**NAME: MAFFO YOKIE BEUTCHA REGINE**

**Report on Customer Churn Prediction**

**Introduction**

Customer churn prediction aims to identify customers who are likely to stop using a service, enabling companies to take proactive measures to retain them. This report outlines the analysis and modeling efforts using a synthetic telecom churn dataset. The process involves data preparation, model training, evaluation, and comparison of multiple models to ensure robust predictions.

**Code Explanation**

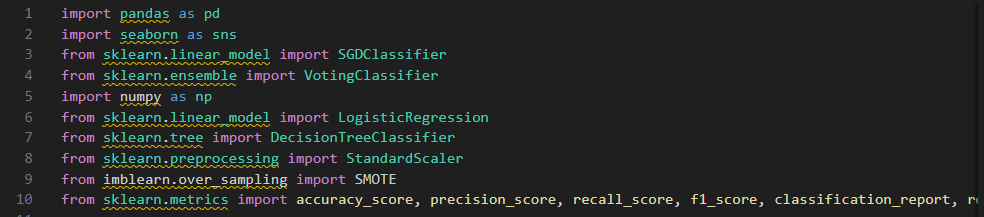
1. **Importing Libraries**

Figure 11. Importing Libraries

1. **pandas**: Used for data manipulation and analysis.

* **seaborn**: A visualization library based on matplotlib, used for creating attractive plots.
* **SGDClassifier**: Implements stochastic gradient descent for classification.
* **VotingClassifier**: Combines multiple models to improve predictions.
* **numpy**: A library for numerical operations.
* **LogisticRegression and DecisionTreeClassifier**: Two machine learning algorithms for classification tasks.
* **StandardScaler**: Standardizes features by removing the mean and scaling to unit variance.
* **SMOTE**: A technique to handle imbalanced datasets by generating synthetic samples.
* **sklearn.metrics**: Contains functions for evaluating model performance.

**2. Data Upload and Inspection**

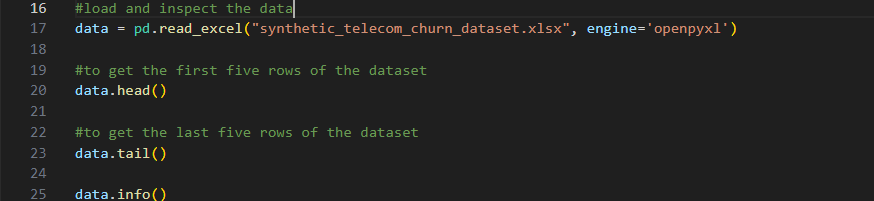
* 

Figure 2 **Data Upload and Inspection**

* The dataset is loaded from an Excel file using pandas. The head() and tail() methods provide a quick look at the first and last few rows, allowing for an initial assessment of the data structure.
* data.info() gives a summary of the DataFrame, including the number of non-null entries in each column, which helps in identifying potential missing values.

**3. Data Cleaning**

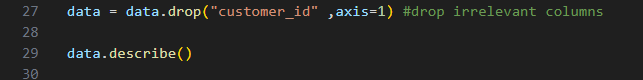
* 

Figure 3 Data Cleaning

* The customer\_id column is dropped as it does not provide predictive value.
* data.describe() generates descriptive statistics for numerical columns, offering insight into the distribution and range of data.

**4. Encoding Categorical Data**

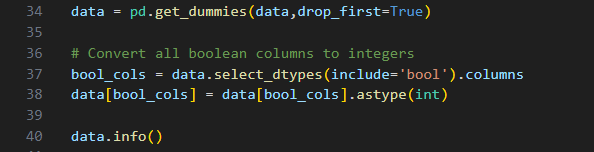


Figure 4 Encoding Categorical Data

* Categorical variables like 'region' are converted into dummy variables using get\_dummies(), which helps in preparing the data for machine learning models.
* Boolean columns are converted to integers for consistency and to ensure compatibility with modeling algorithms.

**5. Handling Missing Values**



Figure 5Handling Missing Values

* The sum of missing values in each column is calculated using isnull().sum(). This helps identify which features may require imputation or further investigation.

