Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Comparativo de kernels sobre predicción de oferta de fuentes alternativas de energía

Trabajo Fin de Máster

Presentado por: Mora Paz, Héctor Andrés

Directora: Mancera Valets, Laura

Codirector: Peluffo Ordoñez, Diego Hernan

Ciudad: San Juan de Pasto
Fecha: 10 de octubre del 2019

Contenido

Pág.

Lista de figuras

Pág.

Lista de tablas

Pág.

Lista de ecuaciones

Pág.

Resumen

La necesidad de mitigar la crisis del cambio climático y suplir la demanda energética, ha girado la atención hacia las fuentes de energías limpias, como la fotovoltaica. Promoviendo estudios enfocados en la oferta como en (Cabrera, 2016), donde se estructuraron para una superficie geográfica determinada, grandes bases de datos desde imágenes satelitales Landsat y MODIS de NASA, produciendo buenos patrones tanto en Redes Neuronales Artificiales (ANN) como en Máquinas de Soporte Vectorial (SVM). No obstante aún no se ha evaluado el desempeño de estos algoritmos haciendo uso de funciones kernel como las propuestas por Belanche (2015). Para solventar esto en esta investigación se aporta con la adquisición eficiente de datos de entrenamiento desde Big data, especialización de los algoritmos ANN y SVM con kernels acoplados a scikit-learn, marco experimental para sintonización de hiperparámetros y discusión, donde se muestra que las funciones kernel consiguen mejorar los resultados del estado del arte.

Palabras Clave: Función kernel, Redes neuronales artificiales, Maquinas de vectores de soporte, Energía fotovoltaica, Imágenes satelitales.

Abstract

The need to mitigate the climate change crisis and meet the energy demand has turned attention to clean energy sources, such as photovoltaics. Promoting studies focused on supply as in (Cabrera, 2016), where large databases of NASA from Landsat and MODIS satellite images were structured for a given geographical area, producing good patterns in both Artificial Neural Networks (ANN) and Support Vector Machines (SVM). However, the performance of these algorithms has not yet been evaluated using kernel functions such as those proposed by Belanche (2015). To solve this in this research, it is contributed with the efficient acquisition of training data from Big data, specialization of the ANN and SVM algorithms with kernels coupled to scikit-learn, experimental framework for hyperparameter tuning and discussion, where it is shown that the functions kernel manage to improve the results of the state of the art.

Keywords: Kernel function, Artificial neural networks, Support vector machines, Photovoltaic energy, Satellite images.

1. Introducción

1.1 Planteamiento del problema

Para la realización de predicciones, son ampliamente utilizadas técnicas de aprendizaje automático capaces de esculpir los aspectos más relevantes de la realidad mediante conjuntos de datos contextualizados. De este modo, poder modelar artefactos inteligentes de gran impacto, logrando detectar patrones como: comportamiento de grupos sociales (Chicco et al., 2001), identificación de emociones (Alm et al., 2005), diagnóstico de enfermedades (Wolberg et al., 1994), interpretación de lenguaje natural (Tesauro, 1992), predicción de oferta de energía (Cabrera et al., 2016), entre otras.

Existen diversas técnicas para la obtención de estos modelos como las Redes Neuronales Artificiales, del inglés Artificial Neuronal Networks (ANN), y las Máquinas de Soporte Vectorial, del inglés Support Vector Machine (SVM), las cuales han venido ganado terreno al ofrecer alternativas para clasificar o predecir datos con un buen compromiso en escenarios donde no se aprecian relaciones de forma intuitiva (Sanchez, 2015). Estas técnicas se caracterizan por tener amplia versatilidad y flexibilidad debido a que contemplan dentro de sus parámetros a las funciones kernel, las cuales permiten llevar el conjunto de características a un espacio de observación lineal, logrando con ello encontrar patrones con un modelo lineal simple y como consecuencia reducir el error de forma eficiente (Yu & Wang, 2016). No obstante, para encontrar patrones y predicciones mediante estas técnicas de clasificación y predicción, generalmente se recurre a la utilización de un par de funciones kernel como la gaussiana y polinómica, dejando vertientes de configuración de entrenamiento para estos algoritmos (SVM, ANN), sin explorar (Pedregosa et al., 2011a).

Los pronósticos para oferta de energías alternativas podrían deducirse de patrones hallados de los anteriores modelos ya que representan un gran atractivo para los planes de energización, ya que su uso se concentra en la transformación, especialmente en energía eléctrica, cuya demanda se proyecta para el año 2030 en aproximadamente 376 millones de kilovatios hora por cubrir en el mundo, suponiendo una tendencia de demanda constante de 3.130.71 Kwh per cápita (Grupo Banco Mundial, 2019) y un crecimiento demográfico del 15% (Naciones Unidas, 2015). Por supuesto que esta demanda de energía podría cubrirse con energías fósiles como las producidas por el carbón, el petróleo o la energía nuclear, sin embargo, esto conlleva a un incremento de riesgos ambientales y biológicos (Badii, M.H. et al., 2016), debidos a sus emisiones contaminantes que han contribuido con el actual calentamiento global y protagonizado desastres como los evidenciados en el Cairo

(González, 2018), Centralia (Jaime González, 2012), Fukushima (Semana, 2018), Chernóbil (EUROPA PRESS, 2018), entre otros.

En este contexto, en Colombia se han venido desarrollando esfuerzos mediante planes de energización sostenibles (PERS) extendidos por el territorio nacional, para estudiar la demanda de energía por parte de los consumidores y la oferta de energía por parte de fuentes alternativas (F. Pantoja, 2014). Esto alineado por un lado a lo estipulado por el acuerdo de París donde se estableció el compromiso que existe por parte de los países en la lucha contra el cambio climático (Acciona, 2015) y por otro lado, debido a que Colombia no es un país con suficiente reserva de energías fósiles. Específicamente, en el departamento de Nariño el desabastecimiento de petróleo es eminente, produciendo escases de combustible para los medios de transporte y zonas sin fluido eléctrico continuo. Por esta razón se han realizado dos fases de planes de energización rural sostenible PERS en el departamento enfocados en la demanda de energía y la formulación de proyectos, en el que se destacan los proyectos Pacifico Pura Energía y Análisis de Oportunidades Energéticas para el Departamento de Nariño (ALTERNAR). Este último enfocado en la oferta de energía donde se realizaron predicciones de oferta energética aplicando algoritmos de aprendizaje automático para obtención de modelos de energía fotovoltaica (Cabrera et al., 2016), y eólica (Cabrera & Pantoja, 2018). En el estudio de obtención de modelos de energía fotovoltaica se utilizaron varios algoritmos de aprendizaje automático, donde para cada algoritmo se utilizó hiperparámetros por defecto propuestos en la biblioteca rminer del lenguaje R, y se obtuvo que ANN y SVM obtuvieron los mejores resultados. Mientras que en el estudio de energía eólica solo se utilizó el LCMSEQ por sus siglas en inglés (Latent Class Model). Las conclusiones que arrojan estas investigaciones indican que los modelos conseguidos, ofrecen un buen compromiso entre desempeño y rendimiento computacional para la predicción de energía fotovoltaica, no obstante, aún falta explorar los resultados frente a una sintonización de hiperparámetros sistemática e incluyendo el uso de funciones kernel. Por lo tanto, existe la probabilidad de conseguir modelos mejores para la predicción de oferta de fuentes alternativas de energía y específicamente energías limpias como la fotovoltaica.

Teniendo en cuenta los anteriores factores, se ve necesario optar por fuentes alternativas de energía no contaminantes (Fotovoltaica, Eólica, Corrientes marinas) para cubrir la demanda de energía. Para ello es importante contar con proyecciones de concurrencia de energía que permitan planificar y orientar la adquisición de manera eficiente. Una mejora en estos modelos puede ser determinante y por ello puede ser relevante explorar las vertientes que una función kernel puede ofrecer.

De no contar con proyecciones robustas desde el punto de vista energético, se desaprovecharía la eficiencia de adquisición sobre puntos estratégicos que hubieran podido ser predichos para la implementación de macro proyectos, se optaría por receptores inadecuados o calibrados incorrectamente para el flujo de una fuente de energía (Revelo et al., 2015), se atrasaría la movilidad de estaciones cuyo flujo de alimentación haya decaído. Lo que puede traer como consecuencia pérdidas de alimentación de energía, monetarias y mala prestación del servicio. Desde la perspectiva del aprendizaje automático al no tener referencia de un antecedente que perfile una ruta para el preprocesamiento, configuración de hiperparámetros y evaluación de funciones kernel, se incrementaría los tiempos para la consecución de un modelo para la detección de estos patrones y se podría incurrir en soluciones apresuradas guiadas por el azar. Igualmente, aquellos estudios que requieran el uso de funciones kernel, al no tener en cuenta las funciones en estudio, podrían optar por una función kernel que pueda no alcanzar mejoras en el desempeño, ya sea por su formulación o a que sus hiperparámetros se encuentren desafinados.

Por lo tanto, con este estudio se pretende validar la hipótesis de que es posible encontrar mejoras en los modelos de predicciones de fuentes alternativas de energía (fotovoltaica) mediante la inspección del desempeño de los algoritmos SVM y ANN, introduciendo funciones kernel recomendadas por la literatura (Belanche, 2016a). Con ello se dotará a los modelos SVM y ANN de un nuevo conjunto de hiperparámetros propio de cada función kernel, los cuales serán sintonizados sistemáticamente para acercar a los resultados en lo posible a la solución óptima de la función de coste. De manera que se genere un marco de discusión sobre los resultados en diferentes métricas de desempeño sobre cada función kernel, la configuración de sus hiperparámetros y cuyos resultados finales sean útiles para estudios relacionados con el objeto de estudio.

1.2 Formulación del problema

Considerando funcionales algebraicas y trascendentes: ¿Cuáles funciones kernel generan un buen compromiso entre desempeño en la predicción y coste computacional en la identificación de patrones de oferta de energía fotovoltaica mediante SVM Y ANN?

1.3 Justificación

Las funciones kernel dotan a los algoritmos de SVM y ANN de amplia flexibilidad, consiguiendo moldear una solución de predicción en amplias distribuciones de datos (López, 2018). Esta característica hace factible conseguir una mejora en los modelos para la

predicción de oferta energética en el departamento de Nariño desarrollados hasta el momento, ya que además de contar con los datos de oferta correspondientes al potencial fotovoltaico y eólico, y disponibilidad de recursos producto de antecedentes de investigación previos (Cabrera et al., 2016), la probabilidad de mejora de estos modelos se sustenta en los resultados obtenidos en experimentos realizados en otros conjuntos de datos, donde se ha evidenciado que funciones kernel como la truncated, triangular y canberra han arrojado mejores resultados que la tradicional gaussiana (Belanche, 2016a).

Por lo tanto con el presente estudio se aspira satisfacer la necesidad de realizar una exploración más exhaustiva en la configuración de algoritmos SVM y ANN, teniendo en cuenta un abanico de funciones kernel más amplio para aproximarse mucho más a un modelo óptimo de aprendizaje automático (Belanche, 2016a). Pretendiendo que se motive a estudios posteriores que estén alineados como referentes de construcción de nuevos modelos predictivos en SVM y ANN, a tener en cuenta las dinámicas que en este estudio se proponen para conseguir sus resultados. Desde luego que considerar un abanico muy amplio de funciones hace que los tiempos para conseguir los modelos de predicción se amplifiquen, es por eso que se enmarcarán las funciones con mejores resultados con la confiabilidad de que han sido sometidas con estricto rigor científico para ser evaluadas, y parametrizadas con cercanía a un óptimo local, sirviendo como apoyo para la construcción de modelos desde un antecedente de partida y así aumentar la probabilidad de disminuir los tiempos de entrenamiento. Además de ello, específicamente en predicción de fuentes de energía limpia, los modelos obtenidos pueden convertirse en punto de referencia para la formulación de proyectos de energización, sintonización de receptores de energía fotovoltaica y contingencia ante desastres ambientales; por supuesto trascendiendo en proyectos de adquisición de energía eléctrica sostenibles en el mediano y largo plazo.

Para que se logre obtener los beneficios enunciados anteriormente, se propone trazar diferentes rutas experimentales para conseguir la mejor función kernel en SVM y ANN, haciendo una comparación de una variedad de funciones kernel, en diferentes tipos de normalización de datos, obteniendo como novedad no solo la mejor función kernel si no un acercamiento a la mejor sintonización de hiperparámetros para cada kernel en estudio, concerniente a oferta de energía obtenida de las imágenes Landsat y MODIS del departamento de Nariño Colombia (Cabrera et al., 2016).

Con las novedades enunciadas anteriormente se aportará a la inteligencia artificial en el área del aprendizaje automático, específicamente en la realización de estudios comparativos de algoritmos, con una ruta de comparación, implementación de funciones kernel en un lenguaje de ciencia de datos como Python para los algoritmos SVM y ANN

aumentando la oferta de funciones kernel a las ofrecidas por bibliotecas de ciencia de datos donde generalmente enmarcan las funciones kernel: lineal, polinómica y gaussiana (Pedregosa et al., 2011a). También se aportará con resultados de las técnicas para afinar hiperparámetros de funcione kernel, modelos entrenados para la obtención de predicciones de oferta energética y visualización de datos.

Finalmente este proyecto se sintoniza con la dinámica de Colombia manifestada en sus planes de energización rural sostenible (Badii, M.H. et al., 2016), siendo Nariño el pionero en desarrollar el macro proyecto para la formulación de planes de energización (F. Pantoja, 2014), desembocando en un proyecto para el análisis de fuentes alternativas de energía (Universidad de Nariño, 2015), en el cual se trabajó con la adquisición de datos mediante la ubicación de 35 estaciones meteorológicas donde se adquirieron datos de radiación solar, viento, temperatura, entre otros y se trabajaron las imágenes satelitales Landsat y MODIS, obteniendo predicciones mediante 13 algoritmos de aprendizaje automático donde los mejores resultados se obtienen para SVM y ANN, mediante validación hold out sobre hiperparámetros por defecto (Cabrera et al., 2016), significando ello que además de tener probabilidad de mejora mediante funciones kernel, se manifiesta la necesidad de refinar los resultados para proporcionar integridad en los modelos predictivos obtenidos.

1.4 Alcances y delimitaciones

Este proyecto se realizará mediante experimentos sobre bases de datos obtenidas de imágenes Landsat y MODIS para el departamento de Nariño (Cabrera et al., 2016), Colombia, cuyo rango temporal va desde los años 2000 a 2015 para Landsat y 2005 a 2015 para MODIS, significando ello que en total se trabajaran con 51.076.512 registros para Landsat y 565.722.468 registros para MODIS.

Los modelos serán entrenados usando Redes Neuronales (ANN) y Máquinas de Soporte Vectorial (SVM), transformando los datos con 3 normalizadores diferentes para cada experimento (l1,l2, MAX) y 7 funciones kernel (RBF,Triangular, Anova, Radial Basic, Rational Quadratic, Canberra y Truncated). Para evaluar los modelos se utilizarán métricas de forecasting y clasificación asociadas al error y al tiempo.

La configuración de los hiperparámetros se realizará utilizando las técnicas de búsqueda en cuadricula, aleatoria y evolutiva por algoritmos genéticos.

1.5 Capitulación

Este documento se estructura de la forma descrita a continuación.

En el presente capítulo, capítulo 1, se describe la problemática y justificación del presente estudio y capitulación.

En el capítulo 2, se enmarca el contexto y estado del arte, para ello en primera instancia se realiza una revisión bibliográfica sobre predicciones de oferta de fuentes alternativas de energía y funciones kernel a saber:

- En el apartado 2.1 se presenta el estado del arte los principales estudios asociados a las energías alternativas limpias, algoritmos SVM y ANN, y funciones kernel.
- En el apartado 2.2 se explica el contexto a saber:
- En el apartado 2.2.1 se explica el funcionamiento de las fuentes de energía alternativa
- En el apartado 2.2.2 se explica groso modo los algoritmos supervisados de clasificación y predicción lineal.
- En el apartado 2.2.3 se detalla el algoritmo SVM.
- En el apartado 2.2.4 se detallan el algoritmo ANN.
- Finalmente en la sección 2.2.5 se profundiza en las funciones kernel.

En el capítulo 3 se aborda los objetivos generales y específicos, la metodología en la que se detallan los aspectos epistemológicos de la investigación y el plan a seguir, en concreto:

- En el apartado 3.1 se presenta el objetivo general.
- En el apartado 3.2 se presentan los objetivos específicos.
- En el apartado 3.3 se presentan la metodología en donde se aborda: El paradigma en el apartado 3.3.1, enfoque en el apartado 3.3.2, tipo de investigación en el apartado 3.3.3, y plan de acción en el apartado 3.3.4.

En el capítulo 4 se aborda el desarrollo de la investigación en contraste con el plan de acción propuesto en el apartado 3.3.4.

2. Contexto y estado del arte

En esta sección, en primera instancia se hace una breve descripción de la bibliografía explorada, destacando los aspectos que rodean al objeto de estudio. En segunda instancia se realiza una conceptualización de los temas que conforman el contexto de esta investigación.

2.1 Estado del arte

Los temas que rodean al objeto de estudio se han explorado en dos vertientes. Por un lado, se realizó una exploración de fuentes bibliográficas relacionadas con fuentes alternativas de energía donde se consultaron artículos relacionados con aspectos conceptuales y las técnicas asociadas para el descubrimiento de patrones en este tipo de datos. La otra vertiente de búsqueda estuvo concentrada en consultar fuentes bibliográficas relacionadas con los algoritmos de aprendizaje automático SVM, ANN y funciones kernel.

2.1.1 Fuentes alternativas de energías limpias

Energía solar. En este apartado se presentan los recursos bibliográficos divididos en aspectos conceptuales y predicciones para la obtención de patrones de energía fotovoltaica.

Aspectos conceptuales. Algunos trabajos que han sido un referente importante en términos de las definiciones y temas asociados a energía solar, son los desarrollados por los autores Suarez & Cervantes (2013) y Gonzales Mario, Cárdenas Victor, & Álvares Ricardo (2019). Dichos trabajos han permitido identificar las variables relevantes y comprender la importancia de realizar limpieza a los datos mediante cotas de irradiación normales predominantes en una región determinada.

Predicciones de oferta de energía solar mediante machine learning. En cuanto a las técnicas para realizar predicciones para estimar la oferta de energía solar se encuentra a (Monger et al., 2016a) donde se analizó el recurso solar en Arizona Estados Unidos mediante un modelo de interpolación geo estadística denominado kriging para generar un modelo de variación de irradiacia solar; (C. Pardo et al., 2010) utiliza un perceptrón multicapa con métodos ensemble tipo bagging para realizar predicciones sobre bases de datos de radiación sola; utiliza Support Vector Machine (SVM) con Firefly (FFA) para predecir la radiación solar global horizontal.

Los artículos (Maldonado et al., 2019) y (Cabrera et al., 2016) Muestran los repositositorios para realizar adquisición de datos como: RetScreen, imágenes de los satélites Lantsat y MODIS. También las herramientas para realizar visualización, tratamiento y validación de datos como: Meteorom, mapa interactivo del IDEAM, el software PVsyst y los lenguajes de programación Matlab, R y Python para la realización de scripts de adquisición, transformación, carga y obtención de patrones. Además de realizar los anteriores aportes, estos artículos se destacan porque muestran aspectos metodológicos, que sirven para determinar la oferta energética y sus pronósticos mediante el uso de técnicas de descubrimiento de patrones en bases de datos (KDD).

Energía eólica. En este apartado se presentan los recursos bibliográficos divididos en aspectos conceptuales y predicciones para la obtención de patrones de energía eólica.

Aspectos conceptuales. En cuanto a energía eólica (Coordinación de Energías Renovables & Dirección Nacional de Promoción, 2008) y (Jaramillo Óscar & Borjas Marco, 2010) explican a grandes rasgos en que consiste la energía eólica, los diferentes usos y trascienden en exponer como ocurre la transformación y adquisición mediante aerogeneradores.

Predicciones de oferta de energía eólica mediante machine learning. En cuanto a las técnicas utilizadas para estimar el recurso eólico: (Luo et al., 2008) utiliza auto correlación espacio temporal, para estimar la potencia generada por aerogeneradores, (Rodman & Meentemeyer, 2006) utiliza MCE por sus siglas Malla Cupular Elíptica para establecer a partir de atributos físicos y ambientales la ubicación más prometedora de estaciones receptoras de viento. (Petrov & Wessling, 2015) usa algoritmos de aprendizaje automático con técnicas de multi-etiqueta o múltiples expertos para modelar predicciones para la ubicación de aerogeneradores en Iowa Estados Unidos. En los Paises Bajos (Yusof et al., 2014) para detectar anomalías en el flujo de viento utilizaron LCMseq por sus siglas del inglés Linear time closed Pattern, con este se buscaron patrones secuenciales significativos de velocidad y dirección del viento simultáneamente y detectar aquellos que salieran de los segmentos normales de la señal.

2.1.2 SVM ANN y funciones kernel

***Algoritmo SVM*.** Los aspectos conceptuales asociados al algoritmo de SVM se tomaron de (Suárez, 2014), (Buitinck et al., 2013a), (Pedregosa et al., 2011a), estos artículos explican en que consiste el algoritmo y su conceptualización matemática, abordando el tema desde los modelos en el espacio lineal, cuasi lineal y no lineal.

Algoritmo ANN. Los aspectos conceptuales asociados al algoritmo ANN se tomaron de (Aldabas-Rubira & Colom, 2015), donde se explica el algoritmo base de las ANN; en (Sharma & Chaurasia, s. f.) Se atiende los aspectos asociados a la arquitectura de ANN denominada kernel perceptron.

***Funciones kernel*.** Los aspectos conceptuales asociados a las funciones kernel se tomaron de (López, 2018) y (Belanche, 2016a) donde se explican los modelos matemáticos de las funciones kernel, la construcción de estas funciones, sus propiedades y funciones kernel más utilizados. (Baudat & Anouar, 2001) muestra como estrategia alterna a una arquitectura multicapa de red neuronal a las funciones kernel, proporciona definiciones complementarias del uso de las funciones kernel tanto para clasificación y regresión, y lista una serie de algoritmos de aprendizaje automático que pueden hacer uso de esta técnica.

Trabajos relacionados. En este apartado se presentan los recursos bibliográficos divididos en primer lugar a trabajos relacionados con los algoritmos SVM, ANN en la obtención de patrones en energía alternativa, en segundo lugar se presentan trabajos relacionados con las funciones kernel y finalmente se enuncian trabajos relacionados con estudios comparativos de algoritmos de aprendizaje automático.

SVM y ANN: Los artículos de (Olatomiwa et al., 2015), (Belaid & Mellit, 2016) muestran que se obtuvo un buen compromiso entre los datos de radiación solar pronosticados y medidos a partir de modelos de SVM al ingresar atributos como la temperatura, luz solar y radiación solar. Ambos estudios coinciden en que estos modelos requieren de pocos parámetros simples para obtener buena precisión. En el artículo (C. Yu et al., 2018) se propone diferentes marcos para la predicción de la oferta eólica utilizando la transformada wavelet como entrada para descomponer el histograma original en segmentos, luego se realiza extracción de características mediante el uso de arquitecturas de ANN, finalmente se toma a SVM como método predictor.

Funciones Kernel. En (Peluffo-Ordóñez et al., 2015) se muestra el uso de las funciones kernel en reducción de dimensión proponiendo una combinación de funciones kernel haciendo uso del algoritmo PCA. En (Diego H. Peluffo-Ordóñez et al., 2015) muestran un marco generalizado de funciones kernel que mediante la incorporación de una SVM mejoran el rendimiento de kernel PCA. (Mohamed Abuella & Badrul Chowdhury, 2017) utiliza el kernel RBF configurado los hiperparámetros mediante una buscada en cuadricula y compara los resultados en redes neuronales para la obtención de predicciones de oferta de energía solar. En (Byon et al., 2016) desarrolla métodos de aprendizaje adaptativo basados en el truco kernel sobre datos no estacionarios, para ello se utilizó datos asociados a la energía eólica,

se utilizó la regresión ridge como algoritmo de aprendizaje y los kernel lineal, polinomial y RBF.

Estudios comparativos. Los artículos de (Ramírez Quintero Juan Pablo, 2018), (Basante-Villota et al., 2018) son estudios comparativos de funciones kernel que proponen la realización de comparaciones de desempeño de algoritmos, el primero se enfoca en comparar el compromiso entre precisión y rendimiento de funciones kernel en patrones descriptivos de clientes de comercio electrónico, y el segundo propone una comparación entre métodos de reducción de dimensión y sus correspondiente versión kernel mediante métricas de calidad a través de una métrica propuesta denominada como curva RNX.

2.2 Contexto

En esta sección se abordan los aspectos teóricos y conceptuales del objeto de estudio, empezando con una introducción a las fuentes alternativas de energías limpias, profundizando en las energías fotovoltaica y eólica. Seguido a lo anterior, se hace una descripción de los algoritmos de aprendizaje automático para clasificación y predicción en distribuciones en espacios lineales, se profundiza en SVN, ANN, en este apartado se pone en evidencia la necesidad de optar por técnicas que aprovechen las características de estos algoritmos en espacios no lineales característicos en escenarios reales y se finaliza profundizando en las funciones kernel como alternativa de solución a la necesidad planteada anteriormente.

2.2.1 Fuentes alternativas de energía limpias

Las fuentes alternativas de energía hacen referencia a aquellas fuentes no tradicionales como las hídricas, o las provistas por recursos fósiles como petróleo y el carbón. Dentro de estas energías no convencionales, se encuentran las energías limpias como la eólica y fotovoltaica, las cuales producen poca contaminación, e impactos ambientales bajos.

Este cambio vertiginoso de las condiciones climáticas desemboca en la preocupación por buscar otras fuentes de generación de energía, y por lo tanto buscar patrones que permitan determinar las mejores zonas geográficas donde se hace más viable el aprovechamiento del potencial energético solar y eólico.

En las siguientes secciones se explica a grandes rasgos en que consiste la energía fotovoltaica y eólica y los estudios de predicción realizado en el departamento de Nariño Colombia.

Energía fotovoltaica. La energía fotovoltaica es aquella energía que se obtiene de aprovechar la radiación producida por el sol para ser transformada en energía eléctrica. Esta energía se percibe a través de paneles solares (ver Figura 1) conformados por celdas cristalinas, que aprovechan la incidencia de la luz y la convierten en energía eléctrica usando el efecto fotoeléctrico (Suarez & Cervantes, 2013), esta energía es aprovechada almacenando la energía eléctrica en baterías que pueden ser conectadas a un inversor para ser inyectada a la red de distribución eléctrica (Gonzales Mario et al., 2019).

Figura 1. Paneles solares percibiendo la radiación del sol

Teniendo en cuenta que energía del sol arroja a la tierra es 4000 veces más energía que la que se consume en todo el planeta (Suarez & Cervantes, 2013), es determinante hacer uso de ella, ya que es una fuente de energía limpia, atractiva y capaz de competir con las energías tradicionales.

Para que esta competitividad tenga lugar es necesario colocar receptores fotovoltaico en lugares geográficos que beneficien la adquisición de energía solar, es por eso que en el proyecto análisis de oportunidades energéticas para el departamento de Nariño (ALTERNAR), se realizaron diversos estudios para el análisis de la oferta de esta energía, para ello se utilizaron las imágenes Landsat y MODIS, de las cuales se extrajeron modelos de predicción para la generación de proyecciones de oferta (Cabrera et al., 2016). En la Figura 2 se observar la mejora en la resolución para el estudio realizado en el proyecto ALTERNAR en comparación con el mapa ofrecido por El Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM). Además, en el mapa más a la derecha de la Figura 2 se observa una zona mucho más amplia de oferta energética fotovoltaica.

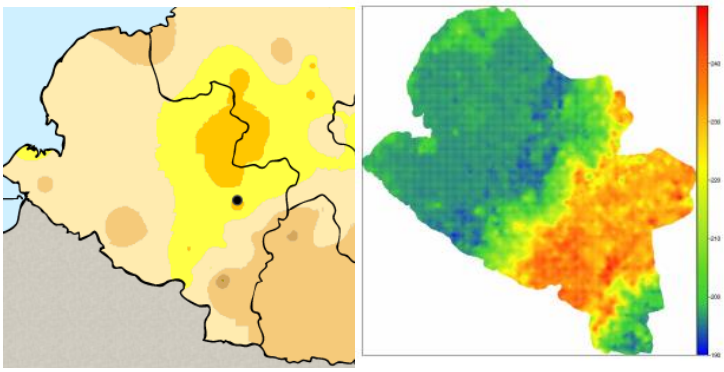


Figura 2. Mapa de oferta de energía para el departamento de Nariño, IDEAM (izquierda), ALTERNAR (derecha).

El proyecto ALTERNAR también realizo estudios similares para el estudio de energía eólica la cual se detalla a continuación.

Energía eólica: Esta energía ha sido ampliamente utilizada a lo largo de la historia de la humanidad para desplazarse de un lugar a otro mediante embarcaciones como los barcos veleros, para uso agrícola para moler granos con los molinos de viento, y en los años ochenta se aprovecha esta energía para la distribución de energía eléctrica de forma industrial. Esta energía es consecuencia de la energía que el sol está irradiando constantemente a la tierra (Jaramillo & Borjas, 2010).

La transformación de esta energía en energía eléctrica se realiza mediante aerogeneradores, los cuales convierten la energía cinética del viento mediante la incidencia de este, sobre las palas del aerogenerador las cuales rotan convirtiendo la velocidad lineal incidente, en velocidad angular. Esta rotación pasa del eje rápido al eje lento el cuál hace rotar la dinamo que transforma esta rotación en energía eléctrica (Coordinación de Energías Renovables & Dirección Nacional de Promoción, 2008). En la Figura 3 puede observarse un esquema de los elementos más destacados de un aerogenerador. La energía es almacenada en baterías de inductores en forma de corriente continua DC, esta suele pasar a la red de servicio eléctrico en forma de corriente alterna AC mediante la utilización de un transformador o inversor.

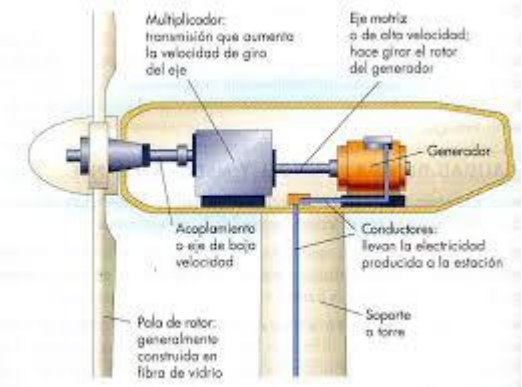


Figura 3. Esquema de un aerogenerador

Para examinar la oferta de este recurso eólico en el departamento de Nariño Colombia, en el proyecto ALTERNAR se analizaron los registros de la base de datos provistos por Vaisaja Inc., y se obtuvieron predicciones mediante el algoritmo LCMseq, los resultado obtenidos se visualizan en la Figura 4.

