Capstone Project: Telecom Churn

BY ~ ROHIT .B. PATEL

Introduction to Capstone Project: Telecom Churn

- LOAD THE DATA
- INSPECTING A DATA SET
- DATA CLEANING
- OUTLIER
- EXPLORATORY DATA ANALYSIS (EDA)
- MODEL BUILDING IN DATA ANALYSIS
- MODEL EVALUATION & RESIDUAL ANALYIS

Loading Data

TaKeykewaysShow

- Loading the dataset is the first step in data analysis.
- Use .shape to check dimensions and.head() to preview the data.
- Readine Data inspection ensures for further preprocessing.

Output Display

- The dataset's shape, such as (99999,226), and the first few rows (as seen in the df. head () output).

In [22]:
```
df = pd.read_csv("telecom_churn_data.csv")
```

In [23]:
```
df.head()
```

Out[23]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_date_of
0	7000842753	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	
1	7001865778	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	
2	7001625959	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	
3	7001204172	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	
4	7000142493	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	

In [24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99999 entries, 0 to 99998
Columns: 226 entries, mobile_number to sep_vbc_3g
dtypes: float64(179), int64(35), object(12)
memory usage: 172.4+ MB
```

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
mobile_number        0
circle_id            0
loc_og_t2o_mou    1018
std_og_t2o_mou    1018
loc_ic_t2o_mou    1018
                  ...
aon                  0
aug_vbc_3g           0
jul_vbc_3g           0
jun_vbc_3g           0
sep_vbc_3g           0
Length: 226, dtype: int64
```

In [26]:
```
df.shape
```

Out[26]:
```
(99999, 226)
```

In [27]:
```
df.head(2)
```

Out[27]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_date_of
0	7000842753	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	
1	7001865778	109	0.00	0.00	0.00	6/30/2014	7/31/2014	8/31/2014	

Clean Data

Key Steps in Data Cleaning

Handle Missing Values

- Use .isnull().sum() or 100 * df.isnull().mean() to identify missing data.
- Impute missing values (e.g., median for numerical columns or mode for categorical).

Remove Duplicates

- Use df.duplicated() and df.drop_duplicates().

Standardize Data Types

- Convert columns to appropriate data types (e.g., numerical, categorical, datetime).

Correct Outliers

- Use visualizations like boxplots or statistical methods to detect and handle outliers.

Normalize/Scale Data

- Apply normalization or scaling for machine learning algorithms.

In [30]:
```
import pandas as pd
```

In [31]:
```
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values[missing_values > 0])
```

```
Missing Values:
 loc_og_t2o_mou          1018
std_og_t2o_mou          1018
loc_ic_t2o_mou          1018
last_date_of_month_7     601
last_date_of_month_8    1100
                        ...
night_pck_user_9       74077
fb_user_6              74846
fb_user_7              74428
fb_user_8              73660
fb_user_9              74077
Length: 166, dtype: int64
```

In [32]:
```
threshold = 0.5 * len(df)
df = df.dropna(thresh=threshold, axis=1)
```

In [33]:
```
for column in df.columns:
    if df[column].dtype == 'object':
        df[column].fillna(df[column].mode()[0], inplace=True)
    else:
        df[column].fillna(df[column].mean(), inplace=True)
```

In [34]:
```
df = df.drop_duplicates()
```

In [35]:
```
df = pd.get_dummies(df, drop_first=True)
```

In [36]:
```
numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
scaler = StandardScaler()
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])
```

In [37]:
```
 df.isnull().sum().sum()
```

Out[37]:
```
np.int64(0)
```

In [38]:
```
print("Shape of cleaned dataset:", df.shape)
```

```
Shape of cleaned dataset: (99999, 296)
```

In [39]:
```
df.head
```

Out[39]:
```
<bound method NDFrame.head of        mobile_number  circle_id  loc_og_t2o_mou  std_og_t2o_mou  \
0              -0.52       0.00            0.00            0.00
```

Inspecting the Dataframe

Key Insights

- Understanding structure:
  Use .head(), .info(),
  and .shape for a quick overview.

- Statistical summary:
  Use .describe() for numeric columns.

