Fraudulent Transactions Data

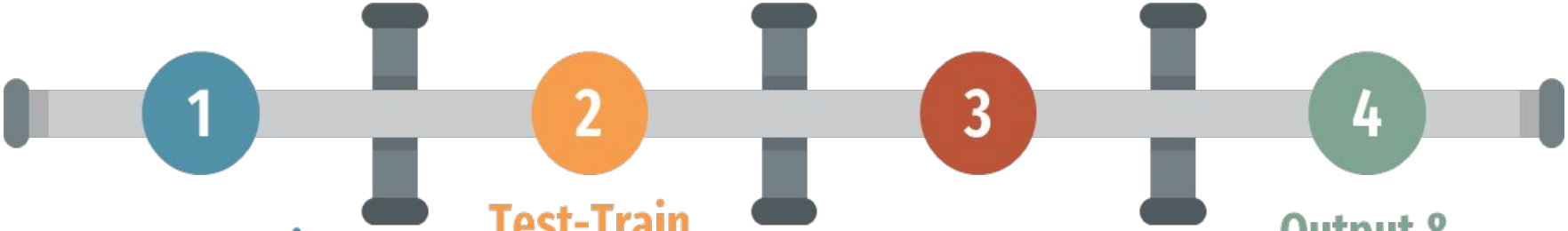
SAPIENZA
UNIVERSITÀ DI ROMA

Claudio Gheorghiu
Giuseppe Di Poce
Angelo Mandara
Enrico Grimaldi
Tito Tamburini

By the order of Peppi Blinders

Data pipeline

SAPIENZA
UNIVERSITÀ DI ROMA



Pre-Processing

- Analysis of data set (https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data)
- Feature Engineering
  - Parsing the feature types in numeric;
  - Dropping non-informative features;
  - Handling multi-collinearity.

Test-Train splitting

- Splitting → 70% - 30 %
- Cross validation for better results

Model selection

Picking the best model and using resamplingtechniques to manage unbalanceness.

Output & Metrics

Reporting our results.

Exploratory Data Analysis (EDA)

SAPIENZA UNIVERSITÀ DI ROMA

Main aspects

- Data set size ⟶ 186 MB
- Number of features ⟶ 10
- Number of examples ⟶ 6362620
- General info:

```
#   Column          Dtype
---  ------          -----
0   step            int64
1   type            object
2   amount          float64
3   nameOrig        object
4   oldbalanceOrg   float64
5   newbalanceOrig  float64
6   nameDest        object
7   oldbalanceDest  float64
8   newbalanceDest  float64
9   isFraud         int64
10  isFlaggedFraud  int64
```

- Target variable ⟶ ‘isFraud’

Fraudulent distribution over features

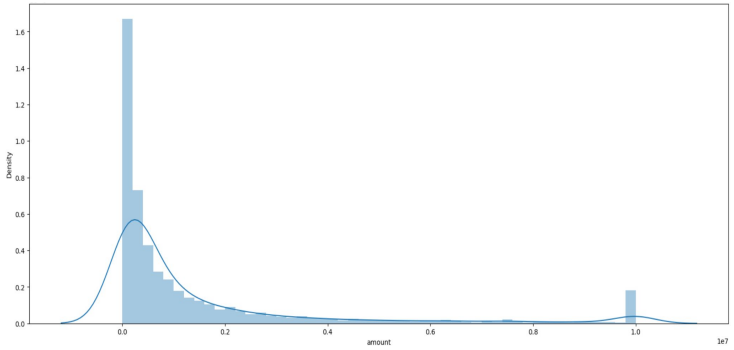
- type of non-fraudulent transactions

```
CASH_OUT    2233384
PAYMENT     2151495
CASH_IN     1399284
TRANSFER     528812
DEBIT         41432
```