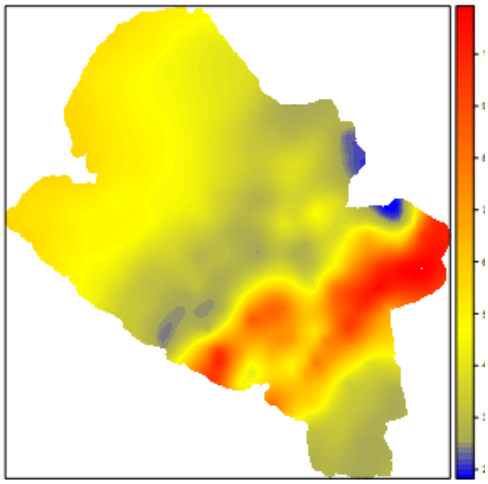


Figura 4. Patrones de velocidad del viento general a 50m

Como puede observarse se logró obtener un modelo de obtención de patrones, no obstante en el artículo (Cabrera & Pantoja, 2018) producido en el proyecto ALTERNAR, no se observan las métricas de calidad obtenidas, sin embargo se destaca la forma de obtener los datos y hace notar que es necesario trabajar la información con algoritmos para tratar series no estacionarias.

En la siguiente sección se detallan algunos algoritmos para la detección de patrones profundizando en los algoritmos SVM, ANN y funciones kernel.

2.2.2 Identificación de patrones mediante modelos lineales

La detección de patrones es el proceso de encontrar en un conjunto de datos caóticos modelos capaces de generalizar el comportamiento de los datos para la obtención de clasificaciones, predicciones o detección de anomalías. Para la obtención de estos patrones se recurre a la sinergia del conocimiento provista por varias ciencias como las matemáticas, estadística, probabilidad, computación entre otras. Actualmente estás técnicas están englobadas en una rama denominada aprendizaje automático, divididas en técnicas de aprendizaje supervisado, no supervisado y por refuerzo. Este apartado se centrará específicamente en las técnicas de aprendizaje supervisado para clasificación y predicción mediante modelos lineales.

Clasificadores lineales: Un clasificador lineal es aquel capaz de encontrar la clase ($\boldsymbol{y}$) discreta a la que pertenece un conjunto de datos basada en una combinación lineal de sus atributos ($\boldsymbol{X}$) como se muestra en la Figura 5.

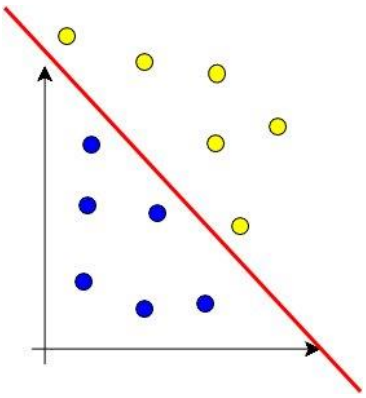


Figura 5. Ejemplo de clasificador lineal

Como se observa en la Figura 5 el clasificador lineal ha trazado un separador en este caso una línea que permite deducir a que clase pertenece (puntos amarillos o azules) así un nuevo registro si se posiciona del lado superior de la línea seria clasificado como punto amarillo, de lo contrario como azul. Por lo tanto una definición formal para un clasificador lineal estaría dado por $y = f(\langle \boldsymbol{w}, \boldsymbol{X} \rangle) = f(\sum_i^n w_i * X_i)$.

Donde $\boldsymbol{w}$ es el vector real de coeficientes f la función que convierte el producto interior de los vectores $\boldsymbol{w}$ y $\boldsymbol{X}$ en la salida y deseada.

Algunos de los algoritmos de clasificación lineal más utilizados para encontrar estos patrones de clasificación se encuentran: Análisis de discriminante lineal, clasificadora de

Bayes lineal, regresión logística, perceptron (ANN), máquinas de soporte vectorial SVM, entre otros.

Predictores Lineales: Un predictor lineal es un algoritmo que se utiliza para establecer al igual que en los clasificadores lineales, una relación entre una variable $(\boldsymbol{y})$ y un conjunto de variables productoras $(\boldsymbol{X})$ donde la diferencia radica en que la variable $(\boldsymbol{y})$ es continua. La figura 6 muestra un esquema de estos modelos.

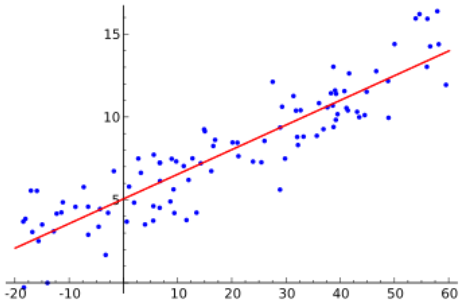


Figura 6. Ejemplo de predictor lineal

En la Figura 6 se observa como el predictor lineal ajusta los datos dispersos a un modelo lineal en este caso una línea recta y su formulación matemática es análoga a la expresada en el apartado anterior.

Entre los algoritmos de predicción lineal más utilizados para encontrar este patrón de predicción se encuentran: método de mínimos cuadrados, Ridge, Lazo, Perceptron (ANN), Maquinas de soporte vectorial (SVM), entre otros.

En las siguientes secciones profundiza en la obtención de estos modelos mediante algoritmos de aprendizaje automático de SVM y ANN.

2.2.3 Máquinas de soporte vectorial (SVM)

Ejemplos separables linealmente. Las máquinas de soporte vectorial son un algoritmo de aprendizaje automático utilizado para problemas de predicción y clasificación (Murillos-Rendón, Peluffo-Ordóñez, Arias-Londoño, & Castellanos-Domínguez, 2013). El algoritmo realiza la separación de clases utilizando el criterio de margen máximo, como se observa en la Figura 7.

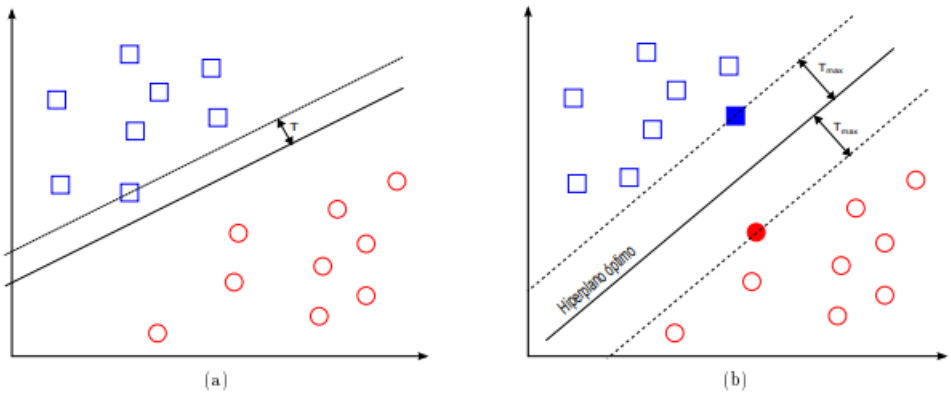


Figura 7. Figura a con margen de separación no optimo, figura b con margen de separación optimo

Para separar un conjunto de datos que es linealmente separable como el mostrado en la Figura 7 existen infinitos hiperplanos (recta en R2) que separan a los dos conjuntos, el criterio del margen máximo especifica que el hiperplano optimo es aquel con margen máximo como el que se observa en b en la Figura 7. La formulación matemática de este criterio se expresa en la Ecuación 1 y Ecuación 2.

$$\boldsymbol{mín}: f(\boldsymbol{w}) = \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle$$

Ecuación 1.Función objetivo SVM para clasificación

$$\boldsymbol{s.a}: y_i(\langle \boldsymbol{w}, \boldsymbol{x_i}\rangle + b) - 1 \geq 0 \;\; i = 1,2,\dots,n$$

Ecuación 2. Restricciones SVM para clasificación

Donde el problema dual a este criterio se resuelve al encontrar la función lagrangiana la cual es resuelta mediante las condiciones Karush-Kuhn-Tucker (KKT), dando como resultado la definición formal en la Ecuación 3 y Ecuación 4.

$$\boldsymbol{max}: L(\alpha) = \sum_{1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x_i}, \boldsymbol{x_j}\rangle$$

Ecuación 3. Lagrangiano al aplicar KKT

$$\boldsymbol{s.a}: \sum_{1}^{n} \alpha_i y_i = 0\,, \alpha_i \geq 0, i = 1,2,\dots,n$$

Ecuación 4. Condiciones del problema dual

Ejemplos cuasi separables linealmente: Los datos en contextos reales poseen generalmente datos ruidosos como se muestra en la Figura 8.

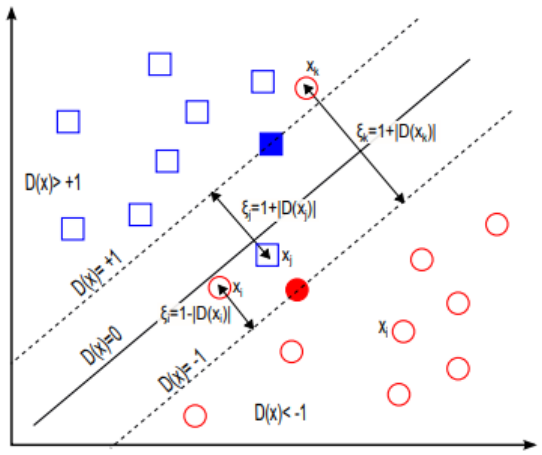


Figura 8. Ejemplo cuasi separable con datos solapados entre clases

En la Figura 8 se observan algunos ejemplos como el x_k, x_j en sectores donde predomina la clase contraria, estos datos son detectados como datos ruidosos, para lidiar con estos ejemplos el modelo descrito en la sección anterior se flexibiliza introduciendo valores de holgura que cuantifiquen la cantidad de ruido que el modelo permitirá captar, por lo tanto la función de coste se transformará como se muestra en la Ecuación 5.

$$\boldsymbol{mín}: f(\boldsymbol{w}, \boldsymbol{\varepsilon}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{n} \boldsymbol{\varepsilon_i}$$

Ecuación 5. Función objetivo SVM para clasificación ejemplos cuasi separables.

Donde C es una constante suficientemente grande para controlar el grado de influencia de las variables de holgura, es decir que para cuando C tienda a infinito se estaría considerando el caso de ejemplos separables, y si C es muy pequeño se estaría admitiendo un número elevado de ejemplos mal clasificados es decir sí C tiende a 0 se considerarían a todos los ejemplos mal clasificados. Teniendo en cuenta la anterior ecuación el modelo para encontrar el hiperplano de separación estaría dado por la Ecuación 6 y Ecuación 7.

$$\boldsymbol{max}: L(\alpha) = \sum_{1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle$$

Ecuación 6. Lagrangiano SVM con condiciones KKT ejemplos cuasi separables

$$\boldsymbol{s.a}: \sum_{1}^{n} \alpha_i y_i = 0\,, 0 \leq \alpha_i \leq C, i = 1,2, \dots, n$$

Ecuación 7. Restricciones SVM KKT ejemplos cuasi separables

SVM para regresión: Este problema busca como se especificó en la sección de predictores lineales una función $f(x) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$, para permitir cierto ruido en los registros de

entrenamiento se introducen las variables de holguras de forma similar como se realizó en el clasificador SVM para ejemplos cuasi separables como se muestra en la Figura 9.

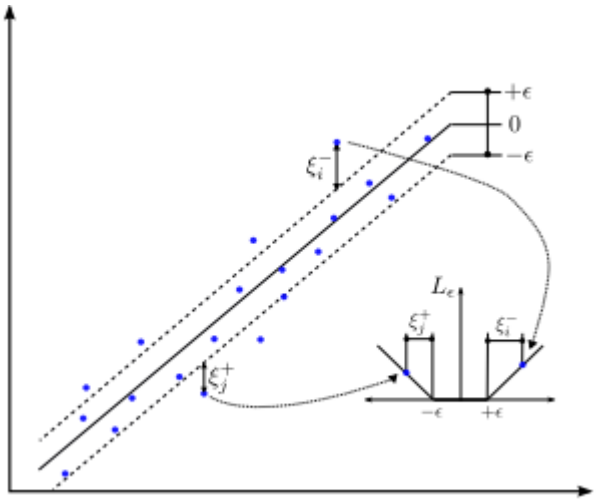


Figura 9. Modelo de SVR relajado mediante un margen máximo a través de variables de holgura.

Utilizando este margen blando se llega a la definición matemática expresada en la Ecuación 8 y Ecuación 9.

$$\boldsymbol{mín}: f(\boldsymbol{w}, \boldsymbol{\varepsilon}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{n} (\boldsymbol{\varepsilon_i^+} + \boldsymbol{\varepsilon_i^-})$$

Ecuación 8. Función objetivo SVM para problemas de predicción

$$\boldsymbol{s.a}: (\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) - y_i - \epsilon - \boldsymbol{\varepsilon_i^+} \leq 0$$

$$y_i - (\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) - \epsilon - \boldsymbol{\varepsilon_i^-} \leq 0$$

$$\boldsymbol{\varepsilon_i^+}, \boldsymbol{\varepsilon_i^-} \geq 0, i = 1,2,3, \dots, n$$

Ecuación 9. Restricciones SVM para problemas de predicción

El problema dual se obtiene de forma análoga al expresado en la sección anterior dando como resultado el problema dual expresado en la Ecuación 10 y Ecuación 11.

$$\boldsymbol{max}: L(\alpha) = \sum_1^n (\alpha_i^- - \alpha_i^+)\, y_i - \epsilon \sum_1^n (\alpha_i^- + \alpha_i^+) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i^- - \alpha_i^+)(\alpha_j^- - \alpha_j^+) \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle$$

Ecuación 10. Lagrangiano SVM condiciones KKT para predicción

$$\boldsymbol{s.a}: \sum_1^n (\alpha_i^- - \alpha_i^+) = 0\,, 0 \leq (\alpha_i^-, \alpha_i^+) \leq C, i = 1,2, \dots, n$$

Ecuación 11. Restricciones SVM KKT para predicción

2.2.4 Redes neuronales ANN

El aprendizaje de una red neuronal ocurre gracias a la sinergia entre neuronas ajustando los pesos de sus conexiones o sinapsis, en la Figura 10 se observa la arquitectura general de una red neuronal.

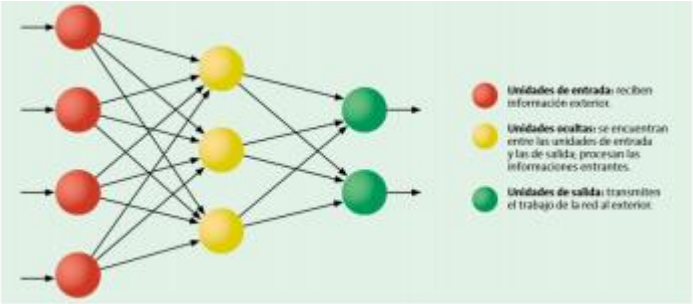


Figura 10. Arquitectura de una red neuronal

En la Figura 11 se puede observar como es la estructura funcional de una neurona en ella se observa que a cada entrada se le realiza una ponderación, y junto a un parámetro de umbral (bias o sesgo) se realiza el sumatorio, al cual se le aplica una función de activación (Sánchez & Paulina, 2015).

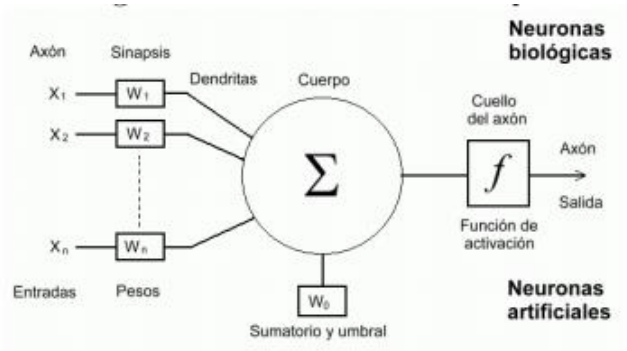
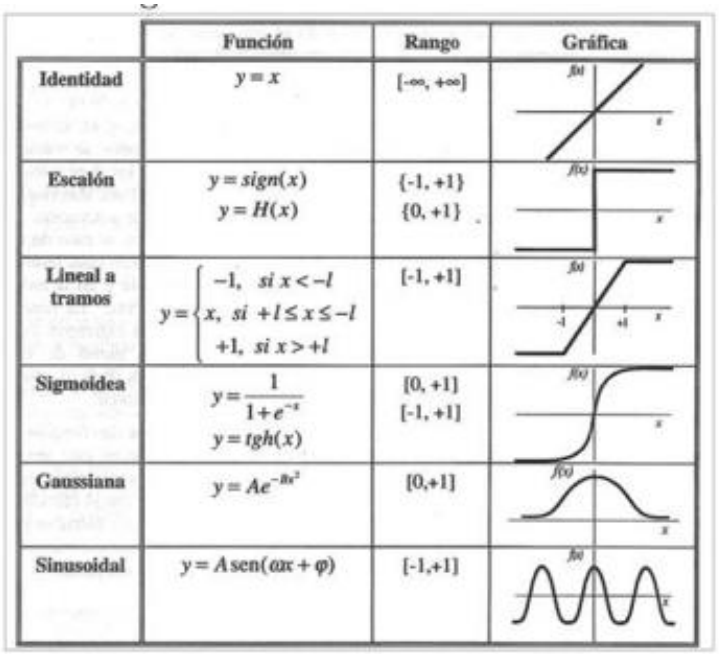


Figura 11. Capas de una red neuronal artificial

Función de activación. Las funciones de activación permiten deformar el espacio y de ese modo encontrar soluciones para aquellos casos que no son linealmente separables o cuyas predicciones se ajusten a una tendencia no lineal. En la Figura 12 se puede apreciar las funciones de activación más comunes y su correspondiente gráfica (Sánchez & Paulina, 2015).



	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & si\ x < -l \\ x, & si\ +l \leq x \leq -l \\ +1, & si\ x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,+1]$	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	$[-1,+1]$	

Figura 12. Funciones de activación más comunes en una red neuronal

Propagación hacia adelante. En este paso la red neuronal calcula las nuevas entradas de cada capa excluyendo la capa inicial para ello se realiza una suma ponderada Z_L (ver Figura 11) equivalente al producto interior de los pesos W_L por la activación de la capa anterior a_{L-1} más el sesgo b_L (ver Ecuación 12), donde L indica el número de la capa, la activación inicial a_0 es el vector de entrada X (ver Ecuación 13), posterior a ello se toma el vector saliente de la suma ponderada y se le aplica la función de activación correspondiente a la neurona como se expresa en la Ecuación 14.

$$Z_L = W_L . a_{L-1} + b_L$$

Ecuación 12. Suma ponderada en ANN

$$a_0 = X$$

Ecuación 13. Activación inicial ANN

$$a_L = f(Z_L)$$

Ecuación 14. Activación capa L ANN

Propagación hacia atrás. En este paso la red neuronal obtiene un error tomado de la diferencia de la salida de la última capa y las etiquetas proporcionadas por el problema a resolver, este error se propaga a cada uno de los parámetros de las neuronas desde la última capa hasta la primera como si se tratara de imputación de errores en una empresa desde el nivel operativo al estratégico (Aldabas-Rubira & Colom, 2015). Para ello es necesario calcular

el valor de coste C las derivadas parciales con respecto a los parámetros W y b de la última capa y con estos determinar la imputación de las capas previas, en concreto esto se expresa formalmente de la Ecuación 15 a la Ecuación 19.

$$C = \frac{1}{n}\sum_{i}(y_i - \hat{y})^2$$

Ecuación 15. Función de coste (ANN)

$$\frac{\partial C}{\partial b_L} = \frac{1}{n}\frac{\partial C}{\partial a_L}\frac{\partial a_L}{\partial z_L} = \delta_L$$

Ecuación 16. Derivada parcial del coste respecto al bias en la capa L (ANN)

$$\frac{\partial C}{\partial W_L} = \delta_L a_{L-1}$$

Ecuación 17. Derivada parcial del coste respecto a los pesos W en la capa L (ANN)

$$\frac{\partial C}{\partial b_{L-1}} = \delta_L W_L \frac{\partial a_{L-1}}{\partial z_{L-1}} = \delta_{L-1}$$

Ecuación 18. Derivada parcial del coste respecto al bias en la capa L-1 (ANN)

$$\frac{\partial C}{\partial W_{L-1}} = \delta_{L-1} a_{L-2}$$

Ecuación 19. Derivada parcial del coste respecto a los pesos W en la capa L-1 (ANN)

Descenso del gradiente. El descenso del gradiente es una técnica que haya la zona de mayor pendiente y mediante pasos llamados ratios encuentra las zonas de mejor coste es decir donde el error tiende a cero. Para redes neuronales este proceso se realiza refrescando los valore b y W como utilizando los valores encontrados en la propagación hacia atrás mediante sus derivadas parciales (ver Ecuación 20 y Ecuación 21).

$$b = b - r\frac{\partial C}{\partial b_L}$$

Ecuación 20. Regla de actualización del bias descenso del gradiente

$$W = W - r\frac{\partial C}{\partial W_L}$$

Ecuación 21. Regla de actualización de los pesos W descenso del gradiente

2.2.5 Funciones Kernel

Una función kernel es una función que dota a los algoritmos de aprendizaje automático como SVM y ANN de la capacidad de llevar a un conjunto de datos N-dimensional a otro espacio M-dimensional mayor a través de una medida de similitud que existe entre dos objetos (Belanche, 2016a), a este espacio M-dimensional se lo conoce como espacio de Hilbert.

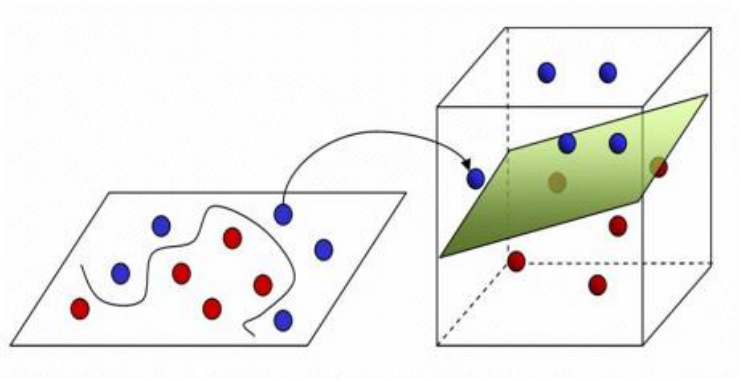


Figura 13.Truco kernel de dos dimensiones a tres

En la Figura 13, se puede observar cómo trabajan las funciones kernel, llevando una distribución de datos de dos dimensiones a tres dimensiones, a este funcionamiento generalmente se lo llama truco kernel. Este truco permite reducir la complejidad de una función que separe las clases de una distribución de datos (Peluffo-Ordóñez et al., 2015), la Figura 13 por ejemplo en dos dimensiones se podría separar mediante funciones no lineales o segmentadas (Elipse, Círculo, Rectángulo), mientras que en tres dimensiones se podría separar mediante una función lineal (Hiperplano) (Baudat & Anouar, 2001).

En términos formales una función kernel define implícitamente una relación $\varphi: \mathrm{X} \rightarrow \mathcal{H}$ a partir de un espacio definido por un vector de características X (Variables independientes) en un espacio $\mathcal{H}$ de Hilbert (llamado espacio de características). El "truco kernel" consiste en realizar la relación y el producto escalar simultáneamente definiendo su función de kernel asociada como se ve en la Ecuación 22.

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}, x, x' \in X$$

Ecuación 22. Definición formal de una función kernel

Donde $\langle ., . \rangle_{\mathcal{H}}$ denota al producto punto en $\mathcal{H}$ (Belanche, 2016a).

Existen diversos kernel empleados comúnmente en bibliotecas de aprendizaje automático como lineal, RBF, polinomial y tangente hiperbólico cuyas definiciones se pueden expresar en la Tabla 1 (Pedregosa et al., 2011a).

Tabla 1 Definición formal funciones kernel

Función kernel	**Ecuación**	**Condición**
Lineal	$k(x, x') = \langle x, x' \rangle$	$x, x \in \mathbb{R}$
RBF	$k(x, x') = \exp(-\sum_{i=1}^{d} \lambda_i (x_i - x_i'))$	$\lambda_i > 0, \beta \in (0,2]$
Polinomial	$k(x, x') = (a\langle x, x' \rangle + 1)^m$	$m \in \mathbb{N}, a > 0$
Tangente hiperbólica	$k(x, x') = \tanh(\, a\langle x, x' \rangle + b)$	$a > 0, b < 0$

Aunque las funciones de la Tabla 1 son funciones de uso común existen muchas más funciones kernel. Estas funciones son utilizadas en algoritmos supervisados para regresión y clasificación; y no supervisadas para detección de anomalías, análisis clúster y extracción de características. Dentro de los algoritmos más destacados se encuentran Procesos gausianos, Spectral clustering, Kernel Linear Discriminant Analysis, Kernel Principal Components Analysis, Kernel Canonical Correlation Analysis, Kernel Independent Component Analysis, SVM, ANN y muchos más (Belanche, 2016a).

En el presente estudio se utilizara las funciones kernel en SVM haciendo uso de la matriz de Gram como se muestra en el sistema de la Ecuación 23.

$$max: L(\alpha) = \sum_1^n \alpha_i - 1/2 \sum_{i=1}^n \alpha_i \alpha_j y_i y_j \, K\langle x_i, x_j \rangle$$

$$s.a: \sum_1^n \alpha_i y_i = 0 \,, 0 \leq \alpha_i \leq C, i = 1,2, \dots, n$$

$$max: L(\alpha) = \sum_1^n (\alpha_i^- - \alpha_i^+)\, y_i - \epsilon \sum_1^n (\alpha_i^- + \alpha_i^+) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i^- - \alpha_i^+)(\alpha_j^- - \alpha_j^+) K\langle x_i, x_j \rangle$$

$$s.a: \sum_1^n (\alpha_i^- - \alpha_i^+) = 0 \,, 0 \leq (\alpha_i^-, \alpha_i^+) \leq C, i = 1,2, \dots, n$$

Ecuación 23. SVM para problema de regresión usando matriz de Gram K

Otra forma de aplicar el truco kernel es mediante una matriz de aproximación como la producida por el método Nyostrem (Buitinck et al., 2013b), método que se usara para ANN, como se muestra en la Ecuación 24.

$$Z_L = W_L . a_{L-1} + b_L$$

$$N_t = NyosTream(kf)$$

$$X = Nt * X$$

$$a_0 = X$$

$$a_L = f(Z_L)$$

$$\hat{y} = predict(X_{batch} * Nt^T)$$

Ecuación 24. ANN usando funciones kernel kf mediante aproximación NyOstrem

3 Objetivos y metodología

3.1 Objetivo general

Determinar de las funciones algebraicas y trascendentes, las funciones kernel que ofrecen los mejores resultados en la predicción de oferta de fuentes alternativas de energía fotovoltaica en los algoritmos de SVM y ANN a partir de un estudio comparativo, haciendo uso de métricas de calidad.

3.2 Objetivos específicos

Procesar los datos de las base de datos Landsat y MODIS del repositorio GeoAlternar, para adquisición eficiente de conjuntos de entrenamiento, exploración y correlación, mediante scripts acoplados en las capas de acceso a datos y aplicación.

Implementar un conjunto de funciones kernel para evaluación del desempeño en los algoritmos SVM y ANN teniendo en cuenta costo computacional y efectividad en la predicción, mediante el acoplamiento de estos algoritmos en la biblioteca scikit-learn.

Clasificar los resultados obtenidos en función de las métricas, técnicas de normalización, función kernel y algoritmos de aprendizaje (SVM, ANN) mediante un marco experimental de sintonización de hiperparámetros para la obtención de modelos suboptimos.

3.3 Metodología

Este estudio se inscribe en el paradigma positivista, enfoque empírico analítico y pertenece a un tipo de investigación aplicada.

3.3.1 Paradigma

La investigación es de tipo positivista, un paradigma que es racional, objetivo y se basa en comprobar hechos y particularidades propias del conocimiento científico (Cuenya y Ruetti, 2010), característico de la esencia disciplinar ingenieril en la que se desarrolla la presente investigación, donde se estudiará el desempeño de los algoritmos SVM y ANN a través del uso de las funciones kernel realizando varias configuraciones de hiperparámetros a través de

una sintonización sistemática, aplicando tales métodos, con datos reales obtenidos de las base de datos Landsat y MODIS de la NASA.

3.3.2 Enfoque

La presente investigación está basada en el enfoque empírico-analítico, el cual permite observar y analizar las causas y efectos de la problemática estudiada, plantear procesos de experimentación y pruebas, que servirán de base para proponer una solución (Gutiérrez, 2014), tal como lo plantea el objetivo del presente proyecto, donde se usará el método de la medición y experimentación, en el cual especifica se realizarán experimentos mediante una serie de rutas para la configuración de datos e hiperparámetros para la obtención de modelos suboptimos en la predicción de fuentes alternativas de energías limpias.

3.3.3 Tipo de investigación

Ejecutar una investigación como la desarrollada en el presente documento, da como resultado la creación de una solución orientada a un problema concreto a través de la aplicación de conocimientos específicos, lo cual clasifica esta investigación como una investigación aplicada, un tipo de investigación que tiene por finalidad producir conocimiento que se aplica directamente al objeto de estudio y que colateralmente, contribuye al incremento del nivel de vida de la sociedad (Lozada, 2014). Esto se verá reflejado en la utilidad que los resultados de la investigación generan en el sector energético y ambiental. De conformidad a la producción de conocimiento el presente estudio contribuye en disminución de búsqueda de sintonización de hiperparámetros para la obtención de predicciones en los algoritmos SVM, ANN en cuanto al conjunto de datos de energía alternativa fotovoltaica.

