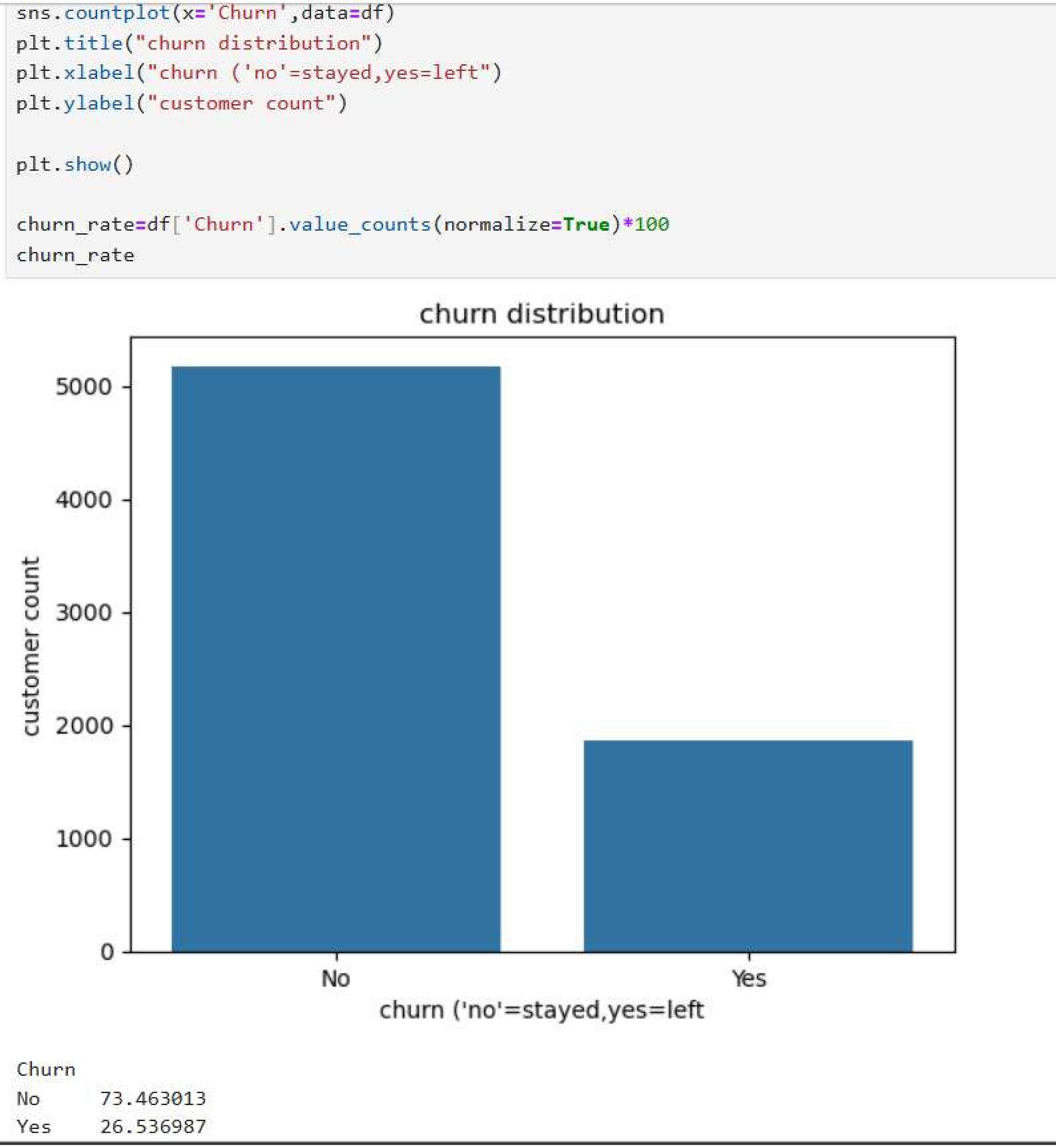
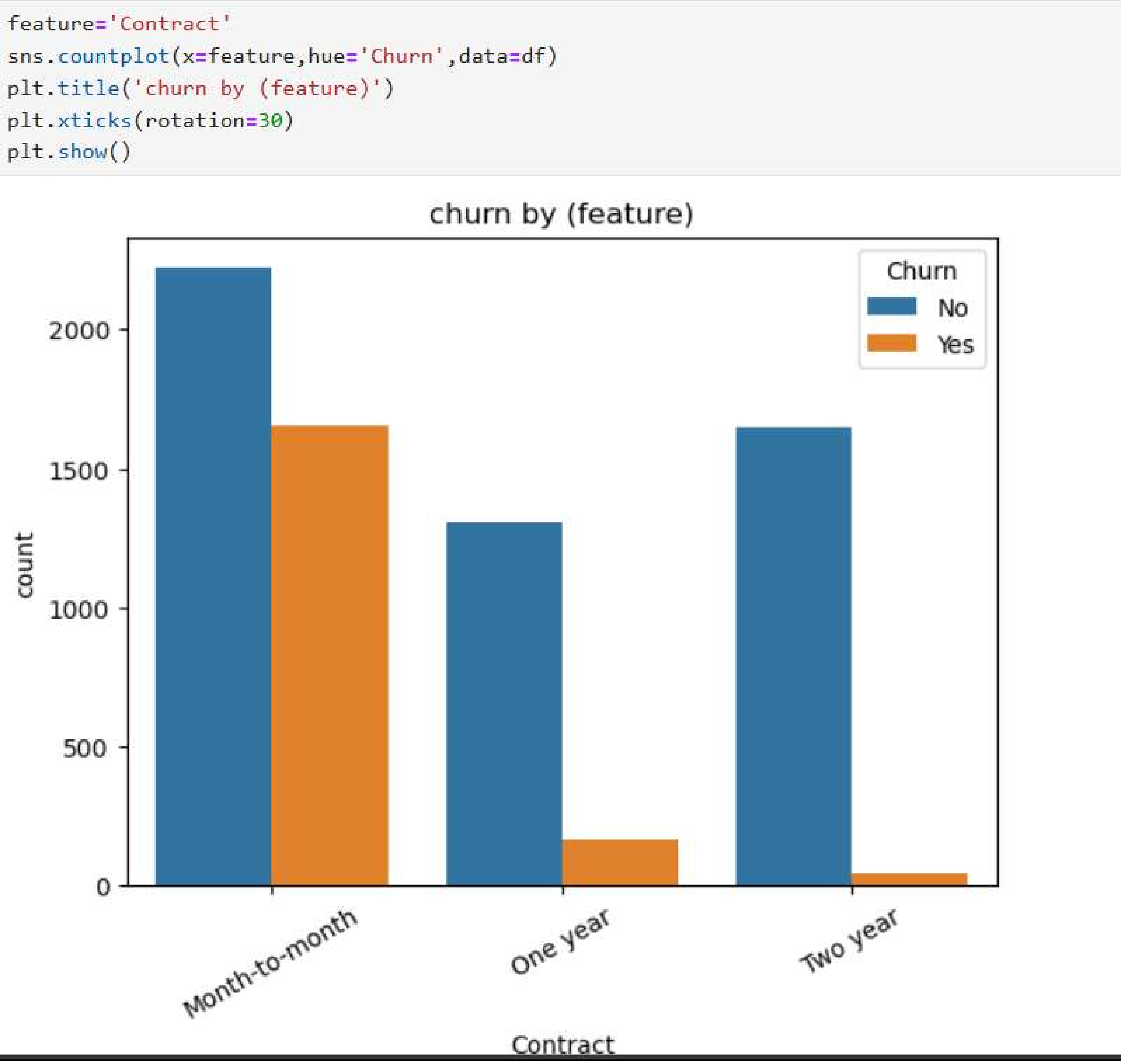
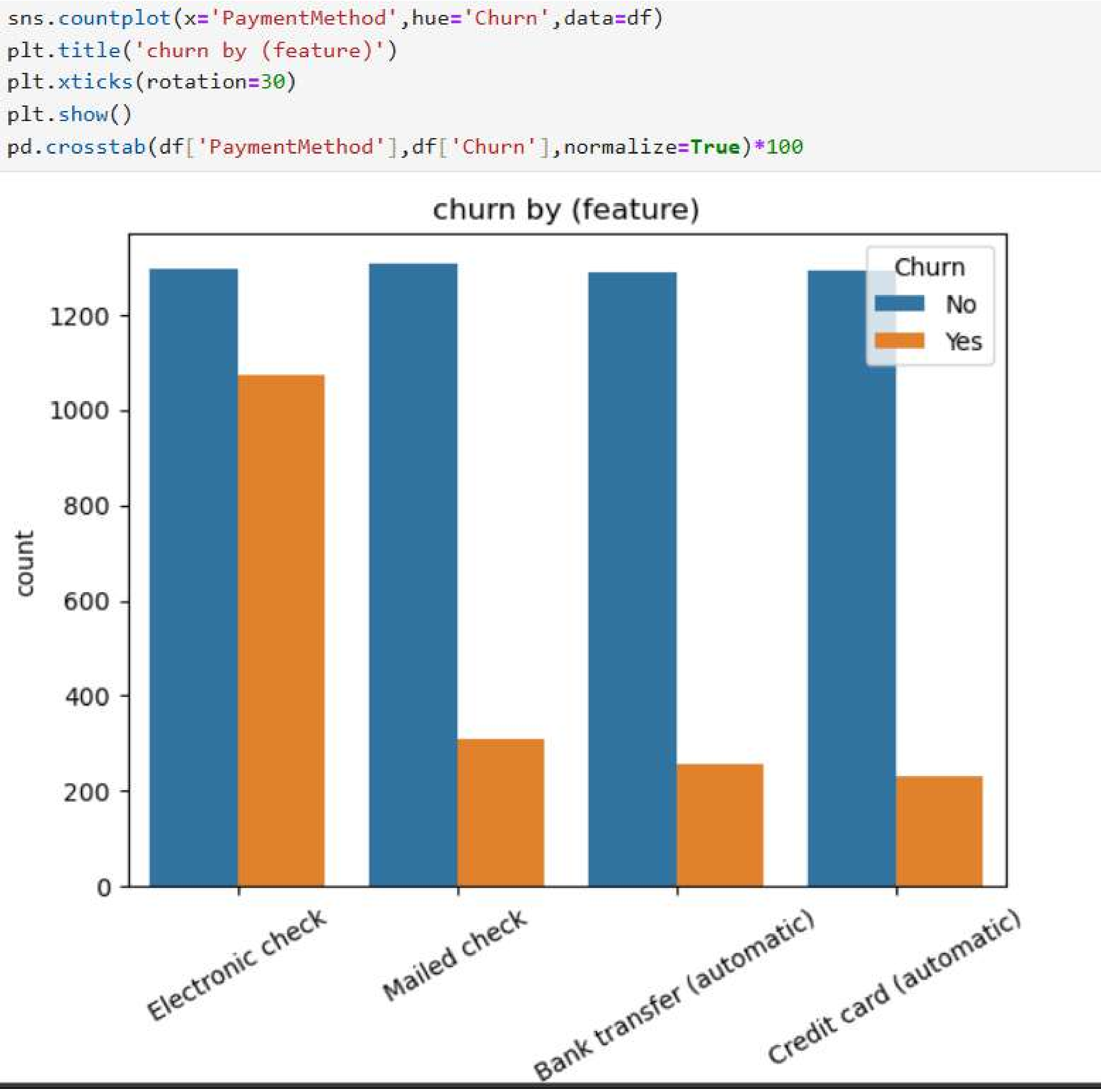


```
sns.countplot(x='Churn',data=df)
plt.title("churn distribution")
plt.xlabel("churn ('no'=stayed,yes=left")
plt.ylabel("customer count")

plt.show()

churn_rate=df['Churn'].value_counts(normalize=True)*100
churn_rate
```

