**NAME:** MOHD IRFAN

**USN:** 1BM20CS409

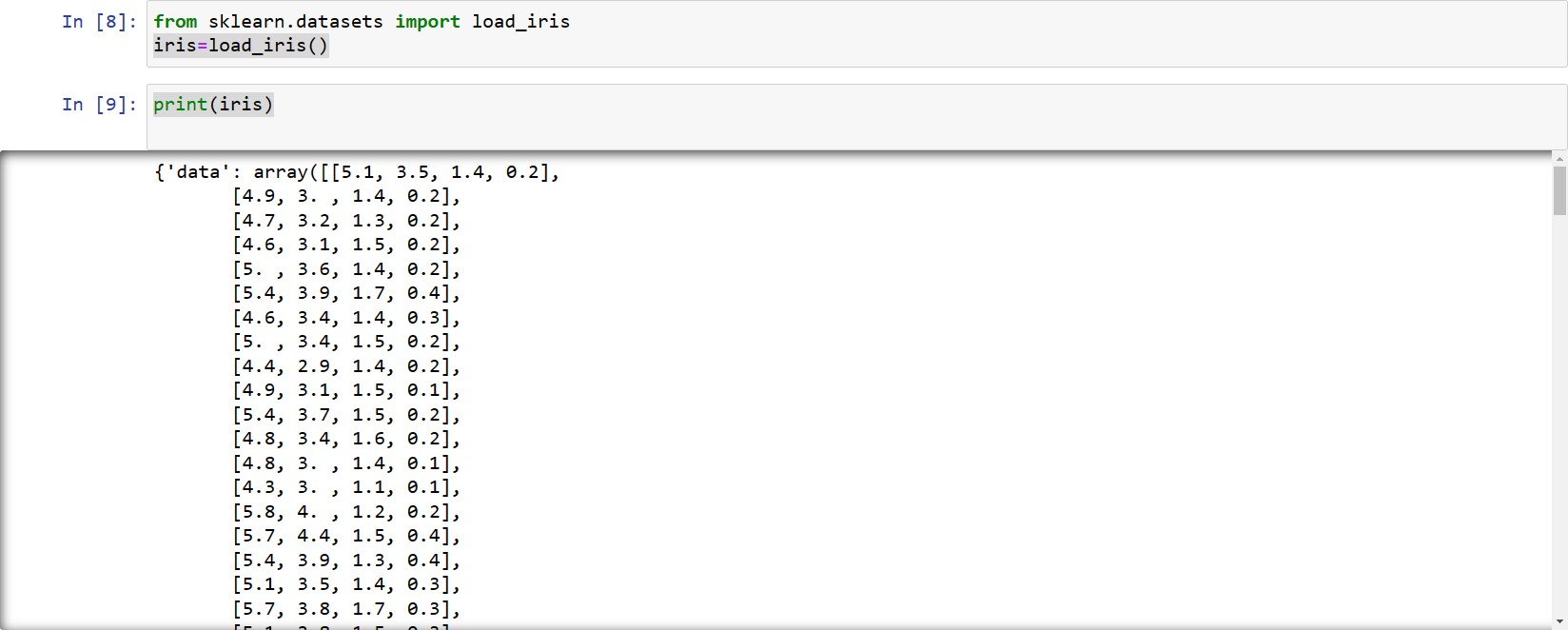
MACHINE LEARNING LAB OBSERVATION

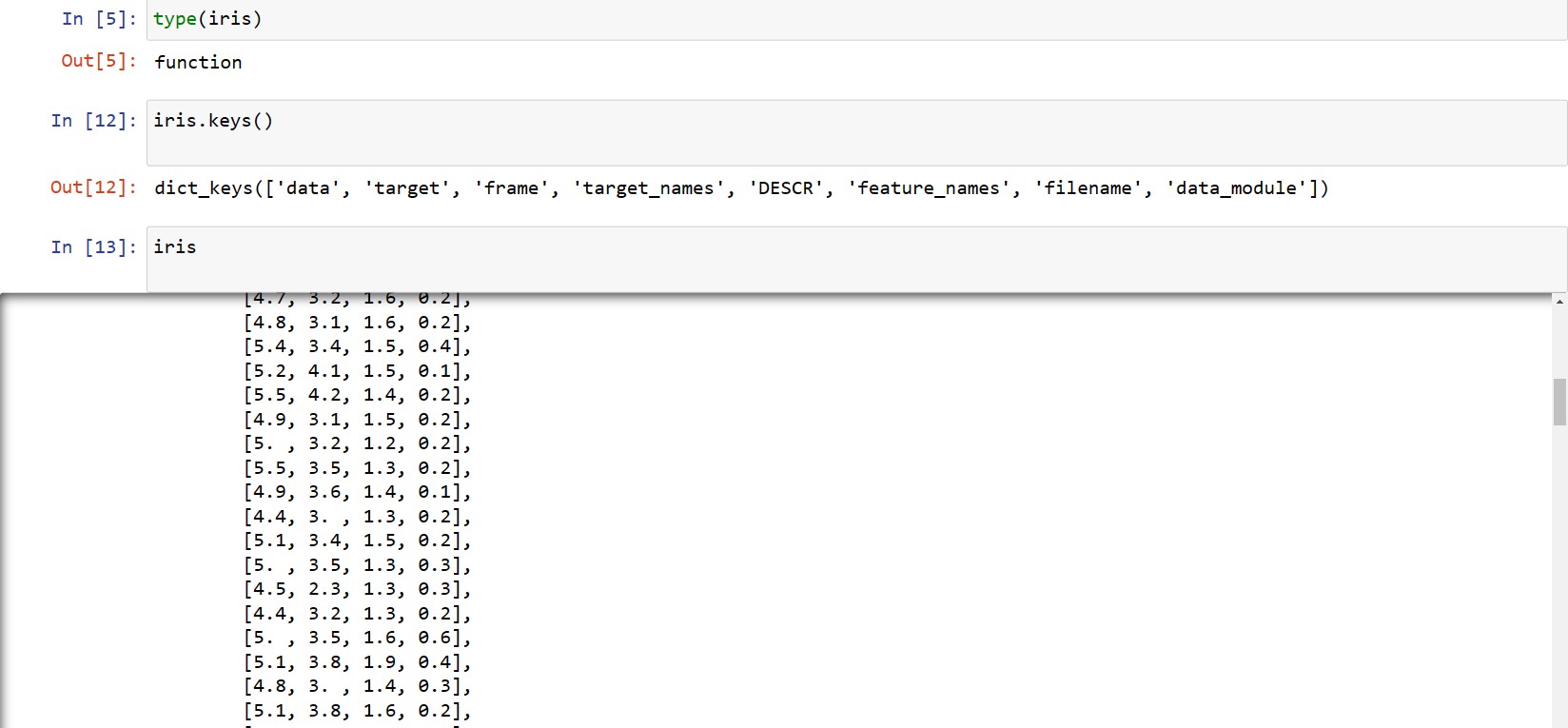
Date: 1-04-2023

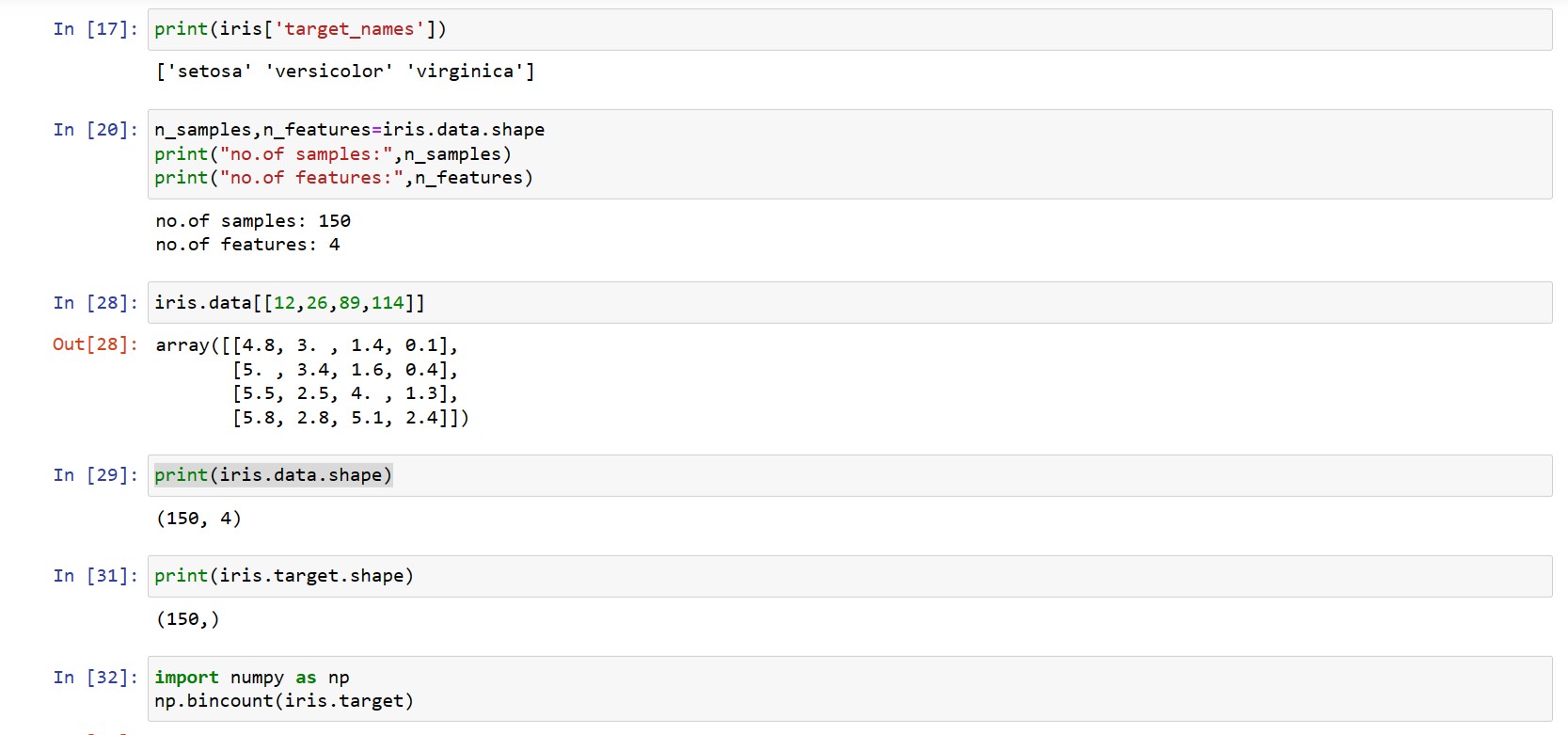
Lab 1: Exploring Datasets IRIS DATASET:

* Features in the Iris dataset:
  1. sepal length in cm
  2. sepal width in cm
  3. petal length in cm
  4. petal width in cm
* Target classes to predict:

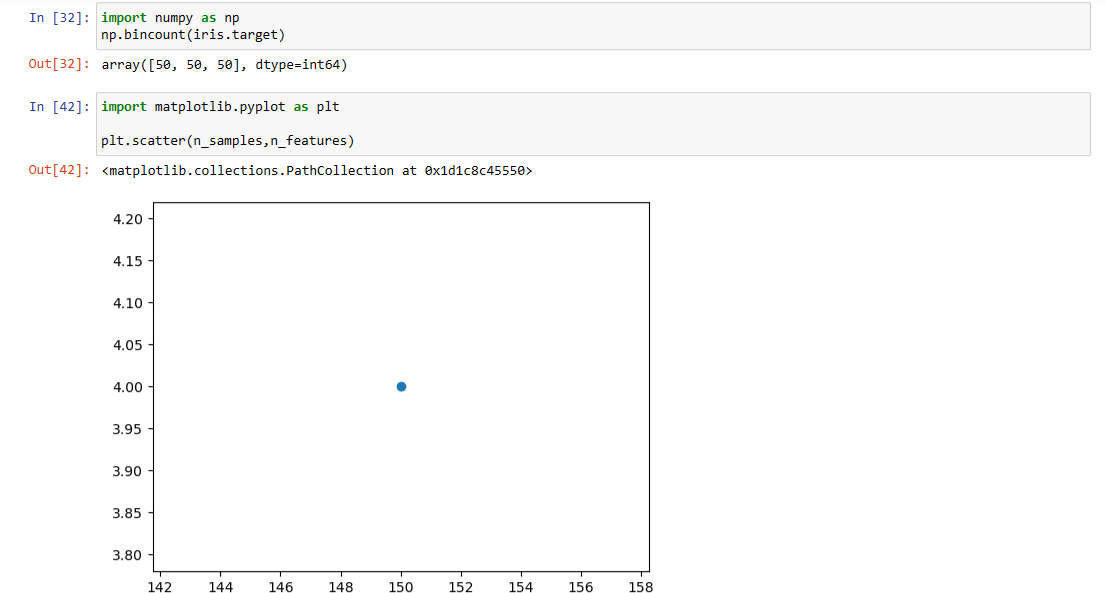
1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica



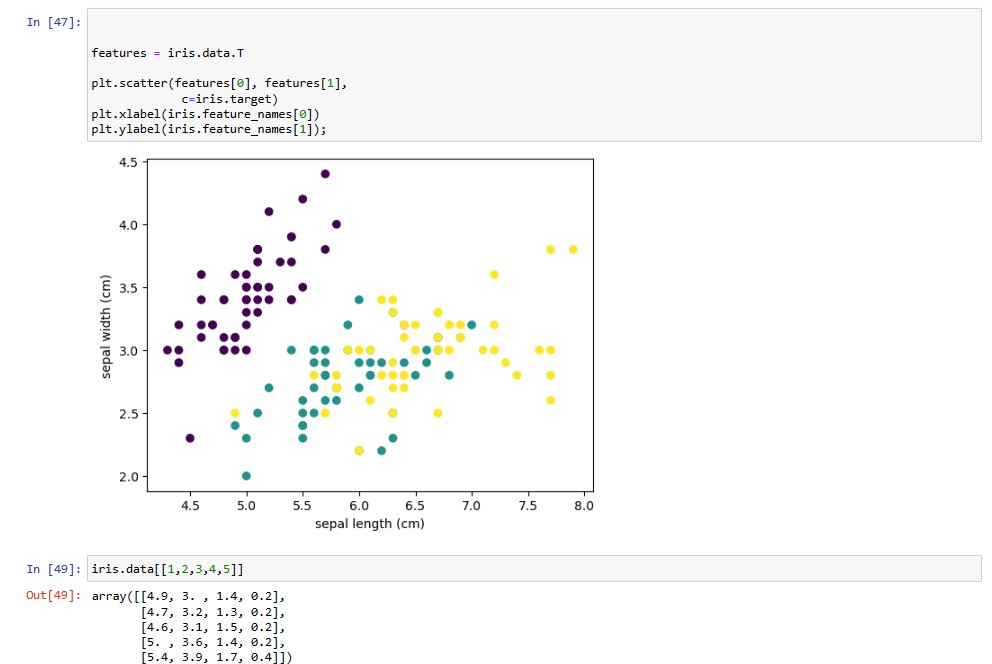




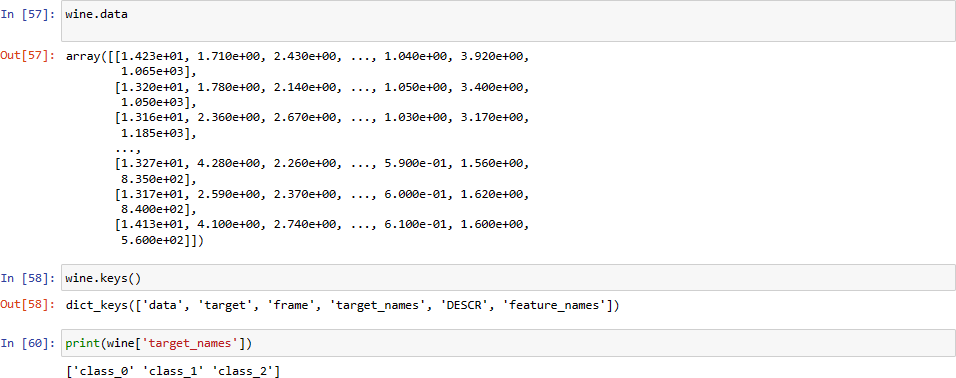
Scattered graph for samples vs features.

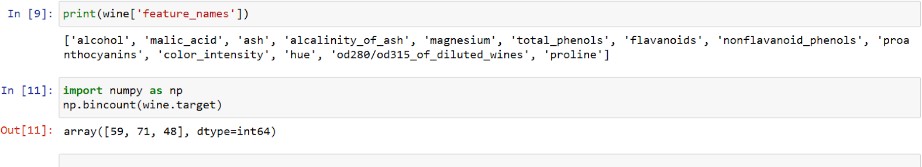


Scattered graph: with first two features( septal width vs septal length) The three colors represents three different classes respectively.



WINE DATASET:

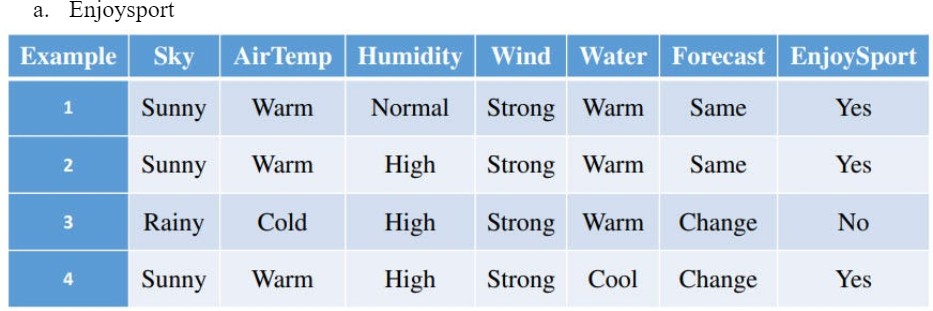




**Date:** 15/04/2023

**Lab 2:** FIND-S ALGORITHM FOR ENJOY SPORT:

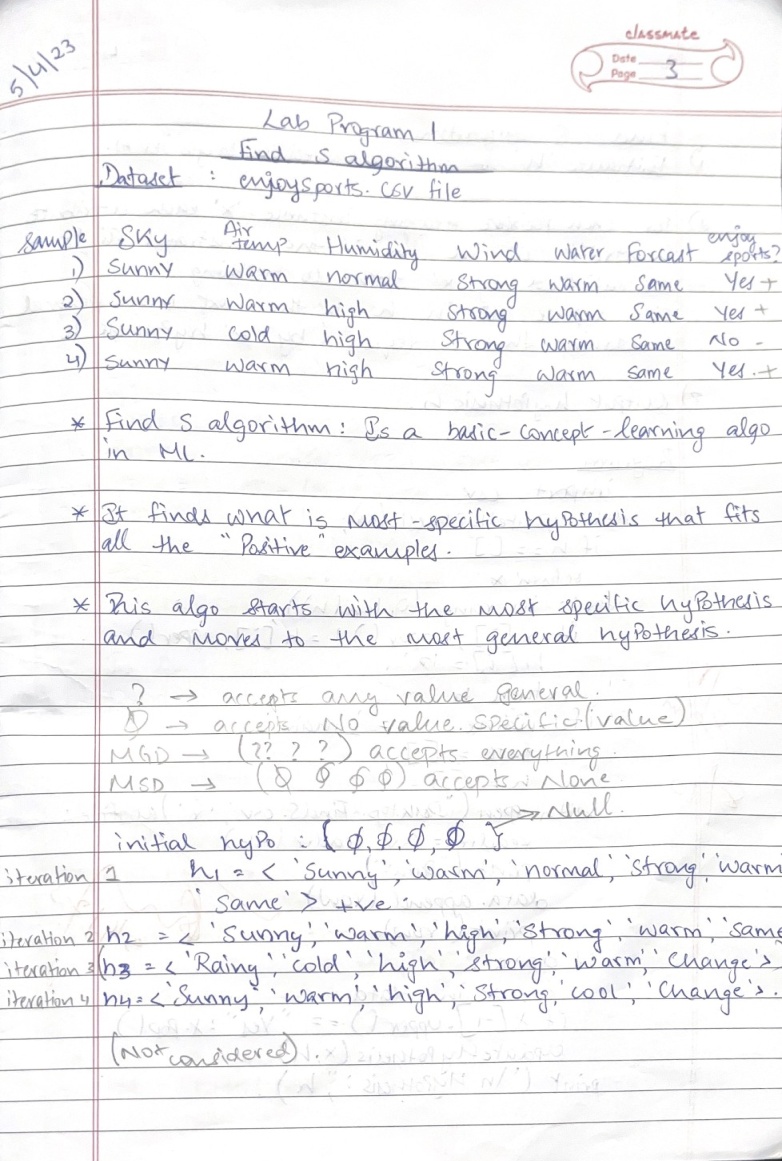
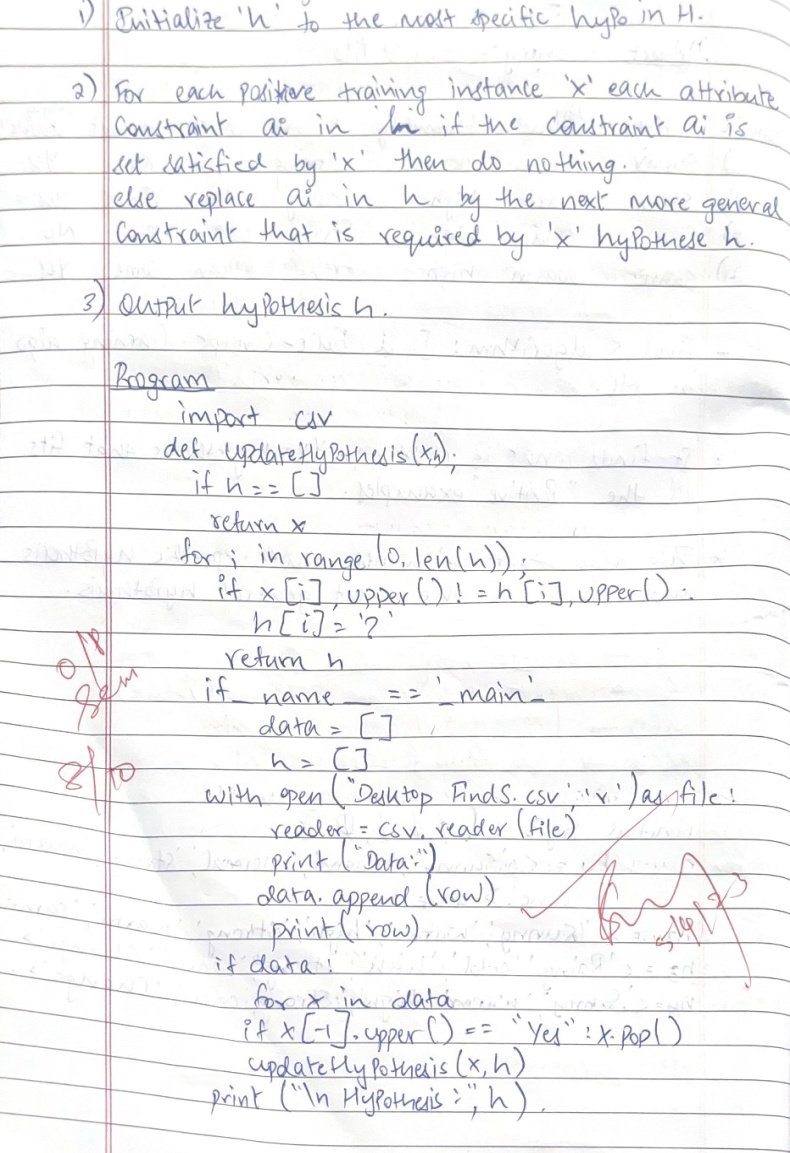
**Program 2** – Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file Data set:Enjoysport



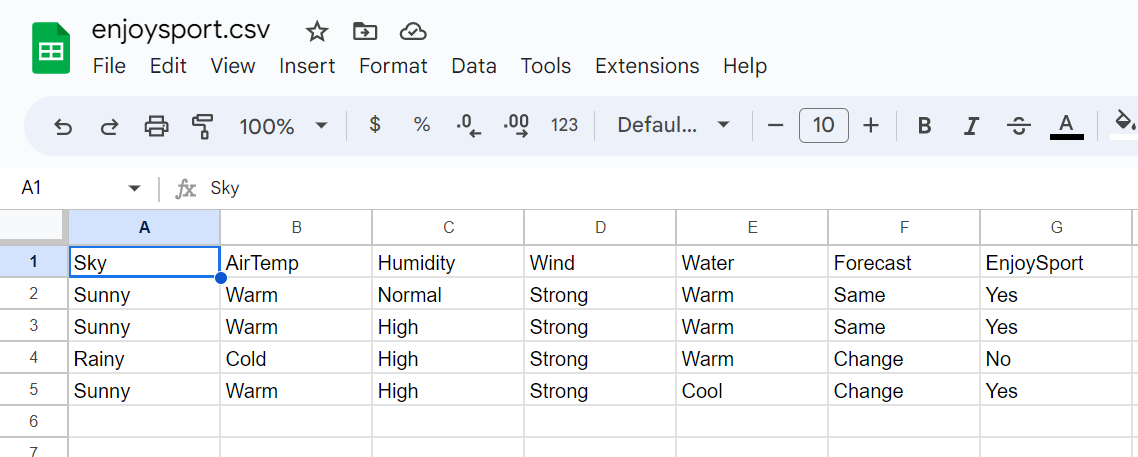
**Algorithm:**

initialize h to the most specific hypothesis in H h-(Ø, Ø, Ø, Ø, Ø, Ø)

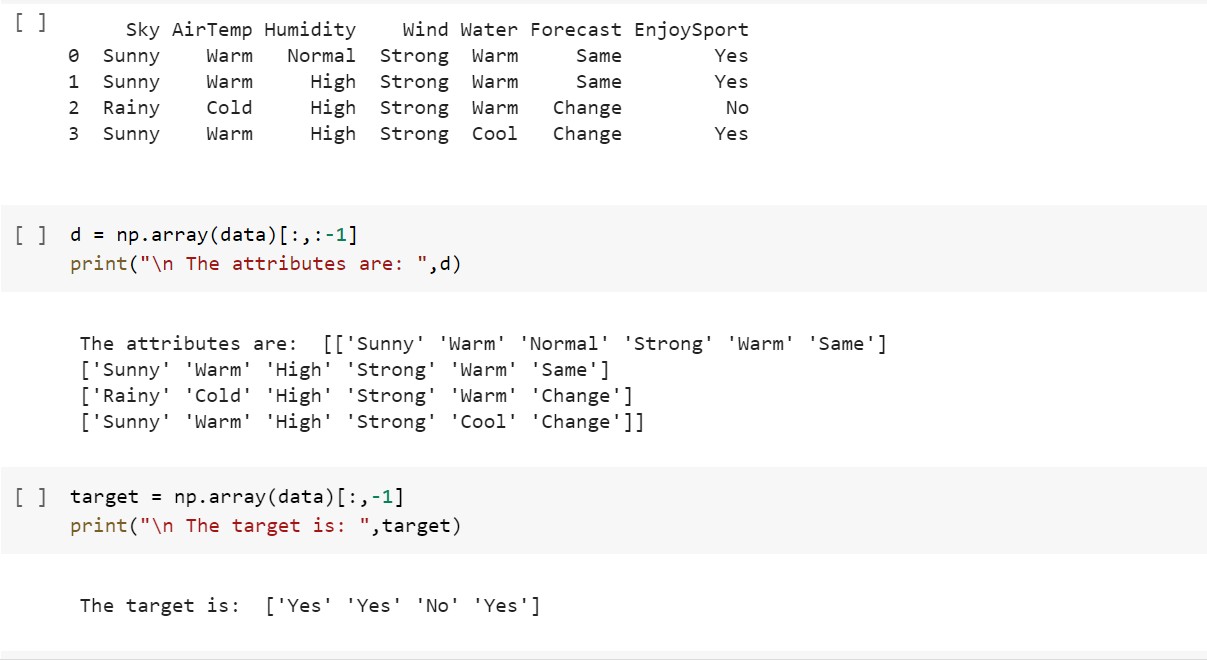
1. First training example X1=< Sunny, Warm. Normal, Strong Warm Same>. EnjoySport=+ve Observing.The first trainin example, it is clear that hypothesis h is too specific. None of the "**Ø**" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example h1 = < Sunny, Warm, Normal, Strong Warm, Same>.
2. Consider the second training example x2 < Sunny, Warm, High, Strong, Warm, Same>. EnjoySport+ve. The second training example forces the algorithm to further generalize h, this time substituting a "?" in place of any attribute value in h that is not satisfied by the new example. Now h2 =< Sunny, Warm, ?, Strong,Warm, Same>
3. Consider the third training example x3< Rainy, Cold, High, Strong, Warm. Change EnjoySport ve. The FIND-S algorithm simply ignores every negative example. So the hypothesis remain as before, so 13=< Sunny, Warm, ?, Strong,Warm, Same>
4. Consider the fourth training example x4 <Sunny, Warm, High. Strong. Cool, Change, EnjoySport +ve. The fourth example leads to a further generalization of h as h4=< Sunny, Warm, ?, Strong, ?, ?>
5. So the final hypothesis is < Sunny, Warm, ?, Strong, ?, ?>

