HAPPINESS INDEX ANALYSIS

A PROJECT REPORT

Submitted by

Sakshi Sawalikar 1921321372146

Adhiraj Jarwal 1921321246050

Deepali Bolsekar 1921321372074

in partial fulfillment of the award of the degree

of

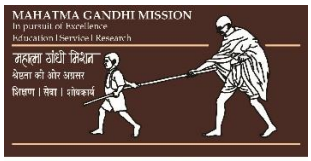
Bachelor of Technology

IN

INFORMATION TECHNOLOGY

Guided by

Mr. Kiran Sonkamble



JAWAHARLAL NEHRU ENGINEERING COLLEGE

Department of Information Technology

MGM's Jawaharlal Nehru Engineering College, Aurangabad

YEAR 2022-2023

CERTIFICATE

This is to certify that the project report

HAPPINESS INDEX ANALYSIS

Submitted by

Sakshi Sawalikar (192132137214), Adhiraj Jarwal (1921321246050), Deepali Bolsekar (1921321372074)

is a bonafide work carried out by them under the supervision of Mr. Kira Sonkamble and it is approved for the partial fulfillment of the award of Bachelor of Technology (Information Technology), from Dr. Babasaheb Ambedkar Technical University Lonere, MS, India.

Date: /12/2022

Mr. Kiran Sonkamble
Guide
Dept. of Information Technology

Dr. S. C. Tamane
Head of Department
Dept. of Information Technology

Dr. H. H. Shinde
Principal
MGM's Jawaharlal Nehru Engineering College, Aurangabad.

CONTENTS

List of Abbreviations

- SVM = Support Vector Machine
- KNN = K-Nearest Neighbor

List of Figures

List of Tables

Abstract

This report examines the Happiness Indexing of the students studying in the college. This report explains how the happiness index of the students studying in college can be calculated using machine learning algorithms. The different algorithms are used to determine the factors affecting happiness in any student's life. These factors can be related to their college, personal, and social life. Different machine learning algorithms are used to reach the goal of maximum accuracy. The variety of questions asked the students pursuing their engineering course. These questions are based on different aspects to determine the behavior of answers collected from students. These questions help collect the required data as input for this machine-learning survey. The data used in the happiness index analysis project is original and free from any reference. The data collected from the students are further examined for analysis and machine learning algorithms are applied to them to perform the happiness index analysis. Different algorithms like multiple linear regression, KNN algorithm, Random Forest algorithm, and Support vector machine algorithm are used.

Introduction

Quantitative research has been carried out to calculate and analyze the happiness index of the respondents, as well as to understand the various factors on which the happiness index depends. We are going to use Google forms to collect the data. The data collected will be inserted in a Microsoft Excel spreadsheet and analyzed further spreadsheet and further, it was analyzed using the measures of central tendency- Mean and Median. The results were presented by using pie charts and graphs. In addition, the tools of ANOVA and multiple regression were also used to test the hypothesis generated for the first objective.

Firstly, we are going to calculate the happiness index of collected data (of an individual student), and then we are going to train our model so that it can predict the happiness index in the future upcoming.

Secondly, we are also classifying the individual student as unhappy, happy, and narrowly happy.

What is an ANOVA?

An ANOVA test is a type of statistical test used to determine if there is a statistically significant difference between two or more categorical groups by testing for differences of means using a variance.

Another Key part of ANOVA is that it splits the independent variable into two or more groups. For example, one or more groups might be expected to influence the dependent variable while the other group is used as a control group, and is not expected to influence the dependent variable.

What is Data Science?

Back in the day, Businesses and other Institutions were able to store most of their data in **Microsoft Excel Sheets**. Even the modest Business Intelligence tools were capable of analyzing and processing this data. The presence of a lesser amount of data made the handling and managing of data easier. But with the passage of time, the amount of data produced every day kept increasing.

You'd have come across this study by Forbes which states that nearly **2.5 Quintillion Bytes** of data are generated every day. According to Raconteur, by 2025, 463 Exabytes of data are expected to be generated every day globally.

This is the scale of data that will be available to be analyzed in the future. For processing data of this magnitude, traditional Spreadsheets and conventional Business Intelligence tools are not going to come in handy. You need sophisticated Data Infrastructure and cutting-edge tools/technologies to process data of such magnitudes. This is where Data Science comes into the picture.

Data Science is all about using data to create as much impact as possible for your company. The impact can be in the form of multiple things. It could be in the form of viewing insights of the audience that **Netflix** mines to produce an original series or in the form of video recommendations for YouTube. Now to do those things, you need to make complicated Models, write code and make use of Data Visualization tools.

The Journal of Data Science described Data Science as "**almost everything that has something to do with data: Collecting, Analyzing, Modeling…… yet the most important part is its applications — all sorts of applications**". Yes, all sorts of applications like Machine Learning. Machine Learning, Deep Learning, and Artificial Intelligence are all used in Data Science for the analysis of data and extraction of useful information from it.

Importance of Data Science

The amount of data has never been this much huge as they are in today's age. Similarly, the complexity of the data is also increasing with time. Today a Data Scientist is simultaneously dealing with a variety of data formats to derive predictions and reach conclusions. This increasing volume and growing complexity gave rise to a need for such techniques, methods, or tools that can help Data Science Data Analysts to analyze more efficiently and quickly.

To fulfill this need, the researchers discovered Data Science, a combination of complex Machine Learning techniques integrated with a variety of tools to help the Data Science Data Analysts in decision making, finding the new patterns, and discovering new ways of Predictive Analysis.

What is Machine Learning?

It is now possible to Train Machines with a **Data-Driven** approach. On a wider spectrum, if you think of Artificial Intelligence as the main umbrella, Machine Learning is a subset of Artificial Intelligence. Machine Learning, a set of Algorithms, gives Machines or Computers the ability to learn from data on their own without any human intervention.

The idea behind Machine Learning is that you teach and Train Machines by feeding them data and defining features. Computers **learn, grow, adapt, and develop** by themselves when they are fed with new and relevant data, without relying on explicit programming. Without data, there is very little that Machines can learn. The Machine observes the dataset, identifies patterns in it, learns automatically from the behavior, and makes predictions.

It is the Machine Learning technology that Online Recommendation Engines use to offer relevant recommendations to the user, be it **YouTube Video Recommendations** or Facebo**ok Friend Recommendations**. One of the most recent technologies, **Google's Self Driving Car** also makes use of Machine Learning Algorithms to understand the patterns and definitions, learn automatically, and execute the operation.

What are the Applications of Machine Learning in Data Science?