- Quality checks:
  Identify missing and duplicate data with .
  isnull() and .duplicated().

```
In [41]: df.shape
Out[41]: (99999, 296)
```

```
In [42]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99999 entries, 0 to 99998
Columns: 296 entries, mobile_number to date_of_last_rech_9_9/9/2014
dtypes: bool(118), float64(178)
memory usage: 147.1 MB
```

```
In [43]: df.describe()
```

Out[43]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	arpu_6	arpu_7	arpu_8	arpu_9	onnet_mou_6	onnet_mou_7	on
count	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	99999.00	
mean	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	0.00	0.00	
std	1.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	
min	-1.74	0.00	0.00	0.00	0.00	-7.74	-6.78	-3.56	-6.32	-0.45	-0.44	
25%	-0.86	0.00	0.00	0.00	0.00	-0.58	-0.57	-0.57	-0.58	-0.43	-0.42	
50%	-0.00	0.00	0.00	0.00	0.00	-0.26	-0.26	-0.25	-0.25	-0.32	-0.32	
75%	0.87	0.00	0.00	0.00	0.00	0.27	0.26	0.26	0.27	-0.00	0.00	
max	1.73	0.00	0.00	0.00	0.00	83.57	103.11	96.57	112.70	24.87	26.50	

```
In [44]: df_missing_columns = (round(((df.isnull().sum()/len(df.index))*100),2).to_frame('null')).sort_values('null', ascending=False)
df_missing_columns
```

Out[44]:

	null
date_of_last_rech_9_9/9/2014	0.00
mobile_number	0.00
circle_id	0.00
loc_og_t2o_mou	0.00
std_og_t2o_mou	0.00
...	...
onnet_mou_9	0.00
onnet_mou_8	0.00
onnet_mou_7	0.00
onnet_mou_6	0.00
arpu_9	0.00

Visualizing Data Distribution Using Histograms

Purpose of Histograms

- Visual representation of the distribution of numerical columns in a dataset.
- Helps identify key patterns, such as normal distribution, skewness, or multimodality.

Code Overview

- df.hist(bins=20, figsize=(15, 10)):Automatically generates histograms for all numerical columns in the DataFrame.
- bins=20: Divides data into 20 intervals for detailed distribution analysis.
- figsize=(15, 10): Ensures clear and large visual output.
- plt.show(): Displays the histograms.

