

Report Assignment 2

(Calculations & Programming)

Applied Machine Learning ELG5255[EG]

**Group Number: G\_26**

**Prepared by:**

**Sawsan Awad (300327224)**

**Sondos Ali (300327219)**

**Toka Mostafa (300327284)**

##### Part 1: Calculations

##### Suppose we have some data collected from a cloth shop, and the dataset contains three features. The first feature is the cloth color (x1), the second feature is the consumer’s gender(x2), and the third feature is the price (x3) (we simplify the problem and use high, medium, and low to present different prices). The label TARGET (y) is whether the consumer buys the cloth. Suppose we have the following training data including 15 training samples. Using Bayesian Rule-Based Classifier to make a prediction when Color = G, Gender = F, Price=H. Please include the detailed calculation process.

##### Notes: In the color row, R, G, and Y are short for Red, Green, and Yellow; in the Gender row, M and F mean Male and Female, respectively; in the Price row, H, M, and L stand for High Prices, Medium Price and Low Prices, respectively and in the Target row, N and Y are No and Yes

A picture containing text, shoji, crossword puzzle, clock

Description automatically generated

##### Solution

P(G,F,H) =

=

1. Consider the following loss table, which contains three actions and two classes. Calculate the expected risk of three actions, and determine the rejection area of P(Class1| x).

Table

Description automatically generated

**Solution**

----------------(1)

--------------------(2)

----------------------------------(3)

We choose if:

We choose if:

The rejection area of P (Class1| x) =

##### Part 2: Programming

##### Naïve Bayesian Classifier

##### Importing the libraries & datasets:

##### We import all libraries we need in our problem.

##### For Preprocessing:

##### Pandas

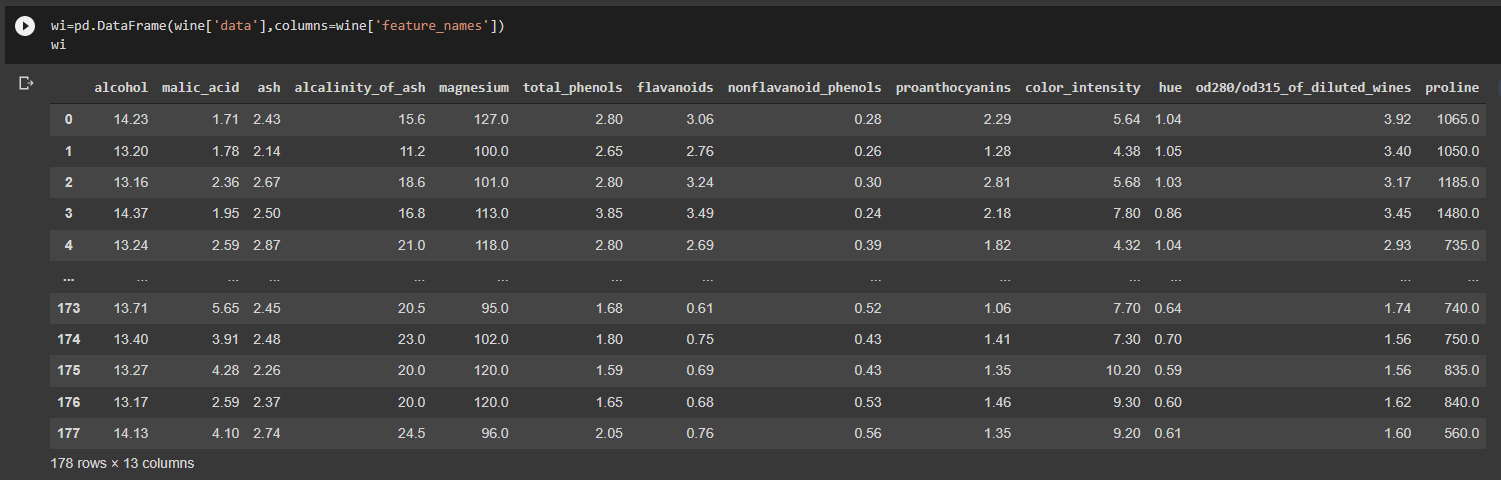
##### NumPy

##### For Visualization:

##### Matplotlib

##### Load datasets:

* We load wine dataset. There are 3 classes in this dataset, and each sample in this dataset has 13 features.

