SPRING SEMESTER

-

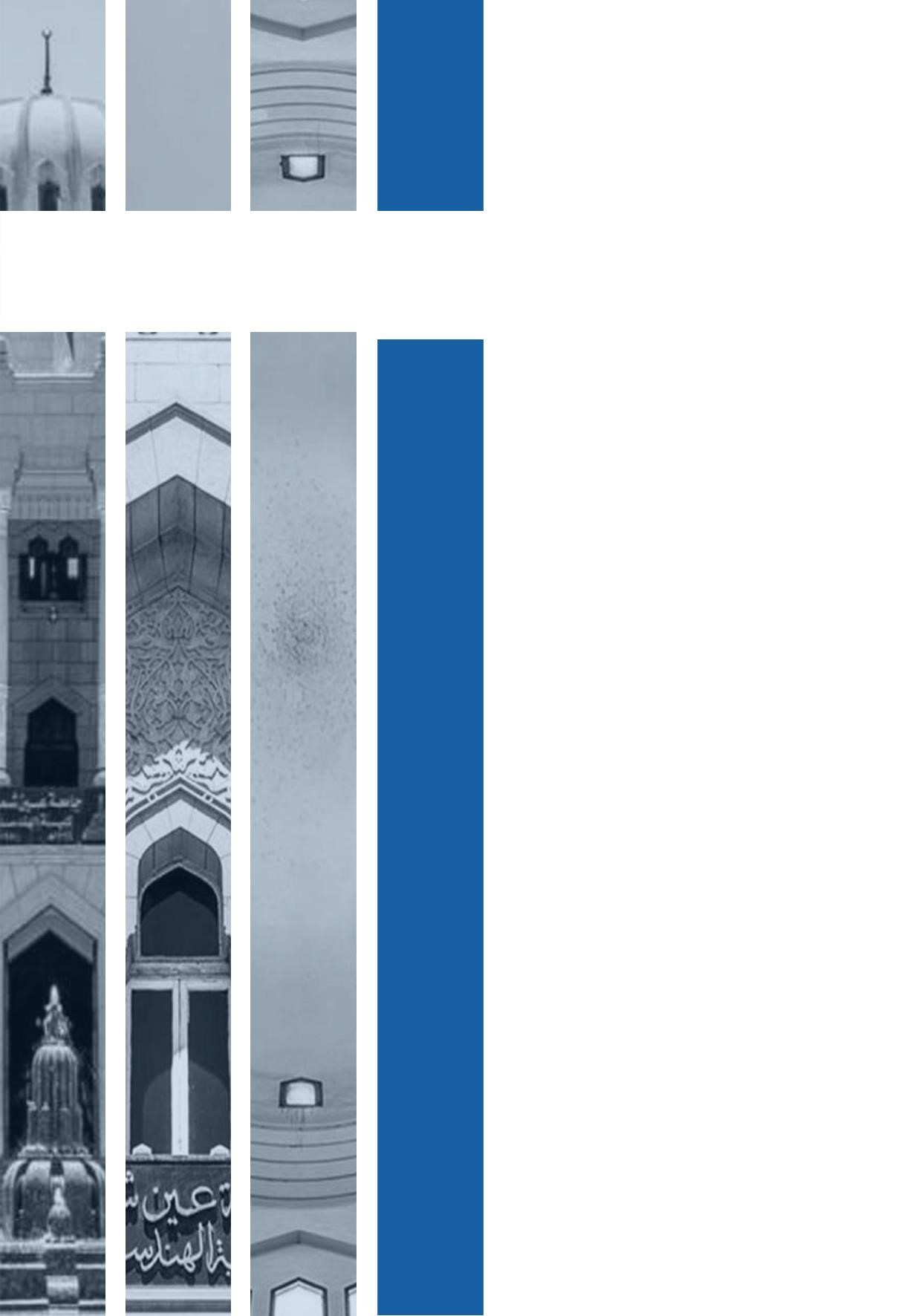
YEAR 201

9

/

20

20



RESEARCH & PROJECT SUBMISSION

S



**Program:**

***Course Code:***

***CSE323(UG2003)***

***Course Name:***

***Data Structure***

***Examination Committee***

**Prof.**

**Islam Ahmed Mahmoud**

**El**

**m**

**adah**

**Ain Shams University**

**Faculty of Engineering**

**Spring Semester**

**–**

**2020**

**Student Personal Information for Group Work**

# Plagiarism Statement

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electron ic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

