Telco Customer Churn Analysis

10Pearls Data Science Internship

Ali Muhammad Asad

Table of Contents

Project Overview

- The objective of this project is to analyze customer churn data from a telecommunications company, develop a predictive model capable of identifying customers who are likely to churn and manage the data via SQL.
- Churn refers to when a customer stops using the services of the telco, which can be due to various reasons such as poor service quality, high rates that are unable to meet the standard, etc.

Customer Churn Dataset

- The data set was provided to us on our notion page as a google sheets, which was downloaded into a csv file
- It included raw information about:
  - Customers who left – Churn (target variable)
  - Services each customer used (phone, internet, etc)
  - Customer Billing Information
  - Customer Demographic Information

Module 1: Python – Data Processing and EDA

Methodology

1. Data Processing
2. Exploratory Data Analysis (EDA)
3. Feature Engineering
4. Correlation Matrix
5. SMOTE

Data Processing

- Main libraries used:
  - Pandas, Seaborn, Matplotlib, imblearn
- The data was first loaded and inspected
- Very clean dataset
- No null values or any missing values

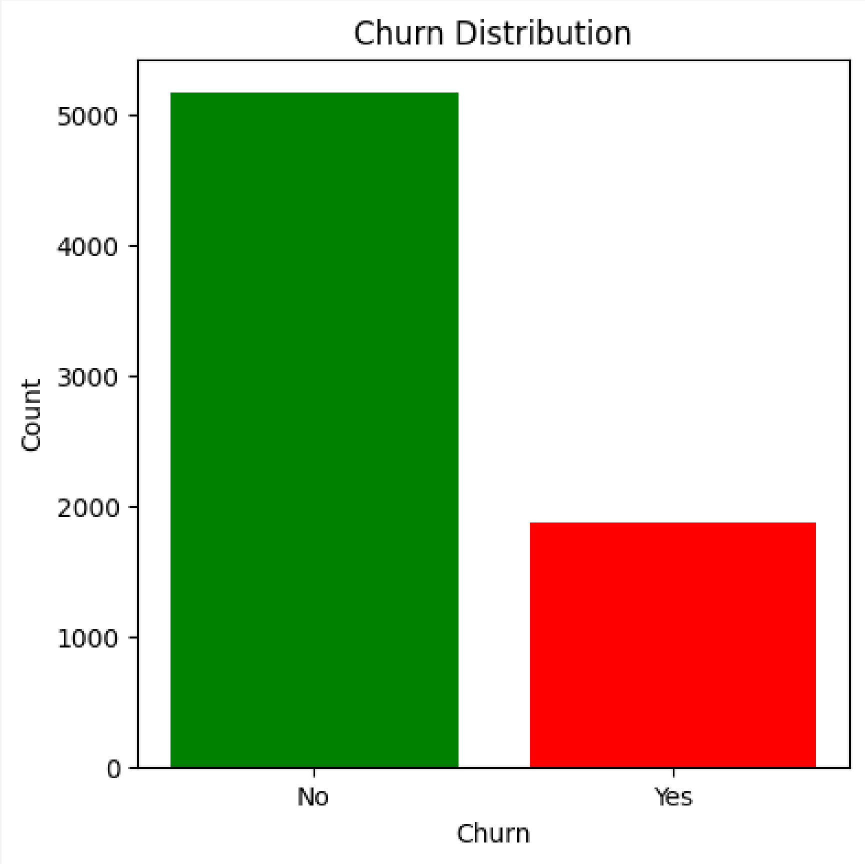
```python
missing_values = df.isnull().sum()
missing_values[missing_values > 0]
df = df.dropna()
df.isnull().sum()
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
```

