SAVITRIBAI PHULE PUNE UNIVERSITY
A PROJECT REPORT
ON

“World Happiness Analysis, Predication And Recommendation For Happier World”

SUBMITTED TOWARDS THE SAVITRIBAI PHULE PUNE UNVERSITY, PUNE IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENIGINNERING

IN

COMPUTER ENGINEERING

Mr. More Amrat Vishwanath	[B150124285]
Miss. Muthe Mayuri Madhukar	[B150124293]
Miss. Pathan Tabassum Faruk	[B150124304]
Miss. Shelar Prajakta Ishwar	[B150124315]

UNDER THE GUIDENCE OF

Prof. C.V.Longani

DEPARTMENT OF COMPUTER ENGINEERING
Sanjivani Rural Education Society’s,
Sanjivani College of Engineering,
Kopargaon - 423603

[27/2021-22]

CERTIFICATE

This is to certify that the Project Entitled

“World Happiness Analysis, Predication And Recommendation For Happier World ”

Submitted By

Mr. More Amrat Vishwanath	[B150124285]
Miss. Muthe Mayuri Madhukar	[B150124293]
Miss. Pathan Tabassum Faruk	[B150124304]
Miss. Shelar Prajakta Ishwar	[B150124315]

is a bonafide work carried out by Students under the supervision of **Prof. C. V. Longani** and It is submitted towards the fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Prof. C.V.Longani
[Internal Guide]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

Dr. A.G.Thakur
[Director]
Sanjivani College of Engineering, Kopargaon-423603

Signature of Internal Examiner **Signature of External Examiner**

PROJECT APPROVAL SHEET
A
Project Report

ON

"World Happiness Analysis, Predication And Recommendation For Happier World"

Is successfully completed by

Mr. More Amrat Vishwanath	[B150124285]
Miss. Muthe Mayuri Madhukar	[B150124293]
Miss. Pathan Tabassum Faruk	[B150124304]
Miss. Shelar Prajakta Ishwar	[B150124315]

(B.E. Computer)

at

DEPARTMENT OF COMPUTER ENGINEERING

Sanjivani College of Engineering

Kopargaon - 423 603.

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

ACADEMIC YEAR 2021-2022

Prof. C.V.Longani
[Internal Guide]

Dr. A. B. Pawar
[Project Coordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

ACKNOWLEDGEMENT

It is with the greatest pleasure and pride that we present this report. At this moment of triumph, it would be neglect all those who helped us in the successful partially completion of this project. We are very much thankful to our respected project guide **Prof. C.V.Longani** for her ideas and help proved to be valuable and helpful during creation of partial project report and set us in the right path.We also extend our gratitude to **Dr. A. B. Pawar (Project Coordinator)** and **Dr. D. B. Kshirsagar** (**H.O.D. Computer Department).** We would also like to thank all the faculties who have cleared all the major concepts that were involved in the understanding techniques behind our project. Lastly, we are thankful to our friends who shared their knowledge in this field with us.

Mr. More Amrat Vishwanath
Miss. Muthe Mayuri Madhukar
Miss. Pathan Tabassum Faruk
Miss. Shelar Prajakta Ishwar
(B.E. COMPUTER)

ABSTRACT

The concept of measuring happiness was introduced to the world by the government of Bhutan as Gross National Happiness which later was adapted by UN in the form of happiness index that measures the happiness and well-being of the population of the country. This project aims to analyze the happiness index and its relationship with various factors by making use of different attributes related to happiness of people. Features like ladder score, GDP per capita, life expectancy score etc. are used to analyze the happiness. The results have been presented by using pie charts, graphs and tables. The algorithms and tools like multiple regression and Naive Bayes have been used to measure the extent of dependence of the factors on happiness index. The Happiness Index is framed to set various parameters on grounds of which a country could be ranked in a list of 156 countries. India's rank has come down the list in the year 2019 to be ranked at the 140th position. This clearly indicates India's deteriorating position down the years. This project elaborates the concept of Happiness Index as a measure and analyses various reasons for India to lose its position in the World Happiness Report.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Problem Definition

The concept of measuring happiness was introduced to the world by the government of Bhutan as Gross National Happiness which later was adapted by UN in the form of happiness index that measures the happiness and well- being of the population of the country. This analysis is very important for the specialists in the fields of financial aspects, brain research, overview examination, public insights, wellbeing, public arrangement, etc. This is very useful to evaluate the advancement of countries.

In this project, we analyze the World and Indian happiness data using Data Science. We answer the question- Are Indians Happy? We discussed various parameters that contribute to provide happiness to the citizens like Log GDP per capita scores, Healthy life expectancy scores, Perceptions of corruption scores, Social support scores, Freedom to make life choices scores, and Generosity scores. We also predict the future values of different parameters related to Happiness of a country. We also propose to generate the recommendations for India to make the country happier, in the ranking of Happiness-Index.

1.2 Literature Review

1. A Comparative Analysis Of The Factors Affecting Happiness Index

Year:-2020

Author:- Parul Oberoi, Shalu Chopra, Yukti Sheth

This paper proposes a study that has been carried out to find the reasons behind thecountry's low happiness index. The factors taken under study for this research are

- Social Support
- Freedom to make life choices
- Physical well-being
- Personal safety
- Generosity

factor that was found to be most influential in determining the happiness of the selected sample The through multiple regression was physical well-being and it was also observed that it was the same factor with the least average i.e., low value of physical well-being among the sample in area of study undertaken is causing low value of happiness index too.

2. Future Prediction of World Countries Emotions Status to Understand Economic Status using Happiness Index and SVM Kernel.

Year:-2019

Author:-B.Prashanthi, Dr. R. Ponnusamy

In this paper, a supervised two-tier ensemble approach for predicting a country's BLI score was proposed. The work presented a cost-effective method of BLI prediction with a high degree of efficiency.

The capability of the model to predict life satisfaction relied on the proper training features, chosen using a recursive elimination method with 10-fold cross-validation. The work combined three of the top four models, with simple averaging, to enhance the performance of the regression. This is forecasting the Better Life Index (BLI) score using machine learning based regression model that can influence the survival of future generations.

3. A Data Analysis of the World Happiness Index and its Relation to the North-SouthDivide

Year:- 2019

Author:- Charles Alba

In this paper, authors performed a detailed data analysis on the World Happiness Report with its relation to the socio-economic North-South Divide. In order to do so, they performed some extensive data cleaning and analysis before querying on the World Happiness Report. The results based on Hypothesis Testing determines the happiness of the Global North is greater than that of the Global South. Furthermore, queries presented in paper show that the mean happiness score for the Global North significantly outweighing that of the South. Likewise, the 10 'Happiest' nations all belong to the Global North whereas the 10 'least happy' nations belong to the Global South.

4. Analyzing Happiness Index as a Measure Along With its Parameters and Strategies for Improving India’s Rank in World Happiness Report

Year: - 2019.

Author: -Sarah Ahtesham

This paper elaborates the concept of Happiness Index as a measure and analyses various reasons for India to lose its position in the World Happiness Report. The author appropriately concludes the paper with suitable suggestion.

1.3 Scope

- In this project the future values related to Happiness of a country, is predicted.
- Recommendations for India about improving the happiness with different attributes to make the country more happy in the ranking of Happiness-Index.
- In this project Analysis results are presented in graphical format.

1.4 Objective

- To perform the analysis of World Happiness data by extracting the knowledge about the Happiness-Index of a country.
- To predict the future values of different parameters related to Happiness of a country.
- To generate the recommendations for India to make the country more happy in the ranking of Happiness-Index.
- To learn and implement the techniques of Multiple Regressions .
- To learn and implement the Naive Bayes.

1.5 System Architecture

The System Design and architecture gives the detail architecture for developing the proposed system.

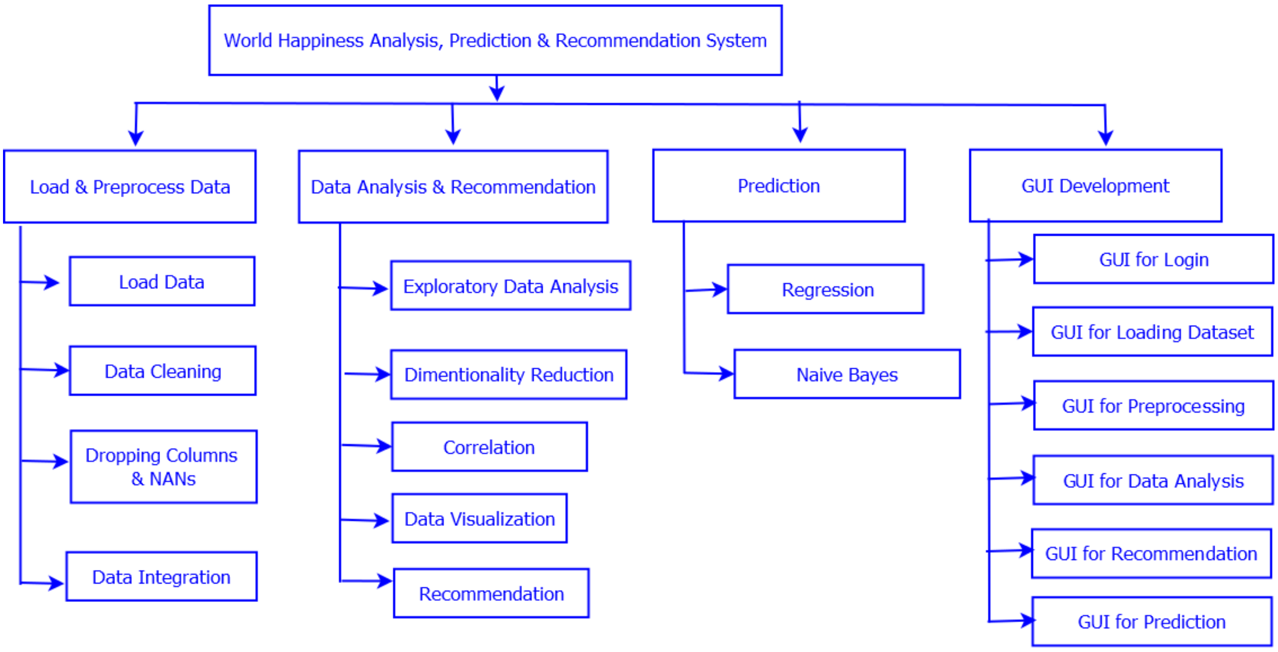


Figure 1.1: System Architecture

CHAPTER 2

REQUIREMENT ANALYSIS

Requirement Analysis and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholder i.e. beneficiaries or users. Requirement analysis is software engineering task that bridges the gap between system level software description and design model. The system description describes overall system functionality of the system including software, hardware, database, human interfaces, and other system element. And website design mainly focuses on application architectural, user interface and component level designs.

A software requirements specification (SRS) is a comprehensive description of the in- tended purpose and environment for software under development. SRS minimizes the time and sort required by developers to achieve desired goals and also minimizes the development cost.

2.1 Requirement Specification

The term specification means different things to different people. A specification can be written document, a graphical model, and a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these. These requirement specifications are of three types i.e.

N: Normal Requirements

E: Expected Requirements

X: Excited Requirements

2.1.1 Normal Requirement

N1: It should provide the facility to upload the dataset.

N2: It should handle the inconsistencies in dataset.

N3: System must be user friendly.

N4: System should do the analysis of World Happiness data so as to present the results in human understandable format.

N5: The system should generate the recommendations for India to make the country more happy in the ranking of Happiness-Index.

N6: Analysis results should be presented in graphical format.

Expected Requirements

These requirements are expected by customer but not clearly stated by customer. This are implicit type of requirements.

E1: The system must work in a reasonable time.

E2: The system should work on the dirty data also.

E3: The System perform analysis of other factors also which extracts more knowledge about happiness.

2.1.2 Excited Requirements

May become normal requirements in the future, highly prized and valued.

X1: Graphical results should contain more details for better understanding.

X2: Graphical User Interface should be aesthetic in nature.

2.2 Validation of Requirement

Requirement validation examines that all system requirements have been stated unambiguously and inconsistencies and errors have been detected and corrected. It ensures that the work products confirmed to the standards established for the process, project and product. The work products produced as a consequence of requirement engineering are assessed during validation.

2.2.1 Validation of Normal Requirements

VN1: Proper and simple GUI can be implemented in the selected programming language and tools.

VN2: The data should be preprocessed before its use for analysis.

VN3: The system should be provided with simple and easy to use GUI for ease of access.

VN4: Standard, well tested, efficient algorithms and techniques should be used in the implementation for better results.

VN5: Proper recommendation algorithm should be used to get the accurate results.

VN6: Visualization tools like matplotlib should be used to present the analysis results in graphical format.

2.2.2 Validation of Expected Requirements

VE1: The selected algorithm should be more efficient in terms of time complexity and space complexity. The developed code should be optimized to execute in less time.

VE2: The code should handle the inconsistencies in data using standard preprocessing techniques.

VE3: It is needed to dive deep into the data to know more insights about data.

2.2.3 Validation of Excited Requirements

VX1: Visualizations should include more details about analysis.

VX2: GUI tools like tkinter should be used with standard procedures of GUI creation.

2.3 System Requirements

2.3.1 Software Requirements

1. Operating System: Windows 10/Linux
2. Database : MySQL
3. Language : Python 3.9
4. Python Packages: Tkinter, matplotlib, numpy, pandas, csv etc.

2.3.2 Hardware Requirements

1. Ram: 4 GB
2. Processor: i5

CHAPTER 3

SYSTEM MODEL

3.1 Process Model

A spiral model is divided into a number of framework activities, also called task regions. For the proposed project we are going to use the following process model. The spiral model, originally proposed by Boehm, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental versions of the software. Using the spiral model, software is developed in a series of incremental releases. During early iterations, the incremental release might be a paper model or prototype. During later iterations, increasingly more complete versions of the engineered system are produced. Typically, there are between three and six task regions. Figure 3.1 depicts a spiral model that contains six task regions: For the proposed project we are going to use the following process model.

3.1.1 Selected Model: Spiral Model

Spiral model required obtaining customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage. Each of the regions is populated by a set of work tasks, called a task set, that are adapted to the characteristics of the project to be undertaken. For small projects, the number of work tasks and their formality is low. For larger, more critical projects, each task region contains more work tasks that are defined to achieve a higher level of formality. In all cases, the umbrella activities (e.g., software configuration management and software quality assurance) are applied. As this evolutionary process begins, the software engineering team moves around the spiral in a clockwise direction, beginning at the centre. The first circuit around the spiral might result in the development of a product specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software. Each pass through the planning region results in adjustments to the project plan. Cost and schedule are adjusted based on feedback derived from customer evaluation. In addition, the project manager adjusts the planned number of iterations required to complete the software. Unlike classical process models that end when software is delivered, the spiral model can be adapted to apply throughout the life of the computer software. An alternative view of the spiral model can be considered by examining the project entry point axis, also shown in Figure 3.1. Each cube placed along the axis can be used to represent the starting point for different types of projects.

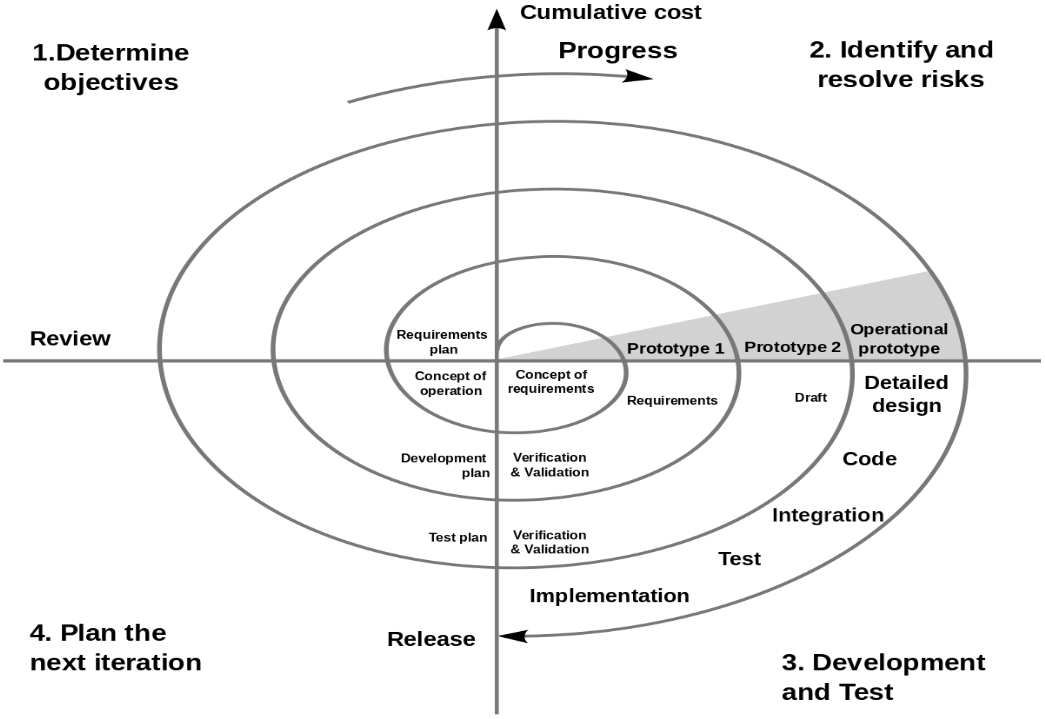


Figure 3.1: Spiral Model

3.1.2 Advantages:-

1. Additional functionality or changes can be done at a later stage.
2. Cost estimation becomes easy as the prototype building is done in small fragments.
3. Continuous or repeated development helps in risk management.
4. Development is fast and features are added in a systematic way in Spiral development.
5. There is always a space for customer feedback.

3.1.3 When to Use Spiral Model:-

1. A Spiral model in software engineering is used when project is large
2. When releases are required to be frequent, spiral methodology is used
3. When creation of a prototype is applicable
4. When risk and costs evaluation is important
5. Spiral methodology is useful for medium to high-risk projects
6. When requirements are unclear and complex, Spiral model in SDLC is useful
7. When changes may require at any time
8. When long term project commitment is not feasible due to changes in economic priorities

3.1.4 Why to use Spiral Models

The main aim of using the model is the reason that we have to add more features in the existing modules to increase project reliability and usability. Using this model we can adapt to the changing requirements of the customer which helps in developing the project in relatively small amount of time. The process is repeated until the project is completed.

3.1.5 Disadvantages of Spiral Models

1. Risk of not meeting the schedule or budget
2. Spiral development works best for large projects only also demands risk assessment expertise
3. For its smooth operation spiral model protocol needs to be followed strictly
4. Documentation is more as it has intermediate phases
5. Spiral software development is not advisable for smaller project, it might cost them a lot

3.2 Project Estimation

3.2.4 Estimation in KLOC

The number of lines required for implementation of various modules can be estimated as follows.

Table 3.1: Estimation of KLOC

Sr. no.	Modules	Estimated KLOC
1	GUI	0.3 KLOC
2	Input Data	0.2 KLOC
3	Data Pre-processing	0.4 KLOC
4	Process Model	0.4 KLOC
5	Summarization	0.8 KLOC
Total		2.81 KLOC

3.2.5 Efforts(E):-

The Efforts required in person/month for implementation can be estimated as follows
E = 3.2 *(KLOC)1.05
E = 3.2 *(2.81)1.05
E = 9.4 persons/month

3.2.6 Development Time (In months):-

D = E/N
D=9.4/4
D = 2.35 months

Development time for project requirement analysis and design requires 3.0 months implementation and testing requires 2.35 months.

D = 3.0 + 2.35 Months
D = 5.35 Months

Total time required for the successful development of project is 5.35 months.

3.2.7 Number of Person Required (N):

Number of Person=Efforts/Duration
Number of Person =9.47/2.36
Number of Person =4 person.

Four Persons are required to complete the project within given time span successfully.

D1: Mr. More Amrat
D2: Miss. Muthe Mayuri
D3: Miss. Pathan Tabassum
D4: Miss. Shelar Prajakta

CHAPTER 4

SYSTEM DESIGN

4.1 Algorithm

4.1.1 Regression

Regression Algorithms will be used to predict certain parameters of happiness whichare dependent on one or more other parameters. A multiple regression considers the effect of more than one explanatory variable on some outcome of interest. It evaluates the relative effect of these explanatory, or independent, variables on the dependent variable when holding all the other variables in the model constant.

This regression algorithm has several application across the industry for product pricing, real estate pricing, marketing departments to find out the impact of campaigns.

Multiple regression formula is used in the analysis of relationship between dependent and multiple independent variables and formula is represented by the equation Y is equal to a plus bX1 plus cX2 plus dX3 plus E where Y is dependent variable, X1, X2, X3 are independent variables, a is intercept, b, c, d are slopes, and E is residual value.

$$\mathbf{y = mx1 + mx2 + mx3 + c}$$

Where,

Y= the dependent variable of the regression

M= slope of the regression

X1=first independent variable of the regression

The x2=second independent variable of the regression

The x3=third independent variable of the regression

C= constant

4.1.2 Naive Bayes

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, These features are still considered independently. This assumption simplifies computation and that’s why it is considered as naïve. This assumption is called class conditional independence.

In order to get a baselin accuracy rate for our data, We are using a Naïve Bayes classifier. Specially, We used the scikit-learn implementation of Gaussian Naive Bayes. This is one of the simplest approaches to classification, in which a probabilistic approach is used, with the assumption that all features are con ditionally independent given the class label. The Naive Bayes Rule is based on the Bayes theorem:

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)}$$

Parameter estimation for naive Bayes models uses the method of maximum likelihood. The advantage here is that it requires only a small amount of training data to estimate the parameters.

Where,

- Let X be a data sample (“evidence”): class label is unknown
- Let H be a hypothesis that X belongs to class C
- Classification is to determine P(H|X), the probability that the hypothesis holds
- given the observed data sample X
- P(H) (prior probability), the initial probability
- E.g., X will buy computer, regardless of age, income, …
- P(X): probability that sample data is observed
- P(X|H) (posteriori probability), the probability of observing the sample X,
- given that the hypothesis holds
- E.g., Given that X will buy computer, the prob. that X is 31..40, medium income

Given training data X, posteriori probability of a hypothesis H, P(H|X), follows the Bayes theorem.

4.2 Project Scheduling and Tracking

Project Scheduling and Tracking is important because in order to build a complex system, many software engineering tasks occurs in parallel, and the result of work performed during one task may have a profound effect on work to be conducted in another task. These inter dependencies are very difficult to understand without detailed schedule.

Project Work and Breakdown Structure (Analysis) :

The project work is decomposed into the following work break down structure as a part of analysis phase.

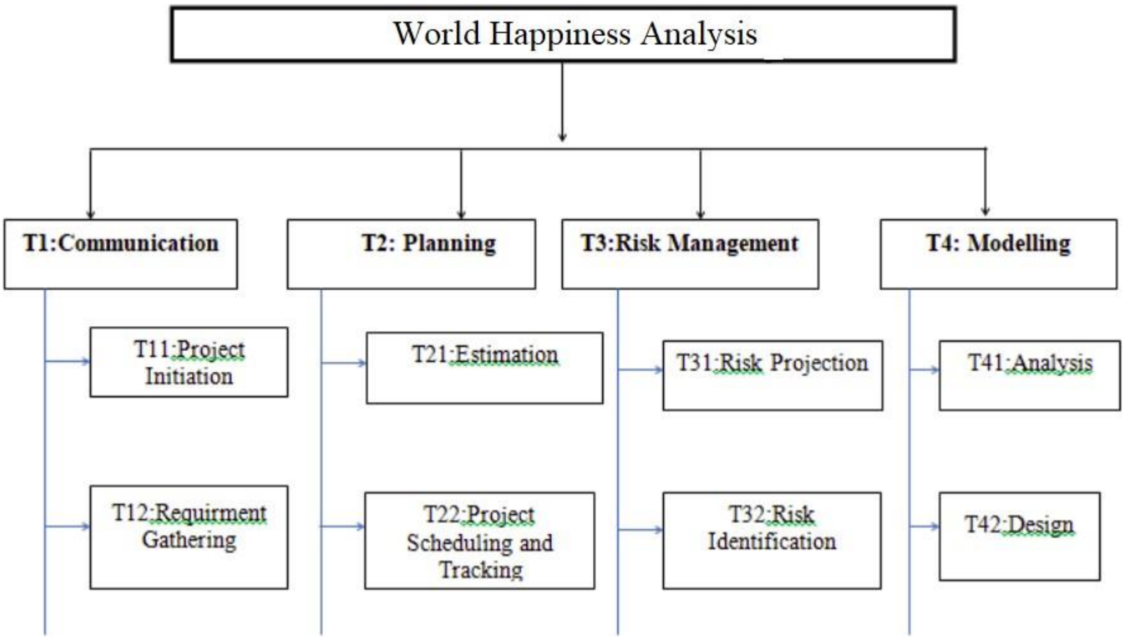


Figure 4.1: Project Work Breakdown Structure (Analysis)

- T1 : Communication Software development process starts with the communication between customer and developer. According to need of project, we gathered the requirements related to project. Requirement gathering is an important aspect as the developer will come to know what customer expects from the project and also he can help a customer to know more features that can be added to project as he is a technical person. The most important thing needed is that communication should be smooth and clear that means developer should easily understand the demands of customer.

- T2 : Planning It includes complete estimation and scheduling (complete time line chart for project development). Before starting the project tasks should be scheduled that means there should be starting and ending date assigned for each and every task and developer should work harder to complete the required task within time chosen at the time of scheduling.

  T3 : Risk Management It is a process of identifying, organizing, assessing and controlling thrats to some organizations capitals and earnings which affects overall or partial software products or performance. These threats or risk could steam from a wide variety of sources, including Financial uncertainty, legal liabilities, strategies, management errors, accidents and natural disasters.

T4 : Modeling It includes detailed requirements analysis and project design (algorithm, flowch Art etc). Flowchart shows complete pictorial view of the projects and algorithm is step by step Solution of problem. Both flowchart and algorithm is helpful in knowing the overall view of – Project and serve as a base for development of whole project.

Project Work and Breakdown Structure(Implementation):

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

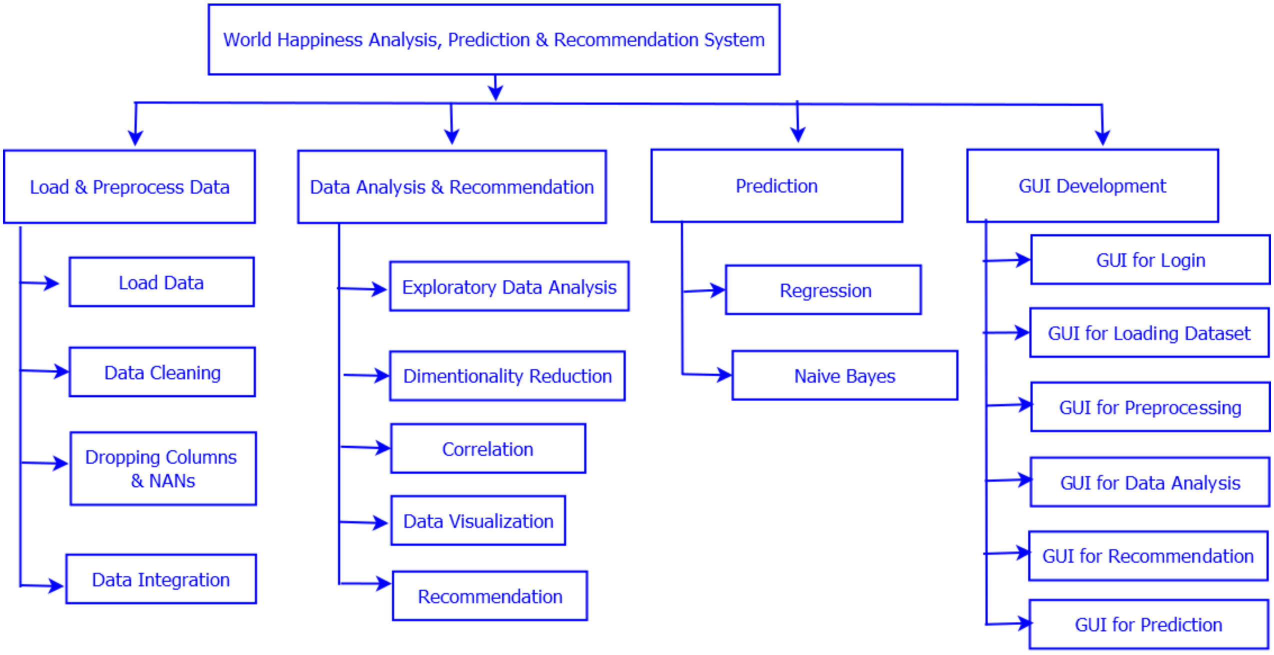


Figure 4.2: Breakdown Structure(Implementation)

4.3 Modules

1. Load & preprocess Data

1.1 Load Data

This module will load the dataset from local computer to the system.

