## Artificial Intelligence and

## Machine Learning

Project Report

Semester-IV (Batch-2022)

Customer Churn Prediction

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Mridul Goyal 2210990584

Mohd Ishaan 2210990581

Mann Sharma 2210990555

Nishant Dinesh 2210990617

Ms Monica Dutta

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

**1. Introduction**

In today's dynamic and hypercompetitive business landscape, customer retention stands as a paramount objective for companies spanning diverse industries. The evolving consumer preferences, coupled with the proliferation of choices available to them, have heightened the significance of retaining existing customers. Central to this endeavor is the ability to predict and preempt customer churn, a phenomenon that poses a significant threat to a company's revenue streams and long-term viability.

**1.1 Background**

The advent of artificial intelligence (AI) and machine learning (ML) techniques has revolutionized the way businesses analyze and interpret customer data. Traditional methods of customer retention, relying on heuristic rules or manual analysis, are increasingly being supplanted by sophisticated predictive analytics powered by AI and ML algorithms. These techniques enable companies to sift through vast volumes of data, discern intricate patterns, and forecast customer behavior with remarkable accuracy.

**1.2 Motivation**

The impetus behind embarking on this project stems from the recognition of the pivotal role that AI and ML can play in helping businesses navigate the complex terrain of customer churn. The imperative to proactively manage churn has never been more pressing, as companies strive to fortify their market positions and sustain growth amidst fierce competition. By harnessing the predictive prowess of AI and ML, businesses can unlock invaluable insights into the drivers of churn, thereby empowering them to devise targeted retention strategies and foster enduring customer relationships.

**1.3 Objective**

At the heart of this project lies a multifaceted objective aimed at leveraging AI and ML techniques to develop a robust customer churn prediction system. By interrogating historical customer data through the lens of advanced analytics, we seek to uncover latent patterns and trends that underpin churn behavior. Subsequently, our aim is to evaluate an array of machine learning algorithms, spanning from conventional models like logistic regression to sophisticated ensemble methods like gradient boosting and neural networks. Through rigorous experimentation and iterative refinement, we endeavor to craft a predictive model endowed with high accuracy and reliability. Ultimately, our goal is to distill actionable insights from the model's predictions, furnishing businesses with the tools needed to stem churn and enhance customer retention rates.

**1.4 Scope**

The ambit of this project encompasses a comprehensive exploration of the myriad factors influencing churn behavior, spanning from demographic attributes to transactional patterns and customer interactions. Drawing upon real-world datasets sourced from [mention your data source if any], we endeavor to construct a holistic understanding of churn dynamics across diverse industry verticals. Moreover, our scope extends to the evaluation of an extensive repertoire of ML algorithms, encompassing both traditional statistical techniques and cutting-edge deep learning architectures. Through this expansive scope, we aim to furnish businesses with a nuanced understanding of churn dynamics and equip them with actionable strategies to mitigate attrition.

**1.5 Significance of the Project**

The significance of this project transcends its immediate applications, resonating profoundly with the overarching imperatives of business sustainability and growth. By arming businesses with a potent arsenal of predictive analytics, we aspire to catalyze a paradigm shift in customer retention strategies. Beyond mere cost savings associated with churn mitigation, our endeavor holds the promise of fostering enduring customer relationships, engendering brand loyalty, and galvanizing sustainable growth trajectories. In a landscape fraught with volatility and uncertainty, the insights gleaned from this project serve as a beacon guiding businesses towards resilience and prosperity.

**2. Problem Definition and Requirement**

**2.1 Problem Statement**

The problem addressed in this project is the prediction of customer churn for businesses operating in various industries. Customer churn, also known as customer attrition, refers to the phenomenon where customers discontinue their relationship with a company or cease using its products or services. Identifying and predicting customer churn is crucial for businesses to implement proactive retention strategies and mitigate revenue loss.

**2.2 Software Requirements**

The development and implementation of the customer churn prediction model require the following software tools and technologies:

* Programming Language: Python or R for data preprocessing, analysis, and model development.
* Data Analysis Libraries: Pandas, NumPy, and SciPy for data manipulation and analysis.
* Machine Learning Libraries: Scikit-learn, TensorFlow, or PyTorch for building and training predictive models.
* Visualization Tools: Matplotlib, Seaborn, or Plotly for data visualization and model performance evaluation.
* Integrated Development Environment (IDE): Jupyter Notebook or Google Colab for interactive development and experimentation.
* Version Control: Git for managing code versions and collaboration among team members.