Listed below are some of the most popular applications of Machine Learning in Data Science:

Real-Time Navigation: Google Maps is one of the most commonly used Real-Time Navigation applications. *But have you ever wondered why despite being of the usual traffic, you are on the fastest route?* It is because of the data received from people currently using this service, and the database of Historical Traffic Data. Everyone who uses this service contributes to making this application more accurate. When you open the application, it constantly sends the data back to Google, providing information about the route being traveled and traffic patterns at any given time of the day. All the information given by the number of users using the application on regular basis has given Google a huge database of traffic data which allows Google Maps not

only to track the traffic at that instance but also predicts what will happen if you continue in the same route.

Image Recognition: Image Recognition is one of the most common applications of Machine Learning in Data Science. Image Recognition is used to identify objects, persons, places, etc. The most popular use cases of this application are Face Recognition in Smartphones, Automatic Friends Tagging Suggestions on Facebook, etc.

Product Recommendation: Product Recommendation is profoundly used by eCommerce and Entertainment companies like Amazon, Netflix, Hotstar, etc. They use various Machine Learning algorithms on the data collected from you to recommend products or services that you might be interested in.

Speech Recognition: Speech Recognition is a process of translating spoken utterances into text. This text can be in terms of words, syllables, sub-word units, or even characters. Some of the well-known examples are Siri, Google Assistant, Youtube Closed Captioning, etc.

What are the Challenges of Machine Learning in Data Science?

Machine Learning in Data science has revolutionized the face of the industries. It has helped companies to take intelligent decisions to grow their business. But it still faces a couple of challenges that a Data Scientist must consider. Listed below are the Top 3 challenges of Machine Learning in Data Science:

Lack of Training Data: Data is the core of any Machine Learning model. However, it is extremely difficult and expensive to obtain labeled data. Training a Machine Learning model without a large amount of data is something that haunts every Data Scientist. Transfer Learning is one of the methods to solve this problem. It enables the model to utilize knowledge from previously learned tasks and applies them to the new related ones. Self-Supervised Learning is another way to solve this problem. It opens up a huge opportunity for better utilizing large amounts of unlabeled data.

Discrepancies between Data: The second challenge is that there are usually some discrepancies between the training data and production data. Sometimes the model works well in your prototyping environment but fails to generalize in real-world cases. For example, the model may work well in one country but fail in another due to geographical differences, the model may work in winter but fail in summer due to seasonal differences, and the model may work on mobile but fail on desktop due to user differences, etc. To solve this problem, you need to be very careful while collecting your training data. To make it as close to your target domain as possible, you need to keep updating your model frequently.

- **Model Scalability:** This is one of the major challenges that industries face. As a Data Scientist, you need to make sure that your model can is fast but at the same time also not very bulky. One of the solutions to this problem is Post-Training Quantization. It is a conversion technique that reduces the model size but at the same time improves CPU and hardware Accelerator Latency, with a little degradation in your model accuracy.

Scope

The Happiness Index Analysis is used to calculate the degree of satisfaction among the students in their day-to-day life. The Happiness Index Analysis project can be helpful to find the most likely factors which affect any individual or group of individuals. Finding the affecting factors can help the source provider to get to know the importance of affecting factors on the elements and can enhance the quality of those affecting factors. This type of analysis helps to improve the quality in any area of improvement.

Research Objectives

Quantitative research has been carried out to calculate and analyze the happiness index of respondents and factors that affect the individual student. The machine learning algorithm is used to generate a hypothesis along with the ANOVA tool.

Firstly, we collected the data from students using Google form and calculated the happiness index, and trained the model to predict the happiness index in the upcoming future.

Secondly, we classified students into different categories unhappy, narrowly happy, extensively happy, and deeply happy.

We created a Google form that includes a total of 25 questions related to student's social life, college life, and personal life. A scale from 0 to 10 has been given for every question so, that students can scale their experience using that scale. This Google form is circulated among different branches and students studying in their respective years. The responses are collected in an excel sheet to apply the formula to find the score of individual students to calculate percentages.

This score is further used in the algorithms which are applied to find the happiness index. After the data collection different machine-learning techniques are used to calculate and analyze the happiness index. The machine learning techniques used in the project are KNN, SVM, Linear regression, Random Forest, Decision Tree, etc.

The best machine learning technique which gives more accuracy is used as the final technique to get the most accurate happiness index.

Literature Survey

Sr no.	Name	Author	Description
1.	Happiness Index-The Footsteps towards sustainable development	Mevawala Jency, Post Graduate student Country ad Town Plannning, Sarvajanik College of Engineering and Technology, Surat, Gujarat.(2019)	The Happiness Index is a comprehensive survey instrument that evaluates happiness, wellbeing, and aspects of sustainability and resilience. The Happiness Alliance developed the Happiness Index to provide a survey instrument to community organizers, researchers, and others seeking to use a subjective well-being index and data. It is the only instrument of its kind freely available worldwide and translated into over ten languages. This instrument can be used to measure satisfaction with life and the conditions of life. It can also be used to define income inequality, trust in government, and sense of community and other aspects of wellbeing within specific demographics of a population. Through, this review paper it will be understood what is happiness index? By which method one can measureGross happiness Index? The indicators of GNH (Gross National Happiness) given by Bhutan.
2.	Applying machine learning to predict Happiness: A case study of 20	Yu Tan, Charuk Singhapreecha, Woraphon Yamaka, Chiang	Happiness is the current important research issues in psychology and social sciences, which is affecting people's daily lifestyle, work habits and thinking patterns, it

	countries	Mai University, Thailand.(2022)	also provides guidance for government policy making. However, in the current analysis of happiness, there are many challenges in variable selection and prediction. Due to the large personal differences and differences in determinants of the experience of happiness, this undoubtedly makes the modeling of happiness more difficult. Based on reviewing several academic literatures, this paper uses questionnaire data from 20 countries in World Values Survey Database, then uses machine learning methods to compare traditional regression approach with machine learning regression approach to predict happiness. Finally, some determinants variables were found. To a certain extent, the elastic net method applied in model was successfully used to predict happiness, social factors, economic factors and personal factors affect the modeling and prediction of happiness in varying degrees, which brought new opportunities for the development of related theories and practices at the study of happiness.
3.	Analysing Happiness Index as a measure along with its parameters and strategies for improving india's rank in world	Sarah Ahtesham, Schol of Business Studies, Vivekananad Institute of Professional Studies,	Measuring happiness in quantifiable terms is a global phenomenon lately. United Nation's World Happiness Report (WHR) is one such means to analyse the level of subjective wellbeing that countries across the world are living with. The Happiness Index is framed to set various

	happiness report	India.(2020)	parameters on grounds of which a country could be ranked in a list of 156 countries. India's rank has come down the list this year (2019) to be ranked at the 140th position. This clearly indicates India's deteriorating position down the years. 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 suggestions.
4.	Happiness index methodology	Laura Musikanski (Happiness Alliance), Scott Cloutier, Erica Bejarano (), Davi Briggs, Julia Colbert, Gracie Strasser, Steven Russell (Arizona State University).(2017)	The Happiness Index is a comprehensive survey instrument that assesses happiness, well- being, and aspects of sustainability and resilience. The Happiness Alliance developed the Happiness Index to provide a survey instrument to community organizers, researchers, and others seeking to use a subjective well-being index and data. It is the only instrument of its kind freely available worldwide and translated into over ten languages. This instrument can be used to measure satisfaction with life and the conditions of life. It can also be used to define income inequality, trust in government, sense of community and other aspects of well-being within specific demographics of a population. This manuscript documents the development the Happiness Index between 2011 and 2015, and includes suggestions for implementation.