1.2 Data Cleaning

Many times the data is dirty. This module will preprocess the data towards removing the dirty components from the dataset.

1.3 Dropping Columns & NANs

In order to further process the data, the unnecessary columns will be dropped and NA entries will be handled by this module.

2. Data Analysis & Recommendation

This module will perform the detail analysis and derive many recommendations to the user.

2.1 Exploratory Data Analysis

The data explored to get the insights of data at abstract level.

2.2 Dimensionality Reduction

The dimensions of data are reduced to enhance the processing time and to focus the analysis on only the targeted data.

2.3 Correlation

The relationships between different parameters in the dataset, is analyzed using the method of correlation.

2.4 Data Visualization

This module display the graphical visualization of analysis and recommendation using the matplotlib & tkinter library.

2.5) Recommendation

Based on the analysis, this module recommends the conclusions based of certain predefined criterion.

3. Prediction

This module will calculate the happiness index and predict the future values of different parameters which are important in the happiness index.

3.1 Multiple Linear Regression

Regression Algorithms will be used to predict certain parameters of happiness which are dependent on one or more other parameters.

3.2 Naive Bayes

In order to get a baseline accuracy rate for our data, we are using a Naive Bayes classifier. Specifically, we used the scikit-learn implementation of Gaussian Naive Bayes.

4 GUI Development

In this module, GUI for different functionalities will be developed using the tkinter library in python

4.4 Task:

following analysis and design tasks are to be carried out in process of analysis and design of project. All project modules are divided into following tasks.

- T1: Project definition searching.
- T2 : Literature collection
- T3 : Give presentation of the topic.
- T4 : Give presentation of the topic.
- T5 :Allocation of responsibilities.
- T6 :Synopsis submission.
- T7 : Detail information gathering of each module.
- T8 : Processes modeling and Estimation.
- T9 : UML diagrams.
- T10 : Risk analysis and management.
- T11 : Mathematical modelling.
- T12 : Preparation of partial project report.
- T13 : GUI Module Implementation.
- T14 : Preprocessing Module Implementation.
- T15 :Data Analysis Module Implementation.
- T16: Data Visualization Module Implementation.
- T17:Recommendation Module Implementation.
- T18:Regration Module Implementation.
- T19:Navie Bayes Module Implementation.
- T20:Integration of all module.
- T21 : Testing and Trouble shooting.
- T22 : Report Generation.
- T23: Report Submission.

4.5 Task Identification

4.5.1 Project Schedule

Table 4.1 : Project Task Table

Task	Days	Dependency	Developer Assigned
T1	12	-	D1, D2, D3, D4
T2	03	T1	D1, D2, D3, D4
T3	02	T1, T2	D1, D2, D3, D4
T4	02	T3	D1, D2, D3, D4
T5	09	T3, T4	D1, D2, D3, D4
T6	03	T3, T5	D1, D2, D3, D4
T7	10	T6	D1, D2
T8	04	T6, T7	D3, D4
T9	05	T7, T8	D1, D3
T10	17	T7, T8	D1, D2, D3, D4
T11	02	T7, T8	D2, D3
T12	06	T7, T8, T9, T10, T11	D1, D2, D3, D4
T13	10	T7, T10	D1, D4
T14	10	T7, T10	D2, D3
T15	09	T7, T10	D3, D4
T16	08	T7, T10	D1, D2
T17	02	T7, T10	D2, D4
T18	10	T7, T10	D1, D3
T19	10	T13, T14,T15,T16,17,18	D1, D2, D3, D4
T20	07	T13, T14,T15,T16,17,18, T19	D1, D2, D3, D4
T21	07	T13, T14,T15,T16,17,18, T19	D1, D2, D3, D4
T22	15	T1 to T21	D1, D2, D3, D4
T23	05	T1 to T21	D1, D2, D3, D4
Total	168	-	-

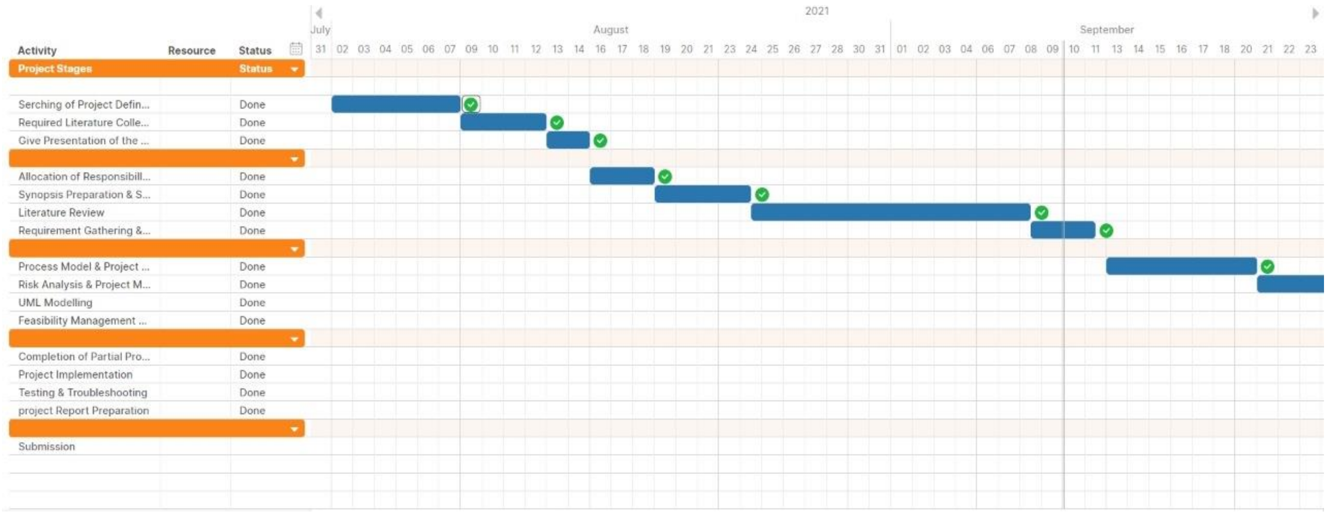
4.5.1 Project Table And Time-Line Chart

Table 4.2 : Project Schedule Time Chart

Task	Exp. Start Time	Act. Start Time	Exp. End Time	Act. End Time	Developers
T1	20/07/2021	20/07/2021	31/07/2021	31/07/2021	D1, D2, D3, D4
T2	01/08/2021	01/08/2021	03/08/2021	03/08/2021	D1, D2, D3, D4
T3	04/08/2021	04/08/2021	05/08/2021	05/08/2021	D1, D2, D3, D4
T4	06/08/2021	06/08/2021	07/08/2021	08/08/2021	D1, D2, D3, D4
T5	08/08/2021	09/08/2021	16/08/2021	20/08/2021	D1, D2, D3, D4
T6	17/08/2021	21/08/2021	19/08/2021	24/08/2021	D1, D2, D3, D4
T7	20/08/2021	25/08/2021	29/08/2021	04/09/2021	D1, D2
T8	30/08/2021	05/09/2021	02/09/2021	10/09/2021	D3, D4
T9	03/09/2021	11/09/2021	07/09/2021	15/09/2021	D1, D3
T10	08/09/2021	16/09/2021	24/09/2021	06/10/2021	D1, D2, D3, D4
T11	25/09/2021	07/10/2021	26/09/2021	09/10/2021	D2, D3
T12	27/09/2021	10/10/2021	03/10/2021	24/10/2021	D1, D2, D3, D4
T13	01/12/2021	7/12/2021	09/12/2021	16/12/2021	D1, D2
T14	10/12/2021	20/12/2022	19/12/2021	30/12/2021	D3, D4
T15	20/12/2021	02/01/2022	28/12/2021	11/01/2022	D1, D2, D3
T16	29/12/2021	12/01/2022	05/01/2022	20/01/2022	D1, D3, D4
T17	06/01/2022	21/01/2022	07/01/2022	23/01/2022	D2, D3, D4
T18	08/01/2022	25/01/2022	17/01/2022	08/02/2022	D1, D2, D3, D4
T19	18/01/2022	10/02/2022	27/01/2022	24/02/2022	D1, D2, D3, D4
T20	28/01/2022	26/02/2022	03/02/2022	05/03/2022	D1, D2, D3, D4
T21	04/02/2022	06/03/2022	10/02/2022	12/03/2022	D1, D2, D3, D4
T22	11/02/2022	13/03/2022	25/02/2022	30/03/2022	D1, D2, D3, D4
T23	26/02/2022	01/04/2022	02/03/2022	08/04/2022	D1, D2, D3, D4

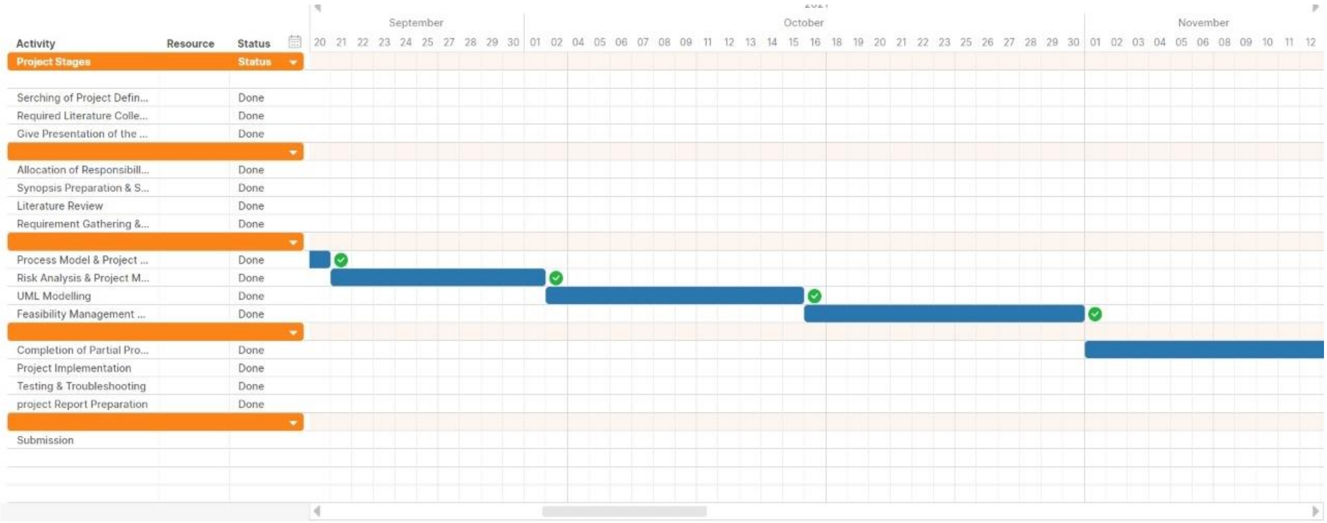
4.5.2 Time-Line Chart

Timeline chart shows the progress of project development in various phases. Timeline chartor the project is divided into 6 months accordingly given in figures below.



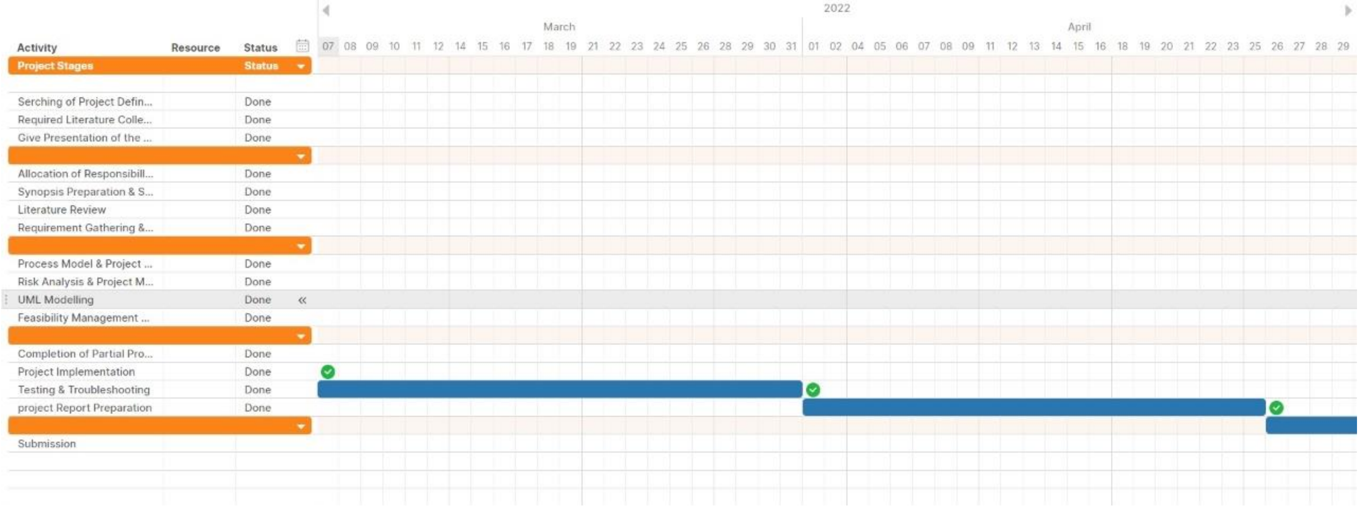
Activity	Resource	Status
Project Stages		Status
Serching of Project Defin...		Done
Required Literature Colle...		Done
Give Presentation of the ...		Done
Allocation of Responsibill...		Done
Synopsis Preparation & S...		Done
Literature Review		Done
Requirement Gathering &...		Done
Process Model & Project ...		Done
Risk Analysis & Project M...		Done
UML Modelling		Done
Feasibility Management ...		Done
Completion of Partial Pro...		Done
Project Implementation		Done
Testing & Troubleshooting		Done
project Report Preparation		Done
Submission		

Figure 4.3: Timeline Chart 1



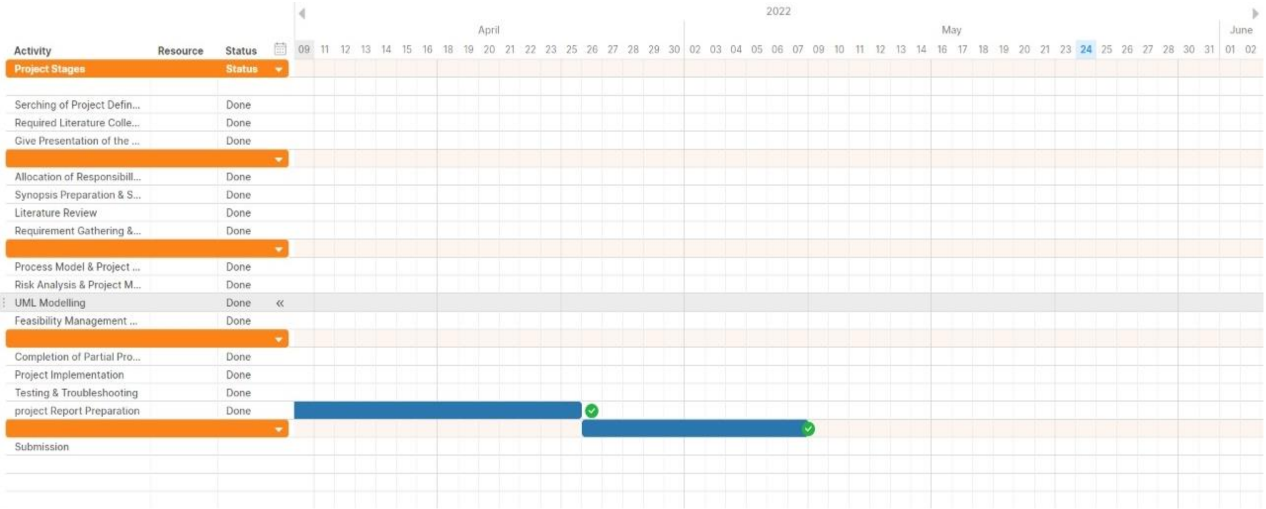
Activity	Resource	Status
Project Stages		Status
Serching of Project Defin...		Done
Required Literature Colle...		Done
Give Presentation of the ...		Done
Allocation of Responsibill...		Done
Synopsis Preparation & S...		Done
Literature Review		Done
Requirement Gathering &...		Done
Process Model & Project ...		Done
Risk Analysis & Project M...		Done
UML Modelling		Done
Feasibility Management ...		Done
Completion of Partial Pro...		Done
Project Implementation		Done
Testing & Troubleshooting		Done
project Report Preparation		Done
Submission		

Figure 4.4: Timeline Chart 2



Activity	Resource	Status
Project Stages		Status
Serching of Project Defin...		Done
Required Literature Colle...		Done
Give Presentation of the ...		Done
Allocation of Responsibill...		Done
Synopsis Preparation & S...		Done
Literature Review		Done
Requirement Gathering &...		Done
Process Model & Project ...		Done
Risk Analysis & Project M...		Done
UML Modelling		Done
Feasibility Management ...		Done
Completion of Partial Pro...		Done
Project Implementation		Done
Testing & Troubleshooting		Done
project Report Preparation		Done
Submission		

Figure 4.5:Timeline Chart 3



Activity	Resource	Status
Project Stages		Status
Serching of Project Defin...		Done
Required Literature Colle...		Done
Give Presentation of the ...		Done
Allocation of Responsibill...		Done
Synopsis Preparation & S...		Done
Literature Review		Done
Requirement Gathering &...		Done
Process Model & Project ...		Done
Risk Analysis & Project M...		Done
UML Modelling		Done
Feasibility Management ...		Done
Completion of Partial Pro...		Done
Project Implementation		Done
Testing & Troubleshooting		Done
project Report Preparation		Done
Submission		

Figure 4.6:Timeline Chart 4

4.6 Analysis Modeling

4.6.1 Behavioral Modeling

Use Case: A use case involves a sequence of interactions between the initiator and the system, possibly involving other actors.

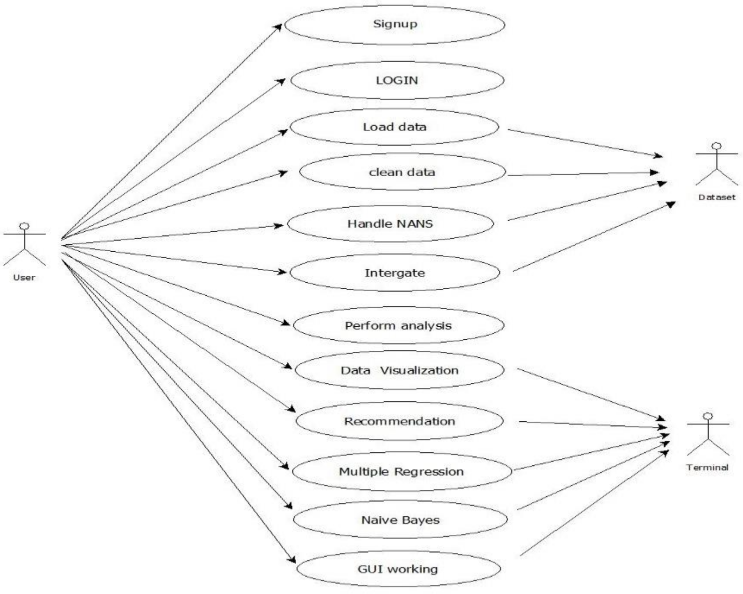


Figure 4.7: Use-Case Diagram

State Machine Diagram: A state machine is a comutation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states.

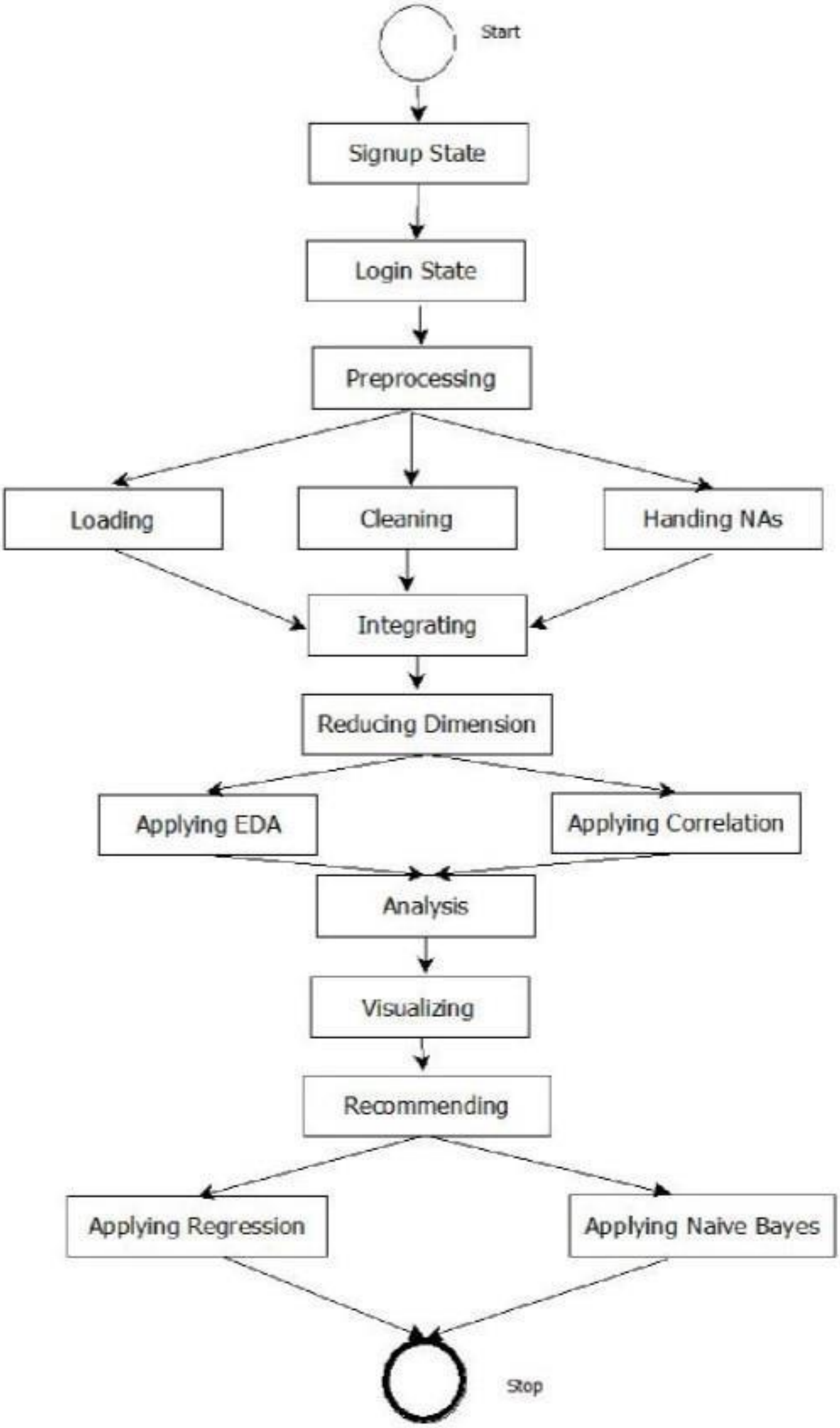


Figure 4.8: State Machine Diagram

Class Diagram: A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

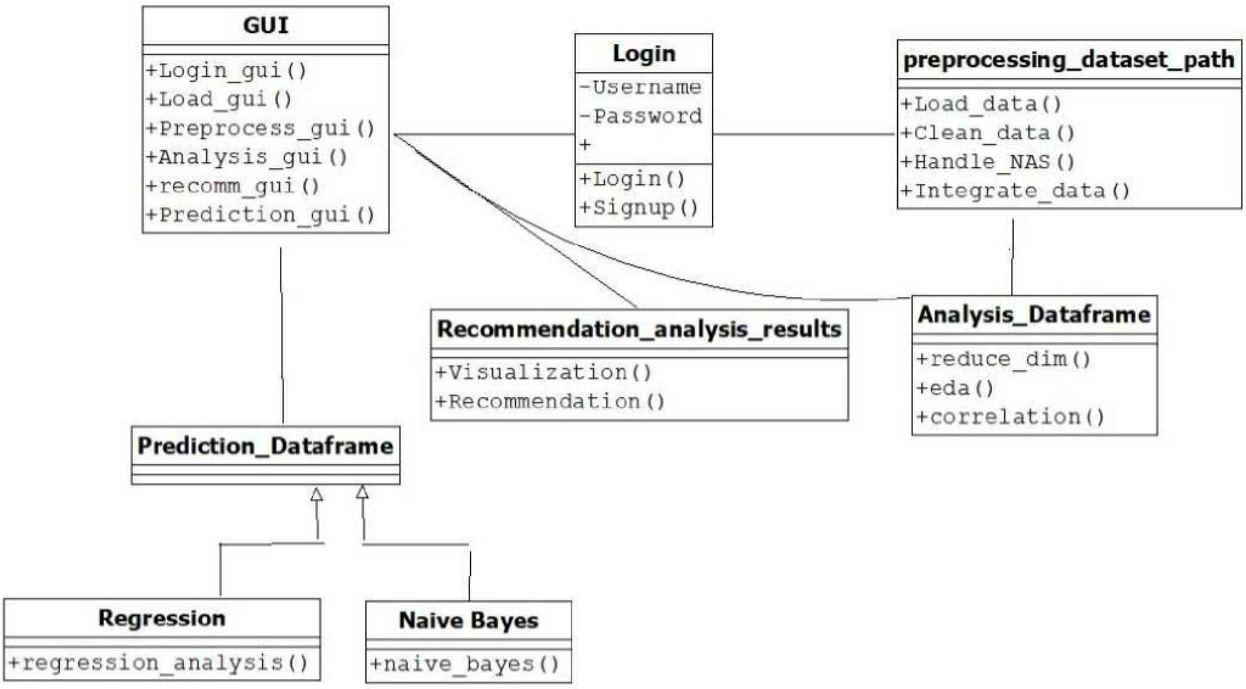


Figure 4.9: Class Diagram

Sequence Diagram: A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

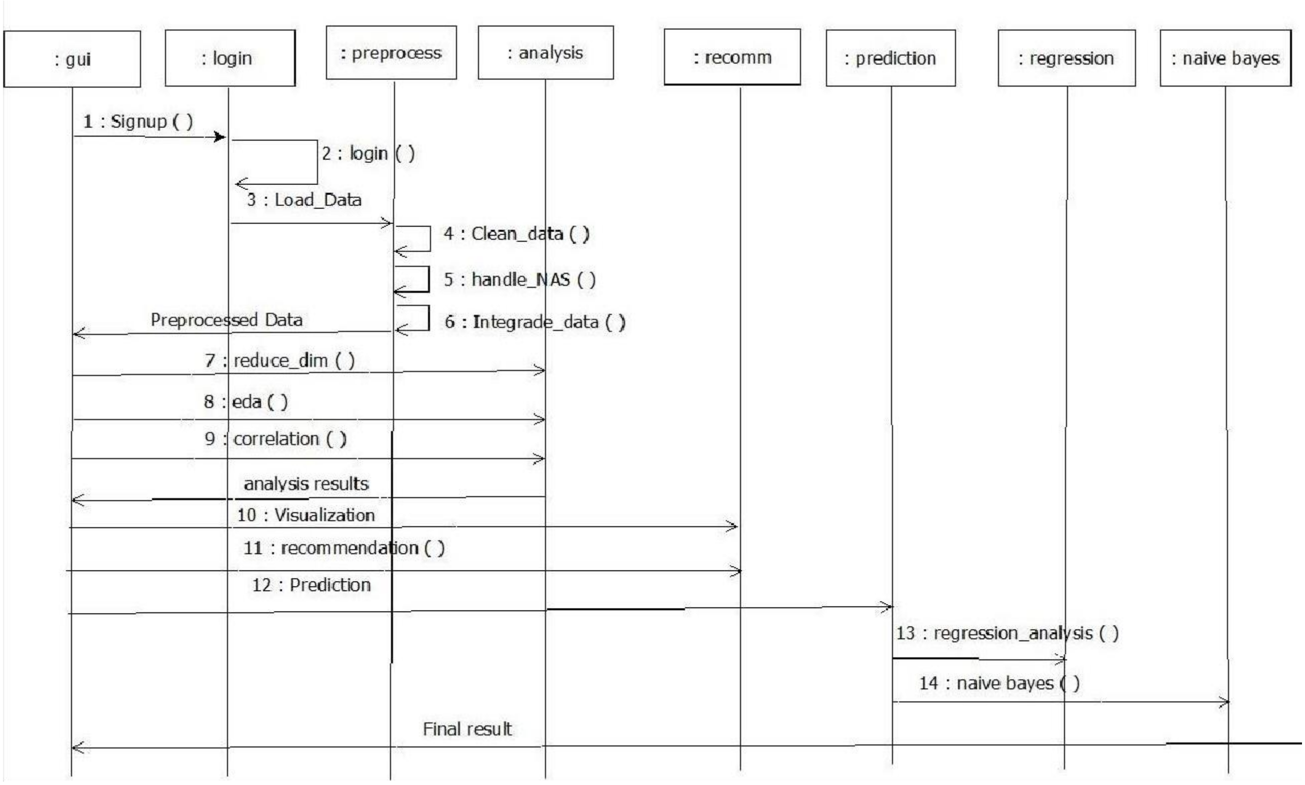


Figure 4.10: Sequence Diagram

Activity Diagram: Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency.

