CSE4062

Introduction to Data Science and Analytics

2023-2024 Spring

Group 5

Company Bankruptcy Prediction

Delivery 3 Report

Ahmet Onat Özalan - 150118054 - Computer Engineering

Email: o171141@gmail.com

Selin Zeydan - 150322823 - Industrial Engineering

Email: selinzeydan9@gmail.com

Mediha Ecem Polat - 150820822 - Bioengineering

Email: medihaecempolat@gmail.com

Fırat Bakıcı - 150120029 - Computer Engineering

Email: firat143@gmail.com

Kardelen Kubat - 150118056 - Computer Engineering

Email: kardelenkubatcse@gmail.com

Berfin Ege Yarba - 150321036 - Industrial Engineering

Email: berfinegeyarba@gmail.com

Osman Buğra Göktaş - 150119565 - Computer Engineering

Email: osmanbugrag@gmail.com

Project Description

Feature Selection

In any machine learning project, the selection of relevant features plays a crucial role in model performance and interpretability. In our project, we employed feature selection techniques to identify the most informative attributes from the dataset. The feature selection process was carried out using scikit-learn, a popular machine learning library in Python.

Dataset Preprocessing

The dataset, sourced from Kaggle, was preprocessed initially to ensure consistency and reliability in the subsequent analysis. We employed the following steps:

Data Import: The dataset was loaded into a pandas DataFrame from a CSV file named "data.csv".

Column Name Cleaning: Leading and trailing whitespaces in the column names were removed for consistency.

Column Removal: The column "Net Income Flag" was dropped as its value was the same for all instances.

Feature-Target Separation: The dataset was divided into features (X) and the target variable (y), where the target variable represents the binary classification of bankruptcy status.

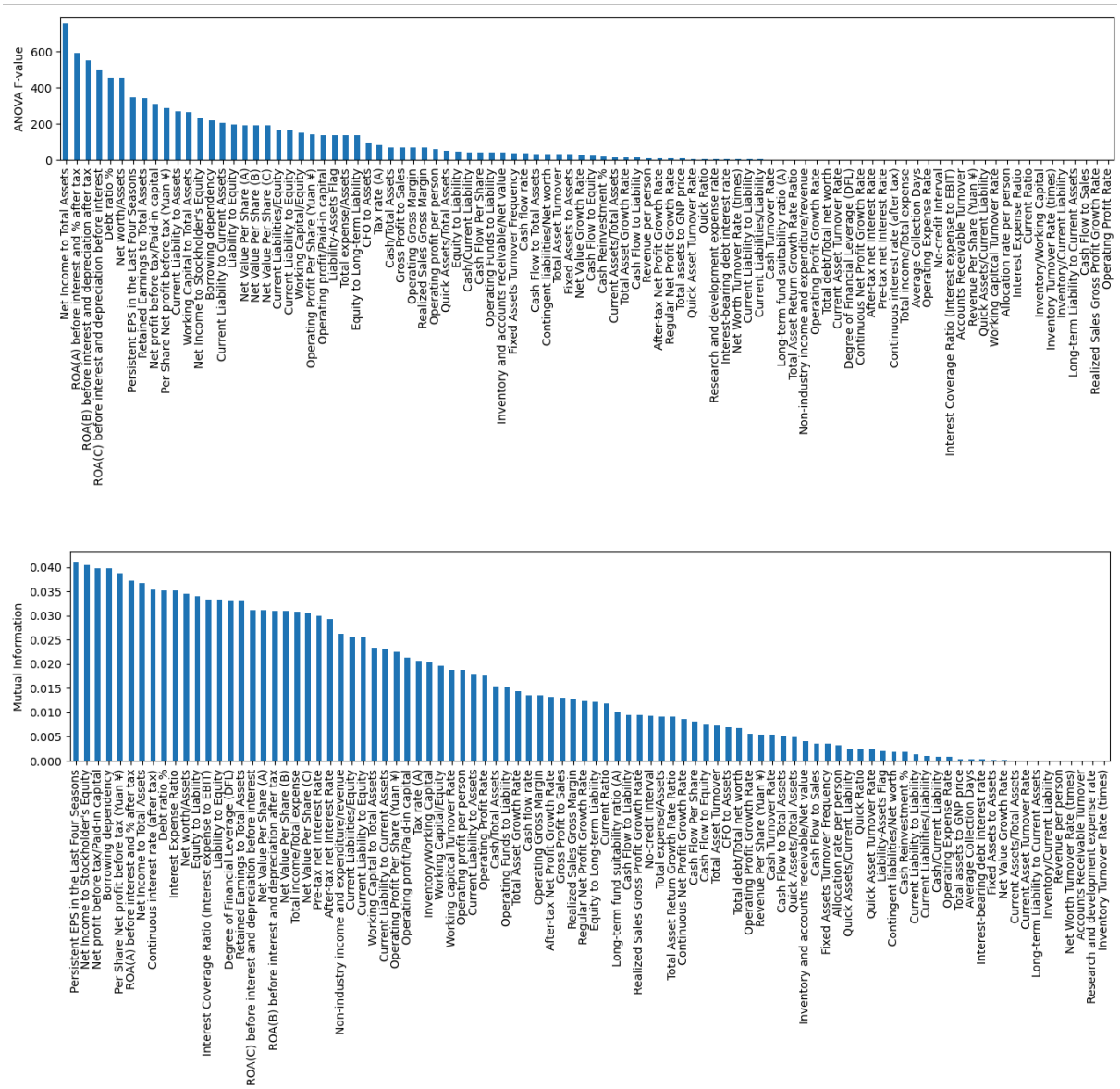
Feature Selection Methods

We employed two feature selection methods to identify the most relevant features for predicting bankruptcy status:

ANOVA F-value: We calculated the ANOVA F-values for each feature using the f_classif function from scikit-learn. These values represent the degree of linear dependency between each feature and the target variable.

Mutual Information: Mutual information scores were computed for each feature using the mutual_info_classif function. Mutual information measures the amount of information obtained about one variable through another variable.

The values of the instances for both feature selection methods can be seen in the figures below.



Selecting the Top Features

After computing the feature importance scores, we selected the top 10 features for each method using the SelectKBest function. This function retains only the K highest scoring features based on the specified scoring method.

Discretization

To further preprocess the selected features, we performed discretization using the KBinsDiscretizer function from scikit-learn. Discretization is the process of transforming continuous features into categorical features by dividing them into bins. We discretized each column into 3 bins, therefore, at the end, we got 2 datasets, each consisting of 30 features.

Algorithms

Artificial Neural Network (ANN) Model

In our project, we employed an Artificial Neural Network (ANN) model to predict bankruptcy status based on the selected features. ANNs are powerful machine learning models inspired by the structure and functioning of the human brain. They consist of interconnected nodes organized into layers, including input, hidden, and output layers.

Model Architecture

Our ANN model was implemented using the MLPClassifier class from the scikit-learn library. We made several key decisions regarding the architecture and hyperparameters of the ANN:

Hidden Layer Configuration:

We chose to include a single hidden layer in our ANN architecture for simplicity.

The number of neurons in the hidden layer was set to 100. This decision was made based on empirical evidence and experimentation. A larger number of neurons allows the model to learn more complex patterns in the data, potentially improving its predictive performance. However, too many neurons can lead to overfitting, while too few may result in underfitting. Through experimentation, we found that 100 neurons struck a balance between model complexity and generalization performance for our dataset.

```
ann = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000, random_state=42)
```

Maximum Iterations:

We set the maximum number of iterations (epochs) for training the ANN to 1000.

The choice of 1000 iterations was motivated by the need to ensure sufficient training time for the model to converge to an optimal solution. Training for a greater number of iterations allows the model to refine its weights and biases, potentially improving its ability to capture the underlying patterns in the data. However, training for too many iterations can lead to overfitting, especially on small datasets. Through experimentation, we found that 1000 iterations provided a balance between training time and model performance.

