Name: Kuldeep Shushil Patel **Roll no: 574**

MALAD KANDIVALI EDUCATION SOCIETY'S
NAGINDAS KHANDWALA COLLEGE OF COMMERCE,
ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS
KHANDWALA COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
(AUTONOMOUS)

(Reaccredited 'A' Grade by NAAC)
(AFFILIATED TO UNIVERSITY OF MUMBAI)
(ISO 9001:2015)

CERTIFICATE

Name: Mr. kuldeep sushil patel

Roll No: 574 **Programme**: BSc CS **Semester**: V

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **Artificial Intelligence** (Course Code: **1851UCSPR**) for the partial fulfillment of Fifth Semester of BSc CS during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

External Examiner **Prof. Elizabeth Leah George**
(Subject-In-Charge)

Date of Examination: **(College Stamp)**

Name: Kuldeep Shushil Patel **Roll No: 574**

Subject: Artificial Intelligence (1851UCSPR)

Class: T.Y.B.SC (CS) **SEMESTER-V**

NO	DATE	TITLE	SIGN
1.	17/07/2020	**Blind Search Algorithms:** 1. Implement Breadth First Search (BFS) algorithm 2. Implement Depth First Search (DFS) algorithm 3. Implement Iterative deepening Search (IDS) algorithm	
2.	31/07/2020	**Heuristic Search Algorithm**: Implement A* search algorithm	
3.	14/08/2020	Implementation of Neural Network on IRIS dataset	
4.	21/08/2020	Implementation of basic neural network model with 4 activation functions on Pima Indians onset of diabetes dataset	
5.	04/09/2020	Performing AND & OR Operations in the Neural Network	
6.	18/09/2020	Prediction Algorithm - Use of different packages on dataset of Cat and Non-Cat images	
7.	25/09/2020	Simple Linear Regression	
8.	09/10/2020	Implement Support Vector Algorithm	
9.	16/10/2020	Decision Tree Learning	

Name: Kuldeep Shushil Patel **Roll no: 574**

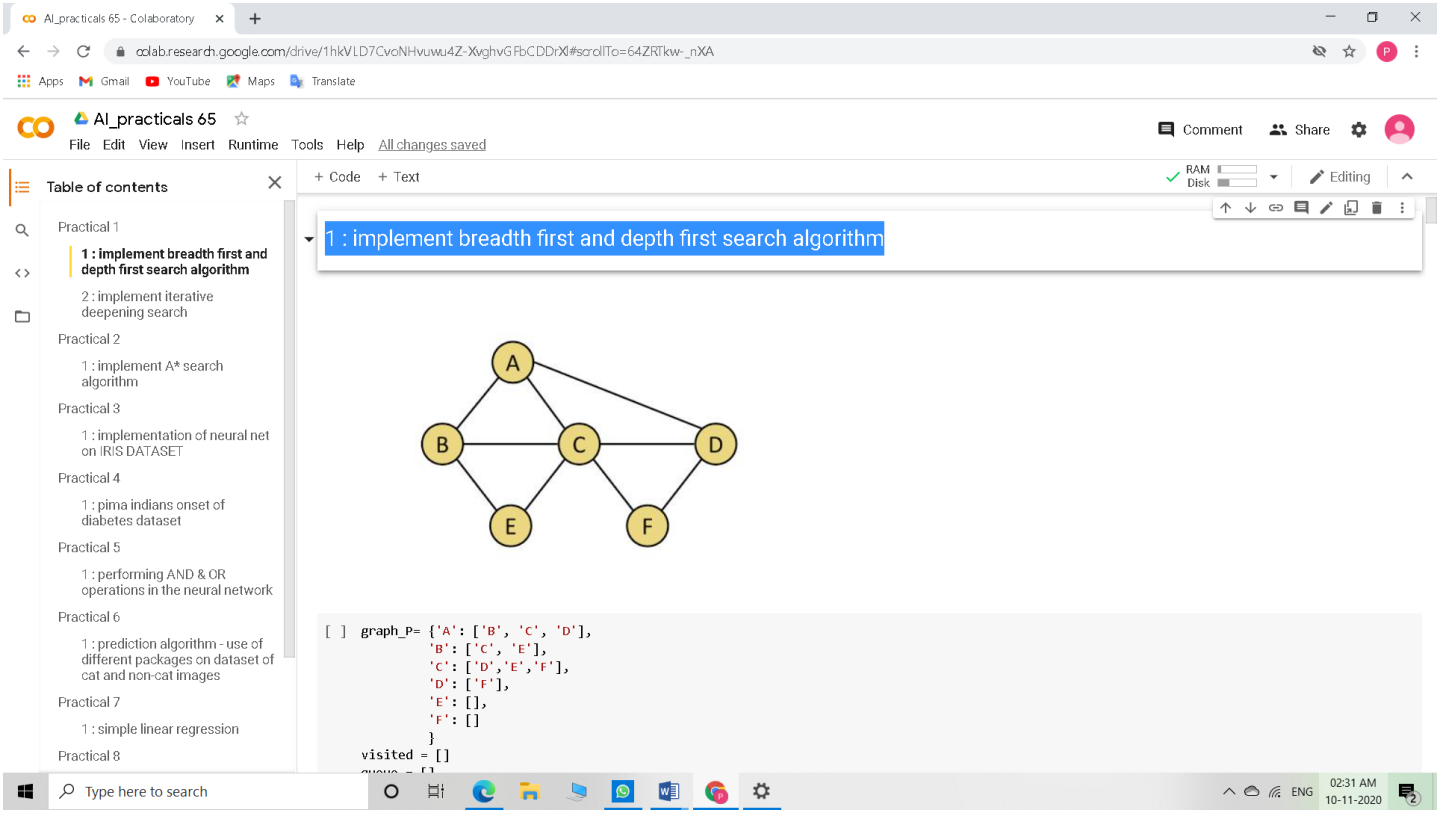
PRACTICAL 1

Aim :1]implement breadth first and depth first search algorithm

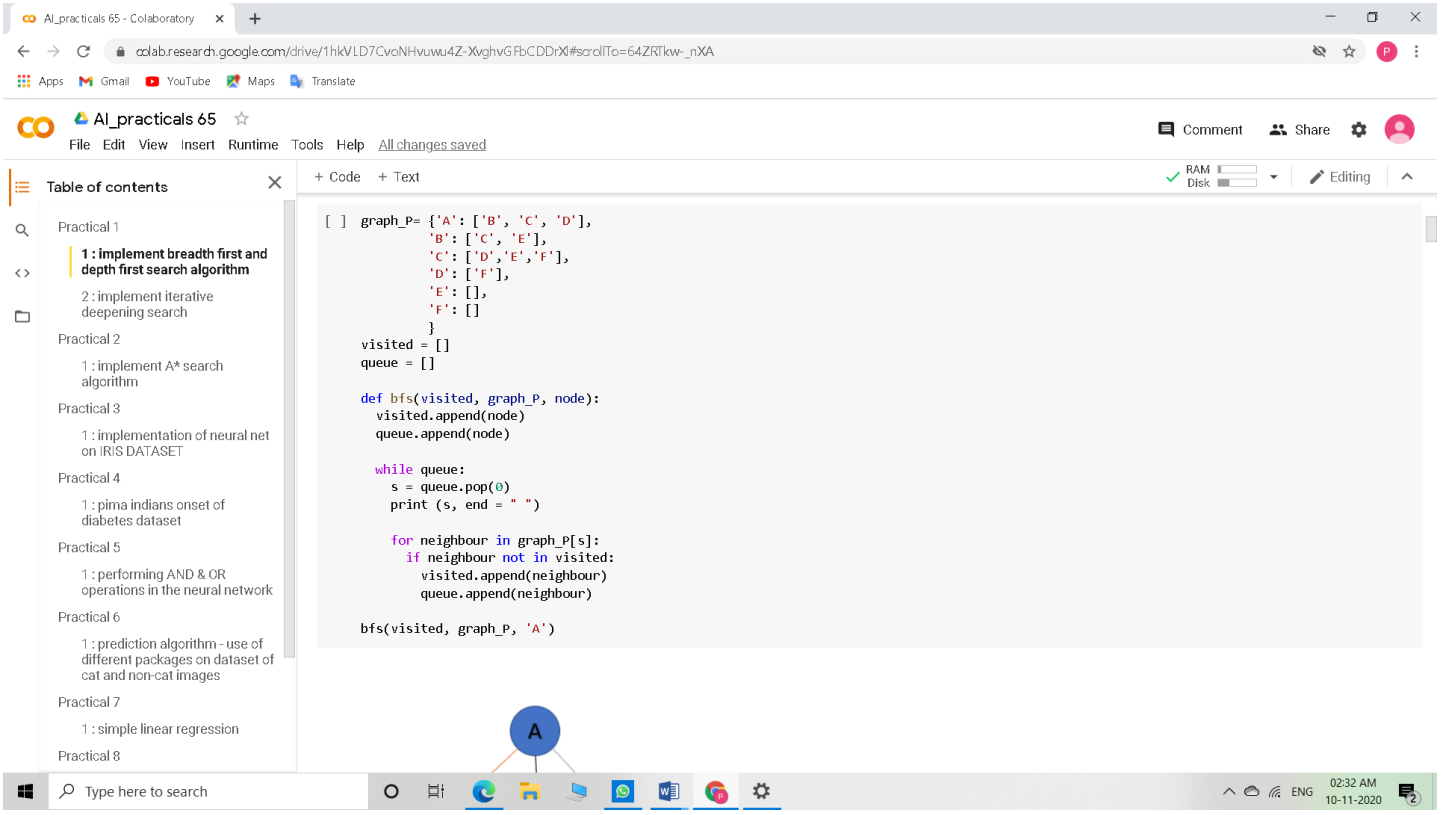
Theory :

- Breadth First Traversal (or Search) for a graph is similar to Breadth First Traversal of a tree.
- The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again.
- To avoid processing a node more than once, we use a boolean visited array.
- For simplicity, it is assumed that all vertices are reachable from the starting vertex

Code:

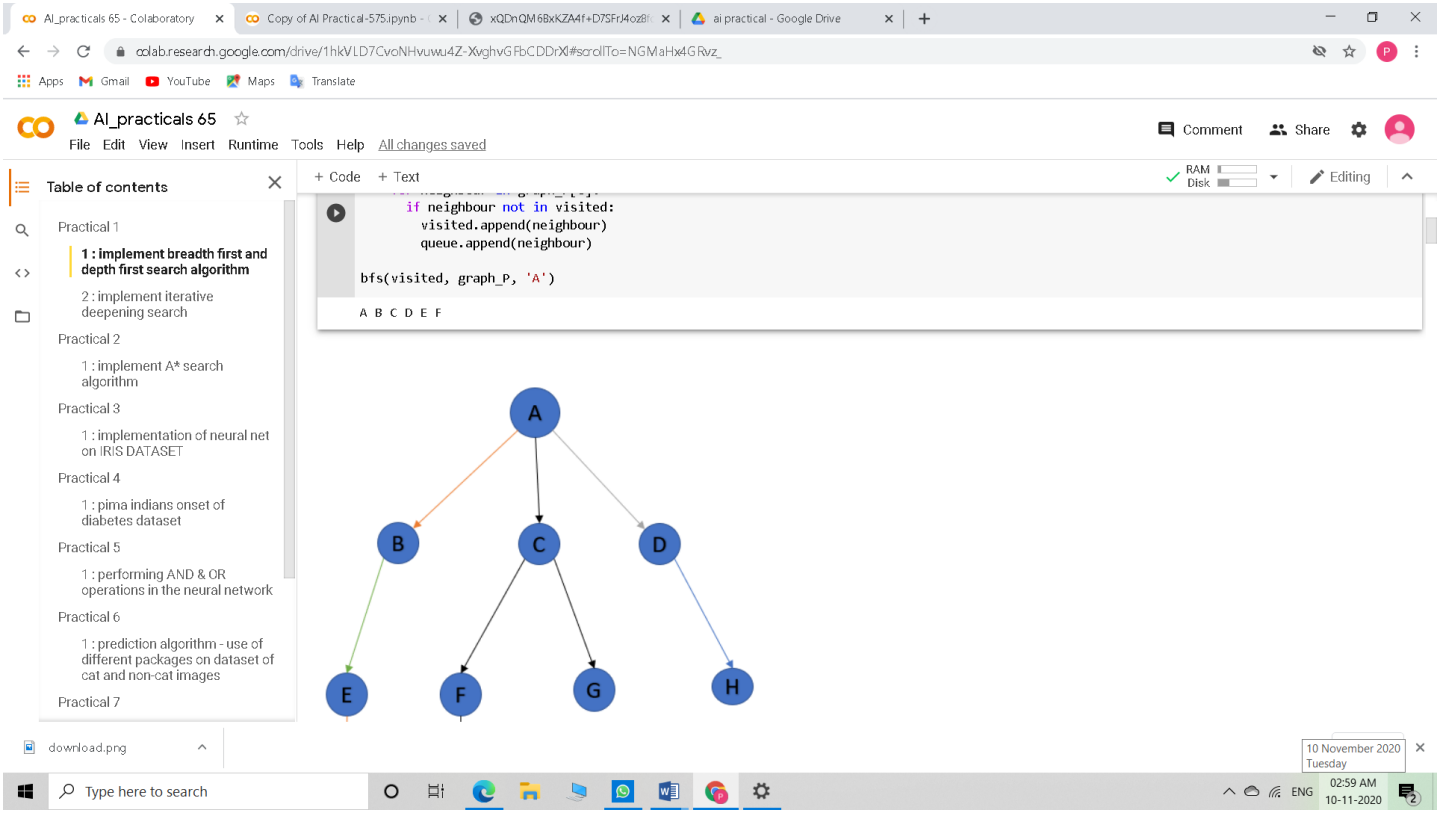


Name: Kuldeep Shushil Patel **Roll no: 574**



Name: Kuldeep Shushil Patel **Roll no: 574**

Output:



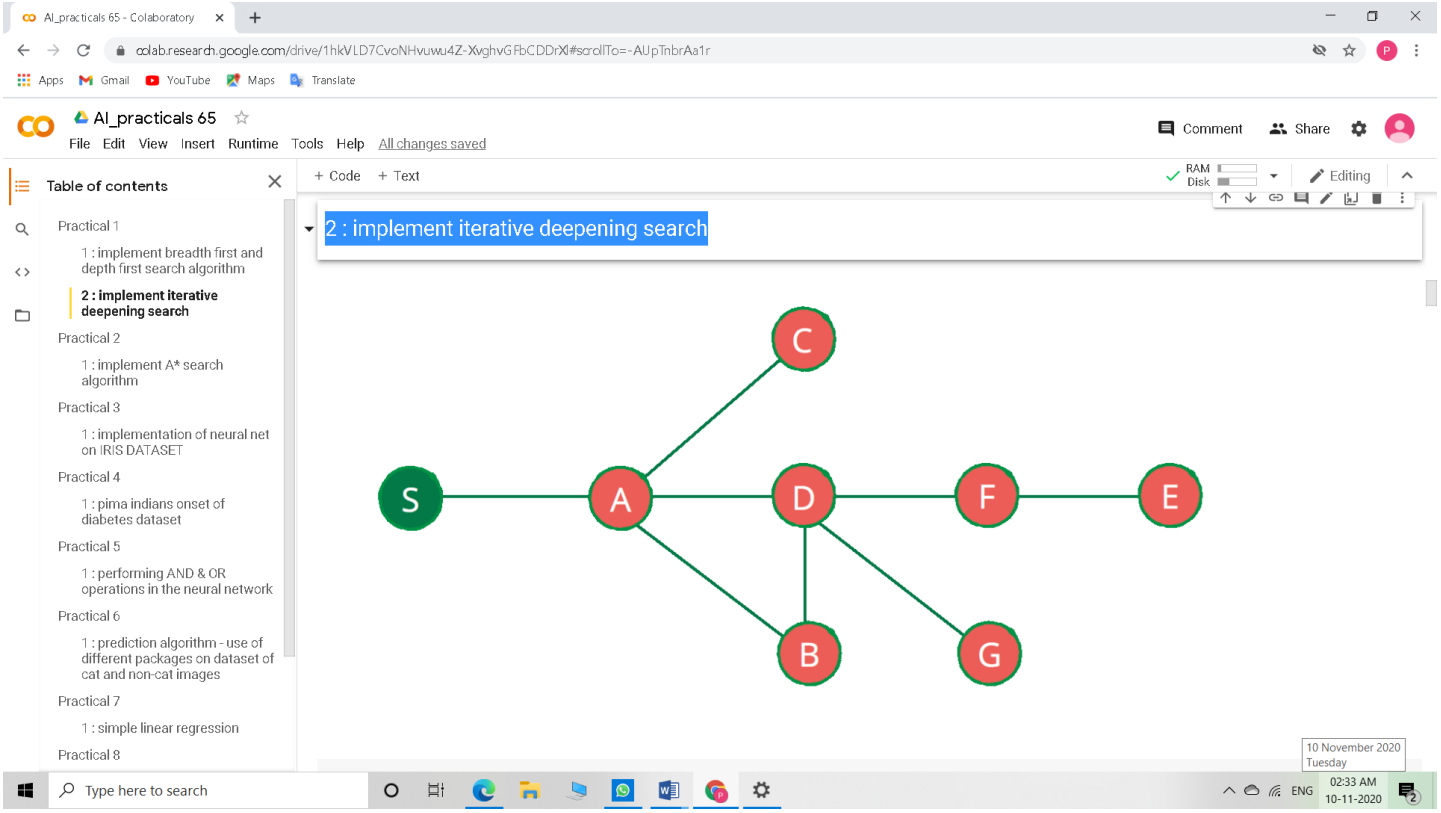
Name: Kuldeep Shushil Patel **Roll no: 574**

2] implement iterative deepening search

Theory :

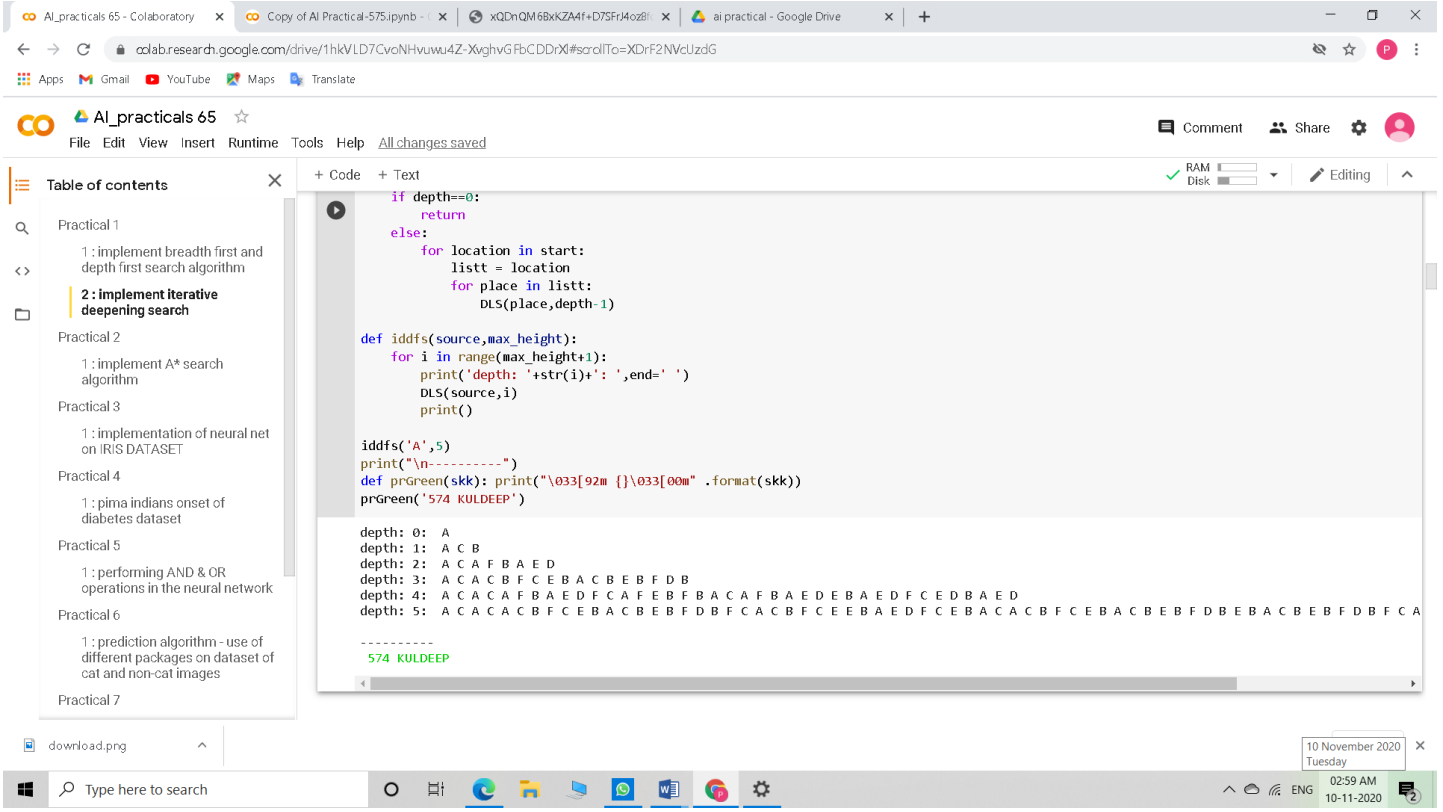
- DFS first traverses nodes going through one adjacent of root, then next adjacent.
- The problem with this approach is, if there is a node close to root, but not in first few subtrees explored by DFS, then DFS reaches that node very late.
- Also, DFS may not find shortest path to a node (in terms of number of edges).

Code:



Name: Kuldeep Shushil Patel **Roll no: 574**

Output :



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=-AUpTnbrAa1r

Name: Kuldeep Shushil Patel **Roll no: 574**

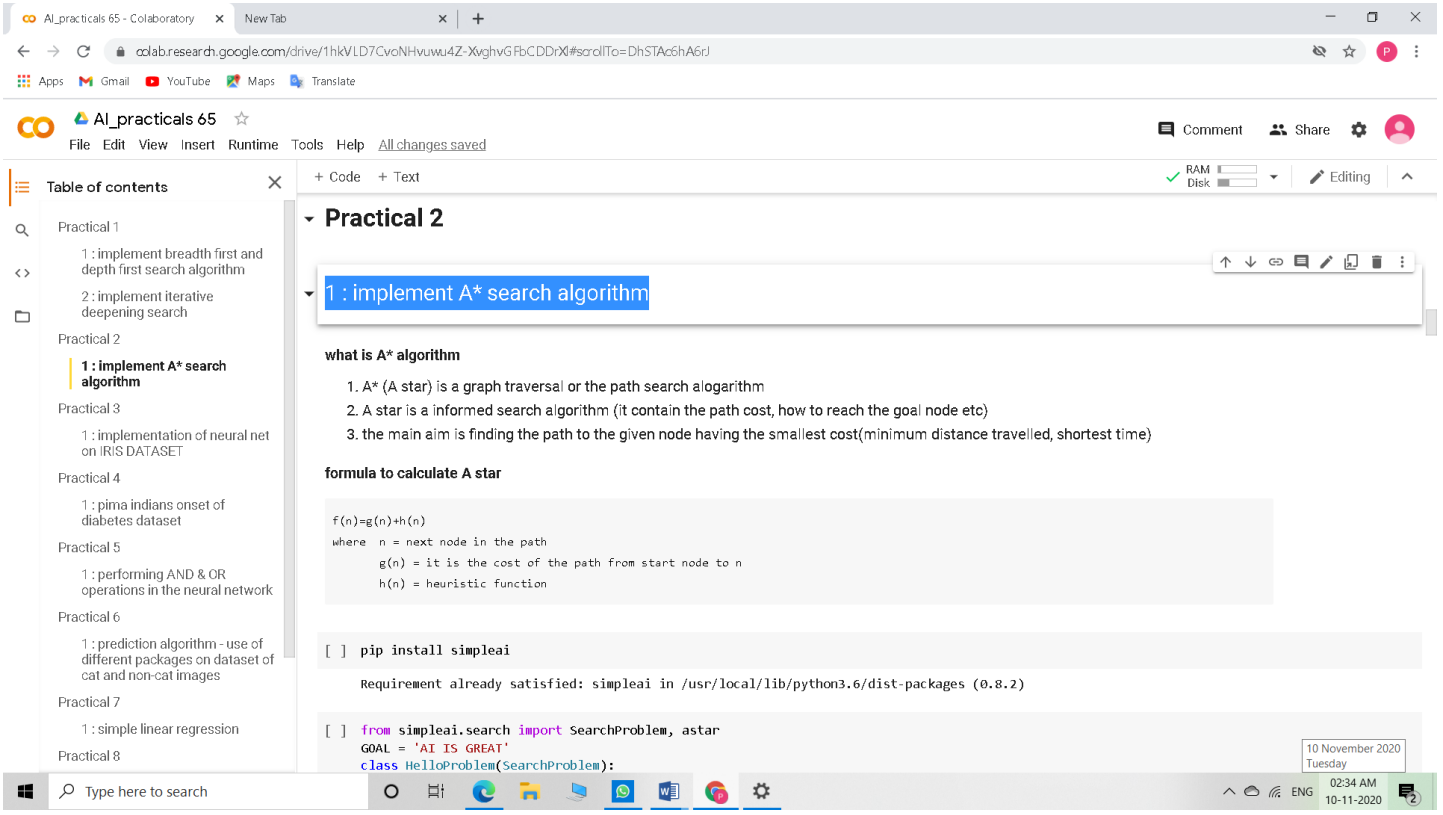
Practical 2

Aim :1] implement A* search algorithm

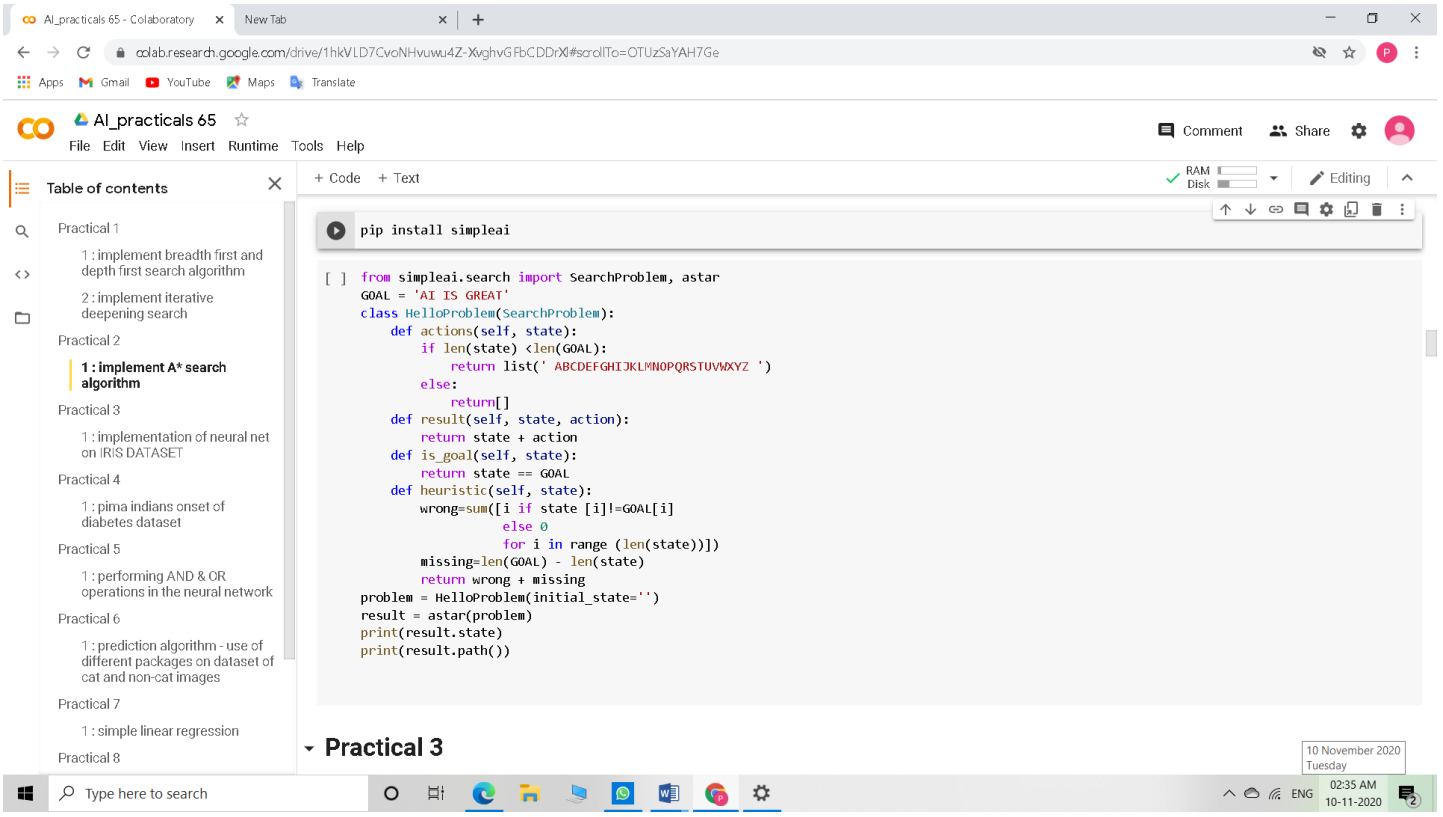
Theory :

- A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.
- Why A* Search Algorithm ?
  Informally speaking, A* Search algorithms, unlike other traversal techniques, it has “brains”. What it means is that it is really a smart algorithm which separates it from the other conventional algorithms.
- This fact is cleared in detail in below sections.
  And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

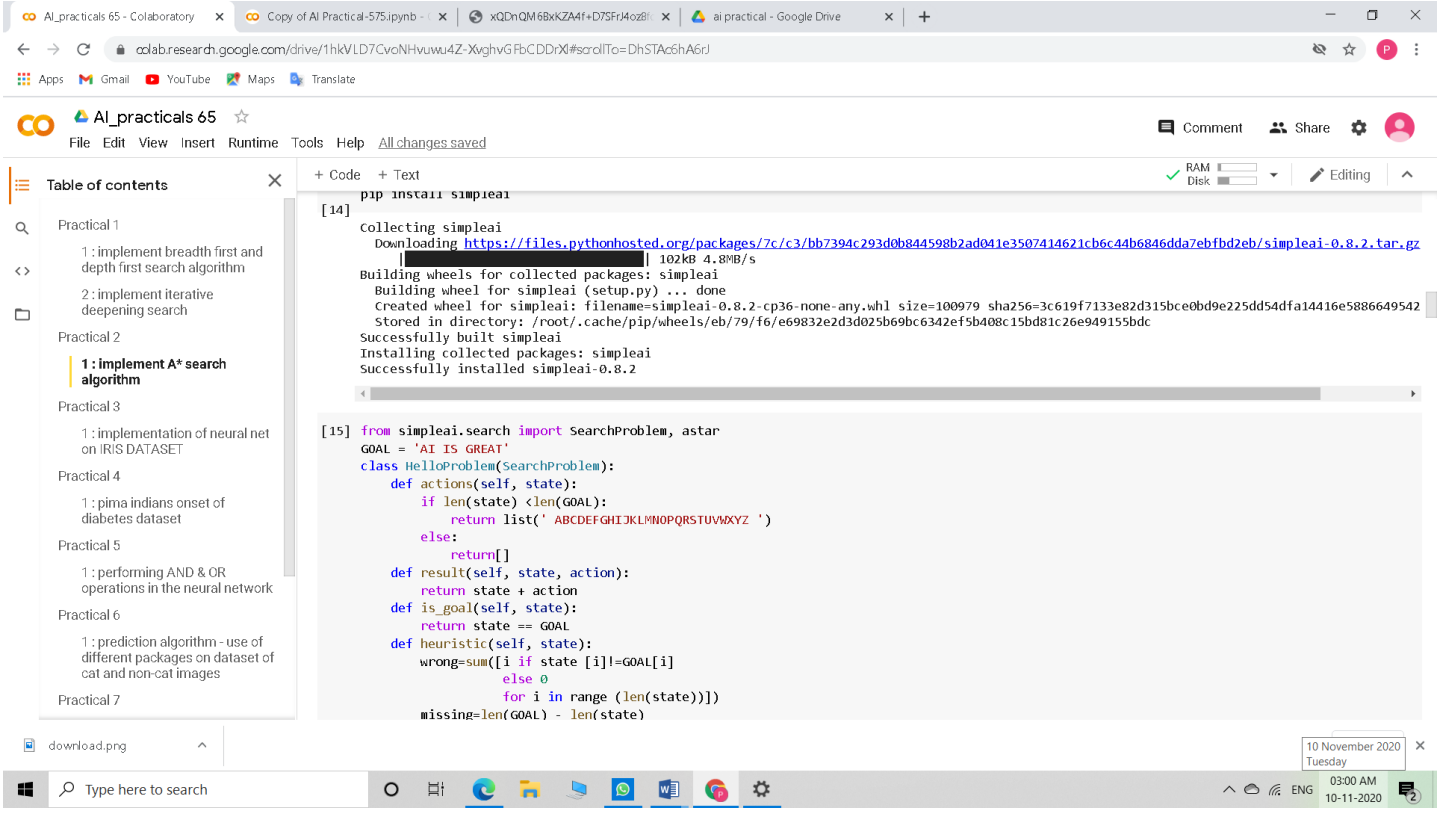
Code



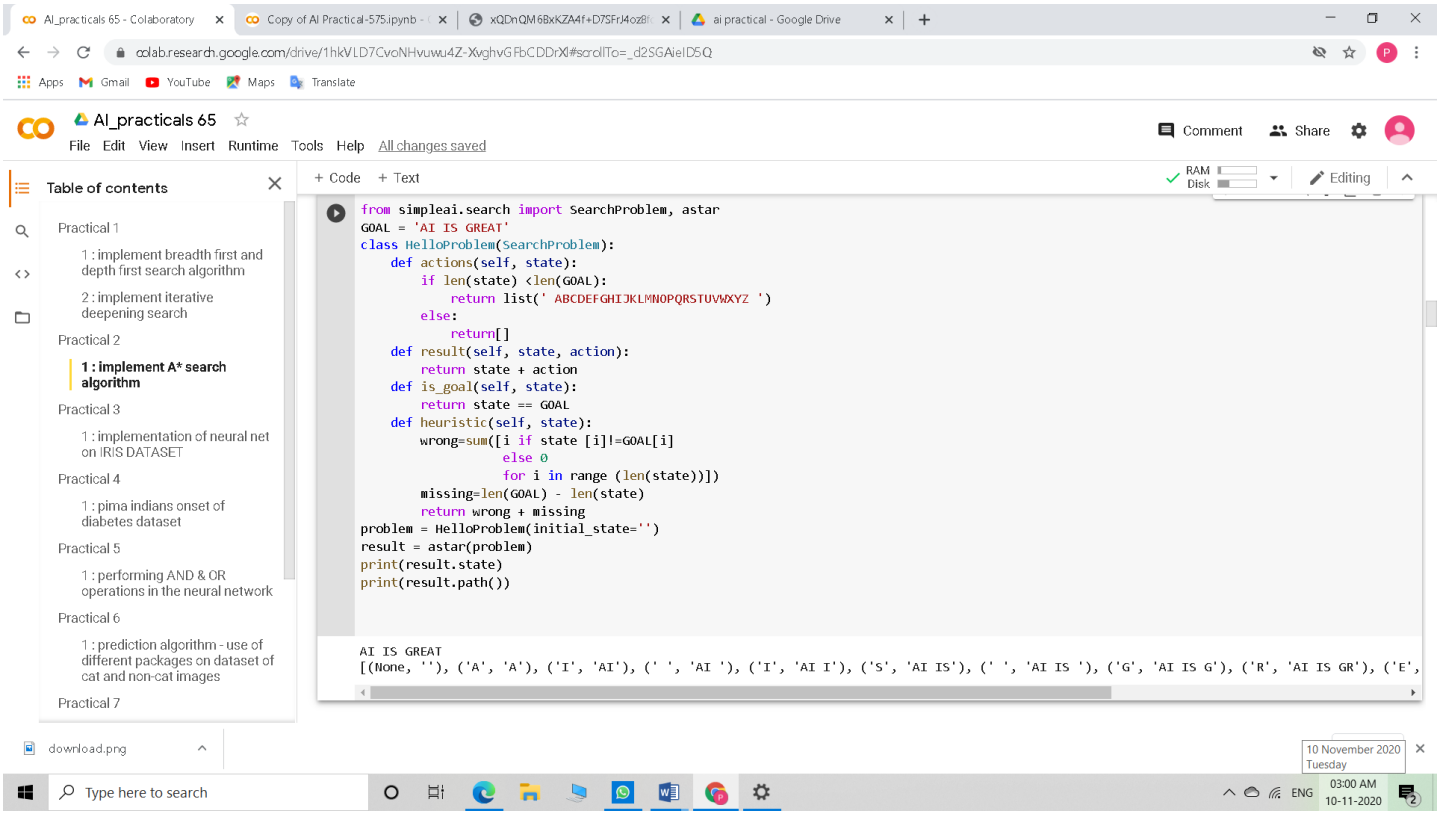
Name: Kuldeep Shushil Patel **Roll no: 574**



Output :



Name: Kuldeep Shushil Patel **Roll no: 574**



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=DhSTAc6hA6rJ

Name: Kuldeep Shushil Patel **Roll no: 574**

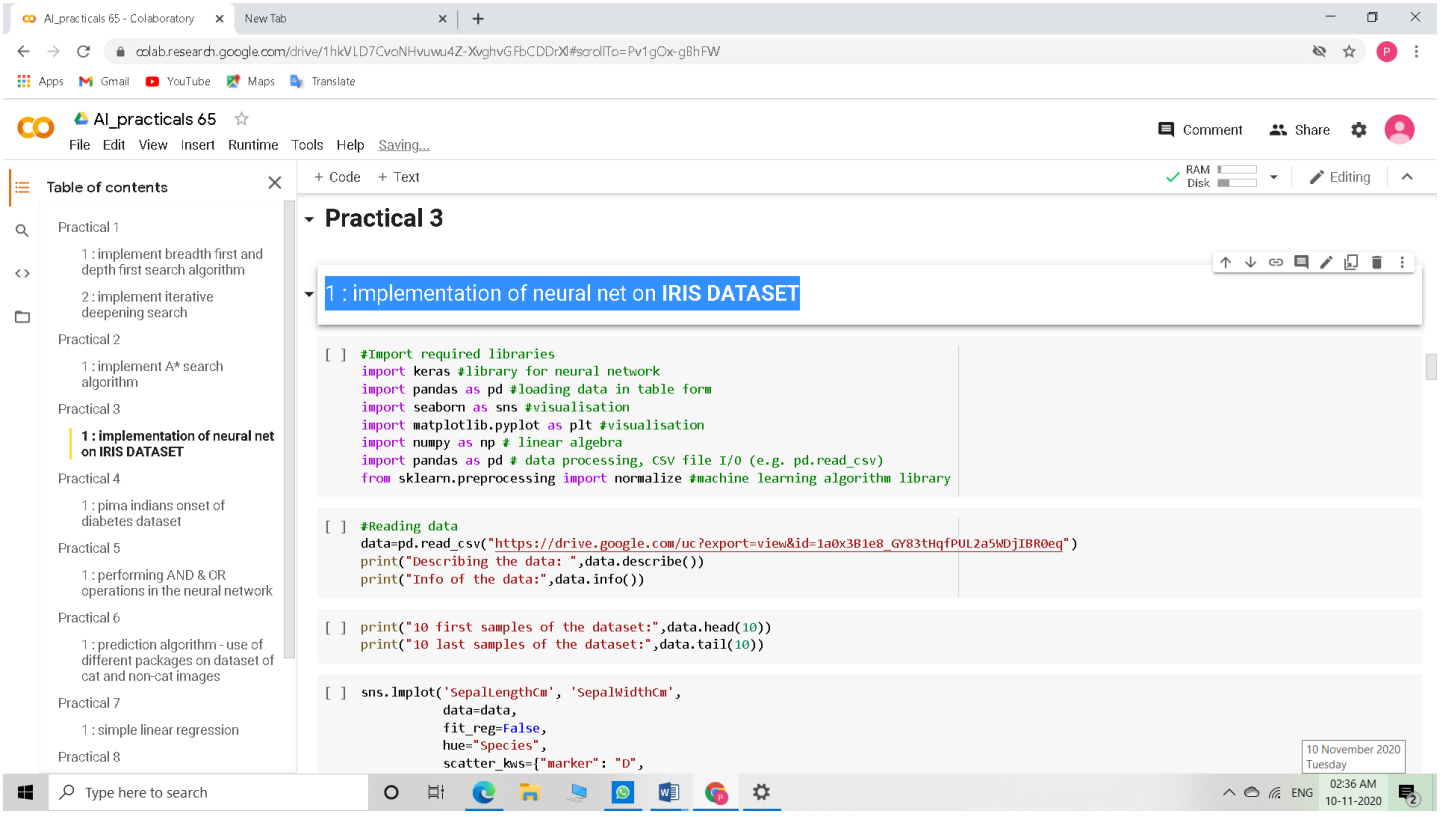
Practical 3

Aim: implementation of neural net on IRIS DATASET

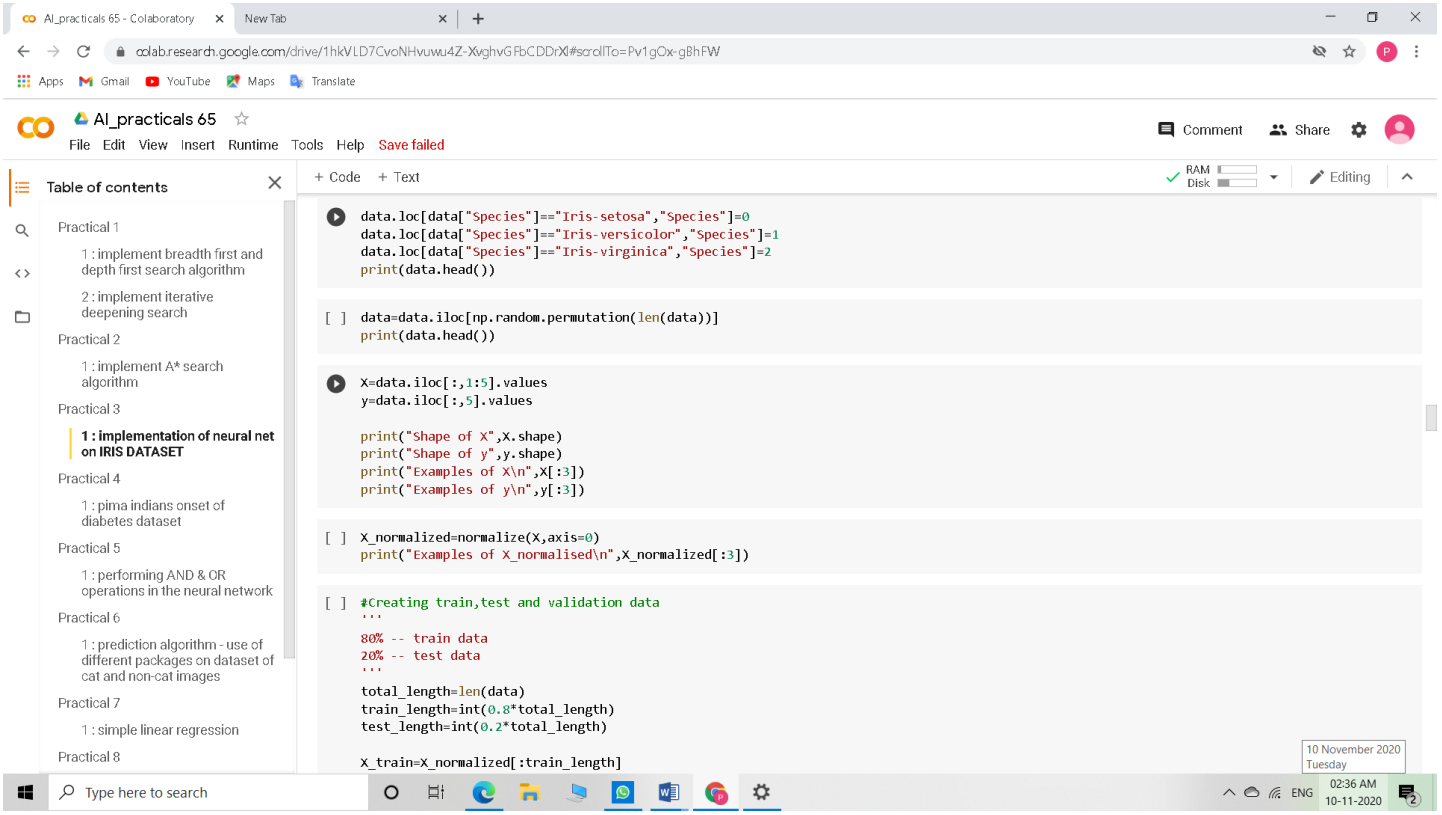
Theory:

- A neuron consists of a dendrite and an axon which are responsible for collecting and sending signals.
- For our artificial neural network, the concept works similar in which a lot of neurons are connected to each layer with its own corresponding weight and biases.

Code :



Name: Kuldeep Shushil Patel **Roll no: 574**



```
data.loc[data["Species"]=="Iris-setosa","Species"]=0
data.loc[data["Species"]=="Iris-versicolor","Species"]=1
data.loc[data["Species"]=="Iris-virginica","Species"]=2
print(data.head())
```

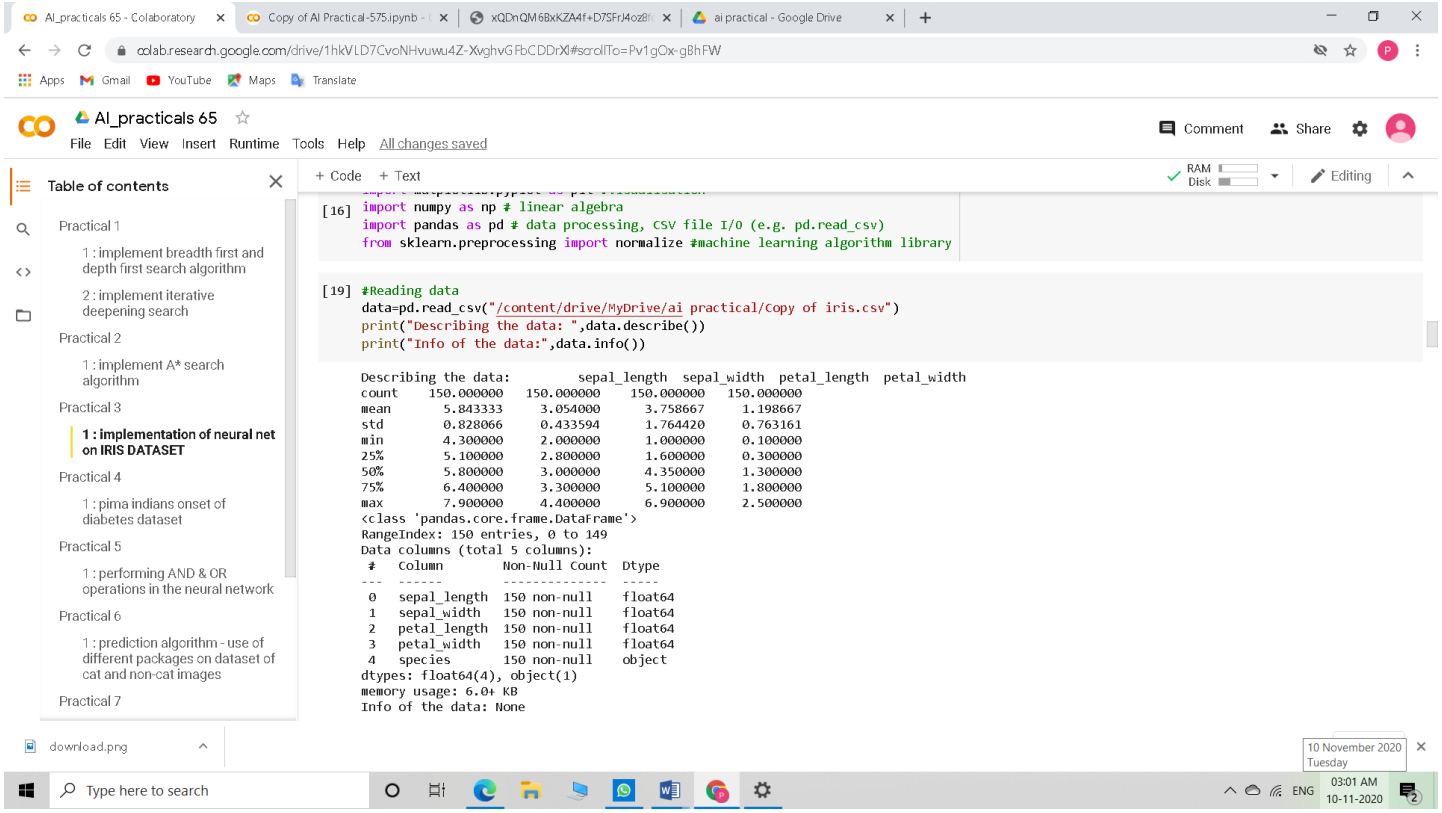
```
data=data.iloc[np.random.permutation(len(data))]
print(data.head())
```

```
X=data.iloc[:,1:5].values
y=data.iloc[:,5].values

print("Shape of X",X.shape)
print("Shape of y",y.shape)
print("Examples of X\n",X[:3])
print("Examples of y\n",y[:3])
```

```
X_normalized=normalize(X,axis=0)
print("Examples of X_normalised\n",X_normalized[:3])
```

```
#Creating train,test and validation data
'''
80% -- train data
20% -- test data
'''
total_length=len(data)
train_length=int(0.8*total_length)
test_length=int(0.2*total_length)

X_train=X_normalized[:train_length]
```

Output:

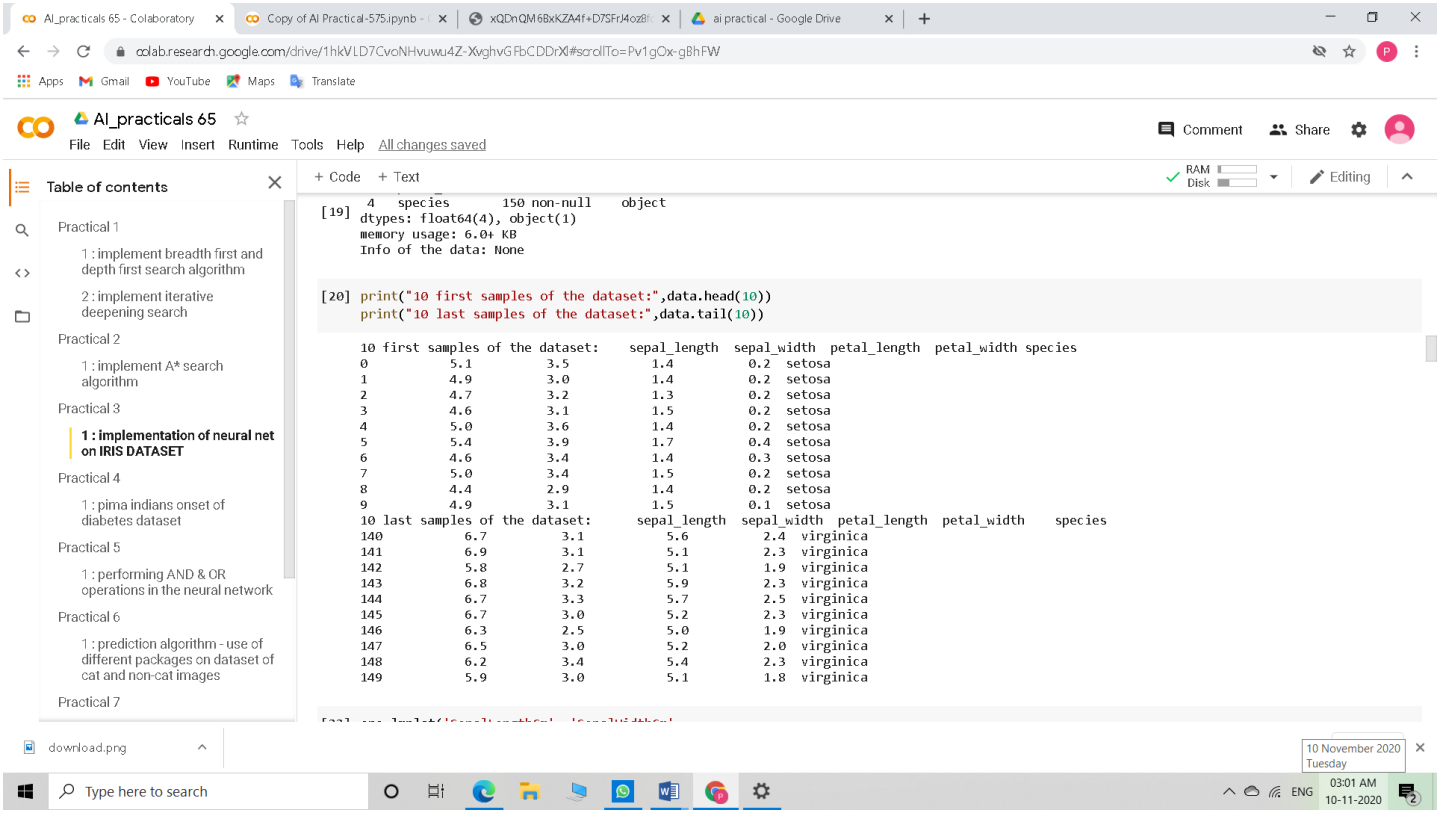


```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.preprocessing import normalize #machine learning algorithm library
```

```
#Reading data
data=pd.read_csv("/content/drive/MyDrive/ai practical/Copy of iris.csv")
print("Describing the data: ",data.describe())
print("Info of the data:",data.info())
```

```
Describing the data:         sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
Info of the data: None
```

Name: Kuldeep Shushil Patel **Roll no: 574**



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=Pv1gOx-gBhFW

Name: Kuldeep Shushil Patel **Roll no: 574**

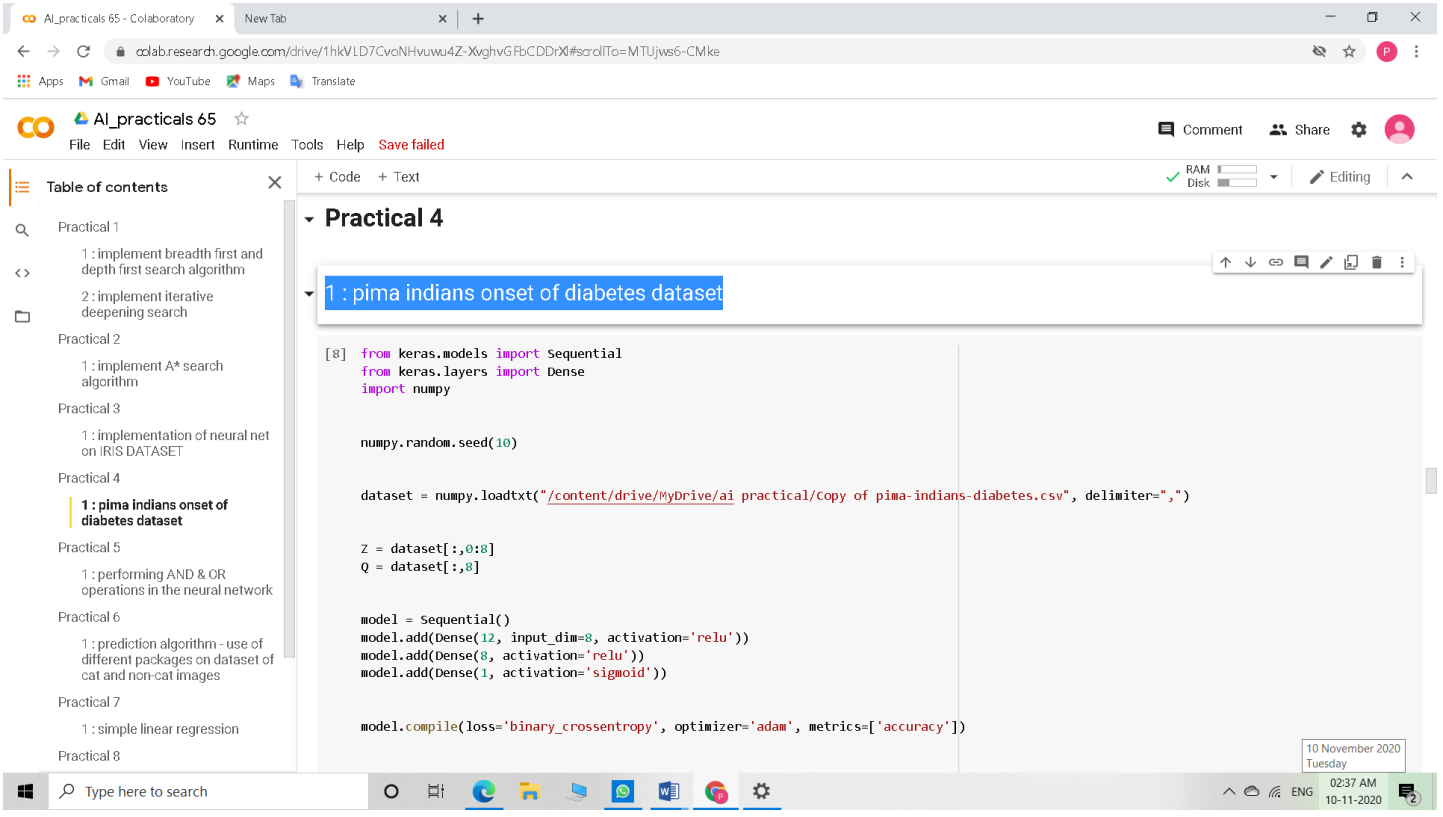
Practical 4

Aim: pima indians onset of diabetes dataset

Theory :

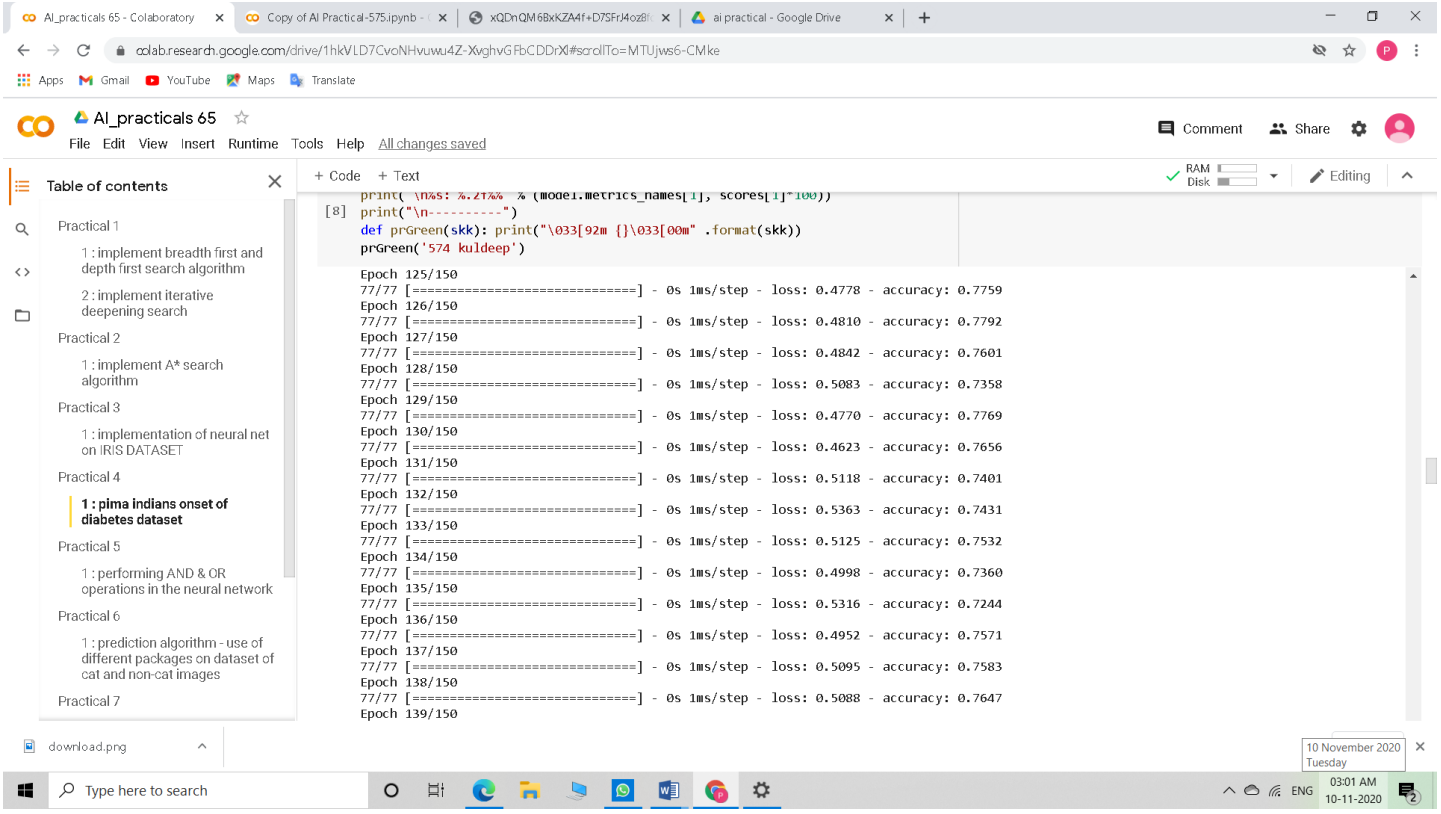
- This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.
- The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset.
- Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Code:



Name: Kuldeep Shushil Patel **Roll no: 574**

Output:



