INTERNSHIP REPORT ON MACHINE LEARNING WITH PYTHON FOR BUSINESS & DATA ANALYTICS

By :

KIRANKUMAR G

Registration No : 20105109018

Submitted To

INFORMATION TECHNOLOGY
DEPARTMENT
JBIET, HYDERABAD

Submitted By

KIRANKUMAR G
IT dept.
(20105109018)

ACKNOWLEDGEMENT

I would like to express my gratitude for the people who were part of my report, directly or indirectly people who gave unending support right from the stage the idea was conceived. It gives me a great pleasure to have an opportunity to acknowledge and to express gratitude those who were associated with me during my Internship at YBI Foundation.

I take this opportunity to thank industrial training coordinator, H.O.D of Computer science and Engineering department. I am highly indebted to my project guide Dr. Alok Yadav (Training Instructor) for his guidance and words of wisdom. He always showed me the right direction during the course of his report project work. I am duly thankful to him for teaching and referring me to various blocks, providing work and for permitting me to have training of duration of4 weeks.

KIRANKUMAR G

(20105109018)

DECLARATION

I hereby declare that the projects done by me at YBI foundation based on Machine Learning with Python for Business & Data Analytics, submitted by me is a record of bona-fide project work completed during internship training. I further declare that the work reported in this project has not been submitted anywhere else and is not copied from anywhere.

KIRANKUMAR G

(20105109018)

Chapter 1

Introduction and Literature Survey

Introduction

Machine Learning is the science of getting computers to learn without being explicitly programmed. It is closely related to computational statistics, which focuses on making prediction using computer. In its application across business problems, machine learning is also referred as predictive analysis. Machine Learning is closely related to computational statistics. Machine Learning focuses on the development of computer programs that can access data and use it to learn themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

History of Machine Learning

The name machine learning was coined in 1959 by Arthur Samuel. Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "**A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E***." This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?". In Turing's proposal the characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.

Types of Machine Learning

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve. Broadly Machine Learning can be categorized into four categories.

I. Supervised Learning
II. Unsupervised Learning
III. Reinforcement Learning
IV. Semi-supervised Learning

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly.

Supervised Learning

Supervised Learning is a type of learning in which we are given a data set and we already know what are correct output should look like, having the idea that there is a relationship between the input and output. Basically, it is learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples.
Supervised learning problems are categorized

Unsupervised Learning

Unsupervised Learning is a type of learning that allows us to approach problems with little or no idea what our problem should look like. We can derive the structure by clustering the data based on a relationship among the variables in data. With unsupervised learning there is no feedback based on prediction result. Basically, it is a type of self-organized learning that helps in finding previously unknown patterns in data set without pre-existing label.

Reinforcement Learning

Reinforcement learning is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best.

Semi-Supervised Learning

Semi-supervised learning fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Literature Survey

Theory

A core objective of a learner is to generalize from its experience. The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias–variance decomposition is one way to quantify generalizationerror.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has underfit the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer.

In addition to performance bounds, learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

The Challenges Facing Machine Learning

While there has been much progress in machine learning, there are also challenges. For example, the mainstream machine learning technologies are black-box approaches, making us concerned about their potential risks. To tackle this challenge, we may want to make machine learning more explainable and controllable. As another example, the computational complexity of machine learning algorithms is usually very high and we may want to invent lightweight algorithms or implementations. Furthermore, in many domains such as physics, chemistry, biology, and social sciences, people usually seek elegantly simple equations (e.g., the Schrödinger equation) to uncover the underlying laws behind various phenomena.Machine learning takes much more time. You have to gather and prepare data, then train the algorithm.There are much more uncertainties. That is why, while in traditional website or application development an experienced team can estimate the time quite precisely, a machine learning project used for example to provide product recommendations can take much less or much more time than expected. Why? Because even the best machine learning engineers don’t know how the deep learning networks will behave when analyzing different sets of data. It also means that the machine learning engineers and data scientists cannot guarantee that the training process of a model can be replicated.

Applications of Machine Learning

Machine learning is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that which makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. We probably use a learning algorithm dozen of time without even knowing it. Applications of Machine Learning include:

- **Web Search Engine:** One of the reasons why search engines like google, bing etc work so well is because the system has learnt how to rank pages through a complex learning algorithm.
- **Photo tagging Applications:** Be it facebook or any other photo tagging application, the ability to tag friends makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.
- **Spam Detector:** Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.

Future Scope

Future of Machine Learning is as vast as the limits of human mind. We can always keep learning, and teaching the computers how to learn. And at the same time, wondering how some of the most complex machine learning algorithms have been running in the back of our own mind so effortlessly all the time. There is a bright future for machine learning. Companies like Google, Quora, and Facebook hire people withmachine learning. There is intense research in machine learning at the top universities in the world. The global machine learning as a service market is rising expeditiously mainly due to the Internet revolution. Theprocess of connecting the world virtually has generated vast amount of data which is boosting the adoptionof machine learning solutions. Considering all these applications and dramatic improvements that ML has brought us, it doesn't take a genius to realize that in coming future we will definitely see more advanced applications of ML, applications that will stretch the capabilities of machine learning to an unimaginable level.

Organization of Training Workshop

Company Profile

(YBI Foundation is a Section 8 Not for Profit Organization) YBIF endeavour is to collaborate with institutions and enable learners to excel in emerging technologies for new age jobs. YBIF vision is to become the most trusted brand by incorporating the latest technology platform, industry skills and best-in-class student support system.

Objectives

Main objectives of training were to learn:

- How to determine and measure program complexity,
- Python Programming
- ML Library Scikit, Numpy , Matplotlib, Pandas.
- Statistical Math for the Algorithms.
- Learning to solve statistics and mathematical concepts.
- Supervised and Unsupervised Learning
- Classification and Regression
- ML Algorithms
- Machine Learning Programming and Use Cases.

Methodologies

There were several facilitation techniques used by the trainer which included question and answer, brainstorming, group discussions, case study discussions and practical implementation of some of the topics by trainees on flip charts and paper sheets. The multitude of training methodologies was utilized in order to make sure all the participants get the whole concepts and they practice what they learn, because only listening to the trainers can be forgotten, but what the trainees do by themselves they will never forget. After the post-tests were administered and the final course evaluation forms were filled in by the participants, the trainer expressed his closing remarks and reiterated the importance of the training for the trainees in their daily activities and their readiness for applying the learnt concepts in their assigned tasks. Certificates of completion were distributed among the participants at the end.

Chapter 2

Technology Implemented

Python – The New Generation Language

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for an emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Features

- Interpreted

In Python there is no separate compilation and execution steps like C/C++. It directly run the program from the source code. Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.

- Platform Independent

Python programs can be developed and executed on the multiple operating system platform. Python can be used on Linux, Windows, Macintosh, Solaris and many more.

- Multi- Paradigm

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming .

- Simple

Python is a very simple language. It is a very easy to learn as it is closer to English language. In python more emphasis is on the solution to the problem rather than the syntax.

- Rich Library Support

Python standard library is very vast. It can help to do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.

- Free and Open Source

Firstly, Python is freely available. Secondly, it is open-source. This means that its source code is available to the public. We can download it, change it, use it, and distribute it. This is called FLOSS (Free/Libre and Open Source Software). As the Python community, we're all headed toward one goal- anever-bettering Python.

Why Python Is a Perfect Language for Machine Learning?

1. A great library ecosystem -

   A great choice of libraries is one of the main reasons Python is the most popular programming language used for AI. A library is a module or a group of modules published by different sources which include a pre-written piece of code that allows users to reach some functionality or perform different actions. Python libraries provide base level items so developers don't have to code them from the very beginning every time. ML requires continuous data processing, and Python's libraries let us access, handle and transform data. These are some of the most widespread libraries you can usefor ML and AI:

   - Scikit-learn for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others.
   - Pandas for high-level data structures and analysis. It allows merging and filtering of data, as well as gathering it from other external sources like Excel, for instance.
   - Keras for deep learning. It allows fast calculations and prototyping, as it uses the GPU in addition to the CPU of the computer.
   - TensorFlow for working with deep learning by setting up, training, and utilizing artificial neural networks with massive datasets.
   - Matplotlib for creating 2D plots, histograms, charts, and other forms of visualization.
   - NLTK for working with computational linguistics, natural language recognition, and processing.
   - Scikit-image for image processing.
   - PyBrain for neural networks, unsupervised and reinforcement learning.
   - Caffe for deep learning that allows switching between the CPU and the GPU and processing 60+ mln images a day using a single NVIDIA K40 GPU.
   - StatsModels for statistical algorithms and data exploration.

   In the PyPI repository, we can discover and compare more python libraries.