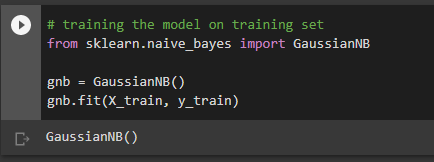


##### Preprocessing:

* We split the data into X, y
* We split X and y into training and testing sets

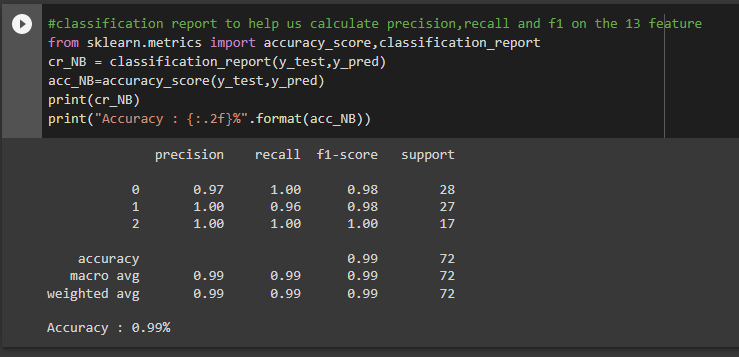
##### Modeling

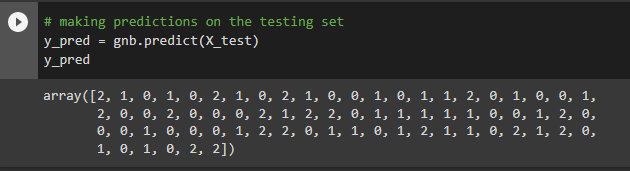
Naïve Bayes Model:

* We train the model on training set. We use Naïve Bayes Model for solving classification problems
* We make predictions on the testing set.

##### Classification report:

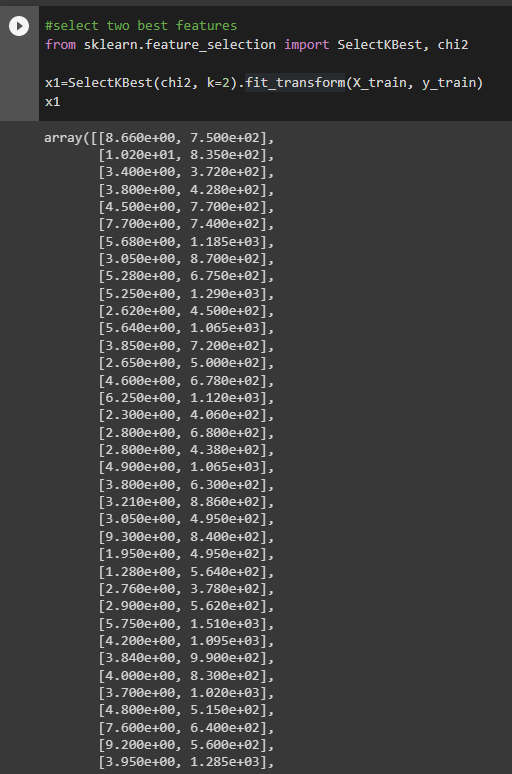
We use classification report function to help us calculate precision, recall and f1.

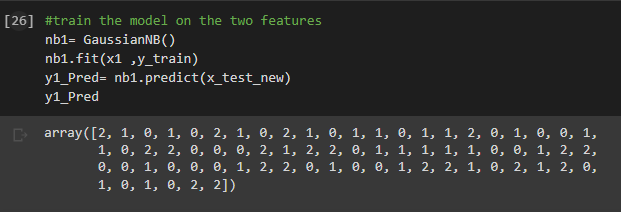
**Model accuracy: 99%**



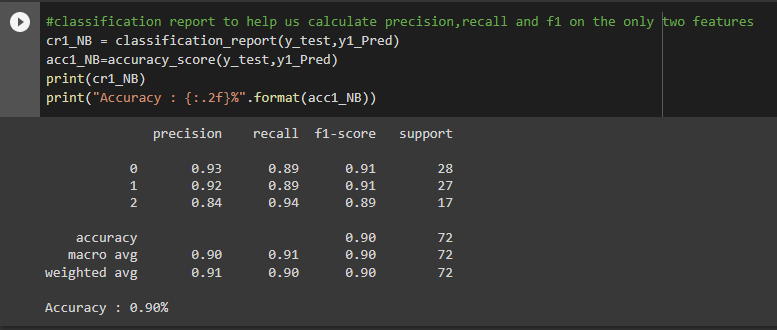
##### Feature Selection:

We use SelectKBest () method to select the best two features. And the best two features were (color\_intensity , proline).

