


```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline
```

```
from google.colab import files
uploaded = files.upload()
```




Choose Files

 No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving customer_churn.csv to customer_churn.csv

```
import pandas as pd
df=pd.read_csv("customer_churn.csv")
print(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.dtypes
```

↗

	0
gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
Contract	object
PaperlessBilling	object
PaymentMethod	object
MonthlyCharges	float64
TotalCharges	object
Churn	object

dtype: object

df.TotalCharges.values

↗ array(['29.85', '1889.5', '108.15', ..., '346.45', '306.6', '6844.5'],
dtype=object)

pd.to_numeric(df.TotalCharges,errors='coerce').isnull()

↗

	TotalCharges
0	False
1	False
2	False
3	False
4	False
...	...
7038	False
7039	False
7040	False
7041	False
7042	False

7043 rows × 1 columns

dtype: bool

df[pd.to_numeric(df.TotalCharges,errors='coerce').isnull()]

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	No
753	Male	0	No	Yes	0	Yes	No	No	No internet service	No internet service
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	Yes
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service
1340	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	Yes
3331	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service
3826	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service
4380	Female	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service
5218	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	Yes
6754	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	Yes

df.shape

(7043, 20)

df.iloc[488].TotalCharges

' '

df[df.TotalCharges!=' '].shape

(7032, 20)

df1 = df[df.TotalCharges!=' ']
df1.shape

(7032, 20)

df1.dtypes



0

gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
Contract	object
PaperlessBilling	object
PaymentMethod	object
MonthlyCharges	float64
TotalCharges	object
Churn	object

dtype: object

df1.TotalCharges.values



```
array(['29.85', '1889.5', '108.15', ..., '346.45', '306.6', '6844.5'],  
      dtype=object)
```

df1[df1.Churn=='No']



	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes
1	Male	0	No	No	34	Yes	No	DSL	Yes	No
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No
6	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes
7	Female	0	No	No	10	No	No phone service	DSL	Yes	No
...
7037	Female	0	No	No	72	Yes	No	No	No internet service	No internet service
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	No
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	No

5163 rows × 20 columns

```
tenure_churn_no = df1[df1.Churn=='No'].tenure  
tenure_churn_yes = df1[df1.Churn=='Yes'].tenure
```

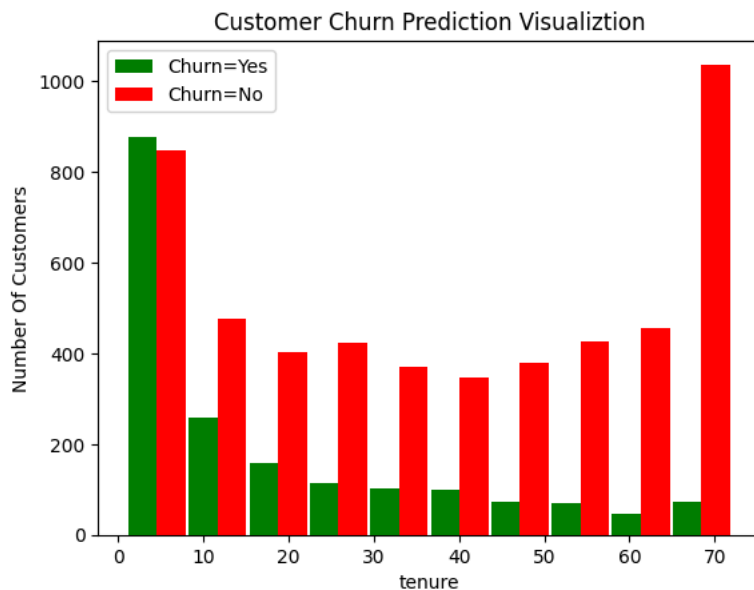
```
plt.xlabel("tenure")  
plt.ylabel("Number Of Customers")
```

```
plt.title("Customer Churn Prediction Visualiztion")

blood_sugar_men = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
blood_sugar_women = [67, 98, 89, 120, 133, 150, 84, 69, 89, 79, 120, 112, 100]

plt.hist([tenure_churn_yes, tenure_churn_no], rwidth=0.95, color=['green','red'],label=['Churn=Yes','Churn=No'])
plt.legend()
```