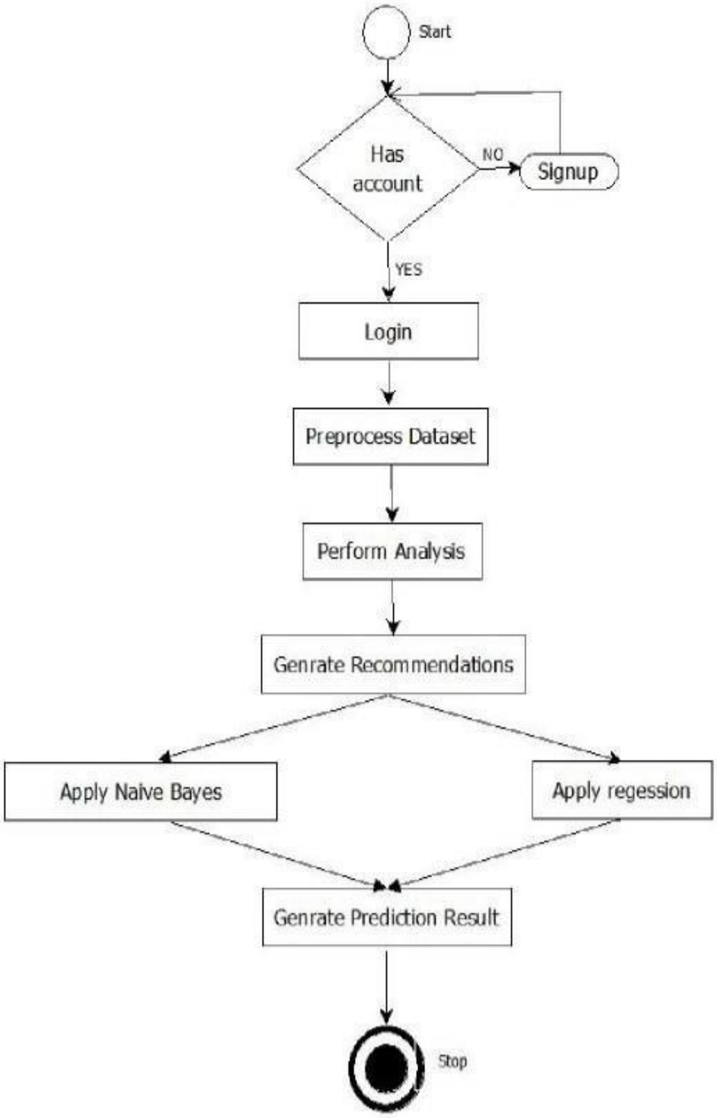


Figure 4.11: Activity Diagram

4.6.2 Architectural Modeling

Component Diagram : Component diagrams are used to model physical aspects of a system. Component diagrams are used to visualize the organization and relationship among components in a system. These diagrams are also used to make executable systems.

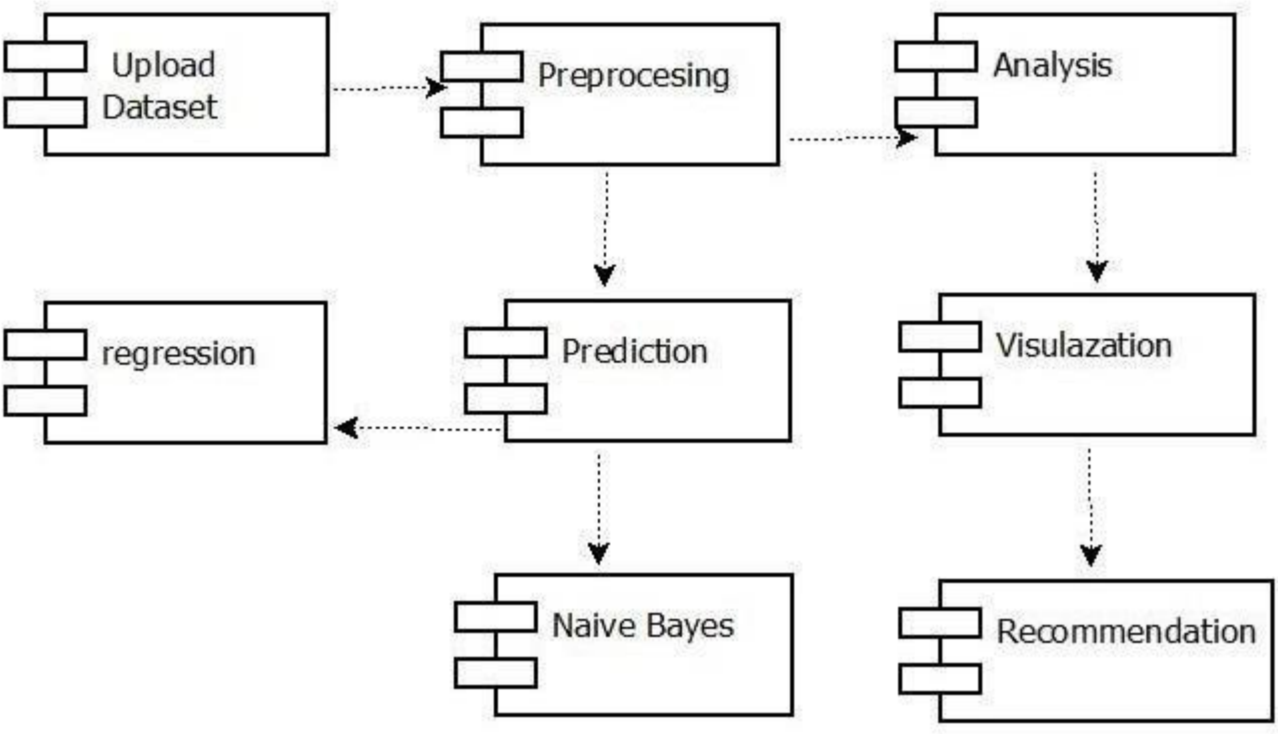


Figure 4.12: Component Diagram

Deployment Diagram : The Deployment Diagram also helps to model the physical aspect of an Object-Oriented software system. It models the run-time configuration in a static view and visualizes the distribution of components in an application.

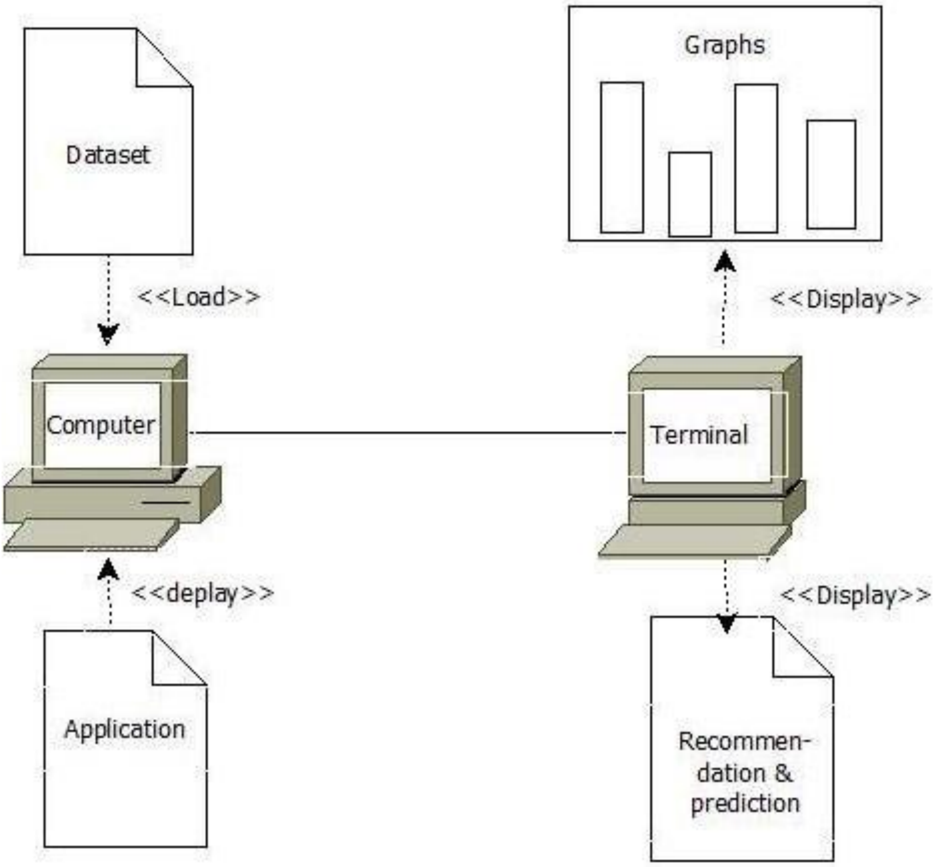


Figure 4.13: Deployment Diagram

4.6.3 Functional Modelling:

Data Flow Diagram : A data flow diagram (DFD) is a graphical representation of the ”flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel. Data flow diagram (DFD) is also called as Bubble Chart is a graphical technique, which is used to represent information flow, and transformers those are applied when data moves from input to output. DFD represents system requirements clearly and identify transformers those becomes programs in design.DFD may further partitioned into different levels to show detailed information flow
e.g. level 0, level 1 etc.

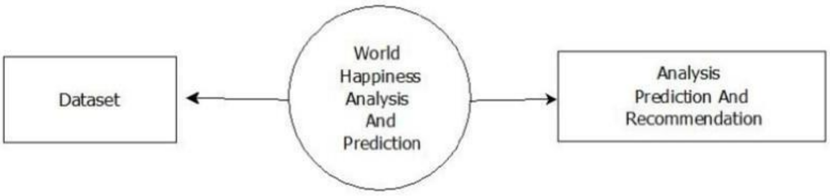

Dataset
World Happiness Analysis And Prediction
Analysis Prediction And Recommendation

Figure 4.14: DFD Level 0

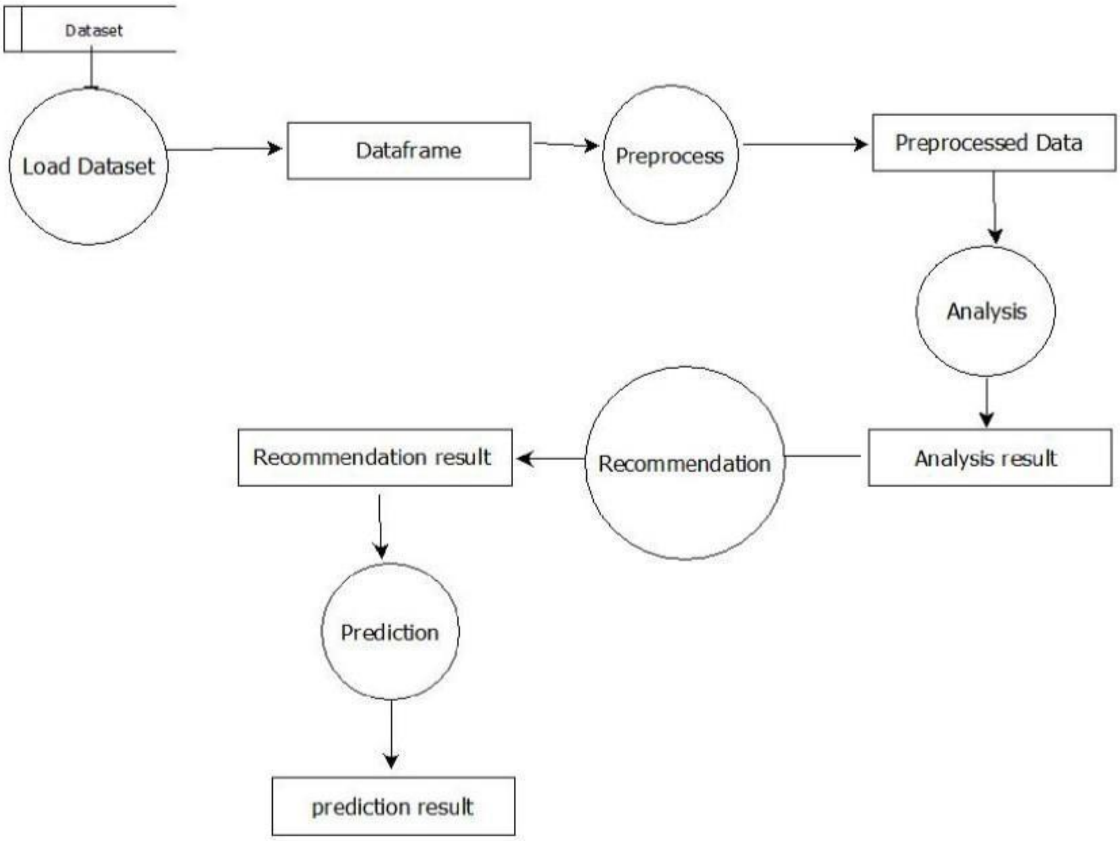


Figure 4.15:DFD Level 1

4.7 Mathematical Modelling:

When solving problems we have to decide the difficulty level of our problem. There are three types of classes provided for that. These are as follows.

1. P Class
2. NP- Hard Class
3.NP- Complete Class

1. P Class

Informally the class P is the class of decision solvable by some algorithm within a number of steps bounded by some fixed polynomial in the length of the input. Turing was not concerned with the efficiency of his machines, but rather his concern was whether they can simulate arbitrary algorithms given sufficient time. However it turns out Turing machines can generally simulate more efficient computer models by at most squaring or cubing the computation time. Thus P is a robust class and has equivalent definitions over a large class of computer models.

2. NP – Hard Class

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Some NP-hard problems are also in NP, some are not. If you could reduce an NP problem to an NP-hard problem and then solve it in polynomial time, you could solve all NP problem. Also, there are decision problems in NP-hard but are not NP complete, such as the halting problems.

3.NP-Complete Class

The complexity class NP-complete is the set of problems that are the hardest problems in NP, in the sense that they are the ones most likely not to be in P. If you can find a way to solve an NP complete problem quick,then you can use that algorithm to solve all NP problems quickly.

Set Theory:-

Set Theory : Set theory is a branch of calculated logic that studies sets, which informally are collections of objects. Although any type of object can be composed into a set, set theory is applied most often to objects that are appropriate to mathematics. The language of set theory can be used in the meanings of nearly all mathematical objects.

Let, System as S set

$$S =(I, P, R, O) ....(1)$$

Where,

I is set of all inputs given to system.

P is set of all processes in system.

R is set of rules that drives your input set.

O is set of output expected from system.

Then Input (I) represented as :

I = {I1,I2}……………… (2)

Where,
I1 : Raw Dataset
I2:History Dataset

Processes (P) is represented as :

P = {P1, P2, P3, P4, P5, P6,P7}……………… (3)

Where,
P1 : Data Preprocessing.
P2 : Data Analysis.
P3 : Data Visualization.
P4 : Predication.
P5: Regression Algorithm.
P6 : Naïve Bayes Algorithm.
P7 : GUI Development.

Rules (R) : is described as :

R = {R1,R2}… ....(4)

Output (O) : is described as :

O = {O1, O2, O3, O4,O5}… .... (5)

Where,
O1:Analysis Findings

O2:Correlation

O3:Graphs

O4:Predication

O5:Recommendation

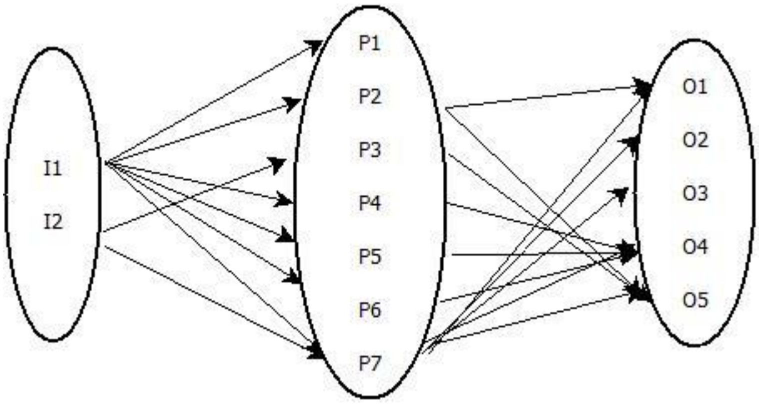


Figure 4.16:Venn Diagram

CHAPTER 5

Risk Management

Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan denotes how risks associated with the project, will be identified, analyzed and managed. It outlines how risk management activities will be performed, recorded and monitored throughout the life cycle of the project and provides templates and practices for recording and prioritizing risks. Risk management is an on-going process that continues through the life of a project. Many of these processes are updated throughout the project life cycle as new risks can be identified at any time. It's the objective of risk management to decrease the probability and impact of events adverse to the project. On the other hand, any event that could have a positive impact should be exploited.

5.1 Risk Identification

Risk details related to proposed project are given below

5.1.1 Product Size Risk

R1 : Project does not complete on time because size of the code and scope of the project is too big.

5.1.2 Business Impact Risk :

R2 : Delay in project delivery, violation of deadline of delivery can impact on customers

economic conditions

R3 : If proposed system is not efficient than the existing system then it may result into economical loss to the customer.

R4 : If we include unrelated code of software into proposed system then it will increase the time for development and complexity of the system.

5.1.3 Customer Related Risk:

R5: If customer demand a certain modification in the proposed system and again requests for modification after completion then it is difficult to change or modify the entire system.

5.1.4 Development Environment Related Risk:

R6: Lack of knowledge and training related to exiting system development environments and new technology could lead difficulties in project completion.

5.1.5 Staff Size Risk:

R7 : Improper initial required training to the staff assigned for the project completion.

R8 : Less Experienced software engineers assigned for project development.

5.1.6 Technical Risks:

R9 : Data is inconsistent.

R10: Data is having missing values.

R11: Operating System failure, Power failure

5.2 Strategies used to manage Risk

- S1: Formulation and follow up of the project plan on regular basis.
- S2: Keep assigned work under certain deadlines.
- S3: Efficient algorithms must be used for implementation of the system.
- S4: Regular meeting with customers reduce the risk to some extent, design system with extensibility and maintain necessary documentation for the same.
- S5: Redefine software process at a higher degree.
- S6: Proper training on required technical tools for development of project reduce risk.
- S7: Ensure that all the members are participating in the design, so that everyone knows and contribute to planning for completion of project.
- S8: Study and understanding of project definition, programming language.
- S9: Data should be carefully selected for the project.
- S10: The method of imputation should be used to handle the missing values in the dataset.
- S11: Take the backup of work on regular basis.

5.3 Risk Projection

5.3.1 Preparing Risk Table:

The risk table shown below lists all possible risks which may occur at any stage during development of project. Table also clearly shows the impact of the risks and RMMM (Risk Mitigation Monitoring and Management) plan to deal with any such risks.

5.3.2 Risk Management Table Along with RMMM Plan

Table 5.1: Risk Management Table Along with RMMM Plan

Risk	Category	Probability	Impact	RMMM Plan
R1	Product Size	More	High	S1
R2	Business Impact	More	High	S2, S11
R3	Business Impact	More	High	S3, S11
R4	Business Impact	Less	High	S6, S9
R5	Customer Related Risk	More	High	S4,S5
R6	Development Environment Related Risk	Less	High	S6, S8
R7	Staff Size Related Risk	More	High	S7
R8	Staff Size Related Risk	Less	High	S9
R9	Technical Risk	Less	Low	S12
R10	Technical Risk	More	High	S3, S10
R11	Technical Risk	Less	High	S13

5.4 Feasibility Study

Feasibility:-Feasibility is defined as an evaluation or analysis of the potential impact of a proposed project.
Technical feasibility:-It is concerned with specifying equipment and software that will successfully support the task required.
SAT (Satisfability):- Boolean formula is satisfability if there exists at least one way of assigning value to its variable so as to make it true and we denote it by using SAT. The problem of deciding whether given formula is satisfability or not.

- Facility to produce output in given times.
- Response time under certain condition.

Operational Feasibility:-It is related to human, organization.

- What changes will be brought in with system.
- How organizational Structure will be distributed.
- what new skill are required.

Economic Feasibility:-It is the most frequently used technique for evaluating the effectiveness of proposed system. Most commonly known as cost/benefit analysis.

CHAPTER 6

Technical Specification

6.1 Software Requirement Specification

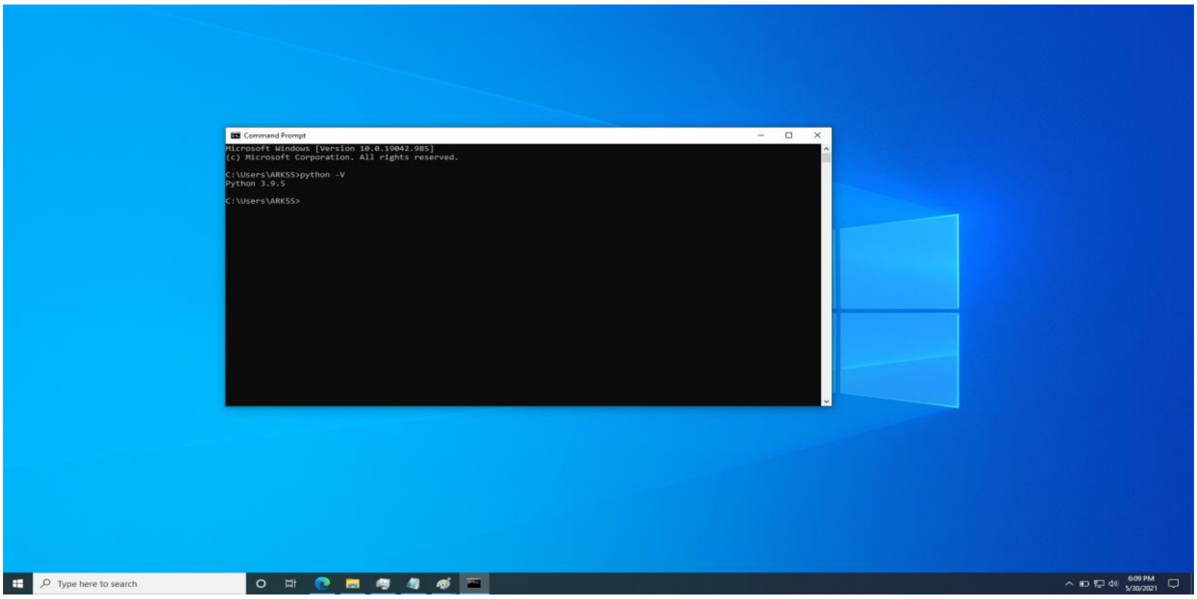
- **Operating System: -**
  We have used windows 10 as the Operating System platform.

- **Languages: -**
  In designing and implementation of this project we have used Python languages.

- **Editors: -**
  In designing and implementation of this project below editoris , used:
  1. Vistual Studio Code: - python code development and testing.

6.1.1 Python Installation Steps

1. Download latest Python language version from https://www.python.org/downloads/
2. Run the Python installer.
3. Install Python silently.
4. Check python version after installation.



```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ARKSS>python -V
Python 3.9.5

C:\Users\ARKSS>
```

Figure 6.4: Python Version

6.1.2 Visual Studio Code Installation Steps

1. Download Visual Studio Code from https://code.visualstudio.com/
2. Run the Visual Studio Code installer.
3. Install Visual Studio silently.
4. After installation the Visual Studio Code will look like this.

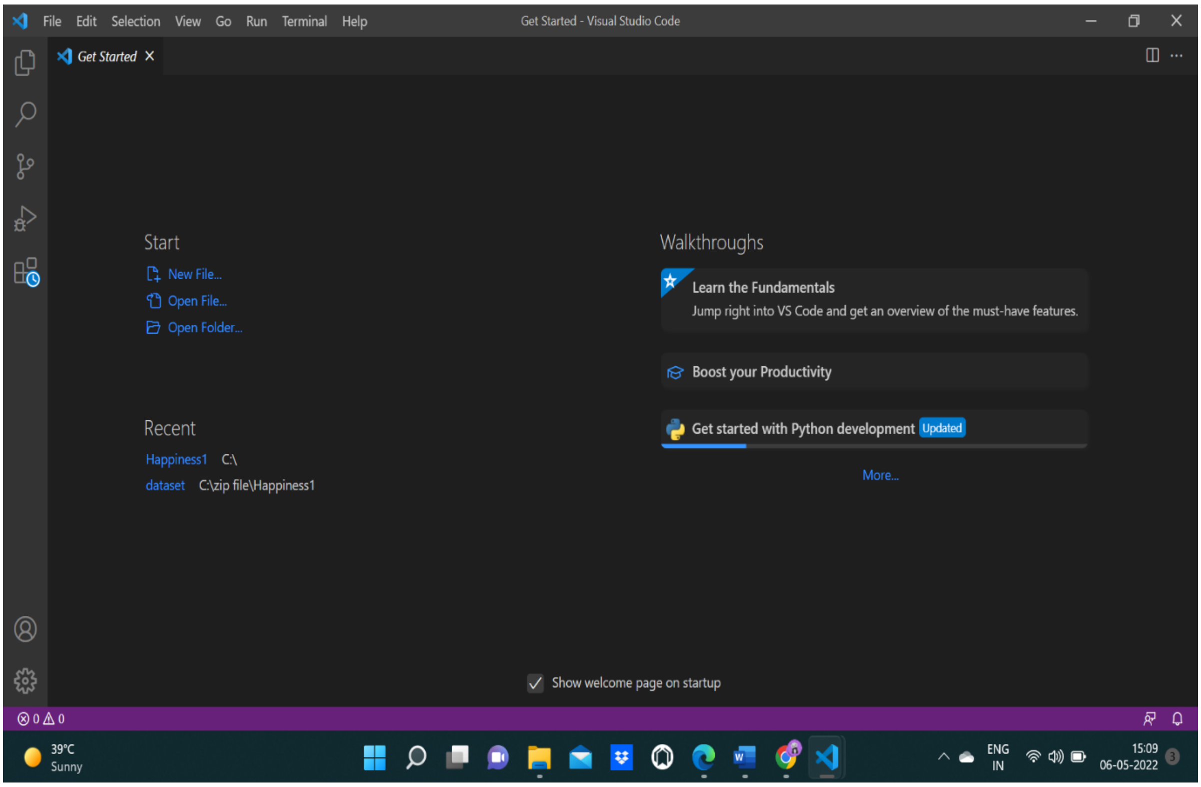

File Edit Selection View Go Run Terminal Help
Get Started - Visual Studio Code
Get Started
Start
New File...
Open File...
Open Folder...
Recent
Happiness1 C:\
dataset C:\zip file\Happiness1
Walkthroughs
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
Boost your Productivity
Get started with Python development Updated
More...
Show welcome page on startup
39°C
Sunny
ENG
IN
15:09
06-05-2022

Figure 6.2: Visual Studio Code

6.2 Hardware Requirement Specification

The algorithms used in the project are computationally intensive and thus require high computingcapabilities in terms of hardware. For the testing and demo purpose we will need good hardware usually found in desktop type of computers. Thus, to reduce the computation time of the project we recommend to use high performance system. Minimum Recommended hardware is:

1. CPU: - Core i3 and above.
2. RAM: - 8 GB and above.
3. Hard Disk: - 256 GB and above.

CHAPTER 7

Implementation

7.1 Introduction

Our project World Happiness Analysis Predication & Recommendation for Happier World is a Desktop Base GUI Application.

