Universidad Internacional de La Rioja (UNIR)

Engineering School

Master in Visual analytics and Big Data

Fraud Detection on European Food and Animal Trade with Machine Learning Algorithms

Master Final Thesis

Author: de Paz Martin, Maria del Pilar

Director/a: Tejeda Lorente, Álvaro

City: Brussels
Date: 15 Sept. 2017

Master	
Visual Analytics and Big Data	Surname: de Paz Martin
	Name: María del Pilar

Abstract

TRACES is an European Commission system to keep traceability of European imports and exports of animal products. Imports are exposed to several checks at entry points (Europe) but, given the high trading volume, only a subset is inspected (based on a human decision) and only a 2% of checked imports are effectively rejected. The principal goal of this work is to find an effective and efficient solution to early detect imports that have a high risk of not being appropriate for import (fraudulent). This work will analyse TRACES system and its generated data to find a predictive model based on machine learning algorithms to help to decision-making. Results of this work show that, even with a highly imbalanced class as we have in this domain, it is possible to have a ratio of true positives near 90% at country level inspection.

Keywords: fraud detection, animal food, international trading, machine learning, help to decision-making

Resumen

Nota: TRACES es un sistema de la Comisión Europea que ofrece trazabilidad de las importaciones y exportaciones europeas. Las importaciones de mercancías de origen animal son objeto de varios controles cuando entran en Europa, pero dado el gran volumen de comercio, solo se pueden controlar un subconjunto de todas las importaciones realizadas (basadas en decisión humana), actualmente alrededor un 2% de estas mercancías son rechazadas. El principal objetivo de este trabajo es desarrollar una solución efectiva y eficiente que permita detectar de manera anticipada las mercancías que tienen un alto riesgo de no ser apropiadas para ser comercializadas en Europa (fraudulentas). Este trabajo, analiza el sistema TRACES y sus datos generados para proporcionar un modelo predicción basado en algoritmos de aprendizaje automático para ayudar a la toma de decisiones. Los resultados de este trabajo demuestran que, a pesar del desequilibrio de clases en este dominio, es posible obtener un ratio de verdaderos positivos cercano al 90% a nivel de inspección nacional.

Palabras Clave: detección de fraude, productos de origen animal, comercio internacional, aprendizaje automático, ayuda a la toma de decisiones

Master	
Visual Analytics and Big Data	Surname: de Paz Martin
	Name: María del Pilar

1. Table of Contents

1. Introduction

This chapter will give an overview to the reader about what this thesis is about, business targets and goals to be reached within the proposed solution.

1.1. Overview

TRACES is a European Commission online application that has been working for more than ten years:

"TRACES' main purpose is to digitalize the certification process and linked procedures storing all relevant data for tracking purposes. These certificates contain valuable information" [1]

TRACES has a database with millions of certificates and history records for each certificate. One certificate contains many fields, such as product category, product, weight, country of origin, country of destination, country of clearance, exporter, importer, means of transport, time, date, etc. Each certificate easily contains between 50 and 150 fields, being several kinds of certificates: animals, products, plants, etc.

In order to fight against fraud and potential risks for consumers, border control authorities perform control checks to the imported goods at the customs border. *"Official control checks are performed by EU countries to verify that businesses comply with agri-food chain rules. These rules cover the safety and quality of food and feed, plant health, animal health and welfare. These rules are also applied to agri-food chain products entering the EU from third countries"* [2].

So, TRACES system allows best risk management practices avoiding health threats coming from imported goods. This is achieved by detecting and rejecting products at the border based on gathered data, i.e., chickens contaminated with salmonella, pork meat contaminated, vegetables with many pesticides, etc. It also helps to fight against fraud that, in some cases, impacts consumer safety or simply poses a financial risk.

1.2. Goals and Scope

The principal goal of this thesis is to find an effective and efficient solution to early detect imports that have a high risk of not being appropriate for import (fraudulent) to Europe at the border inspection post, or entry points, and with a fair performance.

Master	
Visual Analytics and Big Data	Surname: de Paz Martin
	Name: María del Pilar

This thesis is a proof of concept and proposes the usage of IA algorithms to mitigate fraud detection, or irregular products/animals imports, at the border area (customs) and, at the same time, minimizing the number of control checks needed. This proof of concept shall help decision-making actors to detect high-risk consignments (imported product) by using one or several types of Machine Learning algorithms: linear regression, neural networks, regression trees, support vector machines, random forests, etc. We will perform a Binary Classification based on two classes: rejected or accepted (certificates). I am choosing an initial set of well-known linear and not linear algorithms linear and not linear that usually have a good performance in binary classification problem [3].

The select set of machine learning algorithms will be assessed and tuned based on their standard performance and on specific needs since, as we will see, TRACES data has particular characteristics that make our main goal harder to achieve.

1.3. Document Structure

This document is divided into five different sections:

1. Context and the State of Art: this section explains the domain area (European food and animal trading) and explores other works related to the same domain with a similar approach.
2. Methodology: this section defines what exactly we want to achieve, regarding a machine learning model, and what are the different phases needed to design and produce that final model.
3. Proof of Concept development: this section realises the different processes needed to validate the proof of concept: a machine learning model applicable to our case.
4. Results: this section will analyse and assess the produced model adequacy to the domain needs (European food and animal trading).
5. Future lines of work: this section proposes new lines of work and possible improvements over the obtained results.

2. Context and the State of Art

Detection fraud with the help of machine learning algorithms nowadays is widely used in a variety of domains: credit card [4], financial statements [5], automobile insurance fraud [6], etc.; but there are not many publications in the domain of international food and animal, not any under the border post inspections subject. All publications focus either on laboratory/on-field tools to detect food hazards.

In 2015 a paper about fraud detection related to products that contained an associated health alert was made, using Bayesian Networks algorithms [7]. Although the domain is a subset of this work, the techniques used were the same. Results of this research are not very promising: yielding 52% of fraud detection when data regarding fraud committers (actors) was not available to the model, and reaching 80% when such past fraud data was available. The model was highly dependent on the actor's past fraudulent data. One of the main reason for these results could be the few available data for the model (very specific type and a small set of consignments that were the subject of laboratory inspections).

As commented in the introduction of this work, there are millions of registers available for analysis, extraction, preprocess, and to be input to machine learning tools and algorithms applications. Only a small fraction, as we will see, of such big sample is usable for the main objective of this work, I will consider it is enough to have positive results on detecting frauds or irregular consignments at customs borders. This work will prove that it is possible to get valuable information and detect fraud from the analysis of generated data by food and animal trading tracking systems (TRACES) when we have enough data (we will see four years of data in this system provides the most optimal conditions).

Regarding the huge amount of data available and the small subset needed, the main data to be analsyse is what TRACES defines as "certificate". The certificate object represents a consignment in real life; there are several kinds of certificates in TRACES, following we have a brief description of some of them:

- **CVEDA** : Common Veterinary Entry Document for Animals
- **CVEDP**: Common Veterinary Entry Document for products of animal origin
- **CHEDPP**: Common Health Entry Document for Plants and Plant protection
- **DOCOM**: Commercial document for intra-EU exchanges of animal by-products
- **CED**: Common Entry Documents for feed and food of non-animal origin

The system and, by extension, the data model is quite complex, having more than a hundred of attributes per certificate type. For this proof of concept, I will focus in one certificate: **CVEDP**. There are two reasons for this:

- CVEDP is one of the most used certificates in TRACES, so **availability** and **quality** data are suitable for this thesis.
- Consignments belonging to these certificates can be subjet of **controls that are not mandatory** at the border point entry (BCP) in Europe. The proof of concept of this thesis aims to provide advice to the control authority on deciding to perform, or not, complementary controls.

Regarding this proof of concept target user, actors deciding if they must perform inspections over consignment, we can see that deciding which consignments must be checked is a tough decision just showing some figures of current inspections:

- Number of checked certificates (physical check) since 2011 for CVEDP certificate with Germany as entry point: **226894**
- Number of rejected certificates that underwent for a check (physical check) since 2011 for CVEDP certificate with Germany as entry point: **5715**
- Number of valid certificates that underwent for a check (physical check) since 2011 for CVEDP certificate with Germany as entry point: **220407**

So, **the ratio of rejected certificates that are checked is less than 3%.** Authorities have performed many unnecessary checks that led to the acceptance of the consignment:

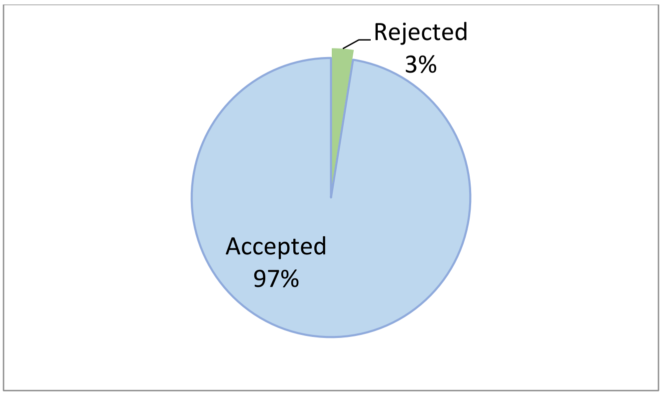


Figure 1: Ratio Rejected/Accepted to Total Inspections.

Above figure shows a false positive rate (if we consider the class "rejected" as the positive class) of 98%. I do not know the number of false negatives (accepted fraudulent consignments), this information is not recorded in TRACES, but for this work, we only need to know the true positives, and I do have the means to identify them, or almost true positives, as defined in section Methodology.

This work shall improve the ratio of true positives by detecting in advance if a consignment has a high probability to be rejected, it will detect with an effective and efficient ratio, consignments with a high risk of being rejected (fraudulent consignments) at customs point.

Regarding fraud detection with machine learning mechanisms, there multiple domains where it is applied, but it is in the financial sector and, more specifically, on payment transactions where we find most of the studies. Actually, a simple search in Google Scholar with the terms "fraud detection machine learning" (all words must be present in any place of the scientific article) yields 49.000 results, and in the first ten results, 8 are related to financial sector on card payments. Regarding algorithms used to detect fraud, it does go from linear algorithms like linear regression or logistic regression, no linear algorithms like Naïve Bayes, k-Nearest Neighbours, Classification Trees, Support Vector Machines, and time-series and recurrent neural networks. There is no really a specific algorithm for fraud detection; it really depends on the data we have available and the underlying problem.

For this proof of concept, I will choose a set of different type of representative algorithms to face the problem. Chosen algorithms represent a good set of linear and no linear algorithms that are demonstrated that can work very well in binary classification problems [8]:

GLM (Model Linear Generalized): This model is a generalization of ordinary linear regression. This model relates the aleatory distribution of the dependent variable with the non-aleatory part (systematic part) in an experiment, through a function called link function [9].

CART (Classification and Regression Tree): CART is a supervised algorithm based on classification and regression trees. Regression and classifications trees have some similar characteristics and also some important differences, like the used procedures to determine where to divide [10].

KNN (K-nearest neighbors): KNN is a method of supervised classification and regression, in both cases, the input is the k closest training instances in the feature space [11].

SVM (Support Vector Machine): SVM is a supervised algorithm for classification and regression, it is based on projecting a hyperplane to categorize input data [12].

RF (Random Forest): the random forest is a supervised algorithm for classification and regression based on building multiple decision trees [13].

After passing the first round with all these algorithms, the one with the best performance will be selected. We will apply several technics for attribute/data selection like eliminating values with variance close to zero, attributes highly correlated and feature selection; results will be analysed ad, after this, we will use the selected algorithm with bagging and boosting technics to try to get the most of it.

3. Methodology

3.1. Hypothesis

We hypothesize that given the actual data stored by TRACES and related systems (the Data Ware House system for instance), it is possible to predict if a consignment (stored as a “certificate” object) should be rejected beforehand. Variables/Fields might affect fraud/rejection could be: time, type of transport, country of origin, consignment (id, type), etc.

3.2. Process

The whole process of creating the proof of concept is divided into three main activities:

1. **Problem Analysis**: of the system, data, relations and relevant information to extract key data. It is **paramount to provide meaningful data**, i.e., aligned with our main objective, to the model in order find a solution at all.
2. **Data mining and Pre-processing**: Cleaning, parsing, filtering, aggregation and other transformation operations over data. Almost as important as meaningful data is **the quality of such data**; quality will allow us to achieve, within the limits of the problem, not only an effective (high recall) model but an efficient one (high precision).
3. **Algorithm Training (modeling)**: application of several machine algorithms with an iterative approach while modifying the data set and tuning algorithm parameters. The last phase focuses on finding the appropriate algorithm and tuning the learning parameters.

This three-activities process is a short version of the de-facto standard process: *Cross Industry Standard Process for Data Mining* [14]:

Figure 2: CRISP-DM phases <REF>

We will pay attention to the most important part of above process: *business understanding* <ref to CRISP again> in section 4.2; there are also loops between activities that will not be shown in this work, for the sake of clarity, but they have happened. Obviously, since this work is a proof of concept, we will not perform the last phase: deployment. Data understanding is shown in section 4.3 and Data Preprocessing in section 4.4. Modeling and Evaluation will be extensively covered in section 4.5 and section 4.6.

4. Proof of Concept development

4.1. Used Technologies and Tools

The following technologies were used in order to conduct above methodology:

- Oracle SQL Developer to connect relational database TRACES and data ware house of TRACES.
- MariaDB to create an aggregated table.
- Microsoft Visio to perform business analysis.
- Microsoft Excel to perform field analysis.
- Microsoft Word to write the thesis
- R and R-Studio.

4.2. Business Understanding.

4.2.1. Introduction

As already explained, before extracting any data, it is utterly important to perform an analysis of available data, and business logic of the application have been carried out. Any attempt of dumping data without a deep knowledge of the domain logic and the produced data will render useless results.

This section will analyse available information regarding TRACES and the data it produces.

4.2.2. Certificate CVEDP creation workflow

First of all, it is important to know the main workflow of how a CVEDP certificate is created and the data that a certificate contains.

For the sake of simplicity, the full reasoning of selecting specific fields is available in Annex I, where we can find a table with all fields belonging to Part I and Part II of CVEDP certificate. The table shows the decision of taking or eliminating certificate fields.

The picture below shows the CVEDP certificate workflow creation and validation; it is a brief example of some of the screens of the real application:

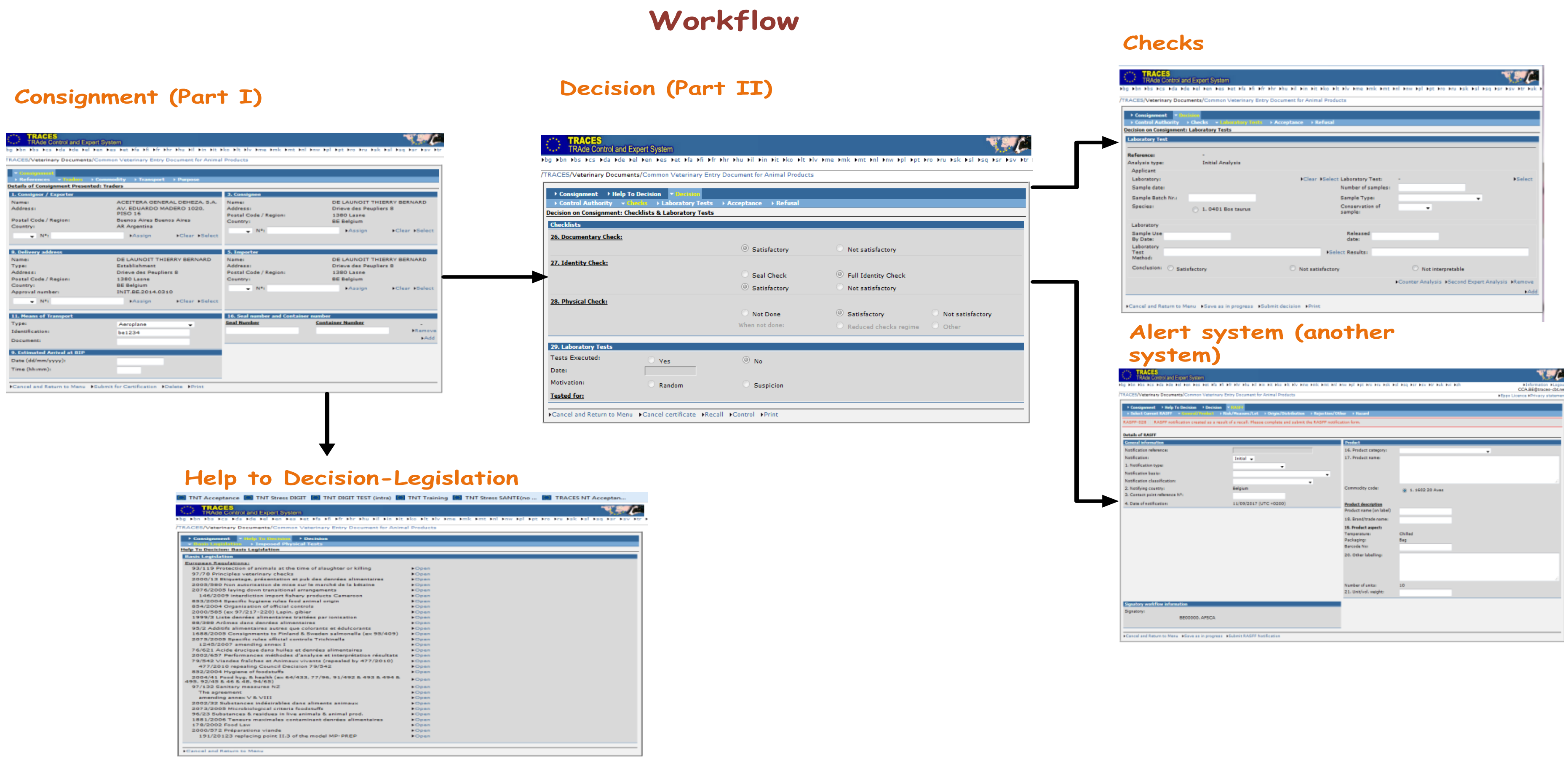


Figure 3.: CVEDP certificate workflow

Consignment, Part I: the economic operator or authority that is creating the certificate will provide mandatory and optional data related to the consignment. At this stage, we have the attributes that we will analyse later on**.**

Decision, Part II: Authority in charge, border control post at customs point will decide if the consignment can be released for free circulation (validation of certificate) or if, on the other hand, will be rejected for any reason.

Following, Figure 1 shows Business Process Model Notation Diagram, more fit for our purposes. This diagram shows the workflow to create a CVEDP certificate with the different actors that can participate in it.

The green process workflows are the one we are interested. This is because to be sure that a consignment is a valid one, we will take only the instances that have passed all mandatory checks (documentary and identity) and no mandatory ones (physical checks). The reason behind this decision will be explained later in this document.

