**Machine Learning for Amazon Product Report**

Kaggle name: Caixu Chen

Dartmouth College

Hanover, NH 03755

caixu.chen.th@dartmouth.edu

**Abstract**

*This report presents the findings of a course project aimed at implementing machine learning models and concepts for analyzing and categorizing product reviews using the Amazon product review dataset. The project focuses on three main tasks: binary classification, multiclass classification, and clustering. In the binary classification task, the objective is to develop classifiers to classify product reviews as either good or bad. The cutoff point for determining "goodness" is an input parameter, and classifiers will be designed for cutoff values of 1, 2, 3, and 4. The report evaluates the performance of at least three different classifiers for each cutoff, employing cross-validation for hyperparameter tuning. The evaluation metrics include the confusion matrix, ROC curve, AUC, macro F1 score, and accuracy. It is required that at least one classifier achieves a macro F1 score equal to or higher than the provided baseline score. For the multiclass classification task, the binary classifiers developed in the previous task are extended to classify product ratings on a five-class scale (1 to 5). Similar to the binary classification, at least three classifiers are evaluated, and the performance is measured using metrics such as the confusion matrix, ROC curve, AUC, macro F1 score, and accuracy. Again, one classifier must achieve a macro F1 score equal to or higher than the provided baseline score. In the clustering task, product reviews in the test dataset are clustered based on word features derived from the data. The k-means clustering algorithm is utilized, with product categories serving as labels. The quality of clustering is assessed using the Silhouette score and Rand index. It is required that the model achieves a Silhouette score equal to or higher than the provided baseline score. The report provides a detailed procedure for hyperparameter tuning for all classification tasks, including the computation of 5-fold cross-validation scores for different combinations of hyperparameters. The best variety of hyperparameters is selected based on these scores, which are reported along with explanations. Using the chosen hyperparameters, the best models are evaluated using various metrics and presented on validation sets. In the case of multiclass classification, the report includes six curves (five category curves and the average curve) in a single plot. The study also compares baseline scores and reports the best test scores achieved in the Kaggle competitions. This project aims to demonstrate the implementation and evaluation of machine learning models for analyzing product reviews, providing insights into binary classification, multiclass classification, and clustering tasks using the Amazon product review dataset.*

# **Introduction**

Amazon needs to assess the factors that influence consumers to make consumption decisions. The background of this project is to analyze and classify product reviews by implementing machine learning models and concepts using the Amazon product review dataset. The project focuses on three tasks: binary classification, multiclass classification, and clustering. The report details the hyperparameter tuning process for all classification tasks, including calculating 5-fold cross-validation scores for different hyperparameter combinations. Based on these scores, the best combination of hyperparameters is chosen, and these scores are reported with corresponding explanations. Using selected hyperparameters, the best model on the validation set is evaluated and displayed using various metrics. In the case of multiclass classification, the report will contain six curves (five class curves and a mean curve) in one plot. The study also compares benchmark scores and reports the best test scores achieved in Kaggle competitions. By analyzing product review datasets, it is possible to gain insights into consumer perceptions, preferences, and product needs. This can help Amazon understand consumers' concerns about different product features, quality, price, etc., to meet consumer needs better and provide personalized product recommendations. Amazon can also learn about the strengths and weaknesses of competitors' products in the market. This can help Amazon optimize its product features, market positioning, and differentiation strategies to maintain a competitive advantage in a highly competitive market.

## **Related Work**

Big data analysis can dig deep into consumers' purchasing history, preferences, interests, and behavioral data. By analyzing this data, sellers can understand consumers' unique needs and preferences and provide personalized product recommendations, offers, and services. This customized experience can meet the individual needs of consumers and increase their satisfaction and loyalty. Big data analytics also enables real-time monitoring and analysis of consumer behavior and feedback, such as purchasing behavior, online reviews, and social media interactions. Real-time analysis of these data can help sellers quickly understand consumer reactions and trends to adjust marketing strategies, product pricing, and inventory management decisions in a timely manner to meet consumer needs better and improve sales performance. In addition, big data can also help sellers predict future trends and demand so that they can better plan their inventory and production.[[1]](#footnote-0)

## **Data Preprocessing**

During the data preparation phase of this project, the main goal was to read and preprocess text data and convert it into a form that could be used in a logistic regression model. This includes reading data, handling missing values, label generation, text preprocessing, feature extraction, feature selection, and data segmentation. Specific steps are as follows:

Data reading: First, two CSV files are read, one of which is training data, and the other is test data.

Handling of missing values: Then, missing values in the two fields "summary" and "vote" are handled. Missing values of the "summary" field are replaced with an empty string while missing values of the "vote" field are replaced with 0.

Label Generation: Next, labels are generated for the training data. The rule defined here is that if the value of the "overall" field is less than or equal to 4, the label is 0. Otherwise, it is 1. The label is set to -1 for the test data as a placeholder.

Text preprocessing: After that, a text preprocessing function is defined to perform a series of processing on the text, including converting to lowercase, removing punctuation marks and numbers, removing redundant spaces and newlines, removing stop words, etc.

Feature Extraction: Next, use ColumnTransformer to combine text data and other relevant variables and perform feature extraction. Here, the TF-IDF vectorizer is used to vectorize the two fields "reviewText" and "summary," SelectKBest is used to select the best features. At the same time, the "verified" field is one-hot encoded, and the "vote" field is left as it is.

Data Splitting: The training data is split into training and validation sets.

Feature selection: Next, apply SelectKBest to the features generated in the previous step for feature selection.

Feature Merging: Finally, the best features obtained by SelectKBest and the features of other related variables are combined to get the final training, validation, and testing features.

These steps aim to extract valuable information for predicting the target variable and provide this information in a form suitable for model learning. After this series of steps, the data is transformed into a format suitable for input to a machine-learning model.

## **Binary Classification**

Binary classification, a central tenet of machine learning, is a process that segregates data into one of two potential classes or categories. The current study's primary objective is to differentiate between positive and adverse product reviews by leveraging thresholds ranging from 1 to 4. The ultimate goal is to craft an efficient classifier capable of accurately deciphering a review's sentiment from text and additional variables.

This research project deploys a minimum of three distinct classifiers for executing the classification tasks. The modus operandi for each classifier comprises determining the relevant hyperparameters for fine-tuning and calculating a 5-fold cross-validation score for each possible combination. The mean of the five-fold validation scores is ascertained for every variety, and the optimal set of hyperparameters is selected based on these cross-validation scores. Several metrics, such as the confusion matrix, ROC, AUC, macro F1 score, and accuracy, are then utilized to report the performance of the best model equipped with the selected hyperparameters.

Different cutoff values serve as a benchmark to classify reviews into positive and negative groups to evaluate the classifiers' effectiveness in identifying review sentiment. The classifiers' proficiency is gauged using the macro F1 score, which considers precision and recall. The outcomes are juxtaposed with a baseline macro F1 score to further evaluate the classifiers' prowess in distinguishing between good and poor reviews. These performance metrics are analyzed on a validation set that comprises 20% of the entire data set.

In summary, the binary classification assignment sheds light on the efficiency of different classifiers in discerning the probable positive or negative sentiment of future product reviews. The results gleaned from this study present businesses that value customer feedback with precious insights into how consumers perceive their products.

### Logistic Regression Classifier

Logistic Regression Classifier is a widely used statistical model that employs an algorithm for binary classification problems. This technique is named after its defining function, the logistic function, which is employed to model a binary dependent variable. In the Logistic Regression Classifier, the log odds of the outcome is modeled as a linear combination of predictor variables. It estimates the probability that a certain event occurs for a given set of input data, such as whether an email is spam or not, or whether a tumor is malignant or benign. The model's output is a number between 0 and 1 that can be interpreted as the probability of occurrence of the predicted class. When utilizing logistic regression, the model is trained by adjusting the weights (also known as coefficients) for each input feature to minimize a cost function, often using methods like gradient descent. Once the model has been trained, new instances can be classified by plugging the features into the logistic function and rounding the output probability to the nearest whole number.

### Decision Tree Classifier

The Decision Tree Classifier is a popular supervised machine learning algorithm for classification tasks. It is a predictive model that utilizes a tree-like structure to make decisions based on feature values. The tree consists of internal nodes, representing tests on the features, and leaf nodes, representing the class labels or outcomes. The Decision Tree Classifier works by recursively partitioning the input data based on the feature values, aiming to minimize the impurity or maximize the information gained at each step. The impurity measures used can include Gini impurity or entropy. The algorithm evaluates different splits and selects the one that optimally separates the data into the purest subsets. This process continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in each leaf node. These metrics are helpful when there is an imbalance in the dataset. Precision measures the proportion of correctly predicted positive instances out of all predicted positives, while recall calculates the proportion of correctly predicted positives out of all actual positives. The F1-score is the harmonic mean of precision and recall, providing a balanced measure. The structure of a decision tree is similar to the decision-making process of human thinking, which is easy to understand and explain. It uses a tree structure to decompose the problem into a series of simple decision steps, each of which makes a decision based on a feature attribute.

### Multinomial Bayes Classifier

The Multinomial Bayes Classifier is a popular machine learning algorithm for text classification tasks. Based on Bayes' theorem, it assumes that the features are conditionally independent given the class. This classifier is particularly well-suited for handling text data, where each element represents the frequency or occurrence of words or terms in a document. A probability distribution over the features in the Multinomial Bayes Classifier represents each class. During the training phase, the algorithm estimates the probabilities by calculating the frequency of each component in each class and applying smoothing techniques, such as Laplace smoothing, to handle unseen features. During the prediction phase, the classifier calculates the probability of a given document belonging to each class and assigns it to the class with the highest probability. These metrics are commonly used for binary classifiers, but they can also be adapted for multi-class problems. The ROC curve plots the true positive rate against the false positive rate at various classification thresholds. At the same time, the AUC quantifies the classifier's overall performance, with higher values indicating better performance. Multinomial Bayes Classifier is suitable for processing data with high-dimensional feature spaces. In natural language processing tasks such as text classification, the feature is usually the frequency of words or phrases, which leads to a prominent feature space. Multinomial Bayes Classifier can handle such high-dimensional features well and usually performs better.

