Final Report

Batch details	PGPDSE-FT CHENNAI SEP23
Team members	Chandru V, Giridaran.D, Keerthi Sharran.S, Rohit.V, Tejaswini.S
Domain of Project	Telecommunication
Proposed project title	Telecommunication churn prediction
Group Number	1
Team Leader	Tejaswini.S
Mentor Name	Ms.Vidya Kannaiah

Date: 10/04/2024

Signature of the Mentor

Signature of the Team Leader

Table of Contents

Project Details

Industry Review

1. Current practices, Background Research:

The telecommunications industry is increasingly leveraging data analytics to enhance operational efficiency and improve service delivery. By analyzing vast amounts of data from various sources such as network logs and customer interactions, telecom companies can optimize their processes, enhance network performance, and provide better services to their customers.

Telecom data analytics plays a crucial role in detecting and preventing fraudulent activities, improving customer service, and optimizing network resources. The primary objectives include providing detailed network performance insights, supporting decision-making processes, offering in-depth customer behavior analysis, identifying potential network issues proactively, and enhancing operational efficiency while reducing costs. Data security is paramount in telecom analytics to protect sensitive information and ensure the integrity of network data. Monitoring networks for anomalies and suspicious activities is essential to maintain data privacy and prevent security breaches.

Telecom Analytics Objective:

- Providing detailed network performance insights.
- Supporting decision-making processes in network management.
- Offering in-depth customer behavior analysis for personalized services.
- Identifying potential network issues proactively to ensure uninterrupted service.
- Enhancing operational efficiency and reducing costs through optimized resource allocation.

2. Literature Review:

1. Information Systems & Information Technology in Telecommunication Sectors:

The telecommunication sector is a key industry where information technology plays a vital role. Information technology is instrumental in enhancing operational efficiency and service delivery within the telecommunication sectors by leveraging data and information effectively.

2. Telecommunication Predictions and Decision Support System (DSS):

Telecommunication prediction involves utilizing data analytics to forecast future trends and optimize operational costs. By analyzing historical data and patterns, predictive analytics can anticipate potential network issues and optimize resource allocation in advance.

3. Role of Predictive Analytics in Telecommunication Industry:

Predictive analytics plays a crucial role in the telecommunication sector by enabling proactive network management and enhancing overall operational efficiency. By identifying potential network issues early on, predictive analytics helps telecom companies improve service quality, prevent downtime, and reduce operational costs.

4. Financial Factors in Telecommunication Predictive Analytics:

Cost-effectiveness is a significant benefit of implementing predictive analytics in the telecommunication industry. Efficient utilization of data and information can lead to cost reductions by optimizing network performance, streamlining operations, and improving resource allocation. By simplifying data management and reducing irregularities, telecommunication companies can enhance their predictive analytics capabilities while minimizing costs.

Dataset and Domain

1. **Domain Background:**

In the telecom industry, customers are able to choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the telecommunications industry experiences an average of 15-25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition.

For many incumbent operators, retaining high profitable customers is the number one business goal. To reduce customer churn, telecom companies need to predict which customers are at high risk of churn**.**

2. **Dataset:**

This is the dataset made available from Kaggle. The dataset consists of 69999 rows, and 172 columns which describe telecom records on different types of parameters.
In order to simplify calculations and model building given the limitations of our machine's specifications, we have decided to drop some columns which are not needed for our model building in the further process.

Our dataset includes the acronyms listed below:

Acronyms:	Description:
0. CIRCLE_ID	Telecom circle area to which the customer belongs to
1. LOC	Local calls within same telecom circle
2. STD	STD calls outside the calling circle
3. IC	Incoming calls
4. OG	Outgoing calls
5. T2T	Operator T to T i.e. within same operator mobile to mobile
6. T2M	Operator T to other operator mobile
7. T2O	Operator T to other operator fixed line

8. T2F	Operator T to fixed lines of T
9. T2C	Operator T to its own call center
10.ARPU	Average revenue per user
11.MOU	Minutes of usage voice calls
12.AON	Age on network number of days the customer is using the operator T network
13.ONNET	All kind of calls within the same operator network
14.OFFNET	All kind of calls outside the operator T network
15.ROAM	Indicates that customer is in roaming zone during the call
16.SPL	Special calls
17.ISD	ISD calls
18.RECH	Recharge
19.NUM	Number
20.AMT	Amount in local currency
21.MAX	Maximum
22.DATA	Mobile internet
23.3G	G network
24.AV	Average
25.VOL	Mobile internet usage volume in MB
26.2G	G network
27.PCK	Prepaid service schemes called PACKS
28.NIGHT	Scheme to use during specific night hours only
29.MONTHLY	Service schemes with validity equivalent to a month
30.SACHET	Service schemes with validity smaller than a month
31.*.6	KPI for the month of June

32.*.7	KPI for the month of July
33.*.8	KPI for the month of August
34.FB_USER	Service scheme to avail services of Facebook and similar social networking sites
35.VBC	Volume based cost when no specific scheme is not purchased and paid as per usage

3. Variable Categorization:

Variables can be categorized in different ways depending on the context. For our dataset, we have used numerical and categorical types of categorization. Using pandas function **df.info()** and **df.isna().sum().sum()**we have checked for the Data types and Null values In the dataset, we have the following numbers of numerical and categorical variables:

i. **Numerical Variables**: A numerical variable is a type of quantitative variable that takes on a numerical value. It can be further categorized as either continuous or discrete. In total, we have float64(135) and int64(28) so, total 163 columns having quantitative variables.

ii. **Categorical Variables**: Categorical variables are a type of qualitative variable that can take on a limited number of values or categories. There are a total of 9 columns present having object data type.

Screenshot from the notebook:

:

```
In [73]: 1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69999 entries, 0 to 69998
Columns: 172 entries, id to churn_probability
dtypes: float64(135), int64(28), object(9)
memory usage: 91.9+ MB
```

```
In [30]: 1 df.isna().sum()

Out[30]: id                        0
         circle_id                 0
         loc_og_t2o_mou          702
         std_og_t2o_mou          702
         loc_ic_t2o_mou          702
         last_date_of_month_6      0
         last_date_of_month_7    399
         last_date_of_month_8    733
         arpu_6                    0
         arpu_7                    0
         arpu_8                    0
         onnet_mou_6            2768
         onnet_mou_7            2687
         onnet_mou_8            3703
         offnet_mou_6           2768
         offnet_mou_7           2687
         offnet_mou_8           3703
         roam_ic_mou_6          2768
         roam_ic_mou_7          2687
```

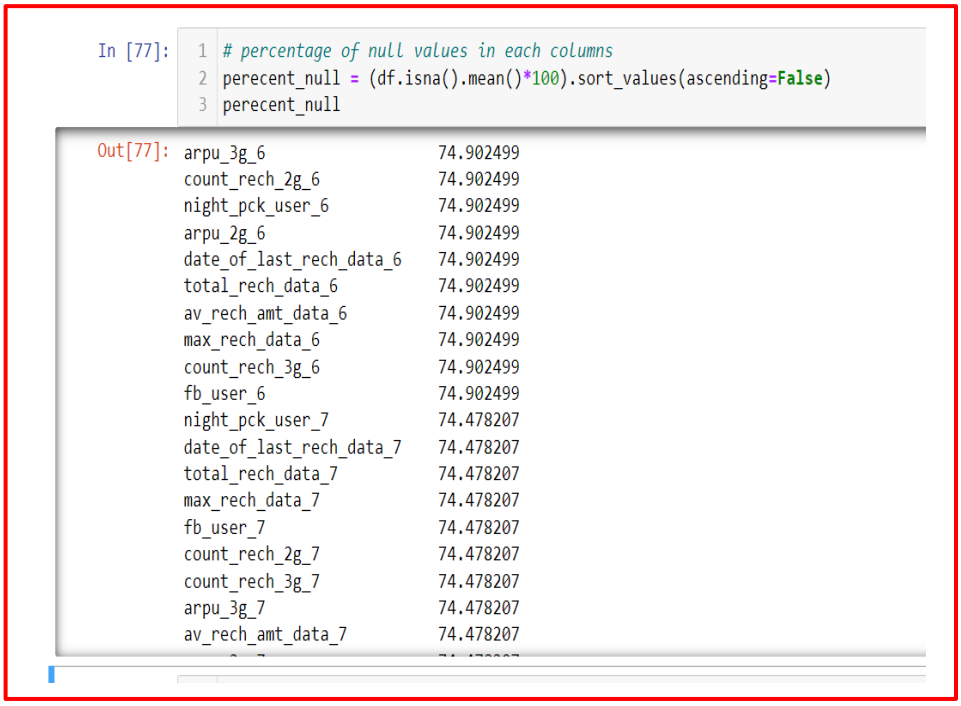
```
In [31]: 1 print("Total number of null values:", df.isna().sum().sum())

Total number of null values: 1835086
```

4. Pre-processing:

In the process of analyzing data, preprocessing plays a vital role as it involves performing activities such as cleaning, transforming, and refining the data to make it suitable for analysis. The primary objective of this step is to ensure the accuracy, completeness, and readiness of the data for further analysis.

From the above-mentioned numbers and the image, we have null values in our dataset. So, in order to identify the percentage of null values we have used the pandas dataframe function isnull(). Further, to find the percentage of the null values, we have used mean().

Screenshot from the notebook:



```
In [77]:
# percentage of null values in each columns
perecent_null = (df.isna().mean()*100).sort_values(ascending=False)
perecent_null
```

```
Out[77]:
arpu_3g_6                   74.902499
count_rech_2g_6             74.902499
night_pck_user_6            74.902499
arpu_2g_6                   74.902499
date_of_last_rech_data_6    74.902499
total_rech_data_6           74.902499
av_rech_amt_data_6          74.902499
max_rech_data_6             74.902499
count_rech_3g_6             74.902499
fb_user_6                   74.902499
night_pck_user_7            74.478207
date_of_last_rech_data_7    74.478207
total_rech_data_7           74.478207
max_rech_data_7             74.478207
fb_user_7                   74.478207
count_rech_2g_7             74.478207
count_rech_3g_7             74.478207
arpu_3g_7                   74.478207
av_rech_amt_data_7          74.478207
```

From the above output, it is feasible to drop columns like **date_of_last_rech_data_6, max_rech_data_6, count_rech_2g_6**, **night_pck_user_6** and so as it contains more than 70% null values and it will be difficult to extract value from them.

