PROJECT COMPLETTRION REPORT

Company Bankruptcy Prediction

For: Joblogic

Date: 01-06-2025

Presented By: Waleed Malik

Project Report

Objective

To develop a machine learning model that can predict whether a company is likely to go bankrupt based on its financial indicators collected over 5 years. The task is framed as a binary classification problem (0 = non-bankrupt, 1 = bankrupt).

Problem-Solving Approach

1. Understanding the Data The dataset contains 64 numerical financial features per company for each year. After merging all yearly datasets, we noticed:

Class imbalance: Very few bankrupt companies compared to non-bankrupt ones

Missing data: Some features had significant missing values

2. Data Preprocessing Dropped columns with more than 40% missing values

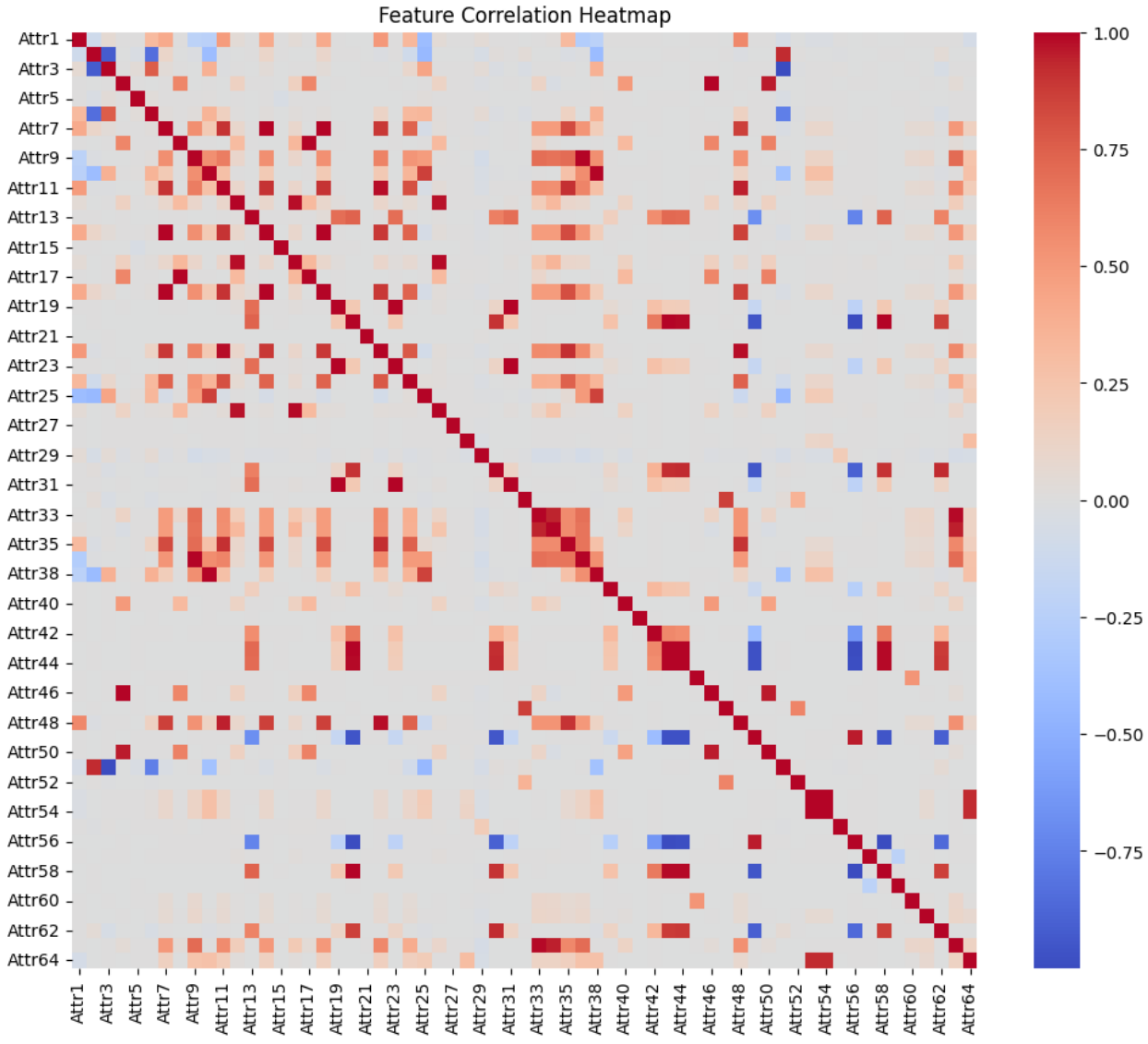
Imputed remaining missing values using the median (to handle outliers better)