Exploratory Data Analysis

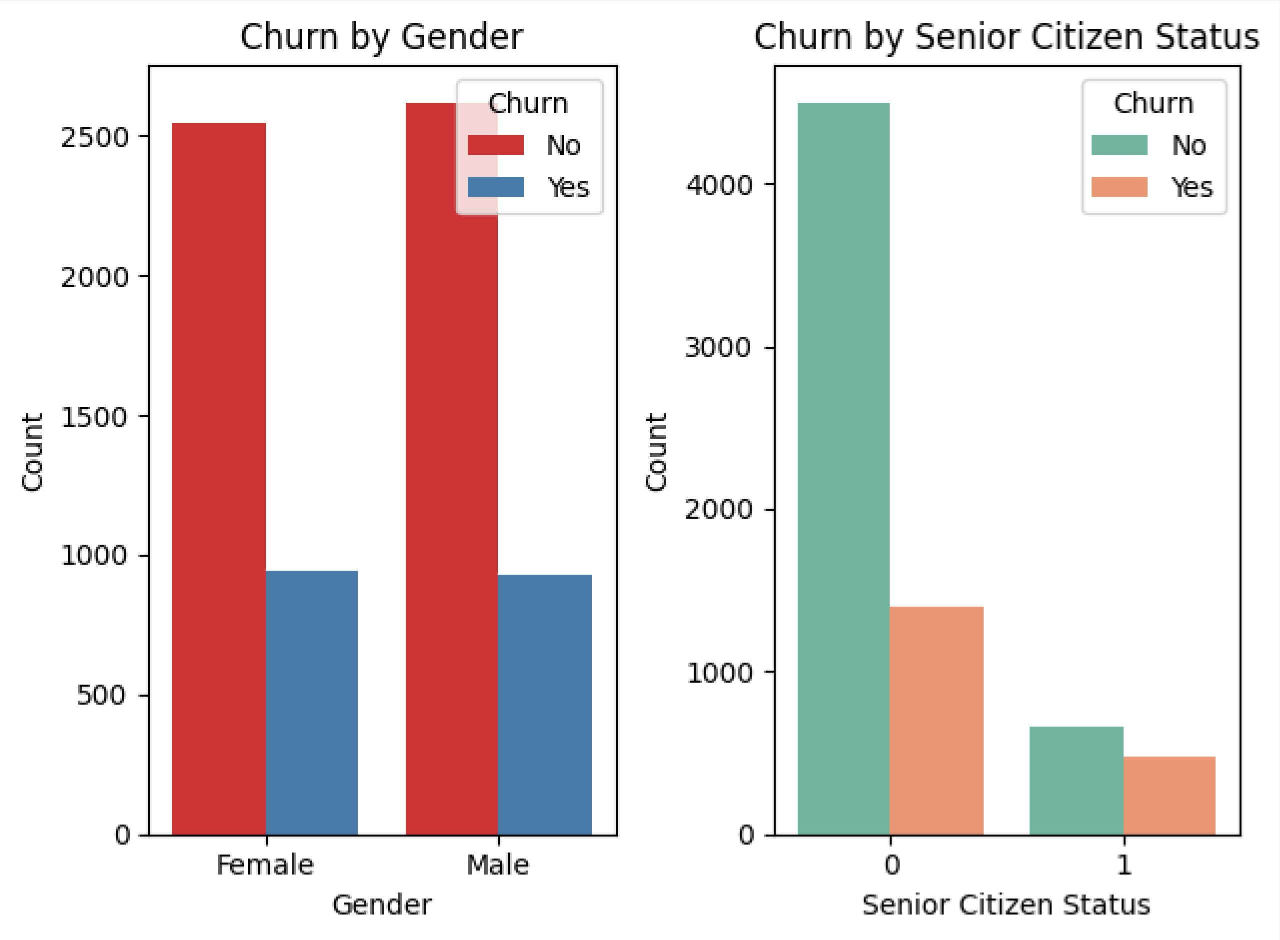
Target Feature Distribution

- Imbalanced distribution of target variable:
  - Churns: roughly 25%
  - Stays: roughly 75%
- Imbalanced distribution handled through SMOTE



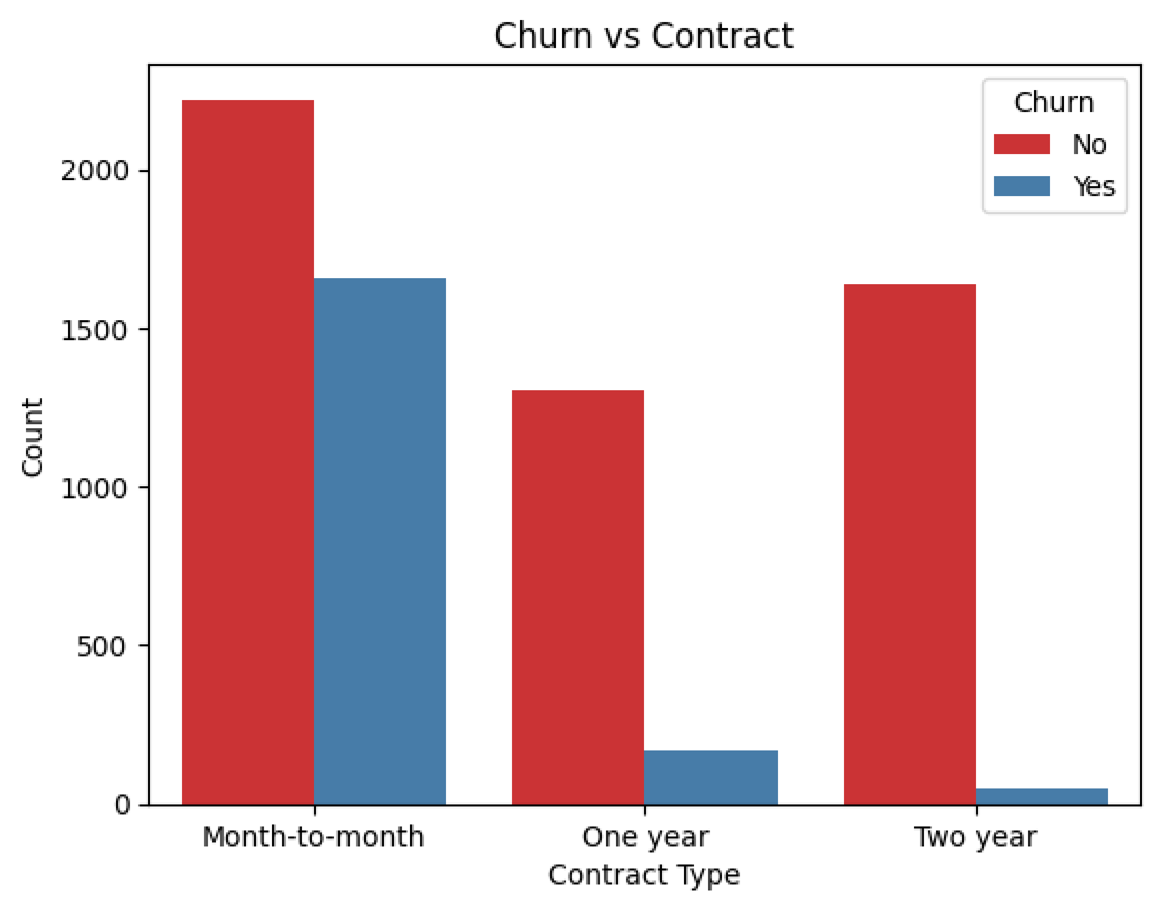
Distribution of Customer Demographics

- Males and Females have roughly the same churn rates
- More non-senior citizens
- Churn rates higher amongst non-senior citizens

