**Telecom Customer Churn Prediction Project**

**Introduction**

The goal of this project is to predict customer churn for a telecom provider. The rate at which customers discontinue using a service over a specified time frame is known as customer churn. The ability to predict customer attrition allows the business to take proactive steps to keep clients, which lowers revenue loss.

**Dataset**

The dataset used in this project contains customer data from a telecommunications company. The dataset includes various features such as customer’s details, account information, and service usage patterns. The target variable is Churn, indicating whether a customer has churned or not.

**MODULE 1: PYTHON**

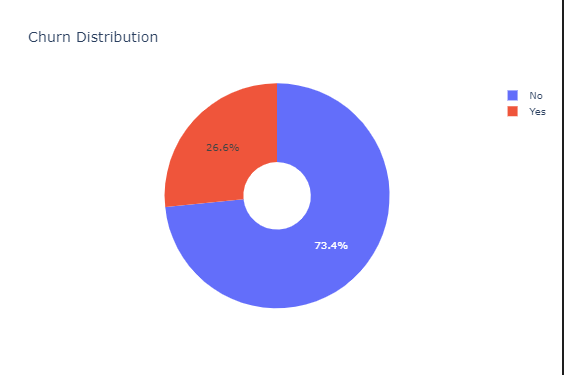
**Data Preprocessing**

Data preprocessing is a crucial step in the machine learning pipeline. The following steps were performed:

1. **Handling Missing Values**: No null values were found
2. **Encoding Categorical Variables**: Converted categorical variables into numerical form using one-hot encoding.
3. **Feature Scaling**: Applied Min-Max scaling to normalize the feature values.

**Exploratory Data Analysis**

Exploratory Data Analysis (EDA) for a telecom churn prediction project involves a thorough examination of the dataset to uncover patterns, correlations, and anomalies that may influence customer retention. A pie chart is used to understand the proportion of customers who have churned and those who have not.

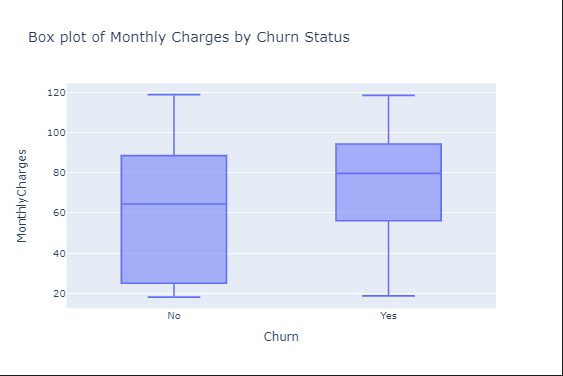
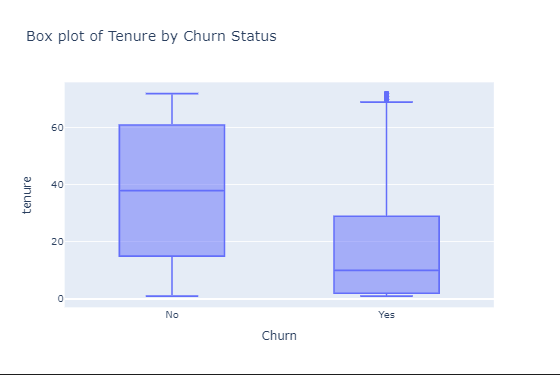


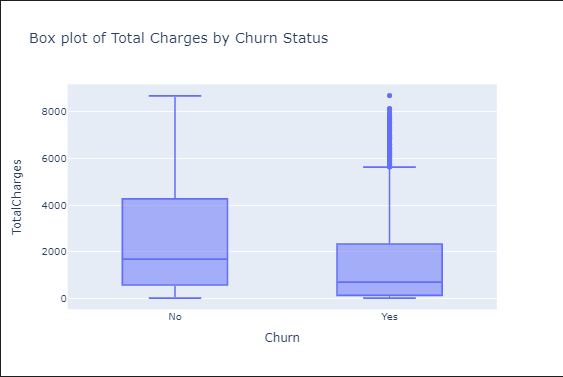
A boxplots for numerical columns like Tenure, Monthly Charges and Total Charges to identify outliers.

1. Tenure: Churned customers tend to have shorter tenures compared to non-churned customers.

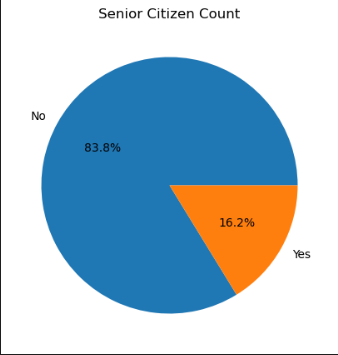
2. Monthly Charges: Churned customers tend to have slightly higher monthly charges.