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=MTUjws6-CMke

Name: Kuldeep Shushil Patel **Roll no: 574**

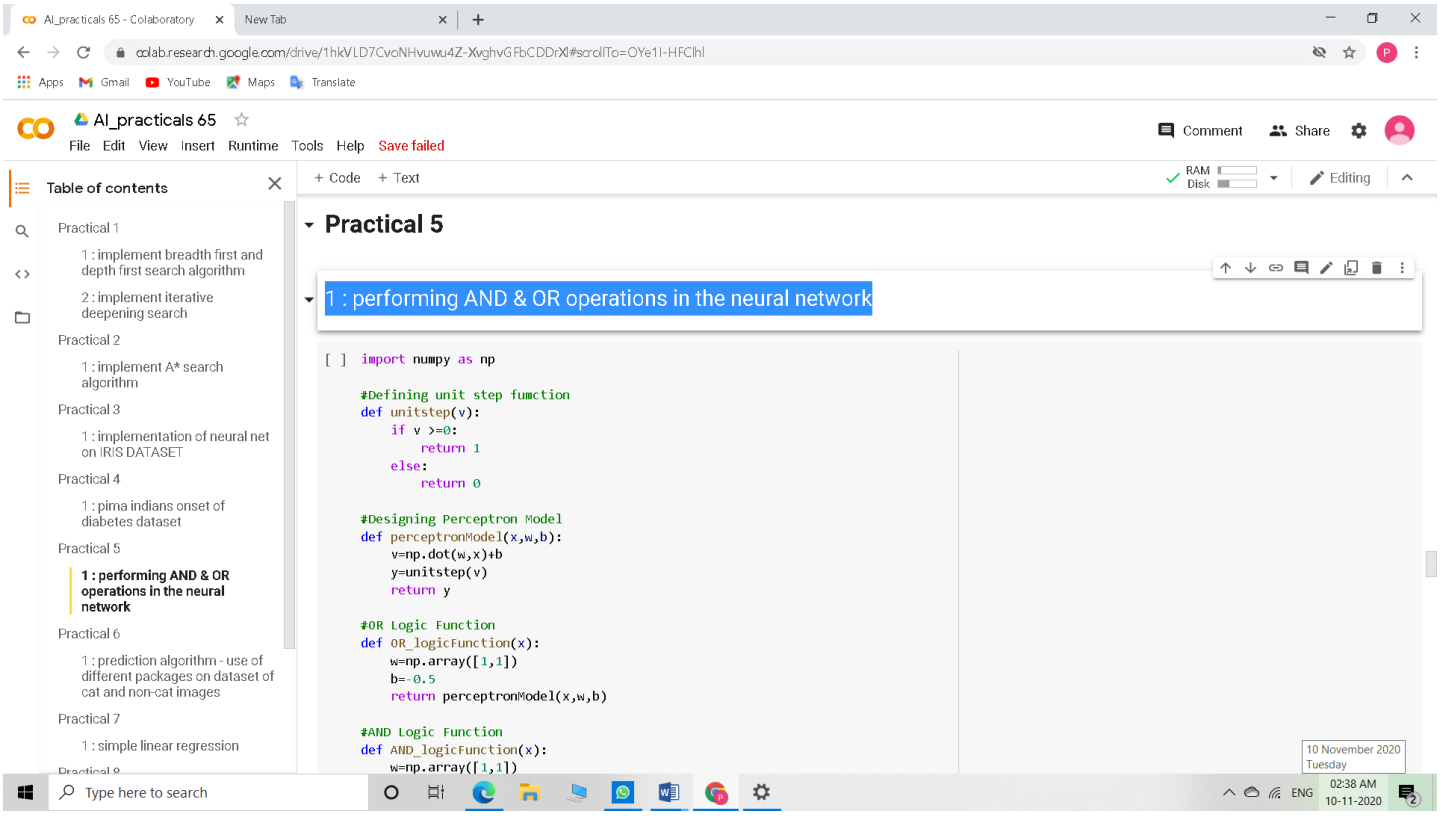
Practical 5

Aim : performing AND & OR operations in the neural network

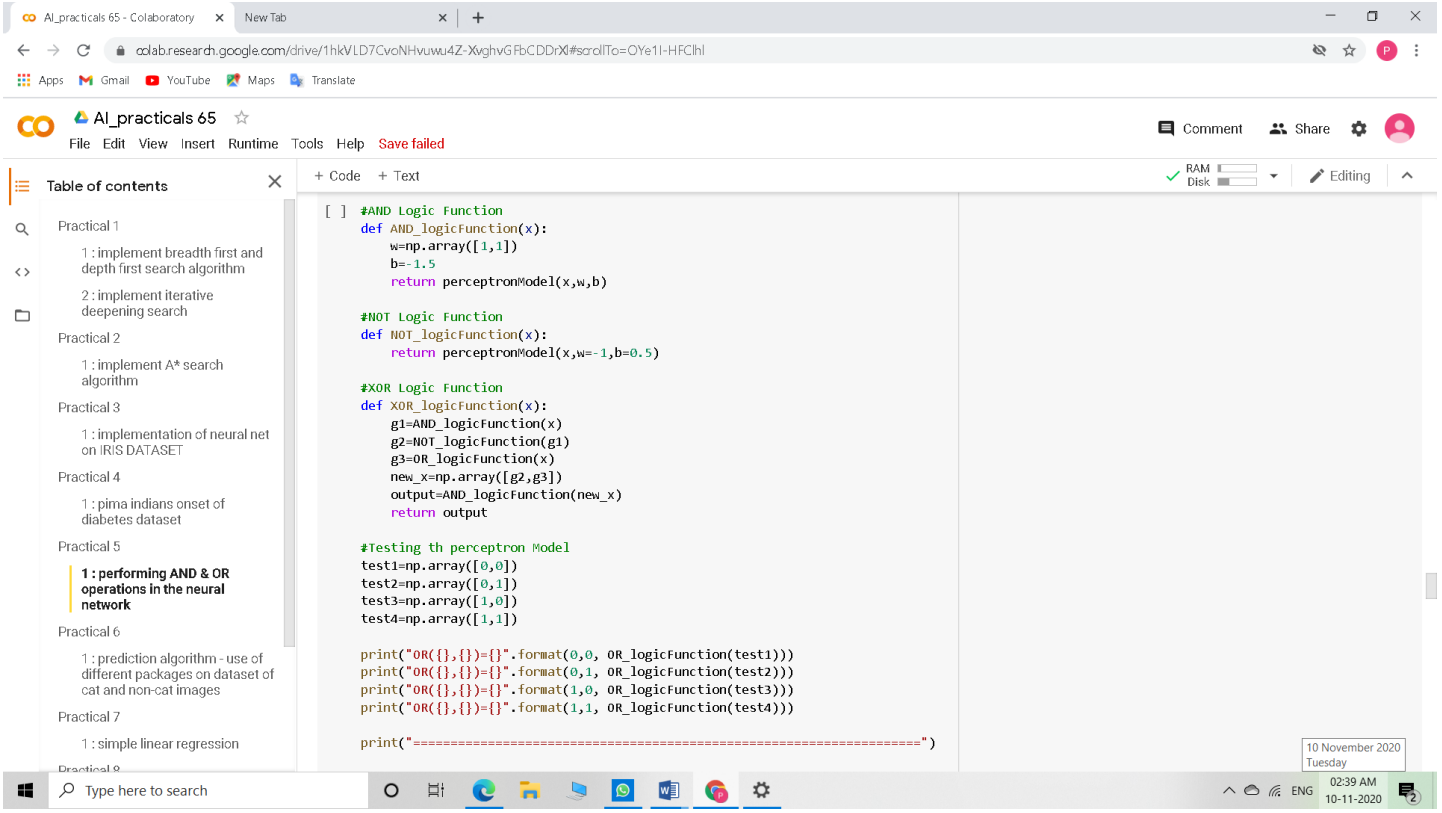
Theory :

- A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
- In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input;
- so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks

Code:

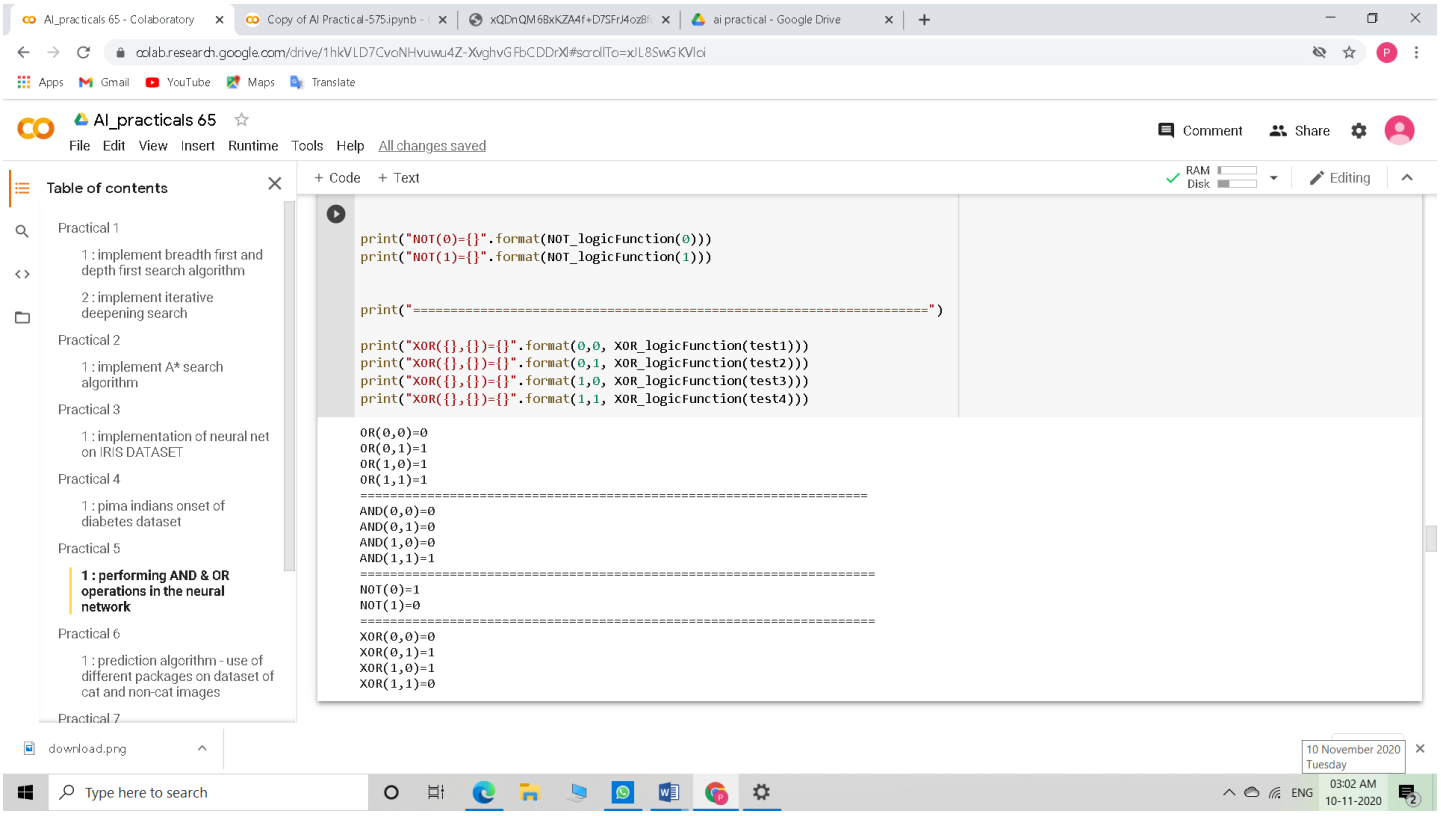


Name: Kuldeep Shushil Patel **Roll no: 574**



Name: Kuldeep Shushil Patel **Roll no: 574**

Output :



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=OYe1I-HFClhl

Name: Kuldeep Shushil Patel **Roll no: 574**

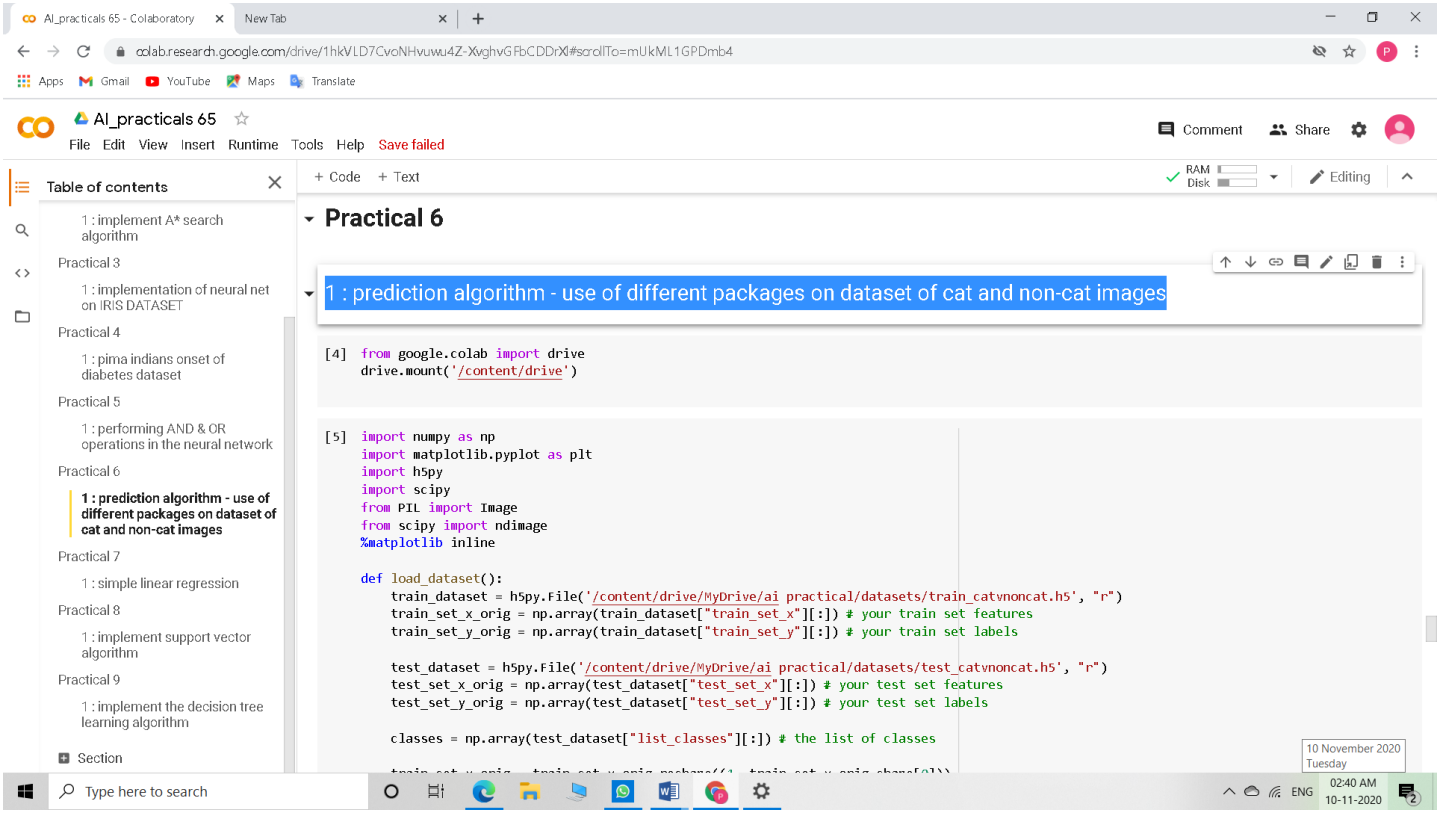
Practical 6

Aim : prediction algorithm - use of different packages on dataset of cat and non-cat images

Theory :

- Predictive analytics algorithms try to achieve the lowest error possible by either using “boosting”
- (a technique which adjusts the weight of an observation based on the last classification) or “bagging” (which creates subsets of data from training samples, chosen randomly with replacement).
- Random Forest uses bagging.

Code:

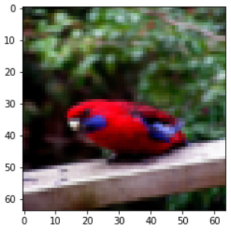


Name: Kuldeep Shushil Patel **Roll no: 574**

```
[5]  import numpy as np
     import matplotlib.pyplot as plt
     import h5py
     import scipy
     from PIL import Image
     from scipy import ndimage
     %matplotlib inline

     def load_dataset():
         train_dataset = h5py.File('/content/drive/MyDrive/ai practical/datasets/train_catvnoncat.h5', "r")
         train_set_x_orig = np.array(train_dataset["train_set_x"][:]) # your train set features
         train_set_y_orig = np.array(train_dataset["train_set_y"][:]) # your train set labels

         test_dataset = h5py.File('/content/drive/MyDrive/ai practical/datasets/test_catvnoncat.h5', "r")
         test_set_x_orig = np.array(test_dataset["test_set_x"][:]) # your test set features
         test_set_y_orig = np.array(test_dataset["test_set_y"][:]) # your test set labels

         classes = np.array(test_dataset["list_classes"][:]) # the list of classes

         train_set_y_orig = train_set_y_orig.reshape((1, train_set_y_orig.shape[0]))
         test_set_y_orig = test_set_y_orig.reshape((1, test_set_y_orig.shape[0]))

         return train_set_x_orig, train_set_y_orig, test_set_x_orig, test_set_y_orig, classes

     # Loading the data (non-cat)
     train_set_x_orig, train_set_y, test_set_x_orig, test_set_y, classes = load_dataset()
     index = 35
     plt.imshow(train_set_x_orig[index])
     print ("y = " + str(train_set_y[:, index]) + ", it's a " + classes[np.squeeze(train_set_y[:, index])].decode("utf-8") +  " picture")
```