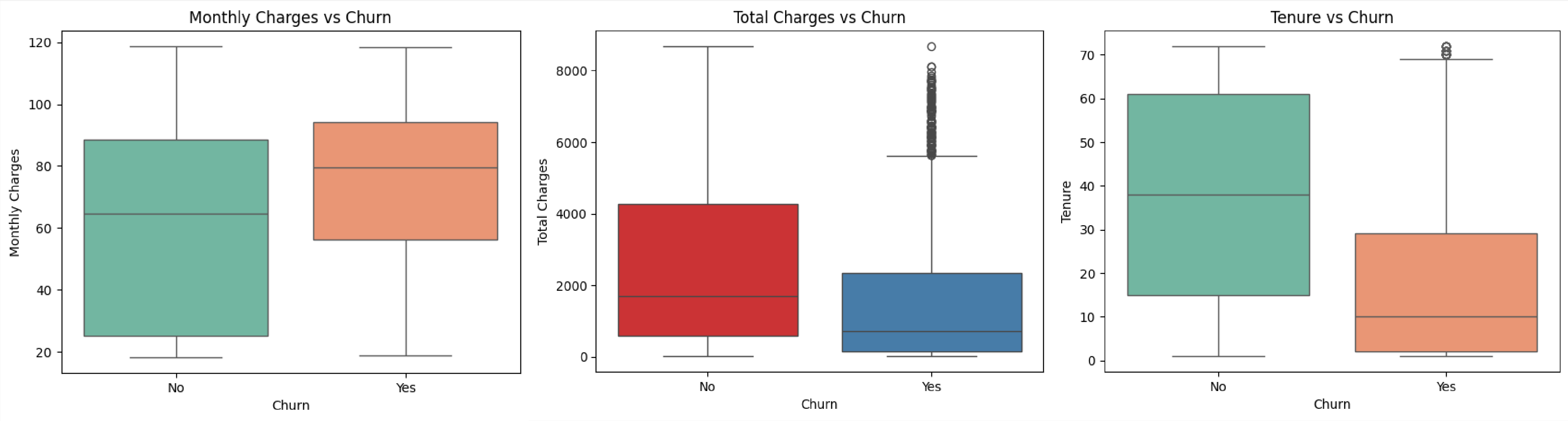


Contract Types and Churn

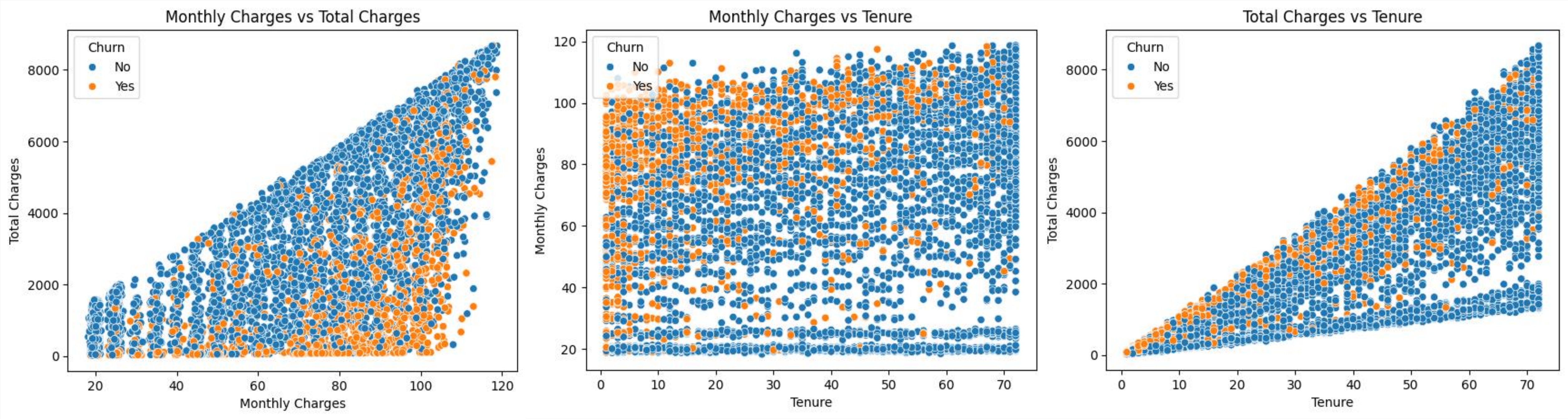
- Month-to-Month contract types show the highest amount of churn rates
- One and Two year contract types have the lowest churn rates
- Customers with longer contracts tend to stay loyal



