1. INTRODUCTION

1.1 Overview

Customer churn has become highly important for companies because of increasing competition among companies, increased importance of marketing strategies and conscious behaviour of customers in recent years. Customers can easily trend toward alternative services. Companies must develop various strategies to prevent these possible trends, depending on the services they provide. During the estimation of possible churns, data from the previous churns might be used. An efficient churn predictive model benefits companies in many ways. Early identification of customers likely to leave may help to build cost effective ways in marketing strategies.

1.2 Purpose

Telecommunication industry always suffers from a very high churn rates when one industry offers a better plan than the previous there is a high possibility of the customer churning from the present due to a better plan in such a scenario it is very difficult to avoid losses but through prediction we can keep it to a minimal level. A machine learning model is built and this helps to identify the probable churn customers and then makes the necessary business decisions

2.LITERATURE SURVEY

2.1 Existing Problem

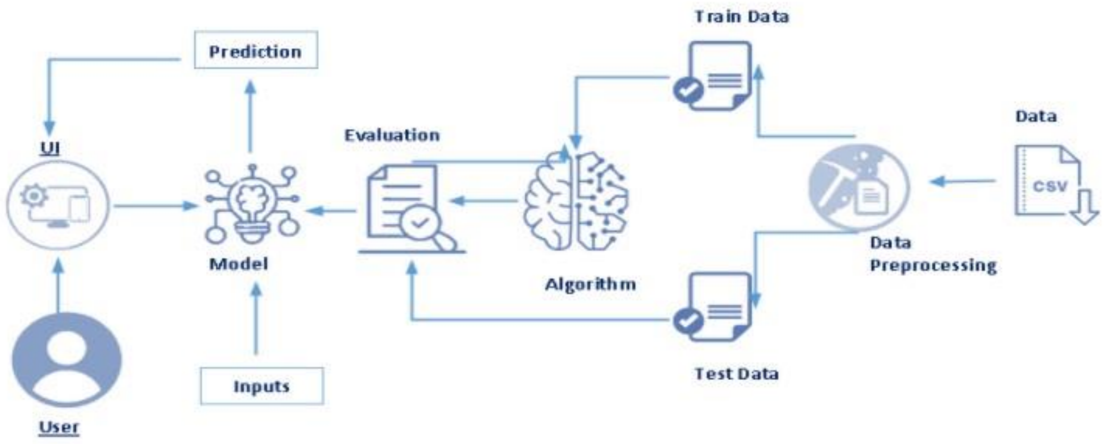
The retention and acquisition of users are the major concerns in telecom industry. The fast growth of marketplace in every business is giving rise to increased subscriber base. Accordingly, companies have recognized the significance of retaining the customers who is on hand. It has become necessary for service-providers to reduce the churn rate of customers since the inattention might negatively influence profitability of the company. Churn prediction contributes to identify those users who are likely to switch a company over another. Telecom is enduring the problem of ever-increasing churn rate. Accordingly, the current study employs machine learning algorithm on big-data platform. Machine learning algorithm techniques facilitate these telecom firms to be protected with efficient approaches for lessening the rate of churn. Silent churn is one type which is considered complicated to predict since there might have such kind of users who might probably churns in the near future. It must be the aim of the decision-maker and advertisers to lessen the churn ratio since it is a recognized fact that comparatively existing customers are the most beneficial resources for companies than acquiring new one.

2.2 Proposed Solution

The goal is to explore the petitions filed and their outcomes for the past six years i.e., from 2011 to 2016, and to find a pattern to predict the outcome by using a predictive model developed using Machine Learning techniques. In order to predict the case status of the applicants, we will be feeding the model with the dataset which contains the required fields by which the machine can predict the certification status of the visa applications.

3.THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

Hardware Requirements:

Processor : Intel Core I3

RAM : 4.00 GB

Operating system : Windows/Linux/MAC

Software Requirements:

Anaconda

Jupyter Notebook

Spyder
IBM Watson Studio.
IBM Watson Machine Learning
IBM Cloud Object Storage

• IBM Watson Studio

Watson Studio provides you with the environment and tools to solve your business problems by collaboratively working with data. It provides a suite of tools for data scientists, application developers and subject matter experts, allowing them to collaboratively connect to data, wrangle that data and use it to build, train and deploy models at scale. Successful AI projects require a combination of algorithms + data + team, and a very powerful compute infrastructure.

• IBM Watson Machine Learning

IBM Watson Machine Learning is a full-service IBM Cloud offering that makes it easy for developers and data scientists to work together to integrate predictive capabilities with their applications. The Machine Learning service is a set of REST APIs that you can call from any programming language to develop applications that make smarter decisions, solve tough problems, and improve user outcomes.

• IBM Cloud Object Storage

4 EXPERIMENTAL INVESTIGATIONS

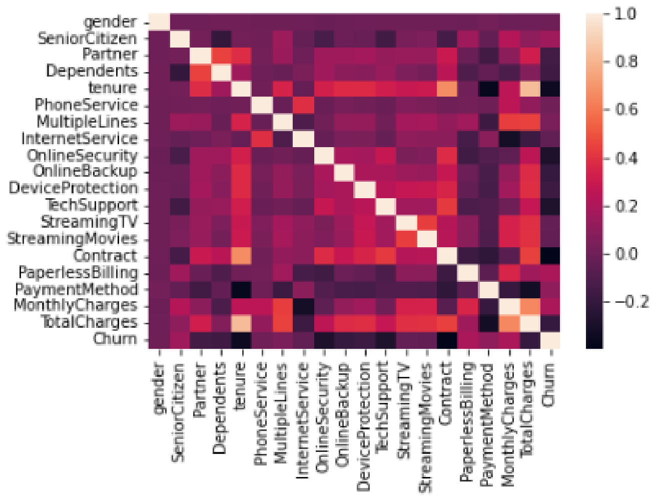
Dataset is downloaded from the Kaggle which has 9 features and 1 feature containing the class label. The total number of records available for us is more than 3 million points. The features provide the following information about our samples.

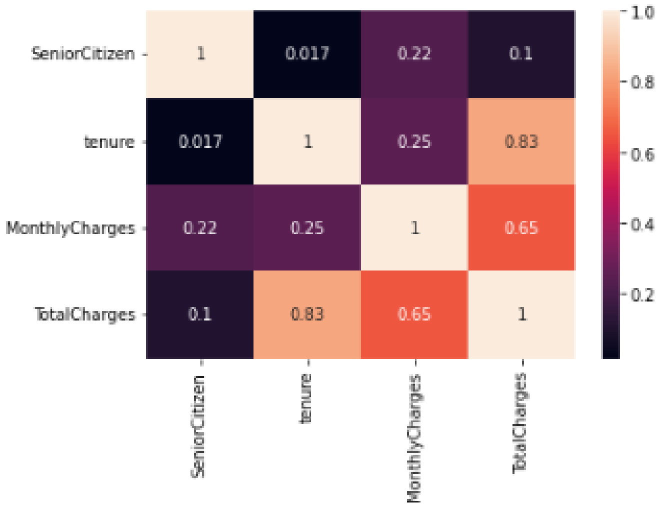
Our Dataset churn data contains following Columns

1. gender
2. SeniorCitizen
3. Partner
4. Dependents

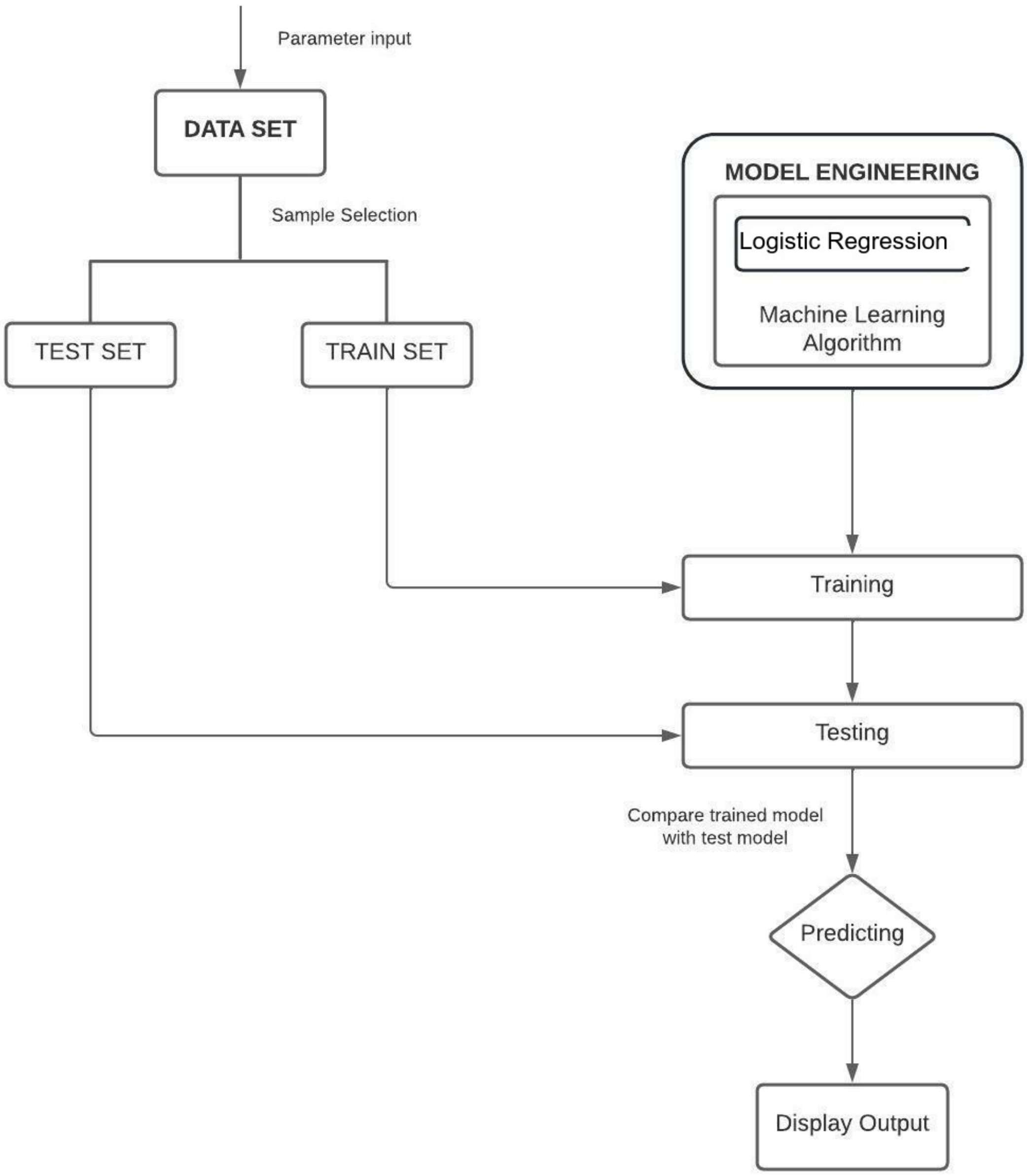
5. tenure
6. PhoneService
7. MultipleLines
8. InternetService
9. OnlineSecurity
10. OnlineBackup
11. DeviceProtection
12. TechSupport
13. StreamingTV
14. StreamingMovies
15. Contract
16. PaperlessBilling
17. PaymentMethod
18. MonthlyCharges
19. TotalCharges
20. Churn

The output column to be predicted is Churn .Based on the input variables we predict the Customer likely to tend or not.



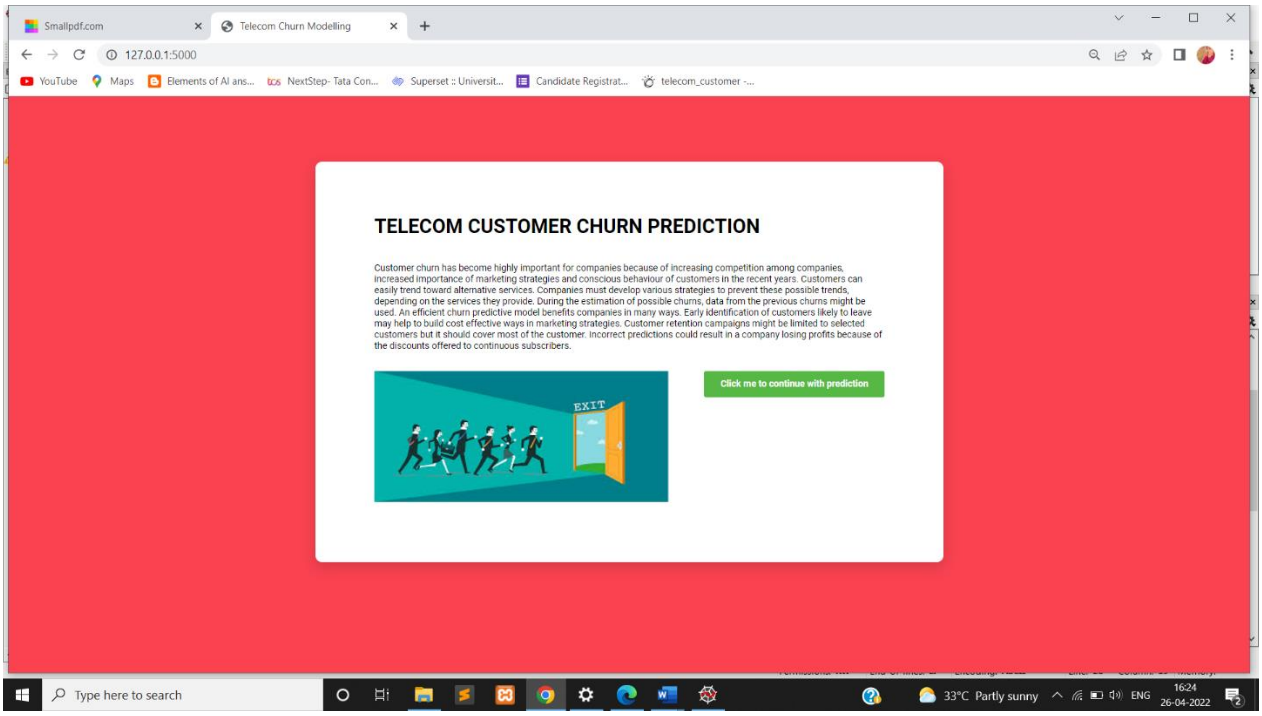

SeniorCitizen
tenure
MonthlyCharges
TotalCharges
1
0.017
0.22
0.1
0.017
1
0.25
0.83
0.22
0.25
1
0.65
0.1
0.83
0.65
1
SeniorCitizen
tenure
MonthlyCharges
TotalCharges
1.0
0.8
0.6
0.4
0.2

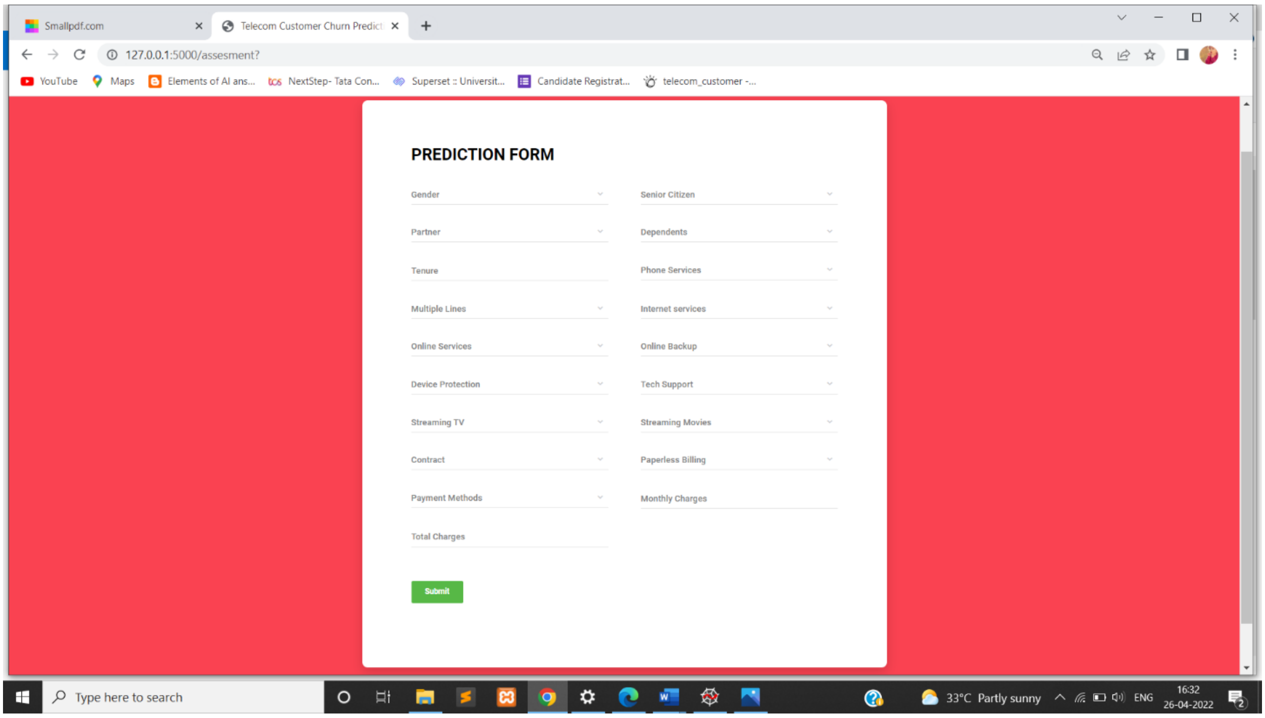
5 FLOWCHART

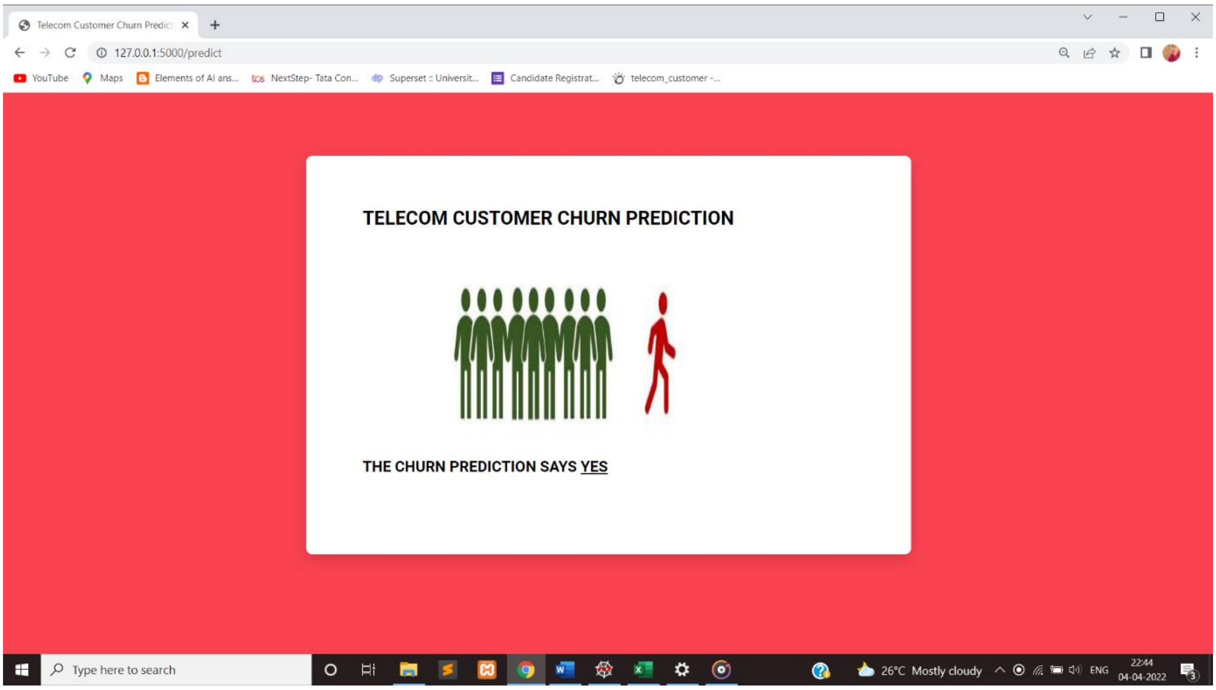

Parameter input
DATA SET
Sample Selection
TEST SET
TRAIN SET
MODEL ENGINEERING
Logistic Regression
Machine Learning Algorithm
Training
Testing
Compare trained model with test model
Predicting
Display Output

