Bank Customer Churn Prediction

Problem Statement

Background: Customer churn, also known as customer attrition, is a critical challenge faced by banks and financial institutions. It refers to the phenomenon where customers terminate their relationship with a bank, either by closing their accounts or discontinuing the use of their banking services. Customer churn can have a significant impact on a bank's profitability and customer base.

To mitigate customer churn and retain valuable customers, banks need to identify potential churners in advance and take proactive measures to retain them. Predictive modeling techniques can be employed to forecast customer churn based on historical customer data and various attributes.

Dataset Information: The dataset used in this project is named "Churn_Modelling.csv". It is a CSV (comma-separated values) file that contains customer information and churn status for a bank. The dataset consists of the following attributes:

RowNumber: The row number in the dataset.

CustomerId: Unique identifier for each customer.

Surname: Customer's surname (last name).

CreditScore: Customer's credit score.

Geography: Customer's geographical location (country).

Gender: Customer's gender (Male or Female).

Age: Customer's age.

Tenure: Number of years the customer has been with the bank.

Balance: Bank account balance.

NumOfProducts: Number of bank products the customer has purchased.

HasCrCard: Whether the customer has a credit card (0 = No, 1 = Yes).

IsActiveMember: Whether the customer is an active member (0 = No, 1 = Yes).

EstimatedSalary: Estimated salary of the customer.

Exited: Whether the customer has churned (0 = No, 1 = Yes).

Problem Statement: The main objective of this project is to develop a predictive model that can accurately predict whether a bank customer is likely to churn based on the given customer attributes. By identifying potential churners, the bank can proactively take measures to retain those customers and minimize customer attrition.

The project involves the following steps:

Data Preprocessing: The dataset is loaded, and any necessary data cleaning or transformation is performed to prepare the data for analysis.

Exploratory Data Analysis (EDA): The dataset is analyzed to gain insights into the relationship between the customer attributes and churn status. Visualizations and statistical summaries are used to explore the data and identify patterns or trends.

Feature Engineering: New features are created or existing features are transformed to enhance the predictive power of the model. For example, the balance-to-salary ratio and tenure-to-age ratio are introduced as additional features.

Model Fitting and Selection: Various predictive models are trained and evaluated on the dataset. This includes logistic regression, support vector machines (SVM), random forest, and extreme gradient boosting (XGBoost). Grid search and cross-validation techniques are used to tune the model hyperparameters and select the best-performing model.

Model Evaluation: The performance of the selected model is assessed using evaluation metrics such as accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC-ROC). The focus is on predicting churn accurately, and the trade-off between precision and recall is considered.

Test Data Prediction: The final model is applied to unseen test data to evaluate its performance on new observations. The accuracy and AUC-ROC are calculated to assess the model's generalization ability.

By developing an accurate churn prediction model, the bank can proactively identify customers at risk of churn and implement targeted retention strategies to maintain a strong customer base and improve overall business performance.

Framework

Importing Required Libraries: Begin by importing the necessary libraries such as NumPy, Pandas, Matplotlib, and Seaborn for data manipulation, visualization, and analysis.

Loading the Dataset: Load the dataset from the CSV file, "Churn_Modelling.csv", using the pd.read_csv() function. Review the shape of the dataset to understand its dimensions.

Data Exploration and Preprocessing:

Checking for Missing Values: Use the isnull().sum() function to identify if there are any missing values in the dataset. Fortunately, the code finds no missing values in this case.

Reviewing Attribute Relevance: Analyze the unique count for each variable using df.nunique() to determine the importance and relevance of each attribute. Decide which attributes are necessary for churn prediction and exclude irrelevant attributes such as "RowNumber", "CustomerId", and "Surname".

Exploring Snapshot Data: Review the top rows of the remaining dataset using df.head() to observe any patterns or anomalies. Ask questions about the data, such as the date relevance, customers with balances despite exiting, and the definition of an active member.

Checking Variable Data Types: Use df.dtypes to check the data types of the variables in the dataset, identifying categorical and continuous variables for further analysis.

Exploratory Data Analysis (EDA):

Proportion of Customer Churn: Visualize the proportion of churned and retained customers using a pie chart. Note that approximately 20% of customers have churned, indicating a baseline for predictive modeling.

Analyzing Categorical Variables: Create count plots for categorical variables (e.g., "Geography", "Gender", "HasCrCard", "IsActiveMember") to examine their relationship with customer churn. Observe patterns such as the impact of geography, gender, credit cards, and inactive membership on churn.

Analyzing Continuous Variables: Use box plots to compare the distributions of continuous variables (e.g., "CreditScore", "Age", "Tenure", "Balance", "NumOfProducts", "EstimatedSalary") for churned and retained customers. Identify any significant differences that could impact churn prediction.