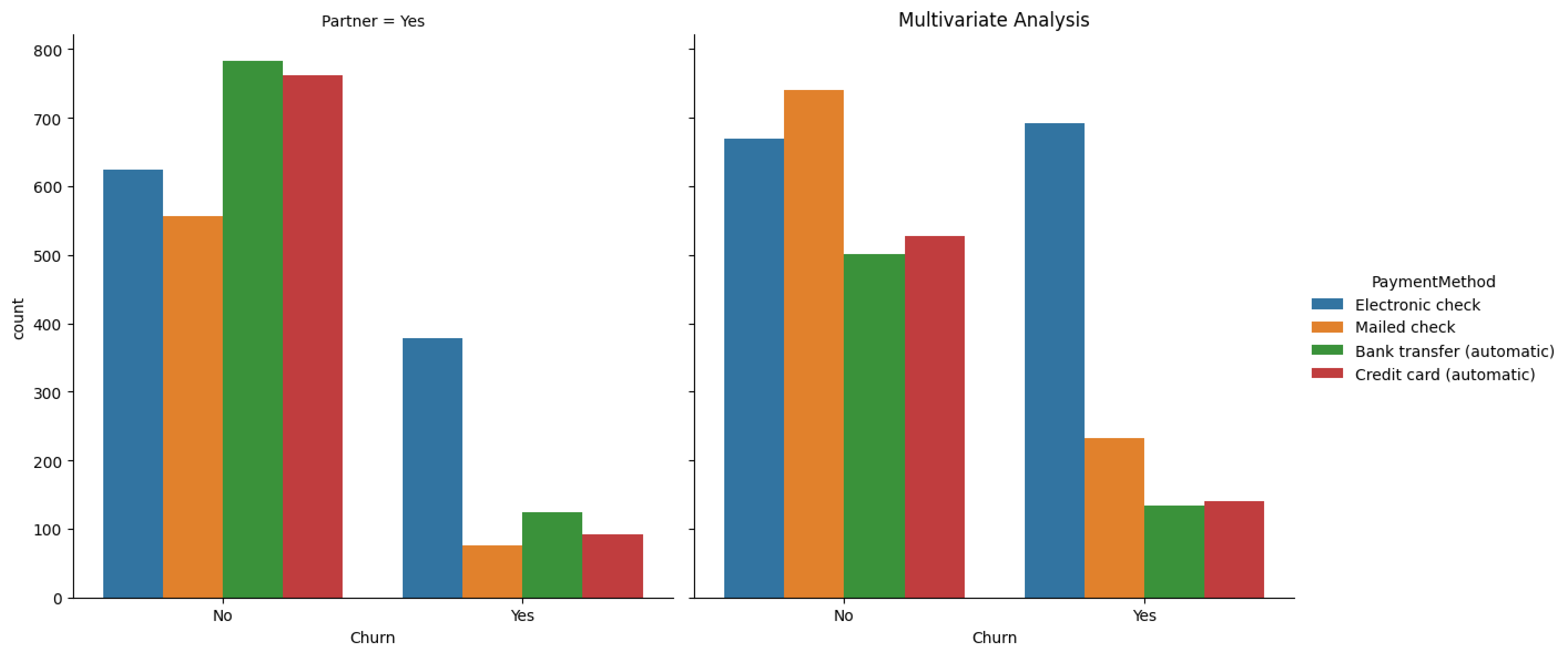
Churn vs Tenure, Monthly Charges, and Total Charges



