CHAPTER 1

INTRODUCTION

Currency or Piece of Paper is an application developed to determine whether the given banknotes are counterfeit or legitimate based on the data set put together. This project is implemented using concepts of data science and machine learning algorithms in real time and analyzing and implementing the same using Python language.

1.1 COURSE OBJECTIVES

- Understand the technological and/or societal needs
- Apply the knowledge of various Python fundamentals.
- Design and develop an algorithm for the given problem statement.
- Analyze and evaluate the algorithm performance metrics.
- Create a software solution for real time application using one or more Python fundamentals.
- Demonstrating our communication skill effectively with a technical presentation.

1.2 PROBLEM DEFINITION

Developing an application that will determine whether the given banknotes are counterfeit or legitimate based on the data set put together. This project is implemented using concepts of data science and machine learning algorithms in real time and analyzing and implementing the same using Python language.

1.3 OUTCOMES OF THE PROJECT

Analyze and understand the technological needs to support tools and develop an application which will determine the authenticity of the given banknotes (currency). This project will establish if there exists a correlation between the attributes and plots graphs depicting the same. The data model built will be optimized to achieve maximum accuracy.

CHAPTER 2
REQUIREMENT SPECIFICATIONS

2.1 HARDWARE REQUIREMENTS

- 1 GB RAM (2 GB+ recommended)
- 9-58 GB of free hard disk space
- Basic GPU
- Intel Pentium or compatible processor 1.6 GHz minimum (2GHz+ recommended)
- 1024x768 or higher resolution monitor

2.2 SOFTWARE REQUIREMENTS

- Operating System
- Python 3.8.3
- Jupyter Notebook (anaconda 3)

CHAPTER 3

FUNDAMENTALS

3.1 PYTHON FUNDAMENTALS

Python is an easy to learn, powerful, high-level programming language. It has efficient data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, makes it an ideal language for scripting and rapid application development in many areas on most platforms.

3.1.1 Data Structures

List – List is a data structure that holds an ordered collection of items i.e. you can store a sequence of items in a list. List is a mutable data structure. The values in a list are enclosed in square brackets [].

Tuple – A Tuple is a collection of comma separated values enclosed within a parenthesis (). It is similar to a list and a major difference is that a tuple is a mutable data structure.

Set – Sets are unordered collections of simple objects. These are used when the existence of an object in a collection is more important than the order or how many times it occurs. Sets are immutable data structures and consists of unique elements. Elements are enclosed within curly braces {}.

Dictionary – A dictionary is like an address-book where you can find the address or contact details of a person by knowing only his/her name i.e. we associate keys (name) with values (details). The keys are unique values. Dictionary is an immutable data structure.

Example: fruits = dict {1 : ”apple”, 2 : “banana”, 3 : “cherry” }

Here, 1, 2, 3 are the keys and “apple”, “banana”, “kiwi” are the values associated with the respective keys.

3.1.2 Functions

Functions are reusable pieces of programs. They allow us to give a name to a block of statements, allowing us to run that block of code using the specified name anywhere in your program, any number of times. Functions are defined using the "def" keyword. After this keyword comes an identifier which is the name for the function, followed by a pair of parentheses which may enclose some parameters. This is followed by the final colon that ends the line. Next follows the block of statements that are part of this function. These statements are written with an indentation to specify they belong to that function. The function can return a value and that is the last statement in the function block.

3.1.3 Programming Paradigms

Python is a multi-paradigm programming language. It supports different programming approaches. One of the popular approaches to solve a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP).

Objects in Python are basically an encapsulation of Python variables and functions, that they get from classes. A class is simply a logical entity that behaves as a prototype or a template to create objects, while an object is just a collection of variables and Python functions. Variables and functions are defined inside the class and are accessed using objects. These variables and functions are collectively known as attributes. Classes provide an easy way of keeping the data members and methods together in one place which helps in keeping the program more organized. Using classes also provides another functionality of this object-oriented programming paradigm, that is, inheritance.

The functional programming paradigm treats program computation as the evaluation of mathematical functions based on lambda calculus. Lambda calculus is a formal system in mathematical logic for expressing computation based on function abstraction and application using variable binding and substitution. Functional programming is handy for parallel processing and is super-efficient for tasks requiring recursion and concurrent execution.

3.1.4 Graphic User Interface (GUI)

Python provides various options for developing graphical user interfaces (GUIs). Some of the most common ones are Tkinter, wxPython, JPython.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. A GUI application using Tkinter can be created by performing the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more widgets to the GUI application as required.
- Enter the main event loop to take action against each event triggered by the user.

3.1.5 Data Visualization

In the trending technological world, there is a massive amount of data that is being consumed, as well as wasted. Thus, handling this data in a rather effective manner becomes the main goal of Data Science. We can make use of various programming languages to deal with the data sets that require operations, like calculating statistics, sales, marketing, plotting on graphical platforms, etc., to be done on them. Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations required in plotting of graphs and diagrams

3.2 ALGORITHM FUNDAMENTALS

An algorithm is a set of rules that specify the order and kind of arithmetic operations that are used on a specified set of data. An algorithm is an effective method for solving a problem using a finite sequence of instructions.

