INTELLIGENT CUSTOMER RETENTION USING MACHINE LEARNING FOR ENHANCED PREDICTION OF TELECOM CUSTOMER CHURN

TEAM ID : NM2023TMID32112

	Name	NM ID
TEAM LEADER	P.MARIKRISHNAN	5CE7AA1AA997D03815E47EFBC42B9207
TEAM MEMBER	A.VIGNESH	9D540FC15246611802941112F4BF00D8
	K.DHANUVARSHAN	8ABBDE7440E766D8AD5476513971C42B
	S.MOHAMED AMJATH	4889CBA5E8044749DE147DEE46E4E5F7
	T.VIGNESH MANIKANDAN	32C485BF671494BB9EC401AB4E92CE1D

GOVERNMENT ARTS & SCIENCE COLLEGE ,KADAYANALLUR

INDEX

TELECOM CUSTOMER CHURN PREDICTION

1.INTRODUCTION

OVERVIEW

Telecom customer churn prediction is the process of using data analysis and machine learning algorithms to identify customers who are likely to cancel their subscription or switch to another provider. Churn prediction is a critical task for telecom companies because it is much more expensive to acquire new customers than to retain existing ones. Therefore, identifying at-risk customers and taking proactive measures to retain them can significantly improve a company's bottom line. The process of telecom customer churn prediction typically involves collecting and analyzing large amounts of customer data, such as call records, billing history, customer demographics, and service usage patterns. Machine learning algorithms are then used to identify patterns and trends in the data that may indicate a customer is at risk of churning. There are several techniques that can be used for telecom customer churn prediction, including logistic regression, decision trees, neural networks, and support vector machines. These algorithms are trained on historical customer data to identify common characteristics of customers who have churned in the past. Once the churn prediction model has been developed, it can be used to identify customers who are at risk of churning in real-time. This allows telecom companies to take proactive measures, such as offering discounts, improving service quality, or providing additional incentives, to retain at-risk customers.Overall, telecom customer churn prediction is an important task for telecom companies to improve customer retention, reduce churn rates, and ultimately, improve their bottom line.

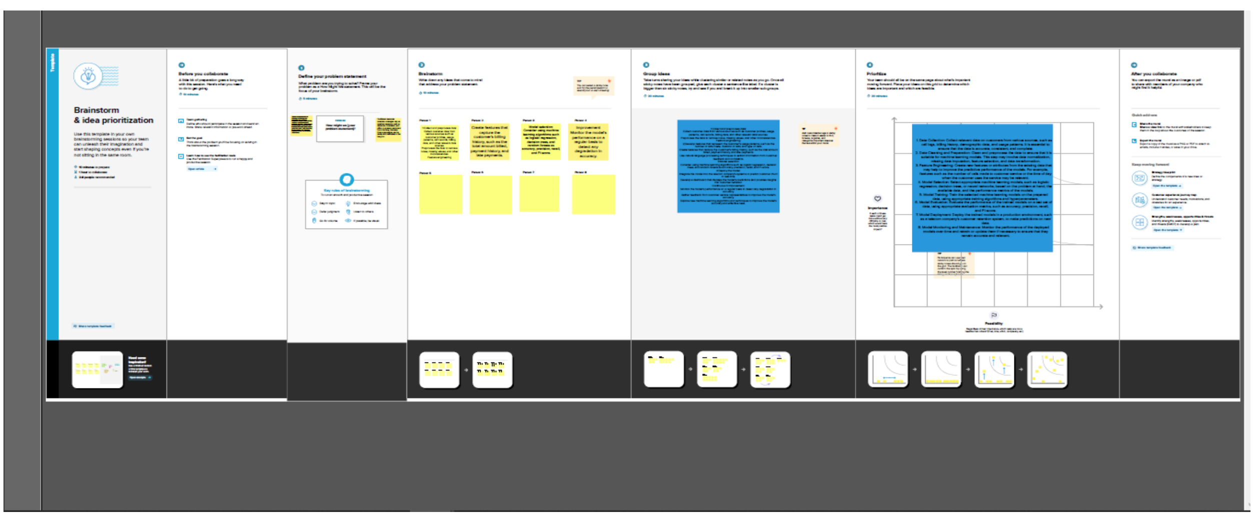
PURPOSE:

The purpose of telecom customer churn prediction is to help telecom companies identify customers who are at risk of cancelling their subscription or switching to a competitor, and take proactive measures to retain them. By analyzing customer data and using machine learning algorithms to predict churn, telecom companies can reduce their customer churn rates and improve customer retention, which can ultimately improve their bottom line. Churn prediction also helps telecom companies to understand the factors that contribute to customer churn, which can inform their marketing, customer service, and product development strategies.

