***Machine Learning: Assignment Report***

# **Prediction using Neural Networks**

**Group Members:**

1. Name: Rishu Kumar

Roll : BT18CSE060

E-mail: [rishu5727@gmail.com](mailto:rishu5727@gmail.com)

1. Name: Neha Pasine

Roll : BT18CSE017

E-mail: [nehapasinegp26@gmail.com](mailto:nehapasinegp26@gmail.com)

## **Abstract :**

Implementation of Back-Propogation Neural network for predicting the occurrence of Heart Disease in a person with the help of a ready dataset.

## 

## **Dataset - Heart Disease Prediction**

*(Ready Dataset)*

* No. of rows - 918
* No. of features - 8
* Output - binary
* Training : Testing - 4:1
* Dataset size - 36 kB

### **Input Attributes:**

Total Input Attributes: 8

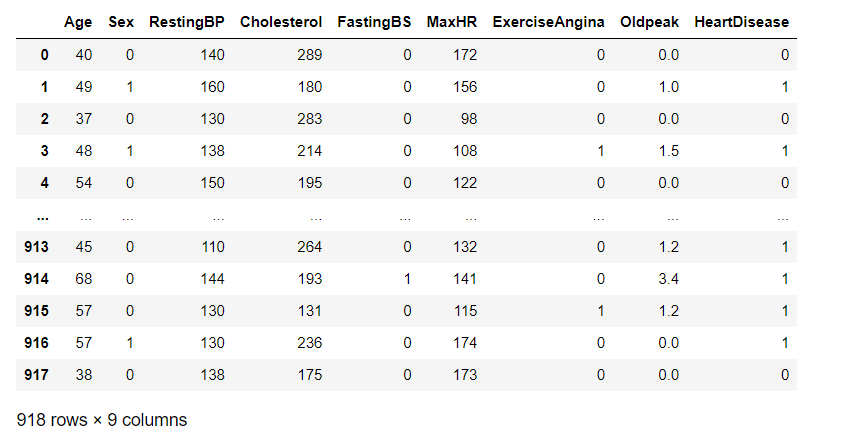
|  |  |  |
| --- | --- | --- |
| **Features** | **Type** | **Range** |
| Age | int | [28,77] |
| Sex | int | [0,1] |
| RestingBP | int | [0,200] |
| Cholesterol | int | [0,603] |
| FastingBS | int | [0,1] |
| MaxHR | int | [60,202] |
| ExerciseAngina | int | [0,1] |
| Oldpeak | float | [-2.6,6.2] |

We are converting all these attributes to float data-type for processing in our code.

### **Output Attribute:**

* Total output Attributes : 1
* Feature Name: HeartDisease
* Range: [0, 1]
* Type: Int (Binary)

