# data-preprocessing-assignment

September 20, 2023

## 1 Assignment 15th Sept

[1]:
```
# Assignment 15 sep
# Perform Data preprocessing on Titanic dataset
# 1.Data Collection.
#    Please download the dataset from
#    https://www.kaggle.com/datasets/yasserh/titanic-dataset
# 2.Data Preprocessing
#     o   Import the Libraries.
#     o   Importing the dataset.
#     o   Checking for Null Values.
#     o   Data Visualization.
#     o   Outlier Detection
#     o   Splitting Dependent and Independent variables
#     o   Perform Encoding
#     o   Feature Scaling.
#     o   Splitting Data into Train and Test
```

## 2 Import the Libraries

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

## 3 Importing the dataset

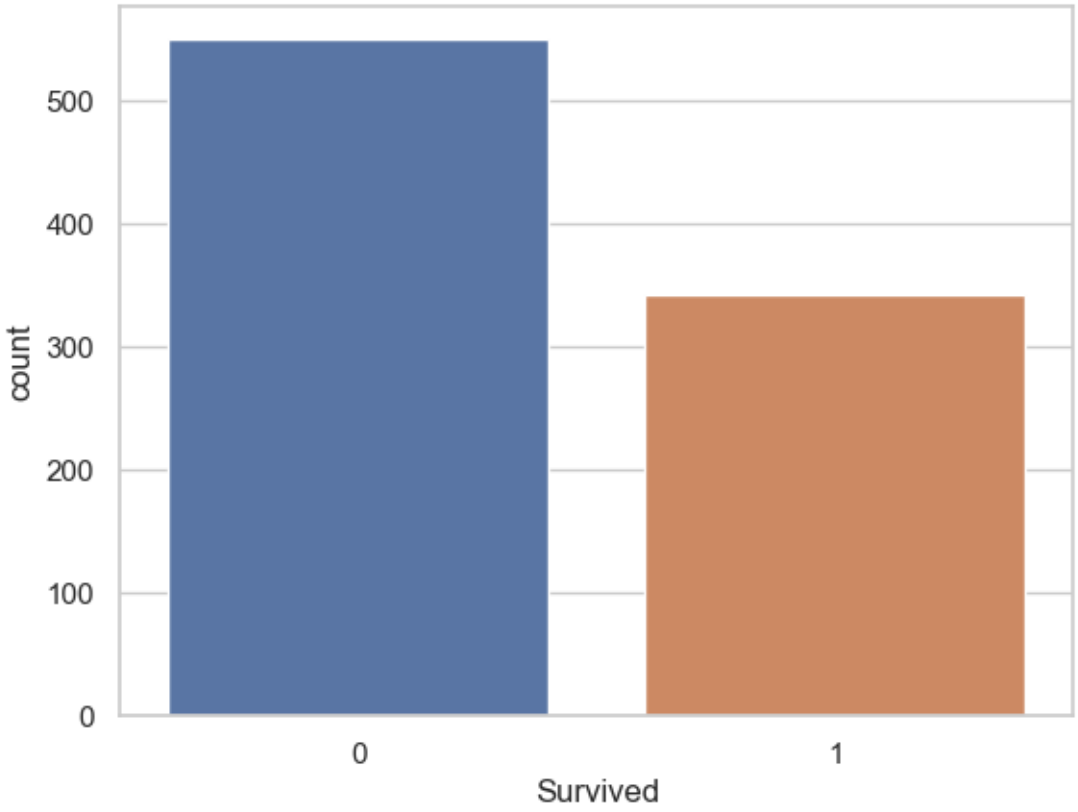
[3]:
```
dataset = pd.read_csv("Titanic.csv")
```

## 4 Checking for Null Values

[4]:
```
null_values = dataset.isnull().sum()
```

## 5 Data Visualization

[5]:
```
sns.set(style="whitegrid")
sns.countplot(x="Survived", data=dataset)
plt.show()
```