**2.3 Hardware Requirements**

The hardware requirements for developing and deploying the customer churn prediction model are relatively modest and can be accommodated by standard computing devices, including:

* Processor: Multi-core processor (Intel Core i5 or higher) for efficient data processing and model training.
* Memory: Minimum 8GB RAM for handling large datasets and running machine learning algorithms.
* Storage: Adequate storage space (at least 256GB SSD) for storing datasets, code, and model files.
* Graphics Processing Unit (GPU): Optional but recommended for accelerating deep learning model training (NVIDIA GeForce GTX or RTX series).
* Internet Connection: Stable internet connection for accessing online resources, libraries, and datasets.

**2.4 Data Sets**

The success of the customer churn prediction model depends significantly on the quality and relevance of the dataset used for training and evaluation. Ideally, the dataset should contain historical customer data, including:

* Customer Attributes: Demographic information such as age, gender, location, income, etc.
* Usage Patterns: Transaction history, purchase frequency, product/service usage, etc.
* Customer Interactions: Customer support interactions, feedback, complaints, etc.
* Churn Label: Binary label indicating whether a customer has churned (1) or not (0).

Several publicly available datasets and commercial sources provide relevant customer churn data, including telecom companies, subscription-based services, e-commerce platforms, and financial institutions.

**2.5 Data Preprocessing**

Prior to model training, the raw dataset needs to be preprocessed to address missing values, handle categorical variables, normalize numerical features, and split the data into training and testing sets. Additionally, feature engineering techniques may be employed to extract relevant features and enhance the predictive power of the model.

**3. Proposed Design/Methodology**

**3.1 Overview**

**The proposed design and methodology for the customer churn prediction system entail a systematic approach encompassing data collection, preprocessing, feature engineering, model selection, training, evaluation, and deployment. This section delineates the workflow, methodologies, algorithms employed, and the structural framework of the project.**

**3.2 Data Collection**

**The foundational step in constructing the customer churn prediction model involves gathering comprehensive datasets encapsulating historical customer information, transactional data, and interaction logs. Leveraging both proprietary datasets and publicly available repositories, such as Kaggle or UCI Machine Learning Repository, ensures a diverse and representative sample. The collected data undergoes rigorous scrutiny to ensure relevance, completeness, and adherence to privacy regulations.**

**3.3 Data Preprocessing**

**Upon acquisition, the raw data undergoes meticulous preprocessing to rectify anomalies, handle missing values, and normalize features. Techniques such as imputation, outlier detection, and scaling are employed to enhance data quality and consistency. Additionally, categorical variables are encoded, and feature scaling is performed to render the data amenable to subsequent modelling.**

**3.4 Feature Engineering**

**Feature engineering constitutes a pivotal stage wherein domain knowledge is harnessed to derive insightful features encapsulating nuanced aspects of customer behavior. Techniques such as aggregations, transformations, and interaction terms are applied to distill meaningful insights from raw data. Domain-specific features such as recency, frequency, and monetary value (RFM) are constructed to capture temporal dynamics and transactional patterns.**

**3.5 Model Selection**

**A plethora of machine learning algorithms are evaluated to ascertain their efficacy in predicting customer churn. Ranging from traditional statistical models like logistic regression to ensemble methods such as random forests and gradient boosting, each algorithm is subjected to rigorous evaluation based on performance metrics such as accuracy, precision, recall, and F1-score. Furthermore, advanced deep learning architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are explored to uncover complex nonlinear relationships inherent in the data.**

**3.6 Training and Evaluation**

**The selected models undergo extensive training on the preprocessed dataset, leveraging techniques such as cross-validation to assess generalization performance. Model hyperparameters are optimized via grid search or Bayesian optimization to maximize predictive accuracy and robustness. Subsequently, model performance is rigorously evaluated on a holdout validation set, with a focus on discerning false positives and false negatives to inform strategic decision-making.**

