POLITECNICO MILANO 1863

DEPARTMENT OF MATHEMATICS

FINANCIAL ENGINEERING

Market Implied Ratings

Project RM3

Federica Maddaloni 10579853
Sebastian Castellano 10582405
Stefano Tomassetti 10747792

UNDER THE SUPERVISION OF

PROF ALDO NASSIGH

Academic Year: 2020/2021

1 Introduction

Even though the role of credit rating agencies (CRAs) such as Standard & Poor's (S&P) and Fitch is to provide credit ratings for different institutions, they are often late in signaling a modification of their credit status. Thus, the objective of this project is to deduce a system of implied credit ratings from the bond and credit default swap markets that aims at anticipating future migrations.

This idea is backed by many studies, which show that market spreads hold valuable information regarding credit ratings and rating migrations. The limit of these studies is that their results come from the analysis of the bond and the CDS markets separately, while this project aims to reach a more accurate implied rating system by joining them in a bi-dimensional framework, extracting meaningful information from each market. We resort to modern classification methods such as Support Vector Machines (SVM) and eXtreme Gradient Boosting (XGB) to deduce the market-implied rating which is associated to each couple of daily spreads.

Once obtained the new system, whenever an implied rating is different from the official one, it is predicting a migration towards the new one.

2 Dataset

The given dataset is composed by S&P credit ratings, daily 5-year maturity CDS and bond spreads of 35 worldwide sovereign issuers, covering the time interval 2008–2018. These sovereign issuers are 24 European Countries (Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom) and 11 countries outside the European continent (Australia, Canada, Hong Kong, Japan, Malaysia, New Zealand, Russia, Singapore, Turkey, United States and Venezuela).

The ratings used for the analysis are S&P's because they are considered to bear more timely information about migrations with respect to other agencies.

3 Data Preprocessing

The given dataset is split among three files:

- `rating_SP`: a spreadsheet containing for each of the 35 sovereign countries the daily S&P rating for each business day in the period between 02-01-2008 and 21-05-2019 (*Date* column, 102,792 elements). The data is grouped alphabetically following the Bloomberg ticker (*BBG_TICKER* column) associated to each country. In some isolated days the ratings are missing (NR), but this issue will be tackled in the `clean_data` function.

- `bnd_cds_5year`: a spreadsheet containing for each date between 02-01-2008 and 31-12-2018 (2869 rows) the Bond Spread and CDS spread for each of the 35 countries. Each countries' spreads have their own column which is labeled as the respective spreads' series name. In some years, for some countries such as Greece or Venezuela, the CDS or Bond Spreads are not available due to the condition of the country, which is close to default.

- `Catalogo` : a spreadsheet containing information on the 35 issuers, most importantly their Bloomberg tickers, CDS series name, Bond Spread series name (creating a link to navigate between the first two files).

To sort out the imprecisions of the dataset and reshape it according to our needs, we implemented two specific functions:

- `clean_data`, that modifies the ratings file in order to:
  - Remove the excess data outside the time window of interest (2008-2018);

- Transform the rating labels to simplify the model, so that only 5 rating classes are considered: AAA, AA, A, BBB, BB and B together. All intermediate ratings are grouped together (for example ratings such as AA+ and AA- are considered just as AA). Each of these five levels of rating are labeled from 1 (AAA) to 5 (B). The ratings under B are labeled as 0 and will be discarded in the training part of the classification.
- Replace with the previous most recent available rating the isolated days where the rating is not available (NR);
- Solve the problem of duplicates: for all countries, the first 10 dates of 2014 are repeated because of the merge of two different datasets. At first, duplicate rows are dropped, but for some countries (Croatia, Ireland, Netherlands, Romania, Turkey) the ratings for the repeated days are different because one of the datasets does not contain information on the notches. After a double-check on *www.worldgovernmentbonds.com*, the wrong ratings are deleted directly on the excel files trying to match the actual ratings and to avoid the presence of fictitious migrations.
- Add the dates which are missing in "*rating_SP*" but present in "*bnd_cds_5year*". The associated rating is obtained by finding the previous most recent available one, like before.

- `reshape_spreads`, to reshape the spreads file in order to have a unique column for all countries' spreads, since the formats of the two main dataset files are not easily comparable. In this way, for each country and for each day on the same row there are the corresponding ratings and spreads.

After reordering the data, we obtain a total of 100415 data points matching daily CDS, Bond Spreads and S&P ratings for the 35 countries.
To save the time needed for data preprocessing, we stored the reordered data in `rating.csv`.

4 Model Explanation and Implementation

Due to the nature of the problem, we try to derive the daily time series of implied ratings using the so-called Support Vector Machine (SVM). This machine learning's technique is commonly used as it allows to classify a dataset by finding a separating hyperplane (both for separable and non-separable classes) and because of its ability to work with a large number of data [1]. We implement the model with two different kernels: the linear kernel, which (as its name suggests) looks for linear separating boundaries, and the radial one, which allows for non-linear boundaries.

