Company Bankruptcy Prediction and Analysis

Xinyi Liu

Introduction:

The prediction of bankruptcy is a significant issue in finance, as accurate predictions enable stakeholders to take early actions to minimize their economic losses. In recent years, numerous studies have investigated the use of machine learning models for predicting bankruptcy, utilizing financial ratios as predictors. As well as for this project, the predictive task is to use financial indicators to predict bankruptcy status, compare the results with different ML models, and finally get the optimal prediction. The project is started with exploratory data analysis (EDA) to gain insights into the features. Then, it moves on to baseline models such as logistic regression and K-nearest neighbors (KNN). Finally, advanced methods including Random Forest and SMOTE were used to tackle class imbalance. The findings show that relying solely on naive accuracy metrics is inadequate; instead, recall, precision, F1 score, and AUC are more suitable metrics for this domain.

Data Description

The dataset used in this prediction project is a well-structured financial dataset that is collected from the Taiwan Economic Journal for the year 1999 to 2009, and accordingly, company bankruptcy is defined based on the business regulations of the Taiwan Stock Exchange. In terms of the data presented, they are static cross-sectional snapshots rather than time-series trends. This dataset is a considerably large dataset, which has a total of 6813 firms observed, 95 financial metrics and most importantly, a target column of “bankruptcy” with binary values. Notably, all features are numerical financial indicators including Profitability Ratios (ROA(A) before interest and % after tax, Net Income to Total Asset, Retained Earnings to Total Asset), Leverage Ratios (Debt Ratio %, Net Worth/Asset), Cash Flow and Earnings Stability (Persistent EPS in the Last Four Season).

