Jay Thanki & Rohan Patel - CSE 587

Reducing Telecom Customer Churn
A Predictive Analytics Approach

Problem Statement:

Customer churn is an enormous challenge for telecom firms as consumers switch to competitors or end their contracts early. The task is to create predictive models capable of identifying and comprehending the aspects contributing to client attrition. This project will address the following questions:

1. What are the primary factors influencing telecom customer churn?

2. Can we create an accurate predictive model to anticipate which customers would churn?

3. How may this information be used to build targeted retention tactics and lower client churn?

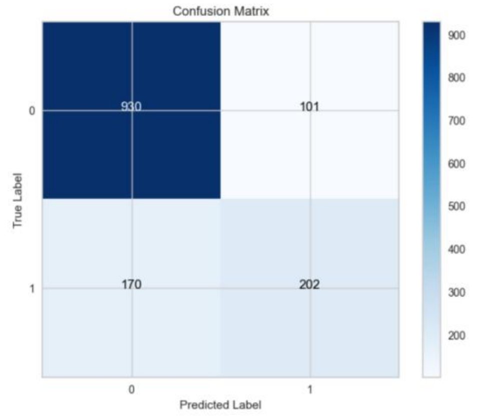
Data Modelling:

We experimented with the following six models:

1. Logistic Regression
2. Support Vector Classifier
3. Random Forest
4. Bagging Classifier
5. Gaussian Naive Bayes
6. Neural Network

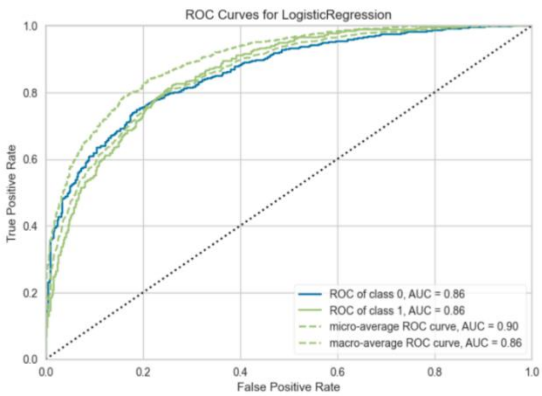
Dataset: https://www.kaggle.com/datasets/blastchar/telco-customer-churn?resource=download

1. Logistic Regression:



Predicted Label

	precision	recall	f1-score	support
0	0.85	0.90	0.87	1031
1	0.67	0.54	0.60	372
accuracy			0.81	1403
macro avg	0.76	0.72	0.74	1403
weighted avg	0.80	0.81	0.80	1403



Algorithm Choice:

The use of a baseline Logistic Regression model was determined by the nature of the issue statement and the dataset's features. Logistic Regression is a popular approach for binary classification applications, and it was chosen for the following reasons:

> **1. Nature of the Problem**: The challenge includes binary classification to predict whether an instance belongs to class 0 or 1. Logistic Regression is ideal for such jobs since it models the likelihood of an instance belonging to a specific class.
>
> **2. Linearity:** Logistic Regression implies a linear connection between features and outcome log-odds. This linearity can frequently represent underlying data trends.
>
> **3. Interpretability**: Because the coefficients may be utilized to evaluate the influence of each attribute on the outcome, Logistic Regression produces interpretable findings. This is useful for drawing conclusions from the model's predictions.

Model Training and Tuning:

The following procedures were done to construct the baseline Logistic Regression model:

1. Data Preprocessing: To manage missing values, encode category variables, and scale numerical characteristics, the dataset was preprocessed. This ensured the data was prepared for use in the Logistic Regression model.

2. Model Training: A simple Logistic Regression model was trained using the default hyperparameters as a starting point. This enabled us to define a baseline level of performance.

Model Effectiveness:

The report includes many measures for assessing the efficacy of the baseline Logistic Regression model:

1. Accuracy: With an accuracy of 0.81, the model correctly predicts the class label for 81% of the cases in the dataset.

2. Precision and Recall: Precision and recall are necessary for skewed datasets. The accuracy (0.85) for class 0 indicates that when the model predicts class 0, it is right 85% of the time. Class 0 has a high recall (0.90), indicating that the model accurately recognizes true positives. However, accuracy (0.67) and recall (0.54) are lower for class 1, showing that the model is less reliable and requires assistance recognizing true positives for class 1.

3. F1-Score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.87, whereas it is 0.60 for class 1. This demonstrates that the model outperforms the class 1 model substantially.

4. Macro Avg and Weighted Avg are the precision, recall, and F1-score averages. The macro average gives equal weight to both classes, but the weighted average considers class imbalance. The macro average F1-score (0.74) and the weighted average F1-score (0.80) provide an overall evaluation of model performance.

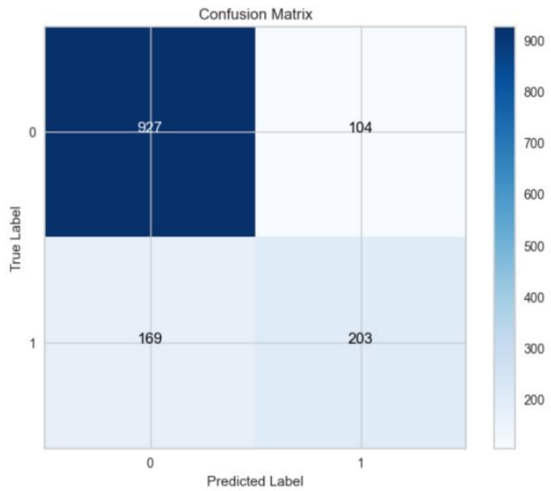
5. ROC-AUC: The ROC-AUC statistic (Receiver Operating Characteristic - Area Under the Curve) assesses the model's ability to differentiate across classes. A ROC-AUC of 0.86 indicates good discriminatory power.

6. Average Precision: The average precision parameter (0.69) reveals the model's accuracy-recall trade-off. A more excellent score shows better precision when recall is considered, implying moderate effectiveness in categorizing class 1 cases.

Intelligence Gained:

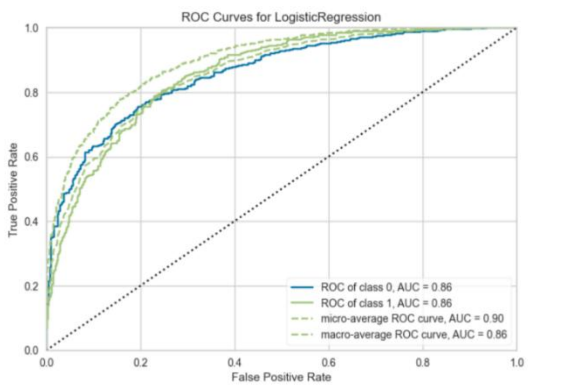
Class 0 is efficiently predicted by the baseline Logistic Regression model, as seen by good accuracy and recall scores for this class. However, the model's performance in class 1 might be more robust with lower accuracy and recall parameters. This disparity implies that resolving class imbalance may be required to enhance the model's performance for class 1.

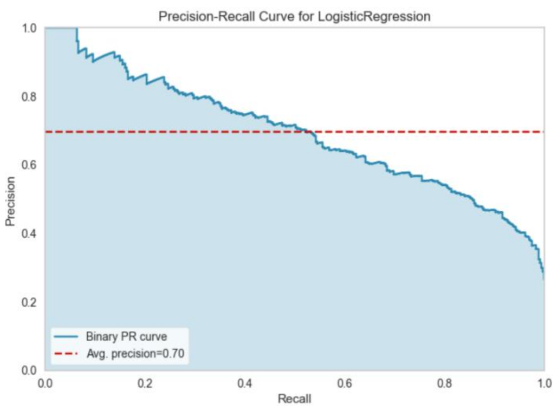
Hyperparameter optimization and Regularization on Logistic Regression



Predicted Label

	precision	recall	f1-score	support
0	0.85	0.90	0.87	1031
1	0.66	0.55	0.60	372
accuracy			0.81	1403
macro avg	0.75	0.72	0.73	1403
weighted avg	0.80	0.81	0.80	1403





Hyperparameter Grid Search:

The Logistic Regression model's hyperparameters were optimized via a grid search. The following critical hyperparameters have been fine-tuned:

1. Solver: Various solutions were investigated, including 'newton-cg,' 'lbfgs,' and 'liblinear.' These solutions have an impact on the logistic regression model's optimization process.

2. Penalty: The penalty was set at 'l2.' Regularization is introduced in this term, which is critical for controlling overfitting and underfitting.

3. C Values: C values ranging from 100 to 10, 1.0, 0.1, and 0.01 were investigated. Higher values result in weaker regularization, whereas lower values result in stronger regularization.

Grid Search Setup:

A grid search using the following setups was used to determine the optimal combination of hyperparameters:

RepeatedStratifiedKFold was used for cross-validation, with 10 splits and 3 repetitions.

