Paolo Rinaldi

Machine Learning Project

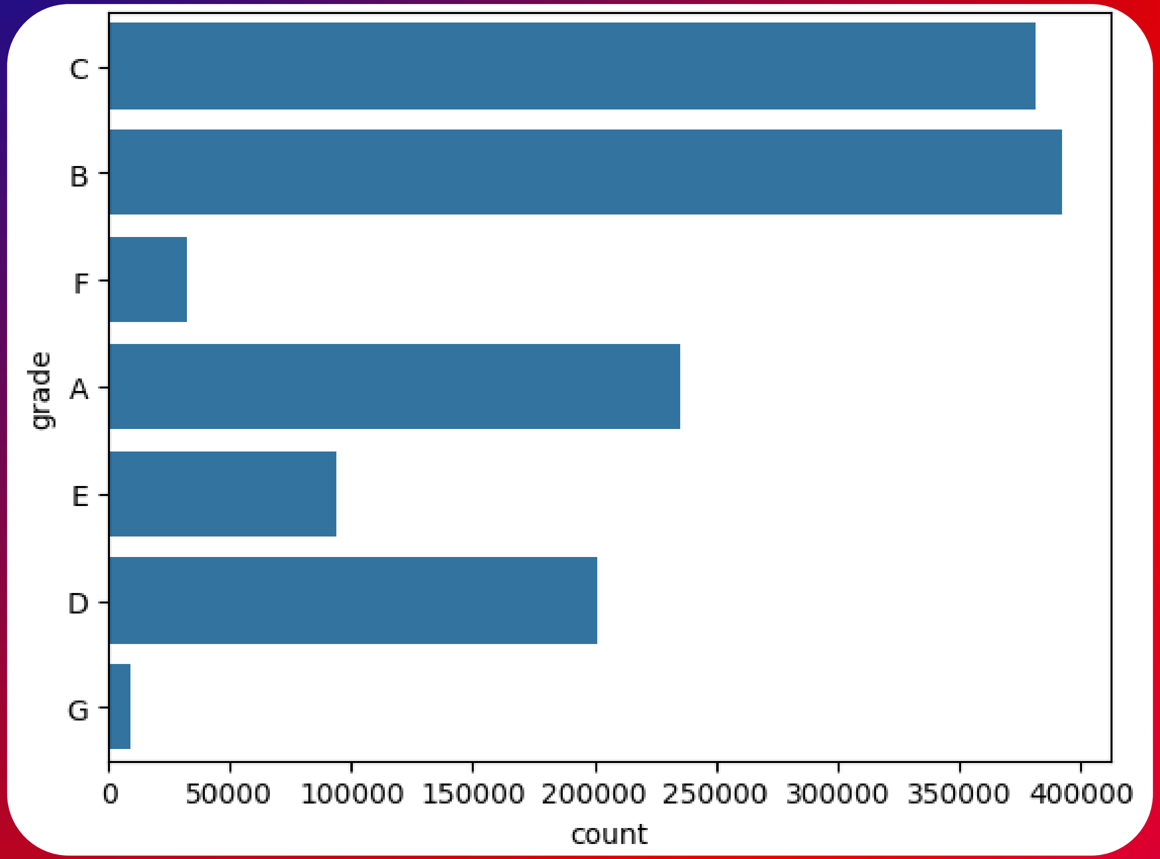
Lending club

Using Python

The project aims to determine whether a new client is worthy or too risky to get credit based on historical data from the Lending Club database.