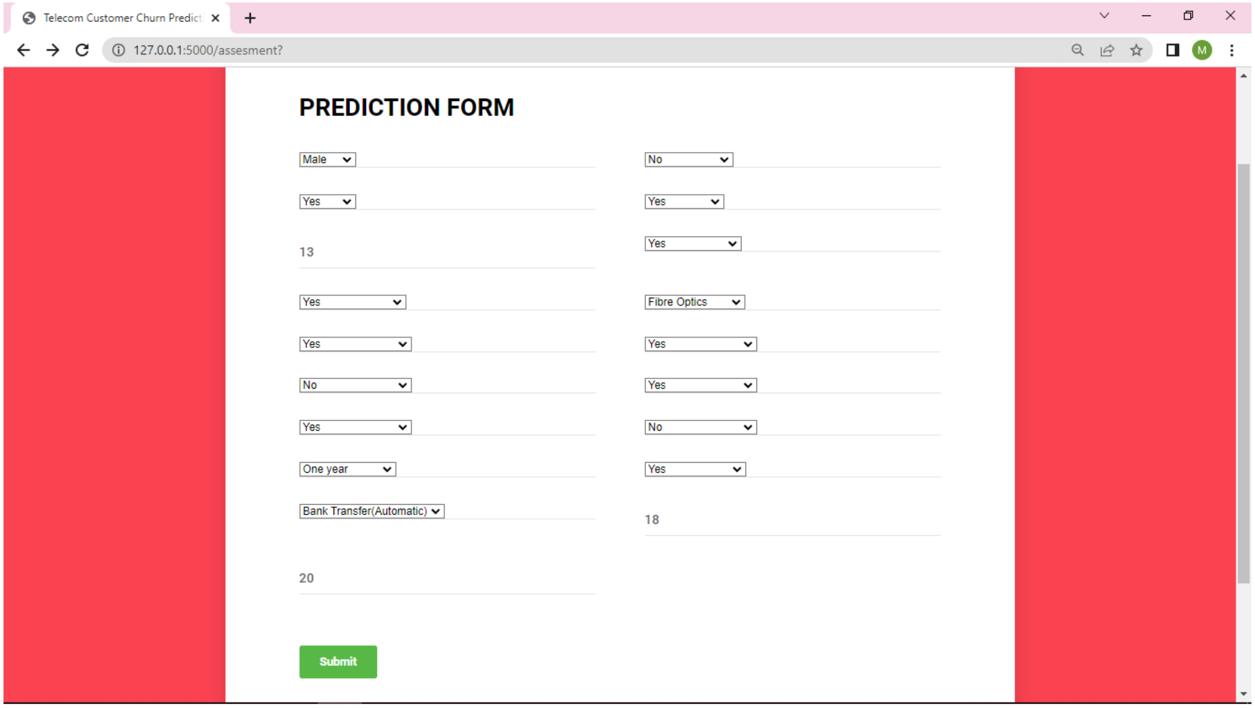
2.PROBLEM DEFINITION & DESIGN THINKING

EMPATHY MAP

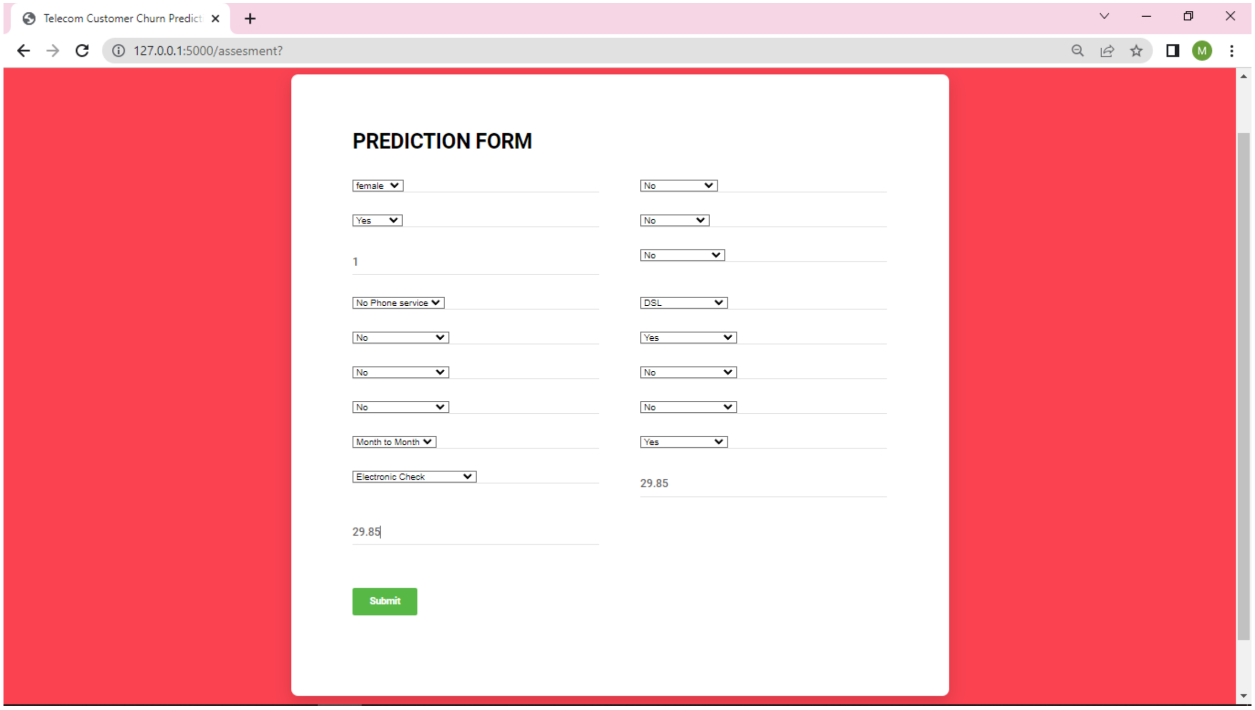
IDEATION & BRAINSTORMING MAP



3.RESULT


Telecom Customer Churn Predict
127.0.0.1:5000/assesment?
PREDICTION FORM
Male
No
Yes
Yes
13
Yes
Yes
Fibre Optics
Yes
Yes
No
Yes
Yes
No
One year
Yes
Bank Transfer(Automatic)
18
20
Submit


Telecom Churn Predict
127.0.0.1:5000/predict
TELECOM CUSTOMER CHURN PREDICTION
THE CHURN PREDICTION SAYS NO


Telecom Customer Churn Predict
127.0.0.1:5000/assesment?
PREDICTION FORM
female
Yes
1
No Phone service
No
No
No
Month to Month
Electronic Check
29.85
No
No
No
DSL
Yes
No
No
Yes
29.85
Submit


Telecom Churn Predict
127.0.0.1:5000/predict
TELECOM CUSTOMER CHURN PREDICTION
THE CHURN PREDICTION SAYS YES

4.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Improved customer retention: Telecom companies can use churn prediction models to identify at-risk customers and take proactive measures to retain them. This can help to reduce churn rates and improve customer retention.
- Cost savings: Acquiring new customers is more expensive than retaining existing ones. By reducing churn rates, telecom companies can save money on customer acquisition costs.
- Increased revenue: Retaining existing customers can lead to increased revenue through upselling and cross-selling opportunities. Churn prediction models can help to identify which customers are most likely to be receptive to these offers.

DISADVANTAGES

- Data privacy concerns: In order to develop and implement churn prediction models, telecom companies need to collect and analyze large amounts of customer data. This can raise concerns about data privacy and security.
- False positives: Churn prediction models may sometimes identify customers as being at risk of churn when they are not. This can result in unnecessary retention efforts, which may annoy customers and damage their relationship with the company.
- Limited accuracy: Churn prediction models may not always accurately predict which customers are at risk of churn. Factors such as changes in market conditions or unexpected events may lead to churn that cannot be predicted by the model.

5.APPLICATIONS

Customer retention: Telecom companies can use churn prediction models to identify at-risk customers and take proactive measures to retain them. This can include offering discounts, improving service quality, or providing additional incentives. Marketing: Churn prediction models can help telecom companies to identify which customers are most likely to be receptive to marketing campaigns. This can help to improve the effectiveness of marketing efforts and reduce marketing costs.

Product development: By understanding the factors that contribute to customer churn, telecom companies can identify areas where their products or services need improvement. This can inform product development efforts and help to create more attractive offerings for customers.

Customer service: Churn prediction models can help telecom companies to identifywhich customers are most likely to have issues with their service or require additionasupport. This can help to improve customer service and reduce the number of customer complaints.

6.CONCLUSION

In conclusion, telecom customer churn prediction is a critical task for telecom companies that can help to improve customer retention, reduce costs, increase revenue, and gain a competitive advantage. By analyzing large amounts of customer data and using machine learning algorithms to predict churn, telecom companies can identify at-risk customers and take proactive measures to retain them. However, it is important to be aware of the potential disadvantages of churn prediction, such as data privacy concerns, false positives, limited accuracy, resource requirements, and customer perception. Despite these challenges, telecom customer churn prediction has various applications that can help telecom companies to improve their bottom line and create a more positive customer experience.