1. **Desktop Application: -**

This application is designed using GUI . This application performs all the main tasks.

7.2 Breakdown Structure with modules and sub modules

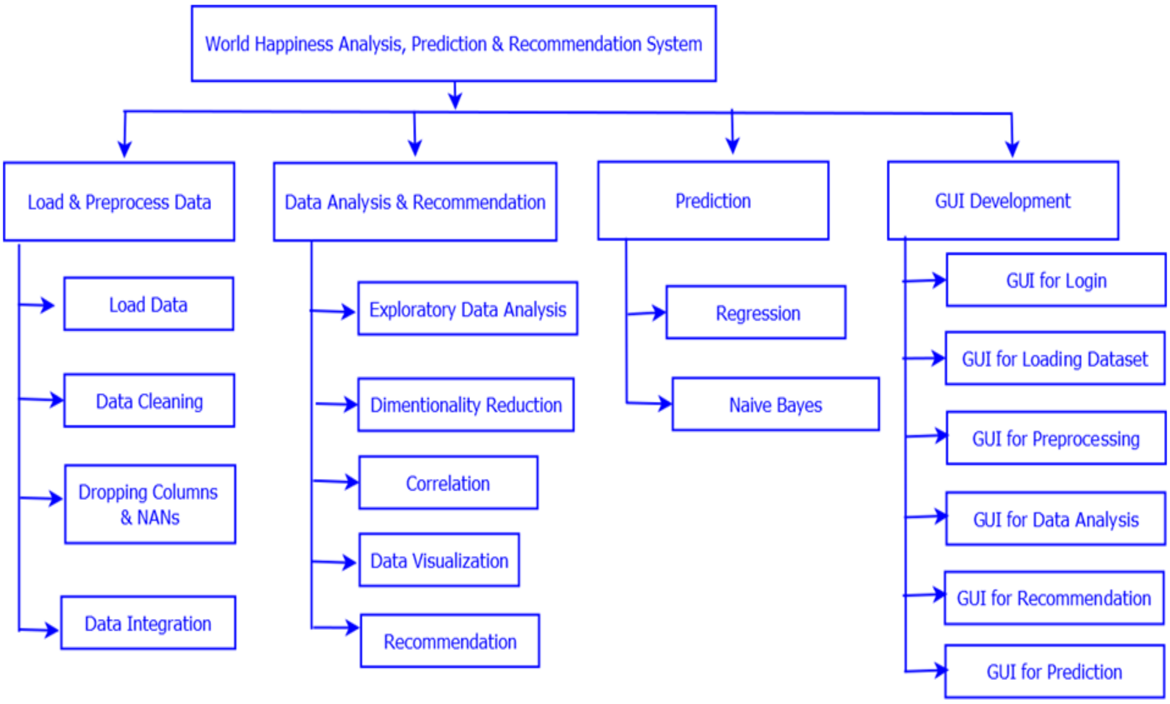


Figure 7.1: Breakdown Structure

7.3 Registration & Login Page

This application will be designed using Python .

7.3.1 Working

In this module/application a user will register first by entering his user ID And Password. After that a user can login with his ID and Password and view the menu.

7.3.2 Major Code

```
#import modules
from tkinter import *
import os
from tkinter import messagebox
import gui

# deletes all the widgets from a frame or window
def clearWindow(win):
    for widget in win.winfo_children():
        widget.destroy()

# Implementing event on register button
def register_user(login_frame):
    username_info = username.get()
    password_info = password.get()

    if(len(username_info)==0 and len(password_info)==0):
        messagebox.showinfo('Input Error', 'Please provide a valid Username & Password')
    elif(len(username_info)==0):
        messagebox.showinfo('Input Error', 'Please provide a valid Username')
    elif(len(password_info)==0):
        messagebox.showinfo('Input Error', 'Please provide a valid Password')
    else:
        file = open('C:\\Happiness\\login' + '\\'+str(username_info), "w")
        file.write(username_info + "\n")
        file.write(password_info)
        file.close()

        username_entry.delete(0, END)
        password_entry.delete(0, END)

        messagebox.showinfo('Success', 'Registration Successful\nPlease Go Back to
Login')
```

```
# Implementing event on login button

def login_verify(login_frame):
    username1 = username_verify.get()
    password1 = password_verify.get()

    username_login_entry.delete(0, END)
    password_login_entry.delete(0, END)

    list_of_files = os.listdir('C:\\Happiness\\login\\')
    if username1 in list_of_files:
        file1 = open('C:\\Happiness\\login\\' + username1, "r")
        verify = file1.read().splitlines()
        if password1 in verify:
            messagebox.showinfo('Success', 'Login Successful')
            clearWindow(login_frame)

            #Label(login_frame, text='Login Successful', bg='white', fg='green',
font='Helvetica 20 bold').pack()

            gui.firstWindow(root, login_frame)

        else:
            messagebox.showinfo('Not Recognised', 'Password Not Recognised')
    else:
        messagebox.showinfo('Not Found', 'User Not Found')

def register(login_frame):
    clearWindow(login_frame)

    global username
    global password
    global username_entry
    global password_entry

    username = StringVar()
    password = StringVar()

    Label(login_frame, text="Register", bg="white", fg='green', font='Arial 18
bold').grid(row=0, column=0, columnspan=2, padx=20, pady=20)

    username_lable = Label(login_frame, text="Username * ", bg='white', fg='green')
    username_lable.grid(row=1, column=0, padx=20, pady=10)

    username_entry = Entry(login_frame, textvariable=username)
    username_entry.grid(row=1, column=1, padx=10, pady=10)

    password_lable = Label(login_frame, text="Password * ", bg='white', fg='green')
    password_lable.grid(row=2, column=0, padx=20, pady=10)
```

```
 password_entry = Entry(login_frame, textvariable=password, show='*')
    password_entry.grid(row=2, column=1, padx=10, pady=10)

    b1 = Button(login_frame, text="Register", width=15, height=1, bg="green", fg='white',
command = lambda:register_user(login_frame))
    b1.grid(row=3, column=1, padx=20, pady=10)

    b2 = Button(login_frame, text="Back to Login", width=15, height=1, bg="green",
fg='white', command = lambda:login(login_frame))
    b2.grid(row=3, column=0, padx=20, pady=10)


# Designing window for login

def login(login_frame):
    clearWindow(login_frame)

    Label(login_frame, text="Login", bg='white', fg='green', font='Arial 18
bold').grid(row=0, column=0, columnspan=2, pady=20)

    global username_verify
    global password_verify

    username_verify = StringVar()
    password_verify = StringVar()

    global username_login_entry
    global password_login_entry

    Label(login_frame, text="Username * ", bg='white', fg='green').grid(row = 1, column =
0, padx=20, pady = 10)

    username_login_entry = Entry(login_frame, textvariable=username_verify)
    username_login_entry.grid(row = 1, column = 1, padx=10, pady = 10)

    Label(login_frame, text="Password * ", bg='white', fg='green').grid(row = 2, column =
0, padx=20, pady = 10)

    password_login_entry = Entry(login_frame, textvariable=password_verify, show= '*')
    password_login_entry.grid(row = 2, column = 1, padx=10, pady = 10)

    Button(login_frame, text="Register", width=15, height=1, bg='green', fg='white',
command = lambda:register(login_frame)).grid(row = 3, column = 0, padx=20, pady = 10)

    Button(login_frame, text="Login", width=15, height=1, bg='green', fg='white', command
= lambda:login_verify(login_frame)).grid(row = 3, column = 1, padx=10, pady = 10)
```

```
def mainWindow():
    global root
    root = Tk()
    root.title("Happiness Analysis")
    root.configure(bg='white')

    titleFrame = LabelFrame(root, width=1200, height=100, borderwidth = 0, fg = 'green',
bg='white')
    titleFrame.grid(row=0, column=0, columnspan=2)
    titleFrame.grid_propagate(0)

    leftFrame = LabelFrame(root, width=400, he ight=400, borderwidth = 0, fg = 'green',
bg='white')
    leftFrame.grid(row=1, column=0)
    leftFrame.grid_propagate(0)

    rightFrame = LabelFrame(root, width=700, height=400, borderwidth = 0, fg = 'green',
bg='white')
    rightFrame.grid(row=1, column=1)
    rightFrame.grid_propagate(0)


    w=Label(titleFrame, text="Happiness Analysis", fg='navy', bg='white', font='Arial 24
bold')
    w.pack(pady=10)

    logo=PhotoImage(file="C:\\Happiness\\images\\sample.png")
    w1=Label(rightFrame, image=logo)
    w1.pack(fill=BOTH, expand=1)

    login(leftFrame)

    #root.eval('tk::PlaceWindow %s center' % root.winfo_pathname(root.winfo_id()))  #
used to disply the window at the center of screen
    root.mainloop()

mainWindow()
exit()
```

7.4 Visualization

This application will be designed using Python .

7.4.1 Working

In this module display the graphical visualization of analysis and recommendation usingthe matplotlib & tkinter library.

7.4.2 Major Code

```
from tkinter import *
import tkinter.font as font
import pandas as pd
from pandastable import Table
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def quitWindow():
    if messagebox.askyesnocancel('Verify', 'Really quit?'):
        messagebox.showwarning('Yes', 'Your Program will Quit')
        exit(0)
    else:
        messagebox.showinfo('No', 'Quit has been cancelled')

def prepareData(happiness_report):
    happiness_report_older = pd.read_csv('C:\\Happiness\\dataset\\world-happiness-report.csv')
    india_1 = happiness_report_older[happiness_report_older['Country name'] == 'India'].reset_index(drop=True)
    india_1 = india_1.drop(['Positive affect','Negative affect'], axis = 1)
    india_1 = india_1.fillna(0)
    happiness_report['year'] = 2021
    india_2 = happiness_report[happiness_report['Country name'] == 'India']
    india_2 = india_2.rename(columns = {'Ladder score':'Life Ladder',
                'Logged GDP per capita':'Log GDP per capita',
                'Healthy life expectancy':'Healthy life expectancy at birth'})
    india_2 = india_2.drop(['Dystopia + residual', 'Regional indicator'], axis = 1)
    global india
    india = pd.concat([india_1, india_2])
    india.reset_index(drop=True, inplace=True)
    india.rename(columns = {'year':'Year'}, inplace=True)
    return india

def ladderScore(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Life Ladder', data=india, marker='o', markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's ladder score over the years", fontsize=18, fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()

def logGDPPerCapita(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Log GDP per capita', data=india, marker='o', markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's Log GDP per capita over the years", fontsize=18, fontweight='bold',
pad=20)
    plt.xlabel('')
    plt.show()

def socialSupport(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Social support', data=india, marker='o', markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's Social support over the years", fontsize=18, fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()

def lifeExpectancy(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Healthy life expectancy at birth', data=india, marker='o',
markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's life expectancy at birth over the years", fontsize=18,
fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()

def freedomToMakeLifeChoices(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Freedom to make life choices', data=india, marker='o',
markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's Freedom to make life choices over the years", fontsize=18,
fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()

def generosity(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Generosity', data=india, marker='o', markersize=10);
sns.set_style('whitegrid')
    sns.despine(left=True)
```

```
  plt.title("India's Generosity over the years", fontsize=18, fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()

def perceptionsOfCorruptions(india):
    plt.figure(figsize=(18, 6))
    sns.lineplot(x='Year', y='Perceptions of corruption', data=india, marker='o', 
markersize=10);
    sns.set_style('whitegrid')
    sns.despine(left=True)
    plt.title("India's Perceptions of corruption over the years", fontsize=18, 
fontweight='bold', pad=20)
    plt.xlabel('')
    plt.show()


def visualiseMenu(root, data):
    helv36 = font.Font(family='Arial', size=12)
    win = Tk()
    win.title('Visualisation')
    prepareData(data)

    b1 = Button(win, text='India’s Ladder Score Over the Years', font=helv36, fg='white', 
bg='green', command=lambda:ladderScore(india))
    b1.pack(fill=X, padx=50, pady=10)

    b2 = Button(win, text='India’s Log GDP per capita Over the Years', font=helv36, 
fg='white', bg='green', command=lambda:logGDPPerCapita(india))
    b2.pack(fill=X, padx=50, pady=10)

    b3 = Button(win, text='India’s Social Support Over the years', font=helv36, 
fg='white', bg='green', command=lambda:socialSupport(india))
    b3.pack(fill=X, padx=50, pady=10)

    b4 = Button(win, text='India’s Life Expectancy At Birth Over the years', font=helv36, 
fg='white', bg='green', command=lambda:lifeExpectancy(india))
    b4.pack(fill=X, padx=50, pady=10)

    b5 = Button(win, text='India’s Freedom To Make Life Choices Over the years', 
font=helv36, fg='white', bg='green', command=lambda:freedomToMakeLifeChoices(india))
    b5.pack(fill=X, padx=50, pady=10)

    b6 = Button(win, text='India’s Generosity Over the years', font=helv36, fg='white', 
bg='green', command=lambda:generosity(india))
    b6.pack(fill=X, padx=50, pady=10)

    b7 = Button(win, text='India’s Perceptions Of Corruption Over the years', 
font=helv36, fg='white', bg='green', command=lambda:perceptionsOfCorruptions(india))
    b7.pack(fill=X, padx=50, pady=10)
```

7.5 Multiple Linear Regression

7.5.1 Working

Regression Algorithms will be used to predict certain parameters of happiness which are dependent on one or more other parameters.

7.5.2 Major Code

```
 def back(current, previous):
    current.withdraw()
    previous.deiconify()

def dropColumns(filepath):
    data = pd.read_csv(filepath)
    global happiness_report
    happiness_report = data.drop(['Standard error of ladder score', 'upperwhisker',
'lowerwhisker','Explained by: Log GDP per capita', 'Explained by: Social support',
'Explained by: Healthy life expectancy',
'Explained by: Freedom to make life choices',
'Explained by: Generosity', 'Explained by: Perceptions of corruption', 'Ladder score in
Dystopia'], axis=1)


def predictScore(predWin, filepath):
    a = float(logged_gdp_per_capita_entry.get())
    b = float(social_support_entry.get())
    c = float(healthy_life_expectancy_entry.get())
    d = float(freedom_to_make_life_choices_entry.get())
    e = float(generosity_entry.get())
    f = float(perceptions_of_corruption_entry.get())
    g = float(dystopia_residual_entry.get())

    # Input: Logged GDP per capita, Social support, Healthy life expectancy, Freedom to make
life choices, Generosity, Perceptions of corruption, Dystopia + residual

    prediction = lm.predict([[a,b,c,d,e,f,g]])

    comparison.regression_prediction = prediction

    print('The Predicted Happiness Score is : ', prediction)
    print(type(prediction))
    msg = 'The Predicted Happiness Score is : ' + str(prediction)
    Label(predWin, text=msg, bg='white', fg='blue', font='Times 24 bold').grid(row=10,
column=0, columnspan=2, pady=20)
```

```
def clearAll():
    logged_gdp_per_capita_entry.delete(0, END)
    social_support_entry.delete(0, END)
    healthy_life_expectancy_entry.delete(0, END)
    freedom_to_make_life_choices_entry.delete(0, END)
    generosity_entry.delete(0, END)
    perceptions_of_corruption_entry.delete(0, END)
    dystopia_residual_entry.delete(0, END)

def predictionWindow(root, filepath):
    predWin = Tk()
    predWin.title("Prediction")
    predWin.configure(bg='white')

    l1 = Label(predWin, text="Prediction of Happiness/Ladder Score using Multiple Linear
Regression", bg='white', fg='green', font='Arial 18 bold')
    l1.grid(row=0, column=0, columnspan=2, padx=20, pady=20)

    global logged_gdp_per_capita
    global social_support
    global healthy_life_expectancy
    global freedom_to_make_life_choices
    global generosity
    global perceptions_of_corruption
    global dystopia_residual

    logged_gdp_per_capita = StringVar()
    social_support = StringVar()
    healthy_life_expectancy = StringVar()
    freedom_to_make_life_choices = StringVar()
    generosity = StringVar()
    perceptions_of_corruption = StringVar()
    dystopia_residual = StringVar()

    global logged_gdp_per_capita_entry
    global social_support_entry
    global healthy_life_expectancy_entry
    global freedom_to_make_life_choices_entry
    global generosity_entry
    global perceptions_of_corruption_entry
    global dystopia_residual_entry

    Label(predWin, text="Logged GDP per Capita * ", bg='white', fg='blue').grid(row=1,
column=0, padx=20, pady=10)
    logged_gdp_per_capita_entry = Entry(predWin, textvariable=logged_gdp_per_capita)
    logged_gdp_per_capita_entry.grid(row=1, column=1, padx=10, pady=10)
```

```
Label(predWin, text="Social Support * ", bg='white', fg='blue').grid(row=2, column=0,
padx=20, pady=10)
    social_support_entry = Entry(predWin, textvariable=social_support)
    social_support_entry.grid(row=2, column=1, padx=10, pady=10)

    Label(predWin, text="Healthy Life Expectancy * ", bg='white', fg='blue').grid(row=3,
column=0, padx=20, pady=10)
    healthy_life_expectancy_entry = Entry(predWin, textvariable=healthy_life_expectancy)
    healthy_life_expectancy_entry.grid(row=3, column=1, padx=10, pady=10)

    Label(predWin, text="Freedom to Make Life Choices * ", bg='white',
fg='blue').grid(row=4, column=0, padx=20, pady=10)
    freedom_to_make_life_choices_entry = Entry(predWin,
textvariable=freedom_to_make_life_choices)
    freedom_to_make_life_choices_entry.grid(row=4, column=1, padx=10, pady=10)

    Label(predWin, text="Generosity * ", bg='white', fg='blue').grid(row=5, column=0,
padx=20, pady=10)
    generosity_entry = Entry(predWin, textvariable=generosity)
    generosity_entry.grid(row=5, column=1, padx=10, pady=10)

    Label(predWin, text="Perceptions of Corruption * ", bg='white',
fg='blue').grid(row=6, column=0, padx=20, pady=10)
    perceptions_of_corruption_entry = Entry(predWin,
textvariable=perceptions_of_corruption)
    perceptions_of_corruption_entry.grid(row=6, column=1, padx=10, pady=10)

    Label(predWin, text="Dystopia / Residual * ", bg='white', fg='blue').grid(row=7,
column=0, padx=20, pady=10)
    dystopia_residual_entry = Entry(predWin, textvariable=dystopia_residual)
    dystopia_residual_entry.grid(row=7, column=1, padx=10, pady=10)

    Button(predWin, text="Predict", width=15, height=1, bg='green', fg='white',
           command=lambda: predictScore(predWin, filepath)).grid(row=9, column=0,
padx=20, pady=10)

    Button(predWin, text="Clear All", width=15, height=1, bg='green', fg='white',
            command  = clearAll).grid(row=9, column=1, padx=10, pady=10)

    predWin.eval('tk::PlaceWindow %s center' %
predWin.winfo_pathname(predWin.winfo_id()))
    predWin.mainloop()


def displayCoefficients():
    coefWin = Tk()
    coefWin.title('Display Coefficients')

    print('Estimated Intercept is: ', lm.intercept_)
    print('No of coefficients in this model are: ', lm.coef_)
    coef = zip(X.columns, lm.coef_)
    coef_df = pd.DataFrame(list(coef), columns=['features', 'coefficients'])
    print(coef_df)

    f = LabelFrame(coefWin, text='Features & their Coefficients', borderwidth = 5, fg =
'blue') # Create a frame
    f.pack(fill=BOTH,expand=1)
    t = Table(f, dataframe=coef_df)
    t.show()
    coefWin.eval('tk::PlaceWindow %s center' %
coefWin.winfo_pathname(coefWin.winfo_id()))
    coefWin.mainloop()


def createModel(filepath):
    dropColumns(filepath)
    # print(happiness_report)

    dropped_happiness_report = happiness_report.drop(['Country name', 'Regional
indicator'], axis=1)
    #print(dropped_happiness_report)
    global X, Y
    X = dropped_happiness_report.drop('Ladder score', axis=1) # All columns except Ladder
Score
    Y = dropped_happiness_report['Ladder score'] # Ladder Score Column

    global lm
    lm = LinearRegression()
    messagebox.showinfo('Model Created', 'Regression Model Successfully Trained')

def trainModel():
    lm.fit(X, Y)
    messagebox.showinfo('Model Trained', 'Regression Model Training Successful')

def plotRegressionLine():
    Y_pred = lm.predict(X)
    plt.title('Regression Line')
    plt.scatter(X, Y)
    plt.plot(X, Y_pred, color='red')
    plt.show()

def calculateAccuracy():
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state
= 45)

    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy1 = 100.0 - np.mean(mean_absolute_percentage_error(y_test,y_pred))
    comparison.regression_accuracy = accuracy1

    #print('Accuracy:', round(accuracy, 2), '%.')
    #s = 'Accuracy of Regression Model: ' + str(round(accuracy, 2)) + '%.'
    s1 = 'Accuracy of Regression Model: ' + str(accuracy1) + ' %'
    messagebox.showinfo('Percentage Accuracy', s1)

def regressionOptions(root, filepath):
    root.withdraw()
    win =Tk()
    win.title('Multiple Regression Algorithm')
    helv36 = font.Font(family='Arial', size=12)

    b1 = Button(win, text='Create Regression Model', font=helv36, fg='white', bg='green',
command=lambda:createModel(filepath))
    b1.pack(fill=X, padx=50, pady=10)

    b2 = Button(win, text='Train the Model', font=helv36, fg='white', bg='green',
command=trainModel)
    b2.pack(fill=X, padx=50, pady=10)

    #b3 = Button(win, text='Plot Regression Line', font=helv36, fg='white', bg='green',
command=plotRegressionLine)
    #b3.pack(fill=X, padx=50, pady=10)

    b4 = Button(win, text='Display Coefficents', font=helv36, fg='white', bg='green',
command=displayCoefficients)
    b4.pack(fill=X, padx=50, pady=10)

    b5 = Button(win, text='Calculate Model Accuracy', font=helv36, fg='white',
bg='green', command=calculateAccuracy)
    b5.pack(fill=X, padx=50, pady=10)

    b6 = Button(win, text='Prediction of Happiness Score', font=helv36, fg='white',
bg='green', command=lambda:predictionWindow(root, filepath))
    b6.pack(fill=X, padx=50, pady=10)

    b7 = Button(win, text='<-- Back to Main Menu', fg='white', bg='green',
command=lambda:back(win, root))
    b7.pack(side=LEFT, expand=1)

    win.eval('tk::PlaceWindow %s center' % win.winfo_pathname(win.winfo_id()))  # used to
disply the window at the center of screen
    win.mainloop()
```

7.6 Naïve Bayes algorithm

7.6.1 Working

In order to get a baseline accuracy rate for our data, we are using a Naive Bayes Algorithm.Specifically, we used the scikit-learn implementation of Gaussian Naive Bayes.

