**AUGMENTED EXPERIMENT**

**WORK ON A REAL DATASET**

Implementation of Data Pre-processing on Real Dataset.

**ABSTRACT :**

A Real Dataset is so realistic to know and observe. Working on real dataset determines to know whether are there any missing values i.e. NULL values, or are there any outliers in it, is standardization and normalization required for the real dataset, does the encoding is applied or not for it. One of the real dataset is Student details. The dataset has 5 features with 40 entries. Before predication of the class variable, all the data pre-processing steps are to be preformed to make the dataset clean and neat. Working on real dataset is not easy as it have may entries with different kind of errors. Different plots can be visualized to know whether the dataset is noise free or not. After all this process, a real dataset is ready for further steps and for prediction of result. Data collection is one of the most important part while working on real dataset. It plays a great role in determining how well the analysis of data goes. Here, the dataset is collected by survey method.

**DATASET : student.csv**

**Student.csv**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | College Name | Age | Marks  (1-100) | Result |
| V.Vineela | GMRIT | 20 | 89 | Pass |
| V.Swetha | GMRIT | 20 | 90 | Pass |
| Upparapalli Ramesh | GMRIT |  | 90 | Pass |
| Jyothi | AU | 19 |  | Pass |
| Ramu Yerramsetti | GMRIT | 20 | 75 | Pass |
| Rajana Sai Manikanta | GMRIT | 20 | 84 | Pass |
| Uma Shankar | AU | 19 | 12 | Pass |
| Michel | GMRIT | 18 | 99 | Pass |
| Dhamareshwarakumar Gandikota | GMRIT | 21 | 70 | Pass |
| MS Rizwan | GMRIT |  | 92 | Pass |
| Thota Prasanth | GMRIT | 20 | 97 | Pass |
| Jhansi | GMR IT | 20 |  | Pass |
| Sneha | AITAM | 21 | 85 | Pass |
| Bukkaptanam Narasimha Swami | AU | 19 | 93 | Pass |
| Renuka kola | GMRIT | 19 |  | Pass |
| M. Suryanarayana | AITAM | 20 | 8.24 | Pass |
| Revathi | GMRIT | 20 |  | Pass |
| P vamsi | GMRIT | 21 | 0 | Fail |
| Surya | GMRIT | 21 | 91 | Pass |
| Pakki Venkata Sai Manasa | GMRIT | 19 | 90 | Pass |
| VINJAMURI ARAVIND | GMRIT | 19 | 46 | Pass |
| Michel | AITAM | 17 | 89 | Fail |
| Paluri.Kali Halkesh | GMRIT | 22 | 72 | Pass |
| Yasin | GMRIT | 20 | 63 | Pass |
| Sharon Kota | GMRIT | 18 | 20 | Pass |
| Aadhya | AU | 19 | 24 | Fail |
| Deepika | GMRIT | 19 | 78 | pass |
| Sri | AU | 21 | 29 | fail |
| Charishma | AITAM | 18 | 63 | pass |
| Surya | AITAM | 22 | 45 | fail |
| Bhavani | AU | 23 | 98 | pass |
| Shyam | AU |  | 23 | fail |
| Laxmi | GMRIT | 20 | 86 | pass |
| Kiran | AU | 19 | 85 | pass |
| Mani | GMRIT | 18 |  | fail |
| Charan | AITAM | 21 | 56 | pass |
| Aravind | AU | 20 | 33 | fail |
| Nagaraj | AITAM | 18 | 12 | fail |
| Sweety | GMRIT | 22 |  | pass |
| Jahnavi | AU | 19 | 39 | pass |

**PROGRAM CODE:**

#Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#Importing the dataset

dataset = pd.read\_csv('student.csv')

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

#Displaying the Rows as a Record

print(X)

**OUTPUT :**

[['V.Vineela' 'GMRIT' 20.0 89.0]

['V.Swetha' 'GMRIT' 20.0 90.0]

['Upparapalli Ramesh' 'GMRIT' nan 90.0]

['Jyothi' 'AU' 19.0 nan]

['Ramu Yerramsetti' 'GMRIT' 20.0 75.0]

['Rajana Sai Manikanta' 'GMRIT' 20.0 84.0]

['Uma Shankar' 'AU' 19.0 12.0]

['Michel' 'GMRIT' 18.0 99.0]

['Dhamareshwarakumar Gandikota' 'GMRIT' 21.0 70.0]

['MS Rizwan' 'GMRIT' nan 92.0]

['Thota Prasanth' 'GMRIT' 20.0 97.0]

['Jhansi' 'GMRIT' 20.0 nan]

