Bankruptcy Prediction: Final Report

1. Introduction

The goal of this project is to predict company bankruptcy using machine learning models trained on financial indicators. A significant challenge is addressing the class imbalance in the dataset, where bankrupt companies constituted only a small portion of the total samples. To tackle this, we used two methods: **Class Weights**: Assigning weights to classes to balance their impact on the model's loss function. **SMOTE (Synthetic Minority Oversampling Technique)**: Generating synthetic samples for the minority class to achieve a balanced dataset.
Before applying these techniques, we trained several **baseline models** on the imbalanced data to evaluate their initial performance. These models included Logistic Regression, Support Vector Machine (SVC), Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). The results from these base models provided a benchmark to compare the improvements achieved after addressing class imbalance.

This report summarizes the findings from different approaches, highlights the best-performing model, and provides insights into feature importance.

2. Methodology

Feature Engineering

There is no missing value or duplicate in our data. We drop the columns having standard deviation 0, or correlation 1 or -1 with other columns. We also use t-test to recognize non-significant features and we drop them if our model performance doesn't reduce, which is the case in this project.

Base Models

Before addressing class imbalance, we trained several baseline models to evaluate their initial performance on the imbalanced dataset. These models were trained without applying any techniques for handling the imbalance, providing a benchmark for subsequent improvements.

Addressing Class Imbalance

Class Weights:

- Applied to models that support class weights (e.g., Random Forest, Logistic Regression, Decision Tree).
- Allows the model to give more importance to the minority class without altering the dataset.

SMOTE:

- Synthesizes new samples for the minority class to balance the dataset.
- Applied during the training phase using pipelines to integrate preprocessing seamlessly.

Models Evaluated

We trained the following machine learning models using the methods:

1. Logistic Regression
2. Support Vector Classifier (SVC)
3. Decision Tree
4. Random Forest
5. K-Nearest Neighbors (KNN)
6. Multi-Layer Perceptron (MLP)

3. Results

3.a. Base Models Performance

In this section, we present the performance of all models trained without addressing class imbalance (baseline models). These models were trained on the original, imbalanced dataset, which highlights the challenge of dealing with highly imbalanced classes. The following table summarizes the test set performance of the baseline models:

Model	Accuracy	Balanced Accuracy	Recall	F1-Score	Precision	Log Loss	ROC-AUC
Decision Tree	0.9582	0.6708	0.3636	0.3596	0.3556	1.5062	0.6708
Multi-Layer Perceptron (MLP)	0.9589	0.6383	0.2955	0.3171	0.3421	0.2956	0.8335
Random Forest	0.9699	0.5780	0.1591	0.2545	0.6364	0.1301	0.9197
K-Nearest Neighbors (KNN)	0.9699	0.5780	0.1591	0.2545	0.6364	0.4959	0.7755
Logistic Regression	0.9670	0.5765	0.1591	0.2373	0.4667	0.1577	0.8798
Support Vector Classifier (SVC)	0.9677	0.5000	0.0000	0.0000	0.0000	NA	NA

Key Observations:

- **Decision Tree** emerged as the best-performing baseline model, achieving the highest balanced accuracy (0.6708), though it still performed poorly in other metrics like recall, F1-score and precision.
- **MLP** showed similar accuracy and balanced accuracy but also struggled with recall and F1-score, indicating potential overfitting to the majority class.
- **Random Forest** and **KNN** performed well in terms of accuracy but struggled with class imbalance, as evidenced by their low balanced accuracy and recall values.
- **SVC** performed poorly across all metrics, with a recall of 0, highlighting its difficulty in dealing with imbalanced data. This model simply predicts all labels to be the majority class.

3.b. Performance Using Class Weights

The table below summarizes the test set performance of all models trained using class weights:

Model	Accuracy	Balanced Accuracy	Recall	F1-Score	Precision	Log Loss	ROC-AUC
Random Forest	0.8944	0.8576	0.8182	0.3333	0.2093	0.2793	0.9317
Decision Tree	0.8585	0.8500	0.8409	0.2772	0.1659	0.3266	0.9217
Logistic Regression	0.8563	0.8489	0.8409	0.2741	0.1637	0.3791	0.8960
Support Vector Classifier (SVC)	0.8944	0.7697	0.6364	0.2800	0.1795	NA	NA
Multi-Layer Perceptron (MLP)	0.9641	0.5750	0.1591	0.2222	0.3684	0.1591	0.8609
K-Nearest Neighbors (KNN)	0.9641	0.5420	0.0909	0.1404	0.3077	0.5739	0.7546

Key Observations:

- **Random Forest** achieved the highest balanced accuracy (0.8576) and ROC-AUC (0.9317), making it the best-performing model using class weights.
- **Decision Tree** showed competitive performance but had lower precision and F1-score compared to Random Forest.
- Models like **MLP** and **KNN** achieved high accuracy but poor balanced accuracy and recall, indicating overfitting to the majority class. The reason is that no class weight is applied to them.

3.c. Performance Using SMOTE

The table below summarizes the test set performance of all models trained using SMOTE:

Model	Accuracy	Balanced Accuracy	Recall	F1-Score	Precision	Log Loss	ROC-AUC
Random Forest	0.8900	0.8663	0.8409	0.3304	0.2056	0.2374	0.9445
Multi-Layer Perceptron	0.8688	0.8663	0.8636	0.2980	0.1801	0.4475	0.9077
K-Nearest Neighbors	0.8372	0.8500	0.8636	0.2550	0.1496	0.9565	0.9124
Logistic Regression	0.8717	0.8458	0.8182	0.2915	0.1773	0.3177	0.9345
Decision Tree	0.8109	0.8254	0.8409	0.2229	0.1285	0.3109	0.8869
Support Vector Classifier	0.1598	0.4341	0.7273	0.0529	0.0274	NA	NA

Key Observations:

- **Random Forest** remains the best-performing model, achieving the highest balanced accuracy (0.8663) and ROC-AUC (0.9445).

- **MLP** matches Random Forest in balanced accuracy but lags in precision and ROC-AUC, making it a close second.
- **KNN**, **Logistic Regression** and Decision Tree showed moderate performance, while **SVC** performed poorly with SMOTE compared to class weight and also to other models.

4. Feature Importance

Using the Random Forest model trained with SMOTE, the most important financial indicators were identified:

Feature	Importance
Borrowing Dependency	0.1196
Persistent EPS in the Last Four Seasons	0.0856
Continuous interest rate (after tax)	0.0644
Liability to Equity	0.0608
Retained Earnings to Total Assets	0.0519

These features highlight key financial indicators associated with bankruptcy risk, providing actionable insights for domain experts.

5. Conclusion

Best Model: The **Random Forest model with SMOTE** emerged as the best-performing approach, achieving the highest balanced accuracy (0.8663) and ROC-AUC (0.9445).

Comparison of Methods: While both **class weights** and **SMOTE** improved model performance, SMOTE yielded slightly better results overall.

Recommendations:

- Deploy the Random Forest model trained with SMOTE for predicting company bankruptcy.
- Regularly update the model with new data and re-evaluate its performance.
- Focus on the top features identified to prioritize financial indicators that contribute most to bankruptcy risk.