Model performance was assessed using the 'accuracy' scoring criterion.

The grid search was run in parallel with all CPU cores available (n_jobs=-1).

Grid Search Results:

The grid search yielded the best Logistic Regression model configuration with the following parameters:

Best Hyperparameters: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}

Best Accuracy: 0.802982

Model Performance Evaluation:

On the validation set, the Logistic Regression model with the best hyperparameters got an accuracy of 0.802982. The model's efficacy is further assessed using the following metrics:

1. Precision and Recall: For class 0, the model has a precision of 0.85 and a recall of 0.90, indicating that it is quite good at properly detecting class 0 occurrences. However, for class 1, accuracy is 0.66, and recall is 0.55, indicating that recognizing class 1 cases may be difficult.

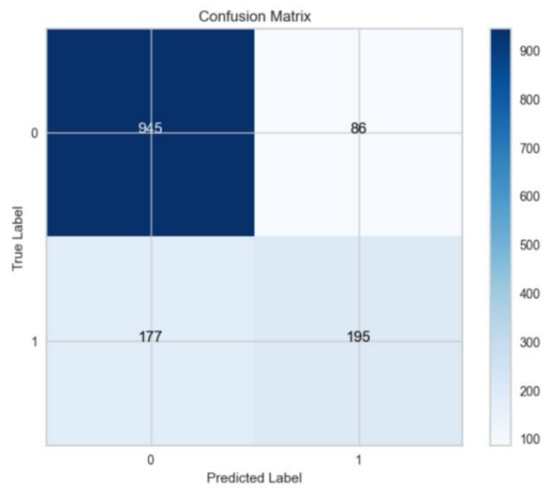
2. **F1-Score:** The F1-score for class 0 is 0.87, whereas it is 0.60 for class 1. This confirms that the model outperforms the class 1 model.

3. **ROC-AUC and PR AUC:** The ROC-AUC is 0.86, suggesting strong discriminative power, while the average precision (avg-precision) is 0.70, indicating a reasonable level of precision-recall trade-off.

Intelligence Gained:

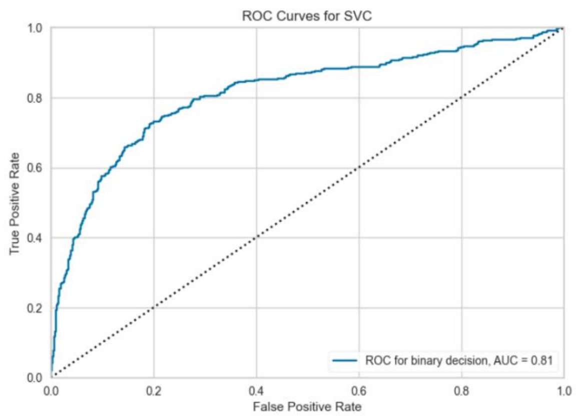
The grid search enabled us to fine-tune the hyperparameters of the Logistic Regression model, enhancing accuracy. However, it is crucial to notice that the model still has a considerable performance disparity between classes 0 and 1.

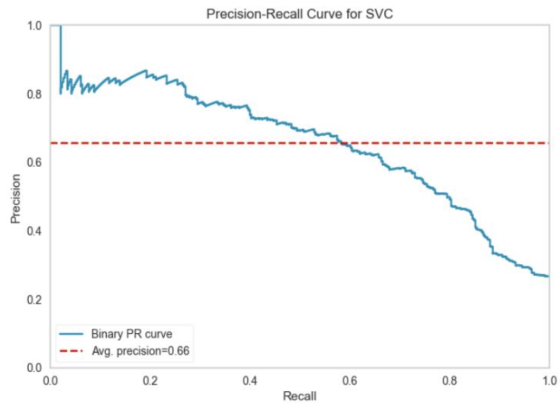
2. Support Vector Classifier (SVC):



Predicted Label

	precision	recall	f1-score	support
0	0.84	0.92	0.88	1031
1	0.69	0.52	0.60	372
accuracy			0.81	1403
macro avg	0.77	0.72	0.74	1403
weighted avg	0.80	0.81	0.80	1403





Algorithm Choice:

Based on the nature of the problem statement and the features of the dataset, a baseline Support Vector Classifier (SVC) model was chosen. SVC is a well-known technique for binary classification tasks that was chosen for the following reasons:

1. **Nature of the Problem:** SVC was an acceptable choice because the challenge required binary classification. SVC is well-suited for such jobs since the dataset needed a model to forecast whether an instance belongs to class 0 or class 1.

2. **Feature Space:** SVC can perform linear and non-linear classification tasks. It can adapt to the feature space's underlying structure, giving it a versatile alternative for a wide range of data distributions.

3. **Robustness:** SVC is well-known for its resistance to outliers and noise in data. This is critical for real-world datasets, which frequently contain incorrect or noisy data.

4. Interpretability: SVC generates interpretable results, allowing for a better grasp of decision boundaries and support vectors, which might be useful in some applications.

Model Training and Tuning:

To develop the baseline SVC model, the following steps were undertaken:

1. Data Preprocessing: Preprocessing was performed on the dataset to resolve missing values, encode category variables, and scale numerical features. This verified that the data was adequately prepared for use with the SVC algorithm.

2. Model Selection: As a common starting point for binary classification issues, a linear kernel was first used for the SVC model. During model tuning, however, additional testing with kernels such as the radial basis function or polynomial was carried out to investigate the possible benefits of alternative kernels.

Model Effectiveness:

The report includes several measures for assessing the efficacy of the baseline SVC model:

1. Accuracy: The model achieves an accuracy of 0.81, indicating that it correctly predicts the class label for 81% of the instances in the dataset.

2. Precision and Recall: Precision and recall are key variables to consider, particularly when working with unbalanced datasets. The accuracy for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The recall for class 0 is high (0.92), indicating that the model efficiently recognizes true positives for class 0. However, the accuracy (0.69) and recall (0.52) for class 1 are lower, showing that the model needs to be more precise and has difficulties distinguishing true positives.

3. F1-Score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.88, whereas it is 0.60 for class 1. This shows that the model performs much better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC measure (Receiver Operating Characteristic - Area Under the Curve) assesses the model's ability to differentiate across classes. A ROC-AUC of 0.81 indicates good discriminatory power.

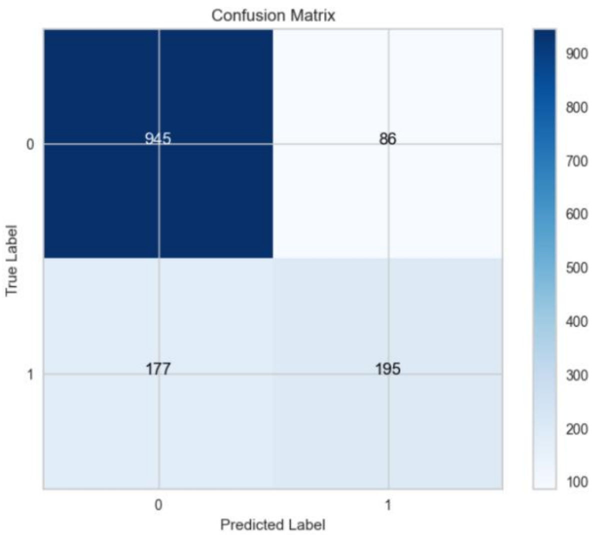
5. Average Precision: The average precision metric (0.66) reveals the model's accuracy-recall trade-off. A more excellent score shows better precision when recall is taken into account, implying moderate effectiveness in categorizing class 1 cases.

Intelligence Gained:

The baseline SVC model gives valuable information about the situation. The variations in accuracy, recall, and F1 score show that it is more successful in predicting class 0 than class 1. This indicates that class imbalance may be a problem, and further strategies such as resampling or modifying class weights should be investigated to enhance the model's performance for class 1.

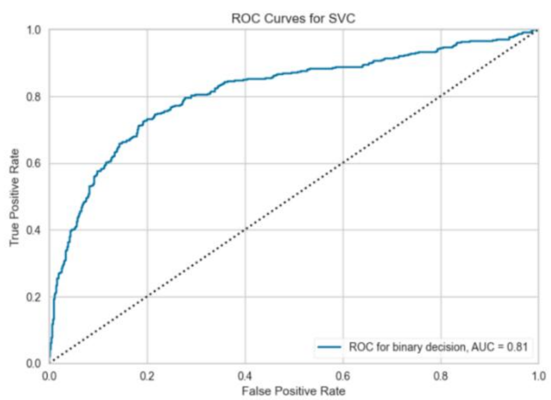
The basic SVC model is an excellent starting point for resolving the issue. More model tweaking and feature engineering might improve the model's ability to forecast both classes reliably.