Tenure, Monthly Charges, Total Charges

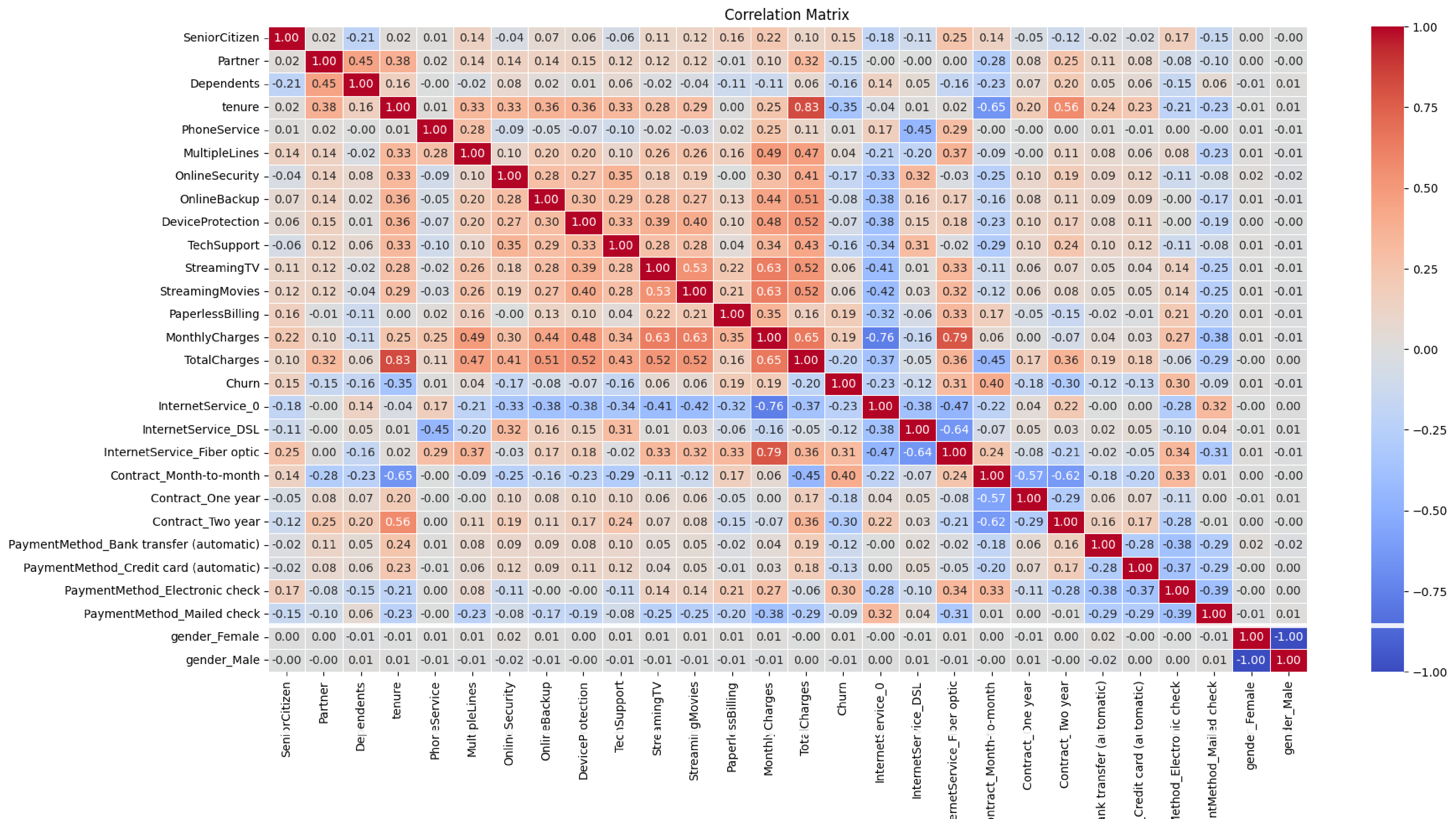


Multivariate Analysis: Partners and Payment Methods



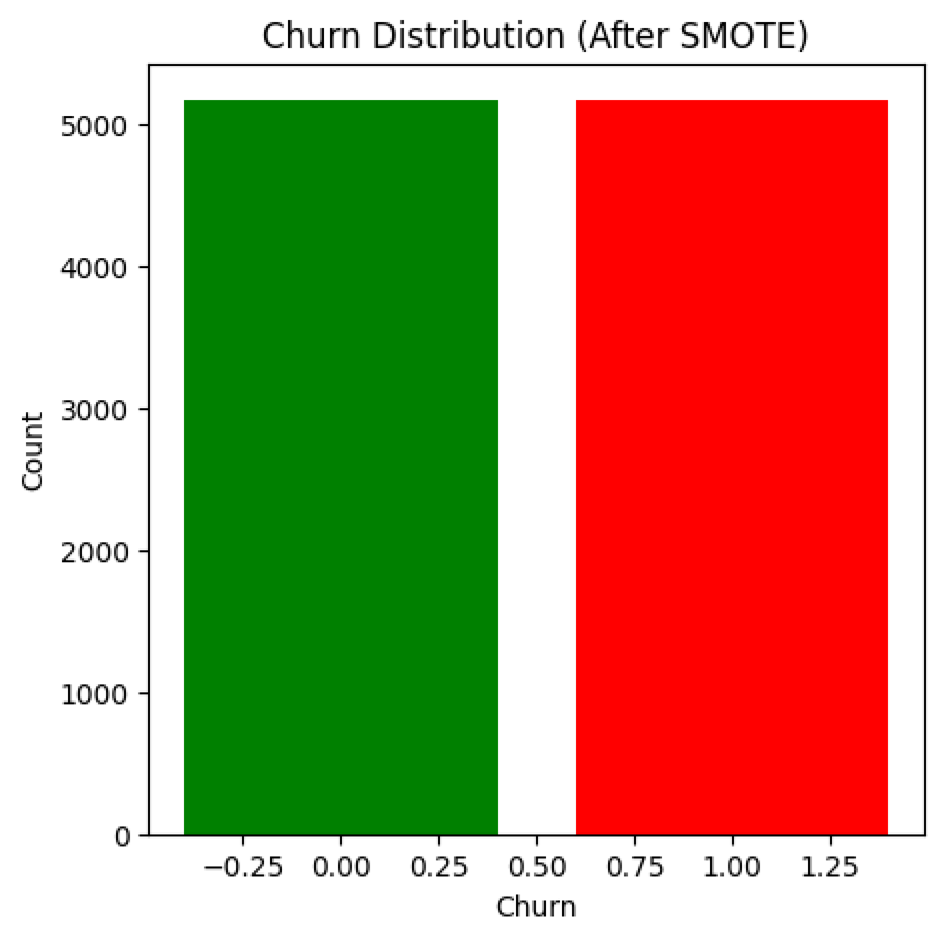
Feature Engineering

- Categorical data converted to numerical data
- Yes and No replaced with 1s and 0s
- Columns with categorical values split to introduce more features in numerical format
  - Payment Methods split
  - Gender split
- Numerical data easier to pass to and train machine learning model on


Correlation Matrix
SeniorCitizen
Partner
Dependents
tenure
PhoneService
MultipleLines
OnlineSecurity
OnlineBackup
DeviceProtection
TechSupport
StreamingTV
StreamingMovies
PaperlessBilling
MonthlyCharges
TotalCharges
Churn
InternetService_0
InternetService_DSL
InternetService_Fiber optic
Contract_Month-to-month
Contract_One year
Contract_Two year
PaymentMethod_Bank transfer (automatic)
PaymentMethod_Credit card (automatic)
PaymentMethod_Electronic check
PaymentMethod_Mailed check
gender_Female
gender_Male
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

SMOTE

- SMOTE used to balance the uneven distribution of the target variable – Churn
- Balances the dataset by introducing more synthetic data from the minority class, thus prevents the machine learning model to overfit on the majority class
- Improved performance



Module 2: AI Algorithms and Machine Learning

Used Machine Learning Models

- Logistic Regression
- Decision Trees
- Random Forest
- Gradient Boosting
- Support Vector Machine
- XGBoost
- Voting Classifier

Performance Metrics:

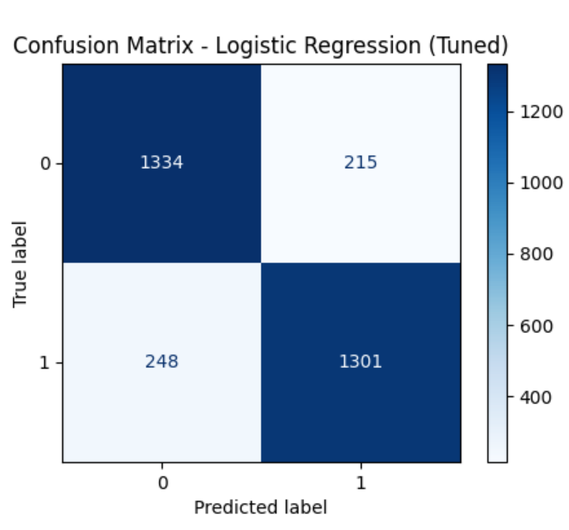
- Accuracy
- Precision
- Recall
- F1-Score
- ROC AUC