An algorithm is finite set of instructions that is followed, accomplishes a particular task. In addition, all algorithms must satisfy the following criteria:

- Input - Zero or more quantities are externally supplied to the algorithm.
- Output - At least one quantity is produced by the algorithm.
- Definiteness - Each instruction is clear and produced.
- Finiteness - If we trace out the instruction of an algorithm, then for all cases, the algorithm terminates after a finite number of steps.
- Effectiveness - Every instruction must be very basic so that it can be carried out, in principal, by a person using only pencil and paper. It is not enough that each operation be definite, it also must be feasible.

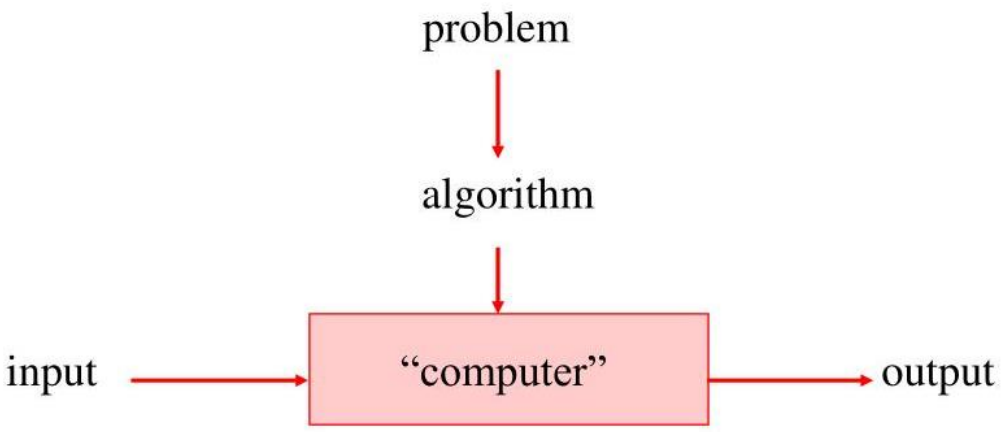

Fig 3.1: Notion of an algorithm

3.3 MACHINE LEARNING FUNDAMENTALS

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Machine Learning is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine Learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine Learning can be classified into three categories -

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

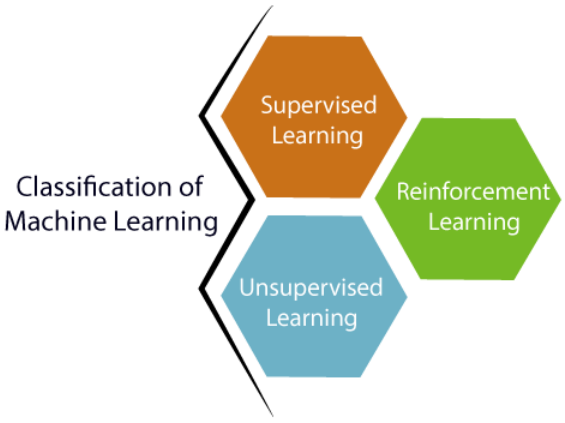

Fig 3.2: Classification of Machine Learning.

3.3.1 Supervised Learning

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

Can be grouped into two categories -

- Regression
- Classification

3.3.2 Unsupervised Learning

Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

Can be grouped into two categories -

- Clustering
- Outlier Detection/ Association

3.3.3 Reinforced Learning

Reinforcement Learning is when the algorithm has a feedback system involved. It accepts positive or negative feedback from the user and sends it back to the algorithm to improve its accuracy.

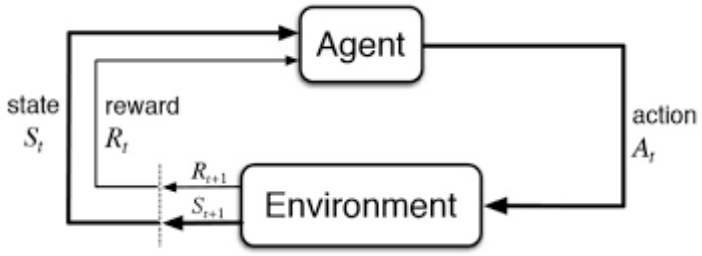

Fig 3.3: Reinforcement Learning

CHAPTER 4
DESIGN AND ALGORITHM

An algorithm is a well-defined procedure that allows a computer to solve a problem. Another way to describe an algorithm is a sequence of unambiguous instructions. The use of the term 'unambiguous' indicates that there is no room for subjective interpretation. Every time you ask your computer to carry out the same algorithm, it will do it in exactly the same manner with the exact same result.

4.1 DESIGN GOALS

This project is designed using the concept of machine learning algorithms (KNN analyzer) in Python language. Some of the operations the functions in this program are designed to do are:

- Asking the user to either select the Image Analysis option or the Data Model Analysis option.
- In the Image Analysis option, the user is asked to enter the entire path of the image he/she wants to be analyzed.
- After entering the image path, the image is processed for further operations.
- The processed image is then used to calculate the variance, curtosis, skewness, and entropy of the image. The values for the same are displayed along with the image specified by the user.
- The values are then plotted on the graph along with the dataset and the algorithm predicts whether the currency is real or counterfeit.
- In the Data Model Analysis option, the user is presented with options to either see the graphs plotted with different values or see the KNN analyzer model.
- The data set is split into training and testing data to train and estimate the performance of the machine learning model.
- The accuracy of the data model is also calculated and it is optimized to achieve maximum efficiency.

4.2 ALGORITHM

STEP 1: Start the program.

STEP 2: Import and install all the Python libraries required in the program.

