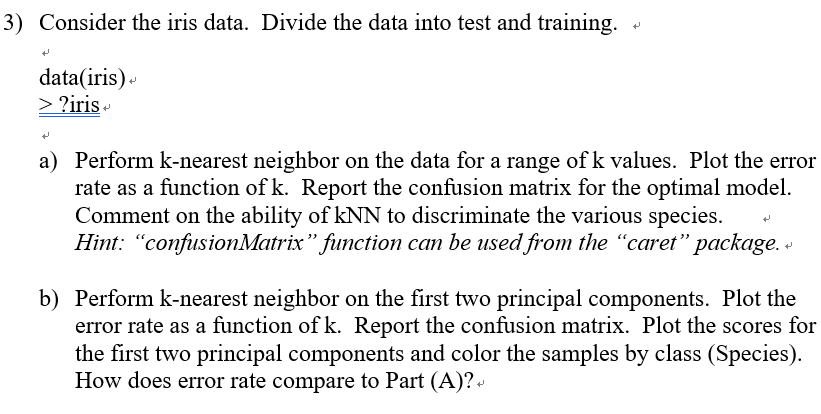
Question 3.



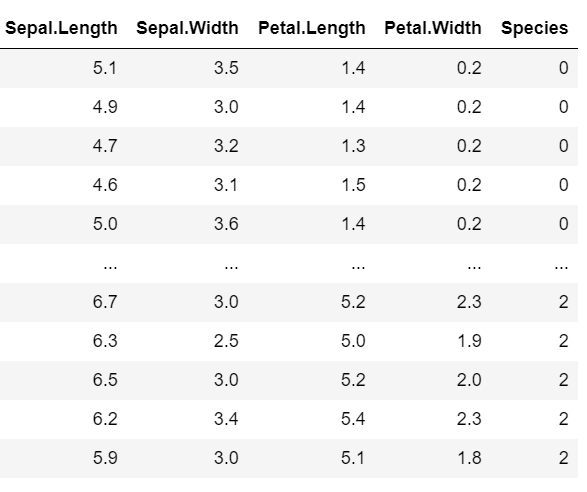
a)

Process:

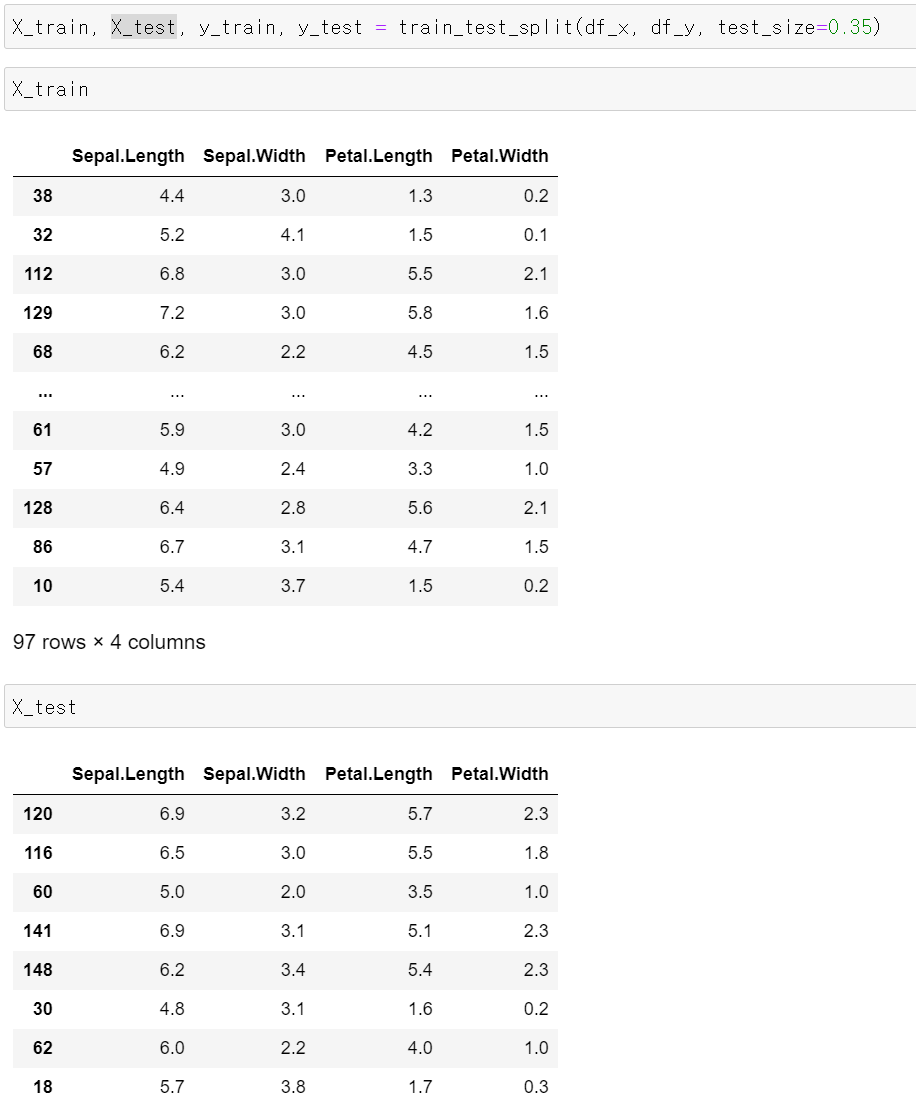
1. **Read data**: I read iris data.
2. **change string Y data into num value**: To verify the Y values, I used the ‘unique’ function and changed it into the number by using the ‘map’ function.
3. **Split Train and Test data**: There are only a few data in ‘iris’. Therefore, to check the test error more carefully, 35% of the data have become Test data.
4. **Fitting the KNN model with the test data**: I used the ’ KNeighborsClassifier’ function to make the KNN model. Also, to find the best k value for the accurate model, I compared the accuracy by the number of K.
5. **Show the Confusion matrix to the optimal model:** I found the best model with the k value from ‘step 4: **Fitting the KNN model with the test data**’. With the best model, I made the confusion matrix.