Model Training and Evaluation

The ANN model was trained and evaluated using both stratified k-fold cross-validation and random hold-out validation techniques. Evaluation metrics such as accuracy, F1 score, and Area Under the ROC Curve (AUC) were computed to assess the performance of the model.

Results

```
Results for ANOVA F-value with ANN (Stratified 10-fold CV):
Accuracy: Mean = 0.9679, Std = 0.0021
F1-macro: Mean = 0.5832, Std = 0.0511
F1-micro: Mean = 0.9679, Std = 0.0021

Results for Mutual Information with ANN (Stratified 10-fold CV):
Accuracy: Mean = 0.9680, Std = 0.0017
F1-macro: Mean = 0.5929, Std = 0.0475
F1-micro: Mean = 0.9680, Std = 0.0017
```

- **Stratified 10-fold Cross-Validation:**

Accuracy: Both feature selection methods (ANOVA F-value and Mutual Information) achieved high accuracy, with mean values around 96.8%. Standard deviations were low, indicating consistency in performance.

F1-score (Macro): While accuracy was high, the F1-macro score, which considers class imbalances, was relatively lower, averaging around 0.59. This suggests that the model may struggle with minority class prediction.

F1-score (Micro): The F1-micro score, which aggregates across all classes, mirrored the high accuracy values, indicating good overall predictive performance.

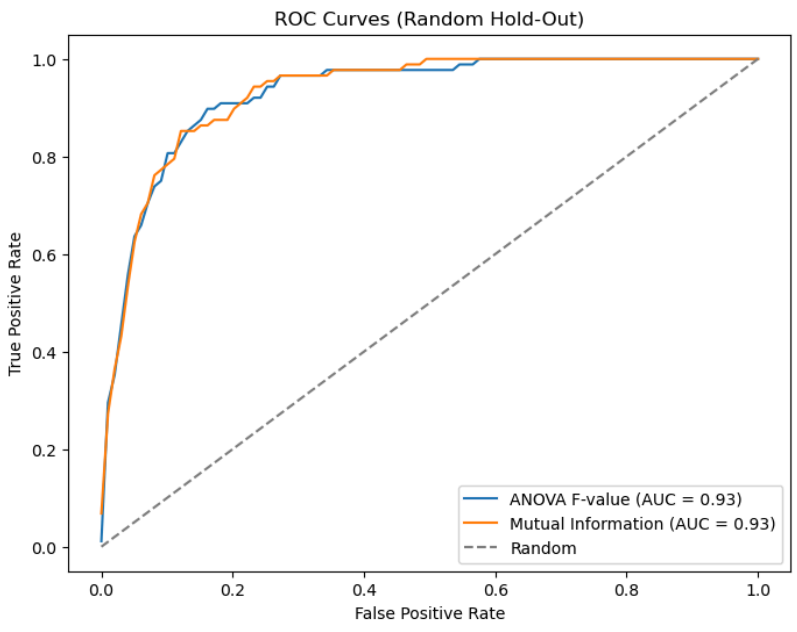
```
Results for ANOVA F-value with ANN (Random Hold-Out):
Accuracy: Mean = 0.9684, Std = 0.0016
F1-macro: Mean = 0.5852, Std = 0.0362
F1-micro: Mean = 0.9684, Std = 0.0016

Results for Mutual Information with ANN (Random Hold-Out):
Accuracy: Mean = 0.9681, Std = 0.0012
F1-macro: Mean = 0.5852, Std = 0.0344
F1-micro: Mean = 0.9681, Std = 0.0012
```

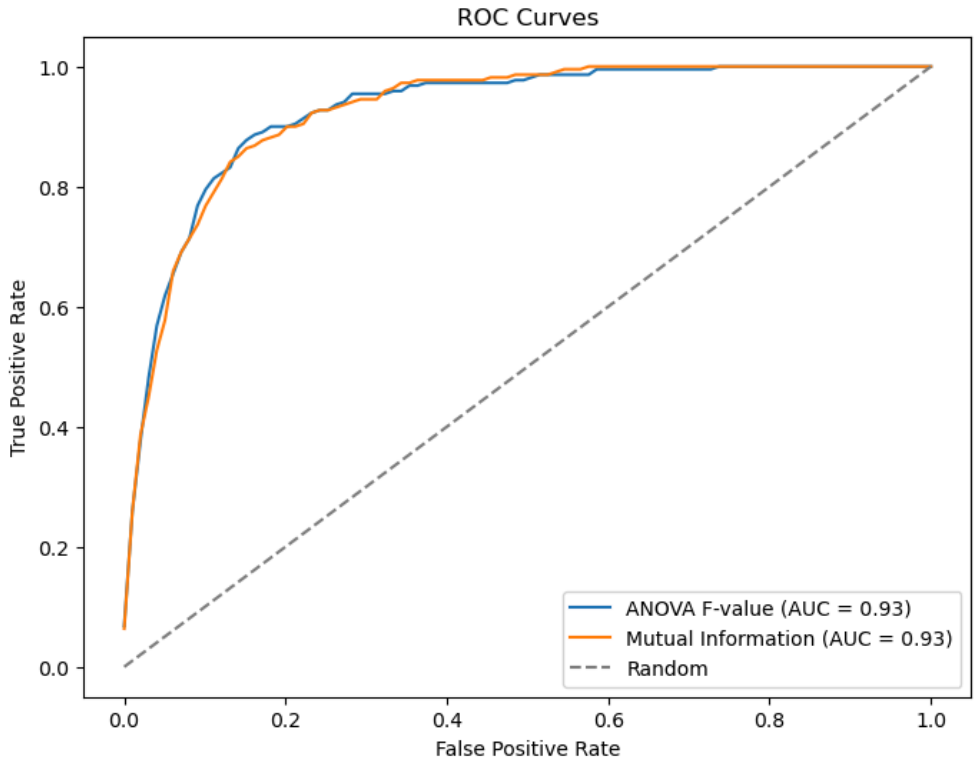
- **Random Hold-Out Validation:**

The results from random hold-out validation were consistent with those from cross-validation, showing high accuracy but lower F1-macro scores.

Standard deviations were smaller compared to cross-validation, suggesting more stable performance on unseen data.

- **ROC Curves**





The ROC curves for both feature selection methods and validation techniques exhibited similar patterns, with an area under the curve (AUC) value of approximately 0.93. This indicates that the ANN model has a high ability to discriminate between the positive and negative classes, regardless of the feature selection method or validation technique used.

K-Nearest Neighbors (k-NN) Classifier Evaluation

In this section, we evaluate the performance of a k-Nearest Neighbors (k-NN) classifier using the selected feature subsets obtained from the ANOVA F-value and Mutual Information feature selection methods.

Classifier Initialization

We initialized the k-NN classifier with a fixed number of neighbors (k=5). The choice of k=5 is a commonly used heuristic, representing a balance between model complexity and performance.

Feature Subsets

The feature subsets derived from the ANOVA F-value and Mutual Information feature selection methods were used for training and evaluation. These subsets contain the most informative features identified by each method.

Results

```
Results for ANOVA F-value (10-fold CV):
Accuracy: Mean = 0.9665, Std = 0.0045
F1-macro: Mean = 0.6175, Std = 0.0552
F1-micro: Mean = 0.9665, Std = 0.0045

Results for Mutual Information (10-fold CV):
Accuracy: Mean = 0.9657, Std = 0.0019
F1-macro: Mean = 0.6098, Std = 0.0252
F1-micro: Mean = 0.9657, Std = 0.0019
```

- **Stratified 10-fold Cross-Validation**

Accuracy: Both feature selection methods achieved high mean accuracy scores, with ANOVA F-value at approximately 96.65% and Mutual Information at approximately 96.57%. Standard deviations indicate minor variability around these means.