3. Total Charges: Non-churned customers have a higher range of total charges due to longer tenures.





A pie chart of Senior Citizen shows that type of customers 83.8 % of the customers are not senior citizen and only 16.2% are senior citizen

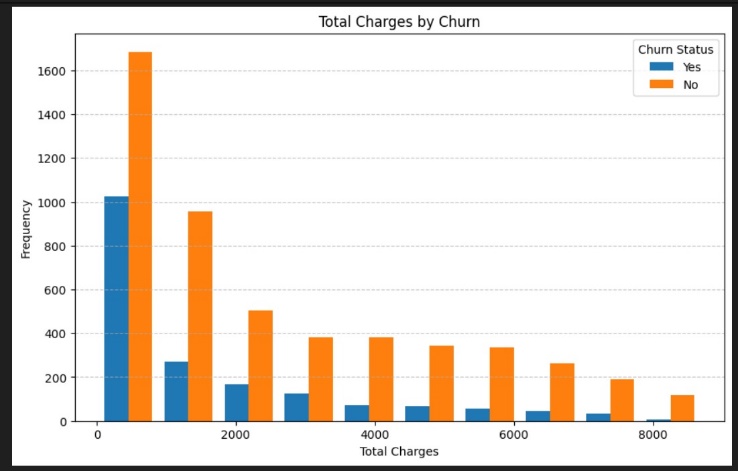
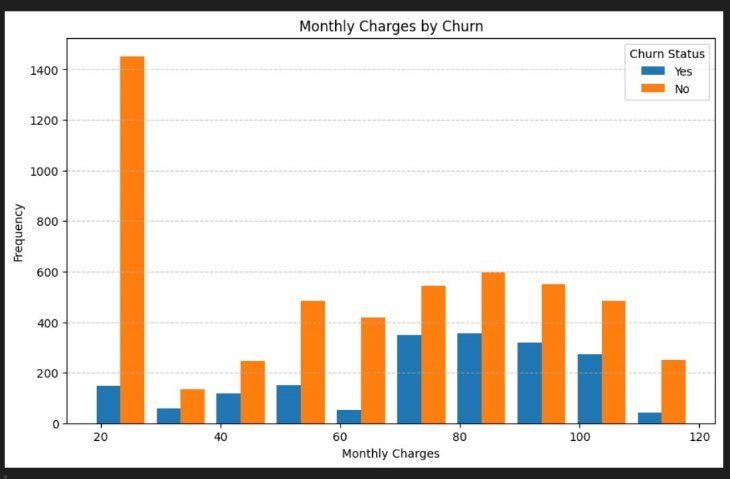


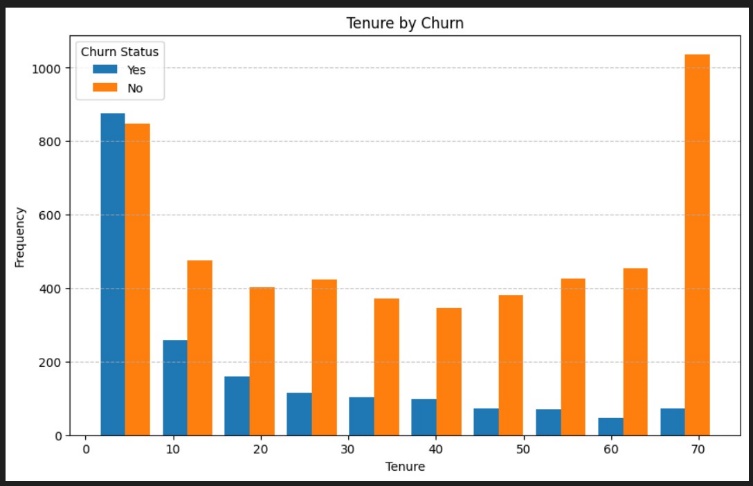
A histogram is used to understand the distribution of Tenure, Monthly charges and total Charges against the target variable ‘Churn’ and shows:

1. Tenure: A higher frequency of churned customers have shorter tenures.

2. Monthly Charges: Higher monthly charges are more common among churned customers.

3. Total Charges: Churned customers have lower total charges, correlating with shorter tenures.



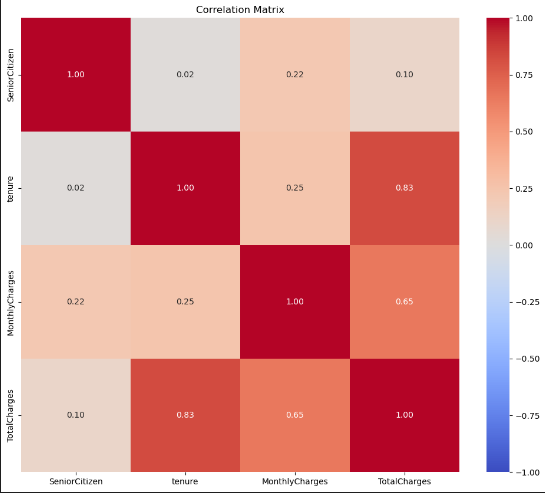


A histogram showing distribution of different features against churn shows

1. Contract type significantly impacts churn rates; customers with month-to-month contracts are more likely to churn compared to those with one or two-year contracts.

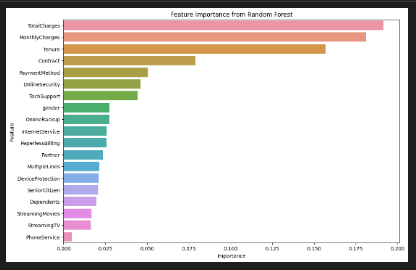
2. Features like 'InternetService', 'OnlineSecurity', 'TechSupport', and 'StreamingTV' show visible differences in churn rates, indicating their potential importance in predicting churn.

Lastly, a correlation matrix shows highly correlated features here Tenure is strongly correlated with total charges, which is expected as longer tenures accumulate higher charges.



**Feature Engineering**

Feature engineering involves creating new features or transforming existing ones to improve model performance. Random Forest Classifier is used to select top 11 important features:



**MODULE 2: AI ALGORITHM**

**Model Selection and Model Tuning**

Several machine learning models were evaluated to identify the best model for predicting customer churn. The models considered include:

1. **Random Forest**
2. **Gradient Boosting**
3. **Logistic Regression**
4. **Decision Tree**

For each model, hyperparameter tuning was performed using RandomSearchCV to find the optimal parameters. The models were evaluated based on their accuracy, training time, and CPU usage.

**Handling Class Imbalance**

The dataset exhibited class imbalance, with a significantly higher number of non-churn customers compared to churn customers. To address this issue, the SMOTEENN (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbors) technique was applied to balance the classes.

**Model Evaluation**

The models were evaluated using a train-test split method. The performance metrics considered include:

1. **Accuracy**: The proportion of correctly predicted instances.
2. **Training Time**: The time taken to train the model.
3. **CPU Usage**: The CPU usage during model training.

**Results**

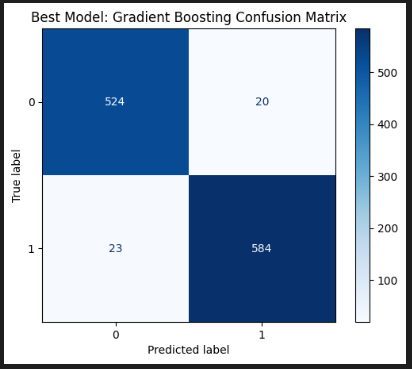
**Original Training Scores**

| **Model** | **Training Time (s)** | **CPU Usage (%)** | **Accuracy** |
| --- | --- | --- | --- |
| Random Forest | 0.69 | 19.1 | 0.79 |
| Gradient Boosting | 0.69 | 9.90 | 0.79 |
| Logistic Regression | 0.18 | 4.80 | 0.79 |
| Decision Tree | 0.03 | 68.7 | 0.73 |

**SMOTEENN Training Scores**

| **Model** | **Training Time (s)** | **CPU Usage (%)** | **Accuracy** |
| --- | --- | --- | --- |
| Random Forest | 0.29 | 11.90 | 0.94 |
| Gradient Boosting | 1.75 | 22.80 | 0.96 |
| Logistic Regression | 8.02 | 25.20 | 0.91 |
| Decision Tree | 0.95 | 3.60 | 0.92 |

**Confusion Matrix**

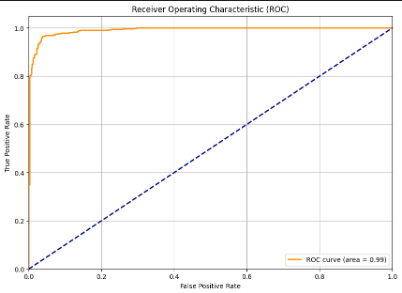
****

**Overall Best Model**

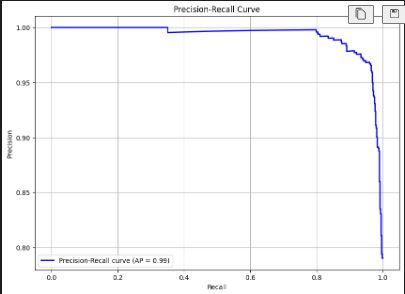
The overall best model was selected based on its accuracy and other performance metrics. The final model and its parameters are as follows:

* **Model**: Gradient Boosting
* **Accuracy**: 96%
* **Training Time**: 1.75 seconds
* **CPU Usage**: 22.80 %

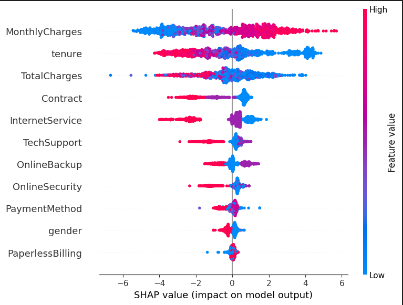
**Receiver Operating Characteristic (ROC)**

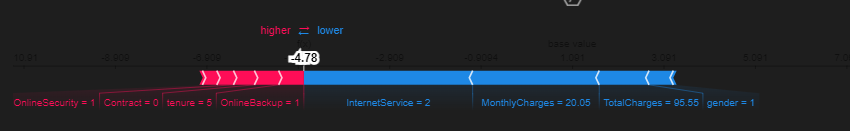
****

**Precision-Recall Curve**

****

**SHAP**

****

****

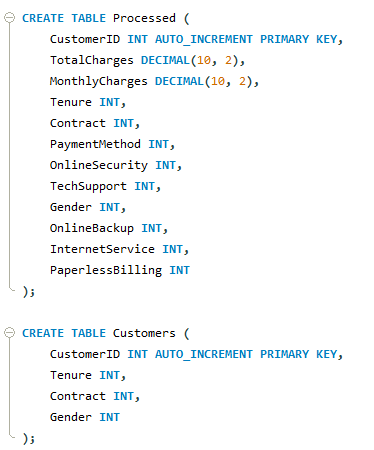
**MODULE 3: SQL**

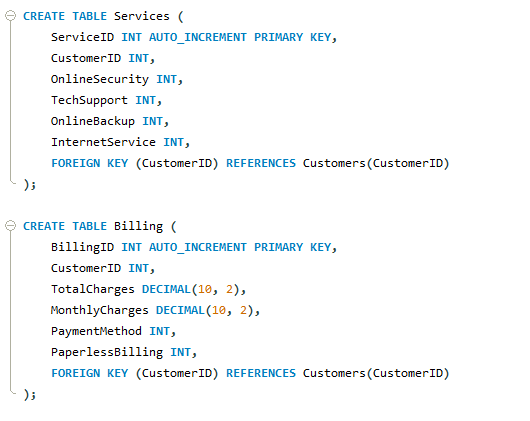