**6. Outlier Detection and Handling**

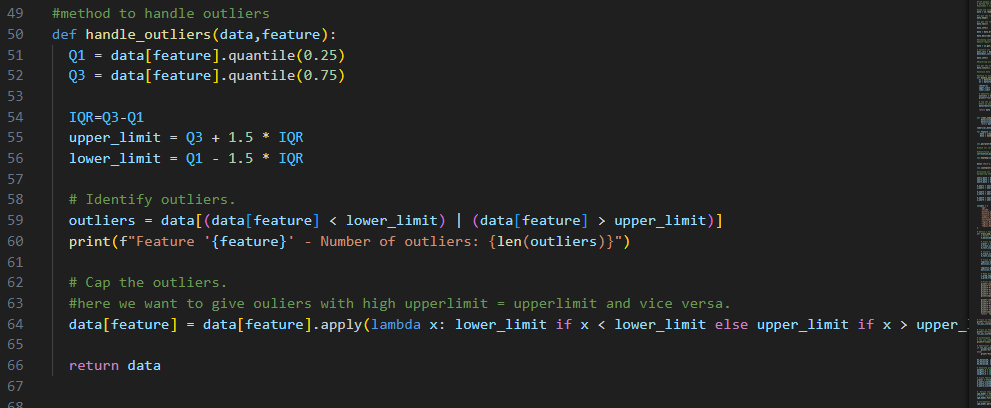


Figure 6Outlier Detection and Handling

* The handle\_outliers function identifies outliers using the Interquartile Range (IQR) method. It calculates the first and third quartiles (Q1 and Q3) and then determines the upper and lower limits for outlier detection.
* Outliers are capped to the defined limits, preventing them from adversely affecting model training and predictions.

**7. Cleaning Numeric Columns**

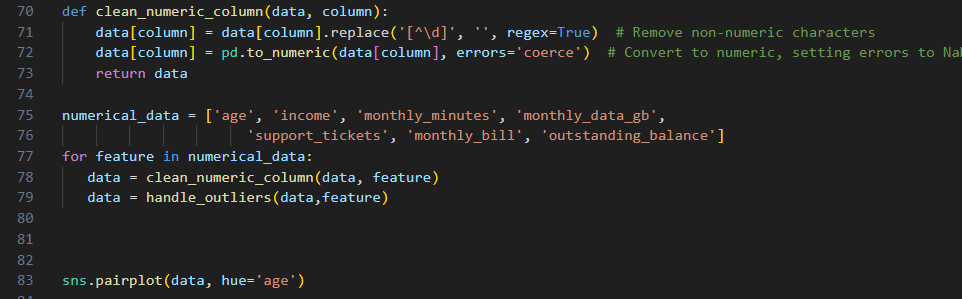


Figure 7Cleaning Numeric Columns

* The clean\_numeric\_column function removes non-numeric characters and converts the column to a numeric type, handling errors by setting them to NaN.
* The specified numerical features in numerical\_data are then processed for cleaning and outlier handling, ensuring that the dataset is ready for model training.

**8. Data Visualization**

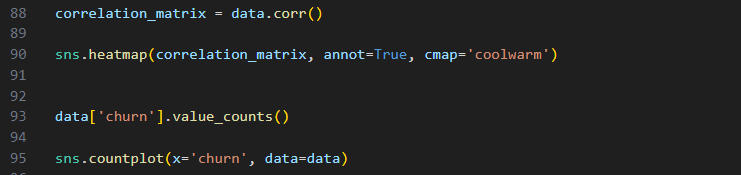


Figure 8Data Visualization

* A pair plot is created to visualize relationships between features, colored by the 'age' variable. This helps in exploring potential correlations visually.
* The correlation matrix is computed to identify linear relationships between features. A heatmap visualizes these correlations, aiding in feature selection for modeling.

