

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and
Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the
award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

ARAVA JAYARAM
REG. NO. 39120045



DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A.JAYARAM(Reg. No: 39120045)** who carried out the project entitled **“Classification of Teaching assistant Evolution based on machine learning”** under my supervision from June 2021 to November2021.

Internal Guide

Dr.V.Nirmalrani

Head of the Department

Submitted for Viva voce Examination held on _____

InternalExaminer

ExternalExaminer

DECLARATION

I , **A.Jayaram** hereby declare that the project report entitled “Classification of **Teaching Assistant Evolution based on machine learning**” done by me under the guidance of **Dr.V.Nirmalrani** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:
THECANDIDATE

SIGNATURE OF

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.V.Nirmalrani** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



Abstract

- The teaching assistant evaluation reviews *academic qualifications , relevant experience , quality of teaching , and professional contribution*. All these aspects can be assessed by the students , peers or by the teachers themselves.
- This paper includes the teacher's evaluation Aleamoni suggests that students are the main source of information about the learning environment, including teacher's ability to motivate them for continued learning , rapport or degree of communication between instructors and students.
 - They are also the most consistent evaluators of the quality, the effectiveness of the learning process and satisfaction with course content, method of instruction, textbooks, homework, and student interest(Aleamoni, 1981).
- This data is collected by a The Wei-Yin Loh (Department of Statistics,UW-Madison).
 - The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison.
- The scores were divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable.
- The given dataset contains 151 rows and 6 columns. The first 5 columns are features, the last column contains the classification label of '3's , '2's and '1's.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	(i)
	LIST OF FIGURES	(ii)
1	INTRODUCTION	1
	1.1 ESTIMATING TEACHING ASSISTANT EVALUATION	1
	1.2 ESTIMATION DATASET INFORMATION	1
	1.3 COMMON MACHINE LEARNING ALGORITHMS AND GOALS	2
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	9
	2.1 AIM	9
	2.2 SCOPE	9
	2.3 ATTRIBUTE INFORMATION	9
	2.4 DISTRIBUTION OF ATTRIBUTE NO 6	10
	2.5 DATA PREPARATION	10

	2.6 READING DATASET	10
3	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	11
	3.1 TYPES OF CLASSIFICATION ALGORITHMS USED	11
	3.2 LOGISTIC REGRESSION ALGORITHM	12
	3.3 IMPORTED LIBRARIES	12
	3.4 IMPLEMENTATION OF LOGISTIC REGRESSION	17
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	19
	4.1 TRAINING AND ACCURACY (MODEL ANALYSIS)	19
5	SUMMARY AND CONCLUSIONS	22
	5.1 SUMMARY AND CONCLUSIONS	22
	REFERENCES	22
	APPENDIX	24
	A.SCREENSHOTS AND OUTPUTS	24
	SOURCE CODE	33

LIST OF FIGURES

Fig no	Fig title	Page no
1.2	excel sheet of the given TA dataset	2
1.3.1	types of machine learning along with the field of use	3
1.3.2	Linear regression scatter plot	4
1.3.3	Logistic regression with probability output in s shape	4
1.3.4	Logistic Regreression	6
1.3.5	Decision tree	7
1.3.6	Random forest	7
1.3.7	scatter plot on K- means clustering	8
2.6.1	code for reading the dataset using pandas	10
2.6.3	Code for null values	11
3.3.1	various python libraries for Machine Learning	13
3.3.4.1	Seaborn(count plot)	15
3.3.4.2	Seaborn(count plot)	15
3.3.4.3	Seaborn(count plot)	16
3.3.5	Bar plot matplotlib lib	16
3.3.7	Imported libraries in jupyter notebook	17
3.4.1	Logistic Regression	17
3.5.1	Given Dataset	18
3.5.2	Testing and Training data	18

3.5.2	Applying Logistic Regression	19
4.1.1	confusion matrix	20
4.1.2	Code for confusion matrix	20
4.1.3	Test Dataset	24
4.1.4	Accuracy	21
4.1.5	TESTING DATA	22

