Inteligência Artificial 2021/2022 - FEUP

Credit Card Fraud

Duarte Sardão - 201905497

Edgar Lourenço - 201604910

Pedro Pereira - 201905508

Work to be performed

- How to classify examples in terms of the concept under analysis.
- Dataset analysis to check for the need for data pre-processing
- An initial exploratory data analysis should be carried out (class distribution, values per attribute, and so on).
- Evaluation of the learning process (in particular on the test set).
- In terms of credit card fraud, the objective is to split the data between fraudulent and non-fraudulent transactions, which at the moment are presented as general generic data.

Related Work and References

- https://moodle.up.pt/pluginfile.php/196462/mod_resource/content/0/IART_Lecture5a_Intro_MachineLearning.pdf
- https://moodle.up.pt/pluginfile.php/196463/mod_resource/content/0/IART_Lecture5b_MachineLearning_Tools.pdf
- https://moodle.up.pt/pluginfile.php/196464/mod_resource/content/0/IART_Lecture5c_MachineLearning_DataPreprocessing.pdf
- https://moodle.up.pt/pluginfile.php/196465/mod_resource/content/0/IART_Lecture5d_MachineLearning_Classification.pdf
- https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud

Algorithms to be used

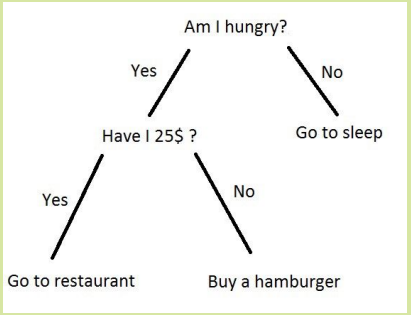
- Decision Trees
- Naive Bayes
- Logistic Regression

Tools to be used

- Jupyter Notebook
- Anaconda

Decision Trees

- We have already partially implemented this algorithm which is already showing promising results in predicting credit card fraud.
- It's a machine learning classification algorithm creates a model (classification tree) that predicts the value of a target variable based on other several input variables



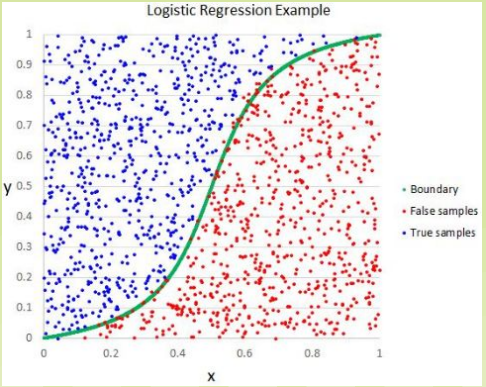
Naive Bayes

- Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Logistic Regression

- This classification algorithm models the probability of one event taking place by having the log-odds for the event be a linear combination of one or more independent variables.
- It's particularly useful because the algorithm only outputs a binary result.



Details of the data pre-processing

```
The shape of the data set is: (1000000, 8)

-------------------------------------------

Checking for missing values:

distance_from_home                0
distance_from_last_transaction    0
ratio_to_median_purchase_price    0
repeat_retailer                   0
used_chip                         0
used_pin_number                   0
online_order                      0
fraud                             0
dtype: int64

-------------------------------------------

       distance_from_home  distance_from_last_transaction  \
count       1000000.000000                  1000000.000000
mean             26.628792                        5.036519
std              65.390784                       25.843093
min               0.004874                        0.000118
25%               3.878008                        0.296671
50%               9.967760                        0.998650
75%              25.743985                        3.355748
max           10632.723672                    11851.104565

       ratio_to_median_purchase_price  repeat_retailer       used_chip  \
count                   1000000.000000   1000000.000000  1000000.000000
mean                          1.824182         0.881536        0.350399
std                           2.799589         0.323157        0.477095
min                           0.004399         0.000000        0.000000
25%                           0.475673         1.000000        0.000000
50%                           0.997717         1.000000        0.000000
75%                           2.096370         1.000000        1.000000
max                         267.802942         1.000000        1.000000

       used_pin_number    online_order           fraud
count   1000000.000000  1000000.000000  1000000.000000
mean          0.100608        0.650552        0.087403
std           0.300809        0.476796        0.282425
min           0.000000        0.000000        0.000000
25%           0.000000        0.000000        0.000000
50%           0.000000        1.000000        0.000000
75%           0.000000        1.000000        0.000000
max           1.000000        1.000000        1.000000

-------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 8 columns):
 #   Column                          Non-Null Count    Dtype
---  ------                          --------------    -----
 0   distance_from_home              1000000 non-null  float64
 1   distance_from_last_transaction  1000000 non-null  float64
 2   ratio_to_median_purchase_price  1000000 non-null  float64
 3   repeat_retailer                 1000000 non-null  float64
 4   used_chip                       1000000 non-null  float64
 5   used_pin_number                 1000000 non-null  float64
 6   online_order                    1000000 non-null  float64
 7   fraud                           1000000 non-null  float64
dtypes: float64(8)
memory usage: 61.0 MB
None

-------------------------------------------

0.0    912597
1.0     87403
Name: fraud, dtype: int64

-------------------------------------------
```