5.	Sentiment analysis using product review data	Xing Fang, Justin Zhan. (2015)	Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gain much attention in recent years. In this paper, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general process for sentiment polarity categorization is proposed with detailed process descriptions. Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. At last, we also give insight into our future work on sentiment analysis.

Problem Statement

The happiness index can help to find the factors affecting the current state of any country or region. The happiness index is the calculation and analysis of different factors affecting the individuals in that region. This project can be done on any level. In this experiment, the problem statement is to be considered as calculating the happiness index of the students studying in college. To calculate the happiness index, find different factors affecting students, and collect the data of as many students as possible to get accuracy.

Major Inputs

Student's data collected through Google form

Major Outputs

A trained machine learning model that will predict the happiness index score based on different attribute values using a regression algorithm.

And a classifier that will classify the happiness index score into four categories

That is happy, unhappy, extremely happy, and deeply happy.

Again we have used a multilinear regression algorithm as a regressor and as a classifier, we have used k-nearest neighbor, decision tree, random forest, and support vector machine algorithms.

We have also calculated the accuracy and confusion matrix and mean squared error for each machine-learning model.

And for the data visualization part different graphs are drawn like pie charts, bar graphs, scatter plots and heat maps.

Hardware Resources Required

Computer device

Software Resources Required

Jupyter notebook

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

Area of Project

Educational institution

Software Requirements Specification

Jupyter Notebook Version 6.4.8

A browser-based tool for interactive authoring of documents that combine explanatory text, mathematics, and computations including inputs and outputs of computations.

Implementation Details

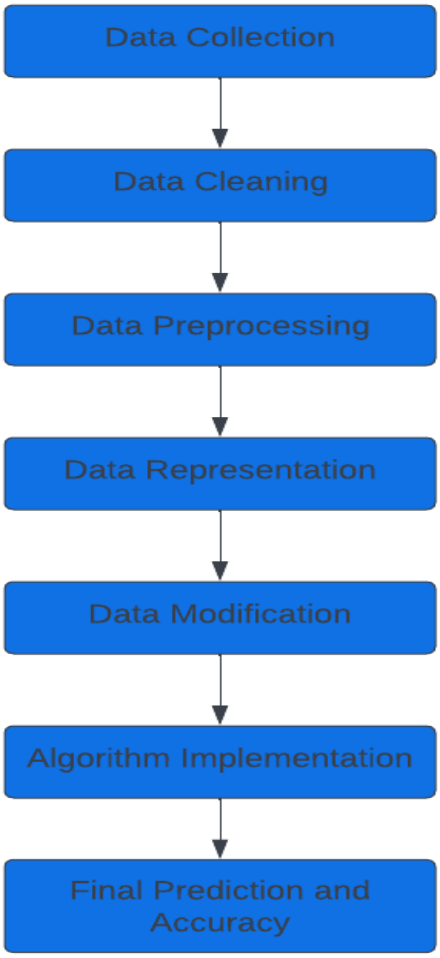


Figure 4.1.1(Implementation flow)

Data Collection:

This process includes the data generation which we have generated using the Google form and circulating them among the college students and collecting and structuring that data manually.

The questions that we have asked to students are

- Importance of family in your life.
- Importance of friends in your life.
- Issues faced due to college politics
- College timing is appropriate.
- Facing issues due to discrimination.
- Your physical fitness level.
- Your mental health level.
- Your social trust level.
- Liberty (Freedom to take decision)
- Feeling of being treated equal among siblings.
- Feeling of being treated equal among all students.
- Your self confidence in your college
- How is your Academic Performance?
- Your participation in extra curricular activities
- Relationship satisfaction (Romantic relationship)
- Level of your academic stress.
- Financial situation of family
- Effect of back bitching.
- Effect of favoritism in class
- Suffering you have endured as a result of ragging
- How much teacher's scolding affects you?
- How supportive is your college regarding your hobbies?
- How much opportunities do you get to follow the tradition and culture at your college during events?

- How balanced are you between leisure, enjoyment, and rushing?
- How do you feel about volunteering, being a part of the community, and feeling safe?

Data Cleaning:

In this process, the data collected from the students through Google form is cleaned so, that it should not have any null value and can provide that clean data to the algorithm for applying the techniques. Data cleaning helps to reduce the unnecessary null data as well as get fewer errors while performing the operations.

Data Preprocessing:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Data Representation:

In this project, the data is represented in the form of a table to read the data for confirming the appropriate data cleaning has been done.

The libraries like seaborn and matplotlib are used to represent the data.

Matplotlib is a fantastic visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. John Hunter introduced it in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.We have used matplotlib to draw bar graphs and pie charts.

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

A Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges. The area of the wedge is determined by the length of the arc of the wedge. The area of a wedge represents the relative percentage of that part with respect to whole data. Pie charts are commonly used in business presentations like sales, operations, survey results, resources, etc as they provide a quick summary.

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of the matplotlib library and is closely integrated to pandas' data structures.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

We have used seaborn to draw multiple graphs taking each feature.

Algorithm Implementation:

- We have used classifier and regressor algorithms in this project.
- The Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.
- The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.
- Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.
- In classification algorithm, a discrete output function(y) is mapped to input variable(x).
- y=f(x), where y = categorical output

Multilinear regression:

Multiple Linear Regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data. The steps to perform multiple linear Regression are almost similar to that of simple linear Regression. The Difference Lies in the evaluation. We can use it to find out which factor has the highest impact on the predicted output and how different variables relate to each other.

Here : $Y = b0 + b1 * x1 + b2 * x2 + b3 * x3 + \ldots\ldots bn * xn$

Y = Dependent variable and x1, x2, x3, …… xn = multiple independent variables

Assumption of Regression Model :

Linearity: The relationship between dependent and independent variables should be linear.

Homoscedasticity: Constant variance of the errors should be maintained.

Multivariate normality: Multiple Regression assumes that the residuals are normally distributed.

Lack of Multicollinearity: It is assumed that there is little or no multicollinearity in the data.