STEP 3: Create a graphic user interface window for the Home Page of the application with two buttons, one for Image Analysis and the other for Data Model Analysis.

STEP 4: Create functions for both, so the user is directed towards the corresponding option.

STEP 5: If the user chooses Image Analysis option, Go to Step 7.

STEP 6: If the user chooses Data Model Analysis option, Go to Step 21.

STEP 7: Create a graphical user interface window for the Image Analysis using the Tkinter class of Python.

STEP 8: Provide a text box for the user to enter the path of the image. The path entered must be the complete path.

STEP 9: If the path of the image exists, Go to Step 10. Else go to Step 8.

STEP 10: Once the path of the image has been verified the image is converted into grayscale image. This can be done using the formula –

Grayscale pixel value = (0.3 * R) + (0.59 * G) + (0.11 * B)

STEP 11: Once the colored image is converted into grayscale image, different values like Variance, Curtosis, Skewness and Entropy are calculated.

STEP 12: These values are displayed on a newly created graphical user interface window along with the image chosen by the user.

STEP 13: A predict class button is provided. If the user clicks on the button, Go to Step 14.

STEP 14: These values that are calculated are used to plot a point on the graph.

STEP 15: The distance from the rest of the points to the new point is calculated and stored in an array.

STEP 16: This array is sorted in ascending order.

STEP 17: A K value is chosen which is odd.

STEP 18: The classes of first K values is considered and counted.

STEP 19: The class of the new point on the graph is predicted to be the class which has the higher majority in the K points.

STEP 20: Display the predicted class of the new point.

STEP 21: Create a graphical user interface window for the Data Model Analysis using the Tkinter class of Python and provide the user with two options of – Data Table Value Analysis and KNN Analysis.

STEP 22: If the user chooses Data Table Value Analysis, Go to Step 24.

STEP 23: If the user chooses KNN Analysis, Go to Step 25.

STEP 24: Display the graph with all the points plotted that are present in the data set that are used to predict the class in the KNN Algorithm.

STEP 25: Display the percentages of the training and testing data that is used and predict the accuracy of the model against the testing set.

STEP 26: Exit from the program.

CHAPTER 5

IMPLEMENTATION

Each aspect of this application is divided into different modules which are then individually implemented. A detailed explanation for each of the module implementation is given below.

5.1 MODULE 1 IMPLEMENTATION

Before the implementation of the modules, all the required packages are imported and installed on the system and the version of python installed needs to be checked. In order to check the version, type the following command on the command prompt on the system –

```
python --version
```

To install a package on the system, the user needs to install it from the internet. In order to do so is to type the following command on the command prompt of the system along with the package name that you want to install.

```
pip install package_name
```

Once all the packages are imported and ready for use, the different aspects of the application are implemented.

When the user runs the application, he/she is provided with two main options. The Image Analysis option and the Data Model Analysis option. To implement this a graphical user interface is developed using the Tkinter class of Python. The window is kept running in a constant loop and closes only when the user explicitly does do. The user is directed towards the respective function based on the choice of the user.

5.2 MODULE 2 IMPLEMENTATION

If the user choses the Image Analysis option, he/she is directed towards a new window. The window consists of a text field wherein the user is asked to type out the path of the image that the algorithm needs to predict. The path of the image must be complete and valid. It should also contain the extension of the image type. Some of the accepted extensions for images are .png, .jpg, jpeg. If the path entered is incomplete or invalid, the program does not proceed further. An example of a complete path is –

C:\Users\arohi_000\Pictures\Example.jpeg

After the user enters the path and it is accepted by the program, the image is then cropped into 400x400 pixels so as to match with the rest of the values in the data set. The image is then converted into a Grayscale Image for easier calculations. This is done by using the formula –

Grayscale pixel value = (0.3 * R) + (0.59 * G) + (0.11 * B)

Where R is the value of red component, G is the value of green component, B is the value of blue component.

After obtaining the Grayscale Image, calculations are done on the data to obtain the variance, skewness, curtosis, and entropy of the image. The value obtained from these calculations is used to predict whether the currency is counterfeit or real. A point is plotted in the graph along with all the other values from the data set. A KNN analysis algorithm is run on the graph and the value obtained from the algorithm is predicted to be the class of the image. This class is displayed on a new window along with the image chosen by the user and the values of variance, skewness, curtosis and entropy of the image. The accuracy of this is based on the model built and the number of neighbors that were considered to calculate the class as the algorithm used to predict the class is K nearest neighbors' algorithm.

5.3 MODULE 3 IMPLEMENTATION

If the user choses, Data Model Analysis option, he/she is directed towards a new window. The window consists of two option: The Data Set Analysis and the KNN Analysis. The Data Set Analysis shows the graph that is used to predict the class of a point. To determine the axes that have a correlation with the final answer many graphs were drawn and analyzed. The Data Set imported had the following columns – variance, skewness, curtosis and entropy. At first, 2-D graphs were drawn. The dark blue points represent Class 0 that is the counterfeit currency, and the golden points represent the Class 1 points that is the real currency. The graphs were plotted using the matplotlib library. Scatter plots were drawn to find which columns have a correlation with the Class of the image and also find which of them have the least overlap of points in order to build our model. A few 2-dimensional graphs that were analyzed are shown below.

- The values of Curtosis and Variance from the data set were plotted taking variance on the x-axis and curtosis on the y-axis.

