INTRODUCTION TO MACHINE LEARNING PROJECT PROGRESS REPORT

ONUR HAKTAN
GÖKSU TURAC

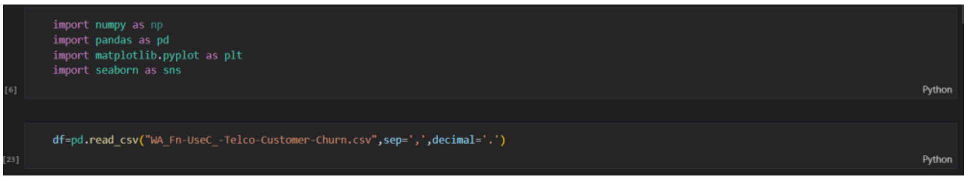
30.11.2022

ESKISEHIR TECHNICAL UNIVERSITY

ANALYSIS OF THE COMPANY'S CUSTOMER LOSS

We have reviewed the data set and finished the preprocessing process.

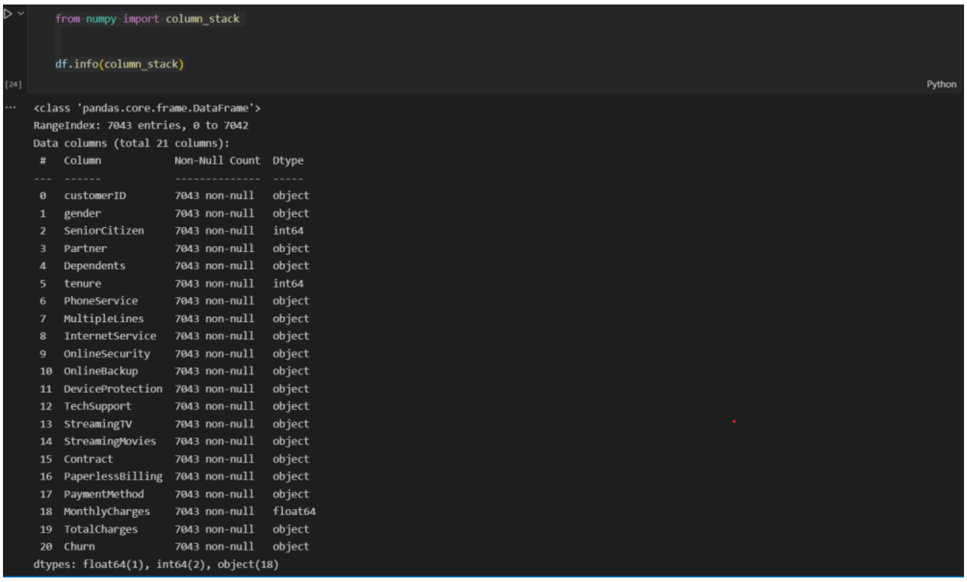
- We took the data set for the lost customer analysis from Kaggle. Then we read the csv file(dataset).



```
[6]
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[23]
df=pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv",sep=',',decimal='.')
```

- A customer data set consisting of 21 characteristics was provided for use in the analysis, and the target variable of these attributes was defined as "Churn".

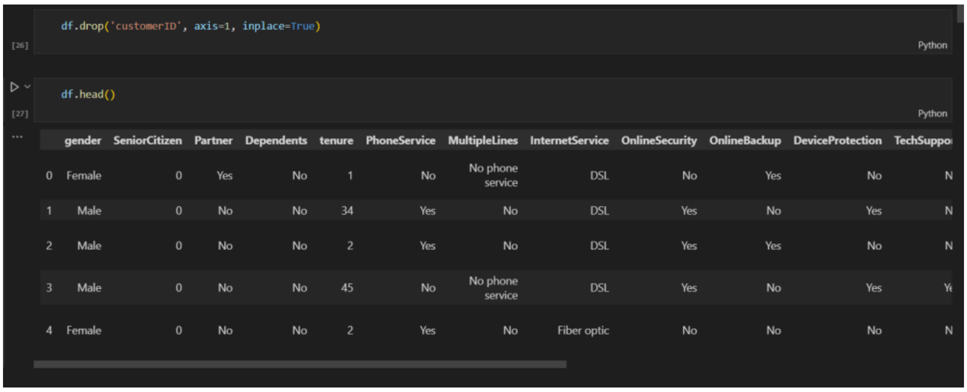


```
[24]
from numpy import column_stack


df.info(column_stack)
```

```
<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)
```