Method of Building Models :

All-in

Backward-Elimination

Forward Selection

Bidirectional Elimination

Score Comparison

Backward-Elimination :

Step#1: Select a significant level to start in the model.

Step#2: Fit the full model with all possible predictors.

Step#3: Consider the predictor with the highest P-value. If P > SL go to STEP 4, otherwise the model is Ready.

Step#4: Remove the predictor.

Step#5: Fit the model without this variable.

Forward-Selection :
Step#1 : Select a significance level to enter the model(e.g. SL = 0.05)

Step #2: Fit all simple regression models y~ x(n). Select the one with the lowest P-value.

Step #3: Keep this variable and fit all possible models with one extra predictor added to theone(s) you already have.

Step #4: Consider the predictor with the lowest P-value. If P < SL, go to Step #3, otherwise the model is Ready.

Steps Involved in any Multiple Linear Regression Model
Step #1: Data Pre Processing

- Importing The Libraries.
- Importing the Data Set.
- Encoding the Categorical Data.
- Avoiding the Dummy Variable Trap.
- Splitting the Data set into Training Set and Test Set.

Step#2: Fitting Multiple Linear Regression to the Training set
Step #3: Predict the Test set results.

KNN Algorithm:

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- Step 1: Select the number K of the neighbors
- Step 2: Calculate the Euclidean distance of K number of neighbors
- Step 3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step 4: Among these k neighbors, count the number of the data points in each category.
- Step 5: Assign the new data points to that category for which the number of neighbor is maximum.
- Step 6: Our model is ready.

Naïve Bayes Algorithm:

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Baye's Theorem.

Bayes' Theorem:

- Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 4.1.2 (Bayes' Theorem)

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example: Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

SVM Algorithm:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed a Support Vector Machine.

Selecting the best hyper-plane:
One reasonable choice as the best hyperplane is the one that represents the largest representation or margin between the two classes.

SVM Kernel:
The SVM kernel is a function that takes low dimensional input space and transforms it into higher dimensional space, i.e. it converts a non-separable problem to a separable problem. It is mostly useful in non-linear separation problems. Simply, put the kernel, does some extremely complex data transformations and then finds out the process to separate the data based on the labels or outputs defined.

Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions for Random Forest:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.

- It can also maintain accuracy when a large proportion of data is missing.

How does the Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and the second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step 1: Select random K data points from the training set.

Step 2: Build the decision trees associated with the selected data points (Subsets).

Step 3: Choose the number N for the decision trees that you want to build.

Step 4: Repeat Steps 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Different Modules and Their Working

- **Pandas**

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

- **Numpy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

- **Maltplotlib**

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.

We have used matplotlib library to draw bar graph between total no of students from each branch .

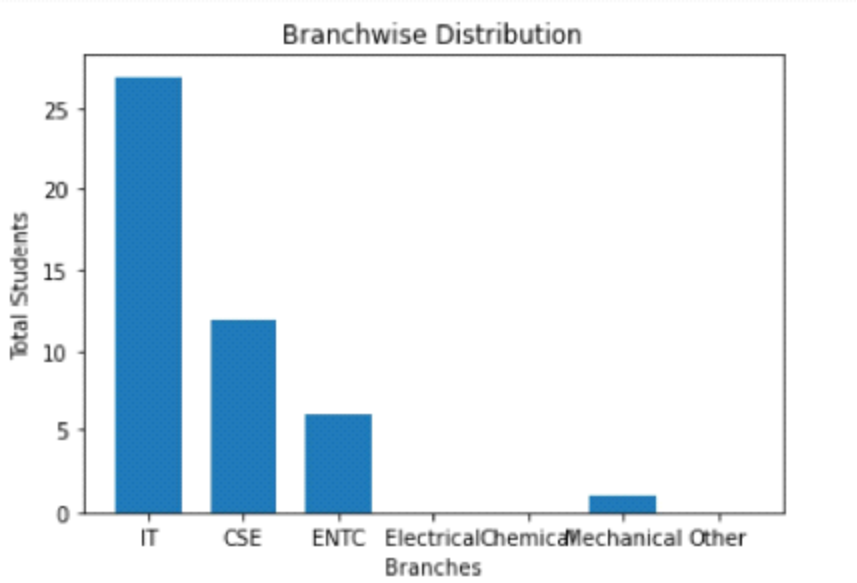


Figure 5.1.1(Branch-wise Distribution)

Again we have used pie chart to represent total no of happy unhappy ,narrowly happy and extreamly happy people.

The criteria which we have used to classify the students into four categories is

Less than 50 percent are marked as unhappy,

Greater than 50 but lesser than 65 are marked as narrowly happy,

Greater than 65 but lesser than 76 are marked as extensively happy,

And in the last greater than 76 are marked as deeply happy.

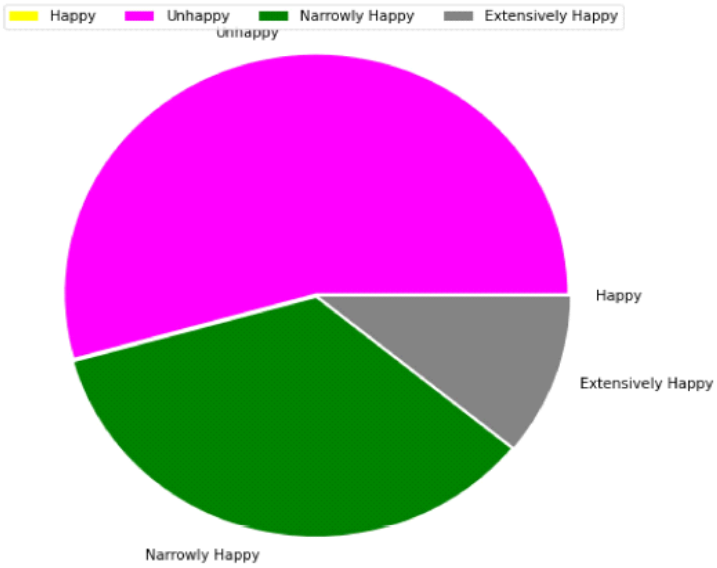


Figure 5.1.2 (Student's happiness Classification)

- **Seaborn**

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

We have used seaborn to plot scatter plot between age vs score i.e age of student and their corresponding happiness index score.

```
sns.scatterplot(x='Age',
                y='Score', data=df)
```

```
<AxesSubplot:xlabel='Age', ylabel='Score'>
```

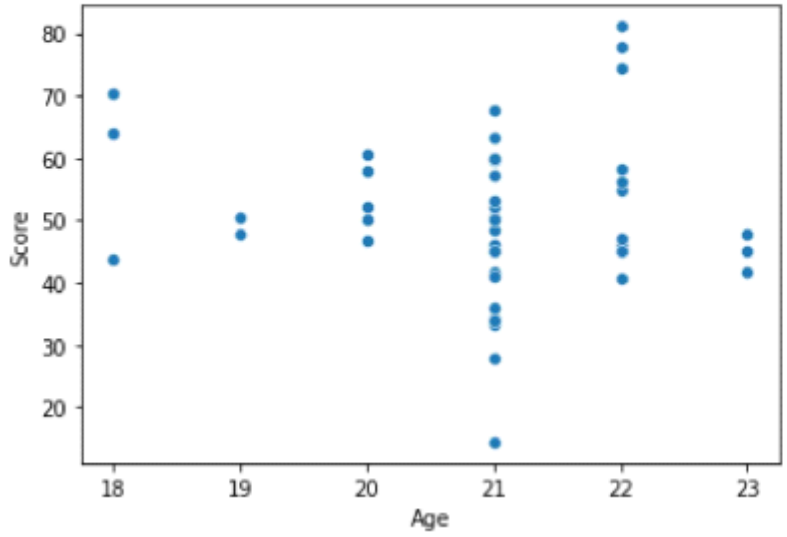


Figure 5.1.3(Age Vs Score)

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred. Heatmap is also defined by the name of the shading matrix. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap() function.

seaborn.heatmap()

Syntax: seaborn.heatmap(data, *, vmin=None, vmax=None, cmap=None, center=None, annot_kws=None, linewidths=0, linecolor='white', cbar=True, **kwargs)

Important Parameters:

•data: 2D dataset that can be coerced into an ndarray.
•vmin, vmax: Values to anchor the colormap, otherwise they are inferred from the data and other keyword arguments.
•cmap: The mapping from data values to color space.
•center: The value at which to center the colormap when plotting divergent data.
•annot: If True, write the data value in each cell.
•fmt: String formatting code to use when adding annotations.
•linewidths: Width of the lines that will divide each cell.
•linecolor: Color of the lines that will divide each cell.
•cbar: Whether to draw a colorbar.
All the parameters except data are optional.

We have used heatmap to show the results .

```
correlation_matrix = df.corr()
sns.heatmap(data=correlation_matrix, annot= True)
# annot = True to print the values inside the sqaure
```

```
<AxesSubplot:>
```

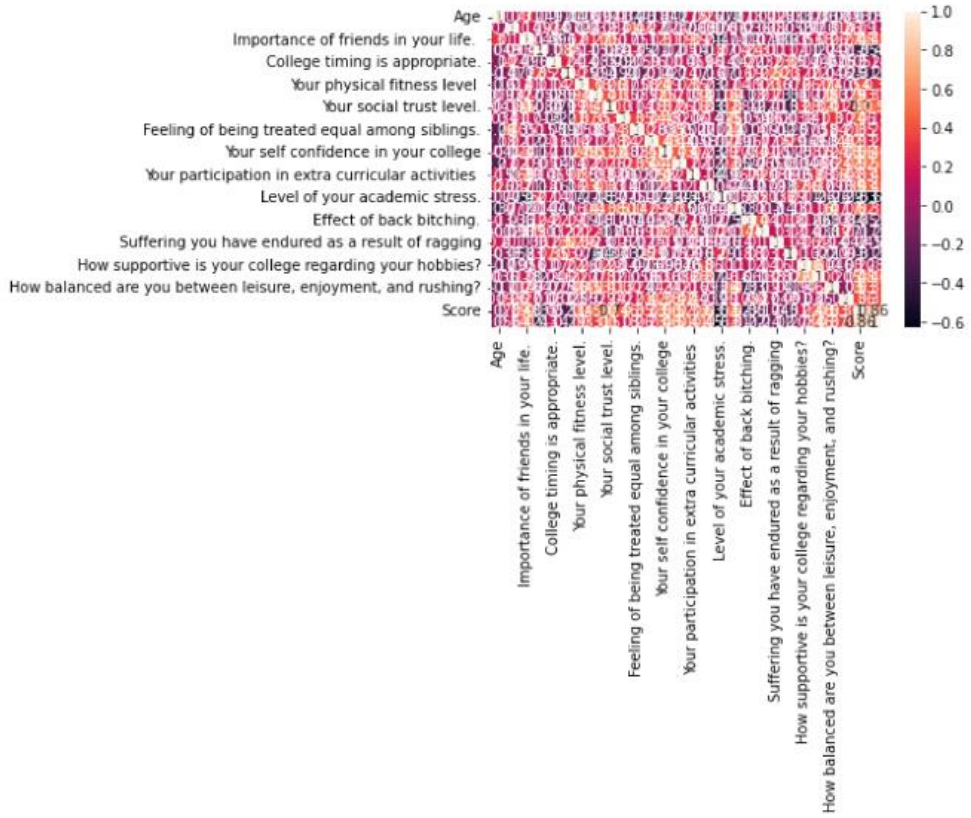


Figure 5.1.4 (Heat Map)

- **sklearn**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Scikit-learn alias sklearn is the most useful and robust library for machine learning in Python. The scikit-learn library provides us with the model_selection module in which we have the splitter function train_test_split().

Syntax:
train_test_split(*arrays, test_size=None, train_size=None, random_state=None, shuffle=True, stratify=None)

Parameters:
•*arrays: inputs such as lists, arrays, data frames, or matrices
•test_size: this is a float value whose value ranges between 0.0 and 1.0. it represents the proportion of our test size. its default value is none.
•train_size: this is a float value whose value ranges between 0.0 and 1.0. it represents the proportion of our train size. its default value is none.
•random_state: this parameter is used to control the shuffling applied to the data before applying the split. it acts as a seed.
•shuffle: This parameter is used to shuffle the data before splitting. Its default value is true.
•stratify: This parameter is used to split the data in a stratified fashion.

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the

related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. Some features of Confusion matrix are given below:

For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.
The matrix is divided into two dimensions, that are predicted values and actual values along with the total number of predictions.
Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
It looks like the below table:

Table 5.1.1(Prediction Table)

n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

The above table has the following cases:

- True Negative: Model has given prediction No, and the real or actual value was also No.
- True Positive: The model has predicted yes, and the actual value was also true.
- False Negative: The model has predicted no, but the actual value was Yes, it is also called as Type-II error.
- False Positive: The model has predicted Yes, but the actual value was No. It is also called a Type-I error.

Need for Confusion Matrix in Machine learning

It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.

It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.

With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

The best value is 1 and the worst value is 0.

What is Accuracy?