7.6.1 Major Code

```
def back(current, previous):
    current.withdraw()
    previous.deiconify()



def selectPredictors(data):
    # Split the Predictors and Prediction Attributes
    # Column No 4 to 9 are predictors, i.e. Logged GDP per capita, Social support, Healthy
life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption
    # Column No 3 is Prediction i.e. Ladder Score (happiness index)

    X = data.iloc[:, 4:11].values
    Y = data.iloc[:, 3].values
    X = X.astype('int')
    Y = Y.astype('int')
    print(X)
    print(Y)
    return X, Y

def calculateAccuracy():
    accuracy = 100 - np.mean(mean_absolute_percentage_error(y_test,y_pred))
    print('Accuracy:', round(accuracy, 2), '%.')
    s = 'Accuracy of Regression Model: ' + str(round(accuracy, 2)) + '%.'
    messagebox.showinfo('Percentage Accuracy', s)


def createModel(filepath):

dropColumns(filepath)

    X, Y = selectPredictors(data)

    global X_train, X_test, y_train, y_test, y_pred
    # Split the dataset into train and test dataset
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state =
45)

    print(X_train.shape, y_train.shape)
    print(type(X_train), type(y_train))
    messagebox.showinfo('Create Model', 'Naive Bayes model is created')


def trainModel():
    # Train Model
    global model
    model = GaussianNB()
    model.fit(X_train, y_train)
    messagebox.showinfo('Train Model', 'Naive Bayes model is trained')

def calculateAccuracy():
    y_pred = model.predict(X_test)
    print(y_pred)
    accuracy = 100 * metrics.accuracy_score(y_test, y_pred)
    comparison.naivebayes_accuracy = accuracy

    print("Accuracy:", accuracy)
    s = 'Accuracy of Naive Bayes Model : ' + str(accuracy) + ' %'
    messagebox.showinfo('Accuracy', s)

def predictScore(predWin, filepath):
    a = float(logged_gdp_per_capita_entry.get())
    b = float(social_support_entry.get())
    c = float(healthy_life_expectancy_entry.get())
    d = float(freedom_to_make_life_choices_entry.get())
    e = float(generosity_entry.get())
    f = float(perceptions_of_corruption_entry.get())
    g = float(dystopia_residual_entry.get())

    # Prediction
    # Input: Logged GDP per capita, Social support, Healthy life expectancy, Freedom to make
life choices, Generosity, Perceptions of corruption, Dystopia + residual

    #prediction = model.predict([[10.77, 0.95, 72, 0.94, -0.098, 0.186, 3.2]])

    prediction = model.predict([[a,b,c,d,e,f,g]])
    print(prediction)
    comparison.naive_bayes_prediction = prediction

    print('The Predicted Happiness Score is : ', prediction)
    print(type(prediction))
    msg = 'The Predicted Happiness Score is : ' + str(prediction)
    Label(predWin, text=msg, bg='white', fg='blue', font='Times 24 bold').grid(row=10,
column=0, columnspan=2, pady=20)

def clearAll():
    logged_gdp_per_capita_entry.delete(0, END)
    social_support_entry.delete(0, END)
    healthy_life_expectancy_entry.delete(0, END)
    freedom_to_make_life_choices_entry.delete(0, END)
    generosity_entry.delete(0, END)
    perceptions_of_corruption_entry.delete(0, END)
    dystopia_residual_entry.delete(0, END)

def predictionWindow(root, filepath):
    predWin = Tk()
    predWin.title("Prediction Using Naive Bayes")
    predWin.configure(bg='white')

    Label(predWin, text="Prediction of Happiness/Ladder Score using Naive Bayes
Algorithm", bg='white', fg='green', font='Arial 18 bold').grid(row=0, column=0,
columnspan=2, pady=20)

    global logged_gdp_per_capita
    global social_support
    global healthy_life_expectancy
    global freedom_to_make_life_choices
    global generosity
    global perceptions_of_corruption
    global dystopia_residual

    logged_gdp_per_capita = StringVar()
    social_support = StringVar()
    healthy_life_expectancy = StringVar()
    freedom_to_make_life_choices = StringVar()
    generosity = StringVar()
    perceptions_of_corruption = StringVar()
    dystopia_residual = StringVar()

    global logged_gdp_per_capita_entry
    global social_support_entry
    global healthy_life_expectancy_entry
    global freedom_to_make_life_choices_entry
    global generosity_entry
    global perceptions_of_corruption_entry
    global dystopia_residual_entry

    Label(predWin, text="Logged GDP per Capita * ", bg='white', fg='green').grid(row=1,
column=0, padx=20, pady=10)
    logged_gdp_per_capita_entry = Entry(predWin, textvariable=logged_gdp_per_capita)
    logged_gdp_per_capita_entry.grid(row=1, column=1, padx=10, pady=10)

    Label(predWin, text="Social Support * ", bg='white', fg='green').grid(row=2,
column=0, padx=20, pady=10)
    social_support_entry = Entry(predWin, textvariable=social_support)
    social_support_entry.grid(row=2, column=1, padx=10, pady=10)

    Label(predWin, text="Healthy Life Expectancy * ", bg='white', fg='green').grid(row=3,
column=0, padx=20, pady=10)
    healthy_life_expectancy_entry = Entry(predWin, textvariable=healthy_life_expectancy)
    healthy_life_expectancy_entry.grid(row=3, column=1, padx=10, pady=10)

    Label(predWin, text="Freedom to Make Life Choices * ", bg='white',
fg='green').grid(row=4, column=0, padx=20, pady=10)
```

```
    freedom_to_make_life_choices_entry = Entry(predWin,
textvariable=freedom_to_make_life_choices)
    freedom_to_make_life_choices_entry.grid(row=4, column=1, padx=10, pady=10)

    Label(predWin, text="Generosity * ", bg='white', fg='green').grid(row=5, column=0,
padx=20, pady=10)
    generosity_entry = Entry(predWin, textvariable=generosity)
    generosity_entry.grid(row=5, column=1, padx=10, pady=10)

    Label(predWin, text="Perceptions of Corruption * ", bg='white',
fg='green').grid(row=6, column=0, padx=20, pady=10)
    perceptions_of_corruption_entry = Entry(predWin,
textvariable=perceptions_of_corruption)
    perceptions_of_corruption_entry.grid(row=6, column=1, padx=10, pady=10)

    Label(predWin, text="Dystopia / Residual * ", bg='white', fg='green').grid(row=7,
column=0, padx=20, pady=10)
    dystopia_residual_entry = Entry(predWin, textvariable=dystopia_residual)
    dystopia_residual_entry.grid(row=7, column=1, padx=10, pady=10)

    Button(predWin, text="Predict", width=15, height=1, bg='green', fg='white',
           command=lambda: predictScore(predWin, filepath)).grid(row=9, column=0,
padx=20, pady=10)

    Button(predWin, text="Clear All", width=15, height=1, bg='green', fg='white',
            command  = clearAll).grid(row=9, column=1, padx=10, pady=10)

    predWin.eval('tk::PlaceWindow %s center' %
predWin.winfo_pathname(predWin.winfo_id()))  # used to disply the window at the center of
screen
    predWin.mainloop()

def naiveBayesOptions(root, filepath):
    root.withdraw()
    win =Tk()
    win.title('Naive Bayes Algorithm')
    helv36 = font.Font(family='Arial', size=12)

    b1 = Button(win, text='Create Naive Bayes Model', font=helv36, fg='white',
bg='green', command=lambda:createModel(filepath))
    b1.pack(fill=X, padx=50, pady=10)

    b2 = Button(win, text='Train the Model', font=helv36, fg='white', bg='green',
command=trainModel)
    b2.pack(fill=X, padx=50, pady=10)

    #b3 = Button(win, text='Plot Regression Line', font=helv36, fg='white', bg='green',
command=plotRegressionLine)
    #b3.pack(fill=X, padx=50, pady=10)

    #b4 = Button(win, text='Display Coefficents', font=helv36, fg='white', bg='green',
command=displayCoefficients)
```

```
    #b4.pack(fill=X, padx=50, pady=10)

    b5 = Button(win, text='Calculate Model Accuracy', font=helv36, fg='white',
bg='green', command=calculateAccuracy)
    b5.pack(fill=X, padx=50, pady=10)

    b6 = Button(win, text='Prediction of Happiness Score using Naive Bayes', font=helv36,
fg='white', bg='green', command=lambda:predictionWindow(root, filepath))
    b6.pack(fill=X, padx=50, pady=10)

    b7 = Button(win, text='<-- Back to Main Menu', fg='white', bg='green',
command=lambda:back(win, root))
    b7.pack(side=LEFT, expand=1)

    win.eval('tk::PlaceWindow %s center' % win.winfo_pathname(win.winfo_id()))  # used to
disply the window at the center of screen
    win.mainloop()
```

CHAPTER 8

Testing

8.1 Introduction

In general, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. Software testing, depending on the testing method employed, can be implemented at any time in the development process. Software testing can be stated as the process of validating and verifying that a software program/application/product. The types of testing can be explained as below.

8.1.1 Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developers to modify the source code without immediate concerns about how such changes might affect the functioning of other units or the program as a whole. Once all of the units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by means of integration testing. Unit testing can be time consuming and tedious.

8.1.2 Integration Testing

Integration Testing is a level of software testing where individual modules are combined and testedas a group. Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network etc. This type of testing is especially relevant to client-server and distributed system. Integration testing occurs after unit testing and before validation testing.Integration testing takes as its input modules thathave been unit tested, groups the min aggregates,applies tests define those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.There are two major ways of carrying out an integration test, called the bottom-up method and the top down method. Bottom-up integration testing begins with unit testing, followed by tests of of progressivelyhigher level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.

8.1.3 Performance Testing

Performance testing is defined as a type of software testing to ensure software application will per- form well under there expected workload.Features and functionality supported by a software systemis not the only concern.A software applications performance like its response time,reliability,resource usage and scalability do matter Performance testing, a nonfunctional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing can help identify the nature or location of a software-related performance prob- lem by highlighting where an application might fail or lag. Performance testing can also verify that a system meets the specifications claimed by its manufacturer or vendor. The process can be used tocompare two or more devices or programs.

8.2 Test Cases

8.2.1Test Cases for Application

ID.	Action	Expected Result	Actual Result	Result
TA-01.	Registration	View success message	Success message visible	Pass
TA-02.	Login	View menu	Menu visible	Pass
TA-03.	Load Dataset	Dataset Load Successfully	Success message Visible	Pass
TA-04.	Visualization	Display the graph	Graph Visible	Pass
TA-05.	History Report	Generate India History Report	History Report Generated	Pass
TA-06.	Multiple Linear Regression	Give Prediction & Accuracy of Module	Display Prediction & Accuracy of Module	Pass
TA-07.	Naïve Bayes	Give Prediction & Accuracy of Module	Display Prediction & Accuracy of Module	Pass
TA-08.	Comparison of Multiple Linear Regression & Naïve Bayes	Display Comparison of Multiple Linear Regression & Naïve Bayes	Multiple Linear Regression & Naïve Bayes Comparison Visible	Pass

CHAPTER 9

Experimental Analysis and Result Snapshot

9.1 Experimental Analysis

The experimental analysis can be explained using training and testing process of the Project.

1.Training the multiple regression model, testing the model and calculating the accuracy.

- **Multiple Regression**

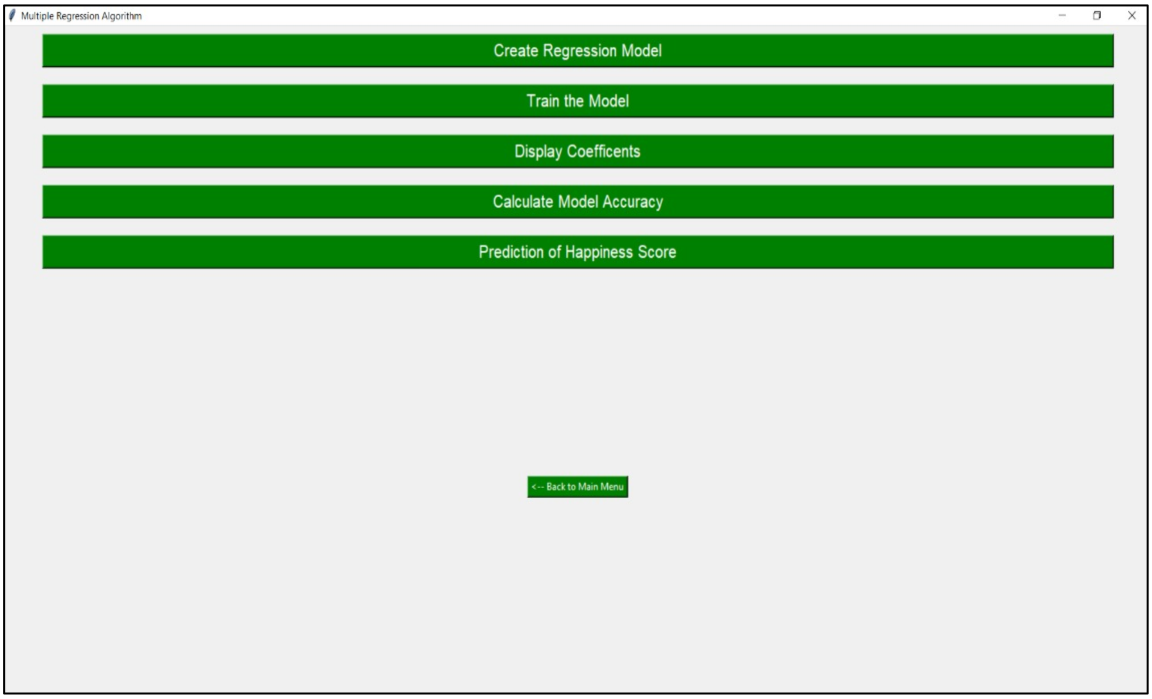


Figure 9.1.1: Multiple Linear Regression Module

2.Training the Naïve bayes model, testing the model and calculating the accuracy.

Naïve Bayes

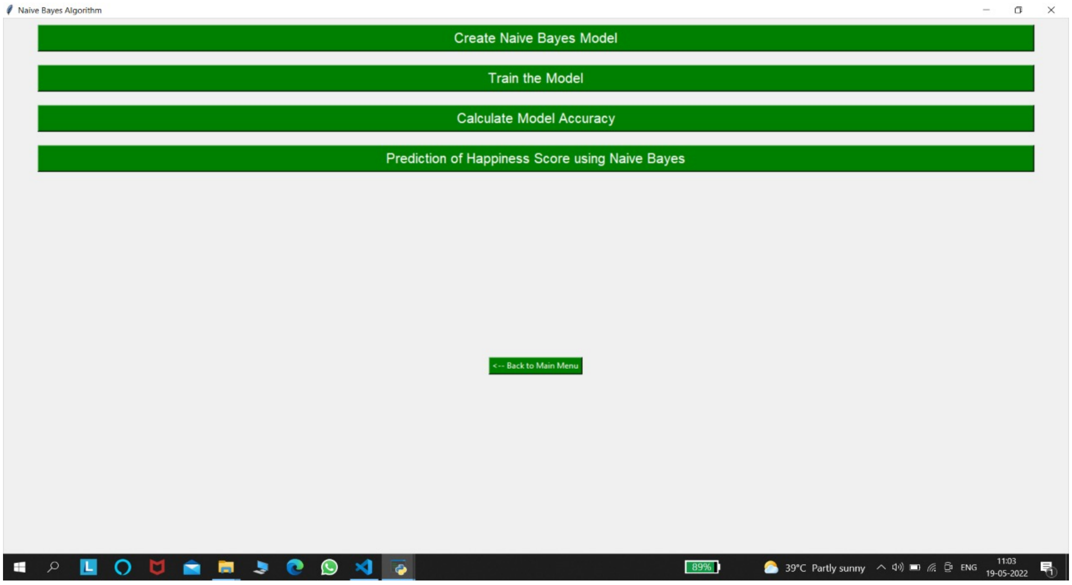
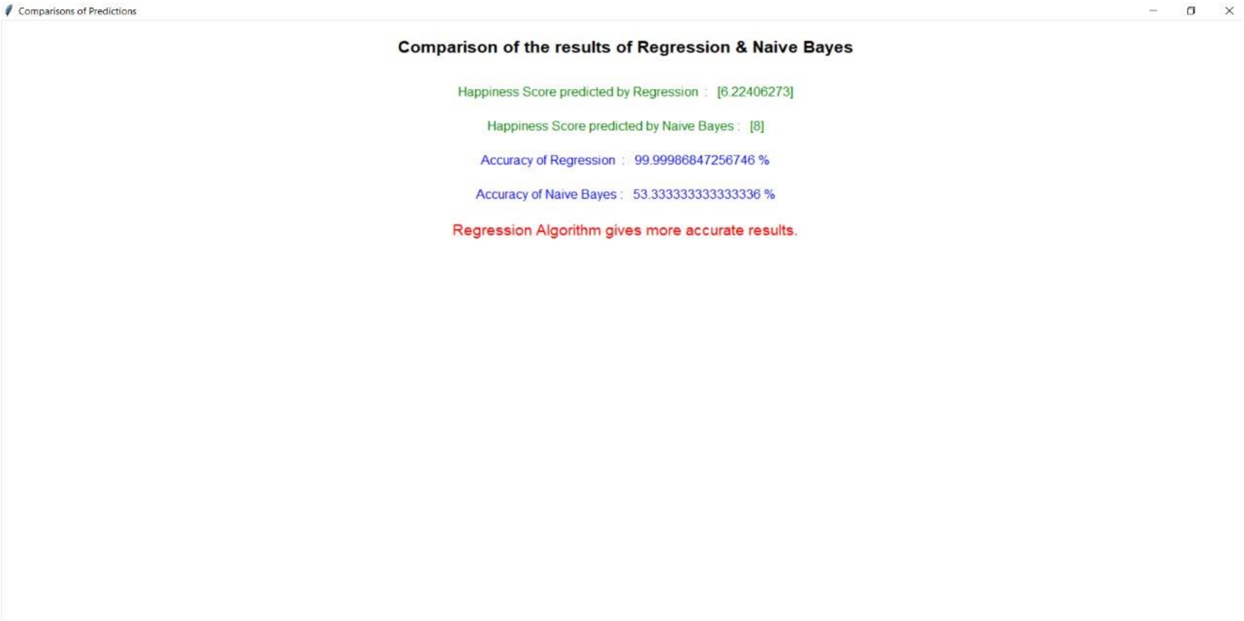


Figure 9.1.2: Naïve Bayes Module

3.Comparison of Multiple Linear Regression & Naïve Bayes Module using Accuracy.



Comparison of the results of Regression & Naive Bayes

Happiness Score predicted by Regression : [6.22406273]

Happiness Score predicted by Naive Bayes : [8]

Accuracy of Regression : 99.99986847256746 %

Accuracy of Naive Bayes : 53.333333333333336 %

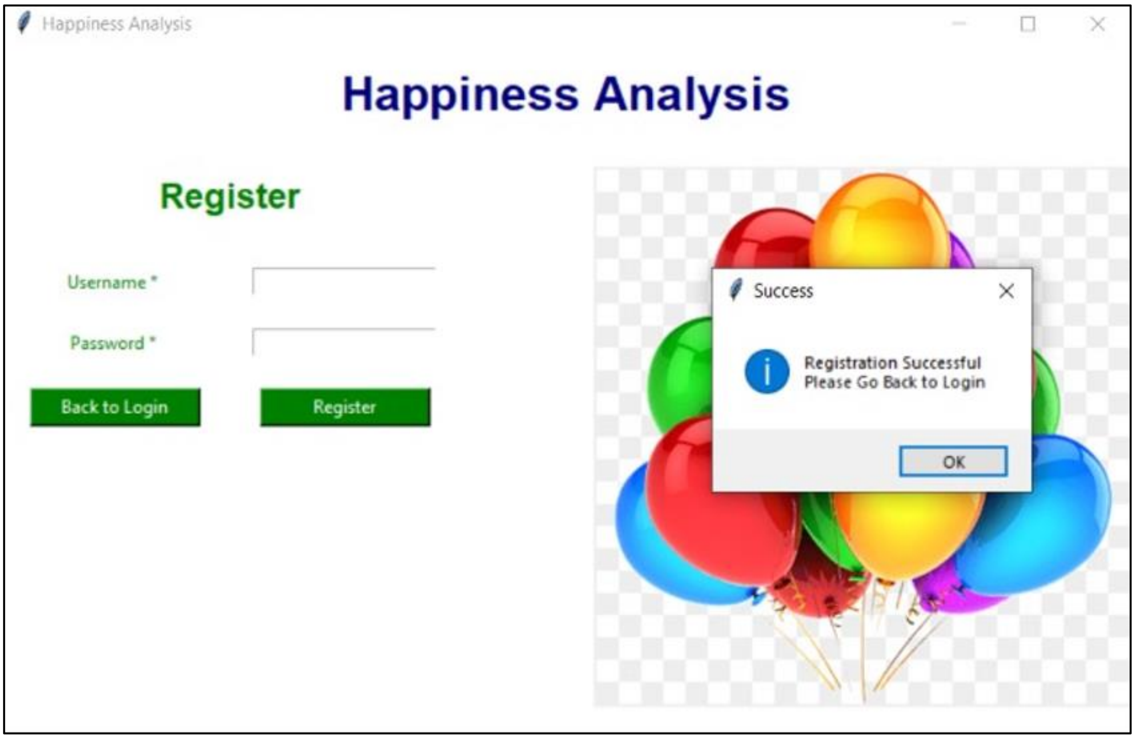
Regression Algorithm gives more accurate results.

Figure 9.1.3: Comparison of Module

9.2 Project Snapshots/Results

The project snapshots/results are shown in sequence in which the whole project workflow take place.

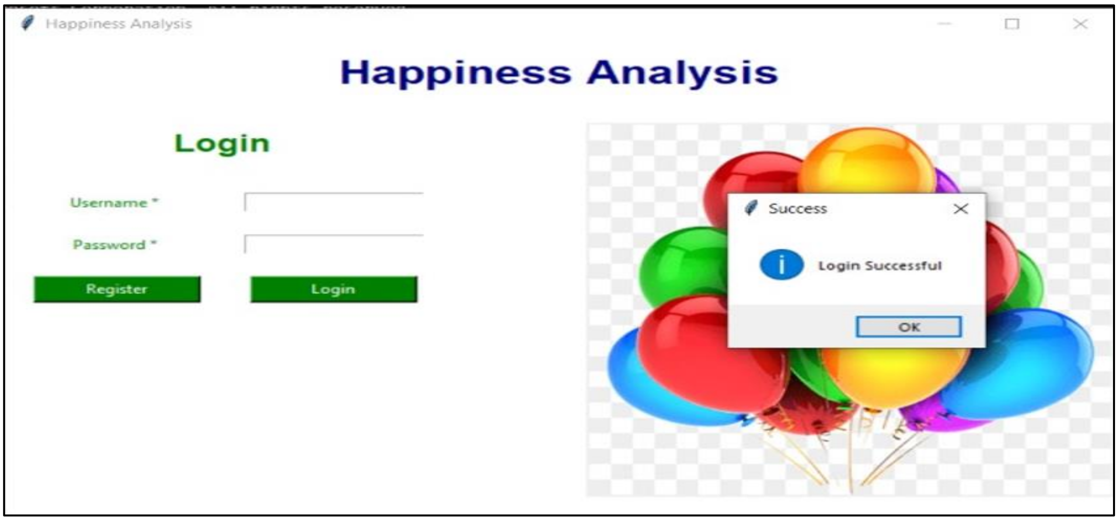
1.This is the registration page where user needs to registered.



Registration Successful
Please Go Back to Login

Figure 9.2.1: Registration Page

2.This is the login screen where user has to enter username and password for login into the system.



Login Successful

Figure 9.2.2: Login Page

3. After Login Successfully Menu page display.

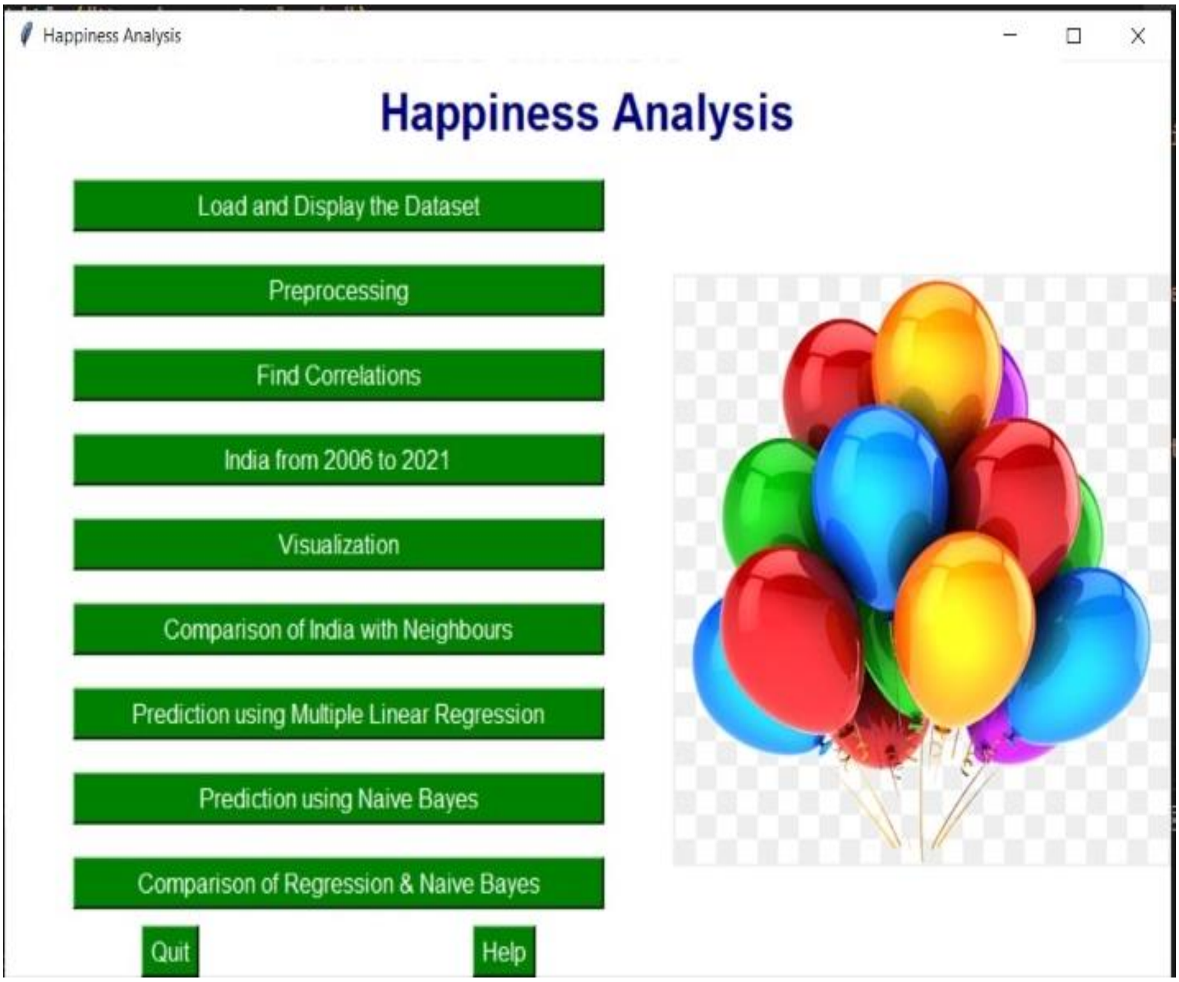
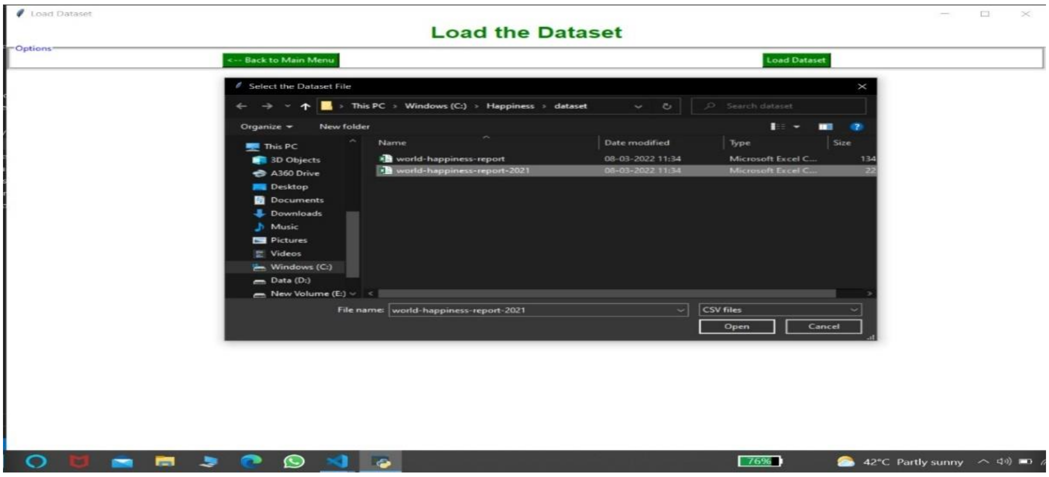

Happiness Analysis
Happiness Analysis
Load and Display the Dataset
Preprocessing
Find Correlations
India from 2006 to 2021
Visualization
Comparison of India with Neighbours
Prediction using Multiple Linear Regression
Prediction using Naive Bayes
Comparison of Regression & Naive Bayes
Quit
Help

Figure 9.2.3: Menu Page

4.Here We need to Select the Dataset.



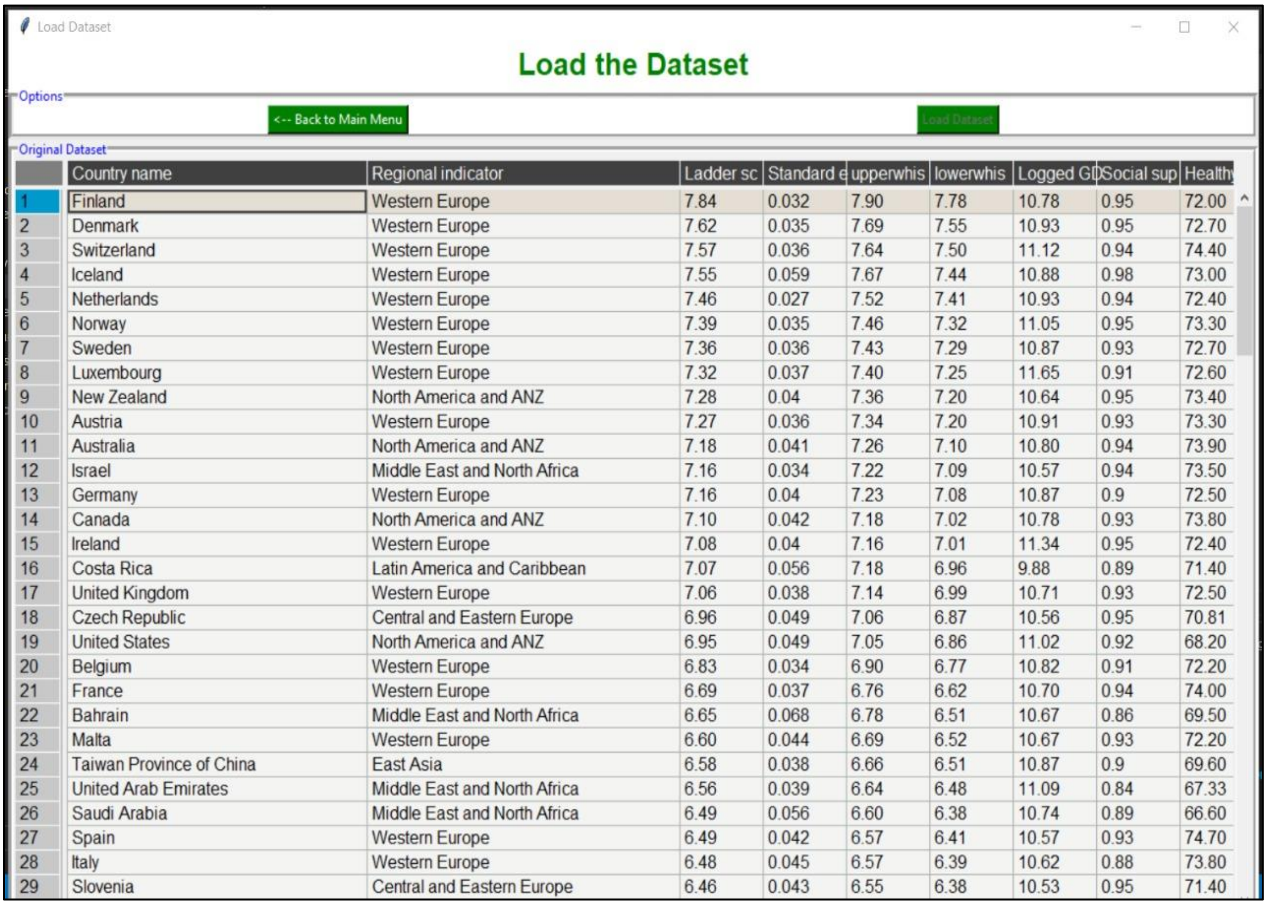
Name	Date modified	Type	Size
world-happiness-report	08-03-2022 11:34	Microsoft Excel C...	134
world-happiness-report-2021	08-03-2022 11:34	Microsoft Excel C...	22

File name: world-happiness-report-2021

CSV files

Figure 9.2.4: Select Dataset

5.This is Loaded Dataset.