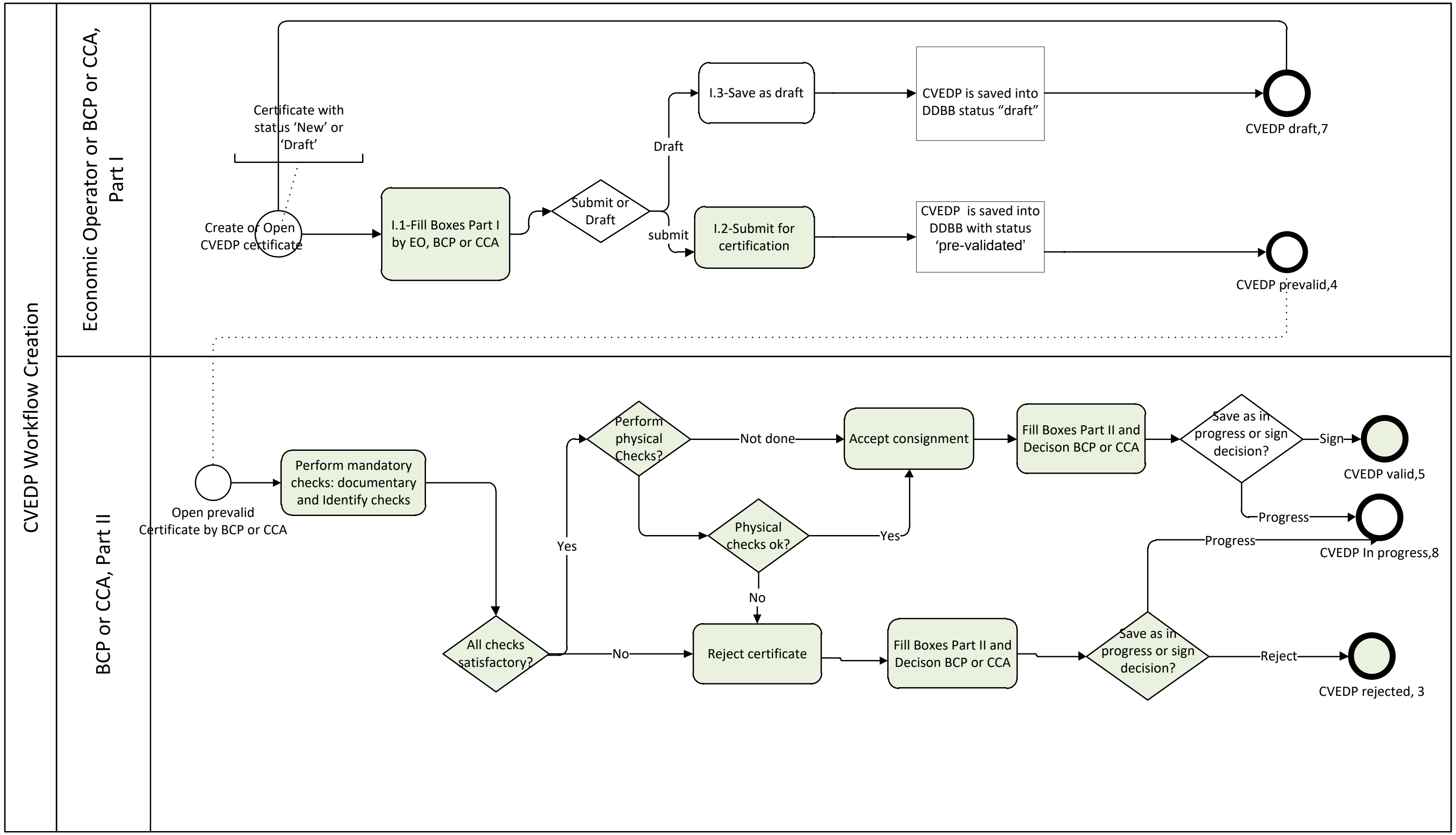


Figure 4: CVEDP certificate processes.

Now that we know how to produce a certificate, let's define which subset of certificates will be collected:

1. Certificates that have been created by authorities or operators and have been validated (**status = 5**) by authorities passing all checks: documentary, identity and physical (**physical check** = **1**).
2. Certificates that have been created by authorities or operators that have been rejected (**status = 3**) by authorities. **For rejection, the physical check is not needed**.

In next sections, an explanation of why these and no other certificates have been selected will follow, before we must understand what a control is.

4.2.3. Certificate Status

For our goal, only data belonging to the first part of the certificate, Part I, is useful and can be analysed to get advice for a new consignment that is intended to be imported into Europe (more info in “Annex I”). Data belonging to Part II and checks of the certificate will help us to validate our model/s and to classify our certificates into two categories: Valid and Not Valid certificate(Accepted / Rejected).
A certificate can have only one status at a given time, but there are eleven possible status :

- 0 = not set
- 1 = new
- 2 = deleted
- 3 = rejected
- 4 = pre-validated
- 5 = valid
- 6 = cancelled
- 7 = draft
- 8 = in progress
- 9 = animo
- 10 = recalled
- 11 = replaced

The workflow to pass from one to another status is as follows; I have shown only the most important status for our purpose, being green colour means an intermediate status and red colour, final status. When passing to the red colour is when the algorithm must give advice; saying if the goods must be controlled or not at customs points.

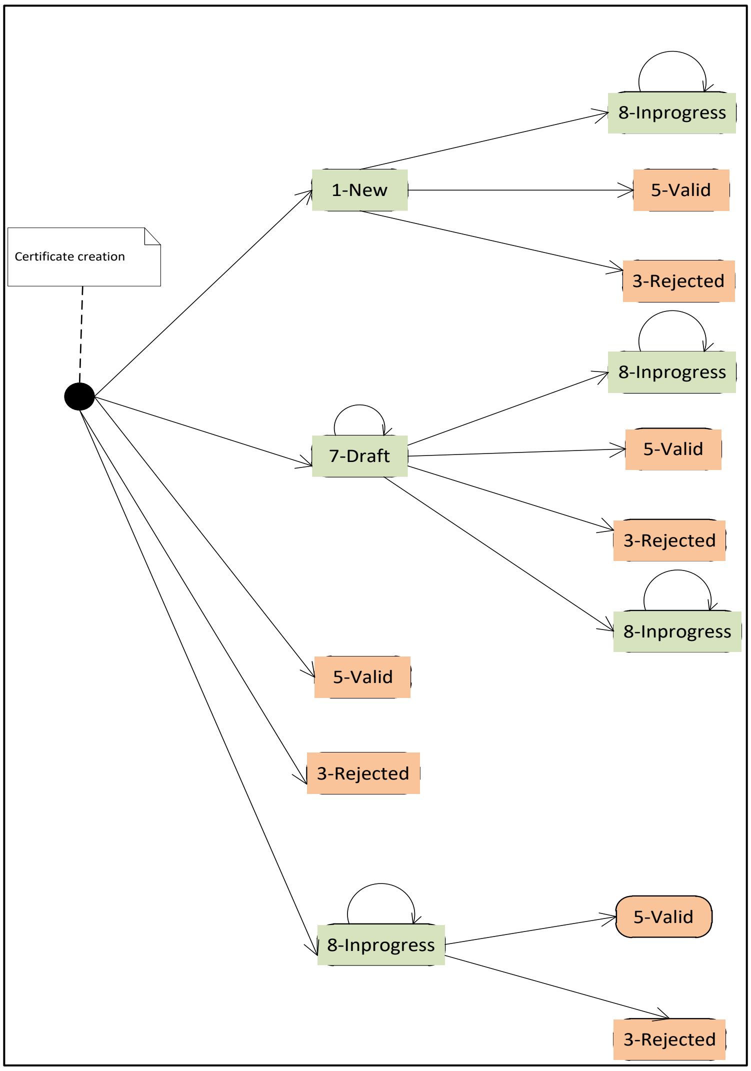


Figure 5: Relevant states for a certificate.

Valid is when a certificate has been signed and validated. Some of those certificates have been subject to controls while others not.

Within a valid certificate we can have several subcategories/purposes:

- Transhipment: consignment arrives by plane or ship to one country, but the final destination is another country at EU level. It is needed the creation of new certificate for the new Border Inspection Post country.
- Transit: when a consignment passes through one or several countries inside of EU area by train, road, etc. Final destination could be any country.
- Internal Market: Intended to be released for free circulation in that country.

- If channelled
- Specific warehouse procedure: customs, direct to a ship, free zone, ship supplier

A certificate with transshipment or transit purpose can be less likely to be controlled that one that is intended for internal market, so, I strongly believe that valid certificates which purpose is not intended for internal market are less likely to be controlled at customs points since they are not going to enter the country. Although this field will be included as a predictor, we will corroborate this during the data filtering activity.

Rejected, within rejected class/category we can have several sub-categories of rejection:

A certificate can be rejected indicating one of the following reasons:

1. Absence/Invalid certificate

2. Non approved country Country:

3. Non approved establishment

4. Prohibited product

5. ID: Mis-match with documents

5.a Invasive alien species

6. ID: Health mark error

7. Physical hygiene failure

8. Chemical contamination

9. Micro biological contamination

10. Other

11. Other, create RASFF notification

We do not have any fields that indicate if a certificate/consignment is fraudulent or not *per se*. The term fraudulent must be interpreted as non-conformant (to law), to avoid including legit consignments that are rejected by simple mistakes or formalities, So we will define fraudulent, rejection, based on the controls and checks that the consignment has been through.

4.2.4. Controls/Checks at the Border Control Post

Checks and controls can be performed at the Europe border control post (BCP), not all goods and consignments can be controlled at the border. Therefore, only the ones with potentially high risks will be checked; also, European legislation establishes a minimum number of checks to be performed at the border entry. Following, a list of controls performed at BCPs:

- **Documentary checks** concerns to the mandatory verification of health certificates or documents accompanying the consignment. These checks are mandatory for all animal products entering in Europe from third countries.
- **Identity checks,** this box concerns checking consistency between the accompanying health certificates or documents and the consignment presented at the EU BIP/DPE/DPI. It is a check by visual inspection to ensure compliance with EU legislation.
- **Physical checks**, 'physical check' means a check on the product itself, which may include checks on packaging and temperature and also sampling and laboratory testing. The aim of the physical check on animal products "*…is to ensure that the products still meet the purpose mentioned in the veterinary certificate or document: the guarantees of origin certified by the third country must accordingly be verified while ensuring that the subsequent transport of the product has not altered the original guaranteed condition, by means of* :
    - *sensory examinations: smell, color, consistency, taste;*
    - *simple physical or chemical tests: cutting, thawing, cooking;*
    - *laboratory tests to detect:*
        - *residues*
        - *pathogens*
        - *contaminants*
        - *observations of alteration"*

  [15]

Physical checks are not always mandatory for CVEDP certificates, so these checks/controls are based on:

- **EC legislation**: legislation establishes the basis of these controls. % of controls that might be performed based on country of origin and kind of product.
- **Personal suspicion of the BIP authority**: when they see there is something wrong or suspicious.
- **Random!** These controls should be eliminated with the help of predictive models.

The next figure shows the different chained checks, each one being an addition over the previous one:

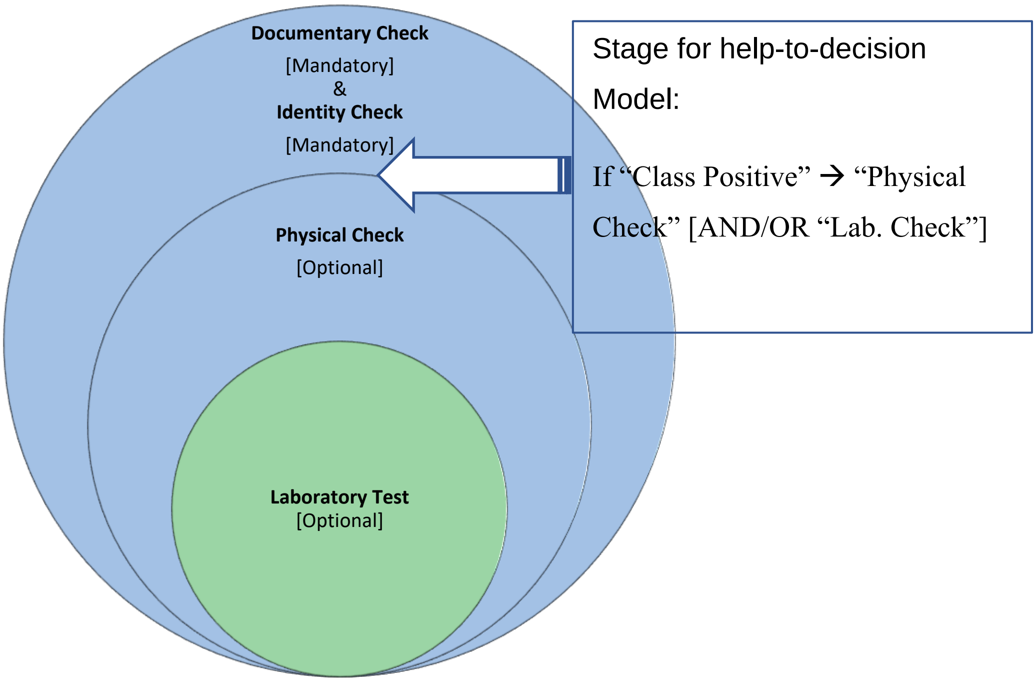


Figure 6: Ttypes of checks on certificates.

Based on the different types of status, nine different ones based on checks passed by a certificate, we define our two classes of our future classification (binary) model as:

Class "Rejected" (positive class): When Cert. STATUS = **3**. → **Phys. Check** Rejected.

Class "Accepted" (negative class): When Cert. STATUS = **5**. → (**Doc. Check** AND **Identity Check** AND **Physical Check**) Accepted.

So in our positive class *Rejected* we have all instances that have been rejected by a physical check (and above since laboratory test is a subset). This our definition of fraudulent.

In our negative class *Accepted,* to be sure we are taking true positives, we take all instances **that at least** have passed the first three test/check.

Why not going further with the check? Wouldn't we have a higher assurance of having a true positive?

Forcing to go down to the third check level (Physical Check) already provides a high assurance: the three checks are of different nature; it is not a perfect assurance, but we cannot go further.

We cannot include the fourth check (Lab. Test) since we would be excluding the different type of products that are less likely to be checked by a laboratory test (e.g. animal fur versus animal food). Including the fourth check implies analysing only a subset of products to be imported, not all of them.

4.2.5. Consignments and CN codes

CN stands for Combined Nomenclature codes. The Combined Nomenclature "..*is a tool for classifying goods, set up to meet the requirements both of the Common Customs Tariff and of the EU's external trade statistics. The CN is also used in intra-EU trade statistics. It is a further development (with special EU-specific subdivisions) of the World Customs Organization's Harmonized System nomenclature. This is a systematic list of commodities applied by most trading nations (and also used for international trade negotiations)*" [16].

One CVEDP certificate can contain several CN codes (in one consignment). In a nutshell, CN codes, or complements, are the products in a consignment.

This is a special attribute we will consider as well; I believe the type of product and the number of them affect are important ones. Since there are many complements per certificate, we will have to de-aggregate multiple rows (as many as complements) per certificate in columns.

4.2.6. Customs, Border Inspection Post

There is the possibility of basing our predictive models exclusively at the Border Inspection Post, so we would develop as many models as BIPs. At the entry of Europe, every custom can have a different way to control consignments and this can affect the number of controls performed over consignment and the rate of rejected consignment. I leave this possibility out of this proof of concept; we will focus at the country level.

4.2.7. Country

The same logic of the Border Inspection Post can apply to country level and, after analysing the existing data, I have decided to base the model in one country in order to reduce the amount of data to work with.

4.3. Data Acquisition and Fields selection

At this stage data extraction from the database has been performed; the analysis of all fields selected can be found on "

Annex I. Used certificate fields". Fields that are not mandatory are very likely to be deleted since most of them will not have any data and this can distort or add noise to the results.

There are several tables needed to be queried and to get all necessary data. Following a brief explanation of some of them

Certificate: the main table where most of the information related to the certificate (CVEDP) is stored: date, reference number, importer id, consignee id, consignor id, the person in charge of signing the certificate (authority), the person responsible for the load, etc.

Authority: This information is related to the authority in charge of validating a certificate.

Business: There many businesses involve into one certificate: importer, consignor, consignee, responsible for the load, place where the consignment is going to be delivered, transporter, etc.

Complements: Products in a certificate. Min 1-Max. X

Decision: When a consignment is signed by the authority. Validated, rejected, etc..

Data is stored in two different databases, so an intermediate temporary table/s have been created (separate database, MariaDB [17]) to ease the process of data extraction and to connect data from both data stores.

Following a brief database model diagram can be found of both:

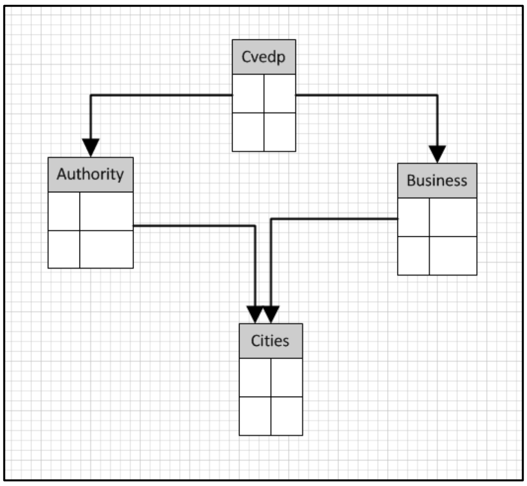


Figure 7: Relational Data Base

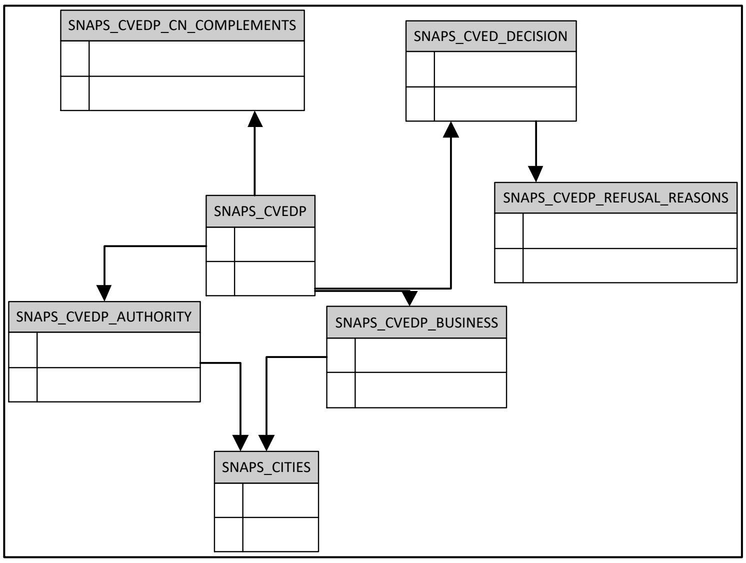


Figure 8: Data Ware House

Once we have all records extracted from both stores, we will face a couple of problems with table's structure.

As already commented, every certificate can have several status/records before reaching a final status; those final statuses can be rejected (3) or valid (5).

So the output can be something similar as follows:

Certificate_Id	Consignee_Id	Consignee_Name	Consignee_Country	Consignor_Id	Consignoor_Name	Status
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	1, new
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	2, draft
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	5, valid

We see that differences between these records are quite small, most of the times, the only status changes across many fields (around 70 fields). Events do not represent changes in the reality but just changes in the system when the data is introduced; for instance, a certificate could have been through different status but only when this data is introduced in the systems (end points are not forced to introduce the data in a time-consistent manner) is when the status of the certificate changes (in the data ware house). **So, at this stage of the project, I consider that information regarding changes to a certificate along his life cycle cannot be useful**. The system, TRACES, needs to be modified to represent real changes faithfully. Therefore we have to discard algorithms with a temporal component (e.g. recurrent neural networks).

After carrying out an investigation and speaking with business, I can conclude that this situation (lack of information between certificates with several statuses) can be explained by two main factors:

- **Many certificates are done at once**. The authority enters the application, save a certificate as new, submit for certification and validate the certificate in one step. So there are three certificate's statuses, but the information on all three is identical.
- **Many certificates are uploaded through web services**. So, the result is the same as before.

Conclusion: As per certificate's status, we will keep the last status snap certificate (latest state of the certificate in the data warehouse).

Another complication with tables structure is that each certificate can have several Complement Codes; this means that one certificate can have several consignments/products associated with it.(i.e.: chicken meat, fish and carrots, these are 3 different complement codes).

