

Pontifícia Universidade Católica do Paraná (PUCPR)
Data Science
Computer Science
Professor Jean Paul Barddal

Project Description

The financial credibility of a person is a factor used to determine whether a loan should be approved or not, and this is quantified by a “credit score,” which is calculated using a variety of factors, including past performance on debt obligations, profiling, amongst others. Machine learning has been widely applied to automate the development of effective credit scoring models over the years. In this project you shall assemble in teams to work on a credit scoring problem. You have received two datasets in Blackboard: **train.csv** and **test.csv**. Both files contain a multitude of variables that can be used (or not) to predict whether a customer will paid its debt in full (column TARGET=0), or default (column TARGET=1). You can consult the meaning of each variable in the data dictionary provided and make sure they are all allowed in credit scoring models according to “Lei Nro 12.414, de 9 de junho de 2011”, also made available in Blackboard.

The report

Throughout this project, you and your team will have to fulfill each step of the python notebook template provided, including mainly:

- Univariate analysis
- Multivariate analysis
- Effective data visualization
- Data preprocessing
- Machine learning and tuning

Detailed information about each of the aforementioned topics can be found in the project template. A partial delivery of the report is expected. This checkpoint will be used to guide your team’s next steps based on peer reviewing.

Probabilities

In addition to the report, your team must deliver the defaulting probability (it is, $P[\text{TARGET}=1]$), of each customer (row) in the test dataset in a CSV file. **Important:** since the output values are probabilities, make sure they are bounded within $[0;1]$. The name of this file should be templated as follows: “NAME_OF_YOUR_TEAM.csv”. **Your team name should have up to 15 characters.** You must also make sure that the test file contains a header with the name of your team. **Submissions that do not adhere to the aforementioned constraints will be automatically disqualified.** An example of a valid header of an output file is as follows.

TEAM_JPB.csv	
1	TEAM_JPB
2	0.1
3	0.05
4	0.7
5	0.55123
6	...

Ranking

Teams will be ranked according to their Kolmogorov-Smirnov (KS) rates obtained in the test set. Teams will NOT receive the ground-truth TARGET values in test set, and the ranking will be made by the professor after the delivery date reported in Blackboard. Competitors can use the following function in Python combined with cross-validation to guide their decision towards machine learning and data preprocessing approaches:

```
from scipy.stats import stats
def computeKS(y_true, y_prob_positive):
    """
    Description:
        Kolmogorov-Smirnov value obtained from ground-truth
        targets (y_true) and
        their probabilities (y_prob_positive).
    Params:
        y_true (pd.Series): Ground-truth labels
        y_prob_positive (pd.Series): The probabilities of
        TARGET=1
    Output:
        ks (float): The KS rate
    """
    vals = list(zip(y_true, y_prob_positive))
    positives = []
    negatives = []
    for a, b in vals:
        if a == 0:
            negatives.append(b)
        else:
            positives.append(b)
    ks = 100.0 * stats.ks_2samp(positives, negatives)[0]
    return ks
```

Important notes:

- **The delivery dates are provided in Blackboard and on the module syllabus. Please check them prior to asking the dates of both the checkpoint and final deliveries.**
- All teams should have between 4 and 6 students. **No exceptions are allowed.**
- **The team that achieves the best KS rate in the test data will get an automatic 10 in the module grade.** If a tie is observed, teams will be ranked in terms of the probability distribution. Team with a probability distribution that best adheres to a Gaussian distribution with a mean of **0.5** will be the winner.
- The test file (test.csv) contains **131,617** rows. Therefore, your submission must have the same number of rows (**header excluded**). Also, you must make sure that the order of the instances in the test file and your predictions match.
- Submissions will be **DISQUALIFIED** if they ignore any of the concepts depicted in “Lei Nro 12.414, de 9 de junho de 2011”.
- Submissions will be **DISQUALIFIED** if the probabilities delivered are not reproducible. You must make sure that your code runs perfectly and that all random states/seeds are set so that the probabilities always match across notebook runs.
- The professor will **NOT** provide any insights or guidance on how to improve your KS rates. You are requested to test all the techniques discussed throughout the lectures, as well as to test other techniques you find in scientific papers, books, and internet.