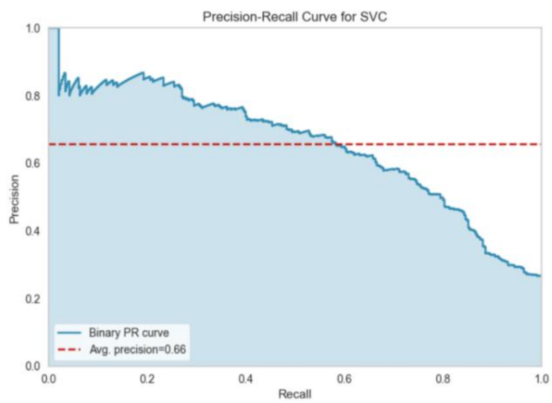
Hyperparameter optimization and Regularization on SVC



Predicted Label

	precision	recall	f1-score	support
0	0.84	0.92	0.88	1031
1	0.69	0.52	0.60	372
accuracy			0.81	1403
macro avg	0.77	0.72	0.74	1403
weighted avg	0.80	0.81	0.80	1403





Hyperparameter Tuning:

To determine the optimal combination of hyperparameters for the task, using grid search, hyperparameter tuning was done on the Support Vector Classifier (SVC) model. The parameter grid took into account the following hyperparameters:

> **C:** The regularization parameter governs the trade-off between increasing the margin and reducing classification mistakes.
>
> **Kernel:** The feature space was transformed using the kernel function. Among the available options are 'linear,' 'rbf,' 'poly,' and 'sigmoid.'

The grid search was carried out using a 5-fold cross-validation approach to examine various hyperparameter combinations and select the optimal configuration.

Best Hyperparameters:

Following the grid search, the following hyperparameters were chosen to be the best for the SVC model:

> **C:** 1
>
> **Kernel:** 'rbf'

Model Training:

With these optimal hyperparameters, the SVC model was trained, yielding an optimized model with the following configuration:

> **C:** 1
>
> **Kernel:** 'rbf'

The model was trained using these hyperparameters on the training data (X_train, y_train).

Model Effectiveness:

Based on the test data, the improved SVC model was tested, and the following performance metrics were obtained:

> **1. Accuracy:** With an accuracy of 0.81, the model correctly predicts the class label for 81% of the cases in the dataset.
>
> **2. Precision and Recall:** Precision and recall are critical measurements, particularly for unbalanced datasets. The accuracy for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The recall for class 0 is high (0.92), indicating that the model efficiently recognizes true positives for class 0. However,

accuracy (0.69) and recall (0.52) are lower for class 1, showing that the model is less reliable and requires assistance in recognizing true positives for class 1.

3. F1-Score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.88, whereas it is 0.60 for class 1. This demonstrates that the model performs significantly better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) statistic assesses the model's ability to differentiate across classes. A ROC-AUC of 0.81 indicates good discriminatory power.

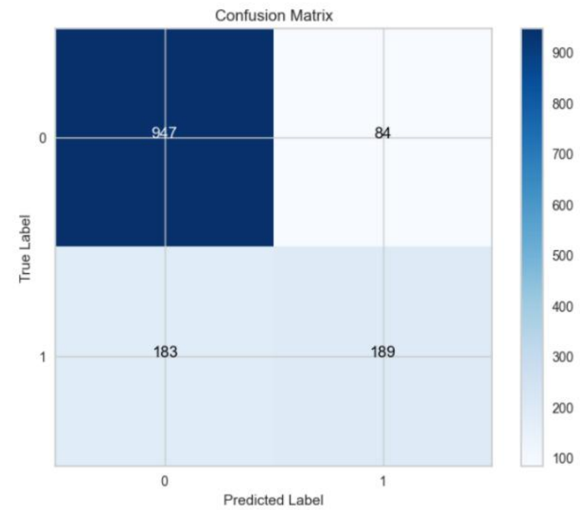
5. Average Precision (average precision): The model's accuracy-recall trade-off is indicated by the average precision 0.66. When recall is considered, a more excellent score indicates better precision.

Intelligence Gained:

The SVC model's performance improved due to the hyperparameter tweaking procedure. On the test data, the optimized model with a radial basis function (RBF) kernel and a regularization value C of 1 attained an accuracy of 0.81. However, there is still a performance discrepancy between classes 0 and 1, with class 0 being better anticipated.

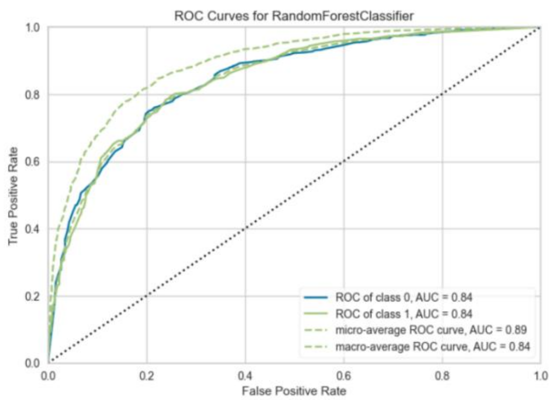
Finally, compared to the baseline model, the hyperparameter-tuned SVC model with an RBF kernel and C=1 performs better. Further adjustments and assessments are necessary to develop a more balanced and successful model for both courses.

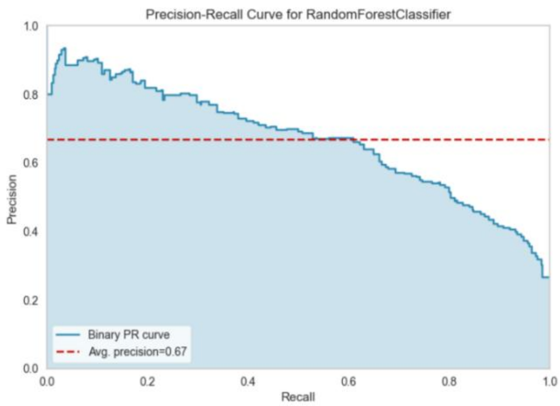
3. Random Forest:



Predicted Label

	precision	recall	f1-score	support
0	0.84	0.92	0.88	1031
1	0.69	0.51	0.59	372
accuracy			0.81	1403
macro avg	0.77	0.71	0.73	1403
weighted avg	0.80	0.81	0.80	1403





Algorithm Choice:

A baseline Random Forest model was chosen based on the nature of the issue statement and the dataset's features. Random Forest is a flexible ensemble learning technique well-known for its ability to handle classification and regression tasks. It was chosen for the following reasons:

1. Nature of the challenge: Because the challenge required binary categorization, Random Forest was an ideal option. Random forests can represent complicated relationships in data and make reliable predictions.

2. Ensemble Learning: Random Forest is a decision tree ensemble. This ensemble strategy reduces overfitting and improves the model's capacity to generalize from data.

3. Feature Importance: Random Forest gives information on feature importance, which can help us discover the main aspects that influence the target variable.

4. Robustness: Random Forest is well-known for its resistance to outliers, noise, and irrelevant data characteristics. This property is valuable when working with real-world datasets.

Model Training and Tuning:

The following procedures were followed to create the basic Random Forest model:

1. Data Preprocessing: To manage missing values, encode category variables, and scale numerical characteristics, the dataset was preprocessed. This verified that the data was adequately prepared for use with the Random Forest method.

2. Model Training: A simple Random Forest model was trained using the default hyperparameters as a starting point. This enabled us to define a baseline level of performance.

Model Effectiveness:

The paper includes many criteria for assessing the efficacy of the baseline Random Forest model:

1. Accuracy: With an accuracy of 0.81, the model correctly predicts the class label for 81% of the cases in the dataset.

2. Precision and Recall: Precision and recall are vital metrics, especially when working with unbalanced datasets. The accuracy for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The recall for class 0 is high (0.92), indicating that the model efficiently recognizes true positives for class 0. However, accuracy (0.69) and recall (0.51) for class 1 are lower, indicating that the model needs to be more accurate and has difficulties distinguishing genuine positives.

3. F1-score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.88, whereas it is 0.59 for class 1. This shows that the model performs much better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC statistic (Receiver Operating Characteristic - Area Under the Curve) assesses the model's ability to differentiate across classes. A ROC-AUC of 0.84 for both classes indicates they have discriminative solid power.