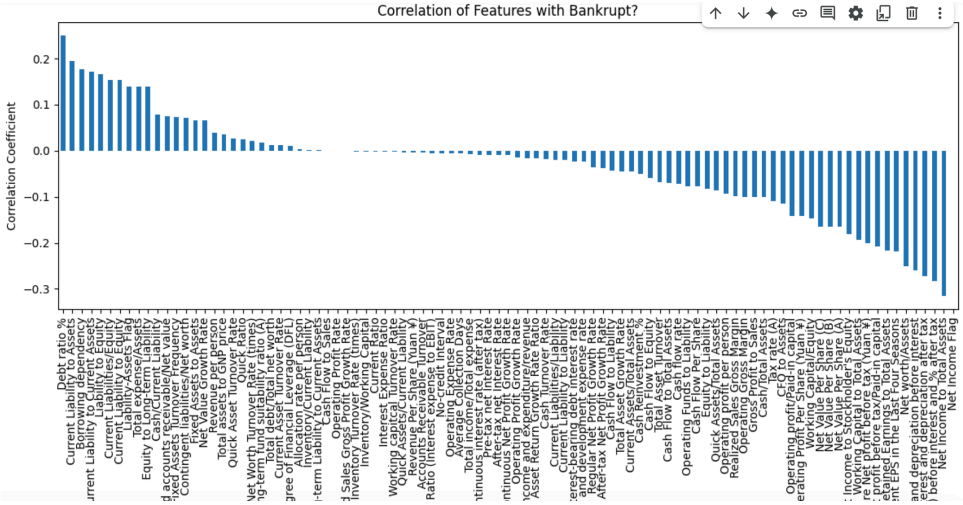
Models and Methods

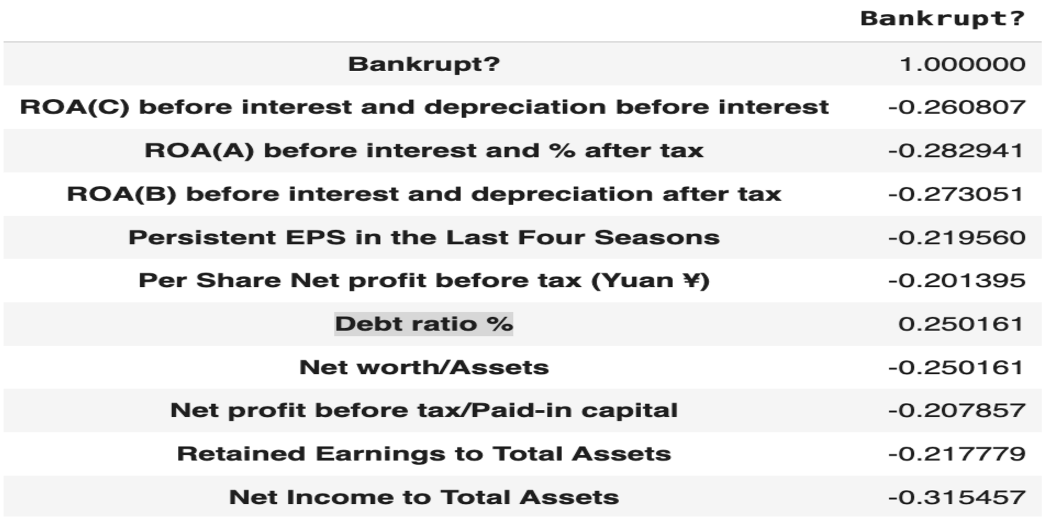
- **<u>Exploratory Data Analysis</u>**

Given that this dataset comprises 96 variables, it is crucial to pinpoint those that exhibit the strongest correlations with bankruptcy. To achieve this, I have utilized a correlation heatmap.

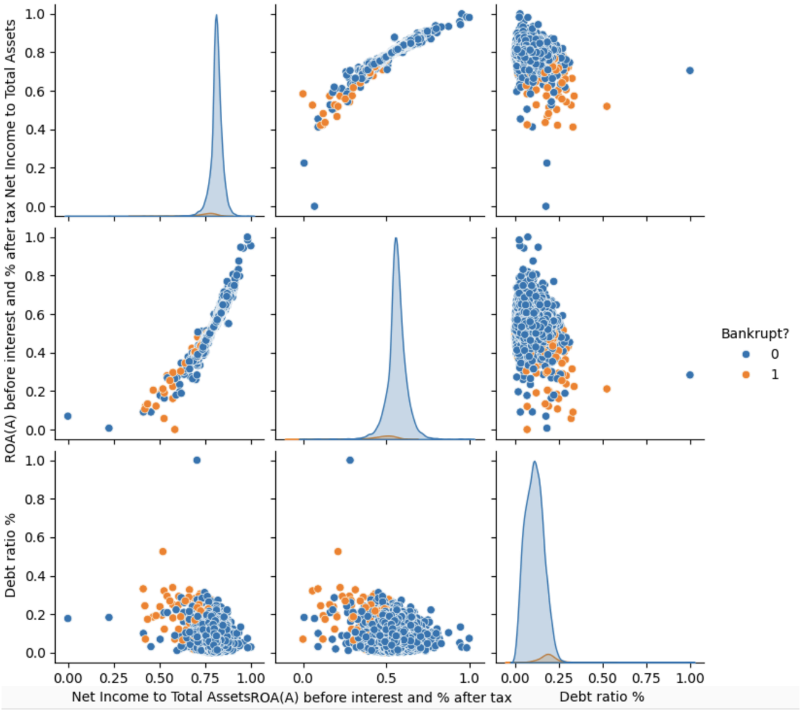
This graph highlights the top 10 variables that are either positively or negatively correlated with bankruptcy. For enhanced accuracy, it's better to concentrate on variables that have correlation values exceeding 0.2 or falling below -0.2. Histograms and boxplots clearly demonstrated the differences in distribution between bankrupt and non-bankrupt firms. In terms of the histogram, it demonstrates clear differences in the financial profiles of bankrupt and non-bankrupt firms. Across various profitability ratios, bankrupt firms consistently display lower returns compared to healthy firms, and this indicates that persistently low profitability is a strong indicator of financial distress. Similarly, metrics like Net Income to Total Assets and Net Profit before Tax relative to Paid-in Capital reveal that bankrupt companies are generally less efficient at generating earnings from their assets and capital.

Leverage-related indicators like Debt Ratio % and Net Worth to Assets, further distinguish the two groups. The targeted groups tend to have slightly higher levels of debt and lower equity, which reflects a greater financial risk. Additionally, the distributions for Persistent EPS and Retained Earnings to Total Assets illustrate that healthy firms maintain more stable earnings over time and accumulate profits, while distressed firms demonstrate greater volatility and weaker capital retention. These visualizations collectively support the selection of these features for predictive modeling, as they consistently and clearly differentiate between the two categories.