CHAPTER-1:INTRODUCTION

1.1 Estimating Teaching Assistant Evaluation

The teaching assistant evaluation reviews *academic qualifications* , *relevant experience* , *quality of teaching* , and *professional contribution*. All these aspects can be assessed by the students , peers or by the teachers themselves

They are also the most consistent evaluators of the quality, the effectiveness of the learning process and satisfaction with course content, method of instruction, textbooks, homework, and student interest(Aleamoni, 1981).This paper includes the teacher's evaluation Aleamoni suggests that students are the main source of information about the learning environment, including teacher's ability to motivate them for continued learning , rapport or degree of communication between instructors and students

They are also the most consistent evaluators of the quality, the effectiveness of the learning process and satisfaction with course content, method of instruction, textbooks, homework, and student interest(Aleamoni, 1981).

1.2 ESTIMATING TA DATASET INFORMATION

- This data is collected by a The Wei-Yin Loh (Department of Statistics,UW-Madison).
- The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison.
- The scores were divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable.
- The given dataset contains 151 rows and 6 columns. The first 5 columns are features, the last column contains the classification label of '3's , '2's and '1's.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Native Eng	Course	Ins	Course	Summer o	Class size	Class attribute																
2	1	23	3	1	19	3																	
3	2	15	3	1	17	3																	
4	1	23	3	2	49	3																	
5	1	5	2	2	33	3																	
6	2	7	11	2	55	3																	
7	2	23	3	1	20	3																	
8	2	9	5	2	19	3																	
9	2	10	3	2	27	3																	
10	1	22	3	1	58	3																	
11	2	15	3	1	20	3																	
12	2	10	22	2	9	3																	
13	2	13	1	2	30	3																	
14	2	18	21	2	29	3																	
15	2	6	17	2	39	3																	
16	2	6	17	2	42	2																	
17	2	6	17	2	43	2																	
18	2	7	11	2	10	2																	
19	2	22	3	2	46	2																	
20	2	13	3	1	10	2																	
21	2	7	25	2	42	2																	
22	2	25	7	2	27	2																	
23	2	25	7	2	23	2																	
24	2	2	9	2	31	2																	
25	2	1	15	1	22	2																	
26	2	15	13	2	37	2																	
27	2	7	11	2	13	2																	
28	2	8	3	2	24	2																	
29	2	14	15	2	38	2																	

Fig 1.2 - excel sheet of the given Teaching Assistant dataset

1.3 COMMON MACHINE LEARNING ALGORITHMS AND GOALS

Then the The variety of machine learning algorithms are classified into three categories as

follows –

Supervised learning algorithms model the relationship between features (independent variables) and a label (target) from given set of observations. Then the model is used to predict the label of new observations using the features. Depending on the characteristics of the target variable i.e., it can be either be classification(discrete variable) or regression (continuous variable) the task is further engaged.

Unsupervised learning finds the structures in unlabeled data.

Reinforcement learning works on action-reward principle. An agent learned to reach the goal by continuously calculating the rewards that it gained from the actions