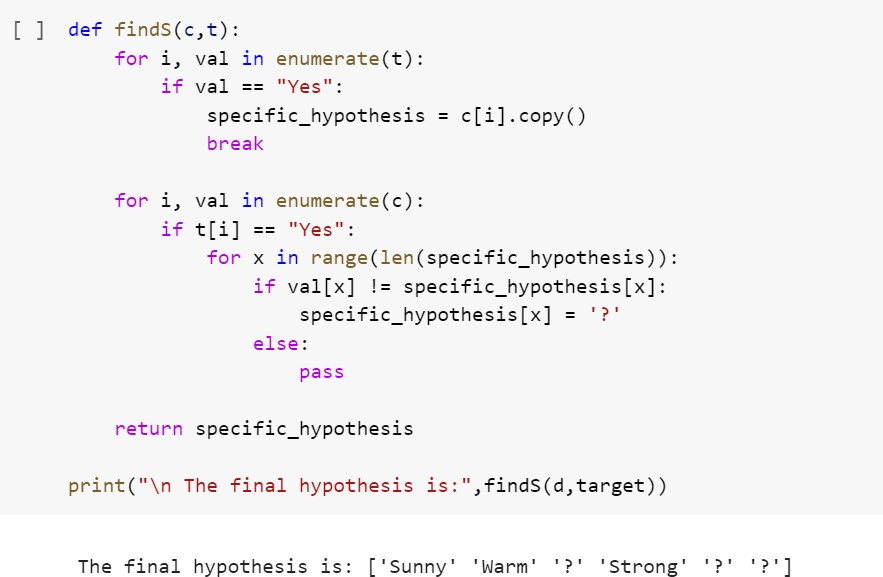


CREATING CSV FILE:

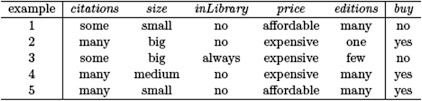




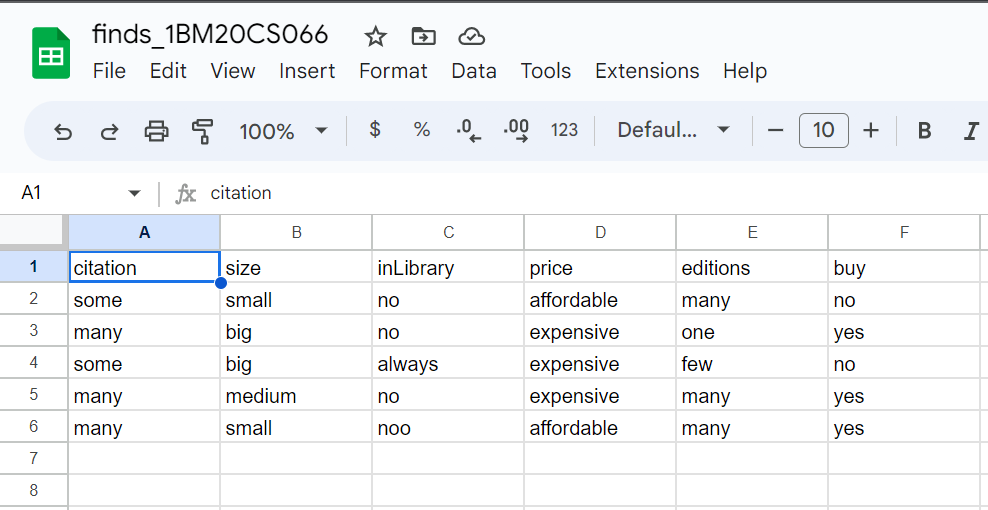


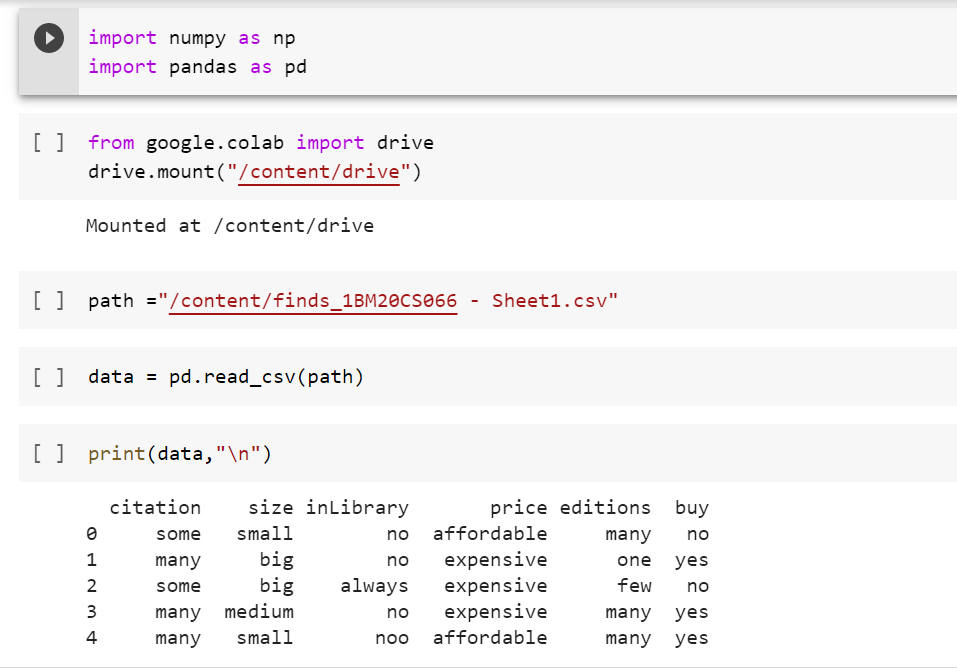


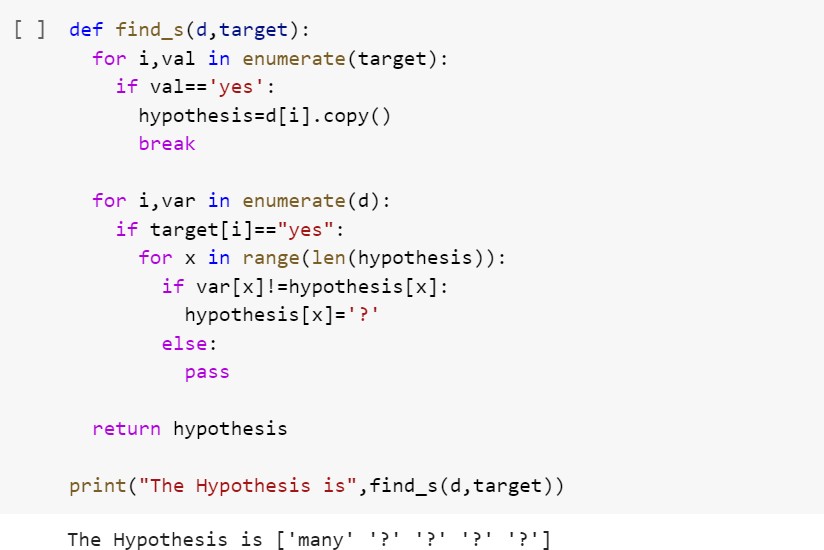
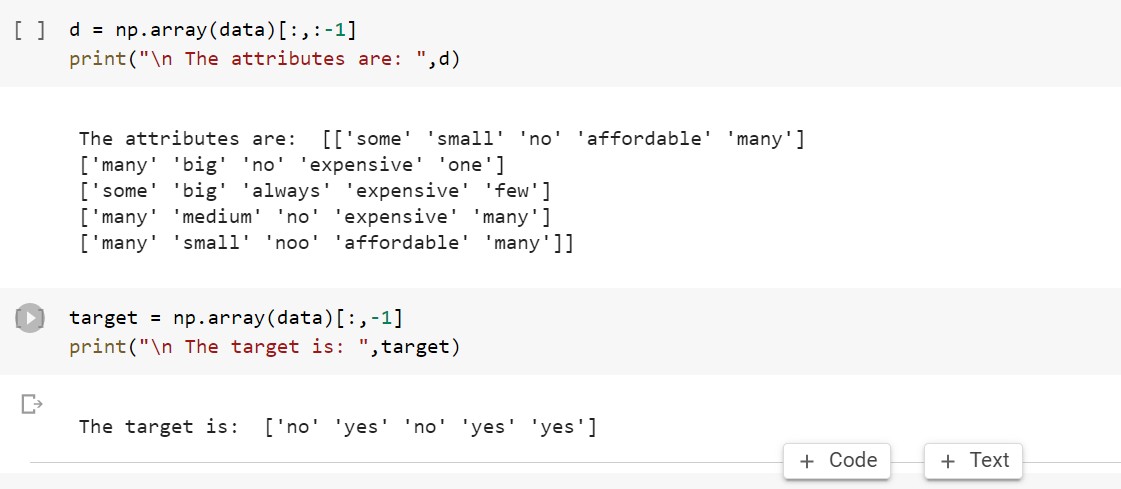
**SECOND DATASET: FIND-S ALGORITHM**



CREATING CSV FILE



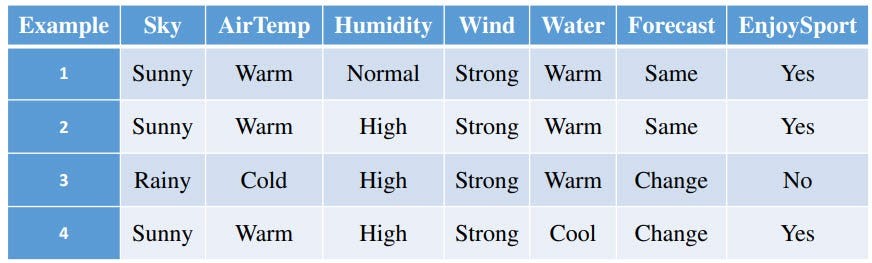




**DATE:** 15/04/2023

**LAB 3:** CANDIDATE- ELIMINATION- ENJOY SPORT

**Program 3:**For a given set of training data examples stored in a .CSV ﬁle, implement and demonstrate the

Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples. Data set:Enjoysport

**ALGORITHM:**

Step1: Load Data set

Step2: Initialize General Hypothesis and Specific Hypothesis. Step3: For each training example

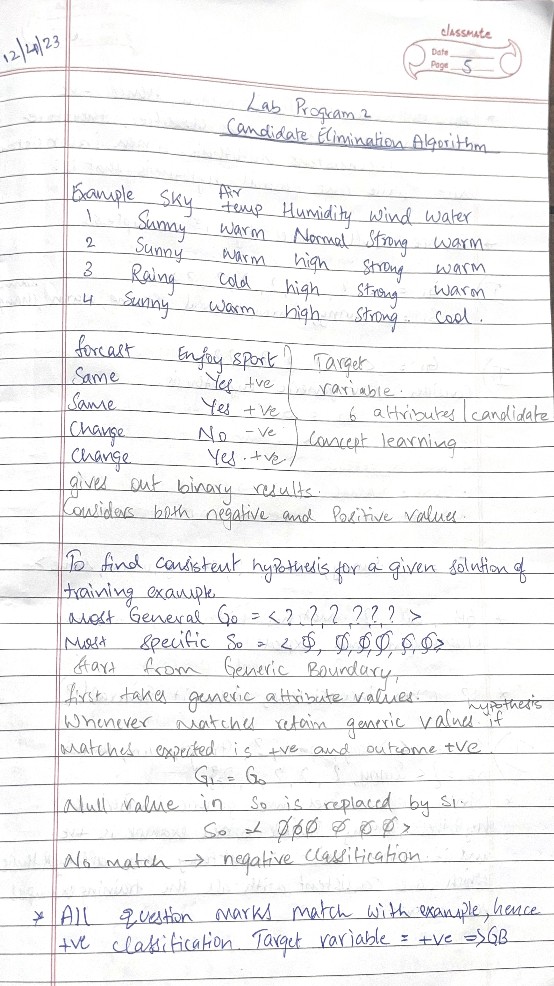
Step4: If example is positive example

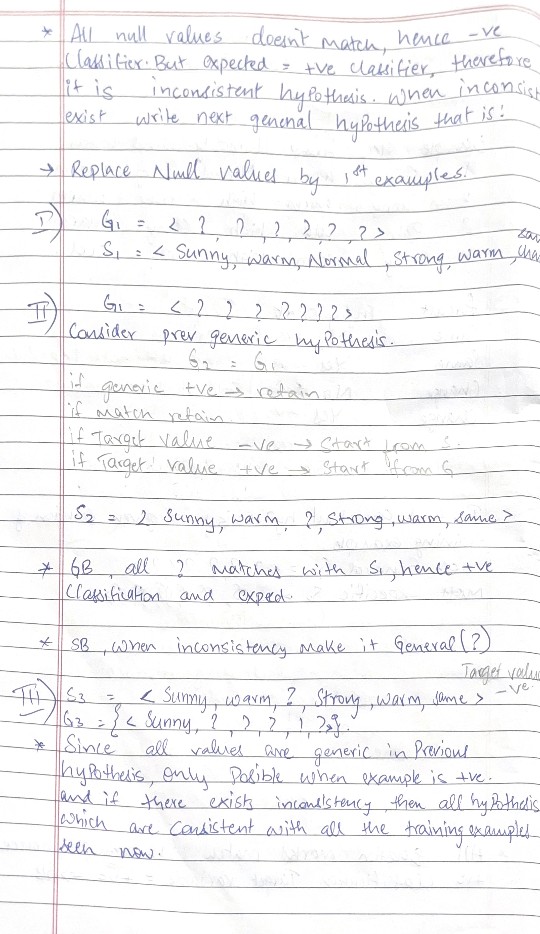
if attribute\_value == hypothesis\_value:

Do nothing else:

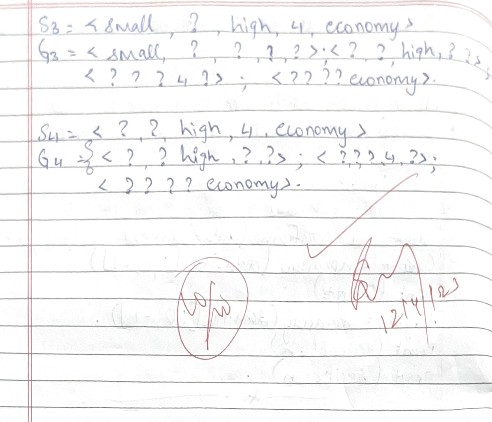
replace attribute value with '?' (Basically generalizing it) Step5: If example is Negative example

Make generalize hypothesis more specific.