15/05/2025

Master’s Degree in Finance

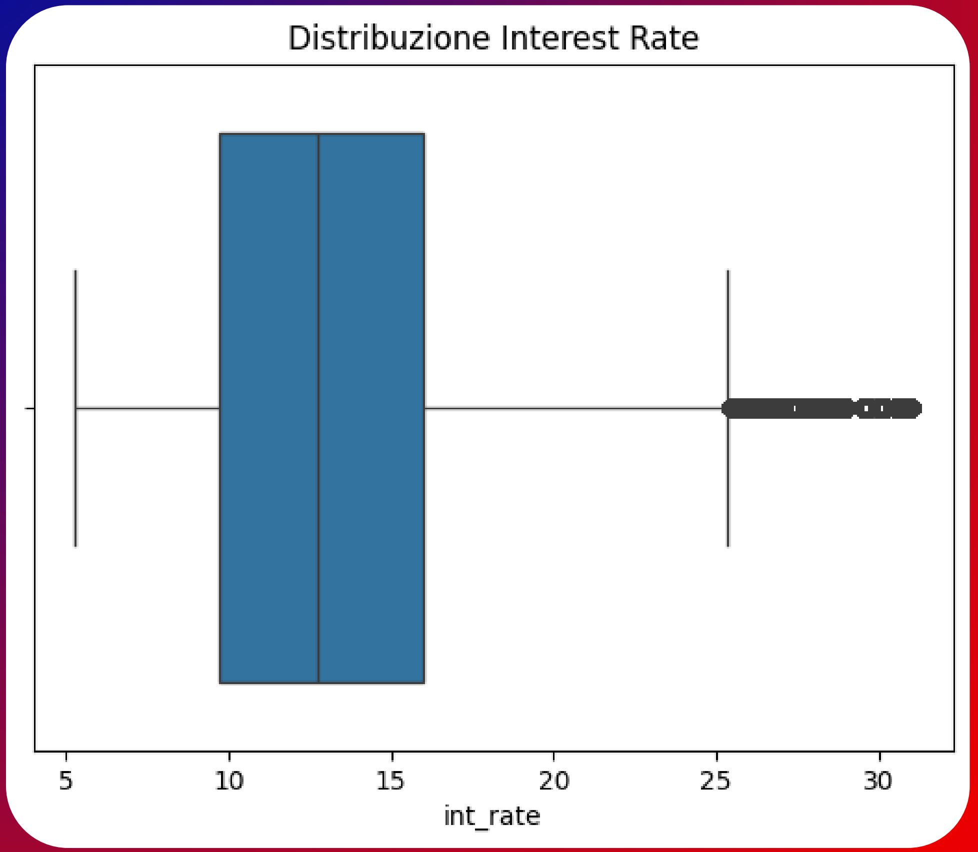


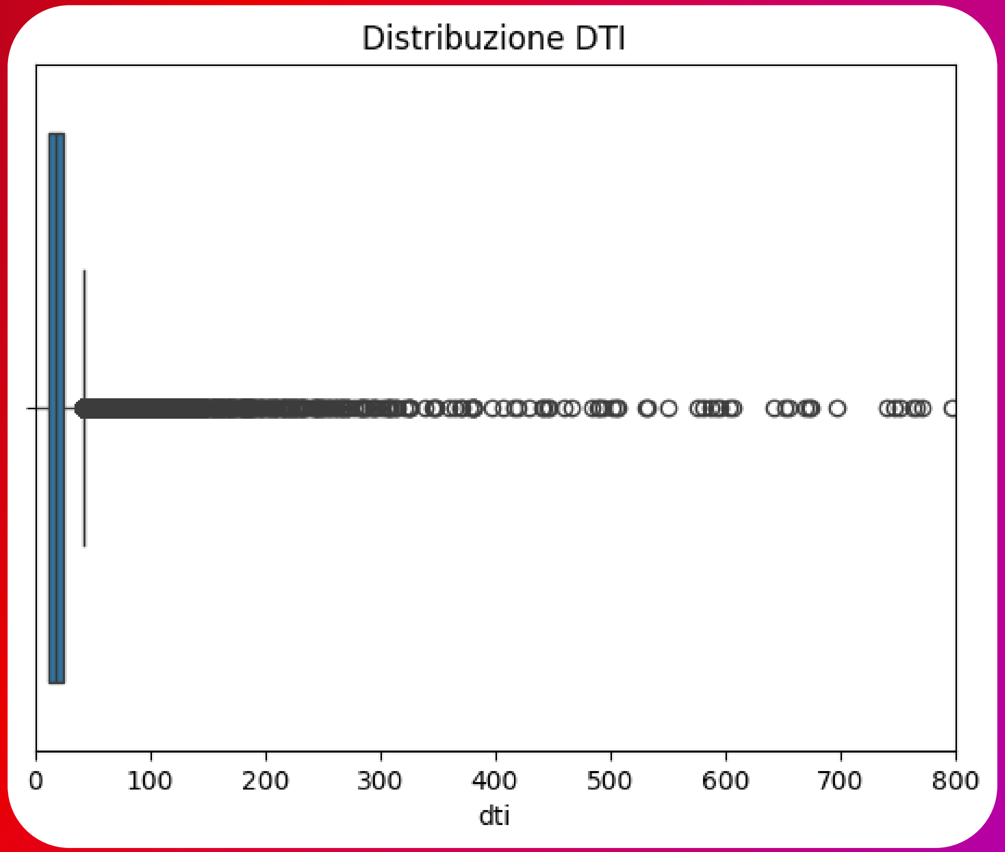

```
# 3. Column removal with >50% missing values
missing_frac = df_mapping.isnull().mean()
df_mapping = df_mapping.loc[:, missing_frac < 0.5]
```

```
print(df_dup.isnull().sum())
```

```
home_ownership                  0
grade                           0
verification_status             0
purpose                         0
term                            0
debt_settlement_flag            0
int_rate                        0
fico_range_low                  0
fico_range_high                 0
emp_length                  78487
loan_status                     0
annual_inc                      0
dti                           374
loan_amnt                       0
open_acc                        0
pub_rec                         0
delinq_2yrs                     0
inq_last_6mths                  1
tot_cur_bal                 67425
revol_bal                       0
revol_util                    854
collections_12_mths_ex_med     54
mths_since_last_delinq     678495
mort_acc                    47179
pub_rec_bankruptcies          694
tax_liens                      37
```

- **Removing duplicates**
- **Checking the composition of each meaningful variable**
  - better comprehension of the database
  - mantaining the variables with highest correlation with target
- **Managing the missing values**
  - substitute with the most suitable values
  - eventually drop when few (<1%)