### Support Vector Classifier (SVC)

The Support Vector Classifier (SVC) is a popular supervised machine learning algorithm for classification tasks. It belongs to the Support Vector Machines (SVM) family and is particularly effective in dealing with complex datasets and non-linear decision boundaries. SVC works by finding the optimal hyperplane that separates different classes by maximizing the margin between the classes. The hyperplane is selected in such a way that it maximally separates the nearest data points from other classes, called support vectors. By maximizing the margin, SVC aims to improve generalization performance on unseen data. SVC can effectively handle non-linear decision boundaries using different kernel functions, such as the radial basis function (RBF) kernel. The kernel trick allows the SVC to implicitly map the input data into a higher-dimensional feature space, where linear separation is possible. This is the fourth approach I try when none of the above three algorithms can achieve the target F1 score.

## **Binary Classification Results**

For Binary Classification, all the results below will be analyzed from the following perspectives. First, the accuracy rate is the ratio of the number of correct predictions (positive and negative cases) to the number of all predictions. This is the most straightforward evaluation metric, but it can lead to misinterpretation in the case of unbalanced samples. Precision is then the ratio of the number of positive examples that were predicted and correct to the number of all positive examples. This metric measures how many of the samples predicted to be positive are positive. Similarly, we will use F1 Score for analysis. Finally, the area under the ROC curve (Receiver Operating Characteristic Curve) is called AUC (Area Under Curve). The ROC curve is a curve that reflects the relationship between sensitivity and 1-specificity, and AUC evaluates the overall performance of the model. The value of AUC is between 0.5 and 1, and the larger the value, the better the model's version. *See appendix.*

### Binary Classification Cutoff 1 Results

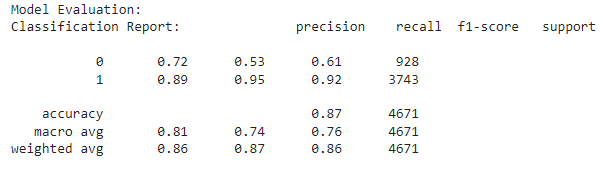
The results of the multiclass classification task showed that the Logistic Regression model performed the best, with the highest F1 Score among the other classifiers. A logistic regression classifier is used for model training and prediction in this binary classification task. The model uses cross-validation for parameter tuning, and the F1 score evaluates the model performance.

By comparing the results of different regularization strength parameters C (including 1, 10, 50, and 100), we found that with the increase of C, the average F1 score of the model gradually improved from 0.705 to 0.748. This might indicate that larger values of C (i.e., lower regularization strength) fit the data better during model training.

The optimal parameter finally selected is C=100, which corresponds to a maximum F1 score of 0.748, which means that the average level of precision and recall of the model is relatively high.

However, we can see from the classification report that there is a significant difference in the model's performance when dealing with different categories of data. For category 1 (predicted positives), the model achieves a precision of 0.89, a recall of 0.95, and an F1 score of 0.92, showing solid predictive power. However, for class 0 (predicted negatives), the model achieves a precision of 0.72, a recall of only 0.53, and an F1 score of 0.61, significantly weaker than the predicted performance of class 1.

This situation may be caused by data imbalance. That is, the number of samples of Category 1 may be much larger than that of Category 0. Therefore, the model may tend to predict the class with more samples, which may lead to weak predictive power for the minority class. In response to this situation, we can consider applying some strategies to deal with unbalanced data, such as oversampling the minority class, under sampling the majority class, or using the synthetic minority class oversampling technique (SMOTE). In addition, it can be seen from the confusion matrix that the model has a higher prediction accuracy for class 1 and a lower prediction accuracy for class 0. This further validates that the data imbalance problem may affect the model.



Classification Report for Logistic Regression with Cutoff of 1

Although the model's AUC(Figure 1) score (0.890) is relatively high, indicating that the model performs well in distinguishing positive and negative examples, the macro-average F1 score (0.764) suggests that the model's ability to deal with imbalanced categories needs to be improved. Improve. Therefore, when further optimizing the model, we need to pay special attention to the problem of data imbalance to improve the model's prediction performance for the minority category.

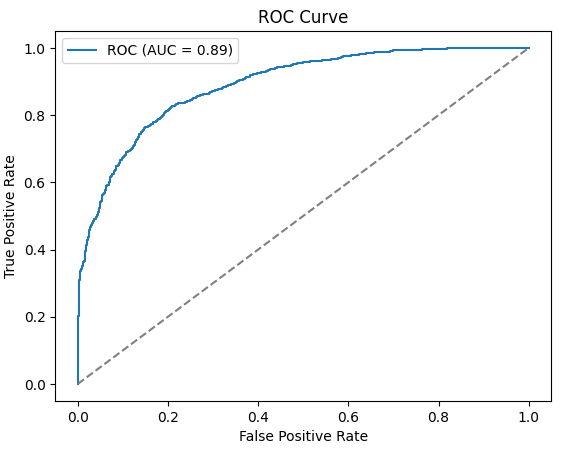


Figure 1: ROC Curve for Logistic Regression Cutoff of 1

For the three models of Cutoff1, logistic regression, decision tree, and Bayes. the F1 score of logistic regression is the best at 0.76. So logistic regression is the best model up to Cutoff1.

### Binary Classification Cutoff 2 Results

Initially, I tried using a Support Vector Machine (SVM) classifier and set its hyperparameters to achieve Cutoff2. SVM is a commonly used classification algorithm that can be used for text classification problems. Here, svc stands for SVM classifier, setting some hyperparameters such as random\_state, class\_weight, and probability. Next, I defined the hyperparameter grid svc\_hyperparameters of the SVM classifier, including different values ​​of the parameter C (penalty term), kernel function type, and gamma (kernel function coefficient). Finally, use grid search (GridSearchCV) and 5-fold cross-validation (cv=5) to tune the hyperparameters of the SVM model. By calling the fit method, grid search iterates over hyperparameter combinations and runs cross-validation on the training data to evaluate model performance. The F1 score (scoring='f1') is used as the evaluation metric in this example. But the effect is not ideal. The score of Kaggle is 0.7903.

Then I fall back on logistic regression to implement Cutoff2. The report gives the mean F1 score for different hyperparameter 'C' values. We can see that the F1 score varies slightly for different 'C' values. In logistic regression, 'C' is a regularization parameter that controls the complexity of the model. The larger 'C' is, the more complex the model, which may lead to overfitting; the smaller 'C' is, the simpler the model, which may lead to underfitting.

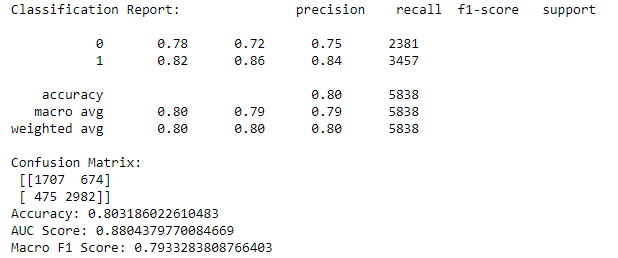
The best parameter choice is {'C': 10} since the model achieved the highest mean F1 score of 0.7903 under this setting.

Next is the classification report, which provides the model's precision, recall, and F1 score in predicting the two classes (0 and 1). Precision is the ratio of correctly predicted positive observations to the total predicted positives; recall is the ratio of correctly predicted positive comments to all observations in actual class - yes, the F1 score is the harmonic mean of precision and recall.

Then comes the confusion matrix, which shows the relationship between the actual and predicted outcomes of the model. The cells at the upper left and lower right (1707 and 2982) represent the number of correct predictions made by the model, and the cells at the upper right and lower left (674 and 475) represent the number of incorrect predictions.

Afterward, the report provides the model's accuracy, which is the ratio of the number of correct predictions to the total number of predictions. Here it's 0.8032. This means that out of all predictions, 80.32% were correct.

Finally, the report gives the AUC score and macro F1 score. The AUC score is the area under the Receiver Operating Characteristic (ROC) curve, reflecting the model's predictive performance across all thresholds. The closer to 1, the better. Here it's 0.8804. The macro F1 score is the average of F1 scores across classes, and here it's 0.7933.



Classification Report for Logistic Regression Cutoff of 2

The ROC curve (Figure 2) showed an AUC value of 0.88. This means that the logistic regression classifier does a good job of distinguishing between the positive and negative classes.

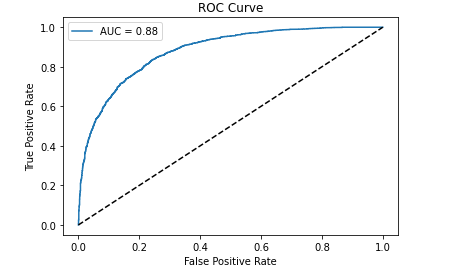
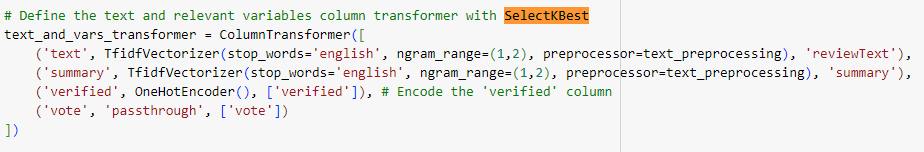


Figure 2: ROC Curve for Logistic Regression Cutoff of 2

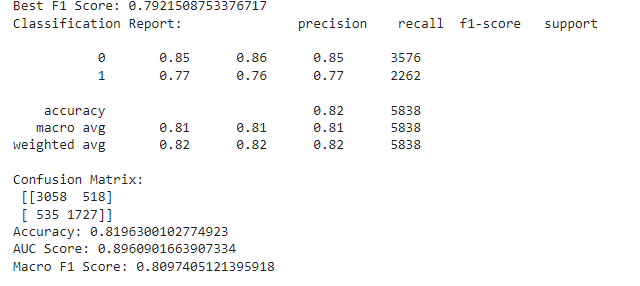
For the three models of Cutoff1, logistic regression, decision tree, and Bayes. the F1 score of logistic regression is the best at 0.79. So logistic regression is the best model up to Cutoff1.