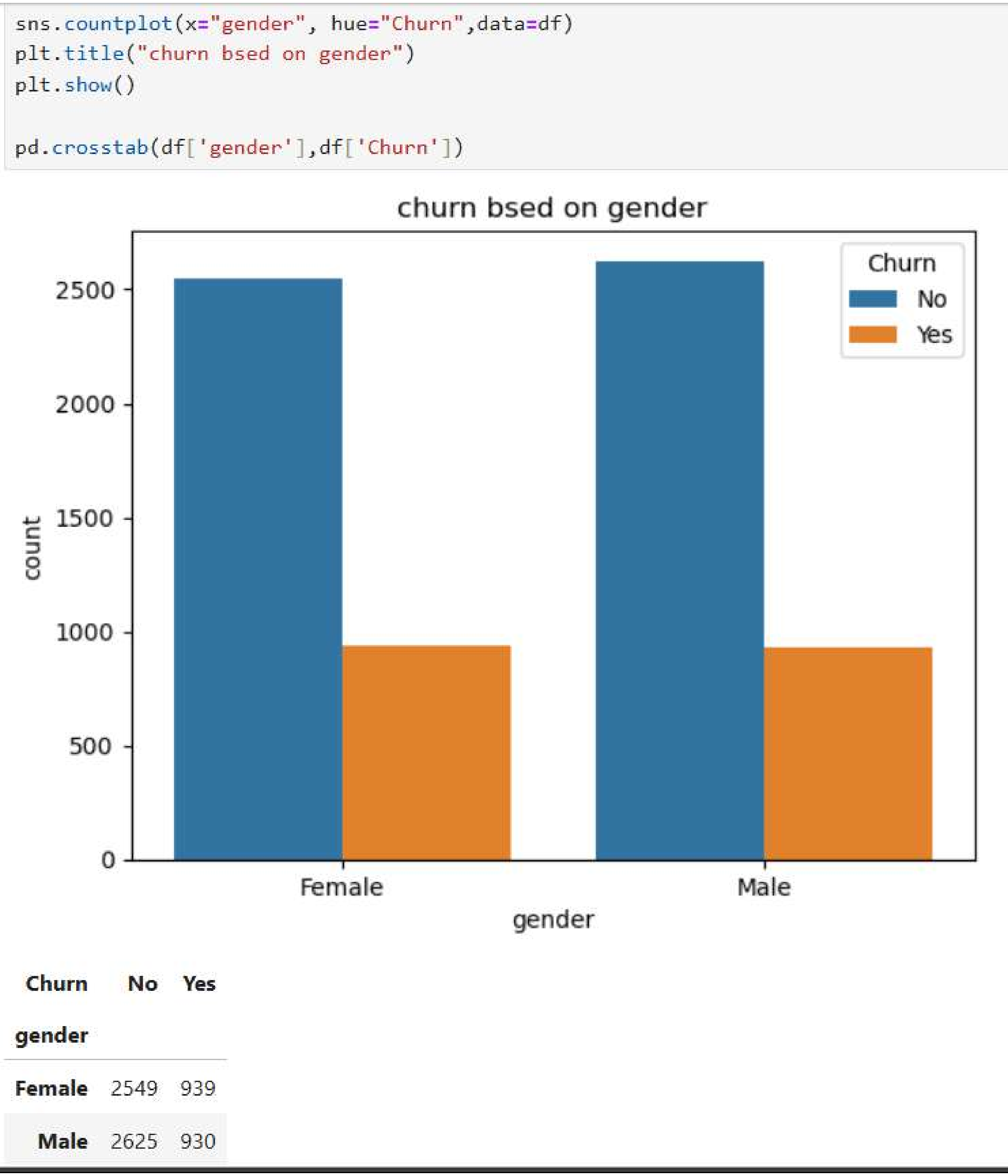
```
Churn
No     73.463013
Yes    26.536987
```



```
feature='Contract'
sns.countplot(x=feature,hue='Churn',data=df)
plt.title('churn by (feature)')
plt.xticks(rotation=30)
plt.show()
```