**9. Churn Distribution Analysis**

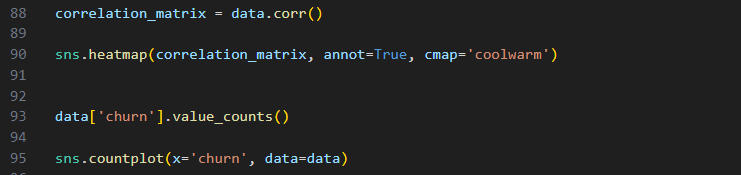


Figure 9Churn Distribution Analysis

* value\_counts() provides a breakdown of the target variable ('churn'), indicating how many customers churned versus those who did not.
* A count plot visualizes this distribution, which is vital for understanding class imbalance in the dataset.

**10. Preparing Data for Training**

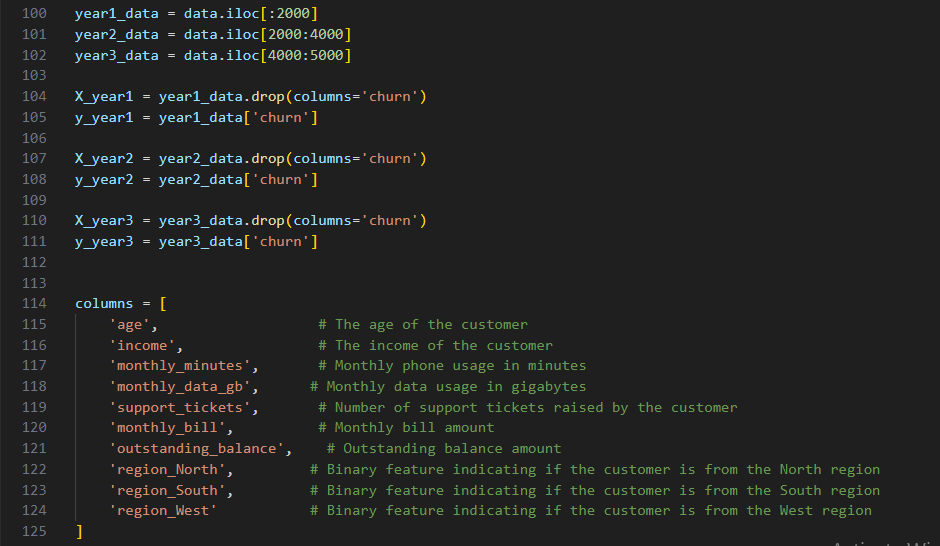
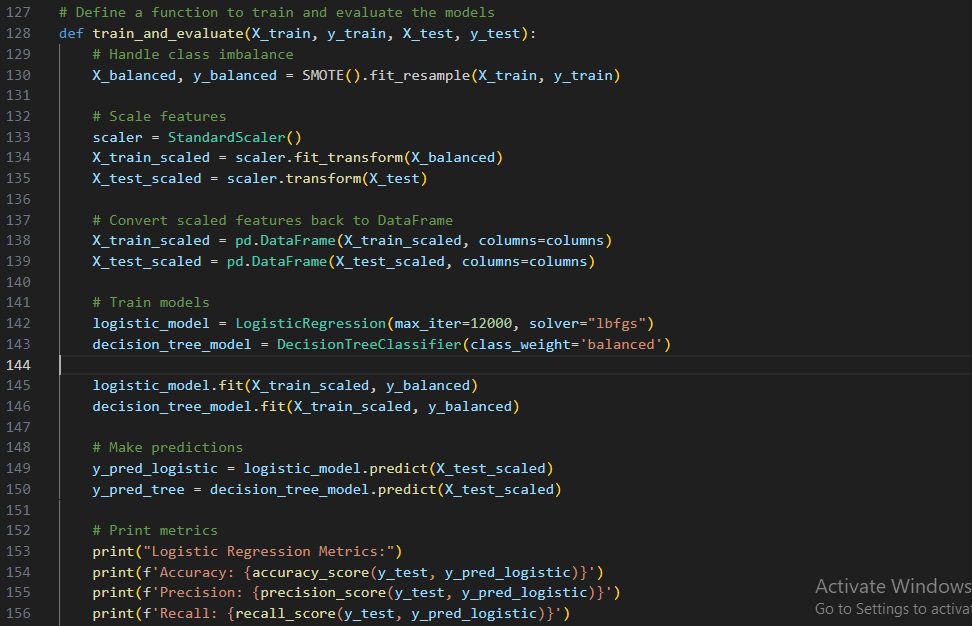


Figure 10Preparing Data for Training

* The dataset is split into three parts representing different years. This temporal split allows for evaluating model performance over time.
* Features (X\_year1, X\_year2, X\_year3) and the target variable (y\_year1, y\_year2, y\_year3) are defined for each year.
* A list of relevant features for modeling is defined. These features are expected to have predictive power regarding customer churn.