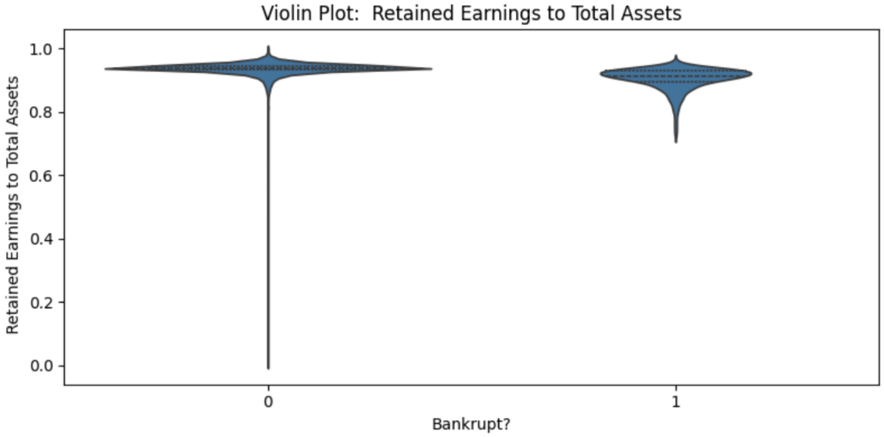




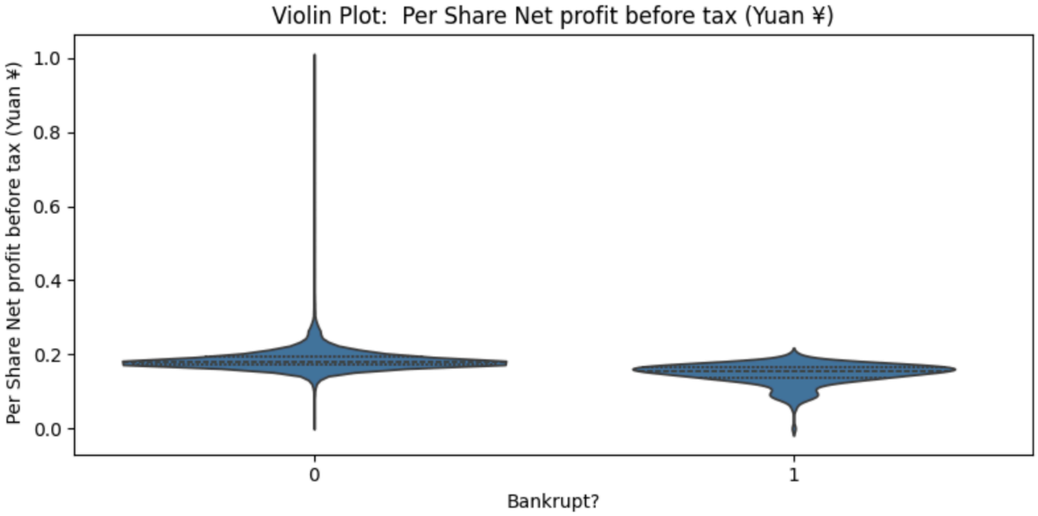
	Bankrupt?
Bankrupt?	1.000000
ROA(C) before interest and depreciation before interest	-0.260807
ROA(A) before interest and % after tax	-0.282941
ROA(B) before interest and depreciation after tax	-0.273051
Persistent EPS in the Last Four Seasons	-0.219560
Per Share Net profit before tax (Yuan ¥)	-0.201395
Debt ratio %	0.250161
Net worth/Assets	-0.250161
Net profit before tax/Paid-in capital	-0.207857
Retained Earnings to Total Assets	-0.217779
Net Income to Total Assets	-0.315457