Screenshot from the notebook:

```
In [44]: 1 df['loc_og_t2o_mou'].unique()
Out[44]: array([ 0., nan])

In [45]: 1 df['std_og_t2o_mou'].unique()
Out[45]: array([ 0., nan])

In [46]: 1 df['loc_ic_t2o_mou'].unique()
Out[46]: array([ 0., nan])

In [47]: 1 df['circle_id'].unique()
Out[47]: array([109], dtype=int64)

In [48]: 1 df['last_date_of_month_6'].unique()
Out[48]: array(['6/30/2014'], dtype=object)

In [49]: 1 df['last_date_of_month_7'].unique()
Out[49]: array(['7/31/2014', nan], dtype=object)

In [50]: 1 df['last_date_of_month_8'].unique()
Out[50]: array(['8/31/2014', nan], dtype=object)
```

We have found some columns with only one unique value, like **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**, so it is of no use for the analysis, hence we have dropped those columns.

5. Alternate sources of data that can supplement the core dataset:

Alternate sources of data are essential in a data science project as they can provide a more comprehensive and accurate analysis. Using alternate sources of data can bring diverse perspectives and viewpoints to a project. This can be especially valuable in cases where the data being analyzed is subjective or complex, as different sources may offer varying viewpoints that can provide a more complete picture.

We have come across two alternate sources that can help us recover at the very least some predictor variables of the dataset we are primarily working on. They are listed as below:

1. **Predicting Customer Churn in Telecommunications on Kaggle:**

   Predicting customer churn is a common and valuable application of data science. This activity focuses on understanding which customers are likely to leave a company. By understanding which customers will churn, and perhaps when or why they might churn, companies can proactively manage these customers and attempt to increase retention. This dataset was provided from the public datasets available on Kaggle. The above-mentioned dataset is linked below: https://www.kaggle.com/code/willturnerau/predicting-customer-churn-in-telecommunications

2. **Telecommunications Industry Customer churn dataset on Kaggle:**

   The Telco customer churn data contains information about a fictional telco company that provided home phone and Internet services to 7043 customers in California in Q3. It indicates which customers have left, stayed, or signed up for their service. Multiple important demographics are included for each customer, as well as a Satisfaction Score, Churn Score, and Customer Lifetime Value (CLTV) index. The above-mentioned dataset is linked below: :https://www.kaggle.com/datasets/aadityabansalcodes/telecommunications-industry-customer-churn-datase

6. Project Justification:

Project Statement:

In the telecom industry, customers are able to choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the telecommunications industry experiences an average of 15-25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition.

For many incumbent operators, retaining high profitable customers is the number one business goal. To reduce customer churn, telecom companies need to predict which customers are at high risk of churn.

Complexity Involved:

The complexities associated with this project include:

1. Predictive Analysis Complexity:

- Developing sophisticated algorithms to predict customer behavior accurately.
- Identifying key indicators and patterns that signal potential churn risks.
- Implementing predictive models that can effectively forecast customer churn probabilities.

2. Data Management Complexity:

- Handling vast amounts of customer data to extract valuable insights.
- Ensuring data accuracy, reliability, and security in predictive modeling processes.
- Integrating data from various sources to enhance predictive accuracy and decision-making.

3. Customer Engagement Complexity:

- Designing personalized retention strategies tailored to individual customer needs.
- Implementing proactive communication and engagement initiatives to reduce churn rates.
- Balancing customer satisfaction with operational efficiency to maximize retention efforts.

Project Outcome:

Our project centers on harnessing telecommunications data to forecast customer churn while simultaneously pinpointing the root causes leading customers to sever ties with the company. To achieve this, we embark on a comprehensive methodology that begins with the collection of diverse telecommunications data, encompassing customer usage patterns, service subscriptions, billing details, and interactions. Following this, an in-depth Exploratory Data Analysis (EDA) is conducted to grasp feature distributions, detect outliers, and uncover correlations. Visualization techniques are employed to discern customer behaviors and patterns, laying the foundation for subsequent analyses.

In the realm of model development, we employ machine learning algorithms, including logistic regression, decision trees, or ensemble methods, to construct a robust churn prediction model. This model is trained on historical data and rigorously validated using cross-validation techniques to ensure its accuracy and generalizability. Simultaneously, we delve into feature engineering to extract pertinent information that may contribute to customer churn. Furthermore, our methodology includes an insightful analysis to interpret the model and unveil the prominent factors influencing churn using advanced analytics techniques such as feature importance and SHAP.

The outcome of our project comprises a formidable churn prediction model, offering the telecommunications company a predictive tool for anticipating customer exits. Moreover, we present a detailed insights report elucidating the underlying reasons for customer churn. By combining the model's findings with customer surveys and feedback analysis, we aim to provide a holistic understanding of the factors influencing customer decisions. The project concludes with strategic recommendations tailored to address the identified reasons for churn, empowering the company to implement targeted retention strategies. We anticipate that these insights will not only help in reducing churn rates but also foster customer loyalty, fortifying the company's competitive position in the dynamic telecommunications market.

Exploratory Data Analysis (EDA):

1. Summary:

Summary of numerical variables:

```
In [99]: df2.describe(include='number')
```

Out[99]:

	arpu_6	arpu_7	arpu_8	onnet_mou_6	onnet_mou_7	onnet_mou_8	offnet_mou_6	offnet_mou_7	offnet_mou_8	roam_ic_mou_6	roam
count	69999.000000	69999.000000	69999.000000	67231.000000	67312.000000	66296.000000	67231.000000	67312.000000	66296.000000	67231.000000	67
mean	283.134365	278.185912	278.858826	133.153275	133.894438	132.978257	198.874771	197.153383	196.543577	9.765435	
std	334.213918	344.366927	351.924315	299.963093	311.277193	311.896596	316.818355	322.482226	324.089234	57.374429	
min	-2258.709000	-1289.715000	-945.808000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	93.581000	86.714000	84.095000	7.410000	6.675000	6.410000	34.860000	32.240000	31.575000	0.000000	
50%	197.484000	191.588000	192.234000	34.110000	32.280000	32.100000	96.480000	91.885000	91.800000	0.000000	
75%	370.791000	365.369500	369.909000	119.390000	115.837500	115.060000	232.990000	227.630000	229.345000	0.000000	
max	27731.088000	35145.834000	33543.624000	7376.710000	8157.780000	10752.560000	8362.360000	7043.980000	14007.340000	2850.980000	4

- The average revenue per user (arpu) slightly fluctuates over the three months but remains relatively stable, indicating consistent revenue generation from users.
- On-net and off-net minutes show similar patterns across the three months, indicating consistent calling behavior.
- Roaming minutes and data usage might vary more, potentially indicating seasonal trends or promotional offers affecting roaming and data usage.
- The churn probability provides an estimate of the likelihood of customers leaving the service provider.
- Analyzing churn probability alongside other metrics can help identify factors influencing customer attrition and inform retention strategies.

Summary of Categorical Variables:

```
In [100]: df2.describe(include='object')
```

Out[100]:

	date_of_last_rech_6	date_of_last_rech_7	date_of_last_rech_8	date_of_last_rech_data_6	date_of_last_rech_data_7	date_of_last_rech_data_8
count	68898	68765	67538	17568	17865	18417
unique	30	31	31	30	31	31
top	6/30/2014	7/31/2014	8/31/2014	6/30/2014	7/31/2014	8/31/2014
freq	11880	12206	10324	1317	1282	1388

- The fact that the most frequent dates are consistently around the last day of the month (e.g., June 30th, July 31st, August 31st) suggests that many users may have a preference for recharging towards the end of the billing cycle.

- While the recharge date columns (date_of_last_rech_6, date_of_last_rech_7, date_of_last_rech_8) exhibit relatively high counts and frequent dates, the recharge data columns (date_of_last_rech_data_6, date_of_last_rech_data_7, date_of_last_rech_data_8) have lower counts and less frequent dates.
- This suggests that recharges for data services may be less common or less regular compared to general recharges.

2. Relationship between variables:

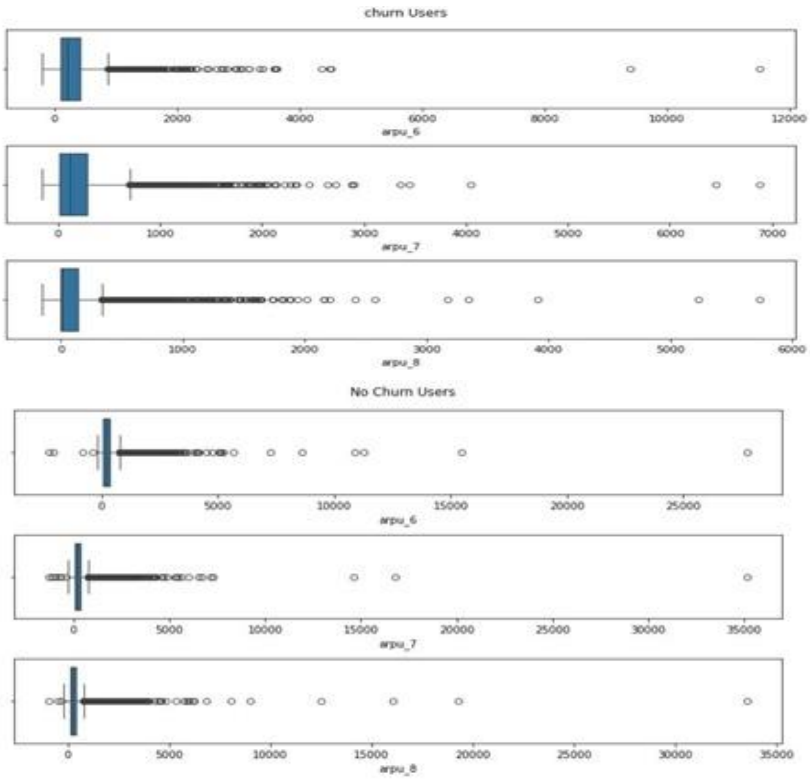
The relationship between variables refers to how two or more variables are associated or related to each other. Understanding the relationship between variables is important data analysis and can help identify patterns, trends, and associations in the data. Here are some common types of relationships between variables:

1. Positive relation
2. Negative relation
3. Linear relation
4. Non-linear relation
5. Correlation

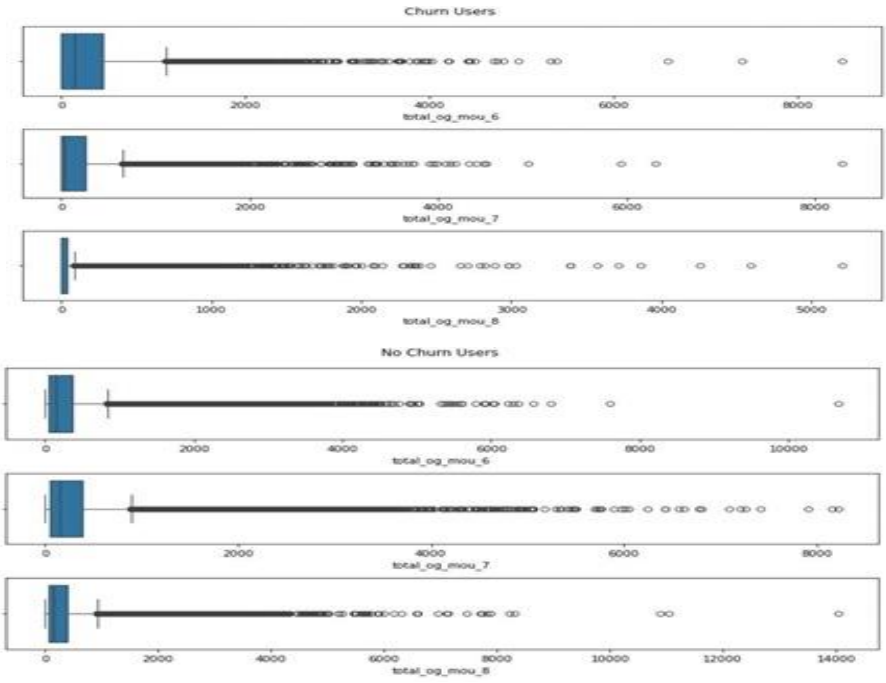
Univariate Analysis:

Firstly we did univariate analysis These plots provide valuable insights into the distribution and trends of average revenue per user, total outgoing calls minutes of usage, roam incoming and outgoing calls minutes of usage over 3 months among users. For users who are not likely to churn, their usage to remains stable, whereas, for users who eventually churn, their usage decreases over the months.

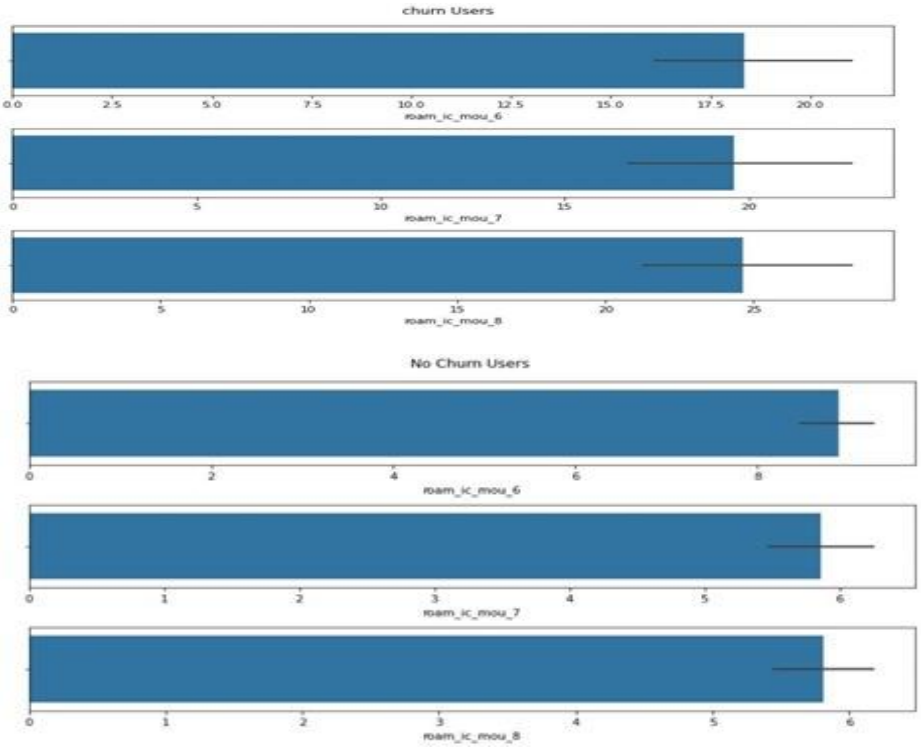
Average revenue per user vs Churn probability:



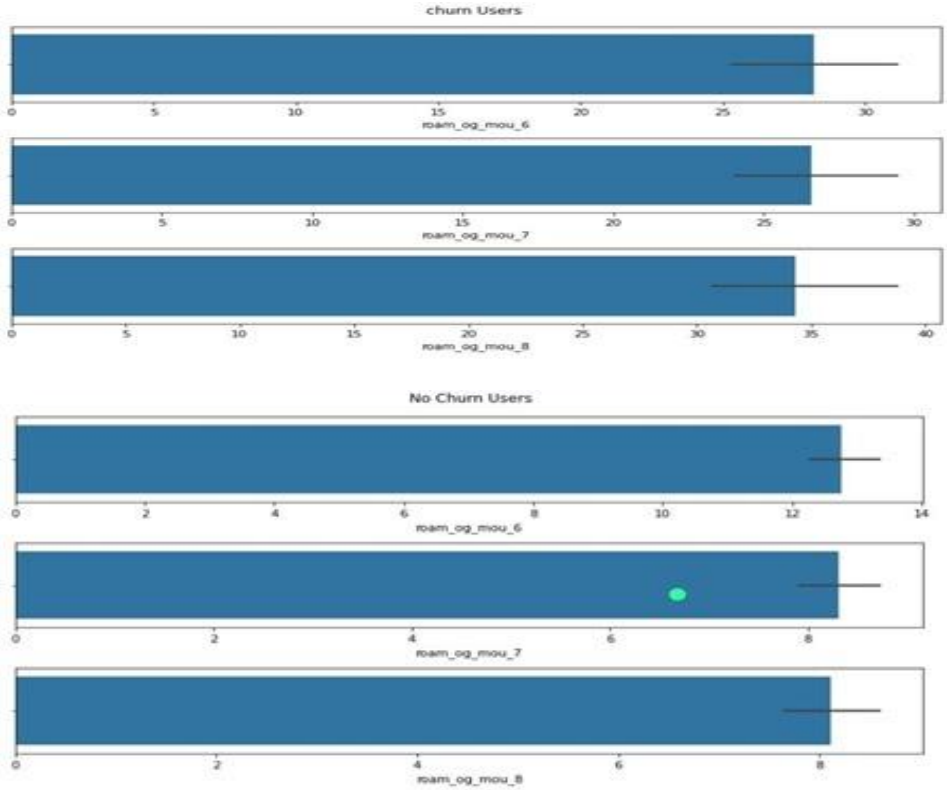
Total Outgoing calls vs Churn Probability:



Roam Incoming calls vs Churn Probability:



Roam Out-going calls Vs Churn Probability:

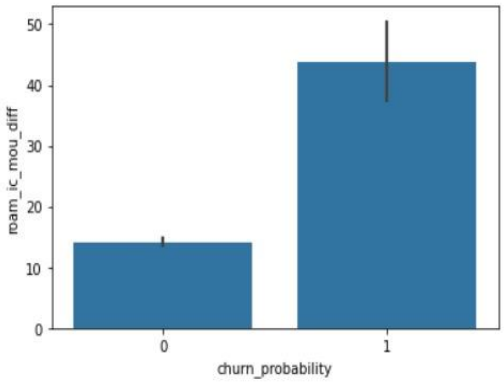


Bivariate Analysis:

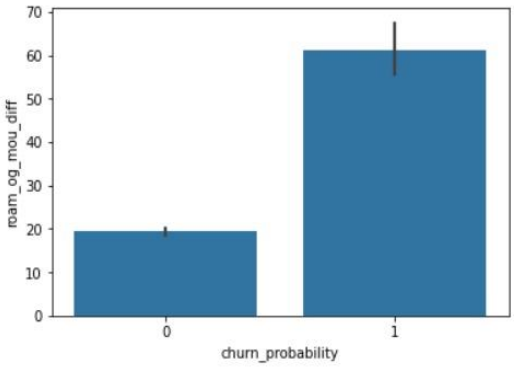
We did bivariate analysis for the numerical variables. In the below image, first graph represents the relationship between roam income minutes of usage difference and churn probability. On the other hand, the second graph concludes the relationship of roam out going minutes of usage difference with churn probability. From both of the graphs, it can be concluded that the usage of roam incoming and outgoing calls of the customers who are not churned are constant, whereas customers who are churned have difference in minutes of usage, indicating decrease in usage.

Screenshot from the notebook:

```
In [128]: 1 df2['roam_ic_mou_diff'] = np.abs(df2['roam_ic_mou_6'] - df2['roam_ic_mou_7'] - df2['roam_ic_mou_8'])

In [132]: 1 sns.barplot(data=df2, x='churn_probability', y='roam_ic_mou_diff')

Out[132]: <Axes: xlabel='churn_probability', ylabel='roam_ic_mou_diff'>
```




```
In [130]: 1 df2['roam_og_mou_diff'] = np.abs(df2['roam_og_mou_6'] - df2['roam_og_mou_7'] - df2['roam_og_mou_8'])

In [133]: 1 sns.barplot(data=df2, x='churn_probability', y='roam_og_mou_diff')

Out[133]: <Axes: xlabel='churn_probability', ylabel='roam_og_mou_diff'>
```



3. Distribution of Variables:

To get a better understanding of the variables, we have plotted graphs for every column in the dataset which is also called univariate analysis. It involves analyzing a single variable in isolation, without considering its relationship with other variables in the dataset. Univariate analysis can be used to determine the central tendency of the variable, its dispersion or variability, and its shape or distribution.

The first step for univariate analysis is that we have plotted graphs for categorical columns.