In case of $K > 2$ classes (multi-class classification), there are two possible approaches: one-vs-rest, which compares each class with all the others transforming the problem into K simple binary classifications, and one-vs-one, where the problem is decomposed into $K(K-1)$ binary classifications and each class is compared with all the others one by one. To be consistent with our reference paper, we use a one-vs-one approach.

To include two different markets in the same analysis and obtain a better classification, we work with a bi-dimensional classification problem, i.e., each daily Rating provided by S&P is associated to a vector x which is composed by a CDS spread and a Bond spread.

As any other model used in machine learning, a SVM requires the dataset to be split into a training and a validation set (in our case no prediction is carried out, so no test set is defined). Before proceeding with the training, we make sure to standardize our data to obtain a better performance, using scikit's `StandardScaler`.

[1] T. Hastie, R. Tibshirani, J. Friedman: The elements of Statistical Learning, pp. 417-455, 2009

As we are dealing with daily time series, we follow what done by Nassigh et al.[2] selecting a training time window of 66 days, since it has been proved that the CDS spreads do not contain any information on migrations for a period longer than three financial months (Ismailescu, Kazemi 2010)[3].

To avoid overfitting, in the function `split_train_val` we perform a cross-validation sampling on the countries, taking the 80% of them for the training and the remaining 20% for the validation, i.e., each country is used either for training or for validation. The over-fitting can be indeed caused by the daily observations of each specific country which tend to fall close to each other in the spreads' space.
In this random selection we check that in the period of interest all the ratings of the country are above CCC, otherwise all the data of said country is removed from the dataset.

A crucial part of machine learning techniques is the tuning of the hyperparameters, which are the parameters not learnt by the algorithm, but that need to be set in advance during the definition of the model. Eventually, we expect a combination of hyperparameters that make the model behave optimally and give the best results possible.

The linear kernel requires only the tuning of C, the penalty parameter of the error term in the non-separable case. It controls the trade-off between having a smooth decision boundary and classifying the training points correctly.
In the radial kernel case, a second parameter must be tuned: γ. It is a control of the precision of fitting the training set, so high values of γ can mean that the boundaries assume more complicated shapes in order to fit the data. In general, too high values of both parameters can lead to overfitting.

We perform the tuning of the hyperparameters through the function `tuning_c` every 10 days using `RandomizedSearchCV` and setting reasonable values in the range $[0.1, 100]$ for C and $[10^{-4}, 10]$ for γ (both ranges are in log scale). The randomized grid search works picking randomly a number n of values (we use $n = 20$) and selecting the one that fits better the data of the training set with a k-fold cross-validation (we use $k = 5$).

For every day in the time window, we train the model on a different (random) set and using the method `predict` we classify each x belonging to the validation set (that consists of around 7 countries chosen randomly). In order to retrieve the implied rating of the day of analysis for every country, we implement the function `implied_ratings`, that selects all the spreads and CRA's ratings of the day for all the countries and, using again `predict`, classifies the x giving as output the implied ratings. At each iteration, the implied ratings are appended into an array that is consistent with the data structure that we have decided to use.
For some specific days, we plot the Decision Surface through the function `plot_decision_surface` (Figure (1)).

Despite the cost functions and the optimization being constructed not to overfit the data, we may end up with a poor classification anyway because of the nature of the problem. Indeed, the delay of official ratings with respect to the market leads to the use of spurious information and thus to misclassification errors that affect deeply the reliability and the performances of the model.

To overcome this intrinsic issue, we try to use the SVM for a purely predictive strategy (SVM*), repeating the analysis with a differently posed problem: for each couple (CDS_spread, $Bond\_spread$), the associated rating is the one corresponding to $h*$ days later. The aim of this approach is to predict the implied ratings at a future horizon ($h*$) through classification by feeding as input this new shifted dataset, and to reach a better performance of the classifiers, having set aside the

[2] A. Nassigh, T. Colozza, S. Marmi, D. Regoli: Bond-CDS implied rating systems, 2020

[3] Ismailescu, I. and Kazemi, H. (2010). The reaction of emerging market credit default swap spreads to sovereign credit rating changes. Journal of Banking and Finance, 34:2861– 2873

less timely official ratings.

Selecting a proper value for $h*$ is not a trivial problem though. For this reason, we employ the value proposed by the reference paper of this project as the one that optimizes the forecast performances ($h* = 22days$, which correspond roughly to one financial month).

Intrigued by the strength of eXtreme Gradient Boosting algorithm and by its great performance in different tasks, we try to implement it for our problem using the package `xgboost`. Instead of looking for the separating hyperplane as SVM, XGB is based on decision trees and random forests.

The split between the training and validation sets is performed through the same function `split_train_val` as for the SVM implementation, while for the tuning of the hyperparameters we have to define a new function: `tuning_xgb`. Indeed, the XGB has a lot more hyperparameters than SVM, almost all with a significant role in preventing overfitting.