If we take as an example the previous certificate and we imagine that this certificate has associated two complement codes, then, the result will be something similar to the following table:

Certificate_Id	Consignee_Id	Consignee_Name	Consignee_Country	Consignor_Id	Consignoor_Name	Status	Complement
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	1, new	256637 pork meet
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	1, new	568933 frozen fish

12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	2, draft	256637 pork meet
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	2, draft	568933 frozen fish
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	5, valid	256637 pork meet
12345	33224351	Friedrich Wilhelm Lübbert GmbH	DE	33224352	Yantai Jiahong Food Co., LTD.	5, valid	568933 frozen fish

This structure needs to be parsed later to get only one row for each certificate. We will use R [18].

We define here what are going to be our two classes with the selected data in order to filter all data available in two different datasets (class positive and class negative).

Class "positive": A German certificate (entry authority is based in Germany) with a rejected "certificate status" created after 2007-01-01 (included) and with a physical check passed.

Class "negative": A German certificate (entry authority is based in Germany) with a valid "certificate status" created after 2007-01-01 (included) and with the following positive controls: physical, documentary and identity.

The reasoning behind this is that Germany, usually, inserts data with high-quality TRACES system.

While analysing data, a bug has been discovered hidden in the system for several years: there are several certificates that it is marked as nonconforming with legislation but as final status has valid value and purpose "internal market" or "for human consumption". This should not be allowed by the system, the number of instances with this problem is not many (order of magnitude of 100), so we will analyse if this can have an impact on the results.

The number of instances in the "positive" class is around 13.000.

The number of instances in the "negative" class is around 210.000.

4.4. Data consolidation, aggregation and preparation

Once data has been extracted, we will perform a series of consistency tests on the data.

There are many tables involved and two databases, so few tests to guaranty the integrity of the collected data are needed.

The test consists on:

- Retrieve data from the temporal table created and compare the results with original tables (See used data in Annex II)
- Several queries to verify that the business logic corresponds to the data that we have retrieved from the databases.

4.4.1. Importing Data (CSV) into R

Once the selected data is stored in a CSV file, we will load it into R Studio [19] to create a dataset and do the necessary transformation to it.

Since the data comes from more than eighty countries and the application itself gives freedom to all operator to be quite "creative" introducing data (free text fields), many fields contain semicolons and commas (apart from new lines that have been stripped out when exporting from the database). So the comma separated values (CSV) file uses this three characters as separator "#$#". R Studio cannot manage this type of separators, so a little transform has been done to the data before being loaded to R Studio (all semicolons have been replaced with commas):

```
~cat DE_2011-2017_with_Complement.csv | sed 's/;/,/g' | sed 's/\#\$\#/;/g' > DE_2011-2017_with_Complement.data
```

R Studio does not manage UTF-8 files properly and, with more than eighty countries, there are lots of non-ASCII characters, so I will be using package "readr" to load the csv file into R Studio:

```
> dim(datos)
[1] 227587     73
```

We have 73 attributes (75 predictors and one class) and around 225.000 instances, before further transformations we must normalize nulls, empties and alike values, i.e., everything meaning empty (there are zeros meaning empty), boolean values (zeros meaning FALSE), dates , and "no values" (inspecting the data we can find many values that implies "no value").

However, before doing any transformation to this dataset, we must "shape" the data in a valid form for the (binary) classification machine learning algorithm. This shaping implies to have only one row per instance where values are rows and attributes (variables) are headers:

Attr1	Attr2	...	AttrN	Class
Cert1	xxx	Xxx	xxx	TRUE
Cert2	xxx	Xxx	xxx	FALSE
...	...	...	...	...
CertN	xxx	Xxx	xxx	TRUE

Data exported from the aggregated database contains multiple rows per certificate since each certificate contains one or more complements (one consignment might contain several products, not just one).

Inspecting the cardinality of certificates for different complements, we find our dataset contains roughly 227.000 rows with a maximum of 11 complements per certificate:

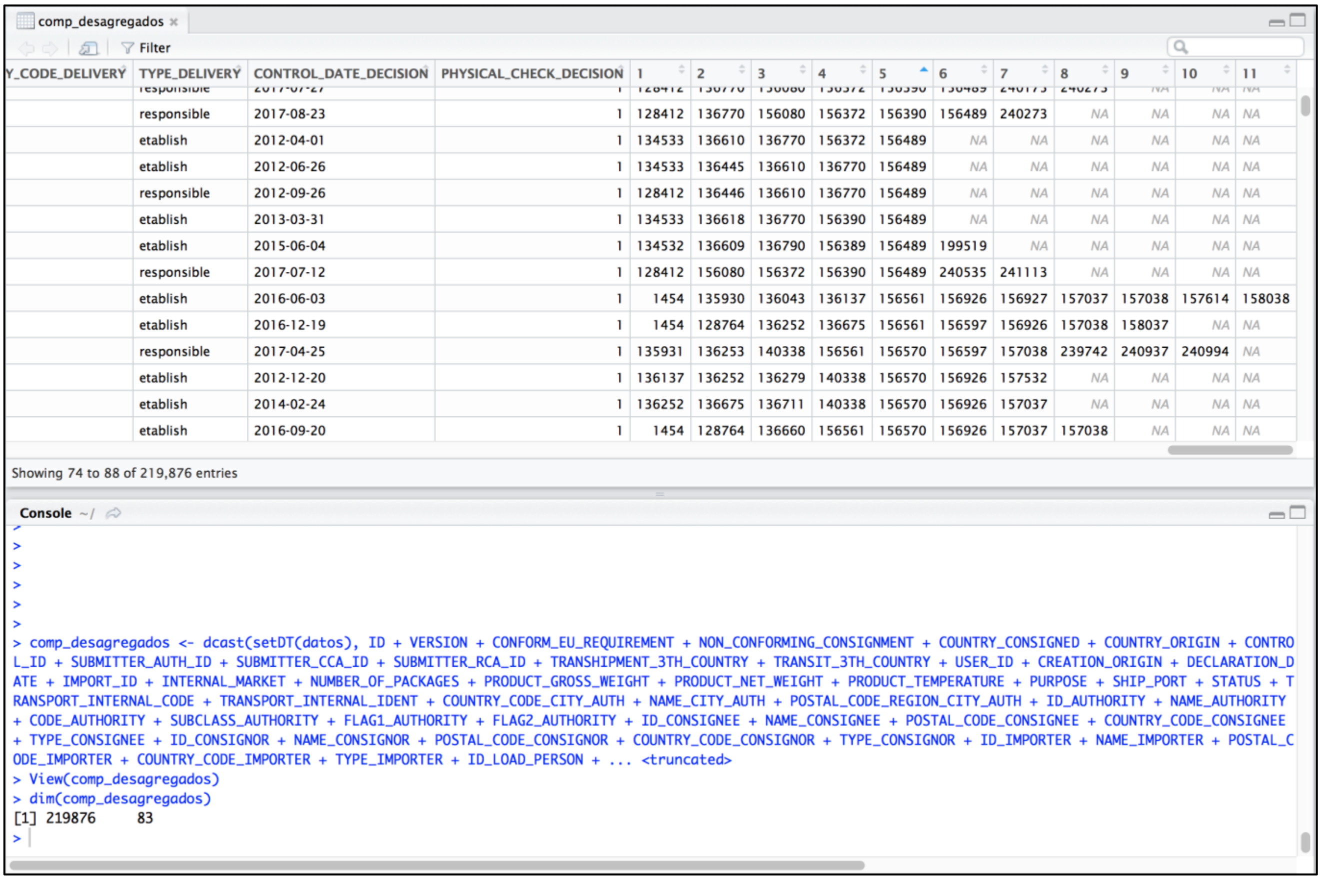
```
> list(datos)
[[1]]
# A tibble: 227,587 x 73
        ID VERSION CONFORM_EU_REQUIREMENT NON_CONFORMING_CONSIGNMENT COUNTRY_CONSIGNED COUNTRY_ORIGIN CONTROL_ID SUBMITTER_AUTH_ID SUBMITTER_CCA_ID SUBMITTER_RCA_ID TRANSHIPMENT_3TH_COUNTRY
     <int>   <int>                  <int>                      <chr>             <chr>          <chr>      <int>             <int>            <int>            <int>                    <chr>
1  2821187       1                      1                       <NA>                CN             CN         NA              1403             1000               NA                     <NA>
2  2821616       1                      1                       <NA>                JP             JP         NA              1403             1000               NA                     <NA>
3  2821694       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
4  2821695       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
5  2821696       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
6  2821697       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
7  2821698       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
8  2821699       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
9  2821740       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
10 2821741       1                      1                       <NA>                US             US         NA              1403             1000               NA                     <NA>
# ... with 227,577 more rows, and 62 more variables: TRANSIT_3TH_COUNTRY <chr>, USER_ID <int>, CREATION_ORIGIN <chr>, DECLARATION_DATE <date>, IMPORT_ID <int>, INTERNAL_MARKET <chr>,
#   NUMBER_OF_PACKAGES <int>, PRODUCT_GROSS_WEIGHT <dbl>, PRODUCT_NET_WEIGHT <dbl>, PRODUCT_TEMPERATURE <chr>, PURPOSE <chr>, SHIP_PORT <chr>, STATUS <int>, TRANSPORT_INTERNAL_CODE <chr>,
#   TRANSPORT_INTERNAL_IDENT <chr>, COUNTRY_CODE_CITY_AUTH <chr>, NAME_CITY_AUTH <chr>, POSTAL_CODE_REGION_CITY_AUTH <chr>, ID_AUTHORITY <int>, NAME_AUTHORITY <chr>, CODE_AUTHORITY <chr>,
#   SUBCLASS_AUTHORITY <chr>, FLAG1_AUTHORITY <int>, FLAG2_AUTHORITY <int>, ID_CONSIGNEE <int>, NAME_CONSIGNEE <chr>, POSTAL_CODE_CONSIGNEE <chr>, COUNTRY_CODE_CONSIGNEE <chr>,
#   TYPE_CONSIGNEE <chr>, ID_CONSIGNOR <int>, NAME_CONSIGNOR <chr>, POSTAL_CODE_CONSIGNOR <chr>, COUNTRY_CODE_CONSIGNOR <chr>, TYPE_CONSIGNOR <chr>, ID_IMPORTER <int>, NAME_IMPORTER <chr>,
#   POSTAL_CODE_IMPORTER <chr>, COUNTRY_CODE_IMPORTER <chr>, TYPE_IMPORTER <chr>, ID_LOAD_PERSON <int>, NAME_LOAD_PERSON <chr>, POSTAL_CODE_LOAD_PERSON <chr>, COUNTRY_CODE_LOAD_PERSON <chr>,
#   TYPE_LOAD_PERSON <chr>, ID_SUBMITTER_BUS <int>, NAME_SUBMITTER_BUS <chr>, POSTAL_CODE_SUBMITTER_BUS <chr>, COUNTRY_CODE_SUBMITTER_BUS <chr>, TYPE_SUBMITTER_BUS <chr>, ID_TRANSPORTER <int>,
#   NAME_TRANSPORTER <chr>, POSTAL_CODE_TRANSPORTER <chr>, COUNTRY_CODE_TRANSPORTER <chr>, TYPE_TRANSPORTER <chr>, DELIVERY_ID <int>, NAME_DELIVERY <chr>, POSTAL_CODE_DELIVERY <chr>,
#   COUNTRY_CODE_DELIVERY <chr>, TYPE_DELIVERY <chr>, CONTROL_DATE_DECISION <date>, PHYSICAL_CHECK_DECISION <int>, COMMODITY_COMPLEMENT_ID <int>
```

Figure 9: Certiticate's complements aggregated

```
> summary(subset(ddply(datos,~ID,summarise,"rep"=length(ID)),rep>1)$rep)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   2.000   2.423   3.000  11.000
```

The last table shows that one ID is duplicated 11 times, this means that this ID has associated 11 different Complement Codes (products). So we should include all these complements in one single row.

To transform repeated rows into new columns, we will cast the data frame into a dataset (data.table library) and then apply dcast (reshape library):



comp_desagregados

Y_CODE_DELIVERY	TYPE_DELIVERY	CONTROL_DATE_DECISION	PHYSICAL_CHECK_DECISION	1	2	3	4	5	6	7	8	9	10	11
	responsible	2017-08-23	1	128412	136770	156080	156372	156390	156489	240273	NA	NA	NA	NA
	etablish	2012-04-01	1	134533	136610	136770	156372	156489	NA	NA	NA	NA	NA	NA
	etablish	2012-06-26	1	134533	136445	136610	136770	156489	NA	NA	NA	NA	NA	NA
	responsible	2012-09-26	1	128412	136446	136610	136770	156489	NA	NA	NA	NA	NA	NA
	etablish	2013-03-31	1	134533	136618	136770	156390	156489	NA	NA	NA	NA	NA	NA
	etablish	2015-06-04	1	134532	136609	136790	156389	156489	199519	NA	NA	NA	NA	NA
	responsible	2017-07-12	1	128412	156080	156372	156390	156489	240535	241113	NA	NA	NA	NA
	etablish	2016-06-03	1	1454	135930	136043	136137	156561	156926	156927	157037	157038	157614	158038
	etablish	2016-12-19	1	1454	128764	136252	136675	156561	156597	156926	157038	158037	NA	NA
	responsible	2017-04-25	1	135931	136253	140338	156561	156570	156597	157038	239742	240937	240994	NA
	etablish	2012-12-20	1	136137	136252	136279	140338	156570	156926	157532	NA	NA	NA	NA
	etablish	2014-02-24	1	136252	136675	136711	140338	156570	156926	157037	NA	NA	NA	NA
	etablish	2016-09-20	1	1454	128764	136660	156561	156570	156926	157037	157038	NA	NA	NA

Showing 74 to 88 of 219,876 entries

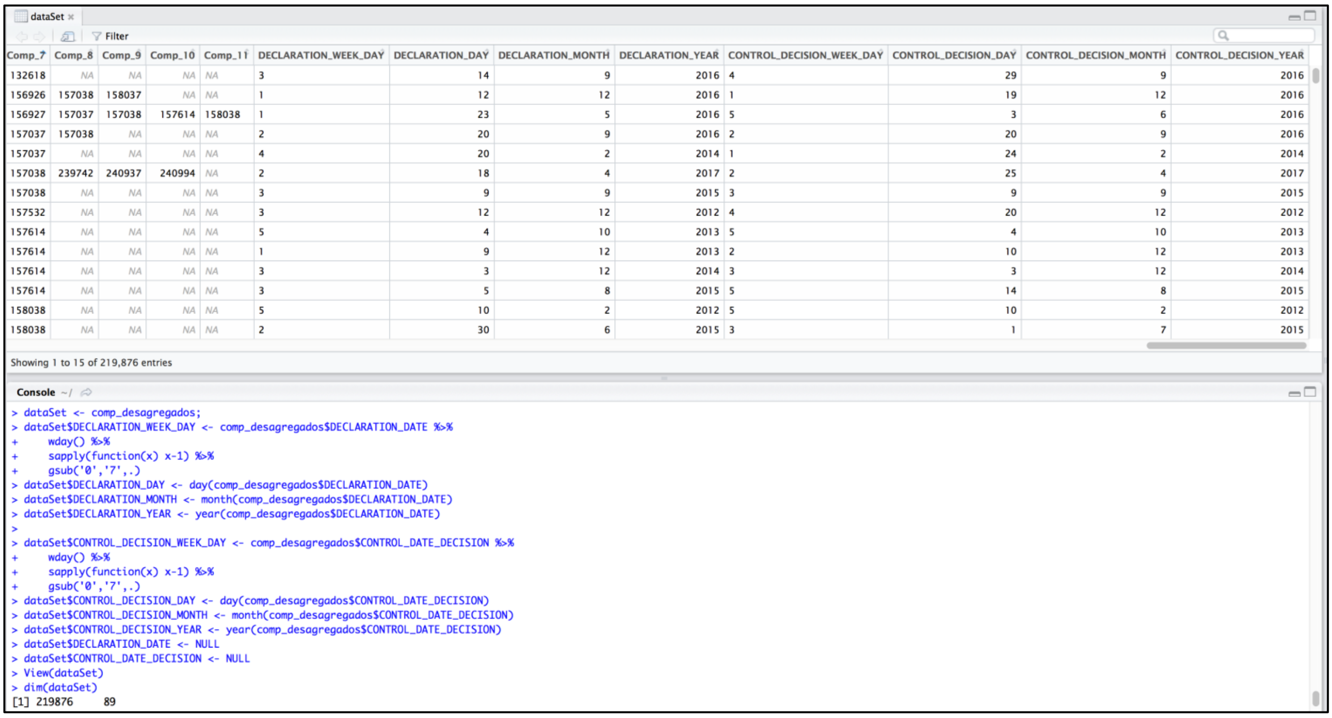
Console ~/

```
>
>
>
>
>
> comp_desagregados <- dcast(setDT(datos), ID + VERSION + CONFORM_EU_REQUIREMENT + NON_CONFORMING_CONSIGNMENT + COUNTRY_CONSIGNED + COUNTRY_ORIGIN + CONTRO
L_ID + SUBMITTER_AUTH_ID + SUBMITTER_CCA_ID + SUBMITTER_RCA_ID + TRANSHIPMENT_3TH_COUNTRY + TRANSIT_3TH_COUNTRY + USER_ID + CREATION_ORIGIN + DECLARATION_D
ATE + IMPORT_ID + INTERNAL_MARKET + NUMBER_OF_PACKAGES + PRODUCT_GROSS_WEIGHT + PRODUCT_NET_WEIGHT + PRODUCT_TEMPERATURE + PURPOSE + SHIP_PORT + STATUS + T
RANSPORT_INTERNAL_CODE + TRANSPORT_INTERNAL_IDENT + COUNTRY_CODE_CITY_AUTH + NAME_CITY_AUTH + POSTAL_CODE_REGION_CITY_AUTH + ID_AUTHORITY + NAME_AUTHORITY
+ CODE_AUTHORITY + SUBCLASS_AUTHORITY + FLAG1_AUTHORITY + FLAG2_AUTHORITY + ID_CONSIGNEE + NAME_CONSIGNEE + POSTAL_CODE_CONSIGNEE + COUNTRY_CODE_CONSIGNEE
+ TYPE_CONSIGNEE + ID_CONSIGNOR + NAME_CONSIGNOR + POSTAL_CODE_CONSIGNOR + COUNTRY_CODE_CONSIGNOR + TYPE_CONSIGNOR + ID_IMPORTER + NAME_IMPORTER + POSTAL_C
ODE_IMPORTER + COUNTRY_CODE_IMPORTER + TYPE_IMPORTER + ID_LOAD_PERSON + ... <truncated>
> View(comp_desagregados)
> dim(comp_desagregados)
[1] 219876     83
>
```

Figure 10: Certificate's complements de-aggregated

Now the dataset (comp_desagragados) contains 83 attributes, and the number of rows is reduced to 220113. It is important to cast back the data.table to a dataframe, or we will face issues when training the machine learning models.