**3.7 Deployment**

**The culmination of the project entails deploying the trained model into production environments, facilitating real-time predictions on incoming customer data. An intuitive user interface is developed to enable seamless interaction with the prediction system, empowering stakeholders to glean actionable insights and devise targeted retention strategies. Furthermore, continuous monitoring mechanisms are established to track model performance and recalibrate parameters as per evolving business requirements.**

**3.8 Schematic Diagram**

**Below is a schematic diagram illustrating the proposed design and workflow of the customer churn prediction system:**

**3.8.1 File Structure**

**The project's file structure is meticulously organized to facilitate seamless collaboration and reproducibility:**

* **data/: Comprises raw and processed datasets.**
* **notebooks/: Houses Jupyter notebooks detailing data preprocessing, model development, and evaluation.**
* **models/: Contains serialized model objects and associated metadata.**
* **scripts/: Encompasses utility scripts for data preprocessing, feature engineering, and model evaluation.**
* **reports/: Hosts project documentation, including the project report, presentation slides, and technical documentation.**

**3.8.2 Algorithms Used**

**The customer churn prediction model harnesses an array of machine learning algorithms, including but not limited to:**

* **Logistic Regression**
* **Decision Trees**
* **Random Forests**
* **Support Vector Classifiers (SVC)**
* **Ad Boost Classifier**
* **K-Means Clustering**

**4. Result**

**4.1 Logistic Regression**

Use in the Code

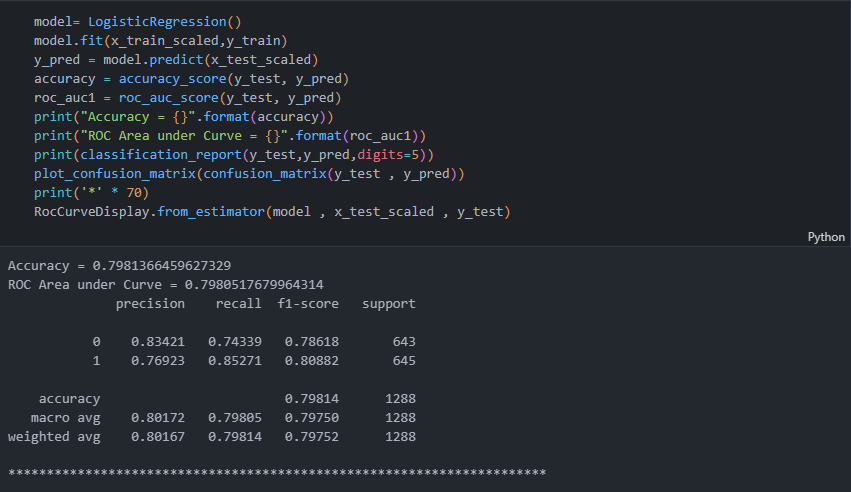
Logistic regression is one of the classification algorithms employed in our code for customer churn prediction. It is initialized as LogisticRegression() and trained on the scaled training data (x\_train\_scaled and y\_train). Predictions are then made on both the training and testing datasets using logistic regression, and accuracy scores are computed to evaluate its performance.

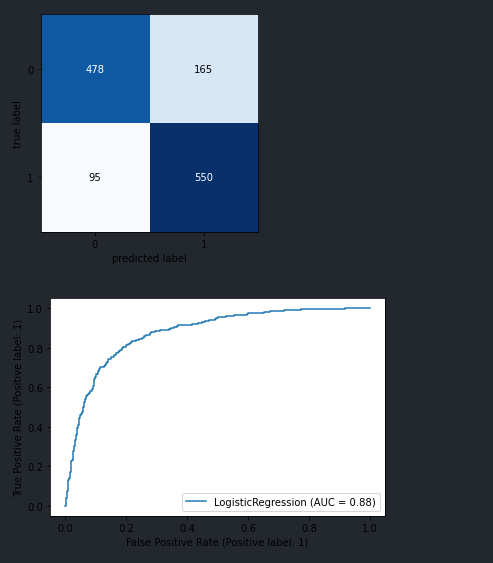
We have used Logistic Regression due to the following reasons:-

Interpretability: Logistic regression provides easily interpretable results, making it conducive for extracting insights into the factors influencing customer churn. The coefficients of the model indicate the strength and direction of the relationship between predictor variables and the likelihood of churn.

Efficiency: Logistic regression is computationally efficient and can handle large datasets with relatively low computational resources. This makes it suitable for real-time applications and scenarios where scalability is essential.