Details of the data pre-processing

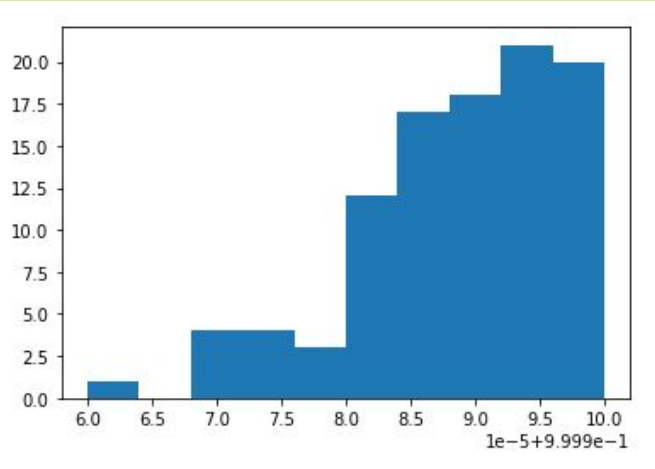
- The dataset contains 10,00,000 data points.
- There are 7 features and 1 target with two classes.
- The class is clearly imbalanced which needs to resolved before proceeding ahead.
- To resolve the issue of class imbalance, we will resample the original datasets and only select 20% data points from the original data set.

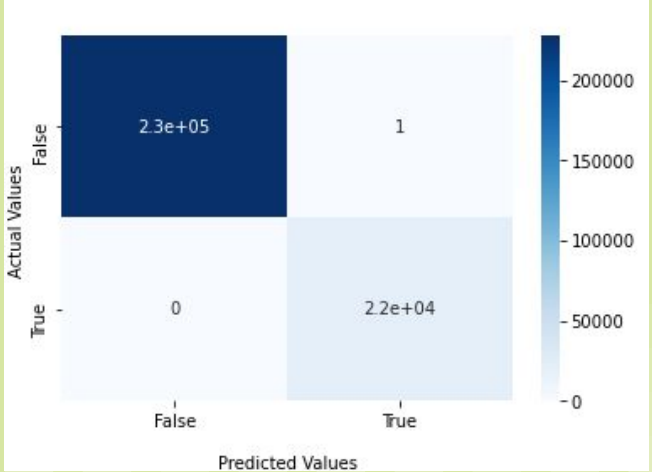
Developed Methods and Evaluation/Comparison

Decision Trees

With an accuracy of over 99,99% this algorithm has proven to be effective on this problem.