**Student Name**

**s**

**:**

Manal Ahmed Mohamed

Nada Emam Kamal

Nermeen Osama Ramadan

Nesma Mohamed Atef

Mariam Abdelrahman Mahmoud

**Student**

**Codes**

**:**

1601449

1601558

1601575

1601579

1601374

**Signature/Student Name: Nesma Mohamed Date: 31/5/2020**

# Submission Contents

**01:** **Background 02:** **Implementation details 03: complexity of operations 04: references and video**

|  |
| --- |
| **Background** |

# 01

## First Topic

**What is machine learning?**

Machine learning is a field that is growing really fast, it’s a field of study whose main concern is the design and analysis of algorithms which allow the computers to learn. with much more awaiting to be discovered than is currently known, nowadays we are using machine learning to teach computers to perform a very wide array of useful operations. This includes operations like the automatic detection of objects in images (self-driving cars), speech recognition (which powers voice command technology), knowledge discovery in the medical sciences (it improves our understanding of complex diseases), and predictive analytics (economic forecasting). In this report we are going to give a high level introduction to a specific field of machine learning which is “The Classification”.

**What is Classification?**

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

|  |
| --- |
| **Implementation details** |

# 02

## Second Topic

**What our project talks about:**

We are required to implement a model that classifies some reviews of a hotel (is it positive or negative?) based on a dataset that contains some features and a text review.

In this classifier model we are going to implement a decision tree to classify whether this review is positive or negative.

**What is the decision tree?**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record’s attribute. On the bases of comparison, we follow the branch corresponding to that value and jump to the next node.

**We are provided with three files:**

1. Train data file: the hotel reviews that we should use to build the decision tree.
2. Validation data file: the hotel reviews that we don’t use to build the tree but we use it to evaluate the already-built tree. we will use it to check how the decision tree works for reviews that it didn’t see before.
3. Test data file: the hotel reviews that we don’t know their ratings.

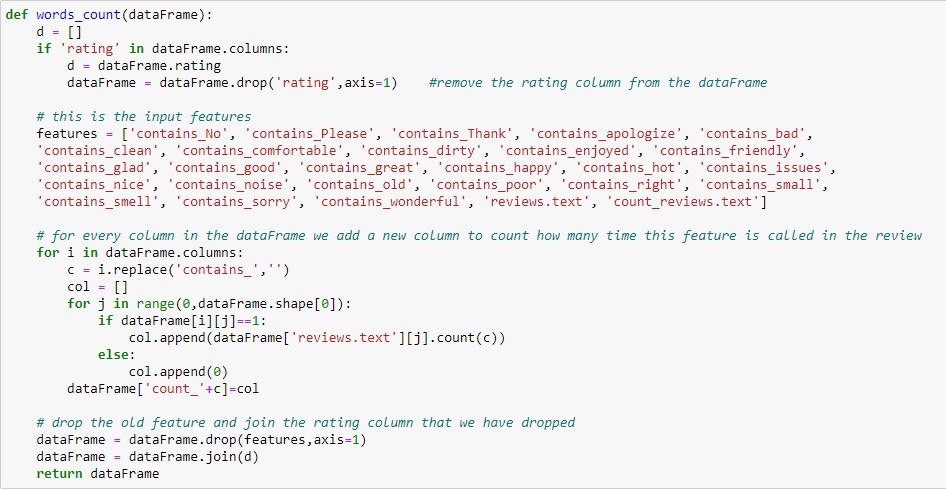
**Before building the tree we are going to prepare our dataset**

1. We use the function data\_init(file) that takes the file from the user and convert it to a data frame



SPRING SEMESTER - YEAR 2019/2020

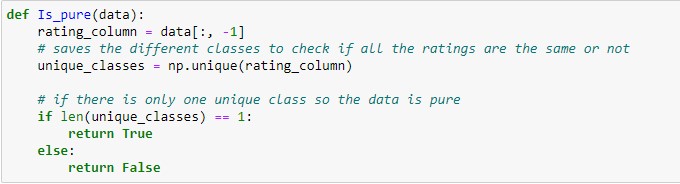
1. Instead of building the decision tree based on a categorical data we will count how many times every feature is called in the text review and use it to build our tree ,now our tree is based on continuous data, we will do this step by using the function words\_count(dataFrame) this function takes the data frame and return the data frame with a new columns (each column represent the number of counts of each feature)