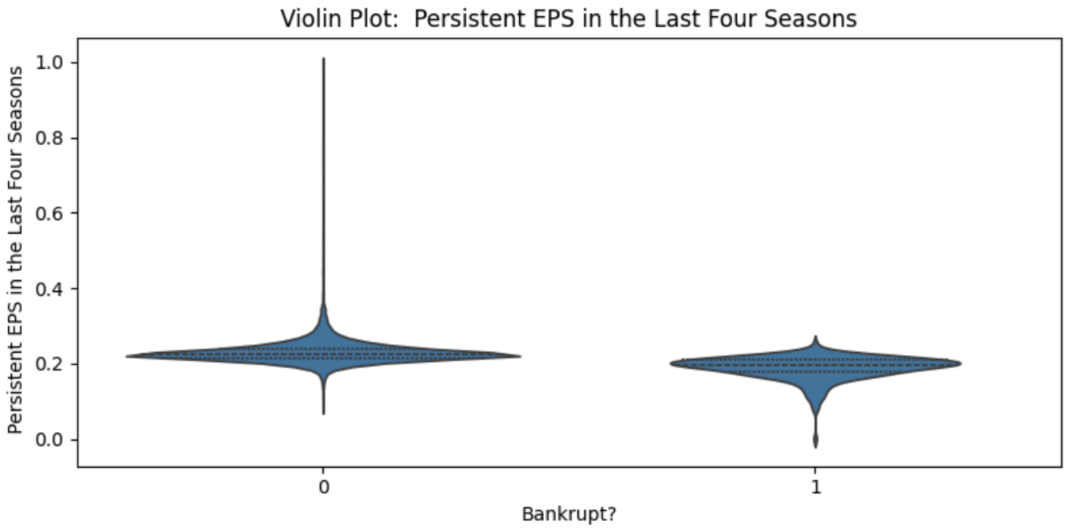
To confirm this relationship, I took three strongest correlated variables to run a pairplot,and this plot clearly illustrates that these three features provide distinct, although not flawless, visual separation between bankrupt and non-bankrupt firms. Bankrupt companies typically exhibit low profitability and relatively high debt levels, while financially healthy firms tend to cluster around high return on assets (ROA), strong net income efficiency, and low leverage. These visual insights validate the use of these features in predictive modeling and support their inclusion in the final feature set.



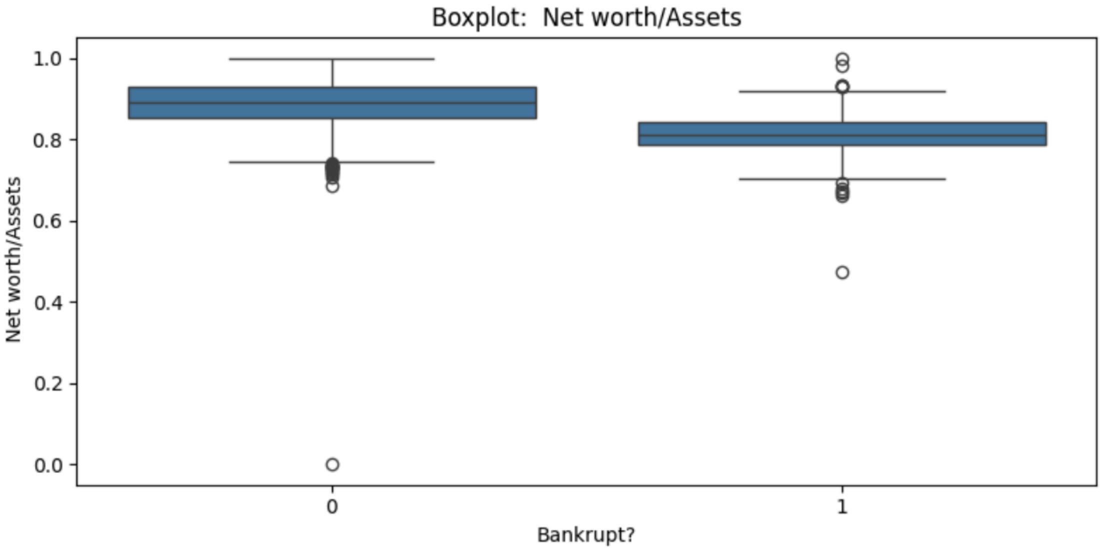
From the graph, we are able to confirm that non-bankrupt firms typically have significantly higher and more concentrated retained earnings, while bankrupt firms tend to show lower and more dispersed values, suggesting that retained earnings are a strong indicator of financial health.



This violin plot indicates that bankrupt firms show a modest shift toward lower profit per share. This gives an assumption that declining profitability may precede bankruptcy, though both groups share similar central ranges.