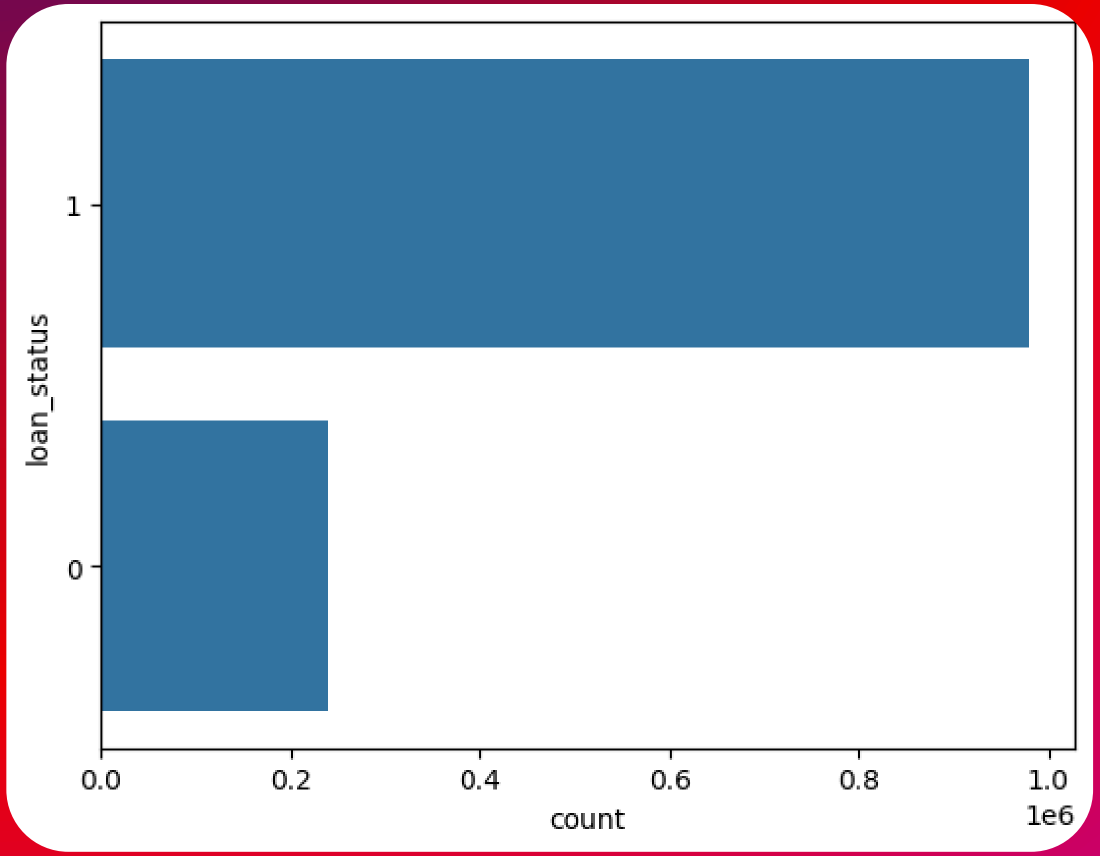


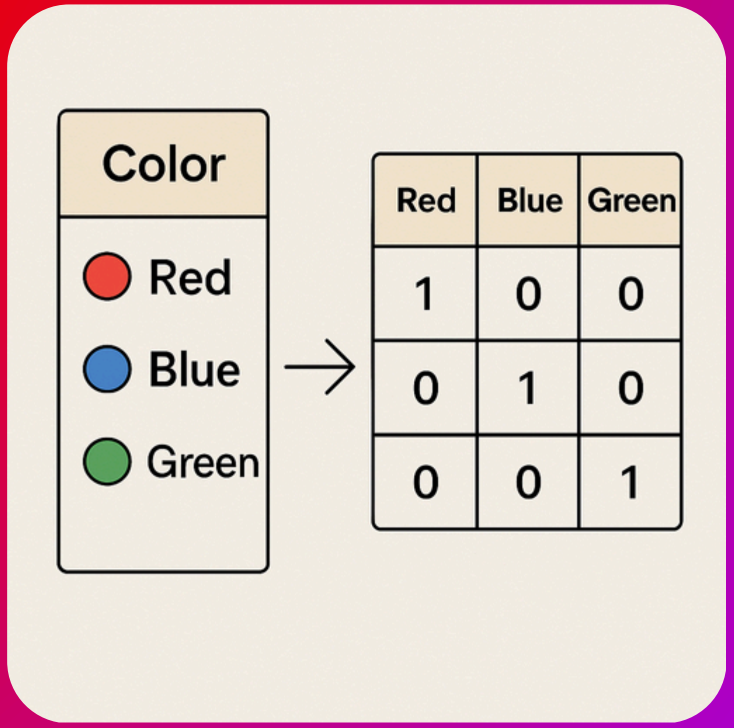

```
wh_lo_dti= stats_3['whislo']
wh_hi_dti= stats_3['whishi']
```

```
bounds = {}
for col in cols:
    stats = boxplot_stats(df_imputed[col], whis=1.5)[0]
    bounds[col] = (stats['whislo'], stats['whishi'])
    # opzionale: stampo per verifica
    print(f"{col:15s} → [{stats['whislo']:.2f}, {stats['whishi']:.2f}]")
```