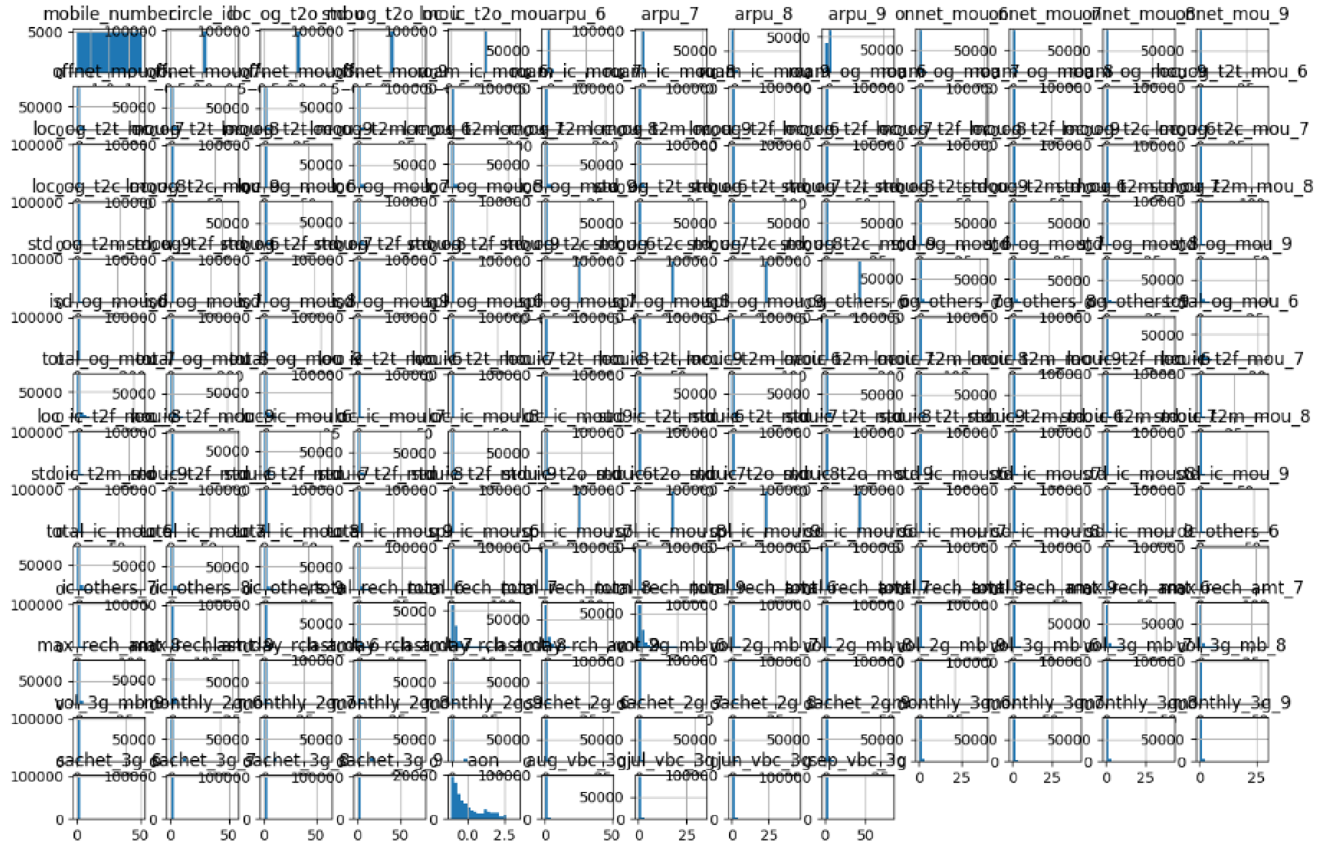
Insights Gained from Histograms

- Shape of Distribution:Normal, skewed, or bimodal distributions.
- Range of Values:The spread of data across intervals.
- Frequency of Values:How often data points fall into each interval.
- Outliers or Gaps:Easily spot unusual data points or missing ranges.

When to Use

- During EDA to understand the structure of numerical data.

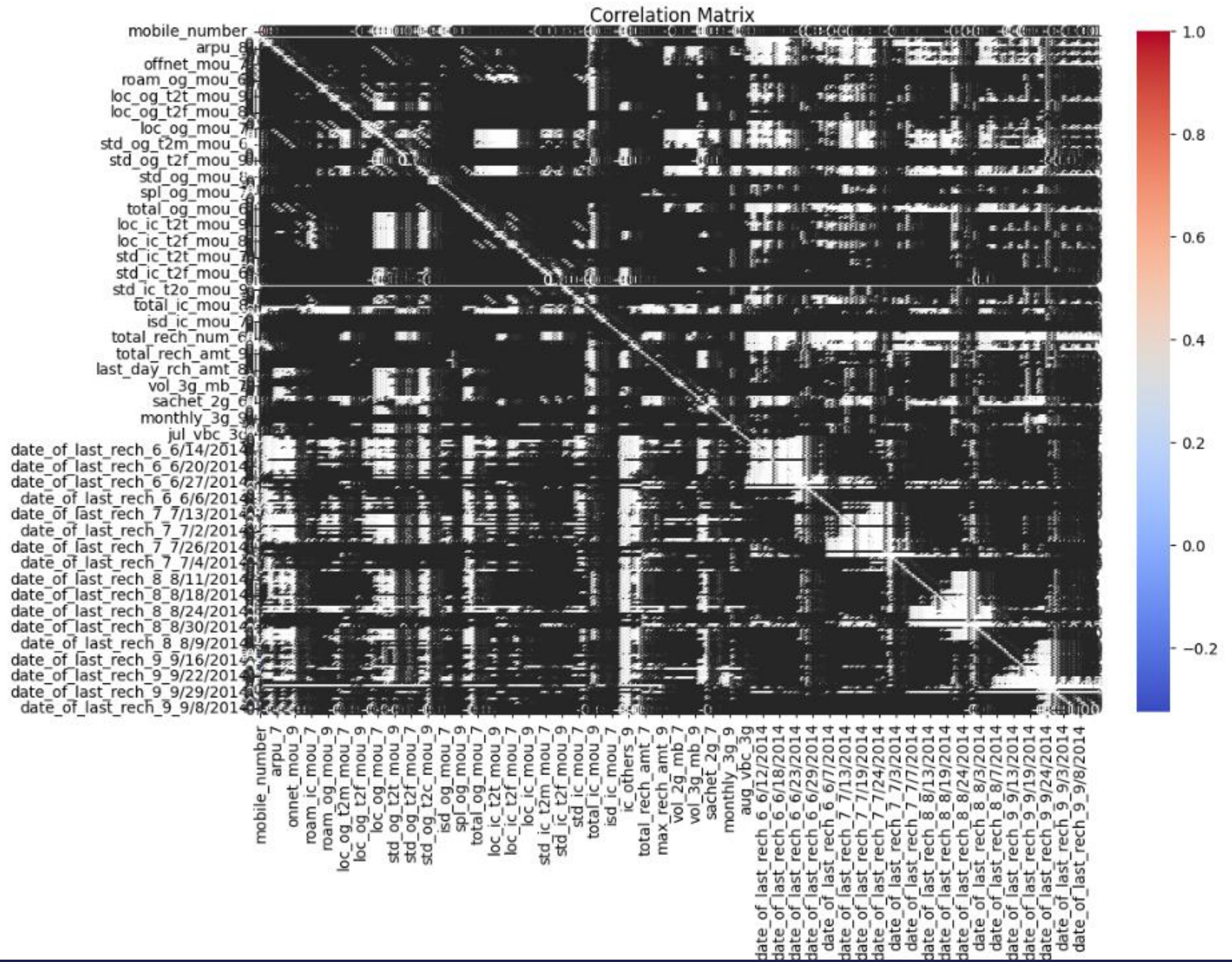
```
In [46]: df = df.drop(col_list_missing_30, axis=1)

In [47]: df.shape

Out[47]: (99999, 296)

In [48]: df.hist(bins=20, figsize=(15, 10))
         plt.show()
```



