FDS_Project(CustomerrChurnPrediction)

October 28, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

[2]:
```
df=pd.read_csv('Telco-Customer-Churn.csv')
```

[3]:
```
df.head(5)
```

[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	…	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	Female	0	Yes	No	1	No	…	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	Male	0	No	No	34	Yes	…	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	Male	0	No	No	2	Yes	…	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	Male	0	No	No	45	No	…	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	Female	0	No	No	2	Yes	…	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

[5 rows x 21 columns]

[4]:
```
df=df.dropna()
```

[5]:
```
df
```

[5]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	…
0	7590-VHVEG	Female	0	Yes	No	1	
1	5575-GNVDE	Male	0	No	No	34	
2	3668-QPYBK	Male	0	No	No	2	
3	7795-CFOCW	Male	0	No	No	45	
4	9237-HQITU	Female	0	No	No	2	
…	…	…	…	…	…	…	…
7038	6840-RESVB	Male	0	Yes	Yes	24	
7039	2234-XADUH	Female	0	Yes	Yes	72	
7040	4801-JZAZL	Female	0	Yes	Yes	11	
7041	8361-LTMKD	Male	1	Yes	No	4	
7042	3186-AJIEK	Male	0	No	No	66	

	PhoneService	MultipleLines	InternetService	OnlineSecurity	…
0	No	No phone service	DSL	No	…
1	Yes	No	DSL	Yes	…
2	Yes	No	DSL	Yes	…
3	No	No phone service	DSL	Yes	…
4	Yes	No	Fiber optic	No	…
…	…	…	…	…	…
7038	Yes	Yes	DSL	Yes	…
7039	Yes	Yes	Fiber optic	No	…
7040	No	No phone service	DSL	Yes	…
7041	Yes	Yes	Fiber optic	No	…
7042	Yes	No	Fiber optic	Yes	…

	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract
0	No	No	No	No	Month-to-month
1	Yes	No	No	No	One year
2	No	No	No	No	Month-to-month
3	Yes	Yes	No	No	One year
4	No	No	No	No	Month-to-month
…	…	…	…	…	…
7038	Yes	Yes	Yes	Yes	One year
7039	Yes	No	Yes	Yes	One year
7040	No	No	No	No	Month-to-month
7041	No	No	No	No	Month-to-month
7042	Yes	Yes	Yes	Yes	Two year

	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges
0	Yes	Electronic check	29.85	29.85

```
1                  No                Mailed check          56.95        1889.5
2                 Yes                Mailed check          53.85        108.15
3                  No  Bank transfer (automatic)          42.30       1840.75
4                 Yes           Electronic check          70.70        151.65
…                 …                           …             …           …
7038              Yes                Mailed check          84.80        1990.5
7039              Yes    Credit card (automatic)         103.20        7362.9
7040              Yes           Electronic check          29.60        346.45
7041              Yes                Mailed check          74.40         306.6
7042              Yes  Bank transfer (automatic)         105.65        6844.5

     Churn
0       No
1       No
2      Yes
3       No
4      Yes
…      …
7038    No
7039    No
7040    No
7041   Yes
7042    No

[7043 rows x 21 columns]
```

[6]:
```
df.drop_duplicates(inplace=True)
```

[7]:
```
df.head(5)
```

[7]:
```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
0  No phone service             DSL             No  …               No
1                No             DSL            Yes  …              Yes
2                No             DSL            Yes  …               No
3  No phone service             DSL            Yes  …              Yes
4                No     Fiber optic             No  …               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
```

```
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

[8]:
```
df.to_csv('Telco-Customer-Churn.csv', index=False)
```

[9]:
```
target=df['Churn']
```

[10]:
```
df=df.drop('Churn',axis=1)
```

[11]:
```
df.head(5)
```

[11]:
```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity OnlineBackup  \
0  No phone service             DSL             No          Yes
1                No             DSL            Yes           No
2                No             DSL            Yes          Yes
3  No phone service             DSL            Yes           No
4                No     Fiber optic             No           No

  DeviceProtection TechSupport StreamingTV StreamingMovies        Contract  \
0               No          No          No              No  Month-to-month
1              Yes          No          No              No        One year
2               No          No          No              No  Month-to-month
3              Yes         Yes          No              No        One year
4               No          No          No              No  Month-to-month

  PaperlessBilling              PaymentMethod  MonthlyCharges TotalCharges
0              Yes           Electronic check           29.85        29.85
1               No               Mailed check           56.95       1889.5
2              Yes               Mailed check           53.85       108.15
3               No  Bank transfer (automatic)           42.30      1840.75
```