After some attempts to understand its computational demand – that results being very high - we decided to apply `RandomizedSearchCV` with a halved number of samples (`n_iter` = 5) to a relatively small grid of parameters that we deem more impactful in the overfitting prevention:

- *eta* (learning rate) $[0.01 - 0.1]$: determines the step size at each iteration;
- *maximumdepth* $[3 - 10]$: maximum depth of a tree, e.g., higher depth allows the model to learn relations specific for a particular sample;
- *minimumchildweight* $[0 - 10]$: minimum sum of weights of all observations required in a child, e.g., higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree, but too high values may lead to underfitting;
- *subsample* $[0.5 - 1]$: fraction of observations to be randomly sampled for each tree, e.g., lower values make the algorithm more conservative (prevent overfitting), but too small values may lead to underfitting.

Thus, with the tuned hyperparameters we go on with the fitting of the model and get back the implied ratings through the usual function `implied_ratings`.

There is room for improving this model either by taking more values from the range of the parameters we already put in for tuning, or by increasing the number of samples to check (`n_iter`), or by adding more hyperparameters to tune, but taking into account an exponential growth of computational time.

Given the amount of time required to run the whole script for the implementation of the various models, we created a .csv file for each one where we store the implied ratings obtained, ready to use in order to assess their peformances (`implied_rating.csv`, `implied_rating_RBF.csv`, `implied_rating_asterisco.csv`, `implied_ratingRBF_asterisco.csv`,`implied_rating_XGB.csv`).

DISCLAIMER: the run of the main scripts of each model (`Run_RM3_SVM.py`, `Run_RM3_SVM_ast.py`, `Run_RM3_XGB.py`) can be avoided since all the meaningful results are stored into the aforementioned .csv files, so only `Run_RM3_Performance.py` can be run.

5 Forecast Performances of Implied Ratings

To compute the performance indexes proposed by the paper, two additional functions are defined: `prec_rec_f1` and `compute_Ns`.

Before defining the indexes used to assess the performance of our ML models, given a forecast horizon h, we have to define the quantities:

$$N_i(t+h) = sign[r_i(t+h) - r_i(t)] \tag{1}$$

$$\hat{N}_i(t+h) = sign[\hat{r}_i(t) - r_i(t)] \tag{2}$$

and we can notice that a positive $N_i(t+h)$ indicates an actual downgrade for country i in the period $(t, t+h]$, while a positive $\hat{N}_i$ (t) indicates that the implied rating system predicts a downgrade.

Also, the following sets need to be introduced:

- the set of issuers whose rating movement in t+h is correctly predicted: $N = \{(i,t) : \hat{N}_i(t) = N_i(t+h), t \in T; i \in I\}$
- the sets of issuers whose rating movement is predicted as α: $\hat{N}_\alpha = \{(i,t) : \hat{N}_i(t) = \alpha, t \in T; i \in I\}$
- the sets of issuers whose actual rating movement in $t+h$ is α: $N_\alpha = \{(i,t) : N_i(t+h) = \alpha, t \in T; i \in I\}$

with $\alpha = \{0, +1, -1\}$, where 0 indicates no migration (stability), +1 a downgrade and -1 an upgrade.

Finally, the performance indexes are the following:

- $accuracy = |N|/n$: number of rating movements correctly predicted over the total number of samples considered (where n is the total sample size $T|I|$)
- $precision_\alpha = \frac{|N \cap N_\alpha|}{|\hat{N}\alpha|} = 1 - P(I\ type\ error)$: number of rating movements correctly predicted as α over the total number of predictions of α
- $recall_\alpha = \frac{|N \cap N_\alpha|}{|N\alpha|} = 1 - P(II\ type\ error)$: number of rating movements correctly predicted as α over the total number of actual α
- $F1_\alpha = \frac{2 * Recall_\alpha * Precision_\alpha}{Recall_\alpha + Precision_\alpha}$

The matrices N and $\hat{N}$ are built with the countries on the columns and the dates on the rows, taking care of the indexes corresponding to implied ratings associated to NaN values of spreads or to CCC (or CCC+ or CCC-) actual ratings by setting them to NaN.

Then, these indexes are computed importing the vectors of implied ratings from the aforementioned .csv files and taking out the first H days (being H the training window, the implied rating for the first H dates are not available). Here $\hat{N}$ indicates if a migration has been predicted to happen at time t; N indicates if there has been an actual migration at time $t+h$.
For the SVM*, the implied ratings are plugged into the matrix `spreads_sort` at the respective position of the actual ratings. This leaves the first 22 ratings of each country with no implied rating (namely with a NaN). This is the reason why the accuracies are computed dividing by $n - 770$.