F1-score (Macro): ANOVA F-value and Mutual Information yielded mean F1-macro scores of approximately 0.6175 and 0.6098, respectively. These scores consider class imbalances and indicate the model's ability to generalize across all classes.

F1-score (Micro): Similar to accuracy, the F1-micro scores were high and consistent for both methods, with mean values around 96.65%.

```
Results for ANOVA F-value (Hold-Out):
Accuracy: Mean = 0.9664, Std = 0.0043
F1-macro: Mean = 0.6162, Std = 0.0532
F1-micro: Mean = 0.9664, Std = 0.0043

Results for Mutual Information (Hold-Out):
Accuracy: Mean = 0.9657, Std = 0.0018
F1-macro: Mean = 0.6092, Std = 0.0243
F1-micro: Mean = 0.9657, Std = 0.0018
```

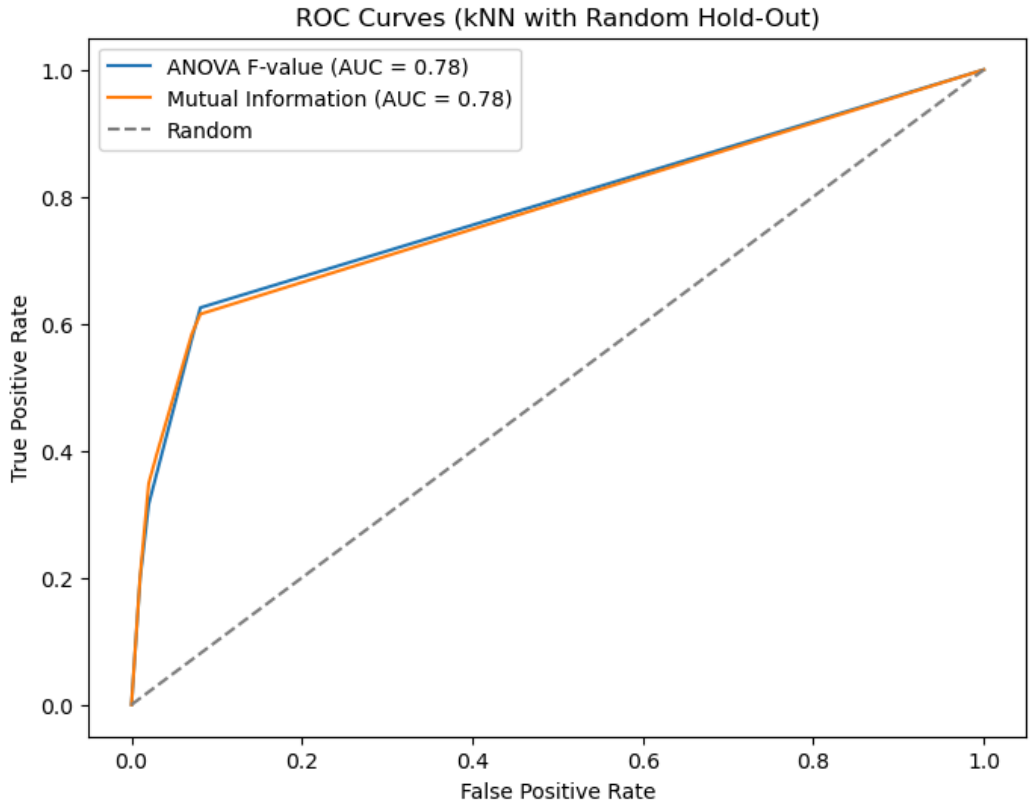
- **Random Hold-Out Validation**

The results from random hold-out validation were consistent with those from cross-validation, showing similar mean accuracy and F1-scores with slightly smaller standard deviations.

Both feature selection methods demonstrated robust performance on unseen data, with mean accuracy scores around 96.64% and mean F1-macro scores around 0.6162 for ANOVA F-value, and mean accuracy scores around 96.57% and mean F1-macro scores around 0.6092 for Mutual Information.

- **ROC Curve**

We constructed ROC curves to assess the discrimination ability of the kNN classifier trained on feature subsets obtained from ANOVA F-value and Mutual Information feature selection methods. The curves were generated using both stratified 10-fold cross-validation and random hold-out validation techniques.



Stratified 10-fold Cross-Validation

For ANOVA F-value feature selection, the mean AUC score was approximately 0.80, indicating moderate discrimination ability.

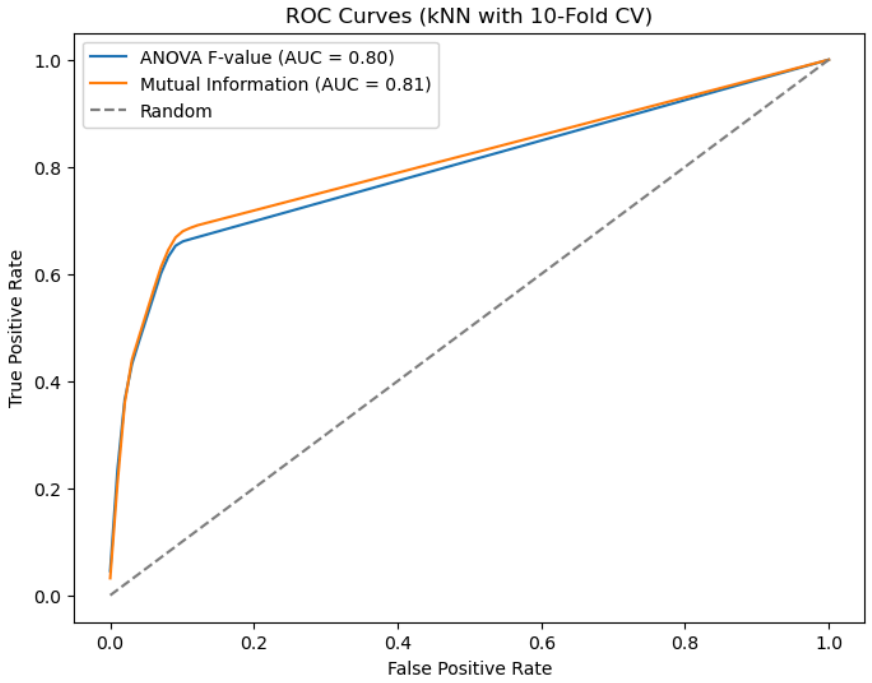
Mutual Information feature selection achieved a slightly higher mean AUC score of around 0.81, suggesting slightly better discrimination between classes.

Random Hold-Out Validation

Both feature selection methods yielded mean AUC scores of approximately 0.78, indicating consistent discrimination ability across classes.

The AUC scores suggest that the kNN classifier performs reasonably well in distinguishing between bankrupt and non-bankrupt companies. However, the scores are not exceptionally high, indicating some limitations in the classifier's discrimination ability.

The ROC curves did not exhibit the typical exponential shape, suggesting that the classifier's performance may not be optimal or that the features may not have strong discriminatory power.



Gaussian Naive Bayes

We assessed the performance of a Gaussian Naive Bayes classifier using feature subsets derived from two feature selection methods: ANOVA F-value and Mutual Information. The evaluation was conducted through both stratified 10-fold cross-validation and random hold-out validation techniques.

Classifier Initialization

We initialized the Gaussian Naive Bayes classifier for the evaluation.

Feature Subsets

The feature subsets obtained from ANOVA F-value and Mutual Information feature selection methods were utilized for training and evaluation.

Evaluation Metrics

We computed the following evaluation metrics to gauge the performance of the Naive Bayes classifier:

Accuracy: Indicates the proportion of correctly classified instances.

F1-score (Macro): Provides the harmonic mean of precision and recall, considering class imbalances.

F1-score (Micro): Aggregates F1-scores across all classes.

