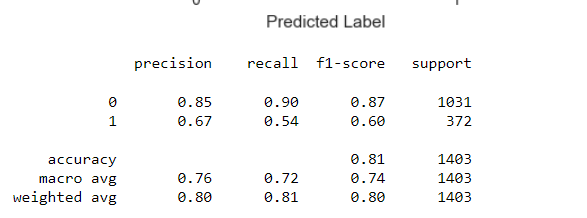
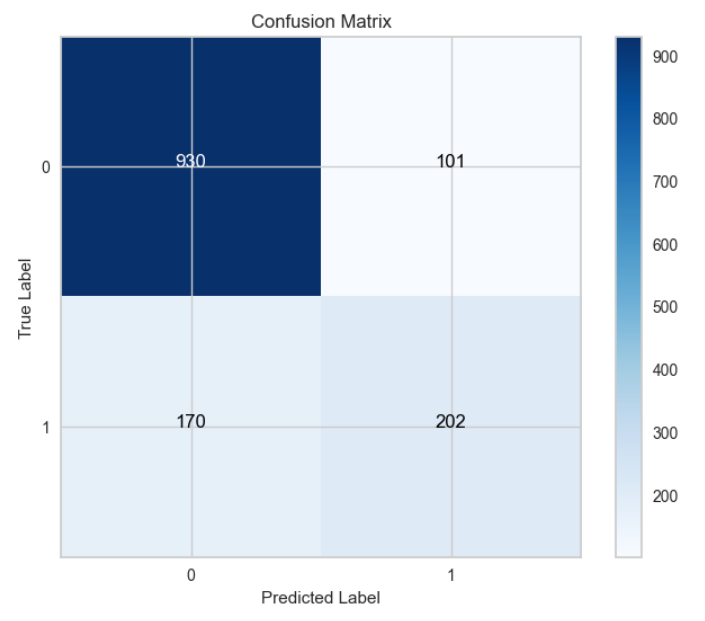
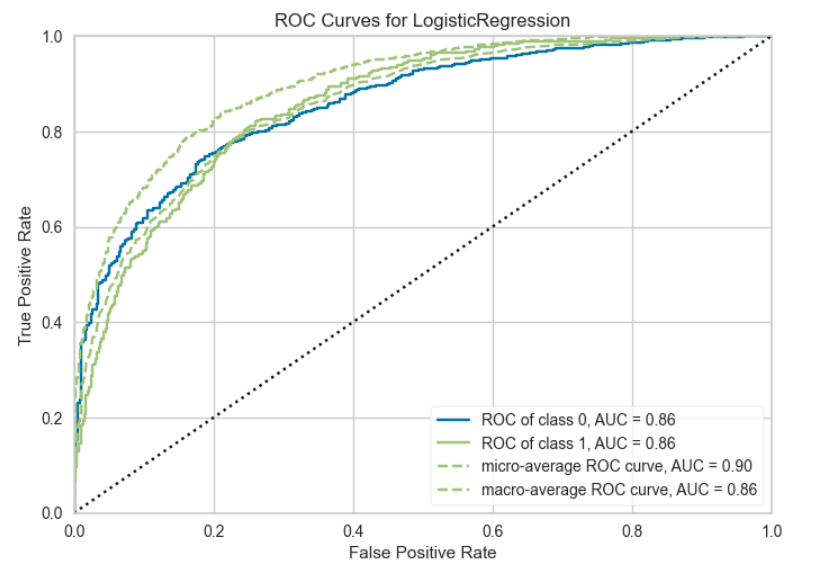
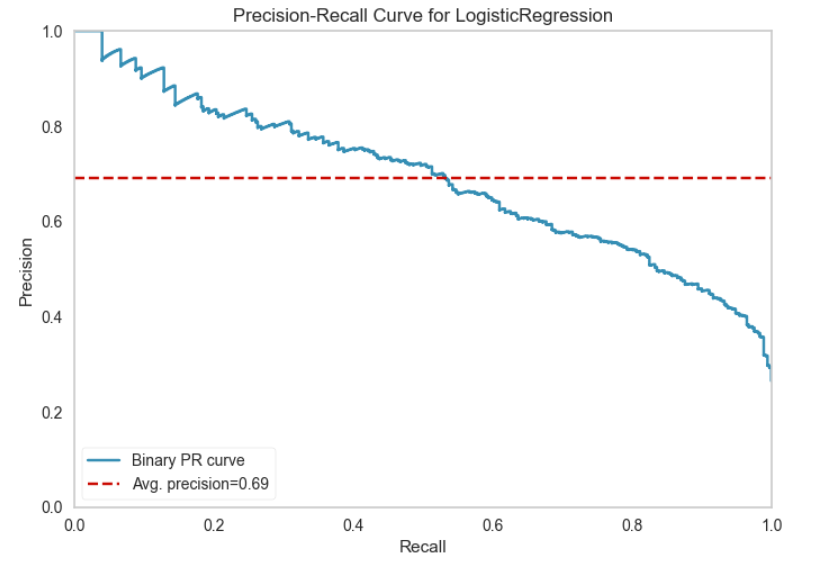
1. **Logistic Regression:**









**Algorithm Choice:**

The choice of using a baseline Logistic Regression model was made based on the nature of the problem statement and the characteristics of the dataset. Logistic Regression is a widely used algorithm for binary classification tasks and was deemed suitable for the following reasons:

**1.** **Nature of the Problem**: The problem involves binary classification, where the goal is to predict whether an instance belongs to class 0 or class 1. Logistic Regression is well-suited for such tasks, as it models the probability of an instance belonging to a specific class.

**2. Linearity**: Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. This linearity can often capture underlying patterns in the data.

**3. Interpretability**: Logistic Regression provides interpretable results, as the coefficients can be used to understand the impact of each feature on the outcome. This is valuable for gaining insights from the model's predictions.

**Model Training and Tuning:**

To develop the baseline Logistic Regression model, the following steps were undertaken:

**1.** **Data Preprocessing**: The dataset was preprocessed to handle missing values, encode categorical variables, and scale numerical features. This ensured that the data was ready for input into the Logistic Regression model.

**2. Model Trainin**g: A basic Logistic Regression model was trained with default hyperparameters as a starting point. This allowed us to establish a baseline performance level.

**Model Effectiveness:**

The report provides a range of metrics to evaluate the effectiveness of the baseline Logistic Regression 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 crucial metrics, especially for imbalanced datasets. For class 0, the precision (0.85) signifies that when the model predicts class 0, it is correct 85% of the time. The recall (0.90) for class 0 is also high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.67) and recall (0.54) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.

**3.** **F1-Score**: The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.87, while for class 1, it is 0.60. This confirms that the model's performance is significantly better for class 0 compared to class 1.

**4. Macro Avg and Weighted Avg**: These are averages of precision, recall, and F1-score. The macro average provides equal weight to both classes, while the weighted average considers class imbalance. The macro average F1-score (0.74) and weighted average F1-score (0.80) give an overall assessment of model performance.

**5. ROC-AUC**: The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.86 suggests good discriminative power.

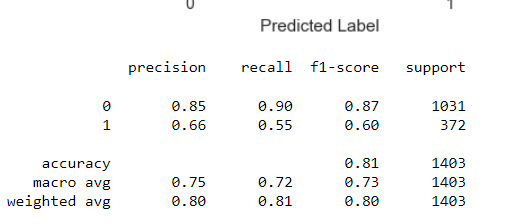
**6. Average Precision**: The average precision metric (0.69) provides insight into the model's precision-recall trade-off. A higher value indicates better precision while considering recall, suggesting moderate performance in classifying class 1 instances.