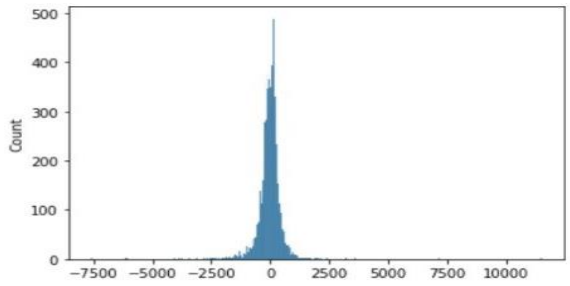
Screenshot from the notebook:

```
churn_df = df1[df1['churn_probability'] ==  1]
```

```
diff = churn_df['arpu_6'] - churn_df['arpu_7'] - churn_df['arpu_8']
```

```
sns.histplot(diff)
```

```
<Axes: ylabel='Count'>
```

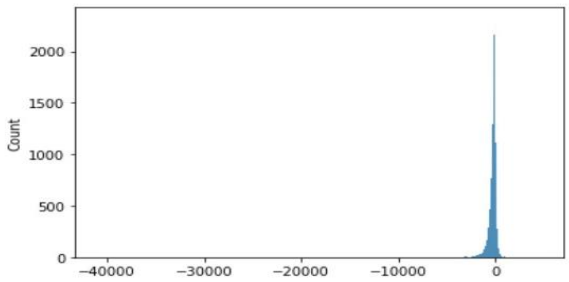



```
no_churn_df = df1[df1['churn_probability'] == 0]
```

```
diff = no_churn_df['arpu_6'] - no_churn_df['arpu_7'] - no_churn_df['arpu_8']
```

```
sns.histplot(diff)
```

```
<Axes: ylabel='Count'>
```



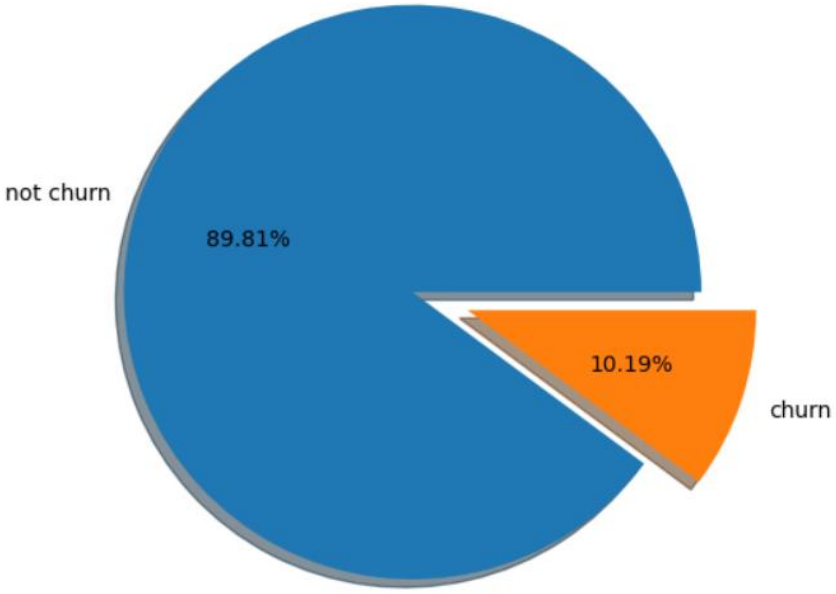
From the above plot came across with the inferences:

- The telecom company has many users with negative average revenues in both phases. These users are likely to churn.
- Most customers prefer the plans of '0' category.
- Revenue generated by the Customers who are about to churn is very unstable.
- The Customers whose average revenue per user (arpu) decreases in 7th month are more likely to churn when compared to ones with increase in average revenue per user (arpu).

The output shows that the variable Ethnicity Not Span/Hispanic has the highest VIF. Removing this feature from the dataset and set the threshold of VIF as to 10, it means considering feature having VIF less than or equal to 10 (can be changed as per business requirement

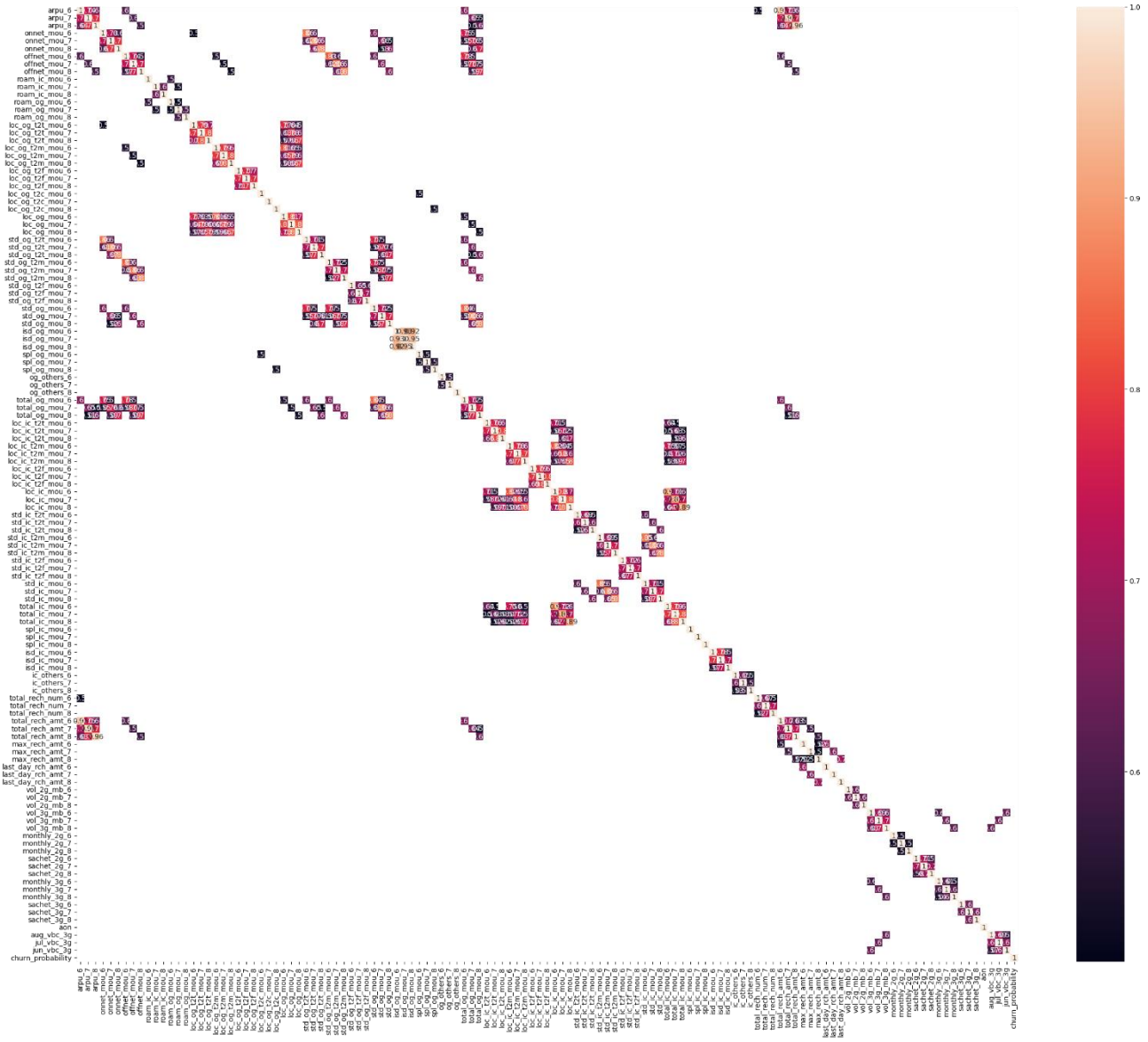
Target Imbalance:

From the below plot we can see we have an imbalanced dataset will affect the recall precision score after ML algorithm.



4.Checking for multicollinearity between the variables:

From the below correlation matrix, we can see there are lot of Multi correlation in the data which should be reduced in the futher process.



Base Model before Feature Engineering:

- Models are created before feature engineering in order to understand machine learning workflows.
- Building a baseline model using raw features can provide a benchmark for performance. This helps in understanding the predictive power of the initial set of features and gives a reference point for improvement after feature engineering.
- In complex datasets, it may not be immediately clear how to engineer features effectively. Base models are created using Logistic Regression, Decision Tree, Random Forest and KNN.

```
1 model_metrics
```

	Model	Sensitivity_train	Specificity_train	Accuracy_train	F1-Score_train	Precission_train	ROC_AUC_Score_train	Sensitivity_test
0	Base Model LogisticRegression	0.445312	0.985843	0.930386	0.565865	0.775916	0.898532	0.445794
1	Base_model_DecisionTreeClassifier	1.000000	0.946978	1.000000	1.000000	1.000000	1.000000	0.576168
2	Base_model_RandomForestClassifier	1.000000	0.981018	1.000000	1.000000	1.000000	1.000000	0.608411
3	Base_model_KNN	0.490585	0.974867	0.934305	0.603425	0.783680	0.961244	0.375701

Feature Engineering:

Screenshot from the notebook:

```
# code to get difference between 3 columns.
new_columns = []
for col in df2.columns:
    if col == 'churn_probability':
        break
    if (('6' in col)) and col not in ['date_of_last_rech_6','date_of_last_rech_7','date_of_last_rech_8','churn_probability']:
        idx = df2.columns.get_loc(col)
        new_col_name = col.split('6')[0] + "diff"
        new_columns.append(new_col_name)
        df2[new_col_name] = np.abs((df2.iloc[:,idx] - df2.iloc[:,idx+1]) + (df2.iloc[:,idx+1] - df2.iloc[:,idx+2]))
```

- From the above description we can see that, the average usage of services such as average revenue per user (arpu), minutes of usage (onnet_mou, offnet_mou), and roaming (roam_ic_mou, roam_og_mou) is notably higher in the sixth month compared to the seventh and eighth months.
- There seems to be a decrease in these usage metrics over time, indicating a potential decline in customer engagement before churning.
- Total recharge amount (total_rech_amt) and the number of recharges (total_rech_num) show a decrease over time, suggesting a decline in customer spending before churning.
- The maximum recharge amount and the number of recharges vary widely, indicating different segments of customers with varying recharge behaviors.
- There's a noticeable decline in that customers who churn tend to exhibit declining usage patterns, reduced recharge amounts, and decreased engagement with additional services over time, indicating potential dissatisfaction or loss of interest in the services provided by the telecom company.
- As a result, Absolute differences between consecutive values are computed for the selected columns, and the absolute differences are summed up for each row. This creates a new feature that represents the overall change across the specified columns.