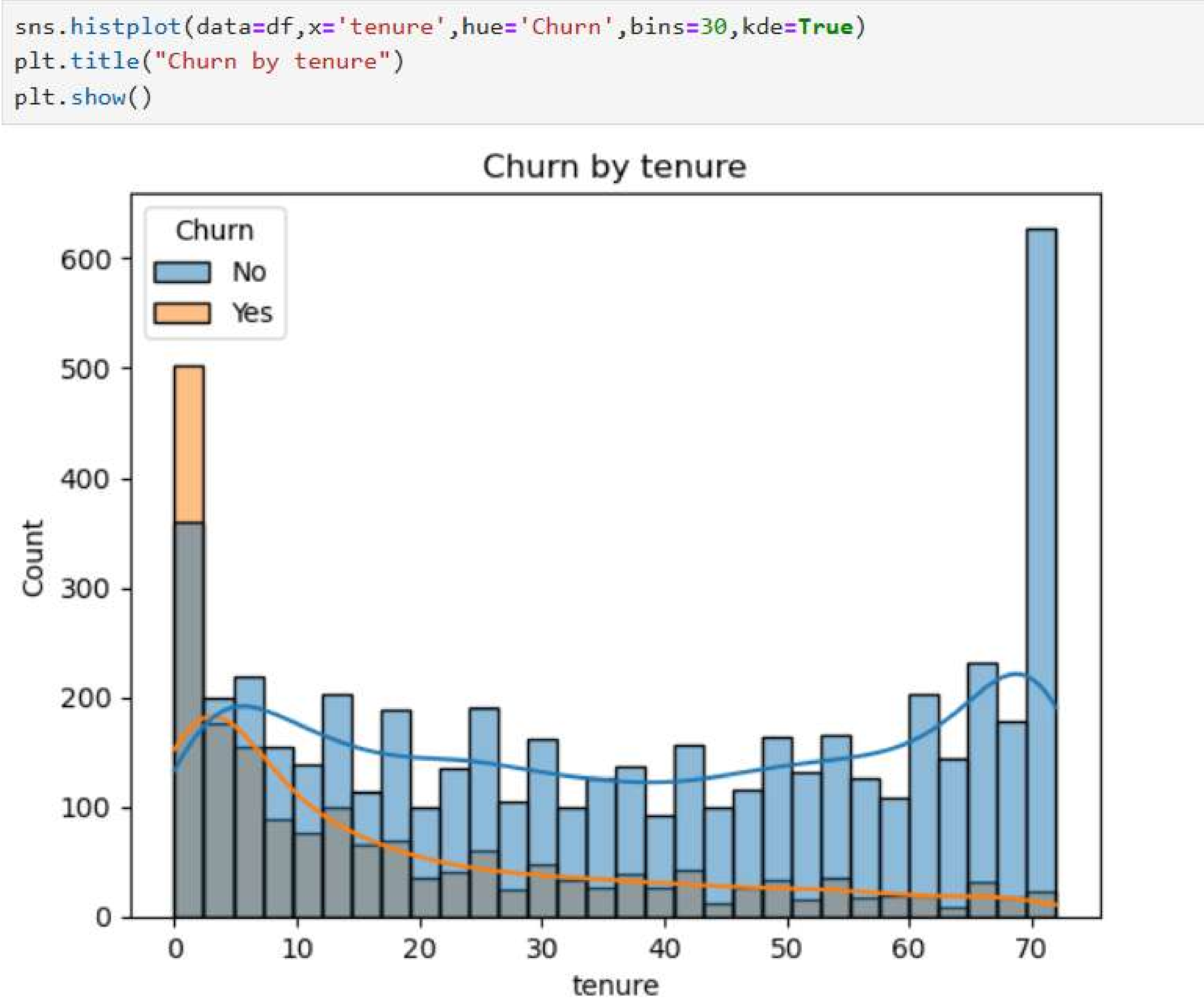


```
sns.countplot(x='PaymentMethod',hue='Churn',data=df)
plt.title('churn by (feature)')
plt.xticks(rotation=30)
plt.show()
pd.crosstab(df['PaymentMethod'],df['Churn'],normalize=True)*100
```

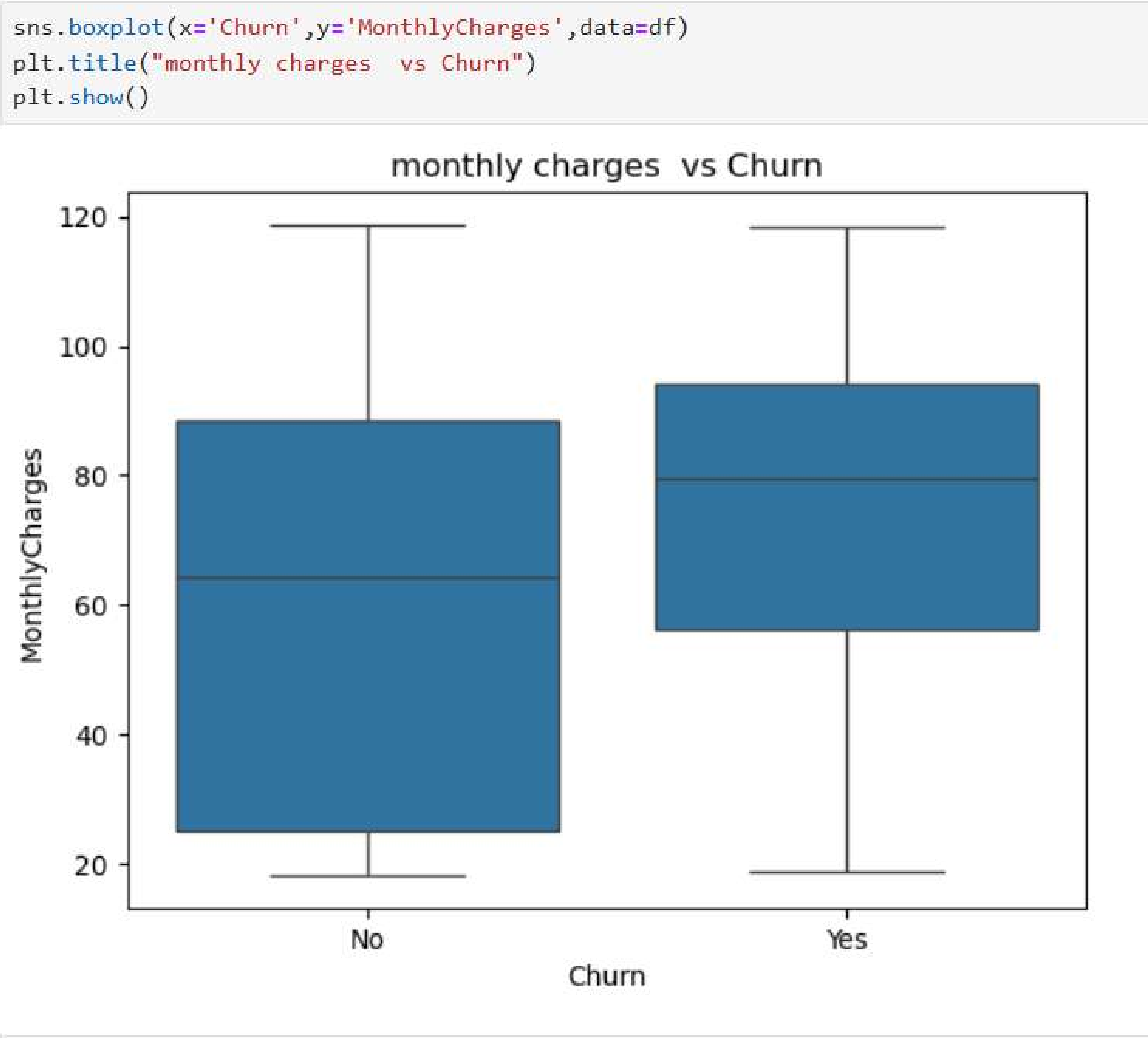


```
sns.countplot(x="gender", hue="Churn",data=df)
plt.title("churn bsed on gender")
plt.show()

pd.crosstab(df['gender'],df['Churn'])
```

