For columns:

Glucose,
BloodPressure,
BMI,
DiabetesPedigreeFunction

If the column value is 0, then they should be considered as **missing data.**

1. Firstly, replace all Missing values with relevant figures.

```
import numpy as np
import pandas as pd
df = pd.read_csv('Dataset_Day7.csv')
print(df.info())
missing_value_percent = df.isna().sum() / len(df)
* 100
print(missing_value_percent)
skewness = df.skew()
print(skewness)
df["Glucose"].fillna(df["Glucose"].median(),
inplace=True)
df["BloodPressure"].fillna(df["BloodPressure"].me
an(), inplace=True)
df["BMI"].fillna(df["BMI"].median(),
inplace=True)
df["Outcome"].fillna(df["Outcome"].mean(),
inplace=True)
print(df.info())
```

```
C:\Users\tejas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\
tejas\PycharmProjects\pythonProject\START\Day8Q1.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None
Pregnancies                 0.0
Glucose                     0.0
BloodPressure               0.0
BMI                         0.0
DiabetesPedigreeFunction    0.0
Age                         0.0
Outcome                     0.0
dtype: float64
Pregnancies                 0.901674
Glucose                     0.173754
BloodPressure              -1.843608
BMI                        -0.428982
DiabetesPedigreeFunction    1.919911
Age                         1.129597
Outcome                     0.635017
dtype: float64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None

Process finished with exit code 0
```