**Database Schema Design:**

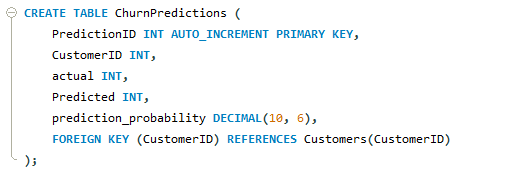
****

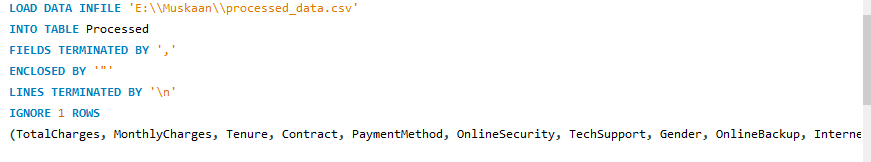
****

**Data Ingestion:**

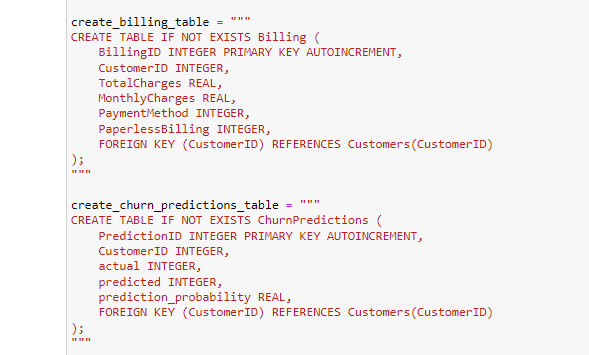
****

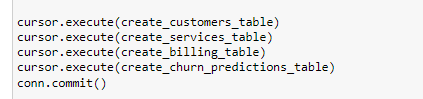
****

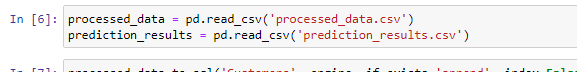
****

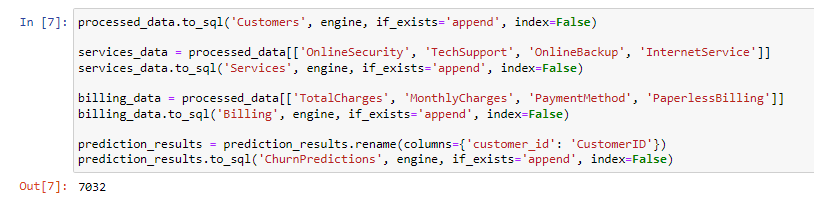
****

****

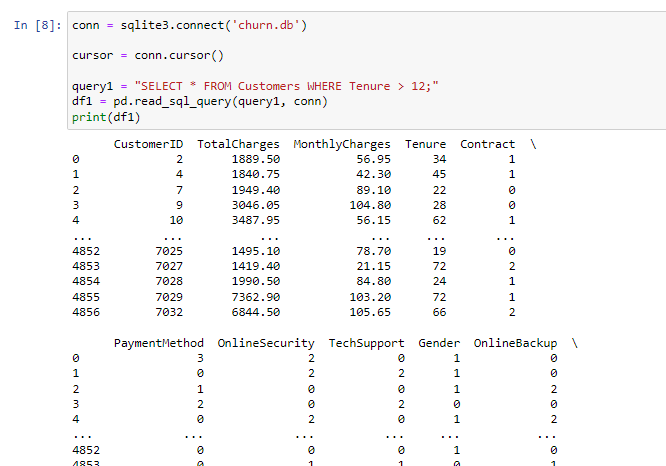