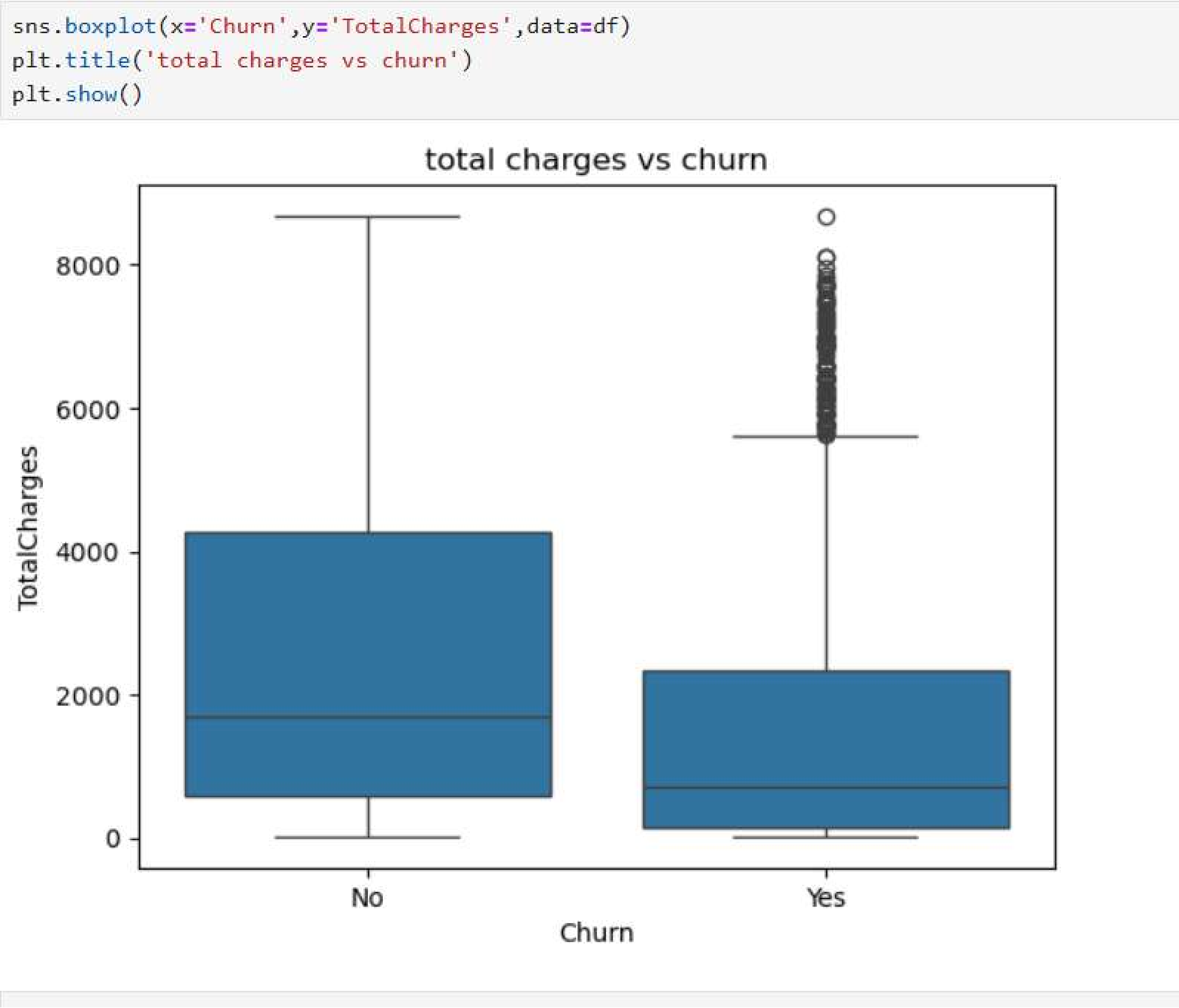
Churn	No	Yes
gender		
Female	2549	939
Male	2625	930



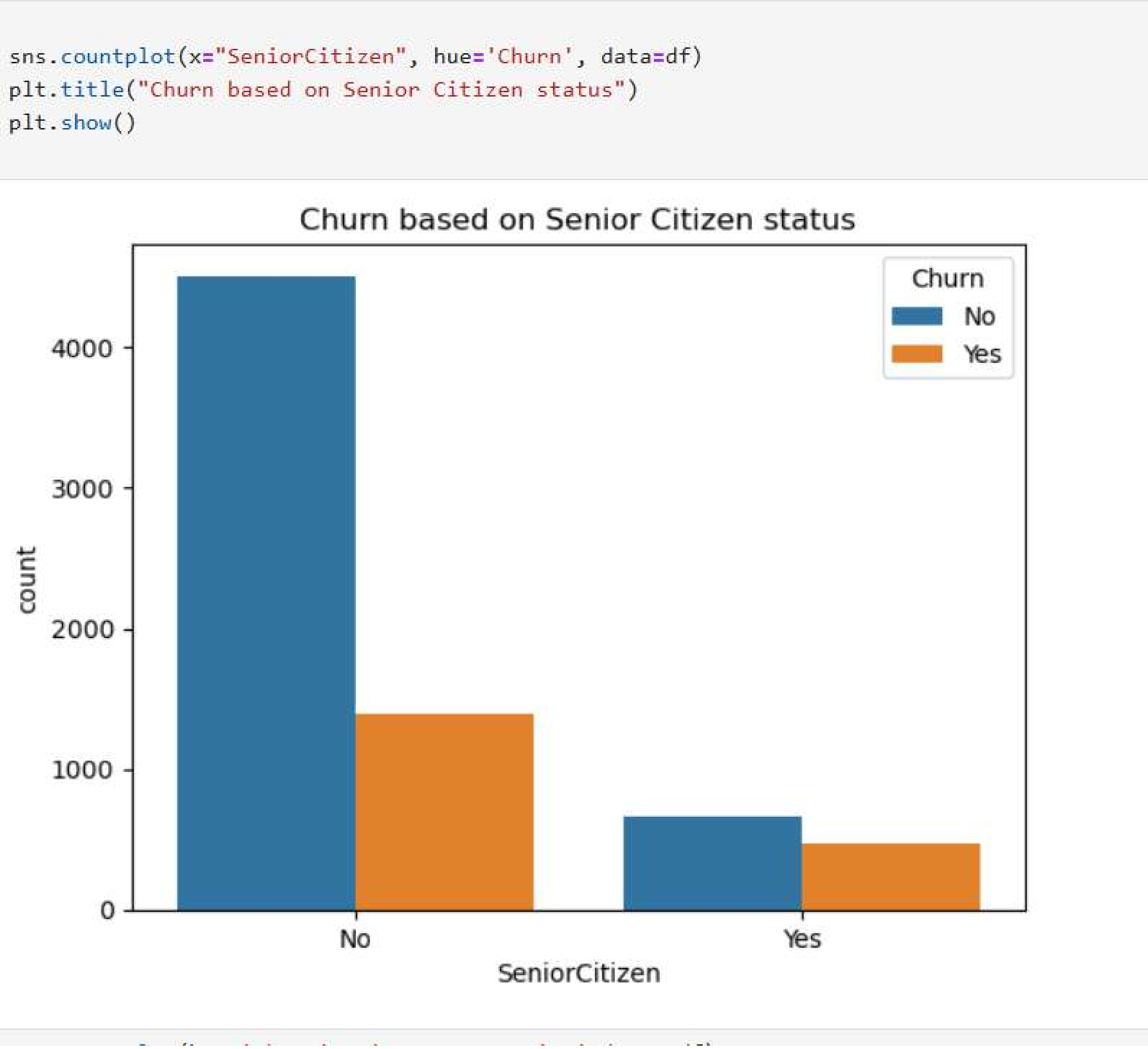
```
sns.histplot(data=df,x='tenure',hue='Churn',bins=30,kde=True)
plt.title("Churn by tenure")
plt.show()
```