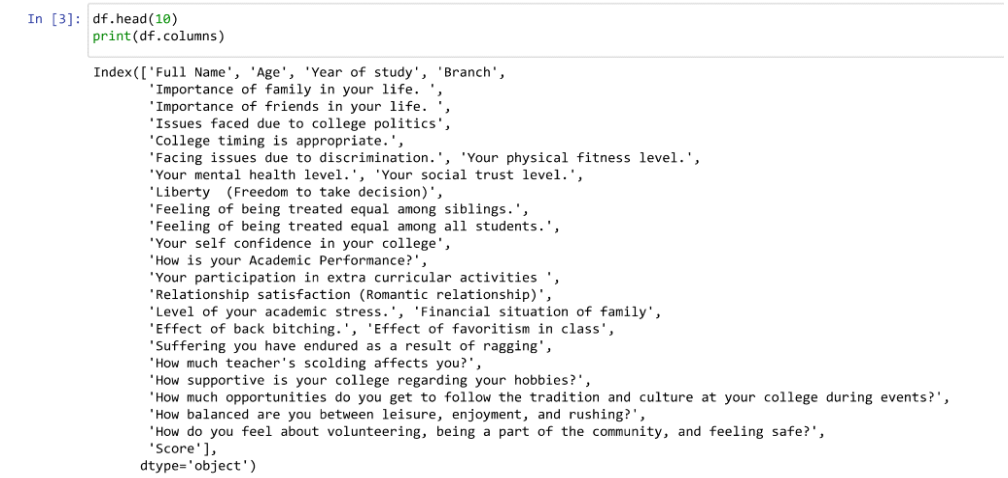
One of the widely used metrics that computes the performance of classification models is accuracy. The percentage of labels that our model successfully predicted is represented by accuracy. For instance, if our model accurately classified 80 of 100 labels, its accuracy would be 0.80.

Creating Function to Compute Accuracy Score

Let's create a Python function to compute the predicted values accuracy score, given that we already have the sample's true labels and the labels predicted the model.y

Output Screens

This image is showing the names of columns from excel sheet where all the data is collected.

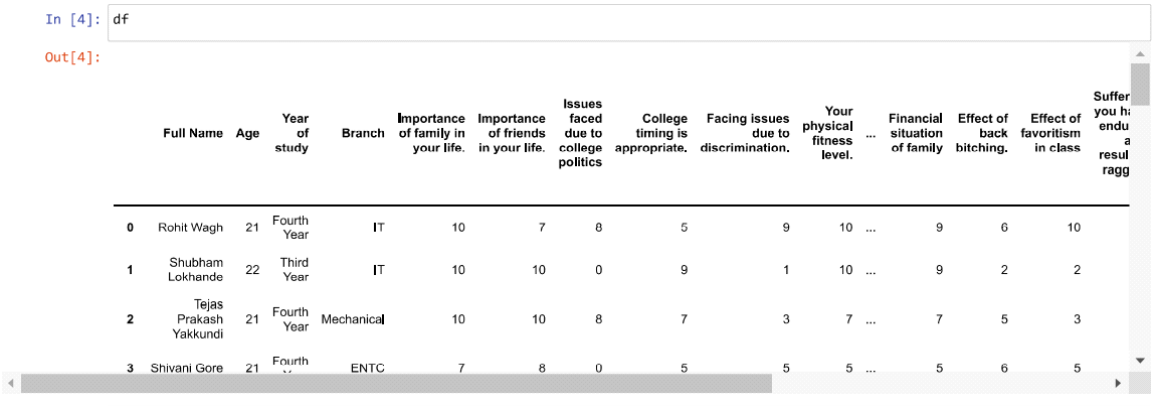


```
In [3]: df.head(10)
        print(df.columns)

Index(['Full Name', 'Age', 'Year of study', 'Branch',
       'Importance of family in your life. ',
       'Importance of friends in your life. ',
       'Issues faced due to college politics',
       'College timing is appropriate.',
       'Facing issues due to discrimination.', 'Your physical fitness level.',
       'Your mental health level.', 'Your social trust level.',
       'Liberty  (Freedom to take decision)',
       'Feeling of being treated equal among siblings.',
       'Feeling of being treated equal among all students.',
       'Your self confidence in your college',
       'How is your Academic Performance?',
       'Your participation in extra curricular activities ',
       'Relationship satisfaction (Romantic relationship)',
       'Level of your academic stress.', 'Financial situation of family',
       'Effect of back bitching.', 'Effect of favoritism in class',
       'Suffering you have endured as a result of ragging',
       'How much teacher's scolding affects you?',
       'How supportive is your college regarding your hobbies?',
       'How much opportunities do you get to follow the tradition and culture at your college during events?',
       'How balanced are you between leisure, enjoyment, and rushing?',
       'How do you feel about volunteering, being a part of the community, and feeling safe?',
       'Score'],
      dtype='object')
```

Figure 5.1.5 (display column name)

This image is showing the all column's representation with the respective data and values.



In [4]: df

Out[4]:

	Full Name	Age	Year of study	Branch	Importance of family in your life.	Importance of friends in your life.	Issues faced due to college politics	College timing is appropriate.	Facing issues due to discrimination.	Your physical fitness level.	...	Financial situation of family	Effect of back bitching.	Effect of favoritism in class	Suffer you h endu a resul ragg
0	Rohit Wagh	21	Fourth Year	IT	10	7	8	5	9	10	...	9	6	10	
1	Shubham Lokhande	22	Third Year	IT	10	10	0	9	1	10	...	9	2	2	
2	Tejas Prakash Yakkundi	21	Fourth Year	Mechanical	10	10	8	7	3	7	...	7	5	3	
3	Shivani Gore	21	Fourth	ENTC	7	8	0	5	5	5	...	5	6	5	

Figure 5.1.6 (Reading Data)

Image showing the scatter plat for age and score data.

```
sns.scatterplot(x='Age',
                y='Score', data=df)
```

```
<AxesSubplot:xlabel='Age', ylabel='Score'>
```

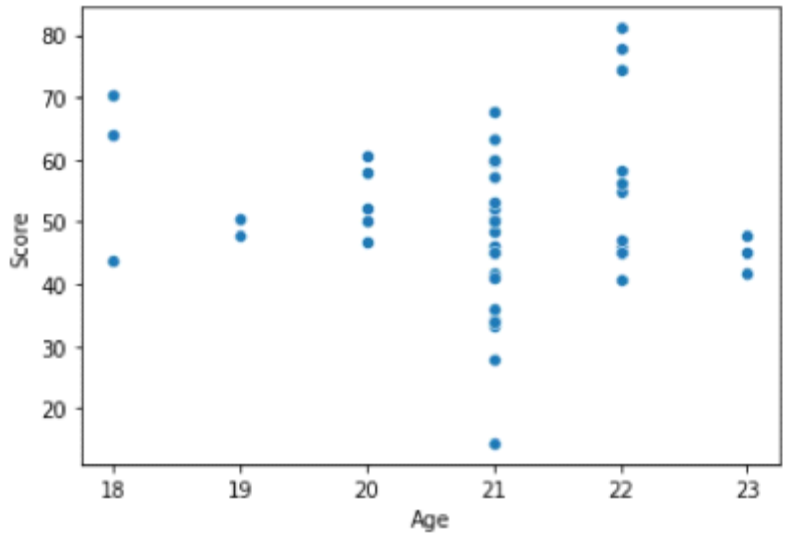


Figure 5.1.7(Age Vs Score)