Binary Classification: Logistic regression is well-suited for binary classification tasks, such as predicting customer churn, where the outcome variable has two possible states (churn or no churn).





Accuracy of Logistic regression model is 79%

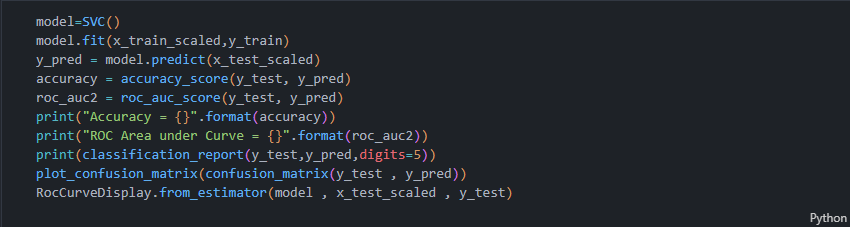
**4.2 Support Vector Classification**

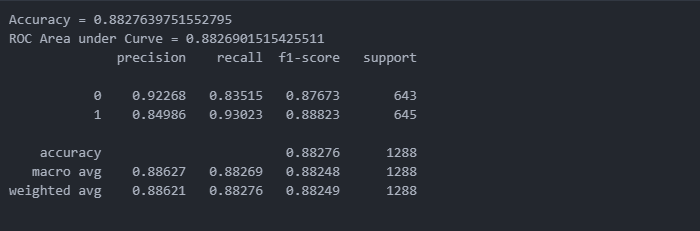
Use in the Code

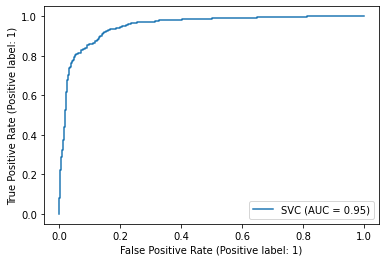
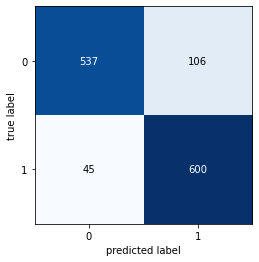
In our project, Support Vector Machine (SVM) is employed as a classification algorithm for predicting customer churn. The SVM model is initialized as SVC() and trained on the scaled training data (x\_train\_scaled and y\_train). Predictions are then made on the testing dataset (x\_test\_scaled), and performance metrics such as accuracy, ROC area under the curve (AUC), and classification report are computed to evaluate the model's effectiveness.

Why Support Vector Machine (SVM)?

1. Non-Linearity: SVM can efficiently model non-linear decision boundaries through the use of kernel functions, enabling it to capture complex relationships in the data. This flexibility makes it suitable for scenarios where the relationship between predictor variables and the target variable is non-linear.
2. Margin Maximization: SVM aims to maximize the margin between the decision boundary and the support vectors, resulting in a robust and generalizable model. By focusing on the most informative data points (support vectors), SVM mitigates the risk of overfitting and improves generalization performance.
3. High-Dimensional Spaces: SVM performs well in high-dimensional feature spaces, making it suitable for datasets with a large number of features. It can handle sparse data efficiently and is robust to the curse of dimensionality.