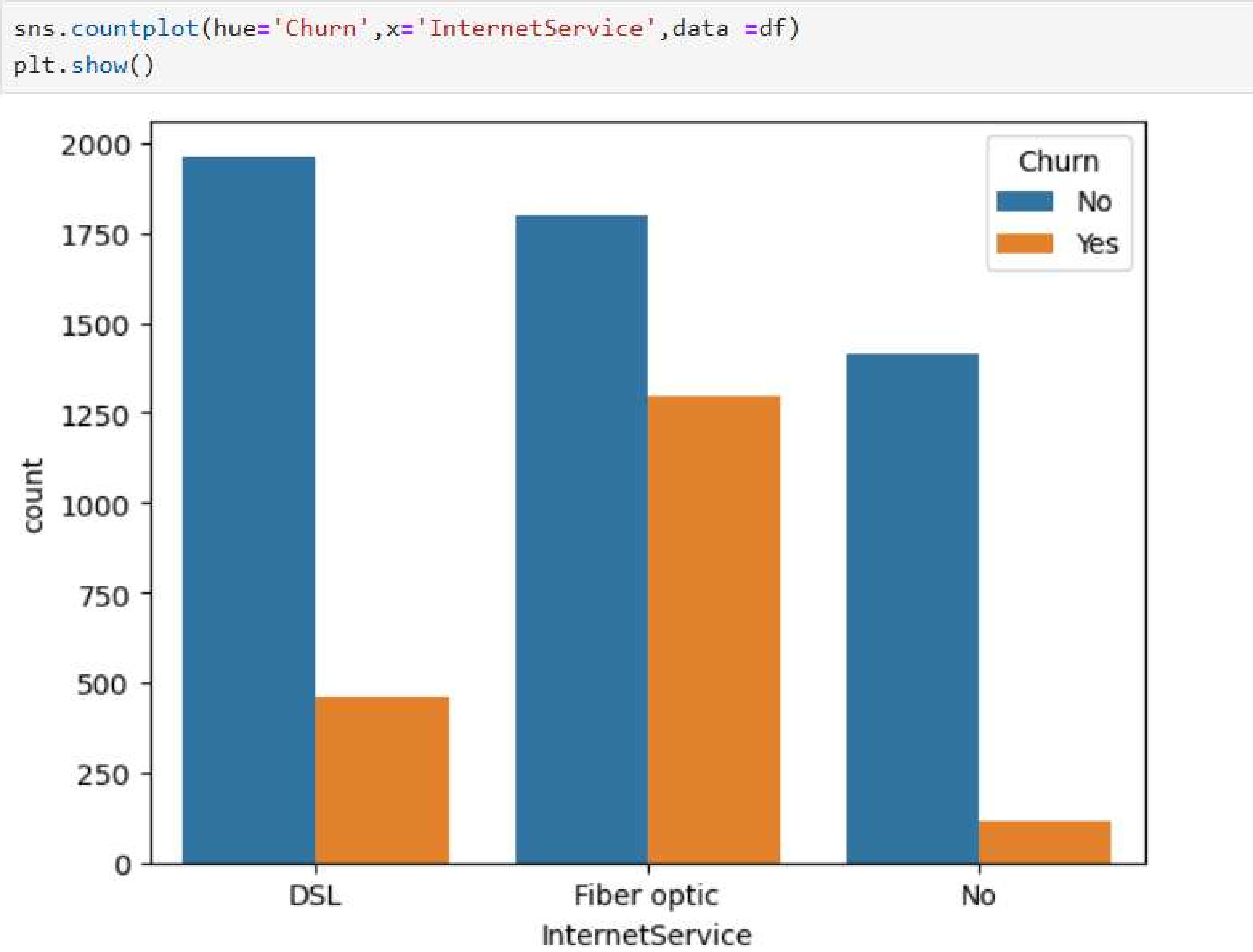
```
sns.boxplot(x='Churn',y='MonthlyCharges',data=df)
plt.title("monthly charges  vs Churn")
plt.show()
```



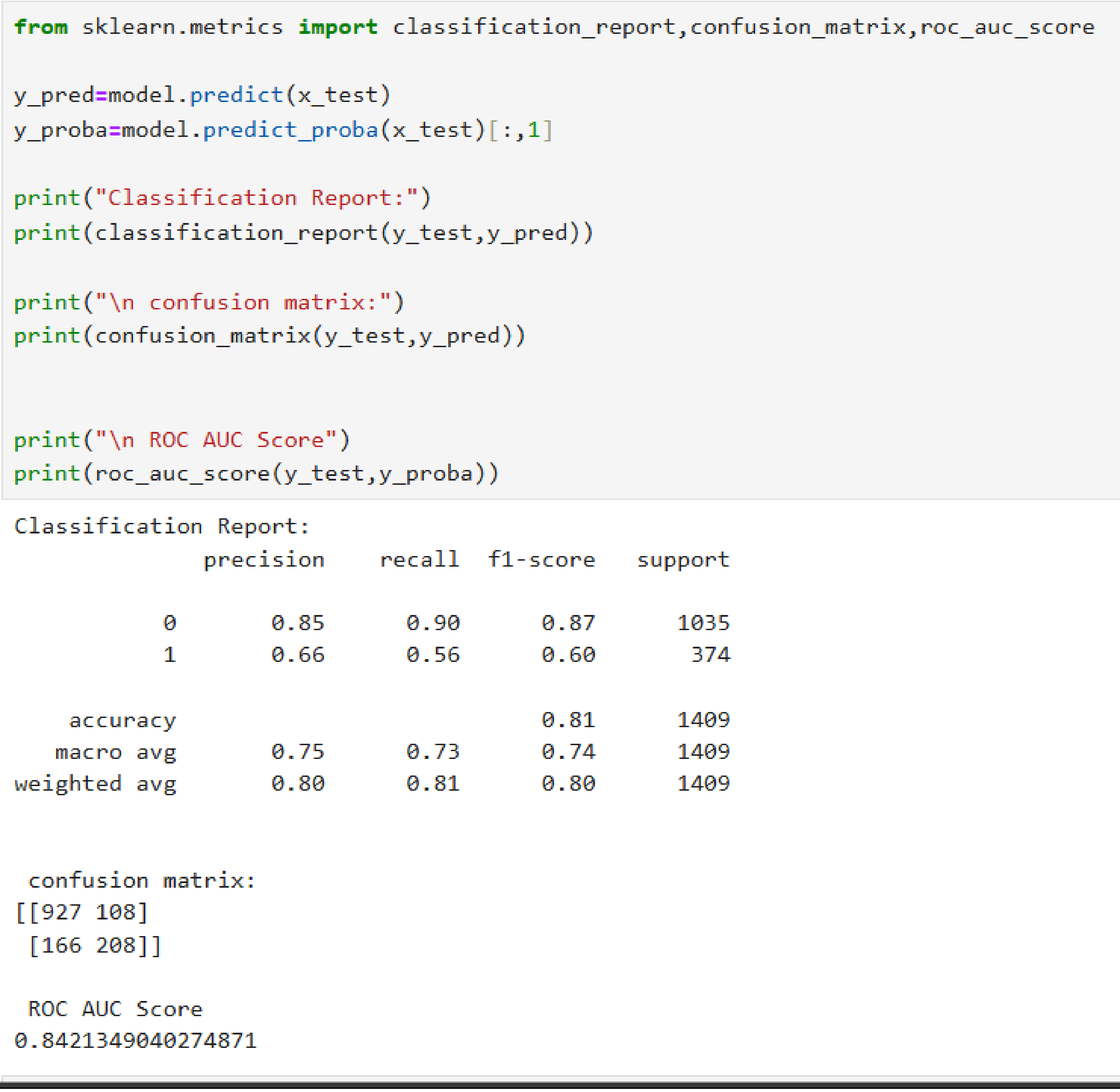
```
sns.boxplot(x='Churn',y='TotalCharges',data=df)
plt.title('total charges vs churn')
plt.show()
```



```
sns.countplot(x="SeniorCitizen", hue='Churn', data=df)
plt.title("Churn based on Senior Citizen status")
plt.show()
```

