Data Preprocessing of Telecom Churn Dataset

Atharv Dusane

200356

- **Data Cleaning:**

Finding and handling missing values-

```
customerID          0
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       11
Churn               0
dtype: int64
```

We find that Total charges is of object data type so we first convert it to numeric. Then we find that there are a total of 11 missing entries. The number of missing.

Total Charges are approximately equal to the monthly charges times the tenure. We now replace the missing values in Total Charges by Monthly charges * Tenure.

In the following example we can see that the missing value of total charges is successfully replaced by 0 (since the tenure was 0 months).

```
customerID                    4472-LVYGI
gender                            Female
SeniorCitizen                          0
Partner                              Yes
Dependents                           Yes
tenure                                 0
PhoneService                          No
MultipleLines            No phone service
InternetService                      DSL
OnlineSecurity                       Yes
OnlineBackup                          No
DeviceProtection                     Yes
TechSupport                          Yes
StreamingTV                          Yes
StreamingMovies                       No
Contract                        Two year
PaperlessBilling                     Yes
PaymentMethod      Bank transfer (automatic)
MonthlyCharges                     52.55
TotalCharges                         0.0
Churn                                 No
```

Finding outliers-

In the Monthly Charges and Total Charges column we are going to find outliers. For this we first find IQR (Q3 – Q1), then all the values lesser that Q1 - 1.5*IQR or greater than Q3 + 1.5*IQR are classified as outliers. In the given data we find that there are no outliers in Total Charges and Monthly charges column.

```
# Function to detect outliers

def detect_outliers(df,n,features):
    outlier_indices = []
    for col in features:

        Q1 = np.percentile(df[col], 25)
        Q3 = np.percentile(df[col],75)
        IQR = Q3 - Q1

        outlier_step = 1.5 * IQR
        outlier_list_col = df[(df[col] < Q1 - outlier_step) | (df[col] > Q3 + outlier_step )].index
        outlier_indices.extend(outlier_list_col)
    outlier_indices = Counter(outlier_indices)
    multiple_outliers = list( k for k, v in outlier_indices.items() if v > n )

    return multiple_outliers

Outliers = detect_outliers(tel_data, 0, ['TotalCharges', 'MonthlyCharges'])
print(Outliers)

# There are no outliers outside [Q1 - 1.5*IQR, 1.5*IQR + Q3] in Monthly Charges and Total Charges
```

```
[]
```

- **Data Integration:**

The data given is already in combined form.

Finding Correlation between having Phone service and Churn (Nominal data type)–

First a contingency table is made for having Phone Service and Churn. Then chi square test is applied to the data. We find that the χ2 value for this data is 0.915, the degrees of freedom is 1 and p-value is 0.339. p-value of less than 0.05 is required for rejecting the null hypothesis. Hence here the null hypothesis can't be rejected.

Contingency Table:

Churn	No	Yes
PhoneService		
No	512	170
Yes	4662	1699

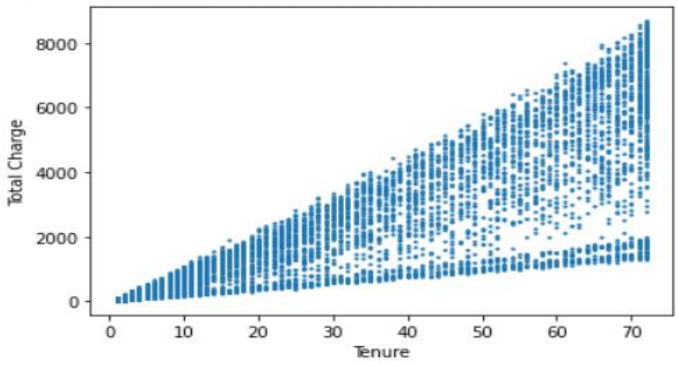
```
chi2_contingency(contingency_table)

# We can see that the chi square value is 0.915, p-value is 0.339 and degrees of freedom is three
# As p-value is not less than 0.05 we cannot reject the null hypothesis
```

```
(0.9150329892546948,
 0.3387825358066928,
 1,
```

Finding Correlation between Tenure and Total Charges (Numeric data type) –

The correlation obtained is 0.8261 between Tenure and Total Charges. This means that both the attributes are strongly positively correlated. From the scatter plot too we can see that they are positively correlated.