3.3.4 Plan de acción

Para la realización del presente trabajo de fin de master se realizarán tres fases: En primera instancia la primera fase denominada procesamiento de datos, en la cual se efectúa, la adquisición de los datos de entrenamiento, la exploración de la base de datos GeoAlternar y se examina la calidad de ellos mediante un análisis estadístico descriptivo y correlación. En una segunda fase se realizará la codificación de las funciones kernel y acoplamiento a SVM y ANN, esta fase inicia con la exploración de fuentes bibliográficas que especifiquen la conceptualización matemática de estas funciones, posteriormente se codifican en el lenguaje Python y se acoplan a una biblioteca de aprendizaje automático. Finalmente se clasificaran

los resultados obtenidos para ello: Inicialmente selecciona un conjunto de métricas para evaluar el desempeño de los algoritmos, seguido a ello se realiza una sintonización de hiperparámetros, se reentrena los algoritmo con los hiperparámetros obtenidos, se almacenan los modelos y resultados, se agrupan los resultados para visualizar las configuraciones con el mejor compromiso entre performance y rendimiento computacional, y se visualizan las predicciones y clasificaciones realizadas por los mejores modelo obtenidos, la *Tabla 2* detalla el plan de acción descrito anteriormente.

Tabla 2. Matriz plan de acción.

Fase	Actividad	Entregable	Recursos
Procesamiento de datos	Adquirir los datos de entrenamiento desde las bases de datos Landsat y MODIS de forma eficiente.	Ecuaciones de algebra relacional. Scripts para la adquisición de datos.	Bases de datos. Motor de base de datos PostgreSQL.
	Explorar los datos de oferta de fuentes alternativas de energía fotovoltaica y eólica provenientes del repositorio GeoAlternar.	Scripts para la obtención de diccionario de datos e histogramas de frecuencia. Diccionario de datos. Diagramas de correlación de Person	Artículos relacionados con los datos recolectados. Bases de datos de sensores Landsat y MODIS. Motor de base de datos PostgreSQL. Lenguaje de programación Python.
Codificación de funciones Kernel y acoplamiento	Seleccionar un grupo de funciones kernel recomendadas en la literatura	Matriz de funciones kernel	Fuentes de información.
	Implementar en las funciones kernel en el lenguaje de programación Python	Scripts de implementación de funciones kernel	Clasificación de funciones kernel Fórmulas matemáticas Lenguaje de programación Python
	Acoplar las funciones kernel a una biblioteca de aprendizaje automático que implemente SVM y ANN.	Scripts de acoplamiento de funciones kernel a biblioteca de aprendizaje automático	Documentación API biblioteca de aprendizaje automático Lenguaje de programación Python
Clasificación de resultados	Seleccionar métricas de calidad para el desempeño de las funciones kernel	Matriz de métricas para evaluación de modelos de aprendizaje automático para predicción	Artículos de métricas de evaluación algoritmos de predicción

		Lenguaje de programación Python
Estructurar una ruta de sintonización de hiperparámetros para las funciones kernel en los algoritmos de SVM y ANN	Algoritmo de sintonización de hiperparámetros	Lenguaje de programación Python
	Scripts de sintonización de hiperparámetros	Bases de datos energía fotovoltaica y eólica
Almacenar los resultados de las métricas para cada función kernel	Base de datos de vectores de resultados de entrenamiento de funciones kernel	Lenguaje de programación Python
	Scripts de entrenamiento	Biblioteca de aprendizaje automático
	Graficas de resultados	Motor de base de datos
	Cuadros comparativos	
Agrupar los resultados obtenidos por métrica, función kernel y algoritmo de aprendizaje	Matriz de comparación de funciones kernel	Lenguaje de programación Python
	Artículo científico estudio comparativo de funciones kernel	Base de datos Landsat y MODIS
	Figuras de resultados kernel más destacados	Motor de base de datos
Visualización de predicciones y clasificaciones	Figuras con mapas de segmentación de oferta de energía solar.	Modelos de predicción y clasificación.
	Scripts para la obtención de predicciones, de energía solar.	Biblioteca grafica para la obtención de mapas.

4. Desarrollo de la contribución

En el presente capitulo se describen los resultados obtenidos en esta investigación los productos relacionados en cada sección como: Scripts de limpieza, inspección, adquisición de datos, acoplamiento y experimentos realizados entre otros se encuentran en el repositorio https://github.com/magohector/fkernel.

4.1 Procesamiento de datos

A continuación se describen las bases datos de irradinacia, se muestran los scripts en SQL utilizados para obtener el diccionario de datos, informe de calidad y limpieza.

4.1.1 Descripción de datos

La base de datos de irradiacia ha sido conformada mediante 1362 imágenes satelitales de Landsat 7 y 3912 imágenes de MODIS y 500 muestras de 3TIER, entre las tres bases de datos se cubren los años desde 1999 a 2015. En la Tabla 3 se pueden observar los datos obtenidos en el procesamiento de datos de LANSAT para el periodo 2000 a 2014.

Tabla 3. Datos de procesamiento y limpieza de datos LANSAT

Nombre	Cantidad
Imágenes procesadas	1.321
Datos Totales	51.076.512
Nube caliente	3.731.768
Nube fría	27.827.009
Ambiguo	11.987.340
Datos válidos reflectancia	4.071.185

En la Tabla 4 se presentan los datos obtenidos en el procesamiento de datos MODIS, para el periodo 2005 a 2015.

Tabla 4. Datos de procesamiento y limpieza de datos MODIS

Nombre	Cantidad
Imágenes procesadas	3.912
Datos Totales	565.722.468
Nubes	192.051.992
Nube tipo 1	160.312.600
Nube tipo 2	31.733.392
Datos válidos	373.670.476

Los datos se estructuraron en tablas relacionadas (Figura 14). Para el caso de Landsat se tiene las tablas discarded, date_landsat, reflectance e irradiance. Para el caso de MODIS se tiene las tablas irradiance, cloud, date_MODIS y bands_7.

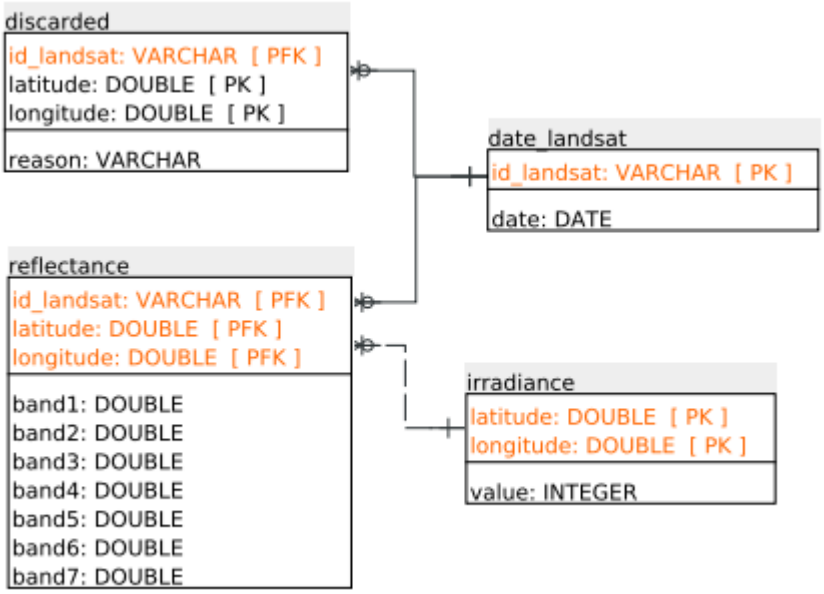


Figura 14. Modelo entidad relación para datos Landsat

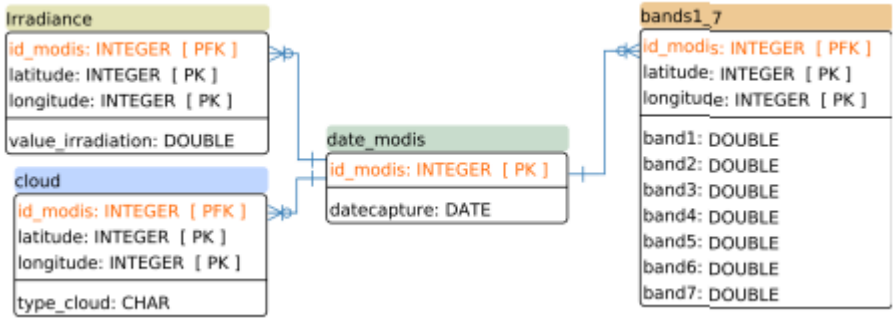


Figura 15. Modelo entidad relación datos MODIS

.

4.1.2 Exploración de base de datos en capa de acceso a datos

Para realizar el procesamiento de los datos se utilizaron scripts escritos en el lenguaje SQL y plPGSQL para obtener el diccionario de datos, análisis de calidad de datos y limpieza. Estos scripts son un producto genérico, lo que significa que servirán para realizar las tareas de preprocesamiento anteriormente enunciadas, tanto para este estudio como para estudios que necesiten realizar la misma tarea, donde los datos residan en un motor de base de datos PostgreSQL.

Tabla 5. Diccionario de datos.

Diccionario de datos

```
SELECT table_name,column_name,data_type  FROM information_schema.columns
       WHERE table_name in (SELECT table_name FROM information_schema.tables
       WHERE table_schema='public' AND table_type='BASE TABLE'
       AND table_name='general')
       ORDER BY table_name, column_name
```

Tabla 6. Extracción de llavez primarias y foráneas

Extracción llaves foráneas y primarias

```
SELECT
           tc.table_schema,
           tc.constraint_name,
           tc.table_name,
         tc.constraint_type,
           kcu.column_name,
           ccu.table_schema  foreign_table_schema,
           ccu.table_name  foreign_table_name,
           ccu.column_name  foreign_column_name

       FROM
           information_schema.table_constraints AS tc
           ,information_schema.key_column_usage AS kcu
           ,information_schema.constraint_column_usage AS ccu

       WHERE
           tc.constraint_name = kcu.constraint_name
             AND tc.table_schema = kcu.table_schema
           AND ccu.constraint_name = tc.constraint_name
             AND ccu.table_schema = tc.table_schema

       AND tc.constraint_type = 'FOREIGN KEY' OR tc.constraint_type='PRIMARY KEY';
       --AND tc.table_name='general';
```

Tabla 7. Extracción de datos numéricos excluidas llavez primarias y foráneas

Extracción de datos numéricos sin llaves

```
SELECT table_name,column_name,data_type FROM information_schema.columns
       WHERE  table_name in (
       SELECT table_name FROM information_schema.tables
       WHERE table_schema='public' AND table_type='BASE TABLE')
       AND column_name not in(
       SELECT
           kcu.column_name

       FROM
           information_schema.table_constraints AS tc
           ,information_schema.key_column_usage AS kcu
           ,information_schema.constraint_column_usage AS ccu

       WHERE
           tc.constraint_name = kcu.constraint_name
             AND tc.table_schema = kcu.table_schema
           AND ccu.constraint_name = tc.constraint_name
             AND ccu.table_schema = tc.table_schema

       AND tc.constraint_type IN ('PRIMARY KEY','FOREIGN KEY'))AND
data_type='numeric'
```

Tabla 8. Cursor para extracción de la moda

Cursor para extracción de modas

```
create or replace function mimoda() returns void as
       $BODY$
       DECLARE
         sql_stmt3 VARCHAR(500);
         reg3 RECORD;
           cur_moda CURSOR FOR
           SELECT table_name,column_name,data_type  FROM
information_schema.columns
           WHERE table_name in (SELECT table_name FROM information_schema.tables
           WHERE table_schema='public' AND table_type='BASE TABLE')
         AND column_name not in(
       SELECT
           kcu.column_name
       FROM
           information_schema.table_constraints AS tc
           ,information_schema.key_column_usage AS kcu
           ,information_schema.constraint_column_usage AS ccu
       WHERE
           tc.constraint_name = kcu.constraint_name
             AND tc.table_schema = kcu.table_schema
           AND ccu.constraint_name = tc.constraint_name
             AND ccu.table_schema = tc.table_schema
       AND tc.constraint_type IN ('PRIMARY KEY','FOREIGN KEY')
       AND tc.table_name='general')
       AND table_name like 'general'
           ORDER BY table_name, column_name;
       BEGIN
         DELETE FROM modas;
         OPEN cur_moda;
         FETCH cur_moda into reg3;
           WHILE(FOUND) LOOP
                  sql_stmt3 := 'INSERT INTO modas (name,moda)
                  SELECT '''||reg3.column_name||''', T FROM
                           (SELECT '||reg3.column_name||', count(*) as c
              FROM general GROUP BY '||reg3.column_name||' HAVING count(*)>10)as T
                          ORDER BY c desc LIMIT 1';
                  EXECUTE   sql_stmt3;
               RAISE NOTICE '%',reg3;
                  FETCH cur_moda into reg3;
           END LOOP;
           CLOSE cur_moda;
           RETURN;
       END
       $BODY$
       LANGUAGE 'plpgsql'
```

Tabla 9. Función PGLSQL para la obtención de frecuencias

Función para obtención de frecuencias

```
create or replace function mifrecuencias() returns void as
       $BODY$
       DECLARE
         sql_stmt VARCHAR(500);
         reg RECORD;
           cur_frecu CURSOR FOR
           SELECT table_name,column_name,data_type  FROM
information_schema.columns
           WHERE table_name in (SELECT table_name FROM information_schema.tables
           WHERE table_schema='public' AND table_type='BASE TABLE') AND
           table_name like 'ventas'
           ORDER BY table_name, column_name;
       BEGIN
```

```
          DELETE FROM frecuencias;
          OPEN cur_frecu;
        FETCH cur_frecu into reg;
            WHILE(FOUND) LOOP
                  sql_stmt := 'INSERT INTO frecuencias(nombre,columna,frecuencia)
                                                  SELECT
'''||reg.column_name||''','||reg.column_name||', count(*) FROM ventas GROUP BY
'||reg.column_name || ' HAVING COUNT(*)>2';
                           EXECUTE   sql_stmt;
                RAISE NOTICE '%',sql_stmt;
                   FETCH cur_frecu into reg;
            END LOOP;
            CLOSE cur_frecu;
            RETURN;
        END
        $BODY$
        LANGUAGE 'plpgsql'
```

Tabla 10. Cursor para obtención de estadísticos de calidad

Cursor para obtención de estadísticos de calidad

```
create or replace function miestadistica() returns void as
        $BODY$
        DECLARE
          sql_stmt2 VARCHAR(1000);
          reg2 RECORD;
            cur_esta CURSOR FOR
            SELECT table_name,column_name,data_type  FROM
information_schema.columns
            WHERE table_name in (SELECT table_name FROM information_schema.tables
            WHERE table_schema='public' AND table_type='BASE TABLE')
          AND column_name not in(
        SELECT
            kcu.column_name
        FROM
            information_schema.table_constraints AS tc
            ,information_schema.key_column_usage AS kcu
            ,information_schema.constraint_column_usage AS ccu
        WHERE
            tc.constraint_name = kcu.constraint_name
              AND tc.table_schema = kcu.table_schema
            AND ccu.constraint_name = tc.constraint_name
              AND ccu.table_schema = tc.table_schema
        AND tc.constraint_type IN ('PRIMARY KEY','FOREIGN KEY')
        AND tc.table_name='general') and data_type = 'double precision'
        AND table_name like 'general'
            ORDER BY table_name, column_name;
        BEGIN
          DELETE FROM estadisticas;
          OPEN cur_esta;
          FETCH cur_esta into reg2;
            WHILE(FOUND) LOOP
                  sql_stmt2 := 'INSERT INTO estadisticas
(nombre,min,max,avg,stddev,ran_min,ran_max, minX, maxX,Q1, medianQ2, Q3, IQR,
count)
                  (SELECT  nombre, count,min,max,avg,stddev_samp,ran_min,ran_max,
                  Q1-1.5*IQR as minX, Q3+1.5*IQR as maxX, Q1, medianQ2, Q3, IQR
                  FROM
                  (SELECT
                nombre, Q3-Q1 as IQR, Q1 as Q1,medianQ2 as medianQ2,
                  Q3 as Q3, min,max,avg,stddev_samp,ran_min,ran_max,count
                  FROM
                  (SELECT '''||reg2.column_name||''' as
nombre,min('||reg2.column_name||'),
                  max('||reg2.column_name||') , AVG('||reg2.column_name||'),
```

```
                stddev_samp('||reg2.column_name||'), AVG('||reg2.column_name||')+
                stddev_samp('||reg2.column_name||') as ran_min,
                AVG('||reg2.column_name||')-stddev_samp('||reg2.column_name||')
as ran_max,
                count('||reg2.column_name||') as count,
                percentile_disc(0.25) within  group (order by
'||reg2.column_name||') as Q1,
              percentile_disc(0.5) within  group (order by
'||reg2.column_name||') as medianQ2,
              percentile_disc(0.75) within  group (order by
'||reg2.column_name||') as Q3
               FROM general )T)T2)';
                        EXECUTE   sql_stmt2;
              RAISE NOTICE '%',reg2;
                FETCH cur_esta into reg2;
          END LOOP;
          CLOSE cur_esta;
          RETURN;
      END
      $BODY$
      LANGUAGE 'plpgsql'
```

Al utilizar los scripts para la obtención de frecuencias por la gran cantidad de información se evidencia cuellos de botella para ejecutar funciones agregadas, por lo tanto primordial realizar una adquisición del modelo de entrenamiento más eficiente como se muestra en la siguiente sección.

4.1.3 Extracción de modelos de base de datos

La extracción de los modelos de base de datos debido a su gran volumen requirió de un procesamiento especial el cual se puede apreciar en la Figura 16.

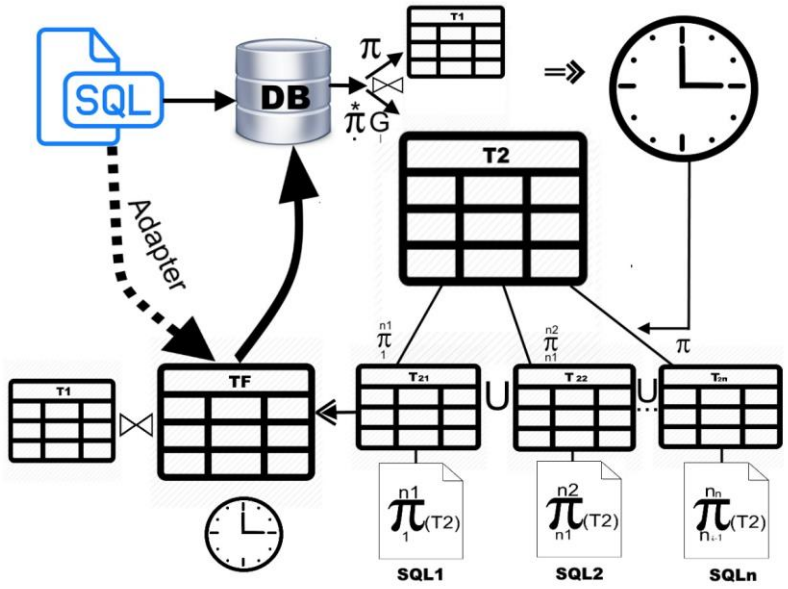


Figura 16 Extracción de modelo BIG DATA

En la Figura 16 se muestra que para la obtención de los modelos es necesario realizar una agrupación y un producto cartesiano, la agrupación corresponde a encontrar el promedio general de cada banda por latitud y longitud y el producto cartesiano al muestreo de estaciones meteorológicas utilizando por (Cabrera & Pantoja, 2018). Estas operaciones especificadas en la Ecuación 25, Ecuación 26 y Ecuación 27 son de un coste computacional elevado y cuya tarea puede durar varias horas o días.

$$\boldsymbol{R_1 = \Upsilon\ pk, avg(x_1)\ x_1, \dots, avg(x_n)\ x_n\ (A)}$$

Ecuación 25. Promedio de las bandas Xi por llave primaria (latitud, longitud) de A (reflectancia)

$$\boldsymbol{R_2 = \pi * (B)}$$

Ecuación 26. Proyección de todos los atributos en B (Radiación)

$$\boldsymbol{R_3 = R1 \bowtie R2}$$

Ecuación 27. Producto cartesiano de R1 con R2

Para mejorar este proceso se fragmento la operación de agrupación en varias agrupaciones, consolidadas en tablas concretas con scripts ejecutados en paralelo dentro del motor de base de datos. Esto debido a que las operaciones de agrupación y producto cartesiano son costosas computacionalmente para grandes cantidades de datos, como el caso de la tabla reflectance con 373.670.476 registros, es necesario realizar una distribución del procesamiento para estas operaciones, por lo tanto la Ecuación 25 se transforma como se indica en la Ecuación 28, Ecuación 29 y Ecuación 30.

$$\boldsymbol{R_{k,m} = \Upsilon_k^{m=\frac{n}{core}}\ pk, \sum x_1\ x_1, \dots, \sum x_n\ x_n, count(*)\ m\ (A)}$$

Ecuación 28. Distribución de la agrupación en m particiones de tamaño m-k

$$\boldsymbol{R_0 = \bigcup_{k=0}^{m} all(R_{i,j})\ |\ \{i = k * m\ , j = t + i\ , t \leq n \rightarrow t = m, t > n \rightarrow t = n\}}$$

Ecuación 29. Unión all, de todas las relaciones de agrupación segmentada

$$\boldsymbol{R_1 = \Upsilon\ pk, \frac{\sum x_1}{\sum m}\ x_1, \dots, \frac{\sum x_n}{\sum m}\ x_n\ (R_0)}$$

Ecuación 30. Obtención de R1 mediante la definición del promedio a través de los datos en R0

Con el anterior principio se estructuró a cada script con el almacenamiento de las sumas y conteo por grupo, posterior a ello se une a cada tabla considerando repeticiones y finalmente se aplica el mismo principio expuesto en la Ecuación 29 consolidada y finalmente se promedia los datos con las sumas de las bandas y conteos. Este procedimiento hace que el resultado de la consulta inicial se obtenga más rápido reduciendo el tiempo de horas o días a minutos.

4.2 Codificación de funciones kernel

La Figura 17 muestra groso modo el algorimo implementado para obtener los insumos que permiten realizar la comparación de kernels utilizando varios conjuntos de datos.

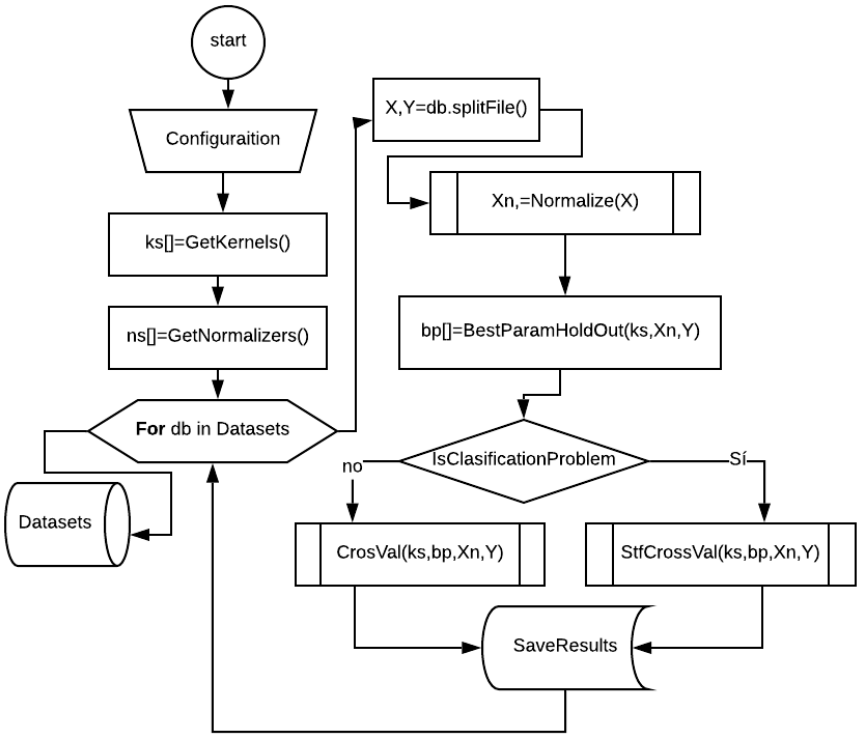


Figura 17. Algoritmo implementado para la comparación de kernel

Inicialmente se realiza una configuración manual donde se especifica: el directorio de entrada, (donde se encuentran los datos a examinar), el directorio de salida (donde se colocaran los resultados de la comparación) y el tipo de problema a tratar (clasificación o regresión). Luego se obtiene el vector de funciones kernel y los normalizadores (funciones que permiten normalizar los datos de entrada X), se leen los conjuntos de datos del directorio de entrada y por cada archivo se realizan los siguientes pasos:

- Se realiza una partición de los datos en variables dependiente e independiente respectivamente.
- Se normalizan los datos y con ellos se escogen los parámetros donde el kernel respectivo haya producido un mejor score (o menor error).
- Con estos parámetros se realiza una validación cruzada con diferentes métricas de error, si el problema es de clasificación se realizará una validación cruzada estratificada, una vez realizada las pruebas se aterrizan los resultados en el directorio de salida.

Con este método inicial se da una aproximación a que función kernel tiene un desempeño mejor no obstante para realizar una aproximación más certera es necesario realizar una sintonización de hiperparámetros con técnicas más sofisticadas. Por ello se ha implementado el siguiente algoritmo enmarcado en la Figura 18.

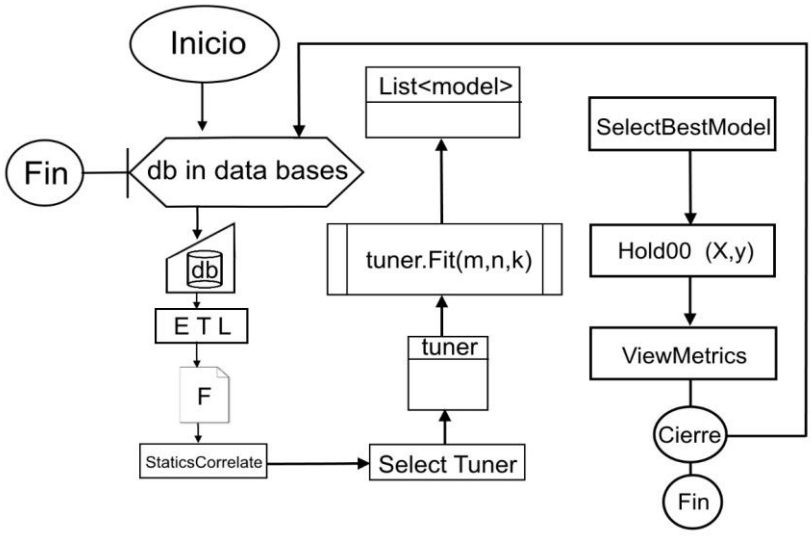


Figura 18 Algoritmo exhaustivo para búsqueda de modelos

En la *Figura 18* se observa que en primera instancia se inicia un ciclo para recorrer cada base de datos, para cada una de ellas se realiza una extracción, transformación y carga de datos, esto finalmente se sintetiza en un archivo el cual es analizado con sus aspectos estadísticos y de correlación, posterior a ello se selecciona un sintonizador por defecto se ha dispuesto a un buscador evolutivo (algoritmo genético), este sintonizador extrae la mejor configuración de parámetros, se selecciona los mejores y se reentrenan nuevamente con un entrenamiento holdout, siguiendo la estrategia establecida por (Pedregosa et al., 2011a) como se muestra en la Figura 19.

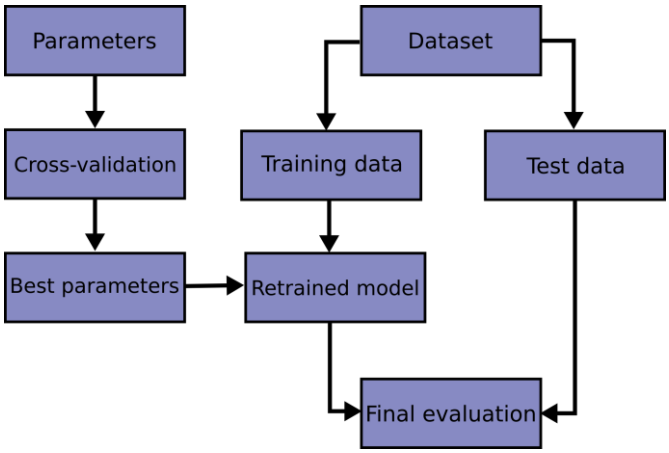
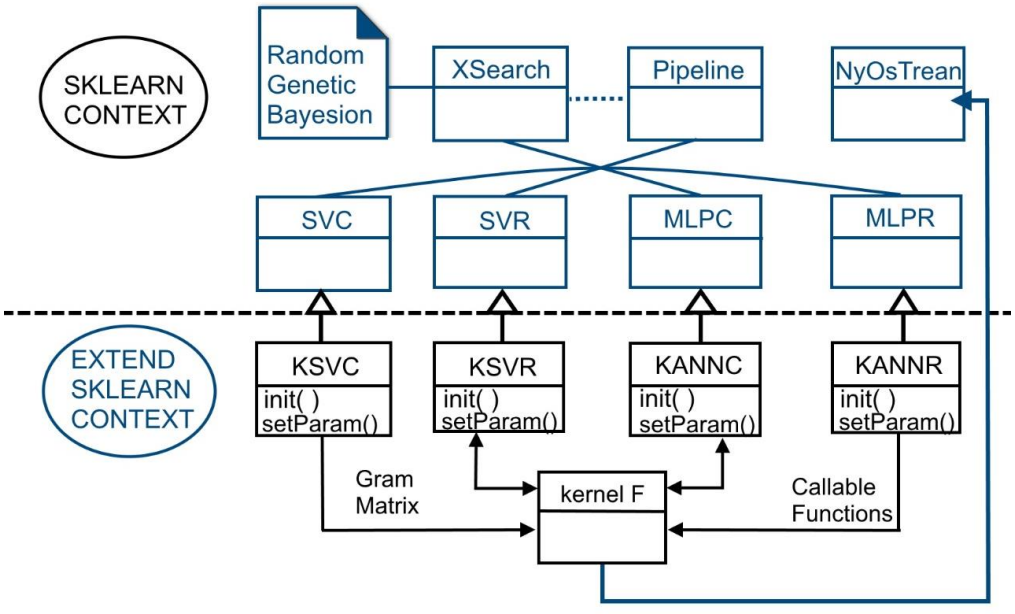


Figura 19 Estrategia de entrenamiento modelo de aprendizaje automático.

A continuación se especifican los procesos más relevantes de los algoritmos establecidos en Figura 17 y Figura 18.

