Summary Report
Data Challenge: Non Performing Loans

Technical overview

Executive summary

This document summarises the work done toward the Data Challenge Non Performing Loans. It is intended to guide the technical review of the code, the methodology and the general strategy adopted to solve the challenge. Additional details might be found in the Jupyter notebooks.
Two supervised machine learning algorithms, trained on the provided historical dataset, are employed to estimate the likelihood to repay and the recovery rate in a 12 months ahead window. The model predictions are then used to re-rank NPL customers. Given the NPL sample as of 09-2017, the expected 12 months recovery rate is between 1.6M€ and 6M€.

Challenge objective

These requirements can be split into two separate tasks:

1. Estimate the likelihood to repay, i.e. the probability that a paiement is performed;
2. Estimate the recovery rate, i.e. the amount recovered w.r.t. the total debt;

Both tasks require the definition of a time horizon since the portfolio manager will be interested to know the likelihood to repay and the estimated recovery rate that should be expected within a certain time window from the moment he/she evaluates the NPL. The definition of time horizon is rather arbitrary, i.e. should be the portfolio manager to suggest which window he/she is interested to look at. For the sake of this challenge, I consider a 12 months window.

Task [1] is implemented with a classification model to be trained against a boolean target variable defined as TRUE in the case any income is received in the next 12 months, FALSE otherwise. The trained classifier will output the probability of Target==TRUE (or Target==FALSE), which can be interpreted as the likelihood of repayment.

Task [2] is implemented with regression model trained on the realised recovery rate 12 month ahead. The trained regressor will provide estimates of the expected recovery rate in the next 12 months.

Target variables

The implementation of both tasks is then based on supervised models and require the definition of target variables to be used in model training.

For a given customer and at a given time, the target variable is defined by considering all the possible amount (*incassi*) that will be received in the next 12 months, divided by the Gross Book

Value (GBV) of the debt at that given date. More technically, the target variable is defined as the 12 month ahead rolling recovery rate:

$$Target(t) = \frac{\sum_{i=}^{t+12} Incassi(i)}{GBV(t)}$$

The target variable is named 'RECOVERY_RATE_12M_AHEAD', it expected to take any continuous value from 0 (no recovery) to 1 (full repayment) and will provide the magnitude of the recovery in the next 12 months.
A boolean target variable named 'HAS_RECOVERED_12M_AHEAD' is also defined to indicate if any income would be possible in the 12 next months (TRUE) or not (FALSE).

Only NPL with a 12 months look ahead are used in the model training/testing. NPLs with a time horizon smaller than 12 months does not allow to define the target values since the information 12 month ahead is not available. NPLs without 12 month horizon are used as prediction sample, i.e. NPLs which the portfolio manager will be interested to know how to treat.

Features selection

I decided to focus on the feature that show a correlation with either of the two target variable bigger than 5%. This choice might seem arbitrary but it is actually based on a measurable and reproducible approach, because only the features that bring measurable information and prediction power with respect to the target variables are considered.

Another approach could have been to take in all the features and leave the model the task to rank them and highlight the most informative. I personally do not like this black-box approach, at least when there is a way to deal with the feature selection in a measurable way.

The following features are considered in the first iteration (most of them are engineered in the notebook *data_cleaning.ipynb*):

1. IMP_GBV
2. IMP_CUSUM_INCASSI
3. MAX_IMP_GBV
4. RECOVERY_RATE_TOTAL
5. NUM_MONTHS_SOFFERENZA
6. Contabile linea interessi di mora
7. Credito vantato
8. NUM_ENTI_SEGNALANTI
9. NUM_COMPONENTI_COINTESTAZIONE
10. NUM_ID_RAPPORTO
11. NUM_ID_GARANTE
12. IMP_INCASSI_ROLLING_12M
13. COD_TIPO_NDG

Classifier

For the classification task, I use the selected features to train a classification model to predict the target variable: 'HAS_RECOVERED_12M_AHEAD'.

I test three models:

1. **Logistic regression**;
2. **Extra random trees classifier**;
3. **Random forest classifier**.

All models are tested with and without PCA for dimensionality reduction using 4 components, which is the minimum number of components that explain the variance of the data.

For each model, the performance are evaluated using different metrics:

- **Precision**: proportion of correctly predicted instances (P = TP / (TP + FP));
- **Recall**: proportion of actual prediction that are correctly identified (R = TP / (TP + FN));
- **F1**: harmonic mean of Precision and Recall;
- **ROC AUC**: measure the probability the classifier perform better than a random choice.