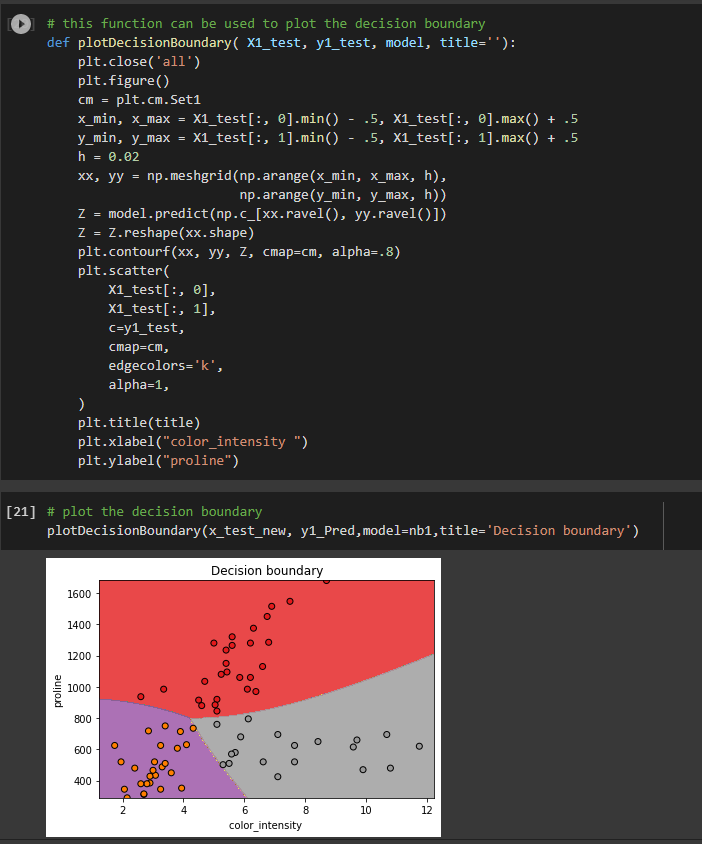


****We used the two features to train the model.

* We calculated accuracy, precision, recall and f1 on the two features

** Model accuracy: 90%**

* The decision boundary on the test set on the two features:



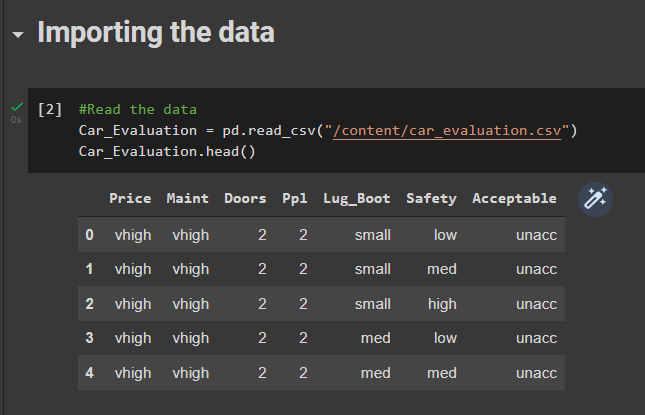
##### KNN Classifier

##### Importing the libraries & datasets:

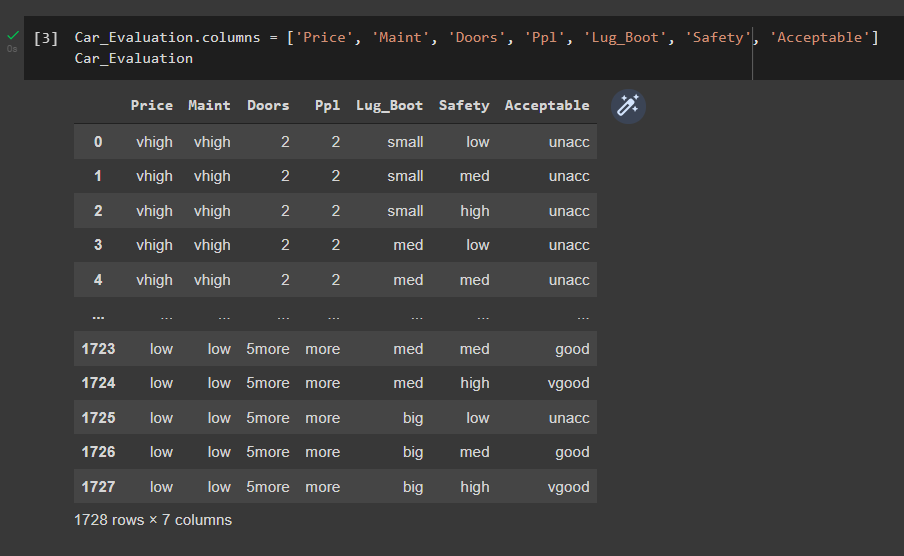
##### We import all libraries we need in our problem.

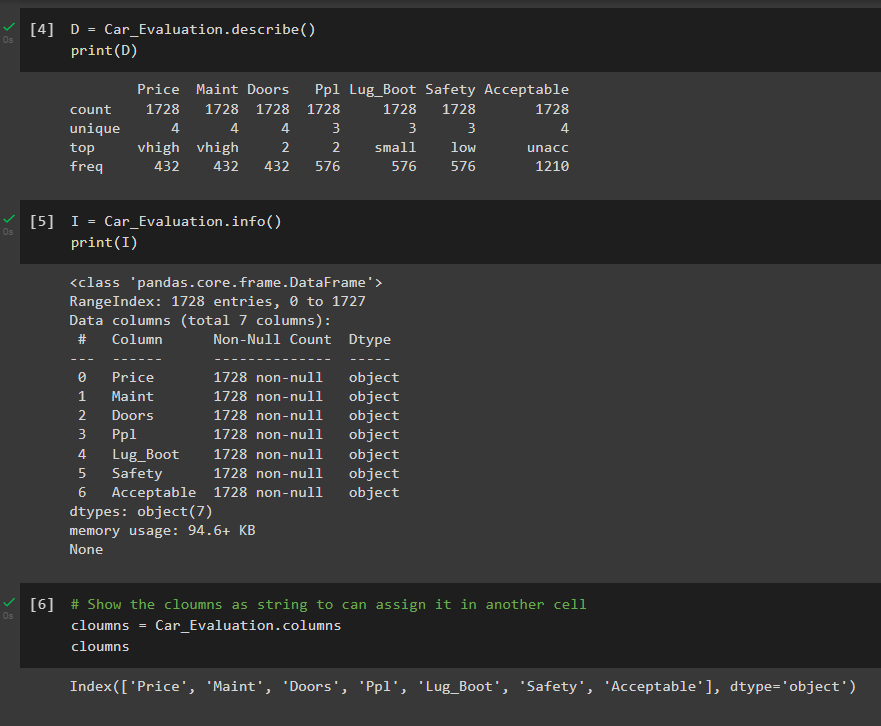
##### Load datasets:

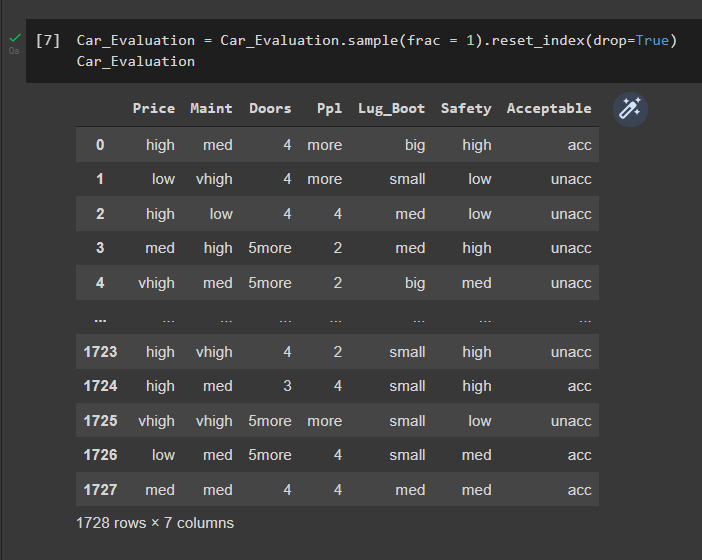
* First: we installed dataset from Kaggle as csv file.
* Second: we loaded it in google colab.
* Third: we used .head() to show first 5 rows.

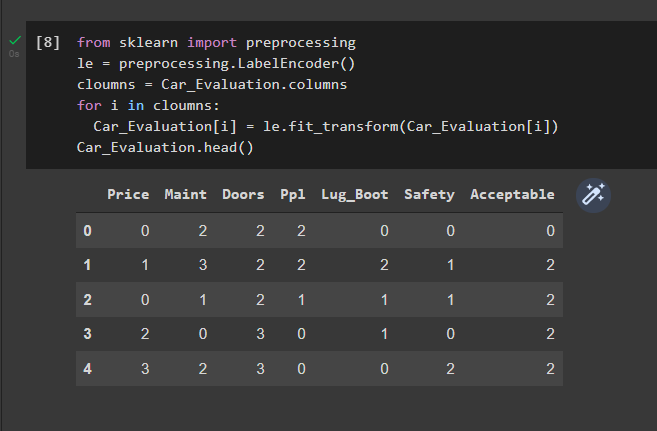


* 1. Manipulating the dataset



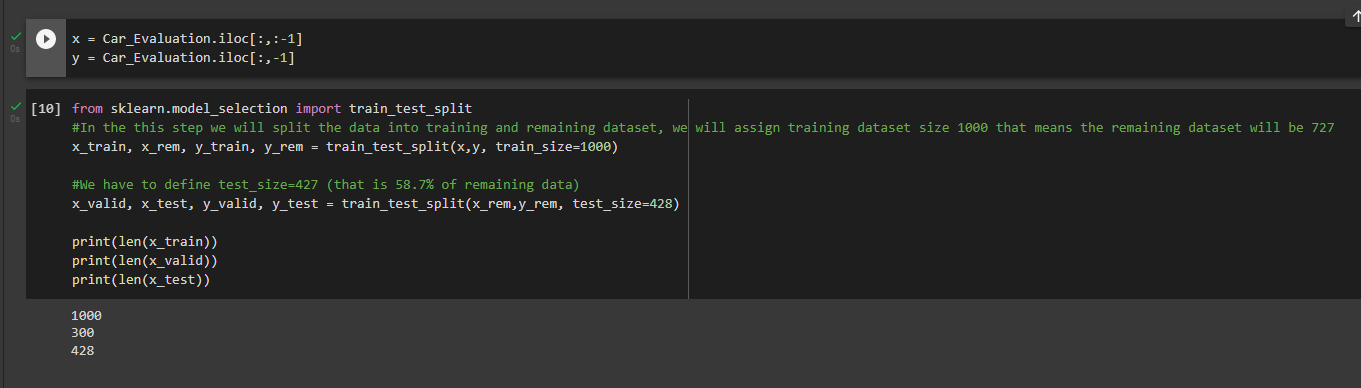


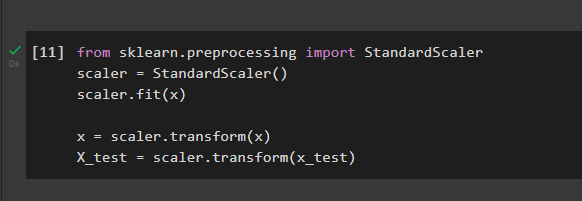
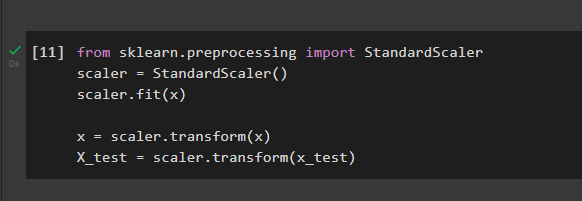
* 1. Shuffle the data
  2. Label Encoder
* To represent the attributes by string values.

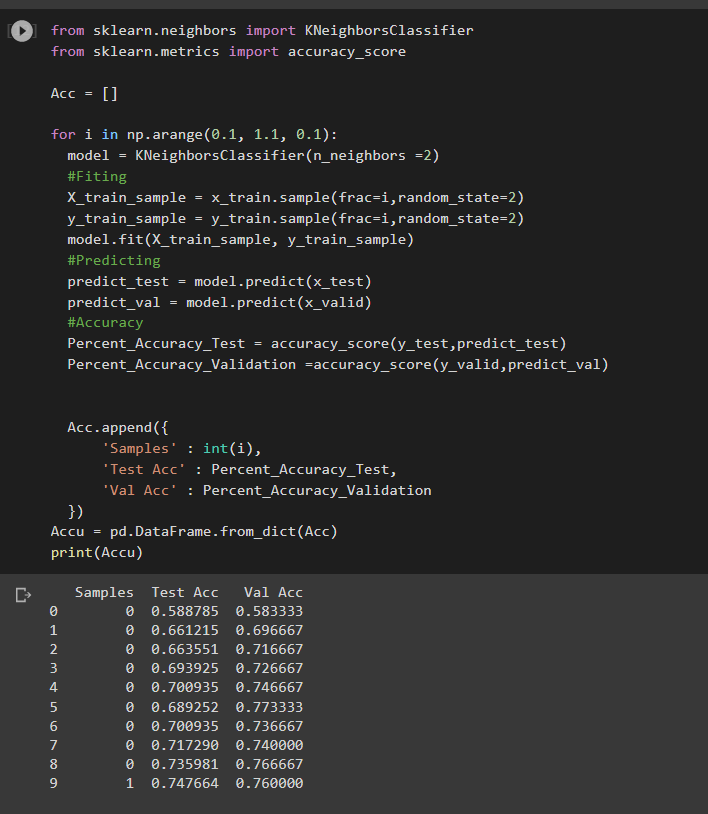


\

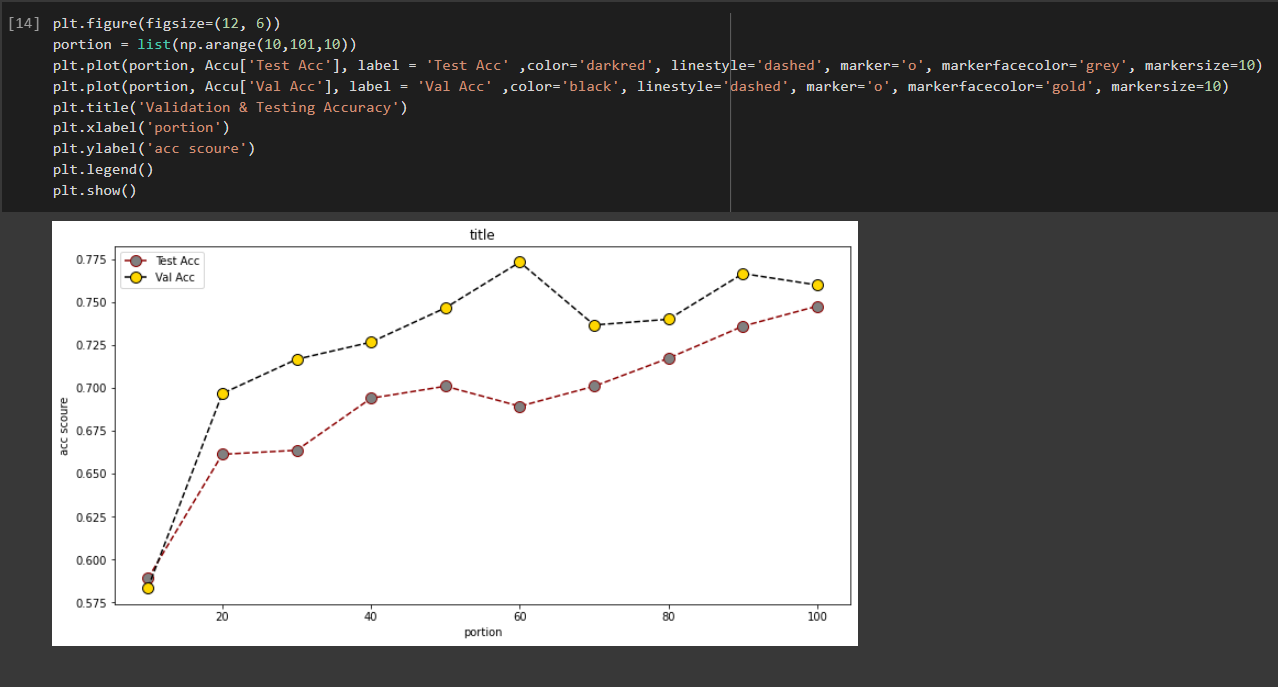
* 1. Training and Splitting the data to 3 sets
* First: we split the data into training and remaining dataset, we assigned training dataset size as 1000 that means the remaining dataset will be 728.
* Second: We split the remaining dataset to test and validation sets, we defined test size as 428.

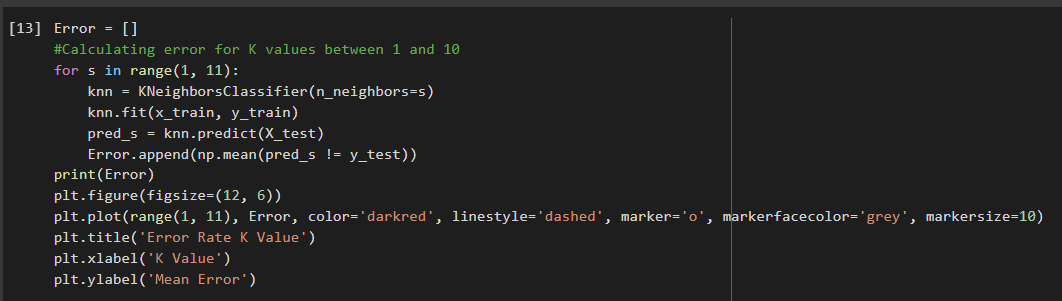


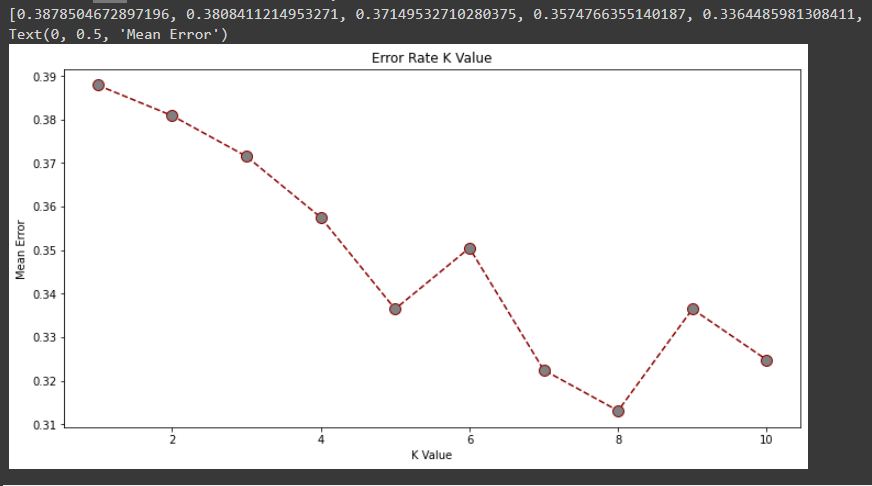
* 1. Feature Scaling
* We scaled the features so that all of them can be uniformly evaluated.
  1. Training and Predictions (KNN Classifier)
* First: we used different number of training samples to show the impact of number of training samples, and used 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training set for 10 separate KNN classifiers and specified a fixed K=2 value.
* Second: model made predicting to x\_test and x\_validation.
* Third: we calculated the accuracy to test and validation.



* 1. Performance of the Validation set and Testing set
* We showed the performance (accuracy score) of the validation set and testing set and made X axis the portion of the training set, Y axis the accuracy score. There are two lines in total, one is for the validation set and another is for the testing set.

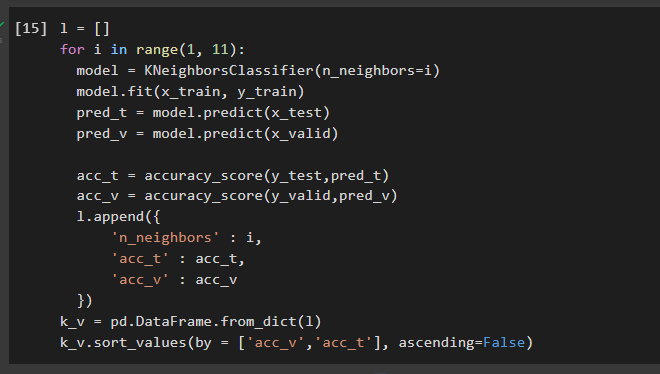


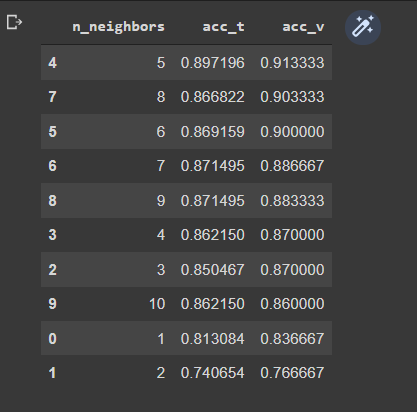
* 1. Comparing Error Rate with the K Value
* We use Error Rate way because it is a one way to help us find the best value of K by plotting the graph of K value and the corresponding error rate for the dataset.
* We noticed in the output that the mean error is zero when the value of the K is 8, but it would be change cause the random sets which model would use it in the next run.



* 1. Use 100% of training samples & Find the best K value
* We selected the best K value with the highest validation and test accuracy by using all of the training data
* **The best K value is: 5**

**Validation Accuracy is: 91%**

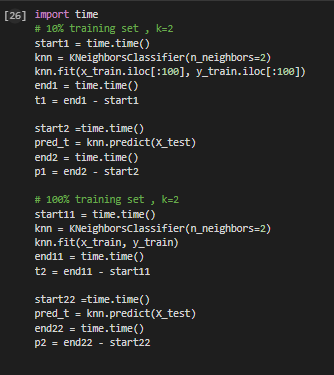
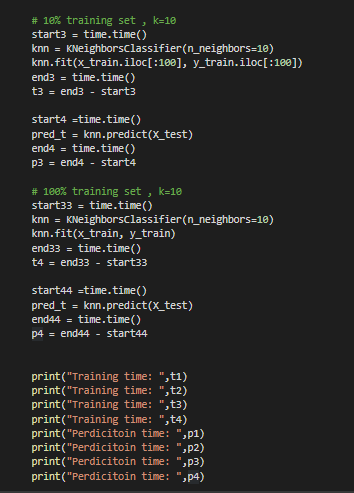