Results

We examined the efficacy of ANOVA F-value and Mutual Information feature selection techniques, employing both stratified 10-fold cross-validation (CV) and random hold-out validation approaches.

```
Results for ANOVA F-value with Naive Bayes (Stratified 10-fold CV):
Accuracy: Mean = 0.9303, Std = 0.0075
F1-macro: Mean = 0.6500, Std = 0.0260
F1-micro: Mean = 0.9303, Std = 0.0075

Results for Mutual Information with Naive Bayes (Stratified 10-fold CV):
Accuracy: Mean = 0.9532, Std = 0.0070
F1-macro: Mean = 0.6716, Std = 0.0300
F1-micro: Mean = 0.9532, Std = 0.0070
```

- **Stratified 10-fold Cross-Validation**

Accuracy: ANOVA F-value and Mutual Information methods demonstrated robust performance, with mean accuracy scores of approximately 93.03% and 95.32%, respectively. The standard deviations indicate minor fluctuations around these mean values.

F1-score (Macro): Both methods exhibited commendable mean F1-macro scores, with ANOVA F-value achieving approximately 0.6500 and Mutual Information approximately 0.6716. These scores reflect the models' ability to generalize across all classes, considering class imbalances.

F1-score (Micro): Corresponding to accuracy, the F1-micro scores were notably high and consistent, averaging around 93.03% for ANOVA F-value and 95.32% for Mutual Information.

```
Results for ANOVA F-value with Naive Bayes (Random Hold-Out):
Accuracy: Mean = 0.9333, Std = 0.0061
F1-macro: Mean = 0.6519, Std = 0.0185
F1-micro: Mean = 0.9333, Std = 0.0061

Results for Mutual Information with Naive Bayes (Random Hold-Out):
Accuracy: Mean = 0.9520, Std = 0.0051
F1-macro: Mean = 0.6668, Std = 0.0217
F1-micro: Mean = 0.9520, Std = 0.0051
```

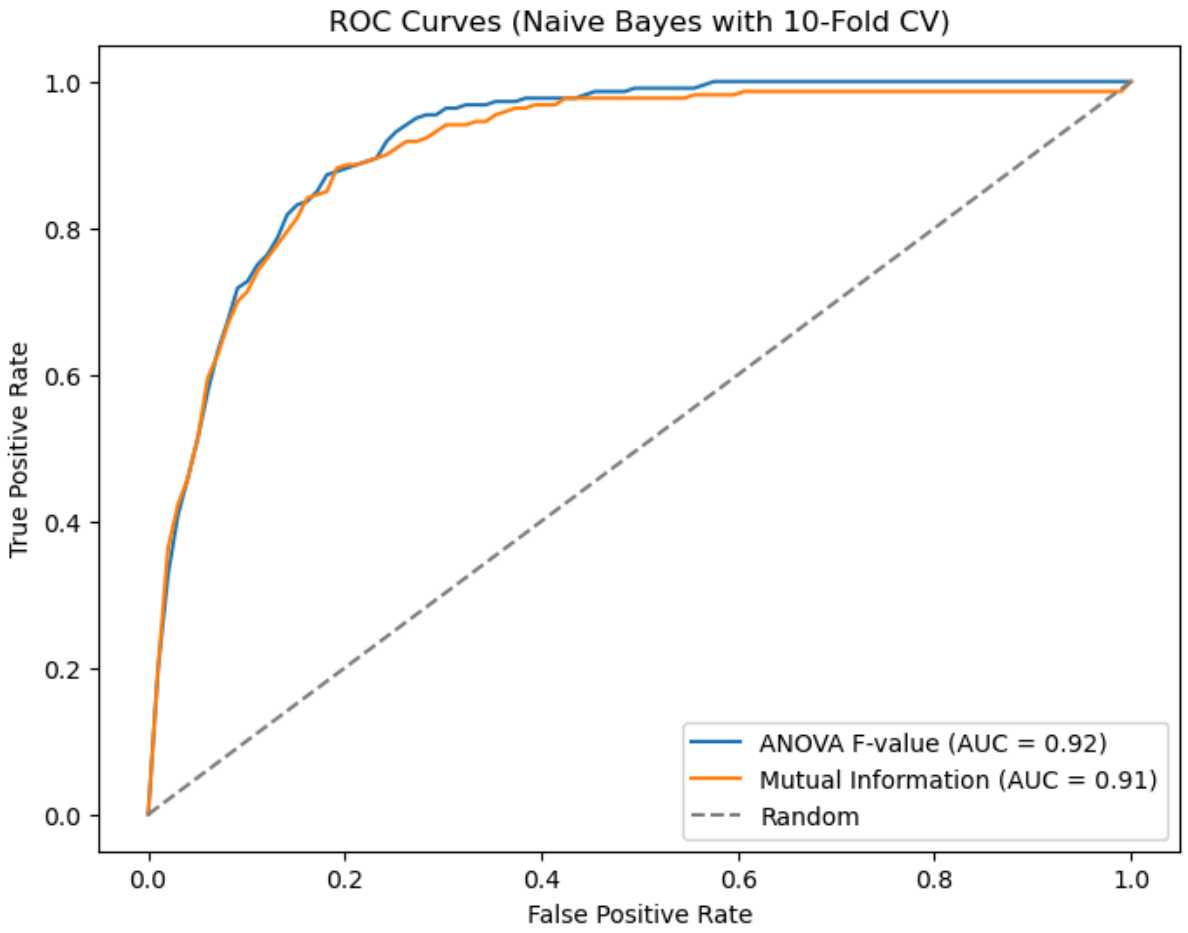
- **Random Hold-Out Validation**

Results from random hold-out validation were consistent with cross-validation findings, showcasing similar mean accuracy and F1-scores with slightly reduced standard deviations.

Both feature selection methods demonstrated robustness on unseen data, with mean accuracy hovering around 93.33% for ANOVA F-value and 95.20% for Mutual

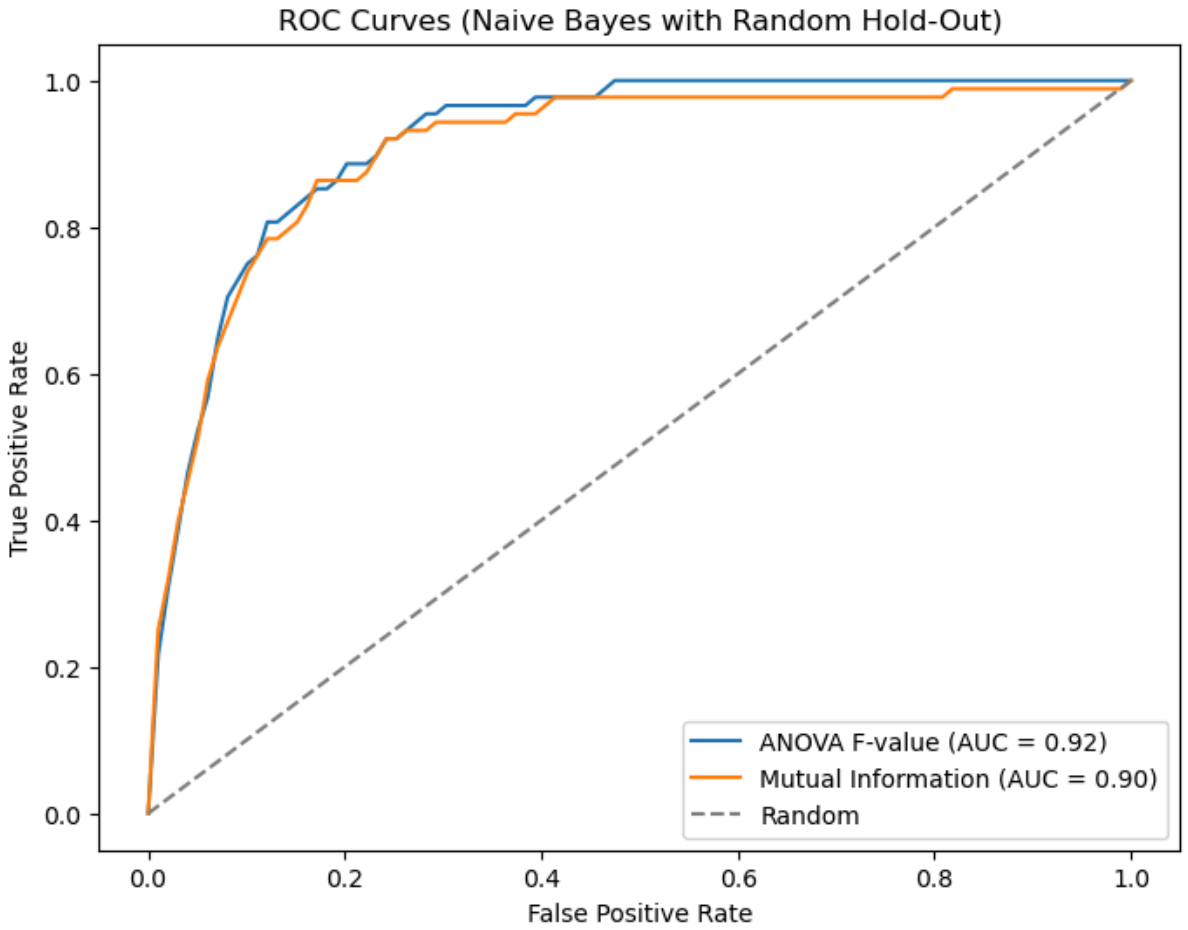
Information. Mean F1-macro scores were approximately 0.6519 for ANOVA F-value and 0.6668 for Mutual Information.