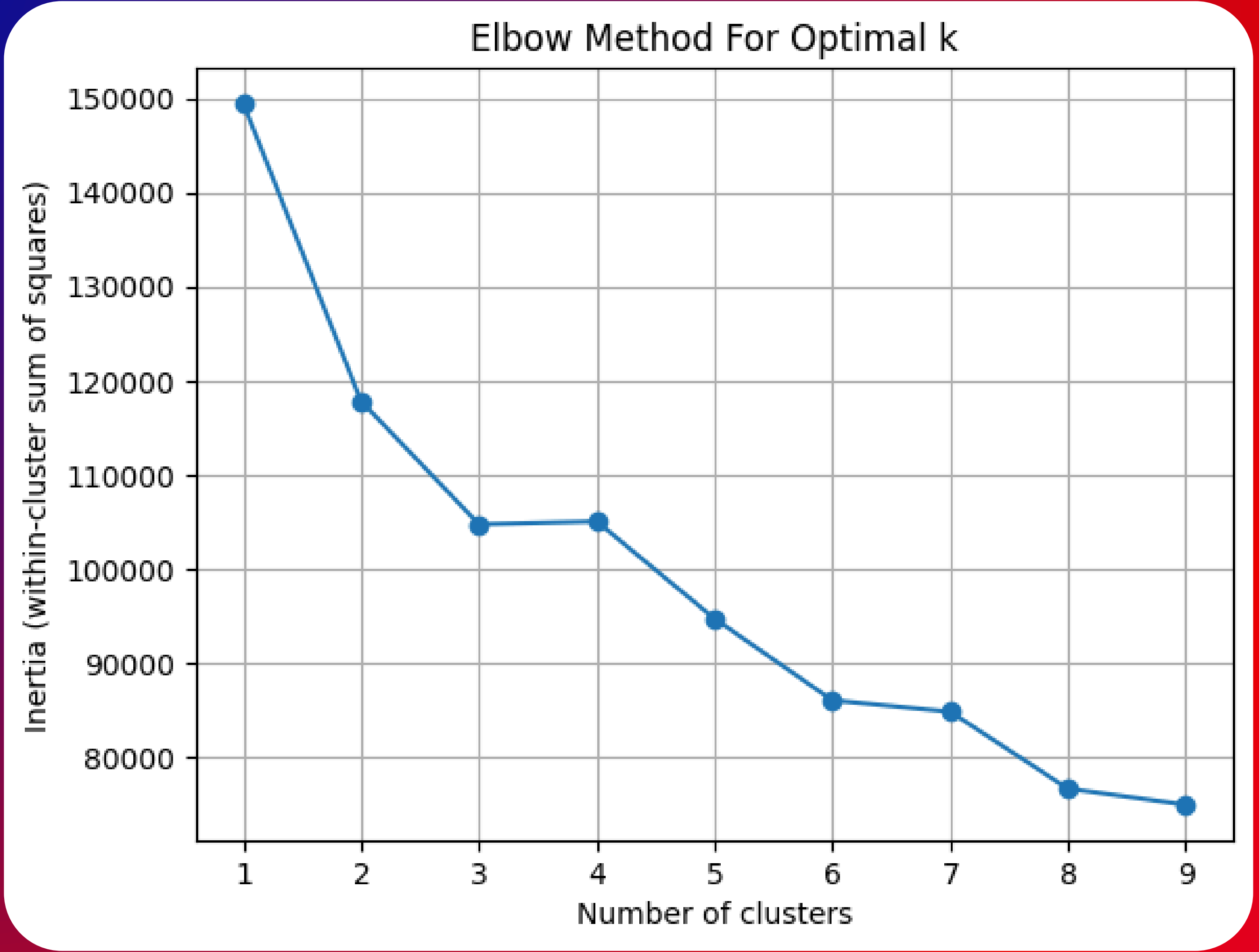
- **Creating boxplots for the most volatile variables**
  - Visualize the data
  - Understanding the distributions
- **Removing the outliers:**
  - DTI
  - Interest rate
  - Loan amount
  - Open accounts
  - FICO
- Percentage of dropped rows : **9.27%**

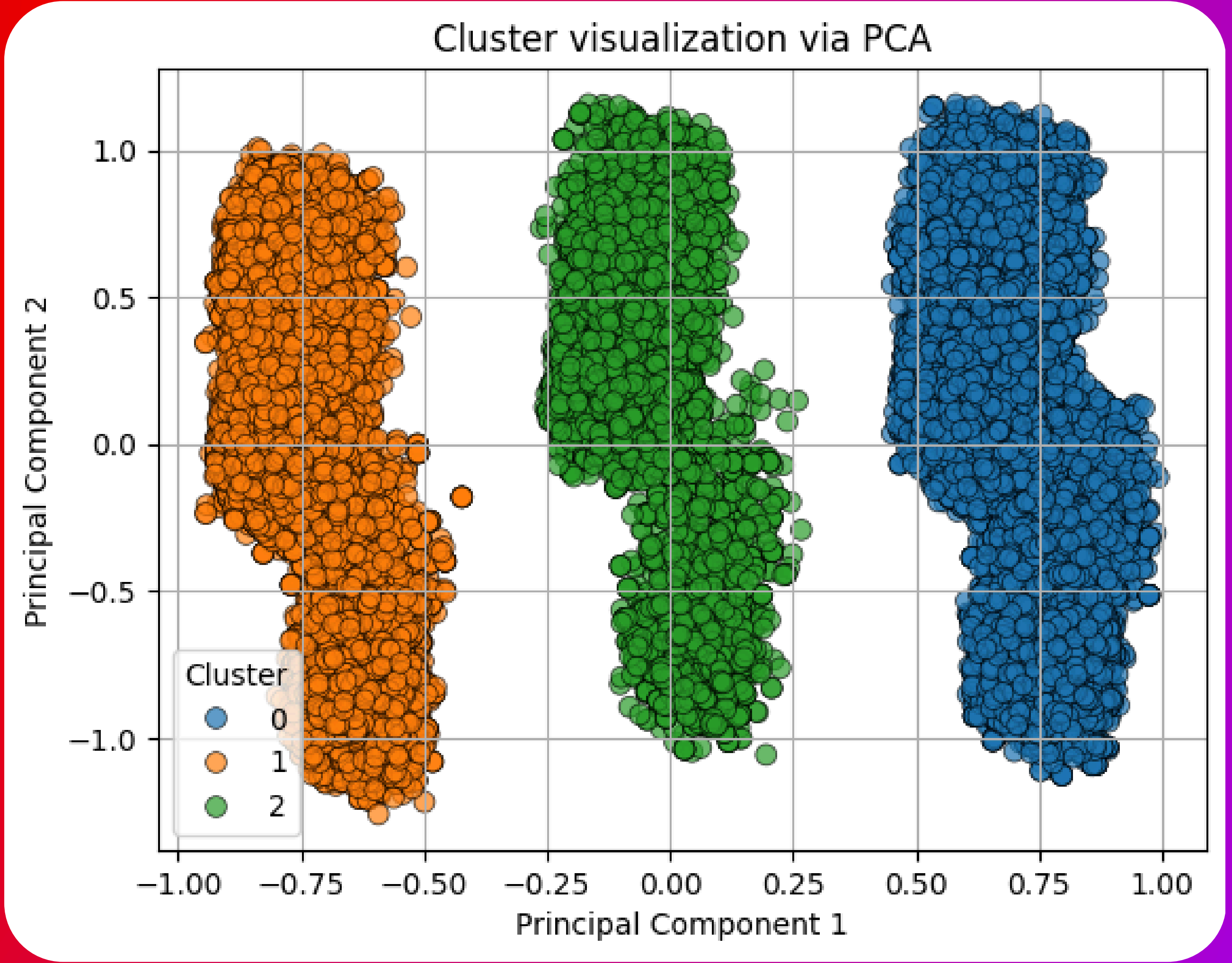
```python
# Create new variables
df_final_2['debt_burden'] = df_final_2['annual_inc'] / df_final_2['dti'] #avoid division by zero
df_final_2['fico_average'] = (df_final_2['fico_range_high'] + df_final_2['fico_range_low'])/ 2
df_final_2['credit_inquiry_density'] = df_final_2['inq_last_6mths'] / (df_final_2['open_acc']+1)

df_final_2['issue_d'] = pd.to_datetime(df['issue_d'], format='%b-%Y')
df_final_2['earliest_cr_line'] = pd.to_datetime(df['earliest_cr_line'], format='%b-%Y')
df_final_2['credit_age_days'] = (df_final_2['issue_d'] - df_final_2['earliest_cr_line']).dt.days
df_final_2['credit_age_years'] = df_final_2['credit_age_days'] / 365
print("\nColumns in DataFrame:", df_final_2.columns.tolist())
```