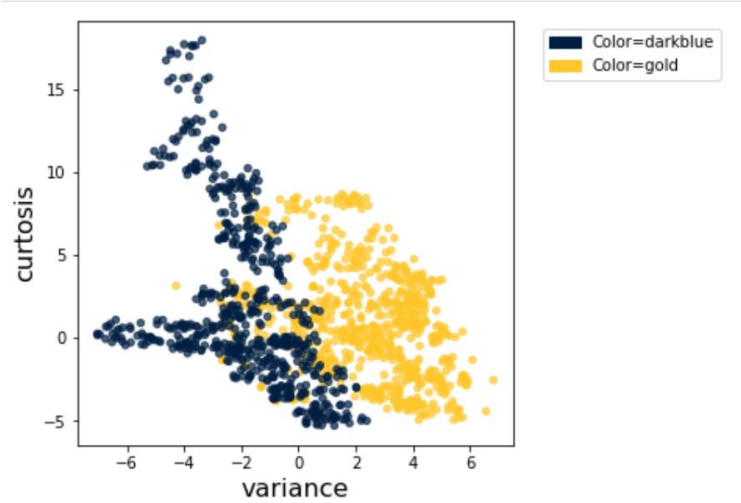

Fig 5.1: Curtosis vs Variance graph

- The values of Entropy and Curtosis from the data set were plotted taking curtosis on the x-axis and variance on the y-axis.

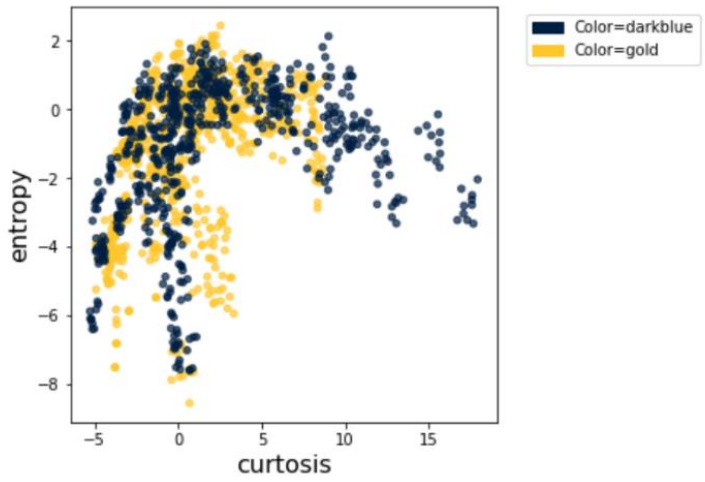

Fig 5.2: Entropy vs Curtosis graph

- The values of Entropy and Skewness from the data set were plotted taking skewness on the x-axis and entropy on the y-axis.

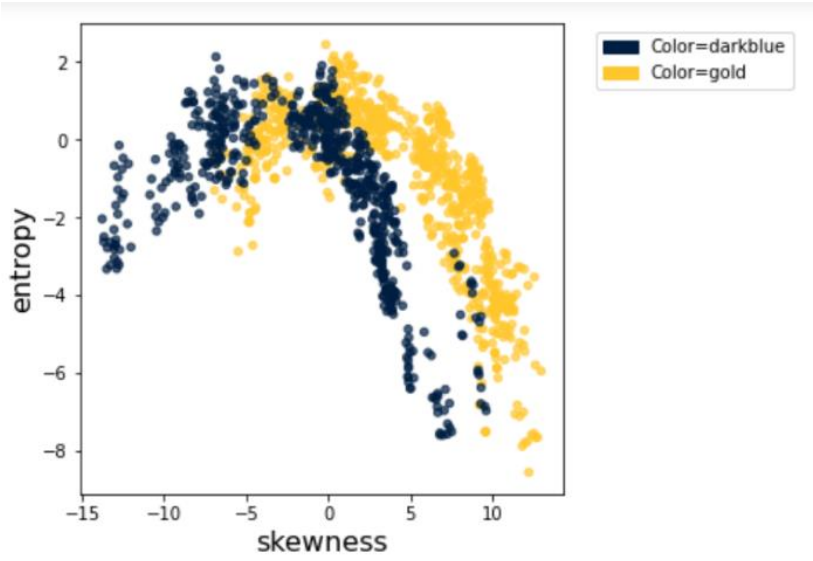

Fig 5.3: Entropy vs Skewness graph

As seen in the above plotted 2-D graphs the overlap between the golden cluster and the blue cluster is very high. Thus, no correlation can be found between the two classes as no clear distinction was seen. After analysing the 2 dimensional grphs, a few graphs were plotted considering three attributes froom the data set. The analysis from those graphs is mentioned below.

- The values of Variance, Skewness and Entropy from the data set were plotted taking Variance on the x-axis, Skewness on the y-axis and Entropy on the z-axis.