4.2.1 Acoplamiento con biblioteca scikit-learn

Para realizar este acoplamiento se utilizó la técnica de herencia de clases, se estudió los aspectos internos enunciados en el API de la biblioteca scikit-learn, y se inspecciono el código interno de las clases a heredar a saber las clases del algoritmo SVM con sus implementaciones SVC para clasificación y SVR para regresión y las clases del algoritmo MLP Multi Layer Perceptron por sus siglas en inglés. La implementación final se visualiza groso modo en la Figura 20. Allí se observa cómo se especializan a las clases anteriormente enunciadas con sus clases derivadas KSVC, KSVR, KANNC y KANNR respectivamente. El prefijo K corresponde a Kernel. También se observa como estas clases derivadas hacen uso de la clase KernelF la cual implementa las funciones a ser introducidas en estos algoritmos, como se observa este uso es a travez de una matriz Gramm para los algoritmos KSVC y KSVR y callable para los algoritmos KANNC y KANNR. En estos últimos algoritmos se utilizó la aproximación de Nystroem ya que a diferencia de SVM no introducían las funciones kernel dentro de su implementación.



,

Figura 20. Modelo de acoplamiento funciones kernel en scikit-learn

En la Figura 21 se observa ecomo se acopla la clase KSVC a la biblioteca scikit-learn se sigue un procedimiento análogo para KSVR, KANNC y KANNR.

```
class KSVC(SVC):

    def __init__(self, C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated',
                 coef0=0.0, shrinking=True, probability=False,
                 tol=1e-3, cache_size=200, class_weight=None,
                 verbose=False, max_iter=-1, decision_function_shape='ovr',
                 random_state=None,a=2):
        super().__init__(
        kernel=kernel, degree=degree, gamma=gamma,
        coef0=coef0, tol=tol, C=C, shrinking=shrinking,
        probability=probability, cache_size=cache_size,
        class_weight=class_weight, verbose=verbose, max_iter=max_iter,
        decision_function_shape=decision_function_shape,
        random_state=random_state)
        self.a=a
    def fit(self, X, y, sample_weight=None):
        if(self.kernel=="linear" or self.kernel== "poly" or self.kernel== "rbf"):
            super().fit(X,y)
            return self
        else:
            if(self.kernel=="mrbf"):
                self.kernel=mrbf_kernel(gamma=self.gamma)
                super().fit(X,y)
                return self
            if(self.kernel=="tru"):
                self.kernel=truncated_kernel(gamma=self.gamma)
                super().fit(X,y)
                return self
            if(self.kernel=="can"):
                self.kernel=canberra_kernel(gamma=self.gamma)
                super().fit(X,y)
                return self
    def set_params(self, **parameters):
        for parameter, value in parameters.items():
            setattr(self, parameter, value)
        return self
```

Figura 21. Código fuente ejemplo de acoplamiento a biblioteca scikit-learn

Este acoplamiento directo con la biblioteca scikit-learn permite heredar desde modelos sofisticados aspectos de entrenamiento, optimización de algoritmos, acoplamiento con sintonizadores de hiperparámetros mediante técnicas como búsqueda aleatoria, algoritmos genéticos o búsqueda bayesiana como se muestra en la Figura 20.

4.2.2 Obtención de funciones kernel (procedimiento GetKernels)

Este procedimiento obtiene una lista de funciones kernel especificado en la configuración inicial donde se gestiona la lista en la cual se suministran las funciones kernel detallados en la Tabla 11 donde se especifica la función kernel su definición formal matemática y su respectiva codificación en el lenguaje Python.

Tabla 11. Formulación matemática del kernel y correspondiente implementación en Python

Kernel	Definición	Kernel codificado en Python

RBF	$\boldsymbol{k^{RBF}(x,x') = e^{-\sum_{i=1}^{d}\gamma(x_i-x_i')^{\beta}}}$
	 $\boldsymbol{\gamma > 0\,,\beta \in (0,2]}$
		`def mrbf(x,xp,gamma=1,beta=2):`
		`  np.sum(gamma*(x-xp)**beta)`
		`  return np.exp(-sm)`

Triangular	$\boldsymbol{k^{Tri}(x,x')}$
	 $\boldsymbol{= \begin{cases} \|x-x'\| \le a \to 1-\frac{\|x-x'\|}{a} \\ \|x-x'\| > a \to 0 \end{cases}}$
	 $\boldsymbol{a > 0}$
		`def triangle(x,xp,a):`
		`  norm=la.norm(np.subtract(x, xp))`
		`  if norm<=a:`
		`  return 1-norm/a`
		`  return 0`

ANOVA Radial basis	$\boldsymbol{k^{RB}(x,x') = \left(\sum_{i=1}^{d} e^{-\gamma(x_i-x_i')^2}\right)^{m}}$
	 $\boldsymbol{\gamma > 0\,,m \in N}$
		`def radial_basic(x, xp, gamma=1,m=1):`
		`  sm=np.sum(np.exp(-gamma*((x-xp)**2)))`
		`  return sm**m`

Rational quadratic	$\boldsymbol{k^{RQ}(x,x') = 1-\frac{\|x-x'\|^2}{\|x-x'\|^2+a}}$
	 $\boldsymbol{a > 0}$
		`def rquadratic(x,xp,a=0.1):`
		`  norm=la.norm(np.subtract(x, xp))`
		`  return 1-(norm**2)/(norm**2+a)`

Canberra	$\boldsymbol{k^{Can}(x,x') = 1-\frac{1}{d}\sum_{1}^{d}\gamma\frac{\|x_i-x_i'\|}{\|x_i\|+\|x_i'\|}}$
	 $\boldsymbol{\gamma \in (0,1]}$
		`def canberra(x,xp,gamma=0.1):`
		`  d=x.shape[0]`
		`  sm=np.sum(gamma*np.abs(x-y)/(np.abs(x)+np.abs(xp)))`
		`  return 1-sm/d`

Truncated Euclidian	$\boldsymbol{k^{Tru}(x,x') = \frac{1}{d}\sum_{1}^{d}\max\left(0,\frac{\|x_i-x_i'\|}{\gamma}\right)}$
	 $\boldsymbol{\gamma > 0}$
		`def truncated(x,xp,gamma=0.1):`
		`  d=x.shape[0]`
		`  val=1-np.abs(x-xp)/gamma`
		`  sm=np.sum(val[val>0])`
		`  return sm/d`

Para la obtención de este procedimiento se construyó la matriz de Gram correspondiente a la función kernel K (ver Figura 22) tomando como kernel a la formulación

de la Tabla 11, enviando los parámetros correspondientes de cada kernel, y los datos de las variables dependientes e independientes extraídas del conjunto de datos. Cabe resaltar que el envío de una función como parámetro brinda un mecanismo de generalización para la implementación de nuevas funciones kernel.

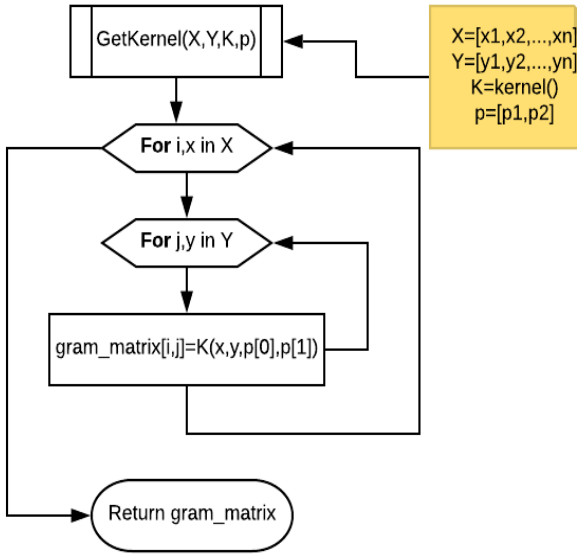


Figura 22. Algoritmo para la obtención de funciones kernel

Con los recursos implementados en la Figura 23 sería suficiente para enviar como parámetro al algoritmo de aprendizaje (SVM, ANN, etc.).

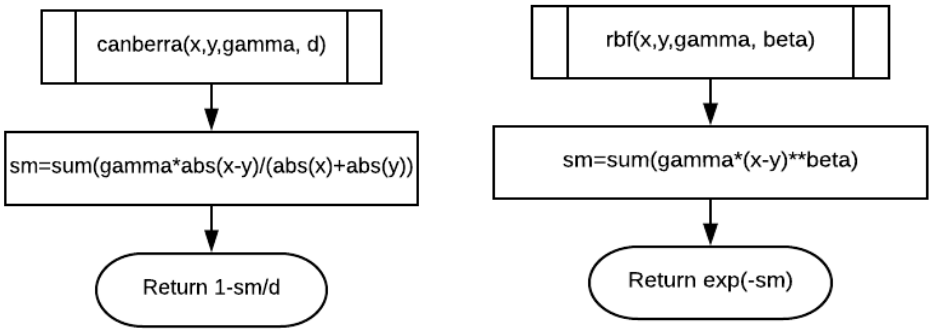


Figura 23. Ejemplo de implementación de un kernel K canberra (izq.), rbf (der)

No obstante por lo general estos algoritmos están diseñados para recibir una función en su inicializador (Constructor) y después enviar los datos de entrenamiento X, Y (funciones callback), para solventar esto es necesario crear un algoritmo que retorne una función como se muestra en la Figura 24.

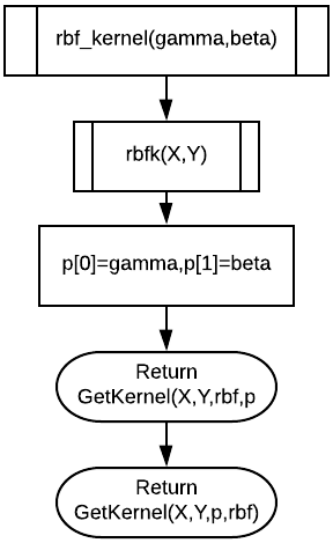


Figura 24. Función rbf kernel con retorno de función rbf mediante procedimiento GetKernel

4.2.3 Obtener normalizador (GetNormalizer)

La función GetNormalizer obtiene un vector normalizado con las técnicas de normalización que se muestran en la Tabla 12. Sea X una matriz de datos con $X = [x^1\ x^2 \dots x^n]$ donde los x^i son los atributos del conjunto de datos, xt^i (Columna) el atributo transformado y xt_i el registro transformado (Fila)

Tabla 12 Formulación matemática de normalizador.

Normalizador	Formula
None	$\boldsymbol{X = X}$
Min Max	$\boldsymbol{xt^i = \frac{x^i - \min(x^i)}{\max(x^i) - \min(x^i)}}$
Normalizer	$\boldsymbol{xt_i = \frac{x_i}{\|x_i\|}}$
Estándar	$\boldsymbol{xt^i = \frac{x^i - mean(x^i)}{std(x^i)}}$

4.2.5 Obtener el mejor parámetro del entrenamiento muestra aleatoria (BestParamHoldOut)

Este procedimiento toma un barrido exponencial para cada parámetro de un kernel específico dentro del rango de definición y realiza una validación Hold Out, dividiendo los datos de entrenamiento y prueba mediante un muestreo aleatorio con una muestra de prueba

del 25%, al finalizar se calcula la puntuación con los datos de prueba y retorna el parámetro con la mejor puntuación.

Tabla 13. Sintonización de hiperparámetros funciones kernel

Kernel	RBF	Tri	Rb		Rq	Tru	Can
Parametro	γ	a	γ	a	γ	γ	
1,00E-06	0,4	0,52	0,42	0,6	0,74	1,00E-11	
1,00E-05	0,62	0,52	0,62	0,6	0,74	1,00E-10	
0,0001	0,78	0,52	0,78	0,6	0,78	1,00E-09	
0,001	0,8	0,52	0,8	0,6	0,86	1,00E-08	
0,01	0,8	0,54	0,82	0,6	0,76	1,00E-07	
0,1	0,78	0,58	0,82	0,6	0,7	1,00E-06	
1	0,62	0,64	0,72	0,6	0,7	1,00E-05	
10	0,54	0,72	0,74	0,6	0,7	0,0001	
100	0,64	0,72	0,66	0,6	0,74	0,001	
1000	0,54	0,72	0,66	0,6	0,84	0,01	
10000	0,54	0,76	0,72	0,6	0,8	0,1	
100000	0,52	0,54	0,72	0,6	0,58	1	
Max	0,8	0,76	0,82	0,6	0,86	0,88	

En la Tabla 13 se muestra el resultado de una validación Hold Out con el algoritmo SVM para clasificación, para las funciones kernel en estudio, tomando el conjunto de datos SVMGMax_4CP sin normalizar (normalizador None).

También puede notarse para qué valor del parámetro se alcanza un score máximo, este parámetro será con el que se procederá a realizar una validación cruzada con diferentes métricas. La Tabla 14 muestra los parámetros que sirven de entrada para la siguiente etapa del algoritmo.

Tabla 14. Mejores parámetros para las funciones kernel después de una validación hold out con el conjunto SVMGMax_4CP

Kernel	parámetro	valor	score
RBF	γ	0,001	0,8
Tri	a	10000	0,76
Rb	γ	0,01	0,82
Rq	a	1,00E-06	0,6
Can	γ	0,01	0,88
Tru	γ	0,001	0,86
Max			0,88

4.2.6 Validación cruzada (CrosVal, StrfCrosVal)

Tomando los mejores parámetros de cada kernel se procede a realizar una validación cruzada para cada kernel, si el algoritmo es de clasificación se realizará una muestreo estratificado para cada prueba de la validación, de lo contrario se realiza un muestreo aleatorio simple. Para las pruebas del presente estudio se tomaron muestras del 20% del total de registros. Del mismo modo en la *Tabla 15* muestra las métricas utilizadas para cada caso (Clasificación, Regresión).

Tabla 15 Métricas utilizadas para comparar los diferentes kernels en problemas de predicción y clasificación

Clasificación	fit time
	score time
	accuracy
	precision macro
	recall macro
	f1 macro
	Confusion matrix
Regresión	fit time
	score time
	explained variance
	neg mean absolute error
	neg mean squared error
	neg median absolute error
	r2
	scatterplot

En la siguiente sección se profundiza en la visualización y clasificación de los resultados obtenidos por el comparador de funciones kernel.

4.3 Clasificación de los resultados

Los resultados del comparador expuesto en la sección anterior almacena los resultados de la prueba reporte y gráficas en el directorio de salida configurado previamente, en la Figura 25 se muestra la estructura de empaquetado que realiza el comparador de funciones kernel.

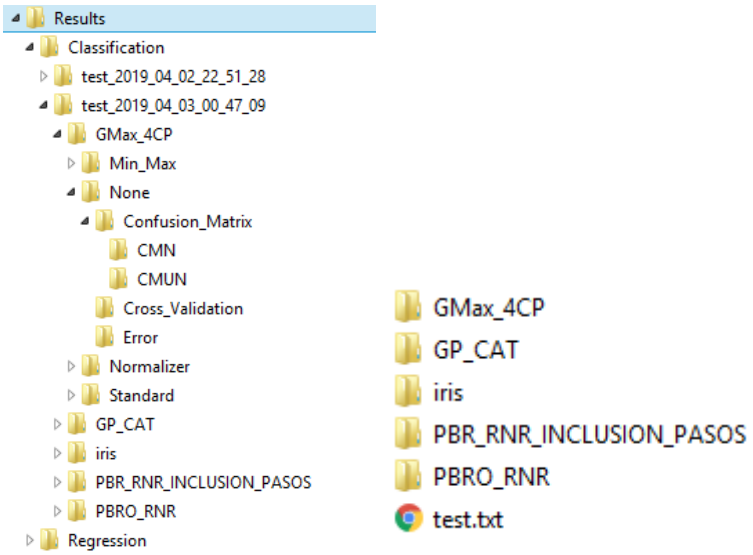


Figura 25. Estructura de empaquetado empleada por el comparador de funciones kernel

La Figura 25 muestra que almacena los diferentes test colocando la fecha en la cual se inició la prueba, dentro de este directorio se muestra un archivo con el nombre test.txt en el cual se almacena el reporte total. Luego a cada conjunto de datos le pertenece un directorio específico con las carpetas de los normalizadores ver *Tabla 12*. Por cada normalizador se podrá encontrar los directorios error, cross_val y confusión_matrix para clasificación y scatter para regresión.

En el directorio error se puede examinar los gráficos de variación del error para cada función kernel con la variación del parámetro respectivo como se ejemplificó en la Tabla 14 (ver Figura 26).

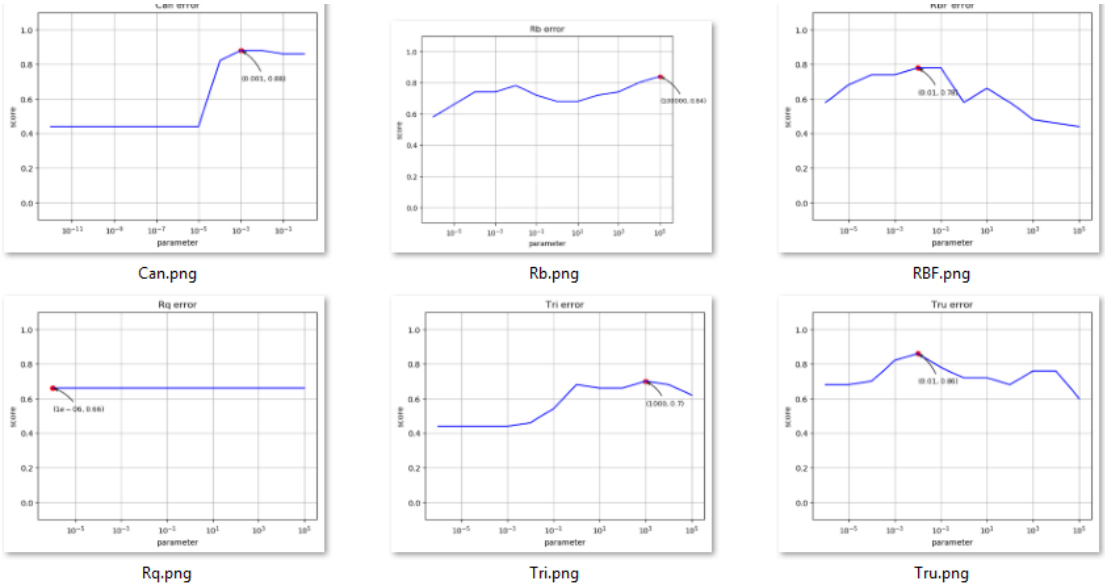


Figura 26. Gráficas de error para SVMGMax_4CP sin normalizar

En el directorio cross_val se encuentran los gráficos de cajas que servirán para evaluar el desempeño de cada kernel frente a cada métrica ver Figura 27.

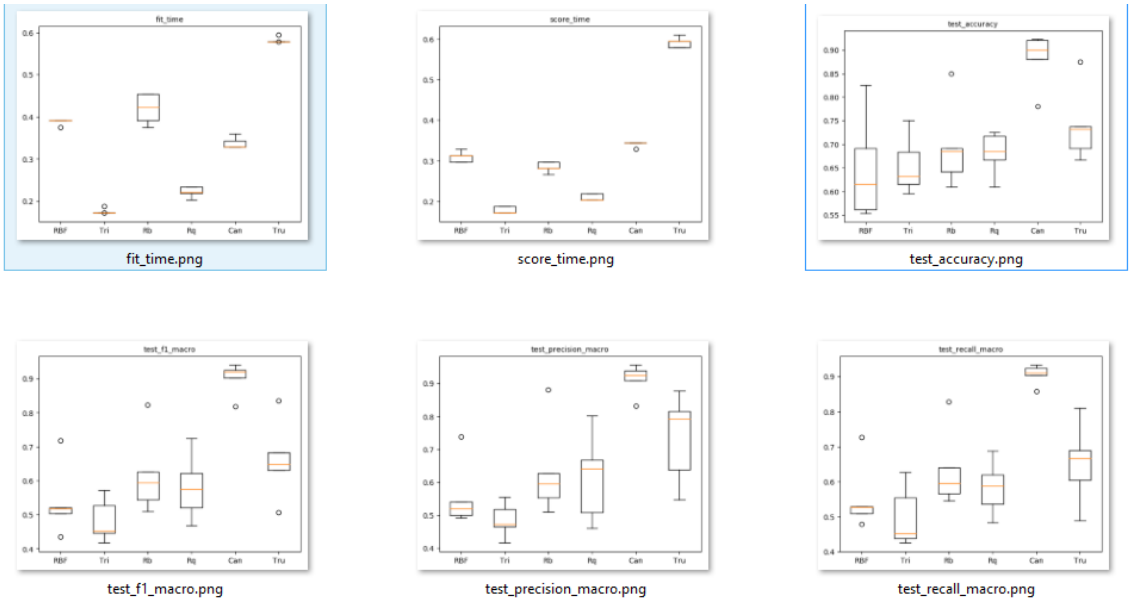


Figura 27. Gráfico de cajas para SVMGMax_4CP sin normalizar

Para el caso de clasificación el comparador genera el directorio confusion_matrix con los directorios CMN y CMUN para generar las matrices de confusión normalizadas y sin normalizar respectivamente (ver

Figura *28* y Figura 29).

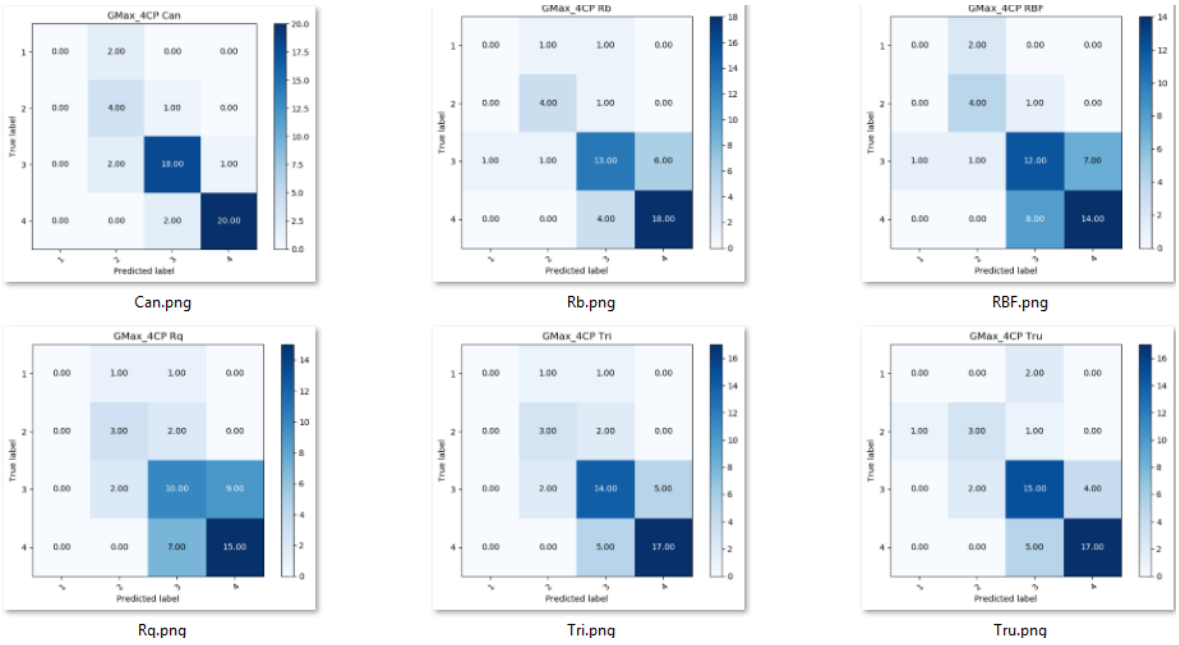


Figura 28. Matrices de confusión directorio CMN SVMGMax_4CP sin normalizar

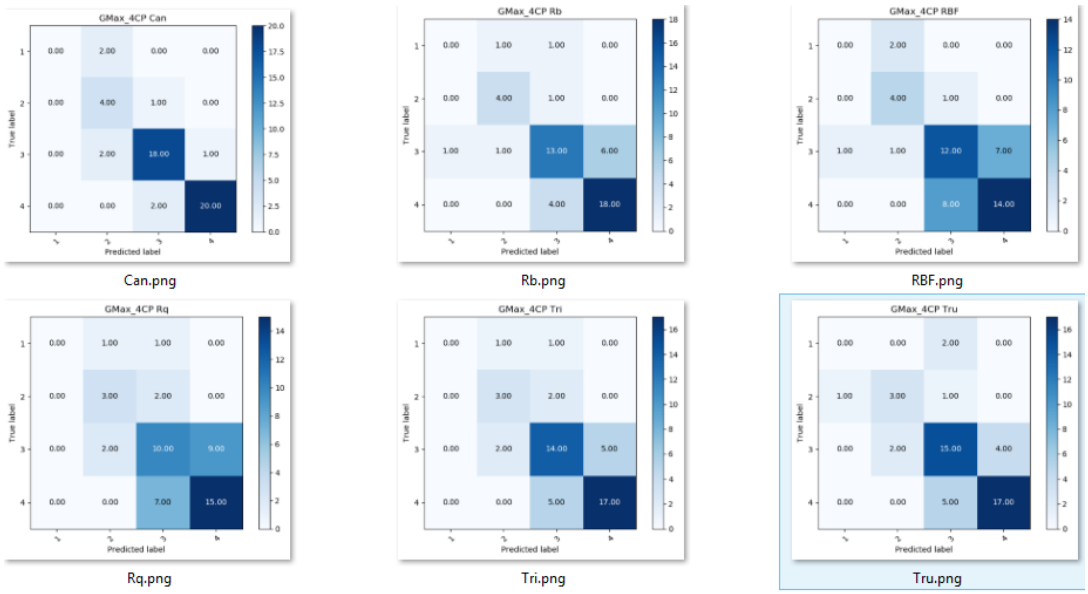


Figura 29. Matrices de confusión directorio CMUN SVMGMax_4CP sin normalizar

Finalmente para el caso de regresión el comparador genera el directorio scatter con los donde se generan los gráficos de dispersión que contrastan el valor predicho y el valor real para cada kernel (ver Figura 30).

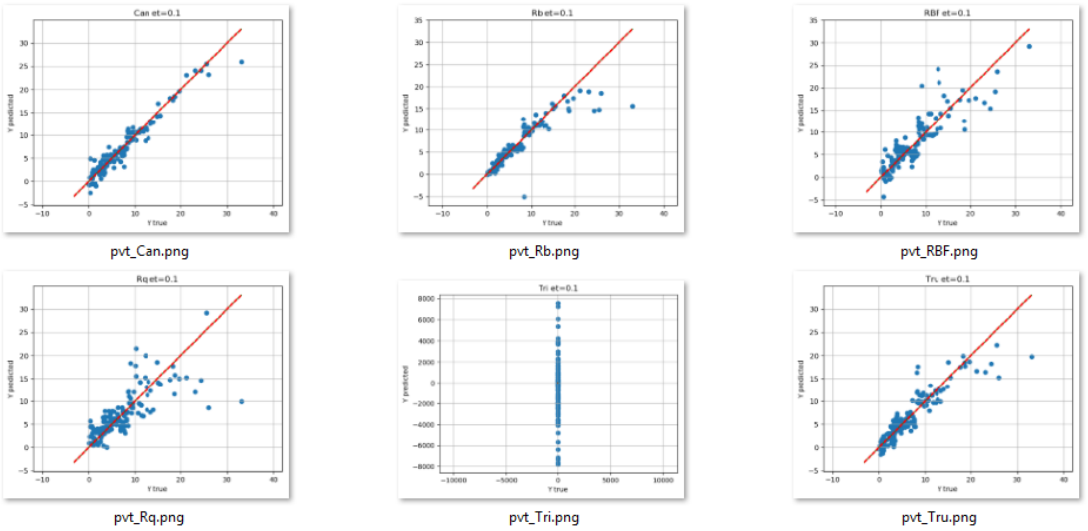


Figura 30. Gráfico de dispersión directorio scatter SVMGMax_4CP sin normalizar

4.3.1 Resultados inspección de funciones kernel en clasificación

A continuación se muestran los resultados de la inspección de las funciones kernel en problemas de clasificación para evaluar el acoplamiento de las funciones kernel y validar lo expuesto en (Belanche, 2016b) donde se afirma que las funciones a considerar en el presente estudio pueden obtener mejores resultados. Los conjuntos de datos a inspeccionar son breastCancer, gmax_4cp, gp_cat, iris, pbr_rnr_inclusion_pasos, pbro_rnr y wine (ver Tabla 16).

Tabla 16. Clasificación de resultados generales

Etiquetas de fila	Máx. de avg	Mín. de std
BreastCancer.csv	**0,941498619**	0,024782929
Can	0,93721203	0,037756544
Rb	0,940079742	0,032363439
RBF	0,891466912	0,047198923
Rq	**0,941498619**	0,031859853
Tri	0,734083882	0,041556042
Tru	0,935753209	0,024782929
GMax_4CP.csv	**0,881113948**	0,028649745
Can	**0,881113948**	0,052608591
Rb	0,705099945	0,075043692
RBF	0,648893599	0,100857699
Rq	0,680716402	0,041454194
Tri	0,7001192	0,028649745
Tru	0,740639709	0,072296653
GP_CAT.csv	**0,995**	0,01
Can	0,954993746	0,018705784
Rb	0,979615385	0,029881422
RBF	0,909981238	0,048998425
Rq	0,869962477	0,036839314
Tri	0,819949969	0,019026758
Tru	**0,995**	0,01
IRIS.csv	**0,96**	0,016329932
Can	0,946666667	0,045215533
Rb	0,953333333	0,026666667
RBF	0,94	0,024944383
Rq	0,953333333	0,033993463
Tri	**0,96**	0,016329932
Tru	**0,96**	0,032659863
PBR_RNR_INCLUSION_PASOS.csv	**0,840662914**	0,043409306
Can	**0,840662914**	0,043409306
Rb	0,831407129	0,067902741
RBF	0,790259537	0,059779688
Rq	0,825778612	0,064629799
Tri	0,800900563	0,045024557
Tru	0,836413383	0,051819483
PBRO_RNR.csv	0,954724828	0,013659023
Wine.csv	**0,96679032**	0,009049149
Can	**0,96679032**	0,031658171
Rb	0,928343301	0,062476499
RBF	0,398989578	0,009049149
Rq	0,398989578	0,009049149
Tri	0,398989578	0,009049149
Tru	0,939304262	0,051799821
Total general	0,995	0,009049149

En la Tabla 16 se han resaltado los mejore score obtenidos por el comparador expuesto en la Figura 17. A continuación se detalla la salida de este comparador para el conjunto de datos GP_CAT, para ver el detalle que el comparador produce en cada base de datos en estudio.