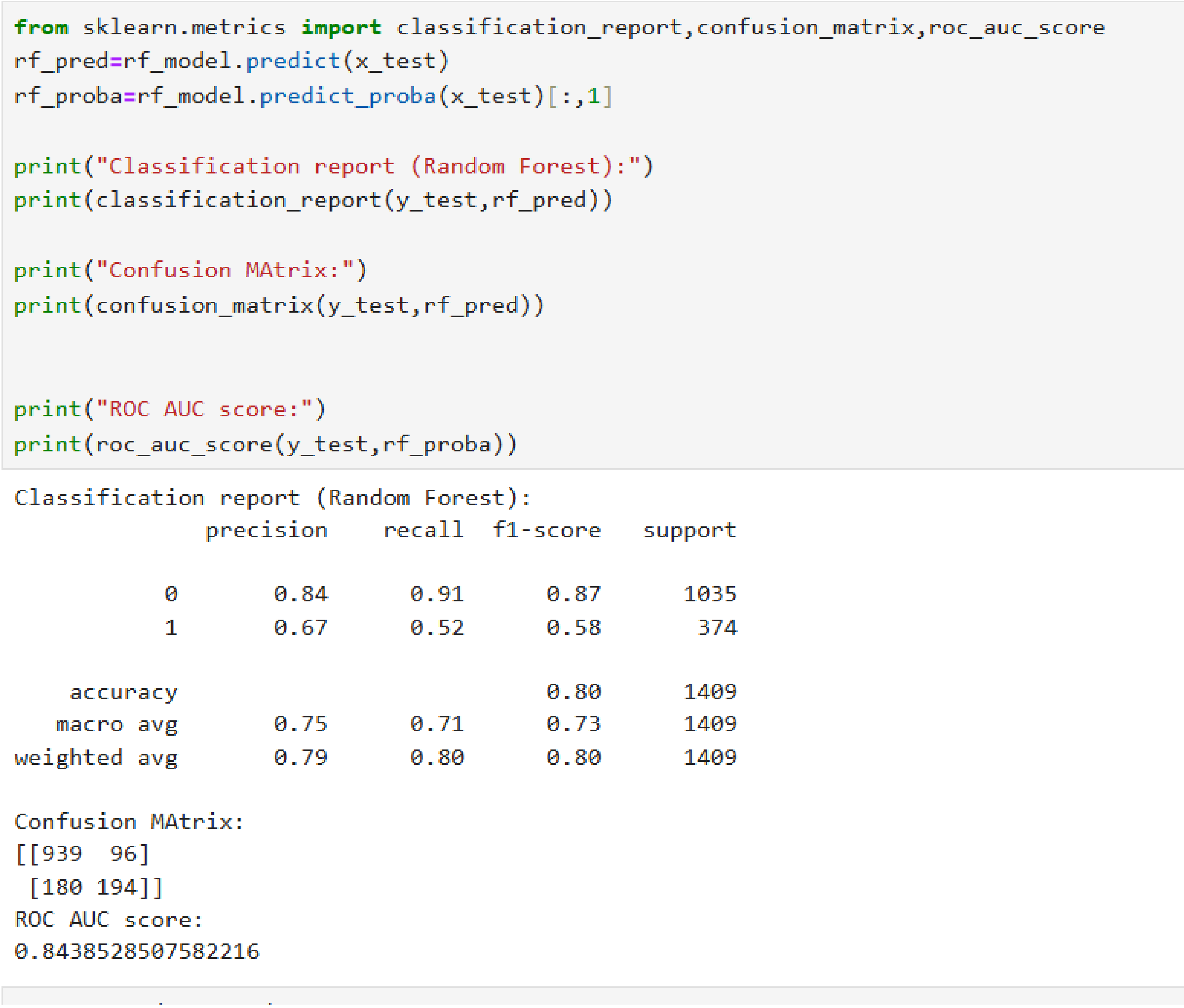


```
sns.countplot(hue='Churn',x='InternetService',data =df)
plt.show()
```



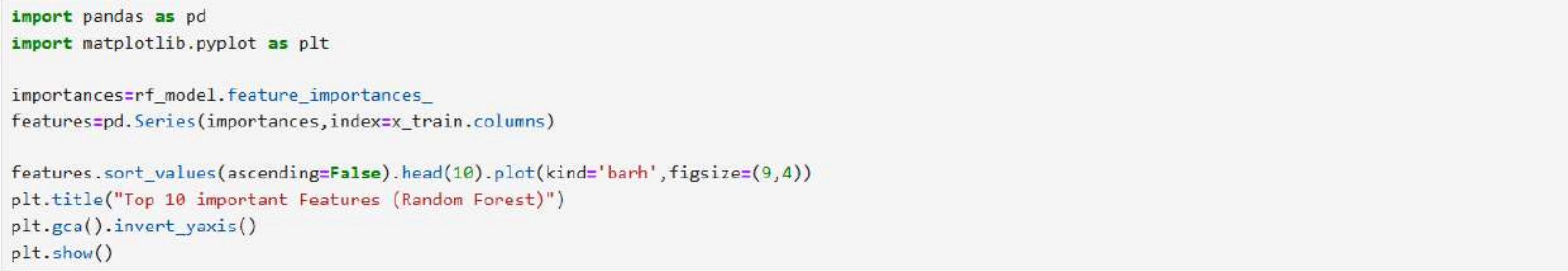
```python
from sklearn.metrics import classification_report,confusion_matrix,roc_auc_score

y_pred=model.predict(x_test)
y_proba=model.predict_proba(x_test)[:,1]

print("Classification Report:")
print(classification_report(y_test,y_pred))

print("\n confusion matrix:")
print(confusion_matrix(y_test,y_pred))



print("\n ROC AUC Score")
print(roc_auc_score(y_test,y_proba))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.90      0.87      1035
           1       0.66      0.56      0.60       374

    accuracy                           0.81      1409
   macro avg       0.75      0.73      0.74      1409
weighted avg       0.80      0.81      0.80      1409


 confusion matrix:
[[927 108]
 [166 208]]

 ROC AUC Score
0.8421349040274871
```