 <matplotlib.legend.Legend at 0x7e432593dad0>



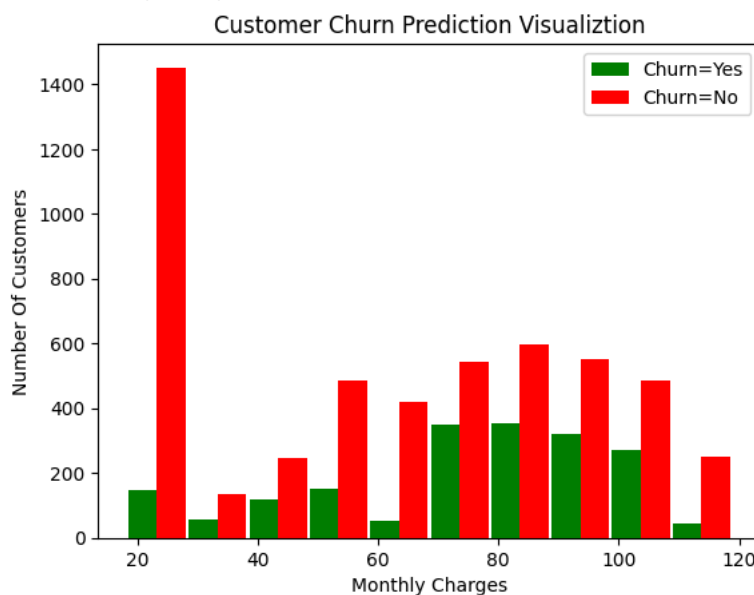
```
mc_churn_no = df1[df1.Churn=='No'].MonthlyCharges
mc_churn_yes = df1[df1.Churn=='Yes'].MonthlyCharges
```

```
plt.xlabel("Monthly Charges")
plt.ylabel("Number Of Customers")
plt.title("Customer Churn Prediction Visualiztion")
```


```
blood_sugar_men = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
blood_sugar_women = [67, 98, 89, 120, 133, 150, 84, 69, 89, 79, 120, 112, 100]
```

```
plt.hist([mc_churn_yes, mc_churn_no], rwidth=0.95, color=['green','red'],label=['Churn=Yes','Churn=No'])
plt.legend()
```

 <matplotlib.legend.Legend at 0x7e432598ee10>



```
def print_unique_col_values(df):
    for column in df:
        if df[column].dtypes=='object':
            print(f'{column}: {df[column].unique()}')
print_unique_col_values(df1)
```

 gender: ['Female' 'Male']
 Partner: ['Yes' 'No']
 Dependents: ['No' 'Yes']

```

PhoneService: ['No' 'Yes']
MultipleLines: ['No phone service' 'No' 'Yes']
InternetService: ['DSL' 'Fiber optic' 'No']
OnlineSecurity: ['No' 'Yes' 'No internet service']
OnlineBackup: ['Yes' 'No' 'No internet service']
DeviceProtection: ['No' 'Yes' 'No internet service']
TechSupport: ['No' 'Yes' 'No internet service']
StreamingTV: ['No' 'Yes' 'No internet service']
StreamingMovies: ['No' 'Yes' 'No internet service']
Contract: ['Month-to-month' 'One year' 'Two year']
PaperlessBilling: ['Yes' 'No']
PaymentMethod: ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
'Credit card (automatic)']
TotalCharges: ['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']
Churn: ['No' 'Yes']

```

```
print_unique_col_values(df1)
```

```

gender: ['Female' 'Male']
Partner: ['Yes' 'No']
Dependents: ['No' 'Yes']
PhoneService: ['No' 'Yes']
MultipleLines: ['No phone service' 'No' 'Yes']
InternetService: ['DSL' 'Fiber optic' 'No']
OnlineSecurity: ['No' 'Yes' 'No internet service']
OnlineBackup: ['Yes' 'No' 'No internet service']
DeviceProtection: ['No' 'Yes' 'No internet service']
TechSupport: ['No' 'Yes' 'No internet service']
StreamingTV: ['No' 'Yes' 'No internet service']
StreamingMovies: ['No' 'Yes' 'No internet service']
Contract: ['Month-to-month' 'One year' 'Two year']
PaperlessBilling: ['Yes' 'No']
PaymentMethod: ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
'Credit card (automatic)']
TotalCharges: ['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']
Churn: ['No' 'Yes']