Inspecting the Dataframe

- **plt.figure(figsize=(12, 8))**
  Sets the figure size to
  ensure the plot is large and readable.
- **sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm' , fmt='.2f')**
  correlation_matrix: The correlation matrix (usually obtained via
  df.corr()).
  annot=True: Displays the correlation values on the heatmap.
- **cmap='coolwarm'**: Sets the color map to coolwarm for
  visual contrast.
- **fmt='.2f':** Formats the correlation values to two decimal places.
- **plt.title("Correlation Matrix")**
  Adds a title to the heatmap
  for better understanding.
- **plt.show()**
  Renders the plot.

```
In [50]: plt.figure(figsize=(12, 8))
         sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
         plt.title("Correlation Matrix")
         plt.show()
```



Data Type Conversion and Outlier Handling

Purpose of Data Type Conversion

- **Clarifies Column Roles**:

Converts mobile_number from numeric to categorical/object, as it represents an identifier, not a numerical value.

Converts churn from numeric to categorical/object, representing binary or categorical data .

- **Optimizes Memory Usage**:

Object data types are more efficient for non-numeric columns.

- **Improves Analytical Accuracy**:

Prevents incorrect operations on non-numeric columns statistical computations on mobile_number.

```
In [82]: df['mobile_number'] = df['mobile_number'].astype(object)
         df['churn'] = df['churn'].astype(object)

In [83]: df.info()

         <class 'pandas.core.frame.DataFrame'>
         Index: 30000 entries, 7 to 99997
         Columns: 136 entries, mobile_number to churn
         dtypes: float64(134), object(2)
         memory usage: 31.4+ MB

In [84]: numeric_cols = df.select_dtypes(exclude=['object']).columns
         print(numeric_cols)

         Index(['loc_og_t2o_mou', 'std_og_t2o_mou', 'loc_ic_t2o_mou', 'arpu_6',
                'arpu_7', 'arpu_8', 'onnet_mou_6', 'onnet_mou_7', 'onnet_mou_8',
                'offnet_mou_6',
                ...
                'monthly_3g_7', 'monthly_3g_8', 'sachet_3g_6', 'sachet_3g_7',
                'sachet_3g_8', 'aon', 'aug_vbc_3g', 'jul_vbc_3g', 'jun_vbc_3g',
                'avg_rech_amt_6_7'],
               dtype='object', length=134)

In [85]: for col in numeric_cols:
             q1 = df[col].quantile(0.10)
             q3 = df[col].quantile(0.90)
             iqr = q3-q1
             range_low  = q1-1.5*iqr
             range_high = q3+1.5*iqr
             # Assigning the filtered dataset into data
             data = df.loc[(df[col] > range_low) & (df[col] < range_high)]

         data.shape

Out[85]: (29693, 136)
```