2. A low entry barrier -

   Working in the ML and AI industry means dealing with a bunch of data that we need to process in the most convenient and effective way. The low entry barrier allows more data scientists to quickly pick up Python and start using it for AI development without wasting too much effort into learning the language. In addition to this, there's a lot of documentation available, and Python's community is always there to help out and give advice.

3. Flexibility-

Python for machine learning is a great choice, as this language is very flexible:

- It offers an option to choose either to use OOPs or scripting.
- There's also no need to recompile the source code, developers can implement any changes and quickly see the results.
- Programmers can combine Python and other languages to reach their goals.

4. Good Visualization Options-

For AI developers, it's important to highlight that in artificial intelligence, deep learning, and machine learning, it's vital to be able to represent data in a human-readable format. Libraries like Matplotlib allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.

5. Community Support-

It's always very helpful when there's strong community support built around the programming language. Python is an open-source language which means that there's a bunch of resources open for programmers starting from beginners and ending with pros. A lot of Python documentation isavailable online as well as in Python communities and forums, where programmers and machine learning developers discuss errors, solve problems, and help each other out. Python programming language is absolutely free as is the variety of useful libraries and tools.

6. Growing Popularity-

As a result of the advantages discussed above, Python is becoming more and more popular among data scientists. According to StackOverflow, the popularity of Python is predicted to grow until 2020, at least. This means it's easier to search for developers and replace team players if required. Also, thecost of their work maybe not as high as when using a less popular programming language.

Data Preprocessing, Analysis & Visualization

Machine Learning algorithms don't work so well with processing raw data. Before we can feed such data to an ML algorithm, we must preprocess it. We must apply some transformations on it. With data preprocessing, we convert raw data into a clean data set. To perform data this, there are 7 techniques -

1. Rescaling Data -

For data with attributes of varying scales, we can rescale attributes to possess the same scale. We rescale attributes into the range 0 to 1 and call it normalization. We use the MinMaxScaler class from scikit- learn. This gives us values between 0 and 1.

2. Standardizing Data -

With standardizing, we can take attributes with a Gaussian distribution and different means and standard deviations and transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

3. Normalizing Data -

In this task, we rescale each observation to a length of 1 (a unit norm). For this, we use the Normalizer class.

4. Binarizing Data -

Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0. For this purpose, we use the Binarizer class.

5. Mean Removal-

We can remove the mean from each feature to center it on zero.

6. One Hot Encoding -

When dealing with few and scattered numerical values, we may not need to store these. Then, we can perform One Hot Encoding. For k distinct values, we can transform the feature into a k-dimensional vector with one value of 1 and 0 as the rest values.

7. Label Encoding -

Some labels can be words or numbers. Usually, training data is labelled with words to make it readable. Label encoding converts word labels into numbers to let algorithms work on them.

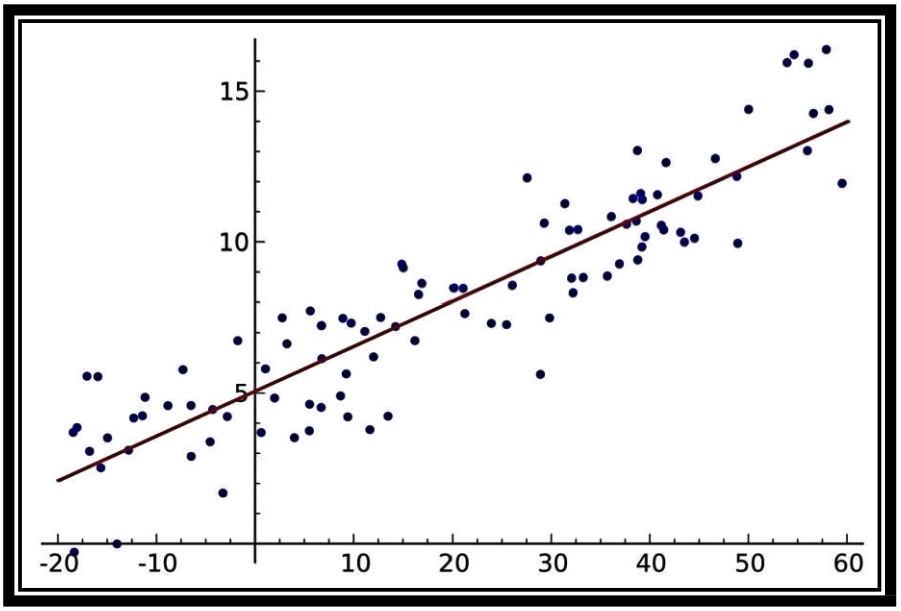
Machine Learning Algorithms

There are many types of Machine Learning Algorithms specific to different use cases. As we work with datasets, a machine learning algorithm works in two stages. We usually split the data around 20%-80% between testing and training stages. Under supervised learning, we split a dataset into a training data and test data in Python ML. Followings are the Algorithms of Python Machine Learning -

1. Linear Regression-

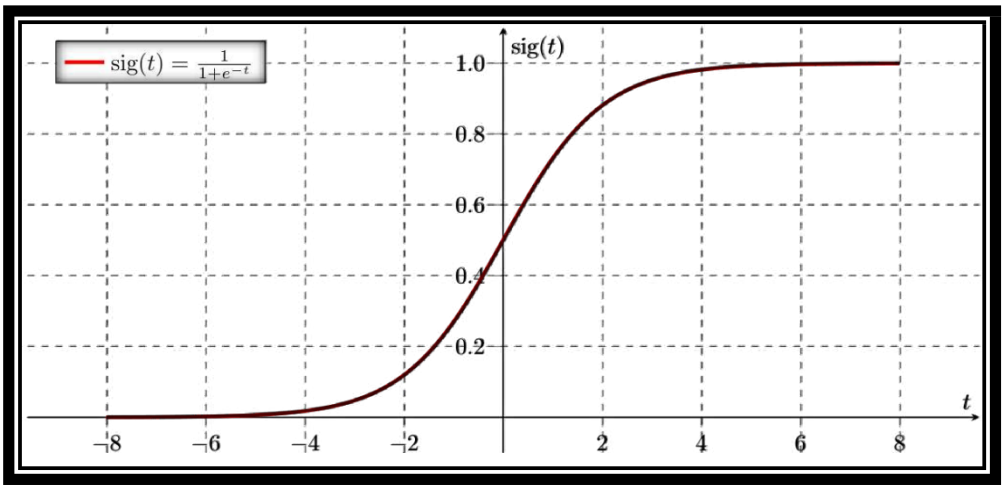
Linear regression is one of the supervised Machine learning algorithms in Python that observes continuous features and predicts an outcome. Depending on whether it runs on a single variable or on many features, we can call it simple linear regression or multiple linear regression.

This is one of the most popular Python ML algorithms and often under-appreciated. It assigns optimal weights to variables to create a line ax+b to predict the output. We often use linear regression to estimate real values like a number of calls and costs of houses based on continuous variables. The regression line is the best line that fits Y=a*X+b to denote a relationship between independent and dependent variables.



2. Logistic Regression -

Logistic regression is a supervised classification is unique Machine Learning algorithms in Python that finds its use in estimating discrete values like 0/1, yes/no, and true/false. This is based on a given set of independent variables. We use a logistic function to predict the probability of an event and this gives us an output between 0 and 1. Although it says 'regression', this is actually a classification algorithm. Logistic regression fits data into a logit function and is also called logit regression.



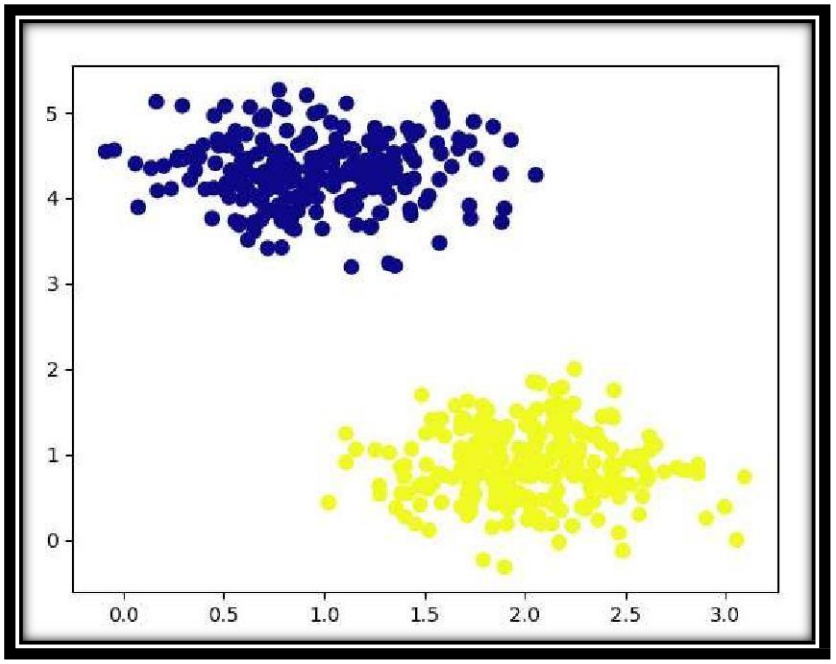
3. Decision Tree -

A decision tree falls under supervised Machine Learning Algorithms in Python and comes of use for both classification and regression- although mostly for classification. This model takes an instance, traverses the tree, and compares important features with a determined conditional statement. Whether it descends to the left child branch or the right depends on the result. Usually, more important features are closer to the root.