****

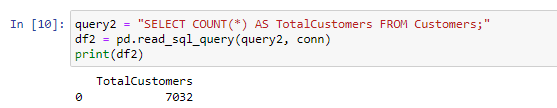
****

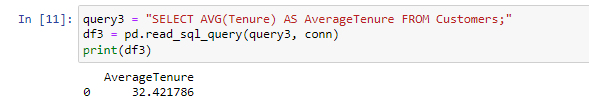
****

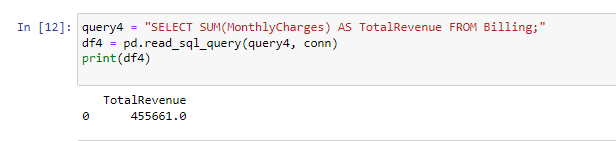
****

**Querying Data:**

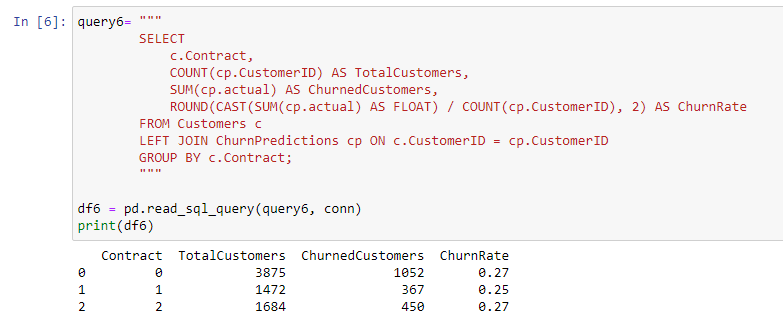
****

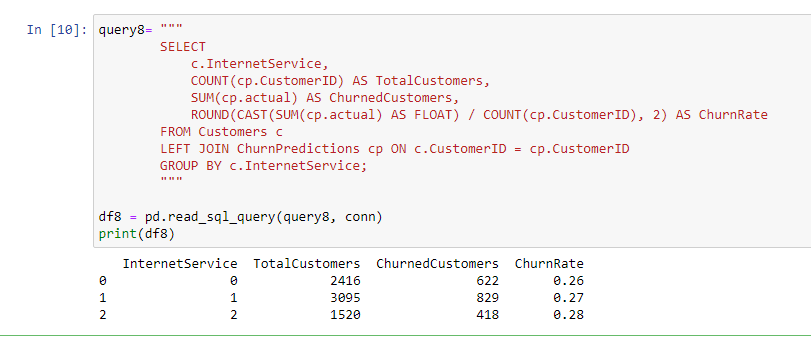
****

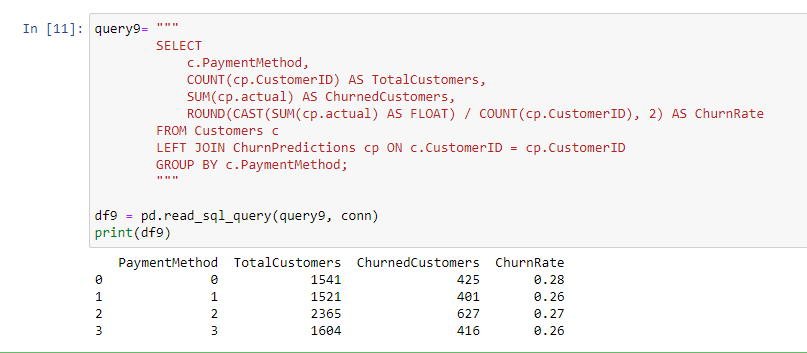
****

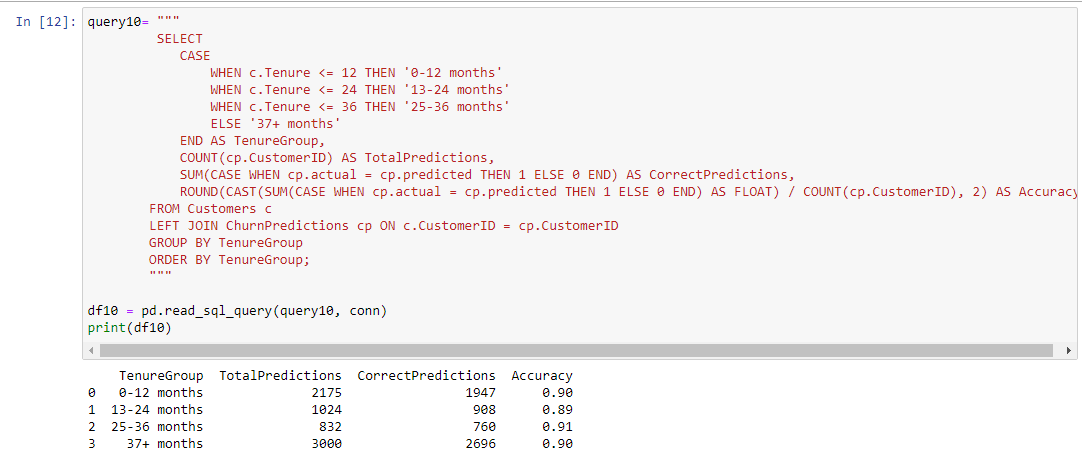
****

**Advanced SQL Analysis:**

****

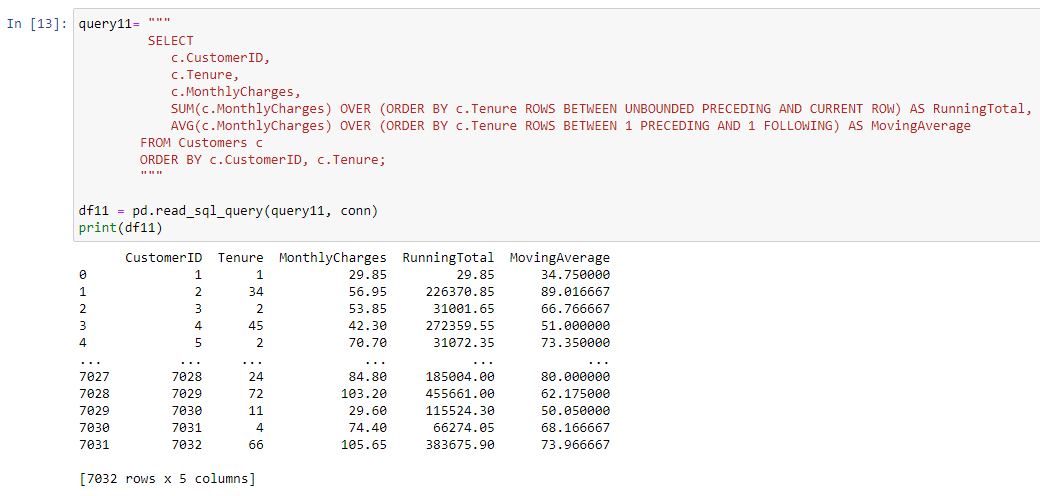