Another useful index we employ to assess the prediction ability of our different models is the average discrepancy between official and implied ratings of the migrating issuer in a backward time window W before the migration, defined as

$$\hat{\nu} := \frac{1}{|W|} \sum_{s \in W} (\hat{r}_i(s) - r_i(s)) \tag{3}$$

and computed through the function `compute_v`.
After having selected the indexes of migrations, for each one we compute the $\hat{\nu}$ with a backward time window of 22 days ($W_1 = [0, 22]$, $W_2 = [22, 44]$ and $W_3 = [44, 66]$).

We also define a function `plot_early_warnings` to represent graphically the time interval from 10 days before to 5 days after a migration, comparing the actual rating (in blue) with the implied one (in red). The discussion of the results of this analysis are carried out in the next section.

6 Discussion of Results

One of the main advantages of using the ML techniques is that they do not require any specific financial background, nor the development of complex analytical models. So, they are simple to use and they come in handy when there is no time for the implementation of new models. Once having invested in powerful devices to sustain the demanding computational side, this kind of tools is very flexible and can be exploited for problems of any kind.

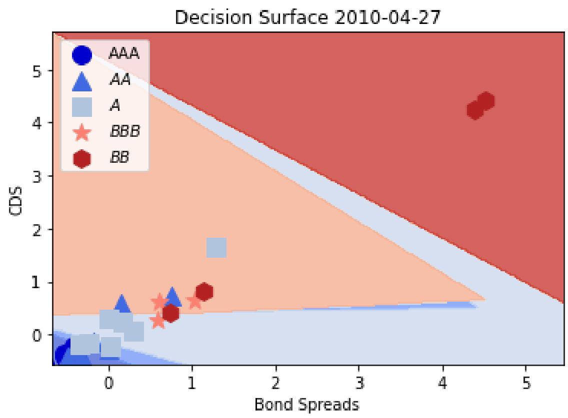
In terms of coding and computational cost, the implementation of the SVM and the SVM* is equivalent. On the other hand, the implementation of XGB is more challenging since it has a lot more hyperparameters which first need to be understood properly. Moreover, from a computational point of view it requires a significant amount of time more than SVM, reason that led us to reduce the number of hyperparameters to tune, the dimension of the grid and the number of iterations of the random grid search.

Thus, since it would require the use of much more powerful devices, we cannot exploit the potential of the model at its fullest and we will have to consider this aspect when comparing the results of the different ML techniques.

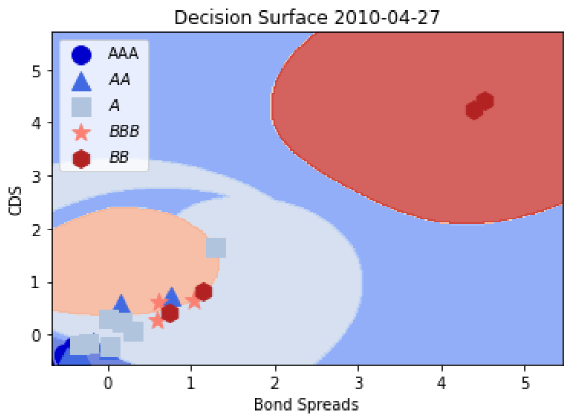
In the plots below (Figure 1), the different shapes of the separating boundaries can be appreciated. In particular, the distinction between the linear and the radial kernel for the SVM becomes clearer: in the first case, the shape is linear, while in the latter the decision surface is more adapted to the data (but this may lead more to overfitting). The separation in the case of XGB, instead, is more irregular due to the fact that it is based on regression trees that, in general, lead to non-linear irregular shapes.

An important remark is that even if in the plots there are points which fall far from the region of the plane where they are expected to be, it is not necessarily a mistake of the model since they may indicate that it is predicting a migration.

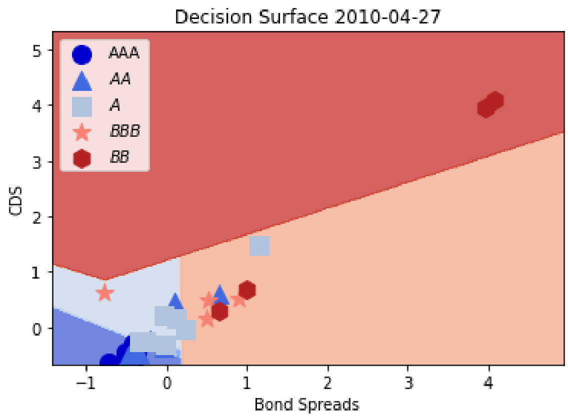
It is also interesting to highlight that, indeed, the classifications obtained using SVM with the usual approach and those obtained through SVM* are deeply different despite being carried out on the same day. This is a visual proof of how associating different ratings to the same couple of spreads actually leads to different results.