## 6 Outlier Detection

[6]:
```
numerical_columns = ['Age', 'Fare']
z_scores = np.abs((dataset[numerical_columns] - dataset[numerical_columns].
  ↪mean()) / dataset[numerical_columns].std())
threshold = 3
outlier_mask = z_scores > threshold
outliers = dataset[outlier_mask.any(axis=1)]
print("Outliers detected using Z-score:")
```

```
print(outliers)
```

```
Outliers detected using Z-score:
     PassengerId  Survived  Pclass  \
27            28         0       1
88            89         1       1
118          119         0       1
258          259         1       1
299          300         1       1
311          312         1       1
341          342         1       1
377          378         0       1
380          381         1       1
438          439         0       1
527          528         0       1
557          558         0       1
630          631         1       1
679          680         1       1
689          690         1       1
700          701         1       1
716          717         1       1
730          731         1       1
737          738         1       1
742          743         1       1
779          780         1       1
851          852         0       3

                                                 Name     Sex   Age  SibSp  \
27                        Fortune, Mr. Charles Alexander    male  19.0      3
88                            Fortune, Miss. Mabel Helen  female  23.0      3
118                             Baxter, Mr. Quigg Edmond    male  24.0      0
258                                   Ward, Miss. Anna  female  35.0      0
299    Baxter, Mrs. James (Helene DeLaudeniere Chaput)  female  50.0      0
311                           Ryerson, Miss. Emily Borie  female  18.0      2
341                        Fortune, Miss. Alice Elizabeth  female  24.0      3
377                            Widener, Mr. Harry Elkins    male  27.0      0
380                                Bidois, Miss. Rosalie  female  42.0      0
438                                    Fortune, Mr. Mark    male  64.0      1
527                                   Farthing, Mr. John    male   NaN      0
557                                  Robbins, Mr. Victor    male   NaN      0
630                  Barkworth, Mr. Algernon Henry Wilson    male  80.0      0
679                    Cardeza, Mr. Thomas Drake Martinez    male  36.0      0
689                     Madill, Miss. Georgette Alexandra  female  15.0      0
700  Astor, Mrs. John Jacob (Madeleine Talmadge Force)  female  18.0      1
716                         Endres, Miss. Caroline Louise  female  38.0      0
730                         Allen, Miss. Elisabeth Walton  female  29.0      0
737                                 Lesurer, Mr. Gustave J    male  35.0      0
```

```
742                Ryerson, Miss. Susan Parker "Suzette"  female  21.0      2
779  Robert, Mrs. Edward Scott (Elisabeth Walton Mc…  female  43.0      0
851                               Svensson, Mr. Johan    male  74.0      0

     Parch    Ticket      Fare            Cabin Embarked
27       2     19950  263.0000      C23 C25 C27        S
88       2     19950  263.0000      C23 C25 C27        S
118      1  PC 17558  247.5208          B58 B60        C
258      0  PC 17755  512.3292              NaN        C
299      1  PC 17558  247.5208          B58 B60        C
311      2  PC 17608  262.3750  B57 B59 B63 B66        C
341      2     19950  263.0000      C23 C25 C27        S
377      2    113503  211.5000              C82        C
380      0  PC 17757  227.5250              NaN        C
438      4     19950  263.0000      C23 C25 C27        S
527      0  PC 17483  221.7792              C95        S
557      0  PC 17757  227.5250              NaN        C
630      0     27042   30.0000              A23        S
679      1  PC 17755  512.3292      B51 B53 B55        C
689      1     24160  211.3375               B5        S
700      0  PC 17757  227.5250          C62 C64        C
716      0  PC 17757  227.5250              C45        C
730      0     24160  211.3375               B5        S
737      0  PC 17755  512.3292             B101        C
742      2  PC 17608  262.3750  B57 B59 B63 B66        C
779      1     24160  211.3375               B3        S
851      0    347060    7.7750              NaN        S
```

## 7 Splitting Dependent and Independent variables

[7]:
```
X = dataset.drop("Survived", axis=1)
y = dataset["Survived"]
```

## 8 Exclude non-numeric columns from X

[8]:
```
X_numeric = X.select_dtypes(include=[np.number])
```

## 9 Perform Encoding (Example: Label Encoding for Sex column)

[9]:
```
label_encoder = LabelEncoder()
X["Sex"] = label_encoder.fit_transform(X["Sex"])
```