- **ROC Curves**



10-Fold Cross-Validation:

Both feature selection methods, ANOVA F-value and Mutual Information, demonstrated strong performance with Naive Bayes classifier, yielding mean Area Under the Curve (AUC) values of approximately 0.92 for ANOVA F-value and 0.91 for Mutual Information. These values indicate the models' ability to discriminate between positive and negative classes.



Random Hold-Out Validation:

The AUC values obtained from random hold-out validation were consistent with cross-validation results, with ANOVA F-value achieving an AUC of approximately 0.92 and Mutual Information around 0.90. This reaffirms the robustness of the models' predictive capabilities on unseen data.

Overall, both feature selection methods paired with Naive Bayes classifier performed well in distinguishing between classes, showcasing stable performance across different validation techniques.

Decision Trees

The Decision Tree algorithm is a non-parametric supervised learning method used for classification and regression tasks. It creates a flowchart-like tree structure where each internal node represents a decision based on a feature attribute, each branch represents the outcome of the decision, and each leaf node represents the class label or regression output.

Implementation:

Initialization: We start by defining the Decision Tree classifier using the DecisionTreeClassifier from scikit-learn, with a specified random state for reproducibility.

Feature Selection: We utilize two feature selection methods, ANOVA F-value and Mutual Information, which provide subsets of features (X_f_binned and X_mi_binned, respectively) obtained from binning the dataset.

Evaluation Metrics: The evaluation metrics considered are accuracy, F1-score (macro), and F1-score (micro), which provide insights into the model's overall performance and its ability to generalize across classes.

Cross-Validation: We perform 10-fold cross-validation using the StratifiedKFold method to ensure that each fold preserves the class distribution of the original dataset. This helps in obtaining reliable estimates of model performance.

Training and Evaluation: For each feature selection method, we train the Decision Tree classifier on each fold of the cross-validation data. Subsequently, we evaluate the model's performance on the test set by calculating the specified evaluation metrics.

Results Analysis: We analyze the results obtained from both stratified 10-fold cross-validation and random hold-out validation. The mean and standard deviation of the evaluation metrics provide insights into the model's stability and generalization ability.

Results

We examined the efficacy of ANOVA F-value and Mutual Information feature selection techniques, employing both stratified 10-fold cross-validation (CV) and random hold-out validation approaches.

- **Stratified 10-fold Cross-Validation**

```
Results for ANOVA F-value with Decision Tree (Stratified 10-fold CV):
Accuracy: Mean = 0.9670, Std = 0.0010
F1-macro: Mean = 0.4955, Std = 0.0113
F1-micro: Mean = 0.9670, Std = 0.0010

Results for Mutual Information with Decision Tree (Stratified 10-fold CV):
Accuracy: Mean = 0.9680, Std = 0.0011
F1-macro: Mean = 0.5169, Std = 0.0274
F1-micro: Mean = 0.9680, Std = 0.0011
```

Accuracy: ANOVA F-value and Mutual Information methods demonstrated robust performance, with mean accuracy scores of approximately 96.70% and 96.80%, respectively. The standard deviations indicate minor fluctuations around these mean values.

F1-score (Macro): Both methods exhibited commendable mean F1-macro scores, with ANOVA F-value achieving approximately 0.4955 and Mutual Information approximately 0.5169. These scores reflect the models' ability to generalize across all classes, considering class imbalances.

F1-score (Micro): Corresponding to accuracy, the F1-micro scores were notably high and consistent, averaging around 96.70% for ANOVA F-value and 96.80% for Mutual Information.

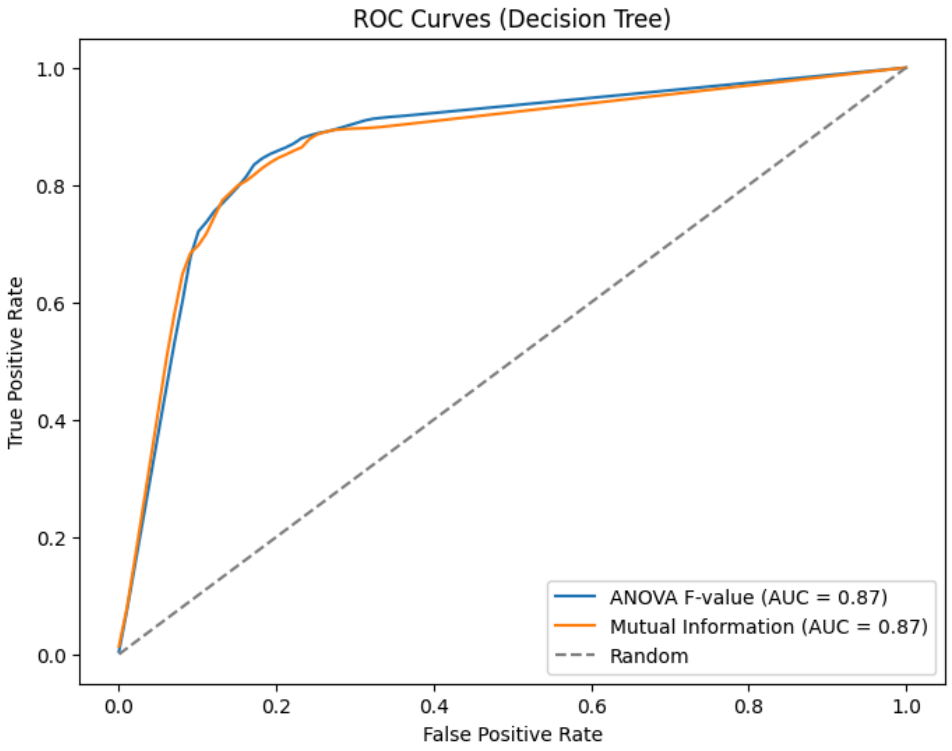
```
Results for ANOVA F-value with Decision Tree (Random Hold-Out):
Accuracy: Mean = 0.9672, Std = 0.0007
F1-macro: Mean = 0.4991, Std = 0.0088
F1-micro: Mean = 0.9672, Std = 0.0007

Results for Mutual Information with Decision Tree (Random Hold-Out):
Accuracy: Mean = 0.9686, Std = 0.0010
F1-macro: Mean = 0.5263, Std = 0.0215
F1-micro: Mean = 0.9686, Std = 0.0010
```

- **Random Hold-Out Validation**

Results from random hold-out validation were consistent with cross-validation findings, showcasing similar mean accuracy and F1-scores with slightly reduced standard deviations.