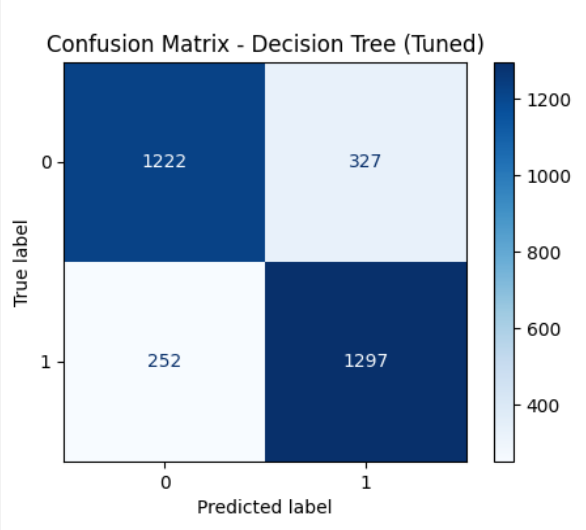
Classification Reports,
Confusion Matrix,
Precision Recll Curves,
ROC AUC Curves

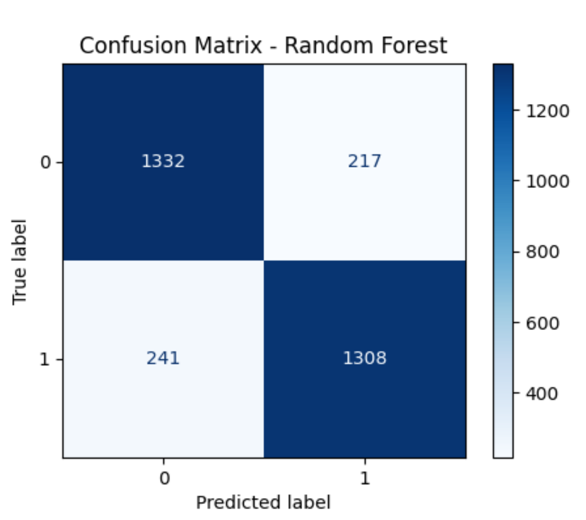
Model Training Steps

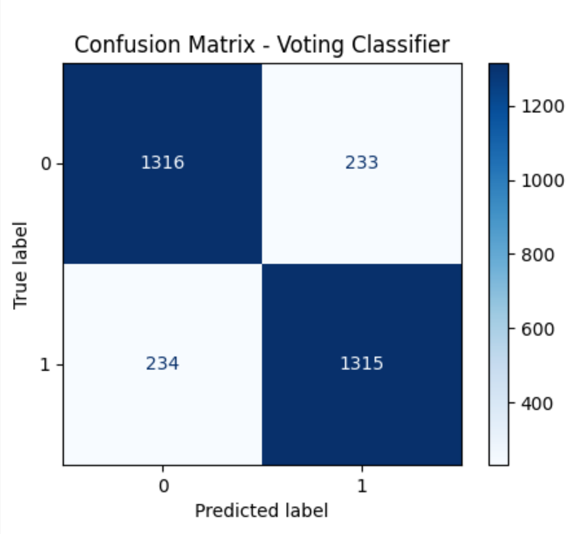
- Data split into train and test set (70-30)
- Model trained on the dataset
- Performance evaluated
- Hyperparameters tuned:
  - Parameters grid
  - GridSearchCV
- Best classifier saved
- Performance compared
- Best model saved as a pickle file

