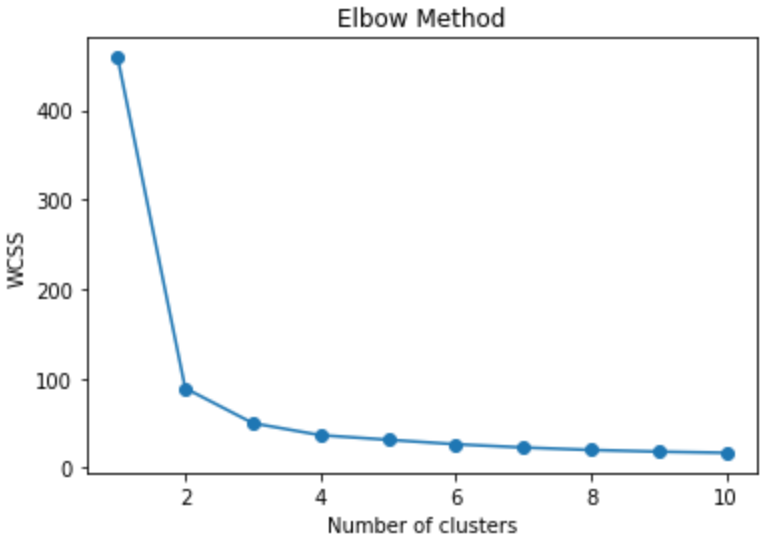
Answer 1:

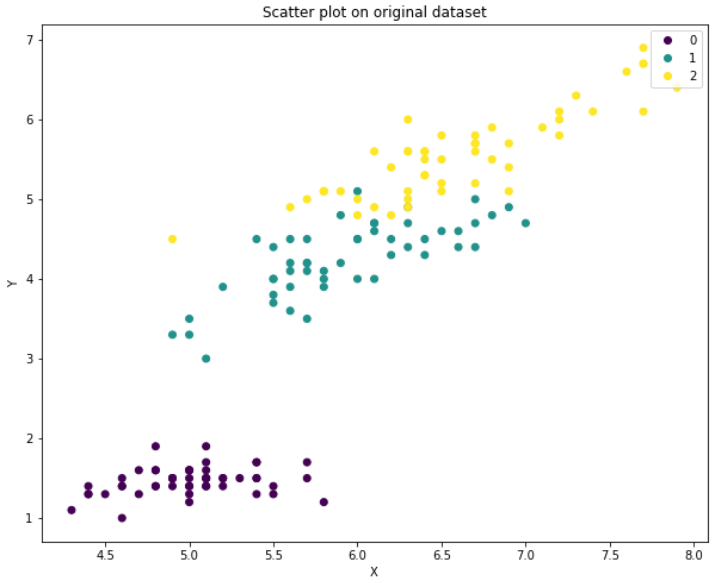
1) The Iris dataset is loaded into the python interface. The Iris dataset contains 150 instances, 4 attributes and a target class column (with classes: iris-setosa, iris-versicolor, iris-virginica). This dataset is splitted into the ratio of 70:30 using train_test_split from sklearn as implemented in the code.

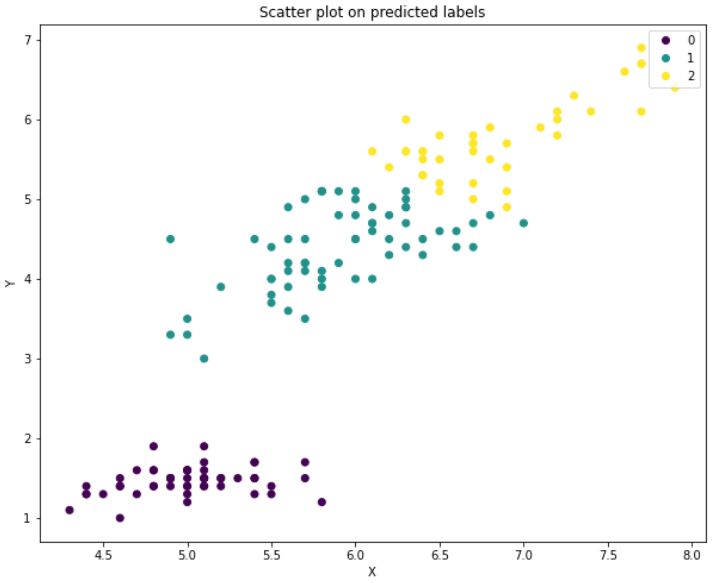
2) After implementing the K Means clustering algorithm, the plot for the errors (inertia or within cluster sum of squares) vs the number of clusters, is shown with the help of the following plot:



Thus from the above plot we see that the elbow forms at “number of clusters = 3”, therefore, we proceed taking 3 as the optimum number of clusters.

3) Scatter plot is used in order to visualize the datasets. In order for comparison, the dataset is visualized one by one with the original class labels and the predicted labels as follows:





Here, 0 refers to iris-setosa, 1 refers to iris-versicolor and 2 refers to iris-virginica (as they were converted into factors for easier data analysis).

4) Here, we calculated the accuracies based on the comparison between the original and the predicted labels and obtained the training and validation accuracies as below:

   Training accuracy: `87.61904761904762`
   Validation accuracy: `93.33333333333333`

Answer 2:

1) Loaded the database and split the dataset using stratify split into 70:30 ratio. Implemented in code.

2) Preprocessed the dataset by -
   (a) Removing punctuation signs
   (b) Lowercasing all words
   (c) Removing stopwords (use nltk library)
   Implemented in code.

3) Created a vocabulary of unique words from the training set.
   This vocabulary was used to design word count feature matrices where the (d,w) entry corresponds to the number of occurrences of word w in document d, both for training set and validation set. Implemented in code.

4) Implemented the multinomial Naive Bayes Algorithm with add-1 smoothing. Shown in code.

5) Training and validation accuracy. -

```
Test_accuracy :  0.7833333333333333
Train_accuracy :  0.9485714285714286
```

Some examples of the misclassified samples and comment as to why they may have been misclassified.

Naive Bayes ignores the order of words and predicts on individual features which are treated independently, thus does not give importance to the meaning or the semantics of the instance.

```
Instance number in test_set :  147   Predicted value :  0   Actual value :  1 original sentence :  Everyone is treated equally special.

Instance number in test_set :  148   Predicted value :  0   Actual value :  1 original sentence :  I didn't know pulled pork could be soooo delicious.

Instance number in test_set :  149   Predicted value :  1   Actual value :  0 original sentence :  Crust is not good.

Instance number in test_set :  150   Predicted value :  0   Actual value :  1 original sentence :  I *heart* this place.
```

Also, there is some noise in the training set also that I noticed.

```
Instance number in test_set :  157   Predicted value :  1   Actual value :  0 original sentence :  say bye bye to your tip lady!
```