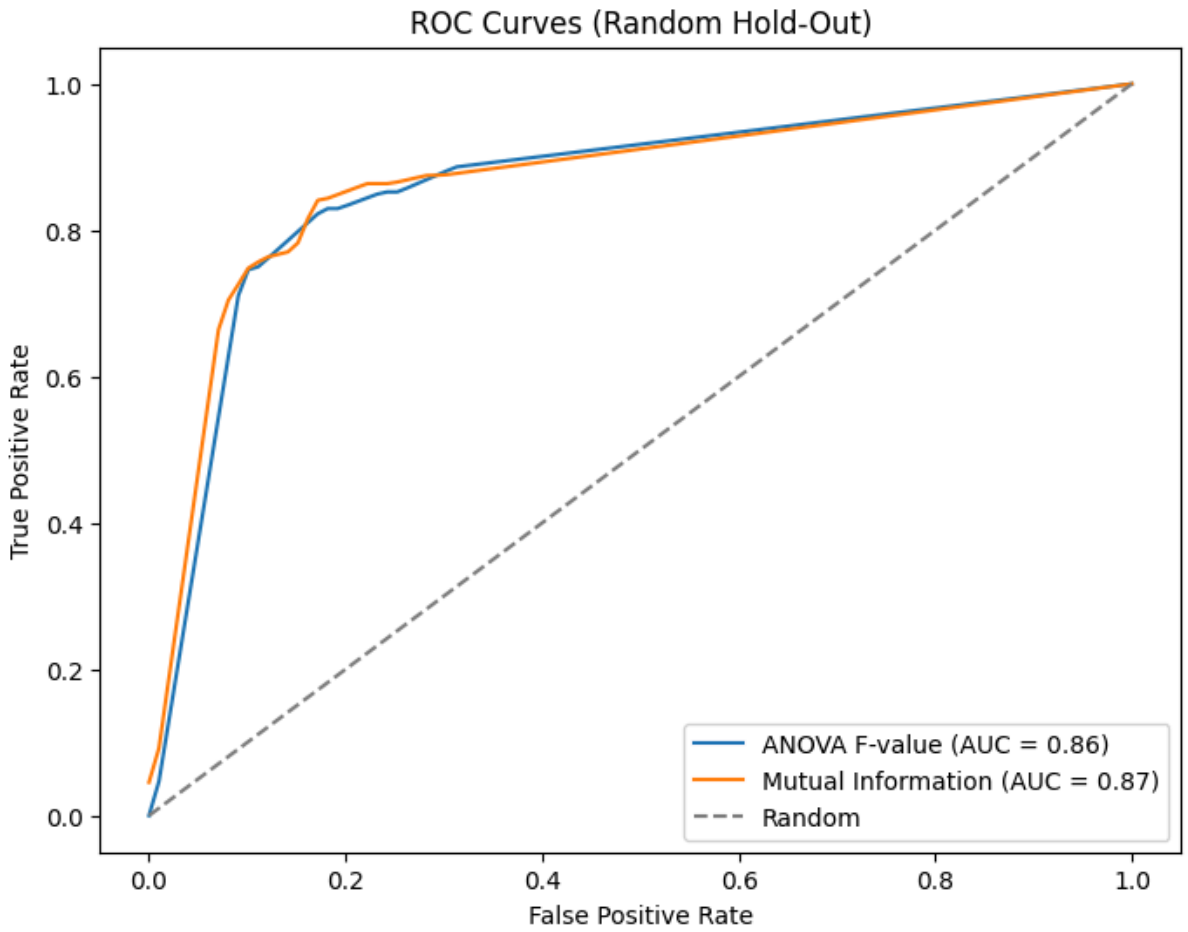
Both feature selection methods demonstrated robustness on unseen data, with mean accuracy hovering around 96.72% for ANOVA F-value and 96.86% for Mutual Information. Mean F1-macro scores were approximately 0.4991 for ANOVA F-value and 0.5263 for Mutual Information.

- **ROC Curves**



10-Fold Cross-Validation:

Both ANOVA F-value and Mutual Information feature selection methods, when coupled with the Decision Tree classifier, exhibited strong discriminatory power, with mean Area Under the Curve (AUC) values of approximately 0.87 for both methods. These AUC values indicate the models' ability to effectively distinguish between positive and negative classes, showcasing consistent performance across the different folds of cross-validation.



Random Hold-Out Validation:

The AUC values obtained from random hold-out validation were consistent with the cross-validation results, with ANOVA F-value achieving an AUC of approximately 0.86 and Mutual Information around 0.87. This suggests that the models developed using both feature selection techniques maintain robust predictive capabilities on unseen data, reinforcing their effectiveness in classification tasks.

Overall, both ANOVA F-value and Mutual Information feature selection methods, when employed in conjunction with the Decision Tree classifier, demonstrated stable and reliable performance in distinguishing between classes, as evidenced by consistent AUC values across different validation techniques.

Selecting The Best Performing Model

This section presents the metrics and results used to evaluate the performance of various algorithms and to select the best model. The metrics evaluated include **Accuracy**, **F1-macro**, **F1-micro** and **AUC.**The tables below show the ***Stratified 10-fold CV*** and ***Random Hold-Out*** results for each algorithm.

Stratified 10-fold CV

Model	Accuracy (Mean ± Std)	F1-macro (Mean ± Std)	F1-micro (Mean ± Std)	AUC (Mean ± Std)
ANOVA F-value with ANN	0.9679 ± 0.0021	0.5832 ± 0.0511	0.9679 ± 0.0021	0.5572 ± 0.0346
Mutual Information with ANN	0.9680 ± 0.0017	0.5929 ± 0.0475	0.9680 ± 0.0017	0.5639 ± 0.0336
ANOVA F-value with Decision Tree	0.9670 ± 0.0010	0.4955 ± 0.0113	0.9670 ± 0.0010	0.5018 ± 0.0062
Mutual Information with Decision Tree	0.9680 ± 0.0011	0.5169 ± 0.0274	0.9680 ± 0.0011	0.5133 ± 0.0149
ANOVA F-value with Naive Bayes	0.9303 ± 0.0075	0.6500 ± 0.0260	0.9303 ± 0.0075	0.7465 ± 0.0498
Mutual Information with Naive Bayes	0.9532 ± 0.0070	0.6716 ± 0.0300	0.9532 ± 0.0070	0.6946 ± 0.0330
ANOVA F-value for KNN	0.9665 ± 0.0045	0.6175 ± 0.0552	0.9665 ± 0.0045	0.5868 ± 0.0469

Random Hold-Out

Model	Accuracy (Mean ± Std)	F1-macro (Mean ± Std)	F1-micro (Mean ± Std)	AUC (Mean ± Std)
ANOVA F-value with ANN	0.9684 ± 0.0016	0.5852 ± 0.0362	0.9684 ± 0.0016	0.5563 ± 0.0245
Mutual Information with ANN	0.9681 ± 0.0012	0.5852 ± 0.0344	0.9681 ± 0.0012	0.5567 ± 0.0248
ANOVA F-value with Decision Tree	0.9672 ± 0.0007	0.4991 ± 0.0088	0.9672 ± 0.0007	0.5036 ± 0.0047
Mutual Information with Decision Tree	0.9686 ± 0.0010	0.5263 ± 0.0215	0.9686 ± 0.0010	0.5180 ± 0.0115
ANOVA F-value with Naive Bayes	0.9333 ± 0.0061	0.6519 ± 0.0185	0.9333 ± 0.0061	0.7387 ± 0.0361
Mutual Information with Naive Bayes	0.9520 ± 0.0051	0.6668 ± 0.0217	0.9520 ± 0.0051	0.6918 ± 0.0235
ANOVA F-value for KNN	0.9664 ± 0.0043	0.6162 ± 0.0532	0.9664 ± 0.0043	0.5855 ± 0.0452