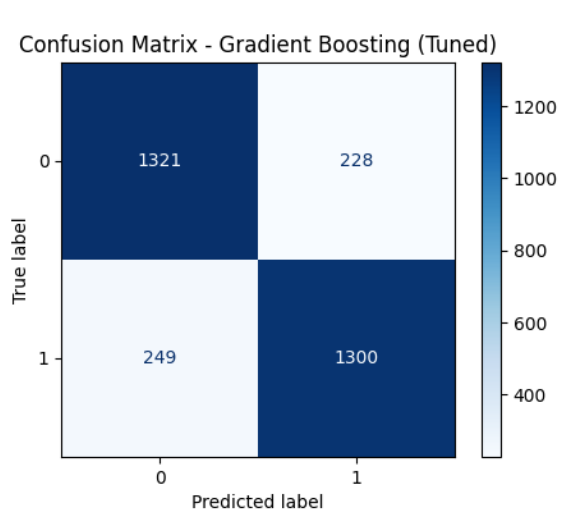
Summary – Confusion Matrices

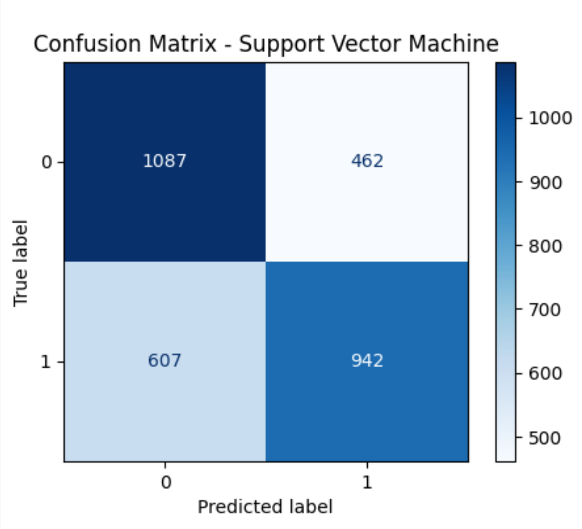


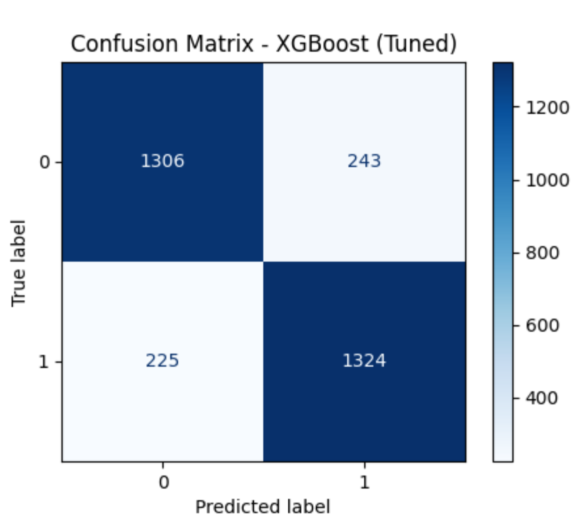












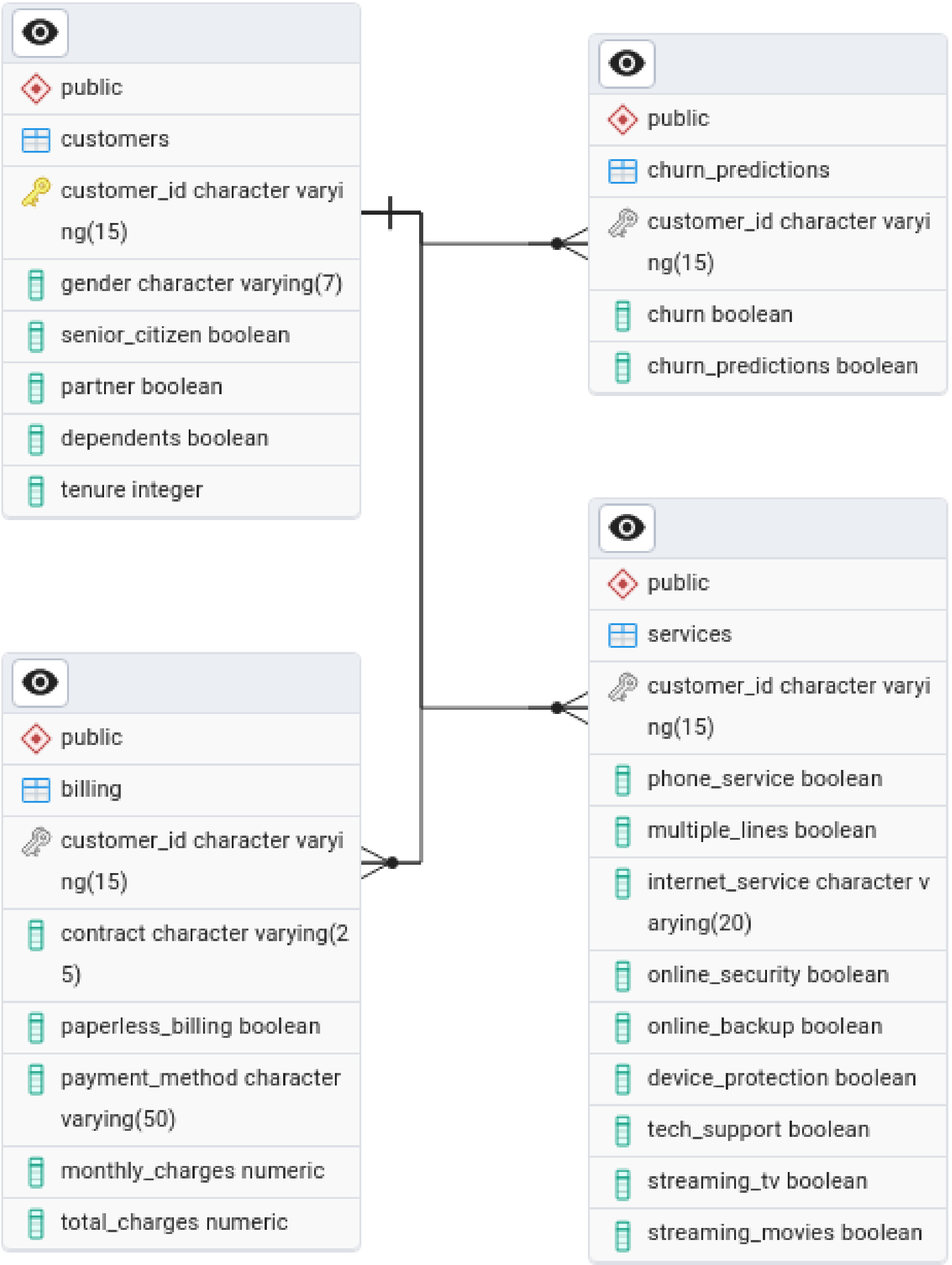
Model Rankings – Based on Metrics

- Metrics evaluated on:
  - Accuracy, Precision, Recall, F1-Score, ROC AUC
- Best Models (rankings based on the highest value given by the mean of the above metrics):
  - Voting Classifier
  - Random Forest
- Important Features using SHAP:
  - Contracts (One and Two years)
  - Monthly Charges
  - Tenure

Module 3: SQL Analysis

Database, Schema and Design

- PostgreSQL and pgAdmin4 used
- Schema generated over pgAdmin4
- Connected to python
- Raw data and prediction results stored in the database
- Queries executed for analysis



Analysis on Contracts, Payment Methods and Internet Services

- Month-to-Month contracts recorded the highest predicted churn rate of 43.40% (1682 out of 3875 customers)
- Two-year contracts recorded the lowest predicted churn rates of 1.90% (32 out of 1685 customers)

- Electronic Check had the highest predicted churn rate of 46.64% (1103 out of 2365 customers)
- Credit Cards have the lowest – 14.33%

- Fiber Optic internet service had the highest predicted churn rates of 42.24% (1308 out of 3096 customers)
- Customers with No Service are least likely to churn – 7.1%

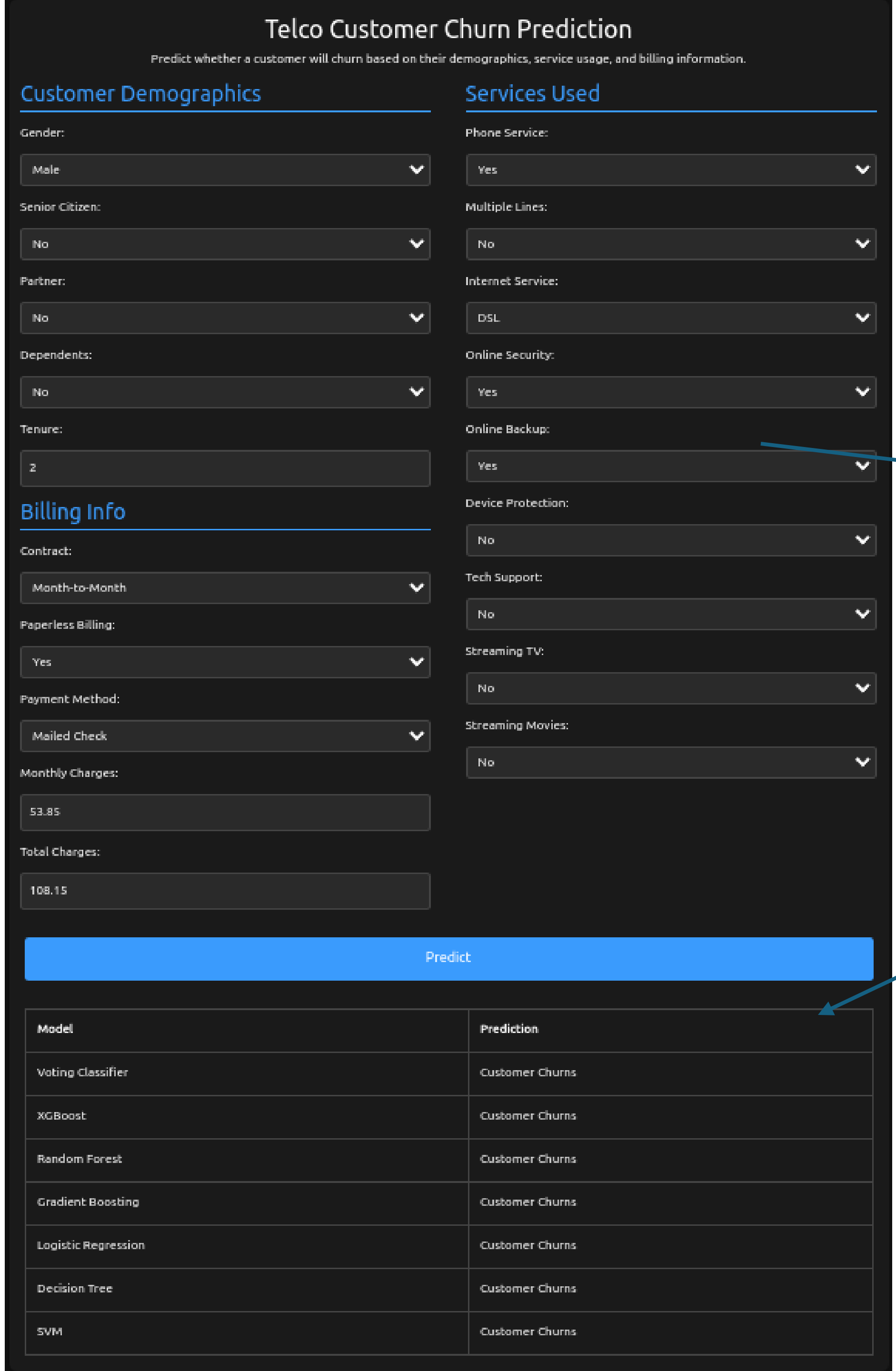
More Analysis

- The total revenue was $16056168.70
- Total revenue generated by churned customers: $2862926.90
- The customers with the lowest tenures (1-5 months) had the highest predicted churns

Module 4: Model Deployment

Flask Web App

- Flask used to deploy the models over a web app
- app.py -> Flask API
    - Loads saved models
    - Receives customer input
    - Computes and returns predictions
- APIs developed for the following functionalities:
    - Health Check: status of API
    - Prediction: receive data and return prediction
- APIs tested via Curl



Customer info

```python
@app.route('/api/predict', methods=['POST'])
def api_predict():
    data = request.json
    data = form_to_numeric(data)

    # Make precictions
    predl = {model_name: show_pred(model.predict(data)[0]) for model_na

    return jsonify(predl)

@app.route('/api/health', methods=['GET'])
def health_check():
    return jsonify({'status': 'Thumbs Up guys'}), 200
```

Predictions

Flask Web App - Demo

Thank you