```
4              Yes           Electronic check           70.70       151.65
```

[12]:
```
from sklearn.linear_model import LogisticRegression
```

[13]:
```
model1=LogisticRegression()
```

[14]:
```
from sklearn import tree
```

[15]:
```
model2=tree.DecisionTreeClassifier(random_state=42)
```

[16]:
```
from sklearn.ensemble import RandomForestClassifier
```

[17]:
```
model3=RandomForestClassifier(n_estimators=80)
```

[18]:
```
df.drop('customerID', axis=1, inplace=True)
```

[19]:
```
target=target.replace({'Yes':1,'No':0})
```

```
C:\Users\aswin\AppData\Local\Temp\ipykernel_33836\3723651564.py:1:
FutureWarning: Downcasting behavior in `replace` is deprecated and will be
removed in a future version. To retain the old behavior, explicitly call
`result.infer_objects(copy=False)`. To opt-in to the future behavior, set
`pd.set_option('future.no_silent_downcasting', True)`
  target=target.replace({'Yes':1,'No':0})
```

[20]:
```
df = pd.get_dummies(df, drop_first=True)
```

[21]:
```
df.head(5)
```

[21]:
```
   SeniorCitizen  tenure  MonthlyCharges  gender_Male  Partner_Yes  \
0              0       1           29.85        False         True
1              0      34           56.95         True        False
2              0       2           53.85         True        False
3              0      45           42.30         True        False
4              0       2           70.70        False        False

   Dependents_Yes  PhoneService_Yes  MultipleLines_No phone service  \
0           False             False                           True
1           False              True                          False
2           False              True                          False
3           False             False                           True
4           False              True                          False

   MultipleLines_Yes  InternetService_Fiber optic  …  TotalCharges_995.35  \
0              False                        False  …                False
1              False                        False  …                False
2              False                        False  …                False
3              False                        False  …                False
```

```
4              False                          True  …                False

   TotalCharges_996.45  TotalCharges_996.85  TotalCharges_996.95  \
0                False                False                False
1                False                False                False
2                False                False                False
3                False                False                False
4                False                False                False

   TotalCharges_997.65  TotalCharges_997.75  TotalCharges_998.1  \
0                False                False               False
1                False                False               False
2                False                False               False
3                False                False               False
4                False                False               False

   TotalCharges_999.45  TotalCharges_999.8  TotalCharges_999.9
0                False               False               False
1                False               False               False
2                False               False               False
3                False               False               False
4                False               False               False

[5 rows x 6559 columns]
```

[22]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(df,target,test_size=0.5)
```

[23]:
```
model1.fit(x_train,y_train)
```

```
C:\Users\aswin\anaconda3\Lib\site-
packages\sklearn\linear_model\_logistic.py:465: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[23]:
```
LogisticRegression()
```

[24]:
```
model1.score(x_test,y_test)
```

[24]:
```
0.7955706984667802
```

[25]:
```
model2.fit(x_train,y_train)
```

[25]:
```
DecisionTreeClassifier(random_state=42)
```

[26]:
```
model2.score(x_test,y_test)
```

[26]:
```
0.7637705848949461
```

[27]:
```
model3.fit(x_train,y_train)
```

[27]:
```
RandomForestClassifier(n_estimators=80)
```

[28]:
```
model3.score(x_test,y_test)
```

[28]:
```
0.7842135150482681
```

[29]:
```
from sklearn.ensemble import VotingClassifier
```

[30]:
```
c_modelslist = [('logreg', model1), ('dtree', model2),('rforest', model3)]
```

[31]:
```
c_model = VotingClassifier(estimators=c_modelslist, voting='soft', n_jobs=-1 )
```

[32]:
```
c_model.fit(x_train,y_train)
```

[32]:
```
VotingClassifier(estimators=[('logreg', LogisticRegression()),
                             ('dtree', DecisionTreeClassifier(random_state=42)),
                             ('rforest',
                              RandomForestClassifier(n_estimators=80))],
                 n_jobs=-1, voting='soft')
```

[33]:
```
c_model.score(x_test,y_test)
```

[33]:
```
0.778534923339012
```

[34]:
```
c_model.predict(x_test)
```

[34]:
```
array([0, 0, 0, …, 0, 1, 0])
```

[35]:
```
test=np.array([[0.0, -1.277, 1.341, -0.992, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0,␣
↪0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.
↪0, 0.0, 1.0, 0.0]])
```

[36]:
```
test = pd.DataFrame(test, columns=x_train.columns[:test.shape[1]])
```

[37]:
```
test = test.reindex(columns=x_train.columns, fill_value=0)
```

[38]:
```
c_model.predict(test)
```

[38]:
```
array([0])
```

[39]:
```
test_input = np.array([[
    0,      # SeniorCitizen = No (0)
    1,      # tenure = very low
    99.90,  # MonthlyCharges = very high
    1,      # gender_Male
    0,      # Partner_Yes
    0,      # Dependents_Yes
    1,      # PhoneService_Yes
    0,      # MultipleLines_No phone service
    1,      # MultipleLines_Yes
    1,      # InternetService_Fiber optic
    1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
]])
```

[40]:
```
test = pd.DataFrame(test, columns=x_train.columns[:test.shape[1]])
```

[41]:
```
test = test.reindex(columns=x_train.columns, fill_value=0)
```

[42]:
```
c_model.predict(test)
```

[42]:
```
array([0])
```

[43]:
```
churned_customer = df[target == 1].sample(1)
```

[44]:
```
churned_customer_array = churned_customer.values.reshape(1, -1)
```

[45]:
```
c_model.predict(churned_customer_array)
```