Tabla 17. Resultados Hold out GP_CAT

	None	Min-Max	Normalizer	Standard	Max
RBF	0.0001,1.0	0.001,1.0	0.0001,1.0	0.0001,1.0	1.0
Tri	10,0.98	10,0.92	10,0.98	10,0.98	0.98
Rb	0.0001,1.0	0.001,1.0	0.0001,1.0	0.0001,1.0	1.0
Rq	1e-06,0.92	1e-06,0.8	1e-06,0.78	1e-06,0.86	0.92
Can	1e-12,0.24	1e-12,0.32	1e-12,0.3	1e-12,0.24	0.32
Tru	1,1.0	1,1.0	1,1.0	10,1.0	1.0
Max	1.0	1.0	1.0	1.0	

El comparador toma el mejor score de la tabla anterior y lo reevalúa por validación cruzada para este caso la función kernel truncated. En general para este conjunto se produjeron muy buenos resultados a excepción del kernel Canberra, por lo que se visualizará solo las salidas del normalizador Normalizer ya que produce mejores resultados en cada kernel. La Figura 31 muestra los diagramas de caja de bigotes producidos por el comparador de la Figura 17 para cada métrica de clasificación.

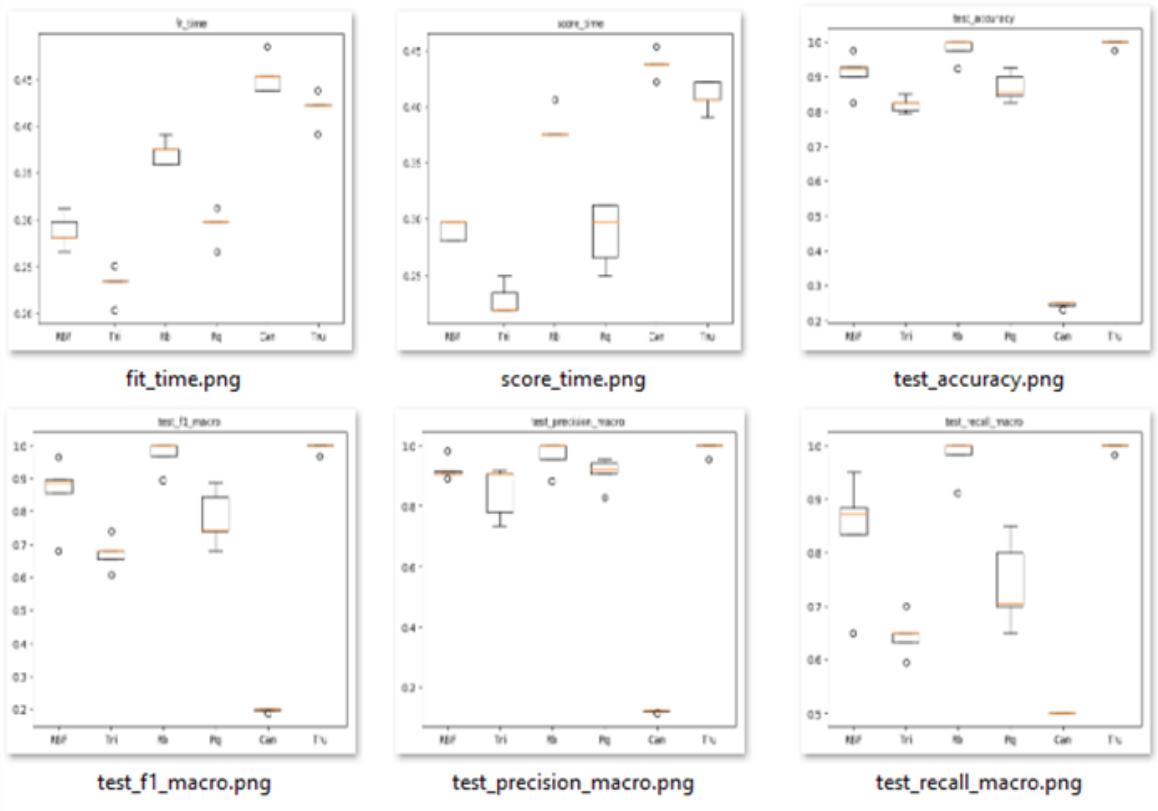


Figura 31.Validación cruzada para cada kernel normalizador Normalizer

Se observa que en las métricas de calidad el Kernel Truncated es el que da mejor resultado para este conjunto de datos. Con una exactitud promedio 0.995000 (±0.010000) [0.985000,1.005000], una precisión promedio macro 0.990909 (0.018182) [0.972727,1.009091], un recall promedio macro (sensibilidad) 0.996667 (±0.006667) [0.990000,1.003333], y un f1 promedio macro de 0.993543 (0.012914) [0.980630,1.006457]. Al ver las matrices de confusión de los dos mejores resultados se observa que el kernel truncated clasifica mejor a las dos clases de este modelo (ver Figura 32).

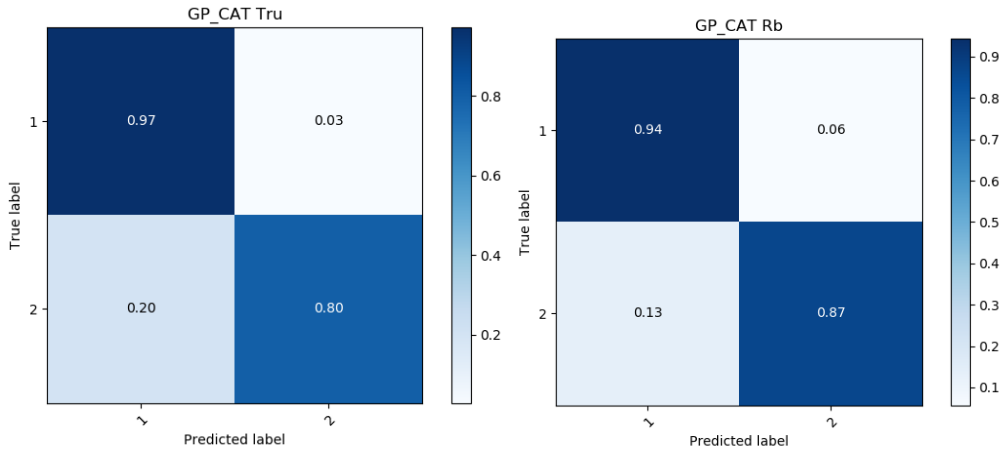


Figura 32. Matriz de confusión kernel truncated y radial basic GP_CAT

Los resultados de cada base datos expuestos anteriormente se sintetizan en la Tabla 18 culminando la primera fase experimental se han resaltado los resultados más altos de los modelos obtenido en cada base de datos.

Tabla 18. Inspección de compromiso funciones kernel en SVM

db	Cn	Rb	rbf	Rq	Tri	Tru
B.C	0,9372	0,9401	0,8915	**0,9415**	0,7341	0,9358
GMC	**0,8811**	0,7051	0,6489	0,6807	0,7001	0,7406
GPC	0,9550	0,9796	0,9100	0,8700	0,8199	**0,9950**
IRIS	0,9467	0,9533	0,9400	0,9533	**0,9600**	**0,9600**
PBRS	**0,8407**	0,8314	0,7903	0,8258	0,8009	0,8364
PBRR	**0,9547**	0,8949	0,8551	0,7504	0,8054	0,8903
Wine	**0,9668**	0,9283	0,3990	0,3990	0,3990	0,9393

Las bases de datos (db) corresponden a bases de datos conocidas breast cáncer (BC), Iris y Wine; y bases de datos financieras, GMC ganancia máxima, GMPC ganancia o pérdida, PRRS perdida con inclusión de pasos, PRRR perdida sin inclusión.

4.3.2 Resultados de regresión en datos Landsat y MODIS

Siguiendo el algoritmo planteado en la Figura 18. Se realizó un entrenamiento con los diferentes buscadores de hiperparámetros búsqueda en cuadricula (grid search), búsqueda aleatoria (random search), búsqueda evolutiva (evolutionary Search). En computación serial para el sistema operativo Windows con procesador Intel core i7 de 2.4Ghz y 8GB de memoria RAM, la búsqueda evolutiva por algoritmos genéticos resultó ser más eficiente. Por lo tanto se escogió a este algoritmo de búsqueda hiperparámetros para realizar los siguientes experimentos configurado con 10 generaciones, tamaño de población igual a 50, gen de mutación 0.1, y tamaño de torneo de 3, para los hiperparámetros de regularización C, kernel, gamma (γ), coef0 (a) y configuraciones para validaciones cruzadas hechas con 5 particiones tomando el 20% de los datos de prueba.

En la *Tabla 19* se observa la salida del primer experimento haciendo una búsqueda por algoritmo y normalizador, descartando a los resultados con r2 menor a 0.7. Se han resaltado los resultados con mejor r2. El experimento se ha realizado para los conjuntos de entrenamiento obtenidos para Landsat y MODIS:

Tabla 19. Resultados experimento 1, sintonización por algoritmo

Db	algoritmo	Normalizador	R2	kernel	γ	a	C
Landsat	**ksvr**	**MM**	**0,93126928**	rq	-	1.0	1000
		Std	0,74816645	rq	-	0.1	10000
		MM	0,91008957	tr	1000.0	-	10000
		Std	0,89532301	rbf	0.001	-	10000
	kannr	MM	0,61504259	rq	-	0.001	10
		MM	0,70187407	hyp	1000.0	-	100
		Std	**0,84376659**	hyp	0.001	-	1000
MODIS	**ksvr**	**MM**	**0,89555281**	rq	-	0.1	1000
		Std	0,89502163	rq	-	1.0	1000
		MM	0,80742541	can	0.1	-	10000
		NM	0,75734473	tri	0.1	-	10000
		Std	0,85986785	rbf	1.0	-	100
	kannr	MM	0,55475843	rq	-	100.0	10000
		MM	0,64854765	tru	1.0	-	1000
		Std	**0,74923367**	hyp	1000.0	-	10

Esta primera fase experimental aunque permite visualizar rápidamente una configuración para la obtención de modelos para los datos en estudio no contempla una amplia gama de combinaciones por kernel, por esta razón se realizó un segundo experimento por kernel y de esa manera tener un ancho de banda de comparación similar para cada

algoritmo. A continuación se muestra la exploración de cada conjunto de datos y los resultados obtenidos en la sintonización de hiperparámetros realizada por kernel.

Exploración de datos sensor Landsat. En la Figura 33 se muestra la matriz de correlación obtenida por el método de Pearson para los datos del sensor Landsat. Se observa que las variables en general tienen una alta correlación. Principalmente entre los datos referentes al espectro electromagnético o bandas. En la Figura 34 se muestra un informe de calidad de datos con estadísticos descriptivos de cada variable.

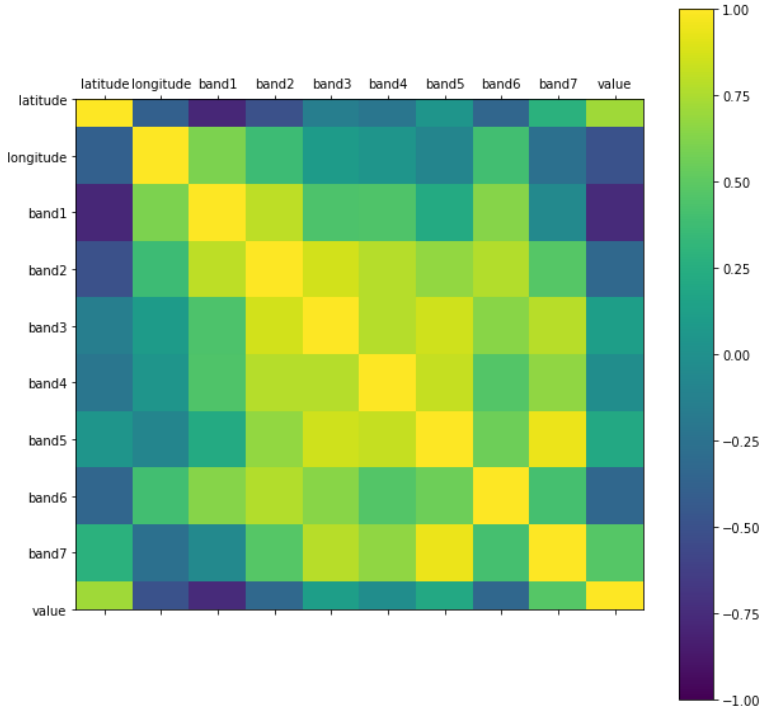


Figura 33. Correlación datos Landsat.

	latitude	longitude	band1	band2	band3	band4	band6	band6	band7	value
count	4.34e+02	434.00	434.00	434.00	4.34e+02	434.00	434.00	434.00	434.00	434.00
mean	-8.67e+06	172098.04	0.10	0.08	5.40e-02	0.29	0.14	295.47	0.05	209.87
std	5.63e+04	52784.26	0.01	0.01	7.79e-03	0.05	0.03	2.30	0.02	17.26
min	-8.79e+06	45000.00	0.06	0.05	3.11e-02	0.16	0.05	286.15	0.02	188.50
25%	-8.71e+06	135000.00	0.09	0.07	4.87e-02	0.25	0.11	294.94	0.04	196.30
50%	-8.67e+06	169650.00	0.10	0.08	5.42e-02	0.29	0.13	296.09	0.05	199.10
75%	-8.62e+06	204750.00	0.11	0.09	5.91e-02	0.33	0.16	296.65	0.06	228.30
max	-8.55e+06	294750.00	0.12	0.10	7.86e-02	0.41	0.26	302.22	0.12	247.30

Figura 34. Análisis estadístico de variables Landsat

Resultado por función kernel sensor Landsat*:* La Tabla 20 muestra los resultados de la búsqueda realizada por el buscador evolutivo al realizar una sintonización por función kernel.

Tabla 20. Sintonización por función kernel datos Landsat

Algoritmo	Normalizador	Kernel	r2	std	t	γ	a	C
KANNR	MM	Rbf	**0,7013**	**0,1039**	**10,6**	**1.0**	**-**	**10000.0**
	St	Hyp	**0,8455**	**0,0261**	**5,8**	**1.0**	**-**	**100.0**
		Can	0,7221	0,0372	4,8	100.0	-	1000.0
		Rbf	0,7170	0,1014	11,7	0.01	-	10.0
KSVR	**MM**	**Rq**	**0,9313**	**0,0150**	**2,3**	-	**1.0**	**1000.0**
		rbf	0,9130	0,0257	0,0	10.0	-	10.0
		Rb	0,9045	0,0127	3,9	0.1	-	1000.0
		Tru	0,8838	0,0169	4,2	100.0	-	10000.0
		can	0,8761	0,0186	4,4	1.0	-	100.0
		hyp	0,8563	0,0167	1,1	0.01	-	10000.0
		Tr	0,7904	0,0306	1,9	1000.0	-	1000.0
		Tru	0,7856	0,0491	3,6	0.01	-	10000.0
	St	**Tr**	**0,9302**	**0,0137**	**2,1**	**100.0**	**-**	**1000.0**
		rbf	0,9079	0,0062	0,5	0.01	-	10000.0
		Rb	0,8993	0,0062	23,5	0.01	-	10000.0
		can	0,8473	0,0117	4,3	10.0	-	1000.0
		hyp	0,8305	0,0194	1,1	0.01	-	100.0
		rq	0,7482	0,0268	2,3	-	0.1	10000.0

En la Figura 35 y Figura 36 se muestran los gráficos de cajas de bigotes para cada normalizador de datos, producto del procesamiento de la búsqueda de hiperparámetros por función kernel para los datos del sensor Landsat.

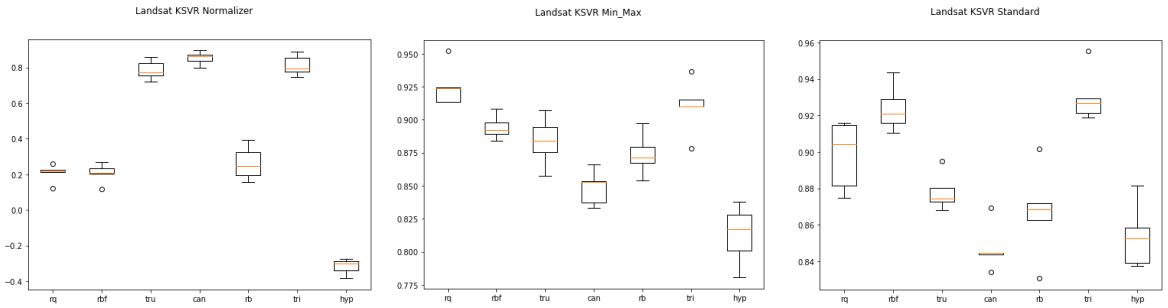


Figura 35 Resultados KSVR sensor Landsat, normalizador Normalizer, Min Max y Standard

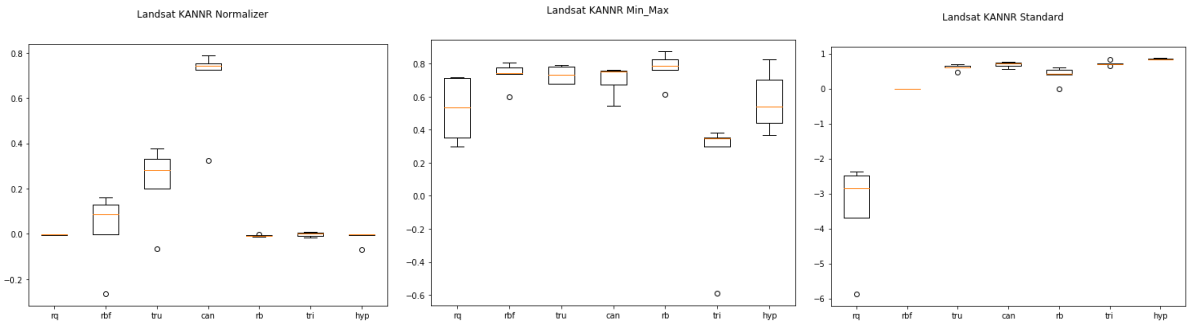


Figura 36 Resultados KANNR sensor Landsat, normalizador Normalizer, Min Max y Standard

Una vez obtenidos los resultados de experimentación se realizó un entrenamiento hold out con mil iteraciones sobre los modelos cuyo rango de confianza (μ±σ) sea superior a 0.9. La Figura 37 y Figura 38 muestran los resultados de este entrenamiento.

	model	normalizer	kernel	sae	mae	rmse	r2
0	KSVR	Min_Max	rquadratic	106.907948	1.228827	2.521034	0.971483
1	KSVR	Min_Max	triangle	263.005449	3.023051	4.941611	0.918701
2	KSVR	Normalizer	can	202.888053	2.332047	4.161511	0.935406
3	KSVR	Standard	triangle	129.161950	1.484620	3.509677	0.954525
4	KANNR	Min_Max	rbf	589.143750	6.771767	9.244043	0.717002
5	KANNR	Min_Max	radial_basic	454.665956	5.226045	7.369464	0.809772
6	KANNR	Normalizer	can	599.178776	6.887112	9.270078	0.689793
7	KANNR	Standard	hyperbolic	477.162513	5.484627	7.112219	0.830680

Figura 37 Resultado final métricas por Holdout sensor Landsat

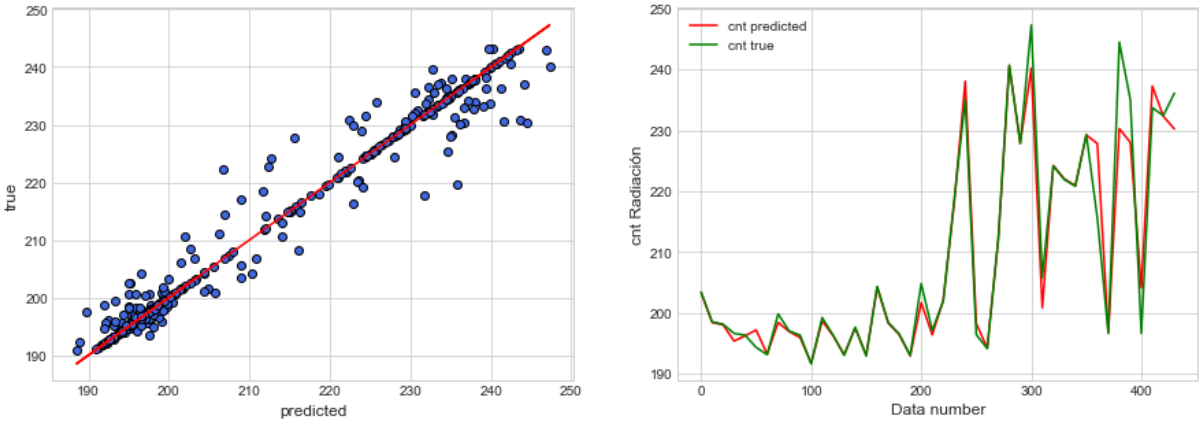


Figura 38. Cruce y traza datos Landsat

Exploración de datos sensor MODIS. En la Figura 39 se muestra la matriz de correlación obtenida por el método de Pearson para los datos del sensor Landsat. Se observa que las variables en general tienen una alta correlación. Principalmente entre los datos

referentes al espectro electromagnético o bandas. En la Figura 40 se muestra un informe de calidad de datos con estadísticos descriptivos de cada variable.

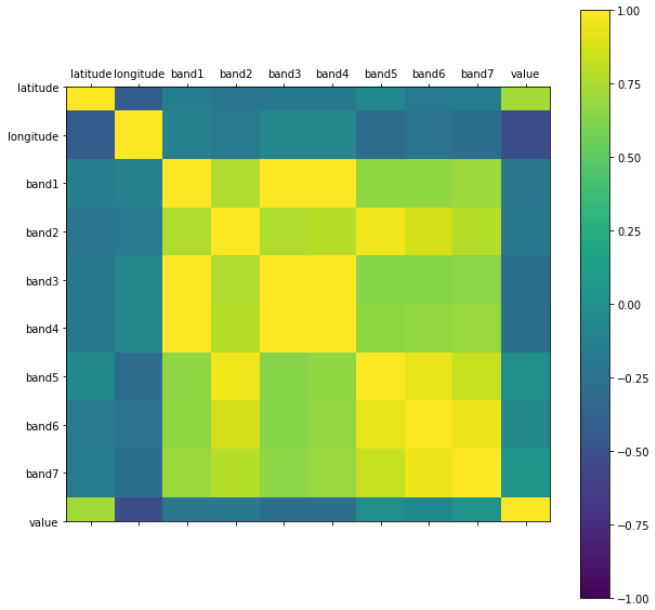


Figura 39 Correlación datos sensor MODIS

	latitude	longitude	band1	band2	band3	band4	band5	band6	band7	value
count	4.45e+02	445.00	445.00	445.00	445.00	445.00	445.00	445.00	445.00	445.00
mean	-8.67e+06	173032.58	3278.96	4504.22	3383.80	3415.47	4397.34	3207.85	1970.02	210.18
std	5.66e+04	53040.04	395.71	349.15	420.19	395.12	291.64	215.47	144.10	17.34
min	-8.79e+06	45000.00	1934.56	3161.65	1897.33	2024.84	3086.36	2409.23	1582.52	188.50
25%	-8.71e+06	135000.00	3115.94	4339.87	3227.32	3267.77	4253.51	3099.44	1890.92	196.40
50%	-8.67e+06	169650.00	3305.41	4540.40	3425.99	3452.63	4431.10	3224.82	1966.49	199.20
75%	-8.62e+06	214650.00	3453.94	4730.22	3593.41	3598.34	4570.24	3345.01	2063.00	228.70
max	-8.55e+06	294750.00	4402.59	5296.90	4513.47	4525.71	5039.51	3700.45	2374.84	248.10

Figura 40. Análisis estadístico de variables MODIS.

Resultado por función kernel sensor MODIS*:* La *Tabla 21* muestra los resultados de la búsqueda realizada por el buscador evolutivo al realizar una sintonización por función kernel.

Tabla 21. Sintonización por función kernel datos MODIS

Modelo	normalizador	kernel	r2	Std	t	γ	a	C
KANNR	MM	rb	**0,7658**	**0,0603**	**8,0**	**10.0**	**-**	**10000.0**
		hyp	0,7547	0,0398	7,1	0.001	-	1000.0
		rb	0,7375	0,0936	8,4	0.1	-	10000.0
KSVR	**MM**	**rbf**	**0,9026**	**0,0082**	**0,1**	**10.0**	**-**	**100.0**
		rq	0,8956	0,0173	2,9	-	0.1	1000.0

	tr	0,8455	0,0264	2,4	10.0	-	100.0
	rb	0,8442	0,0371	3,3	1.0	-	10.0
	tru	0,8187	0,0214	4,0	100.0	-	10000.0
	can	0,8074	0,0268	4,7	1.0	-	1000.0
	hyp	0,7779	0,0332	1,3	0.01	-	10000.0
St	rq	**0,8950**	**0,0097**	**2,6**	**-**	**1.0**	**1000.0**
	rb	0,8826	0,0283	4,2	0.001	-	10000.0
	rbf	0,8599	0,0171	0,1	1.0	-	100.0
	tr	0,8152	0,0300	2,0	10.0	-	10.0
	tru	0,8113	0,0104	4,0	1.0	-	100.0
	can	0,7859	0,0246	4,5	0.01	-	10000.0

En la Figura 41 y Figura 42 se muestran los gráficos de cajas de bigotes para cada normalizador de datos, producto del procesamiento de la búsqueda de hiperparámetros por función kernel para los datos del sensor MODIS.

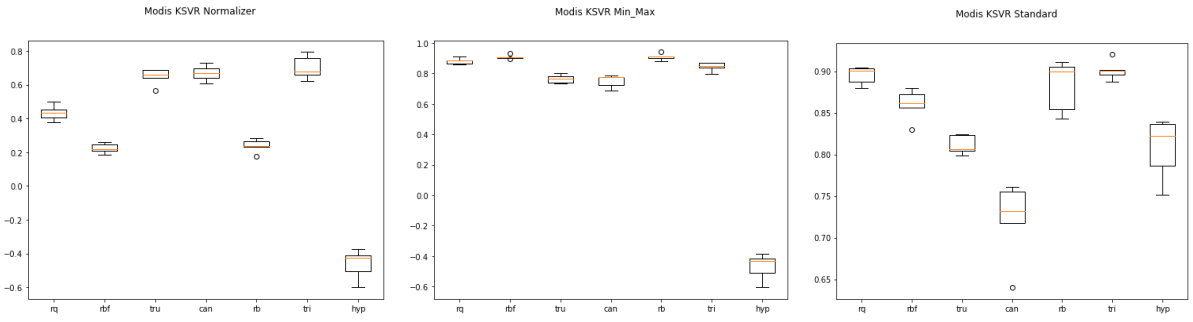


Figura 41 Resultados SVR Sensor MODIS, normalizador Normalizer, Min Max, Standard

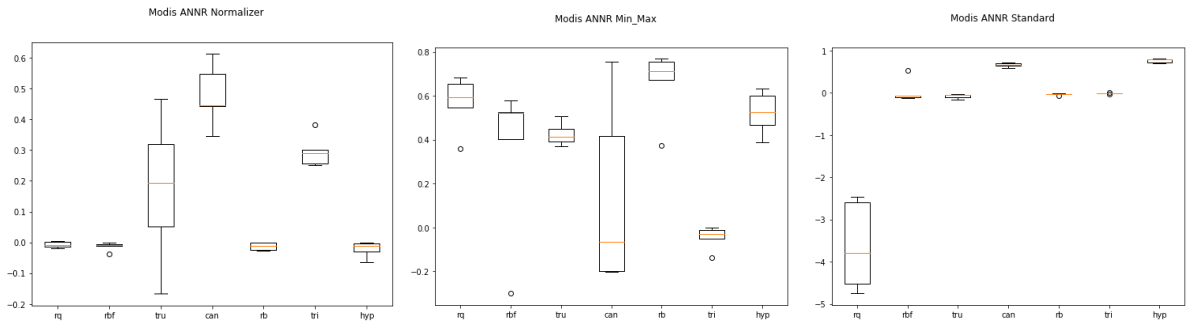


Figura 42 Resultados KANNR sensor MODIS, normalizador Normalizer, Min Max, Standard

Una vez obtenidos los resultados de experimentación se realizó un entrenamiento hold out con mil iteraciones sobre los modelos cuyo rango de confianza (μ±σ) sea superior a 0.87. La Figura 43 y Figura 44 se muestran los resultados de este entrenamiento.

	model	normalizer	kernel	sae	mae	rmse	r2
0	KSVR	Min_Max	rbf	364.687210	4.097609	6.394695	0.865041
1	KSVR	Min_Max	radial_basic	250.598197	2.815710	4.379219	0.933029
2	KSVR	Normalizer	can	405.084245	4.551508	7.427263	0.820131
3	KSVR	Normalizer	tru	480.422018	5.398000	8.872121	0.745402
4	KSVR	Normalizer	triangle	300.886939	3.380752	6.706970	0.847195
5	KSVR	Standard	rquadratic	115.579783	1.298649	2.757646	0.968671
6	KSVR	Standard	triangle	173.691673	1.951592	4.122279	0.941309
7	KANNR	Min_Max	radial_basic	1121.915119	12.605788	14.931880	0.161417
8	KANNR	Standard	hyperbolic	613.987493	6.898736	9.485027	0.687912

Figura 43 Resultado final métricas por Holdout sensor MODIS

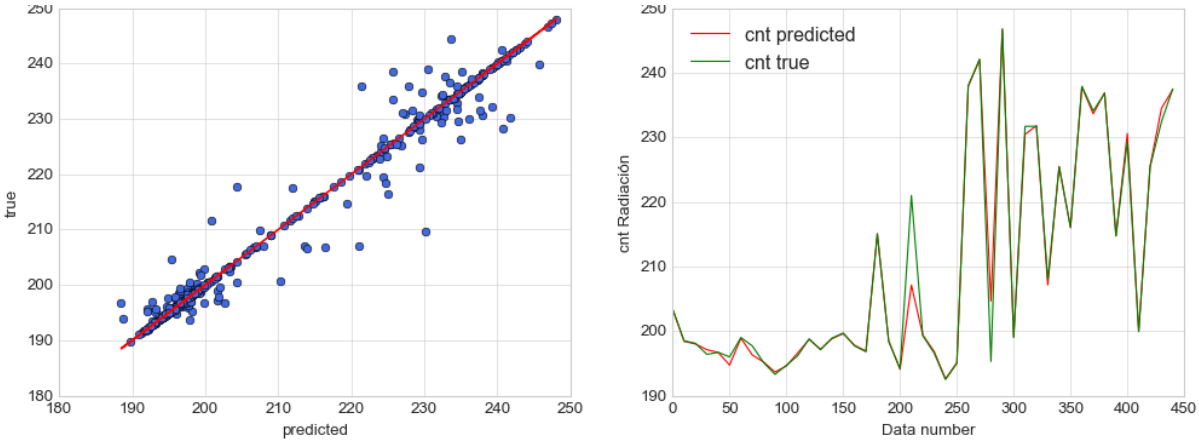


Figura 44. Cruce y traza datos MODIS

Tanto los resultados por hold out para Landsat como para MODIS se obtuvieron con un tercio de los datos prueba en convergencia con el experimento realizado en (Cabrera, 2015), con estos últimos resultados se obtuvieron las métricas de suma de error absoluto (SAE), error absoluto medio (MAE), raíz del error cuadrático medio (RMSE) y coeficiente de determinación r2. Los resultados a comparar están dados por la media de cada métrica y más una desviación estándar para el caso de SAE, MAE y RMSE y resta una desviación estándar para r2. Luego se agregaron los dos mejores resultados multi layer perceptron (mlp), multi layer perceptron ensemble (mlpe), más los resultados de SVM (ksvm) mostrados por (Cabrera, 2015) y finalmente se escalaron las métricas utilizando normalizador min max. Las Figura 45 y Figura 46 muestran los comparativos realizados con las bases de datos Landsat y MODIS.