	Country name	Regional indicator	Ladder sc	Standard e	upperwhis	lowerwhis	Logged GD	Social sup	Healthy
1	Finland	Western Europe	7.84	0.032	7.90	7.78	10.78	0.95	72.00
2	Denmark	Western Europe	7.62	0.035	7.69	7.55	10.93	0.95	72.70
3	Switzerland	Western Europe	7.57	0.036	7.64	7.50	11.12	0.94	74.40
4	Iceland	Western Europe	7.55	0.059	7.67	7.44	10.88	0.98	73.00
5	Netherlands	Western Europe	7.46	0.027	7.52	7.41	10.93	0.94	72.40
6	Norway	Western Europe	7.39	0.035	7.46	7.32	11.05	0.95	73.30
7	Sweden	Western Europe	7.36	0.036	7.43	7.29	10.87	0.93	72.70
8	Luxembourg	Western Europe	7.32	0.037	7.40	7.25	11.65	0.91	72.60
9	New Zealand	North America and ANZ	7.28	0.04	7.36	7.20	10.64	0.95	73.40
10	Austria	Western Europe	7.27	0.036	7.34	7.20	10.91	0.93	73.30
11	Australia	North America and ANZ	7.18	0.041	7.26	7.10	10.80	0.94	73.90
12	Israel	Middle East and North Africa	7.16	0.034	7.22	7.09	10.57	0.94	73.50
13	Germany	Western Europe	7.16	0.04	7.23	7.08	10.87	0.9	72.50
14	Canada	North America and ANZ	7.10	0.042	7.18	7.02	10.78	0.93	73.80
15	Ireland	Western Europe	7.08	0.04	7.16	7.01	11.34	0.95	72.40
16	Costa Rica	Latin America and Caribbean	7.07	0.056	7.18	6.96	9.88	0.89	71.40
17	United Kingdom	Western Europe	7.06	0.038	7.14	6.99	10.71	0.93	72.50
18	Czech Republic	Central and Eastern Europe	6.96	0.049	7.06	6.87	10.56	0.95	70.81
19	United States	North America and ANZ	6.95	0.049	7.05	6.86	11.02	0.92	68.20
20	Belgium	Western Europe	6.83	0.034	6.90	6.77	10.82	0.91	72.20
21	France	Western Europe	6.69	0.037	6.76	6.62	10.70	0.94	74.00
22	Bahrain	Middle East and North Africa	6.65	0.068	6.78	6.51	10.67	0.86	69.50
23	Malta	Western Europe	6.60	0.044	6.69	6.52	10.67	0.93	72.20
24	Taiwan Province of China	East Asia	6.58	0.038	6.66	6.51	10.87	0.9	69.60
25	United Arab Emirates	Middle East and North Africa	6.56	0.039	6.64	6.48	11.09	0.84	67.33
26	Saudi Arabia	Middle East and North Africa	6.49	0.056	6.60	6.38	10.74	0.89	66.60
27	Spain	Western Europe	6.49	0.042	6.57	6.41	10.57	0.93	74.70
28	Italy	Western Europe	6.48	0.045	6.57	6.39	10.62	0.88	73.80
29	Slovenia	Central and Eastern Europe	6.46	0.043	6.55	6.38	10.53	0.95	71.40

Figure 9.2.5: Display Dataset

6.Drop Columns from Dataset

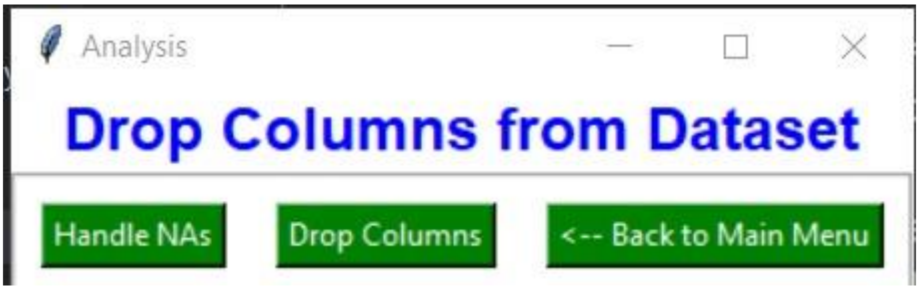


Figure 9.2.6: Drop column

7.This is Correlations between factors in form of Heatmap.

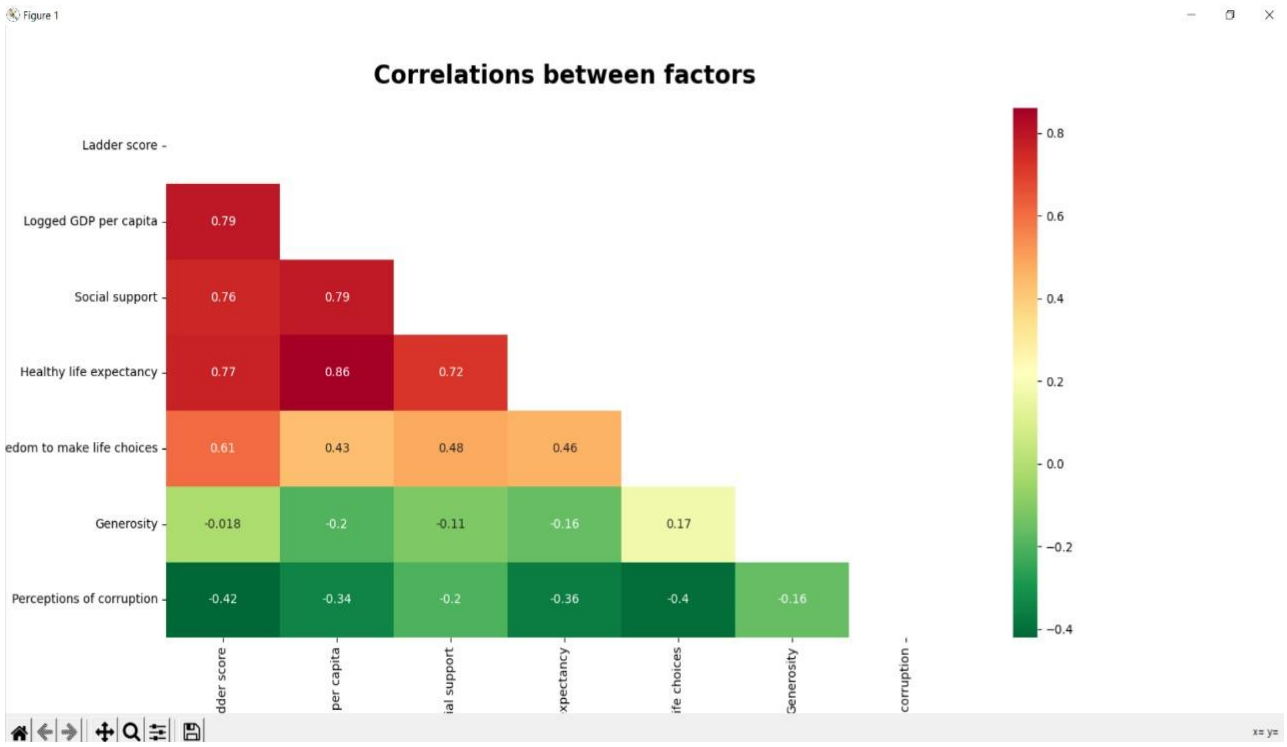
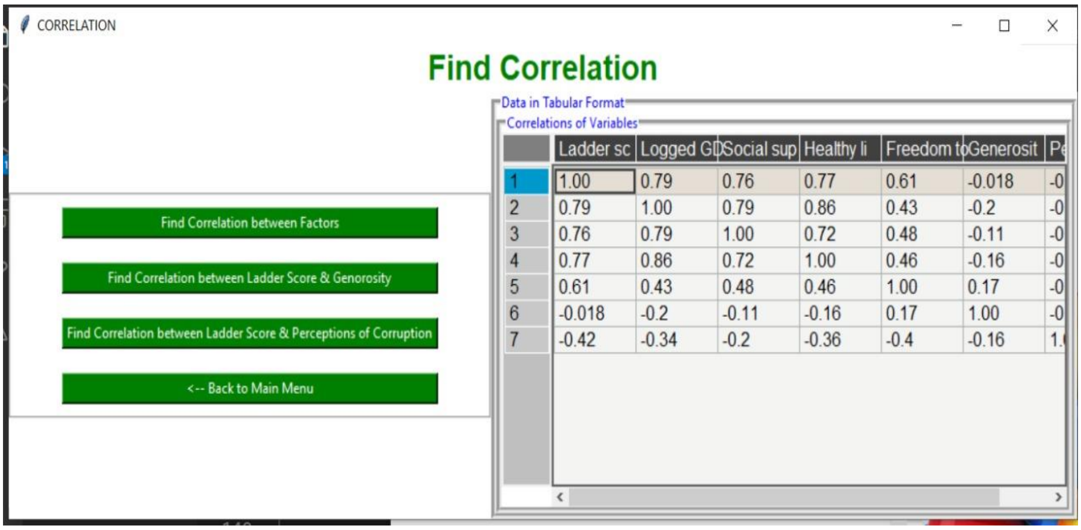


Figure 9.2.7: Correlation between factors using Heatmap

8. This is correlation between factors in form of table


CORRELATION
Find Correlation between Factors
Find Correlation between Ladder Score & Genorosity
Find Correlation between Ladder Score & Perceptions of Corruption
<-- Back to Main Menu

Find Correlation

Data in Tabular Format

Correlations of Variables

	Ladder sc	Logged GD	Social sup	Healthy li	Freedom to	Generosit	Pe
1	1.00	0.79	0.76	0.77	0.61	-0.018	-0
2	0.79	1.00	0.79	0.86	0.43	-0.2	-0
3	0.76	0.79	1.00	0.72	0.48	-0.11	-0
4	0.77	0.86	0.72	1.00	0.46	-0.16	-0
5	0.61	0.43	0.48	0.46	1.00	0.17	-0
6	-0.018	-0.2	-0.11	-0.16	0.17	1.00	-0
7	-0.42	-0.34	-0.2	-0.36	-0.4	-0.16	1.

Figure 9.2.8: correlation between factors in form of table

9.This is Visualization of India

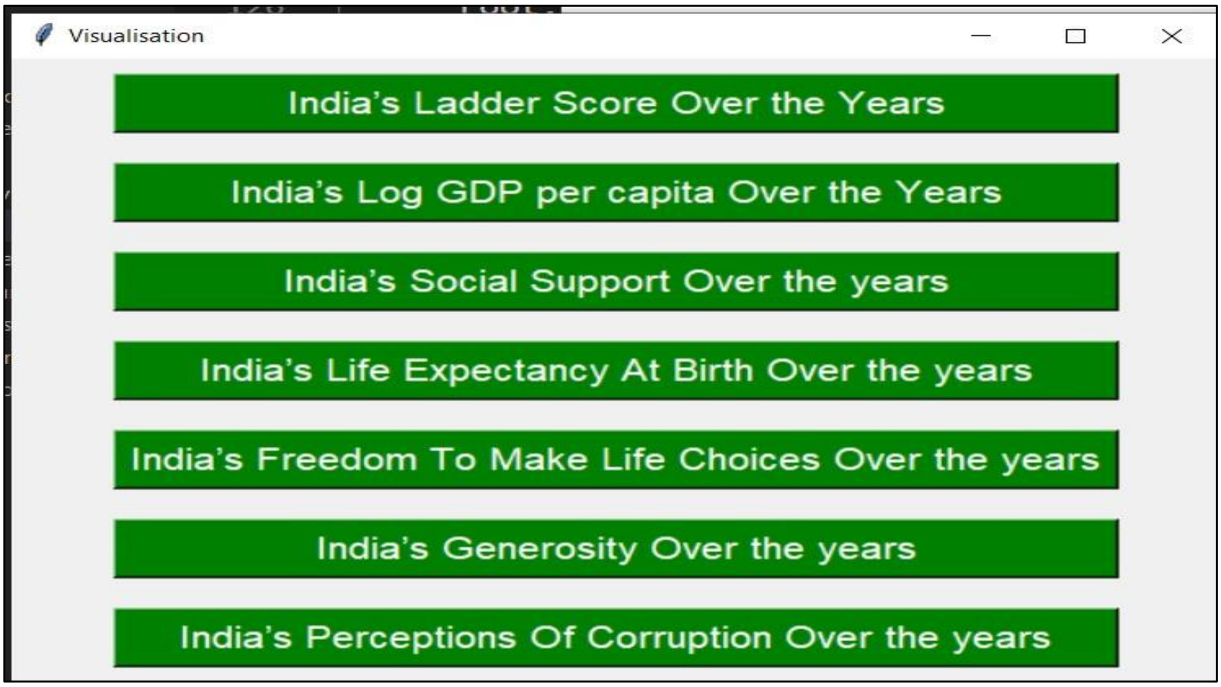
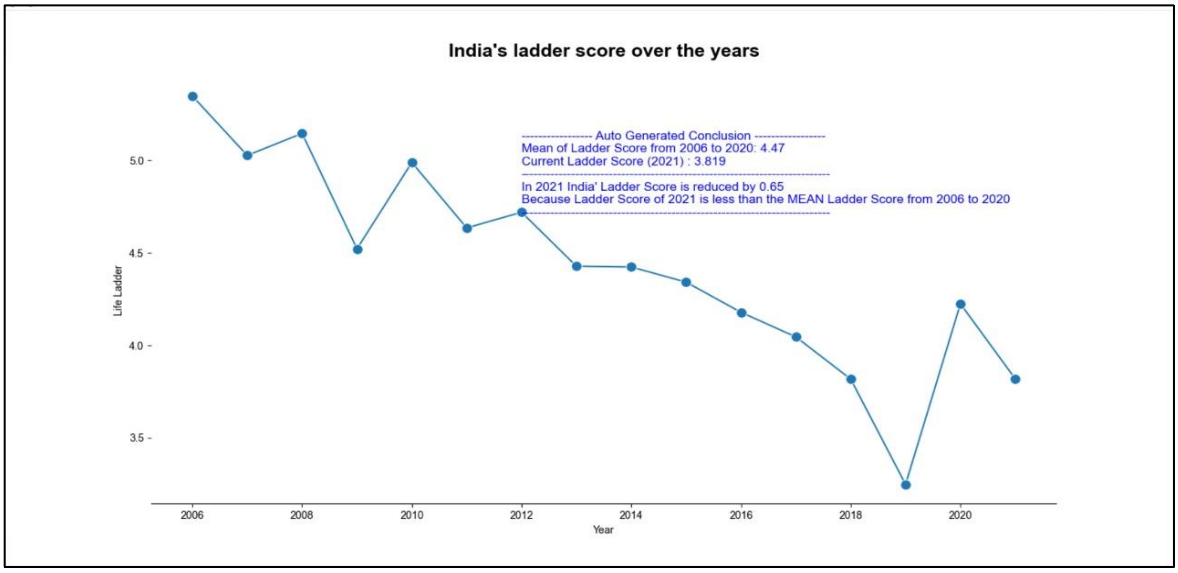

Visualisation
India’s Ladder Score Over the Years
India’s Log GDP per capita Over the Years
India’s Social Support Over the years
India’s Life Expectancy At Birth Over the years
India’s Freedom To Make Life Choices Over the years
India’s Generosity Over the years
India’s Perceptions Of Corruption Over the years

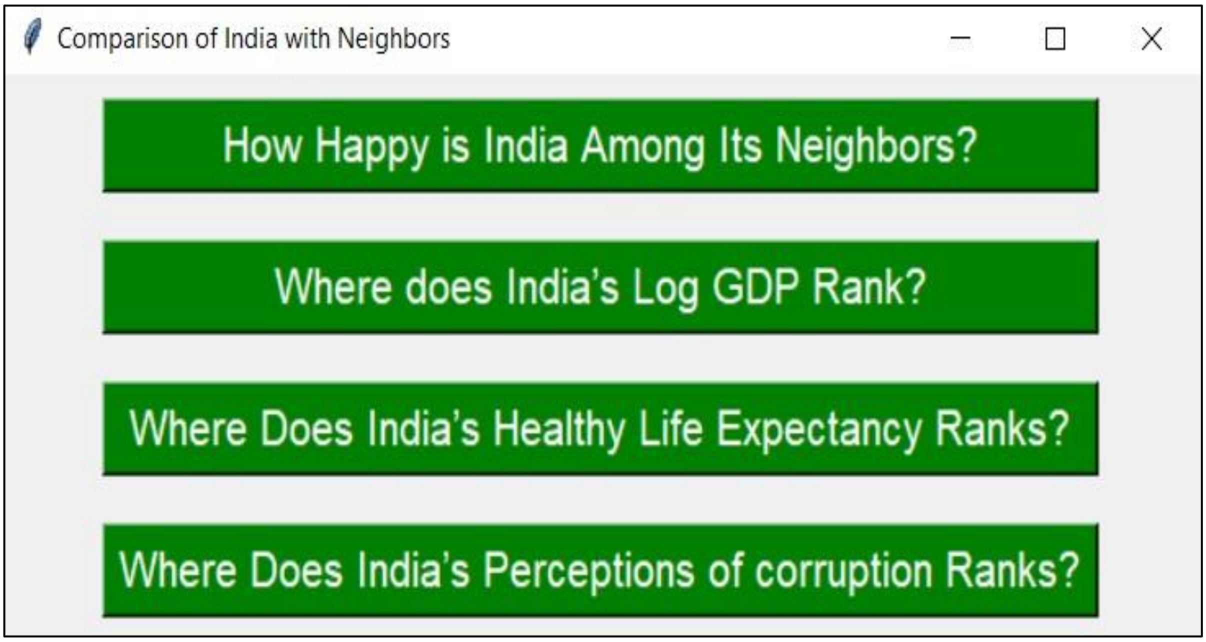
Figure 9.2.9: Menu for Visualization



```
----------------- Auto Generated Conclusion -----------------
Mean of Ladder Score from 2006 to 2020: 4.47
Current Ladder Score (2021) : 3.819
--------------------------------------------------------------
In 2021 India' Ladder Score is reduced by 0.65
Because Ladder Score of 2021 is less than the MEAN Ladder Score from 2006 to 2020
--------------------------------------------------------------
```

Figure 9.2.10: Visualization

10.This is Comparison of India with Neighbors Country.



Comparison of India with Neighbors

How Happy is India Among Its Neighbors?

Where does India's Log GDP Rank?

Where Does India's Healthy Life Expectancy Ranks?

Where Does India's Perceptions of corruption Ranks?

Figure 9.2.11: Menu for comparison

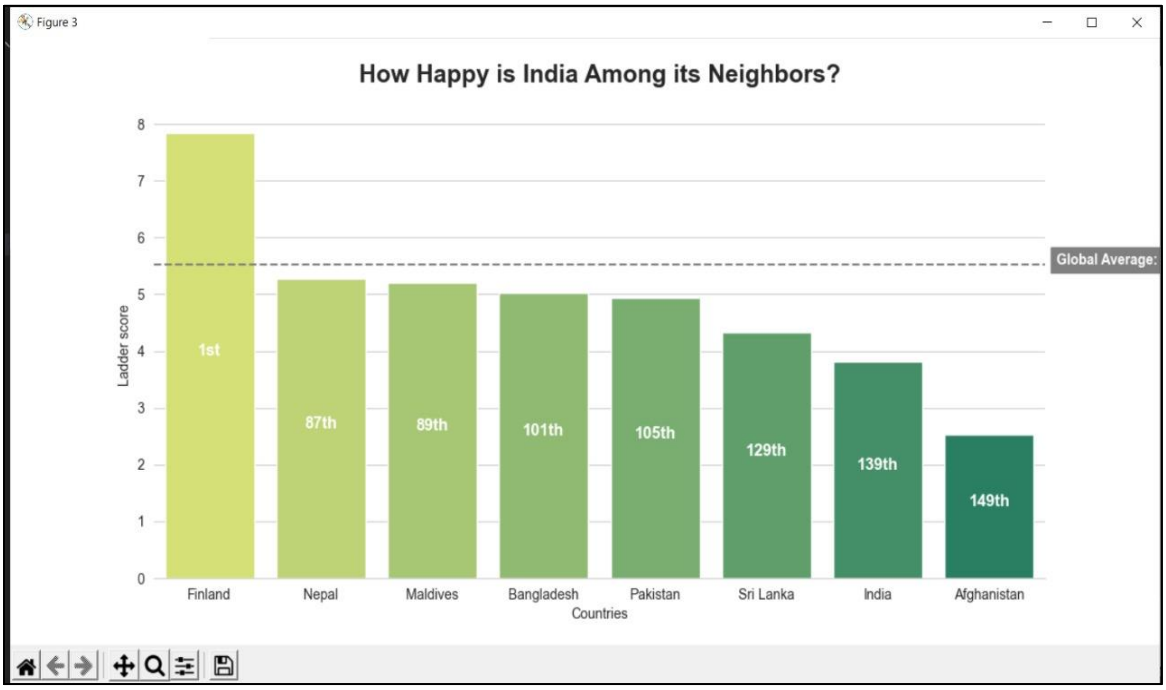


Figure 9.2.12: Comparison of india with neighbors

11.This is menu page of Multiple Linear Regression.

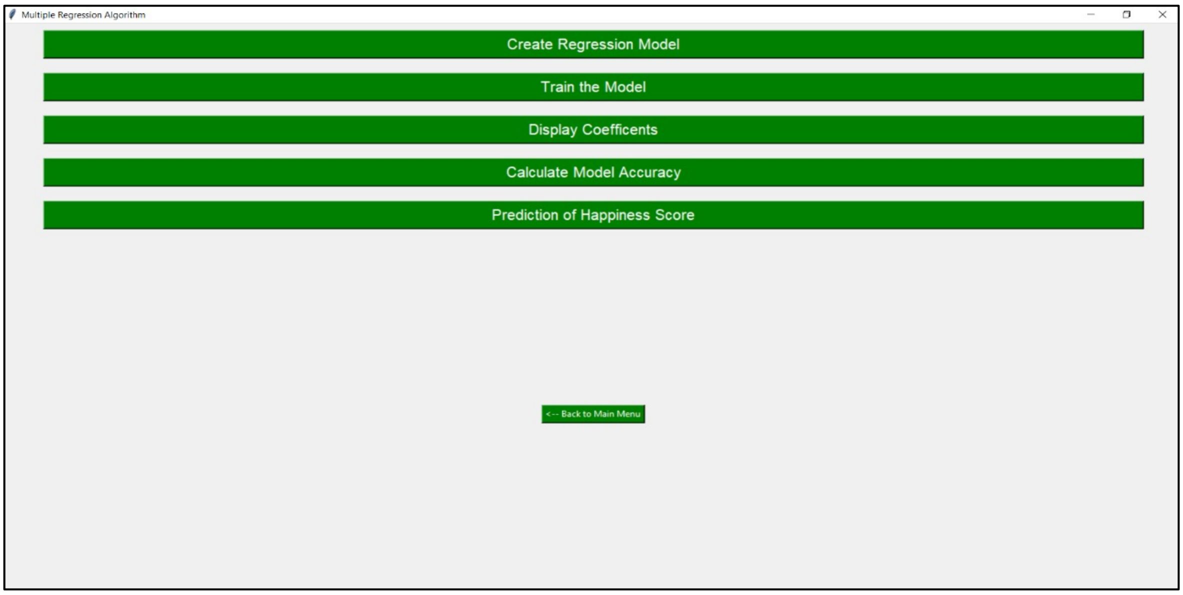
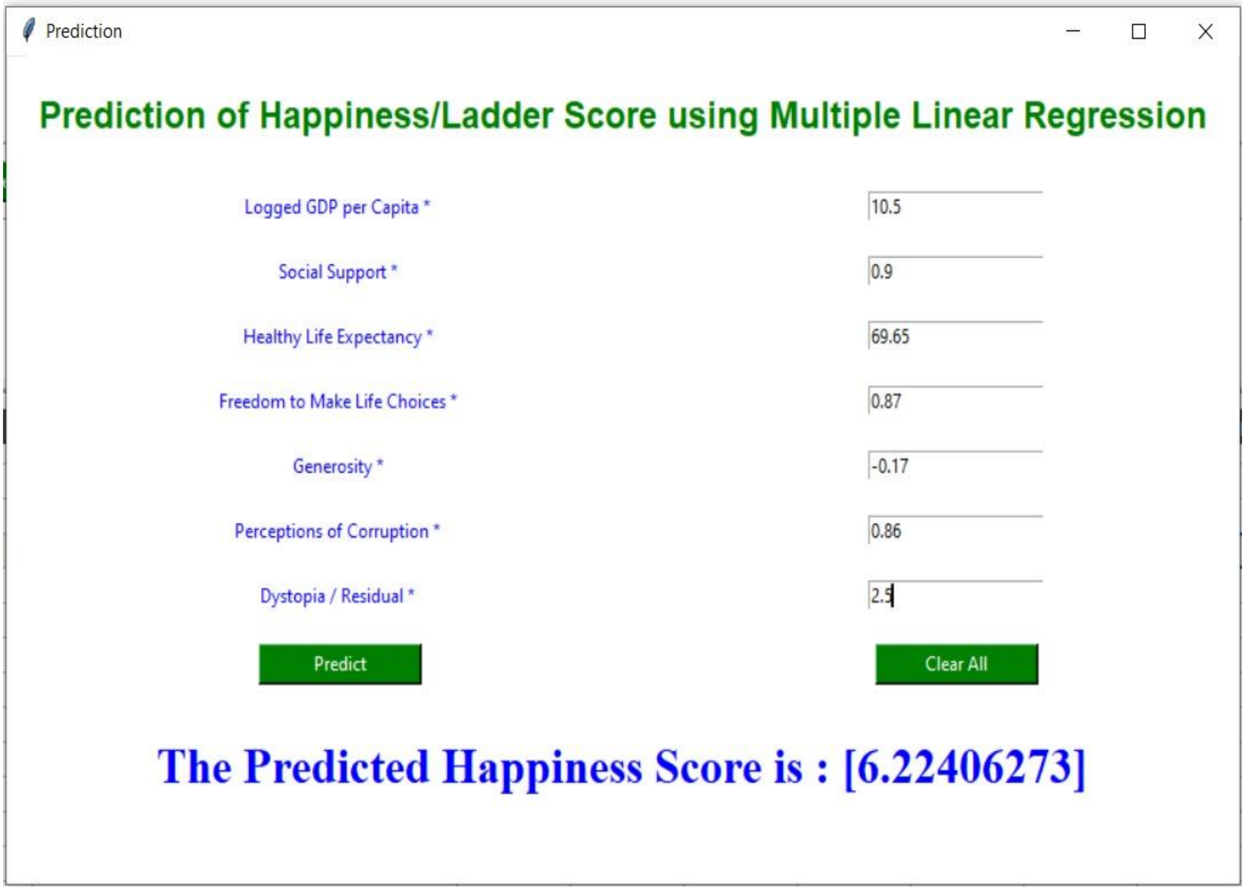