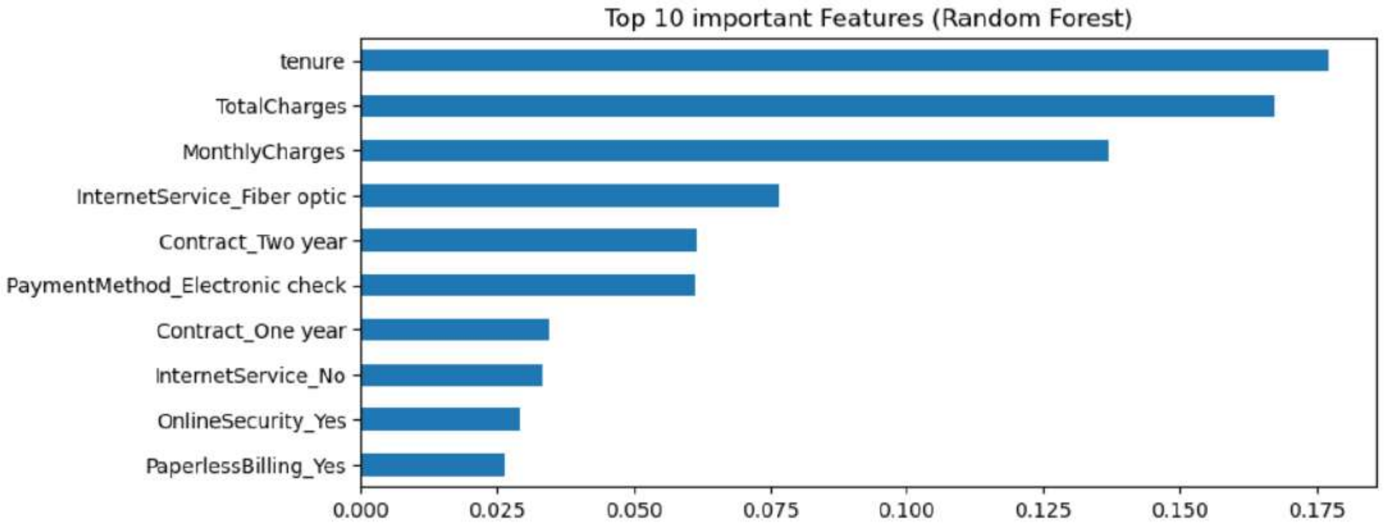


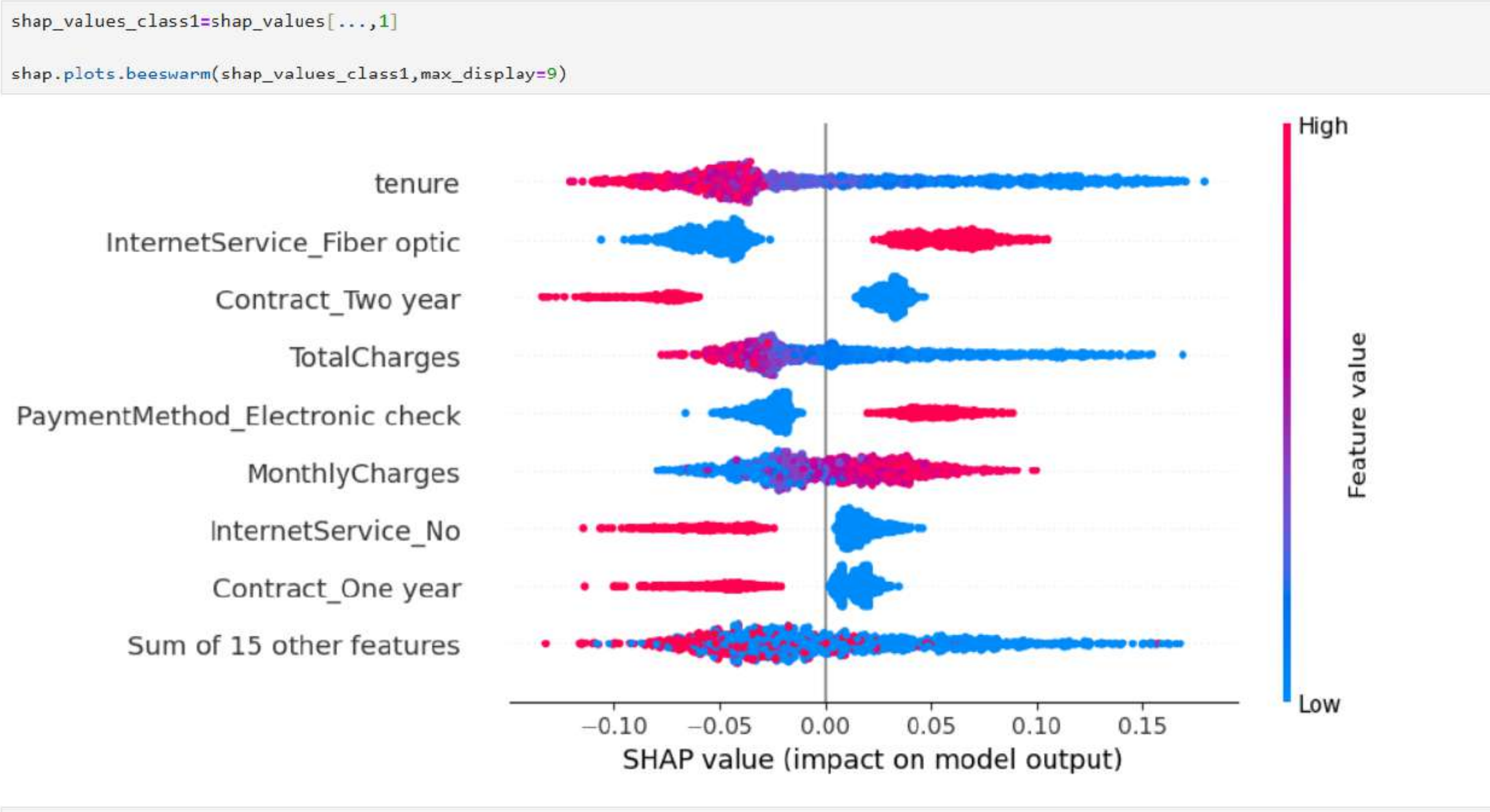
```python
from sklearn.metrics import classification_report,confusion_matrix,roc_auc_score
rf_pred=rf_model.predict(x_test)
rf_proba=rf_model.predict_proba(x_test)[:,1]

print("Classification report (Random Forest):")
print(classification_report(y_test,rf_pred))

print("Confusion MAtrix:")
print(confusion_matrix(y_test,rf_pred))



print("ROC AUC score:")
print(roc_auc_score(y_test,rf_proba))
```

```
Classification report (Random Forest):
              precision    recall  f1-score   support

           0       0.84      0.91      0.87      1035
           1       0.67      0.52      0.58       374

    accuracy                           0.80      1409
   macro avg       0.75      0.71      0.73      1409
weighted avg       0.79      0.80      0.80      1409

Confusion MAtrix:
[[939  96]
 [180 194]]
ROC AUC score:
0.8438528507582216
```

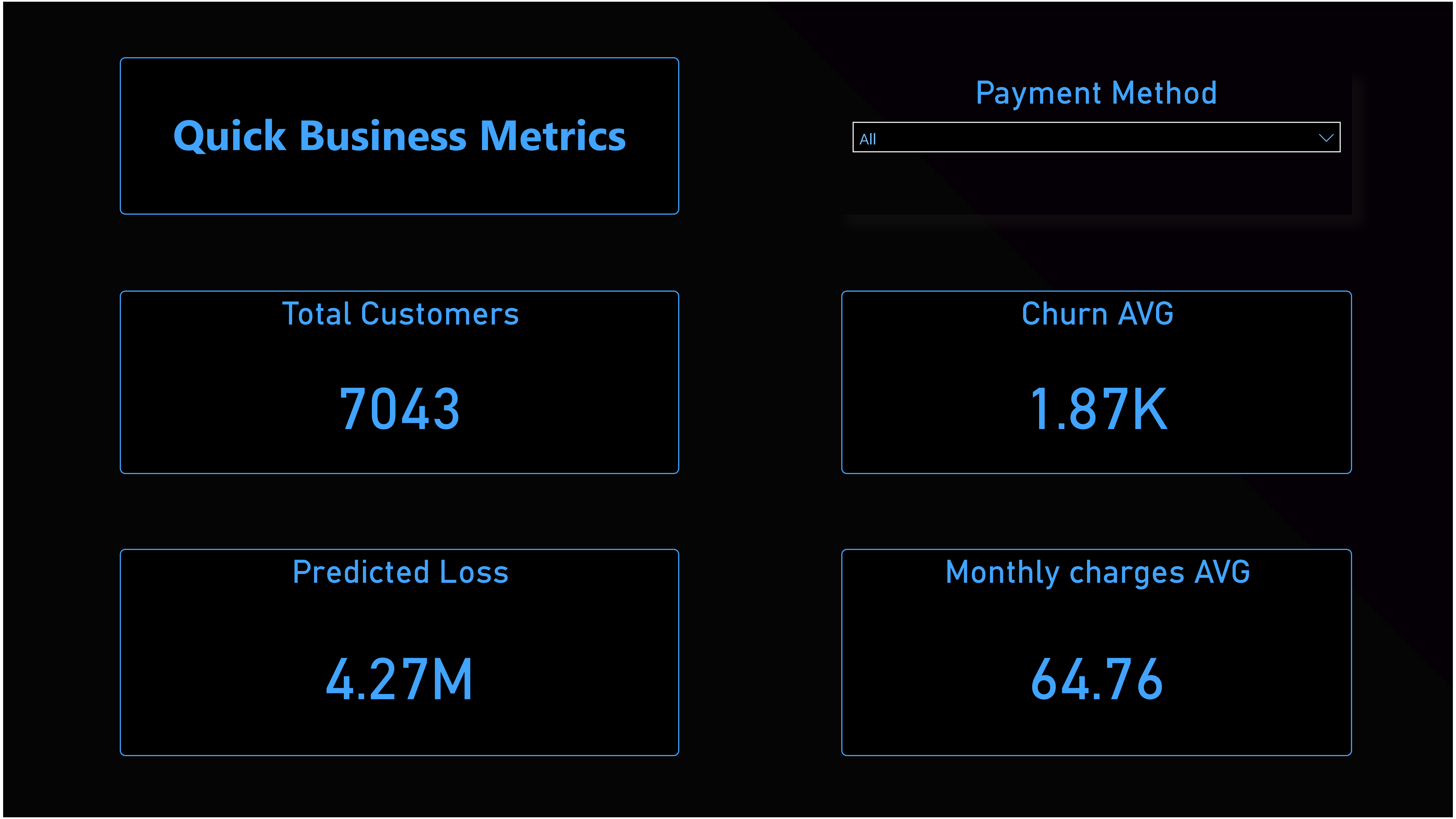


```
import pandas as pd
import matplotlib.pyplot as plt

importances=rf_model.feature_importances_
features=pd.Series(importances,index=x_train.columns)

features.sort_values(ascending=False).head(10).plot(kind='barh',figsize=(9,4))
plt.title("Top 10 important Features (Random Forest)")
plt.gca().invert_yaxis()
plt.show()
```





```
shap_values_class1=shap_values[...,1]

shap.plots.beeswarm(shap_values_class1,max_display=9)
```