**12. Model Training and Evaluation Function**



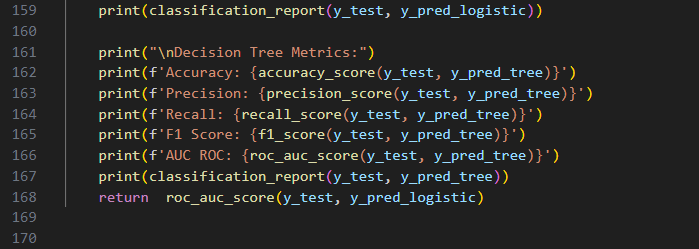
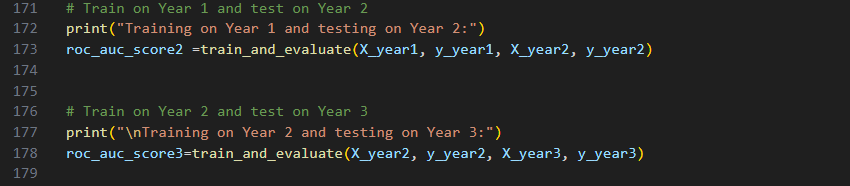


Figure 11Model Training and Evaluation Function

* The train\_and\_evaluate function handles model training and evaluation. It first addresses class imbalance using SMOTE to create synthetic samples for the minority class.
* Features are standardized using StandardScaler, and both Logistic Regression and Decision Tree models are trained. The function then computes and prints various performance metrics, including accuracy, precision, recall, F1 score, and AUC ROC.

**13. Model Training on Different Years**



* The model is first trained on data from Year 1 and tested on Year 2, followed by training on Year 2 and testing on Year 3. This setup helps evaluate how well the model generalizes to unseen data.

**14. Performance Comparison**

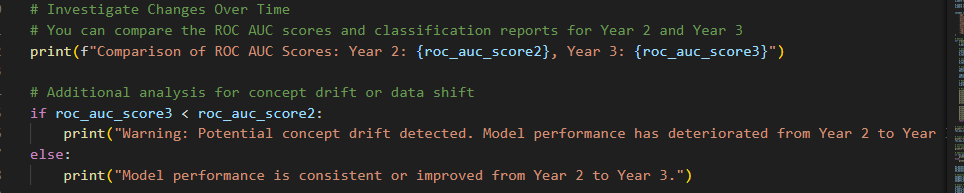


Figure 12Performance Comparison

* The ROC AUC scores from the evaluations are printed to compare model performance across years. A warning is issued if performance worsens, indicating potential concept drift, which suggests that the data distribution may have changed over time.

**15. Balancing Classes for Each Year**

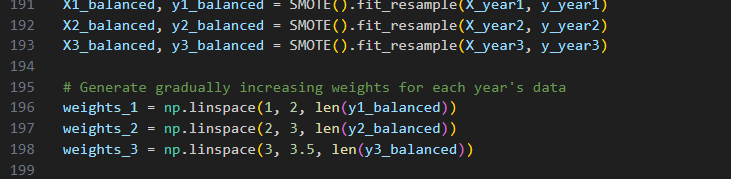


Figure 13Balancing Classes for Each Year

* The SMOTE technique is applied again to balance the training datasets for each year, ensuring that the models trained on each year's data have a more balanced class distribution.
* Gradually increasing sample weights are generated for each year's data to give more importance to more recent data during model training.