Null Value imputation:

```
In [ ]:
# filling null values
for col in df4.columns:
    if col not in ['date_of_last_rech_6','date_of_last_rech_7','date_of_last_rech_8']:
        df4[col].fillna(df4[col].median(),inplace=True)
```

```
In [266]:
df4.isna().sum()
```

```
Out[266]:
arpu_diff                 0
onnet_mou_diff            0
offnet_mou_diff           0
roam_ic_mou_diff          0
roam_og_mou_diff          0
loc_og_t2t_mou_diff       0
loc_og_t2m_mou_diff       0
loc_og_t2f_mou_diff       0
loc_og_t2c_mou_diff       0
loc_og_mou_diff           0
std_og_t2t_mou_diff       0
std_og_t2m_mou_diff       0
std_og_t2f_mou_diff       0
std_og_mou_diff           0
isd_og_mou_diff           0
spl_og_mou_diff           0
og_others_diff            0
total_og_mou_diff         0
loc_ic_t2t_mou_diff       0
loc_ic_t2m_mou_diff       0
loc_ic_t2f_mou_diff       0
loc_ic_mou_diff           0
std_ic_t2t_mou_diff       0
std_ic_t2m_mou_diff       0
std_ic_t2f_mou_diff       0
std_ic_mou_diff           0
total_ic_mou_diff         0
spl_ic_mou_diff           0
isd_ic_mou_diff           0
ic_others_diff            0
total_rech_num_diff       0
```

- Imputing the median helps to preserve the distribution of the data, especially if the distribution is skewed. Since the median is less affected by extreme values, it provides a better representation of the central value in skewed distributions.

- Imputing with the median is less affected by missing values compared to imputing with the mean. If there are missing values in the dataset, using the median ensures that the imputed values do not significantly influence the overall mean.

Statistical test for Variable Significance:

- ANOVA is useful for detecting differences in means across multiple groups and can provide insights into the relationship between categorical variables and the target variable (churn) in the Telecom Churn Case Study.
- It ANOVA can be used to examine whether there are differences in churn rates based on the type of services subscribed to by customers (e.g., internet service type, additional service subscriptions).
- This analysis can provide insights into which services are more strongly associated with churn.
- We can use ANOVA to test the relationship of independent column with the Target column which is a categorical variable.
- H0: There is no relationship between the variables.
- H1: There is relationship between the variables.

```
1 df_stats_test
```

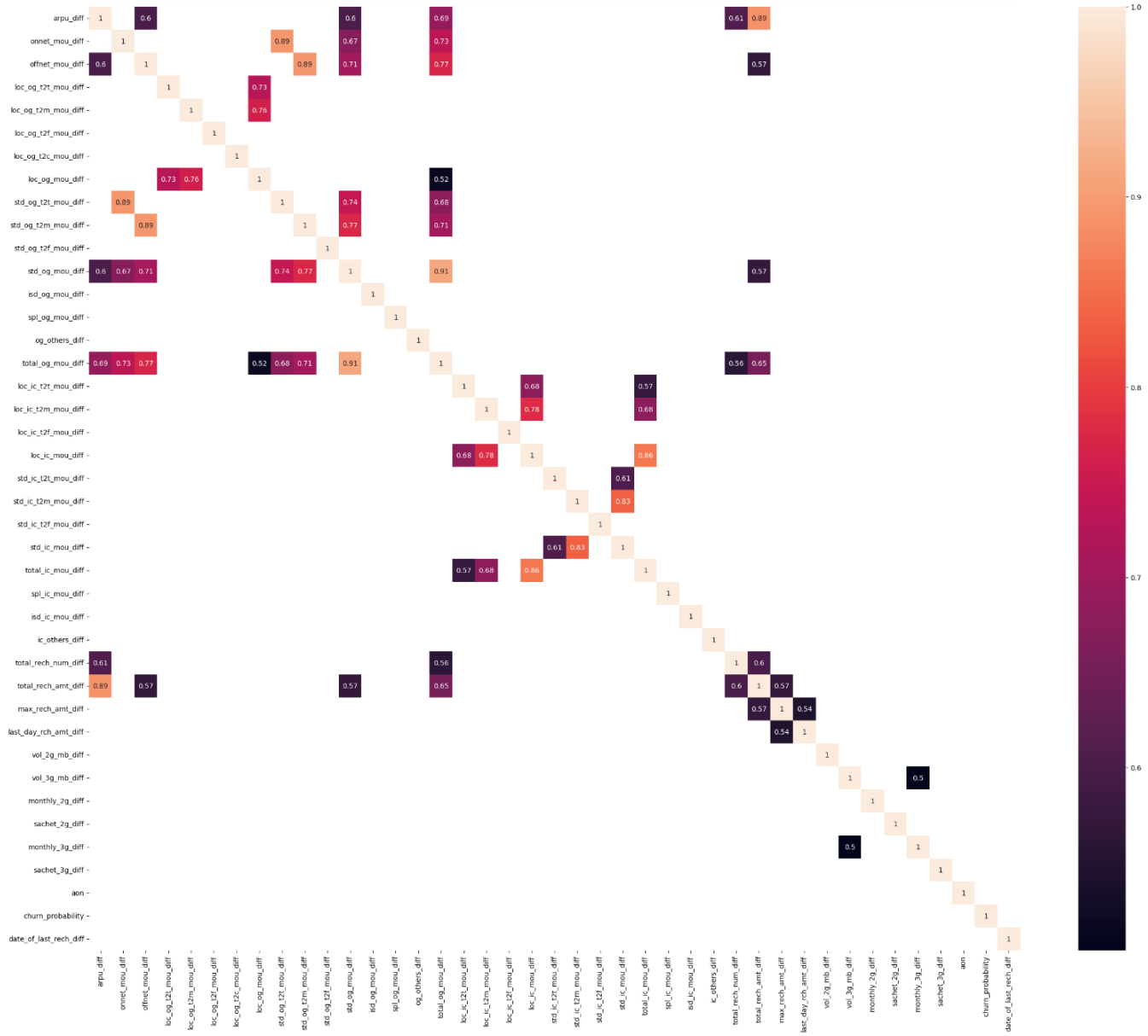
	features	p_value	comment
0	arpu_diff	0.000000e+00	Reject H0
1	onnet_mou_diff	4.859378e-276	Reject H0
2	offnet_mou_diff	1.207461e-285	Reject H0
3	roam_ic_mou_diff	8.502144e-02	Fail to Reject H0
4	roam_og_mou_diff	9.521537e-01	Fail to Reject H0
5	loc_og_t2t_mou_diff	4.726452e-32	Reject H0
6	loc_og_t2m_mou_diff	2.249904e-76	Reject H0
7	loc_og_t2f_mou_diff	2.397146e-15	Reject H0
8	loc_og_t2c_mou_diff	7.090093e-21	Reject H0
9	loc_og_mou_diff	4.384808e-82	Reject H0
10	std_og_t2t_mou_diff	1.855587e-290	Reject H0
11	std_og_t2m_mou_diff	2.228472e-275	Reject H0
12	std_og_t2f_mou_diff	4.867436e-05	Reject H0
13	std_og_mou_diff	0.000000e+00	Reject H0
14	isd_og_mou_diff	6.092321e-12	Reject H0
15	spl_og_mou_diff	8.693341e-73	Reject H0
16	og_others_diff	1.165216e-02	Reject H0
17	total_og_mou_diff	0.000000e+00	Reject H0
18	loc_ic_t2t_mou_diff	6.678584e-40	Reject H0
19	loc_ic_t2m_mou_diff	4.164087e-85	Reject H0
20	loc_ic_t2f_mou_diff	1.060046e-20	Reject H0
21	loc_ic_mou_diff	4.256206e-97	Reject H0
22	std_ic_t2t_mou_diff	1.476158e-45	Reject H0
23	std_ic_t2m_mou_diff	3.269760e-70	Reject H0
24	std_ic_t2f_mou_diff	2.888029e-07	Reject H0
25	std_ic_mou_diff	2.022603e-99	Reject H0
26	total_ic_mou_diff	0.000000e+00	Reject H0
27	spl_ic_mou_diff	5.255129e-19	Reject H0
28	isd_ic_mou_diff	9.768566e-15	Reject H0
29	ic_others_diff	1.288884e-04	Reject H0
30	total_rech_num_diff	0.000000e+00	Reject H0
31	total_rech_amt_diff	0.000000e+00	Reject H0
32	max_rech_amt_diff	0.000000e+00	Reject H0
33	last_day_rch_amt_diff	1.333877e-170	Reject H0
34	vol_2g_mb_diff	6.329836e-32	Reject H0
35	vol_3g_mb_diff	1.872251e-45	Reject H0
36	monthly_2g_diff	6.104409e-23	Reject H0
37	sachet_2g_diff	1.724276e-76	Reject H0
38	monthly_3g_diff	1.632166e-31	Reject H0
39	sachet_3g_diff	7.089041e-18	Reject H0
40	aon	1.093892e-262	Reject H0
41	date_of_last_rech_diff	3.839397e-138	Reject H0

- It's important to interpret the results of ANOVA alongside other analyses and consider potential confounding variables to draw meaningful conclusions.

- From the above metrics we can see that roam_ic and roam_og has no relationship with the target variable. So we can drop those columns to build the better model.

Correlation After Feature Engineering:

From this we can see the Multicollinearity in the data has been reduced.



Presence of outliers and its treatment:

Outliers are data points that are significantly different from other data points in a dataset. They can occur due to measurement errors, data entry errors, or real-world phenomena that deviate from the norm. Outliers can significantly affect the results of statistical analyses, as they can skew the mean and standard deviation of a dataset and lead to incorrect conclusions.