The image is showing the accuracy of linear regression.

```
Linear Regression
mean_squared_error :  4.7782431973387e-29
mean_absolute_error :  4.821539992657823e-15
R2 score is : 1.0
```

Figure 5.1.8(Linear Regression Score)

The image is showing the accuracy of naïve bayes.

```
Naive Bayes
Accuracy: 0.9166666666666666
Confusion Matrix :
 [[6 0 0]
 [0 5 0]
 [0 1 0]]
```

Figure 5.1.9(Naïve Bayes accuracy)

The image is showing the accuracy of knn algorithm.

```
KNN Algorithm
Accuracy: 0.8333333333333334
Confusion Matrix :
 [[4 2 0]
 [0 5 0]
 [0 0 1]]
```

Figure 5.1.10(KNN Algorithm accuracy)

The image is showing the accuracy of random forest algorithm.

```
Random Forest algorithm
Accuracy: 0.9166666666666666
```

Figure 5.1.11(Random Forest Algorithm accuracy)

Analysis

- Data Analysis
    - The data from 45 students from different branches have been collected to analyze the factors which affect the happiness index of students.
    - Branch-wise distribution of students.

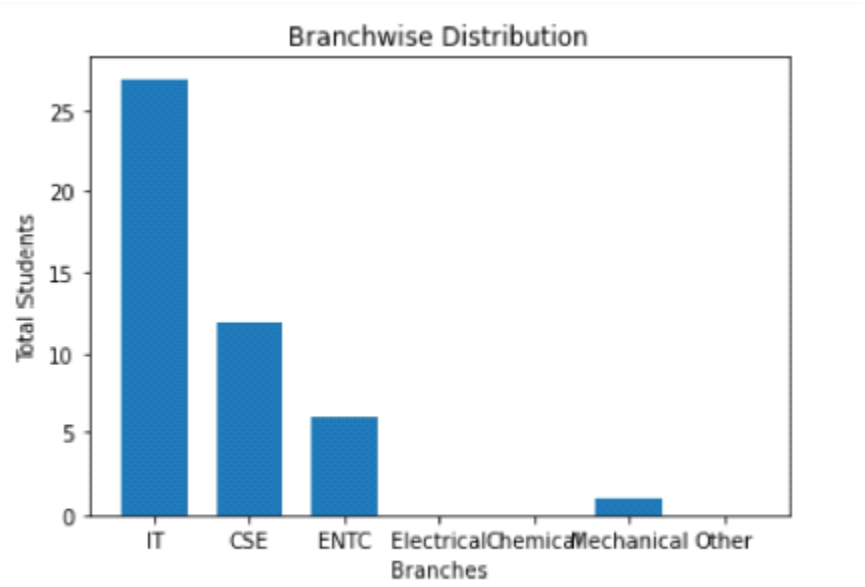


Figure 5.2.1(Branch-wise Distribution)

- Classification of the students according to their happiness index:
  Students have been categorized into different categories like happy, extensively happy, unhappy, and narrowly happy. The below pie chart shows the representation for the same.

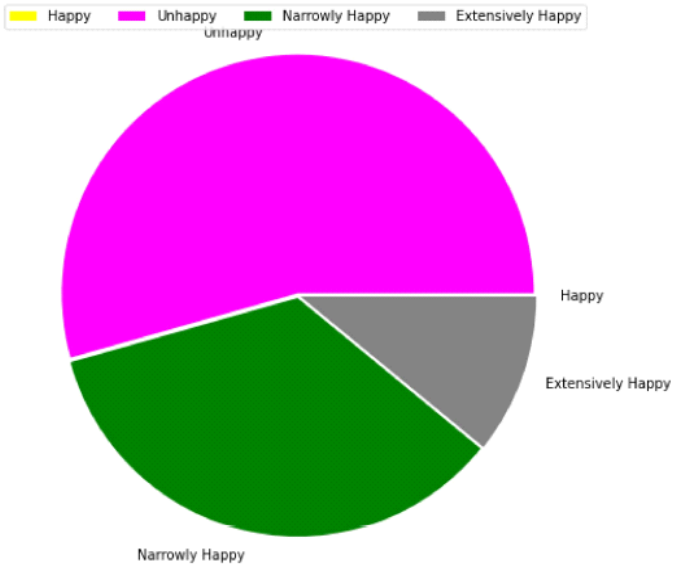


Figure 5.2.2(Student's happiness Classification)

- Algorithms used and their accuracy:

  The table below shows the accuracy of the algorithms after applying algorithms.

Table 5.2.1(Algorithm and Accuracy)

Sr no.	Name of Algorithms used	Accuracy
1.	KNN Algorithm	8.333333
2.	Naïve Bayes Algorithm	0.916667
3.	SVM Algorithm	0.833333
4.	Random Forest	0.916667

- **Heat map**

```
correlation_matrix = df.corr()
sns.heatmap(data=correlation_matrix, annot= True)
# annot = True to print the values inside the sqaure
```

```
<AxesSubplot:>
```

