CUSTOMER CHURN PREDICTION IN TELECOM INDUSTRY

DATA SCIENCE

IBM Preshnave | Senior Software Technical Trainer | IBM

Overview: The study and forecast of customer churn in the telecom industry is a problem nowadays since it's crucial for the sector to examine customer behavior to identify those who are going to cancel their subscriptions. In today's commercial condition gaining a new customer's cost is more than retaining the existing customers. It is therefore possible to forecast whether or not customers would leave a company by gathering knowledge from the telecom industry. The telecom sectors must take the necessary steps to start acquiring their related clients in order to prevent their market value from stagnating.

The master class aims to divide up the client base into different groups and identify the churn-causing elements in each group. This study also attempts to develop a churn prediction model and use it to identify customers who are likely to leave. The analysis of Churn is mainly depends on Poor connectivity, Price/offers and Customer service.

Configuration Details:

1.	**Problem**	Customer Churn Prediction Analysis in Telecom Industry
2.	**Solver**	Data Analysis – Pandas | Numpy | Matplotlib | Seaborn
3.	**Compiler**	Jupyter Notebook
4.	**Variables**	Churn | Account Weeks | Contact Renewal | Data Plan | Data Usage | Customer Services Calls | Day Mins | Day Calls | Monthly Charge | Overage Fee | Roam Mins.

Data Set Information:

This is a sample data for the dataset. The sample dataset contains the consumers who have left the telecom firm, and if we know the difference between it and the population dataset, we will be able to determine that the sample is chosen at random from the population. The dataset link as follows;

/Kaggle/input/telecom-churn/telecom_churn.csv

Process of Data Analysis:

Fig.1 – Data Analysis Process

Data Analysis:

Import required Libraries:

import pandas as pd
import numpy as np
import seaborn as cat

Load Data Files:

data = pd.read_csv("/kaggle/input/telecom-churn/telecom_churn.csv")

data

	Churn	AccountWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins
0	0	128	1	1	2.70	1	265.1	110	89.0	9.87	10.0
1	0	107	1	1	3.70	1	161.6	123	82.0	9.78	13.7
2	0	137	1	0	0.00	0	243.4	114	52.0	6.06	12.2
3	0	84	0	0	0.00	2	299.4	71	57.0	3.10	6.6
4	0	75	0	0	0.00	3	166.7	113	41.0	7.42	10.1
...	...	...	...	...	...	...	...	...	...	...	...
3328	0	192	1	1	2.67	2	156.2	77	71.7	10.78	9.9
3329	0	68	1	0	0.34	3	231.1	57	56.4	7.67	9.6
3330	0	28	1	0	0.00	2	180.8	109	56.0	14.44	14.1
3331	0	184	0	0	0.00	2	213.8	105	50.0	7.98	5.0
3332	0	74	1	1	3.70	0	234.4	113	100.0	13.30	13.7

3333 rows × 11 columns

Dataset Information:

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Churn            3333 non-null   int64  
 1   AccountWeeks     3333 non-null   int64  
 2   ContractRenewal  3333 non-null   int64  
 3   DataPlan         3333 non-null   int64  
 4   DataUsage        3333 non-null   float64
 5   CustServCalls    3333 non-null   int64  
 6   DayMins          3333 non-null   float64
 7   DayCalls         3333 non-null   int64  
 8   MonthlyCharge    3333 non-null   float64
 9   OverageFee       3333 non-null   float64
 10  RoamMins         3333 non-null   float64
dtypes: float64(5), int64(6)
memory usage: 286.6 KB
```

Data Variable Information:

data.columns