- type of fraudulent

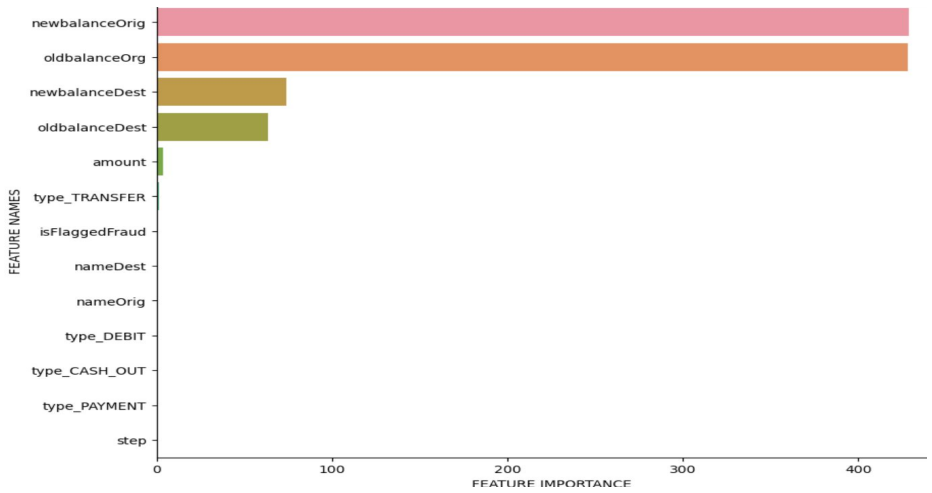
```
CASH_OUT    4116
TRANSFER    4097
```

- number of flagged transactions ⟶ 16
- amount of money for fraudulent transactions



Main issues

- High VIF (Variable Importance Feature) for features about accounts balance and feature hierarchy



	feature	VIF
5	newbalanceOrig	428.693436
4	oldbalanceOrg	428.309108
7	newbalanceDest	73.946539
6	oldbalanceDest	63.765295

critical features in terms of multi-collinearity

- Unbalanced data → fraudulent transaction 0.13%
- Non-informative features

```
0    6362604
1         16
Name: isFlaggedFraud, dtype: int64
```

Feature engineering

- Handling categorical variables:
  - one-hot encoding of transaction types
  - dropping ‘isFlaggedFraud’

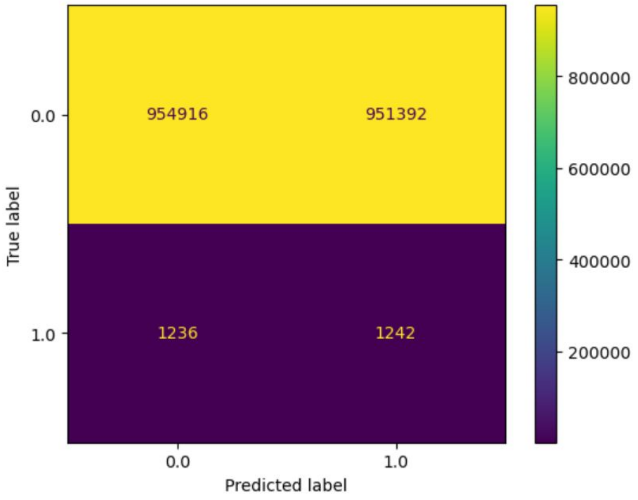
type_CASH_OUT	type_DEBIT	type_PAYMENT	type_TRANSFER
0	0	1	0
0	0	1	0
0	0	0	1

- Hashing strings (Customer ID) in numbers
- Handling the problematic variables in **terms of multi-collinearity**

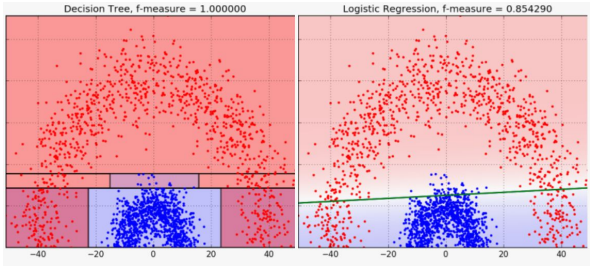
source_bal_change	destinationBal
0	0.0
0	0.0
0	0.0
0	21182.0
0	0.0

