Customer Churn Analysis Report

Introduction

Customer churn analysis is crucial for businesses to understand why customers leave or discontinue their services. By identifying key factors that contribute to customer attrition, businesses can take proactive measures to reduce churn, improve customer retention, and ultimately enhance their overall business performance. This project focuses on analyzing customer churn data, exploring various features, and creating a predictive model to understand customer behavior.

Project Overview

This project involves performing an in-depth analysis of a customer churn dataset, which contains information about customers' demographics, service usage, and whether they have churned or not. The main objective is to identify patterns that might contribute to customer churn and build a machine learning model to predict customer churn based on historical data.

Tools Used

The following tools and libraries were used in this analysis:

- **Python**: Primary programming language for data analysis.
- **Pandas**: Used for data manipulation and preprocessing.
- **NumPy**: Used for numerical operations.
- **Matplotlib and Seaborn**: Used for data visualization.
- **Scikit-learn**: Used for data preprocessing, model training, and evaluation.
- **Joblib**: Used for saving the trained machine learning model.

Data Exploration and Preprocessing

1. Data Loading and Initial Exploration

The dataset, "Customer-Churn-Records.csv," was loaded into a Pandas DataFrame for initial exploration. The first few rows of the dataset were displayed to understand its structure and content. The dataset included a mix of numerical and categorical features.

python

```
# Load dataset
data_path = "Customer-Churn-Records.csv"
df = pd.read_csv(data_path)

# Display dataset overview
df.head()
df.info()
df.describe()
```

2. Churn Column Identification

The target column, which indicates whether a customer has churned, was identified. The dataset contained two potential churn indicators: 'Churn' and 'Exited'. After a check, the churn column was successfully identified.

3. Data Cleaning and Handling Missing Values

Missing values were checked for in the dataset. Rows with missing values were dropped to ensure the dataset is clean and complete for analysis.

python

C

```
df.dropna(inplace=True)
```

4. Feature Encoding

Categorical variables, such as 'Contract', 'PaymentMethod', etc., were encoded into numerical values using **LabelEncoder**. This step ensured that the machine learning model could work with numeric data.

python

```
label_encoders = {}
for column in categorical_cols:
    if column != 'customerID':  # Skip ID columns
        label_encoders[column] = LabelEncoder()
        df[column] = label_encoders[column].fit_transform(df[column])
```

Exploratory Data Analysis (EDA)

1. Churn Distribution

A count plot was created to visualize the distribution of churn (i.e., how many customers have churned vs. retained). This analysis helps to understand the balance between churned and retained customers.

python

```
sns.countplot(x='Churn', data=df, palette='coolwarm')
```

2. Feature Correlation

A correlation heatmap was generated to analyze the relationships between numerical features. This visualization helps to identify any strong correlations between features that may be useful for building the predictive model.

python

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
```

3. Churn Analysis by Tenure

The churn status was analyzed in relation to the "tenure" feature (if available). A histogram was plotted to see if customers with lower tenure were more likely to churn, which is a common observation in subscription-based businesses.

python

```
sns.histplot(df[df['Churn'] == 1]['tenure'], bins=20, kde=True, color='red')

sns.histplot(df[df['Churn'] == 0]['tenure'], bins=20, kde=True, color='blue')
```

4. Churn by Contract Type

An analysis of churn rates by contract type (if the 'Contract' column exists) was performed. The count plot provides insights into whether certain contract types are associated with higher churn.

python

```
sns.countplot(x='Contract', hue='Churn', data=df, palette='Set2')
```

5. Churn by Monthly Charges

A Kernel Density Estimate (KDE) plot was created to analyze the distribution of monthly charges for churned and retained customers. This plot helps to identify if higher monthly charges correlate with higher churn rates.

python

```
sns.kdeplot(df[df['Churn'] == 0]['MonthlyCharges'], label='Retained', fill=True)

sns.kdeplot(df[df['Churn'] == 1]['MonthlyCharges'], label='Churned', fill=True)
```

Feature Importance Analysis

To understand which features are the most important for predicting customer churn, a Random Forest Classifier was trained on the dataset. The importance of each feature was visualized using a bar plot.

python

```
# Train Random Forest model for feature importance

model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)


# Plot feature importance

sns.barplot(x='Importance', y='Feature', data=feature_importance)
```

Model Training and Evaluation

1. Data Splitting

The dataset was split into training and testing sets using a 80-20 ratio. This split is crucial for evaluating the model's performance on unseen data.

python

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2. Feature Scaling

To improve the performance of the machine learning model, the features were scaled using **StandardScaler**. This step ensures that all features contribute equally to the model's predictions.

python

```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

3. Model Training

A **Random Forest Classifier** was used to train the model on the training data. The model was then used to predict the churn status of customers in the testing set.

python

```
model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train_scaled, y_train)


# Predictions

y_pred = model.predict(X_test_scaled)
```

4. Model Evaluation

The model's performance was evaluated using accuracy, classification report, and a confusion matrix. These metrics help to assess the model's ability to correctly predict churn and retention.

python

```
accuracy = accuracy_score(y_test, y_pred)

classification_report(y_test, y_pred)
```

Model Saving

Finally, the trained model, feature scaler, and label encoders were saved using **joblib** to enable deployment and future use.

python

```
import joblib

joblib.dump(model, 'churn_prediction_model.pkl')

joblib.dump(scaler, 'churn_prediction_scaler.pkl')

joblib.dump(label_encoders, 'churn_prediction_encoders.pkl')
```

Conclusion

This analysis provided a detailed understanding of customer churn by identifying important features, visualizing patterns, and building a machine learning model to predict customer churn. The model can be used to predict future churn and help businesses take proactive measures to retain valuable customers. The insights gained from this analysis, such as the impact of contract type, monthly charges, and tenure on churn, can be used to enhance customer retention strategies and improve overall customer satisfaction.