```
Index(['Churn', 'AccountWeeks', 'ContractRenewal', 'DataPlan', 'DataUsage',
       'CustServCalls', 'DayMins', 'DayCalls', 'MonthlyCharge', 'OverageFee',
       'RoamMins'],
      dtype='object')
```

Describe the data – Mean, Median and Standard Deviation Information:

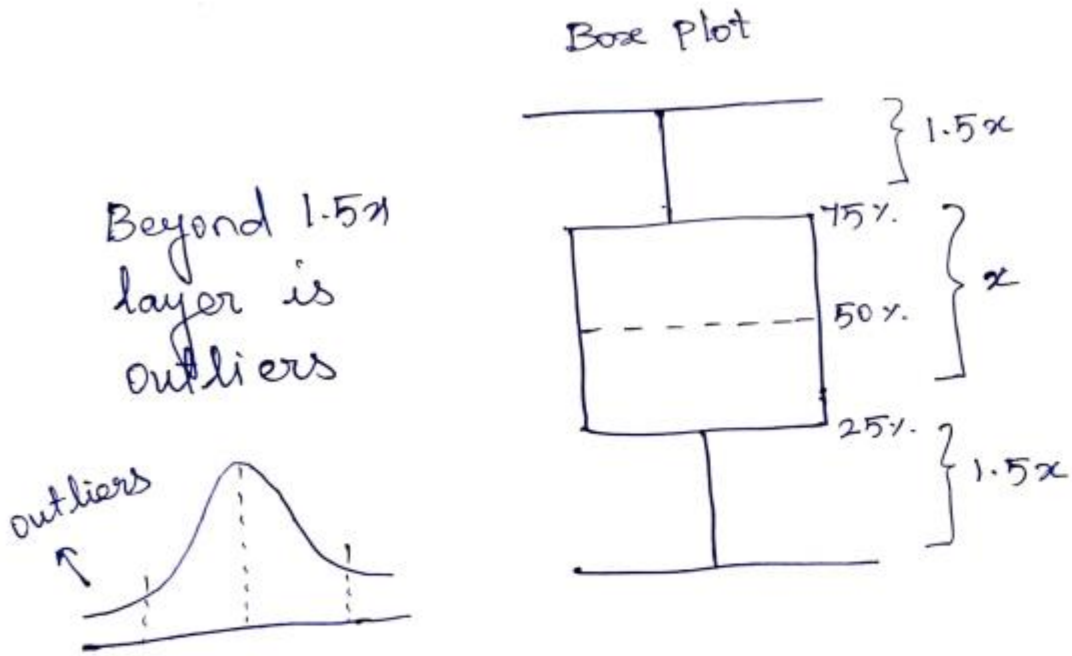
data.describe()

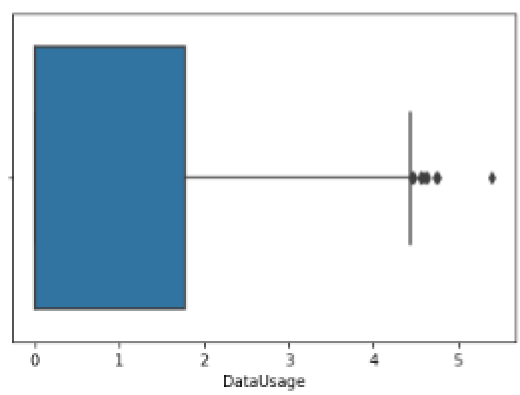
	Churn	AccountWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
mean	0.144914	101.064806	0.903090	0.276628	0.816475	1.562856	179.775098	100.435644	56.305161	10.051488	10.237294
std	0.352067	39.822106	0.295879	0.447398	1.272668	1.315491	54.467389	20.069084	16.426032	2.535712	2.791840
min	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	14.000000	0.000000	0.000000
25%	0.000000	74.000000	1.000000	0.000000	0.000000	1.000000	143.700000	87.000000	45.000000	8.330000	8.500000
50%	0.000000	101.000000	1.000000	0.000000	0.000000	1.000000	179.400000	101.000000	53.500000	10.070000	10.300000