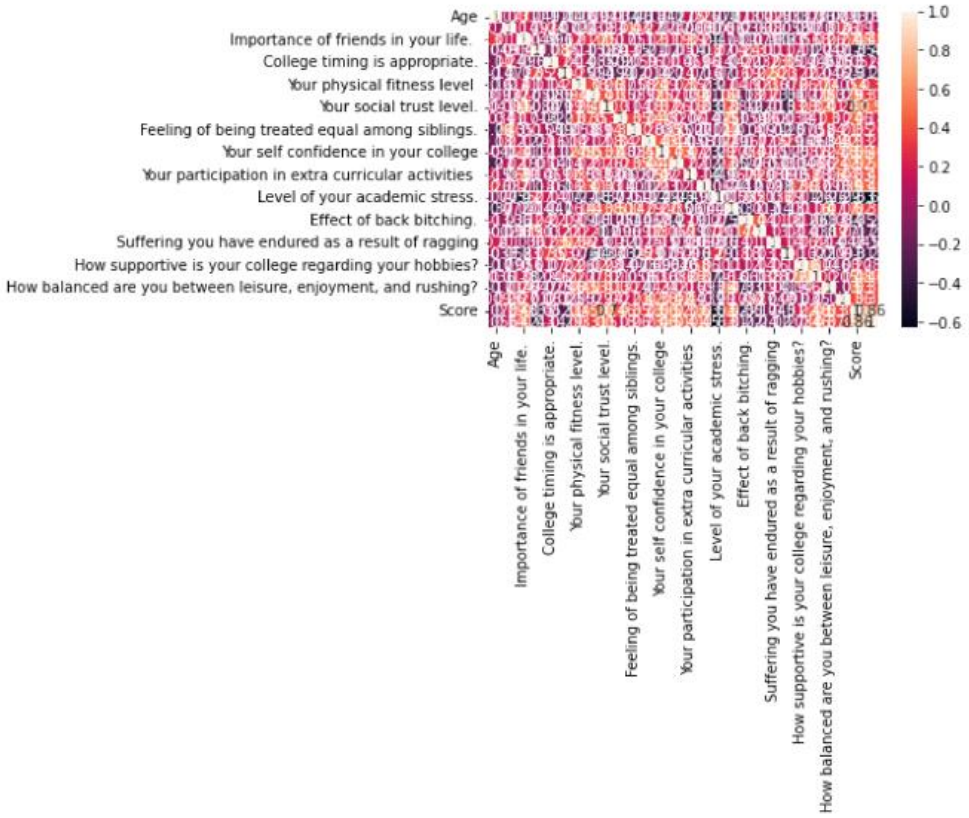


Figure 5.2.3(Age Vs Score Heat map)

Conclusion

Successfully implemented the analysis for finding the happiness index. The algorithms and various methodologies learned from different research papers were successfully applied to this project. The accuracy and the predictions are came out as we wanted it to. Additionally, the website for displaying accuracy for every algorithm the website is created for viewers to ease the task of directly viewing accuracy for any algorithm used in the Happiness Index Analysis.

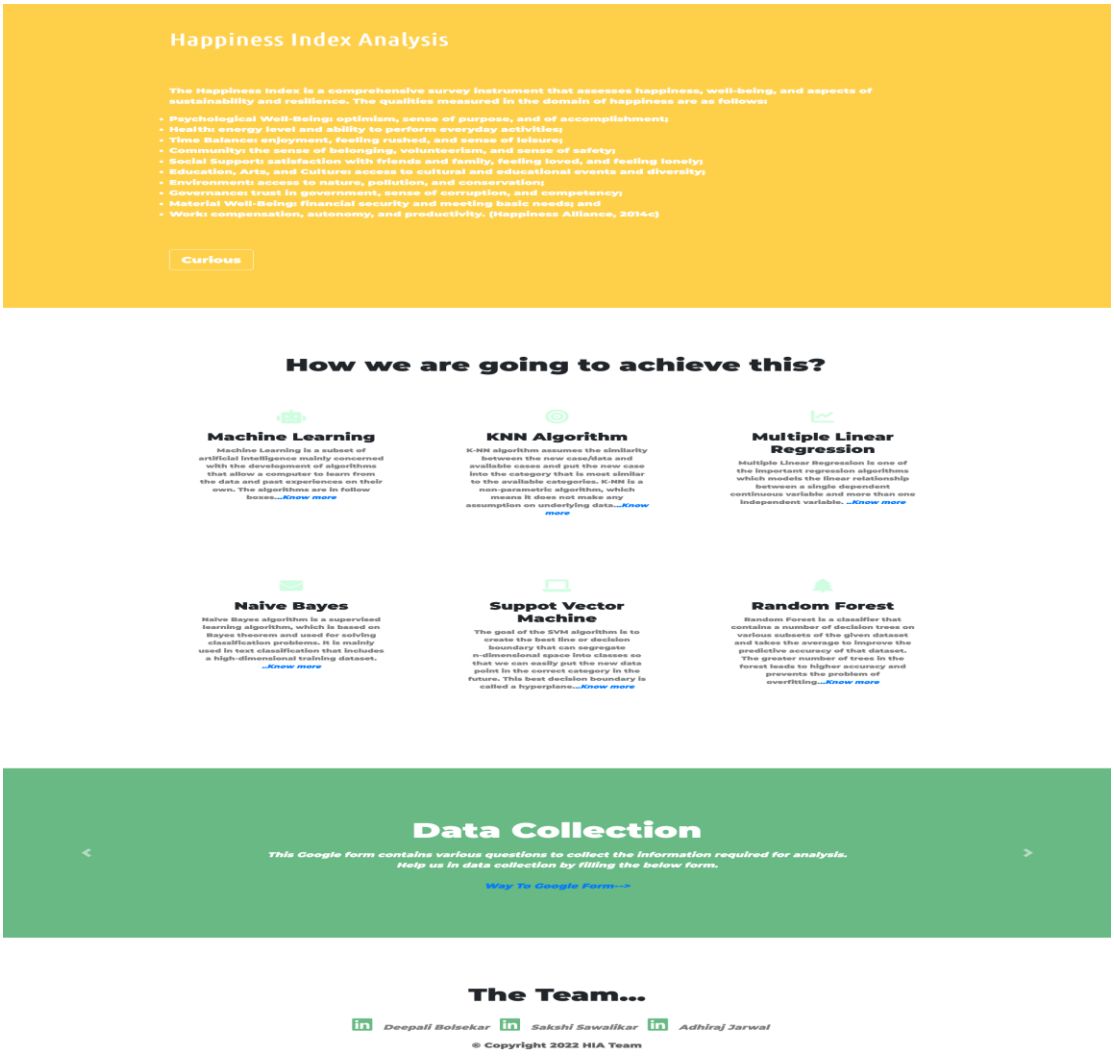


Figure 6.1.1 (Website view)

Future Scope

The Happiness Index Analysis project can help educational institutes to perform the survey in their institutes to find out the factors which affect students' educational interests and the related factors which affect their academic performance. The analysis can also help to implement various changes in their academic curriculum to make students more interested in learning process.

Reference

- Happiness Index- The footsteps towards sustainable development. Mevawala Jency, Gujarat, India, Dec 2019.
- Applying machine learning to predict happiness: a case study of 20 countries. Yu Tan, Charuk Singhapreecha, Woraphon Yamaka, Chaing Mai University, Thailand, 2022.
- Analysing happiness index as a measure along with its parameters and strategies doe improving india's rank in world happiness report. Sarah Ahtesham, India, Feb 2020.
- Happiness index methodology. Laura Musikanski, Scott Cloutier, Erica Bejarano, Davi Briggs, Julia Colbert, Steven Russell, 2017.
- Sentiment analysis using product review data. Xing Fang and Justin Zhan, 2015.
- https://www.geeksforgeeks.org/
- https://www.javatpoint.com/

Acknowledgment

We would like to thank Mr. D.K.Budhwant sir for giving us this opportunity to present our ideas and enhance our knowledge. We would also like to thank our guide Mr.Kiran Sonkamble sir for guiding us in this project to make it a successful one and helping us throughout the project.