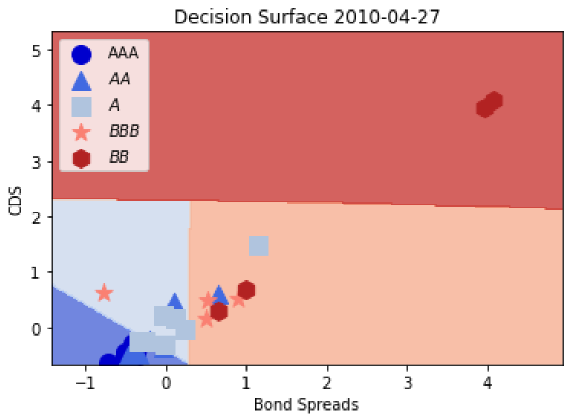
(a) Plot of the decision surface for SVM with linear kernel



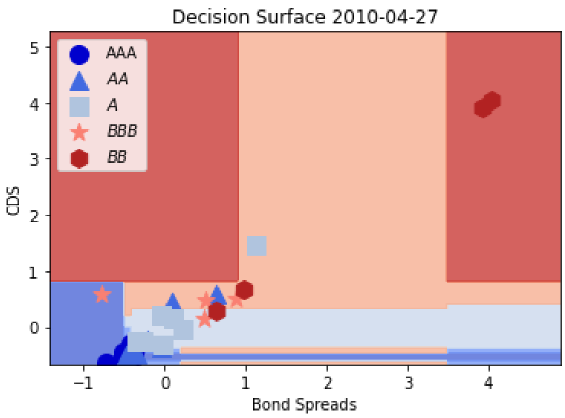
(b) Plot of the decision surface for SVM with radial kernel



(c) Plot of the decision surface for SVM* with linear kernel



(d) Plot of the decision surface for SVM* with radial kernel



(e) Plot of the decision surface for XGB

Figure 1: Plot of the decision surface obtained through different classifiers.

Accuracy	1 m	3 m	6 m	12 m
SVM	65.1	63.92	61.97	58.24
RBF	70.72	69.13	66.46	61.46
SVM*	65.83	64.47	62.51	58.78
RBF*	70.77	68.98	66.41	61.53
XGB	79.74	77.58	74.22	68.05

Precision (%)	Up				Stable				Down			
Pred horizon	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m
SVM	0.61	1.88	3.49	6.98	98.6	95.54	90.96	82.63	2.64	8.01	14.83	24.36
RBF	0.46	1.62	3.06	6.43	98.6	95.49	90.72	81.83	2.15	6.81	12.13	20.52
SVM*	0.74	2.03	3.8	7.65	99.46	96.32	91.67	83.44	3.22	8.61	15.33	24.43
RBF*	0.82	1.89	3.32	7.07	99.48	96.25	91.25	82.34	3.58	8.64	14.09	21.85
XGB	0.73	1.99	3.47	7.14	98.44	95.24	90.48	81.71	1.84	6.18	10.95	19.02

Recall (%)	Up				Stable				Down			
Pred horizon	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m
SVM	54.98	54.69	52.38	54.9	66.7	66.95	67.22	67.95	45.92	45.05	43.04	38.88
RBF	32.03	36.36	35.43	38.98	72.62	72.86	72.99	73.27	34.01	34.78	32	29.77
SVM*	63.64	56.5	54.55	57.54	66.45	66.67	66.91	67.78	57.48	49.62	45.62	39.98
RBF*	55.84	41.4	37.59	41.93	72.31	72.47	72.45	72.76	55.44	43.3	36.45	31.08
XGB	28.57	25.31	22.73	24.5	82	82.18	82.33	82.75	22.11	24.01	21.96	20.97

F1 (%)	Up				Stable				Down			
Pred horizon	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m	1 m	3 m	6 m	12 m
SVM	1.21	3.63	6.54	12.38	79.57	78.73	77.31	74.58	4.99	13.61	22.06	29.95
RBF	0.91	3.11	5.64	11.04	83.63	82.65	80.89	77.31	4.05	11.38	17.59	24.29
SVM*	1.46	3.92	7.1	13.5	79.67	78.8	77.35	74.8	6.1	14.67	22.95	30.32
RBF*	1.62	3.68	6.1	12.1	83.75	82.69	80.77	77.25	6.7	14.41	20.33	25.66
XGB	1.42	3.7	6.02	11.05	89.47	88.23	86.21	82.23	3.4	9.83	14.62	19.95

Table 1: Tables of Performance Indexes. The first table reports the Accuracy index for each model and each prediction horizon (1 month, 3 months, 6 months, 12 months). The other three report, respectively, the Precision, Recall and F1-score in the cases of rating upgrade, downgrade and stability for each model and each prediction horizon (1 month, 3 months, 6 months, 12 months).

From the tables above we can compare the performances of the various methods. For the linear SVM it is evident that there is a low precision but a higher corresponding recall for both the up and down cases. From this we can deduce that this classifier tends to signal more migrations than those effectively happened.