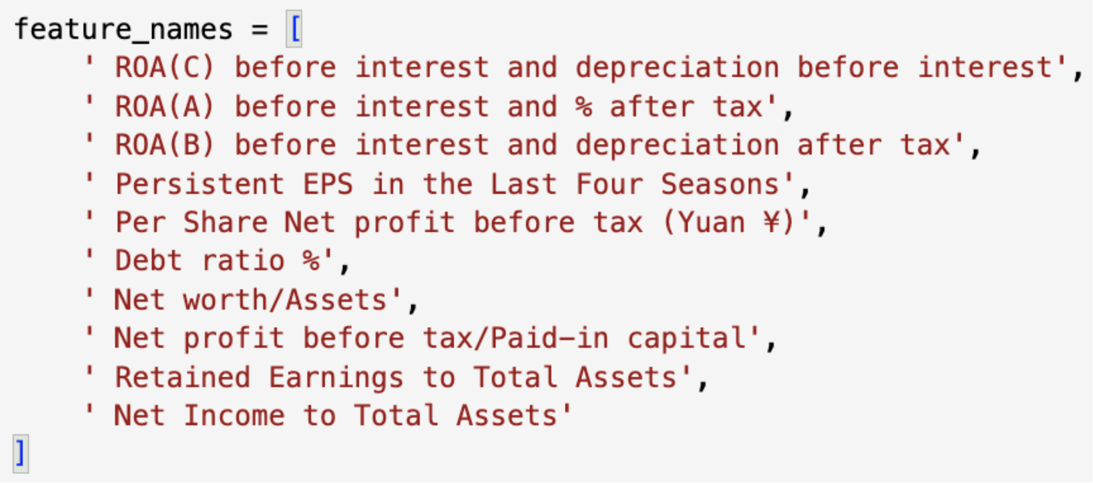
From this graph we see that bankrupt firms exhibit a lower and tighter distribution of persistent EPS, reflecting weaker and less stable earnings performance compared to non-bankrupt firms.



Furthermore, this box plot shows non-bankrupt firms (0) tend to have higher median net worth ratios, with most values tightly clustered between ~0.85 and 0.95. In contrast, bankrupt firms (1) have a lower median and a wider interquartile range, indicating more variability and financial instability. Notably, the bankrupt group contains more low-end outliers, reflecting firms with severely diminished equity positions, confirming it is a strong indicator of bankruptcy.

- **Two Basic Classifier**

Before running the classification regression, we have to split the data into test and train data with a random state equals to 10, confirming that the splitted data will always be the same every time running the code. X data consists of selected financial ratios that describe each company's performance, while y is the binary target variable indicating whether a company went bankrupt. Here, X consists of all the features shown below.

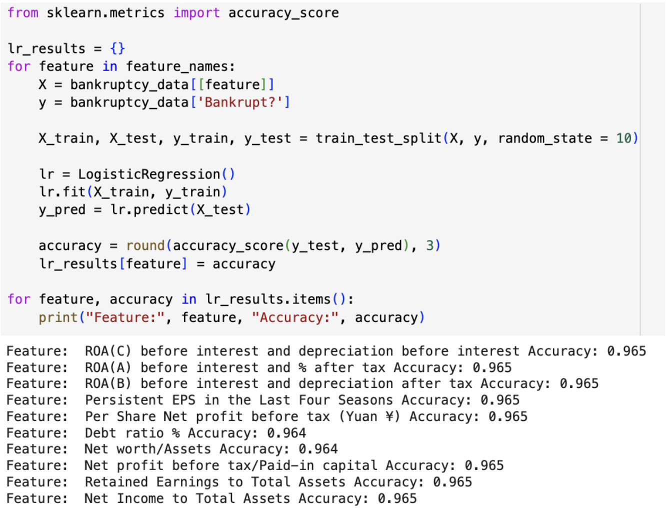


```
feature_names = [
    ' ROA(C) before interest and depreciation before interest',
    ' ROA(A) before interest and % after tax',
    ' ROA(B) before interest and depreciation after tax',
    ' Persistent EPS in the Last Four Seasons',
    ' Per Share Net profit before tax (Yuan ¥)',
    ' Debt ratio %',
    ' Net worth/Assets',
    ' Net profit before tax/Paid-in capital',
    ' Retained Earnings to Total Assets',
    ' Net Income to Total Assets'
]
```

1. *Logistic Regression:*

A logistic regression model was fitted to the training data, predictions were made on the test set, and the accuracy of each model was recorded using "accuracy_score". Because there are several variables incorporated, the loop was repeated for all selected variables, and their respective accuracies were stored for comparison. The final results showed that all features yielded nearly identical accuracy scores of around 0.965, which closely matches the baseline accuracy due to the dataset's strong class imbalance.

This indicates that the models primarily predicted the majority class (non-bankrupt) and failed to capture the minority class (bankrupt), emphasizing that accuracy alone is insufficient for evaluating model performance in imbalanced classification tasks.