The precision for Tree is 0.9999540081865428
The recall for Tree is 1.0



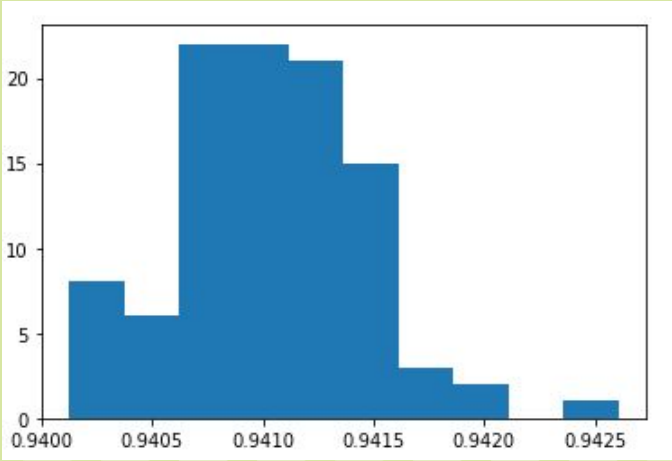


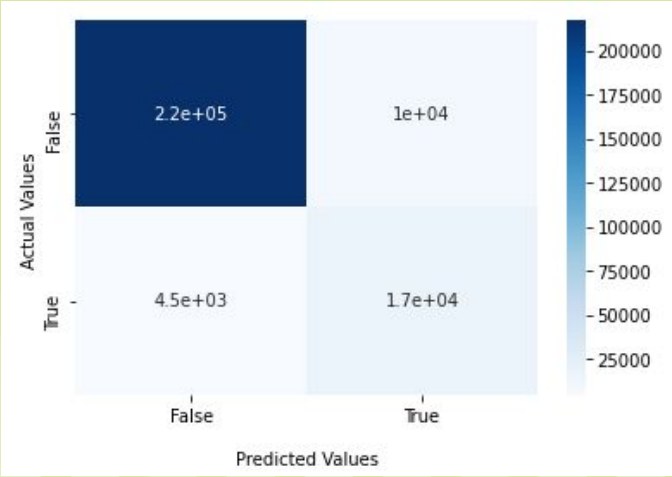
Developed Methods and Evaluation/Comparison

Naive Bayes

Although the accuracy being quite high (~94%), the recall and precision show that this algorithm is not useful.

The precision for NB is 0.6291359630111255
The recall for NB is 0.7950426804217829

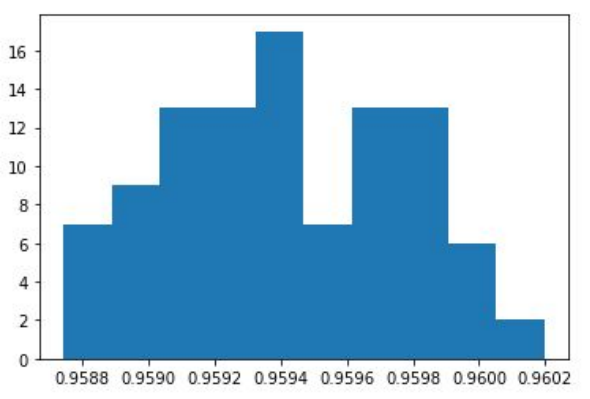




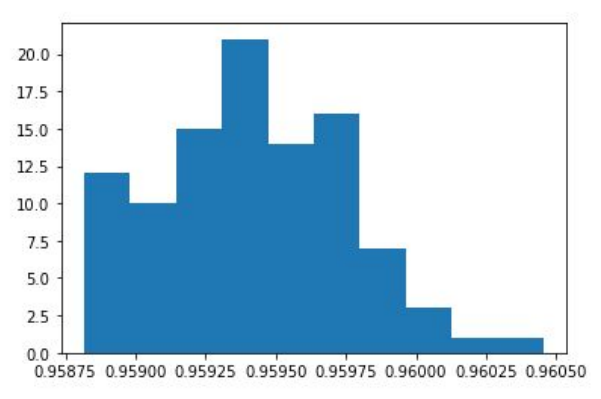
Developed Methods and Evaluation/Comparison

Logistic Regression

With an accuracy of 96% and precision of 85% we suspected this algorithm could prove to be useful with some tweaks to its parameters. After some trial and error tweaking we couldn't improve the algorithm results significantly.



The precision for LR is 0.8508655126498003
The recall for LR is 0.6417126945725111



The precision for LR is 0.8508501240394506
The recall for LR is 0.6418952846122244