- We examined the data in the data set and extracted the data that was not useful to us.(Customer ID)

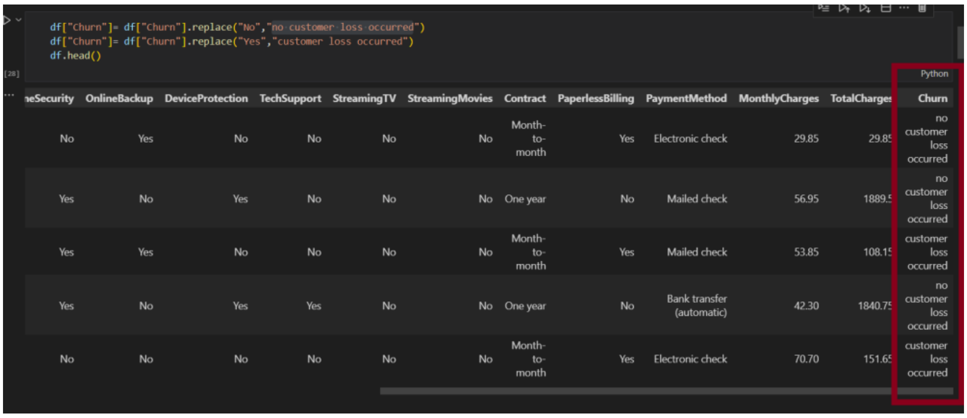


```
[26]
df.drop('customerID', axis=1, inplace=True)
```

```
[27]
df.head()
```

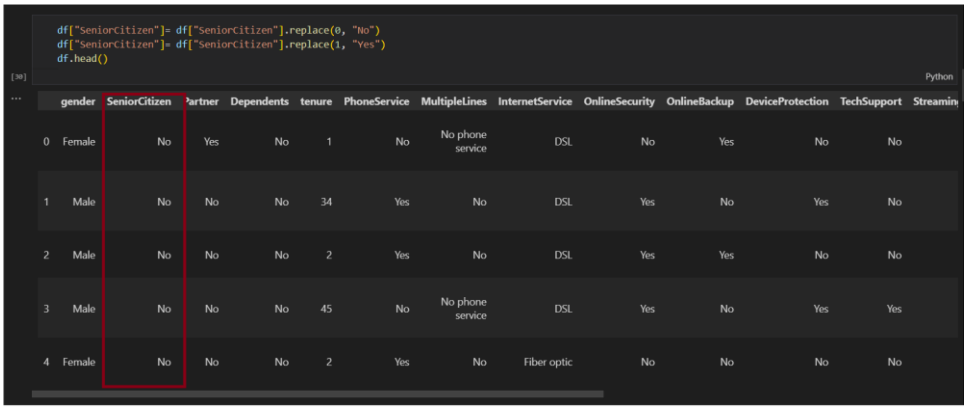
	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Ye
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	N

- We have made changes that make it easier for us to do analysis.



```
[28]
df["Churn"]= df["Churn"].replace("No","no customer loss occurred")
df["Churn"]= df["Churn"].replace("Yes","customer loss occurred")
df.head()
```