**Showing sample entries of the Ready Dataset:**

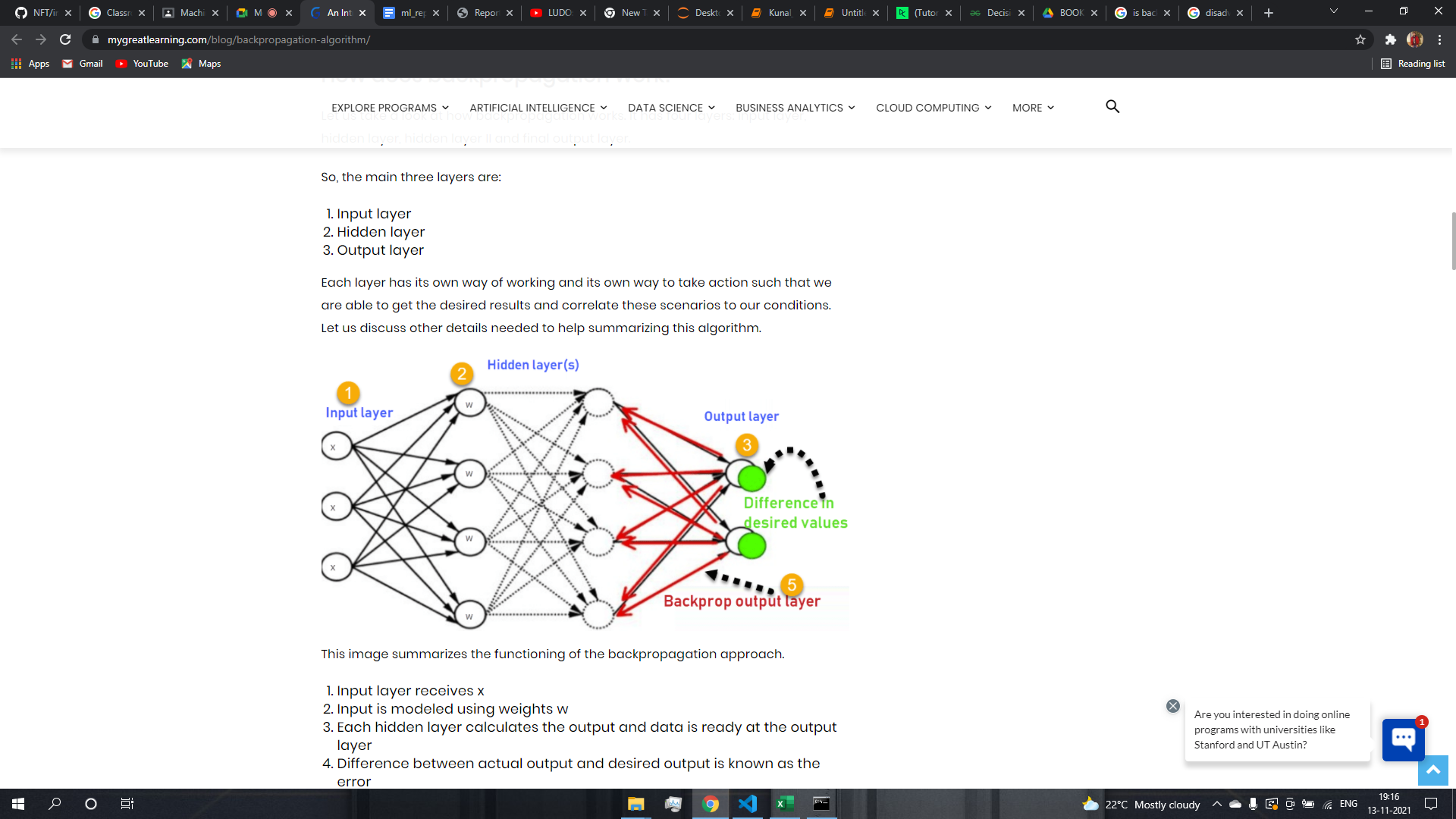
****

## **Candidate Algorithms:**

**Neural Network:** A neural network is a collection of connected units. Each connection has a weight associated with it. This system helps in building predictive models based on huge data sets.

* **Back Propagation** : It is an algorithm that trains some given feed-forward Neural Network for a given input pattern where the classifications are known to us. At the point when every passage of the example set is exhibited to the network, the network looks at its yield reaction to the example input pattern. After that, the comparison done between output response and expected output with the error value is measured. Later, we adjust the connection weight based upon the error value measured.