Outputs:

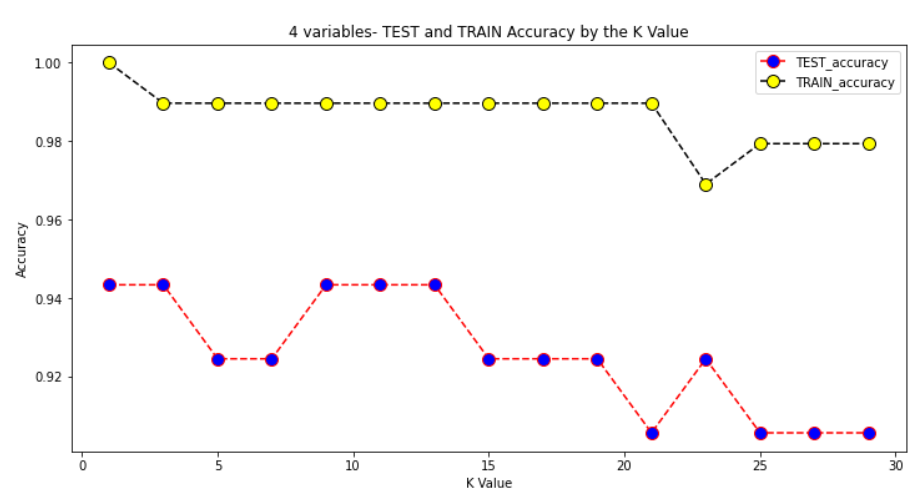
**Read iris data and change string Y data into number**



