**DATA CLEANING CODES**

|  |
| --- |
| import pandas as pd |
| # Info of column names along with the number of non –null values in each column,Categorical DATA TYPES  df = pd.read\_csv('churn\_raw\_data.csv')  df.info() |
| # Finding whether we have null values in the data  df.isnull().sum() |
| # TREATING MISSING VALUES  # EXAMINE THE DISTRIBUTION  df.hist() |
| # Skewed distribution positively skewed right  df.hist(column='Children') |
| # Spread narrow range  df.hist(column='Age') |
| # Categorical variables  # Skewed distribution positively skewed right  df.hist(column='Income') |
| # Bi modal distribution with two modes non-symmetric  df.hist(column='Tenure') |
| # Bi modal distribution non-symmetric  df.hist(column='Bandwidth\_GB\_Year') |
| # Performing Treatment of Outliers using Imputation  df['Children'].fillna(df['Children'].median(), inplace=True)  df['Age'].fillna(df['Age'].mean(), inplace=True)  df['Income'].fillna(df['Income'].mean(), inplace=True)  df['Tenure'].fillna(df['Tenure'].mean(), inplace=True)  df['Bandwidth\_GB\_Year'].fillna(df['Bandwidth\_GB\_Year'].mean(), inplace=True)  df['TechSupport'].fillna(df['TechSupport'].mode()[0])  df['Techie'].fillna(df['Techie'].mode()[0])  df['Phone'].fillna(df['Phone'].mode()[0])  df['TechSupport'].fillna(df['TechSupport'].mode()[0]) |
| # Lets Verify if NAs (NANs) were imputed using the isnull().sum function |
| df.isnull().sum() |
| # Detecting Outliers  import numpy as np  import pandas as pd  from pandas import DataFrame  import scipy.stats as stats |
| # The code for calculating the z-score and a new column/variable for the z-score calculation.  df['Z\_Score\_Age']=stats.zscore(df['Age']) |
| # The code to display the calculated z-score and the values  df[['Age','Z\_Score\_Age']].head |
| # Matplotlibis deployed to plot and Visualize bars  %matplotlib inline |
| import matplotlib.pyplot as plt |
| plt.hist(df['Z\_Score\_Age'])  plt.show() |
| # Using Seaborn to provides visualization  import seaborn |
| # Outliers are not dictated  boxplot=seaborn.boxplot(x='Age',data=df) |
| df['Z\_Score\_Children']=stats.zscore(df['Children']) |
| df[['Children','Z\_Score\_Children']].head |
| plt.hist(df['Z\_Score\_Children'])  plt.show() |
| # Outliers are dictated  boxplot=seaborn.boxplot(x='Children',data=df) |
| df['Z\_Score\_Income']=stats.zscore(df['Income']) |
| df[['Income','Z\_Score\_Income']].head |
| plt.hist(df['Z\_Score\_Income'])  plt.show() |
| # Outliers are dictated  boxplot=seaborn.boxplot(x='Income',data=df) |
| df['Z\_Score\_Tenure']=stats.zscore(df['Tenure']) |
| df[['Tenure','Z\_Score\_Tenure']].head |
| plt.hist(df['Tenure'])  plt.show() |
| # Outliers are not dictated  boxplot=seaborn.boxplot(x='Tenure',data=df) |
| df['Z\_Score\_Bandwidth\_GB\_Year']=stats.zscore(df['Bandwidth\_GB\_Year']) |
| df[['Bandwidth\_GB\_Year','Z\_Score\_Bandwidth\_GB\_Year']].head |
| plt.hist(df['Bandwidth\_GB\_Year'])  plt.show() |
| # Outliers are not dictated  boxplot=seaborn.boxplot(x='Bandwidth\_GB\_Year',data=df) |
| df['Income'].quantile |
| # Dropping outliers systematically  Cleaned\_df = df[df['Children'] < 100] |
| Cleaned\_df.head() |
| df.duplicated() |
| # Performing Treatment of Outliers by Dropping Rows containing outliers  df.drop\_duplicates(inplace = True) |
| df.head() |
|  |
| **PCA** |
| **import pandas as pd** |
| **import numpy as np** |
| **import matplotlib.pyplot as plt** |
| **from sklearn.decomposition import PCA** |
| **from sklearn.preprocessing import StandardScaler** |
| **%matplotlib inline** |
| **df = pd.read\_csv('love')** |
| **df.head(5)** |
| **features = ['population', 'age', 'children', 'income', 'monthly\_charge',]**  **x = df.loc[:, features].values** |
| **y = df.loc[:,['children']]** |
| **x = StandardScaler().fit\_transform(x)** |
| **pd.DataFrame(data = x, columns = features).head()** |
| **pca = PCA(n\_components=2)** |
| **principalComponents = pca.fit\_transform(x)** |
| **principalDf = pd.DataFrame(data = principalComponents**  **, columns = ['principal components 1', 'principal components 2'])** |
| **principalDf.head(5)** |
| **df[['children']].head(5)** |
| **finalDf = pd.concat([principalDf, df[['children']]], axis = 1)**  **finalDf.head(5)** |
| **fig = plt.figure(figsize = (8, 8))**  **ax = fig.add\_subplot(1,1,1)**  **ax.set\_xlabel('Principal Component 1', fontsize = 15)**  **ax.set\_ylabel('Principal Component 2', fontsize = 15)**  **ax.set\_title('2 Component PCA', fontsize = 15)**  **children = ['10-setosa', '9-versicolor', '7-virginica']**  **colors = ['b', 'g', 'r']**  **for children, color in zip(children,colors):**  **indicesToKeep = finalDf['children'] == children**  **ax.scatter(finalDf.loc[indicesToKeep, 'principal components 1']**  **, finalDf.loc[indicesToKeep, 'principal components 2']**  **, c = color**  **, s = 40)**  **ax.legend(children)**  **ax.grid()** |
| **pca.explained\_variance\_ratio\_**  **output =** array([0.20829561, 0.2013675 ]) |