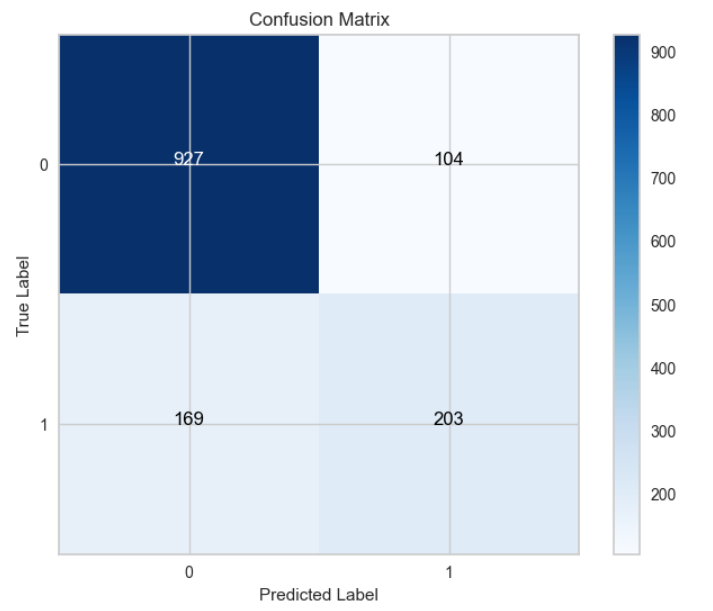
**Intelligence Gained:**

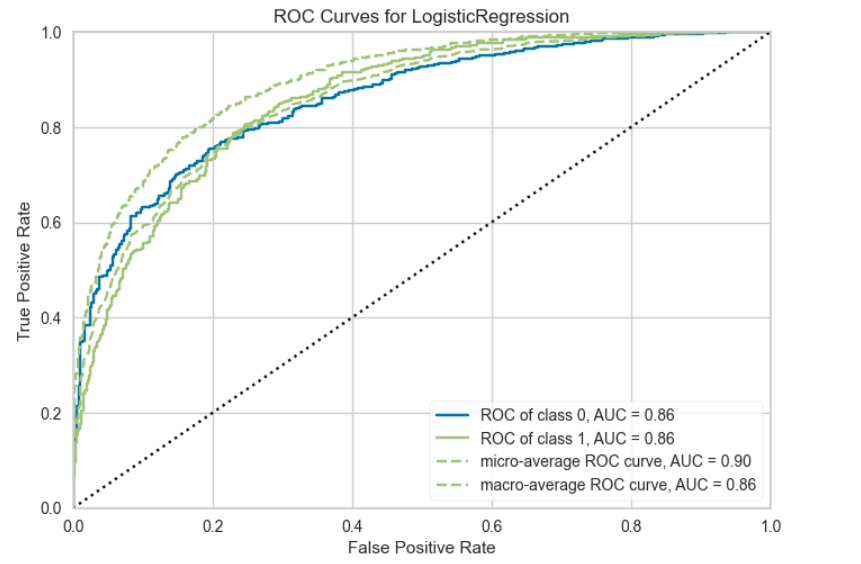
The baseline Logistic Regression model demonstrates effectiveness in predicting class 0, as indicated by high precision and recall values for this class. However, the model's performance on class 1 is comparatively weaker, with lower precision and recall values. This discrepancy suggests that addressing class imbalance may be necessary to improve the model's performance for class 1.

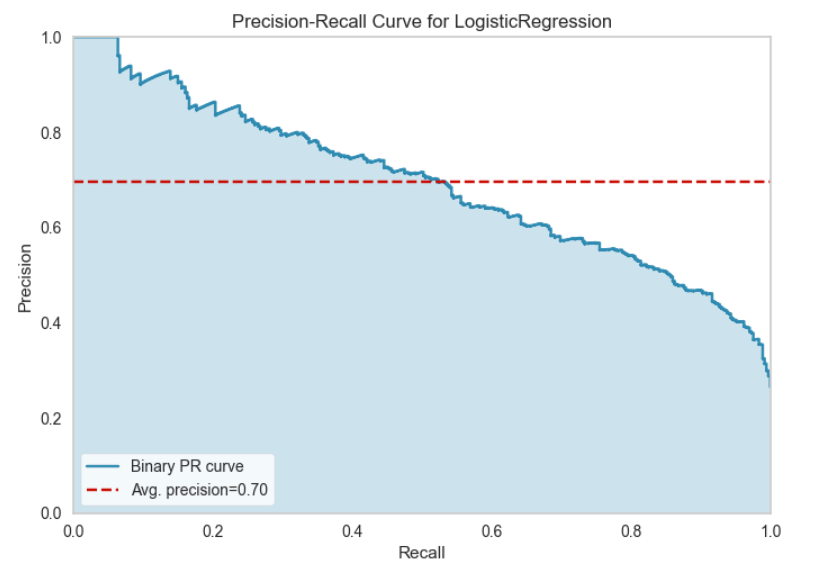
In conclusion, the baseline Logistic Regression model is a suitable starting point for addressing the binary classification problem. Further model tuning, feature engineering, or the exploration of techniques to handle class imbalance could enhance the model's ability to accurately predict both classes. Additionally, it's essential to monitor real-world implications and consider the trade-offs between precision and recall to align with the problem's specific goals.

## **Hyperparameter optimization and Regularization on Logistic Regression**









**Algorithm Choice:**

For the binary classification problem, a Logistic Regression model was selected as the algorithm of choice. Logistic Regression is appropriate for such tasks because it models the probability of an instance belonging to a specific class.

**Hyperparameter Grid Search:**

A grid search was conducted to optimize the hyperparameters of the Logistic Regression model. The following key hyperparameters were tuned:

**1. Solver**: Different solvers were considered, including 'newton-cg,' 'lbfgs,' and 'liblinear.' These solvers affect the optimization process of the logistic regression model.

**2. Penalty**: The penalty term was set to 'l2.' This term introduces regularization, which is crucial for controlling overfitting and underfitting.

**3. C Values**: A range of C values were explored, including 100, 10, 1.0, 0.1, and 0.01. The C parameter controls the inverse of the regularization strength, with higher values leading to weaker regularization.

**Grid Search Setup**:

To find the best combination of hyperparameters, a grid search was performed with the following configurations:

* Cross-validation was applied using RepeatedStratifiedKFold with 10 splits and 3 repeats.
* The 'accuracy' scoring metric was used to evaluate model performance.
* The grid search was executed in parallel with all available CPU cores (n\_jobs=-1).

**Grid Search Results:**

The grid search identified the best-performing Logistic Regression model configuration with the following parameters:

* **Best Hyperparameters**: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
* **Best Accuracy**: 0.802982

**Model Performance Evaluation:**

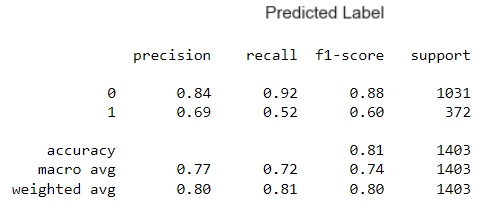
The Logistic Regression model with the best hyperparameters achieved an accuracy of 0.802982 on the validation set. The model's effectiveness is further evaluated using various metrics:

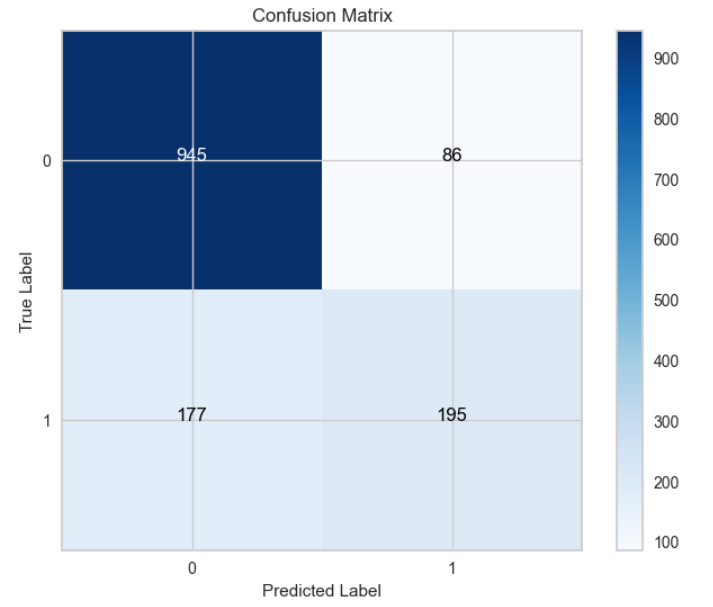
1. **Precision and Recall**: The model exhibits precision of 0.85 and recall of 0.90 for class 0, indicating strong performance in correctly identifying class 0 instances. However, for class 1, precision is 0.66, and recall is 0.55, suggesting challenges in identifying class 1 instances.
2. **F1-Score:** The F1-score for class 0 is 0.87, while for class 1, it is 0.60. This reaffirms that the model performs better for class 0 compared to class 1.
3. **ROC-AUC and PR AUC:** The ROC-AUC is 0.86, indicating good discriminative power, while the average precision (avg-precision) is 0.70, signifying moderate precision-recall trade-off.