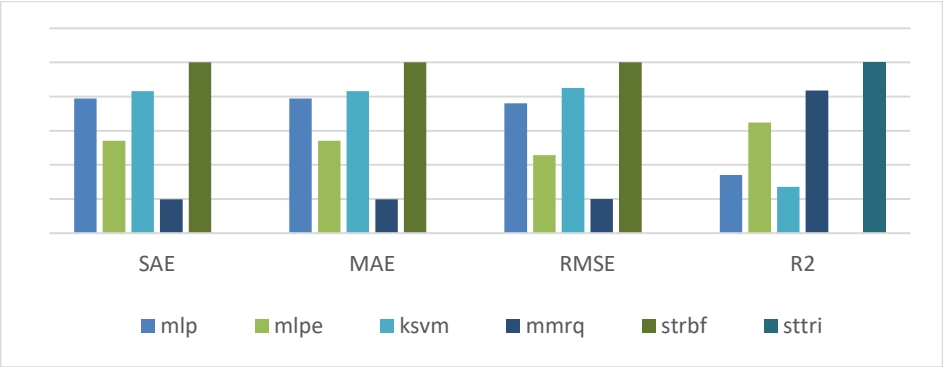


Figura 45. Comparativo funciones kernel base de datos Landsat

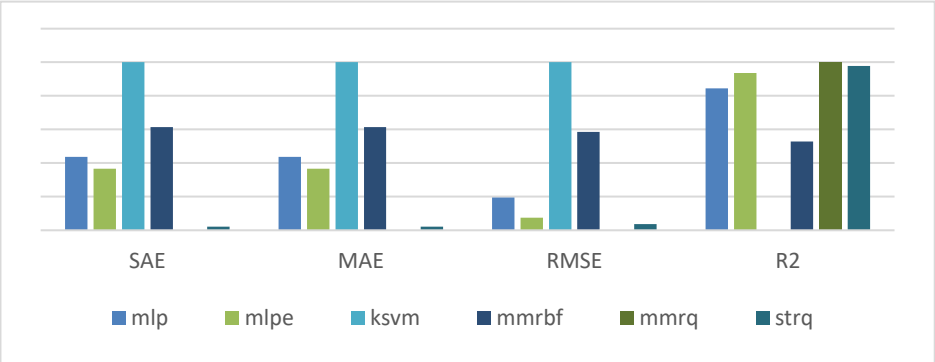


Figura 46. Comparativo funciones kernel base de datos MODIS.

El resultado de los mejores modelos para las base de datos Landsat y MODIS se tomaron para generar las regresiones sobre una malla de puntos que cubre la zona de estudio como se muestra en la Figura 47 y con ellas generar los mapas de radiación realizando interpolación mediante kriking como se indica en (Monger et al., 2016b) y knn con 4 vecinos para contrastar los resultados..

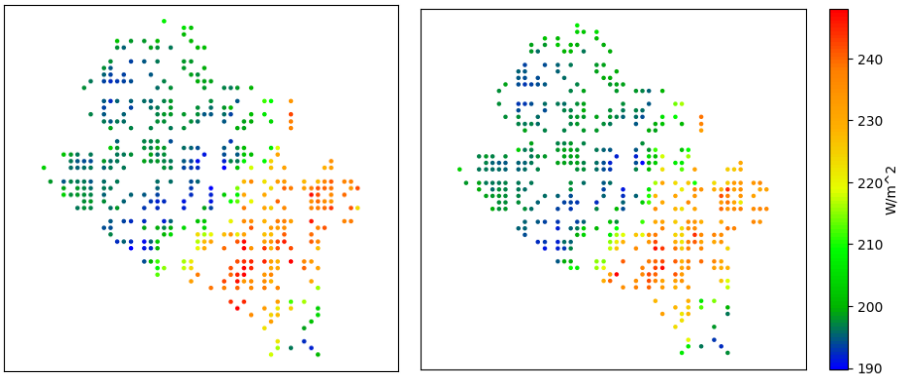


Figura 47. Predicciones geo localizadas Landsat (Derecha) y MODIS (Izquierda)

Los variogramas para realizar la interpolación de kriking se realizaron mediante un modelo esférico (ver Figura 48).

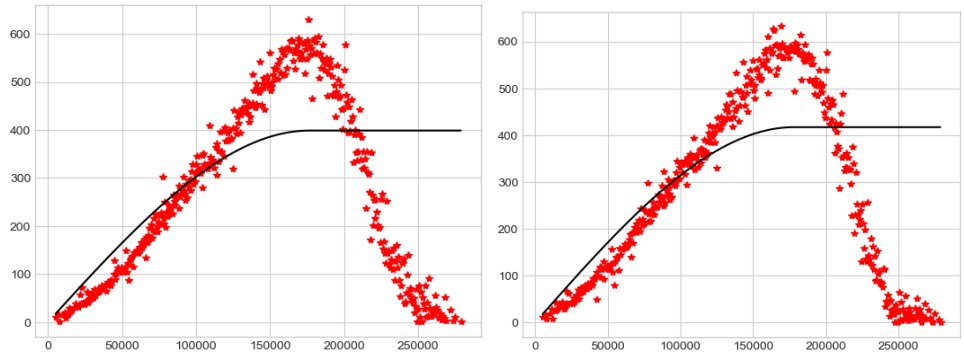


Figura 48. Variograma Landsat (Derecha) y MODIS (Izquierda)

En la Figura 48 se puede notar que los variogramas obtenidos son aproximadamente similares. El resultado de las interpolaciones se puede observar en la Figura 49 y Figura 50.

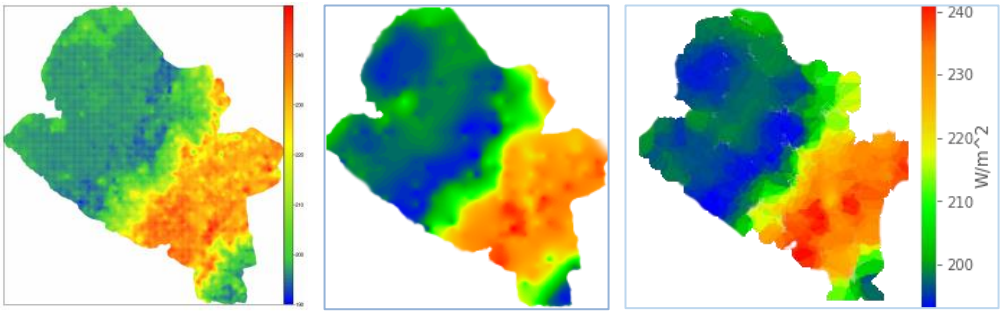


Figura 49. Comparativo Landsat izquierda (Cabrera, 2015), central y derecha está investigación

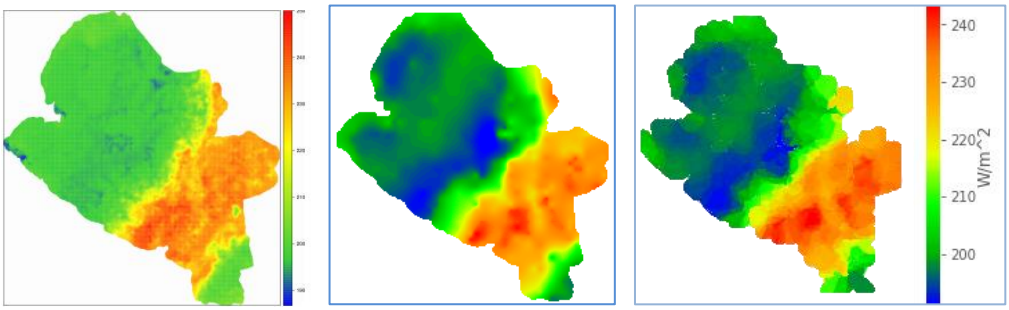


Figura 50. Comparativo MODIS izquierda (Cabrera, 2015), central y derecha está investigación

Los mapas centrales dentro de la Figura 49 y Figura 50 corresponden a los datos interpolados por kriking, los mapas a la derecha de estas figuras corresponden a interpolación por K-vecinos (knn).

5. Discusión y análisis de los resultados

El análisis de los insumos de los sensores Landsat y MODIS ha significado un reto en cada una de las etapas del proceso KDD. En el presente estudio se ha aportado en primera instancia en la etapa de adquisición de datos con un procedimiento de fragmentación de tablas utilizando el principio que denominado principio de fragmentación del promedio, eliminando los cuellos de botella para la obtención de los modelos con grandes volúmenes de datos disminuyendo el procesamiento, de días u horas a minutos.

Seguido a ello es significativamente importante resaltar el acoplamiento de los modelos KSVC, KSVR, KANNC y KANNR a la biblioteca scikit-learn como lo recomienda (Buitinck et al., 2013a). Este esfuerzo permitió ahorrar tiempo en la construcción de los algoritmos de búsqueda de hiperparámetros ya que la biblioteca scikit-learn ya tiene implementado algunos y la comunidad del software libre ha hecho esfuerzos para dotar de mecanismos más sofisticados como los buscadores evolutivos o algoritmos genéticos y la búsqueda bayesiana. Además significa un aporte a la comunidad del software libre y académica a nivel mundial para la construcción de modelos haciendo uso de las funciones kernel y uso de los modelos implementados en esta investigación.

Además, de acuerdo a lo enunciado por (Belanche, 2016a) y (B. Yu et al., 2016) las funciones kernel como la Canberra, Truncated, Triangular proporcionan mejores resultados que la función RBF para cierto tipo de datos esto se puede apreciar en los elementos resaltados en la Tabla 18, destacando que en todos los experimentos función RBF fue superada, también es notorio que en la mayoría de configuraciones la función Canberra obtuvo mejor exactitud qué las demás funciones kernel. Sin embargo esta búsqueda rápida de hiperparámetros puede no ser objetiva ya que ha partido su sintonización desde la técnica holdout. Atendiendo a los estipulado en (Pedregosa et al., 2011a) y (Buitinck et al., 2013b) donde se recomienda realizar configuración de hiperparámetros por validación cruzada.

Se corrobora mediante la propuesta del método expuesto en Figura 18 la hipótesis que las funciones kernel pueden mejorar los modelos de aprendizaje automático, los resultados mostrados en la *Tabla 19* muestran que existen varias configuraciones que pueden acercarse a los mejores resultados encontrados en (Cabrera & Pantoja, 2018) donde el mejor modelo mediante entrenamiento Holdout resultó ser Ensemble Multi Layer Peceptron (EMLP) con un coeficiente de correlación r2 del 0.92721 para Landsat y 0.94394 para MODIS. Entre las configuraciones encontradas en este primer experimento están la de escoger KSVR con normalizador min max, con funciones kernel rquadratic o rbf produciendo r2 de 0.925 y 0.909 respectivamente. Para los datos MODIS los resultados no fueron tan prometedores, no

obstante se considera que con un kernel triangular hay una buena aproximación con un r2 de 0.919.

Cabe resaltar que el experimento inicial se realizó con validación cruzada y la varianza de los resultados puede ser un factor determinante, por esta razón se realizó un segundo experimento buscando hiperparámetros por kernel mejorando los del experimento inicial, con ellos se reconfirma lo expuesto por (Buitinck et al., 2013b), ver Figura 19, de que es necesario usar sintonización de hiperparámetros mediante buscadores que usen validación cruzada.

Teniendo en cuenta lo anterior, los experimentos de sintonización de hiperparámetros muestran que considerando a cada función kernel por separado puede mejorar la calidad de comparación de un conjunto de modelos de aprendizaje automático, ya que en el experimento por funciones kernel independientes se obtienen mejores indicadores de calidad que en el experimento de sintonización por algoritmo. No obstante en el experimento de sintonización por algoritmo *Tabla 19* se observa en primera instancia, que los modelos obtenidos por KANNR y KSVR obtienen mejores resultados para la base de datos Landsat, en segunda instancia, se visualiza que los modelos provistos por KSVR en general funcionan mejor que KANNR, y en tercera instancia, los normalizadores tipo Normalizer han dado resultados de r2 inferiores a 0.5 para la mayoría de las configuraciones sintonizadas en las dos bases de datos.

Las anteriores afirmaciones son corroboradas por los resultados enmarcados en el experimento de sintonización por función kernel (ver Tabla 20, *Tabla 21*). Donde también se logra apreciar configuraciones de hiperparámetros adicionales, donde para la base de datos MODIS se observa que es probable que la mejor ruta posible de configuración en esta base de datos este dada por el uso del algoritmo KSVR, con la combinación de un normalizador Min Max, kernel RBF, esto se puede deducir al ver a r2 mayor a las demás configuraciones y el tiempo de entrenamiento (t) menor a la mayoría, las figuras de cajas de bigotes muestran además que la variabilidad de los resultados de esta configuración son menores a sus dos competidores más cercanos y que generalmente se podrían obtener valores superiores a su media y mediana.

Por otra parte en la base de datos Landsat, desde la Figura 35 pueden distinguirse dos rutas de configuración, ambas partiendo de modelos obtenidos por KSVR, la primera tomando un normalizador Min Max con kernel Rational Quadratic y la segunda tomando un normalizador Standard con kernel Triangle, el tiempo de entrenamiento es similar para las dos configuraciones, pero al observar las figuras se aprecia que el kernel Rational Quadratic tiene menor variabilidad, sin embargo los outliers muestran que pueden obtenerse valores atípicos con r2 mucho más altos o más bajos, aunque existen estos valores atípicos se alcanza a notar

que la concentración de los datos está por debajo de la media con probabilidad de obtener valores más bajos en el entrenamiento.

Al comparar las configuraciones de los modelos relacionados en las Figura 45 con las métricas resultantes por (Cabrera, 2015), se observa que para los datos Landsat los modelos obtenidos por KSVM en las combinación del normalizador standard kernel triangular y normalizador min max kernel Rational Quadratic tienen los mejores compromisos en cada una de las métricas relacionadas, acorde a la expuesto en párrafos anteriores. Empero los resultados por holdout indican que la normalización estándar con kernel triangular como la mejor alternativa.

Por otro lado para los datos MODIS en la Figura 46 el algoritmo KSVM con kernel Rational Quadratic con normalización min max o estándar obtienen el mejor compromiso en ese orden, aunque la brecha para este conjunto de datos está muy cercana al mejor resultado de (Cabrera, 2015). Sin embargo la función kernel rbf aunque mejora los resultados de ksvm (Cabrera, 2015) va en contraposición a lo expresado en párrafos anteriores donde está función prometía los mejores resultados.

Finalmente los mapas resultantes siguen un patrón relativamente similar al obtenido por (Cabrera, 2015), pero con un modelo de predicción más robusto detectando zonas con picos de radiación más elevados y consignando de esta manera menos incertidumbre a la hora de tomar decisiones para la instalación de receptores en zonas geográficas determinadas. De igual forma descartan zonas con media de radiación muy inferiores a los 200 Kw/m2. Igualmente se corrobora que kriking interpola los datos mucho mejor que knn asintiendo a la recomendación hecha por Monger (2016) no obstante knn es una alternativa cuyos patrones son cercanos a los modelos expuesto en la Figura 49 y Figura 50, dando indicios a posibles trabajos futuros.

6. Conclusiones y trabajos futuros

La distribución del proceso de adquisición en volúmenes de grandes datos, como se enuncia en las ecuaciones para procesamiento distribuido representan un aporte en la reducción del tiempo de días u horas a minutos.

Los scripts de diccionario de datos y exploración de datos en PGSQL son insumos genéricos que pueden ser utilizados en cualquier base de datos PostgreSQL.

El acoplamiento de las funciones kernel a la biblioteca scikit-learn permite interactuar de forma familiar con los algoritmos desarrollados, significa un aporte a la comunidad del software libre, y a estudios académicos de punto de partida para construir modelos acoplados a esta biblioteca usada a nivel mundial.

El marco experimental planteado permitió confirmar postulados y corroborar la tesis de que las funciones kernel en estudio se acondicionan mejor frente al tipo de datos suministrados. Al abordar la experimentación con los datos por algoritmo, por función kernel y luego aplicar Holdout permitió observar aspectos generales y particulares del performance que tienen las funciones kernel frente a los datos Landsat y MODIS.

Los resultados del análisis reflejan que SVM y los normalizadores estándar, mínimo máximo y kernel triangular y rational quadratic son los más indicados para realizar regresión sobre los datos Landsat y MODIS.

La visualización de los resultados permitió evaluar de forma más objetiva a cada algoritmo y los mapas finales corroboran que los patrones obtenidos son similares a (Cabrera et al., 2016), con obtención de picos de radiación más altos que sirvan para ubicar receptores de energía fotovoltaica.

Se recomienda realizar una formulación para distribuir el producto cartesiano y agrupaciones en un procesamiento distribuido para grandes volúmenes de datos, ya que es una operación computacionalmente muy costosa para la adquisición de los modelos para bases de datos satelitales.

Se recomienda estudiar la posibilidad de mejorar el performance de la sintonización de hiperparámetros con búsqueda bayesiana.

Se recomienda evaluar la posibilidad de hacer uso de funciones de activación no lineales y funciones kernel para evaluar el compromiso entre métricas de calidad y coste

computacional inspeccionando bases de datos de uso común, tanto para problemas de regresión como clasificación.

Se recomienda hacer uso de técnicas de visión artificial para encontrar patrones difíciles de ver y mejorar el aspecto de los mapas, y cuyos algoritmos de interpolación reflejen los límites de la zona de estudio sin acudir a los límites de la zona en estudio.

Se recomienda realizar un estudio comparativo de las funciones kernel frente a la implementación SVM con funciones callable y aproximadores como Nystroem en varios conjuntos de datos con diferentes tamaños.

Se recomienda realizar un estudio comparativo de funciones kernel acopladas a otros algoritmos de machine learning para clasificación o regresión lineal a imágenes satelitales, para saber que algoritmo trabaja mejor con kernels en estos conjuntos de datos.

Se recomienda mejorar el algoritmo KNN para obtener interpolaciones con menos sobreajuste en la generación de mapas satelitales.

Se recomienda implementar una metodología de sintonización de hiperparámetros escalable introduciendo filtros métricos y aumentando el ancho de banda de inspección de los modelos que sobrepasen las metas expuestas en estos filtros.

Bibliografía

Acciona. (2015). *Compromisos de los países más contaminantes contra el cambio climático* [Sostenibilidad]. https://www.sostenibilidad.com/cambio-climatico/compromisos-paises-contaminantes-cambio-climatico/

Aldabas-Rubira, E., & Colom, U.-C. T.-D.-E. (2015). *Introducción al reconocimiento de patrones mediante redes neuronales*. 3.

Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from Text: Machine Learning for Text-based Emotion Prediction. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 579–586. https://doi.org/10.3115/1220575.1220648

Artaraz, M. (2002). Teoría de las tres dimensiones de desarrollo sostenible. *Revista Ecosistemas*, *11*(2). https://doi.org/10.7818/re.2014.11-2.00

Badii, M.H., A. Guillen, & J.L. Abreu. (2016). *Energías Renovables y Conservación de Energía (Renewable Energies and Energy Conservation)*. http://scholar.googleusercontent.com/scholar?q=cache:aEzmM4FMpBgJ:scholar.google.com/+el+peligro+del+carbon+petroleo+y+energia+nuclear+&hl=es&as_sdt=0,5&as_ylo=2015

Banerjee, D. C., Paul, S., & Ghoshal, M. (2017). An Evolutionary Algorithm based Parameter Estimation using Pima Indians Diabetes Dataset. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(6), 4.

Basante-Villota, C. K., Ortega-Castillo, C. M., Peña-Unigarro, D. F., Revelo-Fuelagán, J. E., Salazar-Castro, J. A., & Peluffo-Ordóñez, D. H. (2018). Comparative Analysis Between Embedded-Spaces-Based and Kernel-Based Approaches for Interactive Data Representation. En J. E. Serrano C. & J. C. Martínez-Santos (Eds.), *Advances in Computing* (Vol. 885, pp. 28-38). Springer International Publishing. https://doi.org/10.1007/978-3-319-98998-3_3

Baudat, G., & Anouar, F. (2001). Kernel-based methods and function approximation. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, *2*, 1244-1249 vol.2. https://doi.org/10.1109/IJCNN.2001.939539

Belaid, S., & Mellit, A. (2016). Prediction of daily and mean monthly global solar radiation using support vector machine in an arid climate. *Energy Conversion and Management*, *118*, 105-118. https://doi.org/10.1016/j.enconman.2016.03.082

Belanche. (2016a). *Developments in kernel design*. 369-378. https://upcommons.upc.edu/handle/2117/23278

Belanche. (2016b). *Developments in kernel design*. 369-378. https://upcommons.upc.edu/handle/2117/23278

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013a). API design for machine learning software: Experiences from the scikit-learn project. *arXiv:1309.0238 [cs]*. http://arxiv.org/abs/1309.0238

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013b). API design for machine learning software: Experiences from the scikit-learn project. *arXiv:1309.0238 [cs]*. http://arxiv.org/abs/1309.0238

Byon, E., Choe, Y., & Yampikulsakul, N. (2016). Adaptive Learning in Time-Variant Processes With Application to Wind Power Systems. *IEEE Transactions on Automation Science and Engineering*, *13*(2), 997-1007. https://doi.org/10.1109/TASE.2015.2440093

Cabrera, Champutiz, Calderon, & Pantoja. (2016). *Landsat and MODIS satellite image processing for solar irradiance estimation in the department of Narino-Colombia*. 1-6. https://doi.org/10.1109/STSIVA.2016.7743306

Cabrera, & Pantoja. (2018). *Análisis del recurso eólico usando R en zonas no interconectadas (ZNI) del departamento de Nariño (Colombia)*. Conferencia Latinoamericana sobre Uso de R en Investigación + Desarrollo (LatinR 2018) - JAIIO 47 (CABA, 2018). http://sedici.unlp.edu.ar/handle/10915/72585

Chicco, G., Napoli, R., & Piglione, F. (2001). Load pattern clustering for short-term load forecasting of anomalous days. *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)*, *2*, 6 pp. vol.2-. https://doi.org/10.1109/PTC.2001.964745

Coordinación de Energías Renovables, & Dirección Nacional de Promoción. (2008). *Energías renovables, energía eólica*. http://www.energia.gov.ar/contenidos/archivos/publicaciones/libro_energia_eolica.pdf

Cuenya, & Rueti. (s. f.). *EPISTEMOLOGICAL AND METHODOLOGICAL CONTROVERSIES BETWEEN THE QUALITATIVE AND QUANTITATIVE PARADIGM IN PSYCHOLOGY | Cuenya | Revista Colombiana de Psicología*. Recuperado 20 de febrero de 2020, de https://revistas.unal.edu.co/index.php/psicologia/article/view/17795

Diego H. Peluffo-Ordóñez, John Aldo Lee, & Michel Verleysen. (2015). *Generalized kernel framework for unsupervised spectral methods of dimensionality reduction—IEEE Conference Publication*. https://ieeexplore.ieee.org/abstract/document/7008664

EUROPA PRESS. (2018, abril 25). *Chernóbil se desató por una explosión nuclear, seguida de otra de vapor*. https://www.europapress.es/ciencia/habitat-y-clima/noticia-chernobil-desato-explosion-nuclear-seguida-otra-vapor-20171117171506.html

Gonzales Mario, Cárdenas Victor, & Álvares Ricardo. (2019). *Energía solar fotovoltaica*. http://www.uaslp.mx/Comunicacion-Social/Documents/Divulgacion/Revista/Dieciseis/universitarios%20potosinos%20238.pdf#page=26

González, R. (2018, septiembre 12). Por qué es El Cairo la ciudad más contaminada del mundo. *El País*. https://elpais.com/internacional/2018/09/10/mundo_global/1536610694_766037.html

Grupo Banco Mundial. (2019). *Consumo de energía eléctrica (kWh per cápita)*. https://datos.bancomundial.org/indicador/EG.USE.ELEC.KH.PC?end=2018&start=1975&view=chart

Gutiérrez, G. U., & Guativa, J. V. (2019). Una revisión desde la epistemología de las ciencias, la educación STEM y el bajo desempeño de las ciencias naturales en la educación básica y media. *Revista Temas*, *0*(13), 109-121. https://doi.org/10.15332/rt.v0i13.2337

Jaime González. (2012). *Centralia, el pueblo que lleva medio siglo ardiendo*. BBC News Mundo. https://www.bbc.com/mundo/noticias/2012/08/120818_eeuu_centralia_fuego_mina_carbon_pensilvania_jg

Jaramillo, & Borjas. (2010). *Energía del viento*. 12.

Jaramillo Óscar, & Borjas Marco. (2010). *Energía del viento*. 12.

López. (2018). *IdUS - Fundamentos matemáticos de los métodos Kernel para aprendizaje supervisado*. https://idus.us.es/xmlui/handle/11441/77547

Lozada, J. (2014). Investigación Aplicada: Definición, Propiedad Intelectual e Industria. *CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica*, *3*(1), 47-50.

Luo, W., Taylor, M. C., & Parker, S. R. (2008). A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales. *International Journal of Climatology*, *28*(7), 947-959. https://doi.org/10.1002/joc.1583

Maldonado, Y. A. M., Roncancio, G. D. A., & Saavedra, J. D. S. (2019). Evaluación del potencial de energía solar en Santander, Colombia. *Prospectiva*, *17*(2), 1.

Mohamed Abuella, & Badrul Chowdhury. (2017). *[1703.09851] Solar Power Forecasting Using Support Vector Regression*. https://arxiv.org/abs/1703.09851

Monger, S. H., Morgan, E. R., Dyreson, A. R., & Acker, T. L. (2016a). Applying the kriging method to predicting irradiance variability at a potential PV power plant. *Renewable Energy*, *86*, 602-610. https://doi.org/10.1016/j.renene.2015.08.058

Monger, S. H., Morgan, E. R., Dyreson, A. R., & Acker, T. L. (2016b). Applying the kriging method to predicting irradiance variability at a potential PV power plant. *Renewable Energy*, *86*, 602-610. https://doi.org/10.1016/j.renene.2015.08.058

Murillo-Rendón, S., Peluffo-Ordóñez, D., Arias-Londoño, J. D., & Castellanos-Domínguez, C. G. (2013). Multi-labeler Analysis for Bi-class Problems Based on Soft-Margin Support Vector Machines. En J. M. Ferrández Vicente, J. R. Álvarez Sánchez, F. de la Paz López, & Fco. J. Toledo Moreo (Eds.), *Natural and Artificial Models in Computation and Biology* (Vol. 7930, pp. 274-282). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38637-4_28

Naciones Unidas. (2015, diciembre 14). *Población*. https://www.un.org/es/sections/issues-depth/population/index.html

Olatomiwa, L., Mekhilef, S., Shamshirband, S., Mohammadi, K., Petković, D., & Sudheer, C. (2015). A support vector machine–firefly algorithm-based model for global solar radiation prediction. *Solar Energy*, *115*, 632-644. https://doi.org/10.1016/j.solener.2015.03.015

Pantoja, A. D., Fajardo, D. F., & Guerrero, D. K. (2015). Hacia el análisis de oportunidades energéticas con fuentes alternativas en el departamento de Nariño. En *Las energías sustentables y sostenibles en el departamento de Nariño* (pp. 38-51). Unimar. http://www.umariana.edu.co/ojs-editorial/index.php/libroseditorialunimar/article/view/705

Pantoja, F. (2014). *PERSN*. PERS. pers.udenar.edu.co

Pardo, C., Rodríguez, J. J., García-Osorio, C., & Maudes, J. (2010). An Empirical Study of Multilayer Perceptron Ensembles for Regression Tasks. En N. García-Pedrajas, F. Herrera, C. Fyfe, J. M. Benítez, & M. Ali (Eds.), *Trends in Applied Intelligent Systems* (pp. 106-115). Springer. https://doi.org/10.1007/978-3-642-13025-0_12

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011a). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825−2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011b). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825−2830.

Peluffo-Ordóñez, D. H., Castro-Ospina, A. E., Alvarado-Pérez, J. C., & Revelo-Fuelagán, E. J. (2015). Multiple Kernel Learning for Spectral Dimensionality Reduction. En A. Pardo & J. Kittler (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision,*

and Applications (Vol. 9423, pp. 626-634). Springer International Publishing. https://doi.org/10.1007/978-3-319-25751-8_75

Petrov, A. N., & Wessling, J. M. (2015). Utilization of machine-learning algorithms for wind turbine site suitability modeling in Iowa, USA. *Wind Energy*, *18*(4), 713-727. https://doi.org/10.1002/we.1723

Ramírez Quintero Juan Pablo. (2018). *Estudio comparativo de funciones kernel para la clasificación de patrones descriptivos en clientes de comercio electrónico*. Universidad tecnológica de Pereira.

Revelo, J., Peluffo, D., & Ramí-rez, C. (2015). Educación y Formación Cultural en Fuentes de Energí-a Alternativa para el Departamento de Nariño. *Libros Editorial UNIMAR*. http://ojseditorialumariana.com/index.php/libroseditorialunimar/article/view/710

Rodman, L. C., & Meentemeyer, R. K. (2006). A geographic analysis of wind turbine placement in Northern California. *Energy Policy*, *34*(15), 2137-2149. https://doi.org/10.1016/j.enpol.2005.03.004

Sanchez Anzola Nicolas. (2015). *Vista de Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario*. https://revistas.uexternado.edu.co/index.php/odeon/article/view/4414/5004

Sánchez, M., & Paulina, A. (2015). *ESTUDIO DE REDES NEURONALES PARA LA RESOLUCIÓN DE PROBLEMAS EN LA ASIGNATURA INTELIGENCIA ARTIFICIAL DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTGACIONALES.* [Thesis, Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería en Sistemas Computacionales]. http://repositorio.ug.edu.ec/handle/redug/10352

Semana. (2018). *El desastre continúa: 7 años después de Fukushima*. 7 años del desastre nuclear de fukushima. https://www.semana.com/vida-moderna/articulo/7-anos-del-desastre-nuclear-de-fukushima/559804

Sharma, R., & Chaurasia, S. (s. f.). International Journal of Computer Network and Information Security(IJCNIS). *International Journal of Computer Network and Information Security(IJCNIS)*, *10*(12), 11.

Suárez, E. J. C. (2014). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*. 25.

Suarez, R. A. C., & Cervantes, J. M. R. (2013). *DISEÑO EXPERIMENTAL DE UN SISTEMA TRADICIONAL DE PANEL SOLAR DE PEQUEÑA ESCALA UBICADO EN LA CIUDAD DE BARRANQUILLA*. 85.

Tesauro, G. (1992). Practical Issues in Temporal Difference Learning. En J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems 4* (pp. 259–266). Morgan-Kaufmann. http://papers.nips.cc/paper/465-practical-issues-in-temporal-difference-learning.pdf

Universidad de Nariño. (2015). *Análisis de Oportunidades Energéticas con Fuentes Alternativas en el Departamento de Nariño*. http://alternar.udenar.edu.co

Wolberg, W. H., Street, W. N., & Mangasarian, O. L. (1994). Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters*, *77*(2), 163-171. https://doi.org/10.1016/0304-3835(94)90099-X

Yu, B., Wang, Y. T., Yao, J. B., & Wang, J. Y. (2016). A COMPARISON OF THE PERFORMANCE OF ANN AND SVM FOR THE PREDICTION OF TRAFFIC ACCIDENT DURATION. *Neural Network World*, *26*(3), 271-287. https://doi.org/10.14311/NNW.2016.26.015

Yu, C., Li, Y., Bao, Y., Tang, H., & Zhai, G. (2018). A novel framework for wind speed prediction based on recurrent neural networks and support vector machine. *Energy Conversion and Management*, *178*, 137-145. https://doi.org/10.1016/j.enconman.2018.10.008

Yusof, N., Zurita-Milla, R., Kraak, M.-J., & Retsios, B. (2014). Mining Frequent Spatio-Temporal Patterns in Wind Speed and Direction. En J. Huerta, S. Schade, & C. Granell (Eds.), *Connecting a Digital Europe Through Location and Place* (pp. 143-161). Springer International Publishing. https://doi.org/10.1007/978-3-319-03611-3_9

Anexo: Artículo científico

Comparativo de funciones Kernel en ANN y SVM mediante métricas de regresión para energía fotovoltaica

Universidad Internacional de la Rioja, Logroño (España)

2020-02-20

Resumen

La necesidad de mitigar la crisis del cambio climático y suplir la demanda energética, ha girado la atención hacia las fuentes de energías limpias, como la fotovoltaica. Promoviendo estudios enfocados en la oferta como en (Cabrera, 2016), donde se estructuraron para una superficie geográfica determinada, grandes bases de datos desde imágenes satelitales Landsat y MODIS de NASA, produciendo buenos patrones tanto en Redes Neuronales Artificiales (ANN) como en Máquinas de Soporte Vectorial (SVM). No obstante aún no se ha evaluado el desempeño de estos algoritmos haciendo uso de funciones kernel como las propuestas por Belanche (2015). Para solventar esto en esta investigación se aporta con la adquisición eficiente de datos de entrenamiento desde Big data, especialización de los algoritmos ANN y SVM con kernels acoplados a scikit-learn, marco experimental para sintonización de hiperparámetros y discusión, donde se muestra que las funciones kernel consiguen mejorar los resultados del estado del arte.

Palabras Clave

Función kernel, Redes neuronales artificiales, Maquinas de vectores de soporte, Energía fotovoltaica, imágenes satelitales

Introducción

Introducción Para detección de patrones mediante regresión, son ampliamente utilizadas técnicas de aprendizaje automático capaces de esculpir los aspectos más relevantes de la realidad mediante conjuntos de datos contextualizados. Existen diversas técnicas para la obtención de estos artefactos como las Redes Neuronales Artificiales, del inglés Artificial Neuronal Networks (ANN), y las Máquinas de Soporte Vectorial, del inglés Support Vector Machine (SVM), las cuales han venido ganado terreno al ofrecer alternativas para clasificar o predecir datos con un buen compromiso en escenarios donde no se aprecian relaciones de forma intuitiva[1]). Estas técnicas se caracterizan por tener amplia versatilidad y flexibilidad, sin embargo cabe la posibilidad de ser mejoradas con funciones kernel, las cuales permite llevar el conjunto de características a un espacio de observación lineal, logrando con ello encontrar patrones con un modelo de optimización lineal y como consecuencia reducir el error de forma eficiente[2]. No obstante, para encontrar patrones y predicciones mediante estas técnicas de clasificación y predicción, generalmente se recurre a la utilización de un par de funciones kernel como la gaussiana y polinómica, dejando vertientes de configuración de entrenamiento para estos algoritmos (SVM, ANN) sin explorar [3].

Los pronósticos para oferta de energías alternativas podrían deducirse de patrones hallados de los anteriores modelos ya que representan un gran atractivo para los planes de energización, debido a que el uso de estas energías se concentra en la transformación eléctrica, cuya demanda se proyecta para el año 2030 en aproximadamente 376 millones de kilovatios hora, suponiendo una tendencia de demanda constante de 3.130.71 Kwh per cápita [4] y un crecimiento demográfico del 15% [5]. Por supuesto que esta demanda de energía podría cubrirse con energías fósiles como las producidas por el carbón, el petróleo o la energía nuclear, sin embargo, esto conlleva a un incremento de riesgos ambientales y biológicos [6], debidos a sus emisiones contaminantes que han contribuido con el actual calentamiento global y protagonizado desastres como los evidenciados en el Cairo [7] Centralia [8], Fukushima [9], Chernóbil [10], entre otros.

En este contexto, en Colombia desde la unidad de planeación minero energética UPME se han venido desarrollando esfuerzos mediante planes de energización sostenibles (PERS) extendidos por el territorio nacional [11], para estudiar la demanda de energía por parte de los consumidores y la oferta de energía por parte de fuentes alternativas. Esto alineado por un lado a lo estipulado por el acuerdo de París donde se estableció el compromiso que existe por parte de los países en la lucha contra el cambio climático [12] y por otro lado, debido a que Colombia no es un país con suficiente reserva de energías fósiles. Específicamente, en el departamento de Nariño el desabastecimiento de petróleo es eminente, produciendo escases de combustible para los medios de transporte y zonas sin fluido eléctrico continuo. Desde los PERS desarrollados en Nariño nace el proyecto Análisis de Oportunidades Energéticas para (ALTERNAR) [13]. Este proyecto está enfocado en la oferta de energía, donde se realizaron predicciones aplicando algoritmos de aprendizaje automático para obtención de modelos de energía fotovoltaica [14]. En este estudio se utilizaron varios algoritmos de aprendizaje automático, donde se configuraron por defecto los hiperparámetros de cada algoritmo, y se obtuvo que ANN y SVM obtuvieron los mejores resultados. Las conclusiones que arroja esta investigación indican que los modelos ahí conseguidos, ofrecen un buen compromiso entre desempeño y rendimiento computacional para la predicción de energía fotovoltaica, no obstante, aún falta explorar los resultados frente a una sintonización de hiperparámetros sistemática incluyendo el uso de funciones kernel. Por lo tanto, existe la probabilidad de conseguir modelos mejores para regresión en datos de oferta energías limpias como la fotovoltaica.

Teniendo en cuenta los anteriores factores, se ve necesario optar por fuentes alternativas de energía no contaminantes, para cubrir la demanda de energía, alineándose a las políticas sostenibles

estructuradas a nivel mundial [15], Para ello es importante contar con proyecciones de concurrencia de energía que permitan planificar y orientar la adquisición de manera eficiente. Una mejora en estos modelos puede ser determinante y por ello puede ser relevante explorar las vertientes que una función kernel puede ofrecer.

De no contar con proyecciones robustas desde el punto de vista energético, se desaprovecharía la eficiencia de adquisición sobre puntos estratégicos que hubieran podido ser predichos para la implementación de macro proyectos, se optaría por receptores inadecuados o calibrados incorrectamente para el flujo de una fuente de energía [16], se atrasaría la movilidad de estaciones cuyo flujo de alimentación haya decaído. Lo que puede traer como consecuencia pérdidas de alimentación de energía, monetarias y mala prestación del servicio. Desde la perspectiva del aprendizaje automático al no tener referencia de un antecedente que perfile una ruta para el pre procesamiento, configuración de hiperparámetros y evaluación de funciones kernel, se incrementaría los tiempos para la consecución de un modelo para la detección de estos patrones y se podría incurrir en soluciones apresuradas guiadas por el azar. Igualmente aquellos estudios que requieran el uso de funciones kernel, al no tener en cuenta las funciones en estudio, podrían optar por una función kernel que pueda no alcanzar mejoras en el desempeño, ya sea por su formulación o a que sus hiperparámetros se encuentren desafinados.

Por lo tanto, con este estudio se pretende validar la hipótesis de que es posible encontrar mejoras en los modelos de predicciones de fuentes alternativas de energía fotovoltaica mediante la inspección del desempeño de los algoritmos SVM y ANN, introduciendo funciones kernel recomendadas por la literatura [17], mediante una sintonización de hiperparámetros sistemática y objetiva que acerque a los resultados en lo posible a la solución óptima de la función de coste. De manera que se genere un marco de discusión sobre los resultados en diferentes métricas de desempeño sobre cada función kernel, la configuración de sus hiperparámetros y cuyos resultados finales sean útiles para estudios relacionados con el objeto de estudio.

Estado del Arte

Los temas que rodean al objeto de estudio se han explorado en dos vertientes. Por un lado, se realizó una exploración de fuentes bibliográficas relacionadas con fuentes alternativas de energía donde se consultaron artículos relacionados con aspectos conceptuales y las técnicas asociadas para el descubrimiento de patrones en este tipo de datos. La otra vertiente de búsqueda estuvo concentrada en consultar fuentes bibliográficas relacionadas con los algoritmos de aprendizaje automático SVM, ANN y funciones kernel.

Energías alternativas

Energía solar. En este apartado se presentan los recursos bibliográficos divididos en aspectos conceptuales y predicciones para la obtención de patrones de energía fotovoltaica.

Aspectos conceptuales. Las definiciones y temas asociados a energía solar son enmarcados por el trabajo de grado de [18] y [7] donde se explica groso modo en que consiste la energía solar o fotovoltaica, los elementos necesarios para percibir este tipo de energía, y los componentes de radiación solar. Aportando en el presente estudio para destacar las variables relevantes y comprender la importancia de realizar limpieza a los datos mediante cotas de irradiación normales predominantes en una región determinada.

Predicciones de oferta de energía solar. En cuanto a las técnicas para realizar predicciones para estimar la oferta de energía solar se encuentra a [19] donde se analizó el recurso solar en Arizona Estados Unidos mediante un modelo de interpolación geo estadística denominado kriging para generar un modelo de variación de irradiacia solar, [20] utiliza un perceptrón multicapa con métodos ensemble tipo bagging para realizar predicciones sobre bases de datos de radiación solar, [21] utiliza Support Vector Machine (SVM) con Firefly (FFA) para predecir la radiación solar global horizontal.

Los artículos [22] y [14] Muestran los repositositorios para realizar adquisición de datos como: RetScreen, imágenes de los satélites Lantsat y MODIS. También las herramientas para realizar visualización, tratamiento y validación de datos como: Meteorom, mapa interactivo del IDEAM, el software PVsyst y los lenguajes de programación Matlab, R y Python para la realización de scripts de adquisición, transformación, carga y obtención de patrones. Además de realizar los anteriores aportes, estos artículos se destacan porque muestran aspectos metodológicos, que sirven para determinar la oferta energética y sus pronósticos mediante el uso de técnicas de descubrimiento de patrones en bases de datos (KDD).

Energía eólica. En este apartado se presentan los recursos bibliográficos divididos en aspectos conceptuales y predicciones para la obtención de patrones de energía eólica.

Aspectos conceptuales. En cuanto a energía eólica (Coordinación de Energías Renovables & Dirección Nacional de Promoción, 2008) y [23] explican a grandes rasgos en que consiste la energía eólica, los diferentes usos y trascienden en exponer como ocurre la transformación y adquisición mediante aerogeneradores.

Predicciones de oferta de energía eólica. En cuanto a las técnicas utilizadas para estimar el recurso eólico: [24]utiliza auto correlación espacio temporal, para estimar la potencia generada por aerogeneradores, [25] utiliza MCE por sus siglas Malla Cupular Elíptica para establecer a partir de atributos físicos y ambientales la ubicación más prometedora de estaciones receptoras de viento. [26] usa algoritmos de aprendizaje automático con técnicas de multi-etiqueta o múltiples expertos para modelar predicciones para la ubicación de aerogeneradores en Iowa Estados Unidos. En los Paises Bajos [27] para detectar anomalías en el flujo de viento utilizaron LCMseq por sus siglas del inglés Linear time closed Pattern, con este se buscaron patrones secuenciales significativos de velocidad y dirección del viento simultáneamente y detectar aquellos que salieran de los segmentos normales de la señal.

SVM ANN y Funciones kernel

Algoritmo SVM. Los aspectos conceptuales asociados al algoritmo de SVM se tomaron de [18] y [3] ambos artículos explican en que consiste el algoritmo y su conceptualización matemática, abordando el tema desde los modelos en el espacio lineal, cuasi lineal y no línea. En [28] se aborda la construcción y uso de la biblioteca scikit-learn, las buenas prácticas y patrones usados.

Algoritmo ANN. Los aspectos conceptuales asociados al algoritmo ANN se tomaron de [29], donde se explica el algoritmo base de las ANN, en [30] Se atiende los aspectos asociados a la arquitectura de ANN denominada kernel perceptron.

Funciones kernel. Los aspectos conceptuales asociados a las funciones kernel se tomaron de [31] y [17] donde se explican los modelos matemáticos de las funciones kernel, la construcción de estas funciones, sus propiedades y funciones kernel más utilizados. [32] Muestra como estrategia alterna a una arquitectura multicapa de red neuronal a las funciones kernel, proporciona definiciones complementarias del uso de las funciones kernel tanto para clasificación y regresión, y lista una serie de algoritmos de aprendizaje automático que pueden hacer uso de esta técnica.

Trabajos relacionados. En este apartado se presentan los recursos bibliográficos divididos en primer lugar a trabajos relacionados con los algoritmos SVM, ANN en la obtención de patrones en energía eólica y solar, en segundo lugar se presentan trabajos relacionados con las funciones kernel y finalmente se enuncian trabajos relacionados con estudios comparativos de algoritmos de aprendizaje automático.

SVM y ANN: Los artículos de [21], [33] muestran que se obtuvo un buen compromiso entre los datos de radiación solar pronosticados y medidos a partir de modelos de SVM al ingresar atributos como la temperatura, luz solar y radiación solar. Ambos estudios coinciden en que estos modelos requieren de pocos parámetros simples para obtener buena precisión. En el articulo [2] se propone diferentes marcos para la predicción de la oferta eólica utilizando la transformada wavelet como entrada para descomponer el histograma original en segmentos, luego se realiza extracción de características mediante el uso de arquitecturas de ANN, finalmente se toma a SVM como método predictor. [28] Muestran en su API técnicas de acoplamiento que pueden aplicar a las funciones kernel mediante métodos formales o aproximados en algoritmos de optimización cuadrática, o iterativa como en el algoritmo gradiente descendente estocástico SGD.

Funciones Kernel. En [34] se muestra el uso de las funciones kernel en reducción de dimensión proponiendo una combinación de funciones kernel haciendo uso del algoritmo PCA. En [34] muestran un marco generalizado de funciones kernel que mediante la incorporación de una SVM mejoran el rendimiento de kernel PCA. [35] Utiliza el kernel RBF configurado los hiperparámetros mediante sintonización por buscada en rejilla y compara los resultados en redes neuronales para la obtención de predicciones de oferta de energía solar. En [36]desarrolla métodos de aprendizaje adaptativo basados en el truco kernel sobre datos no estacionarios, para ello se utilizó datos asociados a la energía eólica, se utilizó la regresión ridge como algoritmo de aprendizaje y los kernel lineal, polinomial y RBF.

Estudios comparativos. Los artículos de [37], [38] son estudios comparativos de funciones kernel que proponen la realización de comparaciones de desempeño de algoritmos, el primero se enfoca en comparar el compromiso entre precisión y rendimiento de funciones kernel en patrones descriptivos de clientes de comercio electrónico, y el segundo propone una comparación entre métodos de reducción de dimensión y sus correspondiente versión kernel mediante métricas de calidad a través de una métrica propuesta denominada como curva RNX.

Objetivos y Metodología

Objetivos

Objetivo general. Determinar de las funciones algebraicas y trascendentes, las funciones kernel que ofrecen los mejores resultados en la predicción de oferta de fuentes alternativas de energía fotovoltaica en los algoritmos de SVM y ANN a partir de un estudio comparativo, haciendo uso de métricas de calidad.

Objetivos específicos: Procesar los datos de las base de datos Landsat y MODIS del repositorio GeoAlternar, para adquisición eficiente de conjuntos de entrenamiento, exploración y correlación, mediante scripts acoplados en las capas de acceso a datos y aplicación.

Implementar un conjunto de funciones kernel para evaluación del desempeño en los algoritmos SVM y ANN teniendo en cuenta costo computacional y efectividad en la predicción.

Clasificar los resultados obtenidos en función de las métricas, técnicas de normalización, función kernel y algoritmos de aprendizaje (SVM, ANN) mediante una sintonización de hiperparámetros para la obtención de modelos suboptimos.

Metodología

Este estudio se inscribe en el paradigma positivista, enfoque empírico analítico y pertenece a un tipo de investigación aplicada.

Paradigma. La investigación es de tipo positivista, un paradigma que es racional, objetivo y se basa en comprobar hechos y particularidades propias del conocimiento científico [39], característico de la esencia disciplinar ingenieril en la que se desarrolla la presente investigación, donde se estudiará el desempeño de los algoritmos SVM y ANN a través del uso de las funciones kernel.

Enfoque. La presente investigación está basada en el enfoque empírico-analítico, el cual permite observar y analizar las causas y efectos de la problemática estudiada, plantear procesos de experimentación y pruebas, que servirán de base para proponer una solución [40], tal como lo plantea el objetivo del presente proyecto, en el cual especifica se realizarán experimentos mediante una serie de rutas para la configuración de datos e hiperparámetros para la obtención de modelos suboptimos en la predicción de fuentes alternativas de energías limpias.

Tipo de investigación. Ejecutar una investigación como la desarrollada en el presente documento, da como resultado la creación de una solución orientada a un problema concreto a través de la aplicación de conocimientos específicos, lo cual clasifica esta investigación como una investigación aplicada, un tipo de investigación que tiene por finalidad producir conocimiento que se aplica directamente al objeto de estudio y que colateralmente, contribuye al incremento del nivel de vida de la sociedad [41]. Esto se verá reflejado en la utilidad que los resultados de la investigación generan en el sector energético y ambiental. De conformidad a la producción de conocimiento el presente estudio contribuye en disminución de búsqueda de sintonización de hiperparámetros para la obtención de predicciones en los algoritmos SVM, ANN en cuanto al conjunto de datos de energía solar y eólica.

Contribución

Para la realización del presente estudio se estructuró un marco experimental sintetizado en la Figura 1, donde se visualiza que inicialmente se estructura un conjunto de datos de entrenamiento desde un sistema gestor de base de datos, posterior a ello se selecciona un sintonizador de hiperparámetros, lista de normalizadores y algoritmos para encontrar los mejores modelos para luego ser evaluados por la técnica holdout y finalmente mostrar y rescatar los resultados.

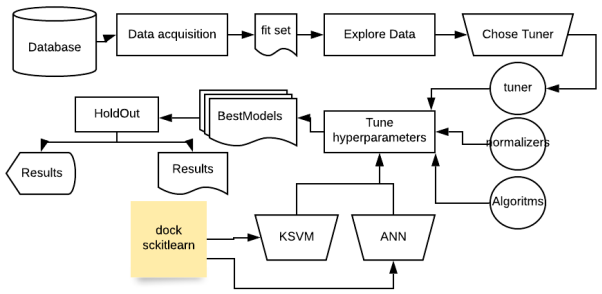


Figura 1. Marco experimental de comparación

Adquisición y exploración de datos

Para la adquisición de datos en primera instancia se restauró las bases de datos Landsat y MODIS estructuradas desde imágenes satelitales [42] en el gestor de base de datos PostgreSQL, cada una con 50Gb y 10Gb respectivamente. Las tablas más relevantes con las que se trabajó en el presente estudio se muestran en la Figura 2, la estructura de las tablas coincide en ambas bases de datos.

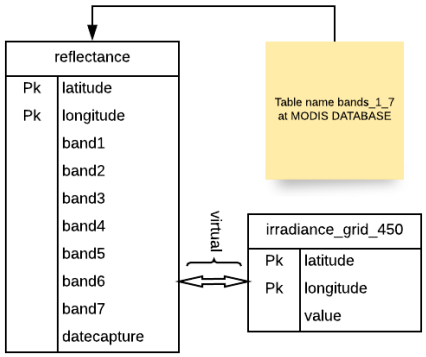


Figura 2. Tablas de base de datos Landsat y MODIS

.

La tabla reflectance almacena las conversiones del numero digital DN de las imágenes a reflectancia electromagnética acorde a lo estipulado por los sensores en Landsat y MODIS [43], y la tabla irradiance_grid_450 representa al mapeo sinusoidal de radiación de una región tomados de VAISALA INC (3TIER) [42].

Los datos de entrenamiento se obtienen desde el sistema gestor de base de datos, mediante las ecuaciones (1), (2) y (3) donde Y, π y $\bowtie$ son los operadores del algebra relacional agrupación, proyección y producto cartesiano natural, respectivamente, pk es la llave primaria, A la tabla reflectance, los elementos x_i representan a los atributos bandas de A y B a la tabla irradiance_grid_450 ver Figura 2.

$$\boldsymbol{R_1 = \Upsilon\, pk, avg(x_1)\, x_1, \dots, avg(x_n)\, x_n\, (A)} \quad (1)$$
$$\boldsymbol{R_2 = \pi * (B)} \quad (2)$$
$$\boldsymbol{R_3 = R1 \bowtie R2} \quad (3)$$

Debido a que las operaciones de agrupación y producto cartesiano son costosas computacionalmente para grandes cantidades de datos, como el caso de la tabla reflectance con 373.670.476 registros, es necesario realizar una distribución del procesamiento para estas operaciones, por lo tanto la ecuación (1) se transforma como se indica en las ecuaciones (4), (5) y (6).

$$\boldsymbol{R_{k,m} = \Upsilon_k^m\, pk, \sum x_1\ x_1, \dots, \sum x_n\ x_n, count(*)\, m\, (A)} \ (4)$$
$$\boldsymbol{R_0 = \cup_{k=0}^{m=\frac{n}{core}} all(R_{i,j}) \mid \{i = k * m, j = t + i\ , t \le n \to t = m, t > n \to t = n\}} \ (5)$$
$$\boldsymbol{R_1 = \Upsilon\, pk, \frac{\sum x_1}{\sum m}\ x_1, \dots, \frac{\sum x_n}{\sum m}\ x_n\, (R_0)} \ (6)$$

Una vez adquirido el modelo por (3) o (6) se cargan los datos y se explora los atributos mediante estadística descriptiva y se inspecciona la correlación entre variables por el método de Pearson.

Sintonización de hiperparámetros.

Para que la comparación de los resultados de las funciones kernel en los algoritmos de ANN y SVM se objetiva, es necesario aplicar un método de sintonización de hiperparámetros común para cada algoritmo [44], en este estudio se contemplaron el buscador en cuadricula, la búsqueda aleatoria y la búsqueda evolutiva por algoritmos genéticos, este último utilizando sklearn-deap. Para los experimentos se tomó a los algoritmos genéticos por obtener sintonizaciones con buen compromiso en sus métricas de calidad en un tiempo menor que la búsqueda en cuadricula y aleatoria [45]. Con el resultado de las búsquedas del sintonizador se analizan los resultados de cada kernel agrupados por algoritmo y normalizador escogiendo por cada grupo los mejores modelos. Estas búsquedas se obtuvieron mediante la biblioteca scikit-learn, acoplando las funciones kernel en los algoritmos SVM y ANN.

Sintonización de hiperparámetros.

Para realizar el acoplamiento de las funciones kernel se tomaron las definiciones formales expresadas por [46] ver Tabla 1. Estas funciones se encapsularon de forma estática en una clase denominada kernelF ver Figura 3.

Tabla 1. Formulación matemática de funciones kernel

Kernel	Kernel
$k^{RBF}(x,x') = e^{-\sum_{i=1}^{d}\gamma(x_i - x_i')^{\beta}}$ $\gamma > 0\,, \beta \in (0,2]$	$k^{Tri}(x,x') = \begin{cases} \|x-x'\| \le a \to 1 - \dfrac{\|x-x'\|}{a} \\ \|x-x'\| > a \to 0 \end{cases}$ $a > 0$
$k^{RB}(x,x') = \left(\sum_{i=1}^{d} e^{-\gamma(x_i - x_i')^2}\right)^m$ $\gamma > 0\,, m \in \boldsymbol{N}$	$k^{RQ}(x,x') = 1 - \dfrac{\|x-x'\|^2}{\|x-x'\|^2 + a}$ $a > 0$
$k^{Can}(x,x') = 1 - \frac{1}{d}\sum_{1}^{d}\gamma\frac{\lvert x_i - x_i'\rvert}{\lvert x_i\rvert + \lvert x_i'\rvert}$ $\gamma \in (0,1]$	$k^{Tru}(x,x') = \frac{1}{d}\sum_{1}^{d}\max\left(0, \frac{\lvert x_i - x_i'\rvert}{\gamma}\right)$ $\gamma > 0$

Para poder acoplar estas funciones se introdujo dos relaciones de uso, la primera mediante la producción de una matriz de Gram a ser utilizada por SVM y el segundo mediante llamadas callable utilizando la aproximación de NyOstrem. Posterior a ello se crean nuevas clases que heredan de los modelos SVM y ANN de biblioteca scikit-learn [28] sobrescribiendo y agregando nuevas funciones kernel a su funcionamiento ver Figura 3.

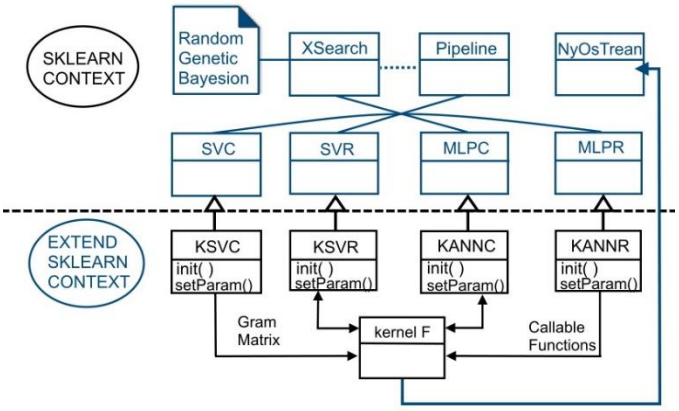


Figura 3 Acoplamiento de funciones kernel a scikit-learn

Formalmente el truco kernel se introduce en SVM para clasificación en la clase SVC y se nutre de más funciones kernel con la clase KSVC mediante la matriz de Gram K como se muestra en (7) y (8). Un proceso similar se sigue para el algoritmo de regresión provista con la clase KSVR cuya expresión formal se muestra en (9) y (10).

$$\boldsymbol{max: L(\alpha) = \sum_1^n \alpha_i - 1/2 \sum_{i=1}^n \alpha_i \alpha_j y_i y_j K\langle x_i, x_j\rangle} \ (7)$$
$$\boldsymbol{s.a: \sum_1^n \alpha_i y_i = 0\,, 0 \le \alpha_i \le C, i = 1, 2, \dots, n} \ (8)$$
$$\boldsymbol{max: L(\alpha) = \sum_1^n (\alpha_i^- - \alpha_i^+)\, y_i - \epsilon \sum_1^n (\alpha_i^- + \alpha_i^+) - \frac{1}{2}\sum_{i,j=1}^n (\alpha_i^- - \alpha_i^+)(\alpha_j^- - \alpha_j^+) K\langle x_i, x_j\rangle} \ (9)$$
$$\boldsymbol{s.a: \sum_1^n (\alpha_i^- - \alpha_i^+) = 0\,, 0 \le (\alpha_i^-, \alpha_i^+) \le C, i = 1,2, \dots, n} \ (10)$$

Para ANN se utilizó la aproximación de NyOstrem [47] mediante una transformación de base sobre el vector de características. Formalmente:

$$Z_L = W_L . a_{L-1} + b_L \text{ (11)}$$
$$N_t = NyosTream(kf) \text{ (12)}$$
$$X = Nt * X \text{ (13)}$$
$$a_0 = X \text{ (14)}$$
$$a_L = f(Z_L) \text{ (15)}$$
$$\hat{y} = predict(X_{batch} * Nt^T) \text{ (16)}$$

Donde (11) representa la suma ponderada, (12) la transformación de NyOstrem preparado con una función kernel kf [48], (13) la generación de la transformación, (14) la activación inicial o capa de entrada, (15) la función de activación que se ha definido lineal por defecto para que las funciones kernel sean las que cumplan el oficio de la transformación al espacio, en este caso el espacio de Hilbert. Cabe resaltar que es necesario sobrescribir el método predict del clasificador o regresor en KANNC y KANNR para realizar la transformación sobre cada batch del descenso del gradiente estocástico SGD (16).

Resultados o Evaluación

Una vez obtenido un método objetivo alineado a una biblioteca de aprendizaje automático se procedió a realizar los experimentos para evaluar el desempeño de los algoritmos en tres experimentos el primero de inspección para corroborar si las funciones kernel mejoraran el desempeño de los modelos dependiendo de los datos [46] ver Tabla 2, el segundo en una búsqueda por algoritmo para obtener una vista panorámica del desempeño de los modelos, y el tercero en una búsqueda por función kernel para afinar la búsqueda y realizar la comparación del compromiso de cada modelo en su r2 y desempeño computacional.

Inspección de compromiso funciones kernel

Mediante una búsqueda mediante una sintonización de hiperparámetros por algoritmos evolutivos con 10 generaciones, tamaño de población igual a 50, gen de mutación 0.1, y tamaño de torneo de 3, para los hiperparámetros de regularización C, kernel, gamma (γ), coef0 (a) y configuraciones para validaciones cruzadas hechas con 5 particiones tomando el 20% de los datos de prueba. Se han obtenido los resultados de la Tabla 3, Tabla 4 y Tabla 5 para los datos Landsat (L) y MODIS (M).

Tabla 2. Inspección de compromiso funciones kernel en SVM

db	**Cn**	**Rb**	**rbf**	**Rq**	**Tri**	**Tru**
B.C	0,9372	0,9401	0,8915	**0,9415**	0,7341	0,9358
GMC	**0,8811**	0,7051	0,6489	0,6807	0,7001	0,7406
GPC	0,9550	**0,9796**	0,9100	0,8700	0,8199	0,9950
IRIS	0,9467	0,9533	0,9400	0,9533	**0,9600**	**0,9600**
PBRS	**0,8407**	0,8314	0,7903	0,8258	0,8009	0,8364
PBRR	**0,9547**	0,8949	0,8551	0,7504	0,8054	0,8903
Wine	**0,9668**	0,9283	0,3990	0,3990	0,3990	0,9393

Las bases de datos (db) corresponden a bases de datos conocidas breast cáncer (BC), Iris y Wine, y bases de datos financieras GMC ganancia máxima, GMPC ganancia o pérdida, PRRS perdida con inclusión de pasos, PRRR perdida sin inclusión.

Sintonización de hiperparámetros por algoritmo

En la Tabla 3 se han descartado los resultados con r2 menores 0.6. Análoga a la configuración de sintonización anterior los resultados del experimentos de la Tabla 4 y Tabla 5 se han obtenido sintonizando a cada función kernel de forma independiente, descartando a los resultados con r2 menor a 0.7.

Tabla 3. Sintonización de hiperparámetros por algoritmo

db	algoritmo	normalizer	bestscore	kernel	γ	a	C
L	**ksvr**	**MM**	**0,93126928**	rq	-	1.0	1000
		Std	0,74816645	rq	-	0.1	10000
		MM	0,91008957	tr	1000.0	-	10000
		Std	0,89532301	rbf	0.001	-	10000
	kannr	MM	0,61504259	rq	-	0.001	10
		MM	0,70187407	hyp	1000.0	-	100
		Std	**0,84376659**	hyp	0.001	-	1000
M	**ksvr**	**MM**	**0,89555281**	rq	-	0.1	1000
		Std	0,89502163	rq	-	1.0	1000
		MM	0,80742541	can	0.1	-	10000
		NM	0,75734473	tri	0.1	-	10000
		Std	0,85986785	rbf	1.0	-	100
	kannr	MM	0,55475843	rq	-	100.0	10000
		MM	0,64854765	tru	1.0	-	1000
		Std	**0,74923367**	hyp	1000.0	-	10

Sintonización de hiperparámetros por función kernel

Los resultados resaltados en la Tabla 5 distinguen al r2 más alto. La Figura 4 muestra aquellos modelos cuyo rango de confianza (μ±σ) sea superior a 0.9 para Landsat y 0.87 para MODIS, se ha descartado al normalizador de la norma vectorial (NM).

Tabla 4. Sintonización por función kernel, base de datos Landsat

model	normalizer	kernel	r2	std	t	γ	a	C
KANNR	MM	rbf	**0,7013**	**0,1039**	**10,6**	**1.0**	-	**10000.0**
	St	hyp	**0,8455**	**0,0261**	**5,8**	**1.0**	-	**100.0**
		can	0,7221	0,0372	4,8	100.0	-	1000.0
		rbf	0,7170	0,1014	11,7	0.01	-	10.0
KSVR	**MM**	**rq**	**0,9313**	**0,0150**	**2,3**	-	**1.**	**1000.0**

							0	
		rbf	0,9130	0,0257	0,0	10.0	-	10.0
		rb	0,9045	0,0127	3,9	0.1	-	1000.0
		tru	0,8838	0,0169	4,2	100.0	-	10000.0
		can	0,8761	0,0186	4,4	1.0	-	100.0
		hyp	0,8563	0,0167	1,1	0.01	-	10000.0
		tr	0,7904	0,0306	1,9	1000.0	-	1000.0
		tru	0,7856	0,0491	3,6	0.01	-	10000.0
	St	**tr**	**0,9302**	**0,0137**	**2,1**	**100.0**	**-**	**1000.0**
		rbf	0,9079	0,0062	0,5	0.01	-	10000.0
		rb	0,8993	0,0062	23,5	0.01	-	10000.0
		can	0,8473	0,0117	4,3	10.0	-	1000.0
		hyp	0,8305	0,0194	1,1	0.01	-	100.0
		rq	0,7482	0,0268	2,3	-	0.1	10000.0

Tabla 5. Sintonización por función kernel, base de datos MODIS

model	normalizer	kernel	r2	Std	t	γ	a	C
KANNR	MM	rb	**0,7658**	**0,0603**	**8,0**	**10.0**	**-**	**10000.0**
		hyp	0,7547	0,0398	7,1	0.001	-	1000.0
		rb	0,7375	0,0936	8,4	0.1	-	10000.0
	MM	**rbf**	**0,9026**	**0,0082**	**0,1**	**10.0**	**-**	**100.0**
		rq	0,8956	0,0173	2,9	-	0.1	1000.0
		tr	0,8455	0,0264	2,4	10.0	-	100.0
		rb	0,8442	0,0371	3,3	1.0	-	10.0
		tru	0,8187	0,0214	4,0	100.0	-	10000.0
		can	0,8074	0,0268	4,7	1.0	-	1000.0
KSVR		hyp	0,7779	0,0332	1,3	0.01	-	10000.0
	St	rq	**0,8950**	**0,0097**	**2,6**	**-**	**1.0**	**1000.0**
		rb	0,8826	0,0283	4,2	0.001	-	10000.0
		rbf	0,8599	0,0171	0,1	1.0	-	100.0
		tr	0,8152	0,0300	2,0	10.0	-	10.0
		tru	0,8113	0,0104	4,0	1.0	-	100.0
		can	0,7859	0,0246	4,5	0.01	-	10000.0

Con los modelos visualizados en la Figura 4 se realizó 1000 entrenamientos con la técnica holdout con un tercio de datos de prueba, con estos últimos resultados se obtuvieron las métricas de suma de error absoluto (SAE), error absoluto medio (MAE), raíz del error cuadrático medio (RMSE) y coeficiente de determinación r2. Los resultados a comparar están dados por la media de cada métrica y más una desviación estándar para el caso de SAE, MAE y RMSE y resta una desviación estándar para r2. Luego se agregaron los dos mejores resultados multi layer perceptron (mlp), multi layer perceptron ensemble (mlpe), más los resultados de SVM (ksvm) mostrados por [42] y finalmente se escalaron las métricas utilizando normalizador min max. Las Figura 5 y Figura 6 muestran los comparativos realizados con las bases de datos Landsat y MODIS.

En las Figura 5 y Figura 6 mmrq corresponde a la combinación de una normalización min max con el kernel Rational Quadratic, strbf corresponde a la combinación de un normalización standard con el kernel RBF, sttri corresponde a la combinación de una normalización standard con kernel triangular, la mmrbf corresponde a una normalización min max con kernel RBF y strrq a la normalización standard con kernel triangular. Los resultados de los mejores modelos para Landsat y MODIS al realizar interpolación kriking y knn respectivamente sobre sus puntos predichos se muestra en las Figura 7 y Figura 8.

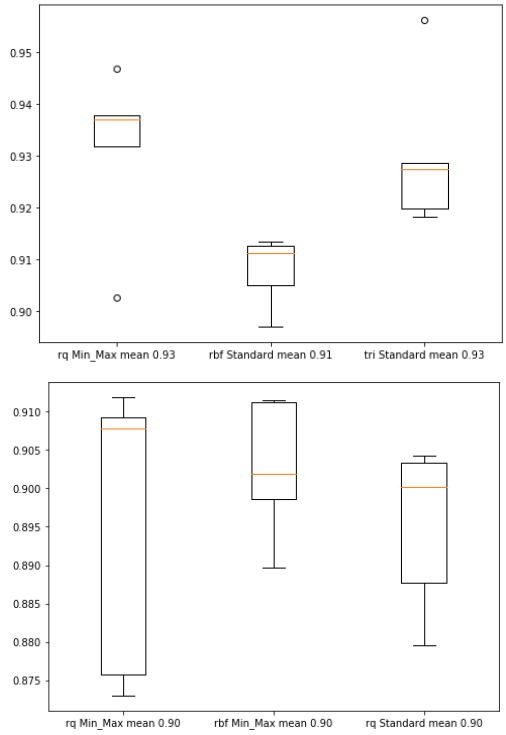


Figura 4. Modelos Landsat (arriba) y MODIS (abajo) dentro del rango de confianza

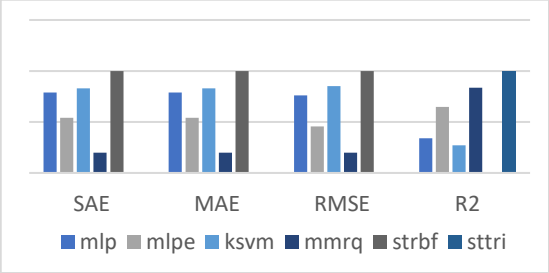


Figura 5. Comparativo funciones kernel base de datos Landsat

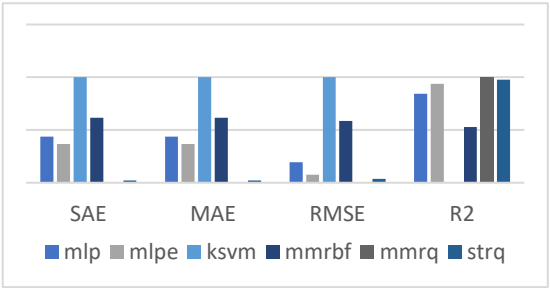


Figura 6. Comparativo funciones kernel base de datos MODIS

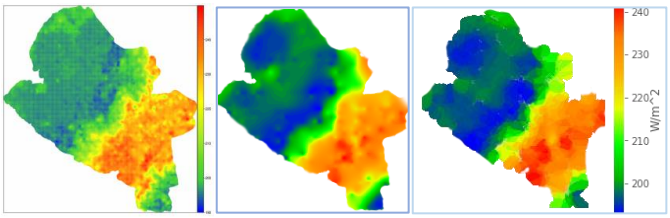


Figura 7 Comparativo Landsat izquierda (Cabrera, 2015), central y derecha está investigación

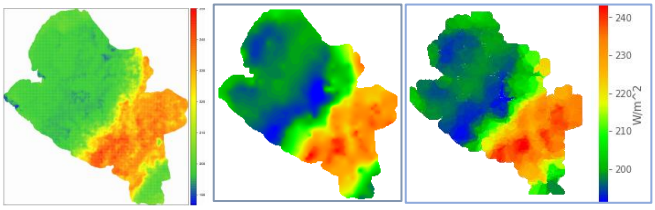


Figura 8 Comparativo MODIS izquierda (Cabrera, 2015), central y derecha está investigación

Discusión o Análisis de Resultados

De acuerdo a lo enunciado por [17]y [2]las funciones kernel como la Canberra, Truncated, Triangular proporcionan mejores resultados que la función RBF para cierto tipo de datos esto se puede apreciar en los elementos resaltados en la Tabla 2, destacando que en todos los experimentos función RBF fue superada, también es notorio que en la mayoría de configuraciones la función Canberra obtuvo mejor exactitud qué las demás funciones kernel. Sin embargo esta búsqueda rápida de hiperparámetros puede no ser objetiva ya que ha partido su sintonización desde la técnica holdout. Atendiendo a los estipulado por [3] y [28]que para la configuración de hiperparámetros se recomienda utilizar validación cruzada.

Teniendo en cuenta lo anterior, los experimentos de sintonización de hiperparámetros muestran que considerando a cada función kernel por separado puede mejorar la calidad de comparación de un conjunto de modelos de aprendizaje automático, ya que en el experimento por funciones kernel independientes se obtienen mejores indicadores de calidad que en el experimento de sintonización por algoritmo. No obstante en el experimento de sintonización por algoritmo Tabla 3 se observa en primera instancia, que los modelos obtenidos por KANNR y KSVR obtienen mejores resultados para la base de datos Landsat, en segunda instancia, se visualiza que los modelos provistos por KSVR en general funcionan mejor que KANNR, y en tercera instancia, los normalizadores tipo Normalizer han dado resultados de r2 inferiores a 0.5 para la mayoría de las configuraciones sintonizadas en las dos bases de datos.

Las anteriores afirmaciones son corroboradas por los resultados enmarcados en la Tabla 4 y Tabla 5. Donde también se logra apreciar configuraciones de hiperparámetros adicionales, donde para la base de datos MODIS se observa que es probable que la mejor ruta posible de configuración en esta base de datos este dada por el uso del algoritmo KSVR, con la combinación de un normalizador Min Max, kernel RBF, esto se puede deducir al ver a r2 mayor a las demás configuraciones y el tiempo de entrenamiento (t) menor a la mayoría, la Figura 4 muestra además que la variabilidad de los resultados de esta configuración son menores a sus dos competidores más cercanos y que generalmente se podrían obtener valores superiores a su media y mediana.

Por otra parte en la base de datos Landsat, desde la Tabla 4 pueden distinguirse dos rutas de configuración, ambas partiendo de modelos obtenidos por KSVR, la primera tomando un normalizador Min Max con kernel Rational Quadratic y la segunda tomando un normalizador Standard con kernel Triangle, el tiempo de entrenamiento es similar para las dos configuraciones, pero al observar la Figura 4 se aprecia que el kernel Rational Quadratic tiene menor variabilidad, sin embargo los outliers muestran que pueden obtenerse valores atípicos con r2 mucho más altos o más bajos, aunque existen estos valores atípicos se alcanza a notar que la concentración de los datos está por debajo de la media con probabilidad de obtener valores más bajos en el entrenamiento.

Al comparar las configuraciones de los modelos relacionados en la Figura 4 con las métricas resultantes por [42], se observa en las Figura 5 que para los datos Landsat los modelos obtenidos por KSVM en las combinación del normalizador standard kernel triangular y normalizador min max kernel Rational Quadratic tienen los mejores compromisos en cada una de las métricas relacionadas, acorde a la expuesto en párrafos anteriores. Empero los resultados por holdout indican que la normalización estándar con kernel triangular como la mejor alternativa.

Por otro lado para los datos MODIS en la Figura 6 el algoritmo KSVM con kernel Rational Quadratic con normalización min max o estándar obtienen el mejor compromiso en ese orden, aunque la brecha para este conjunto de datos está muy cercana al mejor resultado de [14]Sin embargo la función kernel rbf aunque mejora los resultados de ksvm [14]va en contraposición a lo expresado en párrafos anteriores donde está función prometía los mejores resultados.

Finalmente el mapa resultante sigue un patrón relativamente similar al obtenido por [14], pero con un modelo de predicción más robusto detectando zonas con picos de radiación más elevados y consignando de esta manera menos incertidumbre a la hora de tomar decisiones para la instalación de receptores en zonas geográficas determinadas.

Conclusiones

La distribución del proceso de adquisición en volúmenes de grandes

datos, como se enuncia en las ecuaciones (4), (5) y (6) representan un aporte en la reducción del tiempo de horas a minutos.

El acoplamiento de las funciones kernel a la biblioteca scikit-learn permite interactuar de forma familiar con los algoritmos desarrollados, significa un aporte a la comunidad del software libre, y a estudios académicos de punto de partida para construir modelos acoplados a esta biblioteca usada a nivel mundial.

El marco experimental planteado permitió confirmar postulados y corroborar la tesis de que las funciones kernel en estudio se acondicionan mejor frente al tipo de datos suministrados. Al abordar la experimentación con los datos por algoritmo, por función kernel y luego aplicar Holdout permitió observar aspectos generales y particulares del performance que tienen las funciones kernel frente a los datos Landsat y MODIS.

Los resultados del análisis reflejan que SVM y los normalizadores estándar, mínimo máximo y kernel triangular y rational quadratic son los más indicados para realizar regresión sobre los datos Landsat y MODIS.

La visualización de los resultados permitió evaluar de forma más objetiva a cada algoritmo y los mapas finales corroboran que los patrones obtenidos son similares a [14], con obtención de picos de radiación más altos que sirvan para ubicar plantas generadoras de energía eléctrica.

Se recomienda realizar una formulación para distribuir el producto cartesiano y agrupaciones en un procesamiento distribuido para grandes volúmenes de datos, ya que es una operación computacionalmente muy costosa para la adquisición de los modelos para bases de datos satelitales.

Se recomienda estudiar la posibilidad de mejorar el performance de la sintonización de hiperparámetros con búsqueda bayesiana.

Se recomienda evaluar la posibilidad de hacer uso de funciones de activación no lineales y funciones kernel para evaluar el compromiso entre métricas de calidad y coste computacional inspeccionando bases de datos de uso común, tanto para problemas de regresión como clasificación.

Se recomienda hacer uso de técnicas de visión artificial para encontrar patrones difíciles de ver y mejorar el aspecto de los mapas.

Se recomienda realizar un estudio comparativo de las funciones kernel frente a la implementación SVM con funciones callable y aproximadores como NyOstrem en varios conjuntos de datos con diferentes tamaños.

Apéndices

Los scripts de limpieza, inspección, adquisición de datos acoplamiento y experimentos realizados se encuentran en el repositorio https://github.com/magohector/fkernel .

Referencias

[1] Sanchez Anzola Nicolas, «Vista de Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario», 2015. [En línea]. Disponible en: https://revistas.uexternado.edu.co/index.php/odeon/article/view/4414/5004. [Accedido: 11-dic-2019].

[2] B. Yu, Y. T. Wang, J. B. Yao, y J. Y. Wang, «A COMPARISON OF THE PERFORMANCE OF ANN AND SVM FOR THE PREDICTION OF TRAFFIC ACCIDENT DURATION», *NNW*, vol. 26, n.º 3, pp. 271-287, 2016, doi: 10.14311/NNW.2016.26.015.

[3] F. Pedregosa *et al.*, «Scikit-learn: Machine Learning in Python», *Journal of Machine Learning Research*, vol. 12, p. 2825−2830, oct. 2011.

[4] Grupo Banco Mundial, «Consumo de energía eléctrica (kWh per cápita)», 2019. [En línea]. Disponible en: https://datos.bancomundial.org/indicador/EG.USE.ELEC.KH.PC?end=2018&start=1975&view=chart. [Accedido: 29-sep-2019].

[5] Naciones Unidas, «Población», 14-dic-2015. [En línea]. Disponible en: https://www.un.org/es/sections/issues-depth/population/index.html. [Accedido: 30-jun-2019].

[6] Badii, M.H., A. Guillen, y J.L. Abreu, «Energías Renovables y Conservación de Energía (Renewable Energies and Energy Conservation)», 2016. [En línea]. Disponible en: http://scholar.googleusercontent.com/scholar?q=cache:aEzmM4FMpBgJ:scholar.google.com/+el+peligro+del+carbon+petroleo+y+energia+nuclear+&hl=es&as_sdt=0,5&as_ylo=2015. [Accedido: 06-oct-2019].

[7] Gonzales Mario, Cárdenas Victor, y Álvares Ricardo, «Energía solar fotovoltaica», 2019. [En línea]. Disponible en: http://www.uaslp.mx/Comunicacion-Social/Documents/Divulgacion/Revista/Dieciseis/universitarios%20potosinos%20238.pdf#page=26. [Accedido: 14-nov-2019].

[8] Jaime González, «Centralia, el pueblo que lleva medio siglo ardiendo», *BBC News Mundo*, 2012. [En línea]. Disponible en: https://www.bbc.com/mundo/noticias/2012/08/120818_eeuu_centralia_fuego_mina_carbon_pensilvania_jg. [Accedido: 06-oct-2019].

[9] Semana, «El desastre continúa: 7 años después de Fukushima», *7 años del desastre nuclear de fukushima*, 2018. [En línea]. Disponible en: https://www.semana.com/vida-moderna/articulo/7-anos-del-desastre-nuclear-de-fukushima/559804. [Accedido: 06-oct-2019].

[10] EUROPA PRESS, «Chernóbil se desató por una explosión nuclear, seguida de otra de vapor», 25-abr-2018. [En línea]. Disponible en: https://www.europapress.es/ciencia/habitat-y-clima/noticia-chernobil-desato-explosion-nuclear-seguida-otra-vapor-20171117171506.html. [Accedido: 06-oct-2019].

[11] F. Pantoja, «PERSN», *PERS*, 2014. [En línea]. Disponible en: pers.udenar.edu.co. [Accedido: 06-oct-2019].

[12] Acciona, «Compromisos de los países más contaminantes contra el cambio climático», 2015. [En línea]. Disponible en: https://www.sostenibilidad.com/cambio-climatico/compromisos-paises-contaminantes-cambio-climatico/. [Accedido: 11-dic-2019].

[13] A. D. Pantoja, D. F. Fajardo, y D. K. Guerrero, «Hacia el análisis de oportunidades energéticas con fuentes alternativas en el departamento de Nariño», en *Las energías sustentables y sostenibles en el departamento de Nariño*, Unimar, 2015, pp. 38-51.

[14] Cabrera, Champutiz, Calderon, y Pantoja, «Landsat and MODIS satellite image processing for solar irradiance estimation in the department of Narino-Colombia», 2016, pp. 1-6, doi: 10.1109/STSIVA.2016.7743306.

[15] M. Artaraz, «Teoría de las tres dimensiones de desarrollo sostenible», *Revista Ecosistemas*, vol. 11, n.º 2, 2002, doi: 10.7818/re.2014.11-2.00.

[16] J. Revelo, D. Peluffo, y C. Ramí-rez, «Educación y Formación Cultural en Fuentes de Energí-a Alternativa para el Departamento de Nariño», *1*, 2015.

[17] Belanche, «Developments in kernel design», presentado en ESANN 2016 proceedings: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning: Bruges (Belgium), 24-26 April 2013, 2016, pp. 369-378.

[18] R. A. C. Suarez y J. M. R. Cervantes, «DISEÑO EXPERIMENTAL DE UN SISTEMA TRADICIONAL DE PANEL SOLAR DE PEQUEÑA ESCALA UBICADO EN LA CIUDAD DE BARRANQUILLA», p. 85, 2013.

[19] S. H. Monger, E. R. Morgan, A. R. Dyreson, y T. L. Acker, «Applying the kriging method to predicting irradiance variability at a potential PV power plant», *RENEWABLE ENERGY*, vol. 86, pp. 602-610, feb. 2016, doi: 10.1016/j.renene.2015.08.058.

[20] C. Pardo, J. J. Rodríguez, C. García-Osorio, y J. Maudes, «An

Empirical Study of Multilayer Perceptron Ensembles for Regression Tasks», en *Trends in Applied Intelligent Systems*, Berlin, Heidelberg, 2010, pp. 106-115, doi: 10.1007/978-3-642-13025-0_12.

[21] L. Olatomiwa, S. Mekhilef, S. Shamshirband, K. Mohammadi, D. Petković, y C. Sudheer, «A support vector machine–firefly algorithm-based model for global solar radiation prediction», *Solar Energy*, vol. 115, pp. 632-644, may 2015, doi: 10.1016/j.solener.2015.03.015.

[22] Y. A. M. Maldonado, G. D. A. Roncancio, y J. D. S. Saavedra, «Evaluación del potencial de energía solar en Santander, Colombia.», *Prospectiva*, vol. 17, n.º 2, p. 1, 2019.

[23] Jaramillo Óscar y Borjas Marco, «Energía del viento», p. 12, 2010.

[24] W. Luo, M. C. Taylor, y S. R. Parker, «A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales», *International Journal of Climatology*, vol. 28, n.º 7, pp. 947-959, 2008, doi: 10.1002/joc.1583.

[25] L. C. Rodman y R. K. Meentemeyer, «A geographic analysis of wind turbine placement in Northern California», *Energy Policy*, vol. 34, n.º 15, pp. 2137-2149, oct. 2006, doi: 10.1016/j.enpol.2005.03.004.

[26] A. N. Petrov y J. M. Wessling, «Utilization of machine-learning algorithms for wind turbine site suitability modeling in Iowa, USA», *Wind Energy*, vol. 18, n.º 4, pp. 713-727, 2015, doi: 10.1002/we.1723.

[27] N. Yusof, R. Zurita-Milla, M.-J. Kraak, y B. Retsios, «Mining Frequent Spatio-Temporal Patterns in Wind Speed and Direction», en *Connecting a Digital Europe Through Location and Place*, J. Huerta, S. Schade, y C. Granell, Eds. Cham: Springer International Publishing, 2014, pp. 143-161.

[28] L. Buitinck *et al.*, «API design for machine learning software: experiences from the scikit-learn project», *arXiv:1309.0238 [cs]*, sep. 2013.

[29] E. Aldabas-Rubira y U.-C. T.-D.-E. Colom, «Introducción al reconocimiento de patrones mediante redes neuronales», p. 3, 2015.

[30] R. Sharma y S. Chaurasia, «International Journal of Computer Network and Information Security(IJCNIS)», *International Journal of Computer Network and Information Security(IJCNIS)*, vol. 10, n.º 12, p. 11.

[31] López, «idUS - Fundamentos matemáticos de los métodos Kernel para aprendizaje supervisado», 2018. [En línea]. Disponible en: https://idus.us.es/xmlui/handle/11441/77547. [Accedido: 06-oct-2019].

[32] G. Baudat y F. Anouar, «Kernel-based methods and function approximation», en *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, 2001, vol. 2, pp. 1244-1249 vol.2, doi: 10.1109/IJCNN.2001.939539.

[33] S. Belaid y A. Mellit, «Prediction of daily and mean monthly global solar radiation using support vector machine in an arid climate», *Energy Conversion and Management*, vol. 118, pp. 105-118, jun. 2016, doi: 10.1016/j.enconman.2016.03.082.

[34] D. H. Peluffo-Ordóñez, A. E. Castro-Ospina, J. C. Alvarado-Pérez, y E. J. Revelo-Fuelagán, «Multiple Kernel Learning for Spectral Dimensionality Reduction», en *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, vol. 9423, A. Pardo y J. Kittler, Eds. Cham: Springer International Publishing, 2015, pp. 626-634.

[35] Mohamed Abuella y Badrul Chowdhury, «[1703.09851] Solar Power Forecasting Using Support Vector Regression», 2017. [En línea]. Disponible en: https://arxiv.org/abs/1703.09851. [Accedido: 14-nov-2019].

[36] E. Byon, Y. Choe, y N. Yampikulsakul, «Adaptive Learning in Time-Variant Processes With Application to Wind Power Systems», *IEEE Transactions on Automation Science and Engineering*, vol. 13, n.º 2, pp. 997-1007, abr. 2016, doi: 10.1109/TASE.2015.2440093.

[37] Ramírez Quintero Juan Pablo, «Estudio comparativo de funciones kernel para la clasificación de patrones descriptivos en clientes de comercio electrónico». Universidad tecnológica de Pereira, 05-oct-2018.

[38] C. K. Basante-Villota, C. M. Ortega-Castillo, D. F. Peña-Unigarro, J. E. Revelo-Fuelagán, J. A. Salazar-Castro, y D. H. Peluffo-Ordóñez, «Comparative Analysis Between Embedded-Spaces-Based and Kernel-Based Approaches for Interactive Data Representation», en *Advances in Computing*, vol. 885, J. E. Serrano C. y J. C. Martínez-Santos, Eds. Cham: Springer International Publishing, 2018, pp. 28-38.

[39] Cuenya y Rueti, «EPISTEMOLOGICAL AND METHODOLOGICAL CONTROVERSIES BETWEEN THE QUALITATIVE AND QUANTITATIVE PARADIGM IN PSYCHOLOGY | Cuenya | Revista Colombiana de Psicología». [En línea]. Disponible en: https://revistas.unal.edu.co/index.php/psicologia/article/view/17795. [Accedido: 20-feb-2020].

[40] G. U. Gutiérrez y J. V. Guativa, «Una revisión desde la epistemología de las ciencias, la educación STEM y el bajo desempeño de las ciencias naturales en la educación básica y media», *Revista Temas*, vol. 0, n.º 13, pp. 109-121, oct. 2019, doi: 10.15332/rt.v0i13.2337.

[41] J. Lozada, «Investigación Aplicada: Definición, Propiedad Intelectual e Industria», *CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica*, vol. 3, n.º 1, pp. 47-50, 2014.

[42] Cabrera, Champutiz, Calderon, y Pantoja, «Landsat and MODIS satellite image processing for solar irradiance estimation in the department of Narino-Colombia», 2016, pp. 1-6, doi: 10.1109/STSIVA.2016.7743306.

[43] ESRI, «Imágenes y Teledetección», 2019. [En línea]. Disponible en: https://www.arcgis.com/apps/Cascade/index.html?appid=5072b8d56cef4f7bb5d24e5d840461da. [Accedido: 23-feb-2020].

[44] D. C. Banerjee, S. Paul, y M. Ghoshal, «An Evolutionary Algorithm based Parameter Estimation using Pima Indians Diabetes Dataset», *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, n.º 6, p. 4, 2017.

[45] D. C. Banerjee, S. Paul, y M. Ghoshal, «An Evolutionary Algorithm based Parameter Estimation using Pima Indians Diabetes Dataset», *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, n.º 6, p. 4, 2017.

[46] Belanche, «Developments in kernel design», presentado en ESANN 2016 proceedings: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning: Bruges (Belgium), 24-26 April 2013, 2016, pp. 369-378.

[47] L. Yang, M. He, J. Zhang, y V. Vittal, «Support-Vector-Machine-Enhanced Markov Model for Short-Term Wind Power Forecast», *IEEE Transactions on Sustainable Energy*, vol. 6, n.º 3, pp. 791-799, jul. 2015, doi: 10.1109/TSTE.2015.2406814.

[48] L. Buitinck *et al.*, «API design for machine learning software: experiences from the scikit-learn project», *arXiv:1309.0238 [cs]*, sep. 2013.