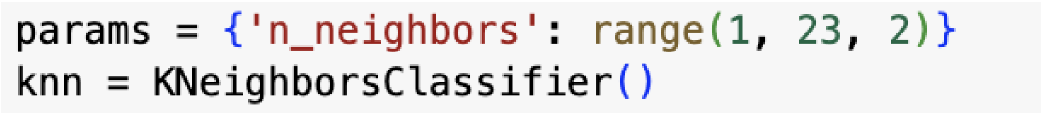


```
from sklearn.metrics import accuracy_score

lr_results = {}
for feature in feature_names:
    X = bankruptcy_data[[feature]]
    y = bankruptcy_data['Bankrupt?']

    X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 10)

    lr = LogisticRegression()
    lr.fit(X_train, y_train)
    y_pred = lr.predict(X_test)

    accuracy = round(accuracy_score(y_test, y_pred), 3)
    lr_results[feature] = accuracy

for feature, accuracy in lr_results.items():
    print("Feature:", feature, "Accuracy:", accuracy)
```

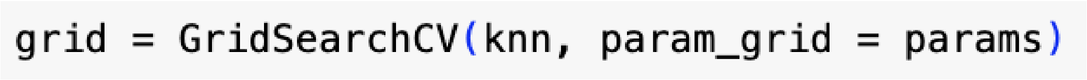
```
Feature:  ROA(C) before interest and depreciation before interest Accuracy: 0.965
Feature:  ROA(A) before interest and % after tax Accuracy: 0.965
Feature:  ROA(B) before interest and depreciation after tax Accuracy: 0.965
Feature:  Persistent EPS in the Last Four Seasons Accuracy: 0.965
Feature:  Per Share Net profit before tax (Yuan ¥) Accuracy: 0.965
Feature:  Debt ratio % Accuracy: 0.964
Feature:  Net worth/Assets Accuracy: 0.964
Feature:  Net profit before tax/Paid-in capital Accuracy: 0.965
Feature:  Retained Earnings to Total Assets Accuracy: 0.965
Feature:  Net Income to Total Assets Accuracy: 0.965
```

2. *KNN model*:

In the K-Nearest Neighbors (KNN) modeling process, a grid search was conducted to determine the optimal number of neighbors (n_neighbors) for classification. The parameter grid included odd values ranging from 1 to 21, and GridSearchCV was employed to evaluate each configuration using cross-validation.



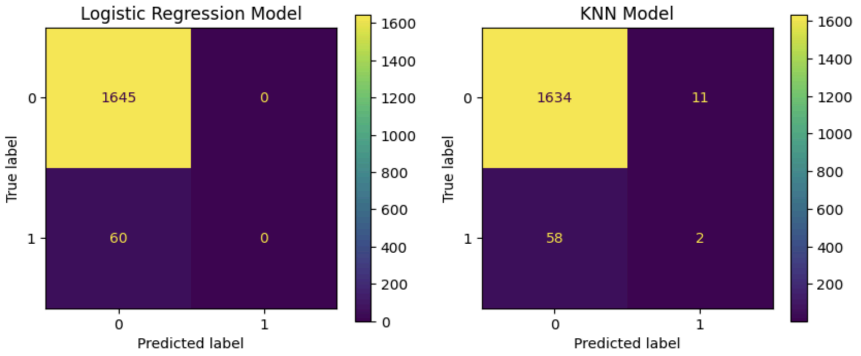
```
params = {'n_neighbors': range(1, 23, 2)}
knn = KNeighborsClassifier()
```



```
grid = GridSearchCV(knn, param_grid = params)
```

The model was trained on the complete dataset (X, y) in order to identify the best KNN setup. After fitting the model, the “grid.score(X, y)” method indicated a high accuracy of approximately 0.969. However, as this evaluation was conducted on the same dataset used for training, the score is likely overestimated and may not accurately represent the model's generalization performance. Similar to the logistic regression model, this elevated accuracy is primarily due to class imbalance, resulting in the model being biased toward predicting the majority class (non-bankrupt firms), which ultimately limits its effectiveness in identifying actual bankruptcies.