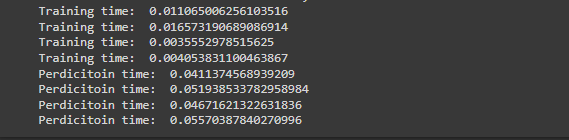


* + 1. . Plotting to Validation Accuracy for each sample

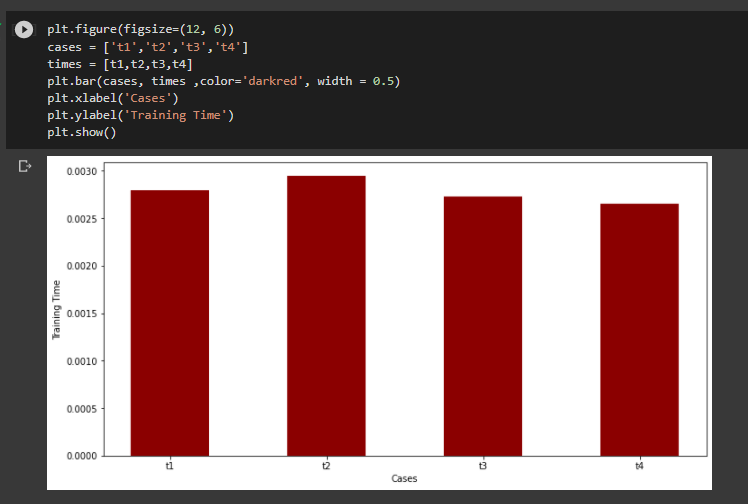


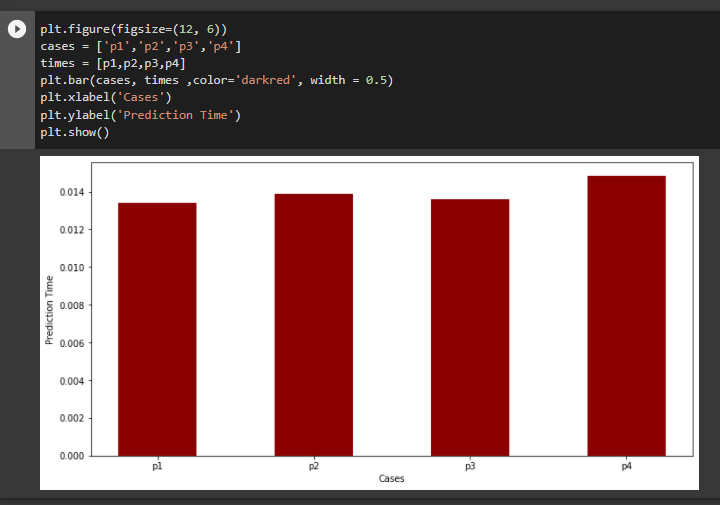
* 1. Analysis the training time when use different number of training samples
* We got 4 cases:
  + 10% of the whole training set and K = 2   
    **Training\_Time = 0.010582208633422852  
    Prediction\_Time = 0.0411374568939209**
  + 100% of the whole training set and K = 2  
    **Training\_Time = 0.002919435501098633  
    Prediction\_Time = 0.051938533782958984**
  + 10% of the whole training set and K = 10  
    **Training\_Time = 0.0032396316528320312  
    Prediction\_Time = 0.04671621322631836**
  + 100% of the whole training set and K = 10.  
    **Training\_Time = 0.002959728240966797  
    Prediction\_Time = 0.05570387840270996**





* + 1. Plotting the Training and Prediction Time to 4 cases



**Conclusion**

From applying KNN model, we found that:

1. According to point(c), after we are modeling the car evaluation with KNN algorithm. when we use 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training set for 10 separate KNN classifiers we notice that the performance increase as training set increase.

* **Note: The variance and testing sets become more closer to each other which avoid overfitting.**

1. According to point(d), when we use 100% training set and varying with k from 1 to 10, we notice that the best k in 5 and 8 but 5 is more efficient because the variance between testing and validation is very small than k=8. Another reason makes k=5 is best is avoiding our model to become confused in future unseen data if the two classes have the same amount of points (4 points in each).

* **Note: It prefers selecting an odd value for K to get the best accuracy.**

1. According to point(e), when we analyze the training time by using different number of training samples, we notice the training time become little bit larger when the k =2 which is make sense because it makes more calculation.

* **Note: The size of the samples doesn’t have affection on the time of training and prediction**