7.FUTURE SCOPE

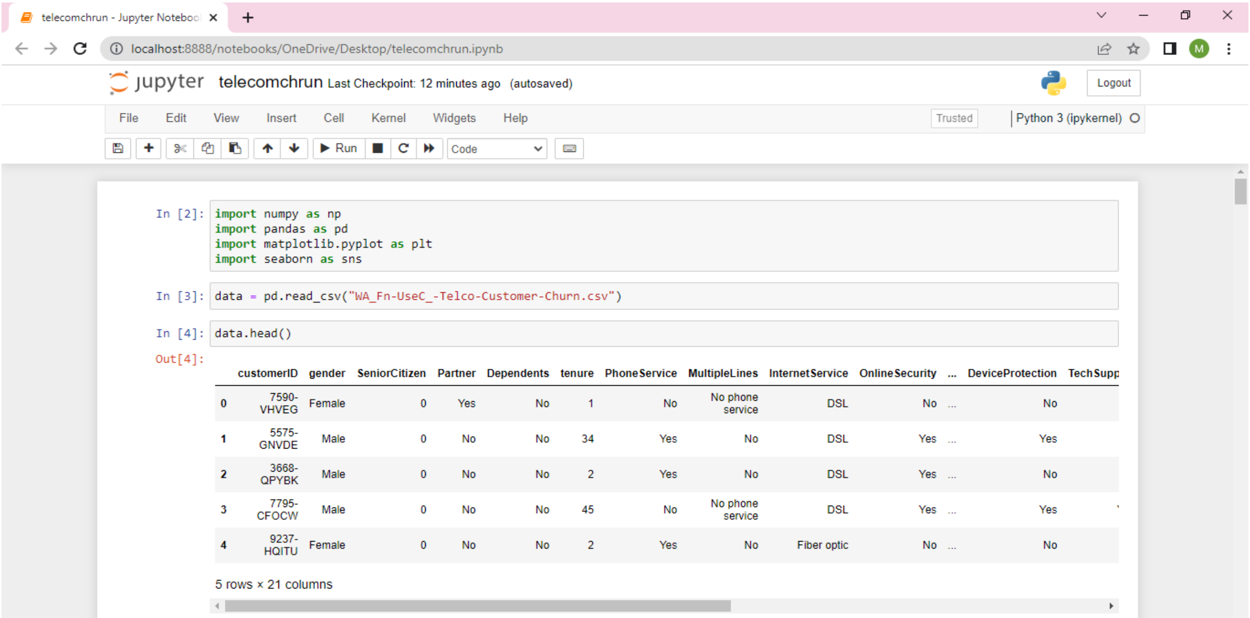
Integration with other data sources: Telecom companies may start to integrate customer churn data with other sources of data, such as social media, to gain a more complete understanding of customer behavior.

Real-time churn prediction: As computing power and data processing speeds continue to increase, churn prediction models may become more responsive and able to predict churn in real-time.

Personalized retention strategies: Churn prediction models may become more personalized, allowing telecom companies to develop targeted retention strategies for individual customers based on their unique characteristics and behavior.

Integration with other telecom services: Churn prediction may be integrated with other telecom services, such as fraud detection and network optimization, to create a more comprehensive view of telecom operations and customer behavior.

8.APPENDIX



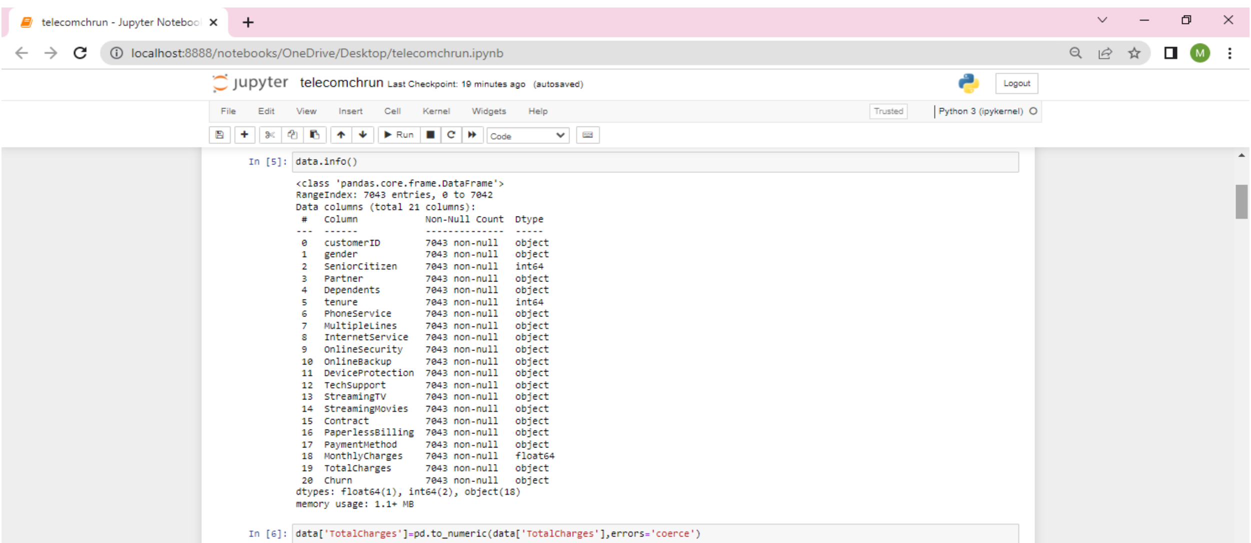
```
In [2]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

In [3]: data = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")

In [4]: data.head()
Out[4]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupp
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 [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

In [6]: data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
```

telecomchrun - Jupyter Notebook
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 21 minutes ago (autosaved)

In [7]:
```
data.head()
```
Out[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupp
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 [8]:
```
data.describe()
```
Out[8]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7032.000000
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2266.771362
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.850000	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

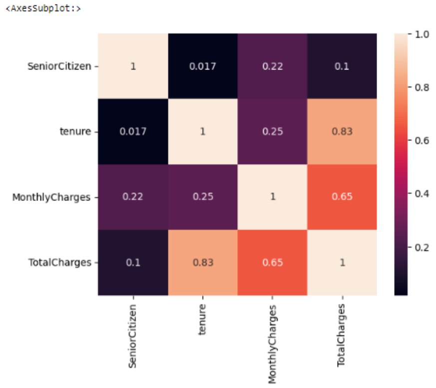
telecomchrun - Jupyter Notebook
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 22 minutes ago (autosaved)