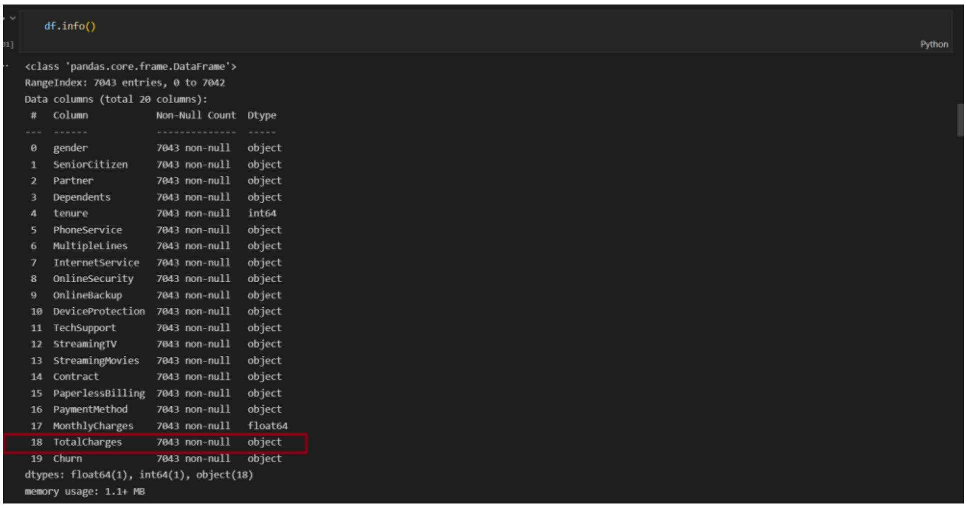
eSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No	Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	no customer loss occurred
Yes	No	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	no customer loss occurred
Yes	Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	customer loss occurred
Yes	No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	no customer loss occurred
No	No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	customer loss occurred



```
[30]
df["SeniorCitizen"]= df["SeniorCitizen"].replace(0, "No")
df["SeniorCitizen"]= df["SeniorCitizen"].replace(1, "Yes")
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Streamin
0	Female	No	Yes	No	1	No	No phone service	DSL	No	Yes	No	No	