3. *ConfusionMatrix:*

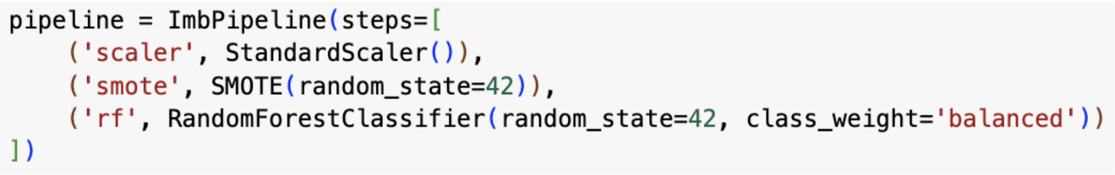


Here, we want to find out the recall for both of the two models using Confusion Matrix. We discovered that both the models have high accuracy because of the majority class dominance and have a recall value of 0 and 0.003 respectively. This demonstrates that neither of the models is efficient at predicting bankruptcies, which is the main goal of the model. More importantly, the logistic regression has completely failed on predicting bankruptcy for a TP = 0; Though KNN performs slightly better than Logistic from this aspect, it is still doing poorly on the minority classification. Therefore, we need to develop a new model.

- **Random Forest**

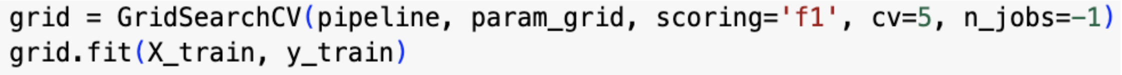
  This regression model is targeted at optimizing the prediction. To run a Random Forest Classifier, the first thing needed is to select the 10 variables that show the strongest correlation with the target variable "Bankrupt?". By focusing on the feature set selected, this model aims to balance the predictive power with interpretability and reduce the risk of overfitting with the majority class.

  To ensure that the model performs well despite the severe class imbalance (with only about 3.2% of companies being bankrupt), we employ a stratified train-test split along with two additional models, thereby maintaining the class ratio in both sets. A key innovation in this modeling pipeline is the integration of SMOTE (Synthetic Minority Over-sampling Technique) during the training process. SMOTE generates synthetic examples of the minority class (bankrupt firms) by interpolating between existing samples. This oversampling strategy is combined with the "class_weight='balanced'" option in the Random Forest classifier, ensuring that both classes contribute equally to the model's learning, despite the skewed distribution of the dataset. Furthermore, the pipeline is built using ImbPipeline, which sequentially applies data standardization with StandardScaler, SMOTE, and Random Forest classification.



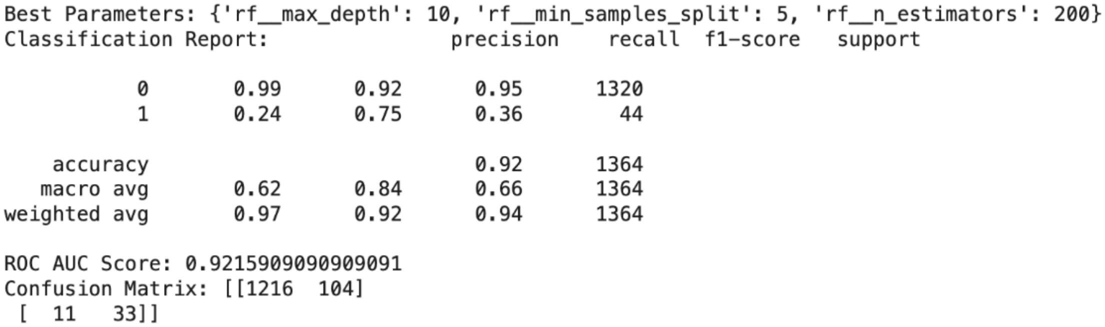
```
pipeline = ImbPipeline(steps=[
    ('scaler', StandardScaler()),
    ('smote', SMOTE(random_state=42)),
    ('rf', RandomForestClassifier(random_state=42, class_weight='balanced'))
])
```

  This model also needs GridSearch for an optimal regression performance. Different from the KNN model, grid search used here coworks with 5-fold cross validation (cv=5) and is applied across a predefined parameter space to test different values for the number of trees (n_estimators), maximum tree depth (max_depth) and minimum sample required to split a node (min_samples_split). Unlike the accuracy_score used for other two models, which is misleading in imbalanced contexts, the optimization metric used here is the F1-score, which is a harmonic mean of precision and recall, which better reflects the model's performance on the minority class.



```
grid = GridSearchCV(pipeline, param_grid, scoring='f1', cv=5, n_jobs=-1)
grid.fit(X_train, y_train)
```