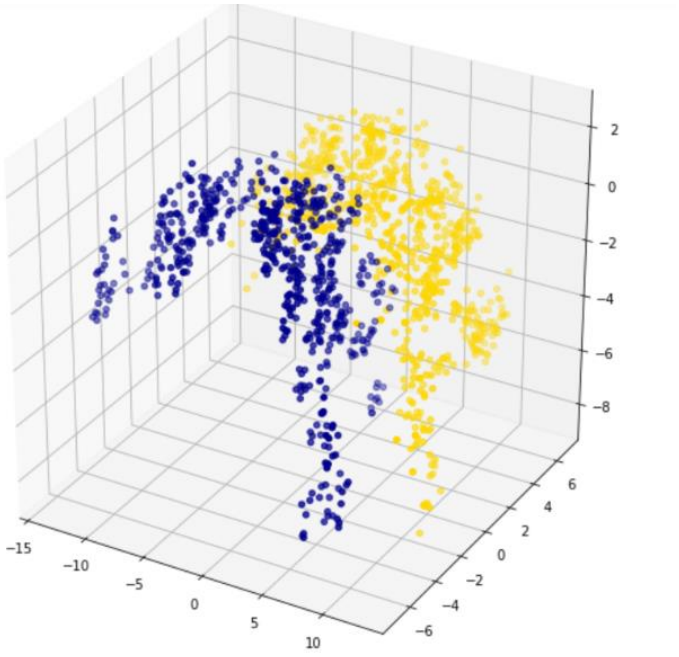

Fig 5.4: Skewness vs Variance vs Entropy graph

As seen from the above graph the overlap between the points has reduced drastically when three attributes were considered to plot the graph. But there is still a considerable overlap and a clear disticton between the classes is not visible. If a new point is introduced to the graph the predicted class will not have a margin of accuracy due to the overlap of points in the above graph. If a K-Nearset Neighbors algorithm is applied on this graph the accuracy of the model built will not be high and thus it will have to be optimised further. Thus the three points considered to plot the above graph will have to be revised.

- The values of Variance, Skewness and Curtosis from the data set were plotted taking Variance on the x-axis, Skewness on the y-axis and Curtosis on the z-axis.

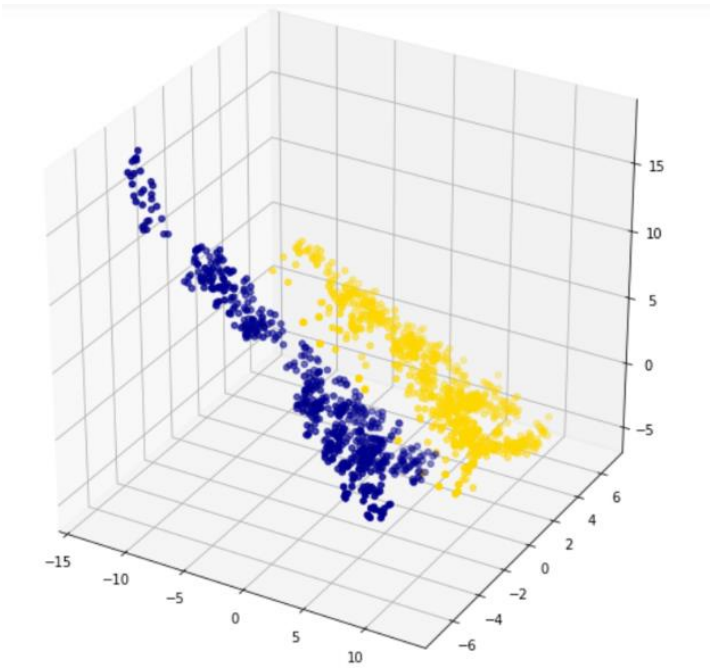

Fig 5.5: Skewness vs Variance vs Curtosis graph

In the above graph the distinction between the two classes is clearly visible and the overlap is the least. It can thus be concluded that the attributes used to plot the above graph have a correlation with the class of the image. The attributes that were considered to plot the above graph were – Variance, Skewness and Entropy. Thus, these are the attributes that are considered while building our KNN model to predict classes of points.

If the user choses the Data Set Analysis option, this graph is embedded on to the GUI window using the Matplotlib library. The user can analyze the graph and if the user hovers the cursor near any particular point the values corresponding to that are displayed. From analyzing this graph, the user gets understands the nature of the graph and also gets a better understanding of how the KNN algorithm uses this graph to predict values.

5.4 MODULE 4 IMPLEMENTATION

If the user choses the KNN Analysis option, information about the KNN model is displayed. This included the percentages in which the data set was split, the number of neighbors considered to build the model and the accuracy of the model. In order to test the efficiency of our model, the data set was split into training and testing data. The model was built on the training data and was checked against the testing data to calculate its accuracy. The values were put into training and testing sets at random so as to eliminate any bias. The percentage at which the training and testing data were split is 75% and 25% respectively.

The number of neighbors that were considered to build the model were – 31. This number had to be chosen very carefully in order to avoid overfitting or underfitting of the model. A higher number would have resulted on overfitting of the mode and a lower number would have resulted in underfitting of the model. The model is built in such a way that if a new point is introduced in the graph, the algorithm searches for the first 31 nearest neighbors and checks the class of these points. The class of the new point is predicted to be the one which has majority among the 31 neighbors. The number of neighbors considered while developing an algorithm is usually an odd number to avoid ties between two classes while predicting.

The accuracy of the graph is calculated against the testing data set. Once the model is built, the points of the training data set are plotted on the graph and their classes are predicted. Since, the classes of those points are already known, we find out how many of those were predicted to be right. In this this way, the accuracy of the model is calculated. The accuracy changes when we change the number of neighbors (K value) in our model. The accuracy of the model for 31 neighbors is calculated to be – 0.9825 that is 98%.

CHAPTER 6

RESULTS

The below screenshots are the working results of the program.