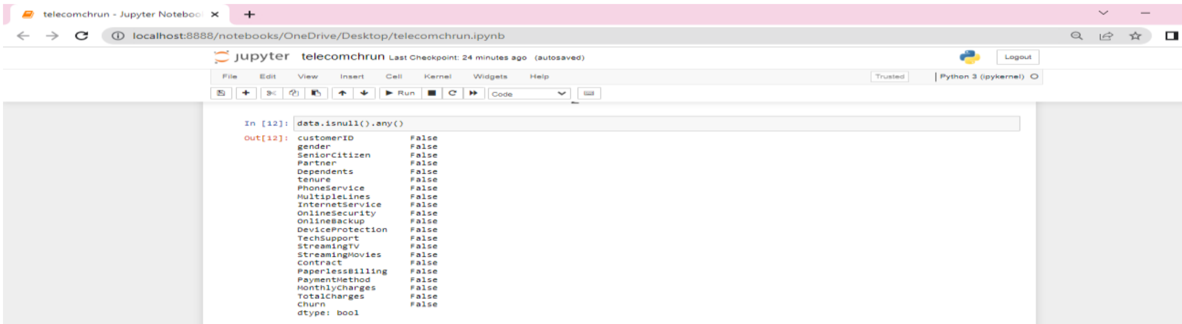
In [9]:
```
data.isnull().any()
```
Out[9]:
```
customerID          False
gender              False
SeniorCitizen       False
Partner             False
Dependents          False
tenure              False
PhoneService        False
MultipleLines       False
InternetService     False
OnlineSecurity      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
```

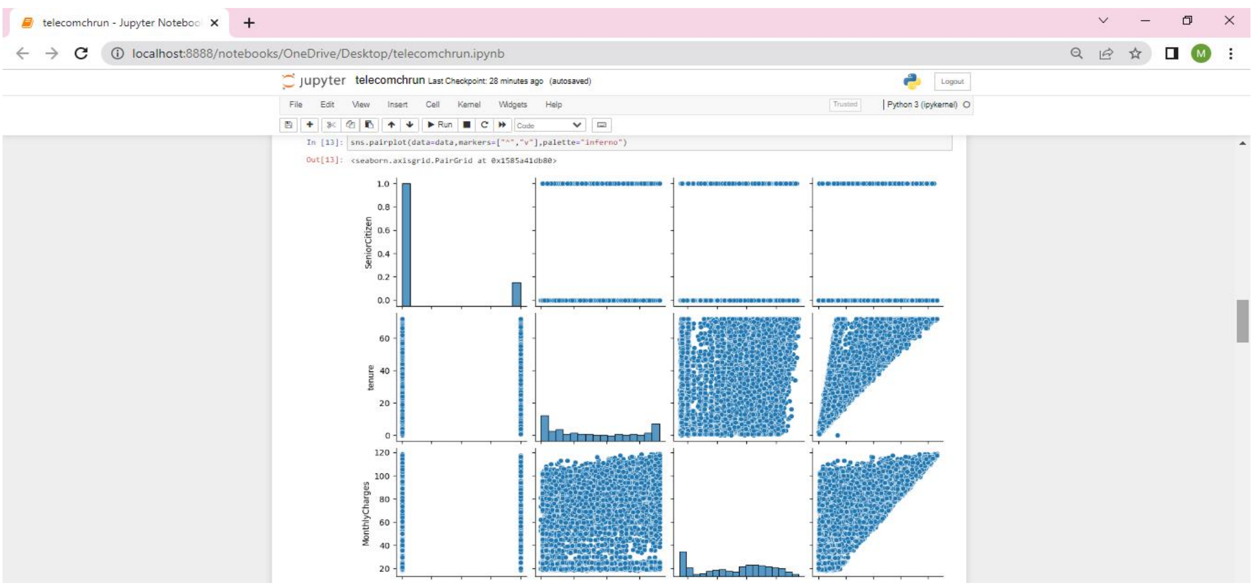
In [10]:
```
data['TotalCharges'].fillna(data['TotalCharges'].median(),inplace=True)
```

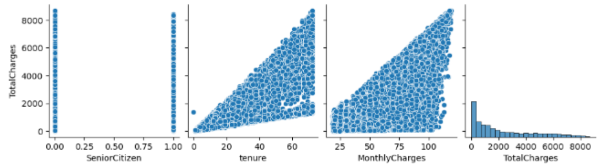
telecomchrun - Jupyter Notebook
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 23 minutes ago (autosaved)

In [11]:
```
sns.heatmap(data.corr(),annot=True)
```
Out[11]: <AxesSubplot:>




telecomchrun - Jupyter Notebook
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 24 minutes ago (autosaved)
In [12]: data.isnull().any()
Out[12]:
customerID False
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 False
Churn False
dtype: bool


telecomchrun - Jupyter Notebook
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 28 minutes ago (autosaved)
In [13]: sns.pairplot(data=data,markers=["^","v"],palette="inferno")
Out[13]: <seaborn.axisgrid.PairGrid at 0x1585a41db80>
SeniorCitizen
tenure
MonthlyCharges