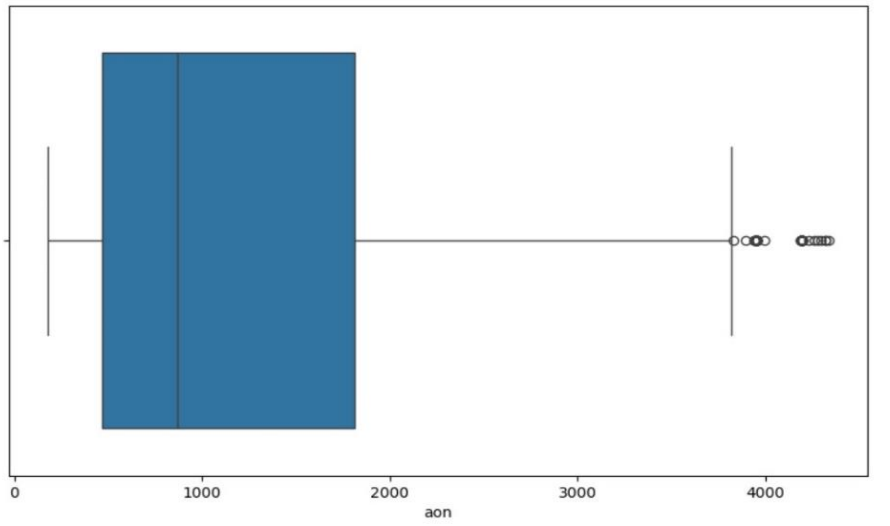
There are various techniques to identify outliers in a dataset, including:

1. **Box plot:** A box plot can provide a visual representation of the distribution of a dataset and highlight any values that fall outside the whiskers of the plot.
2. **Z-score:** The Z-score is a statistical measure that indicates how many standard deviations a data point is from the mean of a dataset. Data points with a Z-score greater than 3 or less than -3 are typically considered outliers.
3. **Interquartile range (IQR):** The IQR is the range between the 25th and 75th percentiles of a dataset. Data points that fall more than 1.5 times the IQR above the 75th percentile or below the 25th percentile is typically considered outliers.

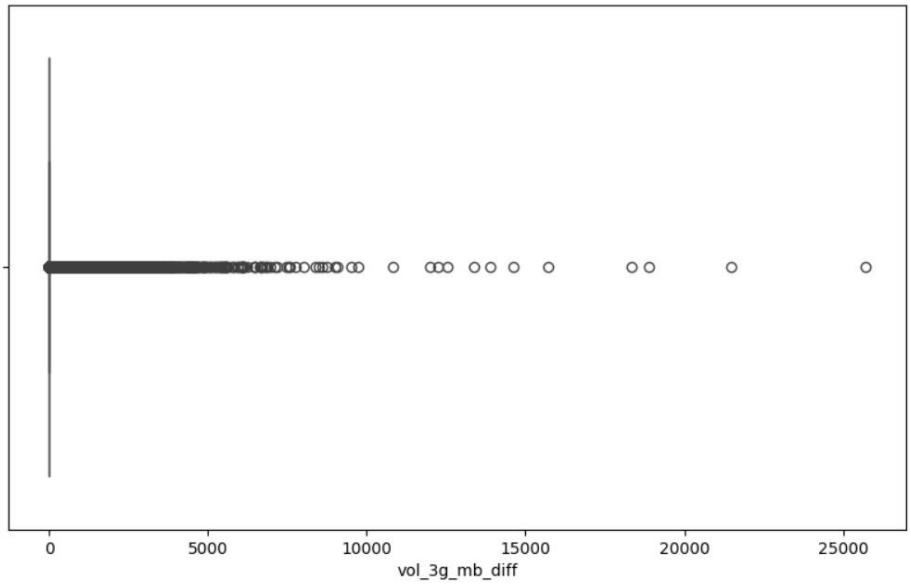
Screenshot from the notebook:

- ## Checking for outliers:

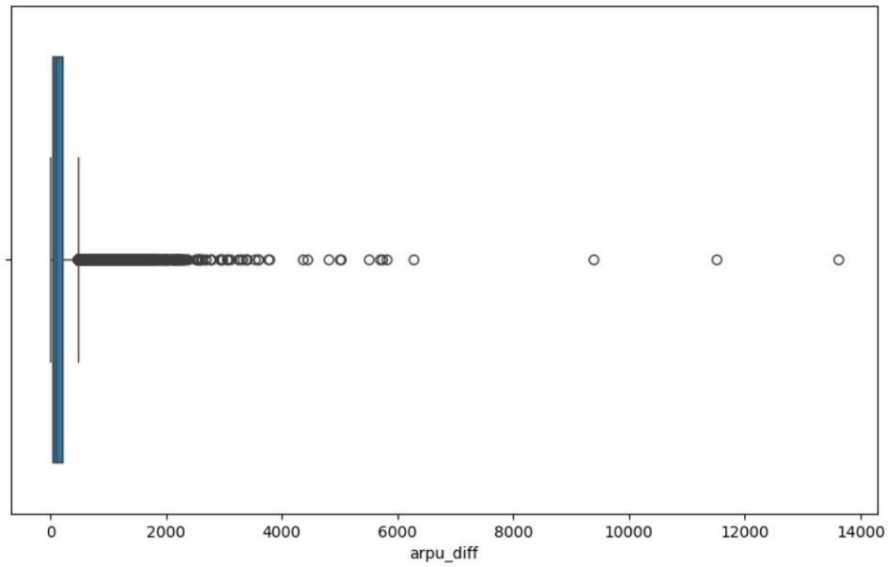
```
plt.figure(figsize=(10,6), dpi=100)
sns.boxplot(data=df3, x='aon')
plt.show()
```



```
plt.figure(figsize=(10,6), dpi=100)
sns.boxplot(data=df3, x='vol_3g_mb_diff')
plt.show()
```

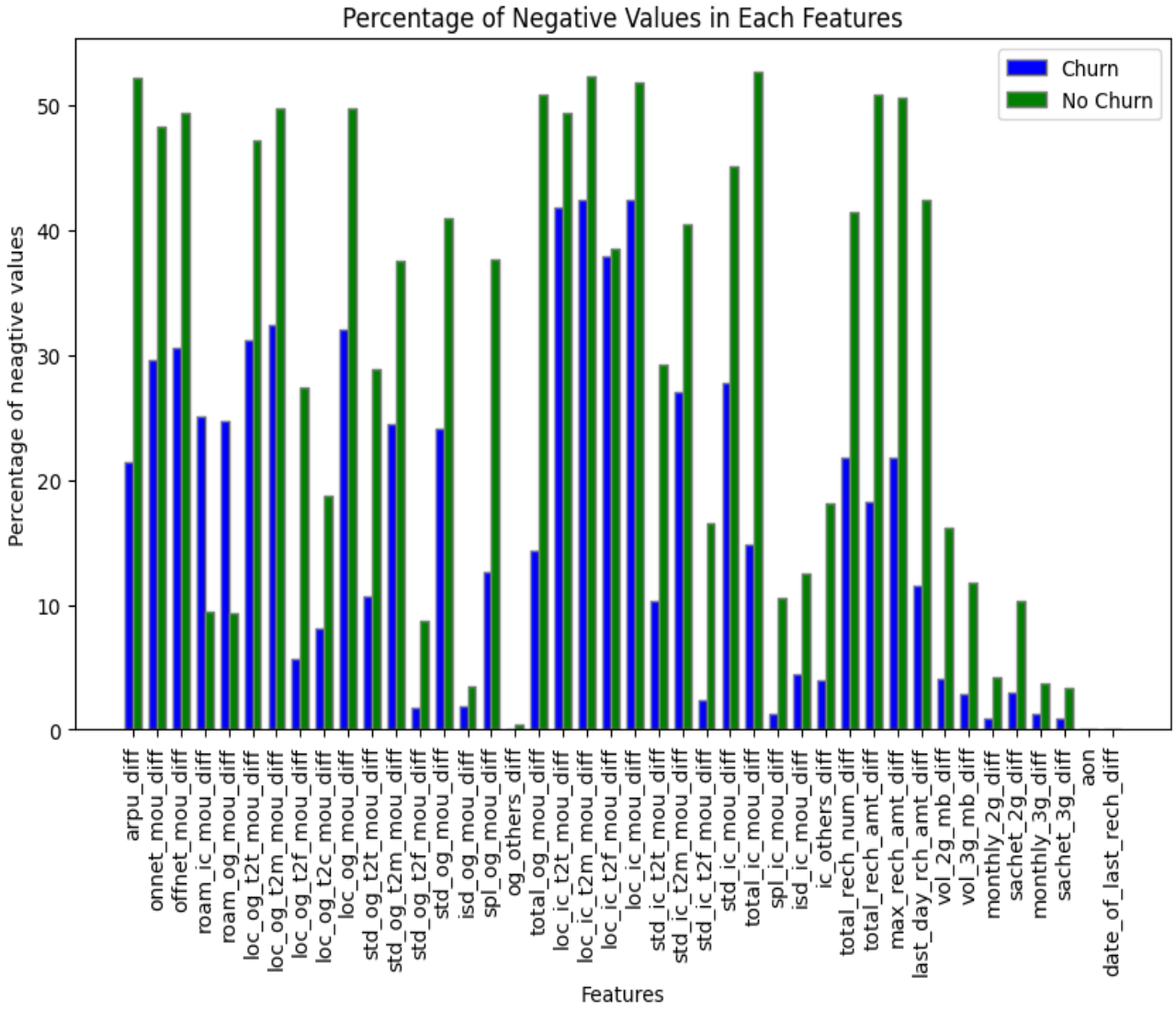



```
plt.figure(figsize=(10,6), dpi=100)
sns.boxplot(data=df3, x='arpu_diff')
plt.show()
```



From the above graphs, we can see that there are outliers in the variables we have taken samples such as vol_3g_mb_diff, aon and, average revenue per user (arpu_diff), which would be handled through transformation in further steps.

Percentage of Negative Values in Each Features:



We can see that the non-churn users has more number of negative value than churn users. So, the feature engineering what we done is correct.

Scaling of the Data:

Standard scaling is a valuable preprocessing step in telecom churn prediction as it improves model stability, convergence, and interpretability, leading to more reliable and accurate predictions.