['Sneha' 'AITAM' 21.0 85.0]

['Bukkaptanam Narasimha Swami' 'AU' 19.0 93.0]

['Renuka kola' 'GMRIT' 19.0 nan]

['M. Suryanarayana' 'AITAM' 20.0 8.24]

['Revathi' 'GMRIT' 20.0 nan]

['P vamsi' 'GMRIT' 21.0 0.0]

['Surya' 'GMRIT' 21.0 91.0]

['Pakki Venkata Sai Manasa' 'GMRIT' 19.0 90.0]

['VINJAMURI ARAVIND' 'GMRIT' 19.0 46.0]

['Michel' 'AITAM' 17.0 89.0]

['Paluri.Kali Halkesh' 'GMRIT' 22.0 72.0]

['Yasin' 'GMRIT' 20.0 63.0]

['Sharon Kota' 'GMRIT' 18.0 20.0]

['Aadhya' 'AU' 19.0 24.0]

['Deepika' 'GMRIT' 19.0 78.0]

['Sri' 'AU' 21.0 29.0]

['Charishma' 'AITAM' 18.0 63.0]

['Surya' 'AITAM' 22.0 45.0]

['Bhavani' 'AU' 23.0 98.0]

['Shyam' 'AU' nan 23.0]

['Laxmi' 'GMRIT' 20.0 86.0]

['Kiran' 'AU' 19.0 85.0]

['Mani' 'GMRIT' 18.0 nan]

['Charan' 'AITAM' 21.0 56.0]

['Aravind' 'AU' 20.0 33.0]

['Nagaraj' 'AITAM' 18.0 12.0]

['Sweety' 'GMRIT' 22.0 nan]

['Jahnavi' 'AU' 19.0 39.0]]

#Displaying the Column as a Record Field

print(y)

**OUTPUT :**

['Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass'

'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Pass' 'Fail' 'Pass' 'Pass'

'Pass' 'Fail' 'Pass' 'Pass' 'Pass' 'Fail' 'Pass' 'Fail' 'Pass' 'Fail'

'Pass' 'Fail' 'Pass' 'Pass' 'Fail' 'Pass' 'Fail' 'Fail' 'Pass' 'Pass']

#Tracing the Missing Data

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

imputer.fit(X[:, 2:4])

X[:, 2:4] = imputer.transform(X[:, 2:4])

# Displaying the Rows as a Record with filled in Missing Values

print(X)

**OUTPUT :**

[['V.Vineela' 'GMRIT' 20.0 89.0]

['V.Swetha' 'GMRIT' 20.0 90.0]

['Upparapalli Ramesh' 'GMRIT' 19.783783783783782 90.0]

['Jyothi' 'AU' 19.0 62.53647058823529]

['Ramu Yerramsetti' 'GMRIT' 20.0 75.0]

['Rajana Sai Manikanta' 'GMRIT' 20.0 84.0]

['Uma Shankar' 'AU' 19.0 12.0]

['Michel' 'GMRIT' 18.0 99.0]

['Dhamareshwarakumar Gandikota' 'GMRIT' 21.0 70.0]

['MS Rizwan' 'GMRIT' 19.783783783783782 92.0]

['Thota Prasanth' 'GMRIT' 20.0 97.0]

['Jhansi' 'GMRIT' 20.0 62.53647058823529]

['Sneha' 'AITAM' 21.0 85.0]

['Bukkaptanam Narasimha Swami' 'AU' 19.0 93.0]

['Renuka kola' 'GMRIT' 19.0 62.53647058823529]

['M. Suryanarayana' 'AITAM' 20.0 8.24]

['Revathi' 'GMRIT' 20.0 62.53647058823529]

['P vamsi' 'GMRIT' 21.0 0.0]

['Surya' 'GMRIT' 21.0 91.0]

['Pakki Venkata Sai Manasa' 'GMRIT' 19.0 90.0]

['VINJAMURI ARAVIND' 'GMRIT' 19.0 46.0]

['Michel' 'AITAM' 17.0 89.0]

['Paluri.Kali Halkesh' 'GMRIT' 22.0 72.0]

['Yasin' 'GMRIT' 20.0 63.0]

['Sharon Kota' 'GMRIT' 18.0 20.0]

['Aadhya' 'AU' 19.0 24.0]

['Deepika' 'GMRIT' 19.0 78.0]

['Sri' 'AU' 21.0 29.0]

['Charishma' 'AITAM' 18.0 63.0]

['Surya' 'AITAM' 22.0 45.0]

['Bhavani' 'AU' 23.0 98.0]