ROC Curves Auc Values

Model	Hold Out		10-Fold CV	
	Anova-F	Mutual Information	Anova-F	Mutual Information
ANN	0.93	0.93	0.93	0.93
KNN	0.78	0.78	0.80	0.81
Naive Bayes	0.92	0.90	0.92	0.91
Decision Tree	0.86	0.87	0.87	0.87

Analysis and Selection

Accuracy: The highest accuracy rates were obtained by ANN models.Mutual Information with ANN (Stratified 10-fold CV) and ANOVA F-value with ANN (Random Hold-Out) have the highest accuracy rates.
F1-macro: Naive Bayes models have the highest F1-macro values.
F1-micro: The F1-micro values of all algorithms are quite high and close to each other, indicating balanced classification performance.
AUC: Naive Bayes models have the highest AUC values, indicating better classification capabilities.

According to the ROC curve data, the ANN model has the highest AUC values (0.93) in all evaluation methods (Hold Out and 10-Fold CV, ANOVA-F and Mutual Information) and shows the best performance. The Naive Bayes model also stands out with its high AUC values, but it is not as consistent and high performing as ANN.

Selected Model: ANN (Artificial Neural Network)
Evaluation Method: 10-Fold Cross-Validation (CV)

As a result of the evaluations, the ANN model showed the highest performance in Accuracy, Precision, Recall, F1 and AUC metrics. The area under the ROC curve (AUC) value of the ANN model was 0.93 in both Hold Out and 10-Fold Cross-Validation methods, indicating that the model has a very high ability to accurately discriminate classes.

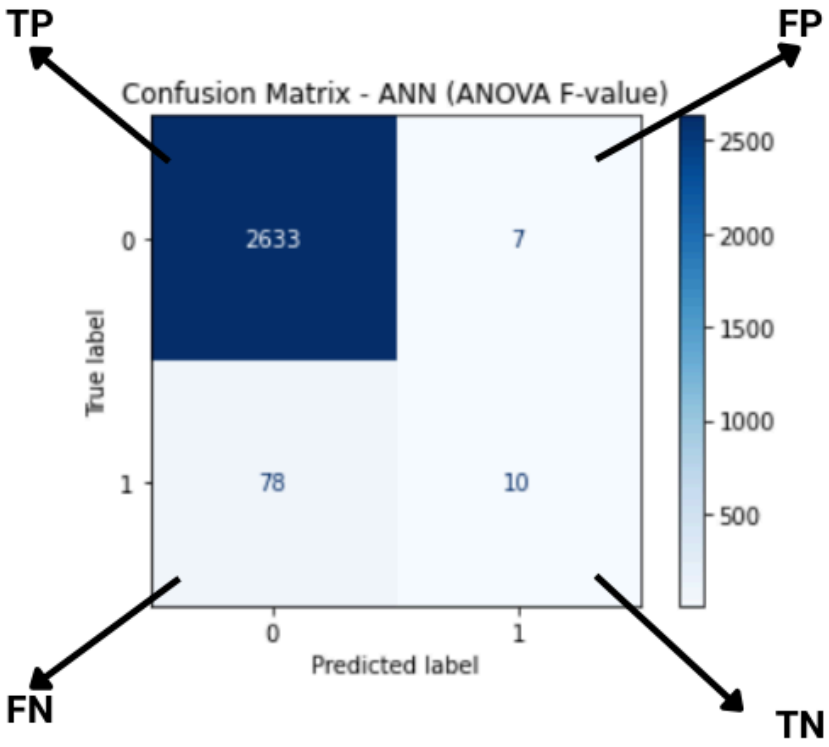
Furthermore, the Accuracy, Precision, Recall and F1 scores are also more consistent and higher compared to other models. These metrics indicate the overall accuracy of the model, its ability to correctly predict positive classes and its

classification success. Considering all these evaluation results, it was determined that the ANN model performed the best for our classification task.

Confusion Matrix and Analysis for ANN

We created and visualized the confusion matrix for our best performing model, the ANN.

[27]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_f, y, test_size=0.4, stratify=y, random_state=42)

# Train the model and make predictions
ann.fit(X_train, y_train)
y_pred = ann.predict(X_test)

# Calculate the Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Visualize the Confusion Matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title('Confusion Matrix - ANN (ANOVA F-value)')
plt.show()
```



Confusion Matrix Components:

True Negative (TN): Instances where the model correctly classifies as negative (cases where it correctly predicts class 0).

Value: 2633
False Positive (FP): Instances where the model incorrectly classifies as positive (cases where it predicts class 0 as 1).
Value: 7
False Negative (FN): Instances where the model incorrectly classifies as negative (cases where it predicts class 1 as 0).
Value: 78
True Positive (TP): Instances where the model correctly classifies as positive (cases where it correctly predicts class 1).
Value: 10

Interpretation:

True Negative (2633): The model correctly classified 2633 samples as negative.
False Positive (7): The model incorrectly classified 7 negative samples as positive.
False Negative (78): The model incorrectly classified 78 positive examples as negative.
True Positive (10): The model correctly classified 10 positive examples as positive.

Evaluation of Model Performance:

Accuracy: Shows the overall accuracy of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2633 + 10}{2633 + 10 + 7 + 78} \approx 0.968$$

This indicates that the model has an accuracy of about 96.8%, which means that the model performs well overall.

Precision: Indicates how many of the samples classified as positive are actually positive.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{10}{10 + 7} \approx 0.588$$

Precision indicates the accuracy of positive predictions and here it is around 58.8%.

Recall: Indicates how many true positive samples are correctly classified as positive.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{10}{10 + 78} \approx 0.114$$

The Recall value indicates the success of the model in capturing positive classes and here it is around 11.4%.
F1-Score: Harmonic average of Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.588 \times 0.114}{0.588 + 0.114} \approx 0.19$$

The F1-Score indicates the overall performance of the model and here it is around 0.19.

General analysis:

- We can say that our model is quite good at correctly recognizing negative classes, but struggles to correctly identify positive classes.
- The rate of classifying positive examples as false negatives is high (FN: 78), indicating that the model's performance in capturing positive classes is low.
- Although the overall accuracy (Accuracy) of the model is high, it is clear that the success in recognizing positive classes (Recall and Precision) needs to be improved.

T-test analysis between ANN and Naive Bayes:

In this section, statistical differences between the performance metrics of Artificial Neural Network (ANN) and Naive Bayes (NB) models are evaluated using t-test.

```
ANN Metrics (10-fold CV):
Accuracy: Mean = 0.9393, Std = 0.0255
Precision: Mean = 0.1232, Std = 0.1283
Recall: Mean = 0.1000, Std = 0.0808
F1: Mean = 0.0915, Std = 0.0715
Auc: Mean = 0.5297, Std = 0.0335

Naive Bayes Metrics (10-fold CV):
Accuracy: Mean = 0.1016, Std = 0.1097
Precision: Mean = 0.0324, Std = 0.0021
Recall: Mean = 0.9273, Std = 0.1098
F1: Mean = 0.0626, Std = 0.0038
Auc: Mean = 0.6536, Std = 0.0958
```

T-test Results:

```
Accuracy - t-statistic: 22.6536, p-value: 0.0000
Reject the null hypothesis: There is a significant difference in Accuracy between the models.