### Binary Classification Cutoff 3 Results



Feature selection is a reprocessing step that is primarily used to reduce model complexity, improve model accuracy, and reduce training time。 To achieve the standard F1 value, I tried to add SelectKBest to the original model, and its goal is to select the K best (or most important) features from the original feature set. In the above code snippet, some additional adjustments may be required if we add SelectKBest. For example, since we use the TF-IDF vectorizer to process text, the number of generated features may be huge (maybe thousands or tens of thousands), and SelectKBest may not be applicable at this time unless we know which features are the most important. But judging from the news given to us by the previous logistic regression model, we can try to select variables directly.

In this model, I used 5-fold cross-validation for training. These lines print the mean F1 score and standard deviation for different regularization parameters C. The F1 score is an indicator of model accuracy, which considers the model's precision and recall. The best F1 score obtained by the model for different C values is 0.792, corresponding to a C value 10. This means for a logistic regression model, C=10 is the best parameter. The F1-scores for categories 0 and 1 are 0.85 and 0.77, respectively, and the average F1-score is 0.81, showing that the model performs relatively well in both categories. In the confusion matrix, the model has 3058 correct and 518 wrong predictions for category 0 and 1727 correct and 535 wrong predictions for category 1, with an accuracy rate of 0.82. These conclusions and insights show that the model performs exceptionally well on this particular task and dataset. The model achieved an accuracy rate of 82%, and the F1 score was also relatively high.



Classification Report for Logistic Regression Cutoff of 3

AUC stands for "Area Under the Curve," which is the area under the ROC curve and is used to evaluate the model's overall performance. The closer the value is to 1, the better the model's performance. Here the AUC score is 0.9, showing that the model performs well. In this binary classification problem, the Macro F1 score is equal to 0.81, indicating that the model's overall performance is good.

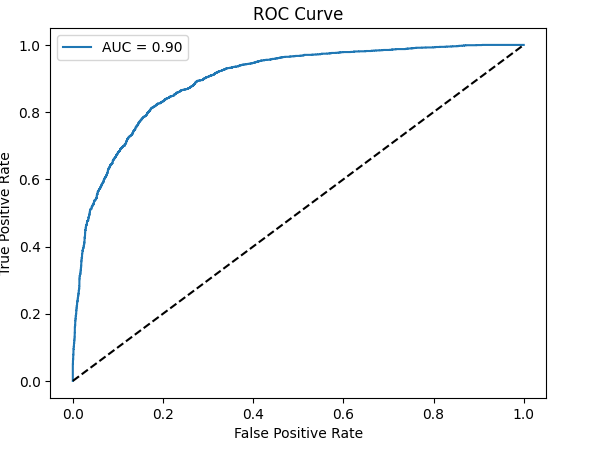
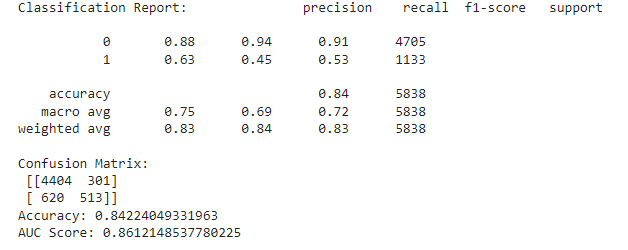


Figure 5: ROC Curve for Logistic Regression Cutoff of 3

### Binary Classification Cutoff 4 Results

The results are based on a Logistic Regression Classifier with a rating cutoff 4. The classifier was trained and evaluated using different values of the regularization parameter C. The F1 score was used as the evaluation metric, considering both precision and recall. The best-performing parameter was found to be C = 150, which achieved a mean F1 score of 0.756. The mean F1 scores for different C values were consistently above 0.70, indicating a reasonable level of classification performance. The classification report provides detailed insights into the model's performance. The classifier achieved 88% precision for class 0 (low rating) and 63% precision for class 1 (high rating). This indicates that the classifier better identifies low-rated instances than high-rated ones. The recall, or sensitivity, was 94% for class 0 and 45% for class 1, indicating that the classifier could capture a higher proportion of low-rated instances than high-rated ones.



Classification Report for Logistic Regression Cutoff of 3

The F1 score is a harmonic mean of precision and recall, providing an overall measure of the classifier's performance. The weighted average F1 score was 0.83, indicating a reasonably good balance between precision and recall across both classes. The confusion matrix provides additional insights into the classification results. It shows that out of 4705 instances classified as class 0, 4404 were true negatives, and 301 were false negatives. Similarly, out of 1133 instances classified as class 1, 620 were false positives, and 513 were true positives. The overall accuracy of the classifier was 84%, indicating that 84% of the instances were correctly classified. The AUC score, which measures the classifier's ability to discriminate between classes, was 0.861, suggesting a good level of discrimination.

The model showed a higher precision and recall for low-rated instances than high-rated ones. Based on these results, it can be concluded that the Logistic Regression Classifier with a rating cutoff of 4 achieved a reasonably good performance in classifying instances into low and high ratings. Further analysis and optimization may be required to improve the classifier's performance on high-rated cases and achieve a better balance between precision and recall across both classes.

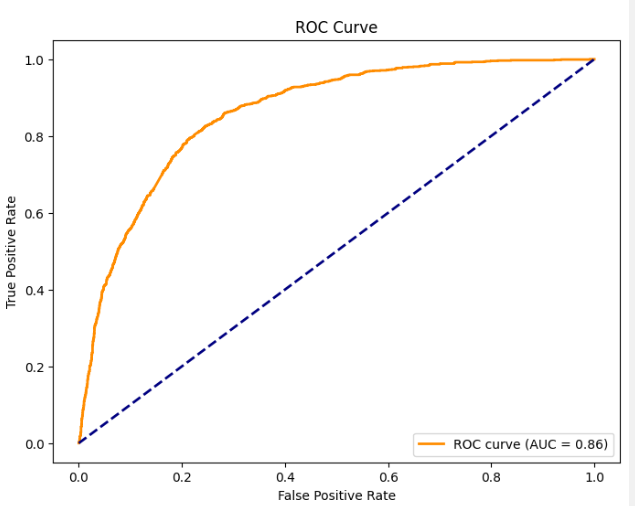


Figure 5: ROC Curve for Logistic Regression Cutoff of 3

## **Multiclass Classification**

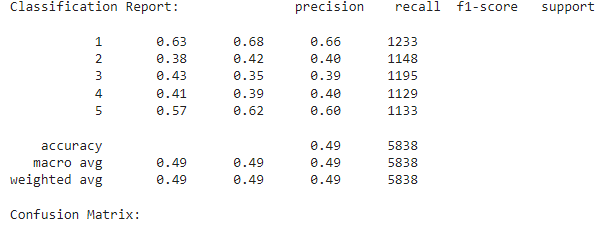
Multiclass Classification is a classification task in machine learning that involves assigning instances to one of multiple categories or labels. Binary classification involves only two classes, but multiclass classification involves three or more classes. For example, you might have a project that needs to identify the main content of an image. If you only need to distinguish whether an image is a dog or a cat, then it is a binary classification problem. But if you also need to determine what kind of dog it is or what kind of cat it is, then it becomes a multiclass classification problem. There are many algorithms for multiclass classification, including logistic regression, decision trees, random forests, support vector machines, naive Bayes, neural networks, etc. There are also many evaluation methods for multiclass classification, including confusion matrix, precision, recall rate, F1 score, etc. A common strategy for dealing with multiclass classification problems is to use the "One-Vs-All" (One-Vs-All, or One-Vs-Rest) strategy, in which we train a binary classifier for each class, which operates to distinguish this category from all other categories.

In this project, I will use three logistic regression algorithms, random forest and gradient boosting, for Multiclass Classification. Understanding and interpreting the logistic regression model is relatively straightforward, giving it certain advantages in scenarios with high interpret ability requirements. Gradient boosting often achieves high scores in machine learning competitions and practices because of its ability to "brainstorm" multiple weak learners to build powerful models. Random forests usually require minimal parameter tuning to get good results, and by integrating the prediction results of multiple decision trees, random forests can significantly improve the accuracy of predictions. Next, I will discuss the Multiclass Classification result in detail.

## **Multiclass Classification Results**

Multiclass classification refers to a classification problem where the goal is to assign an input instance to one of three or more predefined classes or categories. In other words, it involves distinguishing between multiple classes instead of just two (as in binary classification). In multiclass classification, each instance or data point can belong to only one class out of the several available options. The classifier's task is to learn a decision boundary or a set of rules that can effectively separate the different classes based on the given input features.

First, we tuned the hyperparameter C for the logistic regression model. C is the inverse of the regularization strength in the logistic regression model. That is, the larger the C, the smaller the regularization strength. With 5-fold cross-validation, we evaluated models with C values of 0.001, 0.01, 0.1, 0.5, and 1, respectively. The results show that when C=0.5, the F1 score of the model is the highest, which is 0.48183, which indicates that the model has the best prediction precision and recall balance. The classification report shows the precision, recall, and F1-score for five classes of labels. The confusion matrix further reveals the performance of the classifier. The values on the main diagonal represent the number of correctly classified classes. The number of correct classifications of classes 1 and 5 is significant, while the number of suitable varieties of classes 2, 3, and 4 is relatively small. Among them, the F1 scores of categories 1 and 5 are relatively high, reaching 0.66 and 0.60, while the F1 scores of categories 2, 3, and 4 are relatively low, 0.40, 0.39, and 0.40, respectively. This may indicate that the model can distinguish between class 1 and class 5, but it does not perform well in class 2, class 3, and class 4. In all 5838 validation samples, the overall accuracy of the logistic regression model was 0.49229, which means that about 49.23% of the samples were correctly classified.