Accuracy of Support Vector Classification is 88.276 %

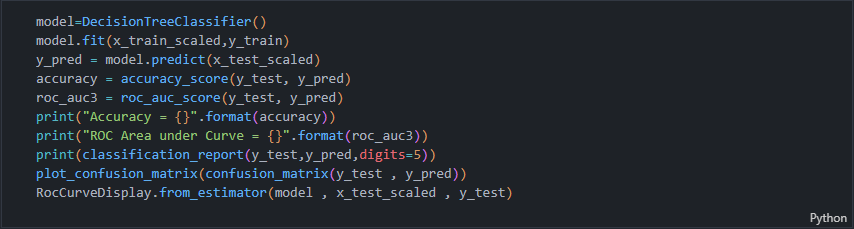
**4.3 Decision Tree Classifier**

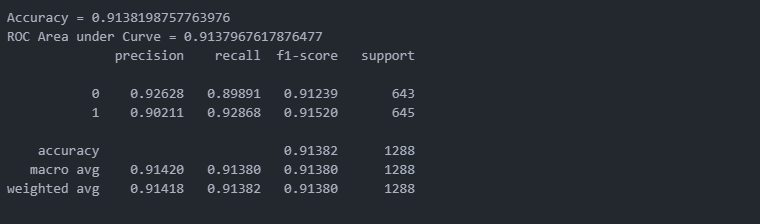
**Use in the Code**

In our project, a Decision Tree classifier is utilized as a classification algorithm for predicting customer churn. The Decision Tree model is initialized as **DecisionTreeClassifier()** and trained on the scaled training data (**x\_train\_scaled** and **y\_train**). Predictions are then made on the testing dataset (**x\_test\_scaled**), and various performance metrics such as accuracy, ROC area under the curve (AUC), and the classification report are computed to assess the model's performance.

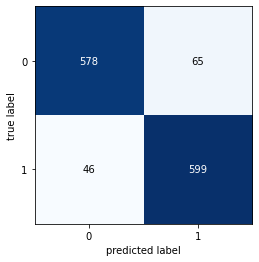
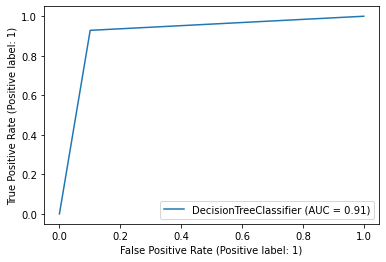
**Why Decision Tree Classifier?**

1. **Interpretability**: Decision Trees provide intuitive explanations for predictions by representing decision rules as a tree structure. This transparency allows stakeholders to understand the factors driving customer churn and derive actionable insights from the model.
2. **Non-Linearity**: Decision Trees can capture non-linear relationships between predictor variables and the target variable, making them suitable for scenarios where the relationship is not linear.
3. **Feature Importance**: Decision Trees inherently rank features based on their importance in predicting the target variable. This information can be leveraged to identify key drivers of customer churn and prioritize retention efforts accordingly.