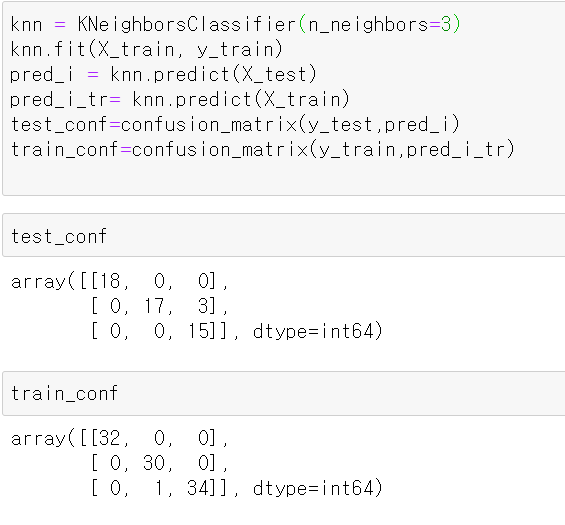
**Split Train and Test data**



**Fitting the KNN model with the test data**

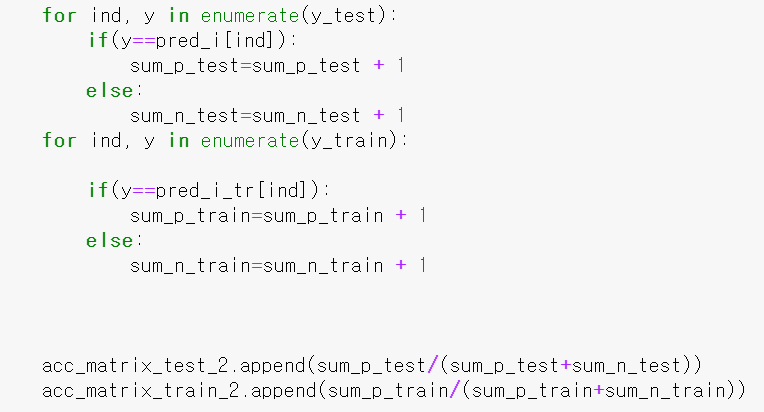


**Show the Confusion matrix to the optimal model (K=3)**



Discussion:

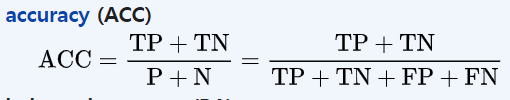
There are many criteria for evaluating models such as MSE, R^2, etc. Since this KNN model is made to classify the data, I evaluated the model with that how many predictions were corrected. It is based on the accuracy in the confusion matrix.



Let’s see the part of the code for finding the best model. After I get the test and train prediction list by each model, compare the prediction list with the list **Y**. If the prediction and the Y are same, increase the number of **correct predictions**. Else, the prediction and the Y are not same, increase the number of **incorrect predictions**. With these two numbers, I can make an accuracy for the model.

* The ACCURACY = **correct predictions** **/ (correct predictions + incorrect predictions)**

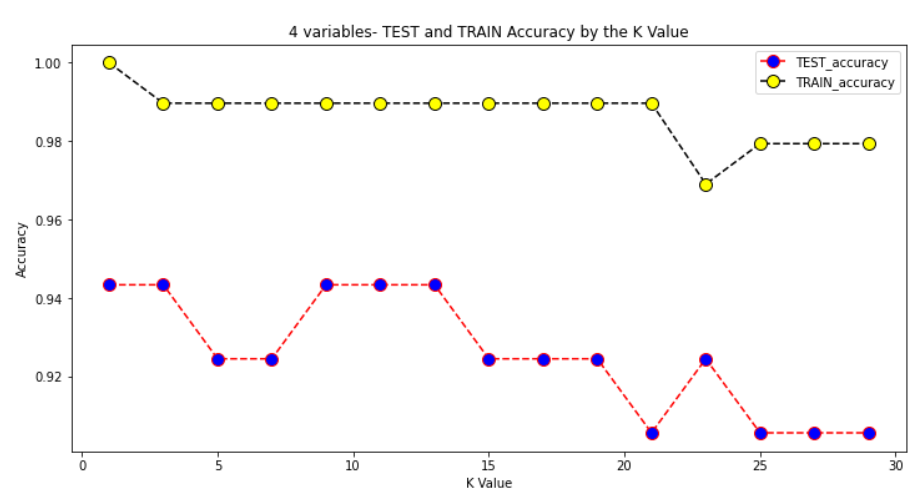
This accuracy also matches with the accuracy in the confusion matrix.



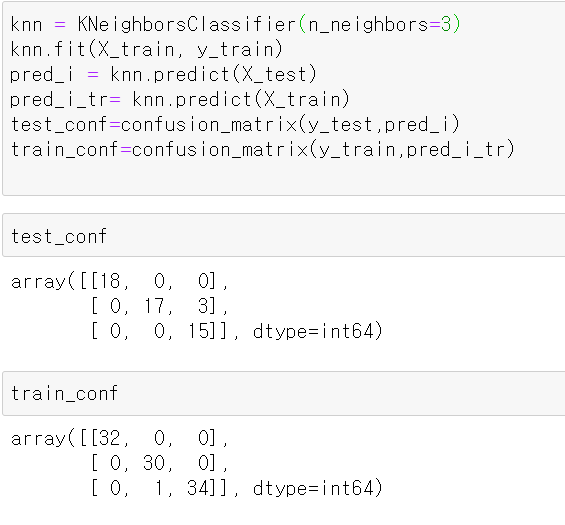
**(correct predictions + incorrect predictions)** indicate total population which is same with (TP+TN) + (FP+FN)