75%	0.000000	127.000000	1.000000	1.000000	1.780000	2.000000	216.400000	114.000000	66.200000	11.770000	12.100000
max	1.000000	243.000000	1.000000	1.000000	5.400000	9.000000	350.800000	165.000000	111.300000	18.190000	20.000000

Describe the data:

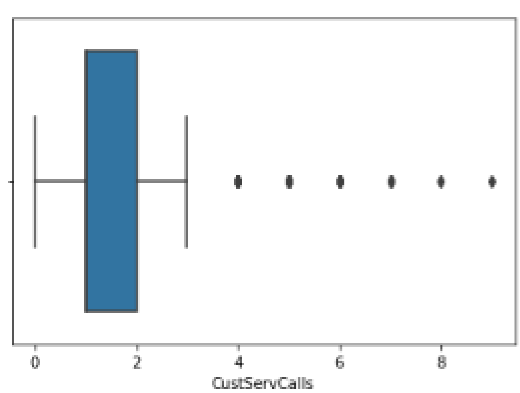
import seaborn as cat

cat.boxplot(data['DataUsage'])

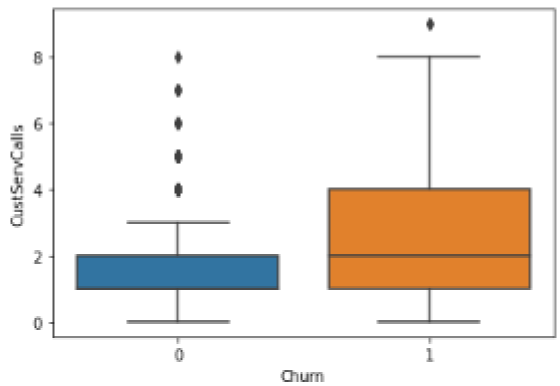




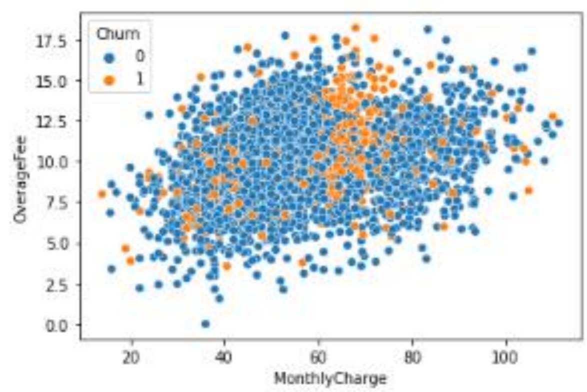
cat.boxplot(data['CustServCalls'])



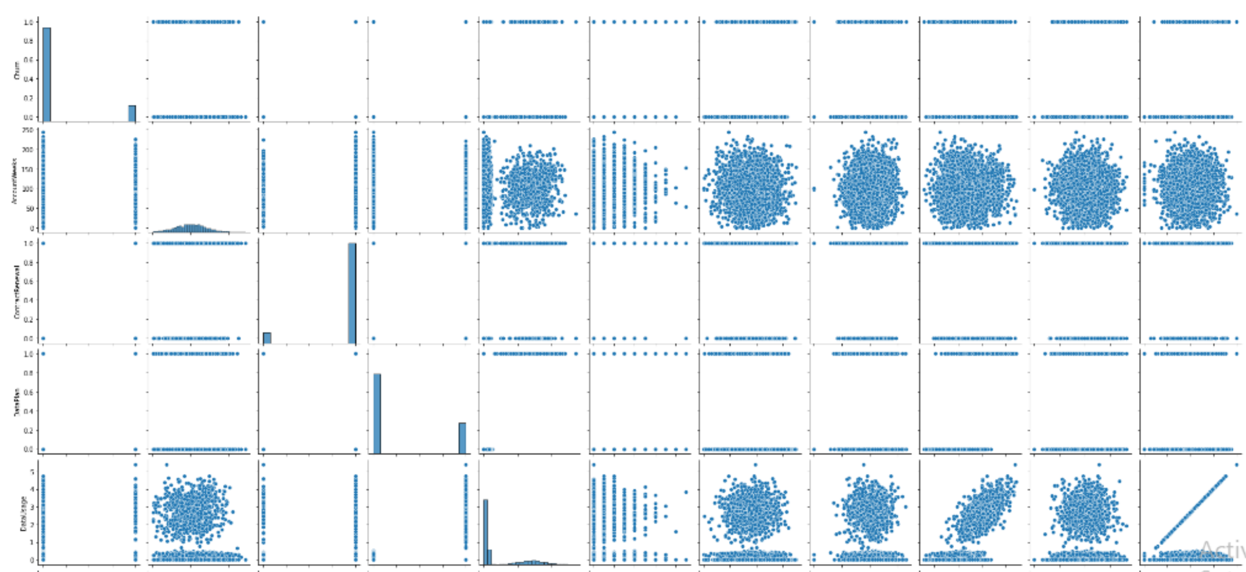
cat.boxplot(data['Churn'], data['CustServCalls'])

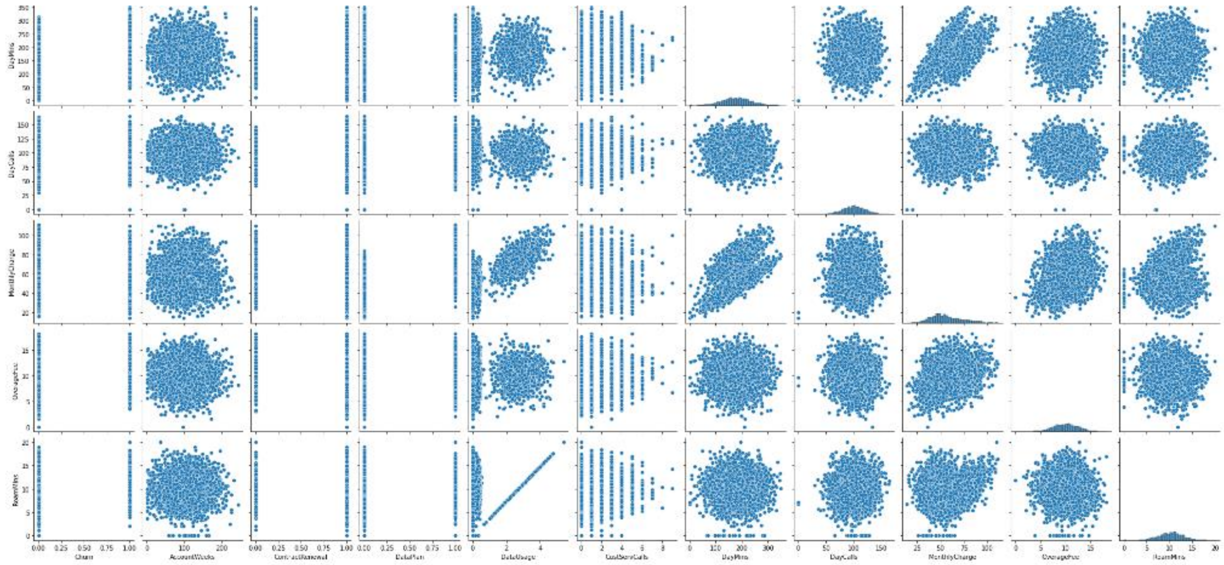


cat.scatterplot(data['MonthlyCharge'], data['OverageFee'], hue=data['Churn'])



cat.pairplot(data)





Data Correlation:

data.corr

	Churn	AccountWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins
Churn	1.000000	0.016541	-0.259852	-0.102148	-0.087195	0.208750	0.205151	0.018459	0.072313	0.092812	0.068239
