AI agent customer churn

```
import pandas as pd
import seaborn as sns
```

```
dt = pd.read_csv('/content/WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

```
dt.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   int16
 1   gender            7043 non-null   int8
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   int8
 4   Dependents        7043 non-null   int8
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   int8
 7   MultipleLines     7043 non-null   int8
 8   InternetService   7043 non-null   int8
 9   OnlineSecurity    7043 non-null   int8
 10  OnlineBackup      7043 non-null   int8
 11  DeviceProtection  7043 non-null   int8
 12  TechSupport       7043 non-null   int8
 13  StreamingTV       7043 non-null   int8
 14  StreamingMovies   7043 non-null   int8
 15  Contract          7043 non-null   int8
 16  PaperlessBilling  7043 non-null   int8
 17  PaymentMethod     7043 non-null   int8
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   int16
 20  Churn             7043 non-null   int8
dtypes: float64(1), int16(2), int64(2), int8(16)
memory usage: 302.8 KB
```

```
for i in dt.columns:
  if dt[i].dtype == 'object':
    dt[i] = dt[i].astype('category').cat.codes
```

```
dt.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   int16
 1   gender            7043 non-null   int8
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   int8
 4   Dependents        7043 non-null   int8
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   int8
 7   MultipleLines     7043 non-null   int8
 8   InternetService   7043 non-null   int8
 9   OnlineSecurity    7043 non-null   int8
 10  OnlineBackup      7043 non-null   int8
 11  DeviceProtection  7043 non-null   int8
 12  TechSupport       7043 non-null   int8
 13  StreamingTV       7043 non-null   int8
 14  StreamingMovies   7043 non-null   int8
 15  Contract          7043 non-null   int8
 16  PaperlessBilling  7043 non-null   int8
 17  PaymentMethod     7043 non-null   int8
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   int16
 20  Churn             7043 non-null   int8
dtypes: float64(1), int16(2), int64(2), int8(16)
memory usage: 302.8 KB
```

```
dt.drop(['customerID'], axis = 1,inplace = True)
```

```
dt.corr()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
gender	1.000000	-0.001874	-0.001808	0.010517	0.005106	-0.006488	-0.006739	-0.000863	-0.015017
SeniorCitizen	-0.001874	1.000000	0.016479	-0.211185	0.016567	0.008576	0.146185	-0.032310	-0.128221
Partner	-0.001808	0.016479	1.000000	0.452676	0.379697	0.017706	0.142410	0.000891	0.150828
Dependents	0.010517	-0.211185	0.452676	1.000000	0.159712	-0.001762	-0.024991	0.044590	0.152166
tenure	0.005106	0.016567	0.379697	0.159712	1.000000	0.008448	0.343032	-0.030359	0.325468
PhoneService	-0.006488	0.008576	0.017706	-0.001762	0.008448	1.000000	-0.020538	0.387436	-0.015198
MultipleLines	-0.006739	0.146185	0.142410	-0.024991	0.343032	-0.020538	1.000000	-0.109216	0.007141
InternetService	-0.000863	-0.032310	0.000891	0.044590	-0.030359	0.387436	-0.109216	1.000000	-0.028416
OnlineSecurity	-0.015017	-0.128221	0.150828	0.152166	0.325468	-0.015198	0.007141	-0.028416	1.000000
OnlineBackup	-0.012057	-0.013632	0.153130	0.091015	0.370876	0.024105	0.117327	0.036138	0.185126
DeviceProtection	0.000549	-0.021398	0.166330	0.080537	0.371105	0.003727	0.122318	0.044944	0.175985
TechSupport	-0.006825	-0.151268	0.126733	0.133524	0.322942	-0.019158	0.011466	-0.026047	0.285028
StreamingTV	-0.006421	0.030776	0.137341	0.046885	0.289373	0.055353	0.175059	0.107417	0.044669
StreamingMovies	-0.008743	0.047266	0.129574	0.021321	0.296866	0.043870	0.180957	0.098350	0.055954
Contract	0.000126	-0.142554	0.294806	0.243187	0.671607	0.002247	0.110842	0.099721	0.374416
PaperlessBilling	-0.011754	0.156530	-0.014877	-0.111377	0.006152	0.016505	0.165146	-0.138625	-0.157641
PaymentMethod	0.017352	-0.038551	-0.154798	-0.040292	-0.370436	-0.004184	-0.176793	0.086140	-0.096726
MonthlyCharges	-0.014569	0.220173	0.096848	-0.113890	0.247900	0.247398	0.433576	-0.323260	-0.053878
TotalCharges	-0.005291	0.037653	0.059568	-0.009572	0.158523	0.083195	0.114955	-0.055724	0.042357
Churn	-0.008612	0.150889	-0.150448	-0.164221	-0.352229	0.011942	0.038037	-0.047291	-0.289309

