
(Un)Bias on Credit Risk Prediction
May 30th, 2020

Celio Rodrigues de Oliveira is a graduate student at the University of Toronto at the Master of Information Human-Centred Data Science with a collaborative Specialization in Knowledge Media Design, and the creator of:

- A Multivariate Bias Calibration (Percentile Error Mapping technique) at the National Research Council Canada (2020),

- A Pattern Recognition solution (B.O.A.) for software testing automation and Finalist of the North America Software Testing and QE Awards under the category of the Best use of technology in a Project at Cloudpipe Inc (2019) https://softwaretesting.news/products/testawards/finalists/

Objective

The purpose of this study is:

a) Stress the importance of Data Scientists when dealing with sensitive information such as the Credit Default Risk Prediction;

b) Provide a reasonable explanation based on socioeconomic variables, historical relationship, and financial health analysis;

c) Compare human analysis versus IBM automated processes through AI Fairness 360 platform.

Technique

We are consenting to transform the whole data set into continuous variables.

This will allow us to treat the problem as a supervised classification.

Thus, it is not considered outlier detection, clustering analysis, dimensionality reduction techniques such as Principal Component Analysis, Pearson correlation, etc.

Once the data is preprocessed, we expose it to multiple machine learning algorithms from Random Forest to Neural Networks aiming to use the better classification score.

Data Engineering

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Financial       1000 non-null   object
 1   Duration        1000 non-null   int64
 2   History         1000 non-null   object
 3   Purpose         1000 non-null   object
 4   Amount          1000 non-null   int64
 5   Bond            1000 non-null   object
 6   Employment      1000 non-null   object
 7   Installments    1000 non-null   int64
 8   Gender          1000 non-null   object
 9   Guarantors      1000 non-null   object
 10  Residence       1000 non-null   int64
 11  Property        1000 non-null   object
 12  Age             1000 non-null   int64
 13  Installments2   1000 non-null   object
 14  Housing         1000 non-null   object
 15  Products        1000 non-null   int64
 16  Job             1000 non-null   object
 17  Liabler         1000 non-null   int64
 18  Telephone       1000 non-null   object
 19  Foreign         1000 non-null   object
 20  Default         1000 non-null   int64
dtypes: int64(8), object(13)
memory usage: 164.2+ KB
```

In a quick analysis, we can see the new dataset from categorical to continuous variables.

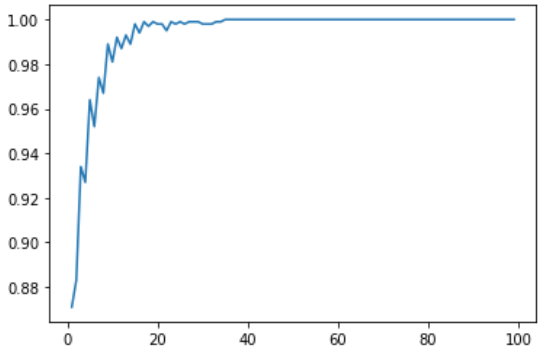
Two new columns from the previous Gender and Job classification

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 23 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Financial       1000 non-null   int64
 1   Duration        1000 non-null   int64
 2   History         1000 non-null   int64
 3   Purpose         1000 non-null   int64
 4   Amount          1000 non-null   int64
 5   Bond            1000 non-null   int64
 6   Employment      1000 non-null   int64
 7   Installments    1000 non-null   int64
 8   Gender          1000 non-null   int64
 9   Guarantors      1000 non-null   int64
 10  Residence       1000 non-null   int64
 11  Property        1000 non-null   int64
 12  Age             1000 non-null   int64
 13  Installments2   1000 non-null   int64
 14  Housing         1000 non-null   int64
 15  Products        1000 non-null   int64
 16  Job             1000 non-null   int64
 17  Liabler         1000 non-null   int64
 18  Telephone       1000 non-null   int64
 19  Foreign         1000 non-null   int64
 20  Default         1000 non-null   int64
 21  Civil           1000 non-null   int64
 22  Resident        1000 non-null   int64
dtypes: int64(23)
memory usage: 179.8 KB
```