Scaling

```
:  1 for col in df4.columns:
   2     if col != 'churn_probability':
   3         sc = StandardScaler()
   4         df4[[col]] = sc.fit_transform(df4[[col]])
```

Transformation Technique:

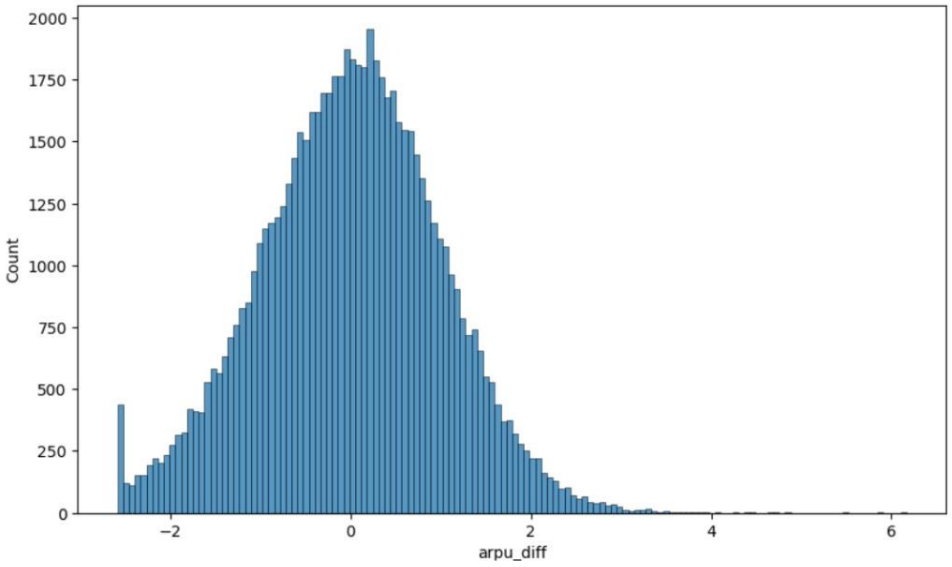
- Power transformations are valuable preprocessing steps in telecom churn prediction as they can improve the performance and interpretability of machine learning models by addressing issues related to skewed distributions, heteroscedasticity, non-linearity, and outliers in the data.
- We have used Yeo-Johnson method to transform the variables in which outliers are present. Since Yeo-Johnson transformation can handle both positive and negative values we have used it.

Transforming

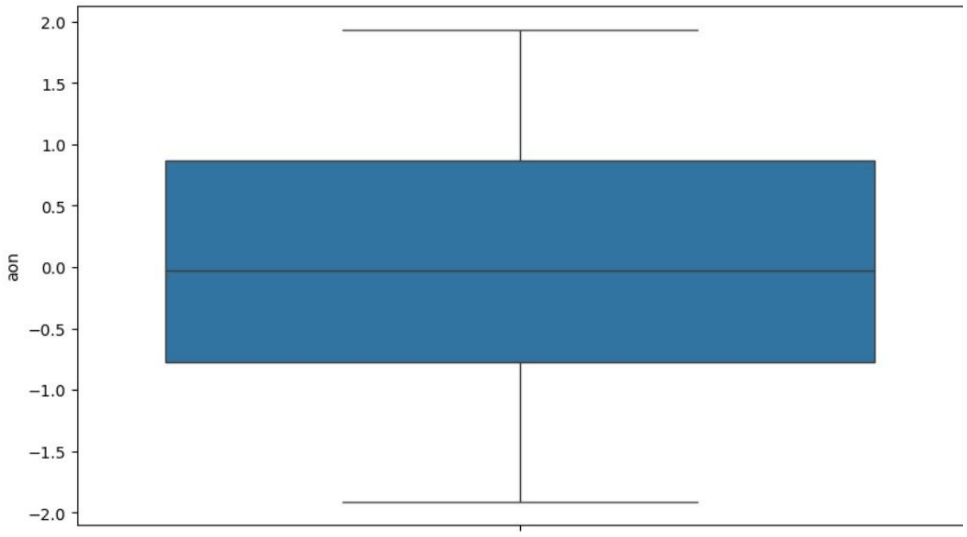
```
1 for col in df4.columns:
2     if col != 'churn_probability':
3         transform = PowerTransformer()
4         df4[[col]] = transform.fit_transform(df4[[col]])
```

Outlier Treatment:

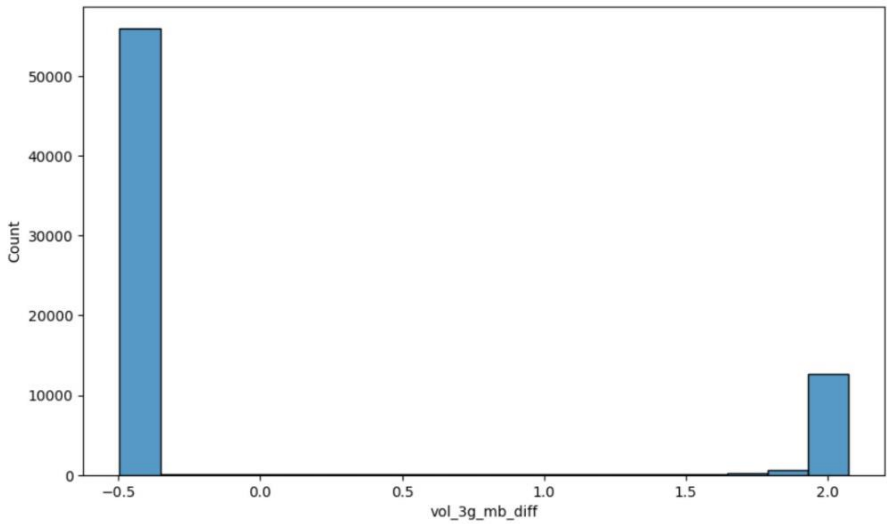
```
plt.figure(figsize=(10,6), dpi=100)
sns.histplot(df5['arpu_diff'])
plt.show()
```




```
plt.figure(figsize=(10,6), dpi=100)
sns.boxplot(df5['aon'])
plt.show()
```



```
plt.figure(figsize=(10,6), dpi=100)
sns.histplot(df5['vol_3g_mb_diff'])
plt.show()
```



From the above plots we can infer that, the outliers are treated, by applying the Yeo-Johnson transformation, the data is normalized and made more symmetric for columns such as difference of average revenue per user(arpu_diff) and age on network(aon) which can improve the performance. Whereas, for difference of mobile internet usage volume in MB(vol_3g_mb_diff) the outliers are not fully treated, but the skewness of the distribution is reduced.

MODEL BUILDING:

Model Building Before Oversampling:

- Building models before SMOTE (Synthetic Minority Over-sampling Technique) can help in understanding the performance of the model without oversampling the minority class.
- And also provides a baseline for model performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
- These metrics give an initial indication of how well the model performs in its raw state, without any adjustments for class imbalance.

```
append_to_metric_df("DecisionTreeClassifier_BOS",model_1_bos,
```

```
train data report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44007
           1       1.00      1.00      1.00      4992

    accuracy                           1.00     48999
   macro avg       1.00      1.00      1.00     48999
weighted avg       1.00      1.00      1.00     48999

test data report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     18860
           1       0.48      0.47      0.48      2140

    accuracy                           0.89     21000
   macro avg       0.71      0.71      0.71     21000
weighted avg       0.89      0.89      0.89     21000
```

```
append_to_metric_df("RandomForestClassifier_BOS",model_2_bos
```

```
train data report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44007
           1       1.00      1.00      1.00      4992

    accuracy                           1.00     48999
   macro avg       1.00      1.00      1.00     48999
weighted avg       1.00      1.00      1.00     48999

test data report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96     18860
           1       0.77      0.48      0.59      2140

    accuracy                           0.93     21000
   macro avg       0.86      0.73      0.78     21000
weighted avg       0.93      0.93      0.93     21000
```

Models such as Decision Tree classifier and Random Forest are built and shows overfitting.

SMOTE:(Synthetic Minority Over-sampling Technique)

- This methos is typically applied to address class imbalance by generating synthetic samples for the minority class.
- Generating synthetic samples using SMOTE can significantly increase the size of the dataset, especially for highly imbalanced datasets with a large minority class.
- Applying SMOTE to the entire dataset may lead to memory and computational constraints.
- Limiting SMOTE to a portion of the data helps manage computational resources more efficiently. Here we are limiting it to 40%.

```
from imblearn.over_sampling import SMOTE
```

```
sm = SMOTE(sampling_strategy=0.4, random_state=10)
```

```
x_sm, y_sm  = sm.fit_resample(x, y)
```

```
y_sm.value_counts()
```

```
0    62867
1    25146
Name: churn_probability, dtype: int64
```

Model Building after Over Sampling:

Building models before and after applying SMOTE allows for a direct comparison of performance metrics. This comparison helps in evaluating the effectiveness of SMOTE in addressing class imbalance and improving model performance.

Logistic Regression:

We are using Logistics Regression here, since this is a Binary classification Problem and due to the high explainable of the model. It is easy to understand and interpret the prediction made by the model. It is also faster and easier to train models for large datasets than complex algorithms.

```
append_to_metric_df("LogisticRegression", model_1, xtrain, xtest, ytrain, ytest)
train data report:
              precision    recall  f1-score   support

           0       0.81      0.97      0.88     44007
           1       0.84      0.43      0.57     17602

    accuracy                           0.81     61609
   macro avg       0.82      0.70      0.72     61609
weighted avg       0.82      0.81      0.79     61609

test data report:
              precision    recall  f1-score   support

           0       0.81      0.97      0.88     18860
           1       0.85      0.44      0.58      7544

    accuracy                           0.82     26404
   macro avg       0.83      0.70      0.73     26404
weighted avg       0.82      0.82      0.80     26404
```

- In churn prediction, the cost associated with misclassifying a churned customer as non-churned (false negative) is typically higher than misclassifying a non-churned customer as churned (false positive).
- When a churned customer is incorrectly identified as non-churned, the telecom company may lose that customer's business, resulting in revenue loss.
- On the other hand, incorrectly identifying a non-churned customer as churned may lead to some inconvenience for the customer (e.g., receiving retention offers), but it does not result in direct revenue loss.
- We need not focus on the false negative rate since the model will predict that those who will abandon the cart as they will buy the product.
- This will make the company miss those who will abandon their cart for any targeted marketing.
- So, we should focus on Sensitivity (Recall score) because that indicates that our customer who are churned.
- The accuracy of the model is 0.82 which means 82% of the data are correctly predicted.
- The training and test data both show the same level of accuracy meaning that the model is

underfit.