+



Accuracy of Decision Tree Classification is 91.38 %

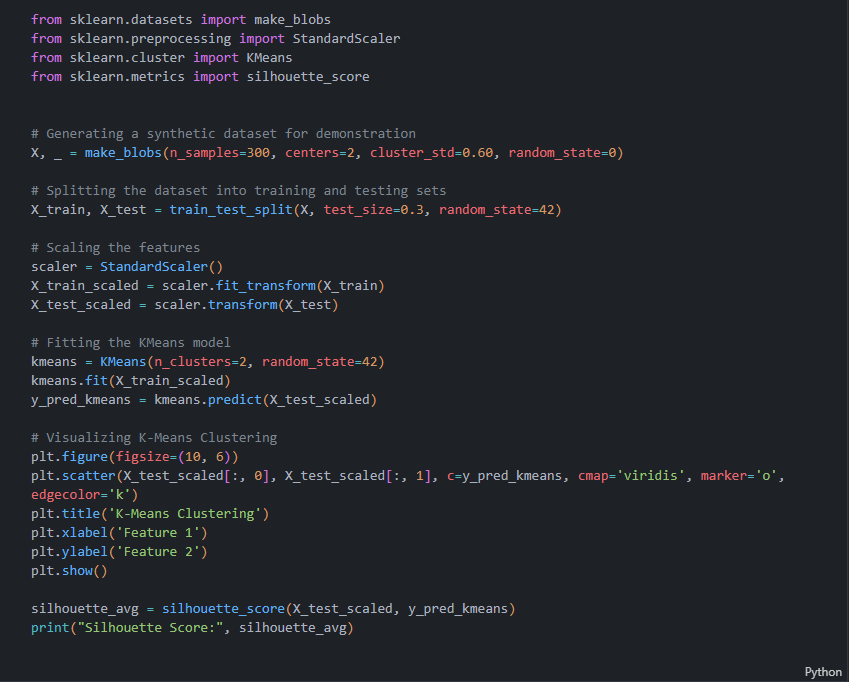
**4.4 K-Means Clustering**

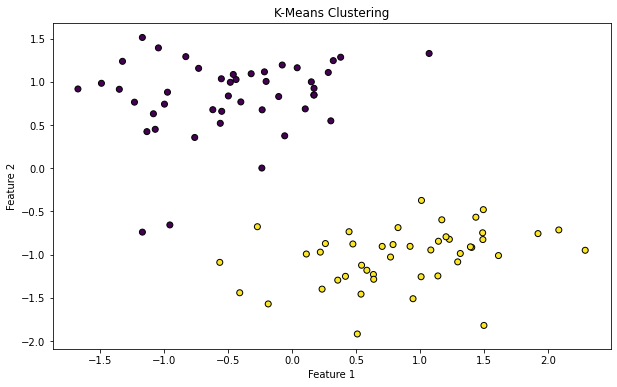
**Use in the Code**

In our project, K-Means clustering is employed to segment customers based on their features and behavior. The synthetic dataset is generated using **make\_blobs**, and then split into training and testing sets. The features are scaled using **StandardScaler** to ensure uniformity across different scales. K-Means clustering is then applied to the scaled testing dataset (**X\_test\_scaled**) to partition the data into two clusters. The resulting clusters are visualized using a scatter plot, and the silhouette score is computed to evaluate the clustering performance.

**Why K-Means Clustering?**

1. **Customer Segmentation**: K-Means clustering facilitates customer segmentation by grouping together customers with similar characteristics or behaviors. This segmentation enables businesses to tailor marketing strategies, product offerings, and customer service initiatives to meet the diverse needs of different customer segments.
2. **Identifying Customer Patterns**: K-Means clustering helps uncover underlying patterns and trends in customer data, such as purchase behavior, preferences, and engagement levels. By identifying distinct customer segments, businesses can gain insights into the factors influencing customer satisfaction, loyalty, and churn.
3. **Market Segmentation**: In addition to customer segmentation, K-Means clustering is used for market segmentation, where it partitions the market into distinct segments based on demographic, geographic, or psychographic attributes. This segmentation aids businesses in targeting specific market segments with tailored marketing campaigns and product offerings.





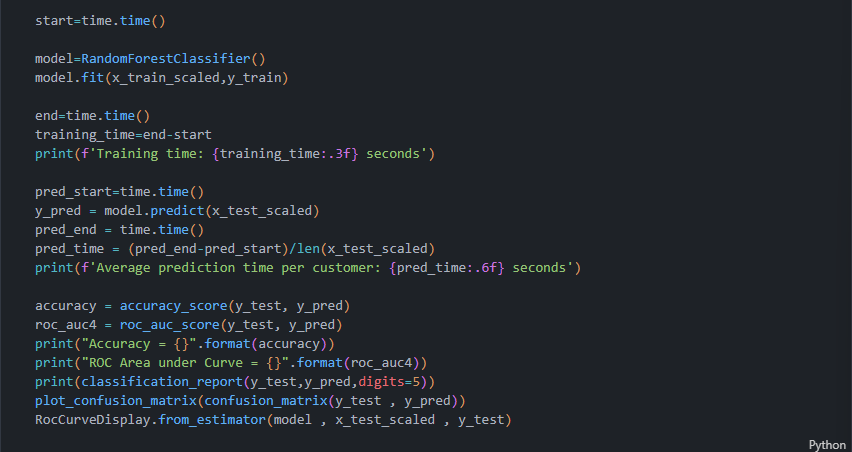
**4.5 Random Forest Classification**

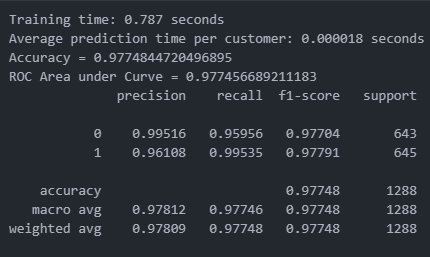
**Use in the Code**

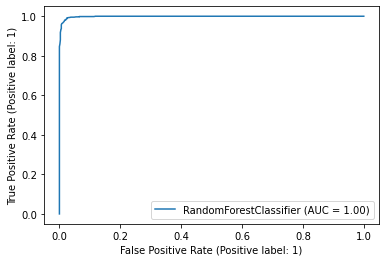
The provided code measures the training time, prediction time, and evaluates the performance of the Random Forest Classifier model on the test dataset. It initializes a Random Forest Classifier model, fits it to the scaled training data (**x\_train\_scaled** and **y\_train**), and makes predictions on the scaled testing data (**x\_test\_scaled**). Performance metrics such as accuracy, ROC area under the curve (AUC), and the classification report are computed and visualized to assess the model's effectiveness.