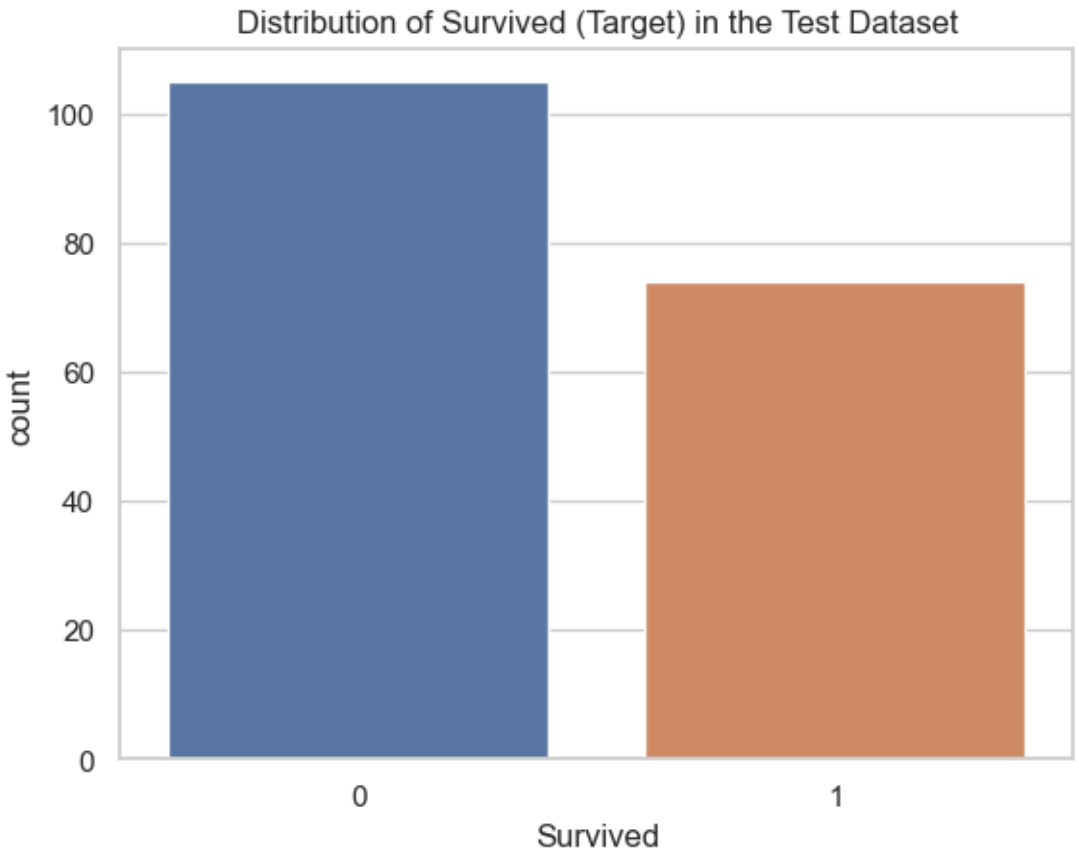
## 10 Feature Scaling (Example: Standardization)

[10]:
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_numeric)
```

## 11 Splitting Data into Train and Test

[11]:
```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,␣
↪random_state=42)
print("Number of samples in the test dataset:", len(X_test))
print("Basic statistics of the test dataset:")
print("Mean:", np.mean(X_test, axis=0))
print("Standard Deviation:", np.std(X_test, axis=0))
print("Minimum:", np.min(X_test, axis=0))
print("Maximum:", np.max(X_test, axis=0))
sns.countplot(x=y_test)
plt.title("Distribution of Survived (Target) in the Test Dataset")
plt.show()
```

```
Number of samples in the test dataset: 179
Basic statistics of the test dataset:
Mean: [-0.03455655 -0.10193644         nan -0.1095822   0.01175234 -0.03059938]
Standard Deviation: [1.00937639 1.04720501        nan 0.6611409  1.06847731
0.79204588]
Minimum: [-1.71066854 -1.56610693         nan -0.4745452  -0.47367361
-0.64842165]
Maximum: [1.72622007 0.82737724        nan 3.15480905 5.73284383 4.6344169 ]
```


Distribution of Survived (Target) in the Test Dataset
count
100
80
60
40
20
0
0
1
Survived