

Assignment NO.9

```
[1]: import pandas as pd #data manipulation
import numpy as np #numerical computations
from sklearn.model_selection import train_test_split
from sklearn import metrics #evaluating the performance of machine learning model

[2]: data = pd.read_csv("Telecom_Customer_Churn.csv")
print(data.index)

RangeIndex(start=0, stop=7043, step=1)

[3]: print(data.columns)

Index(['Customer ID', 'Gender', 'Age', 'Married', 'Number of Dependents',
      'City', 'Zip Code', 'Latitude', 'Longitude', 'Number of Referrals',
      'Tenure in Months', 'Offer', 'Phone Service',
      'Avg Monthly Long Distance Charges', 'Multiple Lines',
      'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
      'Online Security', 'Online Backup', 'Device Protection Plan',
      'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
      'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless Billing',
      'Payment Method', 'Monthly Charge', 'Total Charges', 'Total Refunds',
      'Total Extra Data Charges', 'Total Long Distance Charges',
      'Total Revenue', 'Customer Status', 'Churn Category', 'Churn Reason'],
      dtype='object')

[4]: data.shape

(7043, 38)

[5]: print(data.head())

   Customer ID  Gender  Age  Married  Number of Dependents  City \
0  0002-ORFBO  Female   37     Yes             0  Frazier Park
1  0003-MKNFE   Male   46     No             0    Glendale
2  0004-TLHLJ   Male   50     No             0   Costa Mesa
3  0011-IGKFF   Male   78     Yes             0    Martinez
4  0013-EXCHZ  Female   75     Yes             0   Camarillo

   Zip Code  Latitude  Longitude  Number of Referrals  ...  Payment Method \
0    93225  34.827662 -118.999073             2  ...    Credit Card
1    91206  34.162515 -118.203869             0  ...    Credit Card
2    92627  33.645672 -117.922613             0  ...  Bank Withdrawal
3    94553  38.014457 -122.115432             1  ...  Bank Withdrawal
4    93010  34.227846 -119.079903             3  ...    Credit Card

   Monthly Charge  Total Charges  Total Refunds  Total Extra Data Charges \
0             65.6           593.30           0.00                0
1             -4.0           542.40           38.33               10
2             73.9           280.85           0.00                0
3            98.0          1237.85           0.00                0
4            83.9           267.40           0.00                0

   Total Long Distance Charges  Total Revenue  Customer Status  Churn Category \
0                381.51           974.81         Stayed         NaN
1                96.21           610.28         Stayed         NaN
2               134.60           415.45         Churned    Competitor
3               361.66          1599.51         Churned  Dissatisfaction
4                22.14           289.54         Churned  Dissatisfaction

   Churn Reason
0           NaN
1           NaN
2  Competitor had better devices
3   Product dissatisfaction
4   Network reliability

[5 rows x 38 columns]
```

```
[8]: data.isna().sum()
```

```
[8]: Customer ID          0
      Gender              0
      Age                 0
      Married             0
      Number of Dependents 0
      City                0
      Zip Code            0
      Latitude            0
      Longitude           0
      Number of Referrals  0
      Tenure in Months    0
      Offer               0
      Phone Service       0
      Avg Monthly Long Distance Charges 682
      Multiple Lines      682
      Internet Service     0
      Internet Type       1526
      Avg Monthly GB Download 1526
      Online Security     1526
      Online Backup       1526
      Device Protection Plan 1526
      Premium Tech Support 1526
      Streaming TV        1526
      Streaming Movies    1526
      Streaming Music     1526
      Unlimited Data      1526
      Contract            0
      Paperless Billing    0
      Payment Method      0
      Monthly Charge      0
      Total Charges       0
      Total Refunds       0
      Total Extra Data Charges 0
      Total Long Distance Charges 0
      Total Revenue       0
      Customer Status     0
      Churn Category      5174
      Churn Reason        5174
      dtype: int64
```

```
[9]: data.isnull().sum()
```

```
[9]: Customer ID          0
      Gender             0
      Age               0
      Married           0
      Number of Dependents 0
      City              0
      Zip Code          0
      Latitude          0
      Longitude         0
      Number of Referrals 0
      Tenure in Months   0
      Offer             0
      Phone Service      0
      Avg Monthly Long Distance Charges 682
      Multiple Lines     682
      Internet Service    0
      Internet Type      1526
      Avg Monthly GB Download 1526
      Online Security     1526
      Online Backup       1526
      Device Protection Plan 1526
      Premium Tech Support 1526
      Streaming TV        1526
      Streaming Movies    1526
      Streaming Music     1526
      Unlimited Data      1526
      Contract            0
      Paperless Billing    0
      Payment Method      0
      Monthly Charge      0
      Total Charges       0
      Total Refunds       0
      Total Extra Data Charges 0
      Total Long Distance Charges 0
      Total Revenue       0
      Customer Status     0
      Churn Category      5174
      Churn Reason        5174
      dtype: int64
```

```
[10]: print("Number of rows before removing duplicates:", len(data))
```