Now we can easily transform dates into useful attributes: week day, month day, month, week of the month and year. This will add for attributes more for date attribute; we have two dates so there will be eight more attributes. Notice we are setting Monday as the first day of the week, doing so the Euclidean distance between Sunday and Monday is 6, so there is a clear difference between weekend (actually Sunday) and working days (useful for certain machine learning algorithm)



dataSet

Comp_7	Comp_8	Comp_9	Comp_10	Comp_11	DECLARATION_WEEK_DAY	DECLARATION_DAY	DECLARATION_MONTH	DECLARATION_YEAR	CONTROL_DECISION_WEEK_DAY	CONTROL_DECISION_DAY	CONTROL_DECISION_MONTH	CONTROL_DECISION_YEAR
132618	NA	NA	NA	NA	3	14	9	2016	4	29	9	2016
156926	157038	158037	NA	NA	1	12	12	2016	1	19	12	2016
156927	157037	157038	157614	158038	1	23	5	2016	5	3	6	2016
157037	157038	NA	NA	NA	2	20	9	2016	2	20	9	2016
157037	NA	NA	NA	NA	4	20	2	2014	1	24	2	2014
157038	239742	240937	240994	NA	2	18	4	2017	2	25	4	2017
157038	NA	NA	NA	NA	3	9	9	2015	3	9	9	2015
157532	NA	NA	NA	NA	3	12	12	2012	4	20	12	2012
157614	NA	NA	NA	NA	5	4	10	2013	5	4	10	2013
157614	NA	NA	NA	NA	1	9	12	2013	2	10	12	2013
157614	NA	NA	NA	NA	3	3	12	2014	3	3	12	2014
157614	NA	NA	NA	NA	3	5	8	2015	5	14	8	2015
158038	NA	NA	NA	NA	5	10	2	2012	5	10	2	2012
158038	NA	NA	NA	NA	2	30	6	2015	3	1	7	2015

Showing 1 to 15 of 219,876 entries

Console ~/

```
> dataSet <- comp_desagregados;
> dataSet$DECLARATION_WEEK_DAY <- comp_desagregados$DECLARATION_DATE %>%
+     wday() %>%
+     sapply(function(x) x-1) %>%
+     gsub('0','7',.)
> dataSet$DECLARATION_DAY <- day(comp_desagregados$DECLARATION_DATE)
> dataSet$DECLARATION_MONTH <- month(comp_desagregados$DECLARATION_DATE)
> dataSet$DECLARATION_YEAR <- year(comp_desagregados$DECLARATION_DATE)
>
> dataSet$CONTROL_DECISION_WEEK_DAY <- comp_desagregados$CONTROL_DATE_DECISION %>%
+     wday() %>%
+     sapply(function(x) x-1) %>%
+     gsub('0','7',.)
> dataSet$CONTROL_DECISION_DAY <- day(comp_desagregados$CONTROL_DATE_DECISION)
> dataSet$CONTROL_DECISION_MONTH <- month(comp_desagregados$CONTROL_DATE_DECISION)
> dataSet$CONTROL_DECISION_YEAR <- year(comp_desagregados$CONTROL_DATE_DECISION)
> dataSet$DECLARATION_DATE <- NULL
> dataSet$CONTROL_DATE_DECISION <- NULL
> View(dataSet)
> dim(dataSet)
[1] 219876     89
```

Figure 11: Certificate's dates de-aggregated

As the figure above shows, we have the same number of instances but with 89 attributes now (general date attributes have been deleted). We also renamed de-aggregated attributes with names 1, 2, 3 .. 11, to Comp_1, Comp_2, etc.

We will transform all characters to numeric, to do so first we will convert them from characters to factors and then to numeric. An improvement to this could be to do a hash of the string. This way distances from each "factor" are distributed randomly (nearly). But we will keep it simple and get factors (instead of hashed numbers)

```
> class(dataSet$NON_CONFORMING_CONSIGNMENT)
[1] "factor"
> dataSet$NON_CONFORMING_CONSIGNMENT <- sapply(dataSet$NON_CONFORMING_CONSIGNMENT,as.numeric)
> class(dataSet$NON_CONFORMING_CONSIGNMENT)
[1] "numeric"
> unique(dataSet$NON_CONFORMING_CONSIGNMENT)
[1] NA  1  2  3  4
```

We apply this transformation to all character attributes:

```
> c <- which(sapply(dataSet,function (x) is.character(x)))
> f <- function(x) as.numeric(as.factor(x))
> dataSet[,c] <- sapply(dataSet[,c],f)
```

Now all attributes are numeric:

```
> dataSet[,55:59] <- NULL
> sapply(dataSet[1,],class)
                      ID                  VERSION    CONFORM_EU_REQUIREMENT NON_CONFORMING_CONSIGNMENT        COUNTRY_CONSIGNED           COUNTRY_ORIGIN               CONTROL_ID        SUBMITTER_AUTH_ID
               "integer"                "integer"                 "integer"                  "numeric"                "numeric"                "numeric"                "integer"                "integer"
        SUBMITTER_CCA_ID         SUBMITTER_RCA_ID  TRANSHIPMENT_3TH_COUNTRY        TRANSIT_3TH_COUNTRY                  USER_ID          CREATION_ORIGIN                IMPORT_ID          INTERNAL_MARKET
               "numeric"                "integer"                 "numeric"                  "numeric"                "integer"                "numeric"                "integer"                "numeric"
      NUMBER_OF_PACKAGES     PRODUCT_GROSS_WEIGHT        PRODUCT_NET_WEIGHT        PRODUCT_TEMPERATURE                  PURPOSE                SHIP_PORT                   STATUS  TRANSPORT_INTERNAL_CODE
               "integer"                "numeric"                 "numeric"                  "numeric"                "numeric"                "numeric"                "integer"                "numeric"
TRANSPORT_INTERNAL_IDENT   COUNTRY_CODE_CITY_AUTH            NAME_CITY_AUTH POSTAL_CODE_REGION_CITY_AUTH             ID_AUTHORITY           NAME_AUTHORITY           CODE_AUTHORITY       SUBCLASS_AUTHORITY
               "numeric"                "numeric"                 "numeric"                  "numeric"                "integer"                "numeric"                "numeric"                "numeric"
         FLAG1_AUTHORITY          FLAG2_AUTHORITY              ID_CONSIGNEE             NAME_CONSIGNEE    POSTAL_CODE_CONSIGNEE   COUNTRY_CODE_CONSIGNEE           TYPE_CONSIGNEE             ID_CONSIGNOR
               "integer"                "integer"                 "integer"                  "numeric"                "numeric"                "numeric"                "numeric"                "integer"
          NAME_CONSIGNOR    POSTAL_CODE_CONSIGNOR    COUNTRY_CODE_CONSIGNOR             TYPE_CONSIGNOR              ID_IMPORTER            NAME_IMPORTER     POSTAL_CODE_IMPORTER    COUNTRY_CODE_IMPORTER
               "numeric"                "numeric"                 "numeric"                  "numeric"                "integer"                "numeric"                "numeric"                "numeric"
           TYPE_IMPORTER           ID_LOAD_PERSON          NAME_LOAD_PERSON    POSTAL_CODE_LOAD_PERSON COUNTRY_CODE_LOAD_PERSON         TYPE_LOAD_PERSON           ID_TRANSPORTER         NAME_TRANSPORTER
               "numeric"                "integer"                 "numeric"                  "numeric"                "numeric"                "numeric"                "integer"                "numeric"
 POSTAL_CODE_TRANSPORTER COUNTRY_CODE_TRANSPORTER          TYPE_TRANSPORTER                DELIVERY_ID            NAME_DELIVERY     POSTAL_CODE_DELIVERY    COUNTRY_CODE_DELIVERY            TYPE_DELIVERY
               "numeric"                "numeric"                 "numeric"                  "integer"                "numeric"                "numeric"                "numeric"                "numeric"
 PHYSICAL_CHECK_DECISION                   Comp_1                    Comp_2                     Comp_3                   Comp_4                   Comp_5                   Comp_6                   Comp_7
               "integer"                "integer"                 "integer"                  "integer"                "integer"                "integer"                "integer"                "integer"
                  Comp_8                   Comp_9                   Comp_10                    Comp_11     DECLARATION_WEEK_DAY          DECLARATION_DAY        DECLARATION_MONTH         DECLARATION_YEAR
               "integer"                "integer"                 "integer"                  "integer"                "numeric"                "numeric"                "numeric"                "numeric"
CONTROL_DECISION_WEEK_DAY    CONTROL_DECISION_DAY    CONTROL_DECISION_MONTH      CONTROL_DECISION_YEAR
               "numeric"                "numeric"                 "numeric"                  "numeric"
>
```

Figure 12: Numeric Attributes.

Notice we have deleted attributes related to SUBMITTER_BUSINESS since there was no data, it is a bug in the TRACES system so we cannot use this information (it is available to other countries though, but not for the selected one, Germany).

If we also delete column ID (random attribute used to de-aggregate some rows) we have a dataset with 83 attributes:

```
> dataSet$ID <- NULL
> dim(dataSet)
[1] 219876     83
```

Missing data can have a big impact on modelling, so let's see how many missing values we have per attribute:

```
> sapply(dataSet, function(x) round(sum(is.na(x))/219876*100,digits = 2))
                VERSION   CONFORM_EU_REQUIREMENT   NON_CONFORMING_CONSIGNMENT        COUNTRY_CONSIGNED           COUNTRY_ORIGIN                CONTROL_ID       SUBMITTER_AUTH_ID
                   0.00                     0.00                        99.96                     2.16                     2.16                     96.51                    0.00
      SUBMITTER_CCA_ID         SUBMITTER_RCA_ID       TRANSHIPMENT_3TH_COUNTRY      TRANSIT_3TH_COUNTRY                  USER_ID           CREATION_ORIGIN               IMPORT_ID
                   0.00                    96.54                        99.62                    99.89                    76.73                     12.93                   97.31
       INTERNAL_MARKET       NUMBER_OF_PACKAGES         PRODUCT_GROSS_WEIGHT       PRODUCT_NET_WEIGHT      PRODUCT_TEMPERATURE                   PURPOSE               SHIP_PORT
                   0.71                     0.00                         0.00                     0.00                     0.34                      0.00                   99.99
                 STATUS  TRANSPORT_INTERNAL_CODE      TRANSPORT_INTERNAL_IDENT   COUNTRY_CODE_CITY_AUTH           NAME_CITY_AUTH POSTAL_CODE_REGION_CITY_AUTH       ID_AUTHORITY
                   0.00                     0.00                         0.00                     0.00                     0.00                      0.00                    0.00
        NAME_AUTHORITY           CODE_AUTHORITY           SUBCLASS_AUTHORITY          FLAG1_AUTHORITY          FLAG2_AUTHORITY              ID_CONSIGNEE          NAME_CONSIGNEE
                   0.00                     0.00                         0.00                     0.03                    96.57                      0.04                    0.04
  POSTAL_CODE_CONSIGNEE   COUNTRY_CODE_CONSIGNEE               TYPE_CONSIGNEE             ID_CONSIGNOR           NAME_CONSIGNOR     POSTAL_CODE_CONSIGNOR  COUNTRY_CODE_CONSIGNOR
                   0.24                     0.04                         0.04                     0.00                     0.00                      0.00                    2.16
         TYPE_CONSIGNOR              ID_IMPORTER                NAME_IMPORTER     POSTAL_CODE_IMPORTER    COUNTRY_CODE_IMPORTER             TYPE_IMPORTER          ID_LOAD_PERSON
                   0.00                     0.03                         0.03                     0.22                     0.03                      0.03                    0.00
       NAME_LOAD_PERSON  POSTAL_CODE_LOAD_PERSON    COUNTRY_CODE_LOAD_PERSON         TYPE_LOAD_PERSON           ID_TRANSPORTER          NAME_TRANSPORTER POSTAL_CODE_TRANSPORTER
                   0.00                     0.00                         0.00                     0.00                    98.40                     98.40                   98.40
COUNTRY_CODE_TRANSPORTER        TYPE_TRANSPORTER                  DELIVERY_ID            NAME_DELIVERY     POSTAL_CODE_DELIVERY     COUNTRY_CODE_DELIVERY           TYPE_DELIVERY
                  98.40                    98.40                         0.07                     0.07                     0.26                      0.07                    0.07
  PHYSICAL_CHECK_DECISION                  Comp_1                       Comp_2                   Comp_3                   Comp_4                    Comp_5                  Comp_6
                   0.00                     0.00                        97.53                    99.32                    99.76                     99.92                   99.98
                 Comp_7                   Comp_8                       Comp_9                  Comp_10                  Comp_11      DECLARATION_WEEK_DAY         DECLARATION_DAY
                  99.99                   100.00                       100.00                   100.00                   100.00                      0.00                    0.00
      DECLARATION_MONTH         DECLARATION_YEAR    CONTROL_DECISION_WEEK_DAY      CONTROL_DECISION_DAY   CONTROL_DECISION_MONTH     CONTROL_DECISION_YEAR
                   0.00                     0.00                         0.00                     0.00                     0.00                      0.00
```

Figure 13: Certificate missing values.

There many attributes (more than 10) with more than 25% of empty values., those are candidates to be removed since, most likely, will just add noise.

```
> c <- which(sapply(dataSet[,grepl("^(?!Comp)",colnames(dataSet),perl =  TRUE)],function (x) as.integer(sum(is.na(x))/dim(dataSet)[1]*100) > 25 ))
> dataSetLimpio <- dataSet
> dataSetLimpio[,c] <- NULL
> n <- dim(dataSetLimpio)[1]
> nulos <- function(x) paste(round(as.integer(sum(is.na(x))/n*100),3),"%")
> sapply(dataSetLimpio,nulos)
                 VERSION     CONFORM_EU_REQUIREMENT          COUNTRY_CONSIGNED             COUNTRY_ORIGIN          SUBMITTER_AUTH_ID           SUBMITTER_CCA_ID
                   "0 %"                      "0 %"                      "2 %"                      "2 %"                      "0 %"                      "0 %"
         CREATION_ORIGIN            INTERNAL_MARKET         NUMBER_OF_PACKAGES       PRODUCT_GROSS_WEIGHT         PRODUCT_NET_WEIGHT        PRODUCT_TEMPERATURE
                  "12 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
                 PURPOSE                     STATUS    TRANSPORT_INTERNAL_CODE   TRANSPORT_INTERNAL_IDENT     COUNTRY_CODE_CITY_AUTH             NAME_CITY_AUTH
                   "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
POSTAL_CODE_REGION_CITY_AUTH             ID_AUTHORITY             NAME_AUTHORITY             CODE_AUTHORITY         SUBCLASS_AUTHORITY            FLAG1_AUTHORITY
                   "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
            ID_CONSIGNEE             NAME_CONSIGNEE      POSTAL_CODE_CONSIGNEE     COUNTRY_CODE_CONSIGNEE             TYPE_CONSIGNEE               ID_CONSIGNOR
                   "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
          NAME_CONSIGNOR      POSTAL_CODE_CONSIGNOR     COUNTRY_CODE_CONSIGNOR             TYPE_CONSIGNOR                ID_IMPORTER              NAME_IMPORTER
                   "0 %"                      "0 %"                      "2 %"                      "0 %"                      "0 %"                      "0 %"
    POSTAL_CODE_IMPORTER      COUNTRY_CODE_IMPORTER              TYPE_IMPORTER             ID_LOAD_PERSON           NAME_LOAD_PERSON    POSTAL_CODE_LOAD_PERSON
                   "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
COUNTRY_CODE_LOAD_PERSON           TYPE_LOAD_PERSON                DELIVERY_ID              NAME_DELIVERY       POSTAL_CODE_DELIVERY      COUNTRY_CODE_DELIVERY
                   "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
           TYPE_DELIVERY    PHYSICAL_CHECK_DECISION                    Comp_ 1                    Comp_ 2                    Comp_ 3                    Comp_ 4
                   "0 %"                      "0 %"                      "0 %"                     "97 %"                     "99 %"                     "99 %"
                 Comp_ 5                    Comp_ 6                    Comp_ 7                    Comp_ 8                    Comp_ 9                   Comp_ 10
                  "99 %"                     "99 %"                     "99 %"                     "99 %"                     "99 %"                     "99 %"
                Comp_ 11  CONTROL_DECISION_WEEK_DAY       CONTROL_DECISION_DAY     CONTROL_DECISION_MONTH      CONTROL_DECISION_YEAR       DECLARATION_WEEK_DAY
                  "99 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"                      "0 %"
         DECLARATION_DAY          DECLARATION_MONTH           DECLARATION_YEAR
                   "0 %"                      "0 %"                      "0 %"
```

Figure 14: Empty values removal

Now we have reduced the dimension of the dataset to 62:

```
> dim(dataSetLimpio)
[1] 220113     69
```

We convert empty values to zeros for all cases but PRODUCT_TEMPERATURE since it is a factor converted to a number with a real meaning:

```
> unique(comp_desagregados$PRODUCT_TEMPERATURE)
[1] "ambient" "chilled" "frozen"  NA
```

Value 1 means “ambient”, and it is the most probable option is the observation does not have a value; so we will replace empty observations with one, for the rest of the entries we will use zero:

```
> unique(dataSetLimpio$PRODUCT_TEMPERATURE)
[1]  1  2  3 NA
> dataSetLimpio$PRODUCT_TEMPERATURE[sapply(dataSetLimpio$PRODUCT_TEMPERATURE,function(x) is.na(x))] <- 1
> unique(dataSetLimpio$PRODUCT_TEMPERATURE)
[1] 1 2 3
> dataSetLimpio[sapply(dataSetLimpio,function(x) is.na(x))] <- 0
> sapply(dataSetLimpio,function (x) round(sum(is.na(x))/219876*100,digits = 3))
                VERSION    CONFORM_EU_REQUIREMENT         COUNTRY_CONSIGNED            COUNTRY_ORIGIN         SUBMITTER_AUTH_ID
                      0                         0                         0                         0                         0
       SUBMITTER_CCA_ID           CREATION_ORIGIN           INTERNAL_MARKET        NUMBER_OF_PACKAGES      PRODUCT_GROSS_WEIGHT
                      0                         0                         0                         0                         0
     PRODUCT_NET_WEIGHT       PRODUCT_TEMPERATURE                   PURPOSE                    STATUS   TRANSPORT_INTERNAL_CODE
                      0                         0                         0                         0                         0
TRANSPORT_INTERNAL_IDENT    COUNTRY_CODE_CITY_AUTH            NAME_CITY_AUTH POSTAL_CODE_REGION_CITY_AUTH          ID_AUTHORITY
                      0                         0                         0                         0                         0
         NAME_AUTHORITY            CODE_AUTHORITY        SUBCLASS_AUTHORITY           FLAG1_AUTHORITY              ID_CONSIGNEE
                      0                         0                         0                         0                         0
         NAME_CONSIGNEE     POSTAL_CODE_CONSIGNEE    COUNTRY_CODE_CONSIGNEE            TYPE_CONSIGNEE              ID_CONSIGNOR
                      0                         0                         0                         0                         0
         NAME_CONSIGNOR     POSTAL_CODE_CONSIGNOR    COUNTRY_CODE_CONSIGNOR            TYPE_CONSIGNOR               ID_IMPORTER
                      0                         0                         0                         0                         0
          NAME_IMPORTER      POSTAL_CODE_IMPORTER     COUNTRY_CODE_IMPORTER             TYPE_IMPORTER            ID_LOAD_PERSON
                      0                         0                         0                         0                         0
       NAME_LOAD_PERSON   POSTAL_CODE_LOAD_PERSON  COUNTRY_CODE_LOAD_PERSON          TYPE_LOAD_PERSON               DELIVERY_ID
                      0                         0                         0                         0                         0
          NAME_DELIVERY      POSTAL_CODE_DELIVERY     COUNTRY_CODE_DELIVERY             TYPE_DELIVERY   PHYSICAL_CHECK_DECISION
                      0                         0                         0                         0                         0
                 Comp_1                    Comp_2                    Comp_4                    Comp_5                    Comp_7
                      0                         0                         0                         0                         0
                 Comp_8           DECLARATION_DAY          DECLARATION_YEAR  CONTROL_DECISION_WEEK_DAY      CONTROL_DECISION_DAY
                      0                         0                         0                         0                         0
 CONTROL_DECISION_MONTH     CONTROL_DECISION_YEAR
                      0                         0
```

Figure 15: Clean dataset.