2. Then remove all existing outliers and get the final data for classification.

```python
import numpy as np
import pandas as pd
df = pd.read_csv('Dataset_Day7.csv')
print(df.info())
missing_value_percent = df.isna().sum() / len(df)
```

```
* 100
print(missing_value_percent)
skewness = df.skew()
print(skewness)
df["Glucose"].fillna(df["Glucose"].median(),
inplace=True)
df["BloodPressure"].fillna(df["BloodPressure"].me
an(), inplace=True)
df["BMI"].fillna(df["BMI"].median(),
inplace=True)
df["Outcome"].fillna(df["Outcome"].mean(),
inplace=True)
print(df.info())
OutlierData = pd.DataFrame()
temp = df[["Pregnancies", "Glucose",
"BloodPressure","BMI","DiabetesPedigreeFunction"
, "Age", "Outcome" ]]
for col in ["Pregnancies", "Glucose",
"BloodPressure","BMI","DiabetesPedigreeFunction"
, "Age", "Outcome"]:
    Q1 = temp[col].quantile(0.25)  # Gives 25th
Percentile or Q1
    Q3 = temp[col].quantile(0.75)  # Gives 75th
Percentile or Q3

    IQR = Q3 - Q1

    UpperBound = Q3 + 1.5 * IQR
    LowerBound = Q1 - 1.5 * IQR

    OutlierData[col] = temp[col][(temp[col] <
LowerBound) | (temp[col] > UpperBound)]
    print(len(OutlierData))
    df_OutlierFree = df.drop(OutlierData.index,
axis=0)
    df_OutlierFree.info()
```

```
  tejas\PycharmProjects\pythonProject\START\Day8Q2.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None
Pregnancies                 0.0
Glucose                     0.0
BloodPressure               0.0
BMI                         0.0
DiabetesPedigreeFunction    0.0
Age                         0.0
Outcome                     0.0
dtype: float64
Pregnancies                 0.901674
Glucose                     0.173754
BloodPressure              -1.843608
BMI                        -0.428982
DiabetesPedigreeFunction    1.919911
Age                         1.129597
Outcome                     0.635017
dtype: float64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None
4
<class 'pandas.core.frame.DataFrame'>
Index: 764 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               764 non-null    int64
 1   Glucose                   764 non-null    int64
 2   BloodPressure             764 non-null    int64
 3   BMI                       764 non-null    float64
 4   DiabetesPedigreeFunction  764 non-null    float64
```

```
 59  5   Age                       764 non-null    int64
 60  6   Outcome                   764 non-null    int64
 61 dtypes: float64(2), int64(5)
 62 memory usage: 47.8 KB
 63 4
 64 <class 'pandas.core.frame.DataFrame'>
 65 Index: 764 entries, 0 to 767
 66 Data columns (total 7 columns):
 67  #   Column                    Non-Null Count  Dtype
 68 ---  ------                    --------------  -----
 69  0   Pregnancies               764 non-null    int64
 70  1   Glucose                   764 non-null    int64
 71  2   BloodPressure             764 non-null    int64
 72  3   BMI                       764 non-null    float64
 73  4   DiabetesPedigreeFunction  764 non-null    float64
 74  5   Age                       764 non-null    int64
 75  6   Outcome                   764 non-null    int64
 76 dtypes: float64(2), int64(5)
 77 memory usage: 47.8 KB
 78 4
 79 <class 'pandas.core.frame.DataFrame'>
 80 Index: 764 entries, 0 to 767
 81 Data columns (total 7 columns):
 82  #   Column                    Non-Null Count  Dtype
 83 ---  ------                    --------------  -----
 84  0   Pregnancies               764 non-null    int64
 85  1   Glucose                   764 non-null    int64
 86  2   BloodPressure             764 non-null    int64
 87  3   BMI                       764 non-null    float64
 88  4   DiabetesPedigreeFunction  764 non-null    float64
 89  5   Age                       764 non-null    int64
 90  6   Outcome                   764 non-null    int64
 91 dtypes: float64(2), int64(5)
 92 memory usage: 47.8 KB
 93 4
 94 <class 'pandas.core.frame.DataFrame'>
 95 Index: 764 entries, 0 to 767
 96 Data columns (total 7 columns):
 97  #   Column                    Non-Null Count  Dtype
 98 ---  ------                    --------------  -----
 99  0   Pregnancies               764 non-null    int64
100  1   Glucose                   764 non-null    int64
101  2   BloodPressure             764 non-null    int64
102  3   BMI                       764 non-null    float64
103  4   DiabetesPedigreeFunction  764 non-null    float64
104  5   Age                       764 non-null    int64
105  6   Outcome                   764 non-null    int64
106 dtypes: float64(2), int64(5)
107 memory usage: 47.8 KB
108 4
109 <class 'pandas.core.frame.DataFrame'>
110 Index: 764 entries, 0 to 767
111 Data columns (total 7 columns):
112  #   Column                    Non-Null Count  Dtype
113 ---  ------                    --------------  -----
114  0   Pregnancies               764 non-null    int64
115  1   Glucose                   764 non-null    int64
116  2   BloodPressure             764 non-null    int64
117  3   BMI                       764 non-null    float64
```

```
118  4   DiabetesPedigreeFunction  764 non-null    float64
119  5   Age                       764 non-null    int64
120  6   Outcome                   764 non-null    int64
121 dtypes: float64(2), int64(5)
122 memory usage: 47.8 KB
123 4
124 <class 'pandas.core.frame.DataFrame'>
125 Index: 764 entries, 0 to 767
126 Data columns (total 7 columns):
127  #   Column                    Non-Null Count  Dtype
128 ---  ------                    --------------  -----
129  0   Pregnancies               764 non-null    int64
130  1   Glucose                   764 non-null    int64
131  2   BloodPressure             764 non-null    int64
132  3   BMI                       764 non-null    float64
133  4   DiabetesPedigreeFunction  764 non-null    float64
134  5   Age                       764 non-null    int64
135  6   Outcome                   764 non-null    int64
136 dtypes: float64(2), int64(5)
137 memory usage: 47.8 KB
138 4
139 <class 'pandas.core.frame.DataFrame'>
140 Index: 764 entries, 0 to 767
141 Data columns (total 7 columns):
142  #   Column                    Non-Null Count  Dtype
143 ---  ------                    --------------  -----
144  0   Pregnancies               764 non-null    int64
145  1   Glucose                   764 non-null    int64
146  2   BloodPressure             764 non-null    int64
147  3   BMI                       764 non-null    float64
148  4   DiabetesPedigreeFunction  764 non-null    float64
149  5   Age                       764 non-null    int64
150  6   Outcome                   764 non-null    int64
151 dtypes: float64(2), int64(5)
152 memory usage: 47.8 KB
153
154 Process finished with exit code 0
155
```

3,Split the data into 70% training and 30% testing data. Then, use a k-Nearest Neighbor algorithm with target variable as ‘Outcome’.