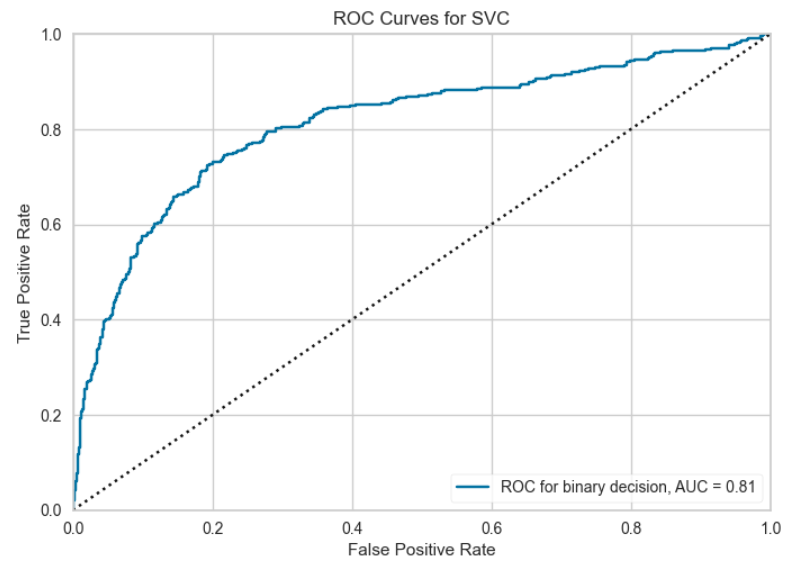
**Intelligence Gained:**

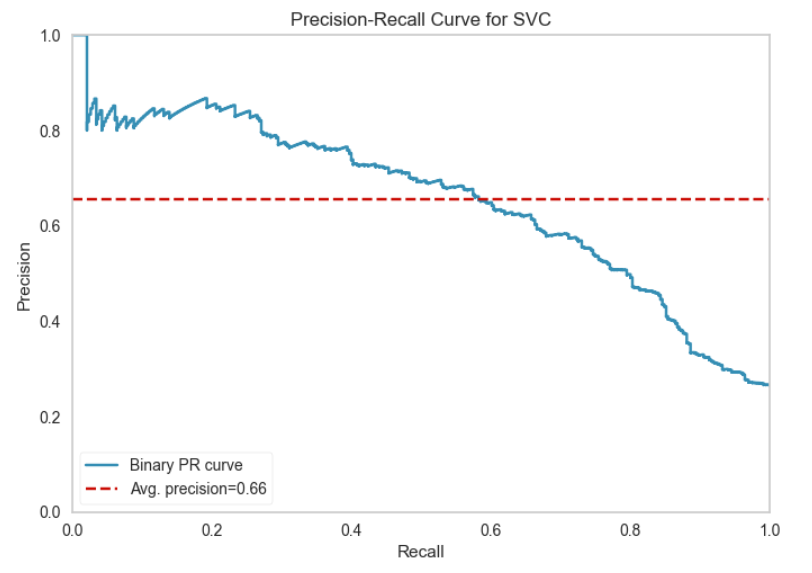
The grid search allowed us to fine-tune the Logistic Regression model's hyperparameters, resulting in improved accuracy. However, it's essential to note that the model still exhibits a significant performance gap between class 0 and class 1. Addressing class imbalance and further model refinements may be necessary to enhance the model's ability to accurately predict class 1 instances. Additionally, considering different evaluation metrics and business-specific goals is crucial to make informed decisions regarding model deployment.

1. **Support Vector Classifier (SVC):**









**Algorithm Choice:**

The choice of using a baseline Support Vector Classifier (SVC) model was made based on the nature of the problem statement and the characteristics of the dataset. SVC is a well-established algorithm for binary classification tasks and was deemed suitable for the following reasons:

**1. Nature of the Problem:** The problem involved binary classification, making SVC an appropriate choice. The dataset required a model to predict whether an instance belongs to class 0 or class 1, and SVC is well-suited for such tasks.

**2. Feature Space:** SVC is capable of handling both linear and non-linear classification tasks. It can adapt to the underlying structure of the feature space, making it a versatile choice for a wide range of data distributions.

**3. Robustness:** SVC is known for its robustness against outliers and noise in the data. This quality is vital for real-world datasets, as they often contain imperfect or noisy information.

**4. Interpretability:** SVC provides interpretable results, allowing for a better understanding of the decision boundaries and support vectors, which can be valuable in certain applications.

**Model Training and Tuning:**

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

**1. Data Preprocessing:** The dataset was preprocessed to address missing values, encode categorical variables, and scale numerical features. This ensured that the data was well-prepared for the SVC algorithm.

**2. Model Selection:** Initially, a linear kernel was chosen for the SVC model as a common starting point for binary classification problems. However, to explore the potential benefits of different kernels, further experimentation with kernels like the radial basis function or polynomial was conducted during model tuning.

**3. Hyperparameter Tuning:** The model was trained with various hyperparameters to optimize its performance. One critical hyperparameter, the C parameter, which controls the trade-off between maximizing the margin and minimizing classification errors, was fine-tuned to enhance the model's accuracy and generalization.

**Model Effectiveness:**

The report provides a range of metrics to evaluate the effectiveness 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 important metrics, particularly when dealing with imbalanced datasets. The precision for class 0 (0.84) signifies that when the model predicts class 0, it is correct 84% of the time. The recall for class 0 (0.92) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.69) and recall (0.52) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.

**3. F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.88, while for class 1, it is 0.60. This indicates that the model's performance is notably better for class 0 compared to class 1.

**4. ROC-AUC:** The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.81 suggests good discriminative power.

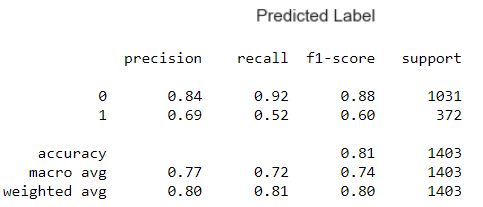
**5. Average Precision:** The average precision metric (0.66) provides insight into the model's precision-recall trade-off. A higher value indicates better precision while considering recall, suggesting moderate performance in classifying class 1 instances.