Feature Engineering:

Creating New Features: Introduce new features to capture additional information that may influence churn prediction. For example, calculate the balance-to-salary ratio and tenure-to-age ratio to assess their impact on churn.

Reordering Columns: Rearrange the columns to ensure the target variable, "Exited", is the first column, followed by the continuous variables and categorical variables.

Data Preparation for Model Fitting:

Data Scaling: Perform min-max scaling on the continuous variables using the minimum and maximum values from the training data.

One-Hot Encoding: Encode the categorical variables using one-hot encoding, creating binary columns for each unique category.

Data Pipeline for Test Data: Define a data preparation pipeline function, DfPrepPipeline(), to apply the same transformations to test data as done with the training data.

Model Fitting and Selection:

Defining Model Candidates: Import the required models, including Logistic Regression, SVM, Random Forest, and XGBoost. Set up the necessary scoring functions and parameter grids for hyperparameter tuning.

Grid Search Cross-Validation: Perform grid search cross-validation on each model using the training data to find the best hyperparameters for each model.

Model Evaluation: Evaluate the performance of each model by analyzing classification reports, precision, recall, and area under the ROC curve (AUC-ROC). Consider the trade-off between precision and recall for identifying potential churners.

Fit Best Models:

Fit Selected Models: Train the best-performing models, such as logistic regression, polynomial logistic regression, SVM with RBF kernel, SVM with polynomial kernel, random forest, and XGBoost, using the training data.

Review Model Accuracy: Assess the precision and recall of each model on the training data, considering the ability to identify churned customers accurately.

Test Data Prediction:

Data Transformation for Test Data: Apply the same data transformations used for the training data to preprocess the test data using the DfPrepPipeline() function.

Model Performance on Test Data: Evaluate the selected model's performance on the unseen test data by calculating precision, recall, and AUC-ROC scores.

Conclusion:

Summarize the results and performance of the predictive model on both training and test data.

Highlight the model's ability to predict customer churn and recommend actions to retain valuable customers.

Code Explanation

The code you provided is aimed at predicting customer churn in a bank. Churn refers to customers who stop using the bank's services or close their accounts. By predicting churn, the bank can take proactive measures to retain valuable customers.

Here's a breakdown of the code and its workflow:

Importing Required Libraries: This section imports the necessary libraries for data manipulation, visualization, and analysis. Libraries like NumPy, Pandas, Matplotlib, and Seaborn are commonly used in data science projects.

Loading the Dataset: The code loads the dataset from a CSV file called "Churn_Modelling.csv" using the pd.read_csv() function. This dataset contains information about bank customers, such as their demographics, account details, and churn status.

Data Exploration and Preprocessing: This section focuses on exploring the dataset and preparing it for analysis. Here are the key steps:

Checking for Missing Values: The code uses the isnull().sum() function to check if there are any missing values in the dataset. Missing values can cause issues in analysis, but luckily, the code finds no missing values in this case.

Reviewing Attribute Relevance: The code analyzes the unique count for each variable using the df.nunique() function. This step helps determine the importance and relevance of each attribute. In this code, the attributes "RowNumber", "CustomerId", and "Surname" are deemed irrelevant and dropped from the dataset.

Exploring Snapshot Data: The code uses df.head() to display the top rows of the remaining dataset. It raises interesting questions, such as the relevance of the snapshot date, customers with balances despite exiting, and the definition of an active member. These questions provide insights into the data and potential challenges in churn prediction.

Checking Variable Data Types: The code uses df.dtypes to check the data types of variables in the dataset. This step helps identify categorical and continuous variables, which will be important for further analysis.

Exploratory Data Analysis (EDA): This section focuses on analyzing the dataset to gain insights into the relationship between variables and customer churn. Here are the key steps:

Proportion of Customer Churn: The code uses visualization techniques to show the proportion of churned and retained customers using a pie chart. This provides an overview of the churn rate in the dataset.

Analyzing Categorical Variables: The code creates count plots for categorical variables like "Geography", "Gender", "HasCrCard", and "IsActiveMember". These plots help understand how these variables relate to customer churn. For example, it examines the proportion of churned customers across different geographical regions, genders, credit card holders, and active members.

Analyzing Continuous Variables: The code uses box plots to compare the distributions of continuous variables like "CreditScore", "Age", "Tenure", "Balance", "NumOfProducts", and "EstimatedSalary" between churned and retained customers. These plots provide insights into whether these variables significantly differ for churned and retained customers.

Feature Engineering: This section involves creating new features from the existing data that could potentially improve churn prediction. Here are the key steps:

Creating New Features: The code introduces new features like the balance-to-salary ratio and tenure-to-age ratio. These features capture additional information that could impact churn prediction. For example, customers with a higher balance-to-salary ratio might be more likely to churn.