Purity and Entropy levels

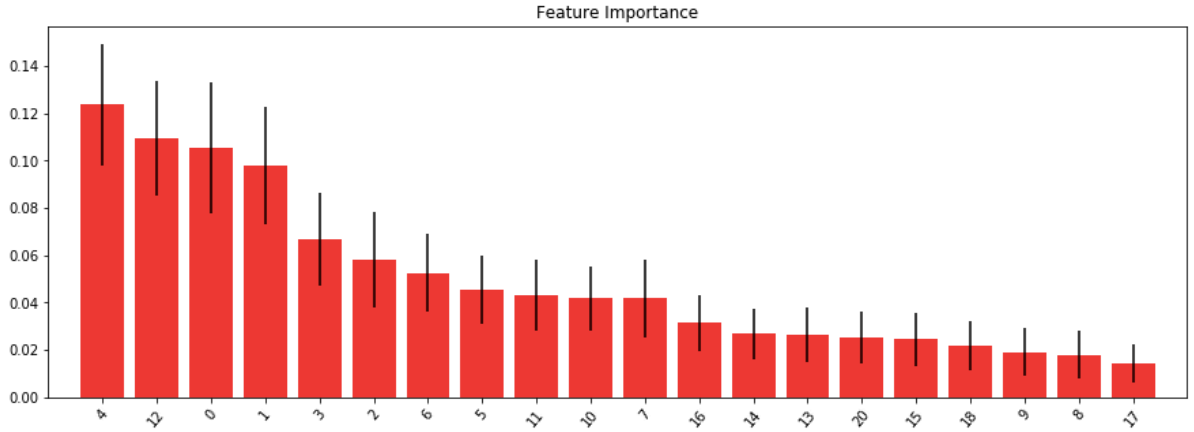
It is measured the entropy and the information gain criteria to define the optimal number of splits in a decision tree grouping observations with similar characteristics.

The purity level was reached after split of 38 terminal nodes.



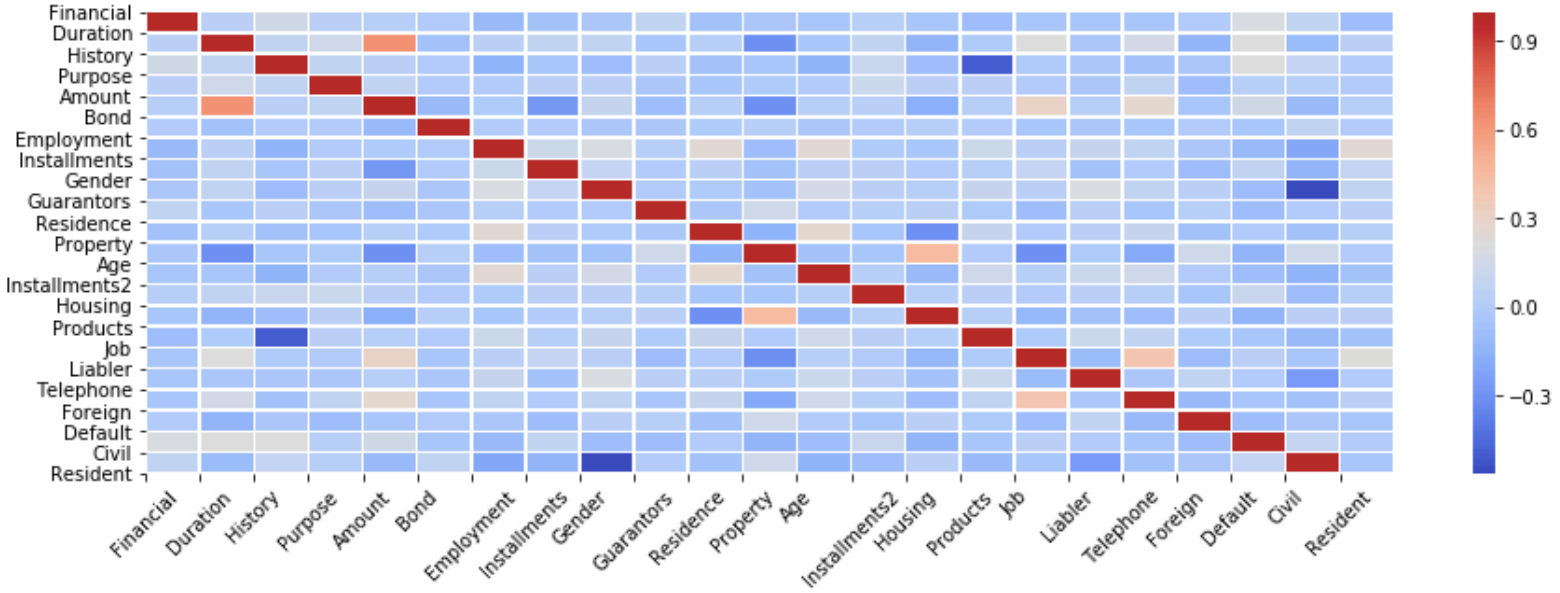
Feature Importance

For this dataset, features were classified according to information gain criteria in the following order:



4 Amount
12 Age
0 Financial
1 Duration
3 Purpose
2 History
6 Employment
5 Bond
11 Property
10 Residence
7 Installments
16 Job
14 Housing
13 Installments2
20 Default
15 Products
18 Telephone
9 Guarantors
8 Gender
17 Liabler
19 Foreign

Feature Correlation - Heatmap



Feature Correlation

This analysis brings some expected correlated criteria and intriguing others:

1. Positive correlation between Credit Amount and the Duration of an existing checking account,

2. Medium Positive correlation between owning a Property, years in the Residence, and status in the Housing as clients who own their houses have a property and do live there for a few years, as well as Job and Foreign, main reason for having a foreigner in the country (other than study),

3. Weak Positive correlation between having a Job and a registered Telephone number,

4. Negative correlation between Civil status, Gender and Permanent Residency status; and surprisingly between Credit History and the number of products with the financial institution,

5. Weak Negative correlation between the Duration of the relationship with the bank, the Amount of money being administrated by the institution, and owning a Property.

Bagging Methods

The best grouped model is the Random Forest using the unbiased* dataset that performed 79.6% average and a F-1 score of 86.7%. This represents a better performance in 3% compared to the biased dataset.

Remembering that F-1 score is based on correctly classified True Positive values. Therefore, credit analysis algorithms have the intention to prevent credit concession to high risk clients, this is the metric that matters most for this study.

```
Accuracy of  RF :  0.796
F-1 score of  RF :  0.8668407310704961
Mean of  RF vanilla:  0.7492332859174964          std: (+/-):  0.04258862696853836
Mean of  RF bagging:  0.7160398293029872          std: (+/-):  0.02084808909483915
```

*Unbiased because it is not considered variables like Gender and Race.

Boosting and Neural Networks

The best boosting model is the Gradient Boosting also applied on the unbiased dataset, and has a 75.5% performance with a standard deviation of 0.046.

```
Mean: 0.742, std: (+/-) 0.045 [Ada Boost]
Mean: 0.755, std: (+/-) 0.046 [Grad Boost]
Mean: 0.748, std: (+/-) 0.031 [Ensemble]
```

NN performed poorly with an overall accuracy of 70%, using a combination of TanH, Softplus and ReLu as activation functions on each neural layer, optimized with Adam.

Benchmark

Both fairness metrics and mitigation algorithms can be performed at various stages of the machine learning pipeline.

It is recommended to check for bias as often as possible, using as many metrics are relevant for the application domain.

We also recommend incorporating bias detection in an automated continuous integration pipeline to ensure bias awareness if this software project evolves.

Next slide shows the IBM Fairness 360 results.

Benchmark – IBM Auto Fairness 360

The automated solution from IBM performed some data transformation and weigh imputation.

However, this model performance is 71.3% outperformed by our Bagging model using Random Forest that performed 79.6%. Furthermore, the IBM model does not compute F1-score (True Positive).

Predictions from transformed testing data

```
 13%|          | 13/100 [00:00<00:00, 127.35it/s]
Classification threshold used = 0.2674
 41%|          | 41/100 [00:00<00:00, 98.54it/s]
Balanced accuracy = 0.7128
Statistical parity difference = -0.0906
Disparate impact = 0.7625
Average odds difference = -0.0266
Equal opportunity difference = -0.0518
Theil index = 0.1294
100%|          | 100/100 [00:01<00:00, 97.34it/s]
```

Benchmark – Classification (AIF360)

There are other classification algorithms using the German dataset in the literature. I relate results from two of them below:

Model 1.

'classification rate in training set via xgboost is 0.84'

'classification rate in testing set via xgboost is 0.76'

Model 2.

Balanced feature selected data (%)			
Acc.	Precision	Recall	F1-score
82	84	82	81
81	81	81	81

It is possible to reproduce the model 2 outcome, but it will require more working hours to fine tune the dataset, extract the best representation of it and applying hyperparameter tuning techniques.

To be continued ...

Celio Oliveira, Data Scientist

Oliveira.celior@gmail.com

+1 437.777.9306