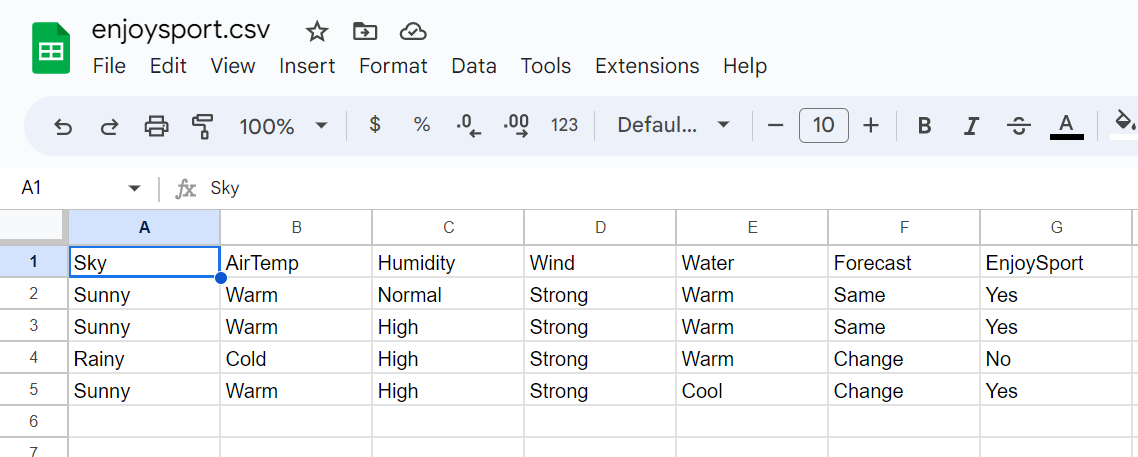


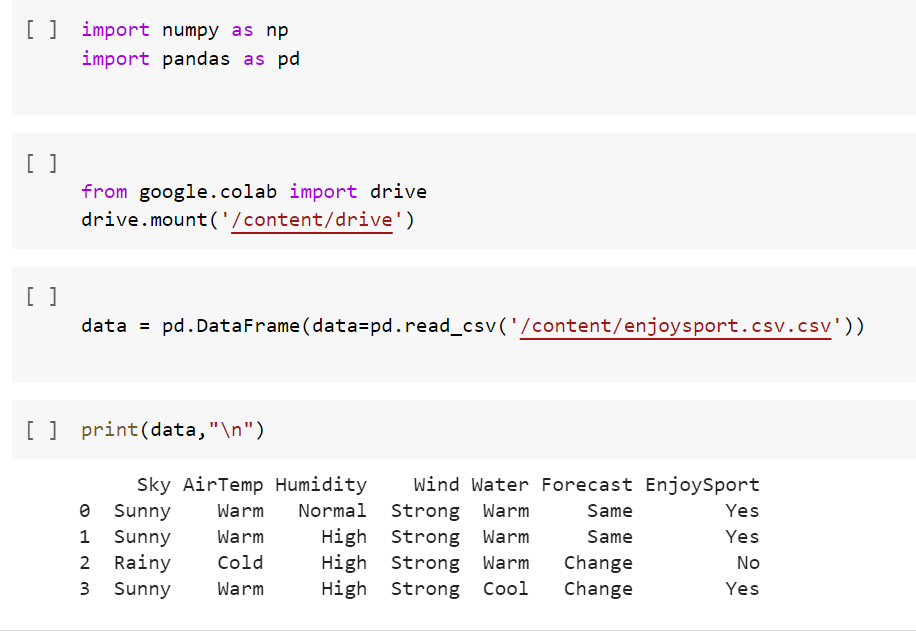


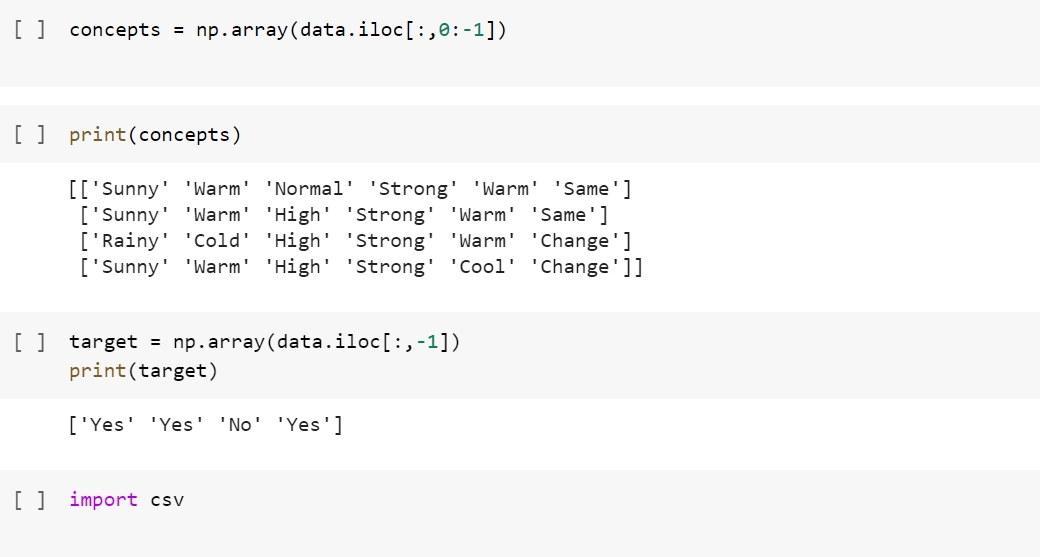




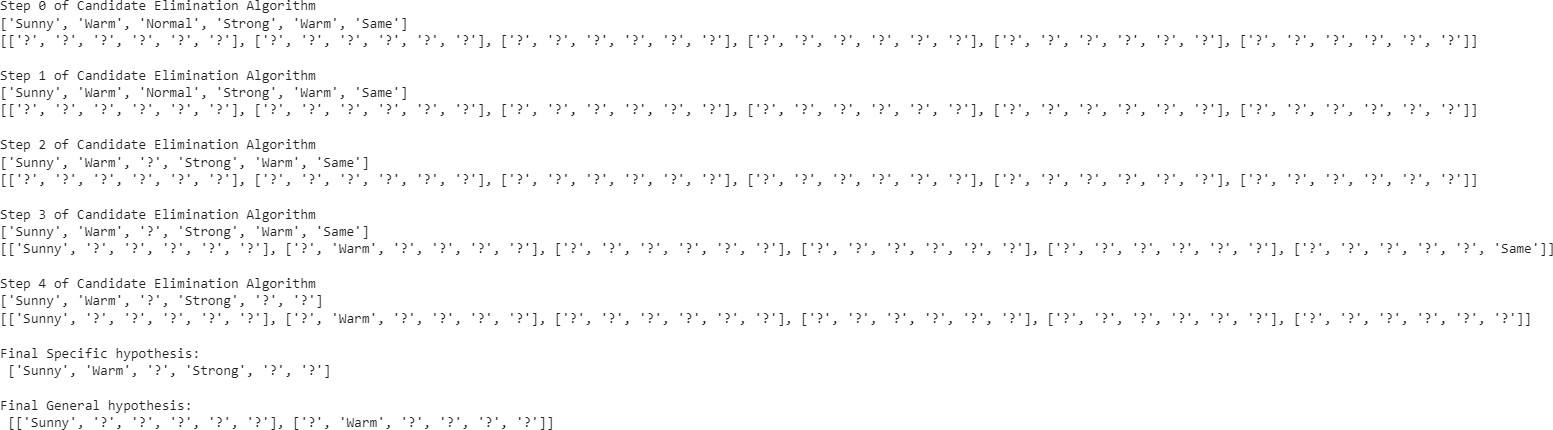
CREATING CSV FILE:

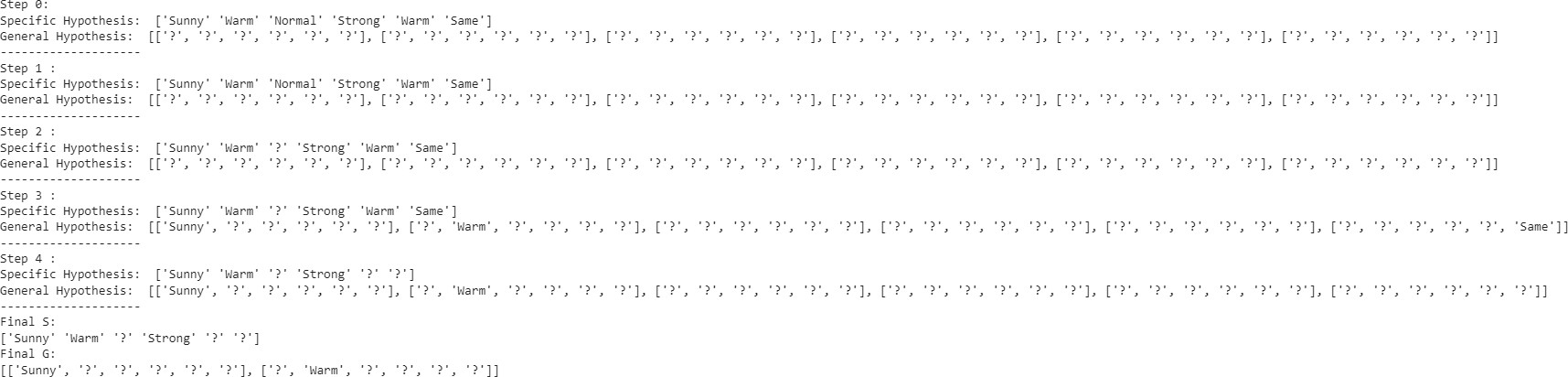




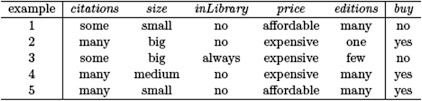




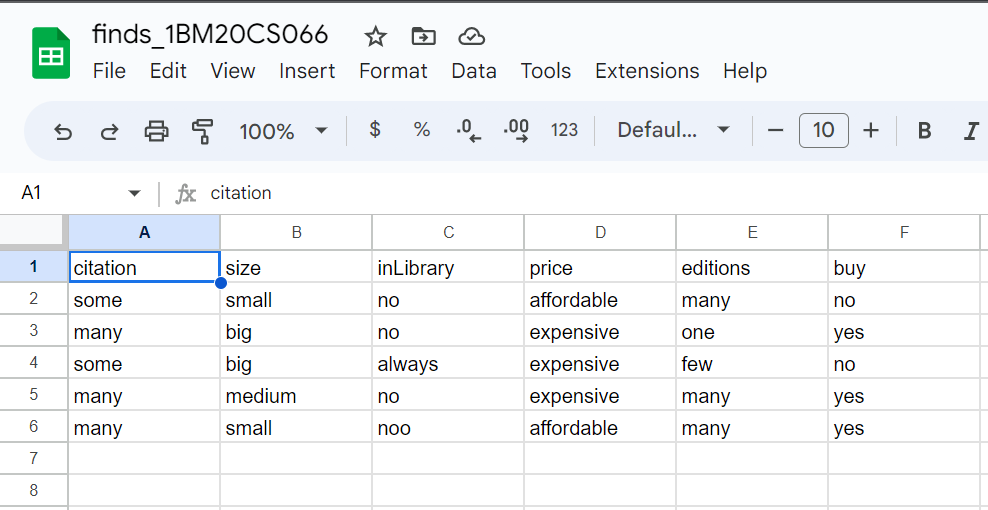


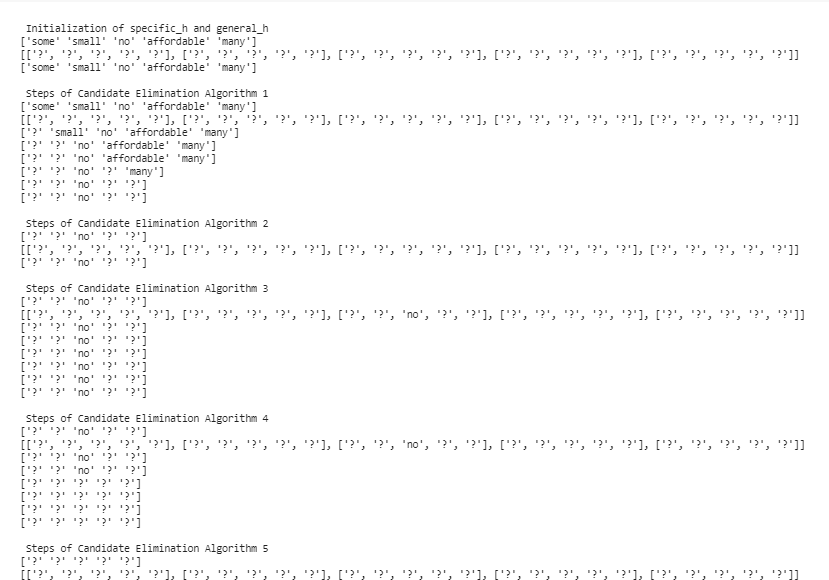


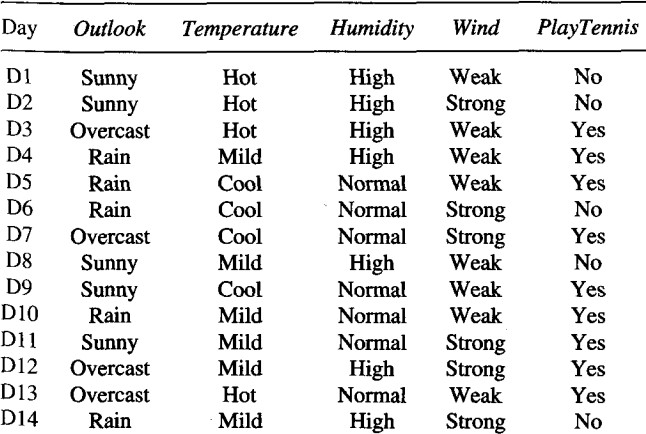
**SECOND DATASET:**



CREATING CSV FILE:





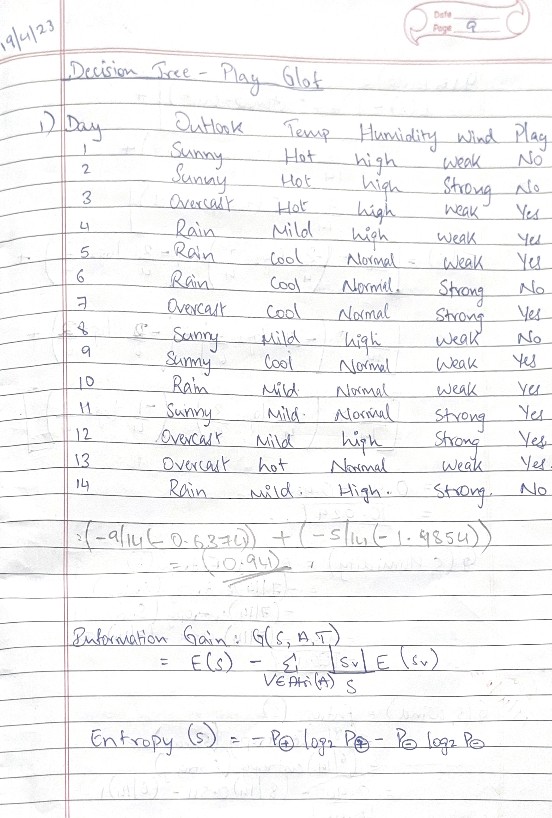
**Program 4:Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.**

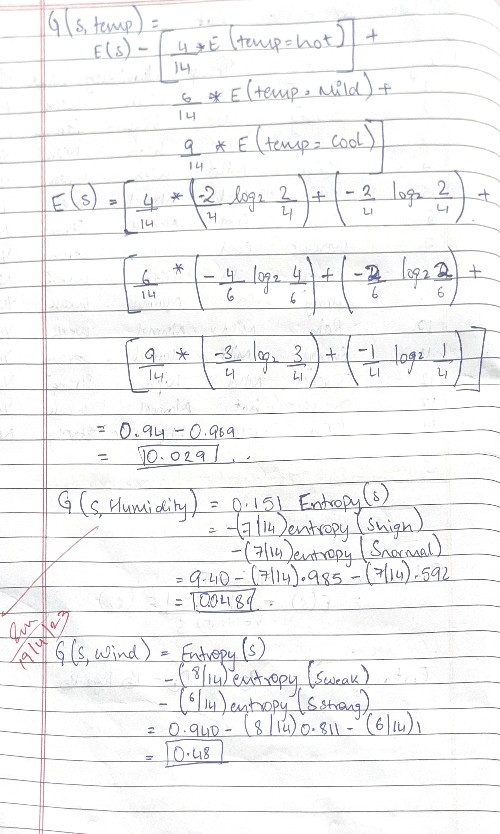
ALGORITHM:

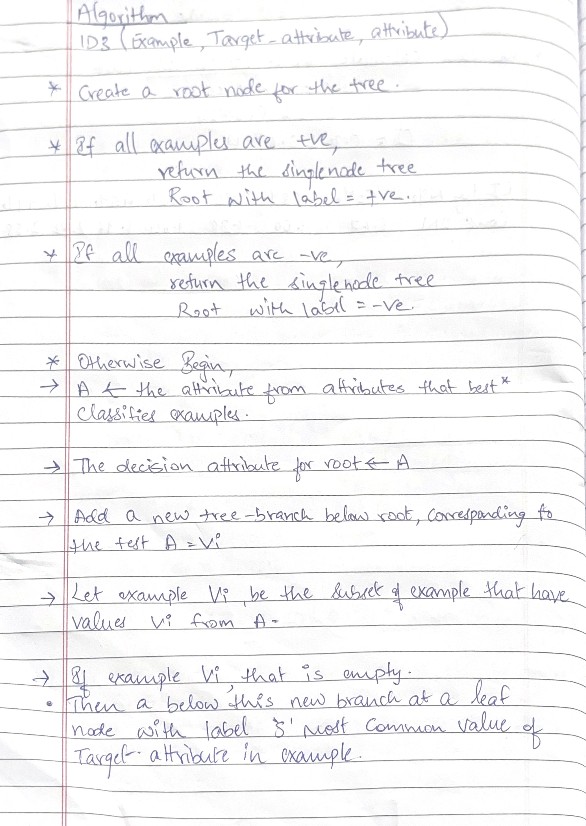
* Create a Root node for the tree
* If all Examples are positive, Return the single-node tree Root, with label = +
* If all Examples are negative, Return the single-node tree Root, with label = -
* If Attributes is empty, Return the single-node tree Root, with label = most common value of Target\_attribute in Examples

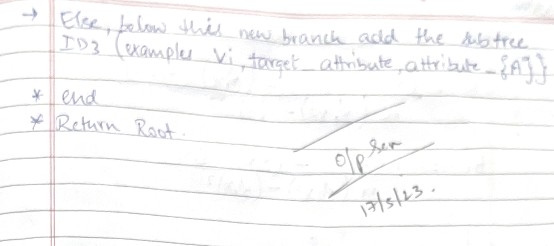
Otherwise Begin

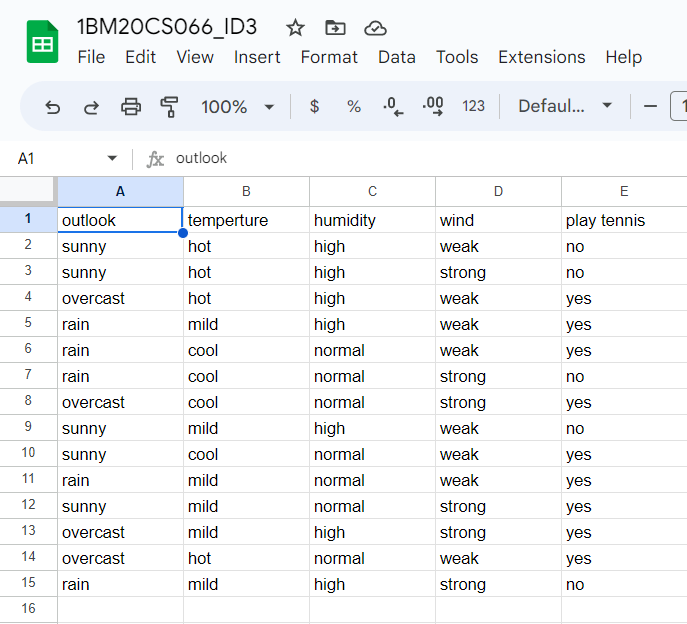
* A ← the attribute from Attributes that best\* classifies Examples
* The decision attribute for Root ← A
* For each possible value, *vi*, of A,
* Add a new tree branch below *Root*, corresponding to the test A = *vi*
* Let *Examples vi*, be the subset of Examples that have value *vi* for *A*
* If *Examples vi* , is empty
* Then below this new branch add a leaf node with label = most common value of Target\_attribute in Examples
* Else below this new branch add the subtree ID3(*Examples vi*, Targe\_tattribute, Attributes – {A}))
* End
* Return Root

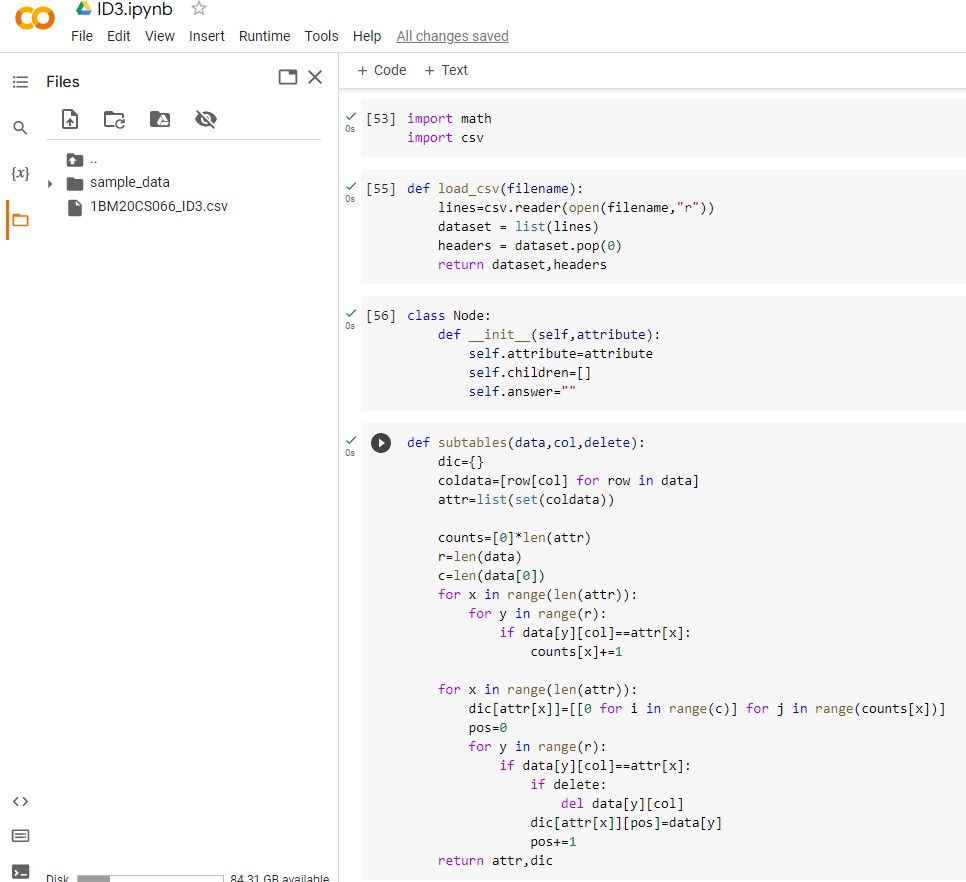


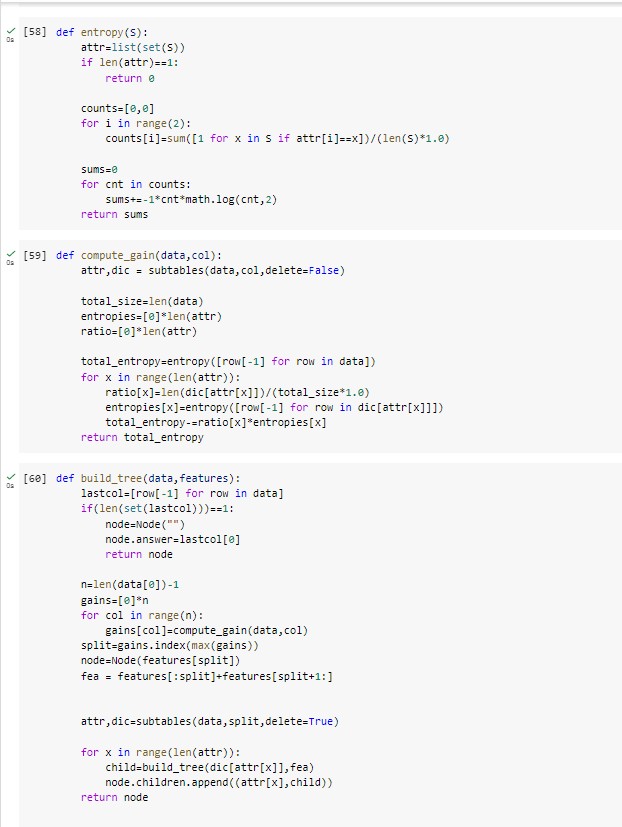
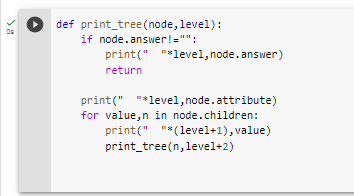


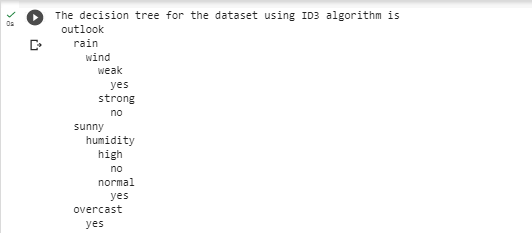


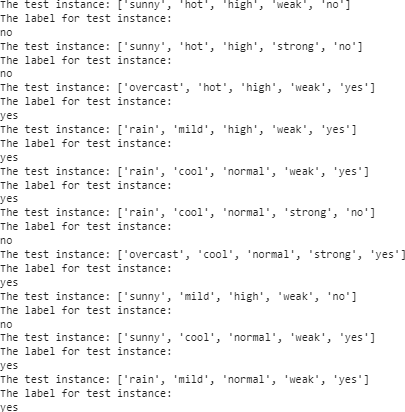


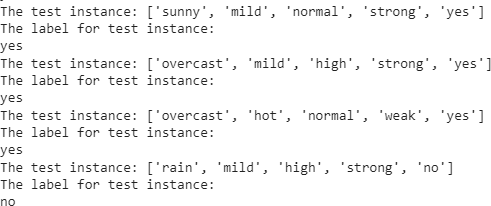










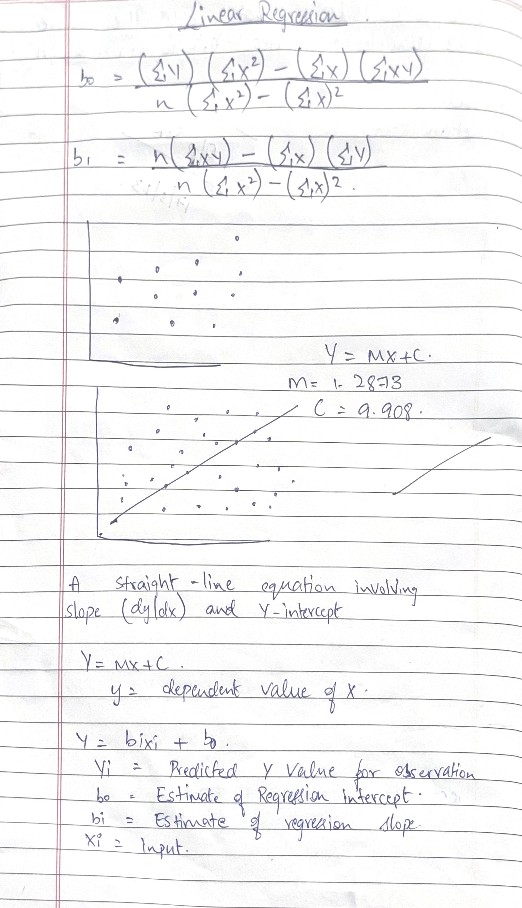


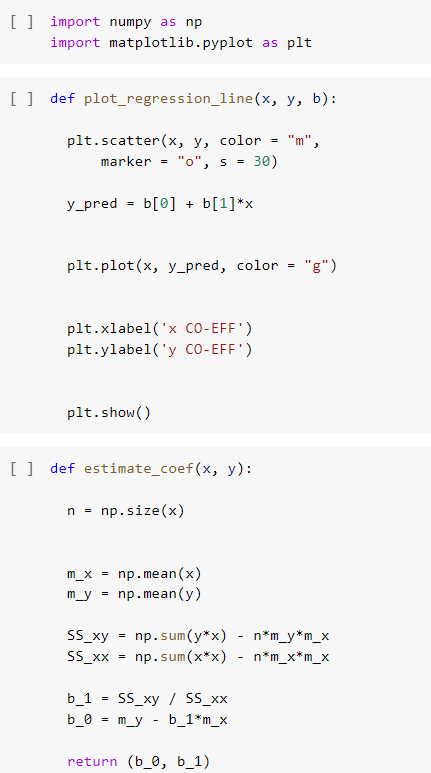
**PROGRAM 5: Simple linear regression program**

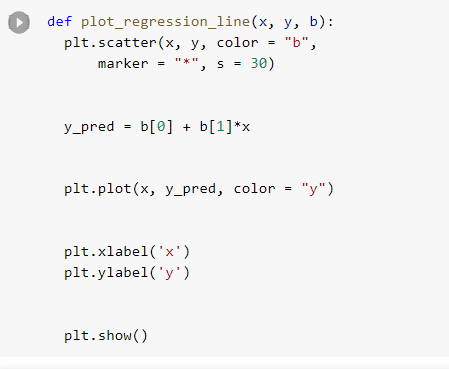
**Dataset used:**

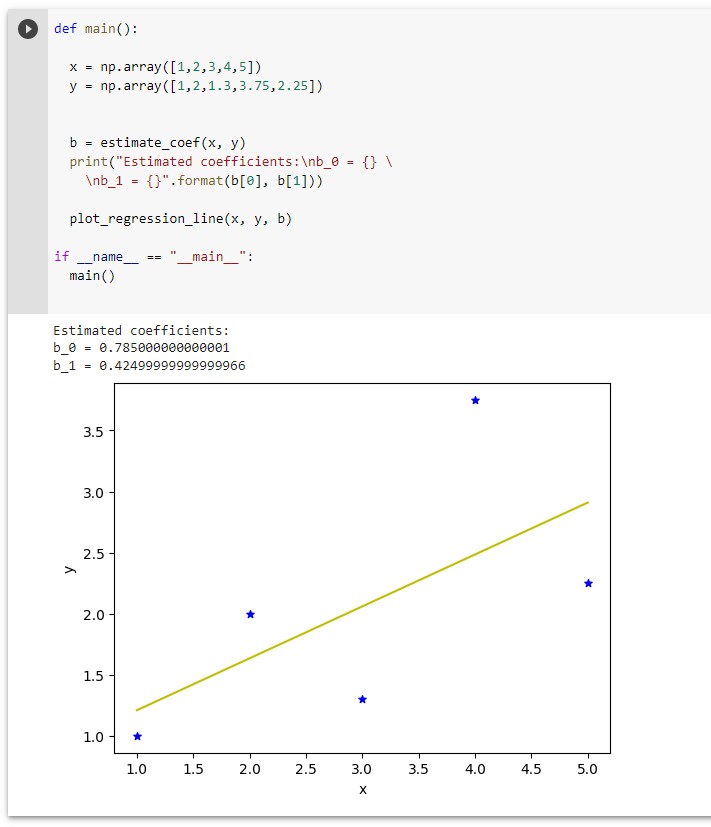
**ALGORITHM:**

* + The main function to calculate values of coefficients
  + Initialize the parameters.
  + Predict the value of a dependent variable by giving an independent variable.
  + Calculate the error in prediction for all data points.
  + Calculate partial derivatives w.r.t a0 and a1.
  + Calculate the cost for each number and add them.
  + Update the values of a0 and a1.





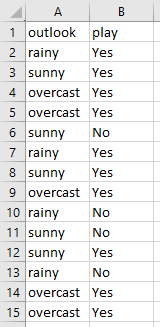




**Conclusion:**

**This model is not appropriate for this model. All the points of this dataset are away from the prediction line.**

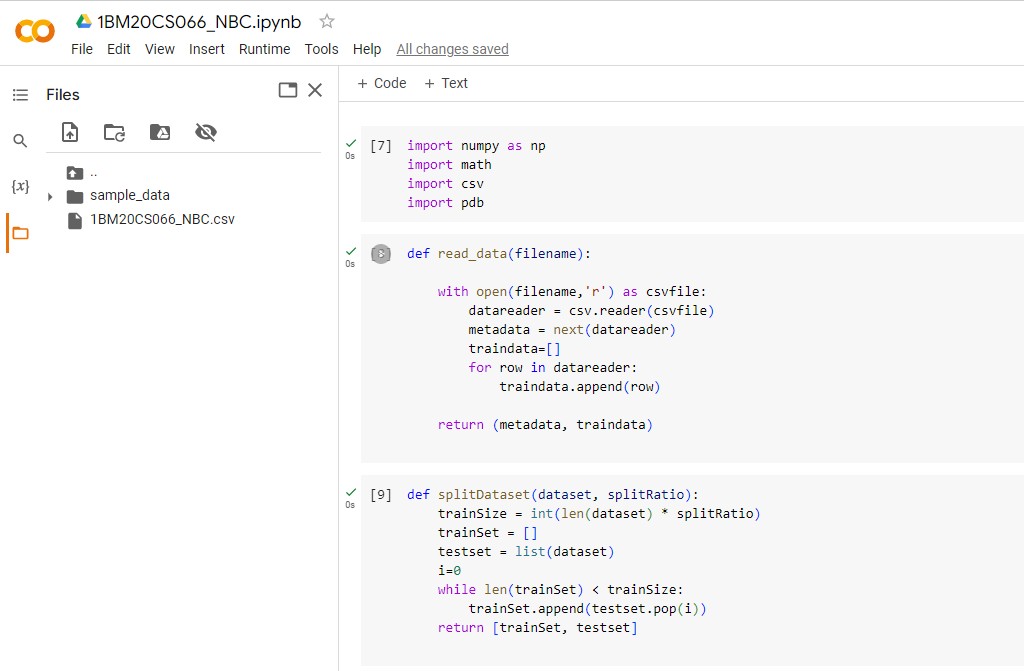
# Program 6:Write a program to implement the naïve Bayesian classiﬁer for a sample training data set stored as a .CSV ﬁle. Compute the accuracy of the classiﬁer, considering few tes data sets.

**Data set used:**

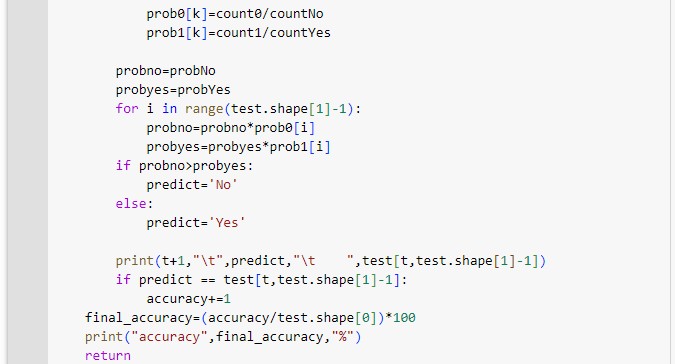
**Algorithm:**

Formula for naive bayes classifier is as follows → 

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.
4. Test accuracy of the result and visualizing the test set result.

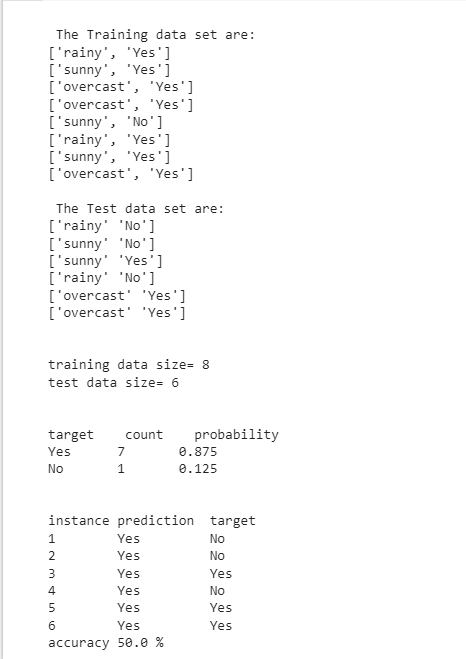


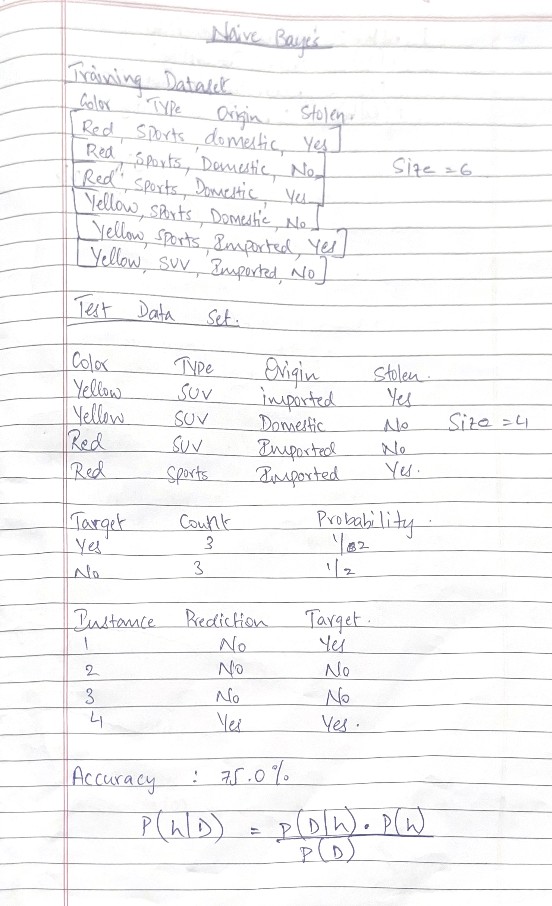


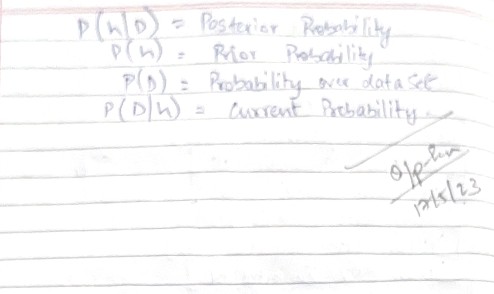




**output:**







# Program 7:K- means clustering

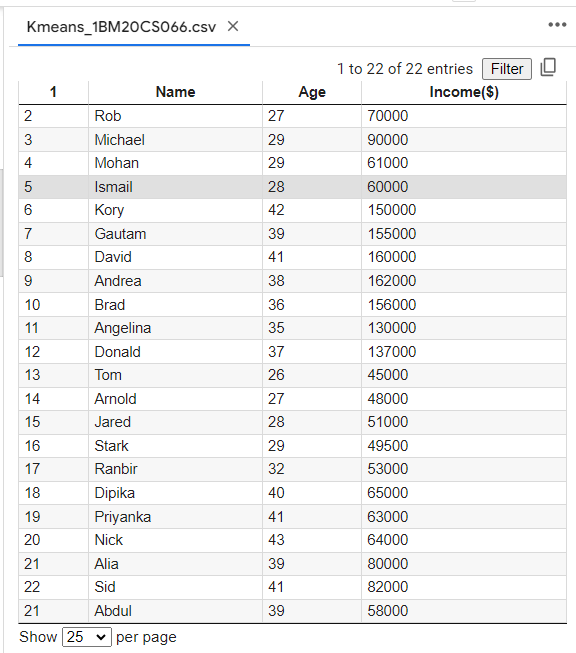
**Algorithm:**

Initialize k means with random values For a given number of iterations:

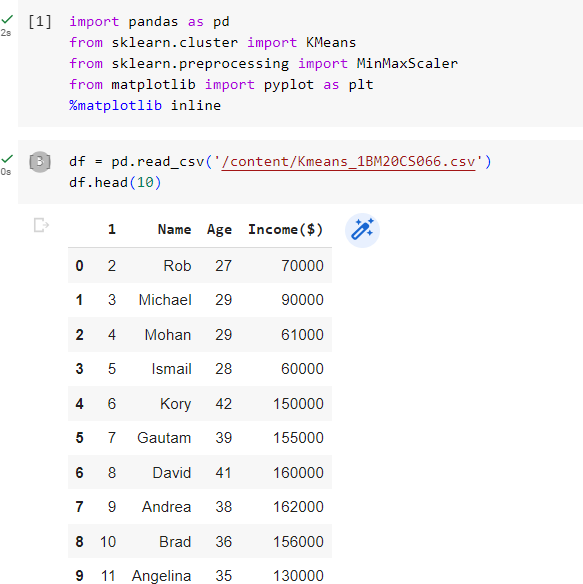
Iterate through items:

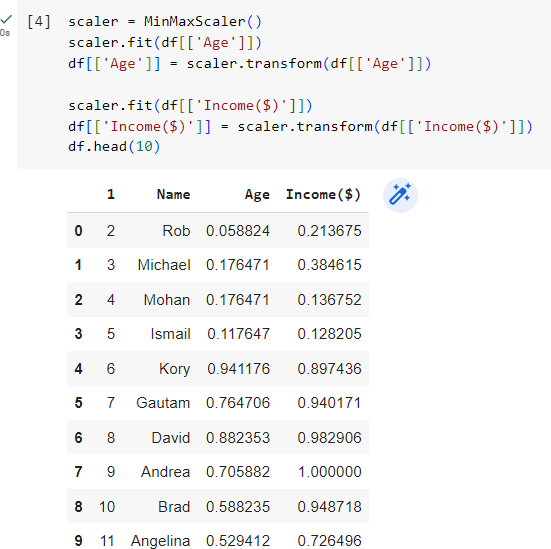
Find the mean closest to the item by calculating the euclidean distance of the item with each of the means Assign item to mean

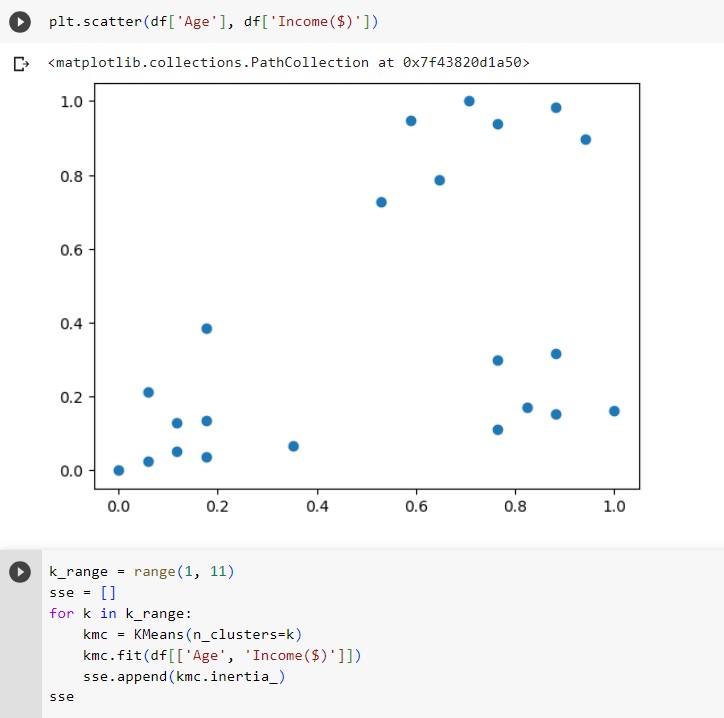
Update mean by shifting it to the average of the items in that cluster

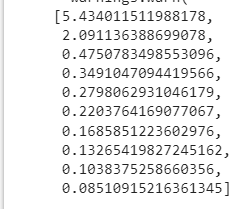
**Dataset:**

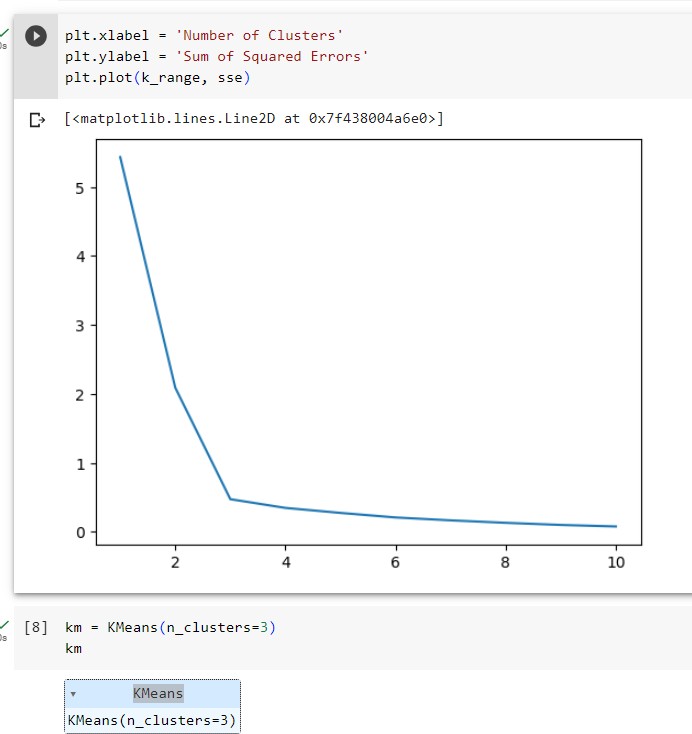


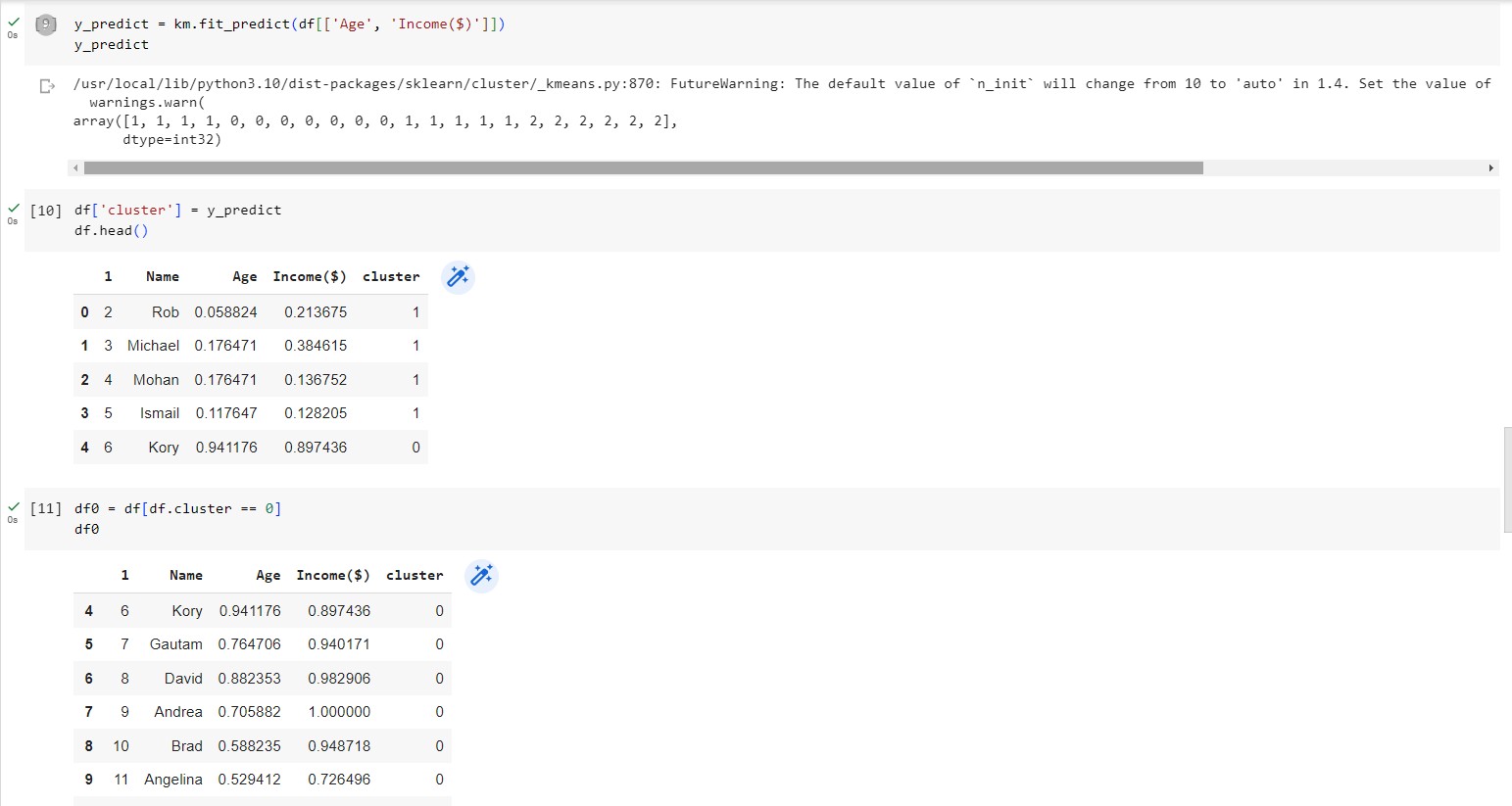


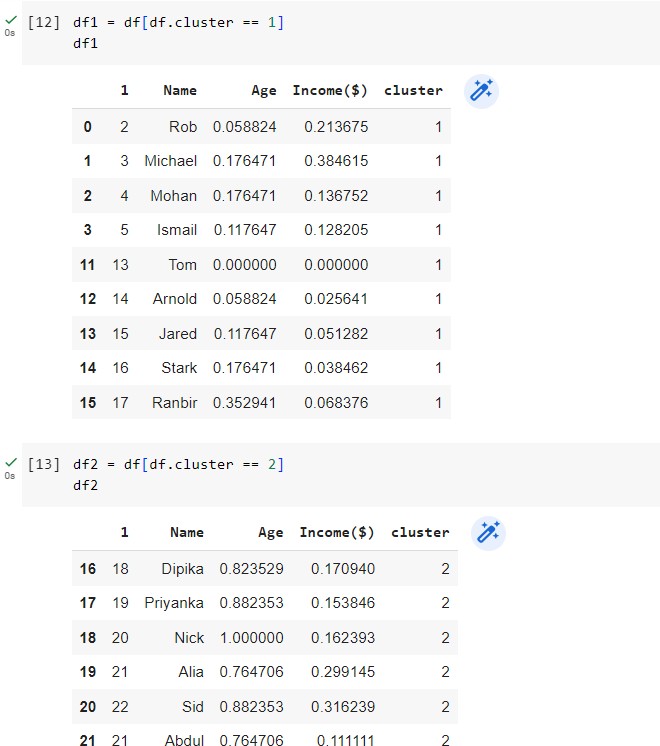


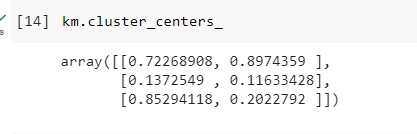


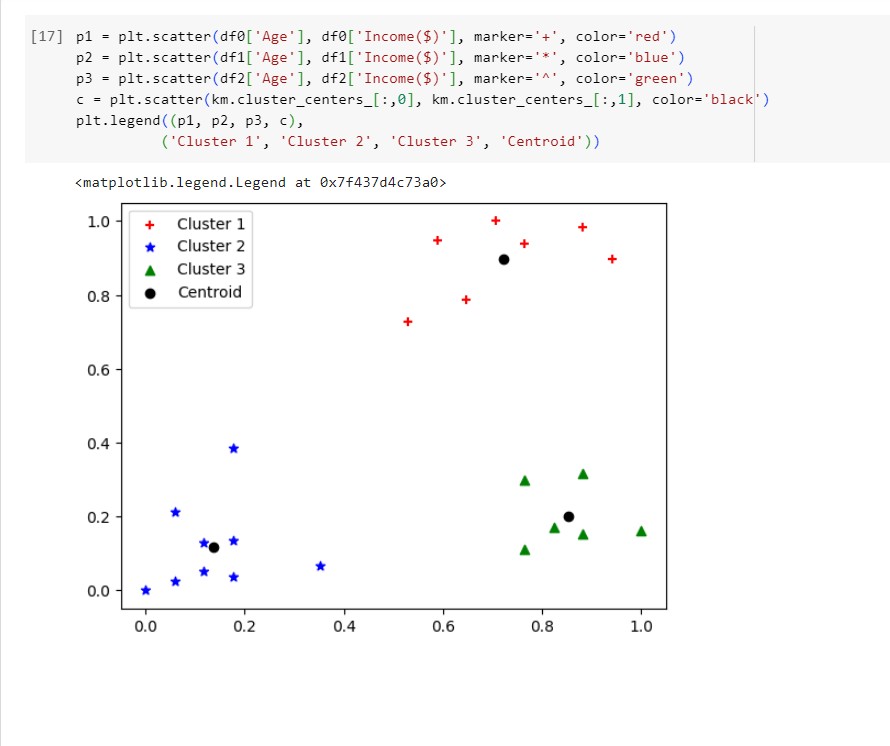








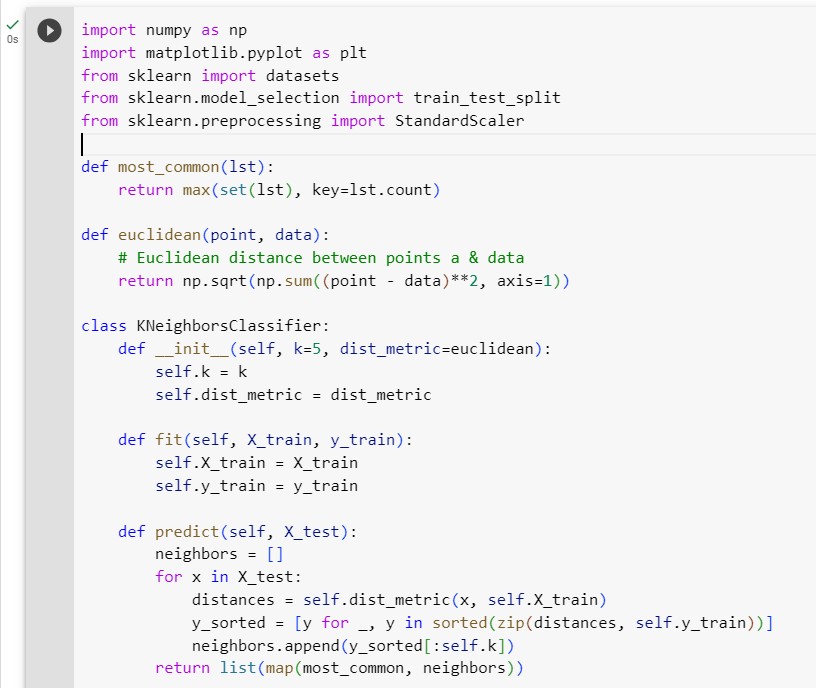


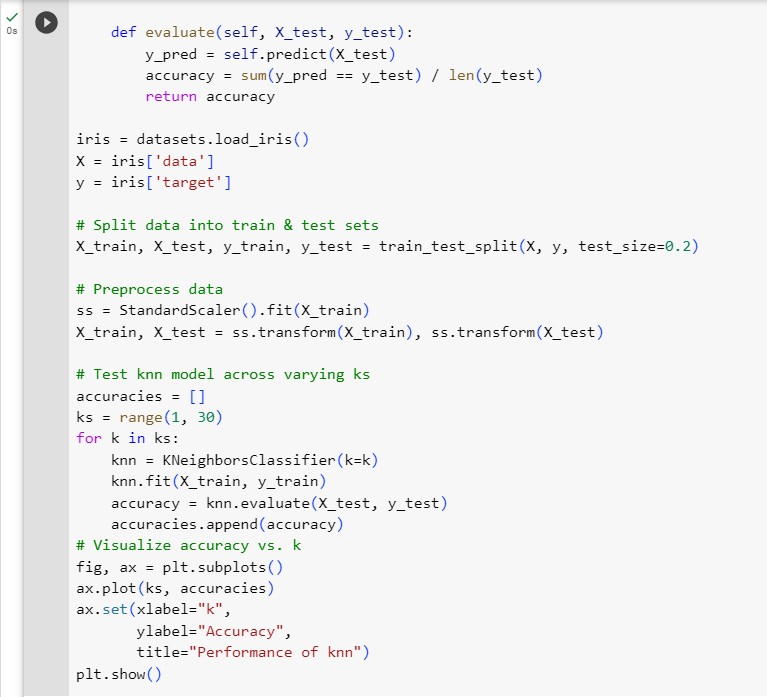


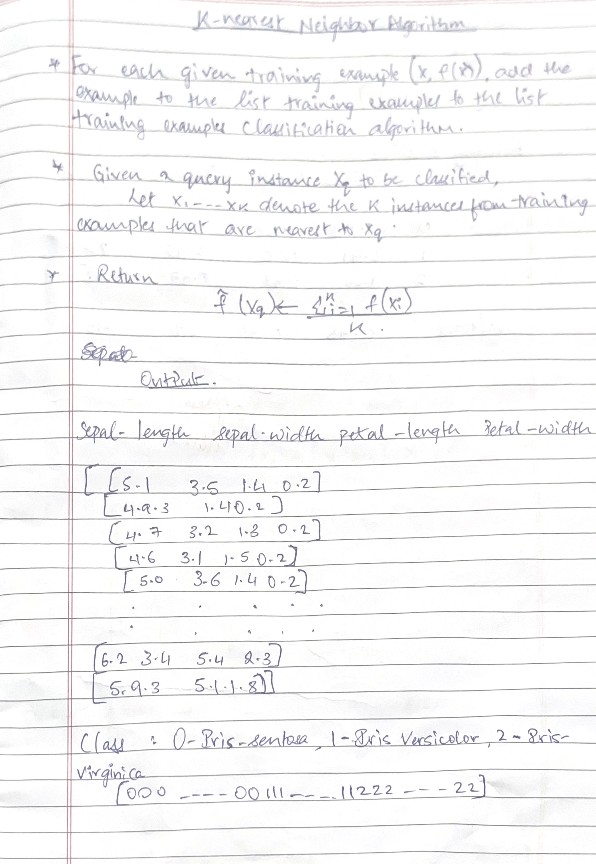
# Program 8: KNN ALGORITHM

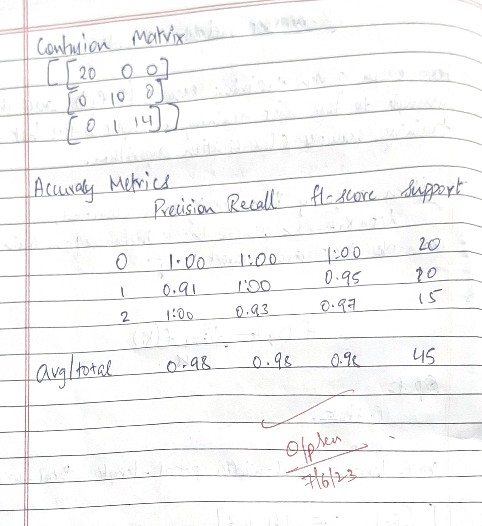
**Dataset used: Iris dataset Algorithm:**

* Select the number K of the neighbor
* Calculate the Euclidean distance of K number of neighbors
* Take the K nearest neighbors as per the calculated Euclidean distance.
* Among these k neighbors, count the number of the data points in each category.
* Assign the new data points to that category for which the number of the neighbor is maximum.







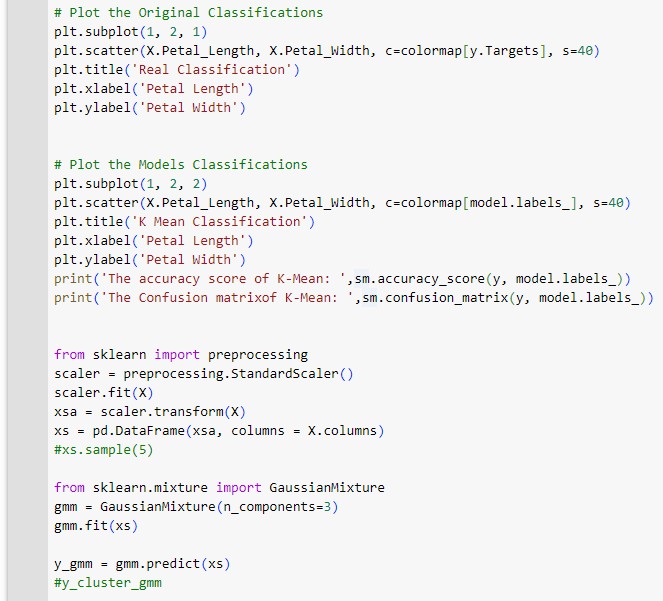


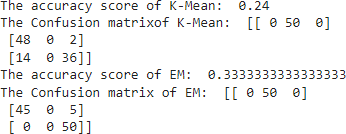
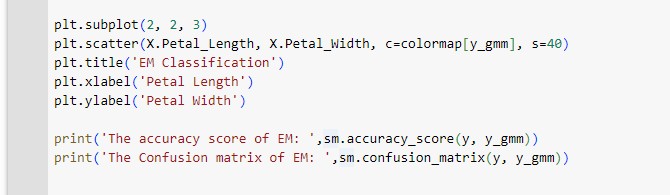
**Program 9:** Apply EM algorithm to cluster a set of data stored in a .CSV file. Compare the results of k-Means algorithm and EM algorithm.

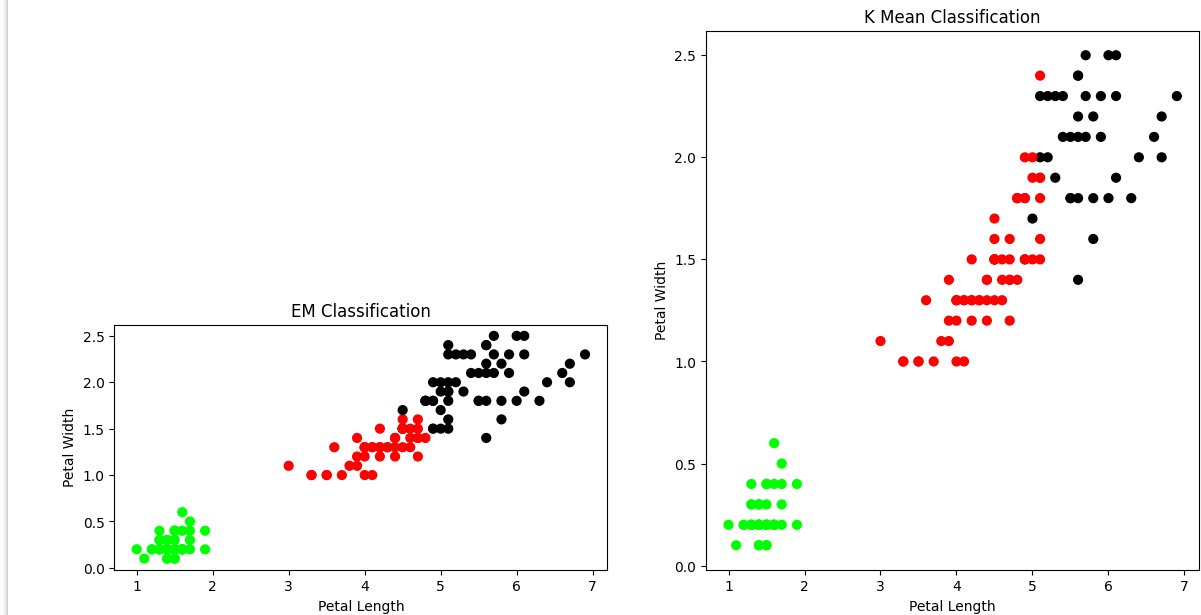
Algorithm for k means clustering:

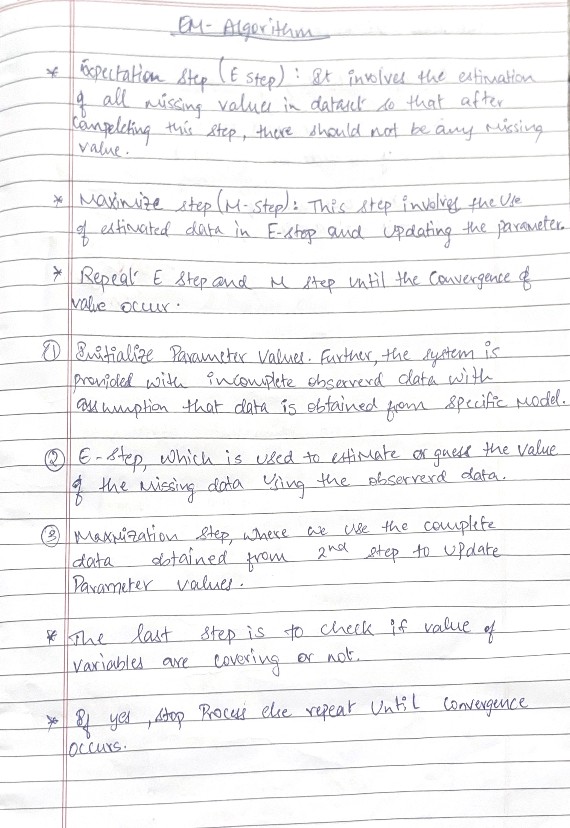
* + Initialize k means with random values
  + For a given number of iterations:
  + Iterate through items:
  + Find the mean closest to the item by calculating the euclidean distance of the item with each of the means
  + Assign item to mean
  + Update mean by shifting it to the average of the items in that clusters Algorithm for EM algorithm:
    - The very first step is to initialize the parameter values. Further, the system is provided with incomplete observed data with the assumption that data is obtained from a specific model.
* This step is known as Expectation or E-Step, which is used to estimate or guess the values of the missing or incomplete data using the observed data. Further, E-step primarily updates the variables.
* This step is known as Maximization or M-step, where we use complete data obtained from the 2nd step to update the parameter values. Further, M-step primarily updates the hypothesis.
* The last step is to check if the values of latent variables are converging or not. Dataset: Iris dataset







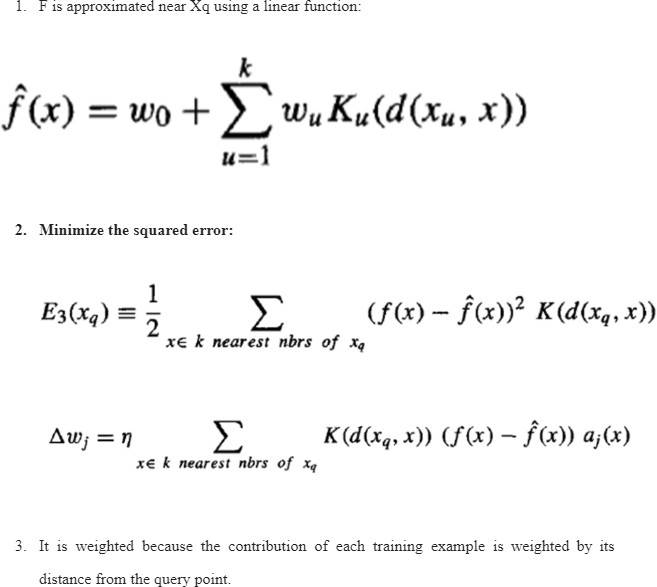






**Program 10:**Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select the appropriate data set for your experiment and draw graphs.

Algorithm:



Dataset: tip.csv