 Classification Report for Muliticlass Classification Logistic Regression

The ideal situation is that the actual positive rate (TPR) is 1, and the false positive rate (FPR) is 0, meaning the model has no misclassifications. In the ROC graph, this point is in the upper left corner. Therefore, the closer the curve is to the upper left corner, the better the classifier does. The larger the area under the curve (AUC), the better the classification effect of the model. Whereas, under this result, (Figure 6) shows that the classifier struggles with classes 2, 3, and 4, which have relatively low recall and precision scores. The ROC-AUC score of the model is 0.80613. The closer this score is to 1, the better the model's performance. The ROC-AUC score measures the ability of the model to distinguish between positive and negative examples. That is, no matter how much the threshold is set, the model's overall ability to distinguish between positive and negative samples. Finally, the model has a macro-average F1 score of 0.48706, which combines the model's balance of precision and recall for all classes. Although this logistic regression model performed well on some indicators (such as ROC-AUC score), it still needs to be improved on some other essential indicators (such as overall accuracy and macro-average F1 score), especially for 2 classes, 3 classes, and The predictive ability of category 4 needs to be further enhanced.

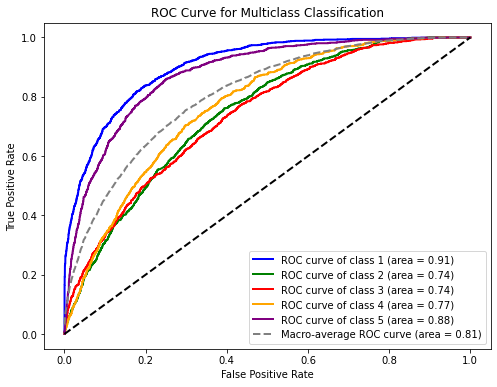


Figure 6: ROC Curve for Logistic Regression Multiclass Classification

## **Clustering**

Clustering (Clustering) is a method of unsupervised learning. Its goal is to divide the samples into the data set into several groups or "clusters" so that the models in the same group are more similar. In comparison, the samples in different groups have a higher similarity. In clustering, we do not need to know the category information in advance but divide it according to the internal structure and distribution of the data. Standard clustering algorithms include K-means, hierarchical clustering, DBSCAN, etc. In this project, we focus on exploring K-means, which divides a set of data points into different categories, and the similarity of data points within it defines each category. The K-means algorithm tries to find the optimal solution that minimizes the intra-category variance and maximizes the inter-category variance. The result of the K-means algorithm depends on the selection of the initial category center point, so there may be multiple locally optimal solutions. In order to obtain better results, the K-means algorithm is usually run multiple times, using different initial center points, and the solution with the minor total error is selected as the final result.

I use the K-means algorithm to cluster the eigenvectors and try different numbers of clusters by looping. For each cluster number k, create a K-Means object and fit the feature vectors using the appropriate method. model.inertia\_ represents the sum of squared errors of the model, which is added to the sum\_of\_squared\_distances list. The silhouette\_score function calculates the Silhouette score of the cluster, adding it to the silhouette\_values list. Then, draw two graphs. One is a graph of the relationship between the number of clusters drawn by the "Elbow Method" and the sum of squared errors, and the other is a graph of the relationship between the number of clusters drawn by the "Silhouette Score Method" and the Silhouette score. These graphs are used to help choose the optimal number of clusters. Finally, print the results and calculate the clustered Silhouette coefficient and adjusted Rand index. These metrics are used to evaluate the quality of clustering. The Silhouette coefficient measures the closeness and separation of clusters, and the closer the value is to 1, the better the clustering quality is. The Rand index measures the consistency between the clustering result and the real category, and the closer the value is to 1, the more consistent the clustering result is with the real category.

## **Clustering Results**

Cluster analysis of text data by using the K-means algorithm. A text vectorization object count\_vector is initialized using CountVectorizer, and the cleaned\_text column is converted into a text vector using the fit\_transform method. This will generate a vector representation for each text. Then, the best value for choosing the number of clusters is evaluated using the "elbow method" and the "Silhouette score" method. The code uses a loop to try different numbers of clusters, calculates the "sum of squared distances" and "Silhouette score" values of the model corresponding to each number of clusters, and stores the results in the sum\_of\_squared\_distances and silhouette\_values lists. Next, I created a graph object containing two subplots and used the plot function to plot the number of clusters versus the "sum of squared distances" and "Silhouette score." The first subplot shows the "elbow method" results, while the second subplot shows the results of the "Silhouette score" method. Then I applied the K-means clustering algorithm with the optimal number of clusters. The variable optimal\_clusters is set to the value of the optimal number of clusters, which can be determined by observing the graphical results of the "elbow method" and "Silhouette score" methods. Then, the text vectors are clustered using the K-means algorithm, and the clustering results are stored in k\_means\_model.labels\_. Finally, I calculated the silhouette coefficient of the clustering results using the silhouette\_score function and the adjusted Rand index between the clustering results and the original categories using the adjusted\_rand\_score function.

In addition to the elbow method, the silhouette score plot in Figure 7 provides further insights into determining the optimal number of clusters. The silhouette score measures how well each sample within a cluster is similar to others in the same cluster compared to samples in neighboring clusters. The score ranges from -1 to 1, where higher values indicate better clustering results.

By examining both the elbow method and silhouette score, we can make a more informed decision about the optimal number of clusters. In this case, we observe that both methods indicate that 5 clusters would be an appropriate choice. This consensus suggests that 5 clusters can effectively capture the underlying patterns and structures in the data.

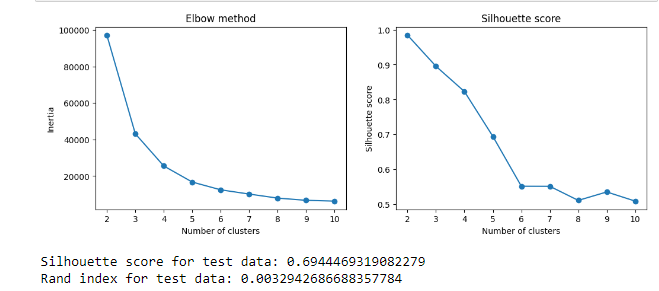


Figure 7: Elbow method and Silhouette score

Silhouette score is used to measure the closeness and separation of clustering results, and its value ranges from -1 to 1, and the closer the value is to 1, the better the clustering result. In this example, the silhouette coefficient of the test data is 0.6944469319082279, indicating that the clustering results are relatively good, with high closeness and separation. The Adjusted Rand index measures the similarity between the clustering result and the real label. Its value ranges from -1 to 1, and the closer the value is to 1, the more consistent the clustering result is with the natural label. In this example, the adjusted Rand index for the test data is 0.0032942686688357784, indicating a low agreement between the clustering results and the true labels.

Overall, the silhouette coefficient indicates better clustering results, but the adjusted Rand index indicates a lower agreement between the clustering results and the true labels. This may mean that the clustering model needs to capture the true category information well for this dataset or that the data itself has high noise or complexity.

## **Conclusion**

Among my four cutoff and multiclass models, logistic regression showed absolute accuracy and high scores, and it is also the best model.

The cutoff is 1: In this case, the best F1 score for the model is 0.748, which corresponds to a C value of 100. Although the accuracy rate is 0.87, its precision, recall, and F1 score (especially for category 0) are relatively low. This could mean that the model is better at identifying positive examples (class 1) but weaker at identifying negative examples (class 0).

The cutoff is 2: In this case, the best F1 score for the model is 0.801, which corresponds to a C value 10. The model achieves an accuracy of 0.80, a relatively balanced performance as it performs fairly close on both classes.

A cutoff of 3: In this case, the best F1 score for the model is 0.792, which corresponds to a C value 10. The model's accuracy is 0.82, which is a relatively balanced performance.

The cutoff is 4: In this case, the best F1 score for the model is 0.756, which corresponds to a C value of 150. Although the accuracy rate is 0.84, the performance of Category 1 (negative examples) is significantly lower than that of Category 0, which may indicate that the model has a more vital ability to identify positive examples (category 0) but less ability to identify negative criteria (category 1). weak.

Multiclass classification: In this case, the best F1 score for the model is 0.482, which corresponds to a C value of 0.5. The accuracy rate is only 0.49, which is the lowest accuracy rate, indicating that the multiclass classification task is more complicated.

Collectively, these results provide the following insights. In binary classification tasks, when the rating cutoff is low (1 or 2) or high (4), the model tends to perform better in one class than the other. This may be because the number of samples of one class in the data is much larger than that of the other class, resulting in the model's more vital recognition ability for most classes. This class imbalance problem can be solved by oversampling the minority class or undersampling the majority class. As the rating cutoff increases, the model's performance may decrease. This may be because the distribution of positive and negative examples becomes closer as the rating cutoff increases, making the classification task more difficult.

In multiclass classification tasks, the performance of the model drops significantly. This may be because multiclass classification tasks are inherently more difficult than binary classification tasks. At the same time, this may also be because some categories in the data have less sharp boundaries with other categories, making it easier for the model to distinguish these categories. C is the regularization parameter of the logistic regression model, which is used to control the complexity of the model. Smaller C values correspond to more robust regularization. The model is simpler; while larger C values correspond to weaker regularization, the model is more complex. If the value of C is not appropriately chosen, it may lead to overfitting or underfitting of the model. In all tasks, the optimal value of C is between 1 and 150, which may indicate that the logistic regression model is susceptible to this parameter.