After training, the best model was evaluated on the test set using several metrics: a classification report, ROC AUC score, and confusion matrix.



```
Best Parameters: {'rf__max_depth': 10, 'rf__min_samples_split': 5, 'rf__n_estimators': 200}
Classification Report:               precision    recall  f1-score   support

           0       0.99      0.92      0.95      1320
           1       0.24      0.75      0.36        44

    accuracy                           0.92      1364
   macro avg       0.62      0.84      0.66      1364
weighted avg       0.97      0.92      0.94      1364

ROC AUC Score: 0.9215909090909091
Confusion Matrix: [[1216  104]
 [  11   33]]
```

The model achieved an AUC of 0.921,which is a better discriminatory power between bankrupt and non-bankrupt firms. The classification report shows a recall of 0.75 for the bankrupt class, which means 75% of actual bankruptcies were correctly identified. Here, we can compare this recall value with the previous two regressions of which have a recall value of 0 and 0.003 respectively.

This represents a significant improvement over baseline models that failed to identify any bankruptcies. Although the precision for bankrupt predictions is modest at around 0.34, this trade-off is generally acceptable in financial risk analysis. In this context, the cost of missing a bankruptcy (a false negative) typically outweighs the cost of a false alarm (a false positive). The confusion matrix supports this, showing 33 true positives, 11 false negatives, and 104 false positives out of over 1,300 samples.

In conclusion, the Random Forest pipeline demonstrates a robust and well-balanced approach to bankruptcy prediction. By effectively addressing class imbalance and optimizing for meaningful performance metrics, the model significantly outperforms naive benchmarks. It offers practical utility in early warning systems for credit risk monitoring, investor due diligence, and regulatory auditing—areas where detecting rare but high-impact events like bankruptcy is critical.

Model Summary and Step Forward

- **Model Summary**

In this bankruptcy analysis, the main goal is to predict the corporate bankruptcy using financial ratios, due to the binary limitation of the target. The process begins with exploratory data analysis (EDA) to understand the distribution of key features and identify those most correlated with bankruptcy. By running a simple heatmap, ten top-performing features were selected, which include variations of Return on Assets (ROA), Debt Ratio %, Net Worth to Assets, Net Income to Total Assets, and Persistent EPS. Histograms and visualizations reveal that bankrupt firms consistently exhibit lower profitability, higher leverage, and weaker earnings stability, supporting the choice of these predictors.

The next step is to implement initial models using logistic regression and K-nearest neighbors (KNN) to establish baselines. Although both models achieve high accuracy (approximately 96%), they fail to detect any bankrupt firms, as indicated by a recall of zero for the minority class. These results emphasize the inadequacy of using accuracy as a performance metric in imbalanced settings, prompting us to shift our focus to more appropriate metrics such as recall, F1-score, and ROC AUC. It's crucial we understand that this bankruptcy dataset itself is a highly imbalanced dataset, therefore the high accuracy result from the Logistic and KNN model is not completely accurate. To address this issue With a more advanced Random Forest regression with F-1 score testing, we were able to get a stronger generalization performance and practical relevance.

- **Step Forward**

Machine Learning:

For further analysis, several directions can significantly enhance the predictive power, interpretability, and practical value of the bankruptcy prediction model.

1) First, use more advanced gradient boosting models such as XGBoost, LightGBM, or CatBoost. These models often outperform Random Forests on structured data due to their ability to capture nonlinear feature interactions and incorporate regularization for better generalization.
2) Cost-sensitive learning techniques can be introduced by explicitly assigning different penalties to false negatives and false positives, aligning the model's optimization with real-world financial risk.

These tools assist in understanding which features most impact individual predictions, enhancing model transparency and improving communication with stakeholders such as analysts or regulators..

Data Base:

1) In terms of data, one promising direction is to incorporate temporal dynamics if longitudinal financial data is available. Time-aware models or sequence-based approaches like rolling averages or LSTM networks can detect deteriorating financial patterns over time, which static models cannot capture.
2) We need to integrate external data like macroeconomic indicators, sector-level financial health, or credit ratings; these resources could enrich the feature set and improve robustness, especially during economic downturns when bankruptcy risk is elevated.
3) Finally, the model should be validated on a truly unseen holdout dataset or applied to a new cohort of firms to test generalizability. Cross-validation is useful during development, but real-world deployment demands out-of-sample validation. These next steps will transform the current predictive pipeline from a technically sound model into a practical early-warning tool that better aligns with the complexities and costs of financial risk assessment.