Purpose of Analysis

- **Objective**

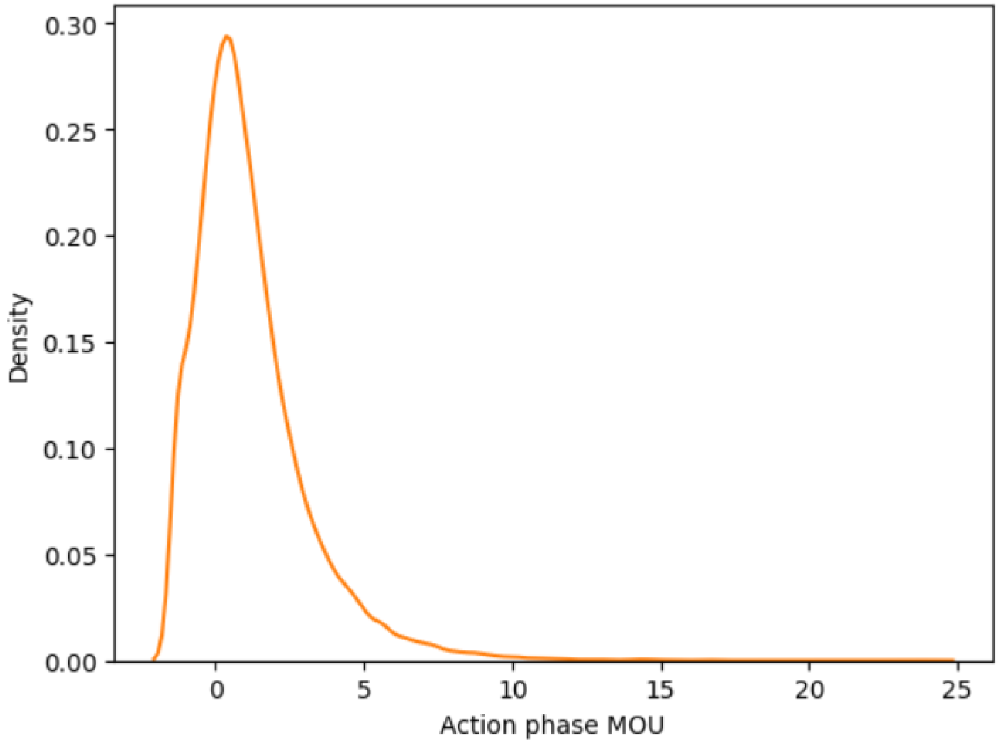
Compare the distribution of ARPU (Average Revenue Per User) during the action phase for churn and non-churn customers.

- **Insight Goal**

Understand revenue behavior differences between the two groups, potentially aiding in identifying churn triggers.

In [114]:
```
ax = sns.distplot(data_churn['total_mou_good'],label='churn',hist=False)
ax = sns.distplot(data_non_churn['total_mou_good'],label='non churn',hist=False)
ax.set(xlabel='Action phase MOU')
```

Out[114]:
```
[Text(0.5, 0, 'Action phase MOU')]
```



Bivariate analysis

EDA Purpose

To understand the data's structure, identify patterns, detect outliers, and spot any anomalies.

- Scatter Plot Overview

Variables:

avg_rech_num_action (X-axis): Average number of recharge actions.

avg_rech_amt_action (Y-axis): Average amount of recharge actions.

churn (Hue): Churn indicator (e.g., Yes/No).

Goal:

To analyze the relationship between recharge behaviors and churn.

- Scatter Plot Analysis

Clusters: Identify any visible patterns or clusters based on churn status.

