import numpy as np   
import pandas as pd   
import matplotlib.pyplot as plt   
import seaborn as sns   
import warnings   
warnings.filterwarnings("ignore")

df=pd.read\_csv("telecom\_churn.csv")

df.head()

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]

df.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

df.isnull().sum()

customerID 0   
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

df.TotalCharges.value\_counts()

11   
20.2 11   
19.75 9   
20.05 8   
19.9 8  
 ..

6849.4 1   
692.35 1   
130.15 1   
3211.9 1   
6844.5 1   
Name: TotalCharges, Length: 6531, dtype: int64

df["TotalCharges"].replace(" ",np.nan,inplace=True)

df.TotalCharges.value\_counts()

20.2 11   
19.75 9   
20.05 8   
19.9 8   
19.65 8  
 ..

6849.4 1   
692.35 1   
130.15 1   
3211.9 1   
6844.5 1   
Name: TotalCharges, Length: 6530, dtype: int64

df["TotalCharges"]=df["TotalCharges"].astype(float)

df.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 7032 non-null float64 20 Churn 7043 non-null object dtypes: float64(2), int64(2), object(17)   
memory usage: 1.1+ MB

df["TotalCharges"].fillna(df["TotalCharges"].median(),inplace=True)

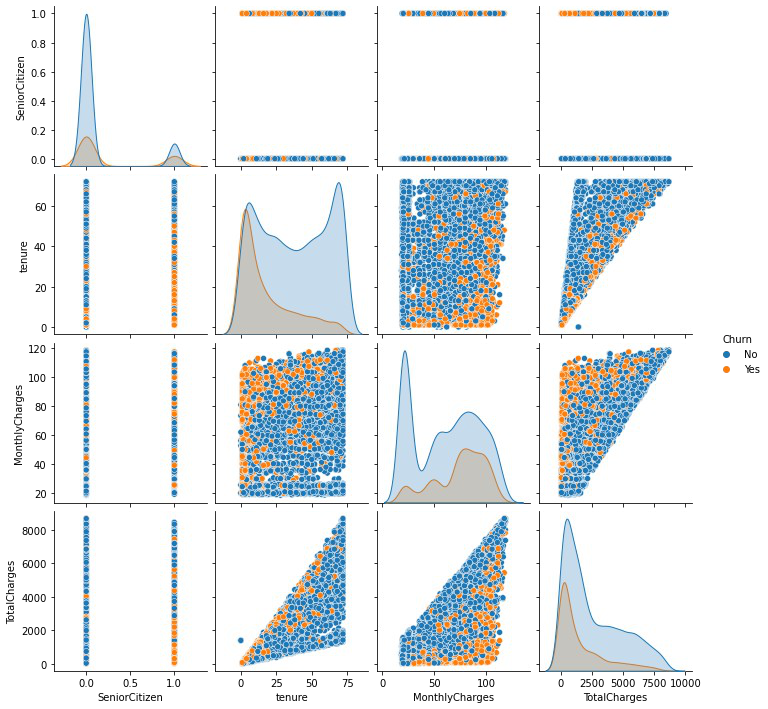
df.TotalCharges.isnull().sum()

0

df.describe()

SeniorCitizen tenure MonthlyCharges TotalCharges count 7043.000000 7043.000000 7043.000000 7043.000000 mean 0.162147 32.371149 64.761692 2281.916928 std 0.368612 24.559481 30.090047 2265.270398 min 0.000000 0.000000 18.250000 18.800000 25% 0.000000 9.000000 35.500000 402.225000 50% 0.000000 29.000000 70.350000 1397.475000 75% 0.000000 55.000000 89.850000 3786.600000 max 1.000000 72.000000 118.750000 8684.800000

sns.pairplot(data=df,hue="Churn")   
plt.show()



from sklearn.preprocessing import MinMaxScaler   
mm=MinMaxScaler()   
df[["TotalCharges"]]=mm.fit\_transform(df[["TotalCharges"]])

num=df.select\_dtypes(['float64','int64']) cat=df.select\_dtypes(['object'])

from sklearn.preprocessing import LabelEncoder le=LabelEncoder()

**for** i **in** cat:  
 le=LabelEncoder()  
 cat[i]=le.fit\_transform(cat[i])

cat

customerID gender Partner Dependents PhoneService MultipleLines \   
0 5375 0 1 0 0 1

1 3962 1 0 0 1 0   
2 2564 1 0 0 1 0   
3 5535 1 0 0 0 1   
4 6511 0 0 0 1 0   
... ... ... ... ... ... ...

7038 4853 1 1 1 1 2   
7039 1525 0 1 1 1 2   
7040 3367 0 1 1 0 1   
7041 5934 1 1 0 1 2   
7042 2226 1 0 0 1 0

InternetService OnlineSecurity OnlineBackup DeviceProtection \   
0 0 0 2 0

1 0 2 0 2

2 0 2 2 0

3 0 2 0 2

4 1 0 0 0

... ... ... ... ...