['Shyam' 'AU' 19.783783783783782 23.0]

['Laxmi' 'GMRIT' 20.0 86.0]

['Kiran' 'AU' 19.0 85.0]

['Mani' 'GMRIT' 18.0 62.53647058823529]

['Charan' 'AITAM' 21.0 56.0]

['Aravind' 'AU' 20.0 33.0]

['Nagaraj' 'AITAM' 18.0 12.0]

['Sweety' 'GMRIT' 22.0 62.53647058823529]

['Jahnavi' 'AU' 19.0 39.0]]

#Encoding the Independent Variable

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])], remainder='passthrough')

X = np.array(ct.fit\_transform(X)) #fitting and encoding happens in parallel

#Displaying the Rows as a Record with Categorical Data 0'sand 1's

print(X)

**OUTPUT :**

[[0.0 0.0 1.0 'V.Vineela' 20.0 89.0]

[0.0 0.0 1.0 'V.Swetha' 20.0 90.0]

[0.0 0.0 1.0 'Upparapalli Ramesh' 19.783783783783782 90.0]

[0.0 1.0 0.0 'Jyothi' 19.0 62.53647058823529]

[0.0 0.0 1.0 'Ramu Yerramsetti' 20.0 75.0]

[0.0 0.0 1.0 'Rajana Sai Manikanta' 20.0 84.0]

[0.0 1.0 0.0 'Uma Shankar' 19.0 12.0]

[0.0 0.0 1.0 'Michel' 18.0 99.0]

[0.0 0.0 1.0 'Dhamareshwarakumar Gandikota' 21.0 70.0]

[0.0 0.0 1.0 'MS Rizwan' 19.783783783783782 92.0]

[0.0 0.0 1.0 'Thota Prasanth' 20.0 97.0]

[0.0 0.0 1.0 'Jhansi' 20.0 62.53647058823529]

[1.0 0.0 0.0 'Sneha' 21.0 85.0]

[0.0 1.0 0.0 'Bukkaptanam Narasimha Swami' 19.0 93.0]

[0.0 0.0 1.0 'Renuka kola' 19.0 62.53647058823529]

[1.0 0.0 0.0 'M. Suryanarayana' 20.0 8.24]

[0.0 0.0 1.0 'Revathi' 20.0 62.53647058823529]

[0.0 0.0 1.0 'P vamsi' 21.0 0.0]

[0.0 0.0 1.0 'Surya' 21.0 91.0]

[0.0 0.0 1.0 'Pakki Venkata Sai Manasa' 19.0 90.0]

[0.0 0.0 1.0 'VINJAMURI ARAVIND' 19.0 46.0]

[1.0 0.0 0.0 'Michel' 17.0 89.0]

[0.0 0.0 1.0 'Paluri.Kali Halkesh' 22.0 72.0]

[0.0 0.0 1.0 'Yasin' 20.0 63.0]

[0.0 0.0 1.0 'Sharon Kota' 18.0 20.0]

[0.0 1.0 0.0 'Aadhya' 19.0 24.0]

[0.0 0.0 1.0 'Deepika' 19.0 78.0]

[0.0 1.0 0.0 'Sri' 21.0 29.0]

[1.0 0.0 0.0 'Charishma' 18.0 63.0]

[1.0 0.0 0.0 'Surya' 22.0 45.0]

[0.0 1.0 0.0 'Bhavani' 23.0 98.0]

[0.0 1.0 0.0 'Shyam' 19.783783783783782 23.0]

[0.0 0.0 1.0 'Laxmi' 20.0 86.0]

[0.0 1.0 0.0 'Kiran' 19.0 85.0]

[0.0 0.0 1.0 'Mani' 18.0 62.53647058823529]

[1.0 0.0 0.0 'Charan' 21.0 56.0]

[0.0 1.0 0.0 'Aravind' 20.0 33.0]

[1.0 0.0 0.0 'Nagaraj' 18.0 12.0]

[0.0 0.0 1.0 'Sweety' 22.0 62.53647058823529]

[0.0 1.0 0.0 'Jahnavi' 19.0 39.0]]

#Encoding the Dependent Variable

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y = le.fit\_transform(y)

# Displaying the Column as a Record Field

print(y)

**OUTPUT :**

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0 1 0

0 1 1]

#Splitting the dataset into the Training set and Test set

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=1)

**OUTPUT :**

[[0.0 0.0 1.0 'Pakki Venkata Sai Manasa' 19.0 90.0]

[0.0 0.0 1.0 'Deepika' 19.0 78.0]

[0.0 0.0 1.0 'Laxmi' 20.0 86.0]