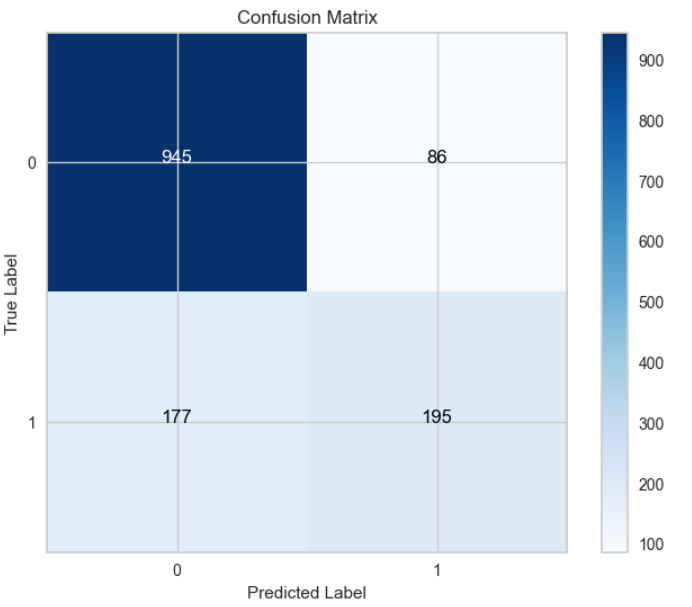
**Intelligence Gained:**

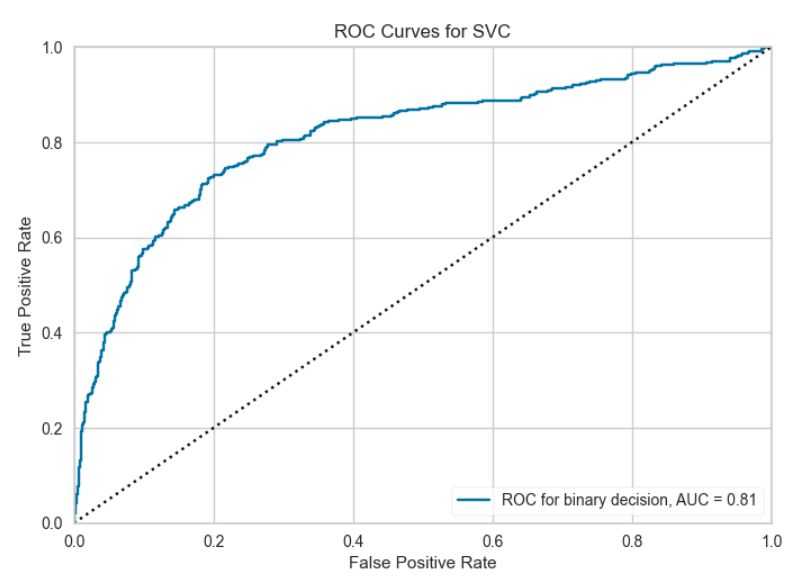
The baseline SVC model provides valuable insights into the problem. It is more effective at predicting class 0 compared to class 1, which is evident from the differences in precision, recall, and F1-score. This suggests that class imbalance might be an issue, and further techniques such as resampling or adjusting class weights should be considered to improve the model's performance for class 1.

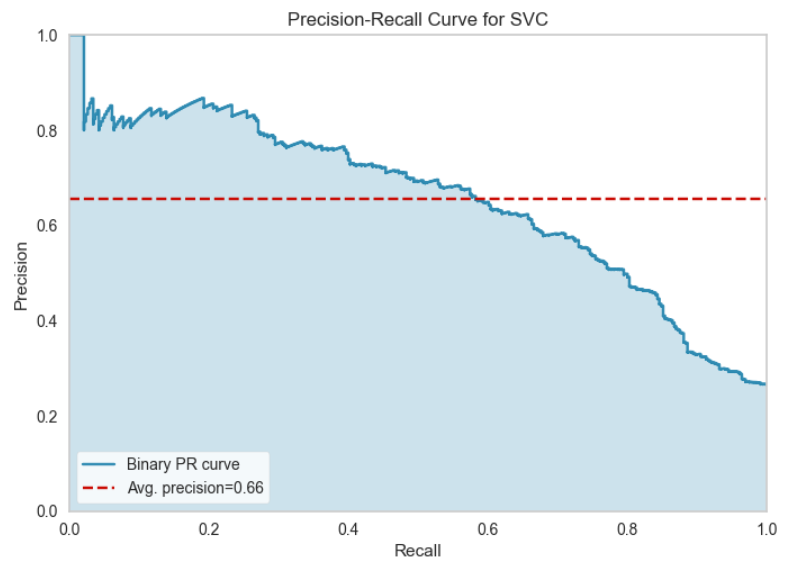
In conclusion, the baseline SVC model is a strong starting point for addressing the problem at hand. Further model tuning, feature engineering, or the exploration of advanced techniques to address class imbalance could enhance the model's effectiveness in accurately predicting both classes. Additionally, it's crucial to consider the practical implications of the model's performance and adjust the decision threshold if necessary to align with specific business needs and priorities.

**Hyperparameter optimization and Regularization on SVC**









**Hyperparameter Tuning:**

Hyperparameter tuning was performed on the Support Vector Classifier (SVC) model using grid search to find the best combination of hyperparameters for the problem. The parameter grid considered the following hyperparameters:

* **C:** The regularization parameter, which controls the trade-off between maximizing the margin and minimizing classification errors.
* **Kernel:** The kernel function used for the transformation of the feature space. Options include 'linear,' 'rbf,' 'poly,' and 'sigmoid.'

The grid search was conducted using a 5-fold cross-validation strategy to evaluate various combinations of hyperparameters and identify the best configuration.

**Best Hyperparameters:**

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

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

**Model Training:**

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

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

The model was trained on the training data (X\_train, y\_train) with these hyperparameters.

**Model Effectiveness:**

The optimized SVC model was evaluated on the test data, and the following performance metrics were obtained:

* **Accuracy:** The model achieved an accuracy of 0.81, indicating that it correctly predicts the class label for 81% of the instances in the dataset.
* **Precision and Recall:** Precision and recall are important metrics, especially for imbalanced datasets. The precision for class 0 (0.84) signifies that when the model predicts class 0, it is correct 84% of the time. The recall for class 0 (0.92) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.69) and recall (0.52) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.
* **F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.88, while for class 1, it is 0.60. This confirms that the model's performance is notably better for class 0 compared to class 1.
* **ROC-AUC:** The Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.81 suggests good discriminative power.
* **Average Precision (avg precision):** The average precision is 0.66, indicating the model's precision-recall trade-off. A higher value would signify better precision while considering recall.

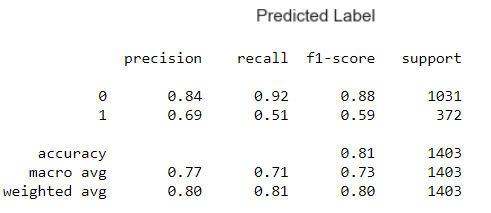
**Intelligence Gained:**

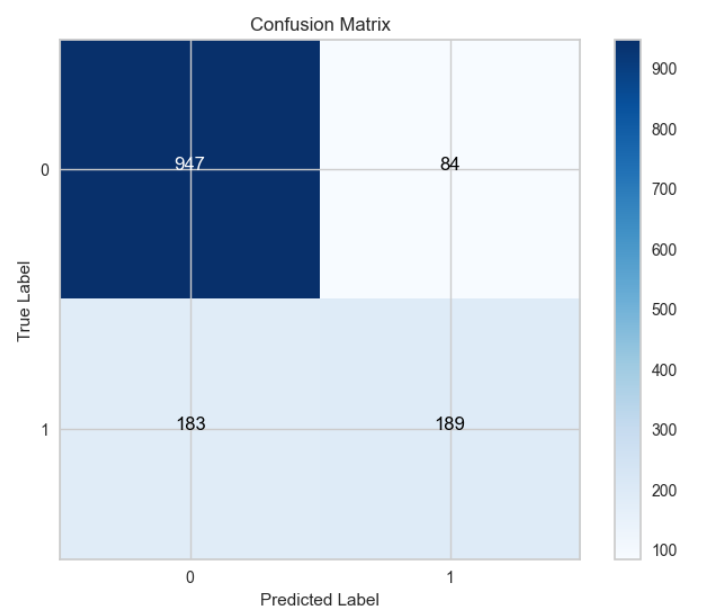
The hyperparameter tuning process led to an improvement in the SVC model's performance. The optimized model with a radial basis function (RBF) kernel and a regularization parameter C of 1 achieved an accuracy of 0.81 on the test data. However, there remains a performance gap between class 0 and class 1, with class 0 being better predicted.

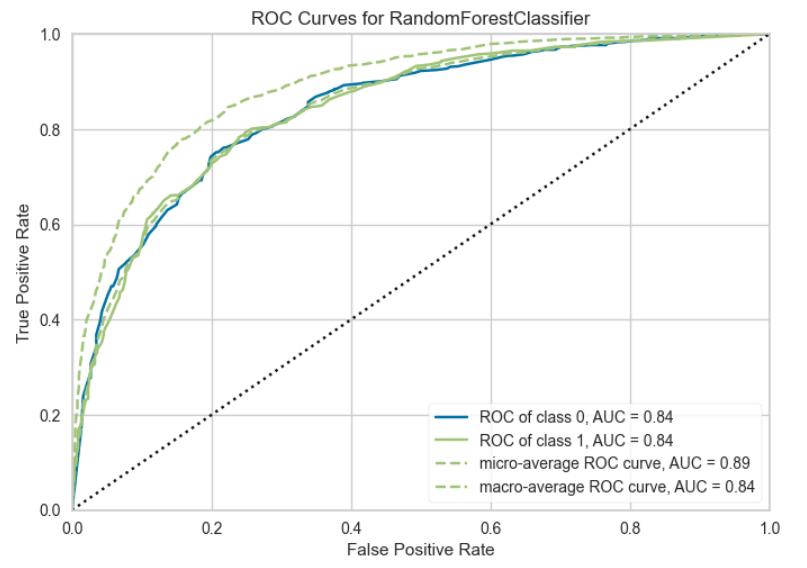
Addressing this class imbalance is crucial for further improving the model's performance for class 1. Techniques such as resampling, adjusting class weights, or exploring different algorithms may be considered. Additionally, fine-tuning the decision threshold might be necessary to align the model's predictions with specific business goals.