Attributes with constant values also (variance near zero) will be removed as well, but keeping an eye on complements (section 4.5.3). For instance, the country authority, in this case, is always Germany, variance of this attribute is obviously zero, it is a candidate to be removed.

4.4.2. Balancing datasets

We have a very unbalanced dataset. The number of rejected certificates is much smaller than certificates that belong to "negative" class; those are the valid certificates.

The existence of an unbalance training dataset can be a problem to obtain a good classifier while using traditional classification techniques like decision trees or neural networks.

There are several techniques at a preprocessing and a processing level to balance datasets. We will apply several on this project. Following it is presented a brief introduction of what are and in which consists these technics.

To fight against this problem are two different approaches: algorithm approach or data approach:

Data approach is done when we are preparing the data before applying any algorithm and consists on re-sample the unbalanced datasets. This technic will allow us to create more instances of one class (over sampling) or eliminate some instances of the class (under sampling)

Algorithm approach, this will apply technics to the algorithms while processing data. Some of this technics can be boost algorithms, applying higher costs/weights to the under sampled class or change threshold to give more importance to the weak class (rejected certificates).

As these technics will depend on the results of the algorithms, we will apply them together with algorithms with an iterative approach, so this will be covered in more detail in the following chapter.

4.5. Machine Learning Algorithm Selection.

The methodology selected to apply the most appropriate algorithm will be as follows:

- Reduce dataset and training folder to see if this impacts a lot the performance of GLM algorithm.
- Application of several algorithms without a balanced dataset:
  - Split dataset between training and validation 80%(training) - 20%(validation)
  - Apply several algorithms to the training dataset
  - See results
  - Validate models with validation datasets
  - Compare results
- Filter attributes and assesses performance with GLM algorithm.
- Applying pre-processing algorithms to balance the dataset:
  - Split dataset between training and validation 80%(training) - 20%(validation)
  - Balance training dataset with under-sampling and SMOTE technics.
  - Apply several algorithms to the training dataset
  - See results
  - Validate models with validation datasets. The validations need to be done with a dataset following the same distribution as the original population. Otherwise, we could get misleading results
  - Compare results

- Applying boosting algorithms to train weak unbalance datasets [phase *Algorithm and training tuning*]:
    - Apply bagging and boost algorithms
    - See results
    - Validate models with validation datasets. The validations need to be done with a dataset following the same distribution as the original population. Otherwise, we could get misleading results
    - Compare results

A process diagram to clarify the methodology:

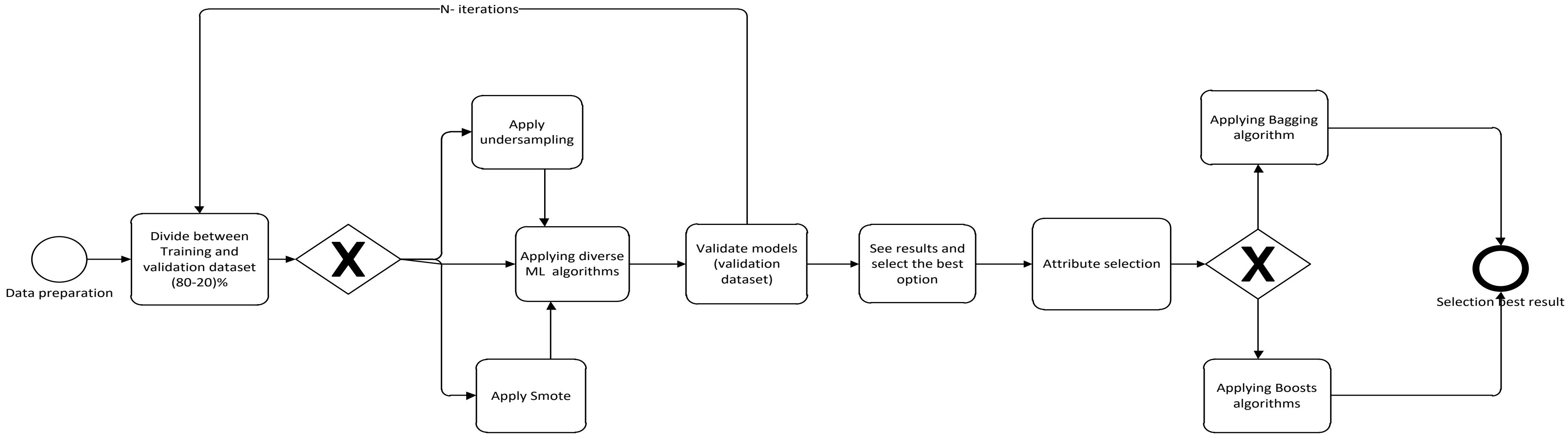


Figure 16: Algorithm selection process.

4.5.1. Creating Training and Validation datasets

We split dataset in training (80%) and validation (20%), the last one will be used to assess the real performance of the algorithm (against date it never saw):

```
#Training and validating data
set.seed(171819)
indice <- createDataPartition(datos$STATUS, p=0.80, list= FALSE, times = 1)
tr <- datos[indice,]
val <- datos[-indice,]
```

```
> dim(tr)[1]/(dim(val)[1]+dim(tr)[1])
[1] 0.8000036
> dim(val)[1]/(dim(val)[1]+dim(tr)[1])
[1] 0.1999964
```

Transformations will apply only to training dataset (tr), leaving validation (val) as it is (imbalanced).

4.5.2. Algorithms test-suite (selection)

We will start defining a test suite of algorithms to compare their performance; for model training, we will be extensively using Caret package [20]. As a warning, caret package is quite unstable; it is recommended to install it directly from GitHub since daily updates are done to fix patches (quite frequent):

devtools::install_github('topepo/caret/pkg/caret')

Since there is enough data we will use ten-fold cross-validation with three repetitions, this is a standard test suite configuration. It is a binary classification problem.

```
#Training: 3 Repeated 10 fold cross validation.
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3, classProbs = TRUE)
```

We will use the following classification algorithms: **cart, glm, knn, svm and random forest.**

4.5.2.1. Comment on computation cost and imbalance

Before starting the training suite, some changes to the original dataset (and therefore the training and validation sets) must be done. The computational cost of running 3 repeated 10 fold cross validation training with this amount of data: ~ 220.000 observations with ~ 70 predictors largely exceeds our available CPU power. Just a 3 repeated 10 folded c.v. for a KKN algorithm takes more than five days of computing time.

To overcome this limitation, I will consider data available as of 2015; this will reduce the original dataset to a sample of ~90.000. Out of this sample, I have tried to reduce it to a 50% by random selection, so we would have half of the data that spans through 2015, 2016, and 2017. Unfortuantely some tests (with GLM algorithm) show this transformation highly affects the performance of the model; it is much more important to keep as many observations as we can than increasing the training repetitions.

If we reduce the dataset to 45000 rows by selecting half of the data since 2015:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      741      215
  Accepted      914    42265

               Accuracy : 0.9744
                 95% CI : (0.9729, 0.9759)
    No Information Rate : 0.9625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5554
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.77510
                 Recall : 0.44773
                     F1 : 0.56760
             Prevalence : 0.03750
         Detection Rate : 0.01679
   Detection Prevalence : 0.02166
      Balanced Accuracy : 0.72134

       'Positive' Class : Rejected
```

*Figure 17: GLM 45K rows **10-3** Rep.Cross.Val.*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      741      215
  Accepted      914    42265

               Accuracy : 0.9744
                 95% CI : (0.9729, 0.9759)
    No Information Rate : 0.9625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5554
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.77510
                 Recall : 0.44773
                     F1 : 0.56760
             Prevalence : 0.03750
         Detection Rate : 0.01679
   Detection Prevalence : 0.02166
      Balanced Accuracy : 0.72134

       'Positive' Class : Rejected
```

*Figure 18: GLM 45K rows. **5-2 Rep**.Cross.Val.*

Performance is an invariance regarding training repetitions. Let's select the whole dataset as of 2014 (~90.000 observations):

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      551      169
  Accepted      384    17068

               Accuracy : 0.9696
                 95% CI : (0.967, 0.972)
    No Information Rate : 0.9485
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6502
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.76528
                 Recall : 0.58930
                     F1 : 0.66586
             Prevalence : 0.05145
         Detection Rate : 0.03032
   Detection Prevalence : 0.03962
      Balanced Accuracy : 0.78975

       'Positive' Class : Rejected
```

Figure 19: GLM 90K rows. 10-3 Rep.Cross.Val

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      551      169
  Accepted      384    17068

               Accuracy : 0.9696
                 95% CI : (0.967, 0.972)
    No Information Rate : 0.9485
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6502
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.76528
                 Recall : 0.58930
                     F1 : 0.66586
             Prevalence : 0.05145
         Detection Rate : 0.03032
   Detection Prevalence : 0.03962
      Balanced Accuracy : 0.78975

       'Positive' Class : Rejected
```

Figure 20: GLM 90K rows. 5-2 Cross Val

We see training repetition remains invariance and we have improved recall value in more than a 10%.

Let's increase the number of observations:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      710      199
  Accepted      824    36779

               Accuracy : 0.9734
                 95% CI : (0.9718, 0.975)
    No Information Rate : 0.9602
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5685
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.78108
                 Recall : 0.46284
                     F1 : 0.58125
             Prevalence : 0.03983
         Detection Rate : 0.01844
   Detection Prevalence : 0.02360
      Balanced Accuracy : 0.72873

       'Positive' Class : Rejected
```

*Figure 21: GLM **190K** rows.*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      741      215
  Accepted      914    42265

               Accuracy : 0.9744
                 95% CI : (0.9729, 0.9759)
    No Information Rate : 0.9625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5554
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.77510
                 Recall : 0.44773
                     F1 : 0.56760
             Prevalence : 0.03750
         Detection Rate : 0.01679
   Detection Prevalence : 0.02166
      Balanced Accuracy : 0.72134

       'Positive' Class : Rejected
```

*Figure 22: GLM 220K rows **5-2** Cross Val.*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      741      215
  Accepted      914    42265

               Accuracy : 0.9744
                 95% CI : (0.9729, 0.9759)
    No Information Rate : 0.9625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5554
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.77510
                 Recall : 0.44773
                     F1 : 0.56760
             Prevalence : 0.03750
         Detection Rate : 0.01679
   Detection Prevalence : 0.02166
      Balanced Accuracy : 0.72134

       'Positive' Class : Rejected
```

*Figure 23: GLM 220K rows **10-3** Cross Val.*

We see the **performance has been reduced with 220K observations** and increasing the number of training repetitions does not help to improve the performance. A sweet point seems to be 2013, with that amount of data (~125.000 observations) we have a precision of 79% and recall of 51%. So, we safely can reduce the number of observations from 220.000 to 125.000, and also reduce the training repetitions from 3 repeated 10 cross validations to 2 repeated 5 cross validations.

This behaviour is a clear consequence of the high imbalance of the dataset, removing observations impacts the positive class (“Rejected”) and the ability to not miss-predict the positive class (recall) decreases.

We can try now to pre-process the data increasing the proportion of the positive class by down-sampling the of negative ones, or introducing synthetic positives observations with SMOTE:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1023     2087
  Accepted      150    21766

               Accuracy : 0.9106
                 95% CI : (0.907, 0.9141)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4396
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.32894
                 Recall : 0.87212
                     F1 : 0.47770
             Prevalence : 0.04687
         Detection Rate : 0.04088
   Detection Prevalence : 0.12427
      Balanced Accuracy : 0.89231

       'Positive' Class : Rejected
```

Figure 24: GLM Down-Sampling 120K rows to 9.3K rows.

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1403     4327
  Accepted      252    38153

               Accuracy : 0.8963
                 95% CI : (0.8934, 0.8991)
    No Information Rate : 0.9625
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3416
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.24485
                 Recall : 0.84773
                     F1 : 0.37996
             Prevalence : 0.03750
         Detection Rate : 0.03179
   Detection Prevalence : 0.12983
      Balanced Accuracy : 0.87294

       'Positive' Class : Rejected
```

Figure 25: GLM Down-Sampling 220K rows to 13K rows

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      955     1526
  Accepted      218    22327

               Accuracy : 0.9303
                 95% CI : (0.9271, 0.9334)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4903
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.38493
                 Recall : 0.81415
                     F1 : 0.52271
             Prevalence : 0.04687
         Detection Rate : 0.03816
   Detection Prevalence : 0.09914
      Balanced Accuracy : 0.87509

       'Positive' Class : Rejected
```

Figure 26: GLM SMOTE 125K rows to 18K rows

Same pattern as before with a number of observations. Selecting since 2011 (~ 220.000 rows) decreases performance, so it is better to downsample data since 2013 (~125.000 rows). SMOTE performs worse than down-sampling and also increasing computational time (downsampling is random). We will select downsampling to reduce the dataset size.

Precision has been reduced highly in favour of the recall, in our case, this is actually much better than having a more balanced result since the data is highly imbalance and we do need a higher recall of the positive class. Notice in the confusion matrix that only 150 positive instances have been miss-predicted, on the other hand, two thirds of predicted positives are negatives (bad precision), this is 2000 miss-classification but if we considered the negative class contains ~23000 instances, **by inspecting 12% of the consignments we detect 87% of consignment that should be rejected.**

With such results, we will *downsample* observations as of 2013 for all algorithms in the test suit, and we will accept the decrease in precision on favour of recall:

```
> dim(val)
[1] 25026    69
> source("TFM-carga.r")
> dim(datos)
[1] 125135     69
> dim(tr)
[1] 100109     69
> dim(val)
[1] 25026    69
> set.seed(171819)
> col_class <- which(sapply(colnames(datos), function(x) x =="STATUS"))
> tr <- downSample(x =  tr[,-col_class], y = tr[,col_class], list = FALSE, yname = "STATUS")
> dim(tr)
[1] 9392    69
> dim(val)
[1] 25026    69
>
```

The training method has been reduced as well to a 2 repeated 5 cross-fold validation. The validation dataset has, obviously, not been down-sampled. **Training data set contains now ~9.9K rows with the following test suite (from Caret package):**

```
#Training: 3 Repeated 10 fold cross validation.
trainControl <- trainControl(method="repeatedcv", number=5, repeats=2, classProbs = TRUE,sampling ="down")

set.seed(171819)
trainControl$classProbs <- TRUE
fit.glm <- train(STATUS ~ ., data = tr,
                 method="glm",
                 trControl=trainControl,
                                preProc=c("center","scale","BoxCox"),
                 metric = "ROC",
                 na.action=na.omit)

set.seed(171819)
col_class <- which(sapply(colnames(tr), function(x) x =="STATUS"))
fit.cart<- train(x=tr[,-col_class], y=tr[,col_class],
                 method="rpart",
                 trControl=trainControl,
                 metric = "ROC",
                 na.action=na.omit)
#KNN needs preprocessing.
set.seed(171819)
fit.knn<- train(STATUS ~ ., data = tr,
                 method="knn",
                 trControl=trainControl,
                 tuneLength=5,
                                preProc=c("center","scale"),
                 metric = "ROC")


set.seed(171819)
fit.rf<- train(STATUS ~ ., data = tr,
                 method="rf",
                 metric="ROC",
                 trControl=trainControl,
                 tuneLength = 2,
                 na.action=na.omit)

set.seed(171819)
fit.svm<- train(STATUS ~ ., data = tr,
                 method="svmRadial",
                 trControl=trainControl,
                 metric="ROC",
                 tunelength = 2,
                 na.action=na.omit)
```

4.5.2.2. Training Results

We will compare now results for the selected algorithms.

```
Call:
summary.resamples(object = results)

Models: GLMNET, CART, KNN, SVM, RF
Number of resamples: 10

Accuracy
            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
GLMNET 0.9207332 0.9218725 0.9223354 0.9237231 0.9255444 0.9295275    0
CART   0.9393667 0.9406778 0.9412896 0.9412390 0.9420505 0.9432125    0
KNN    0.9010039 0.9060873 0.9089275 0.9081201 0.9101103 0.9146439    0
SVM    0.7948257 0.7960794 0.7972878 0.7991439 0.8006942 0.8103586    0
RF     0.9235802 0.9267394 0.9286553 0.9284080 0.9303266 0.9322246    0

Kappa
            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
GLMNET 0.4745662 0.4782723 0.4809351 0.4836603 0.4856822 0.5028893    0
CART   0.5014518 0.5052240 0.5115406 0.5118150 0.5181152 0.5254937    0
KNN    0.4045626 0.4202631 0.4273001 0.4262549 0.4342464 0.4505287    0
SVM    0.1929775 0.1993417 0.2037354 0.2033930 0.2071036 0.2115434    0
RF     0.4865820 0.5012075 0.5038654 0.5052347 0.5132028 0.5218946    0
```

Figure 27: Training results

We can see good accuracy across the board. Almost all algorithms have a mean accuracy above 90%; t**he problem is learnable.**

4.5.2.3. Algorithms test-suite Validation

To select an algorithm, we will compare predictions for the models generated by the validation data (e.g. confusion Matrix: *predictionGLM <- predict(fit.glm, val)*):

GLM

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1014     1774
  Accepted      159    22079

               Accuracy : 0.9228
                 95% CI : (0.9194, 0.926)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4775
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.36370
                 Recall : 0.86445
                     F1 : 0.51199
             Prevalence : 0.04687
         Detection Rate : 0.04052
   Detection Prevalence : 0.11140
      Balanced Accuracy : 0.89504

       'Positive' Class : Rejected
```

CART

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      862     1157
  Accepted      311    22696

               Accuracy : 0.9413
                 95% CI : (0.9384, 0.9442)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5111
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.42694
                 Recall : 0.73487
                     F1 : 0.54010
             Prevalence : 0.04687
         Detection Rate : 0.03444
   Detection Prevalence : 0.08068
      Balanced Accuracy : 0.84318

       'Positive' Class : Rejected
```

KNN

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1023     2160
  Accepted      150    21693

               Accuracy : 0.9077
                 95% CI : (0.904, 0.9113)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4307
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.32139
                 Recall : 0.87212
                     F1 : 0.46970
             Prevalence : 0.04687
         Detection Rate : 0.04088
   Detection Prevalence : 0.12719
      Balanced Accuracy : 0.89078

       'Positive' Class : Rejected
```

RF

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1045     1690
  Accepted      128    22163

               Accuracy : 0.9274
                 95% CI : (0.9241, 0.9305)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5021
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.38208
                 Recall : 0.89088
                     F1 : 0.53480
             Prevalence : 0.04687
         Detection Rate : 0.04176
   Detection Prevalence : 0.10929
      Balanced Accuracy : 0.91001

       'Positive' Class : Rejected
```

SVM

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      911     4431
  Accepted      262    19422

               Accuracy : 0.8125
                 95% CI : (0.8076, 0.8173)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2197
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.17054
                 Recall : 0.77664
                     F1 : 0.27966
             Prevalence : 0.04687
         Detection Rate : 0.03640
   Detection Prevalence : 0.21346
      Balanced Accuracy : 0.79544

       'Positive' Class : Rejected
```

As a note, I have seen KNN is quite sensible to the data format, so I have scaled and centred the data in order to get results closer to the other algorithms.

Given the results above, we will continue with the Random Forest algorithm.

4.5.3. Under-sampling and SMOTE (balancing datasets)

As we have seen imbalanced datasets have a high impact on all models, by balancing the dataset we can achieve better results. In our case, since we needed to reduce the number of observations to reduce the required computation time, we already have a balanced dataset.

