

## Q1)

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.metrics import accuracy_score, confusion_matrix,  
classification_report
```

```
df=pd.read_csv("diabetes.csv")
```

```
df=df.dropna()
```

```
df.isnull().sum()
```

```
X = df.drop("Outcome", axis=1)
```

```
y = df["Outcome"]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random_state=42)
```

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)

accuracy_scores = []

values=range(1,21)

for k in values:

    knn = KNeighborsClassifier(n_neighbors=k)

    knn.fit(X_train, y_train)

    y_pred = knn.predict(X_test)

    acc = accuracy_score(y_test, y_pred)

    accuracy_scores.append(acc)

optimal_k = np.argmax(accuracy_scores) + 1

print("\nOptimal value of K:" ,optimal_k)

plt.plot(values, accuracy_scores)

plt.title("KNN - Accuracy vs K Value")

plt.xlabel("Number of Neighbors (K)")

plt.ylabel("Accuracy")

plt.show()

ypred = knn.predict(X_test)

accuracy= accuracy_score(y_test, y_pred)
```

```
print('Accuracy of the model with optimal K=',accuracy)

newpatient = np.array([[6, 148, 72, 35, 0, 33.6, 0.627, 50]])

new_patient_scaled = scaler.transform(newpatient)

prediction = knn.predict(new_patient_scaled)

if prediction[0] == 1:

    print("The new patient is diabetic.")

else:

    print("The new patient is non-diabetic.")
```

## Q2)

```
import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import AgglomerativeClustering

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read_csv("Wholesalecustomersdata.csv")

data = data.drop(columns=['Channel', 'Region'], errors='ignore')

scaler= StandardScaler()

data_scaled = scaler.fit_transform(data)

print(data_scaled)
```

```
agg_clustering = AgglomerativeClustering(n_clusters=5)
data['Cluster'] = agg_clustering.fit_predict(data_scaled)

plt.figure(figsize=(14, 10), dpi=100)
sns.pairplot(data, hue='Cluster', palette='Set2', markers=["o", "s", "D", "^"])
plt.suptitle('Pairwise Clustering of Wholesale Customers')
plt.tight_layout()
plt.show()
```

```
cluster_sum = data.groupby('Cluster').mean()
print("Cluster Summary (mean spending per category):")
print(cluster_sum)
```