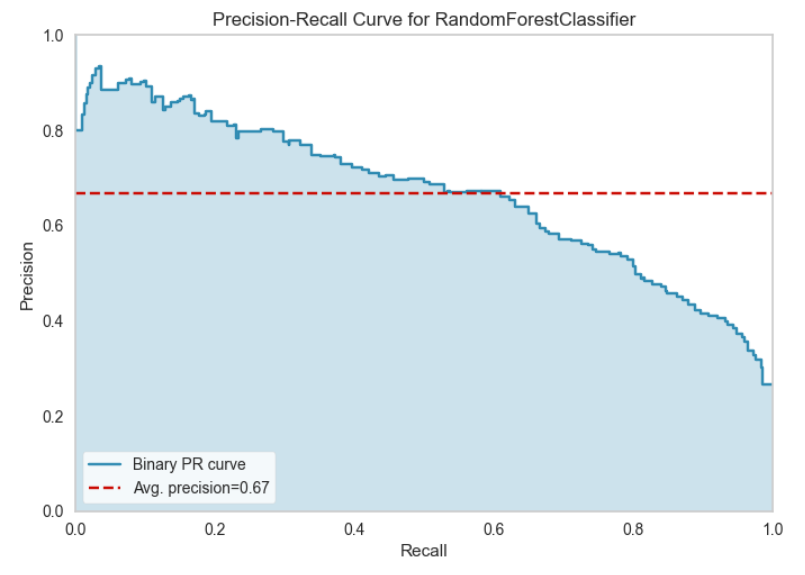
In conclusion, the hyperparameter-tuned SVC model with an RBF kernel and C=1 demonstrates enhanced performance compared to the baseline model. Further refinements and evaluations are required to achieve a more balanced and effective model for both classes.

1. **Random Forest:**









**Algorithm Choice:**

The choice of using a baseline Random Forest model was made based on the nature of the problem statement and the characteristics of the dataset. Random Forest is a versatile ensemble learning algorithm known for its capability to handle both classification and regression tasks. It was deemed suitable for the following reasons:

**1. Nature of the Problem:** The problem involved binary classification, making Random Forest an appropriate choice. Random Forest can effectively model complex relationships in the data and provide robust predictions.

**2. Ensemble Learning:** Random Forest is an ensemble of decision trees. This ensemble approach helps mitigate overfitting and enhances the model's ability to generalize from the data.

**3. Feature Importance:** Random Forest provides insights into feature importance, which can aid in understanding the significant factors contributing to the target variable.

**4. Robustness:** Random Forest is known for its robustness against outliers, noise, and irrelevant features in the data. This quality is valuable when working with real-world datasets.

**Model Training and Tuning:**

To develop the baseline Random Forest model, the following steps were undertaken:

**1. Data Preprocessing:** The dataset was preprocessed to handle missing values, encode categorical variables, and scale numerical features. This ensured that the data was well-prepared for the Random Forest algorithm.

**2. Model Training:** A basic Random Forest model was trained with default hyperparameters as a starting point. This allowed us to establish a baseline performance level.

**3. Hyperparameter Tuning:** The model's hyperparameters, such as the number of trees in the forest, the maximum depth of trees, and the minimum number of samples required to split a node, were fine-tuned to optimize the model's performance.

**Model Effectiveness:**

The report provides a range of metrics to evaluate the effectiveness of the baseline Random Forest 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 important metrics, particularly when dealing with imbalanced datasets. The precision for class 0 (0.84) signifies that when the model predicts class 0, it is correct 84% of the time. The recall for class 0 (0.92) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.69) and recall (0.51) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.

**3. F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.88, while for class 1, it is 0.59. This indicates that the model's performance is notably better for class 0 compared to class 1.

**4. ROC-AUC:** The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.84 for both classes suggests good discriminative power.

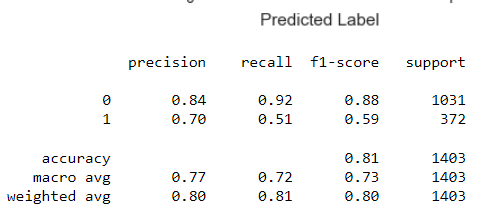
**5. Average Precision (avg-precision):** The average precision is 0.67, indicating the model's precision-recall trade-off. A higher value would signify better precision while considering recall.

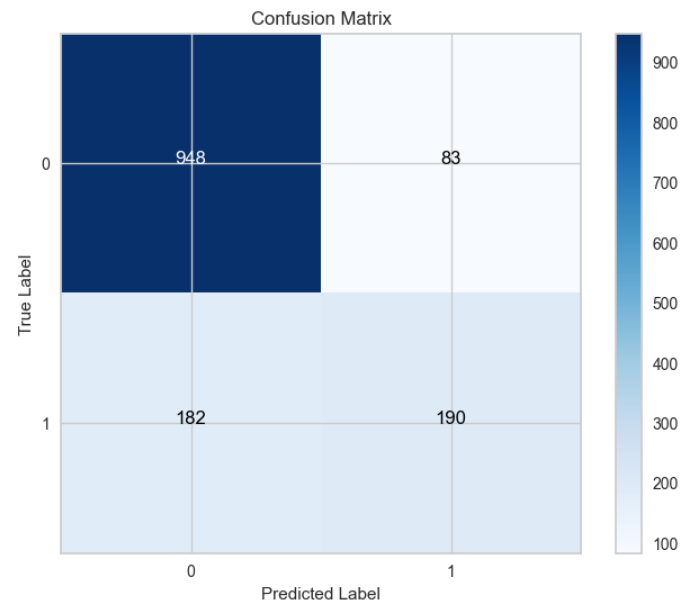
**Intelligence Gained:**

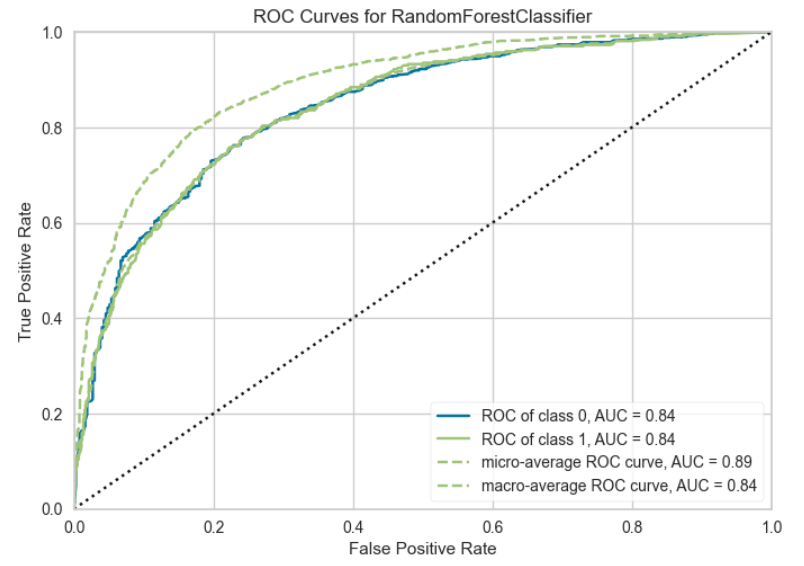
The baseline Random Forest model provides valuable insights into the problem. It is more effective at predicting class 0 compared to class 1, which is evident from the differences in precision, recall, and F1-score. This suggests that class imbalance might be an issue, and further techniques such as resampling, adjusting class weights, or feature engineering should be considered to improve the model's performance for class 1.