AccountWeeks	0.016541	1.000000	-0.024735	0.002918	0.014391	-0.003796	0.006216	0.038470	0.012581	-0.006749	0.009514
ContractRenewal	-0.259852	-0.024735	1.000000	-0.006006	-0.019223	0.024522	-0.049396	-0.003755	-0.047291	-0.019105	-0.045871
DataPlan	-0.102148	0.002918	-0.006006	1.000000	0.945982	-0.017824	-0.001684	-0.011086	0.737490	0.021526	-0.001318
DataUsage	-0.087195	0.014391	-0.019223	0.945982	1.000000	-0.021723	0.003176	-0.007962	0.781660	0.019637	0.162746
CustServCalls	0.208750	-0.003796	0.024522	-0.017824	-0.021723	1.000000	-0.013423	-0.018942	-0.028017	-0.012964	-0.009640
DayMins	0.205151	0.006216	-0.049396	-0.001684	0.003176	-0.013423	1.000000	0.006750	0.567968	0.007038	-0.010155
DayCalls	0.018459	0.038470	-0.003755	-0.011086	-0.007962	-0.018942	0.006750	1.000000	-0.007963	-0.021449	0.021565
MonthlyCharge	0.072313	0.012581	-0.047291	0.737490	0.781660	-0.028017	0.567968	-0.007963	1.000000	0.281766	0.117433