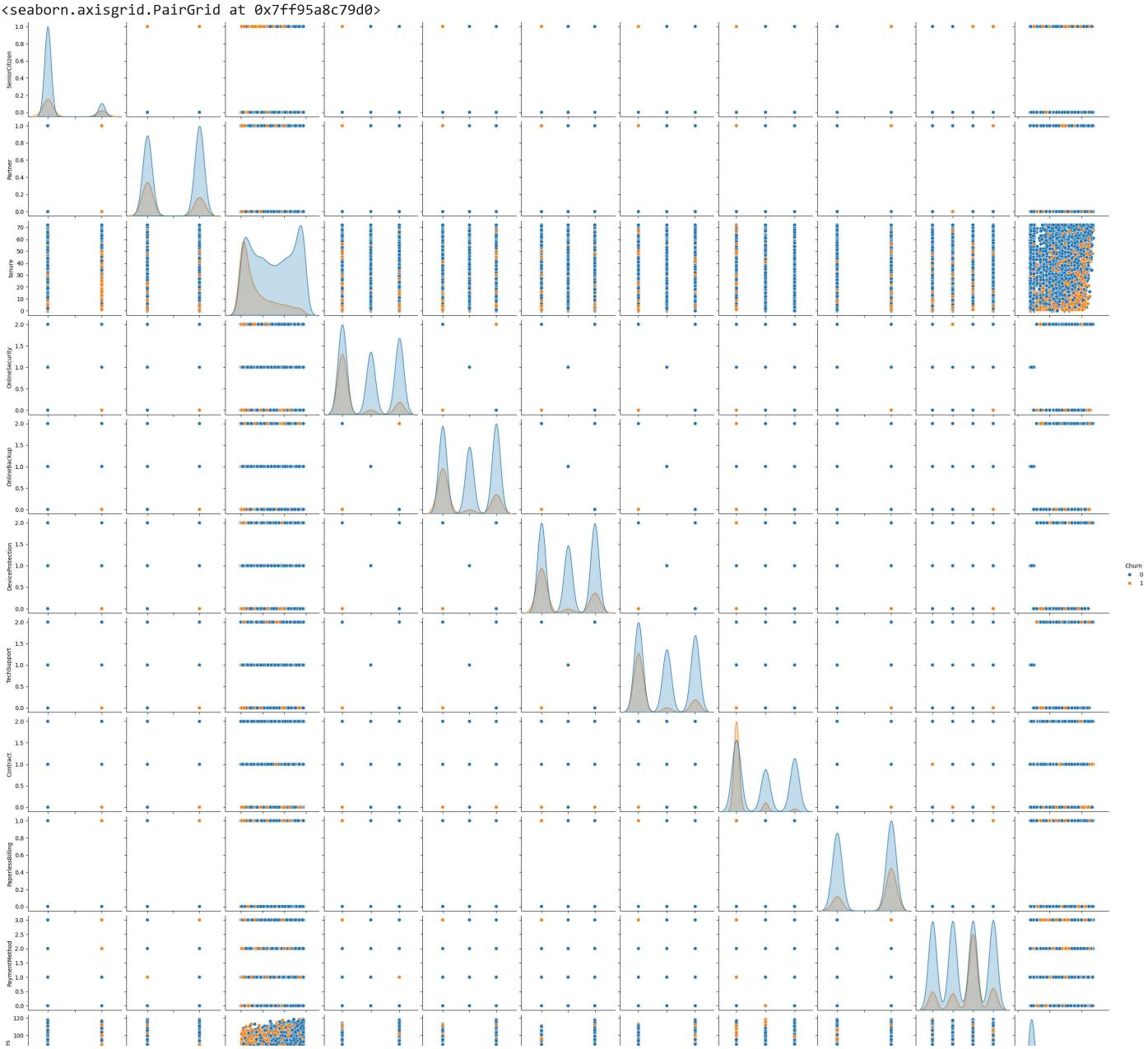
```
dt = dt.drop(["gender","Dependents","PhoneService","MultipleLines","InternetService"],axis = 1)
```

```
x = dt.drop(["Churn"], axis = 1)
y = dt["Churn"]
```

```
from sklearn.model_selection import train_test_split
xtrain , xtest ,ytrain, ytest = train_test_split(x,y,test_size = 0.2)
```

```
sns.pairplot(data = dt,hue= 'Churn')
```

```
<seaborn.axisgrid.PairGrid at 0x7ff95a8c79d0>
```




```
from sklearn.tree import DecisionTreeClassifier
d = DecisionTreeClassifier()
d.fit(xtrain,ytrain)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
ypred = d.predict(xtest)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(ytest,ypred)
```

```
0.7097232079489
```

```
import pickle
```

```
pickle.dump(dt,open('churn.pkl','wb'))
```

```
Start coding or generate with AI.
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
```