**correct predictions** indicate TP + TN.

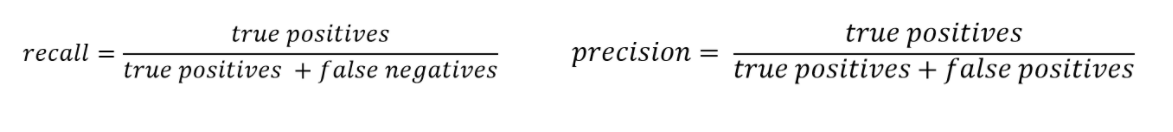
For the accuracy by the number of K values, I plotted Test and Train models and find out what is the best classifier model.

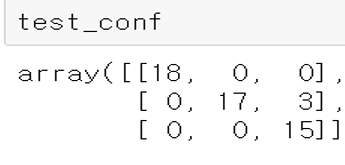


The above picture shows that most of the models have high prediction accuracy. Both Train and Test. The thing we should notice in this model, there are only 150 data to make and test the model. Models were more affected by test size or out liars than K values. Since the data scale is too small to decide the model with an accuracy rate. Therefore, I decided the ‘3’ K value as the best model by the trend that as K values larger, accuracy became lower.



The above picture shows the test and train confusion matrix with the best model. With this confusion matrix, I will examine the model not only the accuracy but also recall and precision value.





**In the test confusion matrix there are three values 'setosa':0, 'versicolor':1, 'virginica':2**

Precision (setosa) = 18 / 18 + 0 + 0 = 1

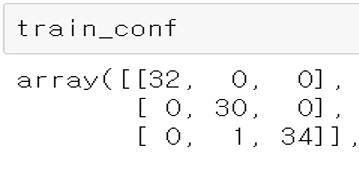
Precision (versicolor) = 17 / 17+ 0 + 0 = 1

Precision (virginica) = 15 / 15+ 3 + 0 = 15/18

Recall(setosa) = 18 / 18 + 0 + 0 = 1

Recall(versicolor) = 17 / 17 + 3 + 0 = 17/20

Recall(virginica) = 15 / 15+ 0 + 0 = 1



In the train confusion matrix there are same three values 'setosa':0, 'versicolor':1, 'virginica':2

Precision (setosa) = 32 / 32 + 0 + 0 = 1

Precision (versicolor) = 30 / 30+ 1 + 0 = 30/31

Precision (virginica) = 34 / 34+ 0 + 0 = 1

Recall(setosa) = 32 / 32 + 0 + 0 = 1

Recall(versicolor) = 30 / 30 + 0 + 0 = 1

Recall(virginica) = 34 / 34+ 1 + 0 = 34/35

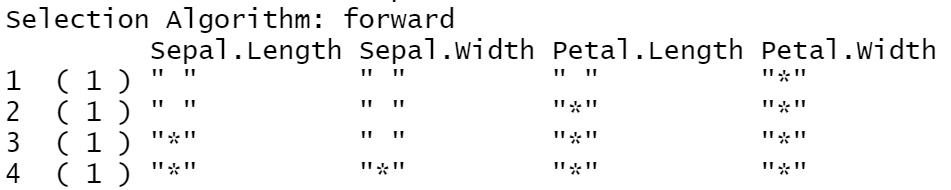
b)

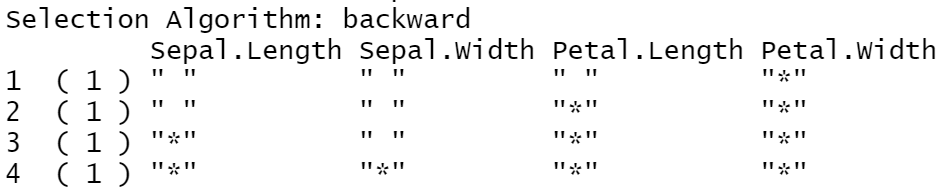
Process:

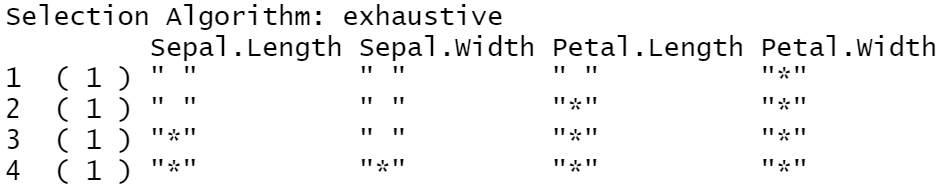
1. **Read data**: I read iris data.
2. **change string Y data into num value**: To verify the Y values, I used the ‘unique’ function and changed it into the number by using ‘map’ function.
3. **Split Train and Test data**: There are only a few data in ‘iris’. Therefore, to check the test error more carefully, 35% of the data have become Test data.
4. **Find the best two columns for the KNN model**: To find the best two columns for the KNN model, I used the best subset selection, forward subset selection, and backward subset selection. After I found two columns for the model, I modified the data into these two columns.
5. **Fitting the KNN model with the modified test data**: I used the ’ KNeighborsClassifier’ function to make the KNN model. Also, to find the best k value for the accurate model, I compared the accuracy by the number of K.
6. **Show the Confusion matrix to the optimal model:** I found the best model with the k value from ‘step 5: **Fitting the KNN model with the modified test data**’. With the best model, I made the confusion matrix.

Outputs:

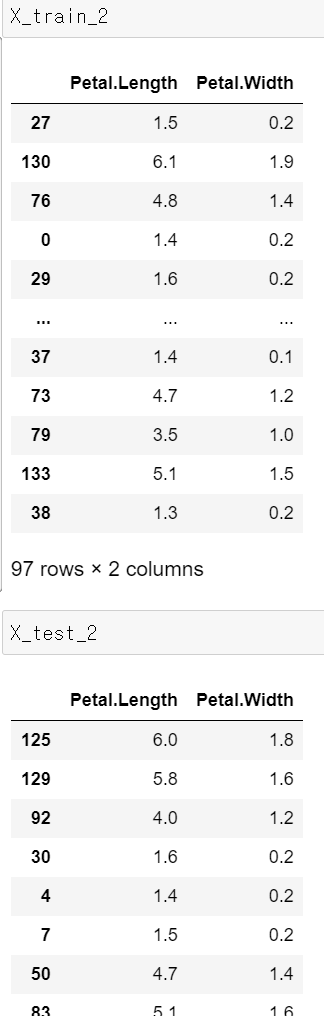
**Find the best two columns for the KNN model using summary function.**



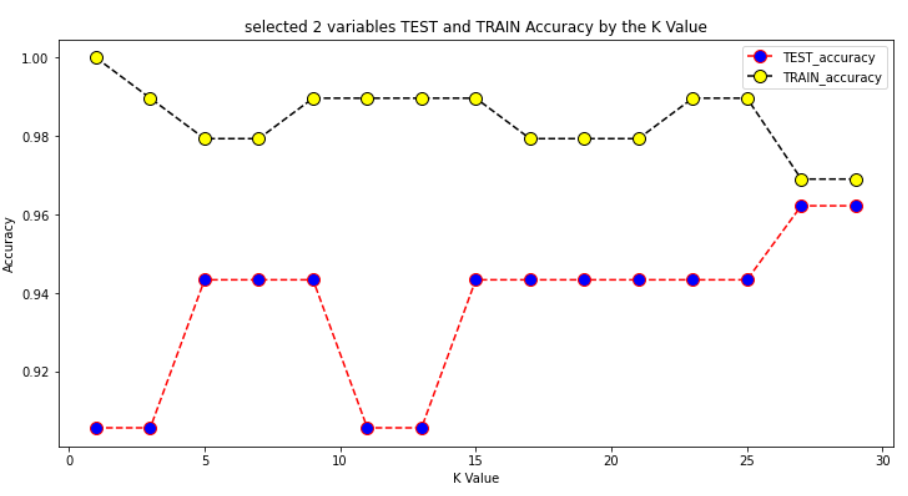




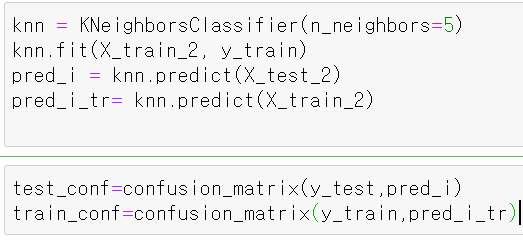
**modified the data into these two columns**

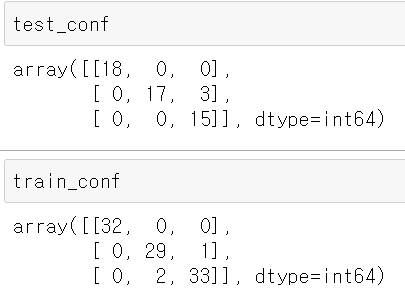


**Fitting the KNN model with the modified test data by K values.**



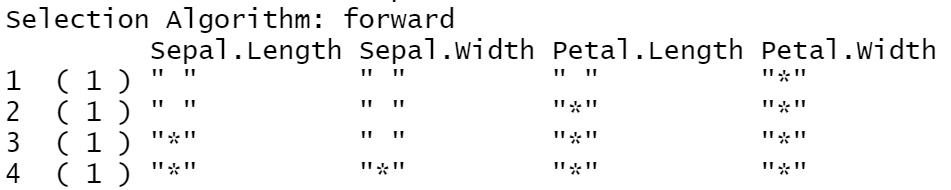
**Show the Confusion matrix to the optimal model(k=5)**

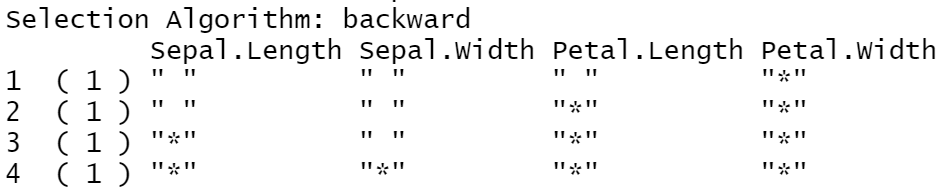


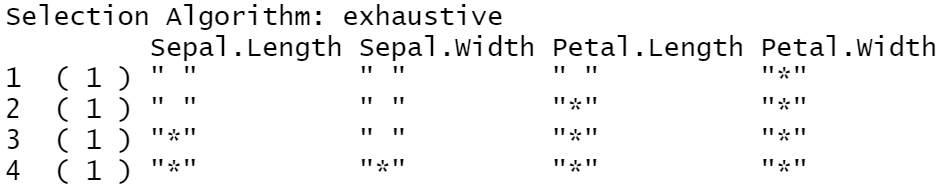


Discussion:

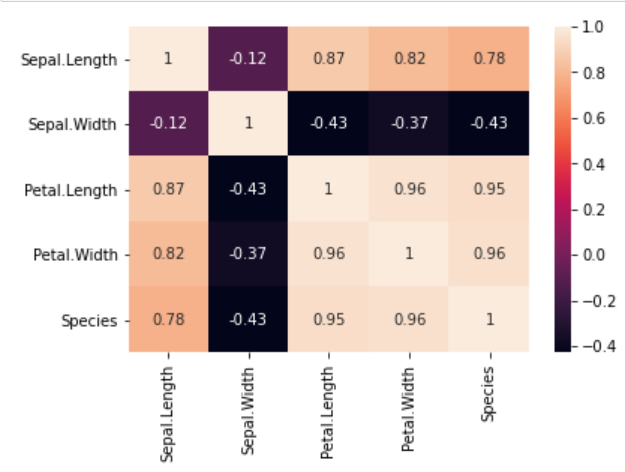
To find out the best two columns for the KNN model, I used forward, backward, and the best subset selection.





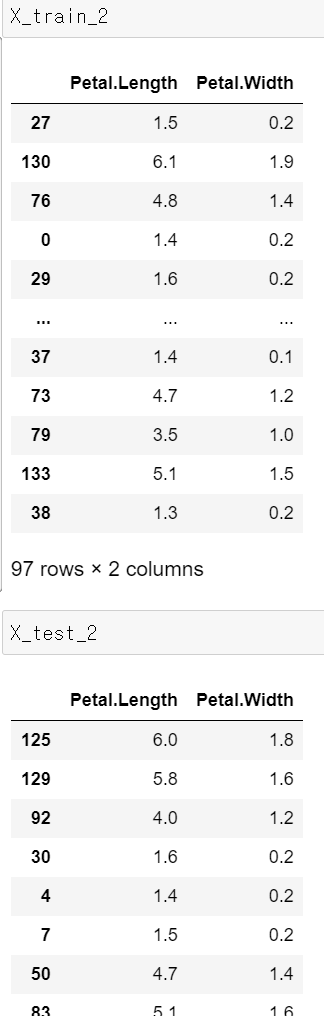


All three-subset selection model show that ‘Petal.Lengh’ and ‘Petal.Width’ are the important variables to make the precise model.