Decision Tree, a Machine Learning algorithm in Python can work on both categorical and continuous dependent variables. Here, we split a population into two or more homogeneous sets. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

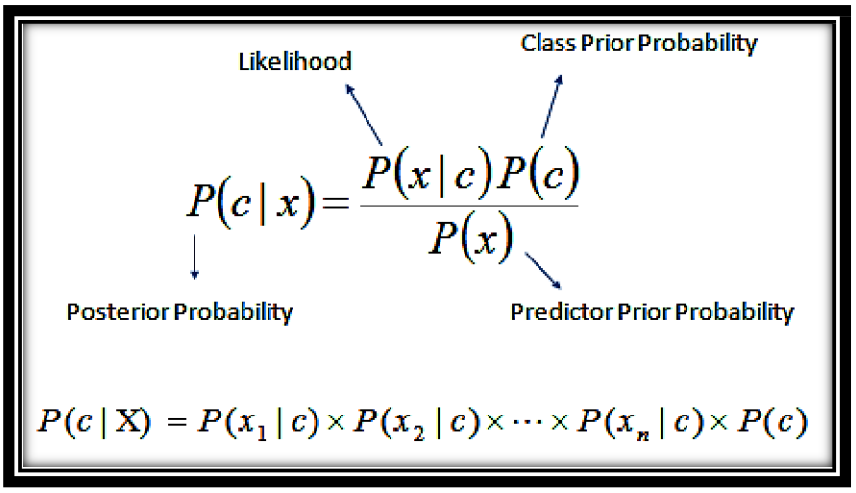
4. Support Vector Machine (SVM)-

SVM is a supervised classification is one of the most important Machines Learning algorithms in Python, that plots a line that divides different categories of your data. In this ML algorithm, we calculate the vector to optimize the line. This is to ensure that the closest point in each group lies farthest from each other. While you will almost always find this to be a linear vector, it can be other than that. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.



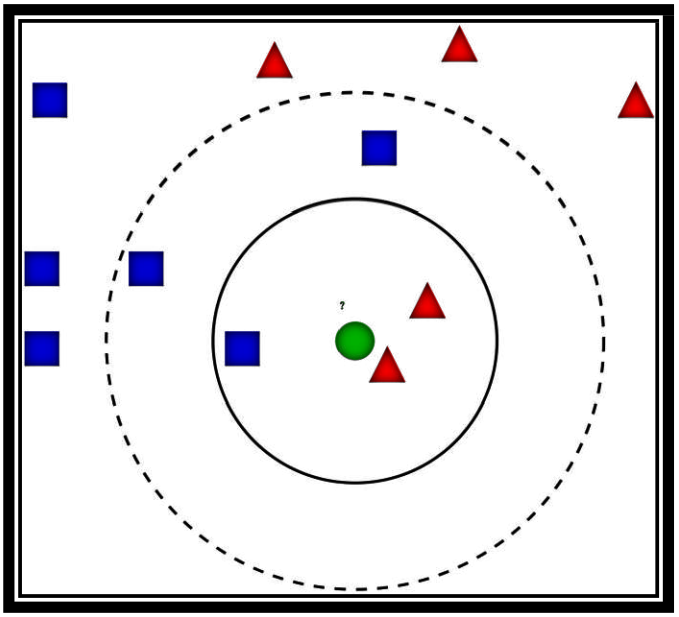
5. Naïve Bayes Algorithm -

Naive Bayes is a classification method which is based on Bayes' theorem. This assumes independence between predictors. A Naive Bayes classifier will assume that a feature in a class is unrelated to any other. Consider a fruit. This is an apple if it is round, red, and 2.5 inches in diameter. A Naive Bayes classifier will say these characteristics independently contribute to the probability of the fruit being an apple. This is evenif features depend on each other. For very large data sets, it is easy to build a Naive Bayesian model. Not only is this model very simple, it performs better than many highly sophisticated classification methods. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.



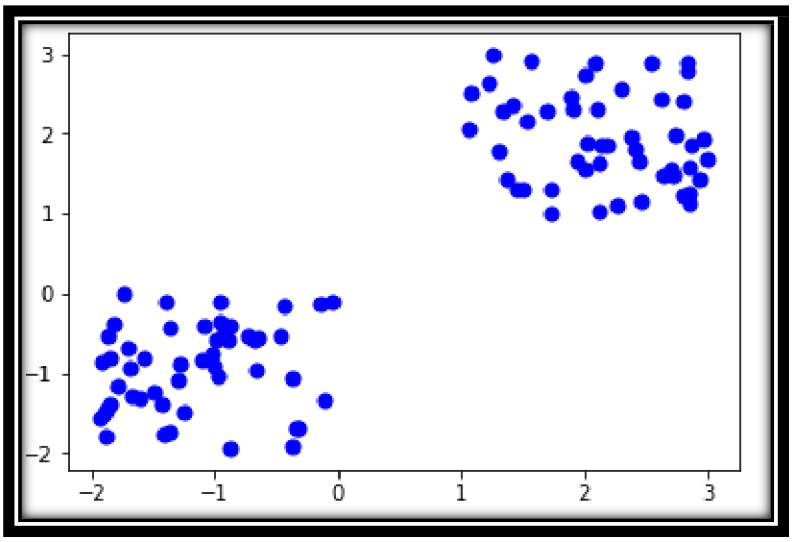
6. kNN Algorithm -

This is a Python Machine Learning algorithm for classification and regression- mostly for classification. This is a supervised learning algorithm that considers different centroids and uses a usually Euclidean function to compare distance. Then, it analyzes the results and classifies each point to the group to optimizeit to place with all closest points to it. It classifies new cases using a majority vote of k of its neighbors. The case it assigns to a class is the one most common among its K nearest neighbors. For this, it uses a distance function. *k*-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. *k*-NN is a special case of a variable-bandwidth, kernel density "balloon" estimator with a uniform kernel.



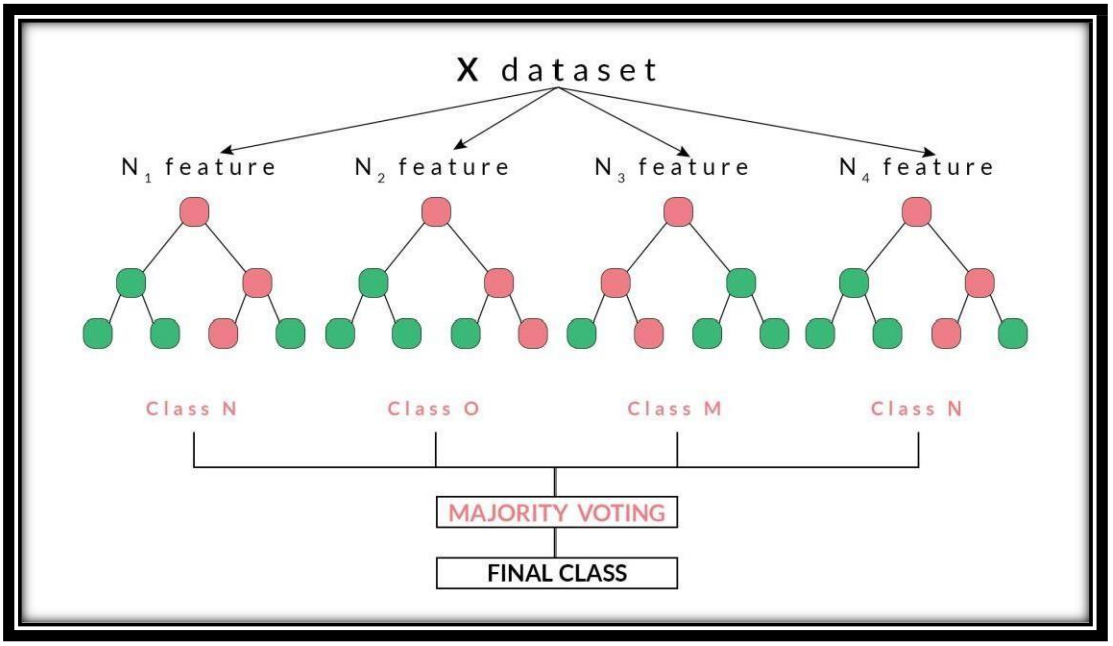
7. K-Means Algorithm -

k-Means is an unsupervised algorithm that solves the problem of clustering. It classifies data using a number of clusters. The data points inside a class are homogeneous and heterogeneous to peer groups. *k*-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. *k*-means clustering is rather easy to apply to even large data sets, particularly when using heuristics such as Lloyd's algorithm. It often is used as a preprocessing step for other algorithms, for example to find a starting configuration. The problem is computationally difficult (NP-hard). *k*-means originates from signal processing, and still finds use in this domain. In cluster analysis, the *k*-means algorithm can be used to partition the input data set into *k* partitions (clusters). *k*-means clustering has been used as a feature learning (or dictionary learning) step, in either (semi-)supervised learning or unsupervised learning.



8. Random Forest -

A random forest is an ensemble of decision trees. In order to classify every new object based on its attributes, trees vote for class- each tree provides a classification. The classification with the most votes wins in the forest. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



Chapter 3

Result Discussion

Result

This training has introduced us to Machine Learning. Now, we know that Machine Learning is a technique of training machines to perform the activities a human brain can do, albeit bit faster and better than an average human-being. Today we have seen that the machines can beat human champions in games such as Chess, Mahjong, which are considered very complex. We have seen that machines can be trained to perform human activities in several areas and can aid humans in living better lives. Machine learning is quickly growing field in computer science. It has applications in nearly every other field of study and is already being implemented commercially because machine learning can solve problems too difficult or time consuming for humans to solve. To describe machine learning in general terms, a variety models are used to learn patterns in data and make accurate predictions based on the patterns it observes.

Machine Learning can be a Supervised or Unsupervised. If we have a lesser amount of data and clearly labelled data for training, we opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If we have a huge data set easily available, we go for deep learning techniques. We also have learned Reinforcement Learning and Deep Reinforcement Learning. We now know what Neural Networks are, their applications and limitations. Specifically, we have developed a thought process for approaching problems that machine learning works so well at solving. We have learnt how machine learning is different than descriptive statistics.

Finally, when it comes to the development of machine learning models of our own, we looked at the choices of various development languages, IDEs and Platforms. Next thing that we need to do is start learning and practicing each machine learning technique. The subject is vast, it means that there is width, but if we consider the depth, each topic can be learned in a few hours. Each topic is independent of each other. We need to take into consideration one topic at a time, learn it, practice it and implement the algorithm/s in it using a language choice of yours. This is the best way to start studying Machine Learning. Practicing one topic at a time, very soon we can acquire the width that is eventually required of a Machine Learning expert.

Chapter 4

Project Report

Objective-

Classification Model to Identify Employee Attrition Project

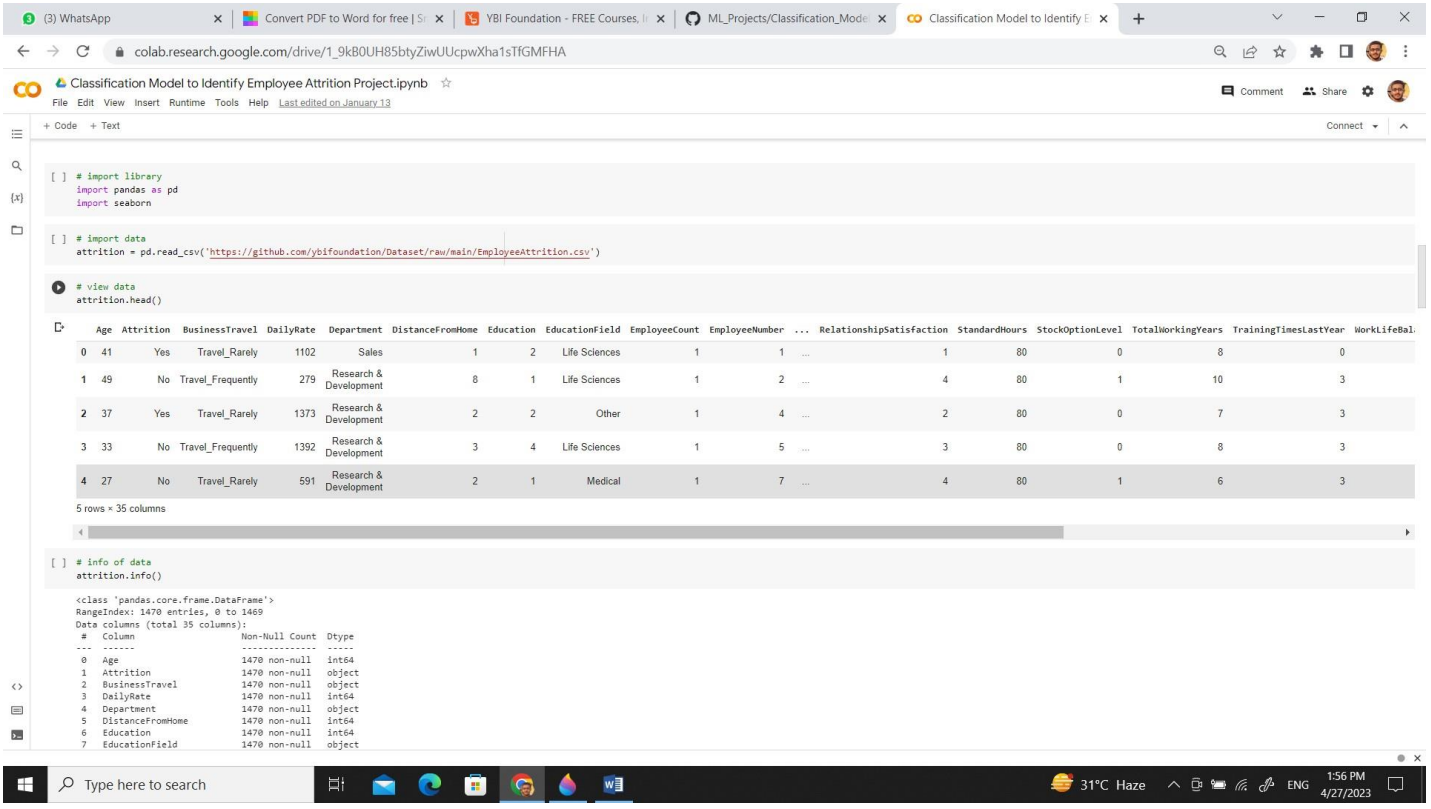
Dataset description-

Dataset source - https://github.com/ybifoundation/Dataset/raw/main/EmployeeAttrition.csv

Dataset variables

1. AGE : Numerical Value
2. ATTRITION : Employee leaving the company (0=no, 1=yes)
3. BUSINESS TRAVEL : (1=No Travel, 2=Travel Frequently, 3=Tavel Rarely)
4. DAILY RATE : Numerical Value - Salary Level
5. DEPARTMENT : (1=HR, 2=R&D, 3=Sales)
6. DISTANCE FROM HOME : Numerical Value - THE DISTANCE FROM WORK TO HOME
7. EDUCATION Numerical Value
8. EDUCATION FIELD (1=HR, 2=LIFE SCIENCES, 3=MARKETING, 4=MEDICAL SCIENCES, 5=OTHERS, 6= TEHCNICAL)
9. EMPLOYEE COUNT Numerical Value
10. EMPLOYEE NUMBER Numerical Value - EMPLOYEE ID
11. ENVIROMENT SATISFACTION Numerical Value - SATISFACTION WITH THE ENVIROMENT
12. GENDER (1=FEMALE, 2=MALE)
13. HOURLY RATE Numerical Value - HOURLY SALARY
14. JOB INVOLVEMENT Numerical Value - JOB INVOLVEMENT
15. JOB LEVEL Numerical Value - LEVEL OF JOB
16. JOB ROLE (1=HC REP, 2=HR, 3=LAB TECHNICIAN, 4=MANAGER, 5= MANAGING DIRECTOR, 6= REASEARCH DIRECTOR, 7= RESEARCH SCIENTIST, 8=SALES EXECUTIEVE, 9= SALES REPRESENTATIVE)
17. JOB SATISFACTION Numerical Value - SATISFACTION WITH THE JOB
18. MARITAL STATUS (1=DIVORCED, 2=MARRIED, 3=SINGLE)
19. MONTHLY INCOME Numerical Value - MONTHLY SALARY
20. MONTHY RATE Numerical Value - MONTHY RATE
21. NUMCOMPANIES WORKED Numerical Value - NO. OF COMPANIES WORKED AT
22. OVER 18 (1=YES, 2=NO)
23. OVERTIME (1=NO, 2=YES)
24. PERCENT SALARY HIKE Numerical Value - PERCENTAGE INCREASE IN SALARY
25. PERFORMANCE RATING Numerical Value - PERFORMANCE RATING
26. RELATIONS SATISFACTION Numerical Value - RELATIONS SATISFACTION
27. STANDARD HOURS Numerical Value - STANDARD HOURS
28. STOCK OPTIONS LEVEL Numerical Value - STOCK OPTIONS
29. TOTAL WORKING YEARS Numerical Value - TOTAL YEARS WORKED
30. TRAINING TIMES LAST YEAR Numerical Value - HOURS SPENT TRAINING
31. WORK LIFE BALANCE Numerical Value - TIME SPENT BEWTWEEN WORK AND OUTSIDE
32. YEARS AT COMPANY Numerical Value - TOTAL NUMBER OF YEARS AT THE COMPNAY
33. YEARS IN CURRENT ROLE Numerical Value -YEARS IN 34. CURRENT ROLE
34. YEARS SINCE LAST PROMOTION Numerical Value - LAST PROMOTION
35. YEARS WITH CURRENT MANAGER Numerical Value - YEARS SPENT WITH CURRENT MANAGER

Code and output-



```
# import library
import pandas as pd
import seaborn
```

```
# import data
attrition = pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/EmployeeAttrition.csv')
```

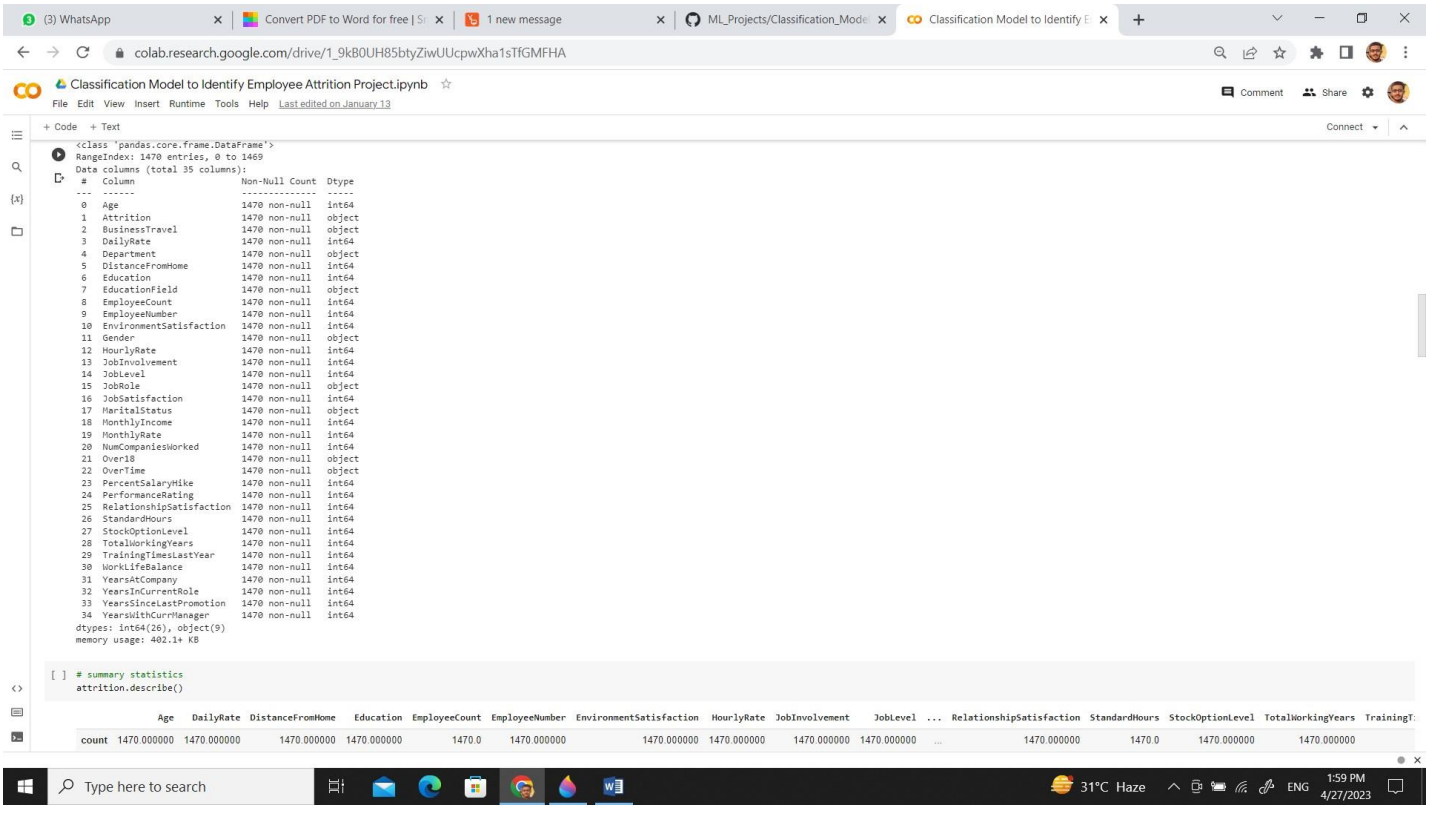
```
# view data
attrition.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBal
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	1	80	0	8	0	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	4	80	1	10	3	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	2	80	0	7	3	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	3	80	0	8	3	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	4	80	1	6	3	

5 rows × 35 columns

```
# info of data
attrition.info()
```

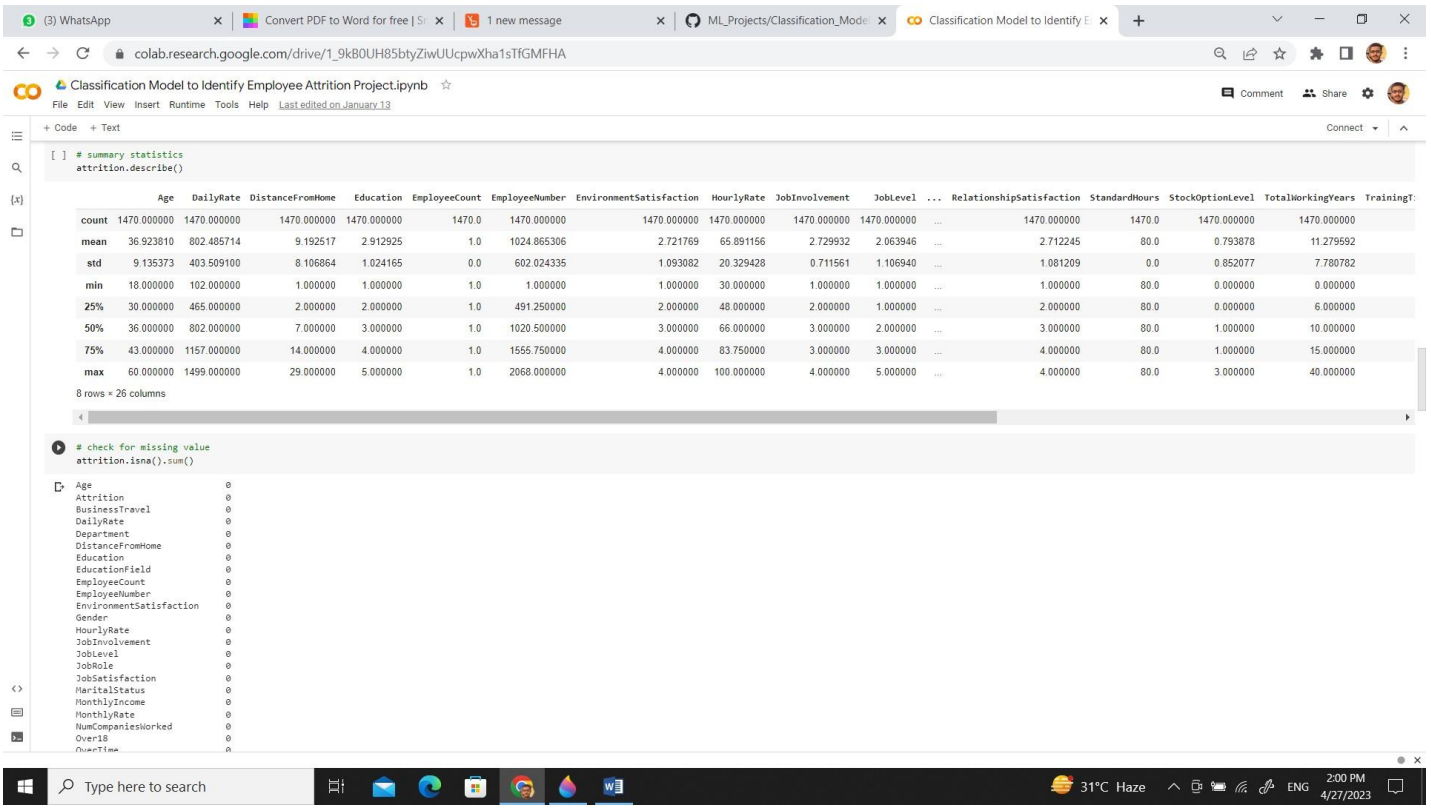
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
```

