Emily Shepherd
CS 645

I chose to use option 2 for my assignment.

Part 1: Prepare Data
I used sklearn to standardize my data and prepare it for analysis.

```
In [4]: from sklearn.preprocessing import StandardScaler

In [8]: # Create a scalar object
        scaler = StandardScaler()

In [10]: #Normalize the data in the bmi and bp columns
         scaler.fit(df.drop('TARGET', axis = 1))
         scaled_features = scaler.transform(df.drop('TARGET', axis = 1))

In [13]: # Create a data frame from the scaled features array
         df_feat = pd.DataFrame(scaled_features, columns = df.columns[:-1])
```

Part II: I split my data into training data and testing data using test-train-split from sklearn.

```
In [15]: from sklearn.model_selection import train_test_split

In [16]: #Split the data into testing and training data
         X = df_feat
         y = df['TARGET']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=47303)
```

Part III: Perform K Nearest Neighbor classification using sklearn

A. K = 1

```
In [17]: from sklearn.neighbors import KNeighborsClassifier

In [18]: knn = KNeighborsClassifier(n_neighbors = 1)

In [19]: knn.fit(X_train, y_train)
Out[19]: KNeighborsClassifier(n_neighbors=1)

In [20]: pred = knn.predict(X_test)

In [23]: from sklearn.metrics import classification_report, confusion_matrix

In [24]: print(confusion_matrix(y_test, pred))
         [[11  1]
          [ 1 13]]
```

From the confusion matrix, I can see that for category 0, 11 entries were classified correctly and 1 was not. For category 1, 1 entry was miscategorized and 13 were categorized correctly.

B. K = 5

```
In [25]: knn = KNeighborsClassifier(n_neighbors = 5)
         knn.fit(X_train, y_train)
         pred = knn.predict(X_test)
         print(confusion_matrix(y_test, pred))
```

```
[[ 9  3]
 [ 0 14]]
```

For K = 5, 9 out of 12 in category 0 were correctly classified. All of the items in category 1 were correctly classified.

C. K = 11

```
In [26]: knn = KNeighborsClassifier(n_neighbors = 11)
         knn.fit(X_train, y_train)
         pred = knn.predict(X_test)
         print(confusion_matrix(y_test, pred))
```

```
[[ 9  3]
 [ 1 13]]
```

For K = 11, 9 out of 12 in category 0 were correctly classified. 13 out of 14 were classified correctly for category 1.