Number of rows before removing duplicates: 7043

```
[11]: # Remove duplicate records
data_cleaned = data.drop_duplicates()
```

```
[12]: # Check the number of rows after removing duplicates
print("Number of rows after removing duplicates:", len(data_cleaned))
```

Number of rows after removing duplicates: 7043

```
[13]: data.describe()
```

```
[13]:
```

	Age	Number of Dependents	Zip Code	Latitude	Longitude	Number of Referrals	Tenure in Months	Avg Monthly Long Distance Charges	Avg Monthly GB Download	Monthly Charge	Total Charges	Total Refunds
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	6361.000000	5517.000000	7043.000000	7043.000000	7043.000000
mean	46.509726	0.468692	93486.070567	36.197455	-119.756684	1.951867	32.386767	25.420517	26.189958	63.596131	2280.381264	1.962182
std	16.750352	0.962802	1856.767505	2.468929	2.154425	3.001199	24.542061	14.200374	19.586585	31.204743	2266.220462	7.902614
min	19.000000	0.000000	90001.000000	32.555828	-124.301372	0.000000	1.000000	1.010000	2.000000	-10.000000	18.800000	0.000000
25%	32.000000	0.000000	92101.000000	33.990646	-121.788090	0.000000	9.000000	13.050000	13.000000	30.400000	400.150000	0.000000
50%	46.000000	0.000000	93518.000000	36.205465	-119.595293	0.000000	29.000000	25.690000	21.000000	70.050000	1394.550000	0.000000
75%	60.000000	0.000000	95329.000000	38.161321	-117.969795	3.000000	55.000000	37.680000	30.000000	89.750000	3786.600000	0.000000
max	80.000000	9.000000	96150.000000	41.962127	-114.192901	11.000000	72.000000	49.990000	85.000000	118.750000	8684.800000	49.790000

```
[15]: #Measure of frequency distribution
unique, counts = np.unique(data['Tenure in Months'], return_counts=True)
print(unique, counts)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72] [613 238 200 176 133 110 131 123 119 127  99 117 109  76  99  80  87  97
 73 71 63 90 85 94 79 79 72 57 72 72 65 69 64 65 88 50
65 59 56 64 70 65 65 51 61 74 68 64 66 68 68 80 70 68
64 80 65 67 60 76 76 70 72 80 76 89 98 100 95 119 170 362]
```