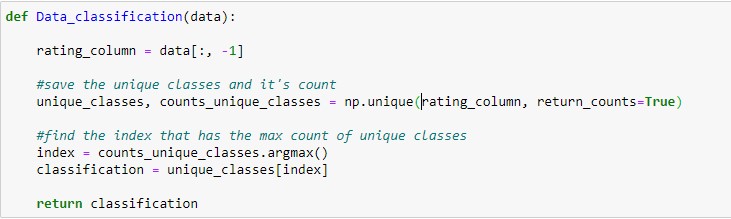
**The steps of building our decision tree:**

Our decision tree is based on some functions

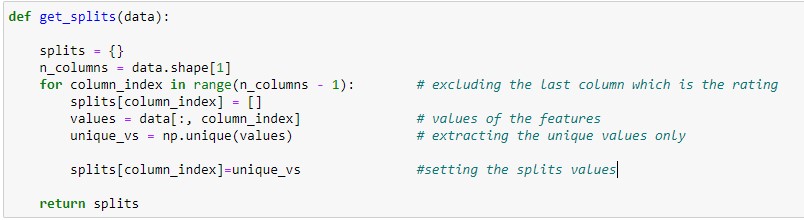
1. “Is\_pure(data) “ this function takes the train data and check if the data has the same rating (positive or negative) and return true or false



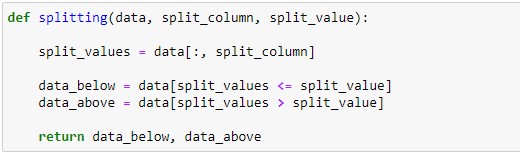
1. “Data\_classification(data) “ this function takes the data and return the rating of the majority class



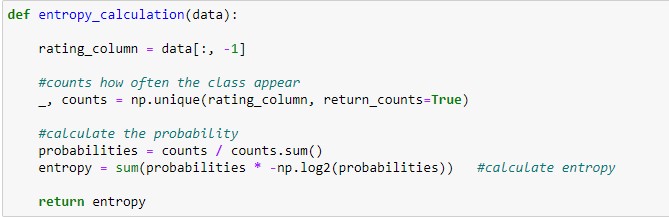
1. “get\_splits(data)” this function takes the data and return all the points that we can split on



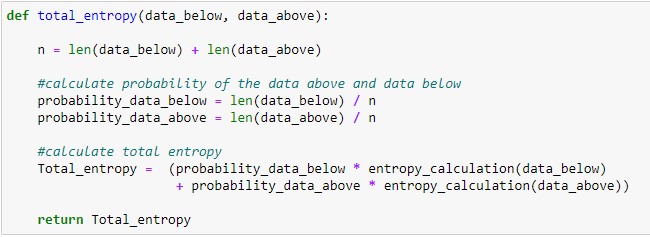
1. “splitting(data, split\_column, split\_value)” this function takes which column I chose to split on and the value of this point ,the function returns the data below the split line and the data above the split line.



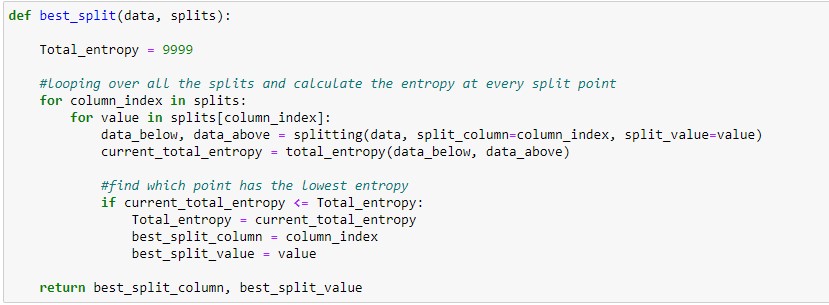
1. “entropy\_calculation(data):” this function takes the data and calculate the entropy of this data . first we calculate the probability of this data and then use it to calculate the entropy using the equation “entropy = sum(probabilities \* -np.log2(probabilities))”