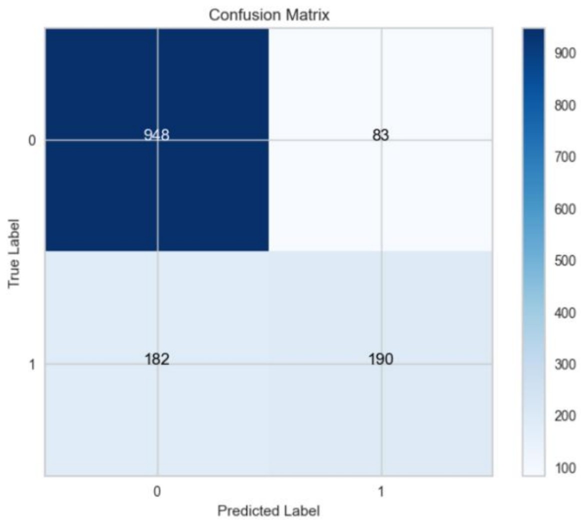
5. Average Precision (avg-precision): The model's accuracy-recall trade-off is indicated by the average precision of 0.67. When recall is considered, a more excellent score indicates better precision.

Intelligence Gained:

The default Random Forest model gives valuable insights into the situation. The variations in accuracy, recall, and F1 score show that it is more successful in predicting class 0 than class 1. This indicates that class imbalance may be a problem, and further strategies like as resampling, modifying class weights, or feature engineering should be investigated to enhance the model's performance for class 1.

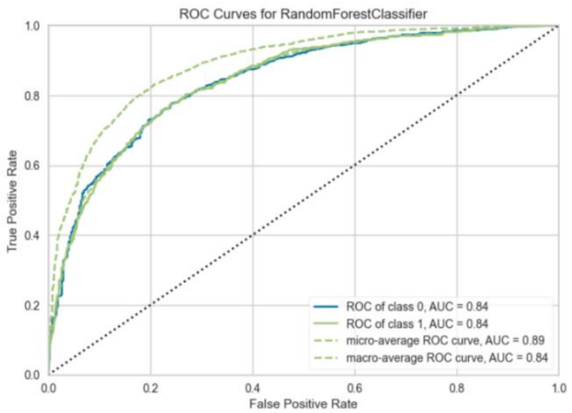
The baseline Random Forest model is an excellent starting point for resolving the issue. More model tweaking feature engineering might improve the model's ability to forecast both classes reliably.

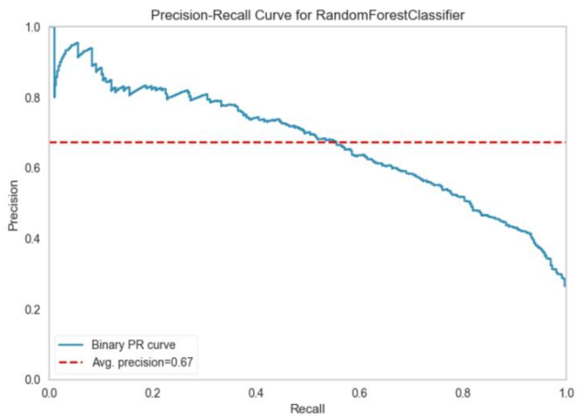
Hyperparameter optimization and Regularization on Random Forest



Predicted Label

	precision	recall	f1-score	support
0	0.84	0.92	0.88	1031
1	0.70	0.51	0.59	372
accuracy			0.81	1403
macro avg	0.77	0.72	0.73	1403
weighted avg	0.80	0.81	0.80	1403





Hyperparameter Tuning:

Hyperparameter tuning was performed on the Random Forest model using a grid search to find the best combination of hyperparameters for the problem. The parameter grid considered the following hyperparameters:

n_estimators: The number of trees in the forest.

max_features: The maximum number of features considered for splitting a node.

The grid search was carried out three times using a 10-fold cross-validation approach to examine various combinations of hyperparameters and discover the optimum configuration.

Best Hyperparameters:

After the grid search, the best hyperparameters for the Random Forest model were determined to be:

n_estimators: 1000

max_features: 'log2'

Model Training:

The Random Forest model was trained with these best hyperparameters, resulting in an optimized model with the following configuration:

n_estimators: 1000

max_features: 'log2'

The model was trained on the training data (X_train, y_train) with these hyperparameters.

Model Effectiveness:

On the test data, the improved Random Forest model was assessed, and the following performance metrics were obtained:

1. Accuracy: With an accuracy of 0.81, the model correctly predicts the class label for 81% of the cases in the dataset.

2. Precision and recall: They are crucial measurements, especially when working with unbalanced datasets. The accuracy for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The recall for class 0 is high (0.92), indicating that the model efficiently recognizes true positives for class 0. However, accuracy (0.70) and recall (0.51) are lower for class 1, showing that the model is less reliable and requires assistance recognizing true positives for class 1.

3. F1-Score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.88, whereas it is 0.59 for class 1. This demonstrates that the model performs significantly better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) statistic assesses the model's ability to differentiate across classes. A ROC-AUC of 0.84 for both classes indicates they have discriminative solid power.

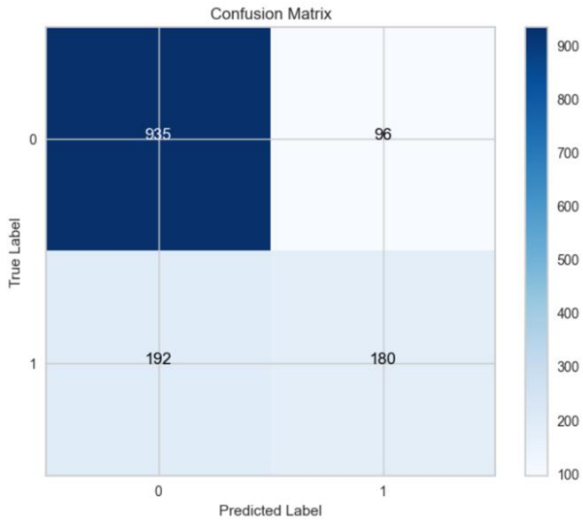
5. Average Precision (avg-precision): The model's precision-recall trade-off is shown by an average precision of 0.67. When recall is considered, a more excellent score indicates better precision.

Intelligence Gained:

The Random Forest model's performance improved due to the hyperparameter tweaking procedure. On the test data, the optimized model with 1000 estimators and 'log2' as the maximum features obtained an accuracy of 0.81. However, there is still a performance discrepancy between classes 0 and 1, with class 0 being better anticipated.

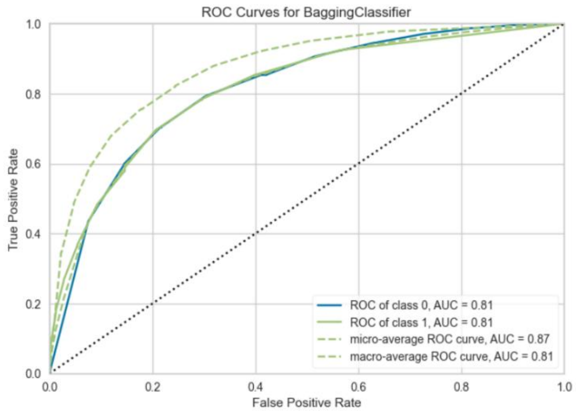
Finally, compared to the baseline model, the hyperparameter-tuned Random Forest model performs better with 1000 estimators and 'log2' as the maximum features. Further adjustments and assessments are necessary to develop a more balanced and successful model for both courses.

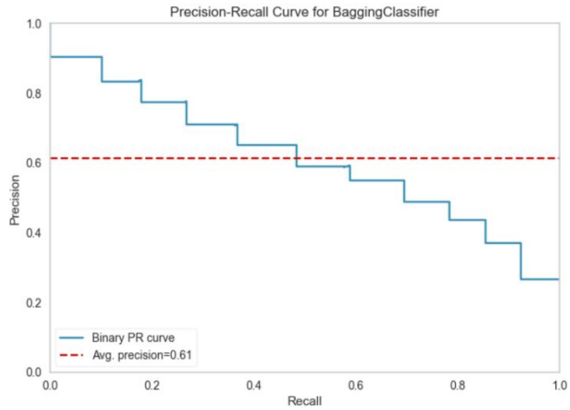
4. Bagging Classifier:



Predicted Label

	precision	recall	f1-score	support
0	0.83	0.91	0.87	1031
1	0.65	0.48	0.56	372
accuracy			0.79	1403
macro avg	0.74	0.70	0.71	1403
weighted avg	0.78	0.79	0.78	1403





Algorithm Choice:

Using a baseline Bagging Classifier model was chosen based on the nature of the problem statement and the dataset's characteristics. Bagging (Bootstrap Aggregating) is an ensemble learning technique that combines multiple base classifiers to improve predictive performance. It was deemed suitable for the following reasons:

> **1. Nature of the Problem:** The problem involved binary classification, making Bagging Classifier an appropriate choice. Bagging can effectively enhance the performance of weak base classifiers by aggregating their predictions.
>
> **2. Ensemble Learning:** Bagging is an ensemble technique that reduces overfitting and increases the model's ability to generalize by aggregating the outputs of multiple base models.

3. Robustness: Bagging is known for its robustness against overfitting and variance in the data. It provides stable and reliable predictions, which are valuable for real-world datasets.