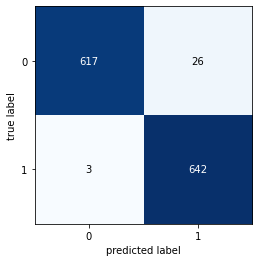
**Why Random Forest Classifier?**

1. **Ensemble Learning**: Random Forest Classifier combines the predictions of multiple decision trees, thereby reducing the variance and improving the overall model performance. By aggregating the predictions of individual trees, Random Forest Classifier achieves higher accuracy and generalization ability compared to single decision trees.
2. **Feature Importance**: Random Forest Classifier ranks features based on their importance in predicting the target variable. This feature importance analysis helps identify key drivers of customer churn, enabling businesses to prioritize retention efforts and allocate resources effectively.
3. **Robustness to Overfitting**: Random Forest Classifier mitigates the risk of overfitting by constructing diverse decision trees through random sampling of features and data points. This ensemble approach enhances the model's robustness to noise and outliers in the data, leading to more reliable predictions.









Accuracy of Random Forest Classification is 97.74%

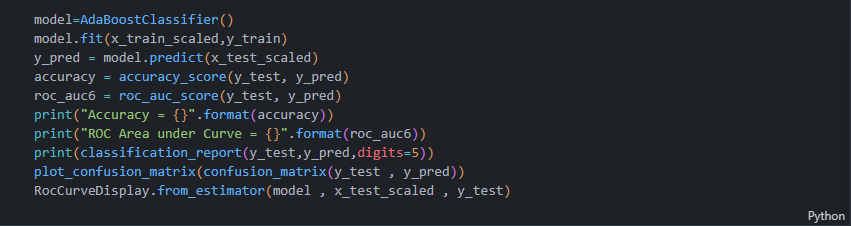
**4.6 Ada Boost Classification**

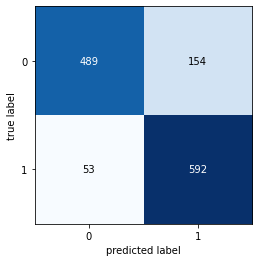
**Use in the Code**

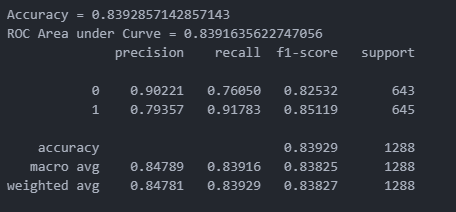
The provided code snippet trains an AdaBoostClassifier model on the scaled training data (**x\_train\_scaled** and **y\_train**) and evaluates its performance on the test data (**x\_test\_scaled**). It computes metrics such as accuracy, ROC area under the curve (AUC), and generates a classification report and confusion matrix to assess the model's effectiveness in predicting customer churn.

**Why AdaBoostClassifier?**

1. **Ensemble Learning**: AdaBoostClassifier employs an ensemble of weak learners, typically decision trees with a single split (stumps), to create a strong classifier. By iteratively combining the predictions of multiple weak learners, AdaBoostClassifier improves predictive accuracy and generalization ability.
2. **Weighted Training**: AdaBoostClassifier assigns higher weights to misclassified instances in each iteration, thereby focusing on the most challenging samples and adapting its training strategy to difficult-to-classify instances. This adaptive boosting technique enhances the model's performance, especially in the presence of class imbalance.
3. **Robustness to Overfitting**: AdaBoostClassifier mitigates the risk of overfitting by combining the predictions of multiple weak learners, each focusing on different subsets of the data. This ensemble approach reduces variance and improves the model's generalization ability, making it suitable for diverse datasets.







Accuracy of ADA Boost Classification is 83.90 %

**5. Reference**

1. DataSet - <https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction/data> .

2. Logistic Regression - [Logistic Regression in Machine Learning - GeeksforGeeks](https://www.geeksforgeeks.org/understanding-logistic-regression/)

3. Random Forest Classification - [Random Forest Algorithm in Machine Learning - GeeksforGeeks](https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/)

4. GitHub -