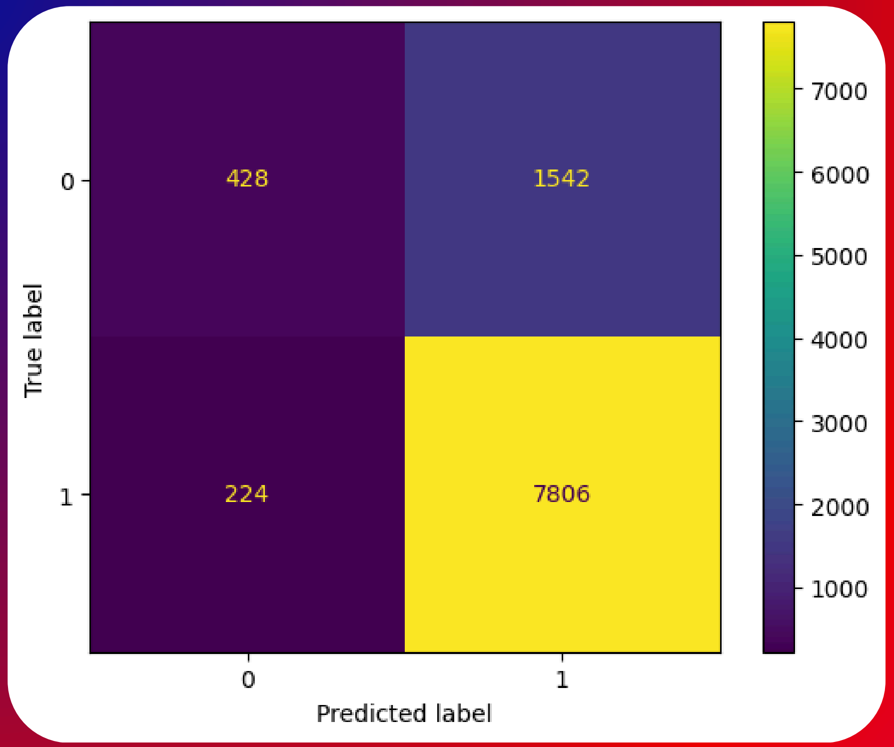
- **Feature engineering: creation of new variables**
  - credit age
  - debt burden
  - fico average
  - credit inquiry density
- **Encoding categorical variables**
  - Grouping
  - Mapping
  - One-hot-encoding
  - MinMaxScaler
- **Train Test and Split**
  - Oversampling