Figure 9.2.13: Menu page of multiple linear Regression

12. This is Prediction of Happiness Score .



Prediction of Happiness/Ladder Score using Multiple Linear Regression

Logged GDP per Capita * 10.5

Social Support * 0.9

Healthy Life Expectancy * 69.65

Freedom to Make Life Choices * 0.87

Generosity * -0.17

Perceptions of Corruption * 0.86

Dystopia / Residual * 2.5

The Predicted Happiness Score is : [6.22406273]

Figure 9.2.14: Prediction Happiness Score using multiple linear Regression

13. This is menu page of Naïve Bayes.

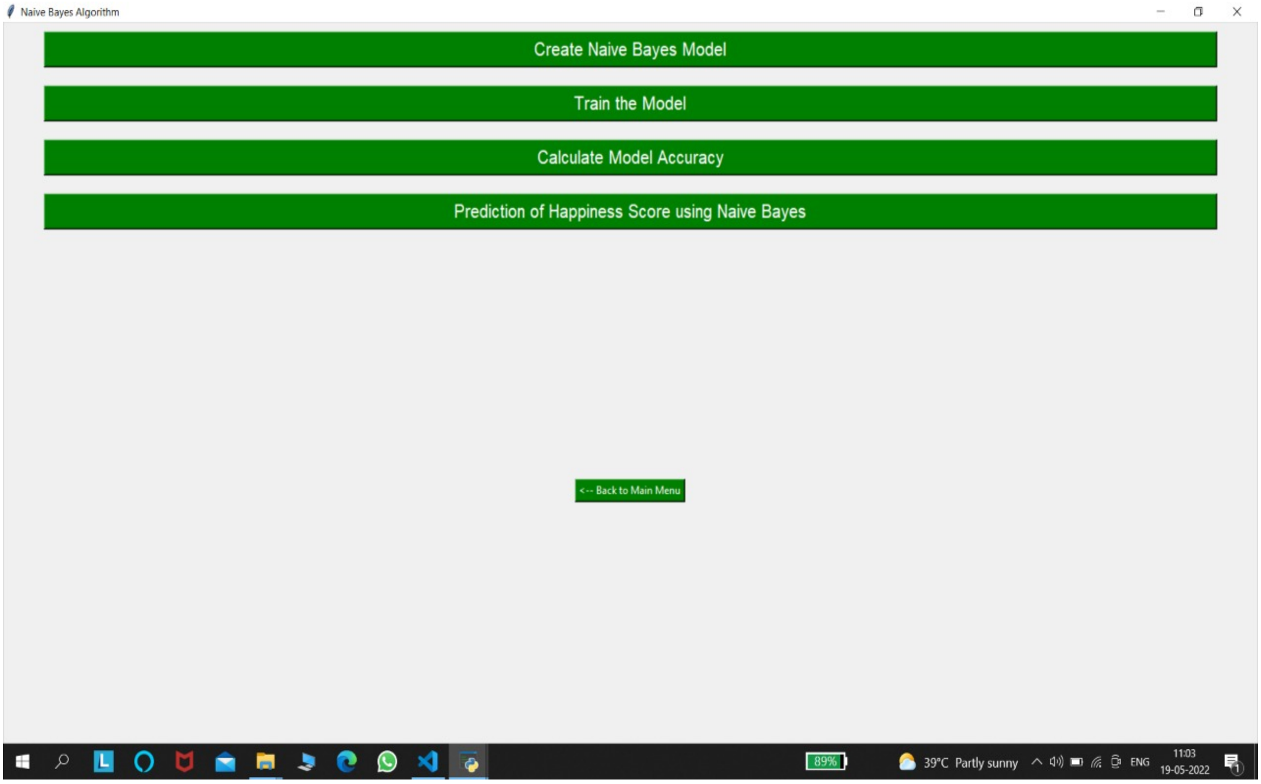
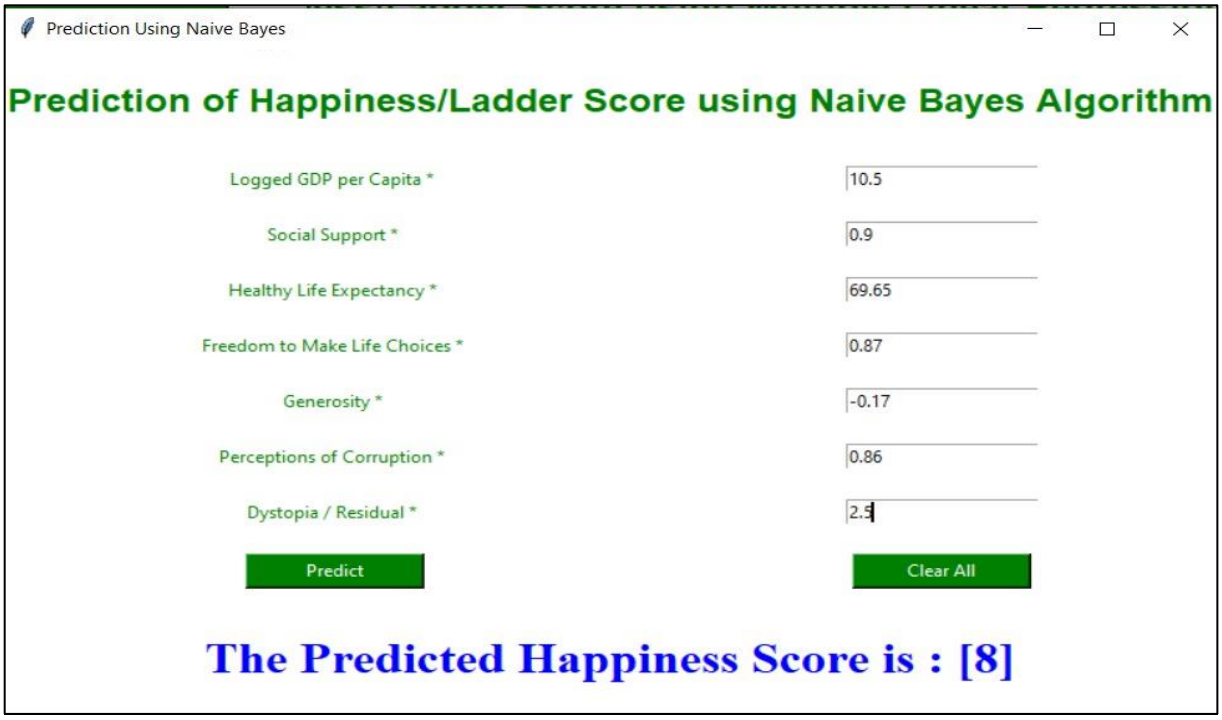


Figure 9.2.15: Menu page of Naïve bayes

14. This is Prediction of Happiness Score using Naïve Bayes.



Prediction of Happiness/Ladder Score using Naive Bayes Algorithm

Logged GDP per Capita * 10.5

Social Support * 0.9

Healthy Life Expectancy * 69.65

Freedom to Make Life Choices * 0.87

Generosity * -0.17

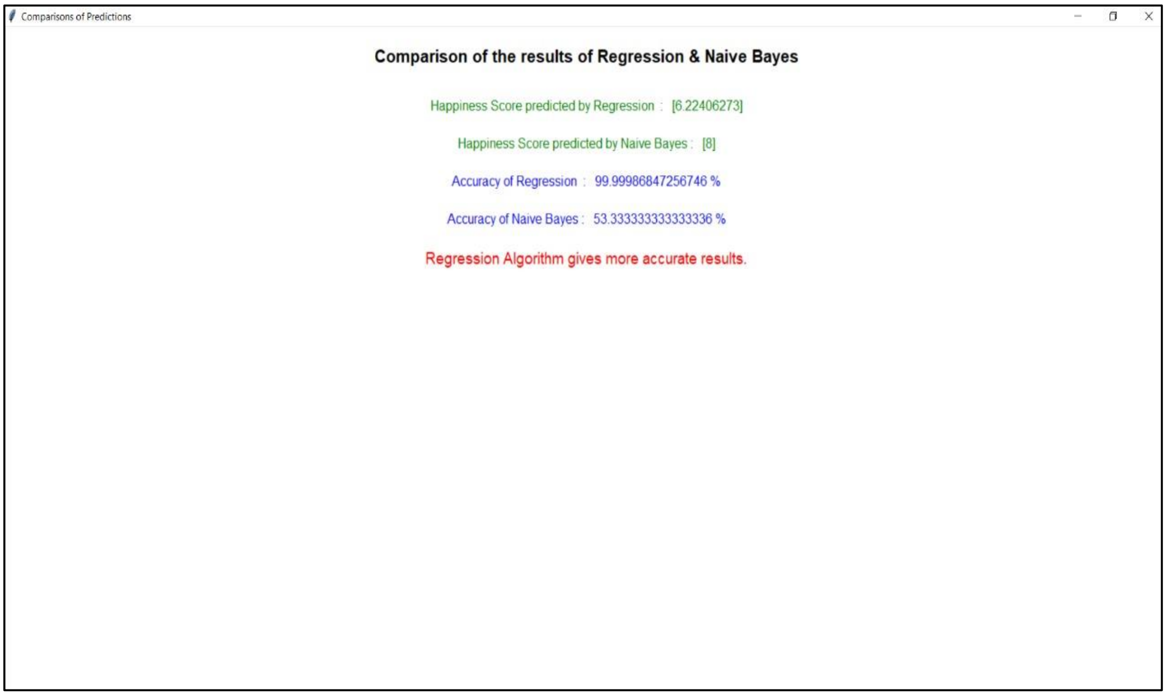
Perceptions of Corruption * 0.86

Dystopia / Residual * 2.5

The Predicted Happiness Score is : [8]

Figure 9.2.16: Prediction Happiness Score using Naïve Bayes

15.Comparison of Multiple Linear Regression & Naïve Bayes



Comparison of the results of Regression & Naive Bayes

```
Happiness Score predicted by Regression :  [6.22406273]
Happiness Score predicted by Naive Bayes :  [8]
Accuracy of Regression :  99.99986847256746 %
Accuracy of Naive Bayes :  53.333333333333336 %
Regression Algorithm gives more accurate results.
```

Figure 9.2.17: Comparison of the result of Regression & Naïve Bayes

CHAPTER 10

Applications of the project

1. This system can be used to predict future values related to Happiness of a country.
2. This system can be used recommendations for India about improving the Happiness with different attributes to make the country more happy in the ranking of Happiness-Index.

CHAPTER 11

Conclusion and Future Scope

11.1 Conclusion

The system is a GUI based desktop applications that is run on desktop systems. The user can be able to upload data and analyze the different aspects of World Happiness. Also there are features to draw the conclusions and thus recommendations from the system, which is very useful to take strategic decisions for happier country. The system can also predict the future values of important parameters related to happiness.

11.2 Future Scope

In our project's future work we will try to design cluster of different country using classification.

References

[1] Parul Oberoi, Shalu Chopra, Yukti Seth, “A Comparative Analysis Of The Factors Affecting Happiness Index”, International Journal of Scientific & Technology Research Volume 9, Issue 03, March 2020.

[2] B. Prashanthi, Dr. R. Ponnusamy, “Future Prediction of World Countries Emotions Status to Understand Economic Status using Happiness Index and SVM Kernel”, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019.

[3] Sarah Ahtesham, “Analyzing Happiness Index as a Measure Along With its Parameters and Strategies for Improving India’s Rank In World Happiness”, 10.21917/ijms.2020.0161.

[4] Charles Alba, “A Data Analysis of the World Happiness Index and Its Relation to the North-South Divide”, Undergraduate Economic Review, Volume 16, Issue 1, Article 6, 2019.

[5] Rajesh K Pillania, India Happiness Report 2020, Retrieved from http://pillania.org › wp-content › uploads › 2020/09.

[6] World Happiness Report 2021 -Amazon AWS, https://happiness-report.s3.amazonaws.com

[7] World Happiness Report – ResearchGate, https://www.researchgate.net › publication.

Annexure A

Weekly Assignment Report

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-I Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	13 Aug 2022
ACADEMIC YEAR	FORMATNO.	REVISIONNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**I**		W.E.F.	02/08/2021	

Week No. - 1 **Dates : 02/08/2021-13/08/2021**

Work done :

The first week dated from 02/08/2021 to 13/08/2021 is the starting of our project where we discussed about the problem definition about the project and the domain on which we will be working. We had couple of meetings with our project guide Prof. C. V. Longani whose outcome was the finalization of the topic and submitting the project title and problem definition along with the domain we will be working in it to project co-ordinator. After the finalization of project topic on 07/08/2021 we extended our work for the completion of synopsis. For the accomplishments of synopsis first we studied the literature review of the topic according to that we define the scope, objective and application of the project. We also designed the tentative model or system architecture of our project and we decided on what basis we meet to proceed for the same. The task of synopsis was divided among the 4 members equally led to completion of synopsis in an efficient manner. After which we submitted it to project guide which he reviewed it and asked us for some changes which we did it and submitted back to guide. After reviewing again, he finally accepted the synopsis which was completed on 13/08/2021.

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani
[Internal Guide]

Dr. A. B. Pawar
[Project Coordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-I Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	23 Aug 2022
ACADEMIC YEAR	FORMA TNO.	REVISIO NNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**I**		W.E.F.	14/08/2021	

Week No. – 2 **Date:-14/08/2021-23/08/2021**

Work done :

After the successful completion of title finalization and synopsis we extended our work to complete the software requirement specification (SRS) of the project. We started our work on 15/08/2021 where we started stating the requirements about our project. According to the tentative project model we thought about various software and hardware requirements which are tentative and predictive and developed our SRS. We also prepared the modular design of our project that is the increment model according to the software specification. And finally, on 21/08/2021 we completed our SRS.

Group ID	**Roll No.**	**Name of Present Student**	**Signature**
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani [Internal Guide]

Dr. A. B. Pawar [Project Coordinator]

Dr. D. B. Kshirsagar [H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-I Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	4 Sep 2022
ACADEMIC YEAR	FORMATNO.	REVISIONNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**I**		W.E.F.	24/08/2021	

Week NO. -3 **Date :-24/08/2021-04/09/2021**

Work done :

After the successful completion of software requirement specification (SRS) we extended our work to next step know as Project planning on 24/08/2021. In this step pf project planning we divided the work of drawing UML diagrams equally among the developer so as to understand the project planning in detail. We also studied the breakdown structure of project model wise which helped us to the better understanding of the project. We draw all the UML diagrams on draw. io and finally completed the assignment on 04/09/2021 and we forwarded to our project guide for verification purpose. After verified from our project guide we finally complete the project planning assignment.

Group ID	**Roll No.**	**Name of Present Student**	**Signature**
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani
[Internal Guide]

Dr. A. B. Pawar
[Project Coordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-I Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	25 Sep 2022
ACADEMIC YEAR	FORMATNO.	REVISIONNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**I**		W.E.F.	06/09/2021	

Week No. -4 **Date:-06/09/2021-25/09/2021**

Work done :

After the successful completion of project plan on 04/09/2021 we extended our work for the completion of the next assignment which was risk management. In risk management we discussed about various kinds of risks associated with our project and we completed that assignment on 18/09/2021. In the same week we also parallelly started developing the partial project report so that the work gets completed in given deadline and we completed approximately 80 same week and remaining of the work will be completed till the end of next week. And we finally submitted the assignment of risk management on 25/09/2021 to the project guide.

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani [Internal Guide]

Dr. A. B. Pawar [ProjectCoordinator]

Dr. D. B. Kshirsagar [H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	12 Feb 2022
ACADEMIC YEAR	FORMATNO.	REVISIONNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	07/02/2022	

Week No. - 5 **Dates : 07/02/2022 - 12/02/2022**

Work Completed :

This Week, We started actual implementation of our project. As we're developed the page for users i.e login page and registration page, Means when the user can register themselves and then login into the system using the username and password which should be displayed. Under the guidance of the Prof. C.V. Longani we have done implementation smoothly.

Guide Remark:-

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani
[Internal Guide]

Dr. A. B. Pawar
[ProjectCoordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	21 Feb 2022
ACADEMIC YEAR	FORMA TNO.	REVISIO NNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	14/02/2022	

Week No. - 6 Dates : 14/02/2022- 21/02/2022

Work Completed : Last Week, We implemented Login page and Registration page. In this week we implemented three modules. Which are Load and display the dataset, Preprocessing and find correlation.

This Load and display dataset module will load the data set from local computer to the system. In Preprocessing we drop the unwanted columns and handle the null values. Correlation is used for finding the relationship between different parameters. We faced problems during developing this module. But under the guidance of Prof. C. V. Longani we have done smoothly.

Guide Remark:-

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani	**Dr. A. B. Pawar**	**Dr. D. B. Kshirsagar**
[Internal Guide]	[ProjectCoordinator]	[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	28 Feb 2022
ACADEMIC YEAR	FORMA TNO.	REVISIO NNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	22/02/2022	

Week No. - 7 **Dates : 22/02/2022 - 28/02/2022**

Work Completed :

This Week, We developed two modules i.e History of India and Visualization. In visualization module display the graphical visualization of analysis and recommendation using the matplotlib tkinter library.

Guide Remark:-

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani [Internal Guide]

Dr. A. B. Pawar [ProjectCoordinator]

Dr. D. B. Kshirsagar [H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	9 Mar 2022
ACADEMIC YEAR	FORMA TNO.	REVISIO NNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	01/02/2022	

Week No. - 8 **Dates : 01/03/2022 - 09/03/2022**

Work Completed :

In this week we implementing algorithms. As we're implementing two algorithms, one of them is multiple linear regression algorithm. We are done with the implementing part of this module and coming to its accuracy we got 99% accuracy results from the module.

Guide Remark:-

Group ID	**Roll No.**	**Name of Present Student**	**Signatur e**
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani
[Internal Guide]

Dr. A. B. Pawar
[ProjectCoordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	17 Mar 2022
ACADEMIC YEAR	FORMA TNO.	REVISIO NNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	10/03/2022	

Week No. - 9 **Dates :10/03/2022 - 17/03/2022**

Work Completed :

After completing multiple linear regression. This week we implemented our 2nd algorithm i.e Naïve bayes algorithm. We got 53.97 accuracy result for naïve bayes algorithm. we will try to optimize the Naïve bayes module in the upcoming days for better accuracy results.

Guide Remark:-

Group ID	**Roll No.**	**Name of Present Student**	**Signatur e**
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

Prof. C. V. Longani	**Dr. A. B. Pawar**	**Dr. D. B. Kshirsagar**
[Internal Guide]	[ProjectCoordinator]	[H.O.D. Computer Dept.]

SRES Sanjivani College of Engineering, Kopargaon - 423603	**BE Project Work Sem-II** **Weekly Progress Report**			Page	1 of 1
	Department of Computer Engineering			Prepared on	26 Mar 2022
ACADEMIC YEAR	FORMATNO.	REVISIONNO.	DATE	CLASS	DIV
2021-22	**ACAD-F-**	0		BE COMP	B
Semester	**II**		W.E.F.	18/03/2022	

Week No. - 10 **Dates : 18/03/2022 - 26/03/2022**

Work Completed :

After completion of everything related to the Experimental Analysis of our project. We have Published Paper in OAIJSE publication which stands for "Open Access International Journal Of Science and Engineering" in volume 7 __ Issue 4 __ Year April 2022 URL for Published paper – https://www.oaijse.com We faced lot of problems while publishing the survey paper of our project. But under the guidance of Prof. C. V. Longani mam we have done everything very smoothly.

Group ID	Roll No.	Name of Present Student	Signature
27	87	1) Amrat Vishwanath More	
	95	2) Mayuri Madhukar Muthe	
	105	3) Tabassum Faruk Pathan	
	117	4) Prajakta Ishwar Shelar	

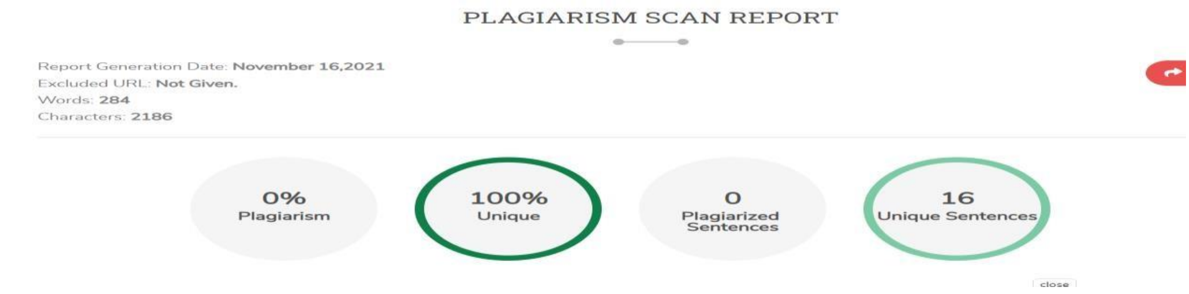
Prof. C. V. Longani
[Internal Guide]

Dr. A. B. Pawar
[ProjectCoordinator]

Dr. D. B. Kshirsagar
[H.O.D. Computer Dept.]

ANNEXURE B

PLAGIARISM REPORT

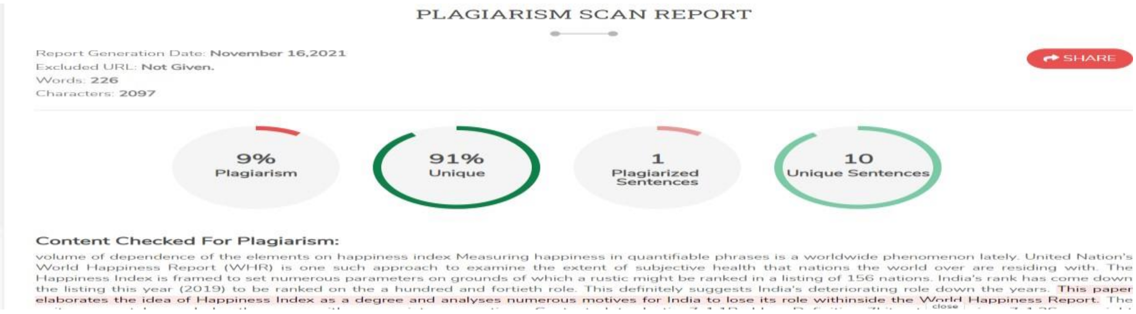


PLAGIARISM SCAN REPORT

Report Generation Date: **November 16,2021**
Excluded URL: **Not Given.**
Words: **284**
Characters: **2186**

0% Plagiarism | 100% Unique | 0 Plagiarized Sentences | 16 Unique Sentences

Figure B.1: Plagiarism Scan Report 1



PLAGIARISM SCAN REPORT

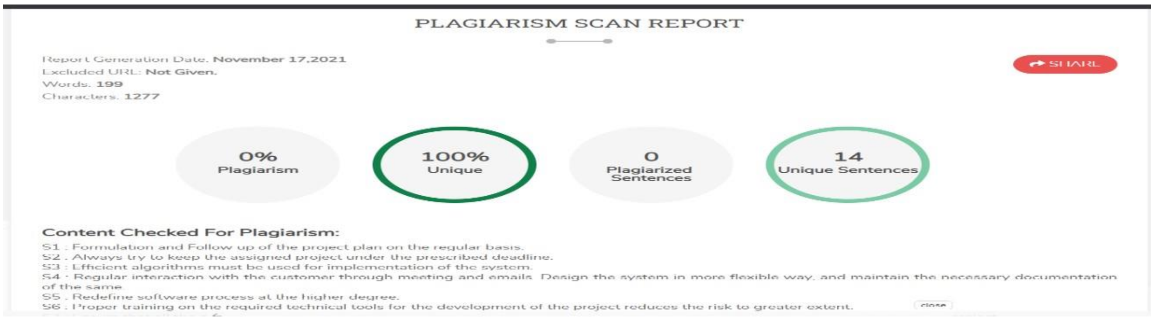
Report Generation Date: **November 16,2021**
Excluded URL: **Not Given.**
Words: **226**
Characters: **2097**

9% Plagiarism | 91% Unique | 1 Plagiarized Sentences | 10 Unique Sentences

Content Checked For Plagiarism:

volume of dependence of the elements on happiness index Measuring happiness in quantifiable phrases is a worldwide phenomenon lately. United Nation's World Happiness Report (WHR) is one such approach to examine the extent of subjective health that nations the world over are residing with. The Happiness Index is framed to set numerous parameters on grounds of which a rustic might be ranked in a listing of 156 nations. India's rank has come down the listing this year (2019) to be ranked on the a hundred and fortieth role. This definitely suggests India's deteriorating role down the years. This paper elaborates the idea of Happiness Index as a degree and analyses numerous motives for India to lose its role withinside the World Happiness Report. The

Figure B.2: Plagiarism Scan Report 2



PLAGIARISM SCAN REPORT

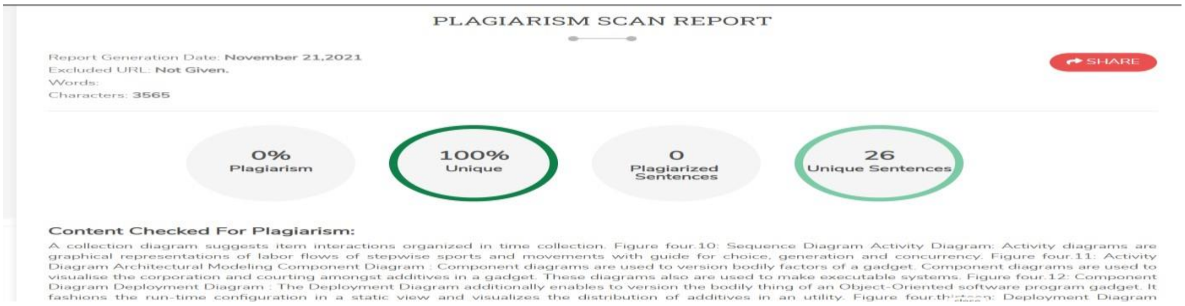
Report Generation Date: **November 17,2021**
Excluded URL: **Not Given.**
Words: **199**
Characters: **1277**

0% Plagiarism | 100% Unique | 0 Plagiarized Sentences | 14 Unique Sentences

Content Checked For Plagiarism:

S1 : Formulation and Follow up of the project plan on the regular basis.
S2 : Always try to keep the assigned project under the prescribed deadline.
S3 : Efficient algorithms must be used for implementation of the system.
S4 : Regular interaction with the customer through meeting and emails. Design the system in more flexible way, and maintain the necessary documentation of the same.
S5 : Redefine software process at the higher degree.
S6 : Proper training on the required technical tools for the development of the project reduces the risk to greater extent.