**16. Online Learning with SGDClassifier**

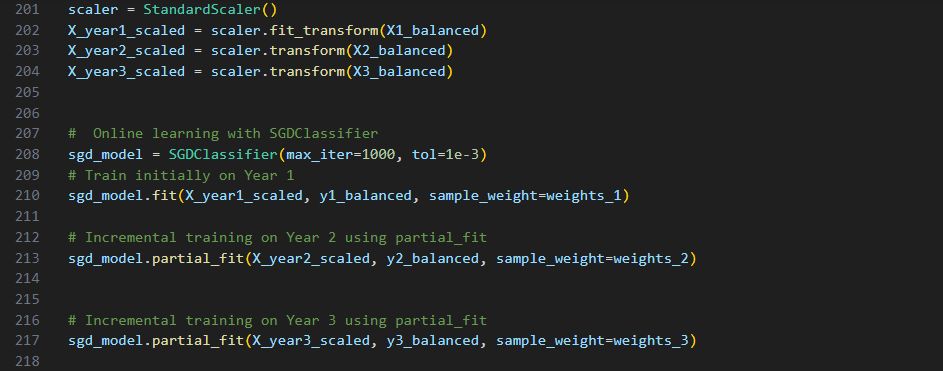


Figure 14 Online Learning with SGDClassifier

* The features for each year are scaled using StandardScaler to ensure consistency across different datasets.
* An SGDClassifier is initialized for online learning. It is first trained on Year 1 and then incrementally updated with data from Years 2 and 3 using the partial\_fit method. This approach is beneficial for adapting to new data without retraining from scratch.

**17. Model Evaluation on Each Year**

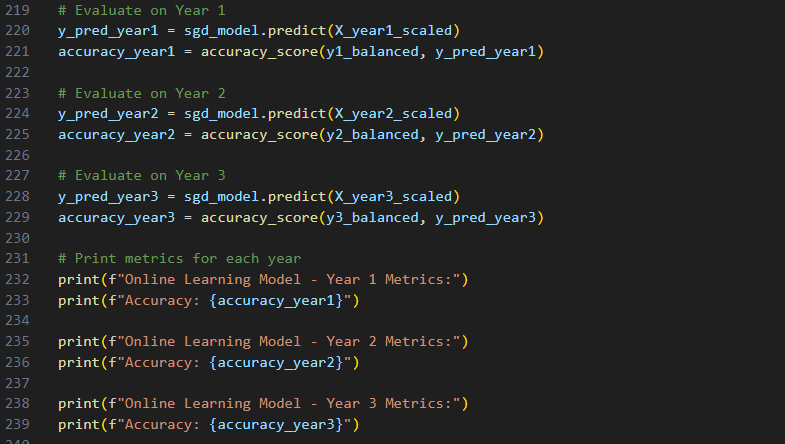


Figure 15Model Evaluation on Each Year

* The trained model is evaluated on each year's data, with predictions made and accuracy scores calculated. This step provides insights into how well the model performs across different time frames.

**18. Training Separate Models for Years 2 and 3**

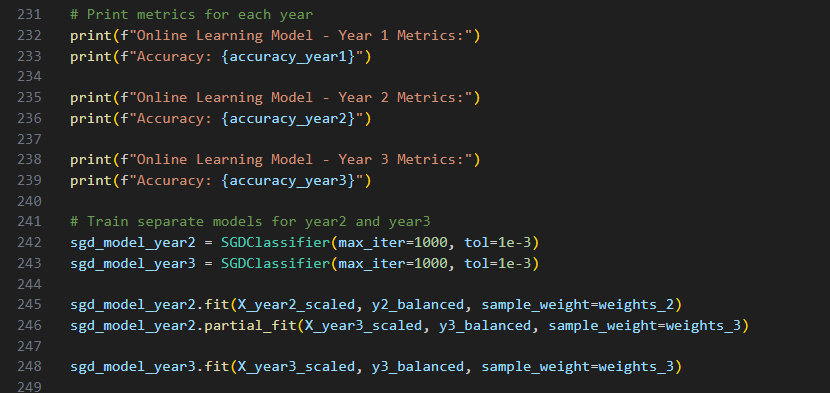


Figure 16Training Separate Models for Years 2 and 3

* Separate SGDClassifier instances are created for Years 2 and 3. The model for Year 2 is trained and then incrementally updated with Year 3 data. This ensures that models can specialize based on the specific data available for each year.