```

```

df2 = pd.get_dummies(data=df1, columns=['InternetService','Contract','PaymentMethod'])
df2.columns

```

```

Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
      'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup',
      'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
      'PaperlessBilling', 'MonthlyCharges', 'TotalCharges', 'Churn',
      'InternetService_DSL', 'InternetService_Fiber optic',
      'InternetService_No', 'Contract_Month-to-month', 'Contract_One year',
      'Contract_Two year', 'PaymentMethod_Bank transfer (automatic)',
      'PaymentMethod_Credit card (automatic)',
      'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check'],
      dtype='object')

```

```

df1['gender'].replace({'Female':1,'Male':0},inplace=True)
df1.gender.unique()

```

<ipython-input-35-310b481c2d5d>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]

```

df1['gender'].replace({'Female':1,'Male':0},inplace=True)
<ipython-input-35-310b481c2d5d>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ve
df1['gender'].replace({'Female':1,'Male':0},inplace=True)
<ipython-input-35-310b481c2d5d>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-array

```

df1['gender'].replace({'Female':1,'Male':0},inplace=True)
array([1, 0])

```

```

for col in df1:
    print(f'{col}: {df1[col].unique()}')

```

```

gender: [1 0]
SeniorCitizen: [0 1]
Partner: ['Yes' 'No']
Dependents: ['No' 'Yes']
tenure: [ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27
  5 46 11 70 63 43 15 60 18 66  9  3 31 50 64 56  7 42 35 48 29 65 38 68
 32 55 37 36 41  6  4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26 39]
PhoneService: ['No' 'Yes']
MultipleLines: ['No phone service' 'No' 'Yes']
InternetService: ['DSL' 'Fiber optic' 'No']
OnlineSecurity: ['No' 'Yes' 'No internet service']
OnlineBackup: ['Yes' 'No' 'No internet service']

```

```

DeviceProtection: ['No' 'Yes' 'No internet service']
TechSupport: ['No' 'Yes' 'No internet service']
StreamingTV: ['No' 'Yes' 'No internet service']
StreamingMovies: ['No' 'Yes' 'No internet service']
Contract: ['Month-to-month' 'One year' 'Two year']
PaperlessBilling: ['Yes' 'No']
PaymentMethod: ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
'Credit card (automatic)']
MonthlyCharges: [29.85 56.95 53.85 ... 63.1 44.2 78.7 ]
TotalCharges: ['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']
Churn: ['No' 'Yes']

```

```

df2 = pd.get_dummies(data=df1, columns=['InternetService','Contract','PaymentMethod'])
df2.columns

```

```

Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup',
'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
'PaperlessBilling', 'MonthlyCharges', 'TotalCharges', 'Churn',
'InternetService_DSL', 'InternetService_Fiber optic',
'InternetService_No', 'Contract_Month-to-month', 'Contract_One year',
'Contract_Two year', 'PaymentMethod_Bank transfer (automatic)',
'PaymentMethod_Credit card (automatic)',
'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check'],
dtype='object')

```

```
df2.sample(5)
```

```

Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'MonthlyCharges', 'TotalCharges', 'Churn', 'InternetService_DSL', 'InternetService_Fiber optic', 'InternetService_No', 'Contract_Month-to-month', 'Contract_One year', 'Contract_Two year', 'PaymentMethod_Bank transfer (automatic)', 'PaymentMethod_Credit card (automatic)', 'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check'], dtype='object')

```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	DeviceProtection
6912	0	0	No	No	9	Yes	Yes	No	No	N
3597	0	0	Yes	No	25	Yes	No	No	No	N
1762	0	0	Yes	Yes	35	Yes	Yes	No	Yes	N
5870	0	0	Yes	Yes	5	Yes	Yes	No internet service	No internet service	No internet service
2998	0	0	No	No	2	Yes	No	No	Yes	N

5 rows x 27 columns

```
df2.dtypes
```



0

gender	int64
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
PaperlessBilling	object
MonthlyCharges	float64
TotalCharges	object
Churn	object
InternetService_DSL	bool
InternetService_Fiber optic	bool
InternetService_No	bool
Contract_Month-to-month	bool
Contract_One year	bool
Contract_Two year	bool
PaymentMethod_Bank transfer (automatic)	bool
PaymentMethod_Credit card (automatic)	bool
PaymentMethod_Electronic check	bool
PaymentMethod_Mailed check	bool

dtvne: object