Figure B.3: Plagiarism Scan Report 3



PLAGIARISM SCAN REPORT

Report Generation Date: **November 21,2021**
Excluded URL: **Not Given.**
Words:
Characters: **3565**

0% Plagiarism

100% Unique

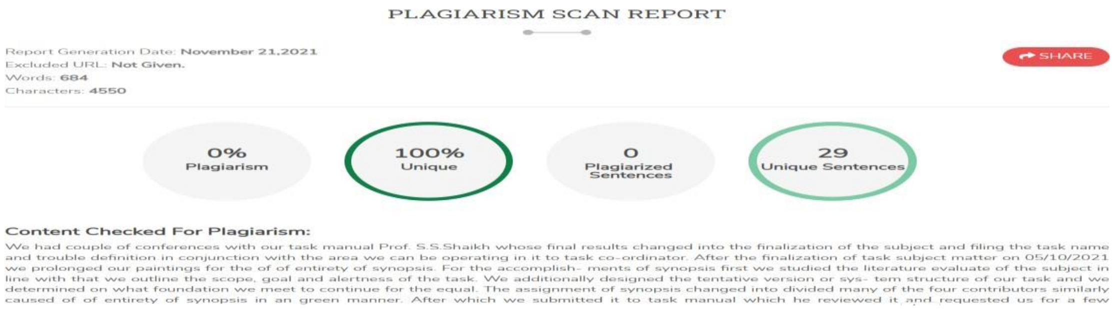
0 Plagiarized Sentences

26 Unique Sentences

Content Checked For Plagiarism:

A collection diagram suggests item interactions organized in time collection. Figure four.10: Sequence Diagram Activity Diagram: Activity diagrams are graphical representations of labor flows of stepwise sports and movements with guide for choice, generation and concurrency. Figure four.11: Activity Diagram Architectural Modeling Component Diagram : Component diagrams are used to version bodily factors of a gadget. Component diagrams are used to visualise the corporation and courting amongst additives in a gadget. These diagrams also are used to make executable systems. Figure four.12: Component Diagram Deployment Diagram : The Deployment Diagram additionally enables to version the bodily thing of an Object-Oriented software program gadget. It fashions the run-time configuration in a static view and visualizes the distribution of additives in an utility. Figure four.thirteen: Deployment Diagram

Figure B.4: Plagiarism Scan Report 4



PLAGIARISM SCAN REPORT

Report Generation Date: **November 21,2021**
Excluded URL: **Not Given.**
Words: **684**
Characters: **4550**

0% Plagiarism

100% Unique

0 Plagiarized Sentences

29 Unique Sentences

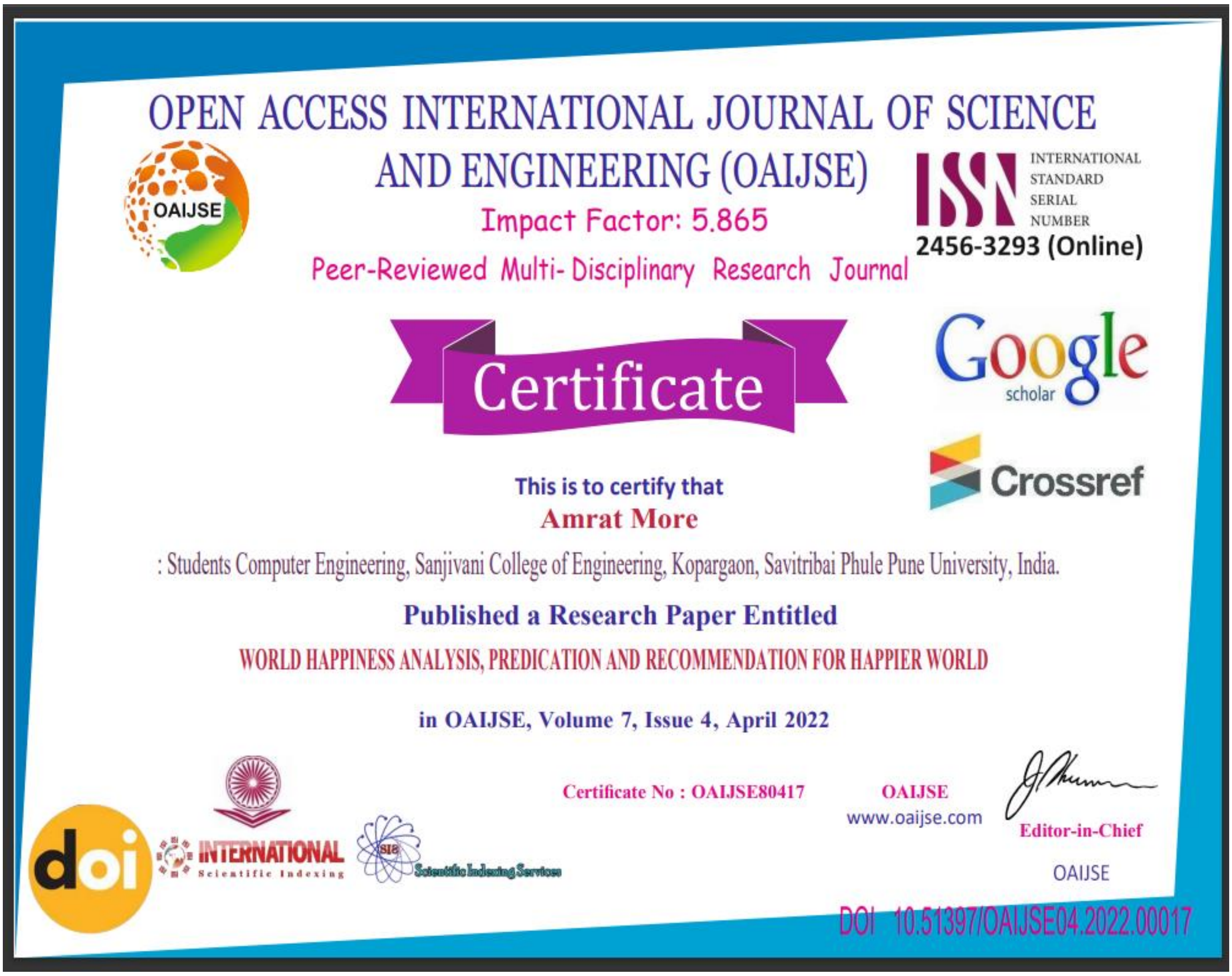
Content Checked For Plagiarism:

We had couple of conferences with our task manual Prof. S.S.Shaikh whose final results changed into the finalization of the subject and filing the task name and trouble definition in conjunction with the area we can be operating in it to task co-ordinator. After the finalization of task subject matter on 05/10/2021 we prolonged our paintings for the of of entirety of synopsis. For the accomplish- ments of synopsis first we studied the literature evaluate of the subject in line with that we outline the scope, goal and alertness of the task. We additionally designed the tentative version or sys- tem structure of our task and we determined on what foundation we meet to continue for the equal. The assignment of synopsis changed into divided many of the four contributors similarly caused of of entirety of synopsis in an green manner. After which we submitted it to task manual which he reviewed it and requested us for a few

Figure B.5: Plagiarism Scan Report 5

Annexure C
Certification

C.1 OAIJSE Certificate of Publication, Amrat More



OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE AND ENGINEERING (OAIJSE)

Impact Factor: 5.865

Peer-Reviewed Multi- Disciplinary Research Journal

ISSN INTERNATIONAL STANDARD SERIAL NUMBER 2456-3293 (Online)

Certificate

This is to certify that

Amrat More

: Students Computer Engineering, Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, India.

Published a Research Paper Entitled

WORLD HAPPINESS ANALYSIS, PREDICATION AND RECOMMENDATION FOR HAPPIER WORLD

in OAIJSE, Volume 7, Issue 4, April 2022

Certificate No : OAIJSE80417

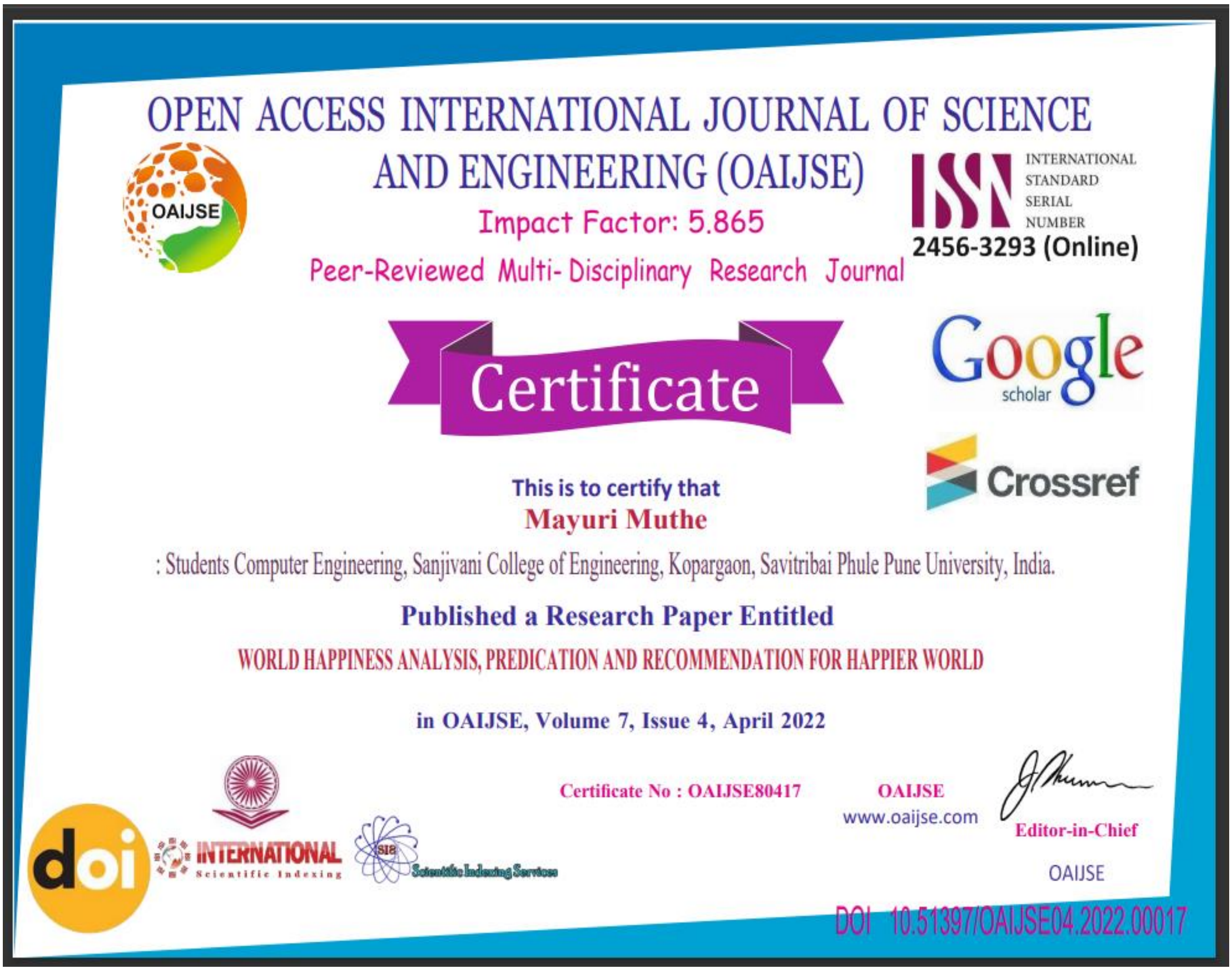
OAIJSE
www.oaijse.com

Editor-in-Chief
OAIJSE

DOI 10.51397/OAIJSE04.2022.00017

Figure C.1: OAIJSE Certificate of Publication

C.2 OAIJSE Certificate of Publication, Mayuri Muthe



OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE AND ENGINEERING (OAIJSE)

Impact Factor: 5.865

Peer-Reviewed Multi- Disciplinary Research Journal

INTERNATIONAL STANDARD SERIAL NUMBER
2456-3293 (Online)

Certificate

This is to certify that
Mayuri Muthe
: Students Computer Engineering, Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, India.

Published a Research Paper Entitled

WORLD HAPPINESS ANALYSIS, PREDICATION AND RECOMMENDATION FOR HAPPIER WORLD

in OAIJSE, Volume 7, Issue 4, April 2022

Certificate No : OAIJSE80417

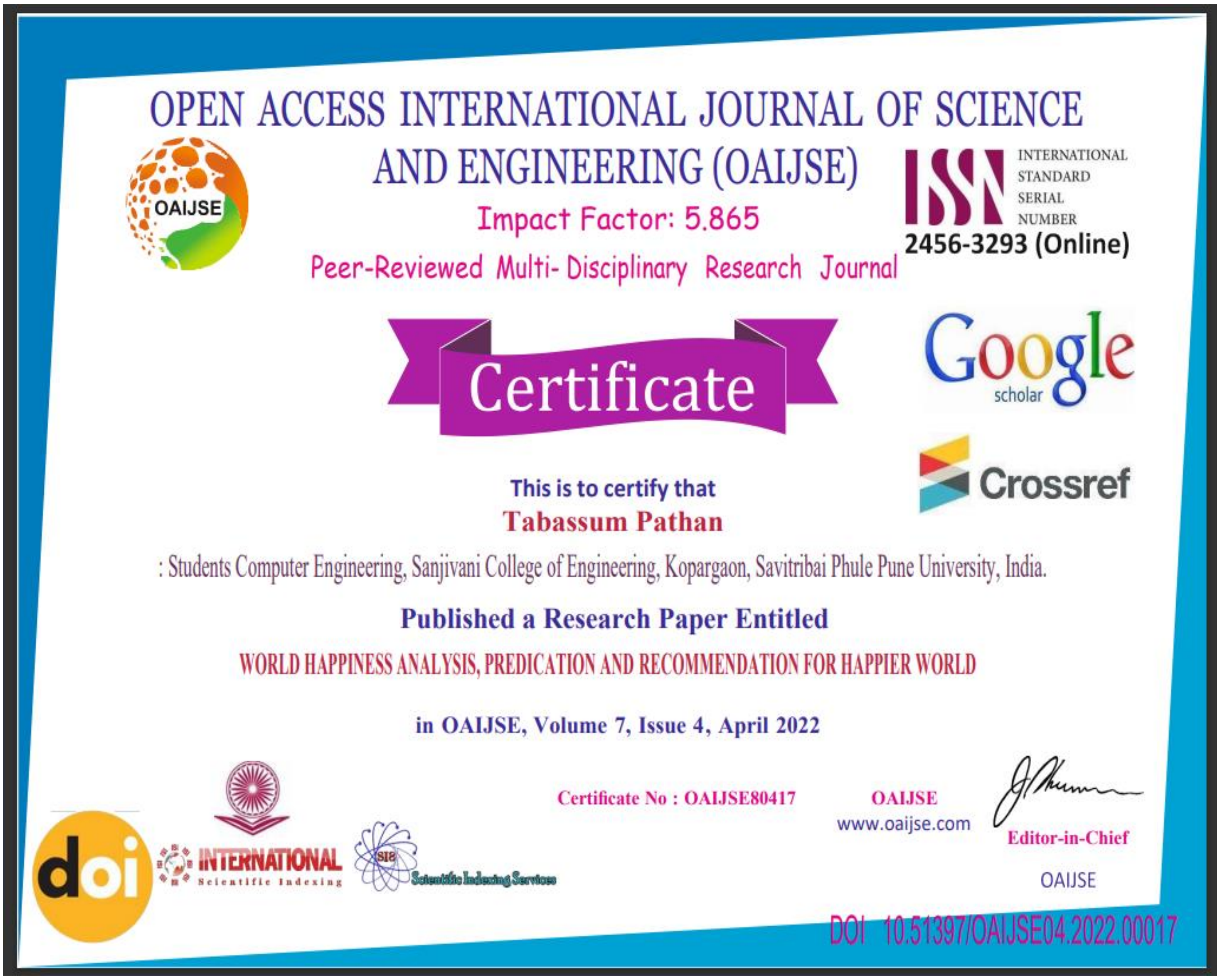
OAIJSE
www.oaijse.com

Editor-in-Chief
OAIJSE

DOI 10.51397/OAIJSE04.2022.00017

Figure C.2: OAIJSE Certificate of Publication

C.3 OAIJSE Certificate of Publication, Tabassum Pthan



OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE AND ENGINEERING (OAIJSE)

Impact Factor: 5.865

Peer-Reviewed Multi- Disciplinary Research Journal

INTERNATIONAL STANDARD SERIAL NUMBER
2456-3293 (Online)

Certificate

This is to certify that

Tabassum Pathan

: Students Computer Engineering, Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, India.

Published a Research Paper Entitled

WORLD HAPPINESS ANALYSIS, PREDICATION AND RECOMMENDATION FOR HAPPIER WORLD

in OAIJSE, Volume 7, Issue 4, April 2022

Certificate No : OAIJSE80417

OAIJSE
www.oaijse.com

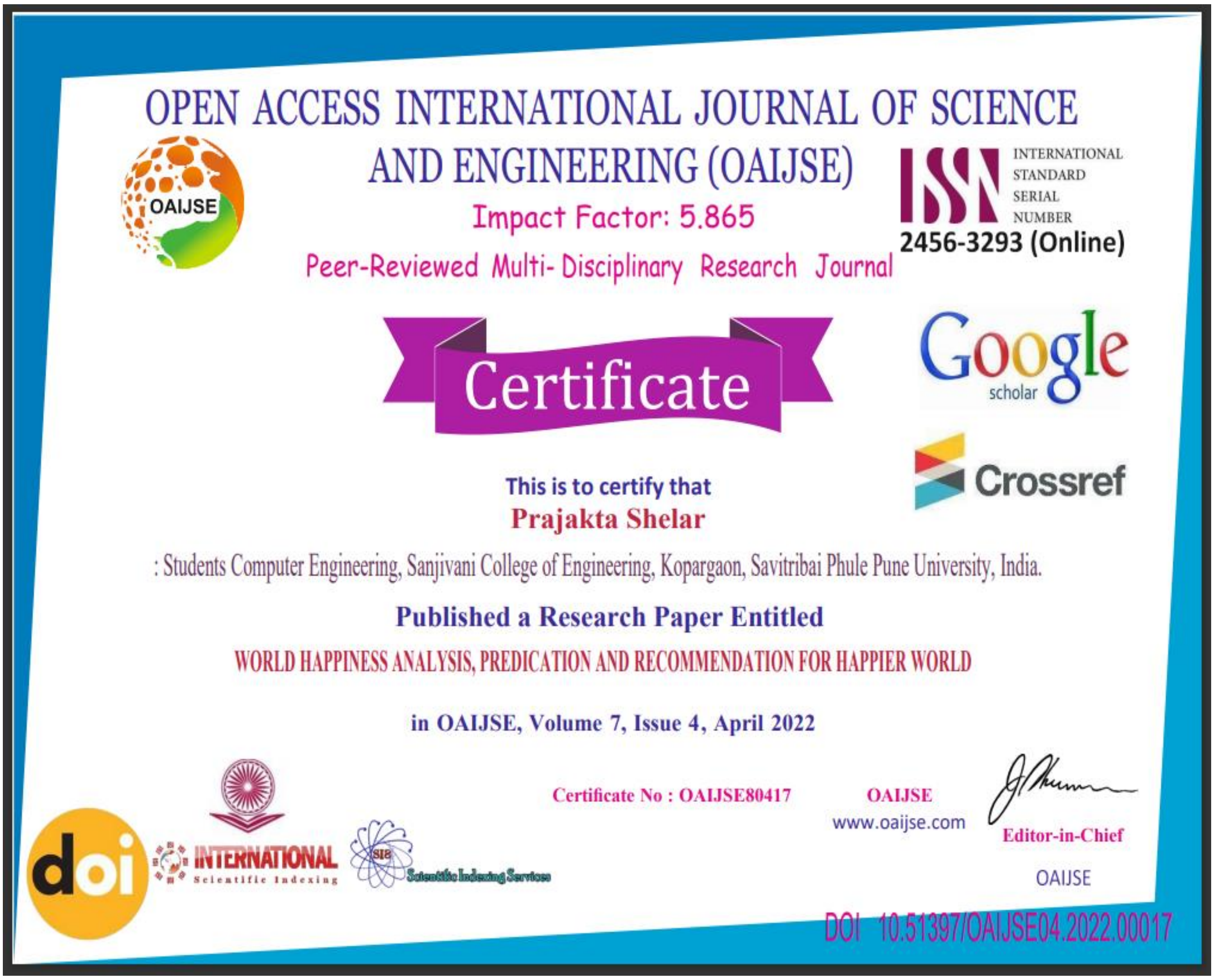
Editor-in-Chief
OAIJSE

DOI 10.51397/OAIJSE04.2022.00017

OAIJSE
ISSN
Google scholar
Crossref
doi
INTERNATIONAL Scientific Indexing
SIS Scientific Indexing Services

Figure C.3: OAIJSE Certificate of Publication

C.4 OAIJSE Certificate of Publication, Prajakta Shelar



OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE AND ENGINEERING (OAIJSE)

Impact Factor: 5.865

Peer-Reviewed Multi-Disciplinary Research Journal

INTERNATIONAL STANDARD SERIAL NUMBER

2456-3293 (Online)

Certificate

This is to certify that

Prajakta Shelar

: Students Computer Engineering, Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, India.

Published a Research Paper Entitled

WORLD HAPPINESS ANALYSIS, PREDICATION AND RECOMMENDATION FOR HAPPIER WORLD

in OAIJSE, Volume 7, Issue 4, April 2022

Certificate No : OAIJSE80417

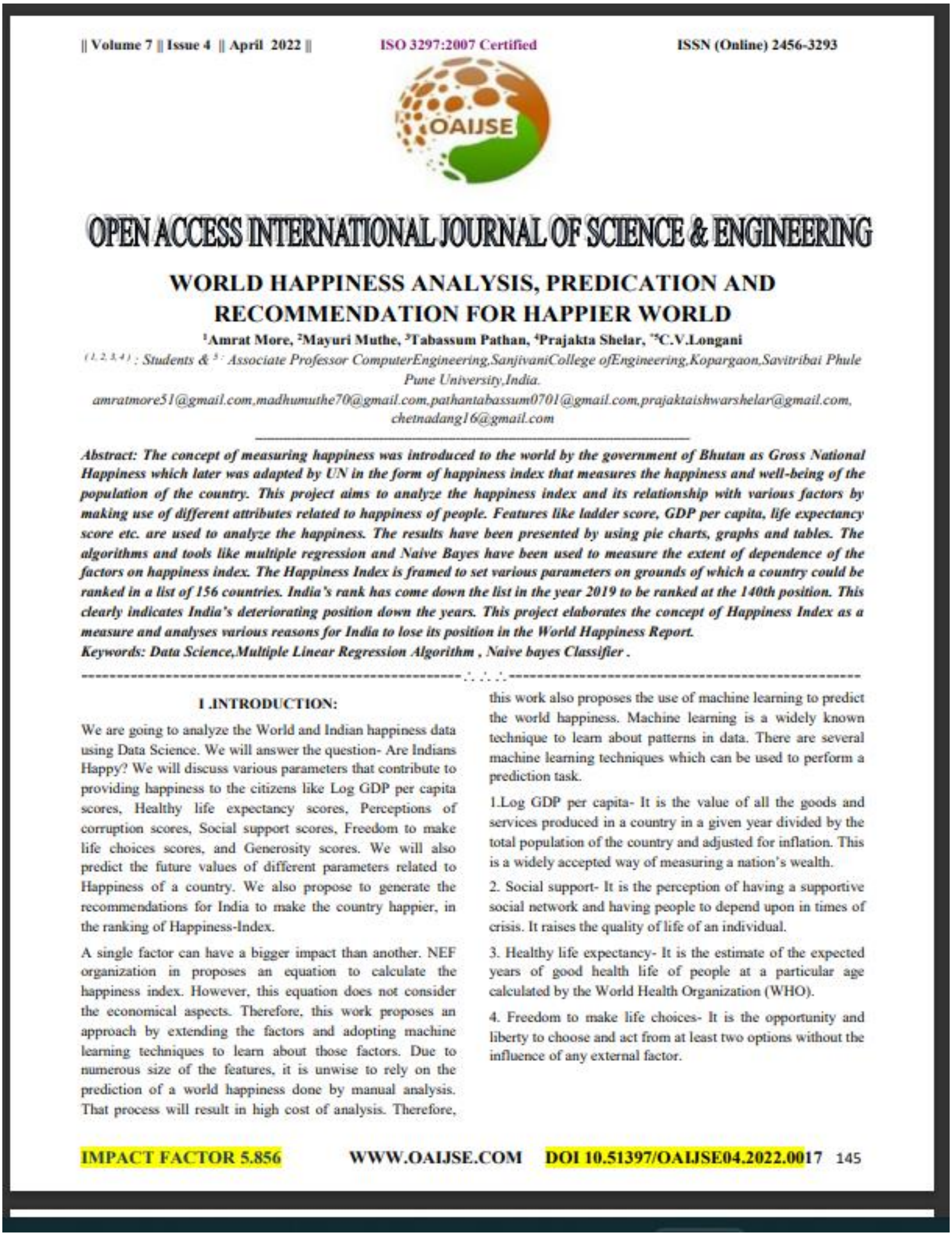
OAIJSE
www.oaijse.com

Editor-in-Chief
OAIJSE

DOI 10.51397/OAIJSE04.2022.00017

Figure C.4: OAIJSE Certificate of Publication

C.5 published paper in journal



|| Volume 7 || Issue 4 || April 2022 || ISO 3297:2007 Certified ISSN (Online) 2456-3293

OAIJSE

OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

WORLD HAPPINESS ANALYSIS, PREDICATION AND RECOMMENDATION FOR HAPPIER WORLD

[1]Amrat More, [2]Mayuri Muthe, [3]Tabassum Pathan, [4]Prajakta Shelar, [*5]C.V.Longani

(1, 2, 3, 4) : Students & [5] : Associate Professor ComputerEngineering,SanjivaniCollege ofEngineering,Kopargaon,Savitribai Phule Pune University,India.

amratmore51@gmail.com,madhumuthe70@gmail.com,pathantabassum0701@gmail.com,prajaktaishwarshelar@gmail.com, chetnadang16@gmail.com

Abstract: The concept of measuring happiness was introduced to the world by the government of Bhutan as Gross National Happiness which later was adapted by UN in the form of happiness index that measures the happiness and well-being of the population of the country. This project aims to analyze the happiness index and its relationship with various factors by making use of different attributes related to happiness of people. Features like ladder score, GDP per capita, life expectancy score etc. are used to analyze the happiness. The results have been presented by using pie charts, graphs and tables. The algorithms and tools like multiple regression and Naive Bayes have been used to measure the extent of dependence of the factors on happiness index. The Happiness Index is framed to set various parameters on grounds of which a country could be ranked in a list of 156 countries. India's rank has come down the list in the year 2019 to be ranked at the 140th position. This clearly indicates India's deteriorating position down the years. This project elaborates the concept of Happiness Index as a measure and analyses various reasons for India to lose its position in the World Happiness Report.

Keywords: Data Science,Multiple Linear Regression Algorithm , Naive bayes Classifier .

I .INTRODUCTION:

We are going to analyze the World and Indian happiness data using Data Science. We will answer the question- Are Indians Happy? We will discuss various parameters that contribute to providing happiness to the citizens like Log GDP per capita scores, Healthy life expectancy scores, Perceptions of corruption scores, Social support scores, Freedom to make life choices scores, and Generosity scores. We will also predict the future values of different parameters related to Happiness of a country. We also propose to generate the recommendations for India to make the country happier, in the ranking of Happiness-Index.

A single factor can have a bigger impact than another. NEF organization in proposes an equation to calculate the happiness index. However, this equation does not consider the economical aspects. Therefore, this work proposes an approach by extending the factors and adopting machine learning techniques to learn about those factors. Due to numerous size of the features, it is unwise to rely on the prediction of a world happiness done by manual analysis. That process will result in high cost of analysis. Therefore, this work also proposes the use of machine learning to predict the world happiness. Machine learning is a widely known technique to learn about patterns in data. There are several machine learning techniques which can be used to perform a prediction task.

1.Log GDP per capita- It is the value of all the goods and services produced in a country in a given year divided by the total population of the country and adjusted for inflation. This is a widely accepted way of measuring a nation's wealth.

2. Social support- It is the perception of having a supportive social network and having people to depend upon in times of crisis. It raises the quality of life of an individual.

3. Healthy life expectancy- It is the estimate of the expected years of good health life of people at a particular age calculated by the World Health Organization (WHO).

4. Freedom to make life choices- It is the opportunity and liberty to choose and act from at least two options without the influence of any external factor.

IMPACT FACTOR 5.856 WWW.OAIJSE.COM DOI 10.51397/OAIJSE04.2022.0017 145

Figure C.5: published paper Page 1