OverageFee	0.092812	-0.006749	-0.019105	0.021526	0.019637	-0.012964	0.007038	-0.021449	0.281766	1.000000	-0.011023
RoamMins	0.068239	0.009514	-0.045871	-0.001318	0.162746	-0.009640	-0.010155	0.021565	0.117433	-0.011023	1.000000

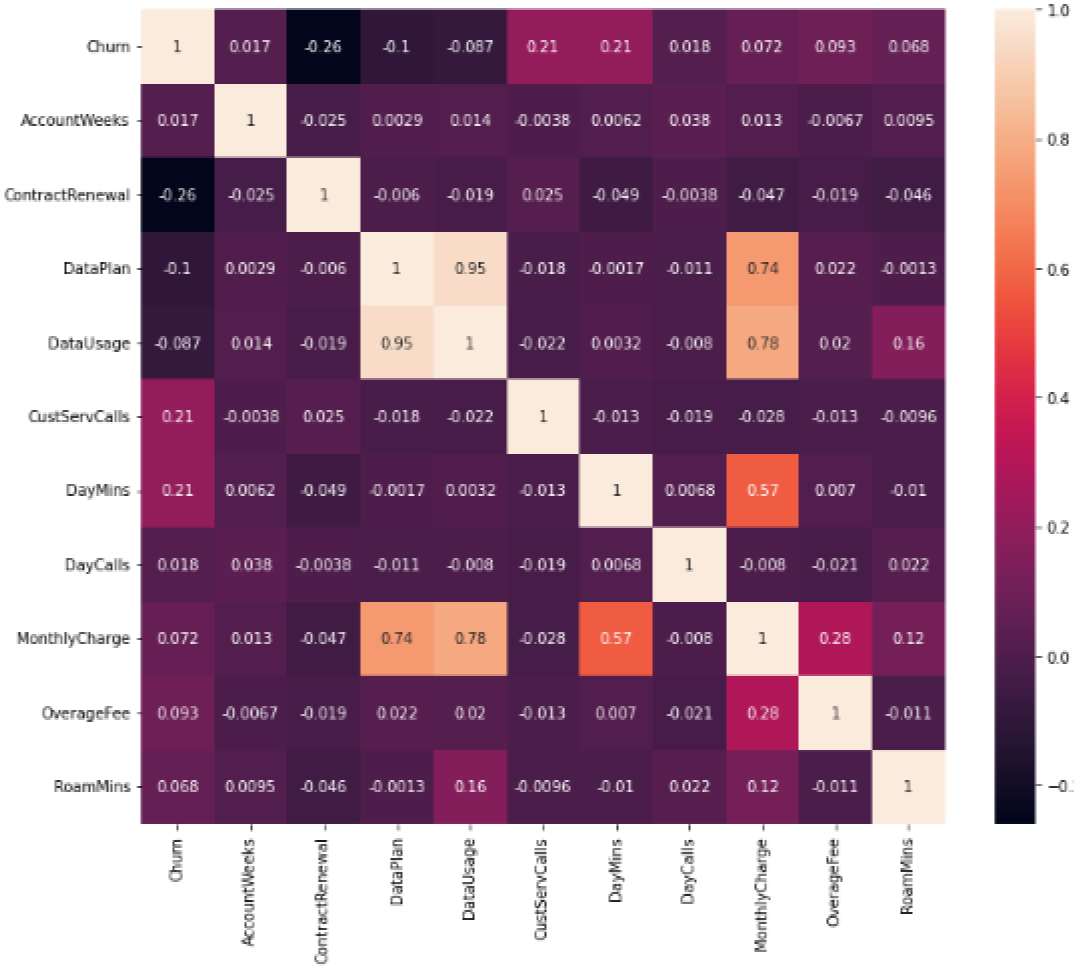
Data Analysis Heat Map:

import matplotlib.pyplot as plt

plt.figure(figsize=(12,10))

data['Churn']=data['Churn'].astype('int64')

cat.heatmap(data.corr(), annot=True)



Inferences:

- If customer service call increases, then the users of any particular network reduce. 75% of possibilities from customer service calls as shown from Data describe section.
- Customer reduction based on Roaming Charges
- Customer reduction based on Monthly charges for data plan, data usage and talk time rate and overage fee. 75% of customers pay more than average amount per month.
- When the churn is high then the charges are high.