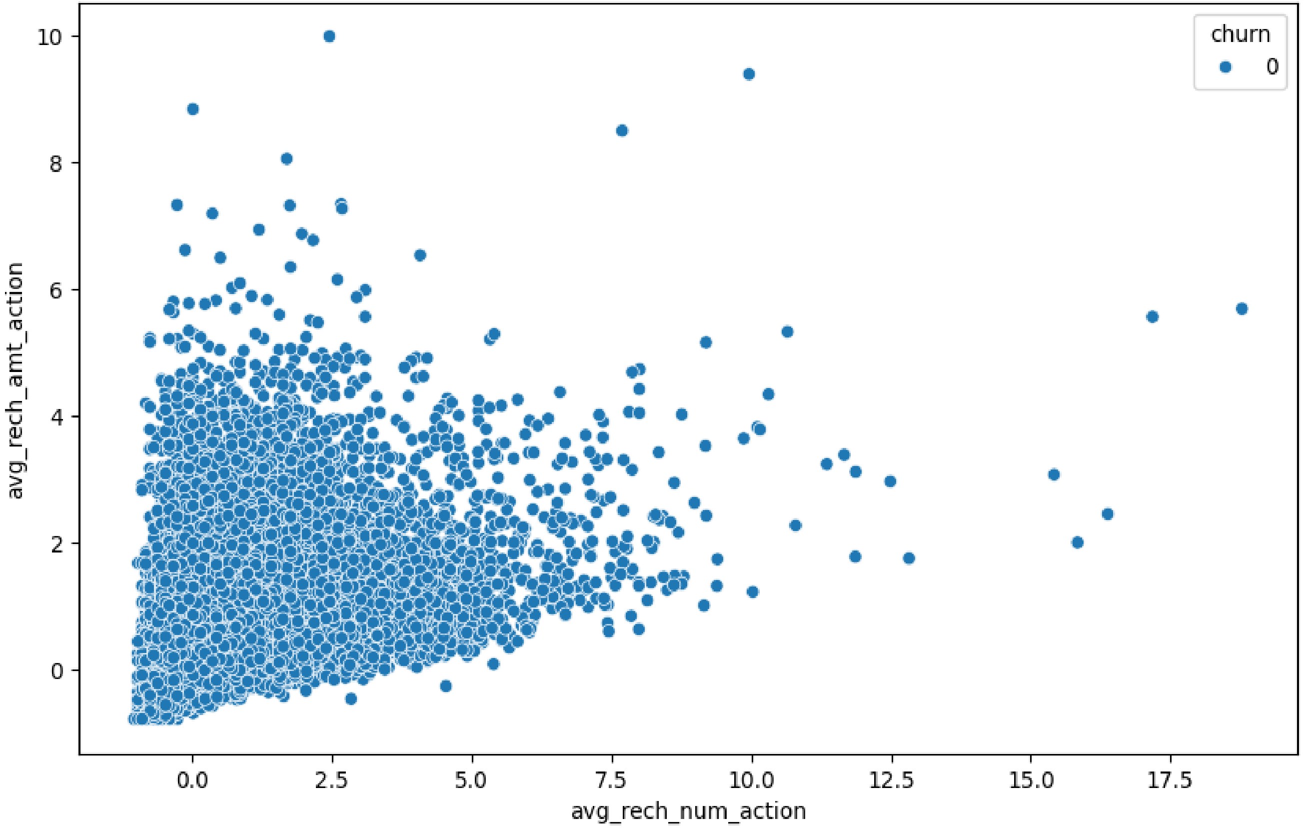
Churn and Recharge Behavior: Check if churners exhibit different recharge behaviors compared to non-churners (e.g., more or less frequent recharges, higher or lower amounts).

Outliers: Look for any outliers, such as customers who have very high recharge amounts or action counts.

- **Observation**

The scatter plot helps to visually differentiate churners from non-churners and may reveal trends or correlations.

```
In [117]: plt.figure(figsize=(10, 6))
          ax = sns.scatterplot(x='avg_rech_num_action', y='avg_rech_amt_action', hue='churn', data=data)
```



Model Evaluation

Purpose:

To assess how well the trained model generalizes to unseen data and to ensure its effectiveness in making predictions.