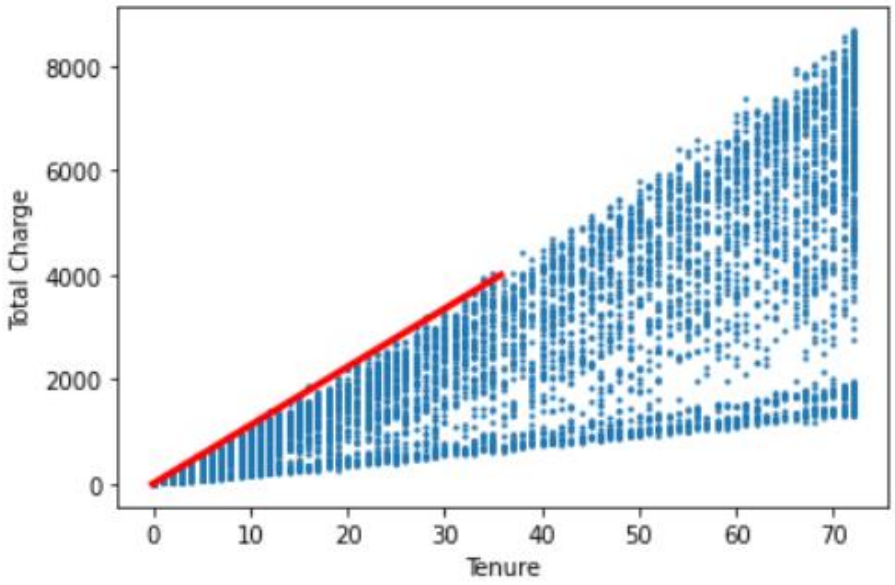
```
Correlation is 0.8258804609332024
```



- **Data Reduction:**

  Principal Component Analysis –

  The first principal component for the two columns tenure and total charge has been found out. Its components are `[0.00895117, 0.99995994]`
  The principal component vector has been plotted in the following graph (Red Line).



  Attribute Selection –

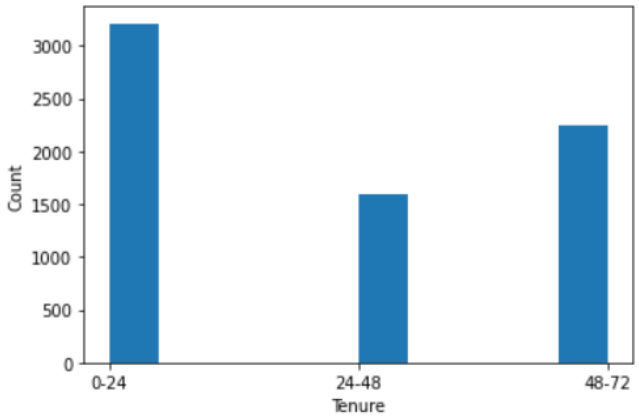
  Customer ID is an irrelevant attribute for our analysis so it is removed.

```
tel_data = tel_data.drop(['customerID'], axis=1)

# Column of customer id is dropped
```

Numerosity Reduction (Non parametric method Histogram) –

Tenure has been divided into 3 equal width bins for reducing numerosity.

```
# Dividing tenure into 3 bins for numerosity redustion
```



Numerosity Reduction (Sampling) –

From the given data 50% of data is randomly sampled without replacement.

```
tel_data_sample = tel_data.sample(frac=0.5, replace=False ,random_state=1)
print(tel_data_sample.shape)
tel_data_sample

# 50% data is sampled randomly without replacement
```

```
(3522, 21)
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	...
3381	Female	0	No	No	41	Yes	No	DSL	Yes	No	...
6180	Female	1	No	No	66	Yes	Yes	Fiber optic	Yes	No	...
4829	Female	0	No	No	12	Yes	No	DSL	No	No	...
3737	Female	0	No	No	5	Yes	Yes	DSL	No	No	...
4249	Female	0	Yes	Yes	10	Yes	No	DSL	No	Yes	...
...	...	...	...	...	...	...	...	...	...	...	...
3220	Male	0	Yes	No	70	Yes	Yes	DSL	Yes	No	...
5909	Female	0	No	No	52	Yes	Yes	Fiber optic	No	Yes	...
5734	Male	0	Yes	Yes	72	Yes	Yes	DSL	Yes	Yes	...

- **Data Transformation:**

Normalization –

Monthly Charges has been normalized by zscore method.

```
tel_data['MonthlyChargesNormalised']

# Monthly Charges has been normalised by zscore
```

```
0      -1.160323
1      -0.259629
2      -0.362660
3      -0.746535
4       0.197365
          ...
7038    0.665992
7039    1.277533
7040   -1.168632
7041    0.320338
7042    1.358961
Name: MonthlyChargesNormalised, Length: 7043, dtype: float64
```

Discretization –

Binning- Monthly Charges has been binned into 4 parts according to the quantiles they belong to.

```
tel_data['MonthlyCharges_Binned']
```

```
0         (18.24, 35.5]
1         (35.5, 70.35]
2         (35.5, 70.35]
3         (35.5, 70.35]
4        (70.35, 89.85]
             ...
7038     (70.35, 89.85]
7039    (89.85, 118.75]
7040      (18.24, 35.5]
7041     (70.35, 89.85]
7042    (89.85, 118.75]
```