It basically has 3 types of layers : Input layer, Hidden layer and Output layer

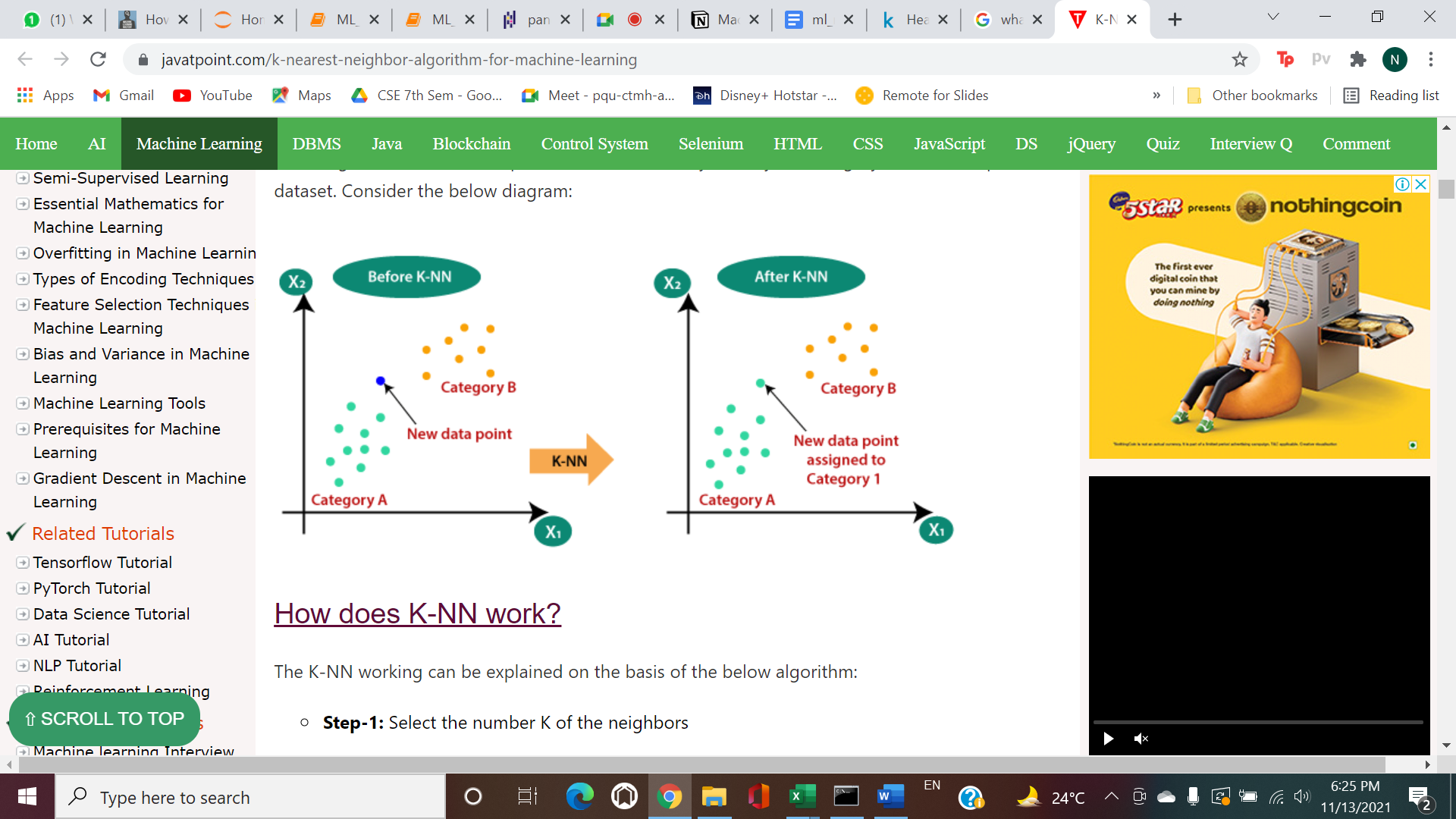


**K- Nearest neighbours(KNN):** It takes a bunch of labeled points and uses them to learn how to label other points.

Based on - “similar cases with same class labels are near each other”.

It is a method for classifying cases based on their similarity to other cases. Cases/data points that are near each other are said to be neighbors.

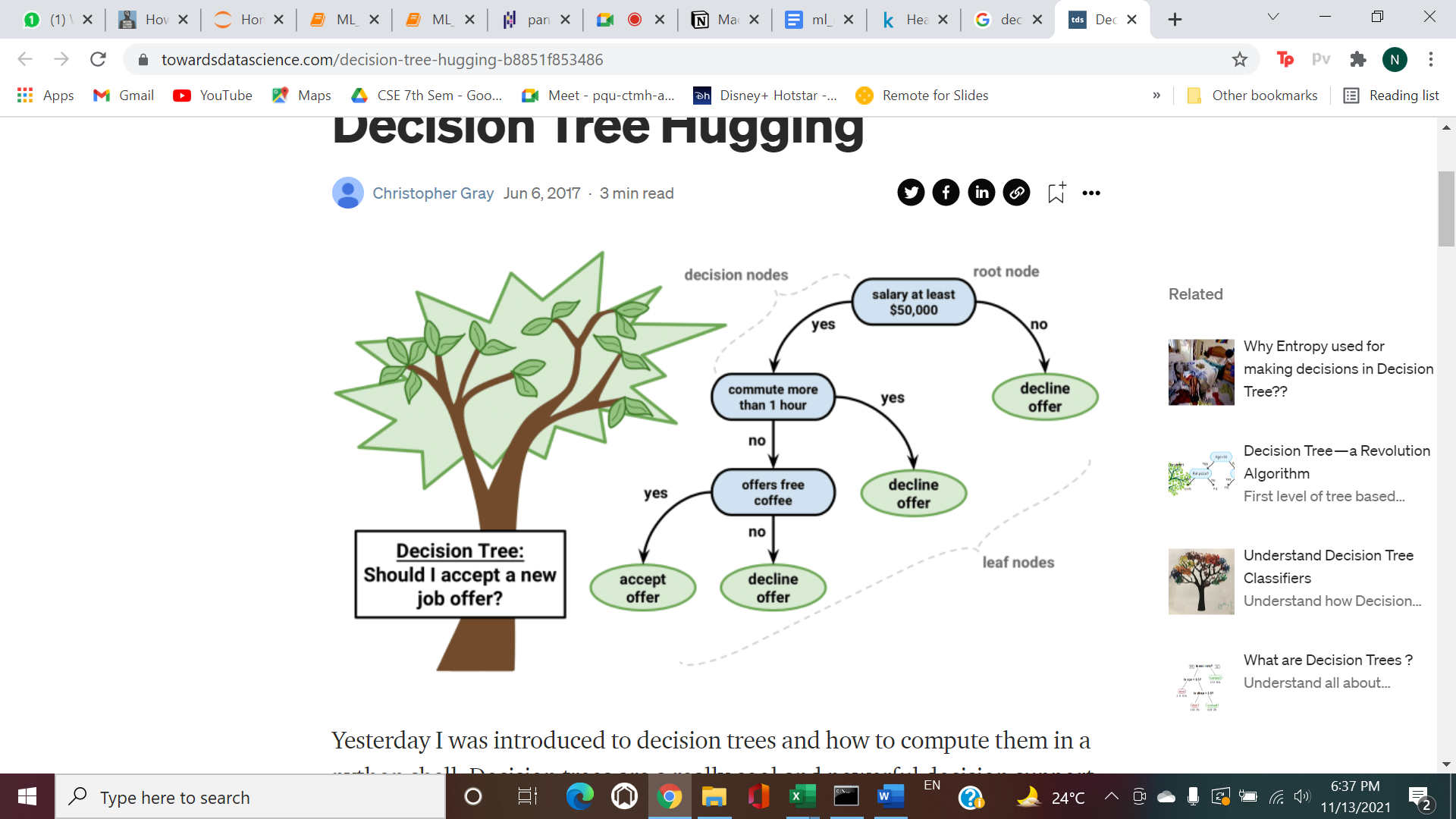
Thus, the distance between two cases is a measure of their dissimilarity.



**Decision Trees:** The basic intuition behind a decision tree is to map out all possible decision paths in the form of a tree.

Decision trees are built by splitting the training set into distinct nodes, where one node contains all of or most of one category of the data.

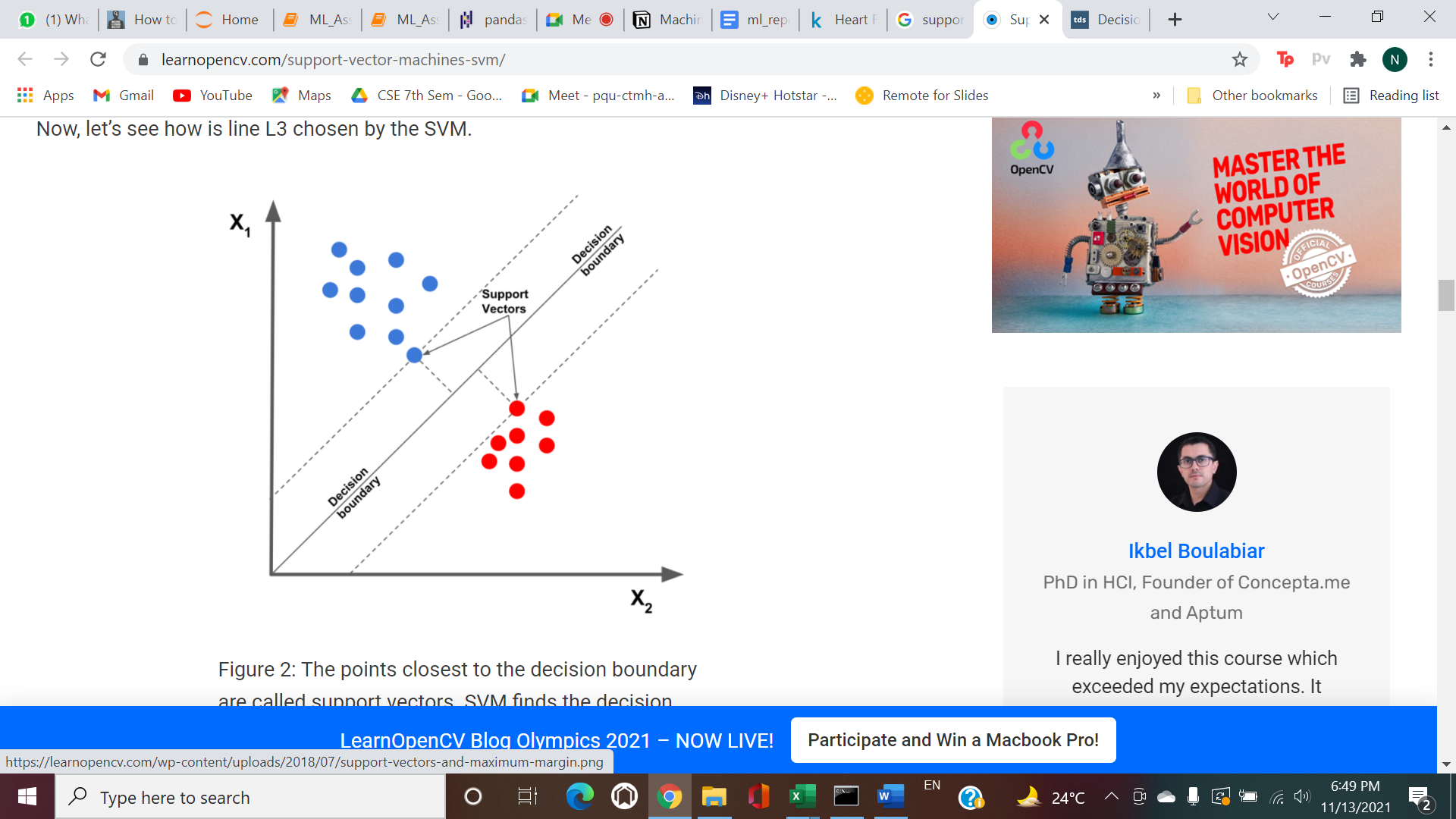
Decision trees are about testing an attribute and branching the cases based on the result of the test.



**Support Vector Machine(SVM):** SVM is a supervised algorithm that classifies cases by finding a separator.

* Mapping data to a high-dimensional feature space so that data points can be categorized, even when the data is not otherwise linearly separable.
* Finding a separator.

The data should be transformed in such a way that a separator could be drawn as a hyperplane.



**Chosen Algorithm : Back Propagation**

## **Why do we prefer Back Propagation ?**

* It is effective for training high dimensional and large data sets, and since our dataset is large(high-dimensional), this becomes more helpful.
* It has no parameters to tune apart from the numbers of input. It is a flexible method as it does not require prior knowledge about the network.
* The Back Propagation process error is measured by a very mature chain method, and its derivation process is rigorous and scientific which effectively trains the model and constantly adjusts the parameters of the neural network to achieve the most desired output.
* Nominal attributes are preferred in algorithms like decision trees, and our attributes are numeric so Back Propagation serves a better purpose for our dataset.

## **Libraries and APIs to implement Back Propagation:**

* PyTorch
* TensorFlow

### [Scikit-Neural Network](https://github.com/aigamedev/scikit-neuralnetwork)

* NeuroLab
* Keras
* ffnet

## **Code:**

We have implemented the code of Neural Network Back Propagation from scratch using the python language and attached the code file along with the Report.

**Libraries used :**

* Numpy
* Pandas
* Matplotlib

We used K-fold cross validation of our dataset to test our model.

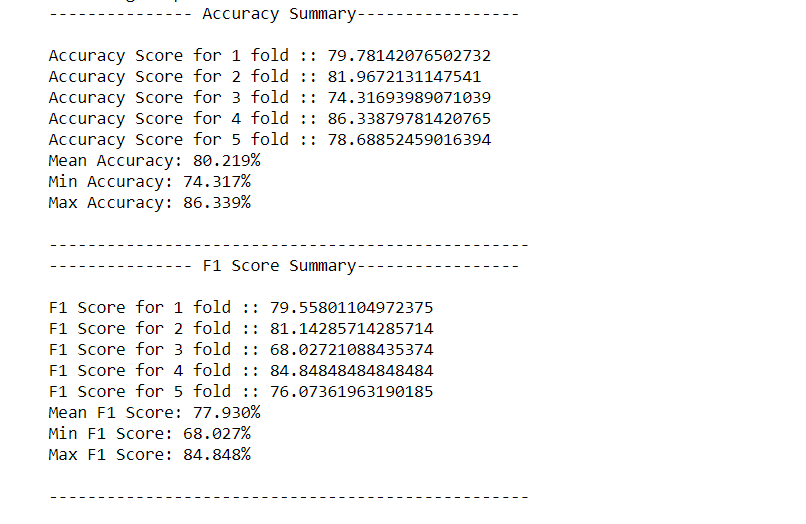
We also implemented the code for other 3 candidate algorithms mentioned above (KNN, SVM, Decision Tree) using **sklearn** library for comparing our model.

## **Performance measure:**

We used 2 different performance Metrics:

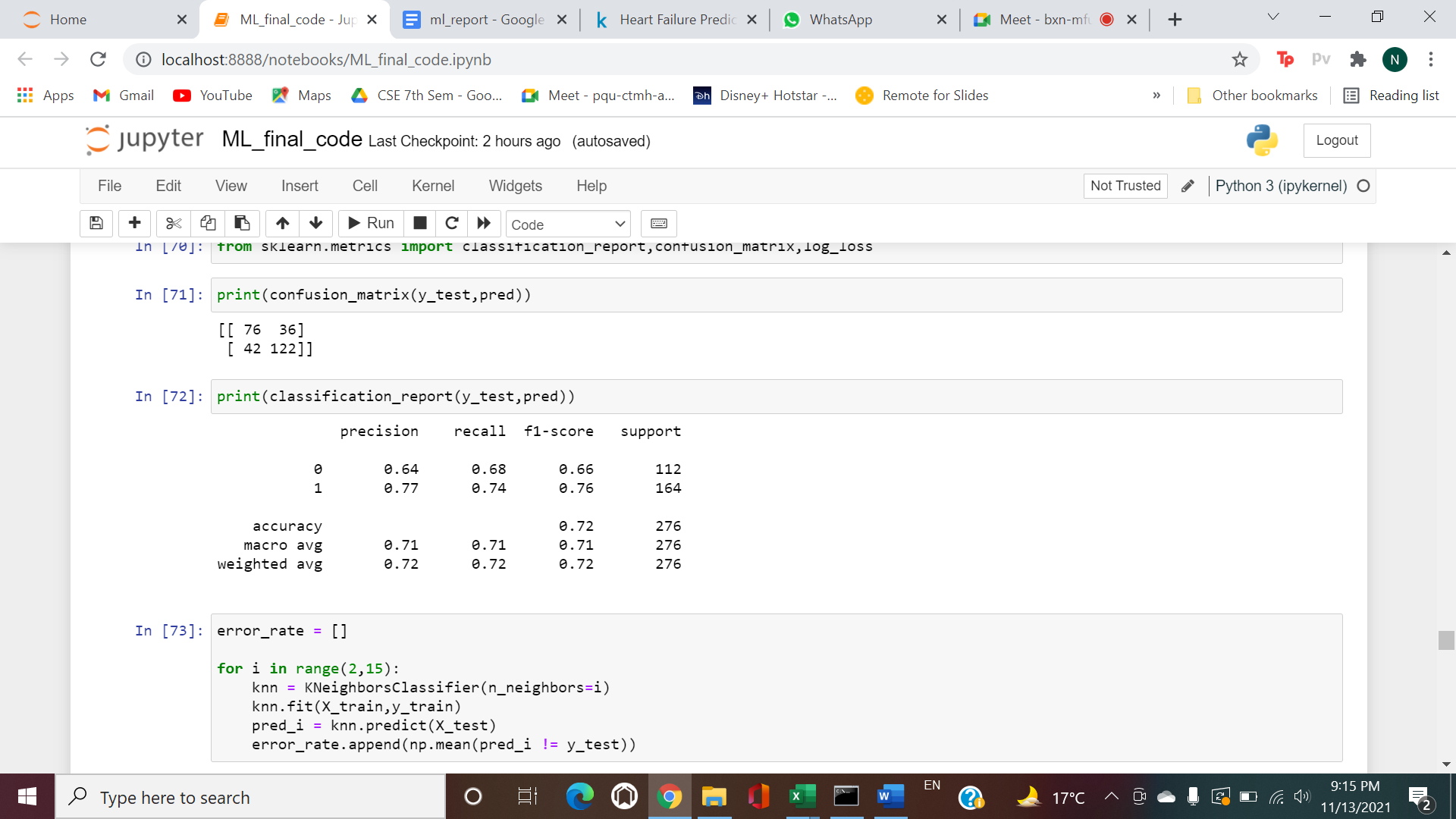
1. Accuracy: The degree to which the result of a measurement conforms to the correct value or a standard.
2. F1 score: It is defined as the harmonic mean between precision and recall. It is used as a statistical measure to rate performance.

