Customer churn prediction

Problem Description :

Customer churn is the term used to describe the loss of customers from a business or service. It is an important metric for businesses to monitor as it can have a significant impact on revenue and profitability. In this project, we will be predicting customer churn for a company.

Dataset Description:

The dataset used for this project contains information about customers in the company. It includes demographic information such as age, gender, as well as information about the geographic information as location.It also contains information such as customer's subscription length, Total usage(gb),monthly bill. The dataset also includes a binary variable indicating whether the customer has churned or not.

Project Objective:

The objective of this project is to predict whether a customer is likely to churn or not based on their demographic and usage patterns. By identifying customers who are at high risk of churning, the company can take proactive measures to retain them and reduce churn rates.

Framework and Steps:

1. **Data Collection:** Collect the dataset from the data source.

2. **Data Preprocessing:** Preprocess the data by handling missing values, removing duplicates, encoding categorical variables, and scaling numerical variables.

3. **Exploratory Data Analysis (EDA):** Perform EDA to understand the relationship between the target variable and the features, and identify any patterns or trends in the data.

4. **Feature Selection:** Select the relevant features for the prediction model using feature selection techniques.

5. **Model Selection**: Select the appropriate machine learning model based on the problem type and data characteristics.

6. **Model Training:** Train the selected model on the preprocessed data.

7. **Model Evaluation:** Evaluate the performance of the model using appropriate evaluation metrics.

8. **Hyperparameter Tuning:** Tune the hyperparameters of the selected model to improve its performance.

9. **Prediction:** Use the trained model to make predictions on new data.

Code Explanation :

Here is the simple explanation for the code which is provided in the code.py file.

The code is for customer churn prediction using a dataset. The dataset contains information about customers and their churn status. We first load the dataset and perform data exploration, checking for missing values, data types, and correlations between features.

Next, we preprocess the data by encoding categorical features, scaling numerical features, and splitting the data into training and testing sets.

Then, we train several classification models, including Logistic Regression, XGBOOST. We evaluate the models using accuracy, precision, recall, and F1-score metrics.

Finally, we select the best performing model based on evaluation metrics and make predictions on new data to predict whether a customer is likely to churn or not.

How to Run the Code

To run the code, follow the steps below:

1. Download the dataset from the source mentioned in the code.

2. Install the required libraries listed at the top of the code, such as pandas, numpy, scikit-learn, and matplotlib.

3. Open the Python environment or Jupyter Notebook and navigate to the directory where the code is saved.

4. Run each section of the code in order, making sure to install any missing libraries as needed.

5. After training and evaluating the models, select the best performing model based on evaluation metrics and use it to make predictions on new data.

Requirements to Run the Code

The code requires the following libraries to be installed:

- Pandas
- Numpy
- Scikit-learn
- Matplotlib

Future Work :

1. **Feature Engineering:** One potential area for future work is feature engineering. This process involves analyzing the existing features and creating new ones that may be more predictive of customer churn.

2. **Advanced Modeling Techniques:** Another area for future work is exploring more advanced modeling techniques. This could include trying out different algorithms such as gradient boosting, neural networks, or ensemble models to improve the predictive power of the model.

3. **Hyperparameter Tuning**: In order to achieve the best possible performance, the model's hyperparameters must be tuned. This involves adjusting the settings of the model to improve its ability to predict customer churn. Hyperparameter tuning can be performed using tools such as grid search, random search, or Bayesian optimization.

4. **Deploying the Model**: After developing and tuning the model, the next step is to deploy it in a production environment. This involves creating an API that can be integrated with other applications and setting up infrastructure to handle predictions in real-time.

5. **Monitoring and Updating the Model:** Once the model is deployed, it is important to monitor its performance and update it as necessary. This can involve tracking the model's accuracy over time, identifying changes in customer behavior that may impact the model's predictions, and retraining the model periodically to ensure that it remains accurate.

Step-by-Step Guide to Implement Future Work:

1. **Feature Engineering:** To improve the model's performance through feature engineering, follow these steps:

- Analyze the existing features to identify any patterns or correlations with customer churn.

- Create new features based on customer behavior or other relevant data.

- Evaluate the impact of these new features on the model's performance using techniques such as cross-validation or A/B testing.

2. **Advanced Modeling Techniques:** To explore more advanced modeling techniques, follow these steps:

- Research different algorithms and identify ones that may be well-suited to the problem at hand.

- Implement these algorithms and compare their performance to the existing model.

- Use techniques such as ensemble models or model stacking to combine multiple algorithms for improved performance.

3. **Hyperparameter Tuning:** To tune the model's hyperparameters, follow these steps:

- Identify the hyperparameters that are most likely to impact the model's performance.

- Use tools such as grid search, random search, or Bayesian optimization to identify the optimal settings for these hyperparameters.

- Evaluate the impact of these hyperparameter settings on the model's performance using techniques such as cross-validation.