```
> dim(datos)
[1] 125135     69
> dim(tr)
[1] 100109     69
> dim(val)
[1] 25026    69
> set.seed(3124234)
> col_class <- which(sapply(colnames(datos), function(x) x =="STATUS"))
> tr <- downSample(x =  tr[,-col_class], y = tr[,col_class], list = FALSE, yname = "STATUS")
> dim(tr)
[1] 9392   69
```

So, by applying under-sampling (SMOTE is computationally more costly and performs worse), we have reduced dramatically the time needed to train the model and increased the recall.

4.5.4. Filtering data

In this section, we will take our dataset, and we will eliminate attributes with variance close to 0 from the data set and also correlated attributes, those who have a correlation above 90%. After this filtration, we will execute the best algorithm from the previous point to see if results vary.

4.5.4.1. Near zero variance

Near zero variance attributes do not, normally, add value to the model, so they just increase their complexity. In our case, we disaggregate the complements columns and created several attributes (as much as the maximum number complements that a certificate can have); doing so we have columns with variance near zero since, for instance, there are few certificates with 11 complements:

```
> col_class <- which(sapply(colnames(tr), function(x) x =="STATUS"))
> nzv <- nearZeroVar(datos[,- col_class],saveMetrics=TRUE)
> head(nzv)
                      freqRatio percentUnique zeroVar   nzv
VERSION                3.295683  0.0151836017   FALSE FALSE
CONFORM_EU_REQUIREMENT 586.488263 0.0015982739  FALSE  TRUE
COUNTRY_CONSIGNED      2.201456  0.1126783074   FALSE FALSE
COUNTRY_ORIGIN         2.209413  0.1182722660   FALSE FALSE
SUBMITTER_AUTH_ID      1.265985  0.0998921165   FALSE FALSE
SUBMITTER_CCA_ID       0.000000  0.0007991369    TRUE  TRUE
> nzv[nzv$nzv == TRUE,]
                          freqRatio percentUnique zeroVar  nzv
CONFORM_EU_REQUIREMENT    586.48826  0.0015982739   FALSE TRUE
SUBMITTER_CCA_ID            0.00000  0.0007991369    TRUE TRUE
PURPOSE                   284.05479  0.0039956847   FALSE TRUE
COUNTRY_CODE_CITY_AUTH      0.00000  0.0007991369    TRUE TRUE
SUBCLASS_AUTHORITY         25.74492  0.0023974108   FALSE TRUE
FLAG1_AUTHORITY            25.47799  0.0015982739   FALSE TRUE
TYPE_CONSIGNOR           1105.65487  0.0039956847   FALSE TRUE
COUNTRY_CODE_LOAD_PERSON  583.77934  0.0159827386   FALSE TRUE
COUNTRY_CODE_DELIVERY      23.74612  0.0487473529   FALSE TRUE
TYPE_DELIVERY              61.98831  0.0167818756   FALSE TRUE
COMPLEMENT_NUMBER          53.43289  0.0087905063   FALSE TRUE
2                         323.14854  0.1901945898   FALSE TRUE
3                         790.49045  0.0934990211   FALSE TRUE
4                        1948.90625  0.0647300915   FALSE TRUE
5                        6249.85000  0.0391577097   FALSE TRUE
6                       31273.50000  0.0239741080   FALSE TRUE
7                       62556.50000  0.0135853278   FALSE TRUE
8                       62565.00000  0.0039956847   FALSE TRUE
9                      125132.00000  0.0031965477   FALSE TRUE
10                     125133.00000  0.0023974108   FALSE TRUE
11                     125134.00000  0.0015982739   FALSE TRUE
```

Figure 28: Near Zero Variance attributes.

There are a lot of near zero variance attributes. Actually, the list complements, going from 1 to 11, including the number of them, are near zero.

Let's remove first near-zero variance attributes but keeping complements attributes:

```
> col_not_remove <- which(colnames(tr) %in% c("STATUS","COMPLEMENT_NUMBER",1:99))
> col_not_remove
 [1] 14 50 51 52 53 54 55 56 57 58 59 60 61
> nz <- nearZeroVar(tr[,-col_not_remove],saveMetrics = TRUE)
> nz[nz$nzv==TRUE,]
                          freqRatio percentUnique zeroVar  nzv
CONFORM_EU_REQUIREMENT    609.42073  0.0019978224   FALSE TRUE
SUBMITTER_CCA_ID            0.00000  0.0009989112    TRUE TRUE
PURPOSE                   291.03216  0.0049945559   FALSE TRUE
COUNTRY_CODE_CITY_AUTH      0.00000  0.0009989112    TRUE TRUE
SUBCLASS_AUTHORITY         25.69931  0.0029967336   FALSE TRUE
FLAG1_AUTHORITY            25.42793  0.0019978224   FALSE TRUE
TYPE_CONSIGNOR           1041.09375  0.0049945559   FALSE TRUE
COUNTRY_CODE_LOAD_PERSON  578.27326  0.0199782237   FALSE TRUE
COUNTRY_CODE_DELIVERY      23.74038  0.0589357600   FALSE TRUE
TYPE_DELIVERY              62.49680  0.0209771349   FALSE TRUE
> col_remove <- rownames(nz[nz$nzv==TRUE,])
> c <- which(colnames(tr) %in% col_remove)
> c
 [1]  2  6 13 17 23 24 34 43 48 49
> tr <- tr[,-c]; val <- val[,-c]
> dim(tr);dim(val)
[1] 100109     59
[1] 25026     59
```

We end up with 59 attributes. Let train our random forest with this data to see if performance has not decreased:

```
> trainControl <- trainControl(method="repeatedcv", number=5, repeats=2, classProbs = TRUE,sampling ="down")
> set.seed(171819)
> fit.rf<- train(STATUS ~ ., data = tr,
+                method="rf",
+                metric="ROC",
+                trControl=trainControl,
+                tuneLength = 2,
+                na.action=na.omit)
```

Validation results show the model has slightly less recall but 1% increment over precision. If we do remove complement attributes with near-zero variance only (similar process ending up with 58 attributes), which actually have near-zero variance all of them, we see a decrease of 1% in recall and almost an increase of 1% in precision:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1044     1604
  Accepted      129    22249

               Accuracy : 0.9308
                 95% CI : (0.9275, 0.9339)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5149
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.39426
                 Recall : 0.89003
                     F1 : 0.54645
             Prevalence : 0.04687
         Detection Rate : 0.04172
   Detection Prevalence : 0.10581
      Balanced Accuracy : 0.91139

       'Positive' Class : Rejected
```

*Figure 29: RF without near-zero variance **attributes** (complements kept).*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1038     1618
  Accepted      135    22235

               Accuracy : 0.93
                 95% CI : (0.9267, 0.9331)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5103
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.39081
                 Recall : 0.88491
                     F1 : 0.54218
             Prevalence : 0.04687
         Detection Rate : 0.04148
   Detection Prevalence : 0.10613
      Balanced Accuracy : 0.90854

       'Positive' Class : Rejected
```

*Figure 30: RF without near-zero variance **complement** attributes (other attrs. kept)*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1045     1690
  Accepted      128    22163

               Accuracy : 0.9274
                 95% CI : (0.9241, 0.9305)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5021
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.38208
                 Recall : 0.89088
                     F1 : 0.53480
             Prevalence : 0.04687
         Detection Rate : 0.04176
   Detection Prevalence : 0.10929
      Balanced Accuracy : 0.91001

       'Positive' Class : Rejected
```

Figure 31: RF original performance.

Removing complements helps a little on precision, not as much as removing the near-zero variance attributes. We can try a mix by keeping “Complement_Number” and one to three complements (the most populated actually):

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1042     1486
  Accepted      131    22367

               Accuracy : 0.9354
                 95% CI : (0.9323, 0.9384)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5332
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.41218
                 Recall : 0.88832
                     F1 : 0.56309
             Prevalence : 0.04687
         Detection Rate : 0.04164
   Detection Prevalence : 0.10101
      Balanced Accuracy : 0.91301

       'Positive' Class : Rejected
```

*Figure 32: RF without near-zero attrs. and with Complement_Number and the **first** complement kept.*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1038     1500
  Accepted      135    22353

               Accuracy : 0.9347
                 95% CI : (0.9315, 0.9377)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5292
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.40898
                 Recall : 0.88491
                     F1 : 0.55942
             Prevalence : 0.04687
         Detection Rate : 0.04148
   Detection Prevalence : 0.10141
      Balanced Accuracy : 0.91101

       'Positive' Class : Rejected
```

*Figure 33: : RF without near-zero attrs. and with Complement_Number and **two** complements kept*

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1038     1500
  Accepted      135    22353

               Accuracy : 0.9347
                 95% CI : (0.9315, 0.9377)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5292
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.40898
                 Recall : 0.88491
                     F1 : 0.55942
             Prevalence : 0.04687
         Detection Rate : 0.04148
   Detection Prevalence : 0.10141
      Balanced Accuracy : 0.91101

       'Positive' Class : Rejected
```

*Figure 34: RF without near-zero attrs. and with Complement_Number and **three** complements kept*

We see the first complement (together with Complement Number) increases Precision by almost 1% and just reduces recall in 0.2%. Introducing more complements does not affect Recall anymore and starts reducing Precision, so we will keep the first configuration of attributes: removing all near-variance attributes but keeping “Complement_Number” and first complement. The final amount of attributes is 49:

```
> col_not_remove <- which(colnames(tr) %in% c("STATUS","COMPLEMENT_NUMBER",1))
> nz <- nearZeroVar(tr[,-col_not_remove],saveMetrics = TRUE)
> nz[nz$nzv==TRUE,]
                           freqRatio percentUnique zeroVar  nzv
CONFORM_EU_REQUIREMENT     609.42073  0.0019978224   FALSE TRUE
SUBMITTER_CCA_ID             0.00000  0.0009989112    TRUE TRUE
PURPOSE                    291.03216  0.0049945559   FALSE TRUE
COUNTRY_CODE_CITY_AUTH       0.00000  0.0009989112    TRUE TRUE
SUBCLASS_AUTHORITY          25.69931  0.0029967336   FALSE TRUE
FLAG1_AUTHORITY             25.42793  0.0019978224   FALSE TRUE
TYPE_CONSIGNOR            1041.09375  0.0049945559   FALSE TRUE
COUNTRY_CODE_LOAD_PERSON   578.27326  0.0199782237   FALSE TRUE
COUNTRY_CODE_DELIVERY       23.74038  0.0589357600   FALSE TRUE
TYPE_DELIVERY               62.49680  0.0209771349   FALSE TRUE
2                          319.51803  0.2177626387   FALSE TRUE
3                          827.39167  0.1048856746   FALSE TRUE
4                         1956.56863  0.0709226943   FALSE TRUE
5                         7142.64286  0.0469488258   FALSE TRUE
6                        33358.66667  0.0249727797   FALSE TRUE
7                        50045.00000  0.0139847566   FALSE TRUE
8                        50052.50000  0.0039956447   FALSE TRUE
9                       100106.00000  0.0039956447   FALSE TRUE
10                      100107.00000  0.0029967336   FALSE TRUE
11                      100108.00000  0.0019978224   FALSE TRUE
```

4.5.5. High correlation

High correlated attributes do not add value to the dataset either, so with the dataset already cleaned.

Below figure shows a map of correlation for all attributes (except the class attribute):

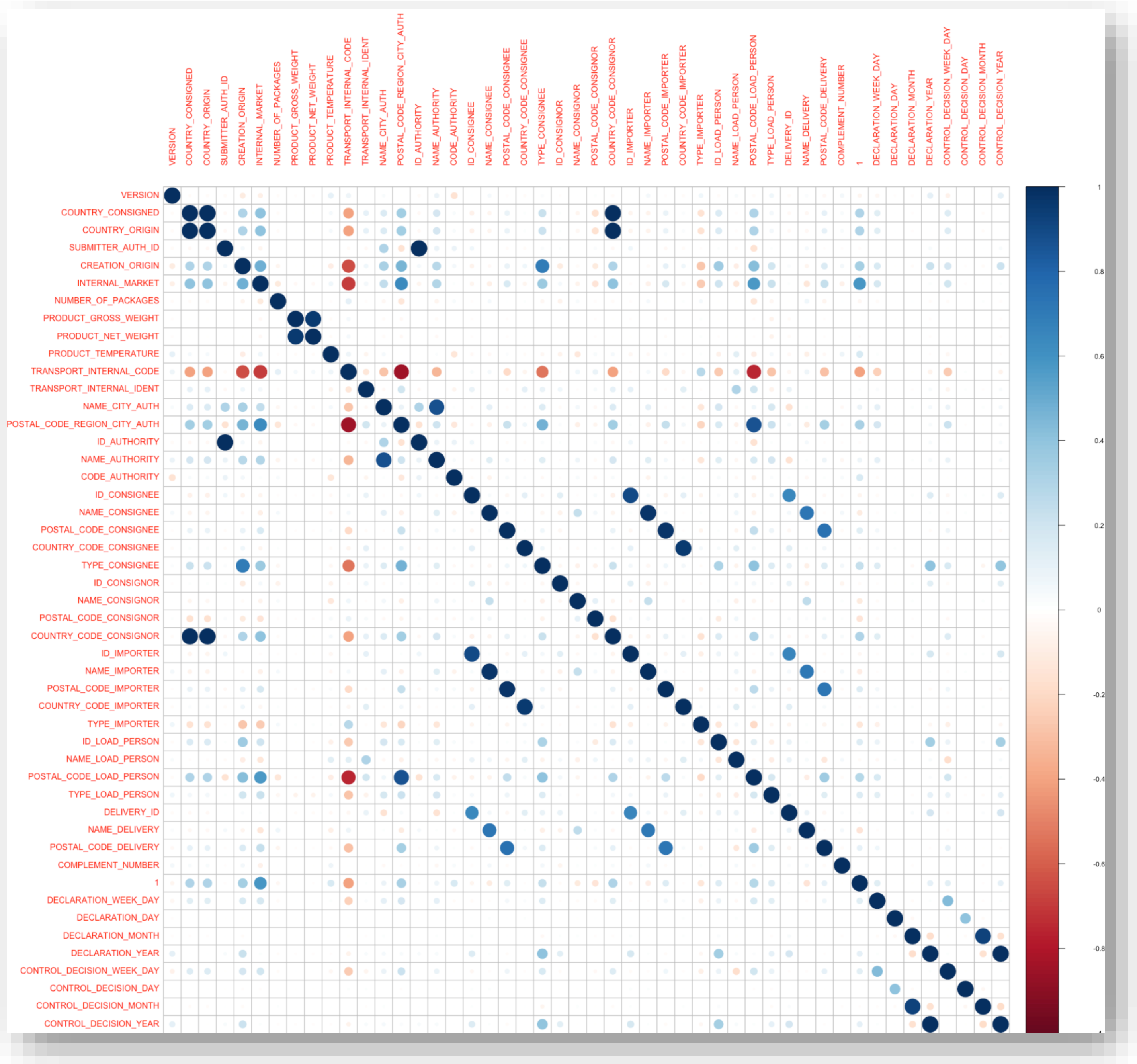


Figure 35: Correlation Map.

If we remove attributes with a correlation value higher than 0.9 we end up with 40 attributes:

```
> status<-which(names(val)=="STATUS")
> correlations <- cor(preprocessed)
> high_corr <- findCorrelation(correlations,cutoff = 0.9,names = TRUE)
> col_remove <- which(colnames(tr) %in% high_corr)
> tr <- tr[,-col_remove]; val <- val[,-col_remove]
> dim(tr) ; dim(val)
[1] 100109     40
[1] 25026    40
```

The validations of the model indicate we have increased precision by 1% and decreased recall by 0,256%:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1039     1417
  Accepted      134    22436

               Accuracy : 0.938
                 95% CI : (0.935, 0.941)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5437
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.42305
                 Recall : 0.88576
                     F1 : 0.57261
             Prevalence : 0.04687
         Detection Rate : 0.04152
   Detection Prevalence : 0.09814
      Balanced Accuracy : 0.91318

       'Positive' Class : Rejected
```

Figure 36: Random Forest performance without high correlated attributes.

4.6. Algorithm and training tuning

4.6.1. Bagging

Bagging mechanisms consist in “average” different outputs of the same model, hoping the average result will be better than the particular ones.

Our selected model, Random Forest, has already a built-in bagging mechanism [8], so we will check if other bagging models can improve our Random Forest (with a performance of 88% recall and 42% precision): Bagged CART (method “treebag”), Bagged Flexible Discriminant Analysis (method “bagEarth”), Bagged Logic Regression (method “logicBag”). As usual, we will use Caret library, so we just follow same training methodology but changing the algorithms (treebag, bagEarth and blacktree):

```
trainControl <- trainControl(method="repeatedcv", number=5, repeats=2, classProbs = TRUE, verbose=TRUE, sampling ="down")
status <- which(names(tr)=="STATUS")

set.seed(171819)
fit.CART_bagged <- train(x =tr[,-status], y = tr[,status],
                 method="treebag",
                 trControl=trainControl,
                 metric = "ROC",
                 tuneLength = 10,
                 importance = TRUE,
                 na.action=na.omit)

set.seed(171819)
fit.Ada_bagged <- train (x =tr[,-status], y = tr[, status],
                         method="AdaBag",
                         trControl=trainControl,
                         metric="ROC",
                         tuneLength = 10,
                         importance = TRUE,
                         na.action=na.omit)
```

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1043     1911
  Accepted      130    21942

               Accuracy : 0.9184
                 95% CI : (0.915, 0.9218)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4699
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.35308
                 Recall : 0.88917
                     F1 : 0.50545
             Prevalence : 0.04687
         Detection Rate : 0.04168
   Detection Prevalence : 0.11804
      Balanced Accuracy : 0.90453

       'Positive' Class : Rejected
```

Figure 37: CART bagged.

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      891     1255
  Accepted      282    22598

               Accuracy : 0.9386
                 95% CI : (0.9355, 0.9415)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.507
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.41519
                 Recall : 0.75959
                     F1 : 0.53691
             Prevalence : 0.04687
         Detection Rate : 0.03560
   Detection Prevalence : 0.08575
      Balanced Accuracy : 0.85349

       'Positive' Class : Rejected
```

Figure 38: ADA bagged.

We see results are far from results achieve by Random Forest although, to be fair, Random Forest is already a type of bagged algorithm.

4.6.2. Boosting

Boosting works chaining output of a model as the input for another model by giving more weight to instances incorrectly classified. We will use the same type of algorithm we have used in previous sections which render good results: linear models and trees. As usual, we use caret for this activity, and the selected models will be several boosted trees: blackboost and adaboost, and a generalised linear model: glmboost.

```
trainControl <- trainControl(method="repeatedcv", number=5, repeats=2, classProbs = TRUE,sampling ="down")
status <- which(names(tr)=="STATUS")


fit.gbm <- train(x = tr[,-status], y=tr[,status],
                    method="gbm",
                    metric="ROC",
                    trControl=trainControl,
                    tuneLength = 5)

fit.blackBoost <- train(x = tr[,-status], y=tr[,status],
                 method="blackboost",
                 metric="ROC",
                 trControl=trainControl,
                 tuneLength = 5)

fit.glmBoost <- train(x = tr[,-status], y=tr[,status],
                 method="glmboost",
                 metric="ROC",
                 trControl=trainControl,
                 tuneLength = 5)
```

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1017     1585
  Accepted      156    22268

               Accuracy : 0.9304
                 95% CI : (0.9272, 0.9336)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5069
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.39085
                 Recall : 0.86701
                     F1 : 0.53881
             Prevalence : 0.04687
         Detection Rate : 0.04064
   Detection Prevalence : 0.10397
      Balanced Accuracy : 0.90028

       'Positive' Class : Rejected
```

Figure 39 GBM (Stochastic Gradient Boosting)

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1019     1516
  Accepted      154    22337

               Accuracy : 0.9333
                 95% CI : (0.9301, 0.9363)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5188
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.40197
                 Recall : 0.86871
                     F1 : 0.54962
             Prevalence : 0.04687
         Detection Rate : 0.04072
   Detection Prevalence : 0.10129
      Balanced Accuracy : 0.90258

       'Positive' Class : Rejected
```

Figure 40: Black Boost.