In [14]:
```
data.corr()
```

Out[14]:

	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

telecomchrun - Jupyter Notebook

localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb

jupyter telecomchrun Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

In [15]:
```
data.shape
```

Out[15]: (7043, 21)

In [16]:
```
data['gender'].unique()
```

Out[16]: array(['Female', 'Male'], dtype=object)

In [17]:
```
data['Partner'].unique()
```

Out[17]: array(['Yes', 'No'], dtype=object)

In [18]:
```
data['Dependents'].unique()
```

Out[18]: array(['No', 'Yes'], dtype=object)

In [19]:
```
data['PhoneService'].unique()
```

Out[19]: array(['No', 'Yes'], dtype=object)

In [20]:
```
data['MultipleLines'].unique()
```

Out[20]: array(['No phone service', 'No', 'Yes'], dtype=object)

In [21]:
```
data['InternetService'].unique()
```

Out[21]: array(['DSL', 'Fiber optic', 'No'], dtype=object)

In [22]:
```
data['OnlineSecurity'].unique()
```

Out[22]: array(['No', 'Yes', 'No internet service'], dtype=object)

In [23]:
```
data['OnlineBackup'].unique()
```

Out[23]: array(['Yes', 'No', 'No internet service'], dtype=object)

In [24]:
```
data['DeviceProtection'].unique()
```

Out[24]: array(['No', 'Yes', 'No internet service'], dtype=object)

In [25]:
```
data['TechSupport'].unique()
```

Out[25]: array(['No', 'Yes', 'No internet service'], dtype=object)

In [26]:
```
data['StreamingTV'].unique()
```

Out[26]: array(['No', 'Yes', 'No internet service'], dtype=object) telecomchrun - Jupyter Notebook

localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb

jupyter telecomchrun Last Checkpoint: a minute ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3 (ipykernel)