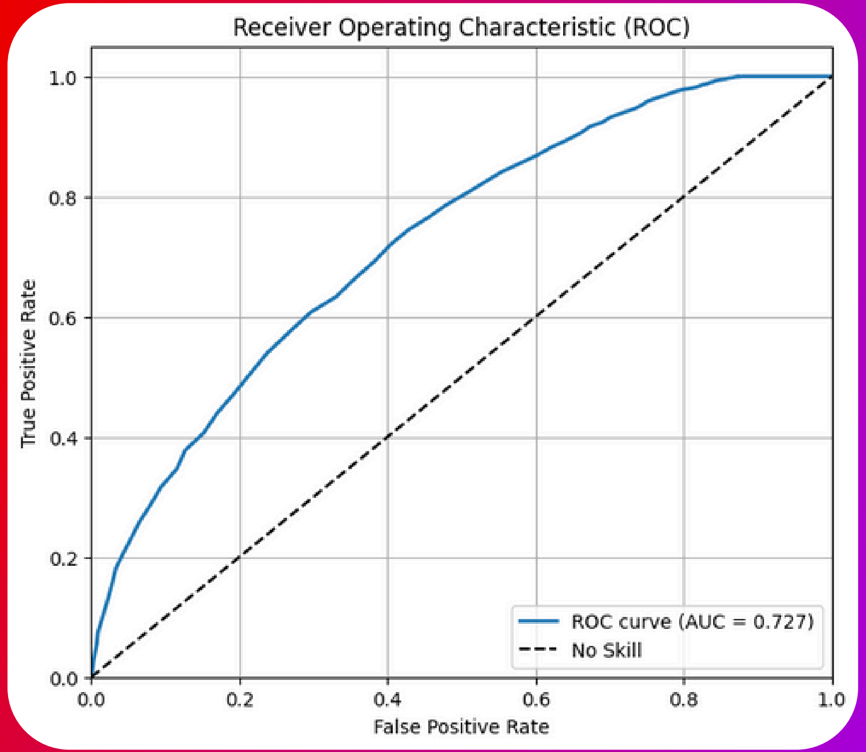




- **Clustering**
  - Elbow method to find clusters
  - Well-defined clusters
  - Not included as a new variable
- **PCA**
  - Dimensionality reduction
  - Noise removal
  - Improves clustering by generating uncorrelated feature

Random Forest Classifier

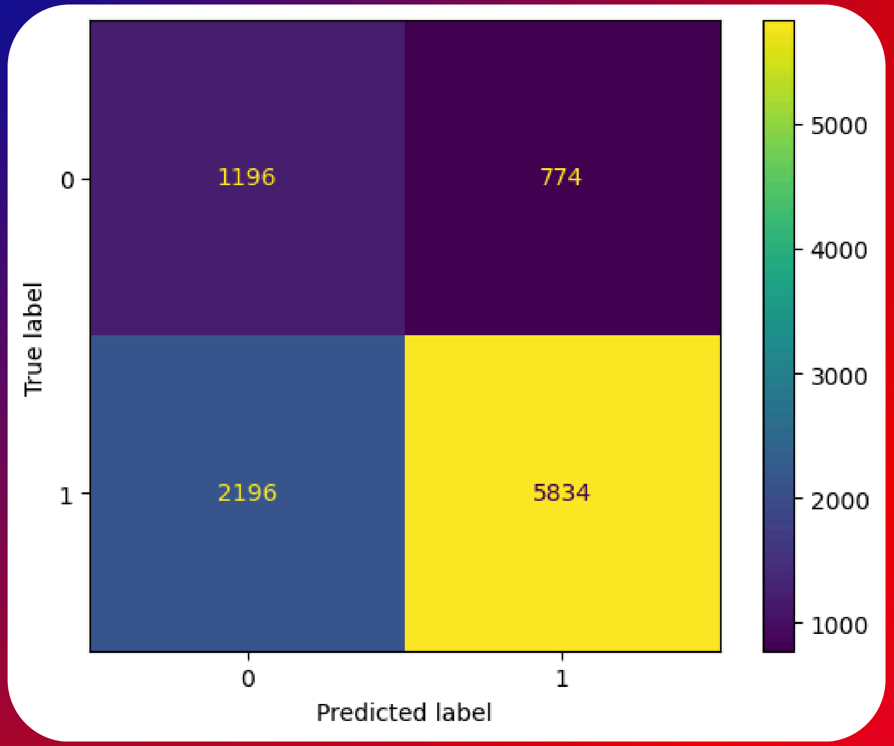


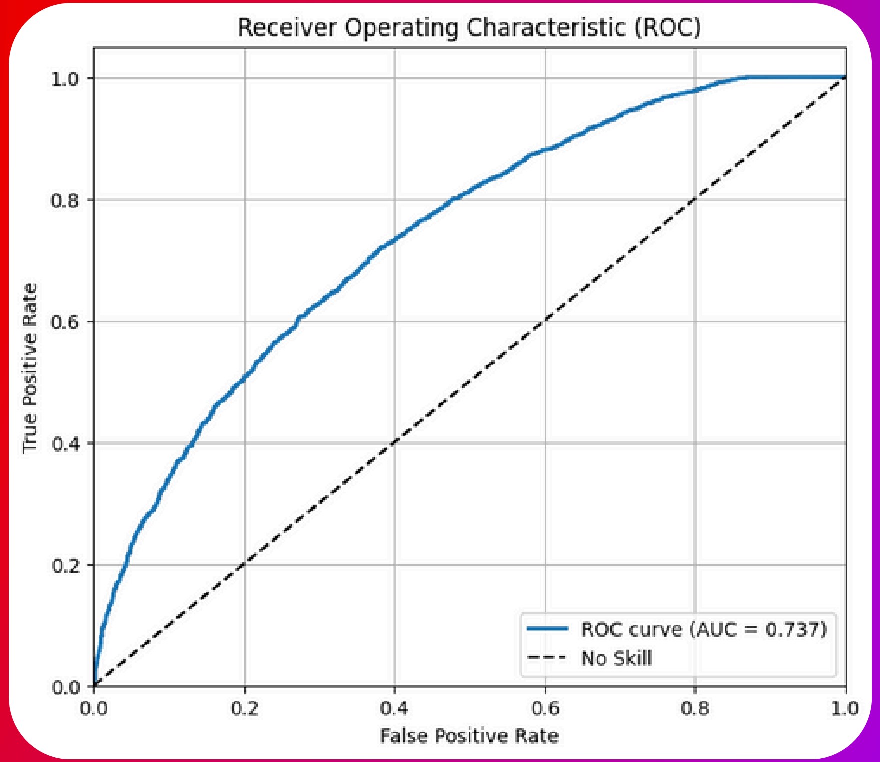


Metrics selection

- Accuracy: 0.823
- ROC accuracy: 0.726
- Precision: 0.835
- Recall: 0.972
- F1-score: 0.898