a. Print the default model performance metrics: Accuracy, Precision, Recall, F1Score

b. Plot a Precision & Recall vs k(no. of neighbours) curve (both Prec and Rec on the same graph). Find the k for which F1-score is the highest. **Use any one Distance Metric for this problem.**

c. Find the best distance metric, no. of neighbors combination for the kNN algorithm

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import
train_test_split
from sklearn.neighbors import
KNeighborsClassifier
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score
import math

df = pd.read_csv('Dataset_Day7.csv')
print(df.info())
missing_value_percent = df.isna().sum() / len(df)
* 100
print(missing_value_percent)
skewness = df.skew()
print(skewness)
df["Glucose"].fillna(df["Glucose"].median(),
inplace=True)
df["BloodPressure"].fillna(df["BloodPressure"].me
an(), inplace=True)
df["BMI"].fillna(df["BMI"].median(),
inplace=True)
df["Outcome"].fillna(df["Outcome"].mean(),
inplace=True)
print(df.info())
OutlierData = pd.DataFrame()
temp = df[["Pregnancies", "Glucose",
"BloodPressure", "BMI",
"DiabetesPedigreeFunction", "Age", "Outcome"]]
for col in ["Pregnancies", "Glucose",
"BloodPressure", "BMI",
"DiabetesPedigreeFunction", "Age", "Outcome"]:
    Q1 = temp[col].quantile(0.25)  # Gives 25th
Percentile or Q1
    Q3 = temp[col].quantile(0.75)  # Gives 75th
Percentile or Q3

    IQR = Q3 - Q1

    UpperBound = Q3 + 1.5 * IQR
    LowerBound = Q1 - 1.5 * IQR

    OutlierData[col] = temp[col][(temp[col] <
```

```
LowerBound) | (temp[col] > UpperBound)]

df_OutlierFree = df.drop(OutlierData.index,
axis=0)
df_OutlierFree.info()

X = df_OutlierFree.drop('Outcome', axis=1)  # all
columns except 'Outcome'
y = df_OutlierFree['Outcome']  # target Variable
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=203)
unique_classes = np.unique(y)
print('Number of unique classes:',
len(unique_classes))
k_start = int(math.sqrt(len(X_train)))
print(k_start)
metric_start = 'euclidean'

knn = KNeighborsClassifier(n_neighbors=k_start,
metric=metric_start)
# fit the model
knn = knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)

print("Model Performance metrics are as below:")
print("Accuracy: ", accuracy_score(y_test,
y_pred))
print("Precision: ", precision_score(y_test,
y_pred, zero_division=1))
print("Recall: ", recall_score(y_test, y_pred))
print("F1-Score: ", f1_score(y_test, y_pred))

k_values = np.arange(2,25)
metric_values = ['euclidean', 'manhattan',
'hamming']
prec = []
rec = []
acc = []
PerfData = pd.DataFrame(columns=['Nearest
Neighbor', 'Distance Metric', 'Precision',
```

```
'Recall', 'Accuracy'])

for dm in metric_values:
    for k in k_values:
        knn = KNeighborsClassifier(n_neighbors=k,
metric=dm)
        knn = knn.fit(X_train, y_train)
        y_pred = knn.predict(X_test)

        row = [[k, dm, precision_score(y_test,
y_pred, zero_division=1), recall_score(y_test,
y_pred),
                accuracy_score(y_test, y_pred)]]

        df2 = pd.DataFrame(row, columns=['Nearest
Neighbor', 'Distance Metric', 'Precision',
'Recall', 'Accuracy'])
        PerfData = pd.concat([PerfData, df2],
ignore_index=True)