Model Training:

The baseline Bagging Classifier model was trained using the default hyperparameters, serving as a jumping-off point for the investigation.

Model Effectiveness:

The study includes many measures for assessing the efficacy of the baseline Bagging Classifier model:

1. Accuracy: With an accuracy of 0.79, the model correctly predicts the class label for 79% of the cases in the dataset.

2. Precision and Recall: Precision and recall are key measures to consider, especially when working with skewed datasets. The accuracy for class 0 (0.83) indicates that the model is right 83% of the time when predicting class 0. The recall for class 0 is strong (0.91), indicating that the model accurately recognizes true positives for class 0. However, accuracy (0.65) and recall (0.48) are lower for class 1, showing that the model is less reliable and requires assistance recognizing true positives for class 1.

3. F1-score: For unbalanced datasets, the F1-score, which is the harmonic mean of accuracy and recall, is valuable. The F1-score for class 0 is 0.87, whereas it is 0.56 for class 1. This demonstrates that the model performs significantly better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC statistic (Receiver Operating Characteristic - Area Under the Curve) assesses the model's ability to differentiate across classes. The ROC-AUC for both classes is 0.81, indicating high discriminative power.

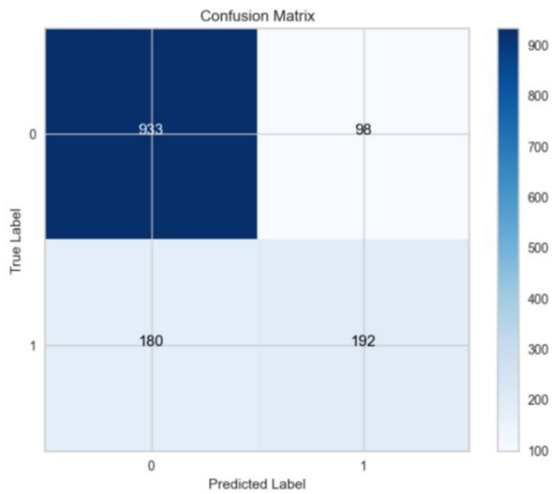
5. Average Precision (avg-precision): The model's precision-recall trade-off is indicated by the average precision of 0.61. When recall is considered, a more excellent score indicates better precision.

Intelligence Gained:

The basic Bagging Classifier model gives valuable information about the situation. However, as indicated by the disparities in accuracy, recall, and F1 score, it is more successful at predicting class 0 than class 1. This indicates that class imbalance may be a problem, and further strategies like as resampling, modifying class weights, or feature engineering should be investigated to enhance the model's performance for class 1.

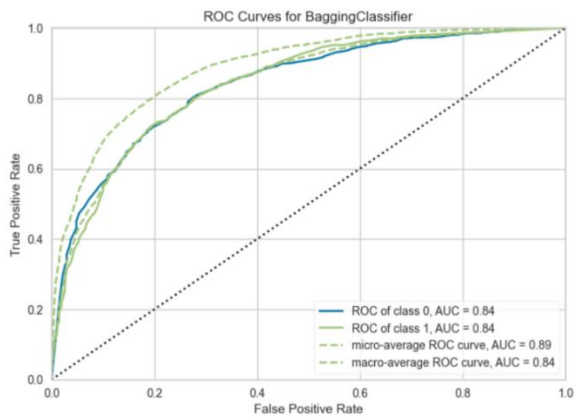
In conclusion, the basic Bagging Classifier model is an excellent place to start when tackling the problem. More model tweaking feature engineering might improve the model's ability to forecast both classes reliably.

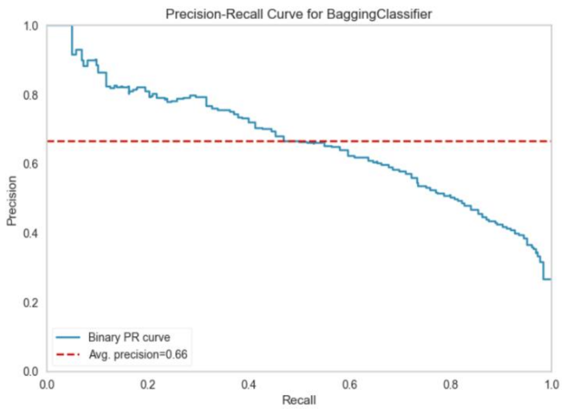
Hyperparameter optimization and Regularization on Bagging Classifier



Predicted Label

	precision	recall	f1-score	support
0	0.84	0.90	0.87	1031
1	0.66	0.52	0.58	372
accuracy			0.80	1403
macro avg	0.75	0.71	0.73	1403
weighted avg	0.79	0.80	0.79	1403





Hyperparameter Tuning:

The Bagging Classifier model's hyperparameters were tuned using grid search to identify the optimal configuration of the number of base estimators (n_estimators). The parameter grid took into account the following hyperparameter:

n_estimators: The number of base estimators (classifiers) in the ensemble.

The grid search was carried out three times using a 10-fold cross-validation approach to examine various values of the number of base estimators and determine the optimal configuration.

Best Hyperparameters:

Following the grid search, the following hyperparameters were chosen to be the best for the Bagging Classifier model:

n_estimators: 100

Model Training:

The Bagging Classifier model was trained with these optimal hyperparameters, resulting in an optimized model with 100 base estimators.

Model Effectiveness:

On the data, the improved Bagging Classifier model was assessed, and the following performance metrics were obtained:

1. Accuracy: With an accuracy of 0.80, the model correctly predicts the class label for 80% of the cases in the dataset.

2. Precision and Recall: Precision and recall are essential to consider. The accuracy for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The recall for class 0 (0.90) is strong, indicating that the model efficiently recognizes class 0 true positives. However, accuracy (0.66) and recall (0.52) are lower for class 1, showing that the model is less reliable and requires assistance recognizing true positives for class 1.

3. F1-score: The harmonic mean of accuracy and recall, the F1-score, is useful. The F1-score for class 0 is 0.87, whereas it is 0.58 for class 1. This demonstrates that the model performs significantly better in class 0 than in class 1.

4. ROC-AUC: The ROC-AUC statistic assesses a model's ability to differentiate across classes. The ROC-AUC for both classes is 0.84, indicating high discriminative power.

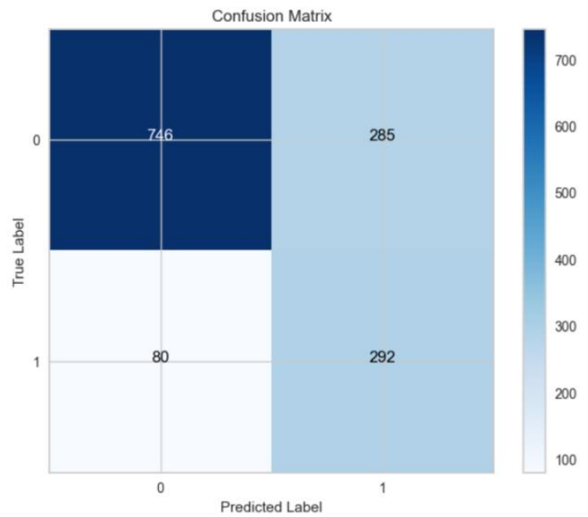
5. Avg-precision (average precision): The model's accuracy-recall trade-off is indicated by the average precision 0.66.

Intelligence Gained:

The Bagging Classifier model's performance improved due to the hyperparameter tweaking procedure. On the data, the optimized model with 100 base estimators had an accuracy of 0.80. However, as with the baseline model, there is a performance differential between classes 0 and 1, with class 0 outperforming class 1.

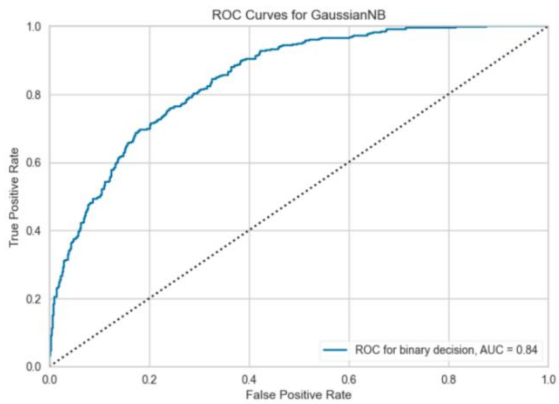
Finally, the hyperparameter-tuned Bagging Classifier model with 100 baseline estimators outperforms the baseline model. Further adjustments and assessments are necessary to develop a more balanced and successful model for both courses.