- This may be due to a lot of reasons like bias in the data, the need for more and better predictor variables or the model may not be able to learn the complex patterns.
- Our Focus metrics – Recall score is very low for this model (0.44). We try to increase the recall score in further models.

Decision Tree:

This Algorithm builds a tree-like model by recursively splitting the dataset based on the most significant features, leading to a set of decision rules. The final leaves of the tree represent the predicted classes or value

```
append_to_metric_df("DecisionTreeClassifier", model_2, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44007
           1       1.00      1.00      1.00     17602

    accuracy                           1.00     61609
   macro avg       1.00      1.00      1.00     61609
weighted avg       1.00      1.00      1.00     61609

test data report:
              precision    recall  f1-score   support

           0       0.91      0.90      0.90     18860
           1       0.76      0.77      0.76      7544

    accuracy                           0.86     26404
   macro avg       0.83      0.84      0.83     26404
weighted avg       0.87      0.86      0.86     26404
```

Ensemble Model – Random Forest:

```
append_to_metric_df("RandomForestClassifier", model_3, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44007
           1       1.00      1.00      1.00     17602

    accuracy                           1.00     61609
   macro avg       1.00      1.00      1.00     61609
weighted avg       1.00      1.00      1.00     61609

test data report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.94     18860
           1       0.91      0.80      0.85      7544

    accuracy                           0.92     26404
   macro avg       0.92      0.88      0.90     26404
weighted avg       0.92      0.92      0.92     26404
```

Ensemble Model – AdaBoost Classifier:

```
1 append_to_metric_df("AdaBoostClassifier", model_4, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       0.89      0.92      0.90     44007
           1       0.79      0.70      0.74     17602

    accuracy                           0.86     61609
   macro avg       0.84      0.81      0.82     61609
weighted avg       0.86      0.86      0.86     61609

test data report:
              precision    recall  f1-score   support

           0       0.88      0.92      0.90     18860
           1       0.79      0.69      0.74      7544

    accuracy                           0.86     26404
   macro avg       0.83      0.81      0.82     26404
weighted avg       0.86      0.86      0.86     26404
```

Ensemble Model – Gradient Boost Classifier:

```
1 append_to_metric_df("GradientBoostingClassifier", model_5, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       0.90      0.95      0.93     44007
           1       0.86      0.75      0.80     17602

    accuracy                           0.89     61609
   macro avg       0.88      0.85      0.86     61609
weighted avg       0.89      0.89      0.89     61609

test data report:
              precision    recall  f1-score   support

           0       0.90      0.95      0.93     18860
           1       0.86      0.74      0.80      7544

    accuracy                           0.89     26404
   macro avg       0.88      0.85      0.86     26404
weighted avg       0.89      0.89      0.89     26404
```

Ensemble Model – XG Boost Classifier:

```
1 append_to_metric_df("XGBClassifier", model_6, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     44007
           1       0.94      0.91      0.93     17602

    accuracy                           0.96     61609
   macro avg       0.95      0.95      0.95     61609
weighted avg       0.96      0.96      0.96     61609

test data report:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95     18860
           1       0.90      0.85      0.87      7544

    accuracy                           0.93     26404
   macro avg       0.92      0.91      0.91     26404
weighted avg       0.93      0.93      0.93     26404
```

KNNeighbors Classifier:

- KNNeighbors classification is a supervised machine learning algorithm used for classification tasks.
- It works by assigning a data point to the majority class among its k-nearest neighbors, determined based on a predefined distance metric.
- The algorithm is simple yet effective, making decisions based on the proximity of data points in the feature space.

```
1 append_to_metric_df("KNN", model_7, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       0.98      0.86      0.92     44007
           1       0.74      0.95      0.83     17602

    accuracy                           0.89     61609
   macro avg       0.86      0.91      0.87     61609
weighted avg       0.91      0.89      0.89     61609

test data report:
              precision    recall  f1-score   support

           0       0.96      0.82      0.88     18860
           1       0.66      0.90      0.76      7544

    accuracy                           0.84     26404
   macro avg       0.81      0.86      0.82     26404
weighted avg       0.87      0.84      0.85     26404
```

Naïve Bayes:

- The "naive" in Naive Bayes comes from the assumption of feature independence.
- The main assumption of Naive Bayes is that all features used to describe an observation are independent of each other given the class label.
- Our Variables are independent of each other based on the correlation coefficients.
- Hence, we use Bernoulli Naïve Bayes model because it is well-suited for handling binary features as it models each feature as a binary random variable following a Bernoulli distribution.

```
1 append_to_metric_df("BernoulliNB", model_8, xtrain, xtest, ytrain, ytest)
```

```
train data report:
              precision    recall  f1-score   support

           0       0.85      0.73      0.79     44007
           1       0.50      0.68      0.58     17602

    accuracy                           0.72     61609
   macro avg       0.68      0.71      0.68     61609
weighted avg       0.75      0.72      0.73     61609

test data report:
              precision    recall  f1-score   support

           0       0.85      0.73      0.79     18860
           1       0.51      0.68      0.58      7544

    accuracy                           0.72     26404
   macro avg       0.68      0.71      0.68     26404
weighted avg       0.75      0.72      0.73     26404
```

Overall Model Metrics:

	Model	Sensitivity_train	Specificity_train	Accuracy_train	F1-Score_train	Precission_train	ROC_AUC_Score_train
0	Base Model LogisticRegression	0.445312	0.985843	0.930386	0.565865	0.775916	0.898532
1	Base_model_DecisionTreeClassifier	1.000000	0.946978	1.000000	1.000000	1.000000	1.000000
2	Base_model_RandomForestClassifier	1.000000	0.981018	1.000000	1.000000	1.000000	1.000000
3	Base_model_KNN	0.490585	0.974867	0.934305	0.603425	0.783680	0.961244
4	DecisionTreeClassifier_BOS	1.000000	0.941729	1.000000	1.000000	1.000000	1.000000
5	RandomForestClassifier_BOS	0.999800	0.983245	0.999980	0.999900	1.000000	1.000000
6	LogisticRegression	0.425520	0.968876	0.812852	0.565070	0.840817	0.801694
7	DecisionTreeClassifier	1.000000	0.900530	1.000000	1.000000	1.000000	1.000000
8	RandomForestClassifier	1.000000	0.967762	1.000000	1.000000	1.000000	1.000000
9	AdaBoostClassifier	0.700432	0.924019	0.860361	0.741349	0.787343	0.911106
10	GradientBoostingClassifier	0.746506	0.953552	0.893068	0.799562	0.860736	0.946700
11	XGBClassifier	0.913760	0.963680	0.959486	0.927994	0.942680	0.991994
12	KNN	0.953130	0.815323	0.889383	0.831182	0.736900	0.976945
13	BernoulliNB	0.677253	0.734730	0.717054	0.577652	0.503591	0.767159

	Model	Sensitivity_test	Specificity_test	Accuracy_test	F1-Score_test	Precission_test	ROC_AUC_Score_test
0	Base Model LogisticRegression	0.445794	0.985843	0.930810	0.567688	0.781327	0.901069
1	Base_model_DecisionTreeClassifier	0.576168	0.946978	0.909190	0.563915	0.552172	0.761573
2	Base_model_RandomForestClassifier	0.608411	0.981018	0.943048	0.685263	0.784337	0.936853
3	Base_model_KNN	0.375701	0.974867	0.913810	0.470451	0.629108	0.825336
4	DecisionTreeClassifier_BOS	0.471963	0.941729	0.893857	0.475406	0.478900	0.706846
5	RandomForestClassifier_BOS	0.484579	0.983245	0.932429	0.593759	0.766445	0.909612
6	LogisticRegression	0.435445	0.968876	0.816467	0.575508	0.848399	0.808525
7	DecisionTreeClassifier	0.773065	0.900530	0.864111	0.764752	0.756617	0.836797
8	RandomForestClassifier	0.796129	0.967762	0.918724	0.848425	0.908074	0.971820
9	AdaBoostClassifier	0.694459	0.924019	0.858431	0.737057	0.785222	0.906607
10	GradientBoostingClassifier	0.742444	0.953552	0.893236	0.798944	0.864752	0.942210
11	XGBClassifier	0.848356	0.963680	0.930730	0.874974	0.903317	0.973484
12	KNN	0.904295	0.815323	0.840744	0.764413	0.662009	0.928982
13	BernoulliNB	0.679215	0.734730	0.718868	0.579933	0.505974	0.768105

- From the above metrics we can see that some models are overfitting, and some model performs really well. But for our business we need a model which should performs well in finding the churn users.
- It is ok if we have predicted a non-churn customer as churn, but we should not miss any churn customer, because in Telecom industry it is very hard to get a new customer rather than retaining the old customer.

- So, in this case we should look for a model which has high sensitivity, and can less focus on having low specificity and precision.
- In this case KNN model performs well, its sensitivity score is high when compared to other models.

Business Interpretation:

- From the data we found that the users who churn are reducing their usage of the service every month where the users who didn't churn are using the services consistently or they increasing their usage.
- So, we did some feature engineering on our data. We found the difference between the 3 months values and made it as a single feature, So that the churn users will have huge difference value and no churn user will have less difference value or negative value.
- Apart from this we found that the usage of roaming is very high for churn users when compared to no churn users. So, we can speculate that some of the users are churning because they use to travel a lot and our service may not performing well in roaming or these users are relocated to different place and in that place our service may not performing well.
- To verify this we can check these customer locations and check for our networks performance in those locations and take necessary action accordingly.

Business / Model Justification:

- The main motive is to improve the performance of the model. As per the Business scenario, we have to predict the users who is going to churn, So that we can take some necessary steps to retain that user.
- As we discussed before in Telecom Industry getting a new customer is harder that retain the old customer, we should predict the maximum number of churn customer, its ok even if we miss classified some of the not churn customers as churn (which mean we can compromise for the precision score).
- So we should look for the model which gives high recall score for class 1 (Sensitivity).So recall for the class 1 is the metric we should look for among all the models we built, and good precision is also important.
- From the above Model metrics table we can see that KNN model and XGBoost performed well according to our requirements, Where KNN model performs very well in Sensitivity (90%) but performs in precision (66%) is not good. The XGBoost model performs moderately in Sensitivity (85%) but the precision in good (90%).
- So, we decide to retain the customers by giving some offers, selecting XGBoost model will help us to reduce the wastage of cost on non-churn customers, or use KNN model accordingly.