****

****

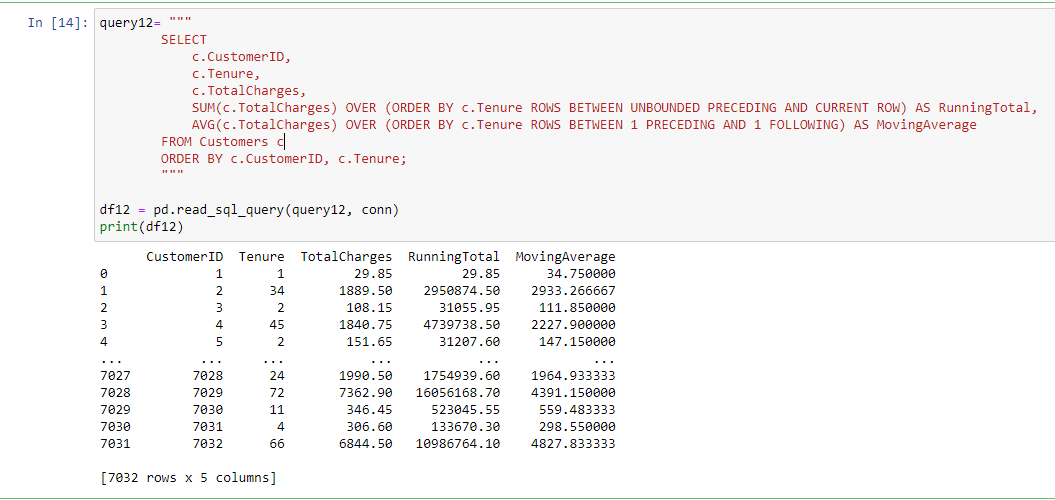
****

**Running Total and Moving Average**

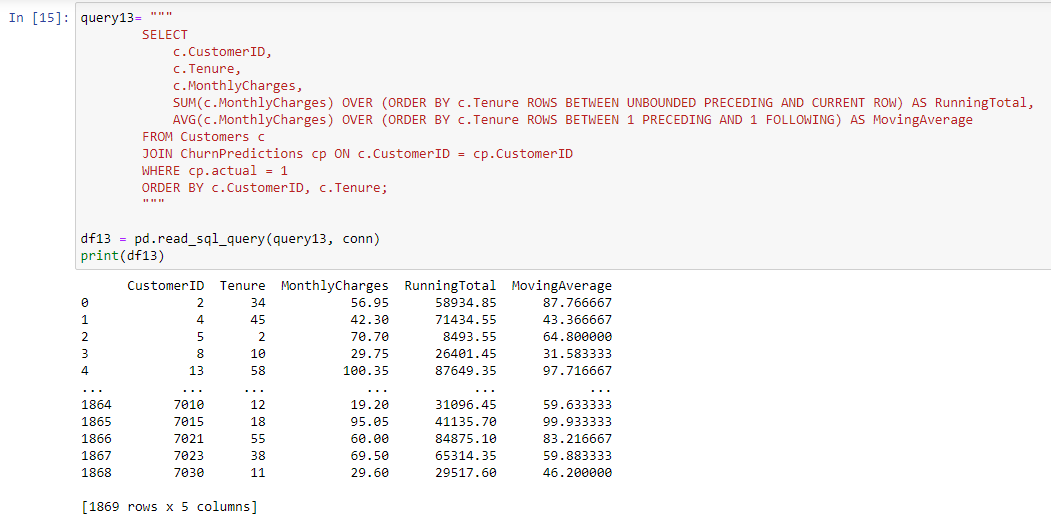
**Monthly Charges:**

****

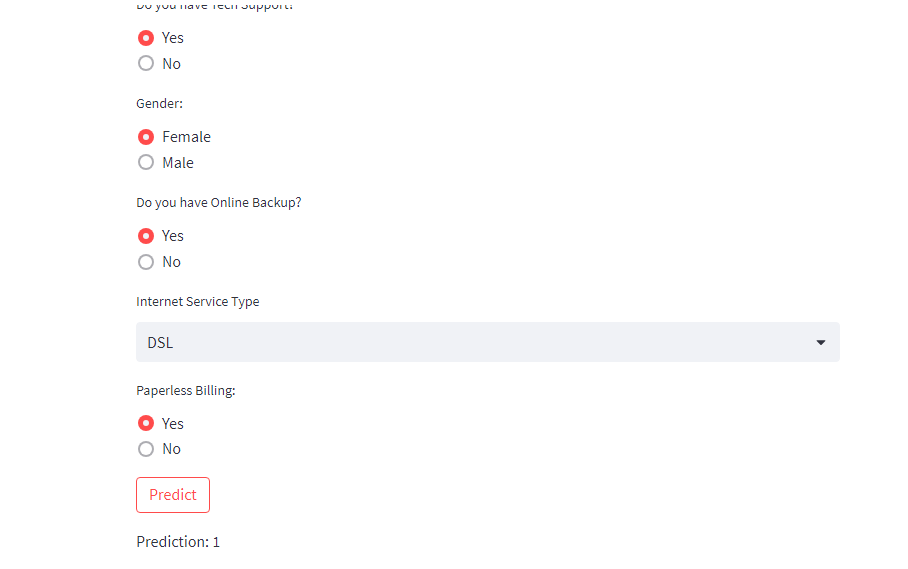
**Total Charges:**

****

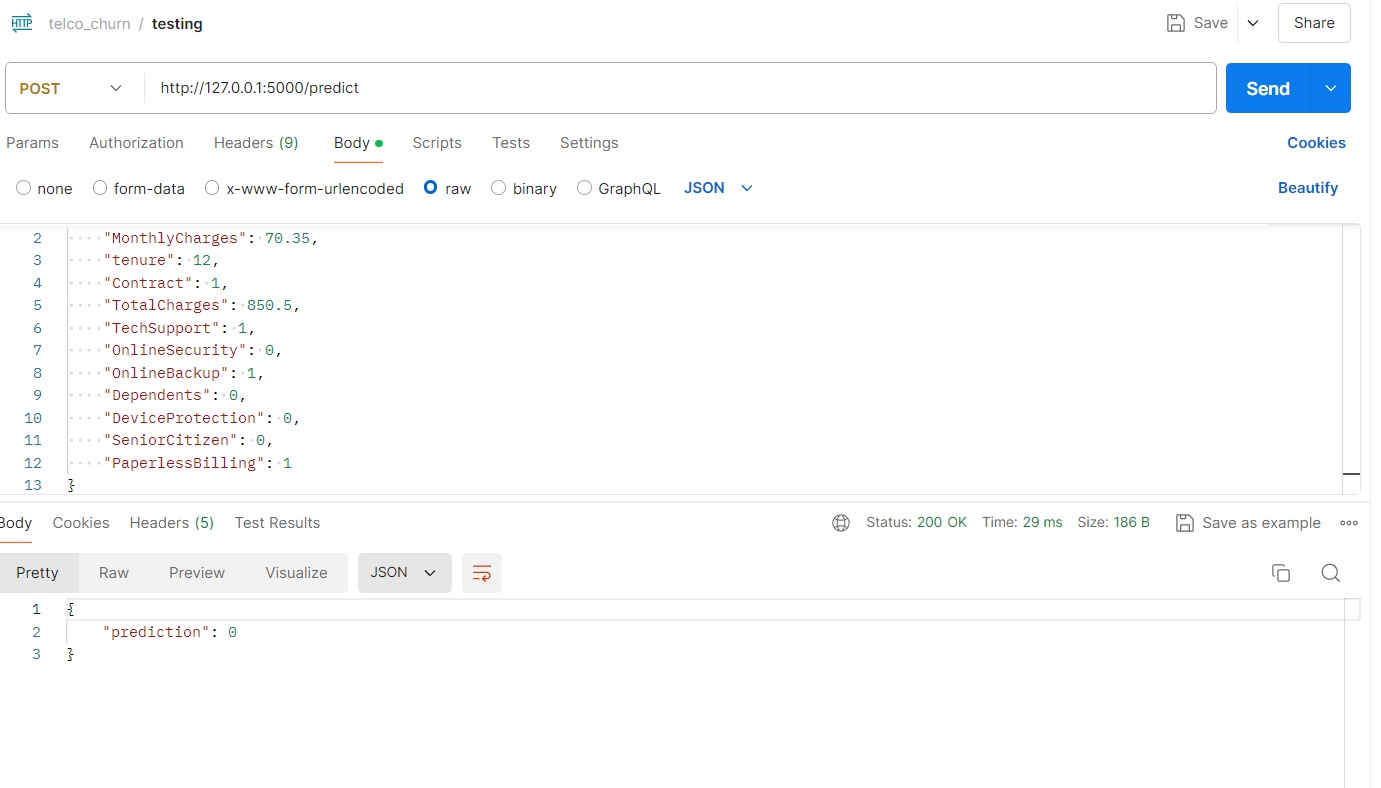
**Monthly Charges for Churned Customers:**

****

**MODULE 3: Model Deployment and API consumption**

****

**Testing and Validation:**

****

**Conclusion**

In this project, we successfully developed a predictive model for customer churn using various machine learning algorithms. The use of SMOTEENN improved the model's performance by addressing class imbalance. The final model can help the telecommunications company identify potential churners and take proactive measures to retain them, thus reducing revenue loss.