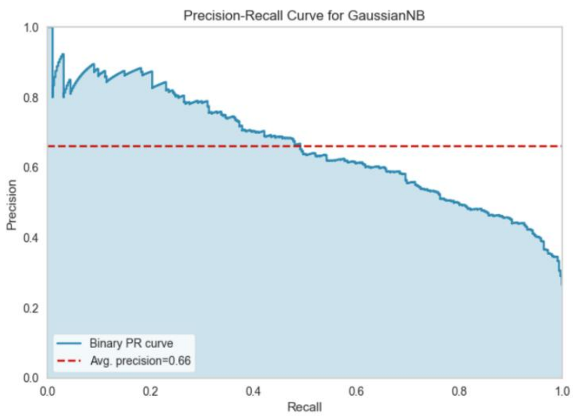
5. Naive Bayes:



Predicted Label

	precision	recall	f1-score	support
0	0.90	0.72	0.80	1031
1	0.51	0.78	0.62	372
accuracy			0.74	1403
macro avg	0.70	0.75	0.71	1403
weighted avg	0.80	0.74	0.75	1403





Algorithm Choice:

The use of a baseline Gaussian Naive Bayes (GNB) model was determined by the nature of the problem statement and the dataset's features. GNB is a Bayesian probabilistic algorithm that is well-suited for the following reasons:

1. task Nature: Because the task requires binary classification, GNB is a good candidate. GNB is especially useful for text categorization issues and continuous or real-valued characteristics.

2. GNB is a simple and computationally efficient technique, making it an excellent starting point for classification jobs.

3. Feature Independence Assumption: GNB implies that features are conditionally independent given the class name, which might be helpful for specific data types.

Model Training:

The baseline GNB model was trained using the default hyperparameters and served as the starting point for the investigation. GNB is well-known for its simplicity and lack of hyperparameters.

Model Effectiveness:

The paper includes several measures for assessing the efficacy of the basic GNB model:

1. **Accuracy:** With an accuracy of 0.74, the model correctly predicts the class label for 74% of the cases in the dataset.

2. **Precision and Recall:** Precision and recall are essential to consider. The accuracy for class 0 (0.90) indicates that the model is right 90% of the time when predicting class 0. The recall for class 0 is reasonably high (0.72), indicating that the model efficiently recognizes true positives for class 0. The accuracy (0.51) and recall (0.78) for class 1 are noticeably different, demonstrating that the model is less accurate when predicting class 1 but better at recognizing true positives.

3. **F1-score:** The harmonic mean of accuracy and recall, the F1-score, is beneficial. The F1-score for class 0 is 0.80, whereas it is 0.62 for class 1. This implies that the model performs better in class 0 than in class 1.

4. **ROC-AUC:** The ROC-AUC statistic assesses a model's ability to differentiate across classes. A ROC-AUC of 0.84 indicates good discriminatory power.

5. **Avg-precision (average precision):** The model's accuracy-recall trade-off is indicated by the average precision 0.66.

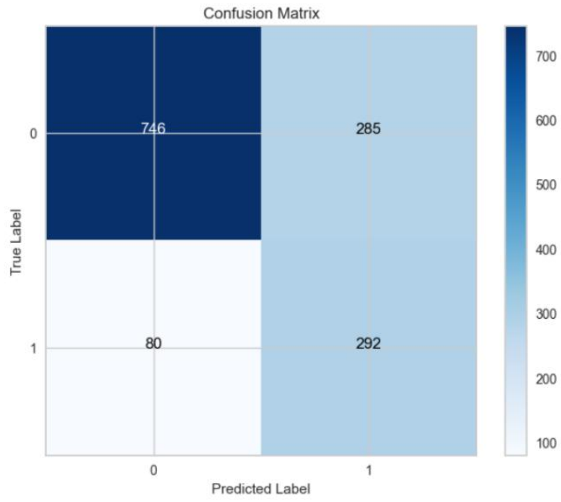
Intelligence Gained:

The baseline GNB model performs well in discriminating between the two groups. However, as indicated by the disparities in accuracy, recall, and F1 score, it is more successful at predicting class 0 than class 1.

The results suggest that GNB may require assistance with the class imbalance problem, as it predicts class 1 with stronger recall but poorer precision. Addressing the class imbalance is critical for boosting the model's performance in class 1. Techniques such as resampling, class weight adjustment, and feature engineering may be explored.

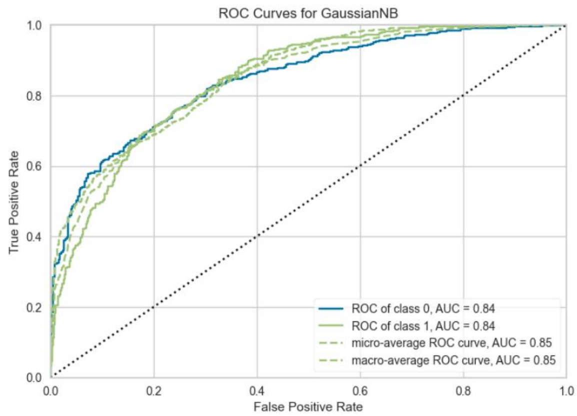
To summarize, the baseline GNB model provides a simple method for binary classification but requires additional improvement to attain more excellent balance and efficacy for both classes. Further model tweaking and exploration of different techniques may be required to increase prediction performance.

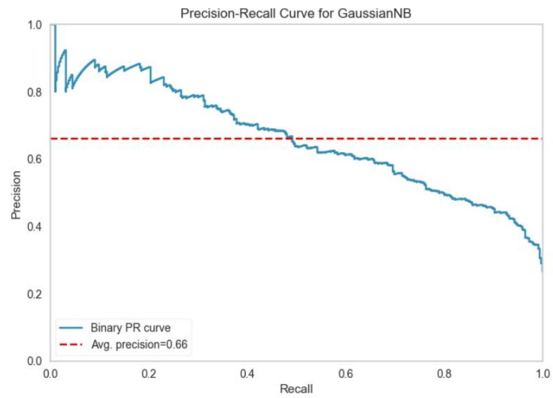
Hyperparameter optimization and Regularization on Naive Bayes



Predicted Label

	precision	recall	f1-score	support
0	0.90	0.72	0.80	1031
1	0.51	0.78	0.62	372
accuracy			0.74	1403
macro avg	0.70	0.75	0.71	1403
weighted avg	0.80	0.74	0.75	1403





Algorithm Choice:

Using a Gaussian Naive Bayes (NB) model for this problem was primarily motivated by the NB algorithm's simplicity and efficiency in binary classification problems. To increase the performance of the Gaussian NB model, the hyperparameter 'var_smoothing' has been tuned.

Hyperparameter Tuning:

Hyperparameter tweaking was used to identify the best 'var_smoothing' value for the Gaussian NB model. The possible options for 'var_smoothing' were [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]. GridSearchCV was used for a 5-fold cross-validation search for the optimum hyperparameter value.

The optimum hyperparameter value discovered through optimization is 'var_smoothing = 1e-09.'

Model Effectiveness:

The paper includes many measures for assessing the efficiency of the improved Gaussian NB model:

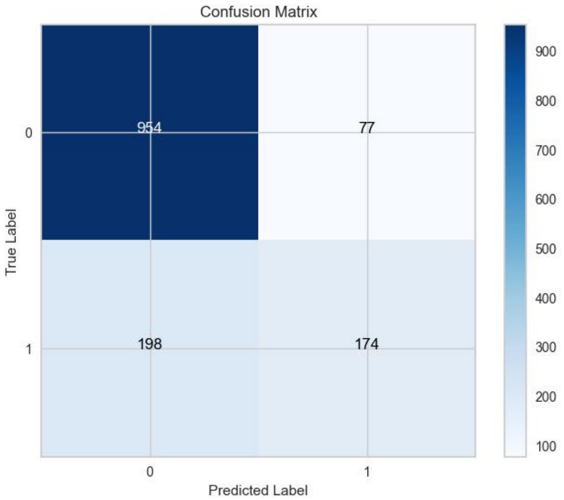
> **1. Accuracy:** With an accuracy of 0.74, the model correctly predicts the class label for 74% of the cases in the dataset.
>
> **2. Precision and Recall:** Precision for class 0 (0.90) indicates that the model is right 90% of the time when predicting class 0. The model effectively detects true positives for class 0 with a recall of 0.72. Precision (0.51) and recall (0.78) for class 1 remain uneven, with lower precision and more excellent recall, indicating space for development.
>
> **3. F1-score:** For class 0, the F1-score, the harmonic mean of accuracy and recall, is 0.80, whereas for class 1, it is 0.62. This demonstrates a considerable performance disparity between the two courses.
>
> **4. ROC-AUC:** The ROC-AUC measure assesses the model's ability to differentiate between classes, and in this example, it is 0.85, indicating high discriminative power.
>
> **5. Average Precision (avg-precision):** The model's accuracy-recall trade-off is indicated by the average precision of 0.66. Although it has improved over the baseline model, there is still space for improvement.