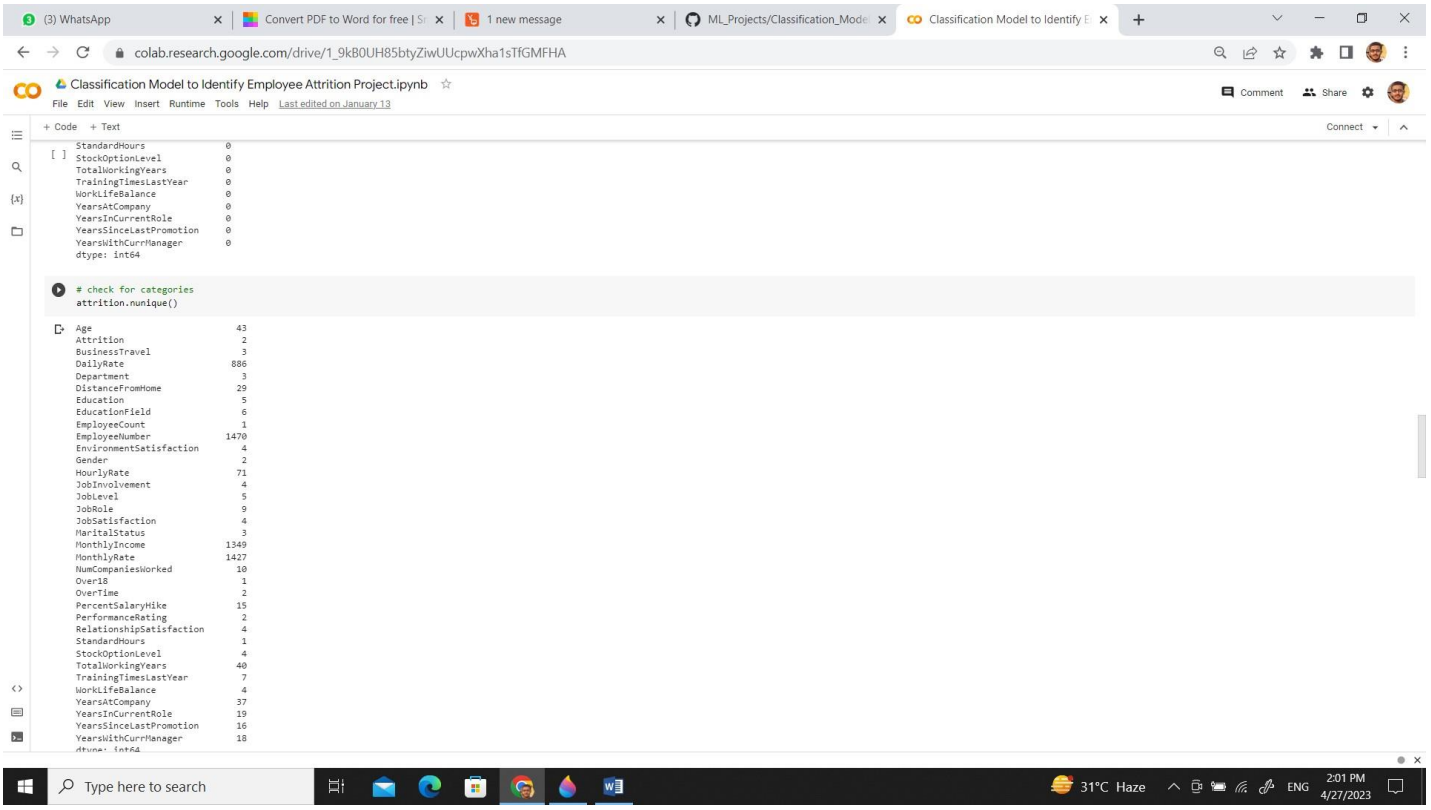


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

```
# summary statistics
attrition.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingT
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	...	1470.000000	1470.0	1470.000000	1470.000000	


(3) WhatsApp
Convert PDF to Word for free | Sr
1 new message
ML_Projects/Classification_Mode
Classification Model to Identify E
colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA
Classification Model to Identify Employee Attrition Project.ipynb
File Edit View Insert Runtime Tools Help Last edited on January 13
Comment
Share
Connect
+ Code + Text
[ ] # summary statistics
attrition.describe()
Age DailyRate DistanceFromHome Education EmployeeCount EmployeeNumber EnvironmentSatisfaction HourlyRate JobInvolvement JobLevel ... RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears
count 1470.000000 1470.000000 1470.000000 1470.000000 1470.0 1470.000000 1470.000000 1470.000000 1470.000000 1470.000000 ... 1470.000000 1470.0 1470.000000 1470.000000
mean 36.923810 802.485714 9.192517 2.912925 1.0 1024.865306 2.721769 65.891156 2.729932 2.063946 ... 2.712245 80.0 0.793878 11.279592
std 9.135373 403.509100 8.106864 1.024165 0.0 602.024335 1.093082 20.329428 0.711561 1.106940 ... 1.081209 0.0 0.852077 7.780782
min 18.000000 102.000000 1.000000 1.000000 1.0 1.000000 1.000000 30.000000 1.000000 1.000000 ... 1.000000 80.0 0.000000 0.000000
25% 30.000000 465.000000 2.000000 2.000000 1.0 491.250000 2.000000 48.000000 2.000000 1.000000 ... 2.000000 80.0 0.000000 6.000000
50% 36.000000 802.000000 7.000000 3.000000 1.0 1020.500000 3.000000 66.000000 3.000000 2.000000 ... 3.000000 80.0 1.000000 10.000000
75% 43.000000 1157.000000 14.000000 4.000000 1.0 1555.750000 4.000000 83.750000 3.000000 3.000000 ... 4.000000 80.0 1.000000 15.000000
max 60.000000 1499.000000 29.000000 5.000000 1.0 2068.000000 4.000000 100.000000 4.000000 5.000000 ... 4.000000 80.0 3.000000 40.000000
8 rows × 26 columns
# check for missing value
attrition.isna().sum()
Age 0
Attrition 0
BusinessTravel 0
DailyRate 0
Department 0
DistanceFromHome 0
Education 0
EducationField 0
EmployeeCount 0
EmployeeNumber 0
EnvironmentSatisfaction 0
Gender 0
HourlyRate 0
JobInvolvement 0
JobLevel 0
JobRole 0
JobSatisfaction 0
MaritalStatus 0
MonthlyIncome 0
MonthlyRate 0
NumCompaniesWorked 0
Over18 0
OverTime 0
Type here to search
31°C Haze
ENG
2:00 PM
4/27/2023


(3) WhatsApp
Convert PDF to Word for free | Sr
1 new message
ML_Projects/Classification_Mode
Classification Model to Identify E
colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA
Classification Model to Identify Employee Attrition Project.ipynb
File Edit View Insert Runtime Tools Help Last edited on January 13
Comment
Share
Connect
+ Code + Text
StandardHours 0
StockOptionLevel 0
TotalWorkingYears 0
TrainingTimesLastYear 0
WorkLifeBalance 0
YearsAtCompany 0
YearsInCurrentRole 0
YearsSinceLastPromotion 0
YearsWithCurrManager 0
dtype: int64
# check for categories
attrition.nunique()
Age 43
Attrition 2
BusinessTravel 3
DailyRate 886
Department 3
DistanceFromHome 29
Education 5
EducationField 6
EmployeeCount 1
EmployeeNumber 1470
EnvironmentSatisfaction 4
Gender 2
HourlyRate 71
JobInvolvement 4
JobLevel 5
JobRole 9
JobSatisfaction 4
MaritalStatus 3
MonthlyIncome 1349
MonthlyRate 1427
NumCompaniesWorked 10
Over18 1
OverTime 2
PercentSalaryHike 15
PerformanceRating 2
RelationshipSatisfaction 4
StandardHours 1
StockOptionLevel 4
TotalWorkingYears 40
TrainingTimesLastYear 7
WorkLifeBalance 4
YearsAtCompany 37
YearsInCurrentRole 19
YearsSinceLastPromotion 16
YearsWithCurrManager 18
Type here to search
31°C Haze
ENG
2:01 PM
4/27/2023

(3) WhatsApp | Convert PDF to Word for free | Sr | 1 new message | ML_Projects/Classification_Mode | Classification Model to Identify E

colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA

Classification Model to Identify Employee Attrition Project.ipynb

File Edit View Insert Runtime Tools Help Last edited on January 13

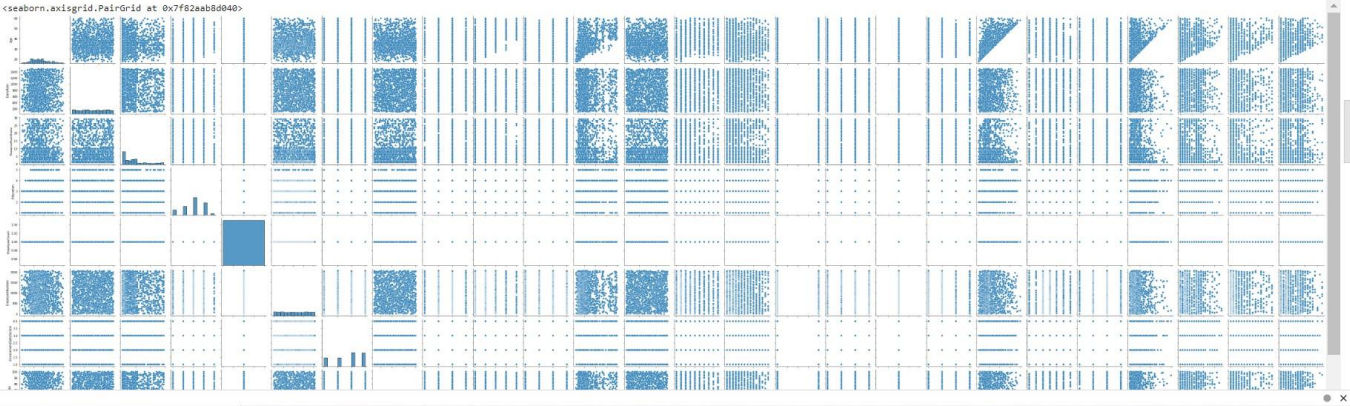
Comment Share Connect

+ Code + Text

```
PercentSalaryHike           15
PerformanceRating            2
RelationshipSatisfaction     4
StandardHours                1
StockOptionLevel             4
TotalWorkingYears           40
TrainingTimesLastYear        7
WorkLifeBalance              4
YearsAtCompany              37
YearsInCurrentRole          19
YearsSinceLastPromotion     16
YearsWithCurrManager        18
dtype: int64
```

```
# visualize pairplot
seaborn.pairplot(attrition)
```

```
<seaborn.axisgrid.PairGrid at 0x7f82aab8d040>
```



Type here to search

31°C Haze ENG 2:01 PM 4/27/2023

(3) WhatsApp | Convert PDF to Word for free | Sr | YBI Foundation - FREE Courses, I | ML_Projects/Classification_Mode | Classification Model to Identify E

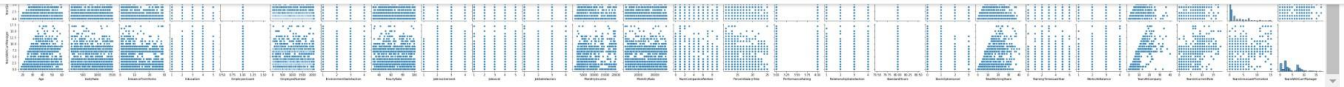
colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA

Classification Model to Identify Employee Attrition Project.ipynb

File Edit View Insert Runtime Tools Help Last edited on January 13

Comment Share Connect

+ Code + Text



```
# columns name
attrition.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

```
# column names of numerical columns
import numpy as np
attrition.select_dtypes(include=np.number).columns
```

```
Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome',
       'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike',
       'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

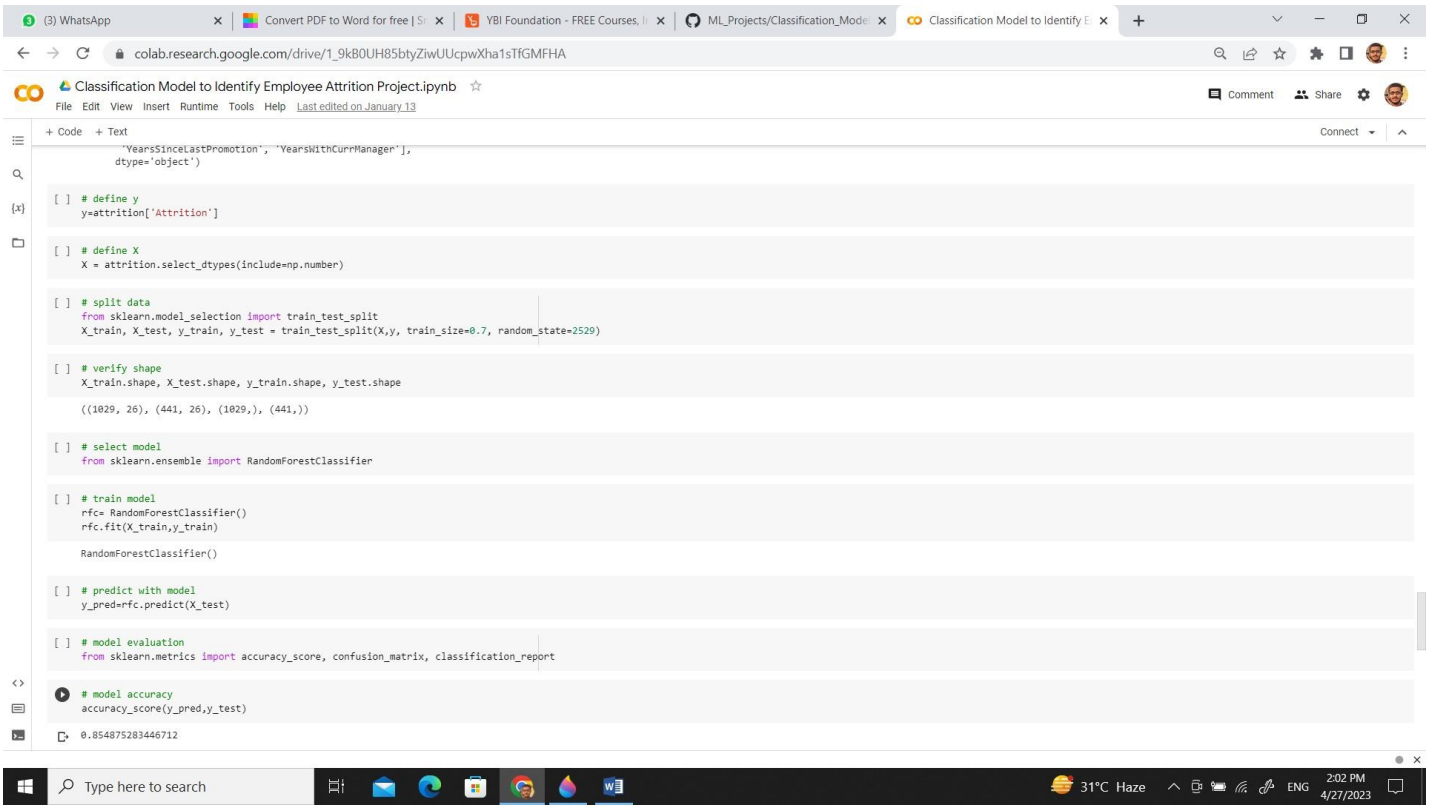
```
# define y
y=attrition['Attrition']
```

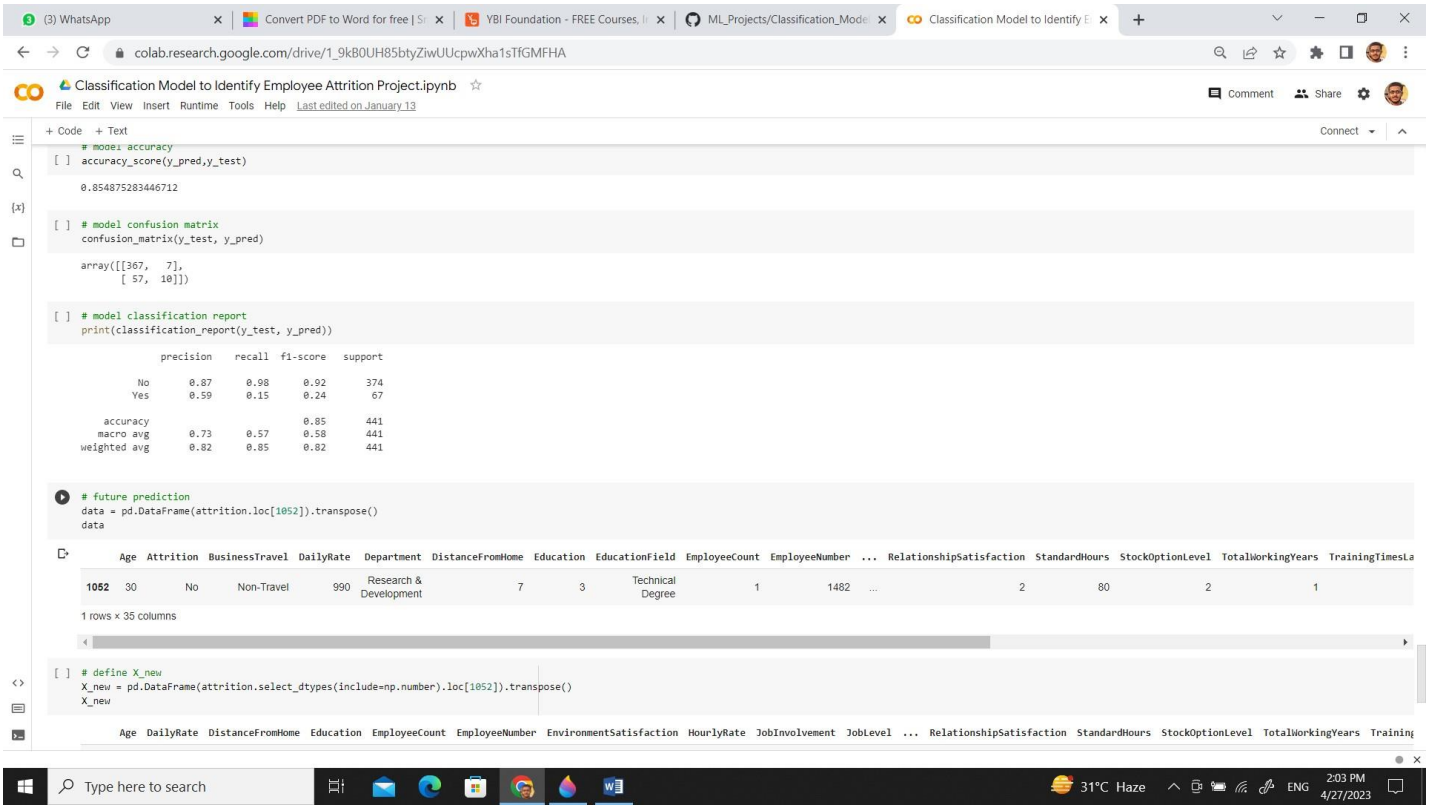
```
# define X
X = attrition.select_dtypes(include=np.number)
```

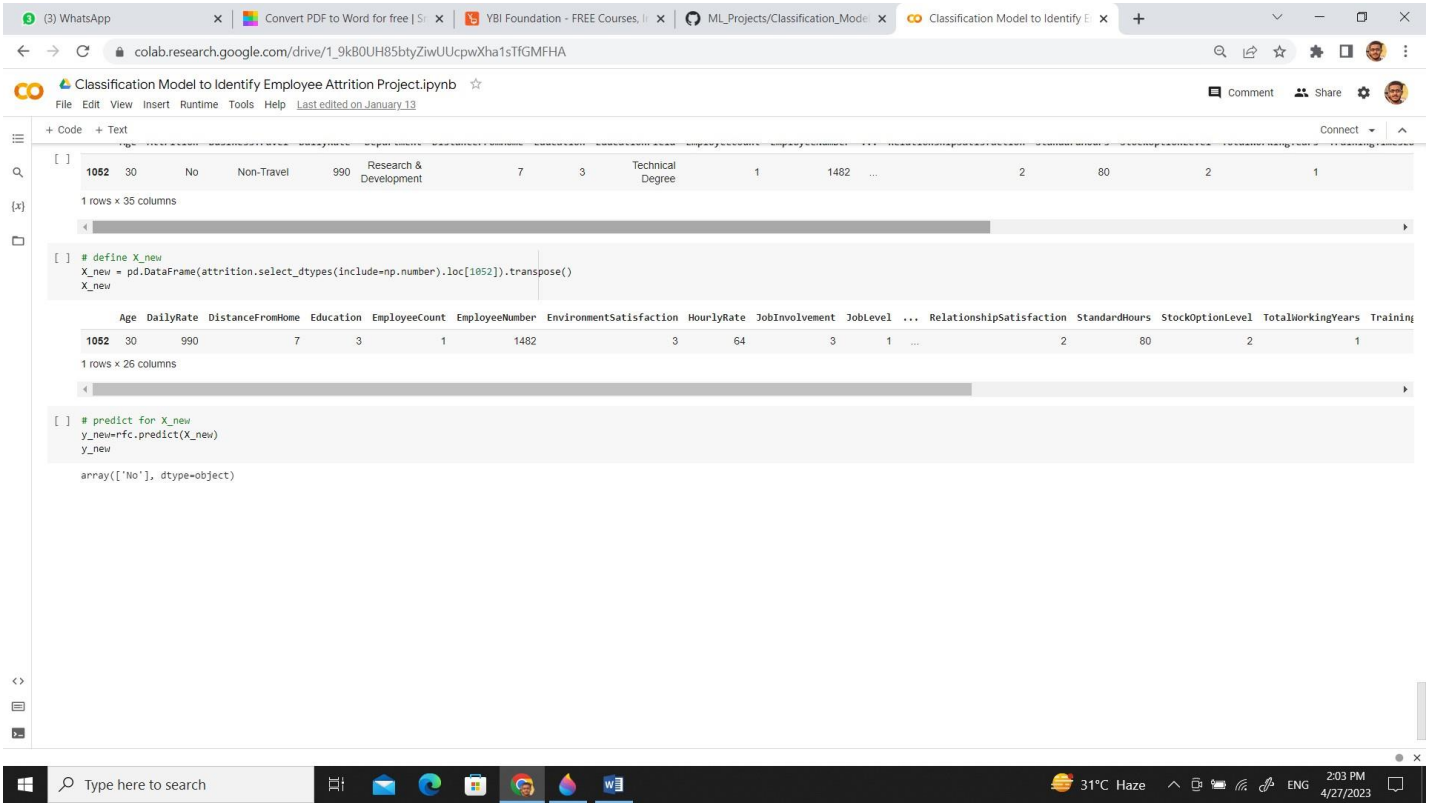
```
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.7, random_state=2529)
```

Type here to search

31°C Haze ENG 2:01 PM 4/27/2023


(3) WhatsApp
Convert PDF to Word for free | S
YBI Foundation - FREE Courses, I
ML_Projects/Classification_Mode
Classification Model to Identify E
colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA
Classification Model to Identify Employee Attrition Project.ipynb
File Edit View Insert Runtime Tools Help Last edited on January 13
Comment
Share
Connect
+ Code + Text
'YearsSinceLastPromotion', 'YearsWithCurrManager'],
dtype='object')
# define y
y=attrition['Attrition']
# define X
X = attrition.select_dtypes(include=np.number)
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.7, random_state=2529)
verify shape
X_train.shape, X_test.shape, y_train.shape, y_test.shape
((1029, 26), (441, 26), (1029,), (441,))
# select model
from sklearn.ensemble import RandomForestClassifier
# train model
rfc= RandomForestClassifier()
rfc.fit(X_train,y_train)
RandomForestClassifier()
# predict with model
y_pred=rfc.predict(X_test)
# model evaluation
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
# model accuracy
accuracy_score(y_pred,y_test)
0.854875283446712
Type here to search
31°C Haze
ENG
2:02 PM
4/27/2023


(3) WhatsApp
Convert PDF to Word for free | S
YBI Foundation - FREE Courses, I
ML_Projects/Classification_Mode
Classification Model to Identify E
colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA
Classification Model to Identify Employee Attrition Project.ipynb
File Edit View Insert Runtime Tools Help Last edited on January 13
Comment
Share
Connect
+ Code + Text
# model accuracy
accuracy_score(y_pred,y_test)
0.854875283446712
# model confusion matrix
confusion_matrix(y_test, y_pred)
array([[367, 7],
[57, 10]])
# model classification report
print(classification_report(y_test, y_pred))
precision recall f1-score support
No 0.87 0.98 0.92 374
Yes 0.59 0.15 0.24 67
accuracy 0.85 441
macro avg 0.73 0.57 0.58 441
weighted avg 0.82 0.85 0.82 441
# future prediction
data = pd.DataFrame(attrition.loc[1052]).transpose()
data
Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EmployeeCount EmployeeNumber ... RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears TrainingTimesLa
1052 30 No Non-Travel 990 Research & Development 7 3 Technical Degree 1 1482 ... 2 80 2 1
1 rows × 35 columns
# define X_new
X_new = pd.DataFrame(attrition.select_dtypes(include=np.number).loc[1052]).transpose()
X_new
Age DailyRate DistanceFromHome Education EmployeeCount EmployeeNumber EnvironmentSatisfaction HourlyRate JobInvolvement JobLevel ... RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears Training
Type here to search
31°C Haze
ENG
2:03 PM
4/27/2023



Link to this project-

https://github.com/himanshumehra250/ML-project/blob/main/ml%20project.ipynb

https://colab.research.google.com/drive/1_9kB0UH85btyZiwUUcpwXha1sTfGMFHA?usp=sharing


Thank You