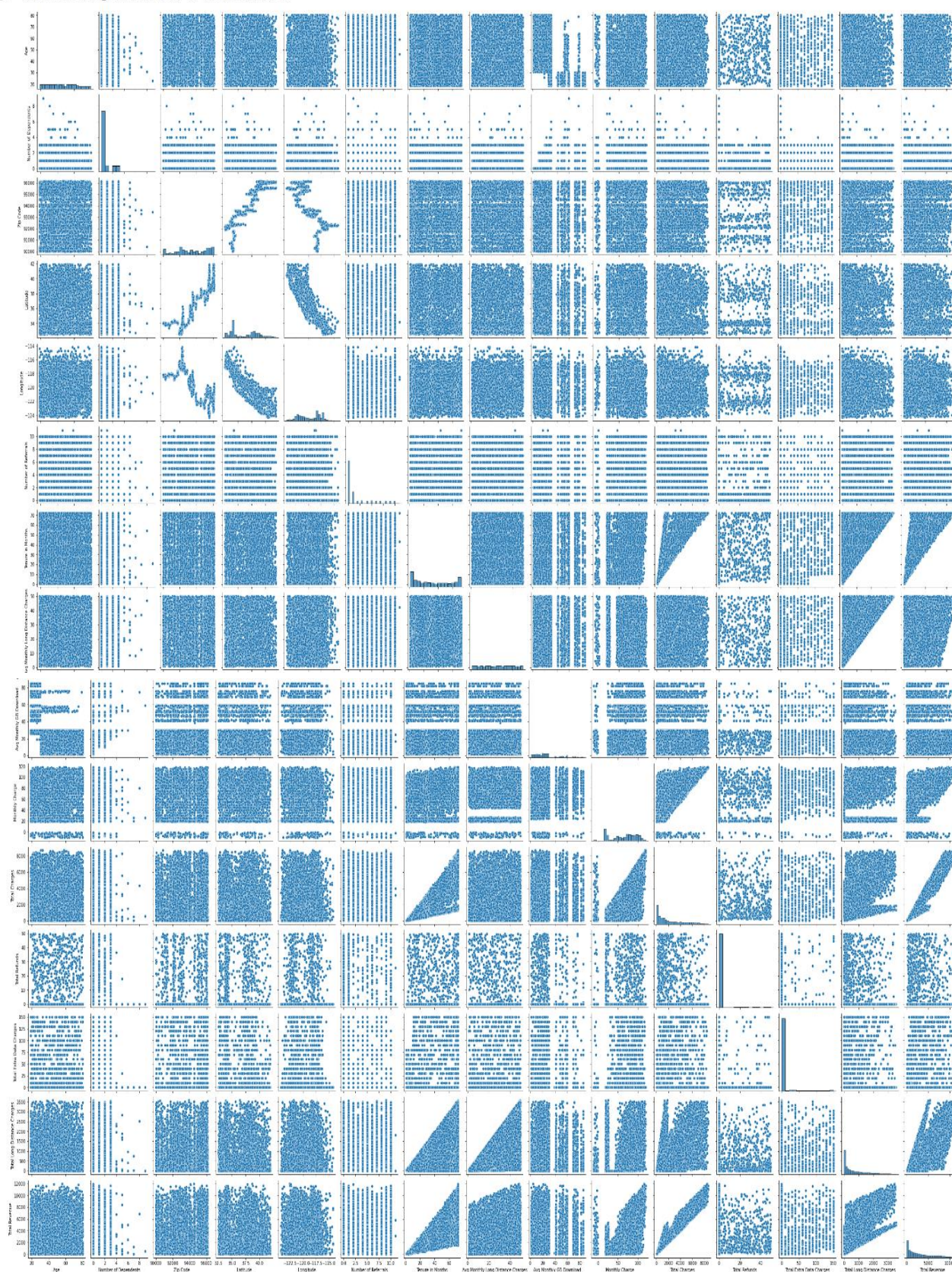
```
[17]: unique, counts = np.unique(data['Total Charges'], return_counts=True)
print(unique, counts)
```

```
[ 18.8   18.85   18.9   ... 8670.1  8672.45 8684.8 ] [1 2 1 ... 1 1 1]
```

```
[18]: import seaborn as sns #Seaborn library for data visualization
sns.pairplot(data)
```



```
[18]: <seaborn.axisgrid.PairGrid at 0x1df9e77f520>
```



```
[22]: X = data.drop("Total Revenue", axis=1)
      y = data["Total Revenue"]
      # Split the dataset into training and testing sets
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[23]: X_train.shape
```

```
[23]: (5634, 37)
```

```
[24]: y_train.shape
```

```
[24]: (5634,)
```

```
[25]: X_test.shape
```

```
[25]: (1409, 37)
```

```
[27]: y_test.shape
```

```
[27]: (1409,)
```

```
[28]: # Export the cleaned dataset to a CSV file
      data.to_csv("Cleaned_Telecom_Customer_Churn.csv", index=False)
```