1) Home screen of the Application showing two options – Image Analysis and Data Model Analysis.

Fig 6.1:Result Screenshot - 1.

2) Image Analysis Window asking the user to enter the path.

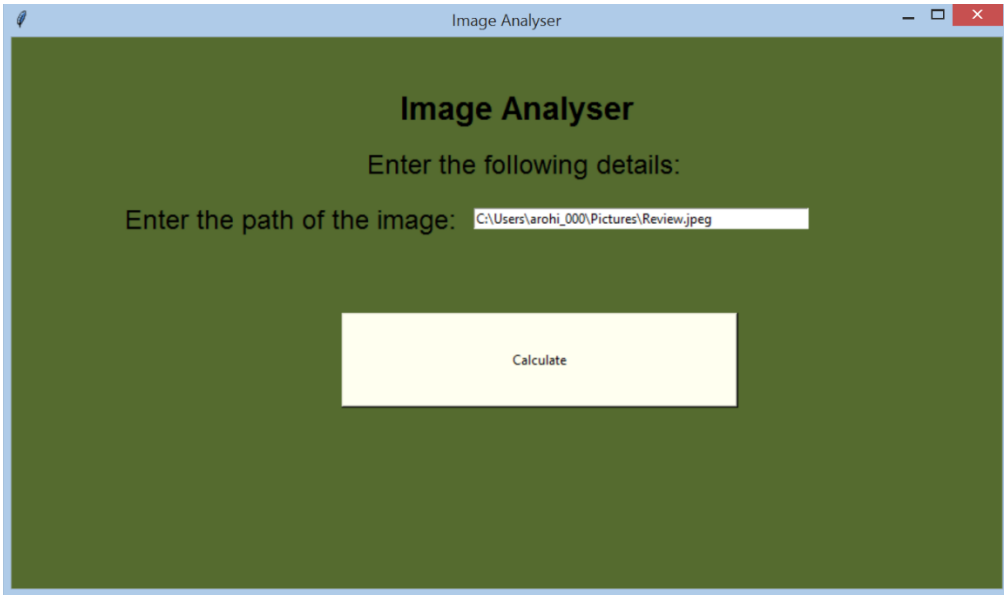

Fig 6.2: Result Screenshot - 2.

3) Sample Image with the values calculated from it.

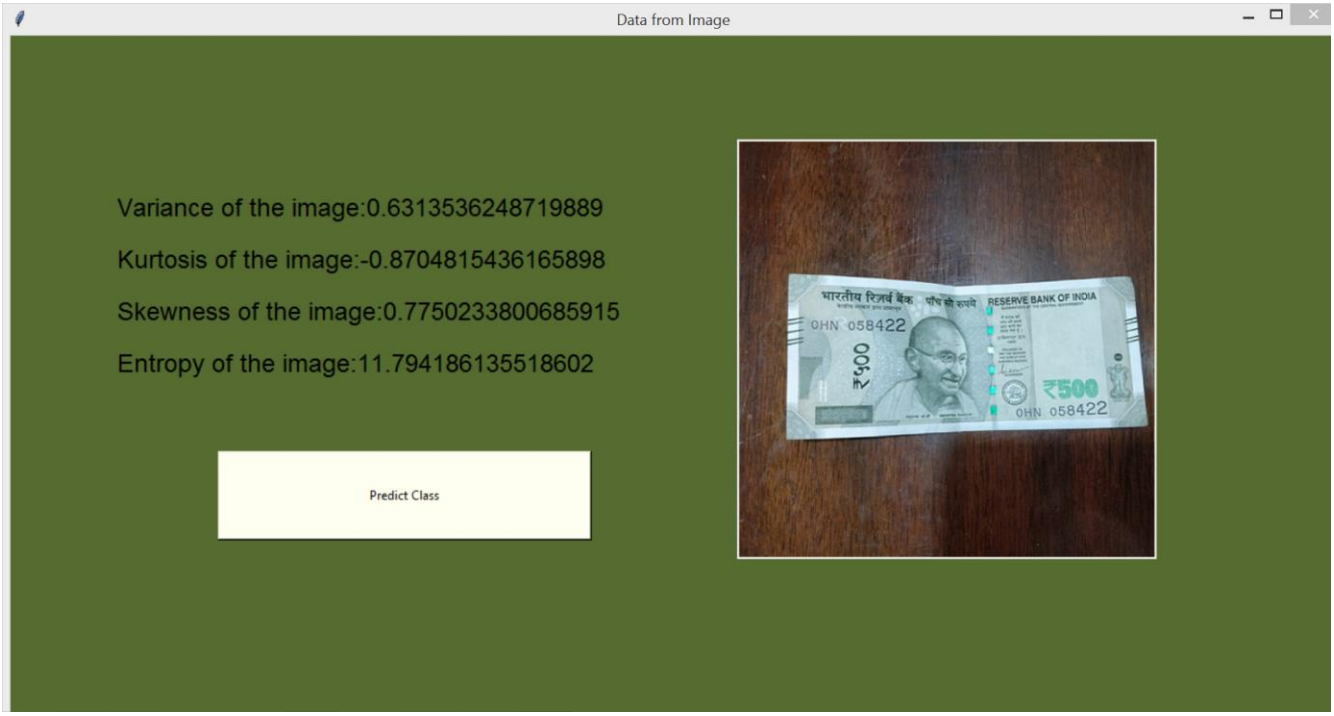

Fig 6.3: Result Screenshot -3

4) The predicted class of the sample image.