print(PerfData.tail())
precision = PerfData['Precision']
recall = PerfData['Recall']
PerfData["F1 Score"] = (2 * PerfData["Precision"]
* PerfData["Recall"]) / (PerfData["Precision"] +
PerfData["Recall"])
print(PerfData[PerfData['F1 Score'] ==
max(PerfData['F1 Score'])])
# Plot Precision & Recall vs k Curve
plt.plot(k_values, PerfData[PerfData["Distance
Metric"]=='euclidean']['Precision'],
label='Precision')
plt.plot(k_values, PerfData[PerfData["Distance
Metric"]=='euclidean']['Recall'], label='Recall')
plt.xlabel('k (Number of Neighbors)')
plt.ylabel('Precision / Recall')
plt.title('Precision & Recall vs k Curve')
plt.legend(loc='upper right')
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None
Pregnancies                 0.0
Glucose                     0.0
BloodPressure               0.0
BMI                         0.0
DiabetesPedigreeFunction    0.0
Age                         0.0
Outcome                     0.0
dtype: float64
Pregnancies                 0.901674
Glucose                     0.173754
BloodPressure              -1.843608
BMI                        -0.428982
DiabetesPedigreeFunction    1.919911
Age                         1.129597
Outcome                     0.635017
dtype: float64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   BMI                       768 non-null    float64
 4   DiabetesPedigreeFunction  768 non-null    float64
 5   Age                       768 non-null    int64
 6   Outcome                   768 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 42.1 KB
None
<class 'pandas.core.frame.DataFrame'>
Index: 764 entries, 0 to 767
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               764 non-null    int64
 1   Glucose                   764 non-null    int64
 2   BloodPressure             764 non-null    int64
 3   BMI                       764 non-null    float64
 4   DiabetesPedigreeFunction  764 non-null    float64
 5   Age                       764 non-null    int64
```

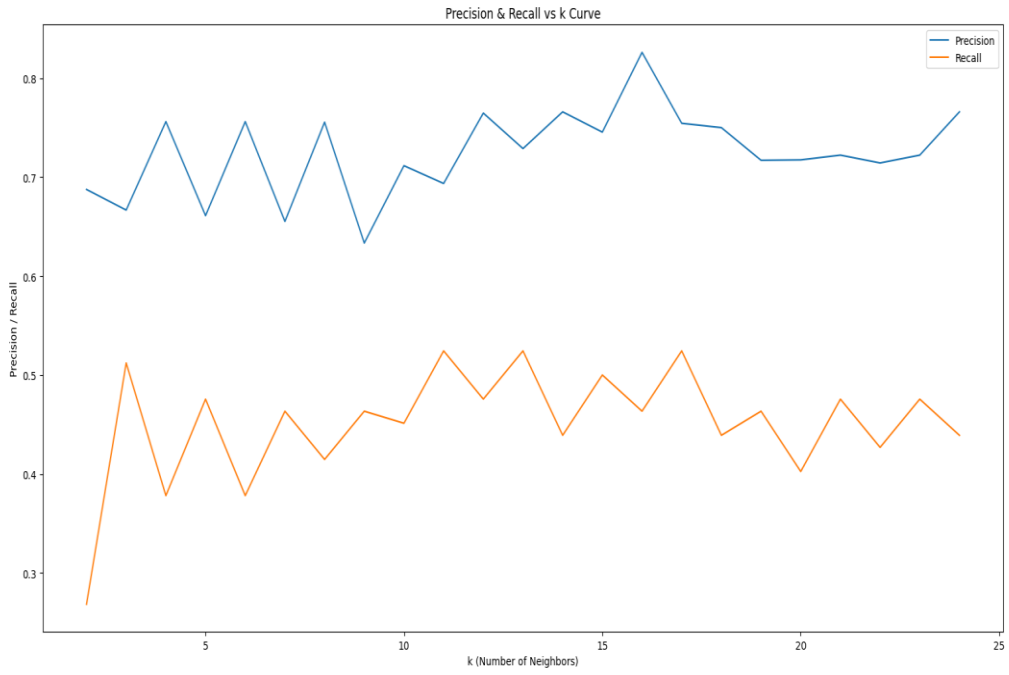
File - Day8Q3

```
 6   Outcome                   764 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 47.8 KB
Number of unique classes: 2
23
Model Performance metrics are as below:
Accuracy:  0.7478260869565218
Precision:  0.7222222222222222
Recall:  0.47560975609756095
F1-Score:  0.5735294117647057
    Nearest Neighbor Distance Metric  Precision  Recall  Accuracy
64                20         hamming        1.0     0.0  0.643478
65                21         hamming        1.0     0.0  0.643478
66                22         hamming        1.0     0.0  0.643478
67                23         hamming        1.0     0.0  0.643478
68                24         hamming        1.0     0.0  0.643478
    Nearest Neighbor Distance Metric  Precision   Recall  Accuracy  F1 Score
15                17       euclidean   0.754386  0.52439  0.769565  0.618705

Process finished with exit code 0

```

Figure 1



x=21.33 y=0.7823