Standardized all features using StandardScaler to normalize the range

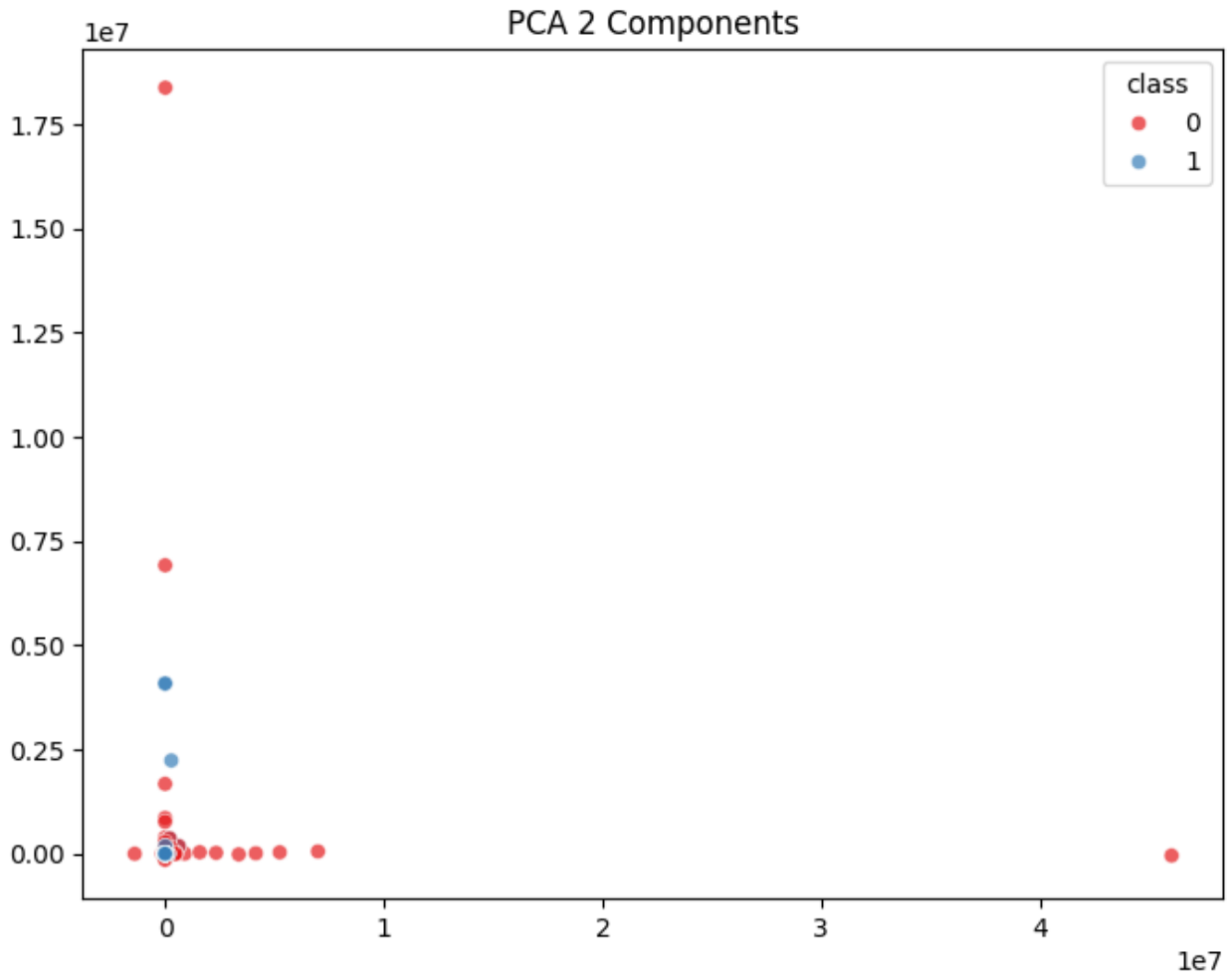
Handled class imbalance using SMOTE, which oversamples the minority class (bankrupt) by synthetically generating new examples

3. Exploratory Data Analysis Histograms showed that many features were skewed, which is common in financial data

Correlation heatmap helped understand redundancy and potential multicollinearity


Feature Correlation Heatmap
Attr1
Attr3
Attr5
Attr7
Attr9
Attr11
Attr13
Attr15
Attr17
Attr19
Attr21
Attr23
Attr25
Attr27
Attr29
Attr31
Attr33
Attr35
Attr38
Attr40
Attr42
Attr44
Attr46
Attr48
Attr50
Attr52
Attr54
Attr56
Attr58
Attr60
Attr62
Attr64
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Applied PCA to visualize class separability in a 2D space



Model Selection

- *Logistic Regression* algorithm for linear and binary classification problems. can be readily generalized to multiclass settings, which is known as multinomial logistic regression or softmax regression.
- *Random Forest* A random forest can be considered as an ensemble of decision trees. The idea behind a random forest is to average multiple (deep) decision trees that individually suffer from high variance to build a more robust model that has a better generalization performance and is less susceptible to overfitting.
- *XGBoost* Often top performer for structured/tabular data; handles imbalance well. It is essentially a computationally efficient implementation of the original gradient boost algorithm.