1. “total\_entropy(data\_below, data\_above) “ this function takes the data above and data below “the output of the function “splitting(data, split\_column, split\_value)” and calculate the overall entropy by calculating the probability of the data below and data above and then calculate the overall entropy by the equation “total\_entropy = (probability of data below \* entropy of data below + probability of data above\* entropy of data above )” .. we calculate the entropy of the data above and data below by just calling the function “entropy\_calculation(data):”

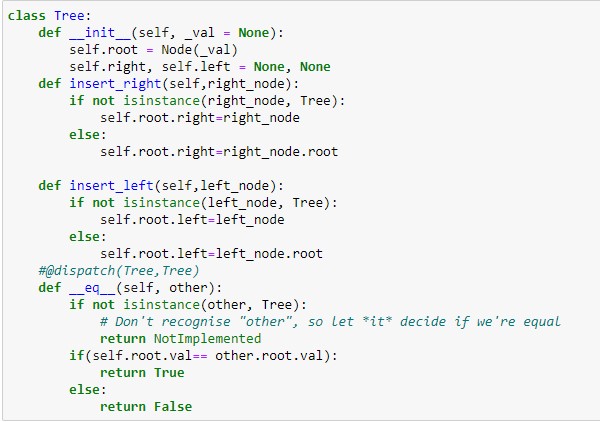


1. “best\_split(data, splits):” this function takes the data and the splits “the points that we can split on” and return the best point we can split on (its column and its value)

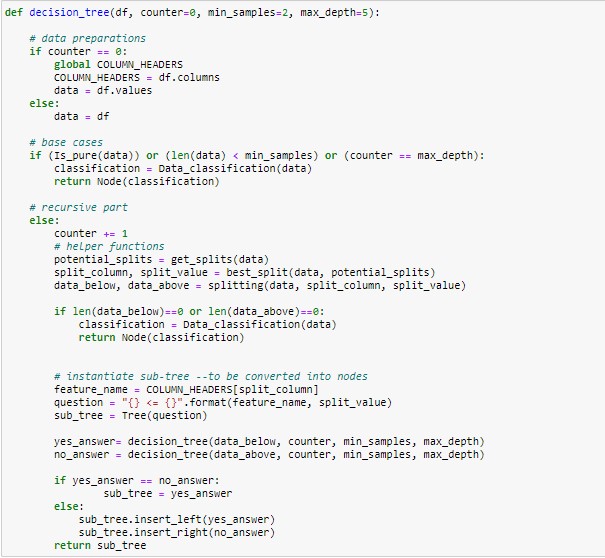


1. At the beginning we built a class node and class tree to help us in building the decision tree

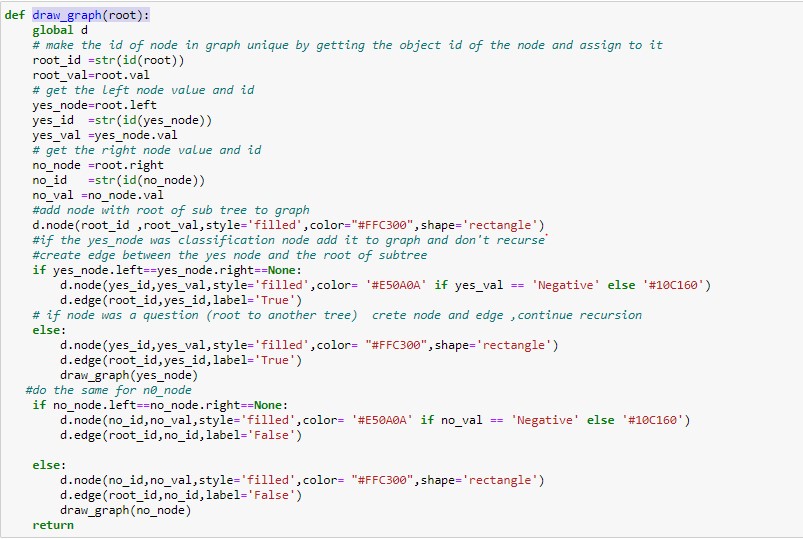




“ decision\_tree(df, counter=0, min\_samples=2, max\_depth=5)”: this function is divided into two main parts , the base case and the recursive part .. the input data will enter the recursive part until it reaches the leaf of the tree then it enters the base part to return the classification