In [27]:
```
data['StreamingMovies'].unique()
```
Out[27]:
```
array(['No', 'Yes', 'No internet service'], dtype=object)
```
In [28]:
```
data['Contract'].unique()
```
Out[28]:
```
array(['Month-to-month', 'One year', 'Two year'], dtype=object)
```
In [29]:
```
data['PaperlessBilling'].unique()
```
Out[29]:
```
array(['Yes', 'No'], dtype=object)
```
In [30]:
```
data['PaymentMethod'].unique()
```
Out[30]:
```
array(['Electronic check', 'Mailed check', 'Bank transfer (automatic)',
       'Credit card (automatic)'], dtype=object)
```
In [31]:
```
data['MonthlyCharges'].unique()
```
Out[31]:
```
array([29.85, 56.95, 53.85, ..., 63.1 , 44.2 , 78.7 ])
```
In [32]:
```
data['Churn'].unique()
```
Out[32]:
```
array(['No', 'Yes'], dtype=object)
```
In [33]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```
In [34]:
```
data['gender']=le.fit_transform(data['gender'])
data['Partner']=le.fit_transform(data['Partner'])
```
In [35]:
```
data['Dependents']=le.fit_transform(data['Dependents'])
data['MultipleLines']=le.fit_transform(data['MultipleLines'])
data['PhoneService']=le.fit_transform(data['PhoneService'])
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 [36]:
```
data