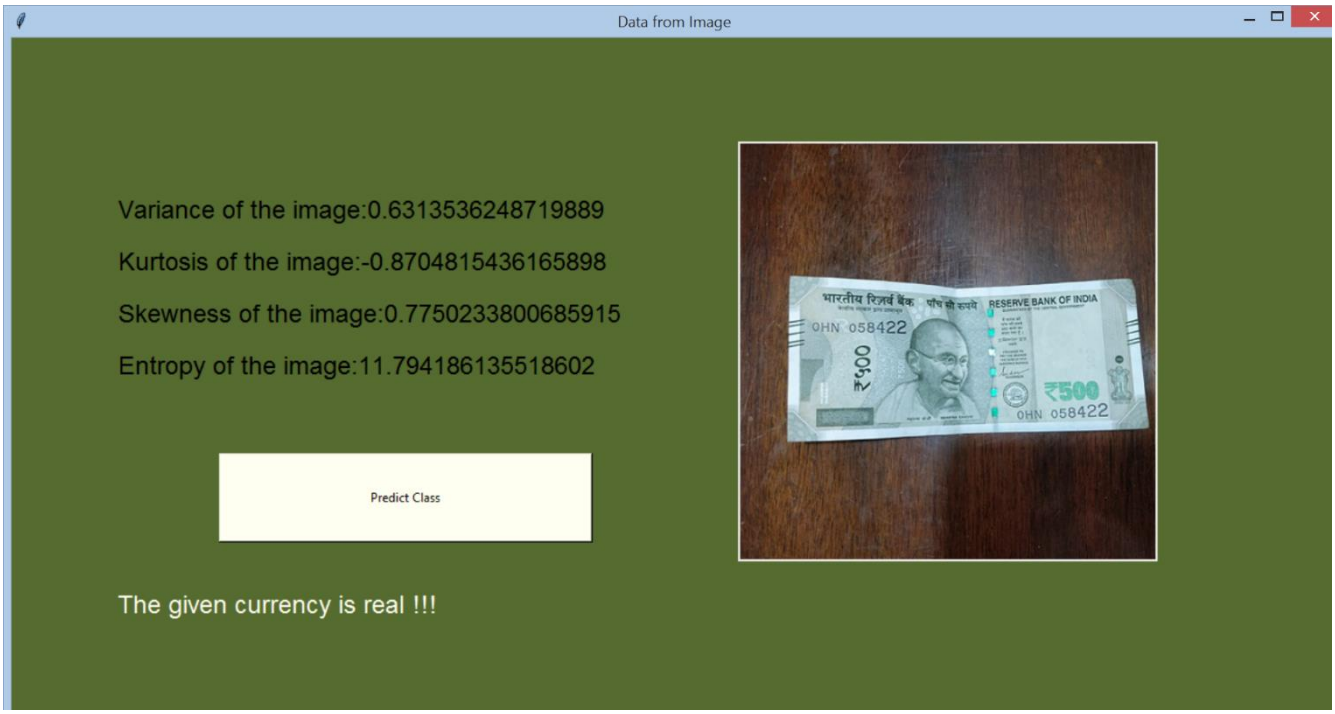

Fig 6.4: Result Screenshot – 4.

5) The Data Model Analysis Window

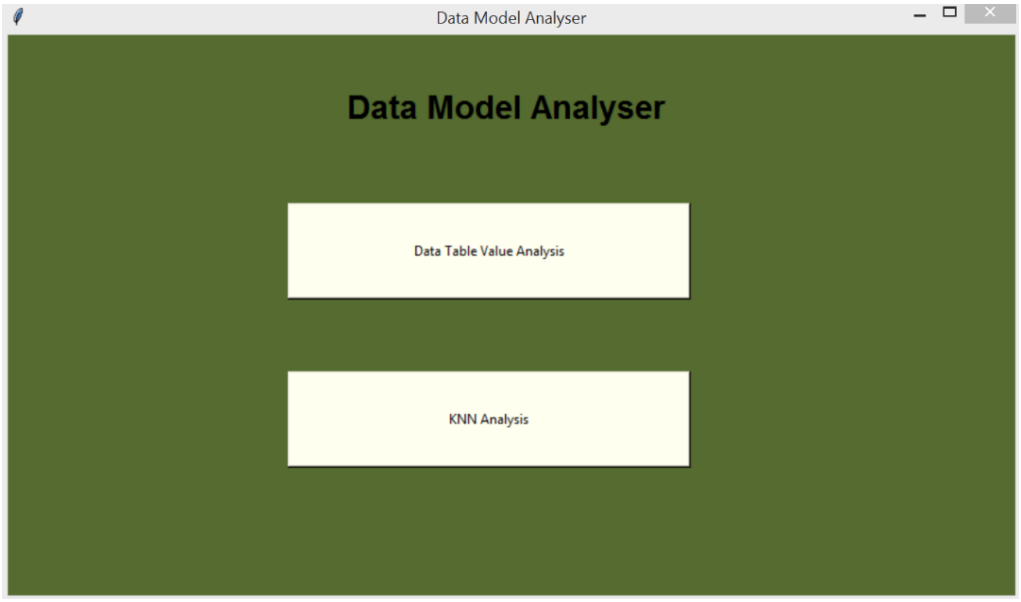

Fig 6.5: Result Screenshot – 5.