[0.0 0.0 1.0 'P vamsi' 21.0 0.0]

[0.0 1.0 0.0 'Bhavani' 23.0 98.0]

[0.0 1.0 0.0 'Aravind' 20.0 33.0]

[0.0 1.0 0.0 'Kiran' 19.0 85.0]

[1.0 0.0 0.0 'Charishma' 18.0 63.0]

[0.0 0.0 1.0 'Ramu Yerramsetti' 20.0 75.0]

[0.0 0.0 1.0 'Renuka kola' 19.0 62.53647058823529]

[0.0 0.0 1.0 'Thota Prasanth' 20.0 97.0]

[1.0 0.0 0.0 'Charan' 21.0 56.0]

[0.0 0.0 1.0 'Yasin' 20.0 63.0]

[0.0 0.0 1.0 'Sharon Kota' 18.0 20.0]

[0.0 0.0 1.0 'Mani' 18.0 62.53647058823529]

[0.0 0.0 1.0 'VINJAMURI ARAVIND' 19.0 46.0]

[0.0 0.0 1.0 'Surya' 21.0 91.0]

[0.0 1.0 0.0 'Aadhya' 19.0 24.0]

[0.0 1.0 0.0 'Uma Shankar' 19.0 12.0]

[0.0 1.0 0.0 'Bukkaptanam Narasimha Swami' 19.0 93.0]

[0.0 0.0 1.0 'Michel' 18.0 99.0]

[0.0 0.0 1.0 'Sweety' 22.0 62.53647058823529]

[0.0 0.0 1.0 'V.Swetha' 20.0 90.0]

[0.0 0.0 1.0 'Revathi' 20.0 62.53647058823529]

[0.0 0.0 1.0 'V.Vineela' 20.0 89.0]

[1.0 0.0 0.0 'M. Suryanarayana' 20.0 8.24]

[0.0 0.0 1.0 'Rajana Sai Manikanta' 20.0 84.0]

[0.0 0.0 1.0 'Jhansi' 20.0 62.53647058823529]

[0.0 0.0 1.0 'MS Rizwan' 19.783783783783782 92.0]

[0.0 0.0 1.0 'Dhamareshwarakumar Gandikota' 21.0 70.0]

[1.0 0.0 0.0 'Sneha' 21.0 85.0]

[1.0 0.0 0.0 'Nagaraj' 18.0 12.0]]

#Displays X Test Set Info

print(X\_test)

**OUTPUT :**

[[0.0 0.0 1.0 'Upparapalli Ramesh' 19.783783783783782 90.0]

[0.0 1.0 0.0 'Shyam' 19.783783783783782 23.0]

[0.0 1.0 0.0 'Jyothi' 19.0 62.53647058823529]

[1.0 0.0 0.0 'Michel' 17.0 89.0]

[0.0 1.0 0.0 'Sri' 21.0 29.0]

[1.0 0.0 0.0 'Surya' 22.0 45.0]

[0.0 0.0 1.0 'Paluri.Kali Halkesh' 22.0 72.0]

[0.0 1.0 0.0 'Jahnavi' 19.0 39.0]]

#Displays y Training Set Info

print(y\_train)

**OUTPUT :**

[1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0]

#Displays y Test Set Info

print(y\_test)

**OUTPUT :**

[1 0 1 0 0 0 1 1]

#Feature Scaling

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

X\_train[:,4:]=sc.fit\_transform(X\_train[:,4:])

X\_test[:,4:]=sc.transform(X\_test[:,4:])

#Displays X Training Set Info

print(X\_train)

**OUTPUT :**

[[0.0 0.0 1.0 'Pakki Venkata Sai Manasa' -0.63429292487538

0.8838528186880219]

[0.0 0.0 1.0 'Deepika' -0.63429292487538 0.47398849789433334]

[0.0 0.0 1.0 'Laxmi' 0.2191193740478599 0.747231378423459]

[0.0 0.0 1.0 'P vamsi' 1.0725316729710999 -2.190129587264642]

[0.0 1.0 0.0 'Bhavani' 2.7793562708175794 1.1570956992171475]

[0.0 1.0 0.0 'Aravind' 0.2191193740478599 -1.0630027050819986]

[0.0 1.0 0.0 'Kiran' -0.63429292487538 0.7130760183573183]

[1.0 0.0 0.0 'Charishma' -1.48770522379862 -0.0383419030977773]

[0.0 0.0 1.0 'Ramu Yerramsetti' 0.2191193740478599 0.3715224176959112]

[0.0 0.0 1.0 'Renuka kola' -0.63429292487538 -0.05417391705784739]