```

telecomchrun - Jupyter Notebook

localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb

jupyter telecomchrun Last Checkpoint: 2 minutes ago (autosaved)

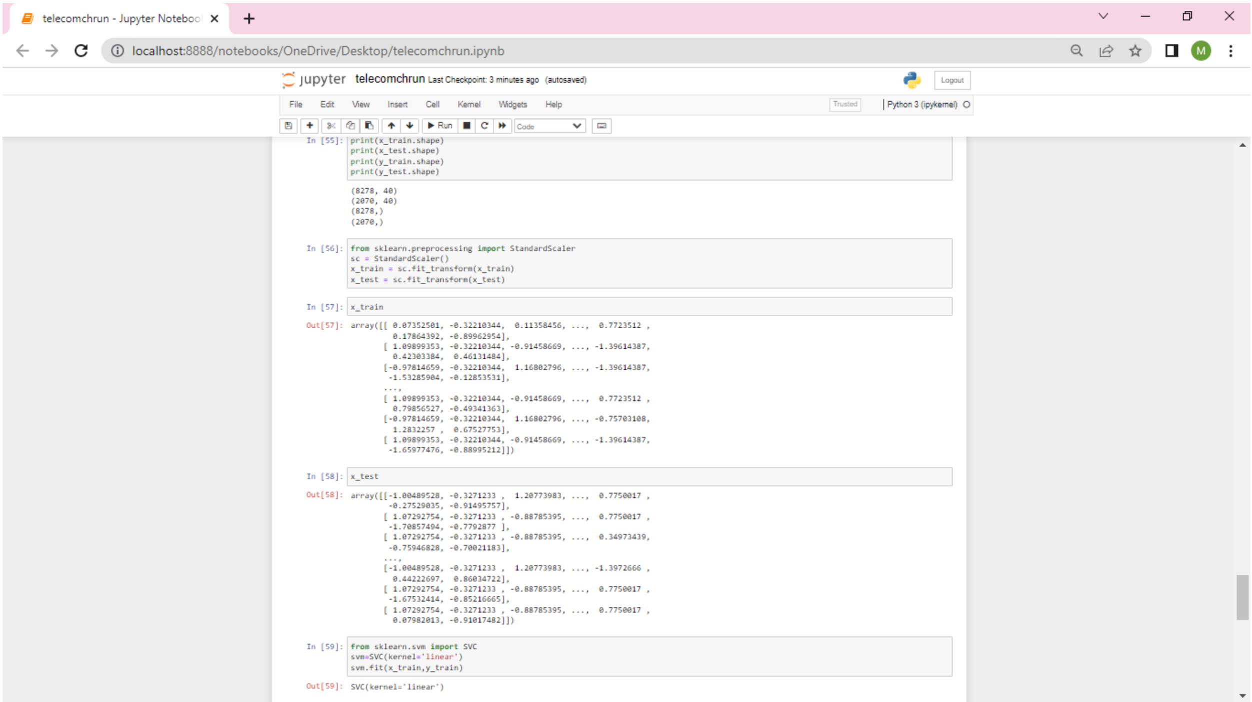
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3 (ipykernel)

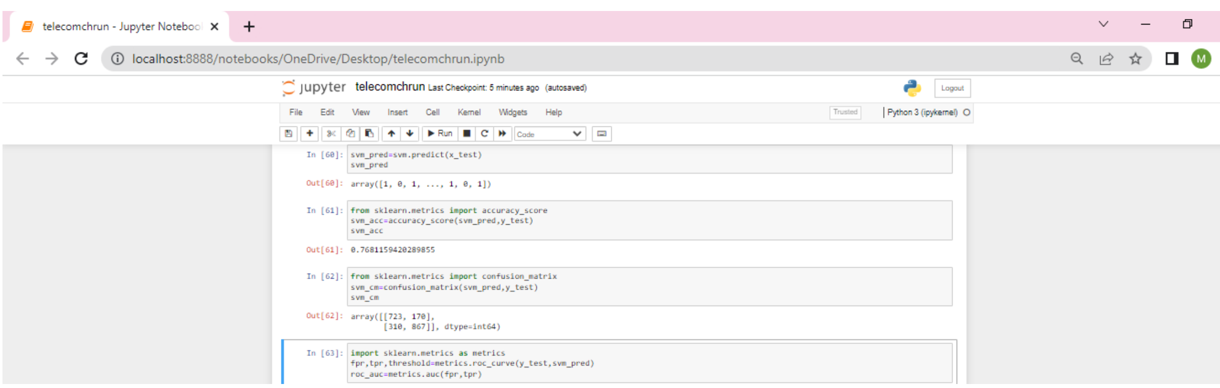
In [48]:
```
data
```
Out[48]:

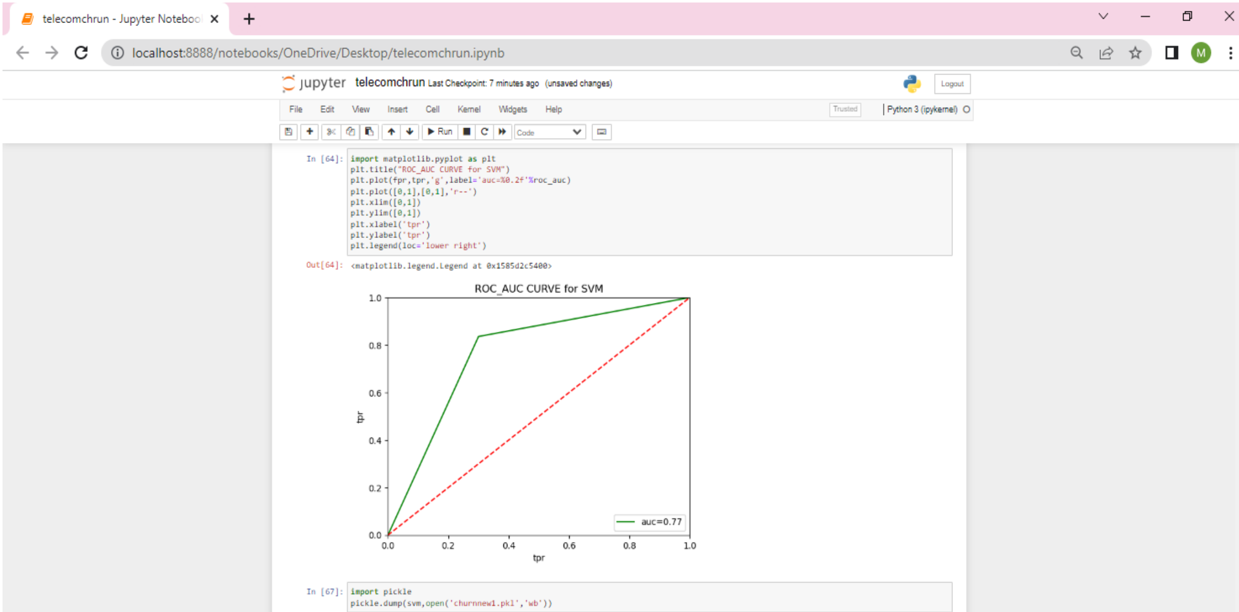
	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSu
0	0	0	1	0	1	0	1	0	0	2	0	
1	1	0	0	0	34	1	0	0	2	0	2	
2	1	0	0	0	2	1	0	0	2	2	0	
3	1	0	0	0	45	0	1	0	2	0	2	
4	0	0	0	0	2	1	0	1	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
7038	1	0	1	1	24	1	2	0	2	0	2	
7039	0	0	1	1	72	1	2	1	0	2	2	
7040	0	0	1	1	11	0	1	0	2	0	0	
7041	1	1	1	0	4	1	2	1	0	0	0	
7042	1	0	0	0	66	1	0	1	2	0	2	

7043 rows × 20 columns

In [49]:
```
x_resample.shape
```
Out[49]:
```
(10348, 40)
```
In [50]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_resample,y_resample,test_size = 0.2, random_state=0)
```
In [51]:
```
print(x_train.shape)
```
```
(8278, 40)
```
In [52]:
```
print(x_test.shape)
```
```
(2070, 40)
```
In [53]:
```
print(y_train.shape)
```
```
(8278,)
```
In [54]:
```
print(y_test.shape)
```
```
(2070,)
```