## **Recommendations**

First, the class imbalance problem may affect the model's performance in some classification tasks, especially for the minority class. I could try using some techniques for dealing with class imbalance, like oversampling the minority class (e.g., SMOTE), undersampling the majority class, or using a model or loss function sensitive to class imbalance. Then, while logistic regression is a powerful model, there may be cases where other models perform better in specific tasks, such as multiclass classification. For example, I could try a support vector machine (SVM), a random forest, or a gradient boosting model, or even try a deep learning model like a multi-layer perceptron (MLP) or a convolutional neural network (CNN). In these tasks, the performance of the logistic regression model is susceptible to the value of C. I can use a finer grid search or other more advanced hyperparameter optimization methods such as random search or Bayesian optimization to find the best value for C. Second, you and I can try more feature engineering strategies like creating interactive features, using polynomial features or making feature selection. At the same time, if the original data has noise or outliers, it should be dealt with accordingly. In the future, obtaining more data can help improve the model's performance, especially for multiclass classification tasks. Due to time reasons, there may still be some hidden data patterns that have a significant impact on the model that has yet to be discovered. A more in-depth data analysis may uncover these patterns and improve models accordingly.

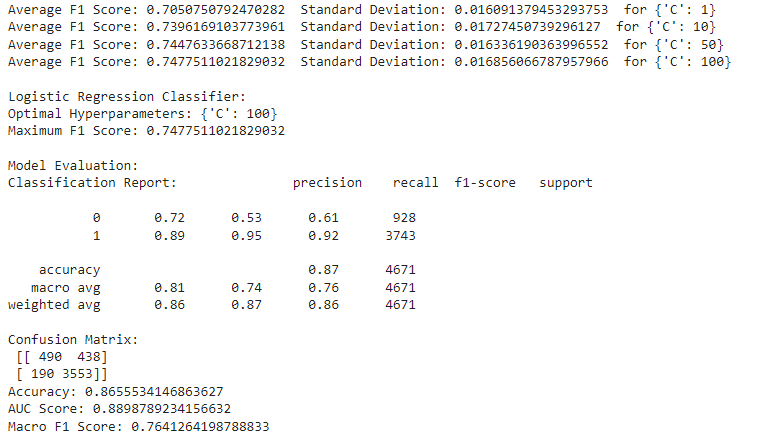
# **References**

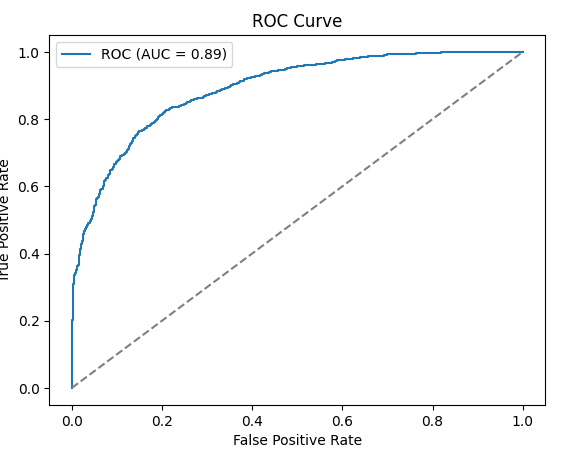
1. Data Lakes and Analytics on AWS - Amazon Web Services. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/big-data/datalakes-and-analytics/

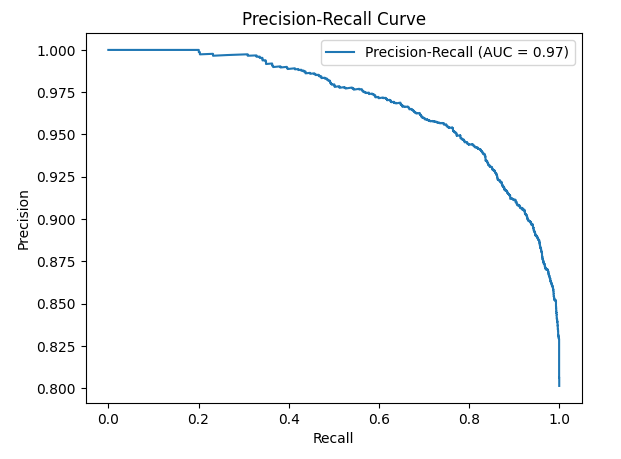
**Appendix: Binary Classification Results**

**Binary Classification Cutoff 1**

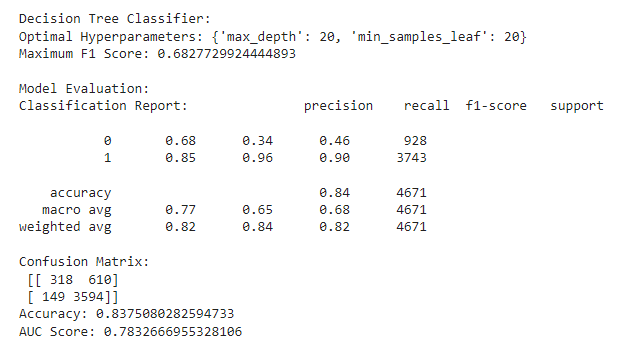
***Logistic Regression Classification Results with cutoff of 1***

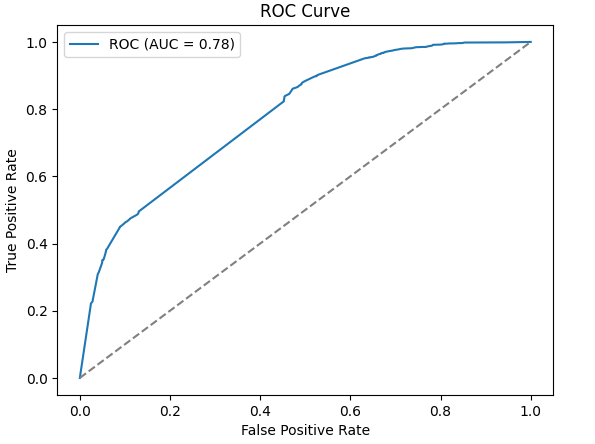


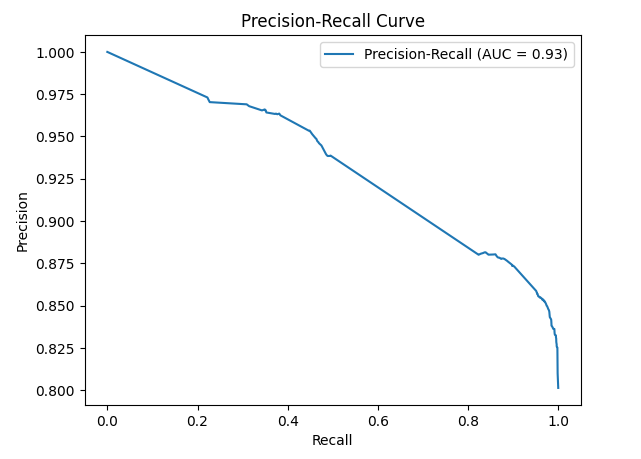




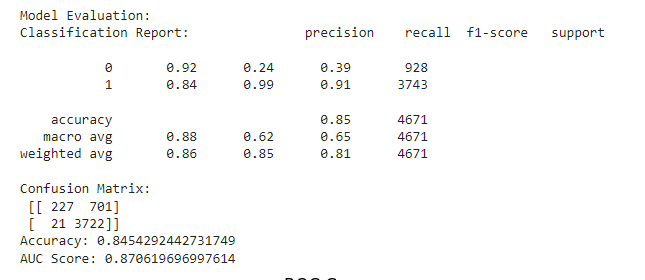
***Decision Tree Binary Classification Results with cutoff of 1***

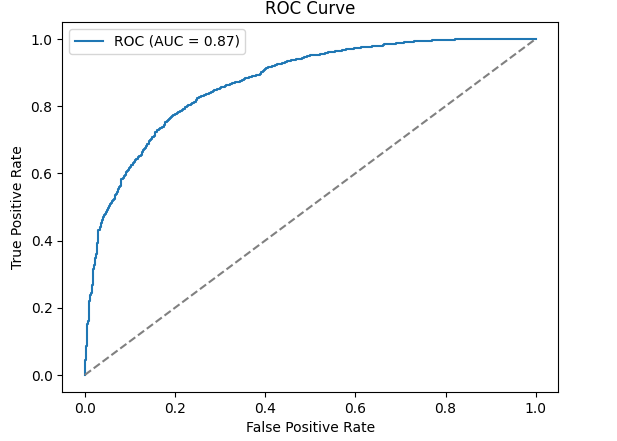


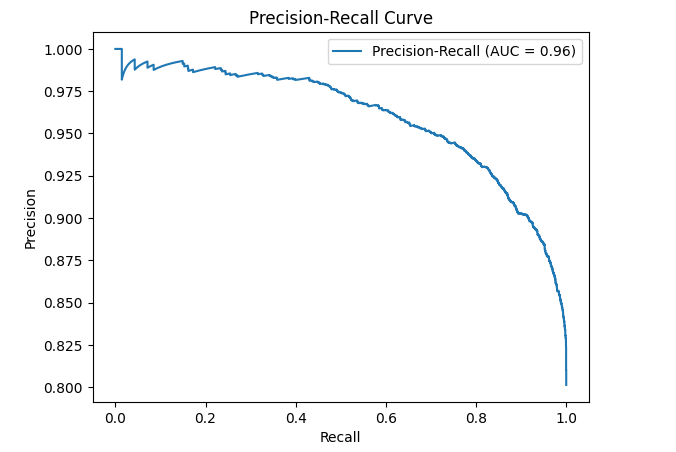




***Multinomial Naïve Bayes Binary Classification Results with cutoff of 1***

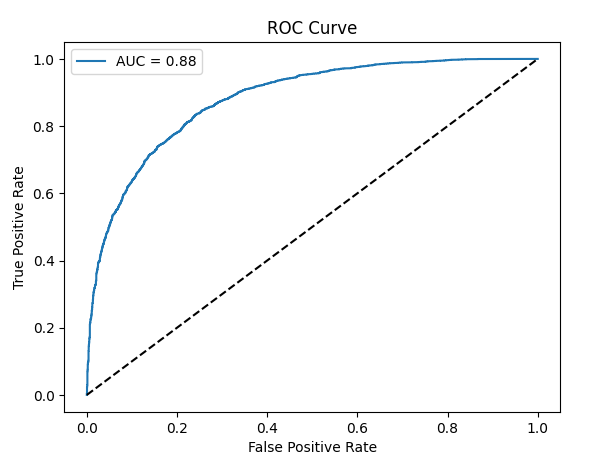
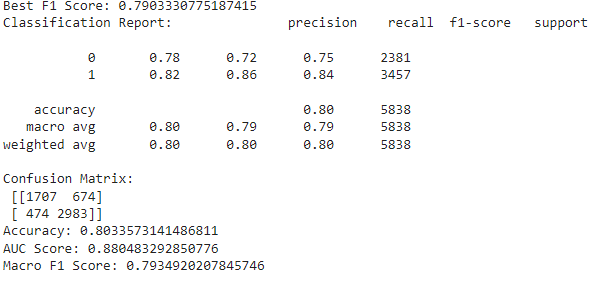