Logistic Regression





Metrics selection

- Accuracy: 0.703
- ROC accuracy: 0.737
- Precision: 0.882
- Recall: 0.726
- F1-score: 0.797

	RGR
collections_12_mths_ex_med	0.999855
purpose_auto	0.999736
tax_liens	0.999649
purpose_business	0.999492
purpose_home	0.999461
purpose_consolidation	0.999437
home_ownership_OWN	0.999387
purpose_personal	0.999326
pub_rec_bankruptcies	0.999135
delinq_2yrs	0.998845
pub_rec	0.998625
home_ownership_MORTGAGE	0.998491
home_ownership_RENT	0.998379
inq_last_6mths	0.997912
verification_status	0.997428
mths_since_last_delinq	0.996751
open_acc	0.996439
credit_inquiry_density	0.996347
revol_util	0.996083
mort_acc	0.995898
revol_bal	0.994905
credit_age_years	0.994728
emp_length	0.994288
annual_inc	0.993705
tot_cur_bal	0.992696
dti	0.991696
fico_average	0.991595
debt_burden	0.990002
loan_amnt	0.989449
term	0.984738
grade	0.982862
int_rate	0.970827
debt_settlement_flag	0.920129

$$RGR = \frac{\sum_{i=1}^{n}\left\{\frac{1}{n\hat{y}}\left(\sum_{j=1}^{i}\hat{y}_{r_{n+1-j}} - \sum_{j=1}^{i}\hat{y}_{r_j^p}\right)\right\}}{\sum_{i=1}^{n}\left\{\frac{1}{n\hat{y}}\left(\sum_{j=1}^{i}\hat{y}_{r_{n+1-j}} - \sum_{j=1}^{i}\hat{y}_{r_j}\right)\right\}}.$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

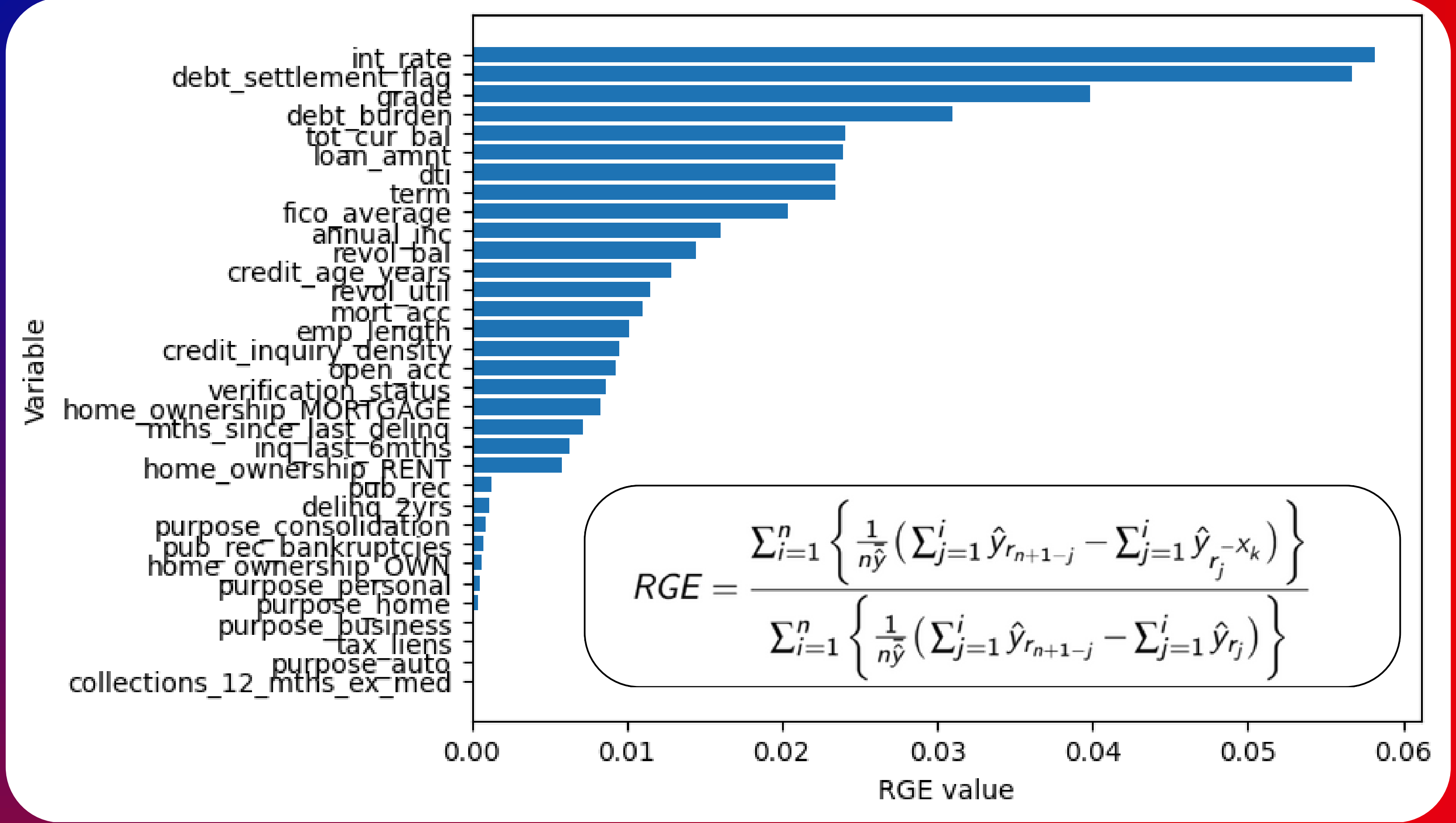
$$\text{TPR}\,(y) = \frac{TP}{TP + FN}$$