  - we identified four problematic variables extremely interrelated
  - we reduced them to two columns (one-hot encoding and reporting the difference between pairs values)
  - considered sources:
    - https://quantifyinghealth.com/vif-threshold/
    - https://www.tandfonline.com/doi/abs/10.1080/09720502.2010.10700699?journalCode=tjim20#:~:text=Multicollinearity%20is%20a%20statistical%20phenomenon%20in%20which%20two%20or%20more,relationships%20among%20the%20explanatory%20variables

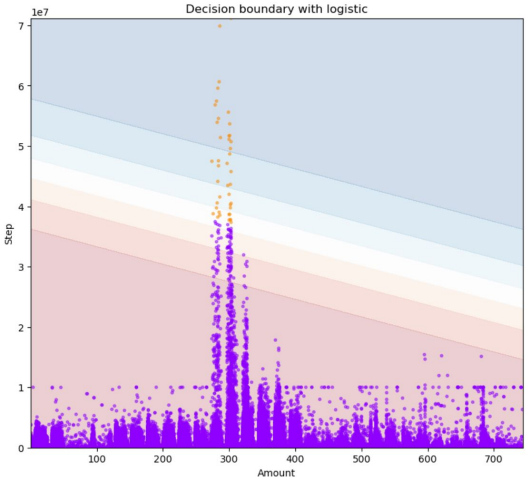
Logistic Regression

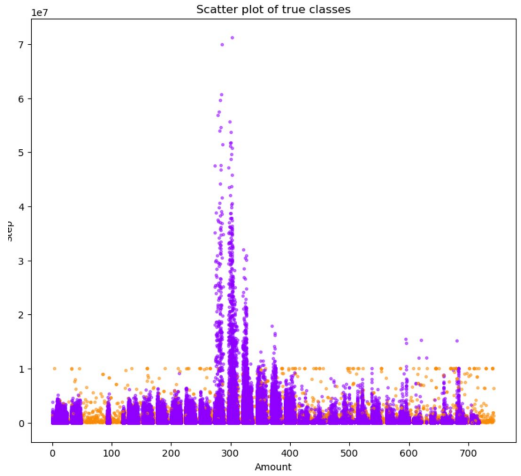


The logistic regression model has a really bad performance due to his ‘linear’ decision boundary