Output :

```
y = [0], it's a non-cat picture
```



Name: Kuldeep Shushil Patel **Roll no: 574**

Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=mUkML1GPDmb4

Name: Kuldeep Shushil Patel **Roll no: 574**

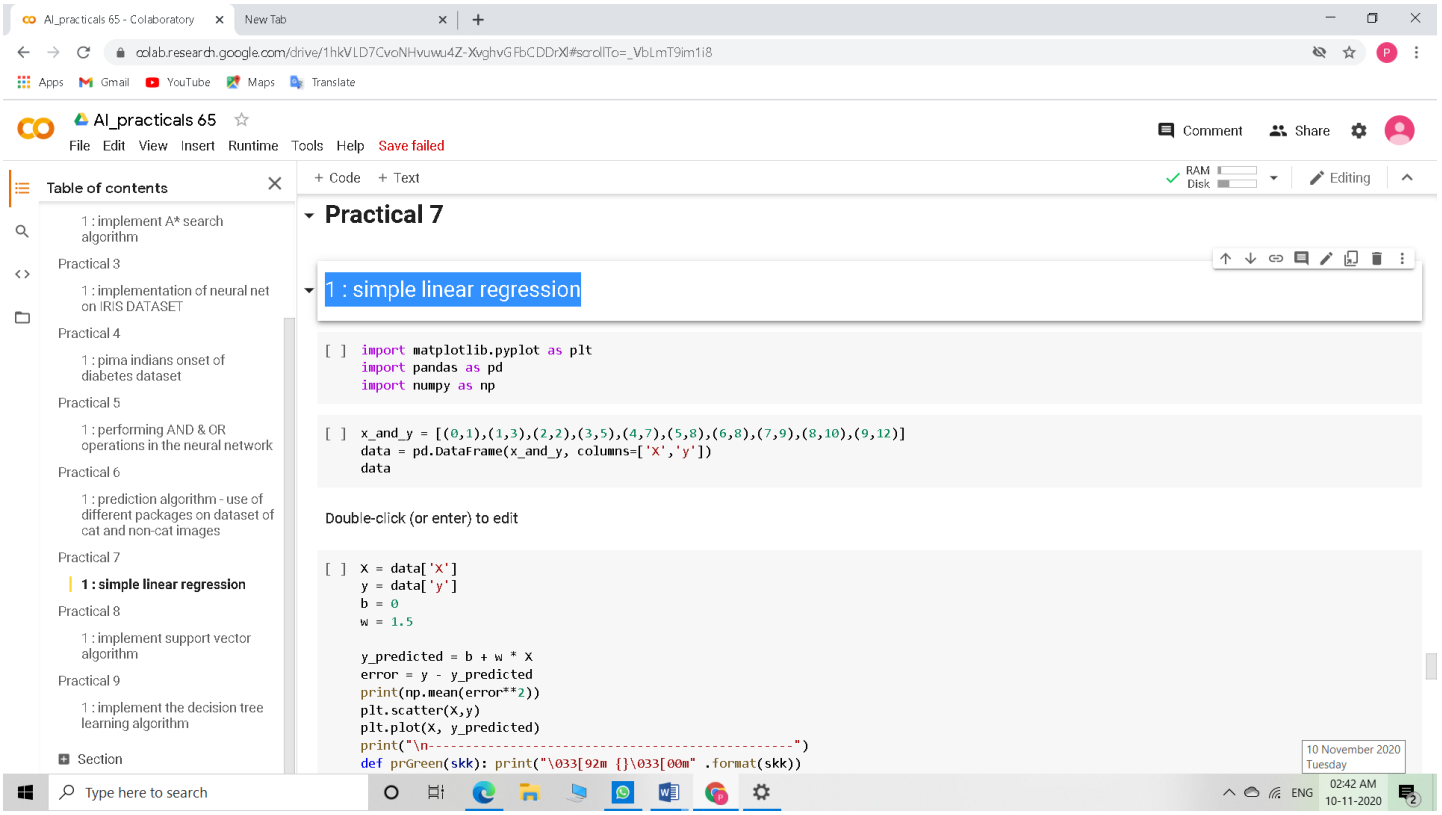
Practical 7

Aim : simple linear regression

Theory :

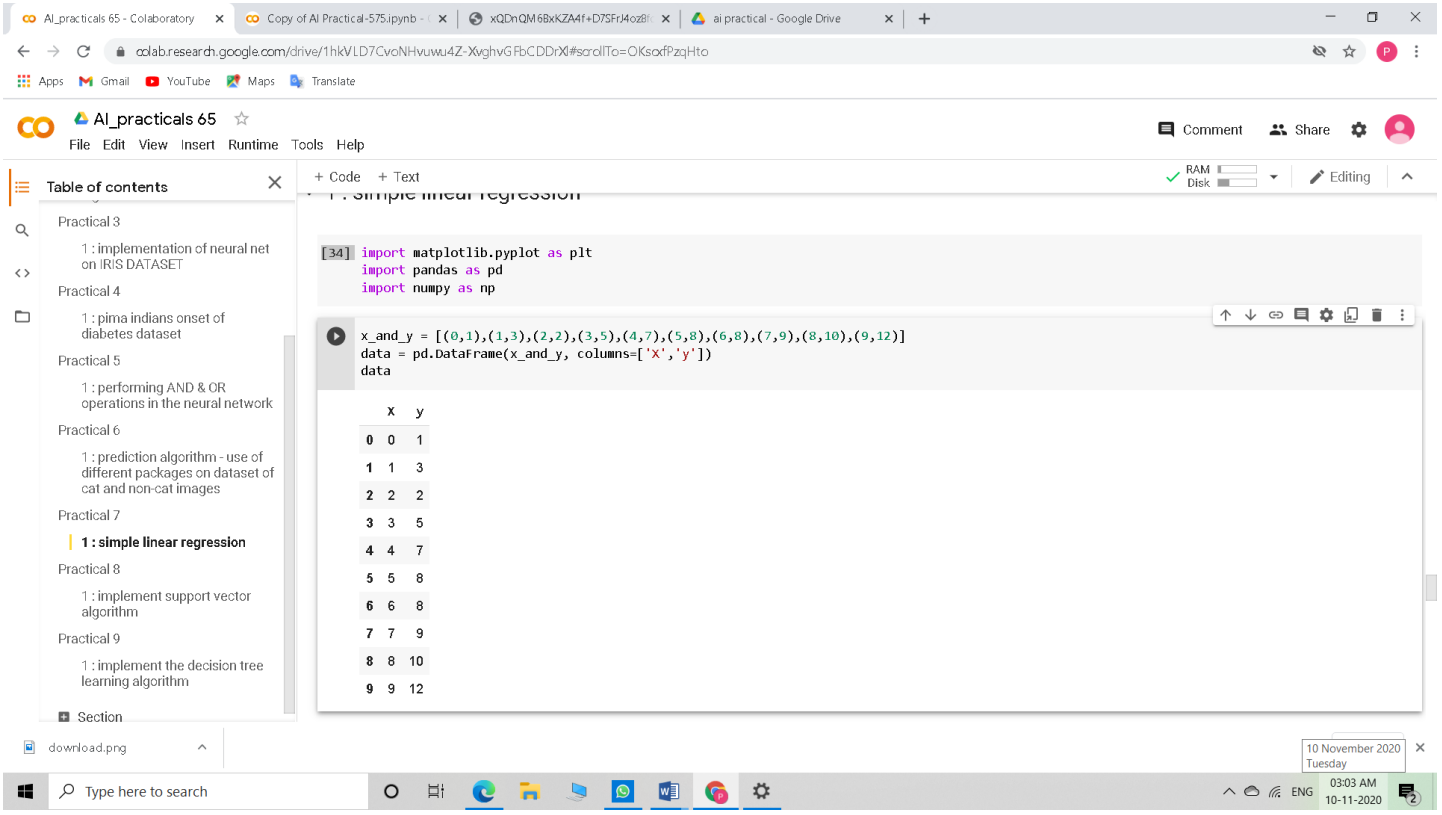
- **Simple linear regression** is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables:
- One variable, denoted *x*, is regarded as the **predictor**, **explanatory**, or **independent** variable.
- The other variable, denoted *y*, is regarded as the **response**, **outcome**, or **dependent** variable.
- Because the other terms are used less frequently today, we'll use the "**predictor**" and "**response**" terms to refer to the variables encountered in this course.

Code :



Name: Kuldeep Shushil Patel **Roll no: 574**

Output :



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=_VbLmT9im1i8

Name: Kuldeep Shushil Patel **Roll no: 574**

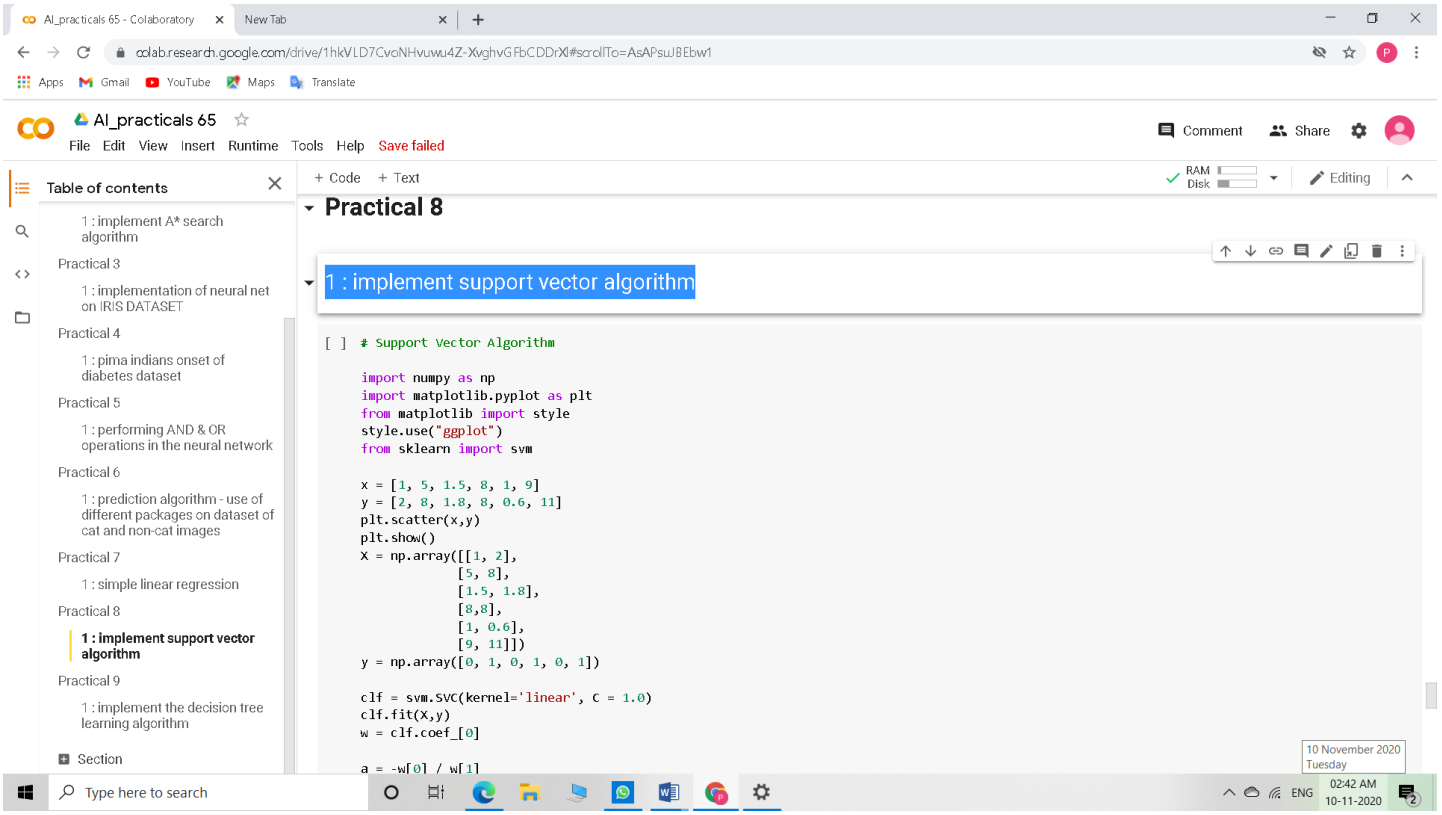
Practical 8

Aim : implement support vector algorithm

Theory :

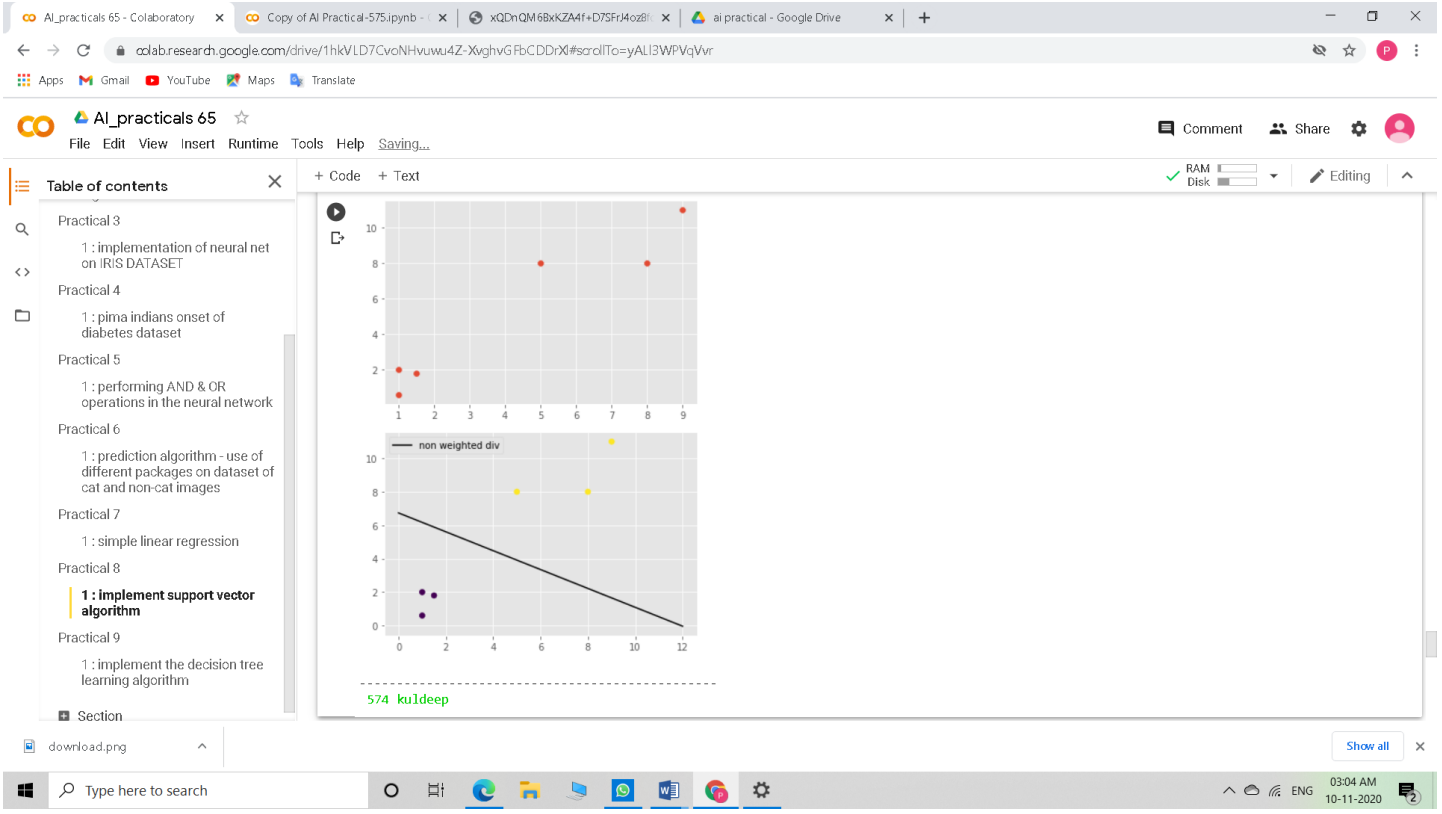
- Support Vector Machine" (SVM) is a supervised machine learning algorithm
- which can be used for both classification or regression challenges.
- it is mostly used in classification problems.
- Support Vectors are simply the co-ordinates of individual observation