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected      960     2670
  Accepted      213    21183

               Accuracy : 0.8848
                 95% CI : (0.8808, 0.8887)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.354
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.26446
                 Recall : 0.81841
                     F1 : 0.39975
             Prevalence : 0.04687
         Detection Rate : 0.03836
   Detection Prevalence : 0.14505
      Balanced Accuracy : 0.85324

       'Positive' Class : Rejected
```

Figure 41: GLM Boosted (Generalized Linear Model).

We got better results than bagging, actually close to the results from Random Forest, but not better.

4.6.3. Stacking

Finally, we will try stacking, it implies building different models, normally different types of them, that will be combined with a different model. The latest one trained to combine the selected models in best possible option, i.e., the last model will train which data needs to be sent to which model, so performance is optimal.

Combining models that do not have a high correlation between them will render better results [8]. We can collect all models trained in previous sections (with similar performances) and, based on their correlation and performance, assess which subset of them should be part of the stacked model:

We will select our previous models to train and then will use a KNN model to combine them. As always, we will use Caret package:

```
status <- which(names(tr)=="STATUS")
set.seed(171819)
trainControl <- trainControl(method="repeatedcv", number=5,repeats=2,
                             savePredictions="final",
                             classProbs=TRUE,
                             verbose=FALSE,
                             index=createFolds(tr[,status],5))
set.seed(171819)
models <- caretList(x = tr[,-status], y=tr[,status],
                    trControl=trainControl,
                    tuneList = list(
                        fda=caretModelSpec(method='fda', metric="ROC",tuneLength=5),
                        treebag=caretModelSpec(method='treebag', metric="ROC",tuneLength=5),
                        rf=caretModelSpec(method='rf', tuneLength=5, metric="ROC"),
                        knn=caretModelSpec(method='knn', preProc=c("center","scale"),metric="ROC"),
                        lda=caretModelSpec(method='lda', metric="ROC"),
                        glm=caretModelSpec(method='glm', metric="ROC"),
                        gbm=caretModelSpec(method='gbm', metric="ROC",tuneLength=5),
                        blackboost=caretModelSpec(method='blackboost', metric="ROC",tuneLength=5),
                        C5.0=caretModelSpec(method='C5.0', metric="ROC",tuneLength=5),
                        adaboost=caretModelSpec(method='adaboost', metric="ROC",tuneLength=5),
                        AdaBag=caretModelSpec(method='AdaBag', metric="ROC",tuneLength=5),
                        glmboost=caretModelSpec(method='glmboost', metric="ROC",tuneLength=5))
                    )
```

Notice this time parameter *sampling* is not set to *down* in the train control, the training set has been down-sampled before; this will save memory since models keep original trainset in memory.

Results from the training are below:

```
Call:
summary.resamples(object = results)

Models: fda, treebag, rf, knn, lda, glm, gbm, blackboost, C5.0, adaboost, AdaBag, glmboost
Number of resamples: 5

Accuracy
             Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
fda        0.8857  0.8909 0.8910 0.8905  0.8914 0.8935    0
treebag    0.8873  0.8875 0.8886 0.8893  0.8901 0.8933    0
rf         0.8943  0.8978 0.9022 0.9003  0.9023 0.9050    0
knn        0.8620  0.8621 0.8627 0.8640  0.8664 0.8669    0
lda        0.8482  0.8485 0.8512 0.8506  0.8524 0.8525    0
glm        0.8488  0.8527 0.8529 0.8539  0.8572 0.8578    0
gbm        0.8958  0.8974 0.8998 0.8991  0.9003 0.9024    0
blackboost 0.8785  0.8814 0.8839 0.8830  0.8849 0.8861    0
C5.0       0.8939  0.8943 0.8955 0.8959  0.8967 0.8993    0
adaboost   0.8964  0.8990 0.9010 0.9011  0.9032 0.9059    0
AdaBag     0.8883  0.8885 0.8891 0.8897  0.8898 0.8930    0
glmboost   0.8448  0.8462 0.8463 0.8469  0.8476 0.8495    0

Kappa
             Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
fda        0.7713  0.7817 0.7820 0.7810  0.7828 0.7870    0
treebag    0.7746  0.7751 0.7772 0.7787  0.7801 0.7865    0
rf         0.7886  0.7956 0.8043 0.8006  0.8046 0.8100    0
knn        0.7239  0.7242 0.7253 0.7280  0.7327 0.7338    0
lda        0.6963  0.6971 0.7024 0.7011  0.7048 0.7051    0
glm        0.6976  0.7054 0.7059 0.7078  0.7144 0.7157    0
gbm        0.7916  0.7948 0.7995 0.7983  0.8006 0.8049    0
blackboost 0.7570  0.7628 0.7679 0.7659  0.7698 0.7721    0
C5.0       0.7879  0.7886 0.7911 0.7919  0.7934 0.7985    0
adaboost   0.7929  0.7980 0.8020 0.8022  0.8065 0.8118    0
AdaBag     0.7767  0.7769 0.7783 0.7795  0.7796 0.7860    0
glmboost   0.6896  0.6923 0.6926 0.6937  0.6952 0.6989    0
```

Figure 42: Stacking training results.

As expected same results as before. Let’s see correlations between all of them:

	fda	treebag	rf	knn	lda
fda	1.0000000	-0.29481963	0.71407834	0.58608455	0.59098807
treebag	-0.2948196	1.00000000	-0.18288855	0.44436223	0.16680940
rf	0.7140783	-0.18288855	1.00000000	0.41500818	0.80699366
knn	0.5860846	0.44436223	0.41500818	1.00000000	0.79729892
lda	0.5909881	0.16680940	0.80699366	0.79729892	1.00000000
glm	-0.3436865	-0.06138428	-0.18679288	0.13441403	0.23166506
gbm	0.6090142	-0.17545054	0.98889125	0.31256651	0.76366334
blackboost	0.4035699	0.06851504	0.67973310	0.68901142	0.94589677
C5.0	0.3768228	-0.45404505	0.85040539	0.01913653	0.60439731
adaboost	0.6754745	-0.44611460	0.95732851	0.22379020	0.69304691
AdaBag	-0.1591491	0.98764729	-0.03602588	0.52914685	0.28380803
glmboost	-0.6270263	0.02505979	-0.44449171	-0.10444738	-0.05413778

	glm	gbm	blackboost	C5.0	adaboost
fda	-0.34368651	0.60901423	0.40356992	0.37682283	0.67547451
treebag	-0.06138428	-0.17545054	0.06851504	-0.45404505	-0.44611460
rf	-0.18679288	0.98889125	0.67973310	0.85040539	0.95732851
knn	0.13441403	0.31256651	0.68901142	0.01913653	0.22379020
lda	0.23166506	0.76366334	0.94589677	0.60439731	0.69304691
glm	1.00000000	-0.17955758	0.52414438	0.11559382	-0.09402574
gbm	-0.17955758	1.00000000	0.65175265	0.89255664	0.95491315
blackboost	0.52414438	0.65175265	1.00000000	0.63028551	0.62646351
C5.0	0.11559382	0.89255664	0.63028551	1.00000000	0.93477230
adaboost	-0.09402574	0.95491315	0.62646351	0.93477230	1.00000000
AdaBag	-0.12761389	-0.03602888	0.15159664	-0.35653827	-0.31427022
glmboost	0.94418345	-0.40774720	0.25536645	-0.07779296	-0.34630791

	AdaBag	glmboost
fda	-0.15914915	-0.62702633
treebag	0.98764729	0.02505979
rf	-0.03602588	-0.44449171
knn	0.52914685	-0.10444738
lda	0.28380803	-0.05413778
glm	-0.12761389	0.94418345
gbm	-0.03602888	-0.40774720
blackboost	0.15159664	0.25536645
C5.0	-0.35653827	-0.07779296
adaboost	-0.31427022	-0.34630791
AdaBag	1.00000000	-0.08146657
glmboost	-0.08146657	1.00000000

If we consider high correlation everything above 0.75:

```
> high_correlations <- findCorrelation(correlations,cutoff = 0.75,names = TRUE,verbose = TRUE)
Compare row 10  and column  3 with corr  0.957
  Means:  0.57 vs 0.425 so flagging column 10
Compare row 3  and column  7 with corr  0.989
  Means:  0.531 vs 0.401 so flagging column 3
Compare row 7  and column  5 with corr  0.764
  Means:  0.448 vs 0.372 so flagging column 7
Compare row 5  and column  8 with corr  0.946
  Means:  0.459 vs 0.357 so flagging column 5
Compare row 12  and column  6 with corr  0.944
  Means:  0.302 vs 0.345 so flagging column 6
Compare row 2  and column  11 with corr  0.988
  Means:  0.379 vs 0.344 so flagging column 2
All correlations <= 0.75
> high_correlations
[1] "adaboost" "rf"       "gbm"      "lda"      "treebag"  "glm"
```

Adaboost is highly correlated with :RF, GBM and C0.5:

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1057     1574
  Accepted      116    22279

               Accuracy : 0.9325
                 95% CI : (0.9293, 0.9355)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5249
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.40175
                 Recall : 0.90111
                     F1 : 0.55573
             Prevalence : 0.04687
         Detection Rate : 0.04224
   Detection Prevalence : 0.10513
      Balanced Accuracy : 0.91756

       'Positive' Class : Rejected
```

Figure 43: Adaboost

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1046     1531
  Accepted      127    22322

               Accuracy : 0.9337
                 95% CI : (0.9306, 0.9368)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5274
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.40590
                 Recall : 0.89173
                     F1 : 0.55787
             Prevalence : 0.04687
         Detection Rate : 0.04180
   Detection Prevalence : 0.10297
      Balanced Accuracy : 0.91377

       'Positive' Class : Rejected
```

Figure 44: Random Forest

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1020     1764
  Accepted      153    22089

               Accuracy : 0.9234
                 95% CI : (0.92, 0.9267)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4813
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.36638
                 Recall : 0.86957
                     F1 : 0.51554
             Prevalence : 0.04687
         Detection Rate : 0.04076
   Detection Prevalence : 0.11124
      Balanced Accuracy : 0.89781

       'Positive' Class : Rejected
```

Figure 45: GBM

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1035     1852
  Accepted      138    22001

               Accuracy : 0.9205
                 95% CI : (0.9171, 0.9238)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4748
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.35850
                 Recall : 0.88235
                     F1 : 0.50985
             Prevalence : 0.04687
         Detection Rate : 0.04136
   Detection Prevalence : 0.11536
      Balanced Accuracy : 0.90236

       'Positive' Class : Rejected
```

Figure 46: C5.0

We can select AdaBoost or RF, being the former one a little better. Notice RF perform worst than our first training, this is because tunelength was reduced (in all algorithms) in order decrease computational time.

Treebag is highly correlated with AdaBag; we will select the latest one:

```
          Reference
Prediction Rejected Accepted
  Rejected      891     1255
  Accepted      282    22598

               Accuracy : 0.9386
                 95% CI : (0.9355, 0.9415)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.507
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.41519
                 Recall : 0.75959
                     F1 : 0.53691
             Prevalence : 0.04687
         Detection Rate : 0.03560
   Detection Prevalence : 0.08575
      Balanced Accuracy : 0.85349

       'Positive' Class : Rejected
```

Figure 47: AdaBag

```
          Reference
Prediction Rejected Accepted
  Rejected     1043     1860
  Accepted      130    21993

               Accuracy : 0.9205
                 95% CI : (0.9171, 0.9238)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4768
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.35928
                 Recall : 0.88917
                     F1 : 0.51178
             Prevalence : 0.04687
         Detection Rate : 0.04168
   Detection Prevalence : 0.11600
      Balanced Accuracy : 0.90560

       'Positive' Class : Rejected
```

Figure 48: TreeBag

GLM is highly correlated with GLMBoost and AdaBoost, confusion matrixes show Adaboost (Figure 43) is a better performer than GLM and GLMBoost so we will drop both GLMs :

```
            Reference
Prediction Rejected Accepted
  Rejected      979     2666
  Accepted      194    21187

               Accuracy : 0.8857
                 95% CI : (0.8817, 0.8896)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3611
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.26859
                 Recall : 0.83461
                     F1 : 0.40639
             Prevalence : 0.04687
         Detection Rate : 0.03912
   Detection Prevalence : 0.14565
      Balanced Accuracy : 0.86142

       'Positive' Class : Rejected
```

Figure 49: GLM

```
            Reference
Prediction Rejected Accepted
  Rejected      958     2770
  Accepted      215    21083

               Accuracy : 0.8807
                 95% CI : (0.8766, 0.8847)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3442
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.25697
                 Recall : 0.81671
                     F1 : 0.39094
             Prevalence : 0.04687
         Detection Rate : 0.03828
   Detection Prevalence : 0.14897
      Balanced Accuracy : 0.85029

       'Positive' Class : Rejected
```

Figure 50: GLM Boost

Finally, we see LDA is highly correlated with many others, its performance is not good so will be drop:

```
Confusion Matrix and Statistics

            Reference
Prediction Rejected Accepted
  Rejected      950     2408
  Accepted      223    21445

               Accuracy : 0.8949
                 95% CI : (0.891, 0.8986)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.376
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.28291
                 Recall : 0.80989
                     F1 : 0.41933
             Prevalence : 0.04687
         Detection Rate : 0.03796
   Detection Prevalence : 0.13418
      Balanced Accuracy : 0.85447

       'Positive' Class : Rejected
```

Figure 51: LDA

Finally, we will train non-high correlated models with a KNN algorithm:

```
models_no_corr <- c(models$fda, models$treebag, models$rf, models$knn, models$blackboost, models$adaboost)
set.seed(171819)
fit.stack <- caretStack(models_no_corr,
                        method="knn",
                        metric="ROC",
                        trControl=trainControl,
                        tuneLength=10)
```

```
Confusion Matrix and Statistics

          Reference
Prediction Rejected Accepted
  Rejected     1057     1759
  Accepted      116    22094

               Accuracy : 0.9251
                 95% CI : (0.9217, 0.9283)
    No Information Rate : 0.9531
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4966
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.37536
                 Recall : 0.90111
                     F1 : 0.52996
             Prevalence : 0.04687
         Detection Rate : 0.04224
   Detection Prevalence : 0.11252
      Balanced Accuracy : 0.91368