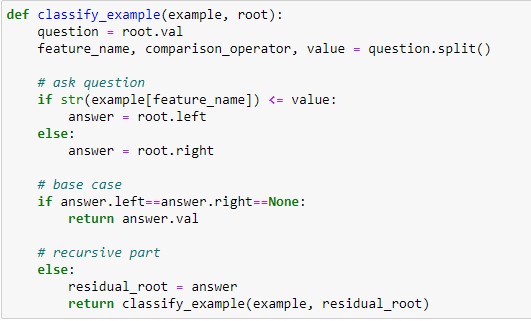


1. “draw\_graph(root):” we built the tree but it’s shown unclearly so we want to show it as a real tree so we used this function ,it takes the root of the tree and draws the graph



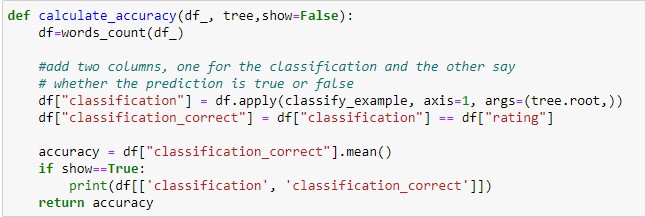
**We officially built our decision tree, now we are going to build the function that will traverse the tree to specify whether the review is positive or negative:**

This function is “classify\_example(example, tree):” takes a line of review and return whether this review is positive or negative.

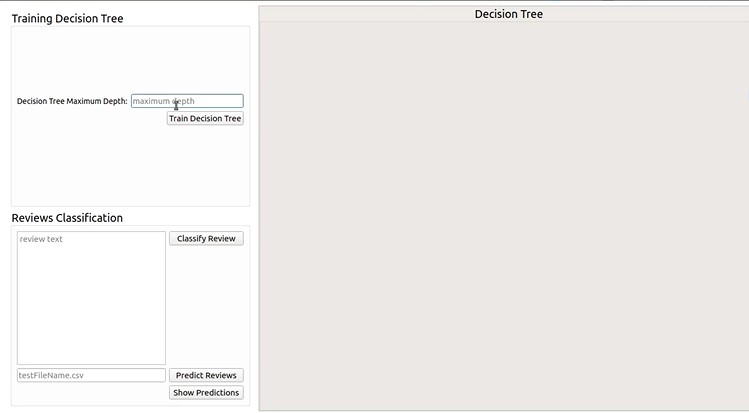


**Now we have reached the point that we want to know the accuracy of our decision tree:**

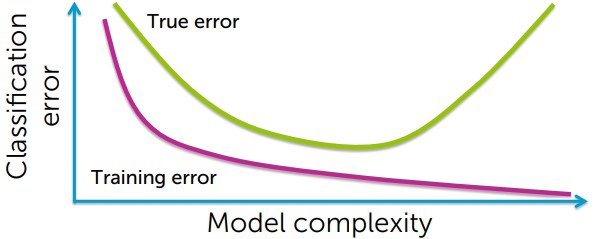
We call the function calculate\_accuracy(df\_, tree,show=False): this function takes the data and the tree and it loops on the whole lines of the reviews and compare the classification I had predicted with the rating of its review .



**Finally, we have built our decision tree using the train data and we test the accuracy based on the dev data. now we want to interface with the user to allow the user to input either a test data csv file or to input a new text review. We designed a GUI to simplify the interfacing between the user and the program**



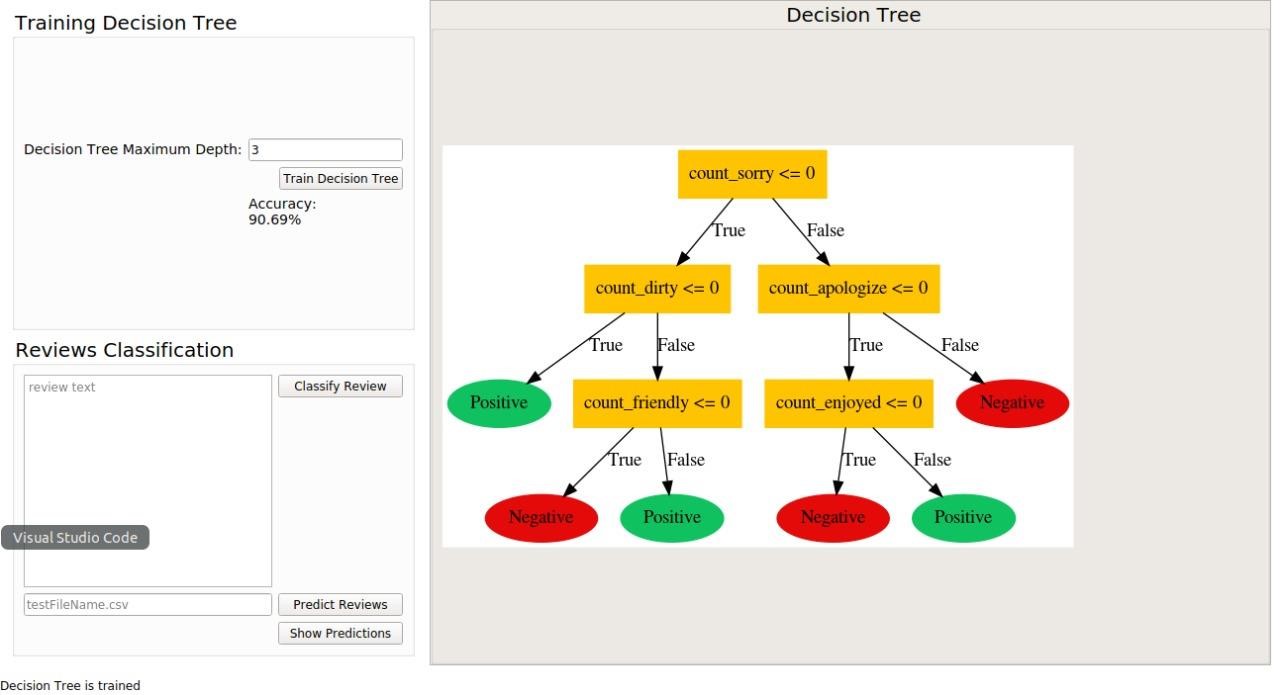
In our model the accuracy depends on something is called the Maximum depth of the tree, the maximum depth is described as the length of the longest path from the tree root to a leaf. When the maximum depth increases the accuracy increases but in a certain depth the overfit is occurred. over-fitting is the phenomenon in which the learning system tightly fits the given training data so much that it would be inaccurate in predicting the outcomes of the untrained data.



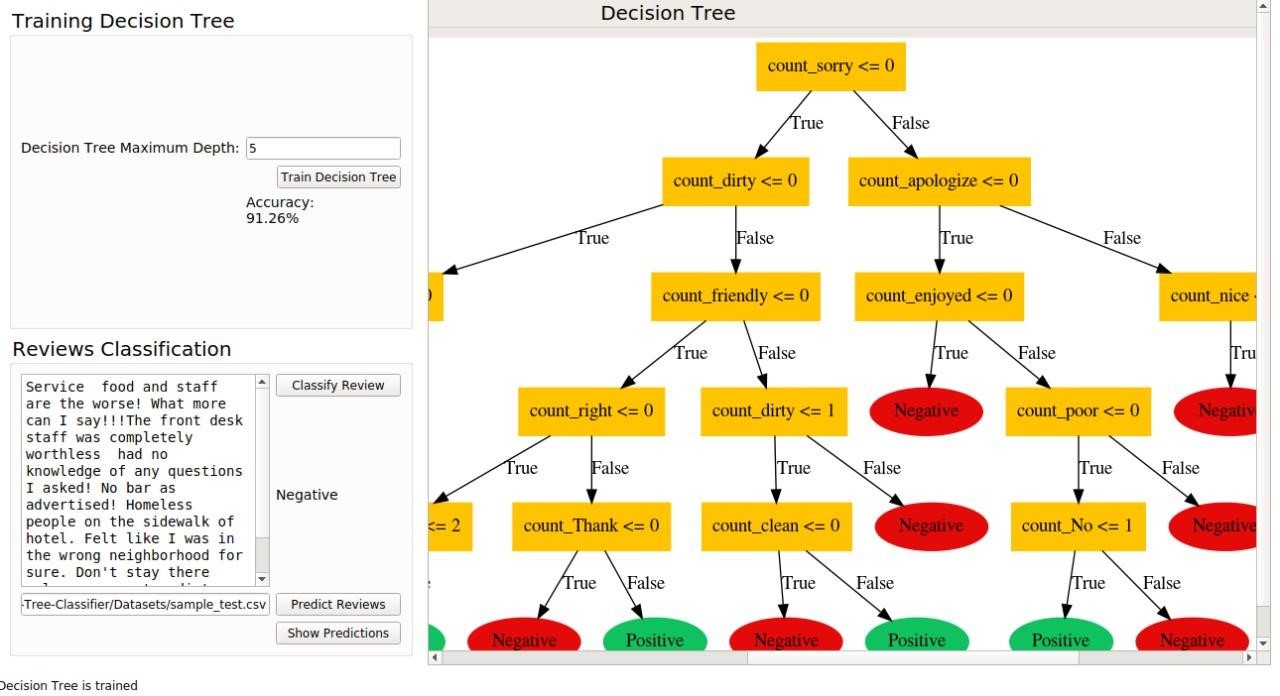
### This graph shows the overfitting when we try to decrease the error of training data

The user can enter the maximum depth to train the decision and calculate the accuracy ,when the user types the max depth, a message “Decision Tree is ready to be trained”, a message “Decision Tree is being trained” appears, and after the tree is shown a message “Decision tree is trained” appears to indicate that the process has been executed and the accuracy is shown here’s an example of max depth =3

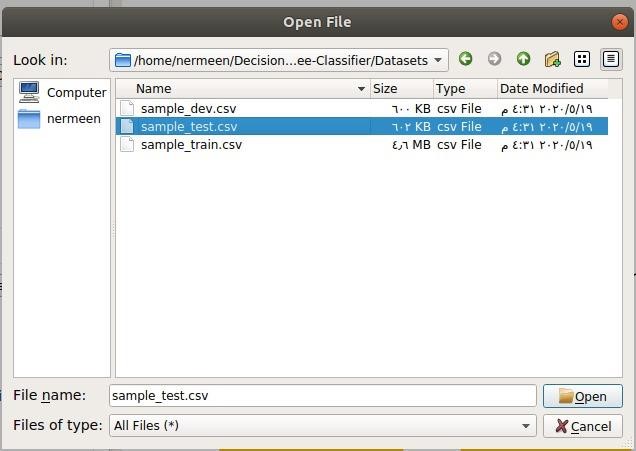
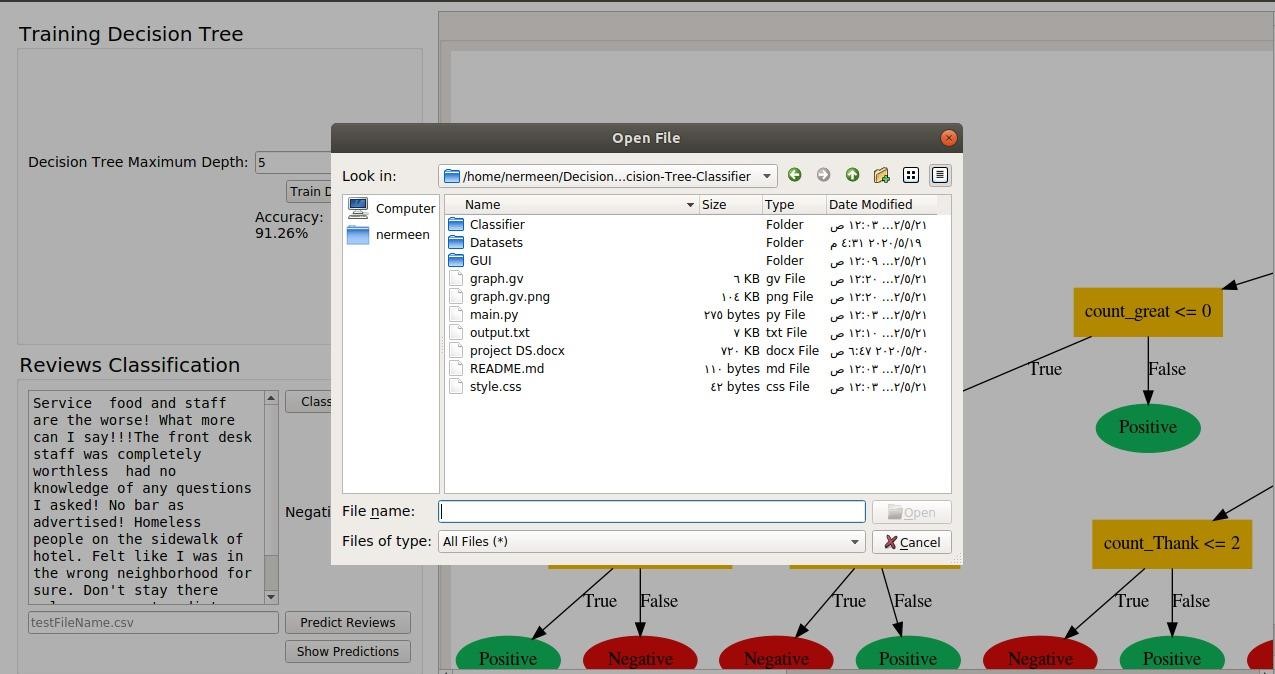




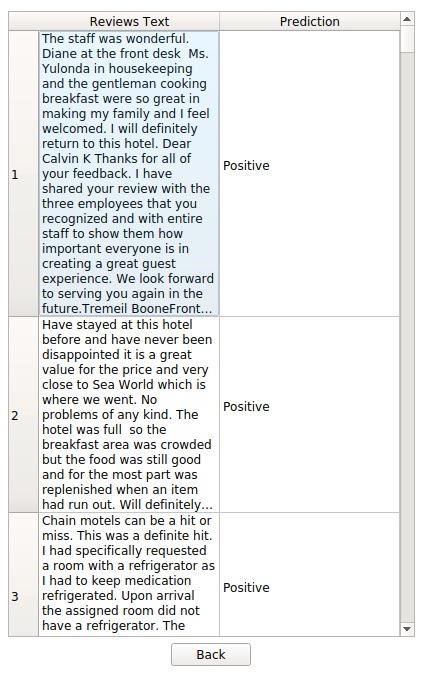
When we change the max depth to be 5: we will see that the accuracy increases



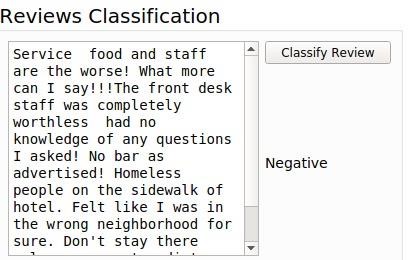
1. The input is test data csv file : the output will be a table contains every review and its prediction, here’s an example



When we press show predictions :



1. The input is new text review : the output will be the prediction of this review (is it positive or negative)



# z03

## Third Topic

**The complexity of the used functions in the project:** data\_init(file) , Is\_pure(data) , Data\_classification(data) , → O(1)

|  |
| --- |
| Complexity of operations |

words\_count(dataFrame) → O(c\*r).. c is the number of columns , r is the number of rows get\_splits(data) → O(n) .. n is the number of columns splitting(data, split\_column, split\_value) → O(1) entropy\_calculation(data) → O(1) total\_entropy(data\_below, data\_above) → O(1) best\_split(data, splits) → O(n^2) .. n is the number of splits

All Functions of class Node → O(1) All Functions Of class Tree → O(1)

decision\_tree(df, counter=0, min\_samples=2, max\_depth=5) → O(2^n) .. n is the max depth + 1 classify\_example(example, root) → O(d) .. d is the max depth of the tree.

calculate\_accuracy(df\_, tree,show=False) → O(c\*r).. c is the number of columns , r is the number of rows

draw\_graph(root) → O(2^n) .. n is the max depth + 1

# 04

## Fourth Topic

|  |
| --- |
| **Resources and Video** |

**resources** :

1. Machine Learning Refined : [http://www.r-5.org/files/books/computers/algolist/statistics/data-mining/Jeremy\_Watt-Machine\_Learning\_Refined-EN.pdf](http://www.r-5.org/files/books/computers/algo-list/statistics/data-mining/Jeremy_Watt-Machine_Learning_Refined-EN.pdf)
2. Machine Learning: Classification Course by University of Washington :

<https://www.coursera.org/learn/ml-classification>

**Link of github** : <https://github.com/alien19/Decision-Tree-Classifier>

**Link of the Video** : <https://www.youtube.com/watch?v=9pPcerApEJk&t=10s>