Code:



Name: Kuldeep Shushil Patel **Roll no: 574**

Output :



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=AsAPsuJBEbw1

Name: Kuldeep Shushil Patel **Roll no: 574**

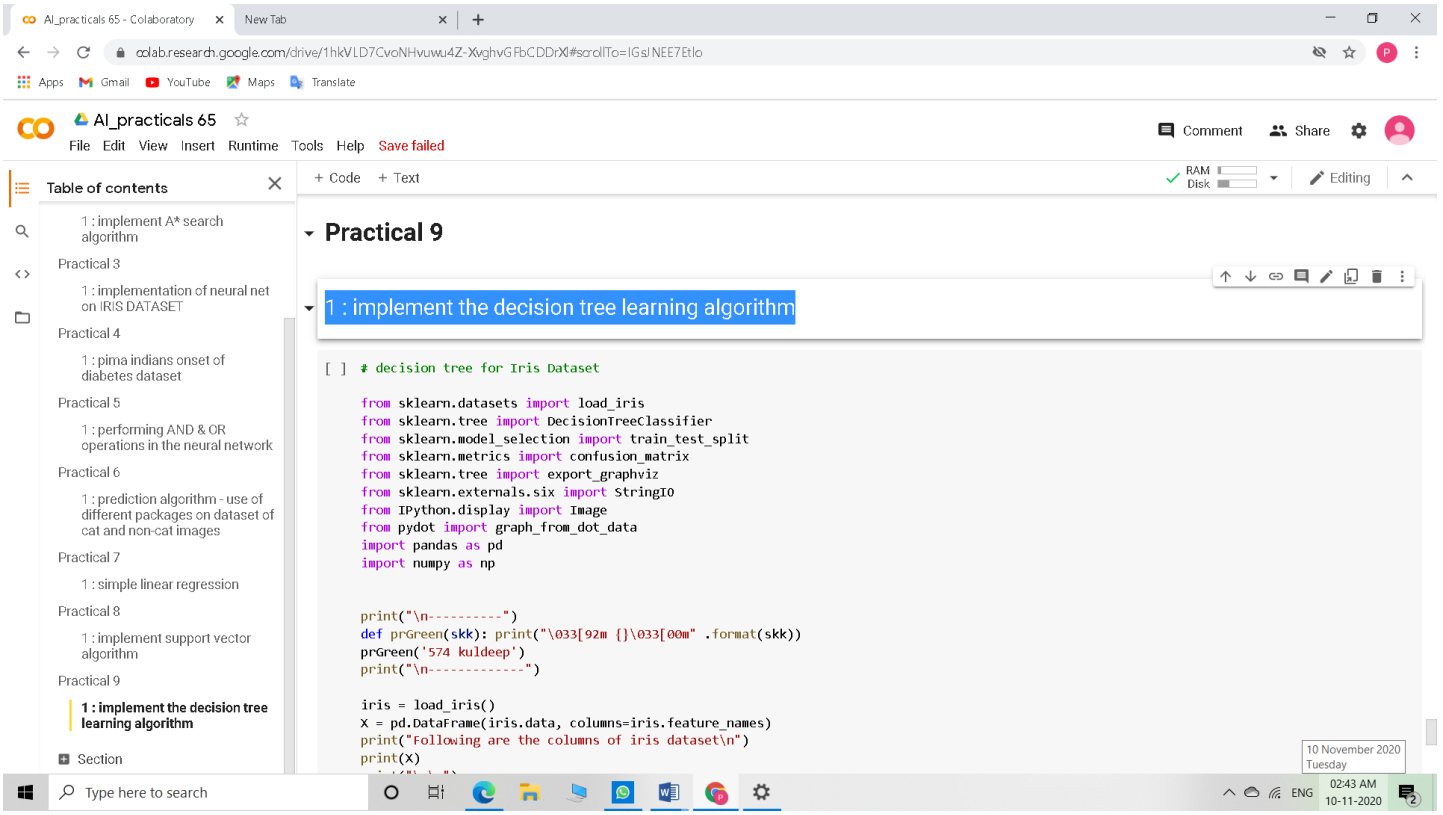
Practical 9

Aim : implement the decision tree learning algorithm

Theory :

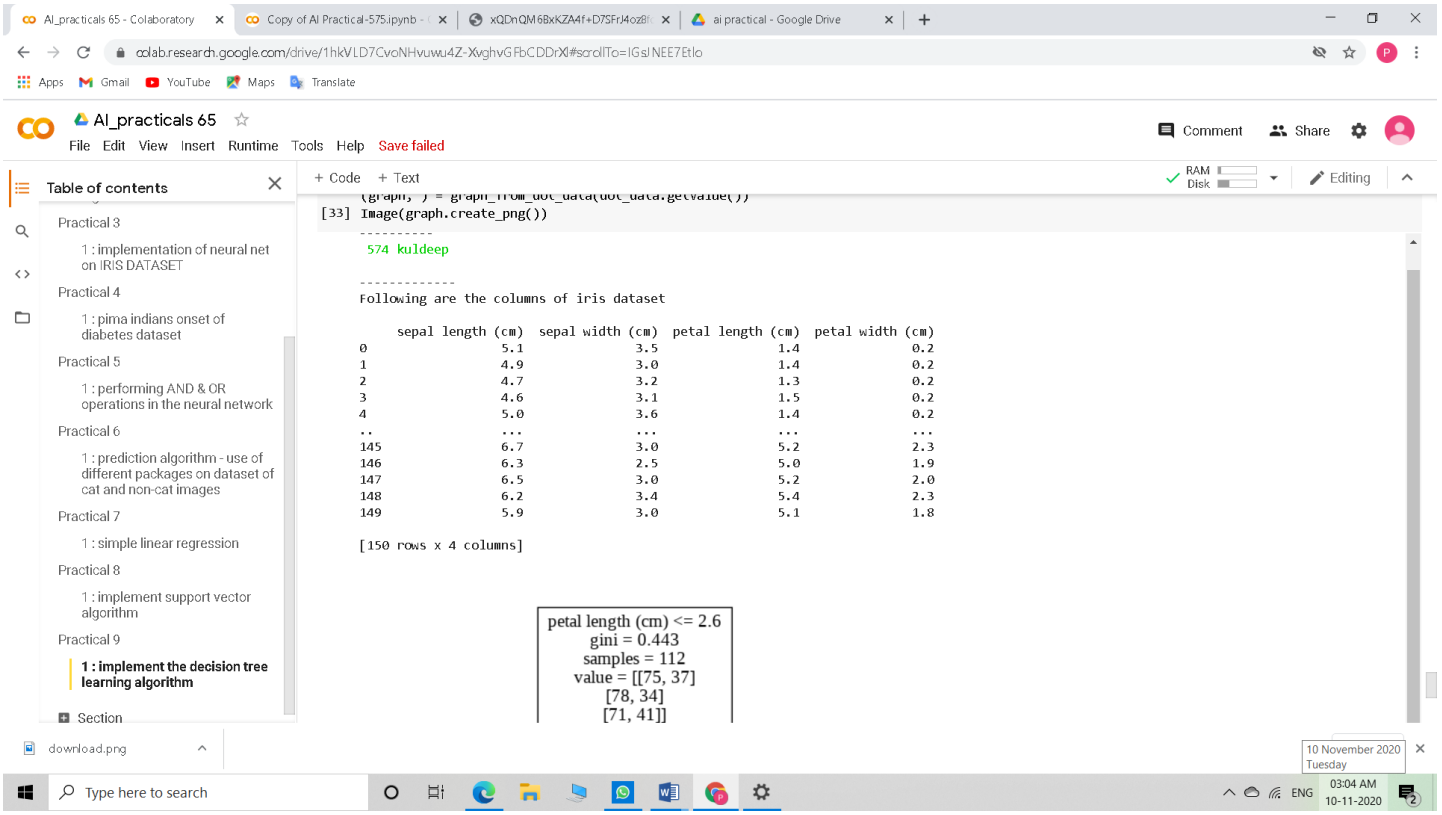
- **Decision Trees** are a type of Supervised **Machine Learning**.
- what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter.
- The **tree** can be explained by two entities, namely **decision** nodes and leaves

Code :

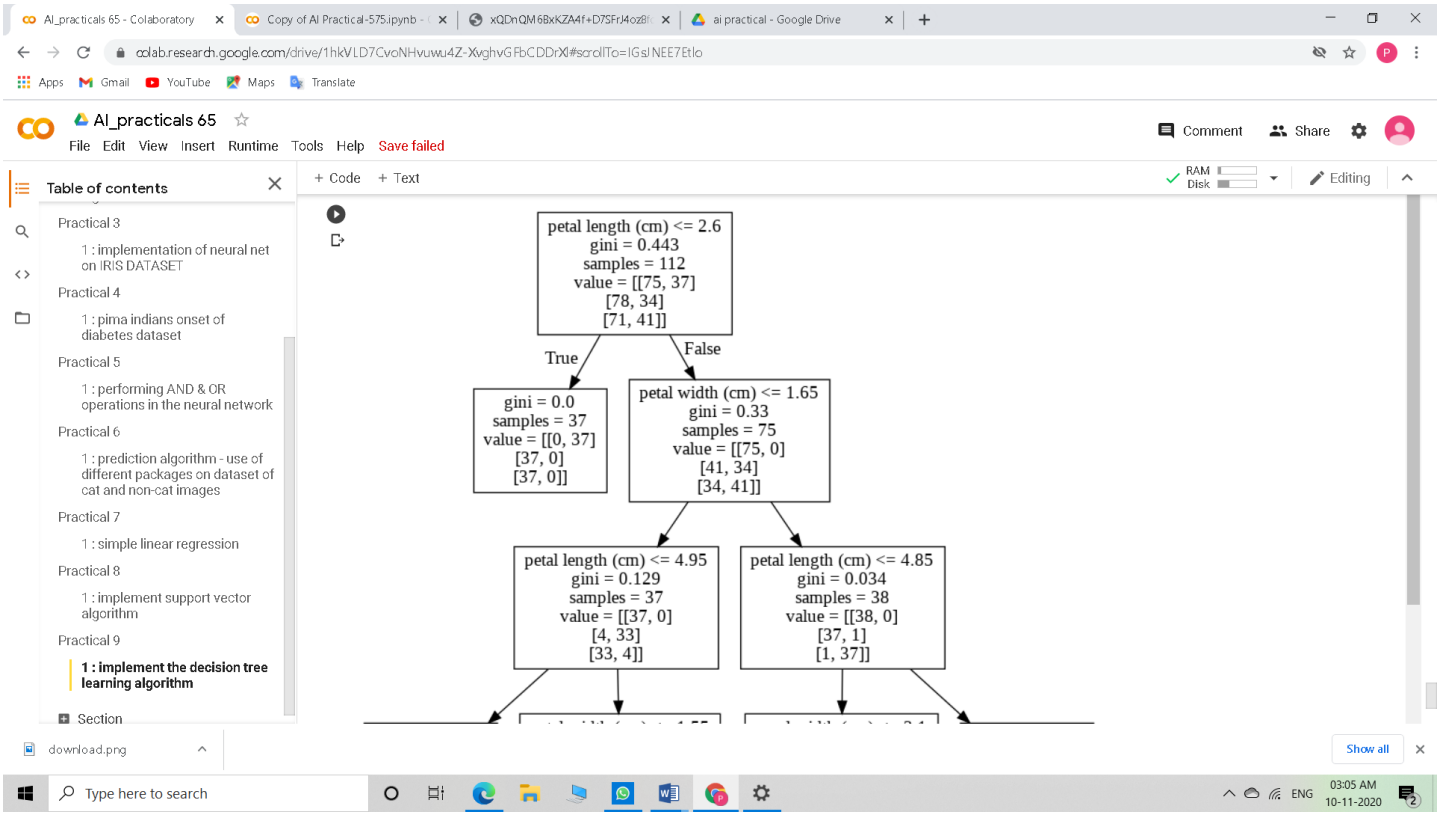


Name: Kuldeep Shushil Patel **Roll no: 574**

Output :



Name: Kuldeep Shushil Patel **Roll no: 574**



Colab link :

https://colab.research.google.com/drive/1hkVLD7CvoNHvuwu4Z-XvghvGFbCDDrXl#scrollTo=IGsJNEE7Etlo