Reordering Columns: The code rearranges the columns to ensure that the target variable, "Exited", is the first column, followed by the continuous variables and categorical variables.

Data Preparation for Model Fitting: This section prepares the dataset for model training by transforming categorical variables into one-hot encoded format and scaling the continuous variables using min-max scaling.

Model Fitting and Selection: This section involves fitting various models to the training data and selecting the best-performing model. Here are the key steps:

Model Fitting: The code fits different models like logistic regression, polynomial logistic regression, SVM (Support Vector Machine), random forest, and XGBoost to the training data using hyperparameter tuning with GridSearchCV.

Reviewing Model Accuracy: The code reviews the accuracy of each model by analyzing classification reports, precision, recall, and AUC-ROC scores. It compares the performance of these models in predicting churned customers.

Fit Best Models: This section fits the selected best models from the previous step on the training data.

Test Data Prediction: This section prepares and preprocesses the test data, applies the same transformations used for the training data, and evaluates the selected model's performance on unseen test data.

Conclusion: The code concludes by summarizing the results and performance of the predictive model on both the training and test data. It highlights the model's ability to predict customer churn and recommends actions to retain valuable customers.

Future Work

Predicting customer churn is an ongoing process that requires continuous improvement and refinement. Here's a detailed plan for future work on this project, along with step-by-step guidance on how to implement it:

Collect Additional Data: To enhance the predictive power of the model, consider collecting additional relevant data. This could include customer interactions, transactional data, customer feedback, or external factors like economic indicators. More comprehensive data can provide deeper insights into customer behavior and improve the accuracy of churn prediction.

Feature Engineering: Explore and engineer new features that capture additional insights into customer churn. Some potential ideas include:

Time-based features: Consider capturing the frequency and recency of customer interactions or transactions.

Customer engagement: Develop features that quantify customer engagement, such as the number of customer service calls, website/app usage, or social media interactions.

Social network influence: Explore features that consider a customer's social connections or influences, as they can impact churn behavior.

Advanced Analytics Techniques: Implement advanced analytics techniques to extract more value from the data and improve prediction accuracy. Some approaches to consider include:

Text mining and sentiment analysis: Analyze customer feedback, reviews, or social media posts to uncover sentiment patterns and identify potential churn indicators.

Customer segmentation: Use clustering algorithms to segment customers based on behavior, preferences, or demographics. This can provide a more personalized approach to churn prediction and retention strategies.

Time series analysis: Apply time series modeling techniques to capture trends and patterns in customer behavior over time, allowing for more accurate predictions.

Ensemble Models and Model Stacking: Explore ensemble models and model stacking techniques to improve prediction performance. Combine multiple models, each with their strengths and weaknesses, to create a more robust and accurate predictive model. This can involve techniques such as bagging, boosting, or model averaging.

Model Interpretability and Explainability: Consider implementing techniques to improve the interpretability and explainability of the predictive model. This is particularly important in banking, where regulatory and compliance requirements necessitate transparency in decision-making. Techniques like feature importance analysis, SHAP values, or LIME can help explain the model's predictions and provide insights into the factors driving customer churn.

Continuous Model Monitoring and Update: Implement a system for continuous monitoring of the model's performance and update it as new data becomes available. This ensures that the model remains effective and up-to-date in predicting churn. Monitor key performance metrics, such as accuracy, precision, recall, and AUC-ROC, and retrain the model periodically using new data.

Step-by-Step Guide:

Data Collection: Gather additional data that can provide valuable insights into customer churn. This could involve accessing internal data sources, partnering with external data providers, or leveraging APIs to collect relevant data.

Feature Engineering: Analyze the collected data and engineer new features that capture important aspects of customer behavior and engagement. This may involve applying domain knowledge, statistical analysis, or machine learning techniques to derive meaningful features.

Data Preparation: Preprocess and transform the data to make it suitable for model training. This includes handling missing values, encoding categorical variables, and scaling numerical variables. Ensure consistency in data preprocessing between the training and future data.

Model Development and Evaluation: Select appropriate machine learning algorithms and train them on the prepared data. Evaluate the performance of different models using appropriate evaluation metrics like accuracy, precision, recall, and AUC-ROC.

Implement Future Work Steps: Implement the future work steps described above, such as advanced analytics techniques, ensemble modeling, model interpretability, and continuous model monitoring. Follow best practices and available libraries or frameworks to implement these steps effectively.

Deploy and Monitor the Model: Deploy the trained model into a production environment, where it can make predictions on new data. Implement a monitoring system to track the model's performance and identify any degradation over time. Regularly evaluate and update the model as new data becomes available or when the model's performance declines.