The RBF kernel SVM tends to underperform with respect to the Linear SVM: this may be due to overfitting, since the boundaries are more complex and more adapted to the official ratings.

From the results of the SVM* approaches, it seems that the intuition was correct since the overall results are consistently better than the first implementation of the SVM.

The XGB does not improve the performances of the SVM in the up and down cases, but it turns out to be the best method for the stable one, even though the stable case is the best classified by all methods, since the dataset is highly unbalanced towards the absence of migration. Summing up, the linear SVM* is the best among the method used: it reaches the highest precision for all the different time windows and it performs better than the others in terms of up and down recall, thus resulting the one with the greatest F1 score.

As expected, precision and recall for the up and down cases increase as the prediction horizon increases, while they decrease for the stable one.

Generally, it can be seen easily that the up and down cases have very low performances. This is due to the dataset which is poor in terms of migrations. In particular for the up case the situation is even worse as the CDS capture better downgrades rather than upgrades[4] [5] [6] [7] and the number of upgrades is lower than the number of downgrades.

		SVM			RBF			SVM*			RBF*			XGB		
Date	**Country**	**I1**	**I2**	**I3**	**I1**	**I2**	**I3**	**I1**	**I2**	**I3**	**I1**	**I2**	**I3**	**I1**	**I2**	**I3**
27/10/08	**ROMANI**	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1
19/01/09	**SPAIN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21/01/09	**PORTUG**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30/03/09	**IRELND**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16/12/09	**GREECE**	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0
27/04/10	**GREECE**	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1
23/11/10	**IRELND**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16/12/10	**CHINA-HongKong**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24/03/11	**PORTUG**	1	1	1	0	0	1	1	1	1	1	0	1	1	1	0
01/04/11	**IRELND**	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1
29/07/11	**CYPRUS**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
05/08/11	**USGB**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24/08/11	**CZECH**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21/12/11	**REPHUN**	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0
13/01/12	**AUST**	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
13/01/12	**CYPRUS**	1	1	1	1	1	0	0	1	1	1	0	1	1	1	0
13/01/12	**FRTR**	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1
13/01/12	**ITALY**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13/01/12	**PORTUG**	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
13/01/12	**SLOVEN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13/01/12	**SPAIN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26/04/12	**SPAIN**	1	1	0	1	0	0	1	1	1	1	0	1	1	1	0
01/01/14	**NETHRS**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01/01/14	**ROMANI**	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
01/01/14	**TURKEY**	1	1	1	0	1	0	1	1	1	1	0	0	1	1	1
06/01/14	**IRELND**	1	1	1	0	0	0	1	1	1	1	1	0	1	0	1
15/01/14	**CROATI**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14/07/14	**FINL**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12/12/14	**BGARIA**	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
16/09/15	**JAPAN**	0	1	0	1	1	0	0	1	1	0	1	0	0	0	0
20/11/15	**NETHRS**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29/06/16	**UKIN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19/07/16	**TURKEY**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20/07/16	**TURKEY**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21/07/16	**TURKEY**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21/09/16	**REPHUN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15/09/17	**PORTUG**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22/09/17	**CHINA-HongKong**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
05/12/17	**BGARIA**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26/03/18	**SPAIN**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17/09/18	**CYPRUS**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Migrations detected out of 41		36	36	34	32	33	30	35	36	35	34	32	32	33	33	32

Table 2: Table of the Migrations Detected. It reports whether each model manages to signal each S&P's migration for each of the three time windows. '0' indicates that the migration was not detected, '1' that it was. The last row of the table reports the total number of migrations correctly identified by each model.

[4] Hull, J., Predescu, M., and White, A. (2004). The relationship between credit default swap spreads, bond yields, and credit rating announcements. Journal of Banking and Finance, 28:2789–2811

[5] Norden, L. (2017). Information in cds spreads. Journal of Banking and Finance, 75:118– 135

[6] Galil, K. and Soffer, G. (2011). Good news, bad news and rating announcements: An empirical investigation. Journal of Banking and Finance, 35:3101–3119

[7] Lee, J., Naranjo, A., and Velioglu, G. (2018). When do cds spreads lead? rating events, private entities, and firm-specific information flows. Journal of Financial Economics, 130:556–578

Through the prediction table (Table 2), we can finally assess whether the methods are able to predict the migrations that actually occur according to the official ratings.
There are some cases in which our classifiers fail badly:

- USA in 2011, which all methods fail to predict for every prediction interval. This confirms the study of Blau and Roseman (2013), whose conclusions are that this event was due to systemic reasons, rather than idiosyncratic. Being trained on market data, none of the ML techniques is expected to be able to catch pieces of information beyond what they are fed.

- Turkey in July 2016, which may be related to some issues of the dataset. While a migration from BB+ to BB is indicated on *worldgovernmentbonds.com* on the 20^{th} (not detectable by our model for how it is built), our `rating_sp` shows ratings oscillating between BB+ and BBB- before settling on BB+. There are, therefore, two fictitious migrations and for this reason the ML models fail in identifying the migration of the 20/07/2016.