Quick Business Metrics

Payment Method

All

Total Customers

7043

Churn AVG

1.87K

Predicted Loss

4.27M

Monthly charges AVG

64.76

Root Cause Analysis:

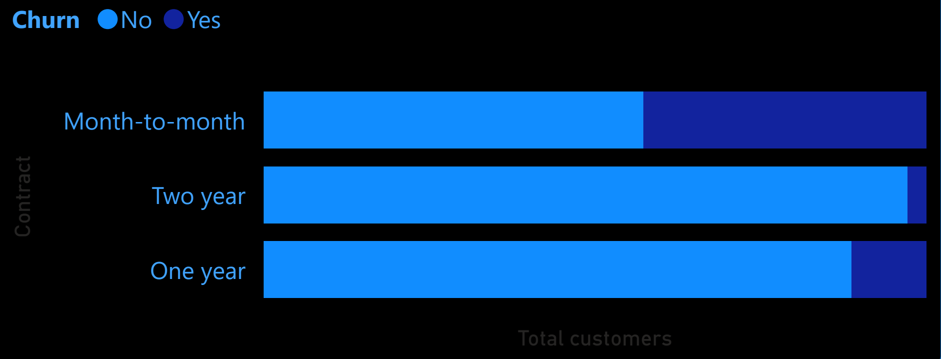
Senior Citizen

All

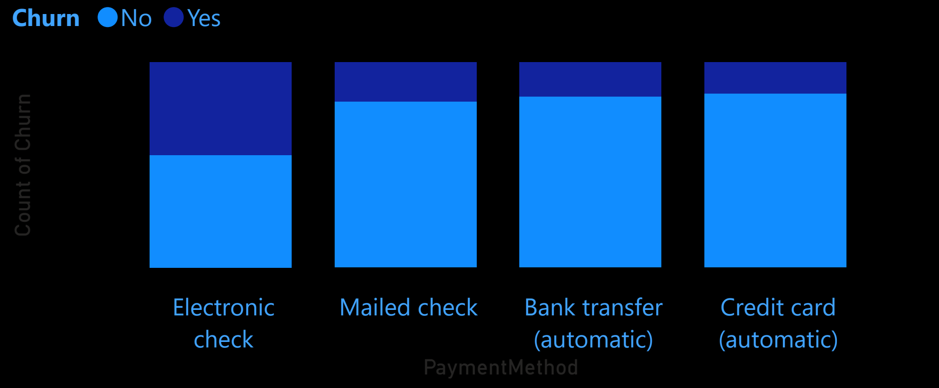
Tenure

All

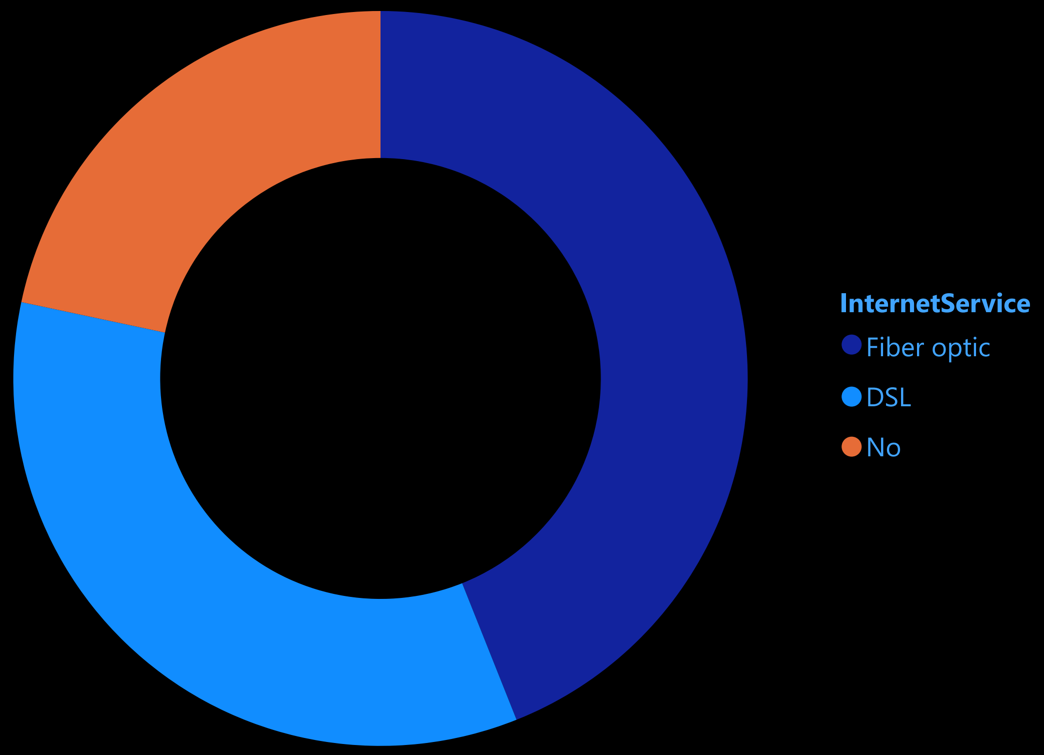
Churn by Contract Type



Count of Churn by PaymentMethod and Churn



contribution of churn by internetservices



SOLUTION SUMMARY

Conversion rate

0.60

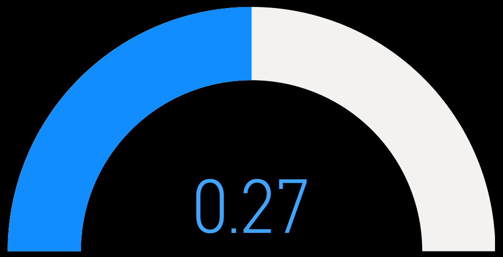
Tenure

12

switch to credit card

0.50

Churn Rate



Simulated churn rate