telecomchrun - Jupyter Noteboo
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 3 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [55]: print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
(8278, 40)
(2070, 40)
(8278,)
(2070,)
In [56]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
In [57]: x_train
Out[57]: array([[0.07352501, -0.32210344, 0.11358456, ..., 0.7723512 ,
0.17864392, -0.89962954],
[1.09899353, -0.32210344, -0.91458669, ..., -1.39614387,
0.42303384, 0.46131484],
[-0.97814659, -0.32210344, 1.16802796, ..., -1.39614387,
-1.53285904, -0.12853531],
...,
[1.09899353, -0.32210344, -0.91458669, ..., 0.7723512 ,
0.79856527, -0.49341363],
[-0.97814659, -0.32210344, 1.16802796, ..., -0.75703108,
1.2832257 , 0.67527753],
[1.09899353, -0.32210344, -0.91458669, ..., -1.39614387,
-1.65977476, -0.88995212]])
In [58]: x_test
Out[58]: array([[-1.00489528, -0.3271233 , 1.20773983, ..., 0.7750017 ,
-0.27529035, -0.91495757],
[1.07292754, -0.3271233 , -0.88785395, ..., 0.7750017 ,
-1.70857494, -0.7792877],
[1.07292754, -0.3271233 , -0.88785395, ..., 0.34973439,
-0.75946828, -0.70021183],
...,
[-1.00489528, -0.3271233 , 1.20773983, ..., -1.3972666 ,
0.44222697, 0.86034722],
[1.07292754, -0.3271233 , -0.88785395, ..., 0.7750017 ,
-1.67532414, -0.85216665],
[1.07292754, -0.3271233 , -0.88785395, ..., 0.7750017 ,
0.07982013, -0.91017482]])
In [59]: from sklearn.svm import SVC
svm=SVC(kernel='linear')
svm.fit(x_train,y_train)
Out[59]: SVC(kernel='linear')


telecomchrun - Jupyter Noteboo
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 5 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [60]: svm_pred=svm.predict(x_test)
svm_pred
Out[60]: array([1, 0, 1, ..., 1, 0, 1])
In [61]: from sklearn.metrics import accuracy_score
svm_acc=accuracy_score(svm_pred,y_test)
svm_acc
Out[61]: 0.7681159420289855
In [62]: from sklearn.metrics import confusion_matrix
svm_cm=confusion_matrix(svm_pred,y_test)
svm_cm
Out[62]: array([[723, 170],
[310, 867]], dtype=int64)
In [63]: import sklearn.metrics as metrics
fpr,tpr,threshold=metrics.roc_curve(y_test,svm_pred)
roc_auc=metrics.auc(fpr,tpr)



telecomchrun - Jupyter Noteboo
localhost:8888/notebooks/OneDrive/Desktop/telecomchrun.ipynb
jupyter telecomchrun Last Checkpoint: 7 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [64]:
import matplotlib.pyplot as plt
plt.title("ROC_AUC CURVE for SVM")
plt.plot(fpr,tpr,'g',label='auc=%0.2f'%roc_auc)
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('tpr')
plt.ylabel('tpr')
plt.legend(loc='lower right')
Out[64]: <matplotlib.legend.Legend at 0x1585d2c5400>
ROC_AUC CURVE for SVM
tpr
auc=0.77
In [67]:
import pickle
pickle.dump(svm,open('churnnew1.pkl','wb'))