Steps in Model Evaluation:

- Train the model on the training set.
- Evaluate the model using the test set.
- Use metrics like accuracy, precision, recall, F1-score, and confusion matrix.

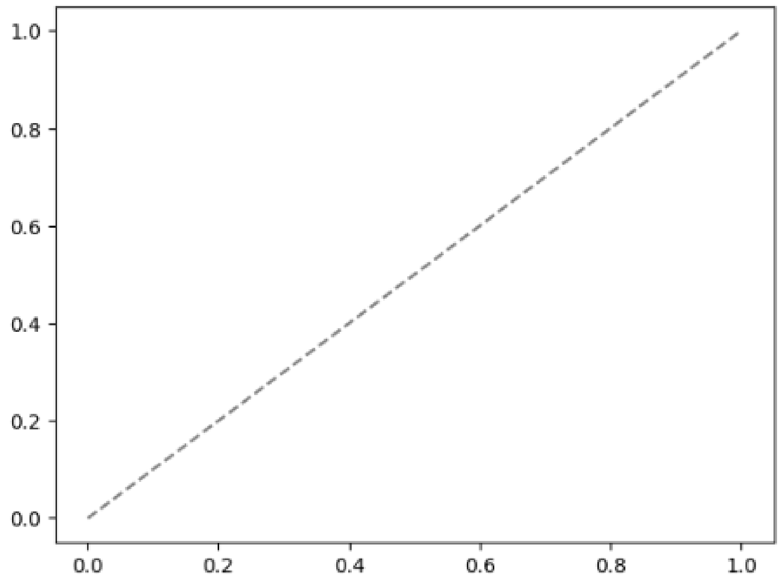
Model Evaluation

```
In [134]: from sklearn.ensemble import RandomForestClassifier

          model = RandomForestClassifier()

          model.fit(X_train, y_train)

Out[134]: RandomForestClassifier()
          In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
          On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [135]: (model.fit)

Out[135]: <bound method BaseForest.fit of RandomForestClassifier()>

In [136]: y_pred = model.predict(X_test)

In [137]: import statsmodels.api as sm
          import matplotlib.pyplot as plt

In [138]: y_actual = data['churn']

In [139]: X = data.drop(columns=['churn'])

In [140]: print(X.dtypes)

          mobile_number                object
          loc_og_t2o_mou              float64
          std_og_t2o_mou              float64
          loc_ic_t2o_mou              float64
          arpu_6                      float64
                                        ...
          decrease_mou_action           int64
          decrease_rech_num_action      int64
          decrease_rech_amt_action      int64
          decrease_arpu_action          int64
          decrease_vbc_action           int64
          Length: 139, dtype: object

In [141]: X = pd.get_dummies(X, drop_first=True)

In [142]: print(y_actual.shape, X.shape)
```

Model Evaluation

- **Purpose:** The dashed gray line represents a reference or baseline, often used to indicate a perfect correlation or comparison benchmark.
- **Appearance:** A diagonal line from (0, 0) to (1, 1) with a dashed style.
- **Context**: In performance graphs like ROC or Precision-Recall curves, it shows random performance or no better than chance.
- **Interpretation**: Curves above this line indicate better-than-random performance, while curves below it indicate worse-than-random performance.

```
In [151]: from sklearn.metrics import roc_curve, auc
          import matplotlib.pyplot as plt
          from sklearn.metrics import roc_curve, auc

In [152]: plt.figure(figsize=(10, 6))

Out[152]: <Figure size 1000x600 with 0 Axes>

          <Figure size 1000x600 with 0 Axes>

In [153]: plt.plot([0, 1], [0, 1], color='gray', linestyle='--')

Out[153]: [<matplotlib.lines.Line2D at 0x20b654dddc0>]
```