- Bulgaria on 12/12/2014, that is hardly detected by the models as the migration has been registered only by S&P and so, as stated by Nassigh et al, it may be related to other factors beyond this analysis.

- Japan in 2015 and Hungary in 2011. We notice that for both the countries the migration is only of a single notch, which could be the cause of a misclassification likely due to minor changes in the market data. The same issue can be the reason behind the bad performance for the migration from A- to BBB+ of Greece of December 2009.
  Anyway, after a check on other countries we found that, for instance, also Cyprus migrates of a notch in 2011 (downgrade) and 2018 (upgrade), but the models manage to catch them. This apparent contradiction may be due to several aspects like the different financial periods with corresponding different market behaviours, or the different views of the countries involved that may have contributed to the migration (political situation, economic outlooks, etc.), that are factors that go beyond the sole market.

Another interesting result is the presence of multiple downgrades of several European countries including Portugal, Italy and Spain on the 13^{th} January 2012, almost all correctly predicted by every model and interval. On that day, the credit rating of nine EU countries has been downgraded as a consequence of the Sovereign Debt Crisis. It looks intriguing that each approach manages to catch the overall trend of a time period, even with different prediction horizons.

In general, as expected, the classifiers work better for short-term prediction horizons due to the higher uncertainty that grows as the the time windows move further away.

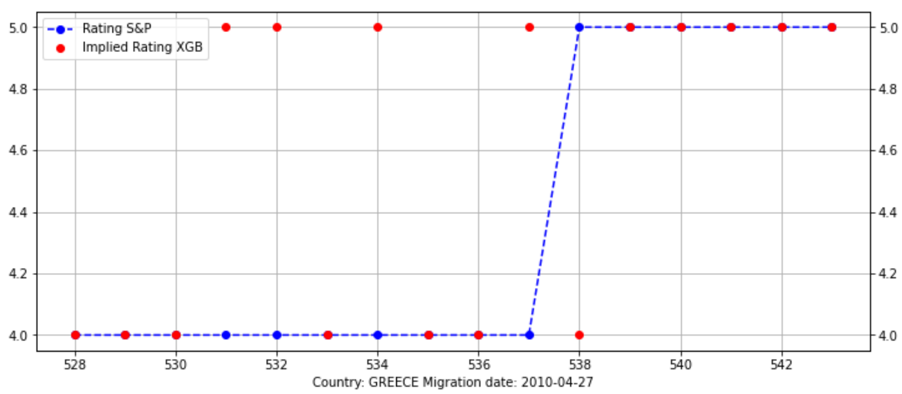


Figure 2: Example of a weak early detection of a migration using XGB

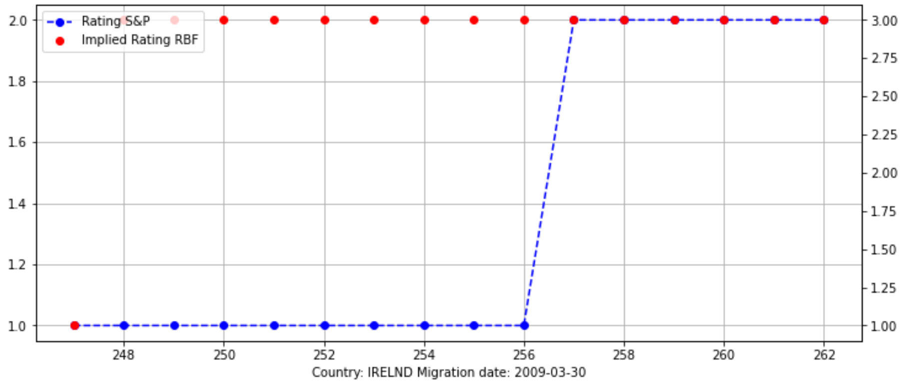


Figure 3: Example of a correct anticipation of a migration using radial SVM

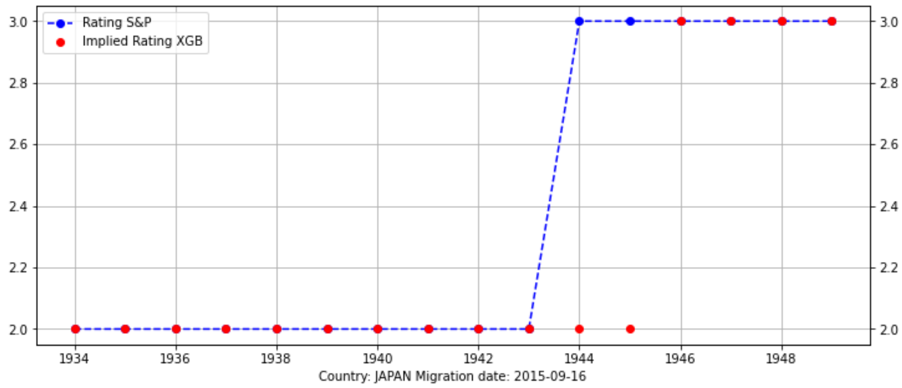


Figure 4: Example of a late detection of the migration using XGB

7 Conclusions

From the analysis portrayed in this project, the final ranking based both on the ability of detecting the migrations and the performance indexes is the following:

1. SVM* with linear kernel
2. SVM with linear kernel
3. SVM* with radial kernel
4. SVM with radial kernel
5. XGB

Overall, as Okkam's razor suggests, the linear models outperform their radial counterparts. One out of many weaknesses of every model is their low performance scores in the up and down cases, but this may be due to the nature of the dataset: out of 100415 total observations, there are only 41 migrations.

As a consequence, we would expect any of our models to perform much better on a more balanced dataset, with (many) more migrations with respect to the total. For instance, we would expect a significant improvement redoing this analysis on a dataset based on corporate issuers[8] [9].

Moreover, as already mentioned, the models suffer from the lack of additional knowledge about the overall situation of the countries considered (e.g. USA in 2011, Bulgaria in 2014), in particular when the ratings are below B. Indeed, in this latter case we do not have enough information for the classification because of unavailability and of illiquidity of market spreads.

[8] Acharya, V. V. and Johnson, T. C. (2005). Insider trading in credit derivatives. Journal of Financial Economics, 84:110–141

[9] Qiu, J. and Yu, F. (2012). Endogenous liquidity in credit derivatives. Journal of Financial Economics, 103:611–631

So, other significant features may be helpful for a more accurate analysis.

Another criticality is that the standard deviation of the spreads corresponding to a single rating increases in periods of crisis (see Figure 5).[10] This supports the use of a ML technique as the overlapping of ratings makes the classification problem too complex to be held using thresholds and it is the reason behind the training procedure used. Anyway, it cannot be neglected that the bigger range of spreads may lead to a more volatile classification which may weaken the significance of the migrations' signals provided by the implied ratings.

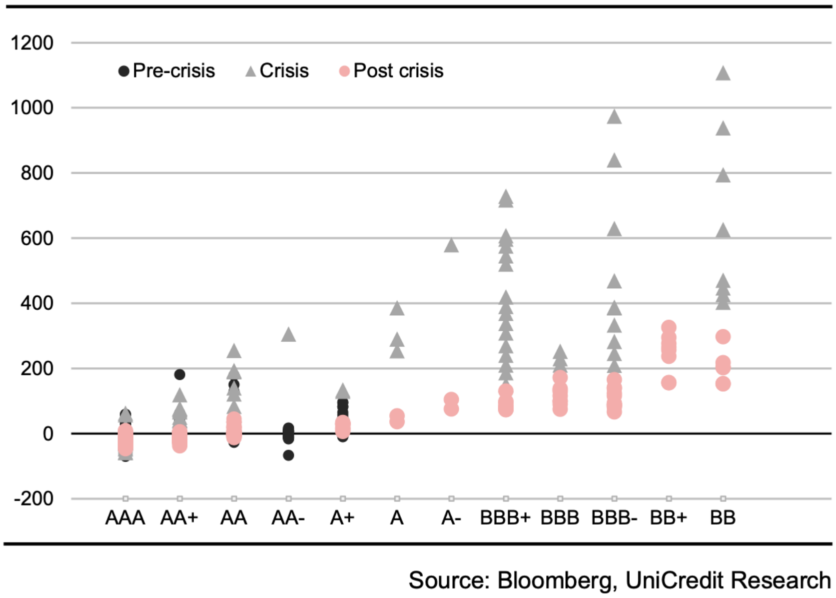


Figure 5: Credit Spreads vs Ratings. Source Unicredit, Bloomberg

If the analysis is extended to non-sovereign obligors (and thus improved), its implications are several. First of all, it could be applied in risk management to anticipate changes in the official ratings of a generic obligor and hedge open positions exposed to these kinds of credit events.

Another application could be to help financial institutions determine their regulatory capital in a more efficient way.

Eventually, it could be interesting to see the social impact for particular events. For instance, we can notice that all the approaches correctly predict the downgrade of the UK in 29/06/2016, 6 days after the referendum for the Brexit, also for longer prediction horizons. This leads to interesting questions, like: can these predictions describe properly the market players' feelings towards an obligor approaching an impactful decision? Can this information influence people's points of view and reflect on their choices?

Projecting these hypothesis on a smaller environment such as a company, is it possible that having prior information on its future rating evolution could affect the behaviour of the board of directors?

[10]The empirical trade-off between rating and credit spreads, by Dr. Luca Cazzulani, Deputy Head of FI Strategy (UniCredit Research, Milan)