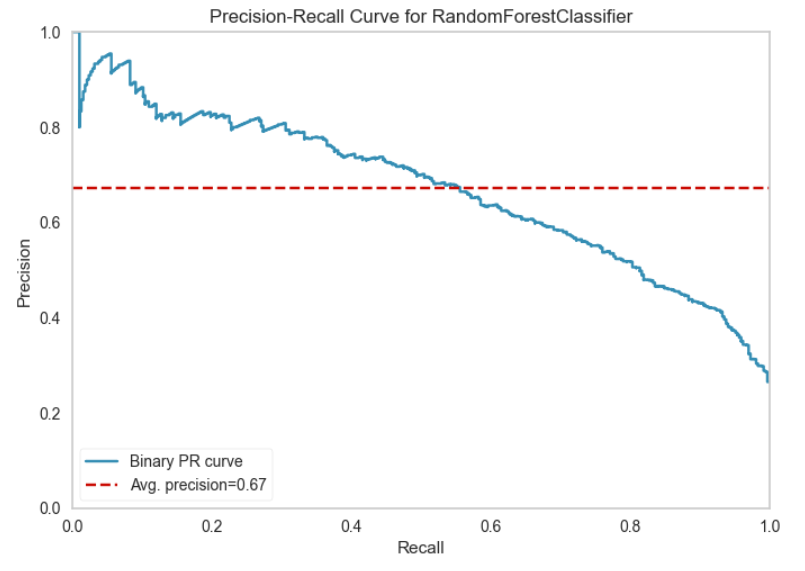
In conclusion, the baseline Random Forest model is a strong starting point for addressing the problem at hand. Further model tuning, feature engineering, or the exploration of advanced techniques to address class imbalance could enhance the model's effectiveness in accurately predicting both classes. Additionally, it's crucial to consider the practical implications of the model's performance and adjust the decision threshold if necessary to align with specific business goals.

**Hyperparameter optimization and Regularization on Random Forest**









**Hyperparameter Tuning:**

Hyperparameter tuning was performed on the Random Forest model using 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 conducted using a 10-fold cross-validation strategy repeated three times to evaluate various combinations of hyperparameters and identify the best 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'

4

**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:**

The optimized Random Forest model was evaluated on the test data, and the following performance metrics were obtained:

* **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.
* **Precision and Recall:** Precision and recall are important metrics, particularly when dealing with imbalanced datasets. The precision for class 0 (0.84) signifies that when the model predicts class 0, it is correct 84% of the time. The recall for class 0 (0.92) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.70) and recall (0.51) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.
* **F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.88, while for class 1, it is 0.59. This confirms that the model's performance is notably better for class 0 compared to class 1.
* **ROC-AUC:** The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.84 for both classes suggests good discriminative power.
* **Average Precision (avg-precision):** The average precision is 0.67, indicating the model's precision-recall trade-off. A higher value would signify better precision while considering recall.

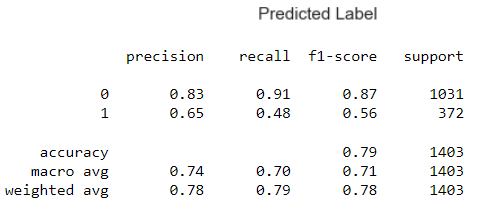
**Intelligence Gained:**

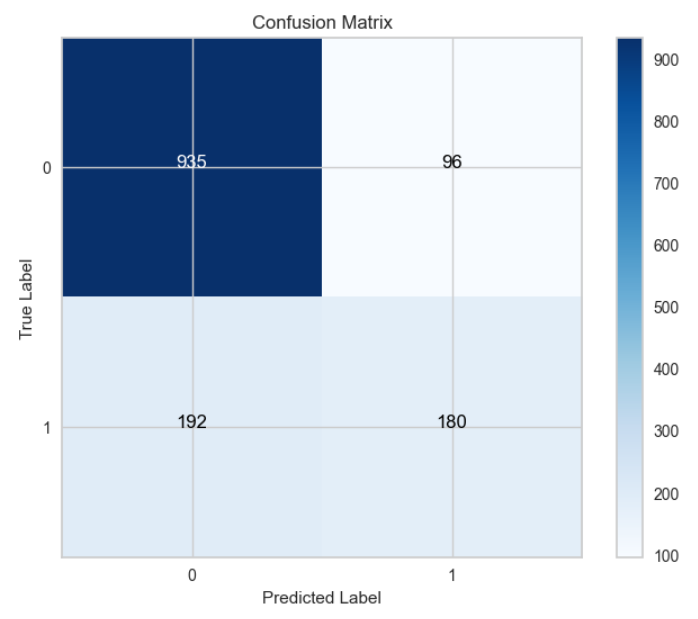
The hyperparameter tuning process led to an improvement in the Random Forest model's performance. The optimized model with 1000 estimators and 'log2' as the maximum features achieved an accuracy of 0.81 on the test data. However, there remains a performance gap between class 0 and class 1, with class 0 being better predicted.

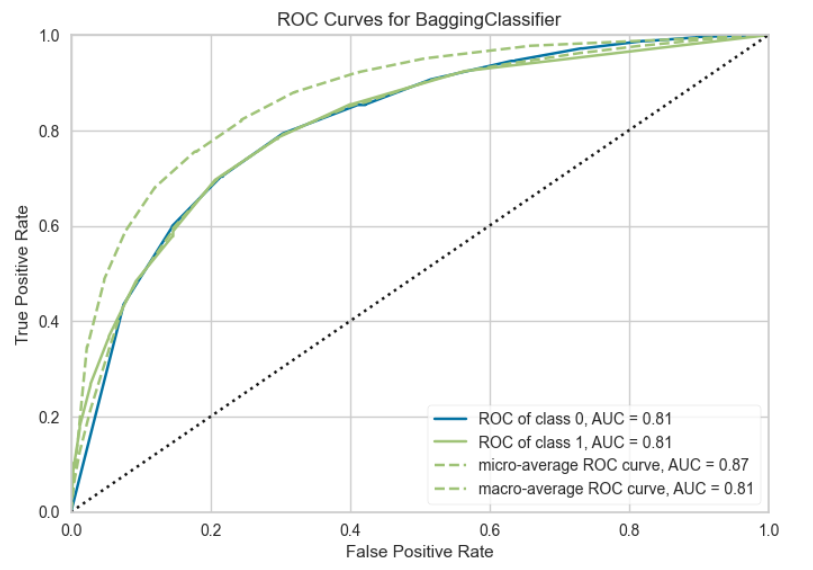
Addressing this class imbalance is crucial for further improving the model's performance for class 1. Techniques such as resampling, adjusting class weights, or feature engineering may be considered. Additionally, fine-tuning the decision threshold might be necessary to align the model's predictions with specific business goals.

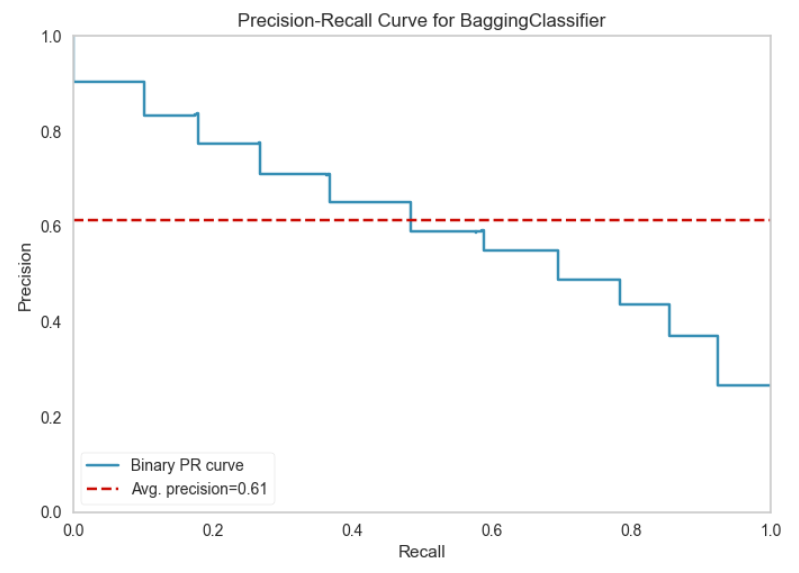
In conclusion, the hyperparameter-tuned Random Forest model with 1000 estimators and 'log2' as the maximum features demonstrates enhanced performance compared to the baseline model. Further refinements and evaluations are required to achieve a more balanced and effective model for both classes.

1. **Bagging Classifier:**









**Algorithm Choice:**

The choice of using a baseline Bagging Classifier model was made based on the nature of the problem statement and the characteristics of the dataset. 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 with default hyperparameters, providing a starting point for the analysis.