Evaluation

Train-Test Split (80/20 stratified) to ensure class balance

Performance Metrics:

- Accuracy
- Precision, Recall, F1-score (to handle imbalance)
- ROC-AUC (for overall class separability)
- Cross-Validation: Used StratifiedKFold to validate model robustness (5 folds)

Model Results

- Logistic Regression gave good baseline performance but struggled with recall missed some bankruptcies

```
[[6035 2228]
 [ 162  256]]
              precision    recall  f1-score   support

           0       0.97      0.73      0.83      8263
           1       0.10      0.61      0.18       418

    accuracy                           0.72      8681
   macro avg       0.54      0.67      0.51      8681
weighted avg       0.93      0.72      0.80      8681

ROC-AUC: 0.7321923348853799
```

- Random Forest performed better overall, especially in identifying bankrupt companies

```
[[8070  193]
 [ 201  217]]
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      8263
           1       0.53      0.52      0.52       418

    accuracy                           0.95      8681
   macro avg       0.75      0.75      0.75      8681
weighted avg       0.95      0.95      0.95      8681

ROC-AUC: 0.9271585675927797
```

- XGBoost had the best ROC-AUC and balanced precision/recall, making it the top choice

```
[[8150  113]
 [ 129  289]]
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      8263
           1       0.72      0.69      0.70       418

    accuracy                           0.97      8681
   macro avg       0.85      0.84      0.85      8681
weighted avg       0.97      0.97      0.97      8681

ROC-AUC: 0.9621926765247976
```

Performing Cross Validation to check the model's performances

```
Stratified K-Fold Cross-validation (5-fold):
LogReg: Mean AUC = 0.6705
RandomForest: Mean AUC = 0.9219
```

- Logistic Regression:

```
Stratified K-Fold Cross-validation (5-fold):
LogReg: Mean AUC = 0.6705
RandomForest: Mean AUC = 0.9219
```

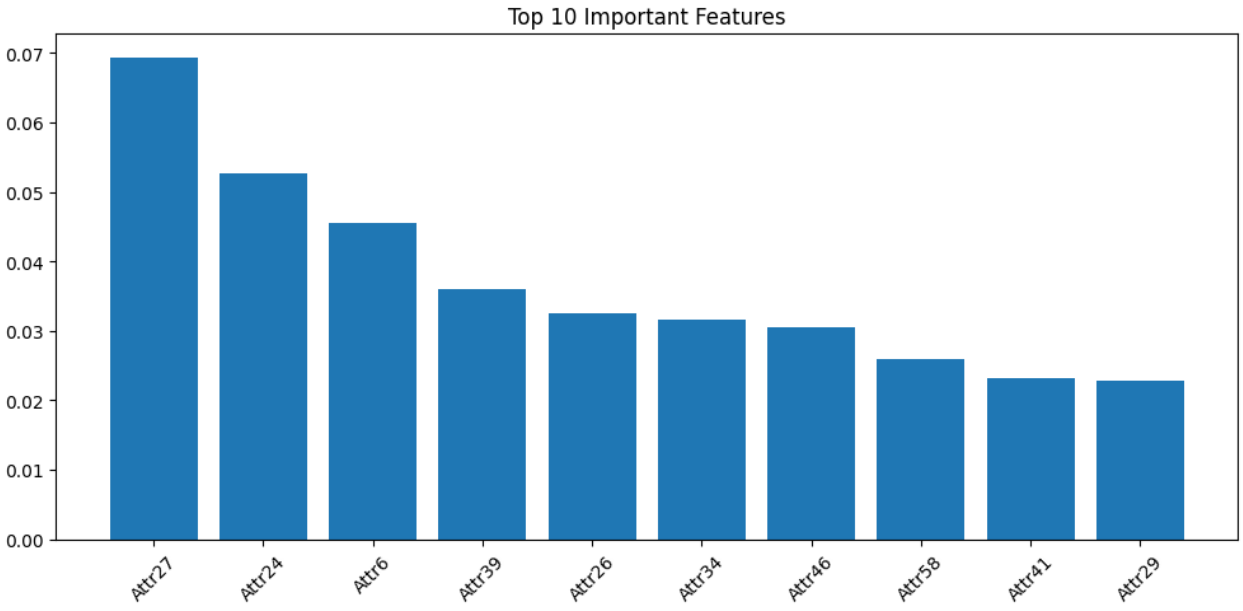
- Random Forest:

```
Stratified K-Fold Cross-validation (5-fold):
LogReg: Mean AUC = 0.6705
RandomForest: Mean AUC = 0.9219
```

- XGBoost:

```
XGBoost: Mean AUC = 0.9694
```

Feature Comparison



Conclusion

For this task, XGBoost was the most effective model due to its ability to handle imbalance, capture complex patterns, and deliver strong predictive performance. Random Forest was a close second and also helped interpret which financial features were most important.