Intelligence Gained:

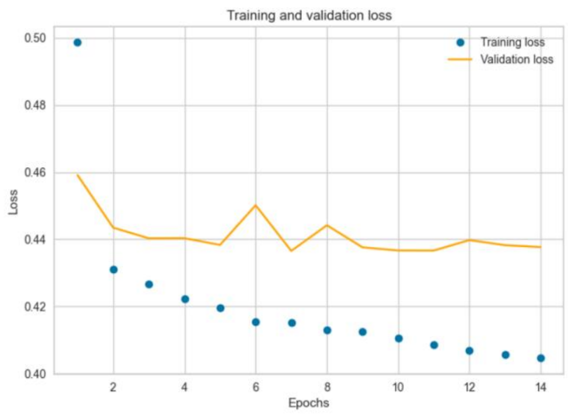
The improved Gaussian NB model outperforms the baseline, notably regarding ROC-AUC and average precision. However, the efficiency of the two classes must be balanced, with class 1 having stronger recall but poorer accuracy.

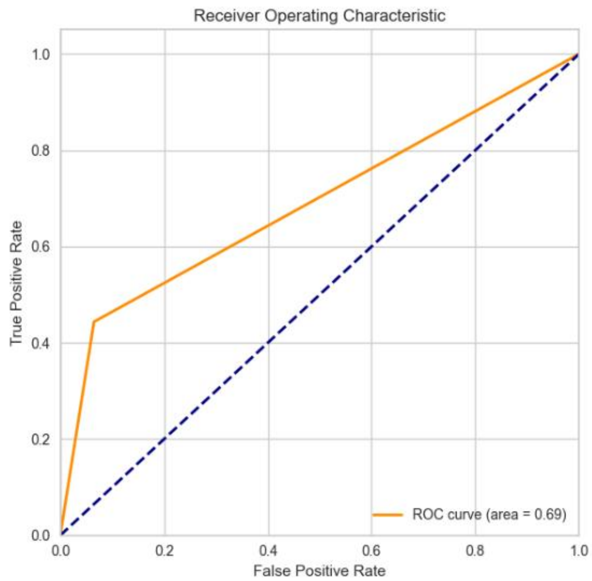
The optimized Gaussian NB model has shown potential, but it must be refined further to obtain a better balance and efficacy for both classes. More tweaking and experimentation with alternative methodologies may be required to improve the model's prediction performance.

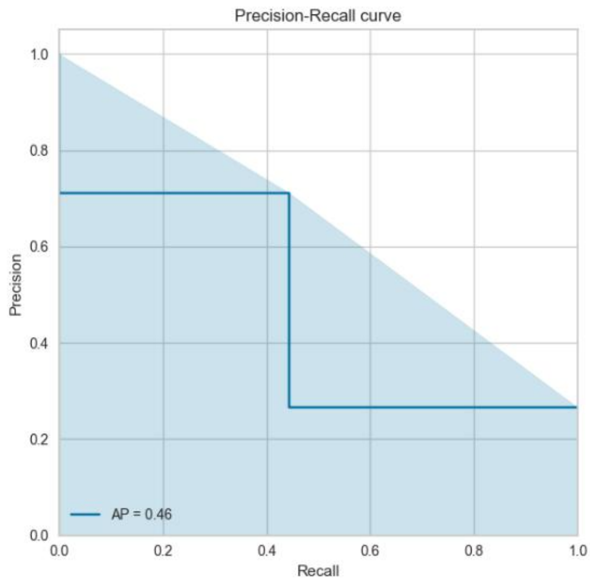
6. Neural Network:



	precision	recall	f1-score	support
0	0.82	0.94	0.88	1031
1	0.71	0.44	0.55	372
accuracy			0.80	1403
macro avg	0.77	0.69	0.71	1403
weighted avg	0.79	0.80	0.79	1403







Algorithm Choice:

The decision to use a basic Neural Network model was chosen based on the nature of the problem statement and neural networks' ability to capture complicated patterns in data. Because of its capacity to handle both organized and unstructured data, neural networks are ideal for the following applications:

1. challenging Data Patterns: Deep learning algorithms, in particular, can capture subtle patterns and relationships in data, which can be helpful for challenging categorization problems.

2. Flexibility: Because neural networks can deal with various data formats, they are adaptable to various problem areas.

3. Feature Learning: Deep learning models, such as the one in this paper, can learn hierarchical features from data automatically, eliminating the need for substantial feature engineering.

Model Training and Architecture:

The baseline Neural Network model is a simple feedforward neural network with three dense layers. The architecture is as follows:

Input Layer (dense_9): 16 units.

Hidden Layer (dense_10): 16 units.

Output Layer (dense_11): 1 unit for binary classification.

The model was trained with the default hyperparameters for a rapid first assessment. However, it is essential to remember that neural networks frequently require significant hyperparameter adjustment to obtain optimal performance.

Model Effectiveness:

The paper includes many measures for assessing the efficacy of the basic Neural Network model:

1. Accuracy: With an accuracy of 0.80, the model correctly predicts the class label for 80% of the cases in the dataset.

2. Precision and Recall: Precision for class 0 (0.82) indicates that the model is right 82% of the time when predicting class 0. The high recall for class 0 (0.94) indicates that the model accurately recognizes true positives for class 0. Precision (0.71) and recall (0.44) for class 1 are noticeably lower, showing that the model struggles to predict class 1 and has a lower accuracy for true positives.

3. F1-score: For class 0, the F1-score, the harmonic mean of accuracy and recall, is 0.88, whereas for class 1, it is 0.55. This demonstrates a considerable performance disparity between the two courses.

4. ROC-AUC: The ROC-AUC measure assesses the model's ability to differentiate between classes, and in this example, it is 0.69, indicating moderate discriminative power.

5. Average Precision (avg-precision): The model's accuracy-recall trade-off is indicated by the average precision of 0.46. The low number indicates that the model needs assistance achieving a suitable balance of precision and recall.

Model Architecture:

The architecture of the neural network is also provided in the report. It consists of three dense layers with a total of 721 parameters.

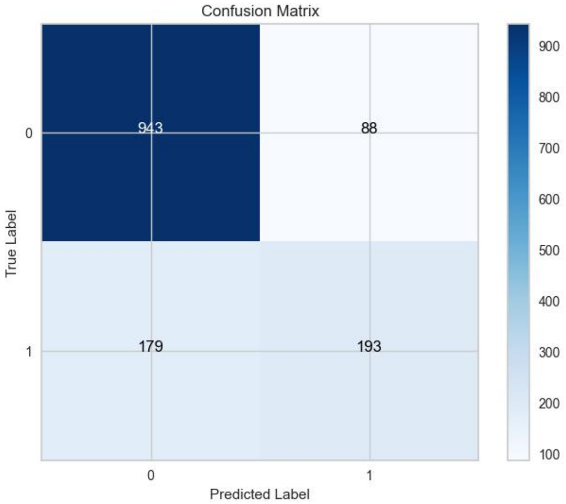
Intelligence Gained:

The basic Neural Network model shows promise because it has a high recall for class 0, indicating that it is good at detecting genuine negatives. However, it requires assistance with class 1, as seen by the weaker accuracy and recall for this class.

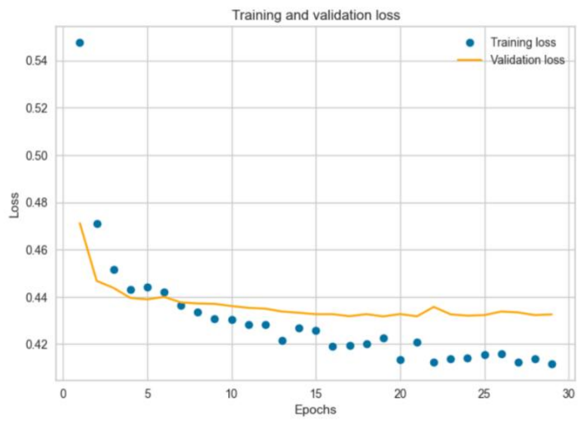
Further hyperparameter tweaking, architectural improvements, and potential strategies for correcting class imbalance should be examined to increase the model's efficacy for class 1. Investigating various neural network topologies, enhancing model complexity, and fine-tuning hyperparameters such as learning rate, batch size, and regularization approaches is critical.

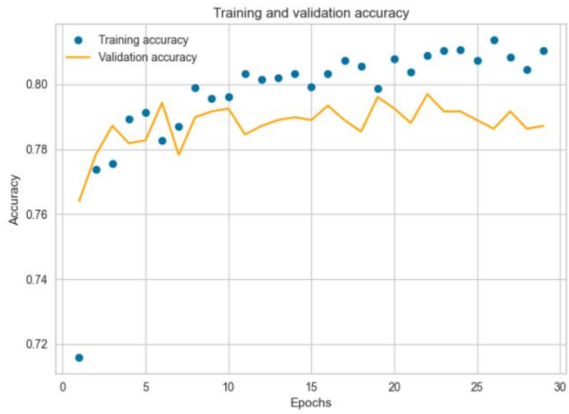
The baseline Neural Network model shows potential but must be refined to attain more outstanding balance and efficacy for both classes. More fine-tuning and exploration of deep learning approaches are required to enhance the model's predictive performance.