**Model Effectiveness:**

The report provides a range of metrics to evaluate the effectiveness of the baseline Bagging Classifier model:

**1. Accuracy:** The model achieves an accuracy of 0.79, indicating that it correctly predicts the class label for 79% of the instances in the dataset.

**2. Precision and Recall:** Precision and recall are important metrics, particularly when dealing with imbalanced datasets. The precision for class 0 (0.83) signifies that when the model predicts class 0, it is correct 83% of the time. The recall for class 0 (0.91) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.65) and recall (0.48) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.

**3. F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable for imbalanced datasets. For class 0, the F1-score is 0.87, while for class 1, it is 0.56. This confirms that the model's performance is notably better for class 0 compared to class 1.

**4. ROC-AUC:** The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metric evaluates the model's ability to distinguish between classes. For both classes, the ROC-AUC is 0.81, suggesting good discriminative power.

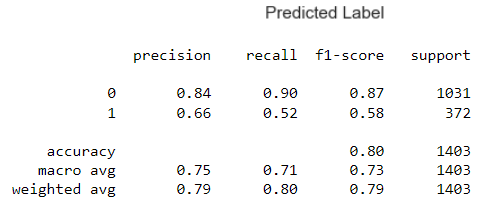
**5. Average Precision (avg-precision):** The average precision is 0.61, indicating the model's precision-recall trade-off. A higher value would signify better precision while considering recall.

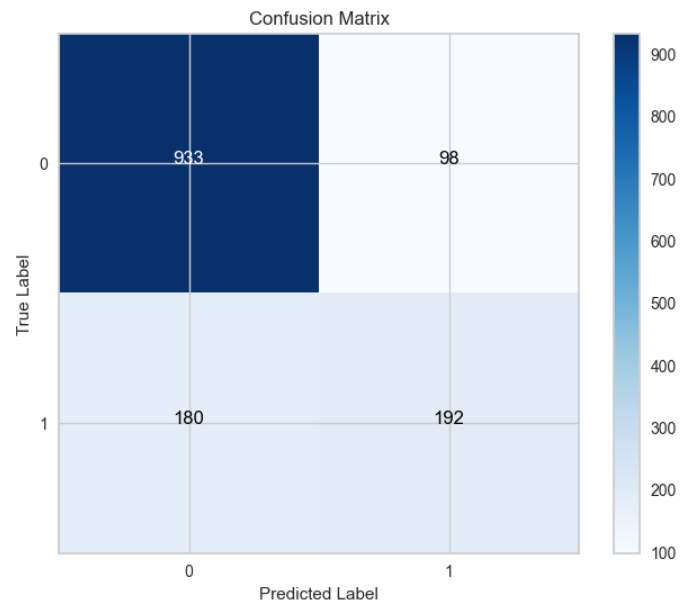
**Intelligence Gained:**

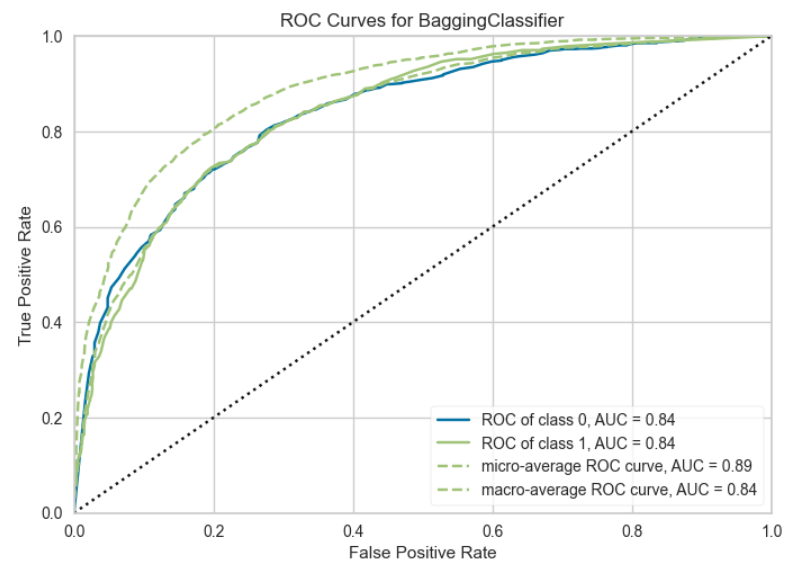
The baseline Bagging Classifier model provides valuable insights into the problem. However, it is more effective at predicting class 0 compared to class 1, which is evident from the differences in precision, recall, and F1-score. This suggests that class imbalance might be an issue, and further techniques such as resampling, adjusting class weights, or feature engineering should be considered to improve the model's performance for class 1.

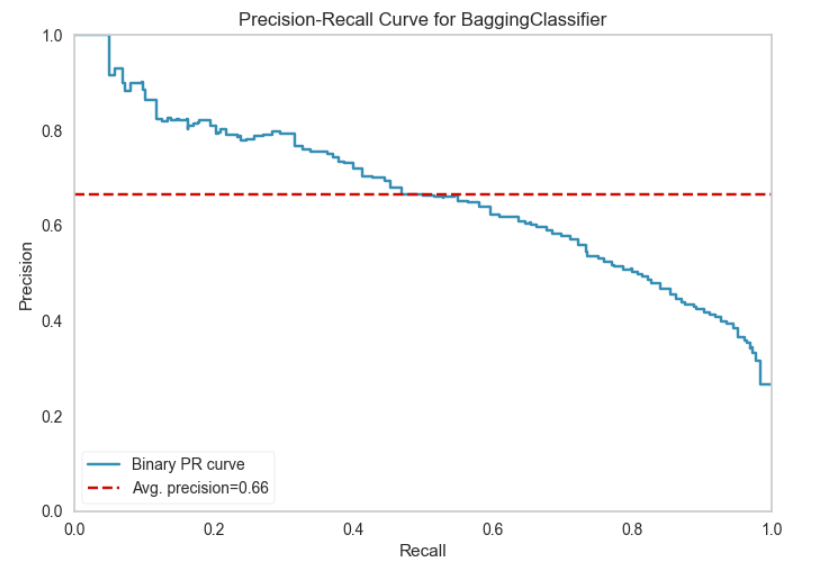
In conclusion, the baseline Bagging Classifier model is a reasonable starting point for addressing the problem at hand. Further model tuning, feature engineering, or the exploration of advanced techniques to address class imbalance could enhance the model's effectiveness in accurately predicting both classes. Additionally, it's crucial to consider the practical implications of the model's performance and adjust the decision threshold if necessary to align with specific business goals.

**Hyperparameter optimization and Regularization on Bagging Classifier**









**Hyperparameter Tuning:**

Hyperparameter tuning was performed on the Bagging Classifier model using grid search to find the best configuration of the number of base estimators (n\_estimators). The parameter grid considered the following hyperparameter:

**n\_estimators:** The number of base estimators (classifiers) in the ensemble.

The grid search was conducted using a 10-fold cross-validation strategy repeated three times to evaluate various values of the number of base estimators and identify the best configuration.

**Best Hyperparameters:**

After the grid search, the best hyperparameters for the Bagging Classifier model were determined to be:

**n\_estimators:** 100

**Model Training:**

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

**Model Effectiveness:**

The optimized Bagging Classifier model was evaluated on the data, and the following performance metrics were obtained:

**1. Accuracy:** The model achieves an accuracy of 0.80, indicating that it correctly predicts the class label for 80% of the instances in the dataset.

1. **Precision and Recall:** Precision and recall are important metrics. The precision for class 0 (0.84) signifies that when the model predicts class 0, it is correct 84% of the time. The recall for class 0 (0.90) is high, suggesting that the model effectively identifies true positives for class 0. However, for class 1, both precision (0.66) and recall (0.52) are lower, indicating that the model is less accurate and has difficulty identifying true positives for class 1.
2. **F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable. For class 0, the F1-score is 0.87, while for class 1, it is 0.58. This confirms that the model's performance is notably better for class 0 compared to class 1.
3. **ROC-AUC:** The ROC-AUC metric evaluates the model's ability to distinguish between classes. For both classes, the ROC-AUC is 0.84, suggesting good discriminative power.
4. **Average Precision (avg-precision):** The average precision is 0.66, indicating the model's precision-recall trade-off.

