Implementation Details:

The program consists of the following:

1. Tweakable variables:
    a. number_of_clusters: This variable refers to the number of clusters i.e., value of 'K' in K-means clustering. (-1 by default which means perform kmeans for k = 1 to 10)
    b. standardize: A Boolean variable to be set 'True' if we want to standardize the data. (True by default)
    c. random.seed(int): The integer value can be anything. This is basically used to make the results more reproducable. (set value of seed to 0)
2. Methods:
    a. extractData(file): This method extracts the input data (X) and targets(y) from the file (iris.data).
    b. standardScaler(X): This function standardizes the given input X and returns X, mean of X, standard deviation of X.
    c. predictClasses(X, clusters): Given the cluster centers and X, this function predicts the labels for each data point in X.
    d. calculateDistortionScore(X,clusters): This function calculates the distortion score (Mean squares) of each point with each cluster center and returns the distortion score.
    e. Kmeans(X,mean_X,std_X): This function creates new cluster centers and then re-adjusts the centers based on Euclidean distances from points near to each cluster.
    f. main(): This function is the basic starting point for the program.

Sequence of steps:

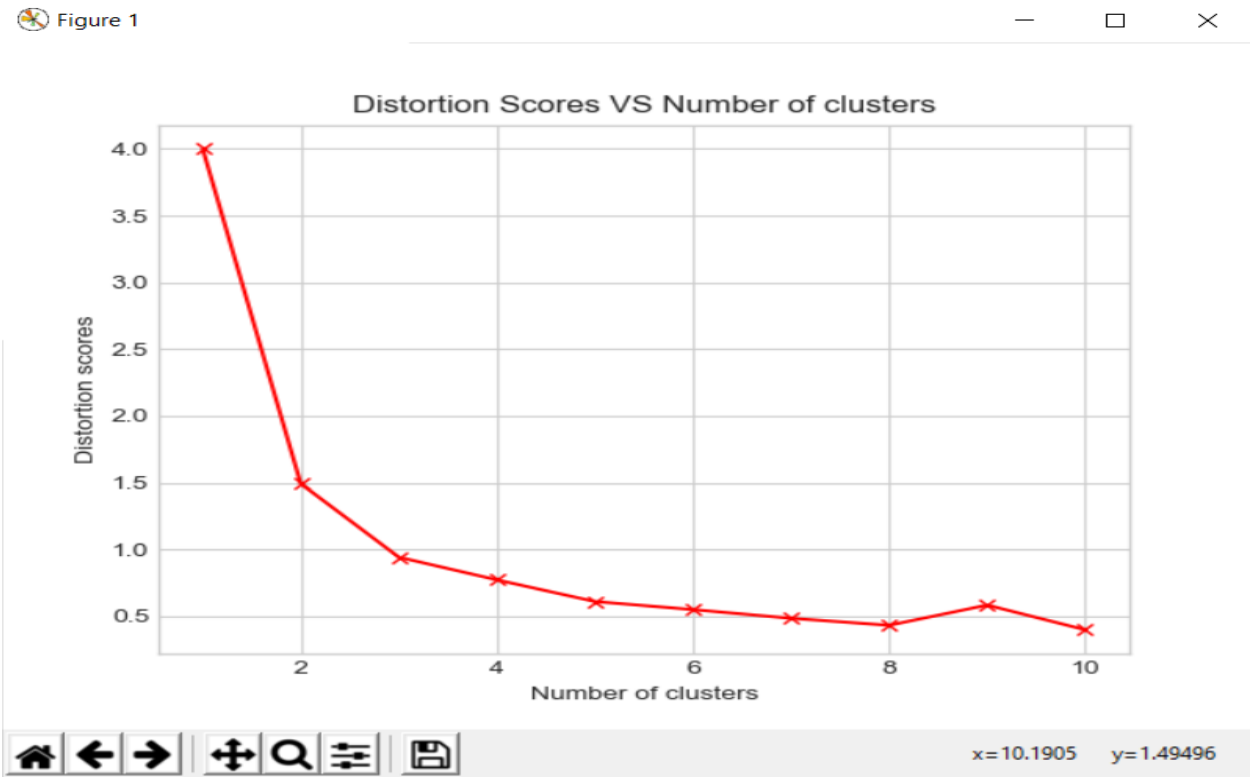
First, the **main** function is called. In this function we extract the features and target from the iris.data file. We standardize the X(We do this, since it makes computation efficient. As we are dealing with distances between centers of clusters and each point multiple number of times, standardizing the data will improve computational efficiency). For each value of k between 1 to 10, generate