```
C:\Users\aswin\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2739:
UserWarning: X does not have valid feature names, but LogisticRegression was
fitted with feature names
  warnings.warn(
C:\Users\aswin\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2739:
UserWarning: X does not have valid feature names, but DecisionTreeClassifier was
fitted with feature names
  warnings.warn(
C:\Users\aswin\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2739:
UserWarning: X does not have valid feature names, but RandomForestClassifier was
fitted with feature names
  warnings.warn(
```

[45]:
```
array([0])
```

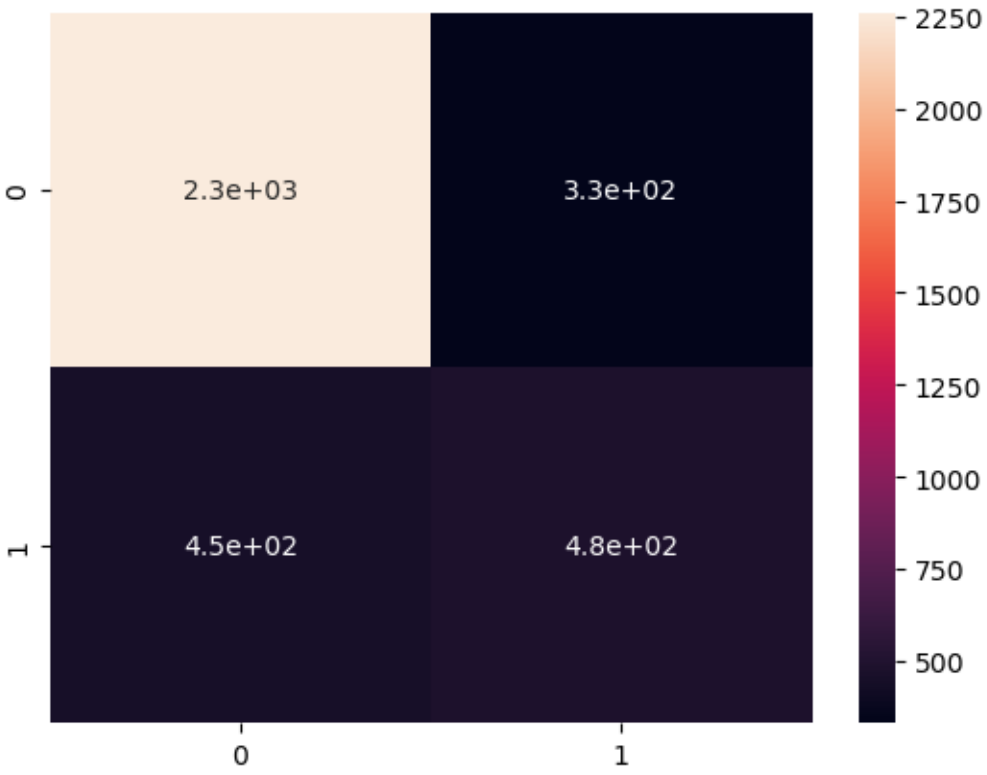
[46]:
```
y_predict=c_model.predict(x_test)
```

[47]:
```
from sklearn.metrics import confusion_matrix
```

[48]:
```
visual=confusion_matrix(y_test,y_predict)
```

[49]:
```
sn.heatmap(visual,annot=True)
```

[49]: <Axes: >



[ ]: