

# **ANALYZE THE DIFFICULTY LEVEL OF A PARTICULAR COURSE**

*FINAL REPORT*

*Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Project Video: <https://youtu.be/QJ3vqljITas>**

*By*

**Registration No.**

11701711

**Name**

Subhadip Mondal

**Roll No.**

RKMO28A13

Under the guidance of

**Usha Mittal**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

MAY 2020

## **ACKNOWLEDGMENT**

---

Place: - Lovely Professional University

Date: - 8<sup>th</sup> April, 2020

I would like to thank Usha Mittal for assigning us with this project. Through the project I am able to grasp more technical and have a hands-on practical experience with python and some machine learning algorithms. Through it I am able to learn how a project is created and how necessary and crucial technical knowledge is. I am really grateful to the faculty that has provided me with the necessary guidelines.

**Registration No.**

11701711

**Name**

Subhadip Mondal

## **DECLARATION STATEMENT**

---

Place: - Lovely Professional University

Date: - 8<sup>th</sup> April, 2020

This is to declare that this report has been written by me. No part of the report is copied from the other sources. All information included from other sources has been duly acknowledged. I aver that if any part of the report is found to be copied, I will take full responsibility for it.

Name : Subhadip Mondal

Reg No. : 11701711

Signature : .....

# **TABLE OF CONTENTS**

---

## **1. INTRODUCTION**

- 1.1. Introduction
- 1.2. Uses
- 1.3. Special Button

## **2. METHODOLOGY**

- 2.1. Requirements
- 2.2. Functionality
- 2.3. Implementation
- 2.4. Debugging
- 2.5. Execution
- 2.6. Working

## **3. TECHNOLOGY**

- 3.1. Python 3.7
- 3.2. Machine Learning
- 3.3. Clustering
- 3.4. Pandas
- 3.5. Numpy
- 3.6. Matplotlib
- 3.7. Sklearn
- 3.8. GUI
- 3.9. IDE
- 3.10. Tkinter
- 3.11. Widgets classes
- 3.12. Option with values

## **4. REFERENCE**

# CHAPTER 1

## INTRODUCTION

---

### 1.1. Introduction

I had performed clustering on dataset using Python, tkinter and Sklearn. I had used machine learning techniques to perform this task we will be doing clustering on a dataset.

### 1.2. Uses

Model created will help the user to predict the difficulty level (Easy or Medium or Difficult) of their given data.

All the button used in this project perform some special kind of operations.

- Predict the difficulty level of user input data
- Know about developer
- Go to user interface

### 1.3. Special Button

Sentiment Analysis on Roman Urdu Dataset project uses some special buttons.

All the button used in this program are

- KNOW ABOUT DEVELOPER
- CLICK TO TAKE USER INPUT
- PREDICT THE DIFFICULTY LEVEL

## CHAPTER 2

### METHODOLOGY

---

#### 2.1. Requirements

First step is gathering all the information required of project. Learn all the technology that maybe used in the project. The functions and the modules are the necessary for the development of the project.

#### 2.2. Functionality

In this step, we will discuss the different function used in the program and how they are working.

The different types of function are used are...

- `user_input()`
- `developer()`
- `predict_difficulty()`

Now we will discuss working of each of the function.

##### **`user_input()`**

This function is defined in the program named “about.py”

Code we used is...

```
def user_input():  
    master.destroy()  
    ticket_obj = user.UserInput()
```

This function help us to move from “about.py” to “user.py” program.

##### **`developer()`**

This function is defined in the program named “user.py”

Code we used is...

```
def developer(self):  
    import os  
    os.system("python about.py")
```

This function gives us all information about the developers those who made this project.

### **predict\_difficulty ()**

This function is defined in the program named “user.py”

Code we used is...

```
def predict_difficulty(self):
    from tkinter import messagebox
    try:
        import pickle
        km = pickle.load(open( "km.sav", "rb" ))
        # fetch the Entered user input data
        user_input = [self.CA_100.get(), self.MTT_50.get(), self.ETT_100.get(), self
.ETP_100.get(), self.Course_Att.get(), self.CA_1.get(), self.CA_2.get(), self.CA_3.get()
, self.CA_4.get()]
        # Predict from the user input data
        y_p = km.predict([user_input])
        level = FALSE
        if y_p == 0:
            lavel = 'Difficult'
        elif y_p == 1:
            lavel = 'Easy'
        else:
            lavel = 'Medium'
        messagebox.showinfo("Model Predicted Difficulty Level", "Course Name : "+sel
f.Course_Name.get()+"\n"+" Difficulty Level : "+lavel)
    except:
        messagebox.showerror("Error", "Please Enter Valid Input!!!")
```

This function predict the difficulty level of the course.

## 2.3. Implementation

This step is to implementing the project by coding. This is most important part of the program. A particular care should be done in order to get error free program.

## 2.4. Debugging

This step is to check the project for any errors in the code. And linking of the module take place after checking individual modules and functions.

## 2.5. Execution

This step is to executing the project and simple testing.

## 2.6. Working

When we run the about.py file this interface will come



On clicking the "Click to enter User Input" Button . Then "user.py" file will run and display this output like



**Predict Difficulty Level**

## Predict the Difficulty Level of Course

Course Name :

CA\_1 Marks (0-100) :

CA\_2 Marks (0-100) :

CA\_3 Marks (0-100) :

CA\_4 Marks (0-100) :

Course Attendance (0-100) :

CA\_100 Marks (0-100) :

MTT\_50 Marks (0-50) :

ETT Marks (0-100) :

ETP Marks (0-100) :

[Click To Predict Difficulty Level](#)
[Click To Know About Developer](#)

Then we will fill the all details and click the “Click To Predict Difficulty Level “ then a new popup will come and show us the difficulty level of the course. A demo is given below

## Predict the Difficulty Level of Course

Course Name :

CA\_1 Marks (0-100) :

CA\_2 Marks (0-100) :

CA\_3 Marks (0-100) :

CA\_4 Marks (0-100) :

Course Attendance (0-100) :

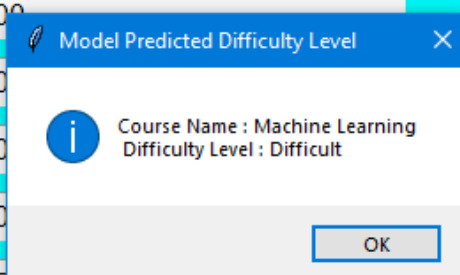
CA\_100 Marks (0-100) :

MTT\_50 Marks (0-50) :

ETT Marks (0-100) :

ETP Marks (0-100) :

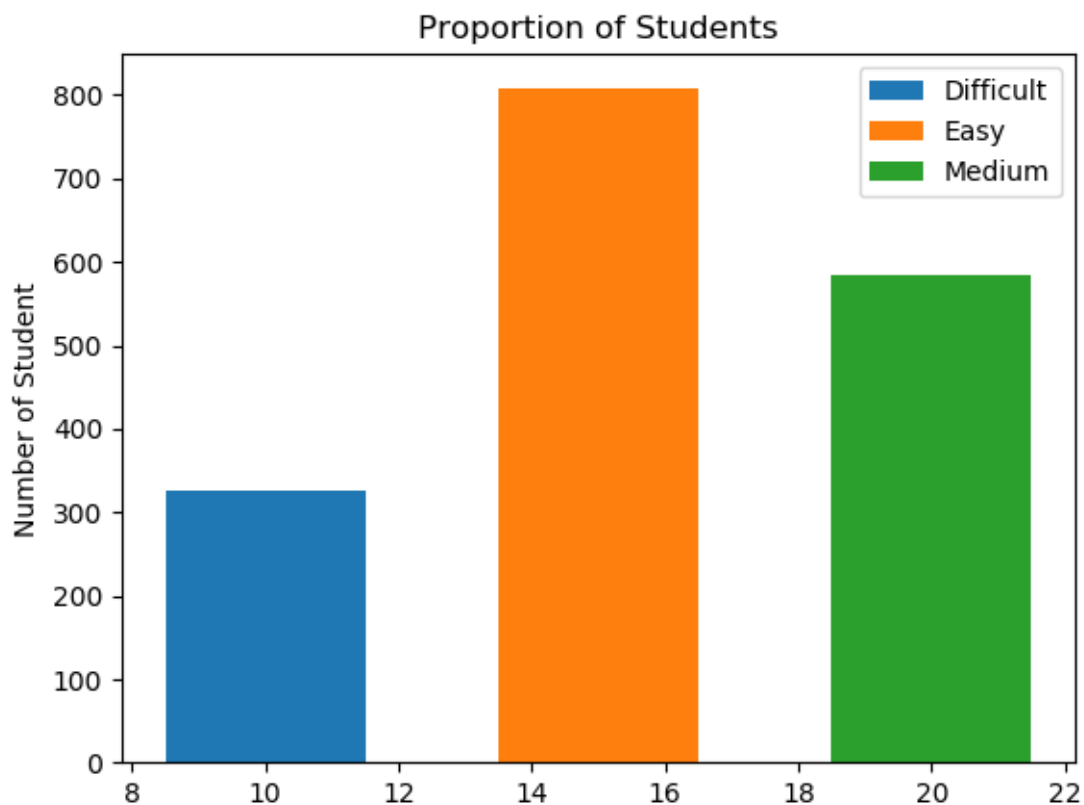
[Click To Predict Difficulty Level](#)
[Click To Know About Developer](#)



Like this our program work. On clicking the “Click To Know About Developer” button we

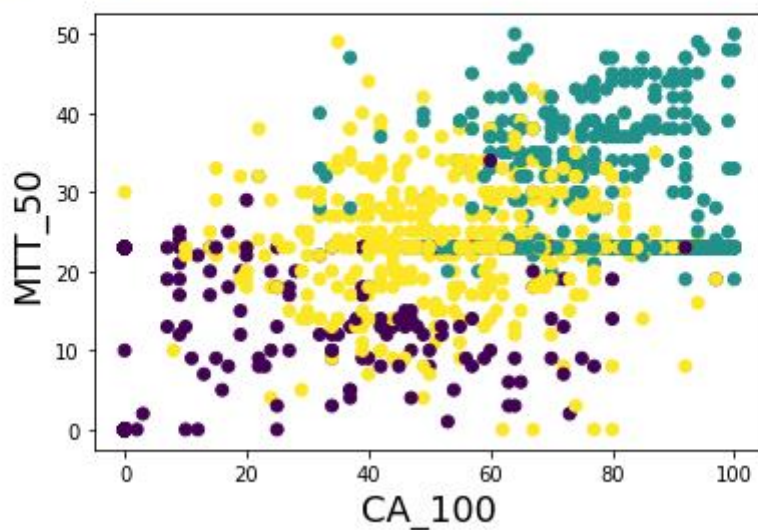
redirect to the “about.py” program.

Below graph show us difficulty level of course with the number of student.

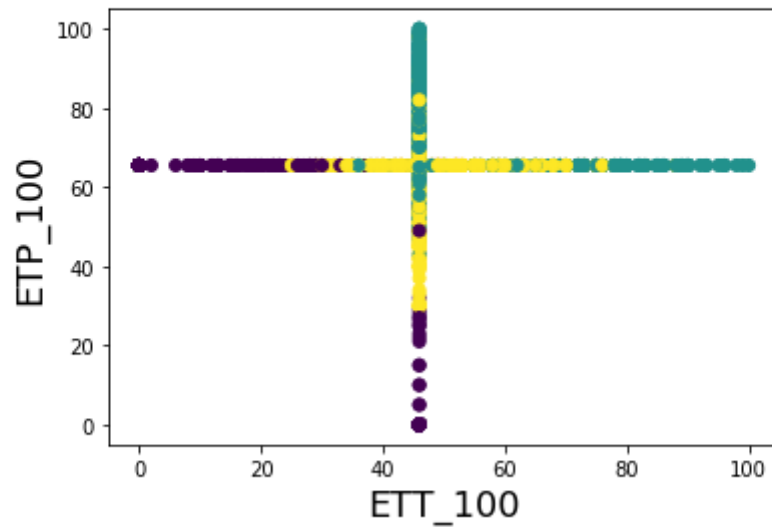


The dependencies between the features are shown in the below graph.

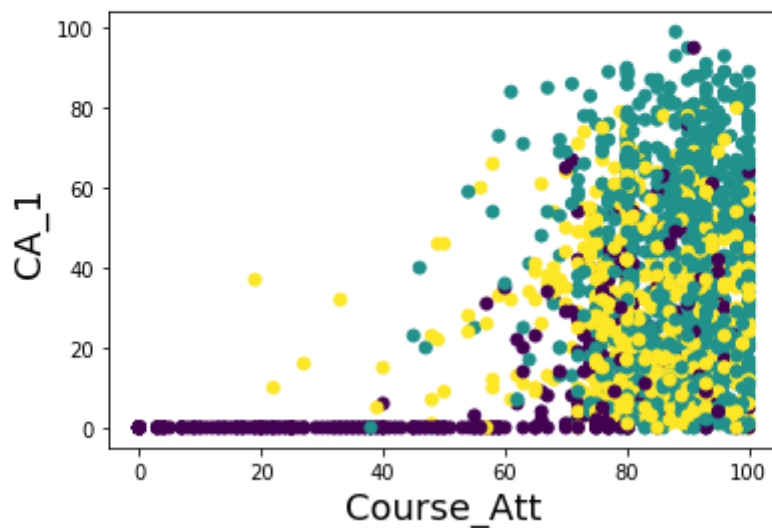
```
In [47]: plt.scatter(dataset[:, 0], dataset[:, 1], c=dataset[:, -1])
plt.xlabel('CA_100', fontsize=18)
plt.ylabel('MTT_50', fontsize=18)
plt.show()
```



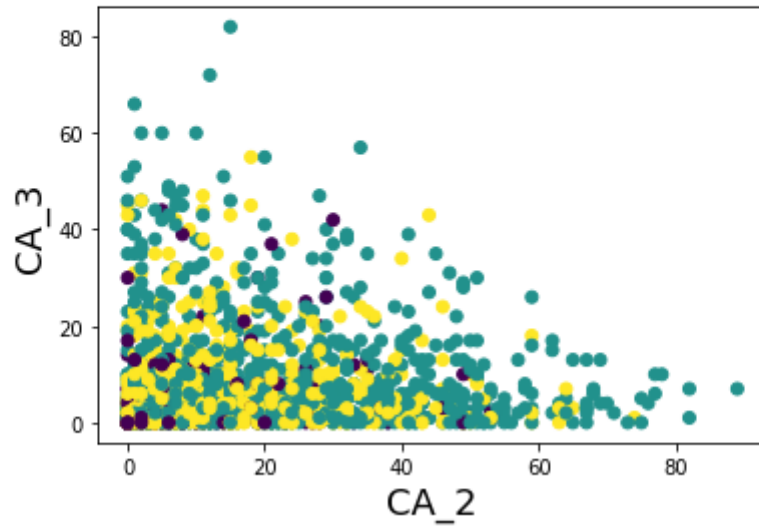
```
In [48]: plt.scatter(dataset[:, 2], dataset[:, 3], c=dataset[:, -1])
plt.xlabel('ETT_100', fontsize=18)
plt.ylabel('ETP_100', fontsize=18)
plt.show()
```



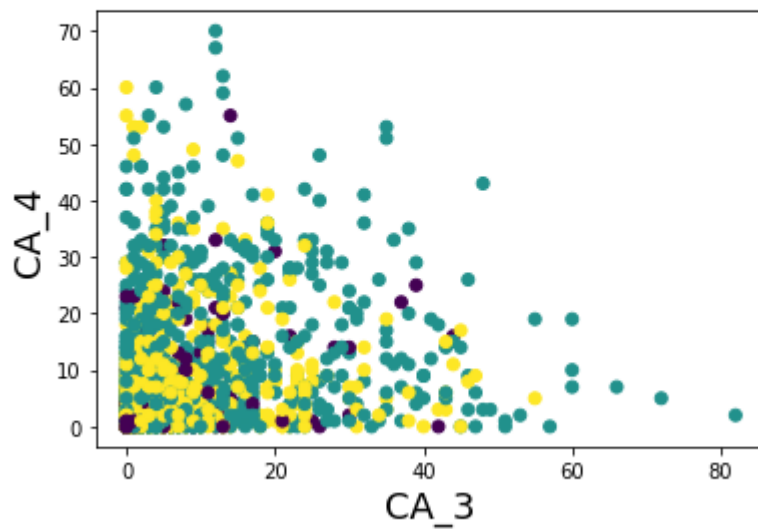
```
In [49]: plt.scatter(dataset[:, 4], dataset[:, 5], c=dataset[:, -1])
plt.xlabel('Course_Att', fontsize=18)
plt.ylabel('CA_1', fontsize=18)
plt.show()
```



```
In [50]: plt.scatter(dataset[:, 6], dataset[:, 7], c=dataset[:, -1])
plt.xlabel('CA_2', fontsize=18)
plt.ylabel('CA_3', fontsize=18)
plt.show()
```



```
In [51]: plt.scatter(dataset[:, 7], dataset[:, 8], c=dataset[:, -1])
plt.xlabel('CA_3', fontsize=18)
plt.ylabel('CA_4', fontsize=18)
plt.show()
```



## **CHAPTER 3**

### **TECHNOLOGY**

---

#### **3.1. Python 3.7**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

#### **3.2. Machine Learning**

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning.

#### **3.3. Clustering**

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

#### **3.4. Pandas**

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

#### **3.5. Numpy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

### **3.6. Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for

embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

### **3.7. Sklearn**

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines and others.

### **3.8. GUI**

Short for Graphical User Interface, a GUI (pronounced as either G-U-I or gooey) allows the use of icons or other visual indicators to interact with electronic devices, rather than using only text via the command line.

A GUI uses icons, and menus to carry out commands, such as opening, deleting, and moving files.

### **3.9. IDE**

Integrated development environment (IDE) used is Visual Studio. It provides code analysis, a graphical debugger, an integrated unit tester etc.

### **3.10. Tkinter**

Python offers multiple options for developing GUI. Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.

### **3.11. Widgets classes**

Tkinter's GUI classes define common GUI widgets such as button, labels, check buttons, frames, canvases and other. We will use some of them in our project.

### **3.12. Option with values**

Widgets classes contains option with their values to change the interface of the widgets classes means appearance of the window. Some of the option are width, height, fg, bg, side, row, column, text, onvalue, offvalue, variable, expand, fill, file, image, command and so on.

## CHAPTER 4

### REFERENCES

---

I went through different website and learn those concept that will used in making this project.

Some of the website are...

- <https://www.google.com/>
- <https://www.python.org/>
- <https://numpy.org/>
- <https://pandas.pydata.org/>
- <https://matplotlib.org/>
- <https://www.nltk.org/>
- <https://scikit-learn.org/>
- I followed the text book.