1	Male	No	No	No	34	Yes	No	DSL	Yes	No	Yes	No	
2	Male	No	No	No	2	Yes	No	DSL	Yes	Yes	No	No	
3	Male	No	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes	
4	Female	No	No	No	2	Yes	No	Fiber optic	No	No	No	No	

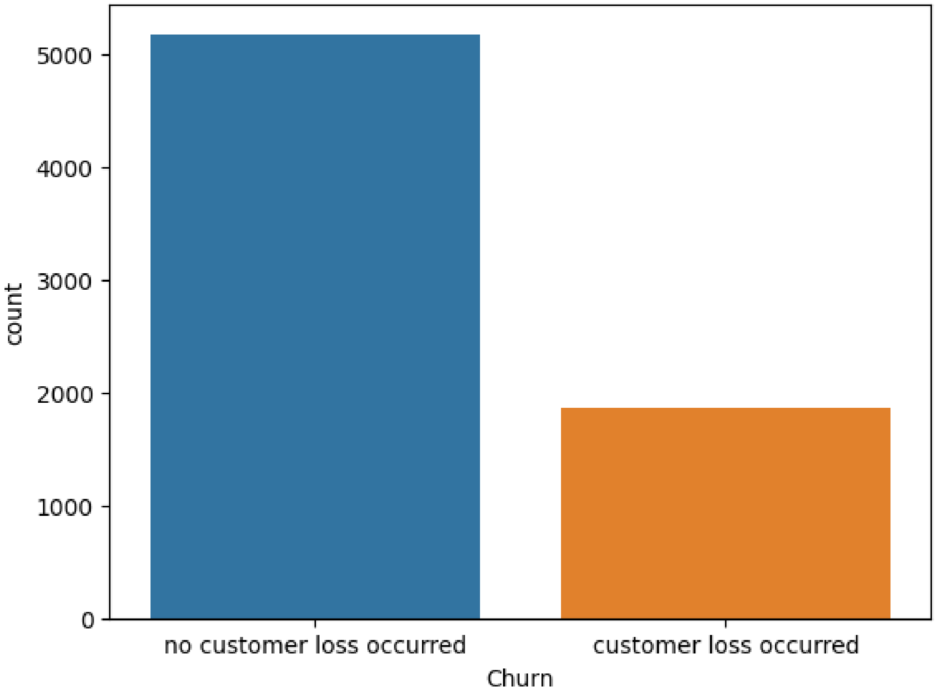
- We decided that we need to change the type of the TotalCharges feature.



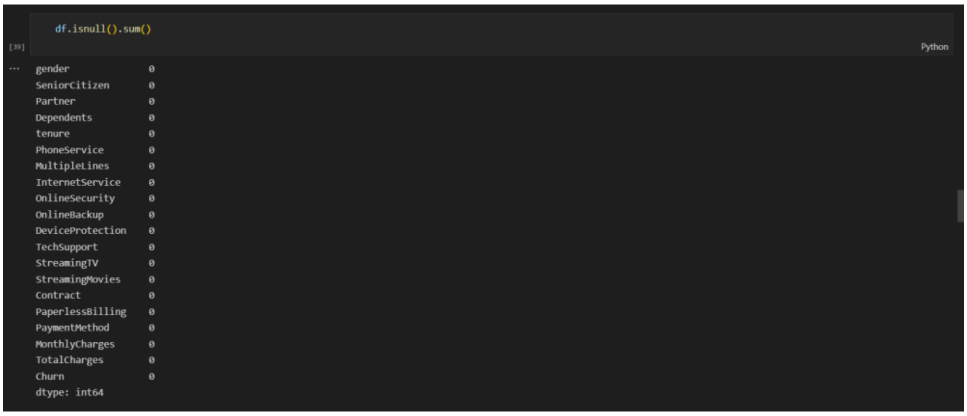
```
df.info()
```

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

- We have drawn a graph of customer loss.



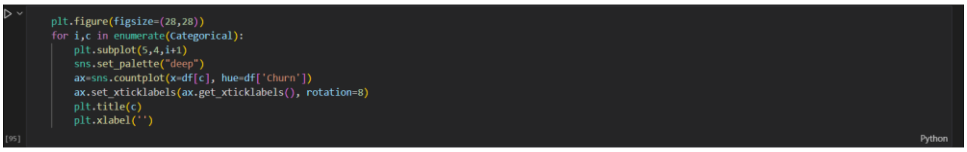
- We checked for lost data.



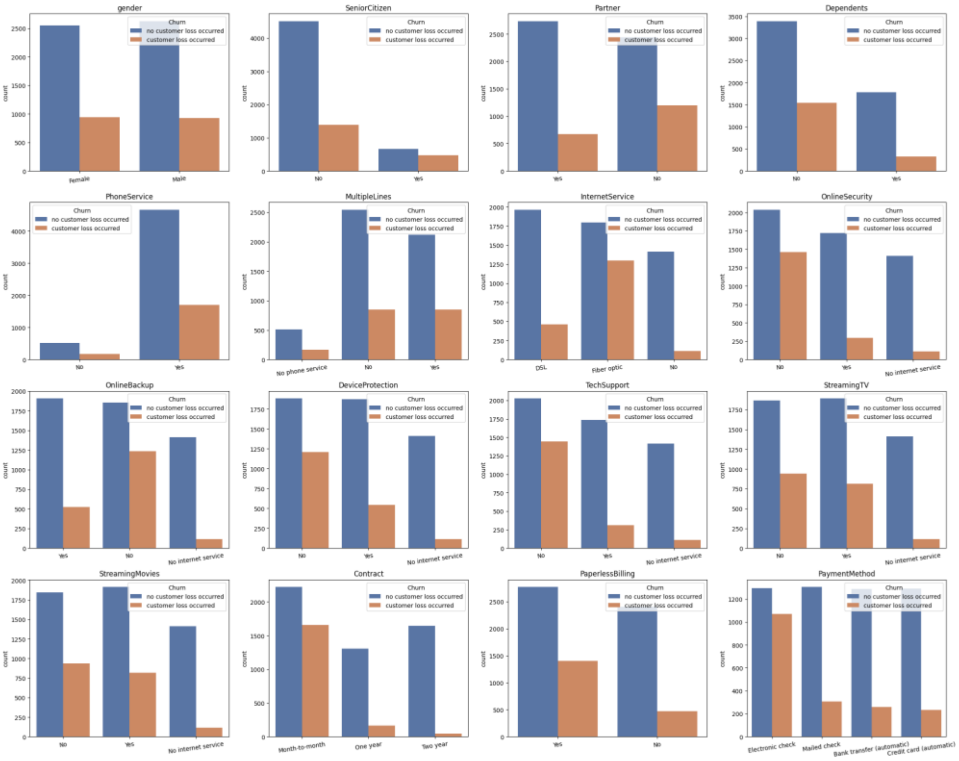
```
df.isnull().sum()
```

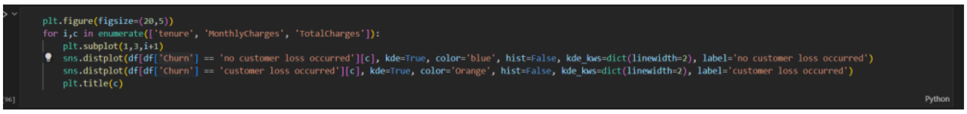
```
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