'k' random cluster centers. Then, we get our predictions by calling predictClasses function(This function calculates the distances between each point and various cluster center and assigns the nearest cluster as the prediction) and recalculate the new centers. We calculate the new centers by taking average of the points within a cluster. If we think about it, there arises a case where no point is near to a cluster. In such cases I'm re-initializing the cluster center. We stop when we converge at a solution (i.e., The cluster centers don't change for 2 consecutive iterations then we can stop, since we have converged.)

Results:

- The output of the program is the list of variances, number of iterations to converge for different values of k, a plot of Distortion scores against the number of clusters in the form of a plot (Elbow plot). The accuracy obtained with the best value of k (i.e., k = 3).

To determine the number of clusters to use:



Elbow plot was used to find the number of clusters needed for the given input data. The plot refers that k=3 is the best value of 'k' to distinguish between different points i.e., to form 3 clusters is the best option.

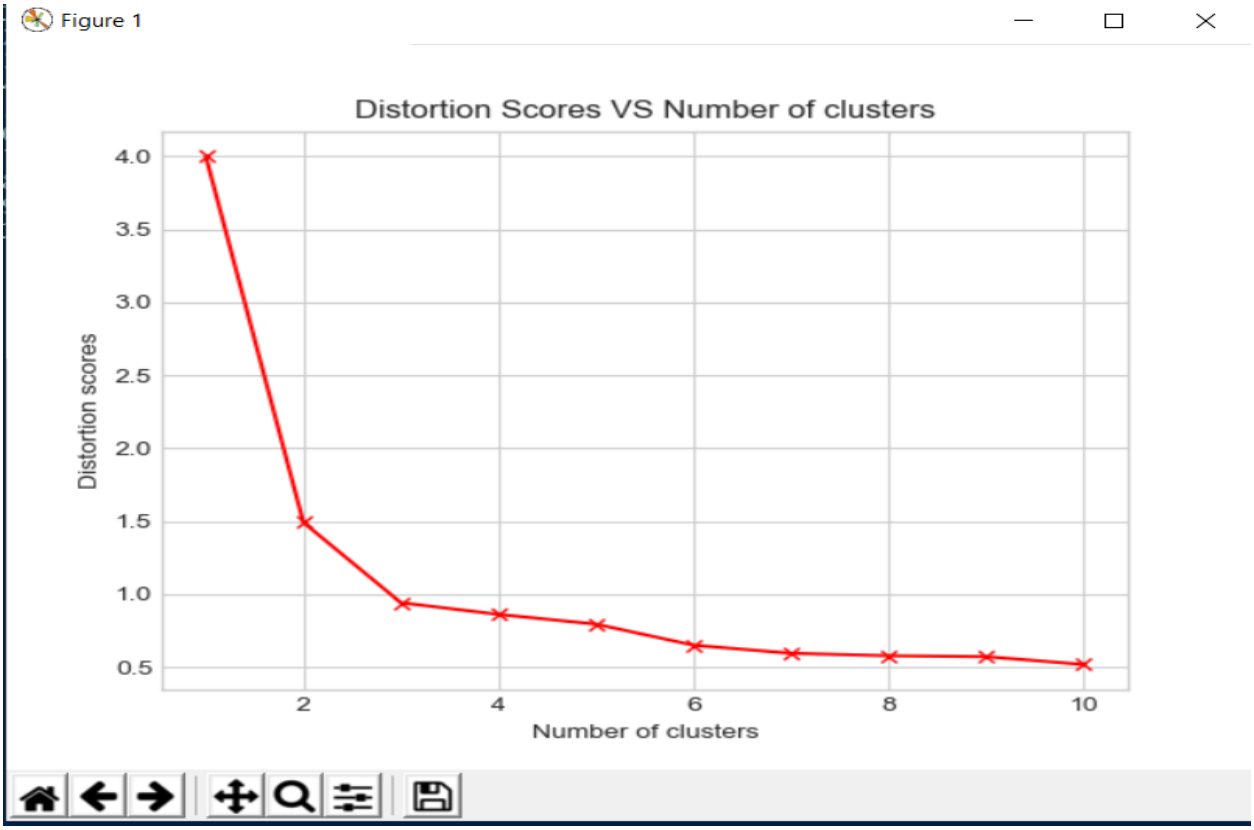
Observation:

1. From the elbow plot shown above. K=2 also works. This is because the iris versicolor and iris virginica have an overlap. If we think about this, iris-versicolor and iris-virginica can also be grouped together as a cluster, which is not what we want, but can be done from seeing the elbow plot.
2. The number of iterations increase as the value of k increases.

```
PS D:\UTA projects & notes\Machine Learning\projects\Project 3> python .\main.py
Finishing calculations and presenting the variances & number of iterations for given values of k(i.e., number of clusters):
For k=1: Cluster_variance = 4.000000000000002. Number of iterations to converge = 2
For k=2: Cluster_variance = 1.4915467049117546. Number of iterations to converge = 6
For k=3: Cluster_variance = 0.94102785422591. Number of iterations to converge = 6
For k=4: Cluster_variance = 0.7712048216573489. Number of iterations to converge = 10
For k=5: Cluster_variance = 0.6086362982711321. Number of iterations to converge = 11
For k=6: Cluster_variance = 0.5484137907838754. Number of iterations to converge = 12
For k=7: Cluster_variance = 0.4832681428900772. Number of iterations to converge = 9
For k=8: Cluster_variance = 0.4307329902619199. Number of iterations to converge = 10
For k=9: Cluster_variance = 0.5823815980320272. Number of iterations to converge = 8
For k=10: Cluster_variance = 0.39686048227394305. Number of iterations to converge = 9
================================================================================================
Based on the shown elbow plot. k=3 is the best value for k...
Results for k=3:
Accuracy = 85.33333333333334%
PS D:\UTA projects & notes\Machine Learning\projects\Project 3>
```

Finally, For k=3, since we already know what the data should be, we can check the accuracy of the predictions. When we check the accuracy its 85.33% on the results.

- If we don't standardize the data and apply kmeans clustering on the data then the number of iterations to converge increases by a lot and also the cluster variances go down by increase in value of k which can be seen from the image shown below.



The best value for k from the above plot is 3. The accuracy obtained for k=3 is 81.33% which can be seen below.

```
PS D:\UTA projects & notes\Machine Learning\projects\Project 3> python .\main.py
Finishing calculations and presenting the variances & number of iterations for given values of k(i.e., number of clusters):
For k=1: Cluster_variance = 4.000000000000002. Number of iterations to converge = 2
For k=2: Cluster_variance = 1.4915467049117546. Number of iterations to converge = 12
For k=3: Cluster_variance = 0.94102785422591. Number of iterations to converge = 17
For k=4: Cluster_variance = 0.8611110495388518. Number of iterations to converge = 16
For k=5: Cluster_variance = 0.7942713603533121. Number of iterations to converge = 87
For k=6: Cluster_variance = 0.649699310496128. Number of iterations to converge = 32
For k=7: Cluster_variance = 0.5944875629619296. Number of iterations to converge = 156
For k=8: Cluster_variance = 0.5777752164029825. Number of iterations to converge = 183
For k=9: Cluster_variance = 0.571461330111709. Number of iterations to converge = 162
For k=10: Cluster_variance = 0.5177421534871798. Number of iterations to converge = 352
================================================================================
Based on the shown elbow plot. k=3 is the best value for k...
Results for k=3:
Accuracy = 81.33333333333333%
PS D:\UTA projects & notes\Machine Learning\projects\Project 3>
```

Note:

1. For further understanding, please go through the code. I have written descriptive comments for all the functions and code written.
2. To run the program read the readme.txt file provided along with the code.