7038 0 2 0 2

7039 1 0 2 2

7040 0 2 0 0

7041 1 0 0 0

7042 1 2 0 2

TechSupport StreamingTV StreamingMovies Contract PaperlessBilling \   
0 0 0 0 0

1   
1 0 0 0 1 0   
2 0 0 0 0 1   
3 2 0 0 1 0   
4 0 0 0 0 1   
... ... ... ... ... ...

7038 2 2 2 1 1   
7039 0 2 2 1 1   
7040 0 0 0 0 1   
7041 0 0 0 0 1   
7042 2 2 2 2 1

PaymentMethod Churn   
0 2 0   
1 3 0   
2 3 1   
3 0 0   
4 2 1   
... ... ...   
7038 3 0   
7039 1 0   
7040 2 0   
7041 3 1   
7042 0 0

[7043 rows x 17 columns]

cat.drop("customerID",axis=1,inplace=True)

df=pd.concat([num,cat],axis=1)

df.head()

SeniorCitizen tenure MonthlyCharges TotalCharges gender Partner \   
0 0 1 29.85 0.001275 0 1   
1 0 34 56.95 0.215867 1 0   
2 0 2 53.85 0.010310 1 0

3 0 45 42.30 0.210241 1 0   
4 0 2 70.70 0.015330 0 0

Dependents PhoneService MultipleLines InternetService OnlineSecurity \   
0 0 0 1 0 0   
1 0 1 0 0 2   
2 0 1 0 0 2   
3 0 0 1 0 2   
4 0 1 0 1 0

OnlineBackup DeviceProtection TechSupport StreamingTV StreamingMovies \   
0 2 0 0 0 0   
1 0 2 0 0 0   
2 2 0 0 0 0   
3 0 2 2 0 0   
4 0 0 0 0 0

Contract PaperlessBilling PaymentMethod Churn 0 0 1 2 0 1 1 0 3 0 2 0 1 3 1 3 1 0 0 0 4 0 1 2 1

x=df.iloc[:,:-1]   
x

SeniorCitizen tenure MonthlyCharges TotalCharges gender Partner \   
0 0 1 29.85 0.001275 0 1   
1 0 34 56.95 0.215867 1 0   
2 0 2 53.85 0.010310 1 0   
3 0 45 42.30 0.210241 1

0   
4 0 2 70.70 0.015330 0 0   
... ... ... ... ... ... ...

7038 0 24 84.80 0.227521 1 1   
7039 0 72 103.20 0.847461 0 1   
7040 0 11 29.60 0.037809 0 1   
7041 1 4 74.40 0.033210 1 1   
7042 0 66 105.65 0.787641 1 0

Dependents PhoneService MultipleLines InternetService \ 0 0 0 1 0 1 0 1 0 0 2 0 1 0 0 3 0 0 1 0 4 0 1 0 1 ... ... ... ... ... 7038 1 1 2 0 7039 1 1 2 1 7040 1 0 1 0 7041 0 1 2 1 7042 0 1 0 1

OnlineSecurity OnlineBackup DeviceProtection TechSupport \ 0 0 2 0 0 1 2 0 2 0 2 2 2 0 0 3 2 0 2 2 4 0 0 0 0 ... ... ... ... ... 7038 2 0 2 2 7039 0 2 2 0 7040 2 0 0 0 7041 0 0 0 0 7042 2 0 2 2

StreamingTV StreamingMovies Contract PaperlessBilling PaymentMethod   
0 0 0 0 1 2   
1 0 0 1 0 3   
2 0 0 0 1 3

3 0 0 1 0 0   
4 0 0 0 1 2   
... ... ... ... ... ...

7038 2 2 1 1 3   
7039 2 2 1 1 1   
7040 0 0 0 1 2   
7041 0 0 0 1 3   
7042 2 2 2 1 0

[7043 rows x 19 columns]

y=df.iloc[:,-1]   
y

0 0   
1 0   
2 1   
3 0   
4 1  
 ..

7038 0   
7039 0   
7040 0   
7041 1   
7042 0   
Name: Churn, Length: 7043, dtype: int32

from sklearn.model\_selection import train\_test\_split   
xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.25,random\_s tate=1)

from sklearn.linear\_model import LogisticRegression   
from sklearn.neighbors import KNeighborsClassifier   
from sklearn.svm import SVC   
from sklearn.tree import DecisionTreeClassifier   
from sklearn.metrics import classification\_report,accuracy\_score

**def** mymodel(model):  
 model.fit(xtest,ytest)  
 ypred=model.predict(xtest)  
 accu=accuracy\_score(ytest,ypred)  
 cr=classification\_report(ytest,ypred) print(f"Accuracy : {accu}\n{cr}")