The metrics are calculated on the test sample as well as on the train sample using a 3-fold cross-validation to test for model overfitting.

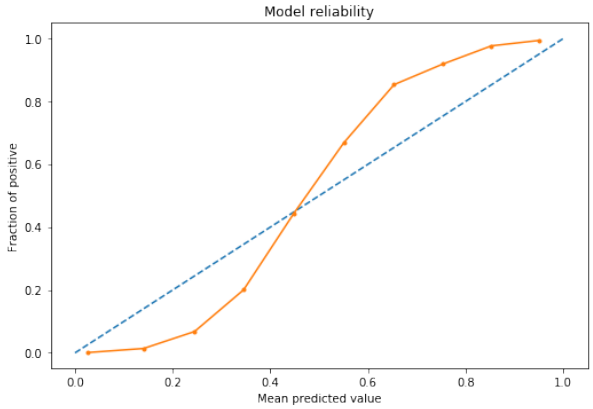
Aside from the numerical metrics, some additional handle on the model performance is given by the plots provided for each model tested:

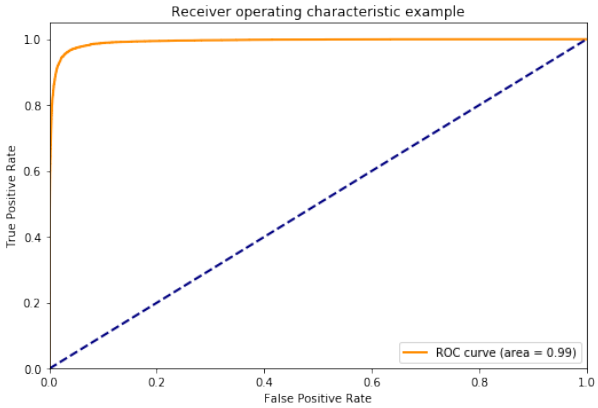
- **Model reliability curve**: check if the classifier output can be interpreted as an estimate of the probability to belong to the given class;
- **Receiver operating characteristic**: shows the performance of the classifier as its discrimination threshold is varied. It also quantify the distance from the performance provided by random choice;
- **Confusion matrix**: overview of the classification accuracy by counting the number of True/False Positive/Negative instances;
- **Classification probability**: distribution of the probability for the True and the Negative class;
- **Precision-Recall curve**: shows the performance of the classifier, in term of Precision and Recall as its discrimination threshold is varied;
- **Feature importance**: the importance given by the model to the features used during the model training (features are enumerated by their position in the feature vector).

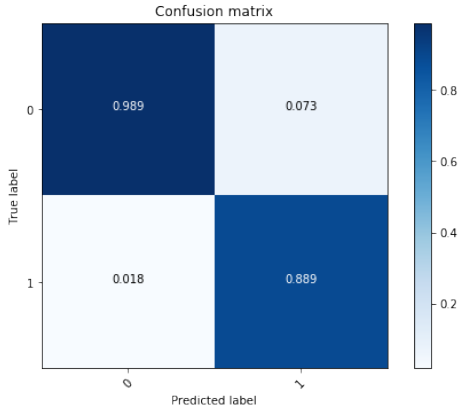
The best model is found to be the Random Forest classifier trained without PCA. The performance are shown in the following table (values within parentheses represent the performance measured on the train sample). The figure on the next page summarises graphically the model performance. The performance for all the models considered and the comparison among them is available in the notebook *modeling_and_performance_part1.ipynb*.

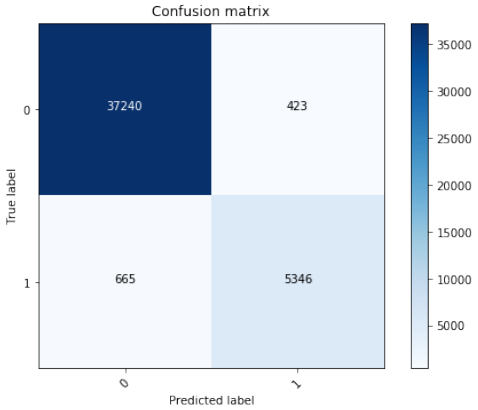
F1	**Precision**	**Recall**	**ROC AUC**
0.85 (0.91)	0.89 (0.93)	0.81 (0.89)	0.98

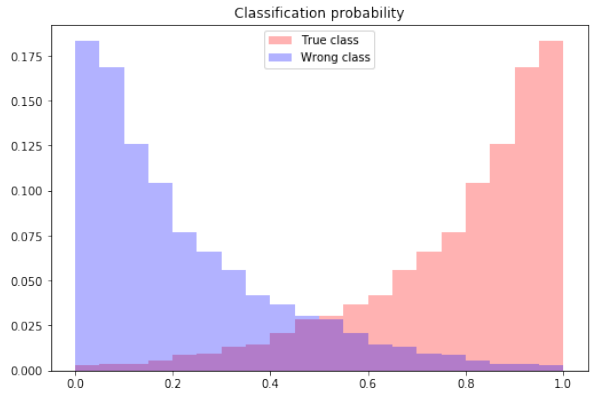
Classification report: Random Forest

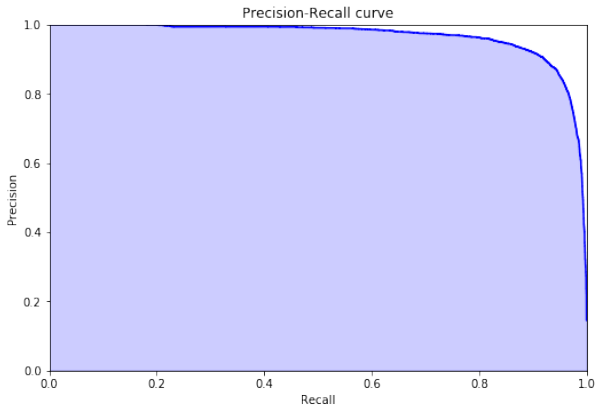


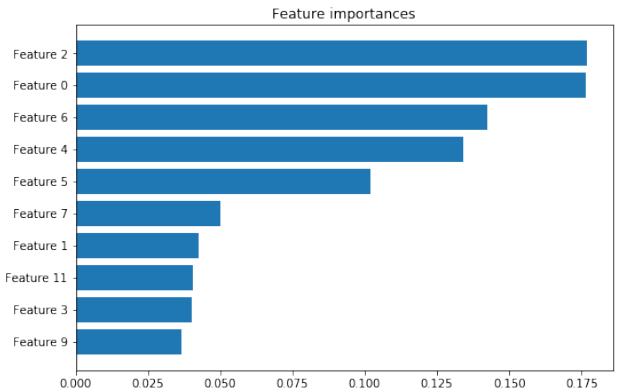












Regression

For the regression task, I use the selected features to train a regression model to predict the target variable 'RECOVERY_RATE_12M_AHEAD'.

I test two models:

1. **Linear regression**;
2. **Random Forest regression**.

All models are tested with and without PCA for dimensionality reduction using the minimum number of components which explain the data variance.

For each model, the performance are evaluated using different metrics:

- **R^2**: represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance;
- **MSE**: measures the average of the squares of the errors.

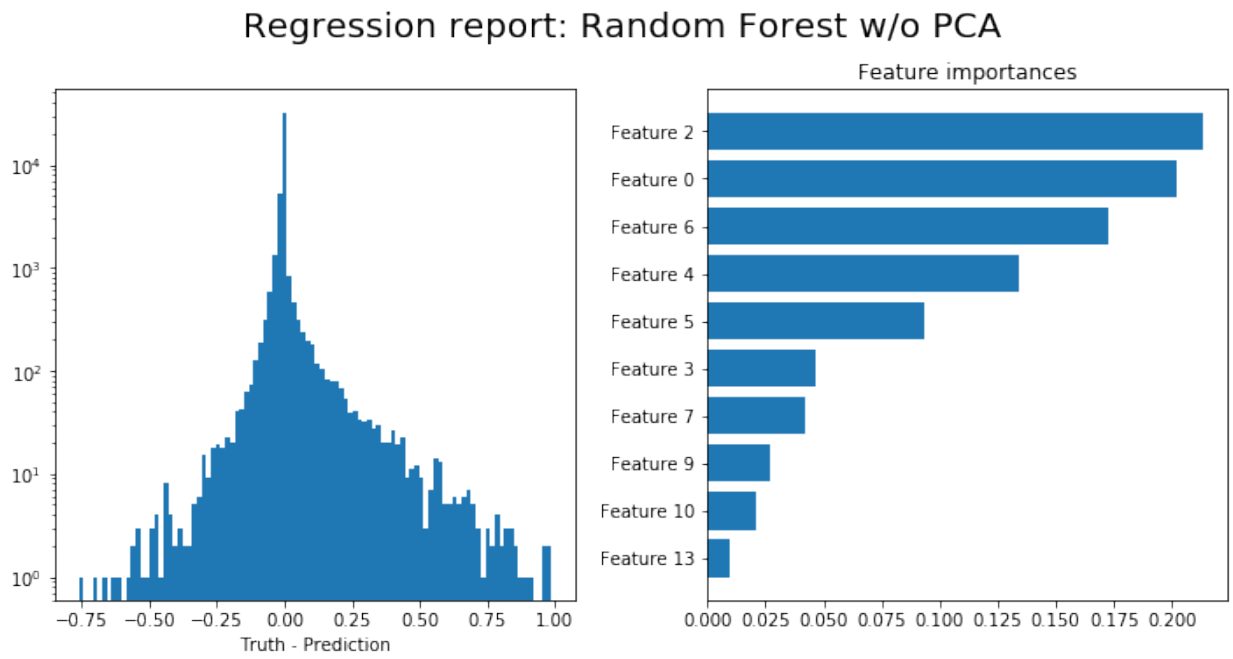
The metrics are calculated on the test sample as well as on the train sample using a 3-fold cross-validation to test for model overfitting.

Aside from the numerical metrics, some additional handle on the model performance is given by the plot provided for each model tested:

- **Residual distribution**: distribution of the residual, i.e. True Value - Predicted Values for each sample;
- **Feature importance**: the importance given by the model to the features used during the model training (features are enumerated by their position in the feature vector).

The best model is found to be the Random Forest regressor trained without PCA. The performance are shown in the following table (values within parentheses represent the performance measured on the train sample). The figure on the next page summarises graphically the model performance.

R2	**MSE**
0.661 (0.777)	0.005 (0.003)



Models improvement

The feature COD_TIPO_NDG divide the type of customers in three different categories: *Persona Fisica*, *Persona Giuridica* and *Cointestazione*. These categories are found in many different contexts and, as for experience, tend to behave differently. In this case, different customer types are expected to undertake different types of loans, are exposed to different types of risks and are driven by different idiosyncratic dynamics.

Following the recommendation given in the summary of *modeling_and_performance_part1.ipynb* and leveraging additional intuitions intuitions matured on route, I improve the first implementation of the modes considering customer types separately.

The main change with respect to the first implementation discussed above is that the Peason's correlation coefficients showed that an additional feature ('IMP_GAR') should be considered for customers of type *Persona Fisica* since its predictive power is not highlighted when considering the sample as a whole. For *Persona Giuridica* and *Cointestazione* the set of features remain unchanged.

Training separated models for different customer types improve the performance by 3-5% (F1) for the classification task and by 5-7% for the regression task.

Customer type	**F1**	**Precision**	**Recall**	**AUC**	**R2**	**MSE**
All customers	0.85 (0.91)	0.89 (0.93)	0.81 (0.89)	0.98	0.66 (0.78)	0.005 (0.003)
Persona Fisica	0.90 (0.91)	0.91 (0.93)	0.88 (0.94)	0.99	0.73 (0.83)	0.004 (0.002)
Persona Giuridica	0.87 (0.93)	0.91 (0.94)	0.84 (0.92)	0.99	0.72 (0.83)	0.004 (0.002)
Cointestazione	0.89 (0.93)	0.91 (0.94)	0.86 (0.92)	0.98	0.70 (0.82)	0.005 (0.003)

Conclusions: Predict and rank NPLs

The predictive models developed in the *modeling_and_performance_part2.ipynb* are applied over the most recent NPLs (09-2017) in order to predict the **probability of recovery** and the **expected recovery rate** in the following 12 months.

The list of NPLs is then ranked by those two predicted values and made available on the *npl_sorted.csv* file. In order to facilitate the work of portfolio managers, the probability of recovery used to define a **recovery class**, i.e. group of NPLs within a given range of probability.

The final result of this data challenge can be summarised as follows:

Recovery Class A contains the NPLs which are more likely to recover (Prob > 75%). A total of 1427 NPLs (7.6%) are in this class. The total debt corresponding to this class is 7.5M€, the predicted recovery rate is 10% which correspond to recover about 0.8M€.

Recovery Class B contains the NPLs which are likely to recover (Prob within 50% and 75%). A total of 135'727 NPLs (7.2%) are in this class. The total debt corresponding to this class is 8.6M€, the predicted recovery rate is 10% which correspond to recover about 0.8M€.

Recovery Class C contains the NPLs which are not likely tro recover (Prob within 25% and 50%). A total of 3'847 NPLs (20%) are in this class. The total debt corresponding to this class is 25.8M€, the predicted recovery rate is about 7.5% which correspond to recover about 1.9M€.

Finally, **Recovery Class D** contains the NPLs which are very unlikely to recover (Prob < 25%). A total of 12'153 NPLs (64%) are in this class. The total debt corresponding to this class is about 72M€, the predicted recovery rate is 3.5% which correspond to recover about 2.5M€.

Given the NPL sample as of 09-2017, the expected recovery rate that is between 1.6M€ (considering only A and B classes) and 6M€ (considering also C and D classes).

List of notebooks:

1. **exploration_1.ipynb**: general exploration of the main dataset of this challenge, i.e. PERIMETRO_INIZIALE.csv. Data cleaning, feature engineering, analytics to develop intuitions to approach the data challenge.
2. **exploration_2.ipynb**: exploration of the dataset CC.csv, answer the questions on point 2. Data cleaning, outliers, seasonalities.
3. **data_cleaning.ipynb:** consider all dataset available, merge them together in order to create a complete and informative dataset to target the challenge objective. Data cleaning, feature engineering, definition of the target variables.
4. **modeling _and_performance_part1.ipynb**: develop predictive models to reach the target objective. Feature selection, model selection, model performance.
5. **modeling_and_performance_part2.ipynb**: improve predictive models leveraging additional intuitions matured on route.
6. **predict_and_rank.ipynb**: apply the models developed on previous notebooks to predict the probability of recovery and the expected recovery rate of NPLs. Re-ranking NPLs.

Required tasks:

Answers are highlighted in red.

1. Data Exploration

- How are they connected? Which are the columns you should use to join them?
    - The datasets are indexed by a common set of keys that can be used to join them together. The following table summarize the keys used for join the dataset considered:

Dataset	Key 1	Key 2
Perimetro iniziale	ID_CUSTOMERS	NUM_YYYYMM
Anagrafica clienti	ID_CUSTOMERS	
Garanzie	ID_CUSTOMERS	
Mutui	ID_CUSTOMERS	NUM_AA_MM
CC	ID_CUSTOMERS	NUM_AA_MM
Centrale Rischi	ID_CUSTOMERS	NUM_AA_MM

- Focus on a single daset: CC.csv
    - How many months of data do you have?
        - There are 33 months from 01-2015 to 09-2015
    - Are there seasonalities over the time? If yes, in which (aggregated) variables? Make some plots.
    - Are there correlations between columns?

This question is addressed in the notebook *exploration_2.ipynb*

2. Data Cleaning

Select at least two datasets

- Detect weird values, outliers and missing values.
    - Which techniques or algorithms do you use to detect them?
    - How do you treat them? Is it ok to remove them from the dataset?
    - Is there any evident inconsistency in the data? If yes, how would you clean the data to overcome it?

These questions are addressed in the notebook data_cleaning.ipynb

- Why did you select these datasets?

    - I considered all the datasets summarised in the table above. The reason to consider all these datasets is because the feature they contain are potentially relevant for the goal of this challenge. All the features are then tested to evaluate if they provide any predictive power to reach the challenge's goals.

3. Target Variable

As in the real world, you are asked to create the target variable for this challenge by keeping in mind the challenge objective stated above.

- Create a data matrix to feed models (you can decide to use only a subset of the given datasets).
- Which granularity did you select and why?
- Which columns identify each occurence?
- Propose a target variable construction, write it with a mathematics formula and implement it.
- Explain and validate every choice/assumption you made in the previous point, also with business intuitions.
- If you decide to not have a target variable to predict, explain carefully your decision and how this could influence next steps in this analysis.

These questions are addressed in the notebook data_cleaning.ipynb

4. Features Engineering

Given or not the target variable defined above, you can choose the features to use in a predictive model.

- Create new features (we suggest ~3) from existing columns. Explain the intuition behind them.
- Select a subset of features to use to train a predictive model. The selection process should be rigorous and reproducible.

These questions are addressed in the notebook data_cleaning.ipynb

5. Modeling and Performance

This is the fancy part, right?

- How would you split the dataset to train, test and validation set?
- Which models have you tried to fit? Explain them.
- Which metrics have you used to measure the performance? Why?
- What have you done to improve models' performance? Are they useful?

These questions are addressed in the notebooks modeling_and_perfomance_part1.ipynb and modeling_and_perfomance_part2.ipynb