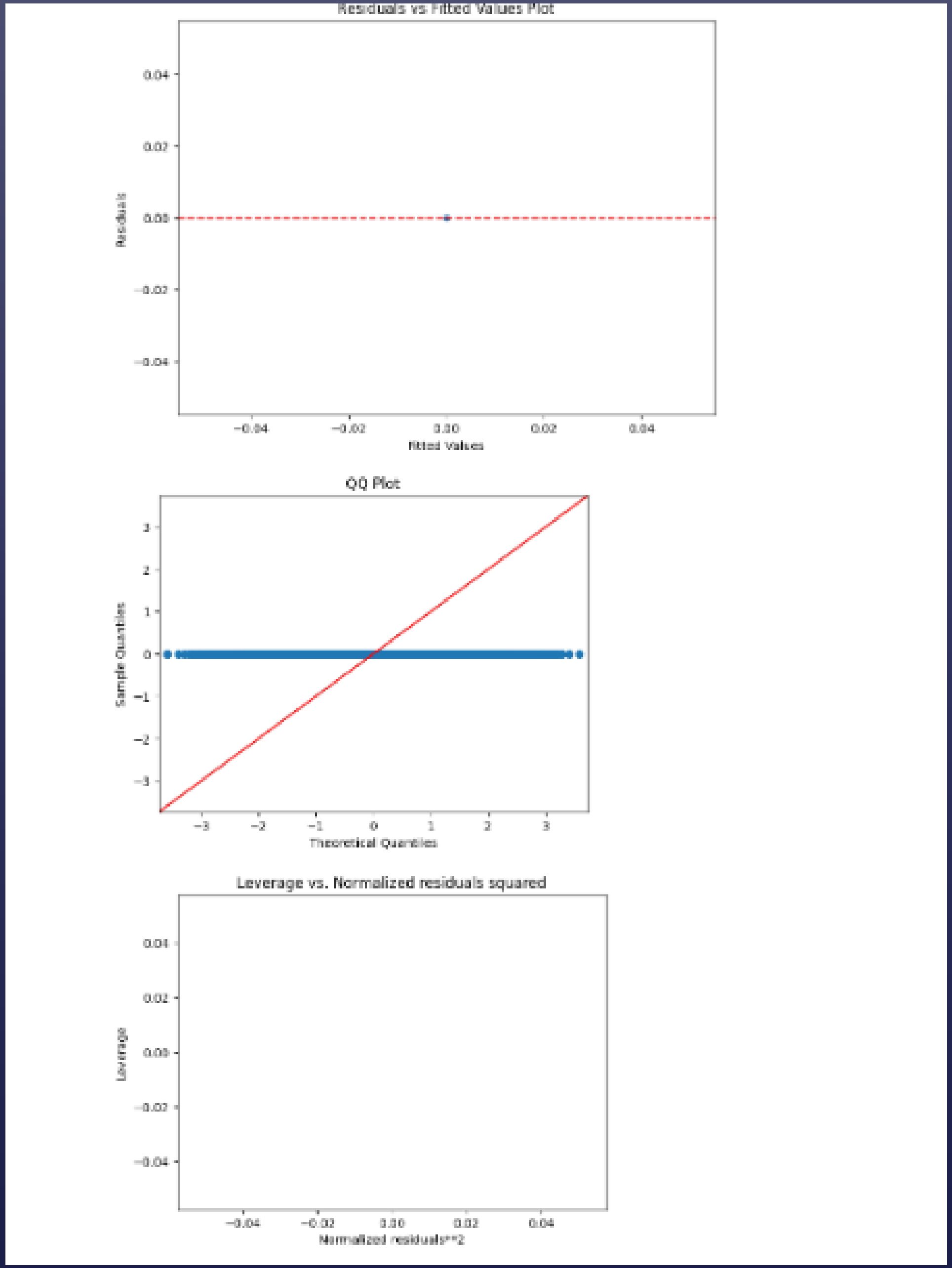
Residuals vs Fitted Values Plot

Residuals

Purpose: To check for patterns in residuals (errors) and assess model fit.
Key Features:
X-axis: Fitted values (predicted values).
Y-axis: Residuals (actual - predicted values).
Red Dashed Line: Represents a zero residual, indicating perfect prediction.
Interpretation: Random scatter around the line suggests a good model; patterns or trends indicate model issues.

QQ Plot
Purpose: To check if residuals follow a normal distribution.
Key Feature: The line represents the expected normal distribution. Residuals should closely follow this line for normality.

Leverage vs Residuals Plot
Purpose: To identify influential data points that have a large impact on the model.
Key Feature: Points far from the center have high leverage and might disproportionately affect the model.



Thank You