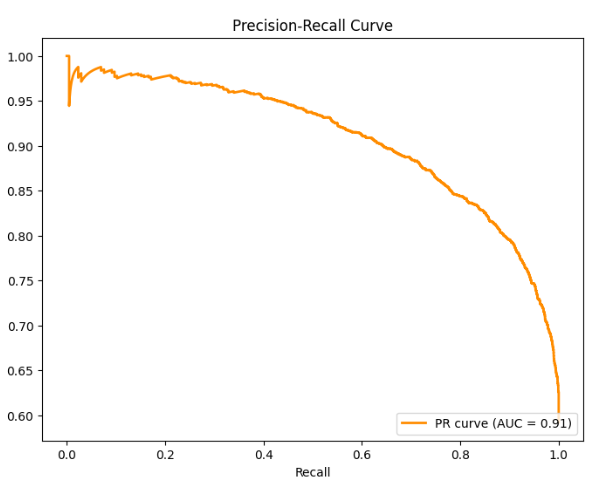




**Binary Classification Cutoff 2**

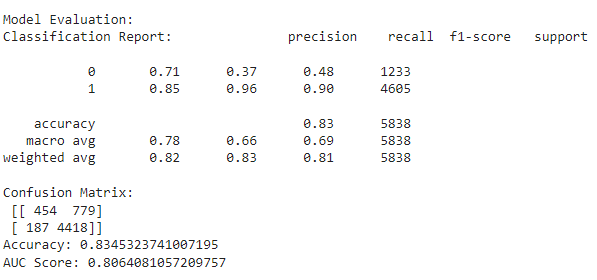
***Logistic Regression Binary Classification Results with cutoff of 2***

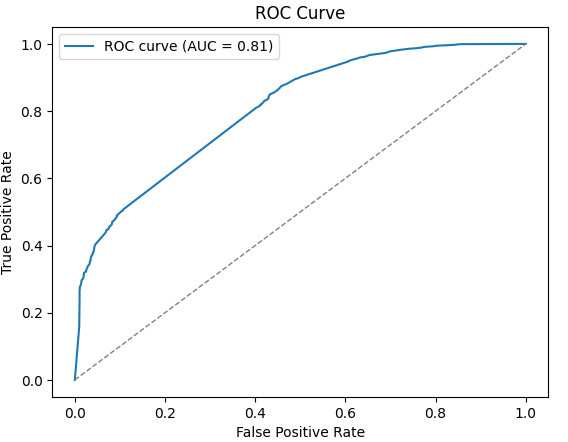


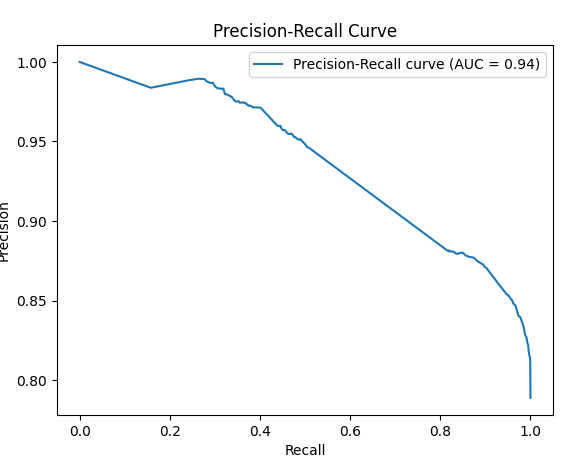


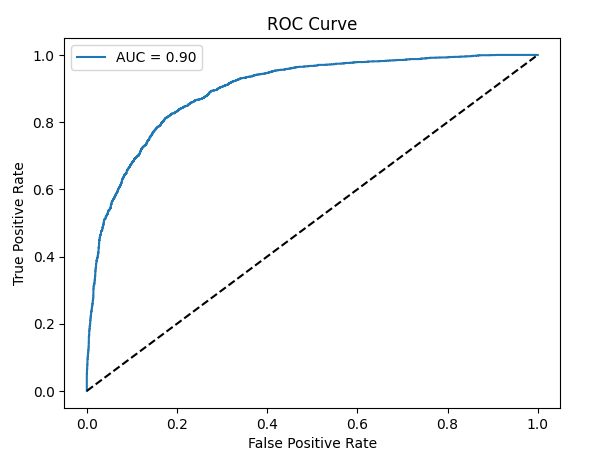
**Binary Classification Cutoff 3**

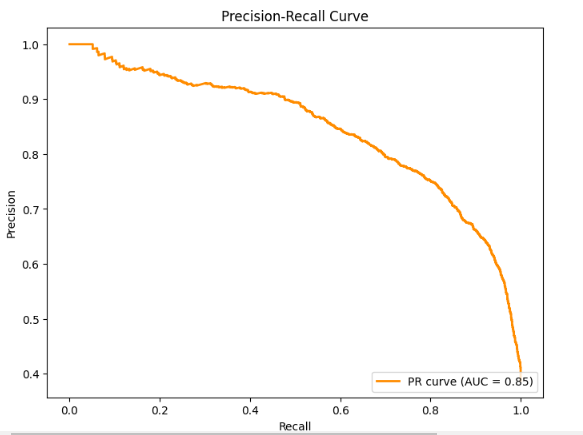
***Logistic Regression Binary Classification Results with cutoff of 2***

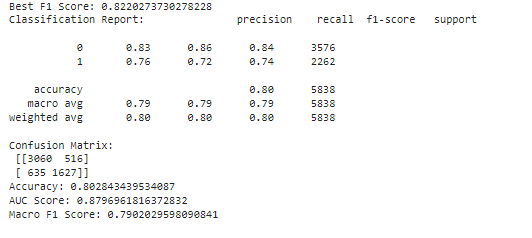




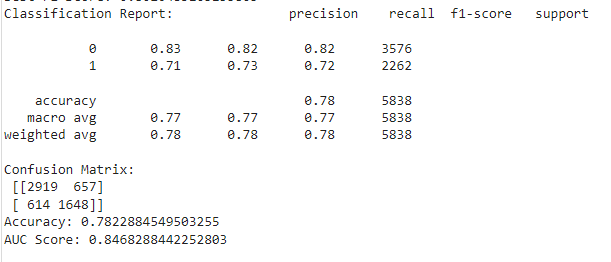


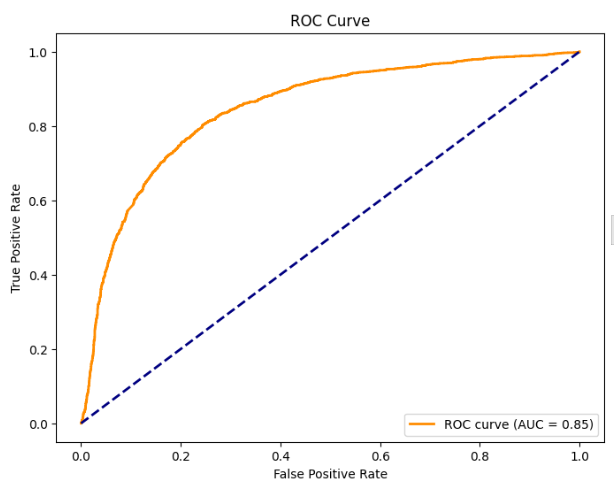


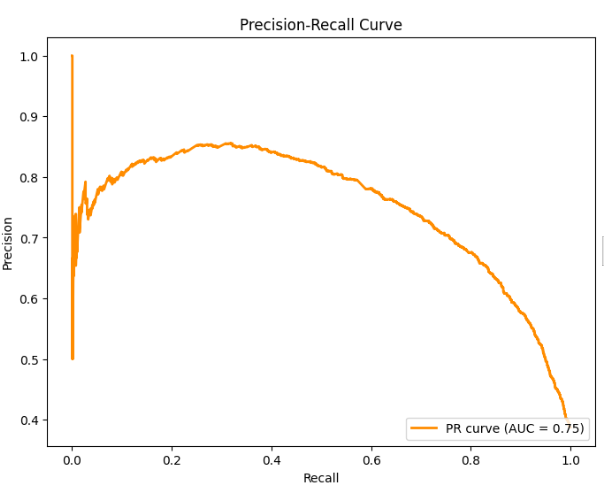


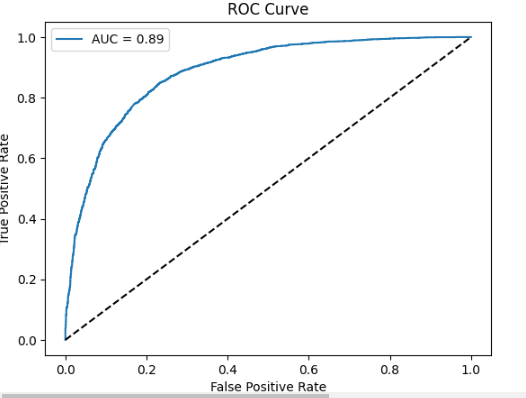
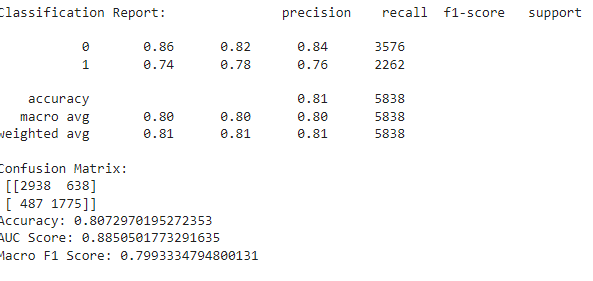