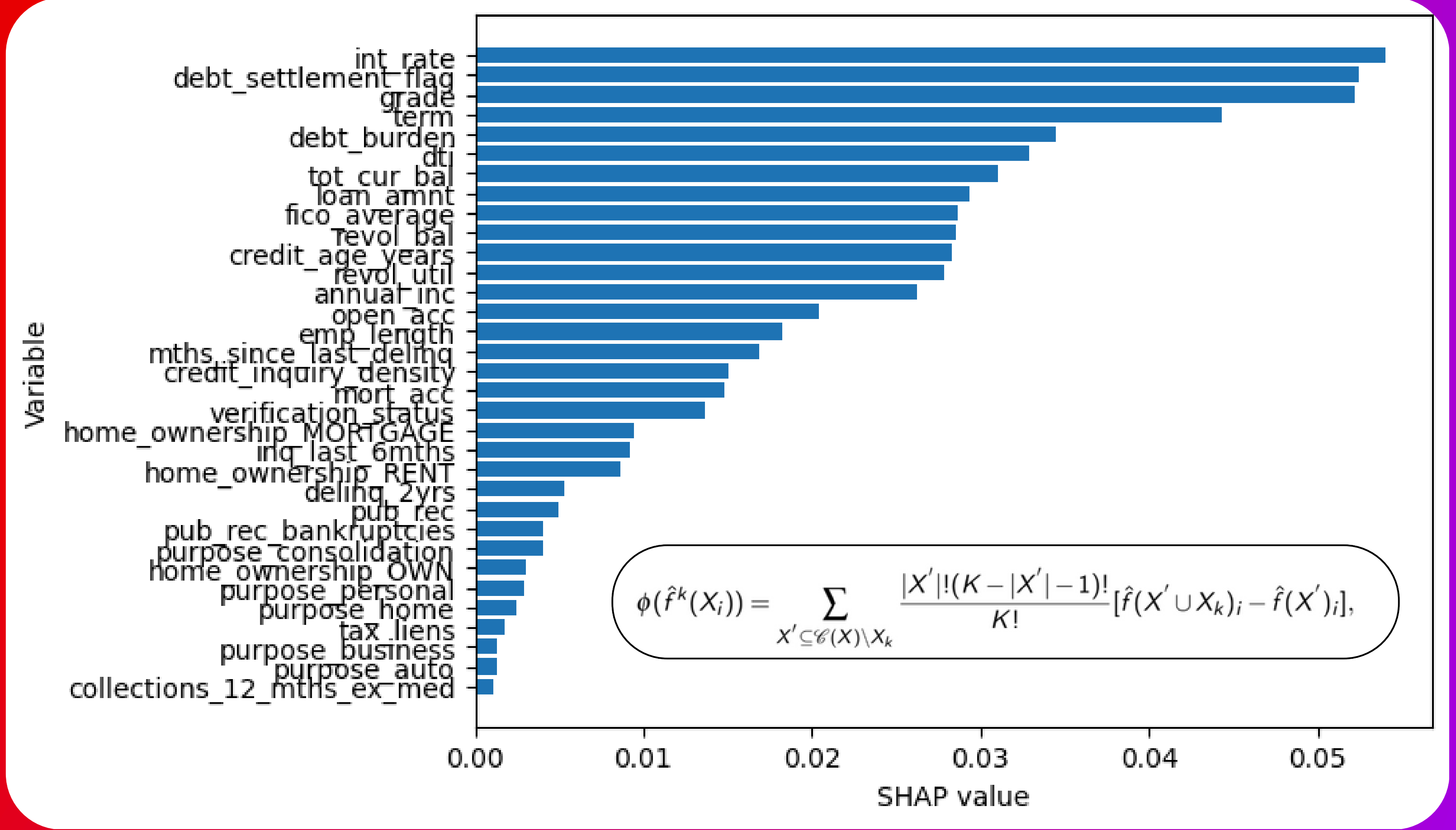
$$\text{FPR}(x) = \frac{FP}{FP + TN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **SUSTAINABILITY**
  - An high level of robustness for each variable (RGR)
- **ACCURACY**
  - Medium-high score for the accountability of the model
- **FAIRNESS**
  - Detect and mitigate bias in the model

Paolo Rinaldi





- **EXPLAINABILITY**
    - Understand how inputs drive predictions.
    - Feature importances
- **SHAP**
    - Provides unique additive explanations based on Shapley values.

Thank you!