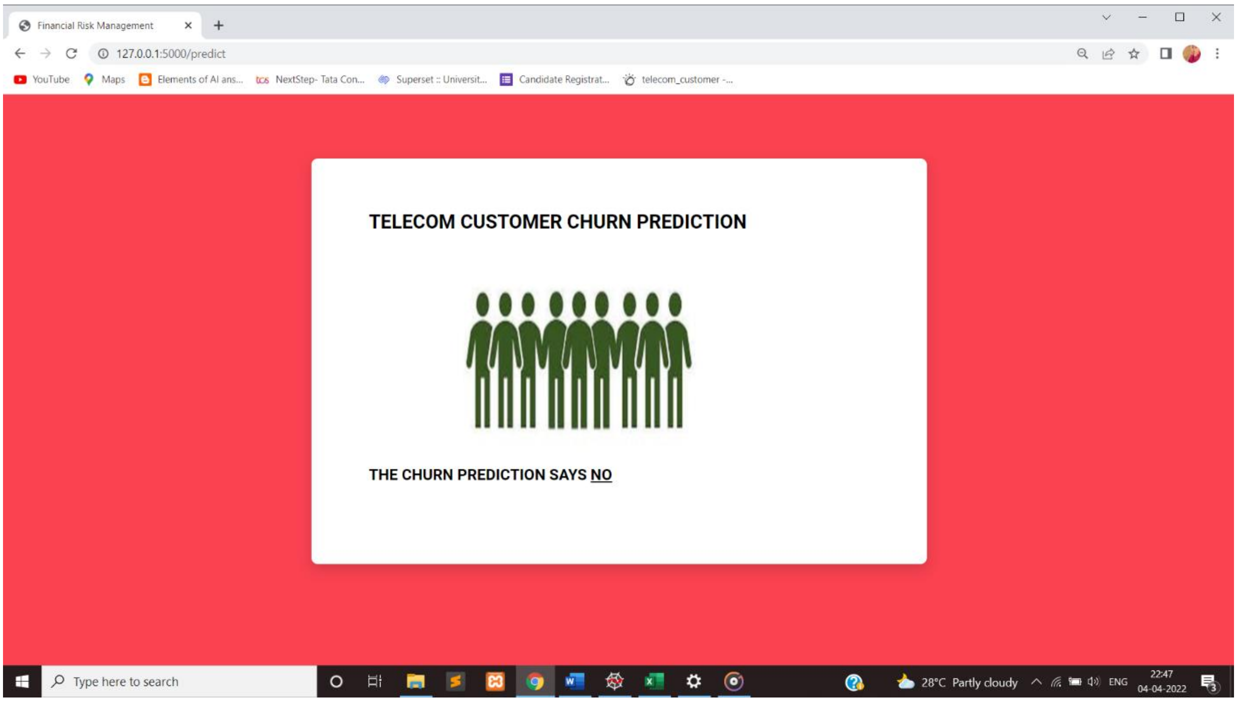
6 RESULT

The final result of the project is to predict the customer churn.






Telecom Customer Churn Predict
127.0.0.1:5000/predict
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
TELECOM CUSTOMER CHURN PREDICTION
THE CHURN PREDICTION SAYS YES
Type here to search
26°C Mostly cloudy
ENG
22:44
04-04-2022


Financial Risk Management
127.0.0.1:5000/predict
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
TELECOM CUSTOMER CHURN PREDICTION
THE CHURN PREDICTION SAYS NO
Type here to search
28°C Partly cloudy
ENG
22:47
04-04-2022

7. ADVANTAGES & DISADVANTAGES

With low switching costs and an abundance of alternative providers, **customer satisfaction** is the most effective means of reducing customer churn in telecom. And the most effective means of improving the customer experience is fully taking advantage of the vast streams of rich telecom customer data.

Disadvantages

1. The number of observations is decent, but if we could have more columns of features like the customers' geographic location, competitor's information, and other important factors, we could draw more insights from the result.

2. Since we have chosen our model not only depends on the complexity and predicting power but more importantly on the ease of interpretation, there are more powerful models outside of our range. For example, neural networks or extreme gradient boosting may perform much better and result in increased accuracy.

3. The nature of our dataset is a cross-sectional dataset. This means that there are no time series factors inside it. Since our goal is to predict churn rate, we have the option of contracts from monthly, one year to two years. It is best that we can find a time series dataset containing all the customer's information for up to two years to obtain better results for predicting and making decisions for the future market.

8. APPLICATIONS

Churn Prediction is essentially **predicting which clients are most likely to cancel a subscription i.e 'leave a company' based on their usage of the service**. From a company point of view, it is necessary to gain this information because acquiring new customers is often arduous and costlier than retaining old ones.

9.CONCLUSION

After successfully completing the project we learned about various machine learning models that were performed to predict the customer tend to an alternative service. The machine learning models included are logistic regression, k-nearest neighbor, random forest, and linear regression. Experiments show that the logistic regression model surpasses other models on accuracy . As for the future work, more experiments can be conducted on models to find out which model gives the best performance.

11. BIBILOGRAPHY

In [2]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
import seaborn as sns
import numpy as np

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_37a942429d3547b29a73ca52a76d33d7 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='MXCd7Y69IvWpc5_alh47jBSjvCIh7P9qmN7v9DiNDThf',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

body = client_37a942429d3547b29a73ca52a76d33d7.get_object(Bucket='telecomcustomerchurnprediction-donotdelete-pr-ihqq7q5brxbun5',
Key='DataSet.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

data = pd.read_csv(body)
data.head()
```

Out[2]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProte
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No

5 rows × 21 columns

In [3]:

```
data['Churn'].value_counts()  # Data is imbalanced
```

Out[3]:

```
No     5174
Yes    1869
Name: Churn, dtype: int64
```

In [4]:

```
data.drop(["customerID"], axis =1, inplace = True)
```

In [5]:

```
data.head()
```

Out[5]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtec
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No

In [6]:
```
data.describe()
```

Out[6]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   gender            7043 non-null   object 
 1   SeniorCitizen     7043 non-null   int64  
 2   Partner           7043 non-null   object 
 3   Dependents        7043 non-null   object 
 4   tenure            7043 non-null   int64  
 5   PhoneService      7043 non-null   object 
 6   MultipleLines     7043 non-null   object 
 7   InternetService   7043 non-null   object 
 8   OnlineSecurity    7043 non-null   object 
 9   OnlineBackup      7043 non-null   object 
 10  DeviceProtection  7043 non-null   object 
 11  TechSupport       7043 non-null   object 
 12  StreamingTV       7043 non-null   object 
 13  StreamingMovies   7043 non-null   object 
 14  Contract          7043 non-null   object 
 15  PaperlessBilling  7043 non-null   object 
 16  PaymentMethod     7043 non-null   object 
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   object 
 19  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```

In [8]:
```
data.TotalCharges = pd.to_numeric(data.TotalCharges, errors='coerce')
data.isnull().any()
```

Out[8]:
```
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      False
OnlineBackup        False
DeviceProtection    False
TechSupport         False
StreamingTV         False
StreamingMovies     False
Contract            False
PaperlessBilling    False
PaymentMethod       False
MonthlyCharges      False
TotalCharges         True
Churn               False
dtype: bool
```

In [9]:
```
data.isnull().sum()
```
Out[9]:
```
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       11
Churn               0
dtype: int64
```

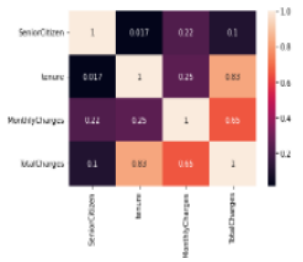
In [10]:
```
data["TotalCharges"].fillna(data["TotalCharges"].median() , inplace =True)
```

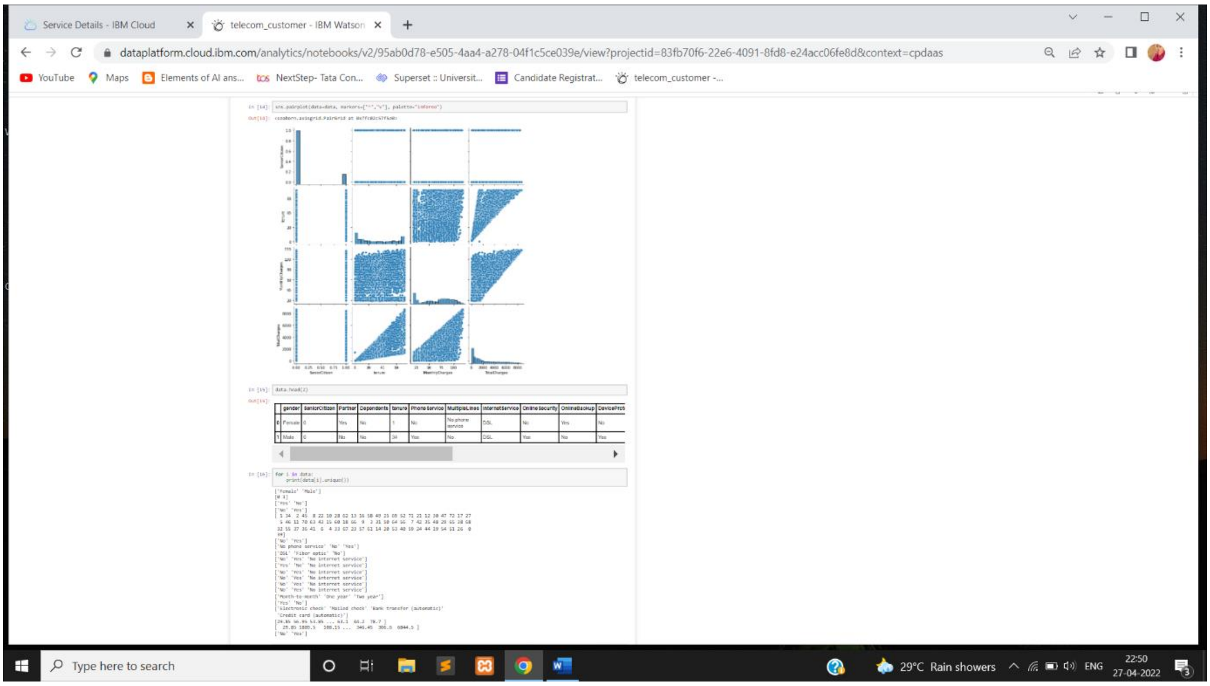
In [11]:
```
data.isnull().sum()
```
Out[11]:
```
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

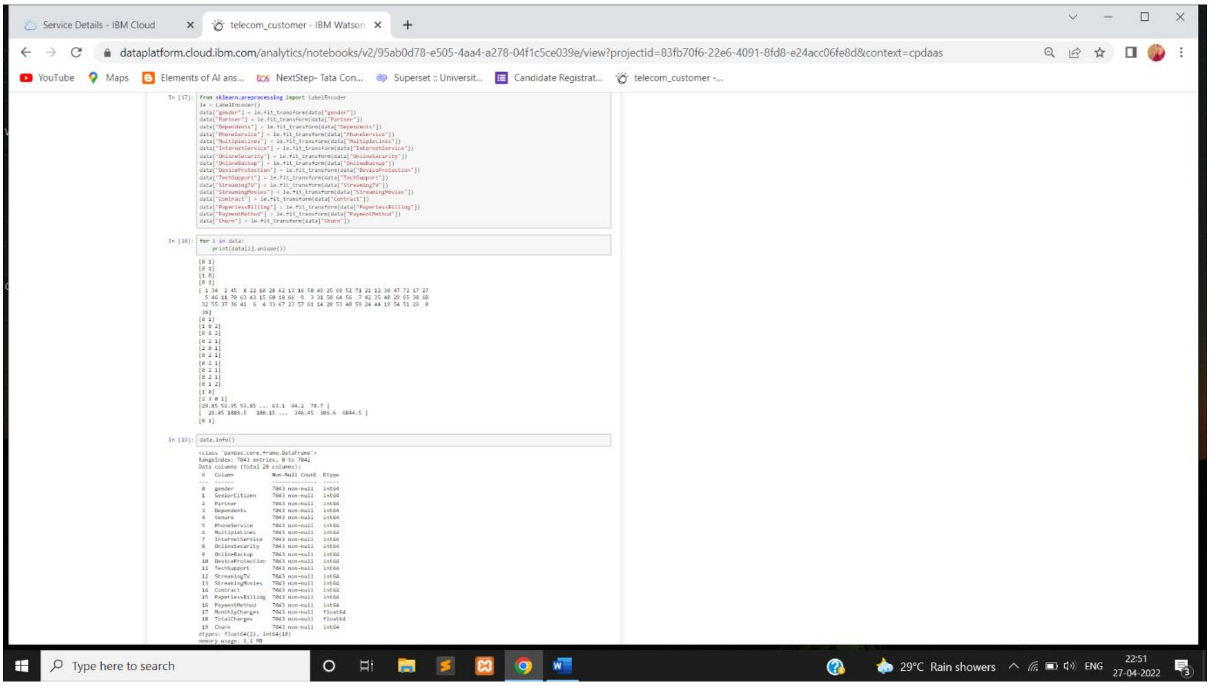
In [12]:
```
data.corr()
```
Out[12]:

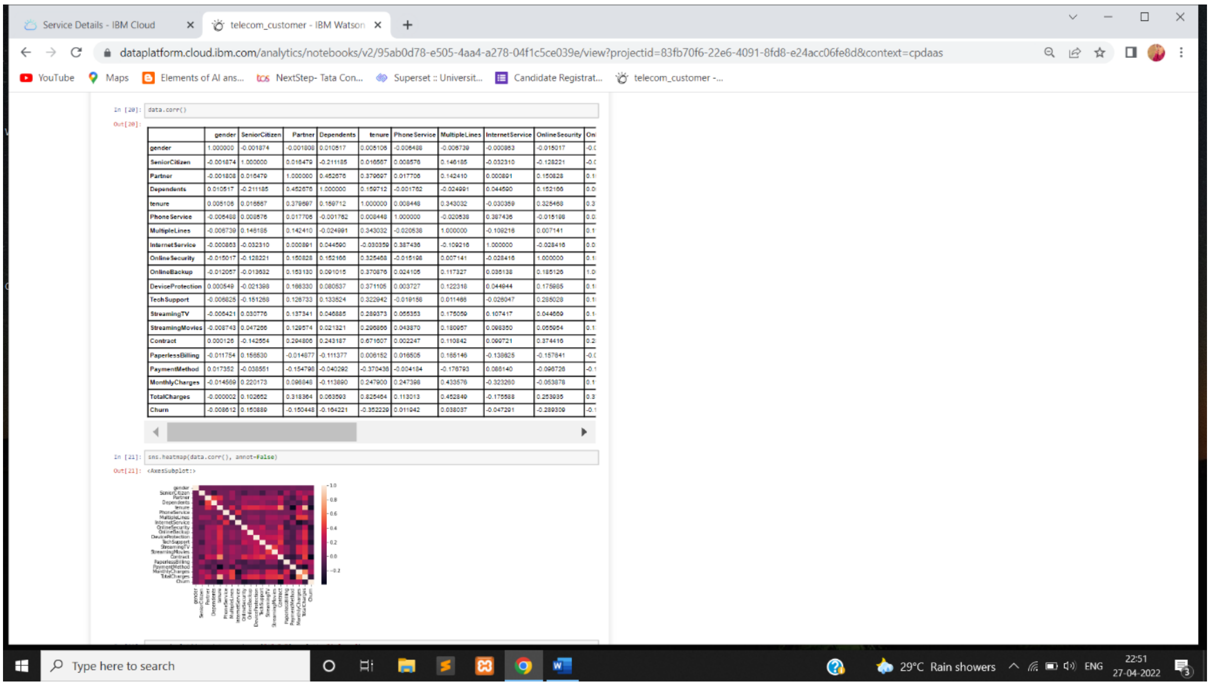
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
SeniorCitizen	1.000000	0.016567	0.220173	0.102652
tenure	0.016567	1.000000	0.247900	0.825464
MonthlyCharges	0.220173	0.247900	1.000000	0.650864
TotalCharges	0.102652	0.825464	0.650864	1.000000

In [13]:
```
sns.heatmap(data.corr(), annot=True)
```
Out[13]: <AxesSubplot:>

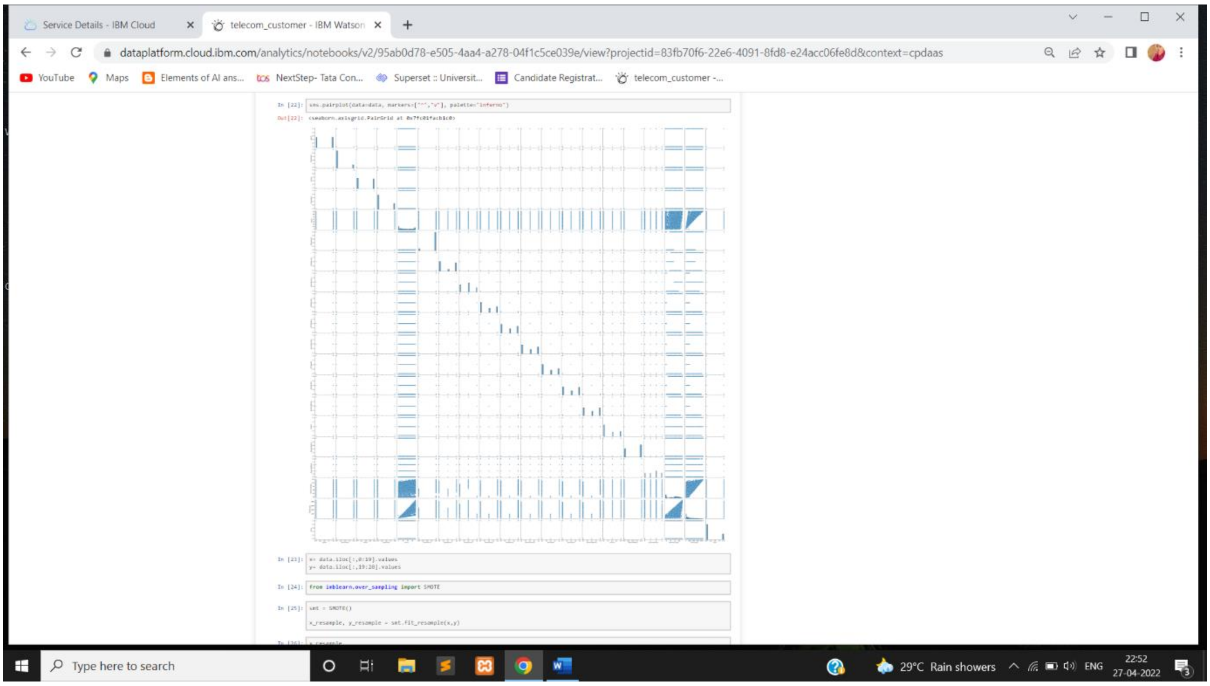



Service Details - IBM Cloud
telecom_customer - IBM Watson
dataplatform.cloud.ibm.com/analytics/notebooks/v2/95ab0d78-e505-4aa4-a278-04f1c5ce039e/view?projectid=83fb70f6-22e6-4091-8fd8-e24acc06fe8d&context=cpdaas
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
In [15]: data.head(2)
gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBackup | DeviceProt
0 | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | Yes | No
1 | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | No | Yes
In [16]: for i in data:
print(data[i].unique())
['Female' 'Male']
['Yes' 'No']
['No' 'Yes']
['No' 'Yes']
['No phone service' 'No' 'Yes']
['DSL' 'Fiber optic' 'No']
['Month-to-month' 'One year' 'Two year']
['Yes' 'No']
['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
'Credit card (automatic)']
['No' 'Yes']
Type here to search
29°C Rain showers
ENG
22:50
27-04-2022

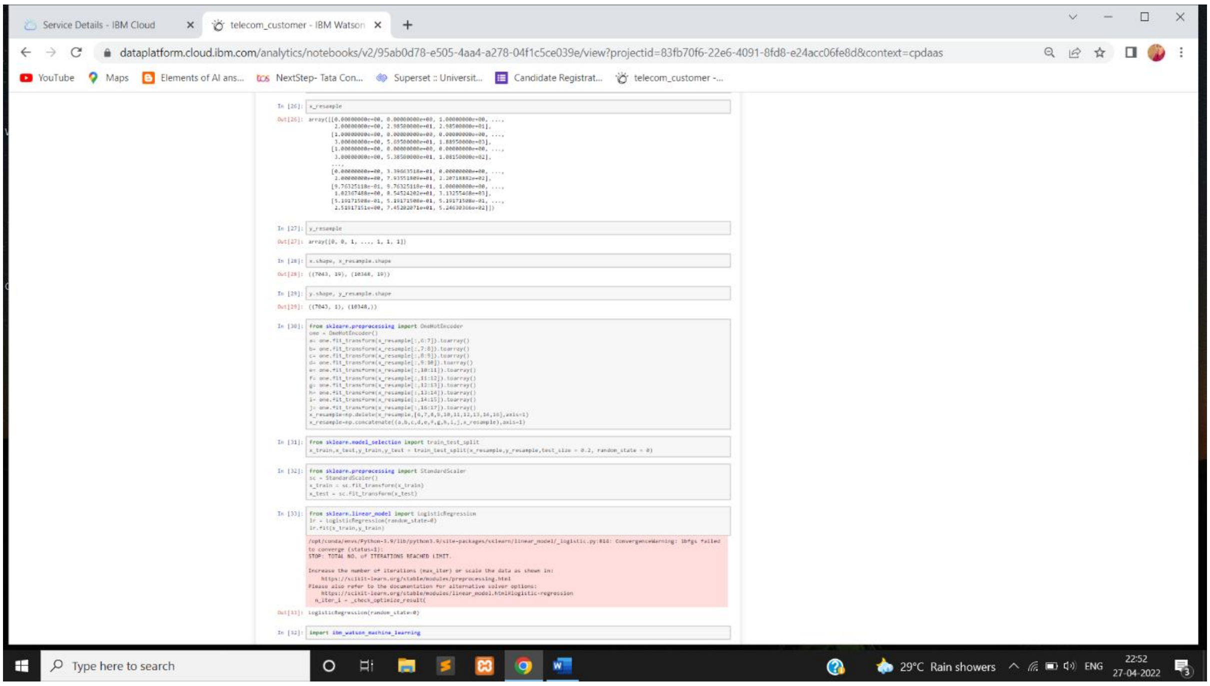

Service Details - IBM Cloud
telecom_customer - IBM Watson
dataplatform.cloud.ibm.com/analytics/notebooks/v2/95ab0d78-e505-4aa4-a278-04f1c5ce039e/view?projectid=83fb70f6-22e6-4091-8fd8-e24acc06fe8d&context=cpdaas
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
In [17]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["gender"] = le.fit_transform(data["gender"])
data["Partner"] = le.fit_transform(data["Partner"])
data["Dependents"] = le.fit_transform(data["Dependents"])
data["PhoneService"] = le.fit_transform(data["PhoneService"])
data["MultipleLines"] = le.fit_transform(data["MultipleLines"])
data["InternetService"] = le.fit_transform(data["InternetService"])
data["OnlineSecurity"] = le.fit_transform(data["OnlineSecurity"])
data["OnlineBackup"] = le.fit_transform(data["OnlineBackup"])
data["DeviceProtection"] = le.fit_transform(data["DeviceProtection"])
data["TechSupport"] = le.fit_transform(data["TechSupport"])
data["StreamingTV"] = le.fit_transform(data["StreamingTV"])
data["StreamingMovies"] = le.fit_transform(data["StreamingMovies"])
data["Contract"] = le.fit_transform(data["Contract"])
data["PaperlessBilling"] = le.fit_transform(data["PaperlessBilling"])
data["PaymentMethod"] = le.fit_transform(data["PaymentMethod"])
data["Churn"] = le.fit_transform(data["Churn"])
In [18]: for i in data:
print(data[i].unique())
In [19]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
Column Non-Null Count Dtype
0 gender 7043 non-null int64
1 SeniorCitizen 7043 non-null int64
2 Partner 7043 non-null int64
3 Dependents 7043 non-null int64
4 tenure 7043 non-null int64
5 PhoneService 7043 non-null int64
6 MultipleLines 7043 non-null int64
7 InternetService 7043 non-null int64
8 OnlineSecurity 7043 non-null int64
9 OnlineBackup 7043 non-null int64
10 DeviceProtection 7043 non-null int64
11 TechSupport 7043 non-null int64
12 StreamingTV 7043 non-null int64
13 StreamingMovies 7043 non-null int64
14 Contract 7043 non-null int64
15 PaperlessBilling 7043 non-null int64
16 PaymentMethod 7043 non-null int64
17 MonthlyCharges 7043 non-null float64
18 TotalCharges 7043 non-null float64
19 Churn 7043 non-null int64
dtypes: float64(2), int64(18)
memory usage: 1.1 MB
Type here to search
29°C Rain showers
ENG
22:51
27-04-2022

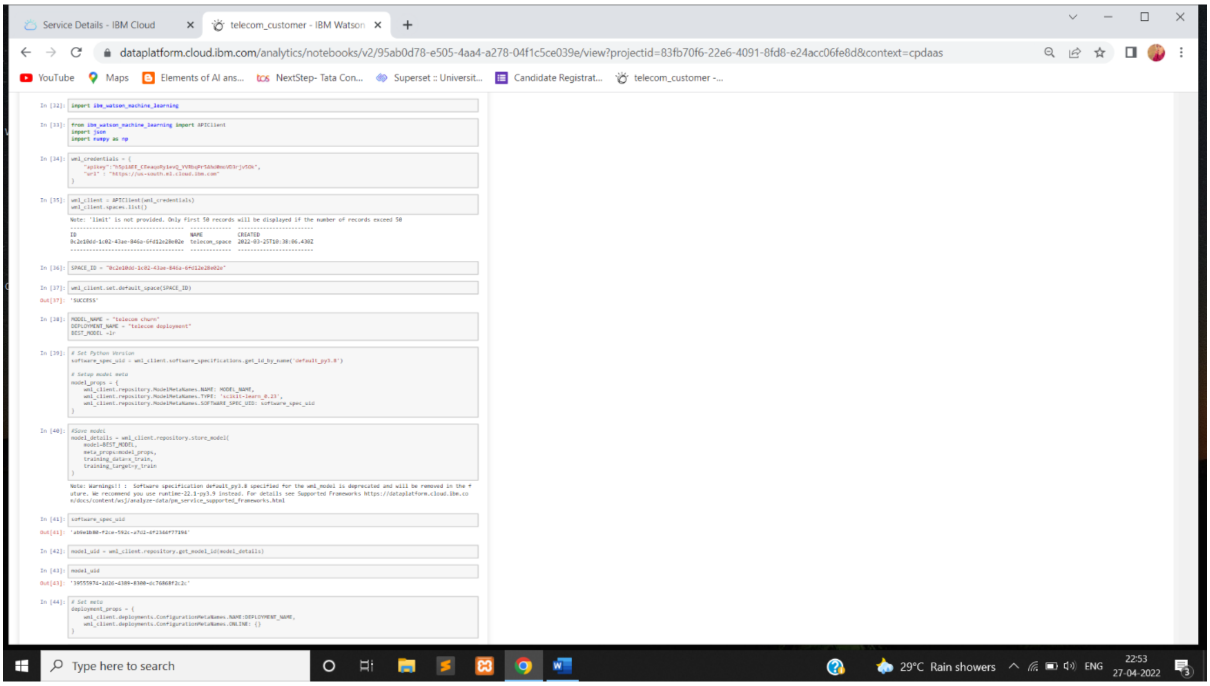


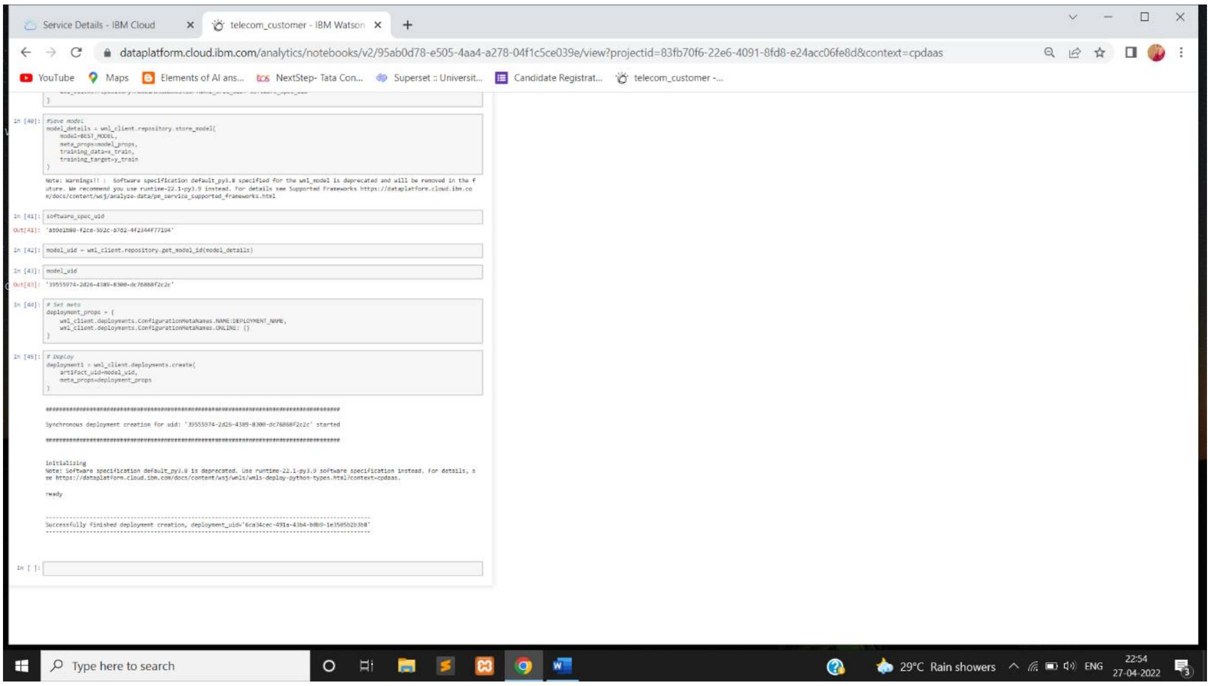
In [20]: data.corr()
Out[20]:
In [21]: sns.heatmap(data.corr(), annot=False)
Out[21]: <AxesSubplot:>



In [22]: sns.pairplot(data=data, markers=["^","v"], palette="inferno")
Out[22]: <seaborn.axisgrid.PairGrid at 0x7fc01facb1c0>
In [23]: x= data.iloc[:,0:19].values
y= data.iloc[:,19:20].values
In [24]: from imblearn.over_sampling import SMOTE
In [25]: smt = SMOTE()
x_resample, y_resample = smt.fit_resample(x,y)


Service Details - IBM Cloud
telecom_customer - IBM Watson
dataplatform.cloud.ibm.com/analytics/notebooks/v2/95ab0d78-e505-4aa4-a278-04f1c5ce039e/view?projectid=83fb70f6-22e6-4091-8fd8-e24acc06fe8d&context=cpdaas
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
Type here to search
29°C Rain showers
ENG
22:52
27-04-2022


Service Details - IBM Cloud
telecom_customer - IBM Watson
dataplatform.cloud.ibm.com/analytics/notebooks/v2/95ab0d78-e505-4aa4-a278-04f1c5ce039e/view?projectid=83fb70f6-22e6-4091-8fd8-e24acc06fe8d&context=cpdaas
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
Type here to search
29°C Rain showers
ENG
22:53
27-04-2022



Service Details - IBM Cloud
telecom_customer - IBM Watson
dataplatform.cloud.ibm.com/analytics/notebooks/v2/95ab0d78-e505-4aa4-a278-04f1c5ce039e/view?projectid=83fb70f6-22e6-4091-8fd8-e24acc06fe8d&context=cpdaas
YouTube
Maps
Elements of AI ans...
NextStep- Tata Con...
Superset :: Universit...
Candidate Registrat...
telecom_customer -...
In [40]: #Save model
model_details = wml_client.repository.store_model(
model=BEST_MODEL,
meta_props=model_props,
training_data=x_train,
training_target=y_train
)
Note: Warnings!! : Software specification default_py3.8 specified for the wml_model is deprecated and will be removed in the future. We recommend you use runtime-22.1-py3.9 instead. For details see Supported Frameworks https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/pm_service_supported_frameworks.html
In [41]: software_spec_uid
Out[41]: 'ab9e1b80-f2ce-592c-a7d2-4f2344f77194'
In [42]: model_uid = wml_client.repository.get_model_id(model_details)
In [43]: model_uid
Out[43]: '39555974-2d26-4389-8300-dc76868f2c2c'
In [44]: # Set meta
deployment_props = {
wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
In [45]: # Deploy
deployment1 = wml_client.deployments.create(
artifact_uid=model_uid,
meta_props=deployment_props
)
Synchronous deployment creation for uid: '39555974-2d26-4389-8300-dc76868f2c2c' started
initializing
Note: Software specification default_py3.8 is deprecated. Use runtime-22.1-py3.9 software specification instead. For details, see https://dataplatform.cloud.ibm.com/docs/content/wsj/wmls/wmls-deploy-python-types.html?context=cpdaas.
ready
Successfully finished deployment creation, deployment_uid='6ca34cec-491a-43b4-b0b9-1e3505b2b3b8'
In [ ]:
Type here to search
29°C Rain showers
ENG
22:54
27-04-2022