- In order to understand how the variables affect the target value, we had them plotted.

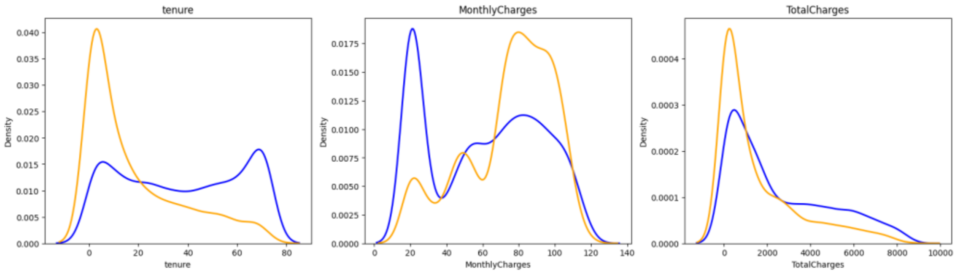


```
plt.figure(figsize=(28,28))
for i,c in enumerate(categorical):
    plt.subplot(5,4,i+1)
    sns.set_palette("deep")
    ax=sns.countplot(x=df[c], hue=df['Churn'])
    ax.set_xticklabels(ax.get_xticklabels(), rotation=8)
    plt.title(c)
    plt.xlabel('')
```