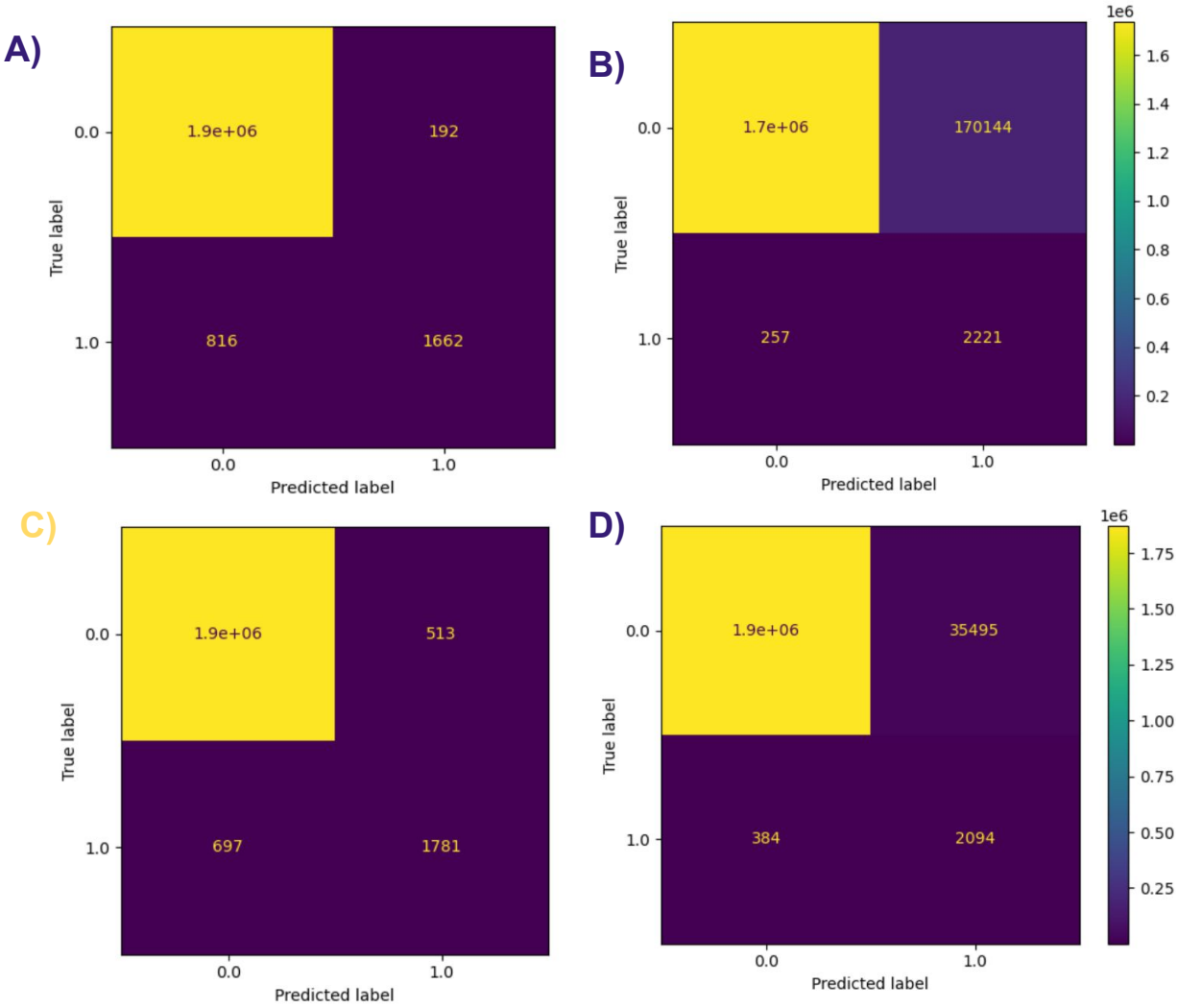
The example on the left is not related to our data set but well represents how decision boundaries are defined for **decision tree** models.
Source:
https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/#:~:text=Decision%20Trees%20bisect%20the%20space,generalize%20to%20planes%20and%20hyperplanes.





Decision tree & resampling techniques

SAPIENZA UNIVERSITÀ DI ROMA



A) DT without resampling
B) DT with **undersampling**
C) DT with **oversampling**
D) DT with **SMOTE**

Goal : our intention is to to minimize the quantity of **false negatives** and we would increase the **recall** but without losing too much efficiency in terms trade-off (precision-recall).

R = TP/(TP+FN)

Further improvements

- Random forest
  - Pros.
    - Powerful and highly accurate
    - Can handle several features at once
    - Run trees in parallel ways
    - Can perform both regression and classification tasks.
    - Produces good prediction that is easily understandable
  - Cons.
    - They are biased to certain features sometimes
    - The algorithm can become quite slow and ineffective for real-time predictions.
    - Worse for high dimensional data

→ In our case we DO NOT gain any improvement in metrics with random forest

- Cross validation

Conclusions

Suggestions for the Bank/Company:

- gather more information about time of a transaction → ‘step’ feature is important according to our analysis but is not really informative given the fact that we don’t know the ‘zero time’ and we can’t extract further information;
- more attention on the ‘transfer’ type cards → they include the most of fraudulent transactions;
- a fundamental point for the company is to concentrate part of their investments on predictive models that minimize **FALSE NEGATIVES**

  → False alarmism preferred over risk of non-detection.

Bibliography

- https://www.kaggle.com/code/kartik2khandelwal/predicting-fraudulent-transactions
- https://github.com/sosamandara/FDS_Final_Project
- Pattern recognition and machine learning, C. Bishop