       'Positive' Class : Rejected
```

Figure 52: Stacking results.

Although it has improved average performances, recall of 90%, it did not improve AdaBoost with a 90% of recall and 40% of precision. For this data it seems AdaBoost is the best model to be used, capturing 90% of rejected certificates and misclassifying 8% of accepted certificates.

5. Results

Due to computational problems, we have reduced the initial dataset. The first idea was to take certificates since 2011, but after doing some tests applying GLM algorithms, we have observed better results taking certificates from 2013, this helped on reducing computational time and also increasing performance results. More than four years of data seem to increase noise in the data.

Also, we have verified that performance results remain the same for a two repeated five cross-fold validations as for a three repeated ten fold cross validation. With this, we could reduce even more the computational time needed.

After this, the objective was getting a balanced dataset, and down-sampling was selected as the best candidate, not only delivering better results than adding synthetic observations (algorithm SMOTE) but also being much faster. Downsampling also highly reduced the dataset size, so improvements on speed were considerable high (from hours of training to minutes).

The results with several algorithms and previous modifications were as follows:

- **GLM** with a precision of 36% and Recall of 68%. Accuracy 92%
- **CART** with a precision of 42% and Recall of 73%. Accuracy 94%
- **KNN** with a precision of 32% and Recall of 87%. Accuracy 90%
- **RF** with a precision of 38% and Recall of 89%. Accuracy 93%
- **SVM** with a precision of 17% and Recall of 77%. Accuracy 81%

Having into account the nature of our problem, i.e., detecting the maximum number of potential rejected certificates (max Recall with a decent precision), we have chosen Random Forest as the most suitable algorithm.

When filtering data, we have removed some attributes with variance close to zero and highly correlated attributes; we observed better results than unfiltered data:

- **RF** with a precision of 42% and Recall of 88%. Accuracy 94%.

We got a bit less of Recall 88% versus 89%, but on the other hand, we got 42% of precision versus 38% we had before. Since we are interested in recall, we considered this an improvement over previous results.

Further attempts of applying ensemble mechanism yield slightly better performances. Bagging algorithms did not improve our Random Forest results, although Random Forest is actually a type of bagged mechanism. Boost algorithms, AdaBoost, did improve Random Forest by an increase in Recall (but decreasing a 2 on Precision). Results and stacking did not improve Random Forest or AdaBoost.

As final predictive model, we will keep the one given by **AdaBoost algorithm with 40% of precision** and **90% of recall**.

6. Conclusions

This work proves that it is possible to create an effective and, at great extent efficient, help-to-decision model for the system based on the data that it produces nowadays. With actual results, we can get a model, AdaBoost, able to **capture 90% of rejected certificates** and **misclassifying only 7% of accepted certificates**.

Although there is room for improvements (on data quality mainly), we see validation of the model shows high ratios of true positives predictions.

7. Future Lines of Work

In this work, I have considered only BIPs at the national level, Germany in this case, but there is the possibility of building models at European level and at lower levels (local authorities). The exact same process can be followed to verify it is possible to detect consignments that should be rejected, probably with a different set of attributes.

Another line of work not explored in this work is to incorporate a temporal variable, i.e. considering the different status, the certificate passes through as a time series classification problem. This is actually a very promising line of work since temporal variable would add a completely new type of information that could lead finding a model with even higher performance. But in order to follow such path, the system, TRACES, needs to be modified in order to represent real changes faithfully, actual temporal data is not useable.

Regarding changes in TRACES, a deep review of data input processes must be followed in order get data with higher data. During this work several issues with data consistency have been found, impacting the data available to be used with the classification model. We do not know how much this lack of quality has impacted actual results, but is clear that improving quality data will allow not only reduce noise (and possibly increase the model performance) but to include more attributes that could add useful information to the model.

References

[1] European Commission - TRACES, “TRACES: TRAde Control and Expert System,” [Online]. Available: http://ec.europa.eu/food/animals/traces_en.

[2] European Commission - Fact Sheet, “Enforcement of rules along the agri-food chain in the EU,” [Online]. Available: http://europa.eu/rapid/press-release_MEMO-17-611_en.htm.

[3] J. Brownlee, Machine Learning Mastery With R, 2017.

[4] P. K. Chan and S. J. Stolfo, “Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection,” 1998.

[5] P. Ravisankar, V. Ravi, G. R. Rao and I. Bose, “Detection of financial statement fraud and feature selection using data mining techniques,” 2011.

[6] M. Artís, M. Ayuso and M. Guillén, “Detection of automobile insurance fraud with discrete choice models and misclassified claims,” 2002.

[7] Y. Bouzembrak and H. J. Marvin, “Prediction of food fraud type using data from Rapid Alert System for Food and Feed (RASFF) and Bayesian network modelling,” *Food Control,* vol. 61, 2016.

[8] M. Kuhn and K. Johnson, Applied predictive modeling, Springer, 2013.

[9] P. McCullagh, “Generalized linear models,” 1984.

[10] W.-Y. Loh, “Classification and regression trees,” 2011.

[11] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” 2000.

[12] G. Cauwenberghs and T. Poggio, “Incremental and decremental support vector machine learning,” 2001.

[13] C. Chen, A. Liaw and L. Breiman, “Using random forest to learn imbalanced data,” 2004.

[14] C. Shearer, The CRISP-DM model: the new blueprint for data mining, 2000.

[15] THE COUNCIL OF THE EUROPEAN UNION, “COUNCIL DIRECTIVE 97/78/EC,” *Official Journal of the European Communities,* 1997.

[16] European Commission, “The Combined Nomenclature,” [Online]. Available: https://ec.europa.eu/taxation_customs/business/calculation-customs-duties/what-is-common-customs-tariff/combined-nomenclature_en.

[17] MariaDB Corporation Ab, “MariaDB,” [Online]. Available: https://mariadb.com/.

[18] The R Foundation, “The R Project for Statistical Computing,” [Online]. Available: https://www.r-project.org/.

[19] RStudio , “RStudio,” [Online]. Available: https://www.rstudio.com/.

[20] M. Kuhn, “The caret Package,” [Online]. Available: http://topepo.github.io/caret/index.html.

[21] X. P. Monné, “TRACES: Annual report 2015,” 2015. [Online]. Available: https://ec.europa.eu/food/sites/food/files/animals/docs/traces_report_annual_2015_final_eng.pdf.

Annex I. Used certificate fields

Certificate

Column name	Description	Comments
ID_SNAP	The unique identifier. Sequence : CVEDP_SEQ	Yes, to group by certificate and identify each certificate
VERSION	Certificate's version used to prevent simultaneous updates	It is for internal control, not real version
ARRIVAL_BIP	Estimate date arrival at BIP	No, this date is introduced by the user, but it is an estimation, not a real value
CONFORM_EU_REQUIREMENT	Indicate if the consignment is conform to the EU requirement - 0 = false - 1 = true	Yes
COUNTRY_CONSIGNED/Commodity tab	The ISO2-Code of the country where the consignment is consigned	Yes
COUNTRY_ORIGIN/Commodity tab	The ISO2-Code of the country of origin	Yes
CONTROL_ID/Part II	The unique identifier of the CVEP control, link to CVEDP_control table	No, only to retrieve data from this table
DECISION_ID/Part II	The unique identifier of the CVEDP decision, link CVED	No, only to retrieve data from this table
DECLARATION_DATE/ Responsible for load, references	Certificate's declaration date	Yes

Column name	Description	Comments
INTERNAL_MARKET, see Purpose	Indicate which type of internal market the consignment is for. If it's applicable - human - animal - pharma - technical - other	Yes
TRANSPORT_INTERNAL_CODE, traders tab	Indicate the type of transport - other - plane - rail - road - ship	Yes, how the consignment is travelling
TRANSPORT_INTERNAL_IDENT	Indicate the identification of transport	Yes, it can be repetead
TRANSPORT_INTERNAL_DOC	Indicate the document of transport	No, many nulls and it does not add value, it is a number given by the operator
NON_CONFORMING_CONSIGNMENT, see Purpose	Indicate the non-conforming consignment - customs - free - supplier - ship	??Yes
NUMBER_OF_PACKAGES, commodity tab	Indicate the number of packages	Yes
PRODUCT_GROSS_WEIGHT	Indicate the gross weight of a package	Yes
PRODUCT_NET_WEIGHT	Indicate the net weight of a package	Yes

Column name	Description	Comments
PRODUCT_TEMPERATURE	Indicate the temperature of the product - chilled - frozen - ambient	Yes
PURPOSE	Indicate the type of purpose - internalmarket - nonconforming - tranship - transit - import	Yes
REFERENCE_NUMBER	Certificate's reference number. (Unique)	Yes
REGISTER_NUMBER	Indicate the register number in case of non conforming consignment	No, it is random number
SHIP_PORT	Indicate the ship port in case of non conforming consignment	No
STATUS	Certificates status - 0 = not set - 1 = new - 2 = deleted - 3 = rejected - 4 = pre-validated - 5 = valid - 6 = cancelled - 7 = draft - 8 = in progress - 9 = animo - 10 = recalled - 11 = replaced	Yes

Column name	Description	Comments
TRANSHIPMENT_3TH_COUNTRY, see purpose	Indicate the ISO2-Code of the transhipment third country if its applicable	yes
TRANSIT_3TH_COUNTRY, see purpose	Indicate the ISO2-Code of the transit third country if it's applicable	yes
TYPE_OF_PACKAGES_OLD	Indicate the type of packages	no
USER_ID	The unique identifier of the user who signed this certificate. Only filled in if the certificate is created by a transitair user	Yes
VET_DOC_DATE	Date of the veterinary's document	It is free text introduced by the user, this id does not belong to Traces
VET_DOC_NUMBER	The number of the veterinary's document	It is a date introduced by the user, it does not belong to TRACES
CONSIGNEE_ID, traders tab	The unique identifier of the consignee business.	yes, get data from other tables
CONSIGNOR_ID, traders tab	The unique identifier of the consignor business.	yes, get data from other tables
IMPORTER_ID, trades tab	The unique identifier of the business responsible for the import.	yes, get data from other tables
DELIVERY_ID, traders tab, delivery address	The unique identifier of the business where the products are delivered.	yes, get data from other tables
LOAD_PERSON_ID, references tab RFL	The unique identifier of the business responsible for the consignment (4.).	yes, get data from other tables
CUSTOMS_NUMBER/LOCAL REFERENCE NUMBER, references tab	The Local reference number (I.2) attributed by the local authorities.	??
CREATION_DATE	Creation date of the record.	No, date when certificate was created, it does not add any value
LAST_CHANGE_DATE	Date of last modification of the record.	It does not add value

Column name	Description	Comments
IMPORT_ID, see purpose	The unique identifier of the IMPORT certificate. (if generated from an import certificate)	Yes, if certificate is created from an import document
REPLACING_ID, check, if this field is not null don't take this certificate, only mark has been replaced.	Id of the certificate replacing this cancelled certificate.	no
REPLACING_REF	Reference number of the certificate replacing this cancelled certificate (for display).	no
REPLACED_ID	Id of the certificate replaced by this certificate.	no
REPLACED_REF	Reference number of the certificate replaced by this certificate (for display).	no
CERTIFICATE_VERSION	Version number of the certificate (before validation).	yes
SUBMITTER_BUSINESS_ID	The unique identifier of the certificate's business submitter, used to link a draft certificates to its owner	yes, to get data from other table
SUBMITTER_AUTH_ID	The unique identifier of the certificate's authority submitter, used to link a draft certificates to its owner	yes
SUBMITTER_RCA_ID	The unique identifier of RCA of the certificate's submitter, used to link a draft certificates to its owner	yes
SUBMITTER_CCA_ID	The unique identifier of CCA of the certificate's submitter, used to link a draft certificates to its owner	yes

Column name	Description	Comments
DEPARTURE_DATE	The departure date of the transport	??
TRANSPORTER_ID	Unique identifier of the business found in the field "Transporter"	yes
TRANSPORT_EXTERNAL_CODE	Indicate the type of transport: - other - plane - rail - road - ship	yes
TRANSPORT_EXTERNAL_IDENT	Identification of the transport	yes
TRANSPORT_EXTERNAL_DOC	Document of the transport	no
REPLACED_DATE	Date on which the certificate has been replaced by another one.	no
REPLACING_DATE	Date on which the certificate has been created for replacing another one.	no
PREVIOUS_CVEDP_ID	CVEDP ID of the parent certificate (for certificates resulting from a split operation). Same information as the related CVEDP_DECISION.PREVIOUS_CVED_NUMBER, but created at consignment creation in order to make this value available at decision time. Consignments and decision might be created at diff times.	no

Column name	Description	Comments
STATUS_DATE	Date on which the certificate status has been changed	It does not add valuable information to the model,
STATUS_USER	User who changes the status of the certificate, Indicate the status - 0 = new, 1 = confirmed, 2 = valid, 3 = inactive, 4 = deleted, 5 = rejected, 6 = body_suspend (when the related authority has been suspended)	yes
TRANSHIPPED_DATE	Date on which the certificate has been transhipped to another place	Not value
TRANSHIPPING_DATE	Date on which the certificate has been created for transhipping another one.	Not value
CREATION_ORIGIN	Technical Origin of this certificate: Online: online, B2B: b2b, New-Zealand Tool: nz	yes
EXPORT_ID	The unique identifier of the corresponding EXPORT_ID	yes

Complements

Column name	Description	Comments
COMMODITY_COMPLEMENT_ID	The unique identifier of the complements associated to this certificate.	Needed information
CREATION_DATE	Creation date of the record	
CVEDP_ID	The unique identifier of the concerned certificate.	
LAST_CHANGE_DATE	Date of last modification of the record	
POSITION	Indicate the position of the parameter in the list (starting with 0). It stores the order wherein cn codes have been added	
SUBTOTAL_NET_WEIGHT	Subtotal of Net Weight for this complement id, on this certificate	Needed information to improve the model

Decision

Column name	Description	Comments
ACCEPTABLE	Indicate if the consignment is acceptable or not - 0 = false - 1 = true	Not needed, it belongs to Part II of the certificate
ACCEPTCHANNELLED	Indicate the action if consignment is accepted for channel - article8 - article15	Not needed, it belongs to Part II of the certificate
ACCEPT_MARKET_FREE_CIRCULATION	Indicate which type of free circulation, the consignment is accepted for - human - animal - pharma - technical - other	Not needed, it belongs to Part II of the certificate
ACCEPT_SPECIFIC_WAREHOUSE	Indicate which type of specific warehouse, the consignment is accepted for - customs - free - supplier - ship	Not needed, it belongs to Part II of the certificate
CONTROLLED_DESTINATION	Unique identifier of the business where the control has been made.	Not needed, it belongs to Part II of the certificate
CONTROL_DATE	Date of the decision	Not needed, it belongs to Part II of the certificate

Column name	Description	Comments
CREATION_DATE	Creation date of the record.	Not needed, it belongs to Part II of the certificate
CREATION_ORIGIN	Technical Origin of this decision: Online: online, B2B: b2b, New-Zealand Tool: nz	Not needed, it belongs to Part II of the certificate
CUSTOMS_DOCUMENT_REFERENCE	Decision's customs document reference	Not needed, it belongs to Part II of the certificate
CVED_SUBSEQUENT_NUMBERS_OLD	Decision's subsequent numbers	Not needed, it belongs to Part II of the certificate
DOCUMENTARY_CHECK	Result of the check of document - 1 = satisfactory - 2 = not satisfactory	Not needed, it belongs to Part II of the certificate
ID	DB ID of the decision	Not needed, it belongs to Part II of the certificate
IDENTY_CHECK	Result of the check of identity (of given type) - 1 = satisfactory - 2 = not satisfactory	Not needed, it belongs to Part II of the certificate
IDENTY_CHECK_FULL	Type of the check of identity - 6 = Full identity check - 5 = Seach check	Not needed, it belongs to Part II of the certificate
LAST_CHANGE_DATE	Date of last modification of the record.	Not needed, it belongs to Part II of the certificate
NOT_ACCEPTABLE_ACTION	Indicate the action if the consignment is not accepted - destruction - reexport - transformation	Not needed, it belongs to Part II of the certificate
NOT_ACCEPT_DATE	Indicate the date of the non-acceptable action	Not needed, it belongs to Part II of the certificate
OFFICIAL_VETERINARIAN	The unique identifier of the Official Veterinarian (Authority) who made de decision	Not needed, it belongs to Part II of the certificate
OFFICIAL_VET_FIRST_NAME	Copy of the official veterinarian's first name who made the decision	Not needed, it belongs to Part II of the certificate
OFFICIAL_VET_LAST_NAME	Copy of the official veterinarian's last name who made the decision	Not needed, it belongs to Part II of the certificate

Column name	Description	Comments
PHYSICAL_CHECK	Result of the physical check - 1 = satisfactory - 2 = not satisfactory - 7 = not done	Interesting to know if a physical check has been carry out on the consignment
PHYSICAL_CHECKNOT_DONE	- 8 = Reduced checks regime - 9 = Other	Not needed, it belongs to Part II of the certificate
PREVIOUS_CVED_NUMBER	Decision's previous CVED number	Not needed, it belongs to Part II of the certificate
RASFF_INFORMATION_ID_OLD	The unique identifier of the RASFF form	Not needed, it belongs to Part II of the certificate
REFUSAL_COUNTRY	Indicate the country if the certificate has been refused because of non-approved country	Not needed, it belongs to Part II of the certificate
REFUSAL_ESTABLISHMENT	Indicate the establishment if the certificate has been refused because of non-approved establishment	Not needed, it belongs to Part II of the certificate
TEST_EXECUTED	Indicate if the laboratory test has been executed - 0 = false - 1 = true	Not needed, it belongs to Part II of the certificate
TEST_EXECUTED_DATE	The test laboratory date	Not needed, it belongs to Part II of the certificate
TEST_MOTIVATION	Indicate the motivation to execute the test - random - suspicion - reinforced	Not needed, it belongs to Part II of the certificate
VERSION	Decision's version	

Refusal Reasons

Column name	Description	Comments
CREATION_DATE	Creation date of the record.	
CVEDP_DECISION_ID	The unique identifier of the decision of the CVED for Products	No add value
LAST_CHANGE_DATE	Date of last modification of the record.	
REASON	"Indicate the refusal reason - nocertificate - country - establishment - products - document - error - physical - chemical - biological - other "	

Cities

Field name	Description	Comments
AUTHORITY_ID	The unique identifier of the local authority which is responsible for this city	Yes, authority of entry ID
COUNTRY_CODE	The country code where the city is located	Yes, to know the authority and business countries
CREATION_DATE	Creation date of the record.	Not add value
GEO_SOURCE	Source of the geolocation coordinates (0: Initial, 1: Provided by Member State, 7: Copied from the city of the LVU)	Not add value
ID	Unique identifier; Sequence : CITIES_SEQ	Only to query the table, not needed
LAST_CHANGE_DATE	Date of last modification of the record.	Not add value
LATITUDE	Latitude of the city	
LONGITUDE	Longitude of the city	
NAME	City's name	Yes, city name of authority and business
POSTAL_CODE_REGION	City's postal code	Yes, postal code of authority and business
QUALITY	Reflects how accurate are the coordinates (longitude/latitude) of the city, set in function of the origin of the geo information	Not add value
STATUS	"City's status	Not add value

Authority

The most important fields have been retrieved from authority table.

Field name	Description	Comments
ALTERNATE_CODE	Authority alternate code. Used, for instance, to link an UNLOCODE to a Super LVU	
CITY_ID	The unique identifier of the city where the authority is located	
CODE	Authority code	
CREATION_DATE	Creation date of the record.	
EMAIL	The e-mail address	
FAX	The fax number	
FLAG1	"If subclass L: restricted If subclass V: official If subclass D: border inspection post Else: not used 0=false 1=true"	
FLAG2	"If veterinary: restricted Else: not used 0=false 1=true"	
ID	Unique identifier. Sequence : AUTHORITY_SEQ	Entry EU authority ID
LAST_CHANGE_DATE	Date of last modification of the record.	
NAME	Authority name	Entry EU authority name
PARENT_ID	The authority parent unique identifier	
PHONE	The phone number	

STATUS	"Authority status - 0 = not set - 2 = deleted - 3 = suspended - 5 = valid"	
STREET	Authority address	
SUBCLASS	"Indicate the authority type: L = LVU V = Veterinary D = Customs office (Douane) C = CCA R = RCA"	This is important to know which kind of authority
TIME_ZONE	Time Zone of this authority	
VERSION	Version used to detect simultaneous updates	
VETERINARY_CONTROL_ALLOWED	Only valid for veterinary (subclass = V). If 1, this veterinary can control certificates that concern him.	
VETERINARY_MANUALLY_ASSIGNED	Only valid for veterinary (subclass = V). If 1, EO can select this veterinary while submitting an IntraTrade.	
WEB	Authority's Internet address	

Business

Field name	Description	Comments
BUSINESS_ID	The unique identifier of the business	Id of the concerned business: consignee, consignor.
CITY_NAME	The name of the city where the business is located.	We have the postal_code

CODE	The official code of the business.	With id is enough, this one is not mandatory
COUNTRY_CODE	The country where the business is located.	Valuable to be studied
CREATION_DATE	Creation date of the record.	Not important
ID	The unique identifier of the CVED for Products	
LAST_CHANGE_DATE	Date of last modification of the record	Not important
NAME	The name of the business.	Included in the model
POSTAL_CODE	The postal code of the city where the business is located.	
STREET	The street and number where the business is located.	
TYPE	The type of the business.	

Annex II. Integrity and quality data tests

Outcome	ID	cn number	NAME CITY AUTHO	CITY_ID	POSTAL CODE CITY AUTHO	ID_AUTHORIT	NAME_AUTHO	CODE AURHORITY	ID CONSIGNEE	NAME	POSTAL CODE	COUNTRY CODE	TYPE
Exit	2827884	4	Frankfurt Am Main, Stadt		60549	1395	Frankfurt/Main	DEFRA4	12307108	STAR IMPEX	44339	DE	etablish
Real	2827884		Frankfurt Am Main, Stadt	6017	60549	1395	Frankfurt/Main	DEFRA4	12307108	STAR IMPEX	44339	DE	etablish
Exit	2871346	2	Frankfurt Am Main, Stadt	6017	60549	1395	Frankfurt/Main	DEFRA4	12561574	ELAFOOD	94626	FR	etablish
Real	2871346		Frankfurt Am Main, Stadt	6017	60549	1395	Frankfurt/Main	DEFRA4	12561574	ELAFOOD	94626	FR	etablish
Exit	2866375	1	Bremen, Stadt		28207	1391	Bremen	DEBRE1	12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish
Real	2866375	1	Bremen, Stadt	3887	28207	1391	Bremen	DEBRE1	12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish

ID CONSIGNOR	NAME	POSTAL CODE	COUNTRY CODE	TYPE	ID IMPORTER	NAME	POSTAL CODE	COUNTRY CODE	TYPE	ID LOAD PERSON	NAME	POSTAL CODE	COUNTRY CODE	TYPE
12307109	St John's Sea Foods	Tamil Nadu	IN	exporter	12307220	STAR IMPEX	44339	DE	etablish	12307222	STAR IMPEX	44339	DE	importer
12307109	St John's Sea Foods	Tamil Nadu	IN	exporter	12307220	STAR IMPEX	44339	DE	etablish	12307222	STAR IMPEX	44339	DE	importer
12561575	DAREL CO INC	Massachusetts	US	exporter	12561576	ELAFOOD	94626	FR	etablish	12561578	Nagel Airfreight GmbH	60549	DE	responsible

12561575	DAREL CO INC	Massachusetts	US	exporter	12561576	ELAFOOD	94626	FR	etablish	12561578	Nagel Airfreight GmbH	60549	DE	responsible
12532055	BRF - BRASIL FOODS S.A.	Santa Catarina	BR	exporter	12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish	12532056	Preuss Logistik GmbH	28197	DE	importer
12532055	BRF - BRASIL FOODS S.A.	Santa Catarina	BR	exporter	12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish	12532056	Preuss Logistik GmbH	28197	DE	importer

ID SUBMITTER	NAME	POSTAL CODE	COUNTRY CODE	TYPE	ID TRANSPORTER	NAME	POSTAL CODE	COUNTRY CODE	TYPE	DELIVERY ID	NAME	POSTAL CODE	COUNTRY CODE	TYPE	decision_id	CONTROL DATE DECISION
										12307221	STAR IMPEX	44339	DE	etablish	2625531	03/02/2011
										12307221	STAR IMPEX	44339	DE	etablish		03-Feb-11
										12561577	PERISHABLE CENTER FRANKFURT	60549	DE	warehouse	2666644	04/03/2011
										12561577	PERISHABLE CENTER FRANKFURT	60549	DE	warehouse		04-Mar-11
										12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish	2661957	01/03/2011
										12532054	Allfein Feinkost GmbH & Co. KG	49393	DE	etablish		01-Mar-11

PHYSICAL CHECK DEC	COMMODITIES	Result
1	1416 1436 1440 11036	☑
1	1416 1436 1440 11036	☑
1	1436 1440	☑
1	1436 1440	☑
1	10931	☑
1	10931	☑