[0.0 0.0 1.0 'Thota Prasanth' 0.2191193740478599 1.1229403391510069]

[1.0 0.0 0.0 'Charan' 1.0725316729710999 -0.2774294235607623]

[0.0 0.0 1.0 'Yasin' 0.2191193740478599 -0.0383419030977773]

[0.0 0.0 1.0 'Sharon Kota' -1.48770522379862 -1.5070223859418277]

[0.0 0.0 1.0 'Mani' -1.48770522379862 -0.05417391705784739]

[0.0 0.0 1.0 'VINJAMURI ARAVIND' -0.63429292487538 -0.6189830242221693]

[0.0 0.0 1.0 'Surya' 1.0725316729710999 0.9180081787541625]

[0.0 1.0 0.0 'Aadhya' -0.63429292487538 -1.370400945677265]

[0.0 1.0 0.0 'Uma Shankar' -0.63429292487538 -1.7802652664709535]

[0.0 1.0 0.0 'Bukkaptanam Narasimha Swami' -0.63429292487538

0.986318898886444]

[0.0 0.0 1.0 'Michel' -1.48770522379862 1.1912510592832883]

[0.0 0.0 1.0 'Sweety' 1.9259439718943396 -0.05417391705784739]

[0.0 0.0 1.0 'V.Swetha' 0.2191193740478599 0.8838528186880219]

[0.0 0.0 1.0 'Revathi' 0.2191193740478599 -0.05417391705784739]

[0.0 0.0 1.0 'V.Vineela' 0.2191193740478599 0.8496974586218812]

[1.0 0.0 0.0 'M. Suryanarayana' 0.2191193740478599 -1.9086894203196425]

[0.0 0.0 1.0 'Rajana Sai Manikanta' 0.2191193740478599

0.6789206582911776]

[0.0 0.0 1.0 'Jhansi' 0.2191193740478599 -0.05417391705784739]

[0.0 0.0 1.0 'MS Rizwan' 0.034597795902293345 0.9521635388203032]

[0.0 0.0 1.0 'Dhamareshwarakumar Gandikota' 1.0725316729710999

0.20074561736520766]

[1.0 0.0 0.0 'Sneha' 1.0725316729710999 0.7130760183573183]

[1.0 0.0 0.0 'Nagaraj' -1.48770522379862 -1.7802652664709535]]

#Displays X Test Set Info

print(X\_test)

**OUTPUT :**

[[0.0 0.0 1.0 'Upparapalli Ramesh' 0.034597795902293345

0.8838528186880219]

[0.0 1.0 0.0 'Shyam' 0.034597795902293345 -1.4045563057434056]

[0.0 1.0 0.0 'Jyothi' -0.63429292487538 -0.05417391705784739]

[1.0 0.0 0.0 'Michel' -2.3411175227218597 0.8496974586218812]

[0.0 1.0 0.0 'Sri' 1.0725316729710999 -1.1996241453465615]

[1.0 0.0 0.0 'Surya' 1.9259439718943396 -0.6531383842883101]

[0.0 0.0 1.0 'Paluri.Kali Halkesh' 1.9259439718943396 0.2690563374974891]

[0.0 1.0 0.0 'Jahnavi' -0.63429292487538 -0.8580705446851543]]

#Name,College Name,Age,Marks(1-100),Result

import seaborn as sns

data=pd.read\_csv('student.csv')

data.head()

**OUTPUT :**

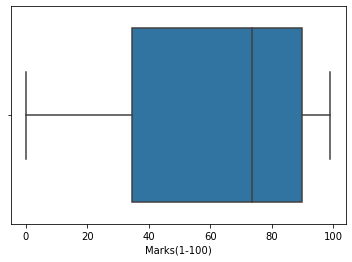
| **Name** | **College Name** | **Age** | **Marks(1-100)** | **Result** |
| --- | --- | --- | --- | --- |
| **0** | V.Vineela | GMRIT | 20.0 | 89.0 | Pass |
| **1** | V.Swetha | GMRIT | 20.0 | 90.0 | Pass |
| **2** | Upparapalli Ramesh | GMRIT | NaN | 90.0 | Pass |
| **3** | Jyothi | AU | 19.0 | NaN | Pass |
| **4** | Ramu Yerramsetti | GMRIT | 20.0 | 75.0 | Pass |

# Box Plot

import seaborn as sns

sns.boxplot(data['Marks(1-100)'])

**OUTPUT :**



# Position of the Outlier

print(np.where(data['Marks(1-100)']>95))

**OUTPUT :**

(array([ 7, 10, 30], dtype=int64),)