***Multinomial Naïve Bayes Binary Classification Results with cutoff of 3***



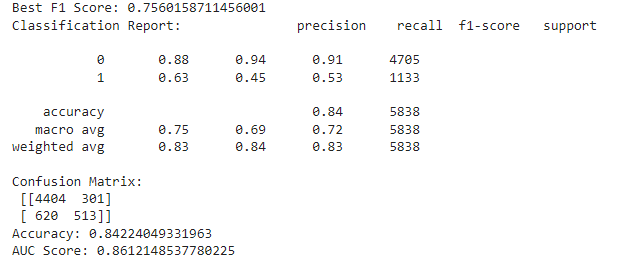


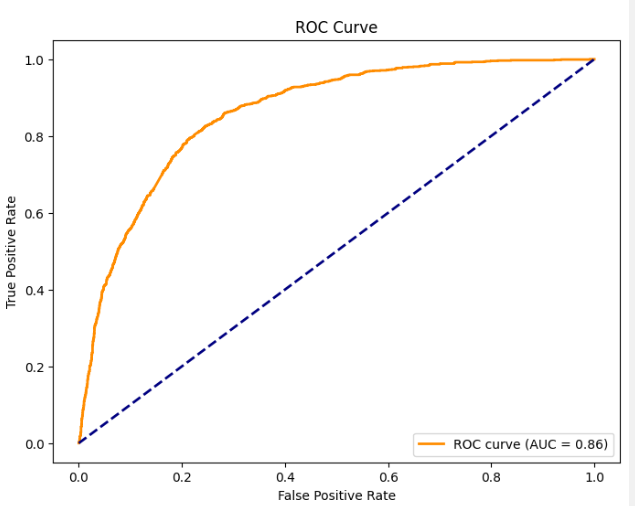


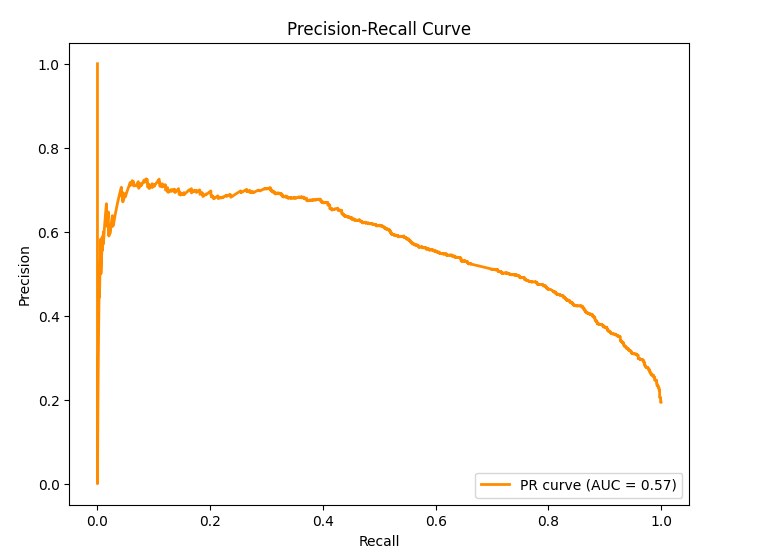


**Binary Classification Cutoff 4**

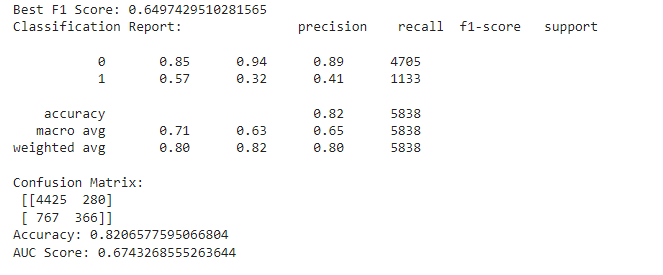
***Logistic Regression Binary Classification Results with cutoff of 4***

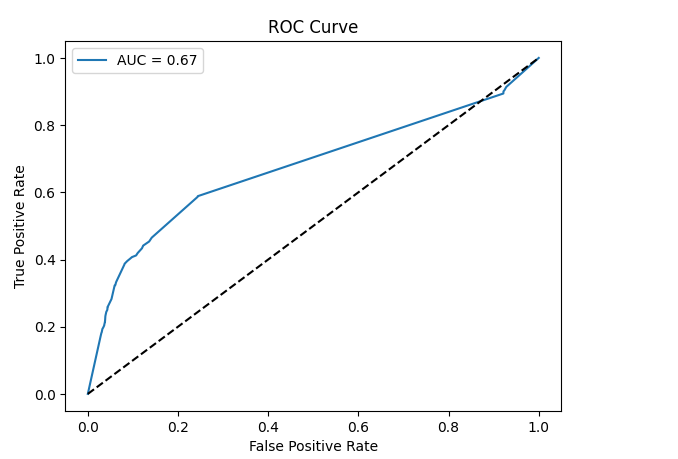


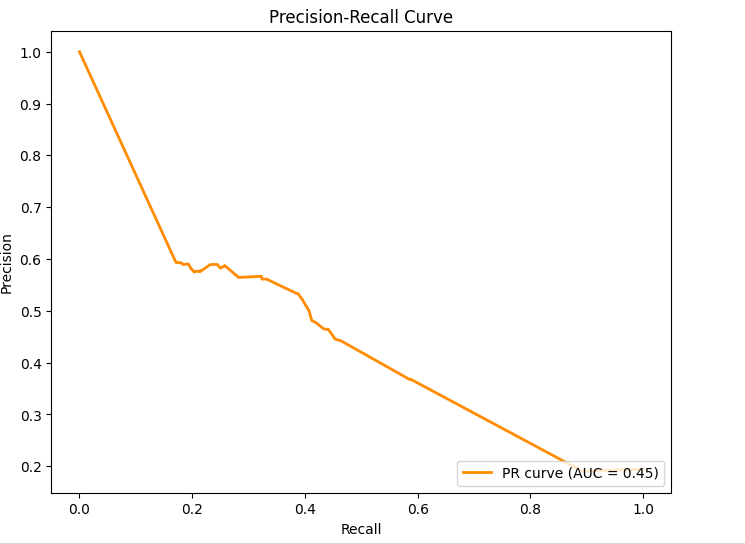




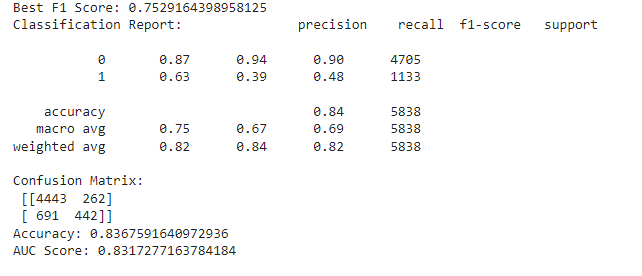
***Decision Tree Binary Classification Results with cutoff of 4***