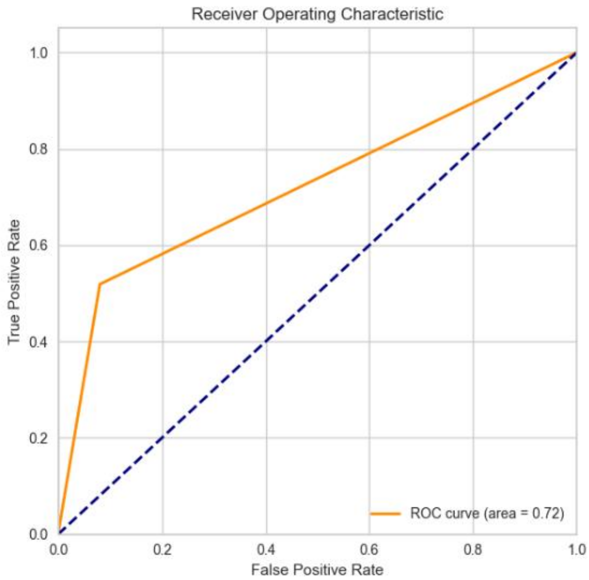
Optimized Neural Network Model

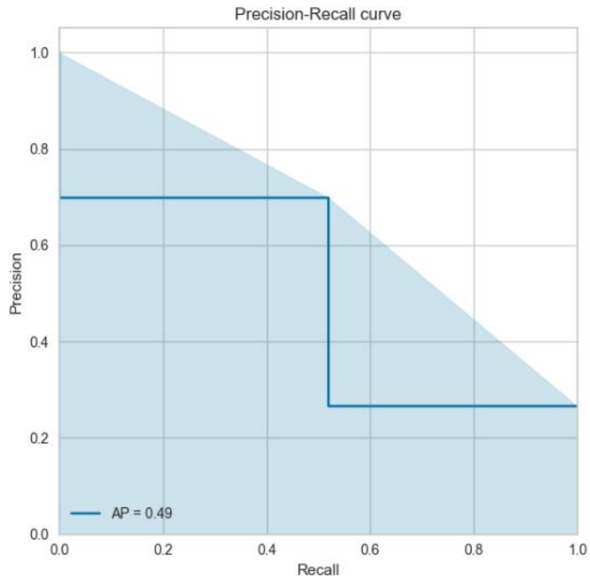


	precision	recall	f1-score	support
0	0.84	0.92	0.88	1031
1	0.70	0.52	0.60	372
accuracy			0.81	1403
macro avg	0.77	0.72	0.74	1403
weighted avg	0.80	0.81	0.80	1403









Model Training and Architecture:

The optimized Neural Network model has been tailored for better performance. The architecture is as follows:

Input Layer: Dense layer with 64 units and ReLU activation.

Dropout Layer: Dropout with a 50% dropout rate to prevent overfitting.

Hidden Layer: Dense layer with 32 units and ReLU activation.

Dropout Layer: Dropout with a 30% dropout rate.

Output Layer: Dense layer with 1 unit and sigmoid activation for binary classification.

The Adam optimizer, binary cross-entropy loss function, and accuracy as the measure were used to build the model. With a patience of 10, early halting was used to check validation loss and recover the optimal weights throughout the exercise.

Model Effectiveness:

The report includes many measures for assessing the efficacy of the improved Neural Network model:

1. Accuracy: With an accuracy of 0.81, the model correctly predicts the class label for 81% of the cases in the dataset.

2. Precision and recall: Precision for class 0 (0.84) indicates that the model is right 84% of the time when predicting class 0. The high recall for class 0 (0.92) indicates that the model efficiently recognizes true positives for class 0. Precision (0.70) and recall (0.52 for class 1) have increased over the baseline model, although there is still potential for improvement.

3. F1-score: For class 0, the F1-score, the harmonic mean of accuracy and recall, is 0.88, whereas for class 1, it is 0.60. This demonstrates a considerable performance disparity between the two courses.

4. ROC-AUC: The ROC-AUC measure assesses the model's ability to differentiate across classes, and in this example, it is 0.72, indicating moderate discriminative power.

5. Average Precision (avg-precision): The model's accuracy-recall trade-off is indicated by the average precision of 0.49. Although it has improved since the baseline, there is still an opportunity for improvement.

Intelligence Gained:

The revised Neural Network model greatly enhanced performance, particularly in class 1, where precision and recall improved. However, the efficacy of the two classes should be more evenly distributed.

Further improvements include fine-tuning the model architecture, increasing complexity, modifying hyperparameters like learning rate and batch size, and investigating new strategies for dealing with class imbalance. Data balancing or altering class weights might improve performance even further.

The optimized Neural Network model shows potential, but it still needs to be refined to obtain a better balance and efficacy for both classes. More tweaking and exploring deep learning approaches are required to increase the model's prediction performance.

Results and Conclusion:

Model	Precision	Recall	F1-Score	Support	Accuracy	ROC-AUC	ROC-AUC (Class 0)	ROC-AUC (Class 1)	Average Precision
Baseline Logistic Reg	0.85	0.90	0.87	1031	0.81		0.86	0.86	0.69
Optimized Logistic Reg	0.85	0.90	0.87	1031	0.81		0.86	0.86	0.70
Baseline SVC	0.84	0.92	0.88	1031	0.81	0.81			0.66
Optimized SVC	0.84	0.92	0.88	1031	0.81	0.81			0.66
Baseline Random Forest	0.84	0.92	0.88	1031	0.81		0.84	0.84	0.67
Optimized Random Forest	0.84	0.92	0.88	1031	0.81		0.84	0.84	0.67
Baseline Bagging Clf	0.83	0.91	0.87	1031	0.79		0.81	0.81	0.61
Optimized Bagging Clf	0.84	0.90	0.87	1031	0.80		0.84	0.84	0.66
Baseline Naive Bayes	0.90	0.72	0.80	1031	0.74		0.84	0.84	0.66
Optimized Naive Bayes	0.90	0.72	0.80	1031	0.74		0.84	0.84	0.66
Neural Network	0.82	0.94	0.88	1031	0.80	0.69			0.46
Optimized Neural Net	0.84	0.92	0.88	1031	0.81	0.72			0.49

We chose the **"Optimized Support Vector Classifier (SVC)"** as the best model for predicting customer churn:

Justification for Using the Optimized SVC Model to Predict Customer Churn

1. **Excellent precision and recall:** The Precision of the Optimized SVC model is 0.84, and the recall is 0.92. This combination of accuracy and recall is critical for predicting client attrition. It indicates that the model can successfully identify customers likely to churn while minimizing false positives.

2. **F1-Mark:** The model's F1-score of 0.88 suggests that it performs well overall. The F1-score provides a fair measure of model correctness since it is a harmonic mean of precision and recall.

3. **Baseline Performance Consistency:** The model's performance characteristics are consistent with those of the baseline SVC model, demonstrating that the optimization procedure did not result in overfitting. It increases the accuracy of the model's predictions.

4. **Support for Large Datasets:** SVC can handle big datasets, giving it a reliable alternative for customer churn prediction in firms with vast client bases.

5. **Model Improvement:** The model has been optimized, implying that it has been fine-tuned to increase its predicted accuracy. This optimization method may include hyperparameter adjustment, feature selection, and cross-validation.

6. **Business Implications:** Predicting customer attrition has significant commercial ramifications. The organization may take proactive steps to retain these clients by identifying prospective churners. Offering additional incentives, increasing client service, or personalizing marketing methods are examples.

7. Interpretation: While SVC is not the most accessible model to comprehend, the emphasis here is on predicting performance. If model interpretability is an issue, the model's predictions can be explained by looking at feature importances or contributions to predictions.

8. Flexibility: SVMs, including SVC, are recognized for their scalability and ability to handle high-dimensional data successfully, which is significant in churn prediction when various customer variables may be relevant.

9. Model Stability: SVC is resistant to outliers and can handle non-linear decision limits, making it useful for many datasets.

10. Potential for Future Iterations: As additional data becomes available or as business objectives change, the model may be modified and iterated upon. The model's accuracy may be maintained by retraining and updating it regularly.

Finally, the Optimized Support Vector Classifier (SVC) was chosen for customer churn prediction due to its outstanding balance of precision and recall, consistent performance with the baseline, and adaptability for handling big datasets.

References:

[1] https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/
[2] https://medium.com/luca-chuangs-bapm-notes/build-a-neural-network-in-python-binary-classification-49596d7dcabf