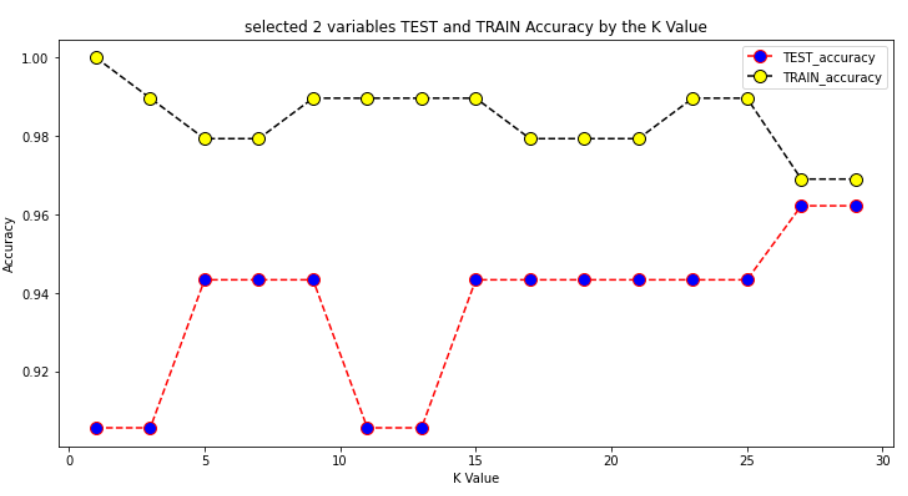


According to Correlation Heatmap shows ‘Petal.Lengh’ and ‘Petal.Width’ are highly related to Species.

Therefore I decided to select ‘Petal.Lengh’ and ‘Petal.Width’ for the new KNN model.

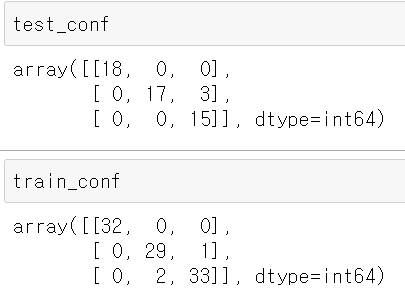


I evaluated the model with that how many predictions were corrected. It is based on the accuracy in the confusion matrix same as above.



The above picture shows that most of the models have high prediction accuracy as like the previous model. Both the Train and the Test. The problem was similar to the previous model. Data for train and test was too small. The Models were more affected by test size or out liars data than K values. Since the data scale is too small to decide the model with an accuracy rate. Although Test Accuracy looks good at the high value of K, it was hard to decide the model only with the accuracy rate because of the test data scale.

I selected the best KNN model with the 5 K values, and made a confusion matrix with that.



**In the test confusion matrix there are three values 'setosa':0, 'versicolor':1, 'virginica':2**

Precision (setosa) = 18 / 18 + 0 + 0 = 1

Precision (versicolor) = 17 / 17+ 0 + 0 = 1

Precision (virginica) = 15 / 15+ 3 + 0 = 15/18

Recall(setosa) = 18 / 18 + 0 + 0 = 1

Recall(versicolor) = 17 / 17 + 3 + 0 = 17/20

Recall(virginica) = 15 / 15+ 0 + 0 = 1

**In the train confusion matrix there are three values 'setosa':0, 'versicolor':1, 'virginica':2**

Precision (setosa) = 32 / 32 + 0 + 0 = 1

Precision (versicolor) = 29 / 29+ 2 + 0 = 29/31

Precision (virginica) = 33 / 33+ 1 + 0 = 33/34

Recall(setosa) = 32 / 32 + 0 + 0 = 1

Recall(versicolor) = 29 / 29 + 1 + 0 = 29/30

Recall(virginica) = 33 / 33+ 2 + 0 = 33/35

Both models (the 4 variables KNN model, the 2 variables KNN) have similar accuracy. This result reveals that with only 2 columns, it was enough to make the precise model. Since the data is too small, it is hard to make a conclusion about which model is better. However, I can infer with this conclusion that the 2 variable models can make a clearer model with a large data set because the other 2 unselected variables can cause some noise or the dimensional cures for the model.

reference

https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c