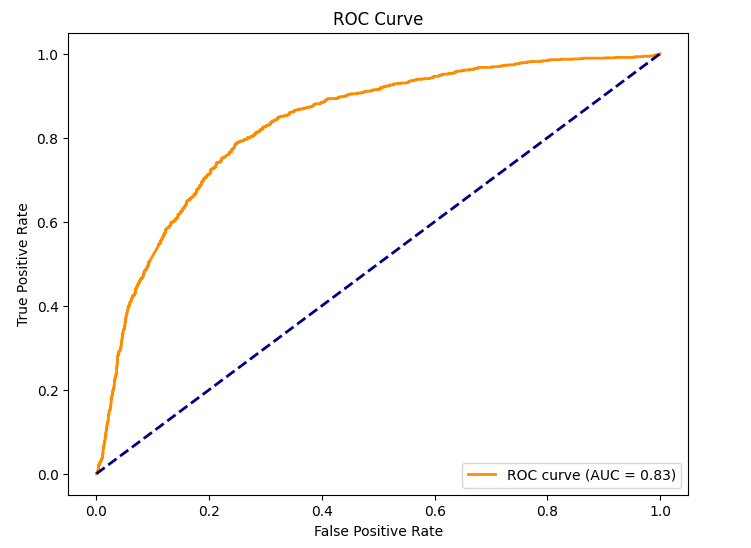


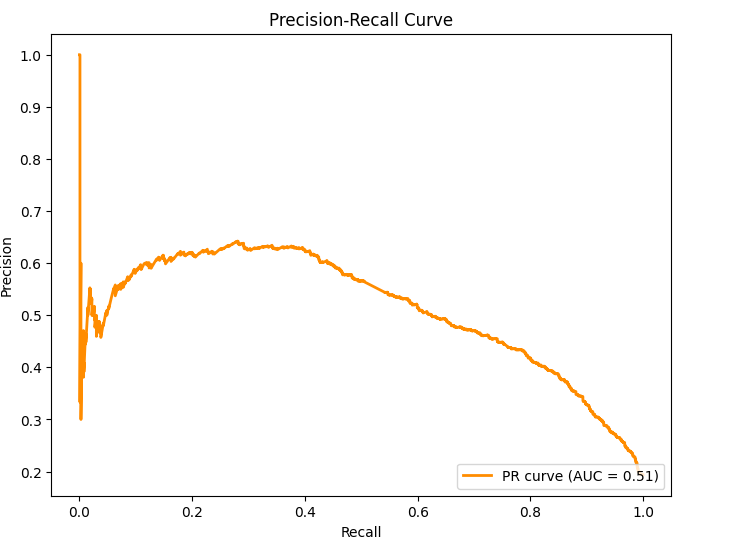




***Multinomial Naïve Bayes Binary Classification Results with cutoff of 4***

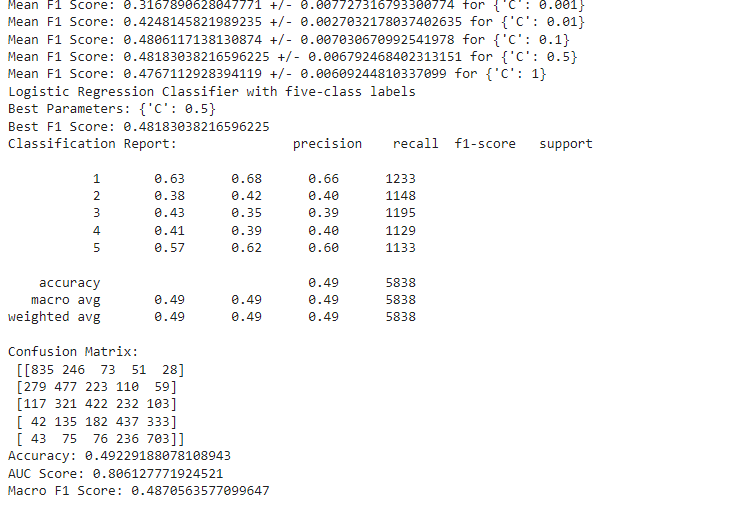


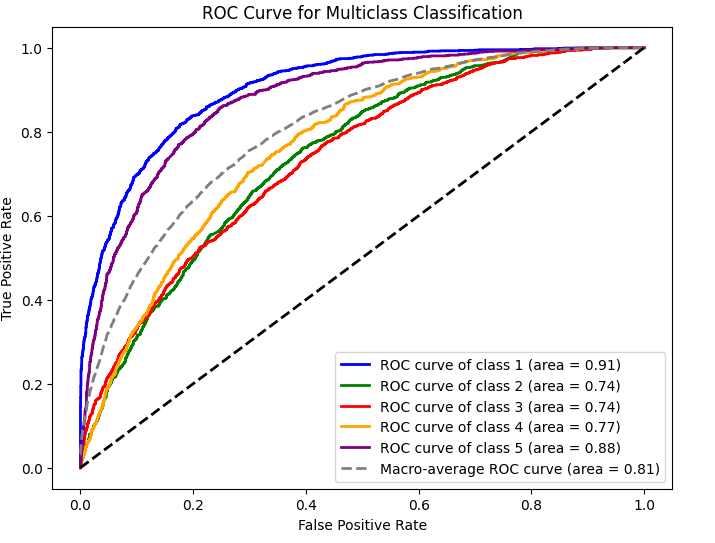




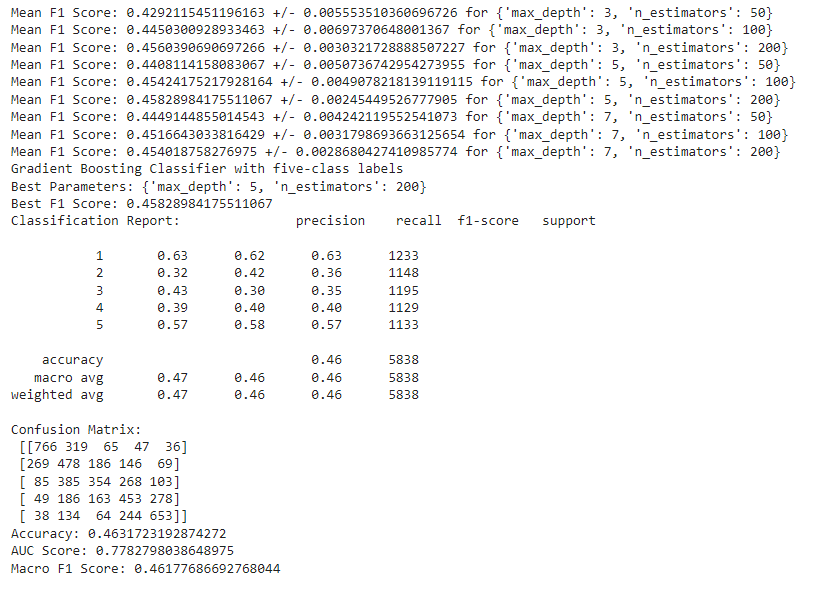
**Multiclass Classification**

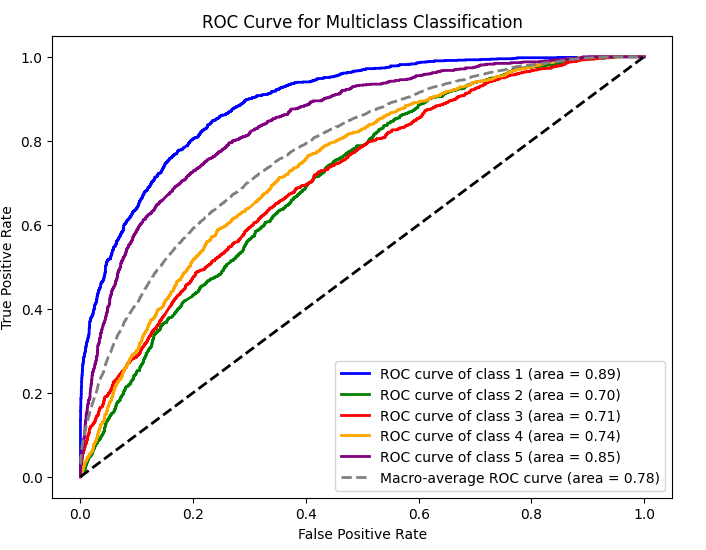
***Logistic Regression***



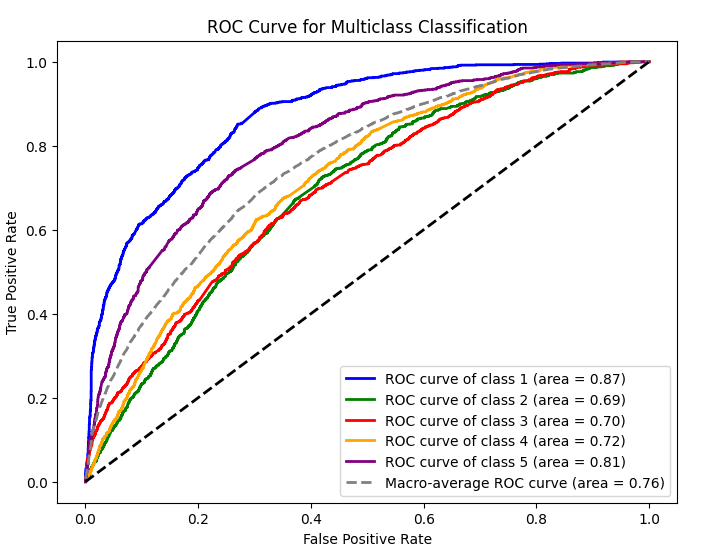
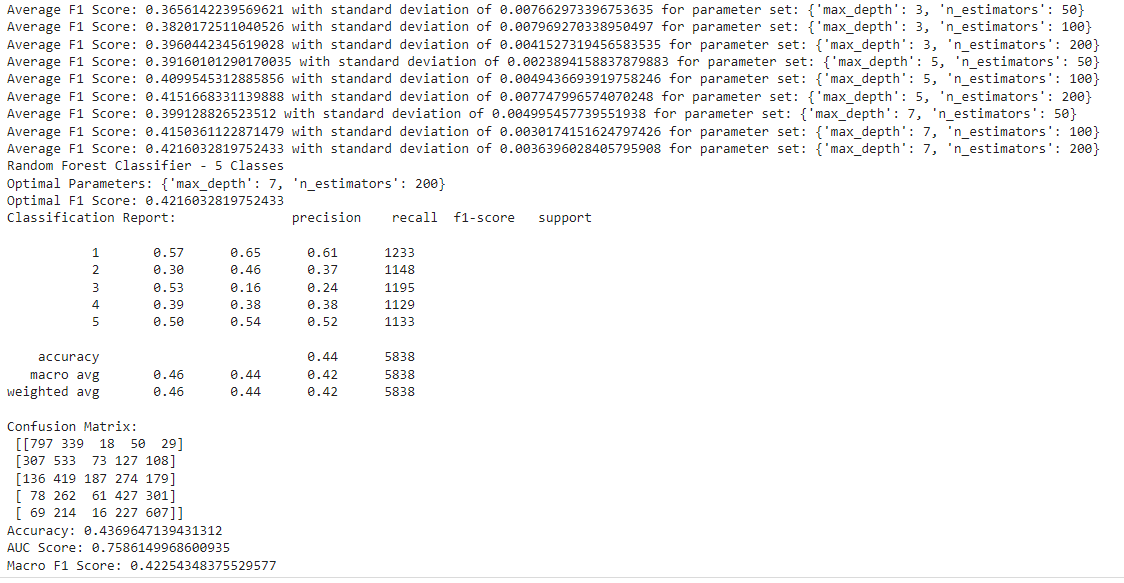


***Random Forest***





***Gradient Boosting Multiclass Classification Results***



1. *Data Lakes and Analytics on AWS - Amazon Web Services.(n.d.). Amazon Web Services, Inc.* [↑](#footnote-ref-0)