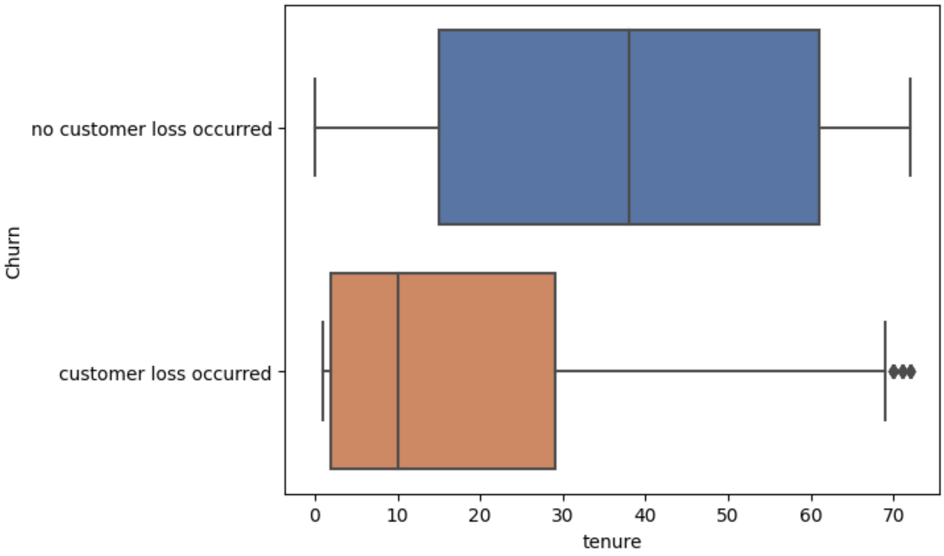


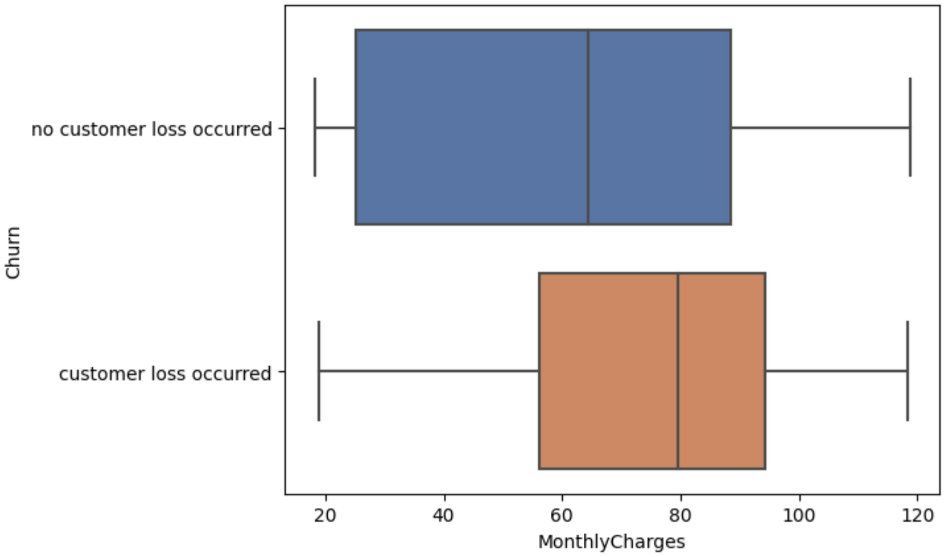


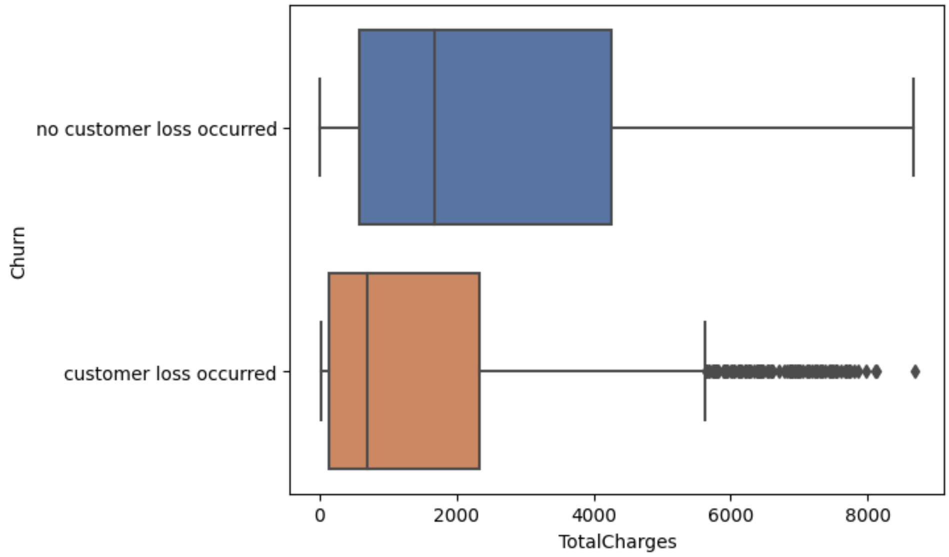
```python
plt.figure(figsize=(20,5))
for i,c in enumerate(['tenure', 'MonthlyCharges', 'TotalCharges']):
    plt.subplot(1,3,i+1)
    sns.distplot(df[df['Churn'] == 'no customer loss occurred'][c], kde=True, color='blue', hist=False, kde_kws=dict(linewidth=2), label='no customer loss occurred')
    sns.distplot(df[df['Churn'] == 'customer loss occurred'][c], kde=True, color='Orange', hist=False, kde_kws=dict(linewidth=2), label='customer loss occurred')
    plt.title(c)
```



- We looked to see if there were any outlier data.







In conclusion; In the part of the Total Charges that resulted in customer loss, we saw some outlier data. There are some outlier data also appears in the tenure. We have cleared the outlier data from the dataset with LabelEncoder. In the next part, we will divide the final dataset into two (test and train) and perform machine learning.