```
cols_to_scale = ['tenure', 'MonthlyCharges', 'TotalCharges']
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df2[cols_to_scale] = scaler.fit_transform(df2[cols_to_scale])
for col in df2:
    print(f'{col}: {df2[col].unique()}')
```



```
gender: [1 0]
SeniorCitizen: [0 1]
Partner: ['Yes' 'No']
Dependents: ['No' 'Yes']
tenure: [0.          0.46478873 0.01408451 0.61971831 0.09859155 0.29577465
 0.12676056 0.38028169 0.85915493 0.16901408 0.21126761 0.8028169
 0.67605634 0.33802817 0.95774648 0.71830986 0.98591549 0.28169014
 0.15492958 0.4084507  0.64788732 1.          0.22535211 0.36619718
 0.05633803 0.63380282 0.14084507 0.97183099 0.87323944 0.5915493
 0.1971831  0.83098592 0.23943662 0.91549296 0.11267606 0.02816901
 0.42253521 0.69014085 0.88732394 0.77464789 0.08450704 0.57746479
 0.47887324 0.66197183 0.3943662  0.90140845 0.52112676 0.94366197
 0.43661972 0.76056338 0.50704225 0.49295775 0.56338028 0.07042254
 0.04225352 0.45070423 0.92957746 0.30985915 0.78873239 0.84507042
 0.18309859 0.26760563 0.73239437 0.54929577 0.81690141 0.32394366
 0.6056338  0.25352113 0.74647887 0.70422535 0.35211268 0.53521127]
PhoneService: ['No' 'Yes']
MultipleLines: ['No phone service' 'No' 'Yes']
OnlineSecurity: ['No' 'Yes' 'No internet service']
OnlineBackup: ['Yes' 'No' 'No internet service']
DeviceProtection: ['No' 'Yes' 'No internet service']
TechSupport: ['No' 'Yes' 'No internet service']
StreamingTV: ['No' 'Yes' 'No internet service']
StreamingMovies: ['No' 'Yes' 'No internet service']
PaperlessBilling: ['Yes' 'No']
MonthlyCharges: [0.11542289 0.38507463 0.35422886 ... 0.44626866 0.25820896 0.60149254]
```



```
TotalCharges: [0.0012751 0.21586661 0.01031041 ... 0.03780868 0.03321025 0.78764136]
Churn: ['No' 'Yes']
InternetService_DSL: [ True False]
InternetService_Fiber optic: [False  True]
InternetService_No: [False  True]
Contract_Month-to-month: [ True False]
Contract_One year: [False  True]
Contract_Two year: [False  True]
PaymentMethod_Bank transfer (automatic): [False  True]
PaymentMethod_Credit card (automatic): [False  True]
PaymentMethod_Electronic check: [ True False]
PaymentMethod_Mailed check: [False  True]
```

```
X = df2.drop('Churn',axis='columns')
y = df2['Churn']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=5)
```

```
X_train.shape
```

```
(5625, 26)
```

```
X_test.shape
```

```
(1407, 26)
```

```
X_train[:10]
```

```
(5625, 26)
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	DeviceProtecti
5664	1	1	No	No	0.126761	Yes	No	No	No	
101	1	0	Yes	Yes	0.000000	Yes	No	No internet service	No internet service	No internet serv
2621	0	0	Yes	No	0.985915	Yes	No	No	Yes	
392	1	1	No	No	0.014085	Yes	No	No	No	
1327	0	0	Yes	No	0.816901	Yes	Yes	No	No	
3607	1	0	No	No	0.169014	Yes	No	Yes	No	
2773	0	0	Yes	No	0.323944	No	No phone service	No	No	
1936	1	0	Yes	No	0.704225	Yes	No	Yes	Yes	
5387	0	0	No	No	0.042254	No	No phone service	No	No	
4331	0	0	No	No	0.985915	Yes	Yes	No internet service	No internet service	No internet serv

10 rows × 26 columns

```
len(X_train.columns)
```

```
26
```