**Accuracy and F1 Score of our model (Back Propagation NN)** :

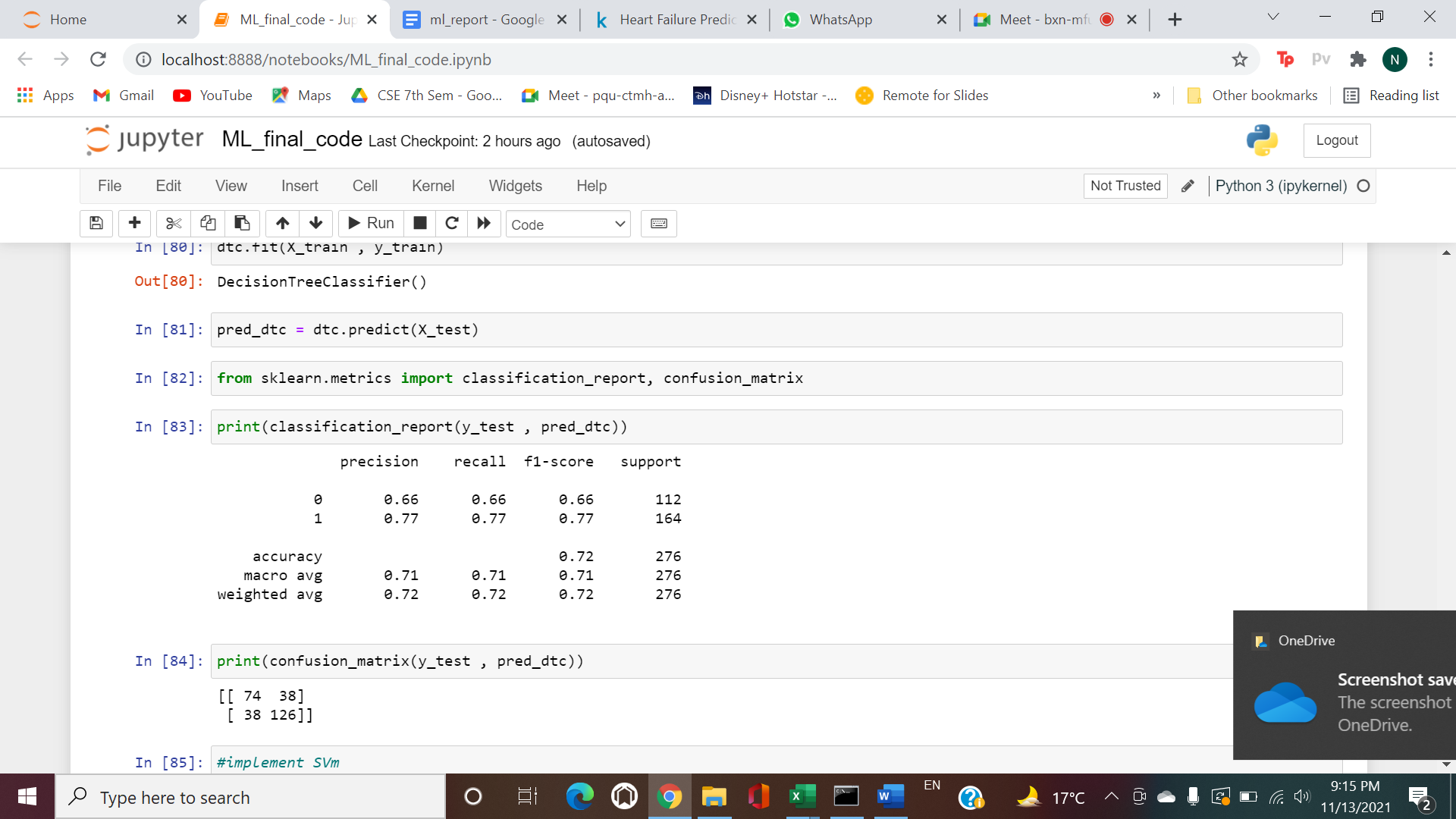
****

**Accuracy measure of candidate algorithms (using sklearn) :**

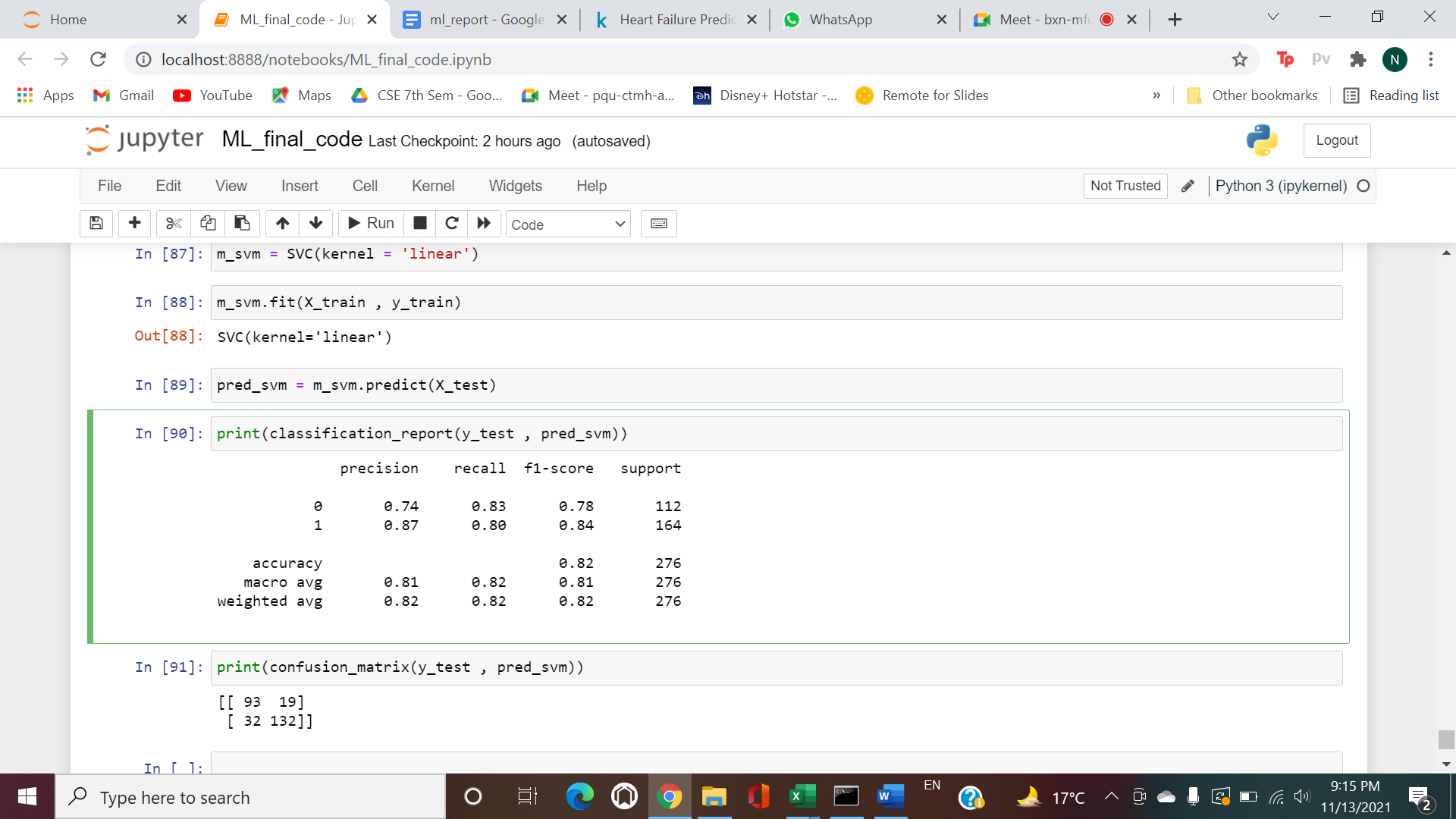
**KNN:**

****

**Decision Tree:**

****

**SVM:**

****

As we can see that the accuracy, we are getting in various candidate algorithms is either lesser (KNN, decision tree) compared to our model or comparable (SVM) to our model.

## **ADVANTAGES OF BACK PROPAGATION ALGORITHM**

* Backpropagation is fast, simple, and easy to implement
* There are no parameters to be tuned
* Prior knowledge about the network is not needed thus becoming a flexible method
* There is no need to specify the features of the function to be learned.
* Allows efficient computation of the gradient at each layer.
* BackPropagation searches the space of possible hypotheses using gradient descent to iteratively reduce the error in the network in order to fit to the training examples.
* It is able to adapt and learn independently.

## **DISADVANTAGES OF BACK PROPAGATION ALGORITHM**

* Sensitive to complex/noisy data which lead to inaccurate results.
* It needs the derivatives of activation functions for the network design time.
* Sensitive of the initial weights and bias.