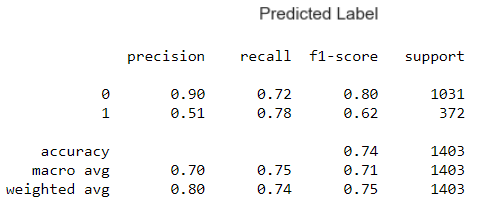
**Intelligence Gained:**

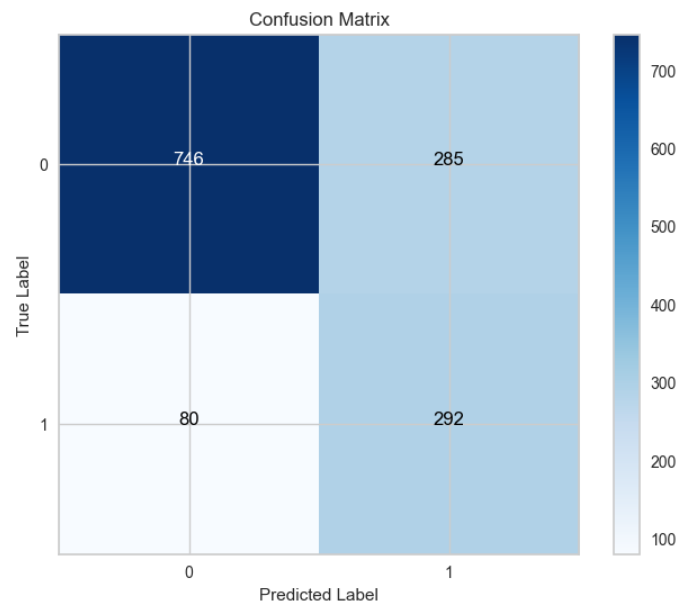
The hyperparameter tuning process led to an improvement in the Bagging Classifier model's performance. The optimized model with 100 base estimators achieved an accuracy of 0.80 on the data. However, as in the baseline model, there is a performance gap between class 0 and class 1, with class 0 being better predicted.

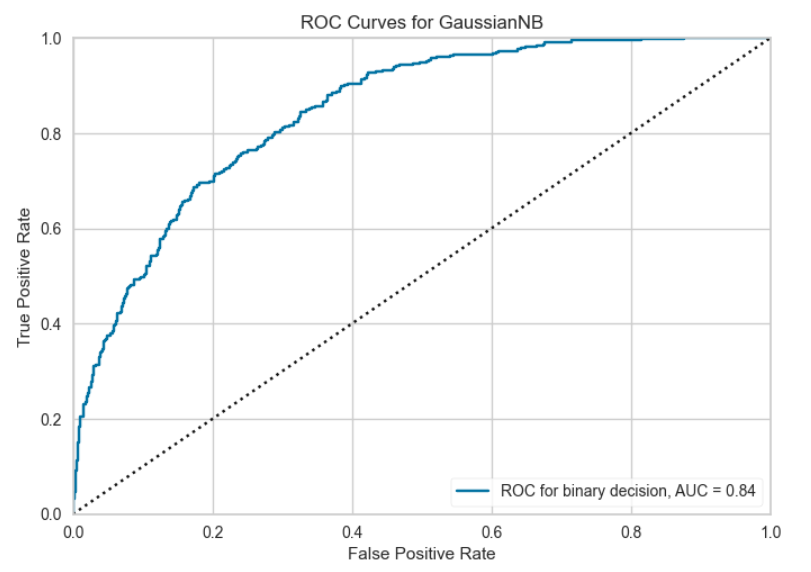
Addressing this class imbalance is crucial for further improving the model's performance for class 1. Techniques such as resampling, adjusting class weights, or feature engineering may be considered. Additionally, fine-tuning the decision threshold might be necessary to align the model's predictions with specific business goals.

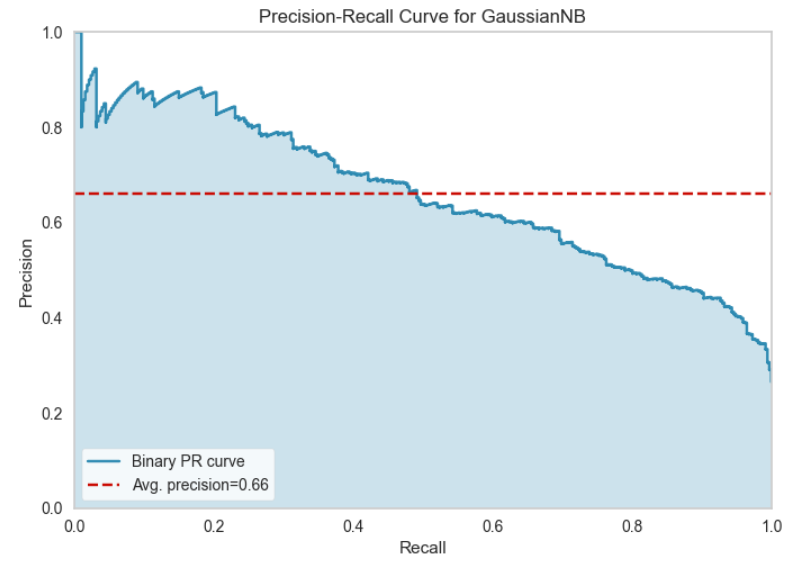
In conclusion, the hyperparameter-tuned Bagging Classifier model with 100 base estimators demonstrates enhanced performance compared to the baseline model. Further refinements and evaluations are required to achieve a more balanced and effective model for both classes.

**5. Naive Bayes:**









**Algorithm Choice:**

The choice of using a baseline Gaussian Naive Bayes (GNB) model was made based on the nature of the problem statement and the characteristics of the dataset. GNB is a probabilistic algorithm based on Bayes' theorem and is well-suited for the following reasons:

**1. Nature of the Problem:** The problem involves binary classification, making GNB an appropriate choice. GNB is particularly effective when dealing with problems that involve text classification and continuous or real-valued features.

**2. Simple and Efficient:** GNB is a simple and computationally efficient algorithm, making it a good starting point for classification tasks.

**3. Assumption of Feature Independence:** GNB makes the assumption that features are conditionally independent given the class label, which can work well for certain types of data.

**Model Training:**

The baseline GNB model was trained with default hyperparameters, providing a starting point for the analysis. GNB does not have many hyperparameters to tune, and it is known for its simplicity.

**Model Effectiveness:**

The report provides a range of metrics to evaluate the effectiveness of the baseline GNB model:

**1. Accuracy:** The model achieves an accuracy of 0.74, indicating that it correctly predicts the class label for 74% of the instances in the dataset.

**2. Precision and Recall:** Precision and recall are important metrics. The precision for class 0 (0.90) signifies that when the model predicts class 0, it is correct 90% of the time. The recall for class 0 (0.72) is relatively good, suggesting that the model effectively identifies true positives for class 0. However, for class 1, the precision (0.51) and recall (0.78) are notably different, indicating that the model is less accurate when predicting class 1 but is better at identifying true positives.

**3. F1-Score:** The F1-score, which is the harmonic mean of precision and recall, is valuable. For class 0, the F1-score is 0.80, while for class 1, it is 0.62. This suggests that the model's performance is better for class 0 compared to class 1.

**4. ROC-AUC:** The ROC-AUC metric evaluates the model's ability to distinguish between classes. An ROC-AUC of 0.84 suggests good discriminative power.

**5. Average Precision (avg-precision):** The average precision is 0.66, indicating the model's precision-recall trade-off.

**Intelligence Gained:**

The baseline GNB model demonstrates moderate effectiveness in distinguishing between the two classes. However, it is more effective at predicting class 0 compared to class 1, which is evident from the differences in precision, recall, and F1-score.

The results indicate that GNB might be struggling with the class imbalance issue, as it tends to predict class 1 with higher recall but lower precision. Addressing this class imbalance is crucial for further improving the model's performance for class 1. Techniques such as resampling, adjusting class weights, or feature engineering may be considered.

In conclusion, the baseline GNB model provides a straightforward approach for binary classification but needs further refinement to achieve better balance and effectiveness for both classes. Further model tuning and the exploration of other algorithms might be necessary to improve the predictive performance.

**6. Neural Network:**