6) Data Set Analysis window displaying the graph.

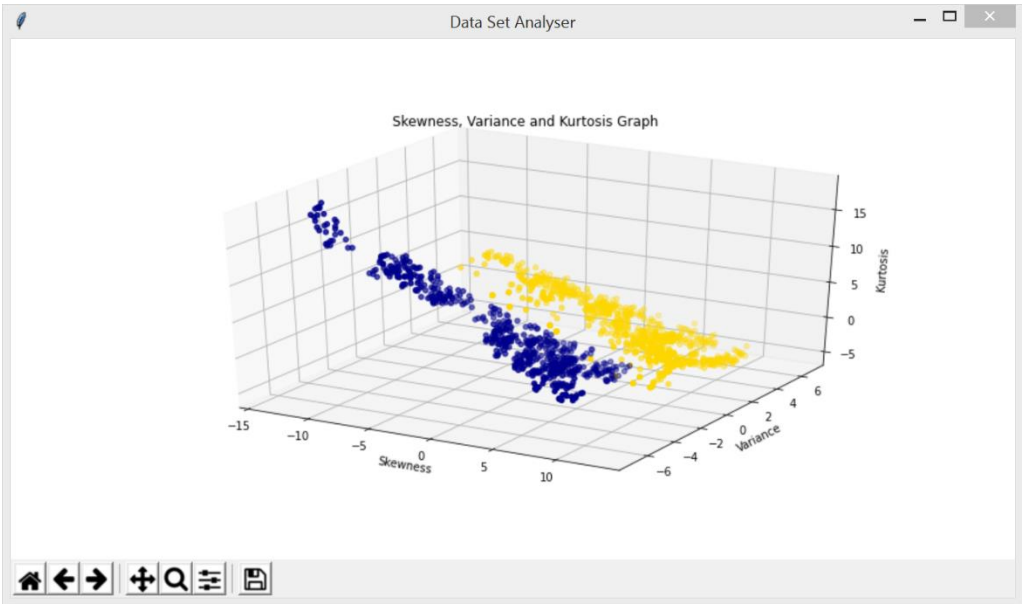

Fig 6.6: Result Screenshot – 6.

7) KNN Analysis window.

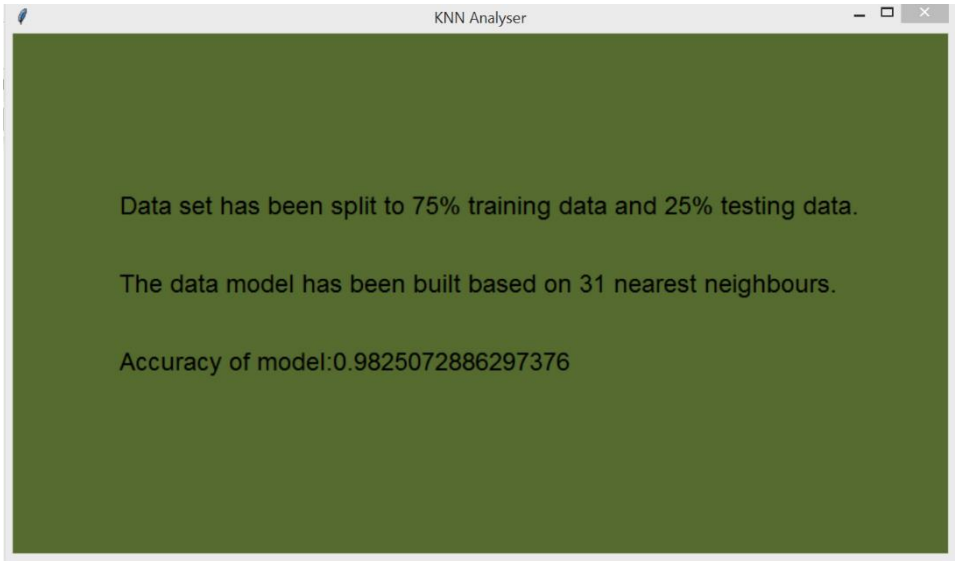

Fig 6.7: Result Screenshot – 7.

CHAPTER 7

CONCLUSION

A platform where people can check if their currency is counterfeit or real can help the society. This project is developed based on a data set and is for reference purposes only as it is not 100% accurate. The images used by the user can have a different exposure, contrast thus affecting the end result. Though, efforts have been made to resemble the images used by user to the images used in the data set, the accuracy of this project is based on the values of the data set and it may vary for images shot on a different camera with different settings . This project is developed using the concepts of machine learning algorithms in Python language.

Developing this project has also led us to have a better understanding of the machine learning concepts. Implementing those concepts in real life has made us more confident in the topic and we can henceforth be able to develop real time applications with much more clarity in mind. This has also led us to have a better grasp on the Python language. The main objective of understanding and analyzing the technological needs of the society has been achieved.

REFRENCES

[1] David Beazley - Python Essential Reference, 4th Edition.

[2] Guido van Rossum - Python Library Reference 1999.

[3] Wesley Chun – Core Python Programming, 2nd Edition, 2006.

[4] https://www.programiz.com/python-programming

[5] https://intellipaat.com/blog/tutorial/python-tutorial/

[6] https://www.tutorialspoint.com/python/python_gui_programming.htm

[7] https://www.python.org/

[8] https://www.kaggle.com/datasets