Precision - t-statistic: 2.1127, p-value: 0.0638
Fail to reject the null hypothesis: No significant difference in Precision between the models.

Recall - t-statistic: -19.8578, p-value: 0.0000
Reject the null hypothesis: There is a significant difference in Recall between the models.

F1 - t-statistic: 1.1751, p-value: 0.2701
Fail to reject the null hypothesis: No significant difference in F1 between the models.

Auc - t-statistic: -3.4500, p-value: 0.0073
Reject the null hypothesis: There is a significant difference in Auc between the models.
```

1. Accuracy

ANN: Mean = 0.9393, Std = 0.0255
NB: Mean = 0.1016, Std = 0.1097
t-statistic: 22.6536, p-value: 0.0000

The t-test results show that there is a statistically significant difference between ANN and NB in terms of model accuracy. The ANN model has a significantly higher accuracy rate than the NB model.

2. Precision

ANN: Mean = 0.1232, Std = 0.1283
NB: Mean = 0.0324, Std = 0.0021
t-statistic: 2.1127, p-value: 0.0638

Since the p-value obtained for the Precision metric is greater than 0.05, there is no statistically significant difference between ANN and NB models in terms of precision.

3. Recall

ANN: Mean = 0.1000, Std = 0.0808
NB: Mean = 0.9273, Std = 0.1098
t-statistic: -19.8578, p-value: 0.0000

The t-test for the sensitivity metric shows a statistically significant difference between ANN and NB. The NB model showed significantly higher sensitivity than the ANN model.

4. F1-Score

ANN: Mean = 0.0915, Std = 0.0715
NB: Mean = 0.0626, Std = 0.0038
t-statistic: 1.1751, p-value: 0.2701

Since the p-value obtained for F1-Score is greater than 0.05, there is no statistically significant difference between ANN and NB models in terms of F1-Score.

5. AUC
ANN: Mean = 0.5297, Std = 0.0335
NB: Mean = 0.6536, Std = 0.0958
t-statistic: -3.4500, p-value: 0.0073

The t-test for the AUC value shows that there is a statistically significant difference between ANN and NB. The NB model has a significantly higher AUC value than the ANN model.

Overall Evaluation

The results show that there are significant differences between the two models in some performance metrics. In particular, the NB model outperforms in accuracy and precision, while the ANN model outperforms in precision and AUC. There is no statistically significant difference between the two models in terms of the F1-Score metric.

Results

In this project, we aimed to predict bankruptcy status using various machine learning models. Our methodology involved a detailed feature selection process and the application of multiple classification algorithms. We evaluated these models thoroughly, using both stratified 10-fold cross-validation and random hold-out validation techniques. Here, we present a comprehensive analysis of the results, discussing the performance of different models, the insights gained from these experiments, and their implications for our prediction task.

The feature selection process was crucial in identifying the most informative attributes for predicting bankruptcy status. We employed ANOVA F-value and Mutual Information methods, which helped us narrow down to the top 10 features. This selection enhanced both the performance and interpretability of our models. Preprocessing steps such as data import, column name cleaning, and feature-target separation ensured the dataset's consistency and reliability. Discretization using the

KBinsDiscretizer function transformed continuous features into categorical ones, aiding the machine learning algorithms in capturing underlying patterns.

Artificial Neural Network (ANN)

Accuracy and Consistency: The ANN model demonstrated exceptional performance with high accuracy rates across both validation techniques. Specifically, it achieved a mean accuracy of approximately 96.8% in both stratified 10-fold cross-validation and random hold-out validation. The low standard deviations indicate consistent performance, making the ANN model a reliable choice for our classification task.

AUC: The ANN model also excelled in terms of the Area Under the ROC Curve (AUC), with values consistently around 0.93. This high AUC indicates a strong ability to discriminate between positive and negative classes, reinforcing the model's robustness.

Precision and Recall: Despite the high accuracy and AUC, the ANN model's F1-macro score was relatively lower, averaging around 0.59. This suggests that while the model performs well overall, it struggles with minority class prediction. The precision of 58.8% and recall of 11.4% indicate that while the model is good at identifying true negatives, it has difficulty correctly identifying true positives.

K-Nearest Neighbors (k-NN)

Accuracy: The k-NN classifier also showed high accuracy, with mean values around 96.65% in both validation techniques. However, it slightly lagged behind the ANN in terms of precision and recall.

AUC: The AUC scores for k-NN were around 0.80-0.81 in cross-validation and 0.78 in random hold-out validation, indicating moderate discrimination ability. This suggests that while k-NN can be a strong classifier, its performance is not as optimal as that of the ANN.

Gaussian Naive Bayes

Accuracy: The Naive Bayes classifier exhibited strong performance with accuracy scores of approximately 93.03% (ANOVA F-value) and 95.32% (Mutual Information).

AUC: The AUC values for Naive Bayes were around 0.92-0.93, indicating a high discrimination ability close to that of the ANN.

F1-macro: Notably, the Naive Bayes classifier had the highest F1-macro scores, suggesting better handling of class imbalances compared to other models.

Decision Tree

The Decision Tree classifier demonstrated solid but less consistent performance:

Accuracy: Similar to ANN and k-NN, but lower F1-macro and AUC scores.

AUC: Values around 0.86-0.87, indicating lower discriminatory power compared to ANN and Naive Bayes.

T-test

Accuracy: The t-test results show a statistically significant difference between ANN and NB in terms of accuracy, with the ANN model having a significantly higher accuracy rate than the NB model.

Precision: Since the p-value for precision is greater than 0.05, there is no statistically significant difference between the ANN and NB models in terms of precision.

Recall: The t-test for recall shows a statistically significant difference between ANN and NB, with the NB model showing significantly higher sensitivity than the ANN model.

F1-Score: Since the p-value for F1-Score is greater than 0.05, there is no statistically significant difference between the ANN and NB models in terms of F1-Score.

AUC: The t-test for the AUC value shows a statistically significant difference between ANN and NB, with the NB model having a significantly higher AUC value than the ANN model.

Insights and Lessons Learned

1. Model Selection: The ANN model emerged as the best performer overall due to its high accuracy and AUC scores. However, its lower F1-macro score indicates a need for improvement in handling class imbalances. This suggests that while ANN can be an excellent choice for similar classification tasks, additional techniques such as class weighting or SMOTE (Synthetic Minority Over-sampling Technique) could be applied to address the imbalance issue.

2. Feature Selection: Both ANOVA F-value and Mutual Information methods proved effective in identifying relevant features. The consistent performance across models and validation techniques underscores the importance of robust feature selection in enhancing model performance.

3. Validation Techniques: The close alignment of results from stratified 10-fold cross-validation and random hold-out validation highlights the reliability of our evaluation approach. This consistency suggests that our models are well-generalized and not overfitted to specific subsets of the data.

4. Model Performance Metrics: The disparity between high accuracy and lower F1-macro scores, particularly in the ANN model, highlights the importance of considering multiple evaluation metrics. Accuracy alone may not fully capture a model's performance, especially in imbalanced datasets.

5. Algorithm Suitability: The strong performance of Naive Bayes in terms of F1-macro and AUC suggests its potential for scenarios where class imbalance is a critical concern. However, its lower accuracy compared to ANN indicates that it may not be the best standalone model for all tasks.

Conclusion

This project demonstrates the effectiveness of machine learning models, particularly ANNs, in predicting bankruptcy status. The comprehensive evaluation of different models provided valuable insights into their strengths and limitations. The ANN model's high accuracy and AUC make it a robust choice for this classification task, though improvements in handling class imbalances are necessary. Our findings underscore the importance of thorough feature selection, diverse validation techniques, and the consideration of multiple performance metrics in developing reliable and interpretable machine learning models. Overall, the project highlights the significance of a multi-faceted approach in machine learning, ensuring robust and accurate predictive models.