I used Euclidean distance since it is the most correct distance function that can be used in these types of models. The disadvantage of Euclidean Distance is that it can take too long when the data is big in terms of number of features. Since our data has 8 features and it is such a small number relatively, we can use Euclidean distance for better results. I also tried Chi-square distance in order to prove my claim. It took nearly quarter of the time that Euclidean Distance take but the results were much worse.

Sometimes the features can be useless or indicate the same thing in machine learning algorithms and when the number of features is much, we can reduce the number of features and can reduce the cost without getting too much error. Even when we reduce the data, sometimes we get better results since the data can overfit.

When I train the data with kNN classifier using all the features, the confusion matrix accuracy and time are shown below.

```
TP =  30 FN =  25
FP =  19 TN =  80
```

```
accuracy wiith all features:  0.7142857142857143
12.314371109008789  time for 8 feature training
```

After I trained with all the 8 features, I used Backward elimination method for feature selection method. This method can be explained basically as I trained the data with taking out one of the features for all features and then eliminate the feature that you get the most accuracy. I did this until I get a worse accuracy before eliminating a feature and when the accuracy dropped, I stopped.

The accuracies, dropped features and validation times are listed below.

- ```
  BloodPressure  is the dropped feature
  accuracy after first feature dropped:  0.7402
  ```
- ```
  95.8690357208252  time for 8 featue 8 times
  ```

- ```
  Insulin  is the dropped feature
  accuracy after second feature dropped:  0.7597
  ```
- ```
  84.92653512954712  time for 7 feature 7 times
  ```

- ```
  Age  is the dropped feature
  accuracy after third feature dropped:  0.77272
  ```
- ```
  73.38230609893799  time for 6 feature 6 times
  ```

- ```
  SkinThickness  is the dropped feature
  ```

When I analyzed the time data, I see that feature selection takes much more than the training time as I expected because in this method to select the features, I ran the same training with omitting a single feature and do this for every feature. Also, as the feature
  ```

numbers decrease, the time decrease rationally which is also an expecting result. We can easily say that this method cannot be used when the feature count is too much since we cannot train the data again and again for all the features.