**19. Metrics for Year 3 Evaluation**

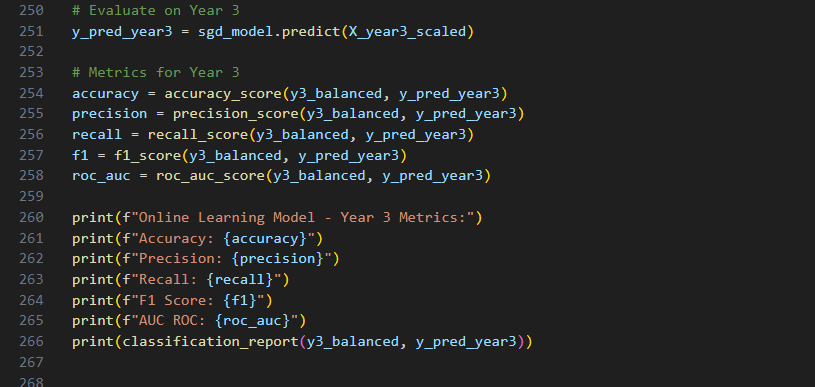


Figure 17Metrics for Year 3 Evaluation

* The performance of the model is evaluated for Year 3, with metrics such as accuracy, precision, recall, F1 score, and AUC ROC computed and printed. This comprehensive evaluation highlights the model’s strengths and weaknesses in predicting churn.

**20. Ensemble Model Training: Voting Classifier**

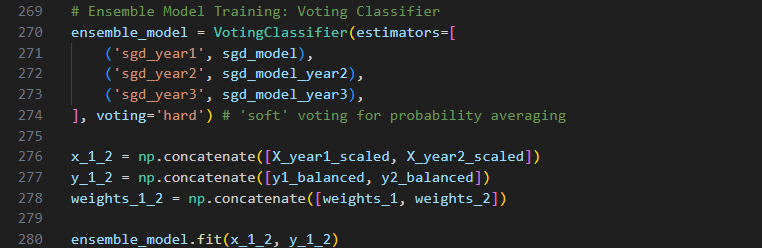


Figure 18Ensemble Model Training: Voting Classifier

* An ensemble model using a Voting Classifier is created, combining predictions from the SGDClassifier models trained on each year. This method leverages the strengths of each individual model to improve overall predictive performance.
* The training data for Years 1 and 2 is concatenated to form a comprehensive dataset for fitting the ensemble model.

**21. Predictions with the Ensemble Model**

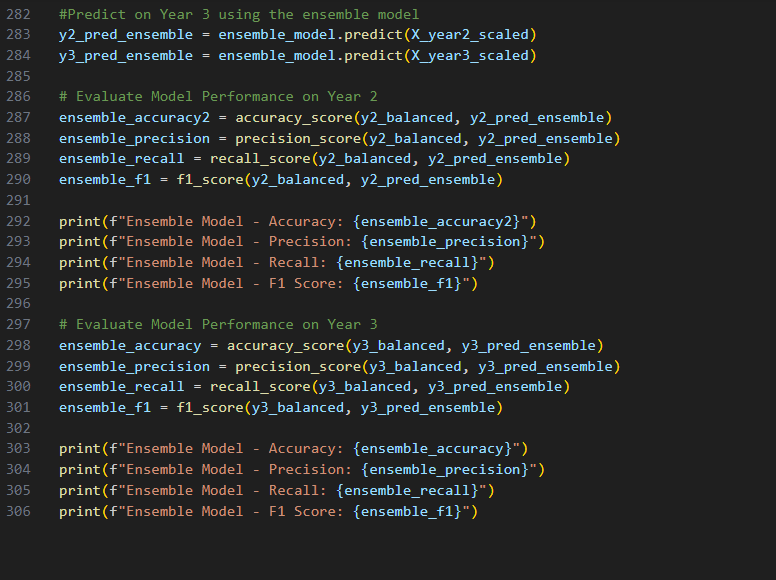


Figure 19Predictions with the Ensemble Model