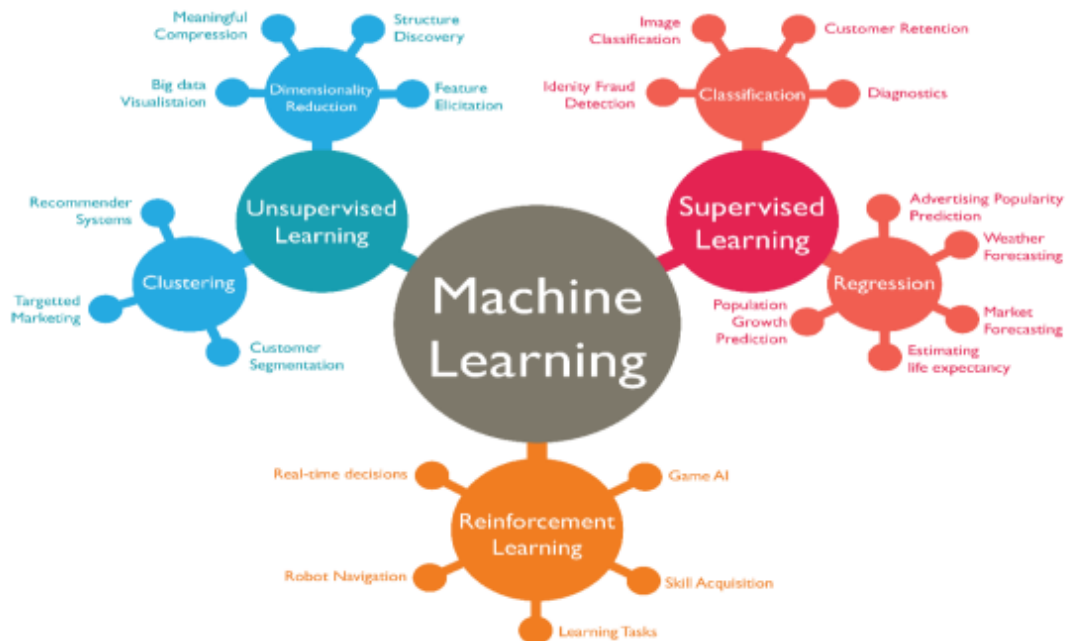


Fig 1.3.1 - types of machine learning along with the field of use

ALGORITHMS

1. Linear Regression is a supervised learning algorithm and tries to be a bridge between a continuous target variable and one or more independent variables by fitting a linear equation to the data. For choosing this algorithm, there needs to be a linear relation between independent and target variable. As scatter plot shows the positive correlation between an independent variable(x-axis) and dependent variable (y-axis).

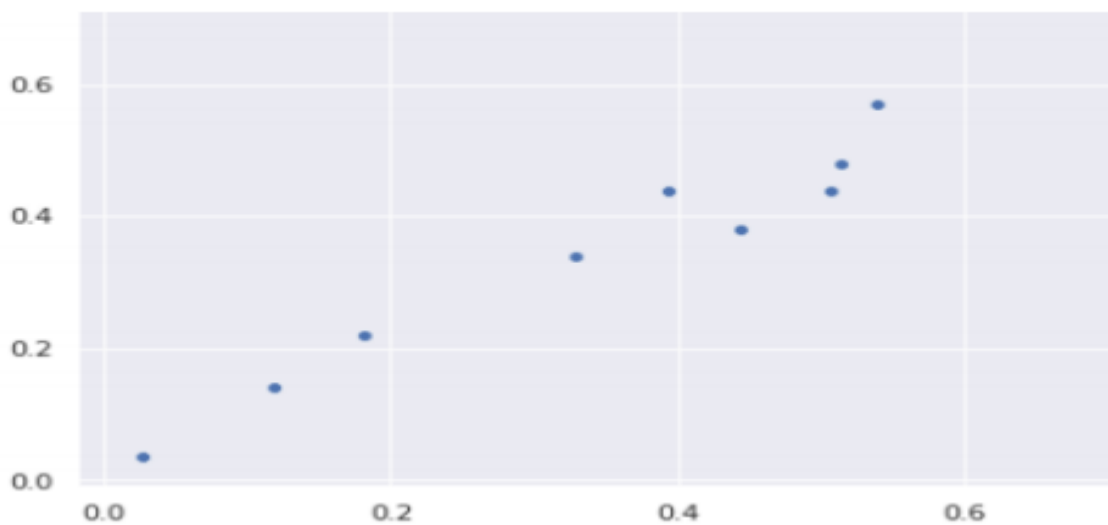


Fig 1.3.2 - Linear regression scatter plot

This would try to put regression line to represent relations. Common technique is ordinary-least squares (OLS). As a result, we could get a regression line as a outcome by minimizing sum square of distance between data points and regression line.

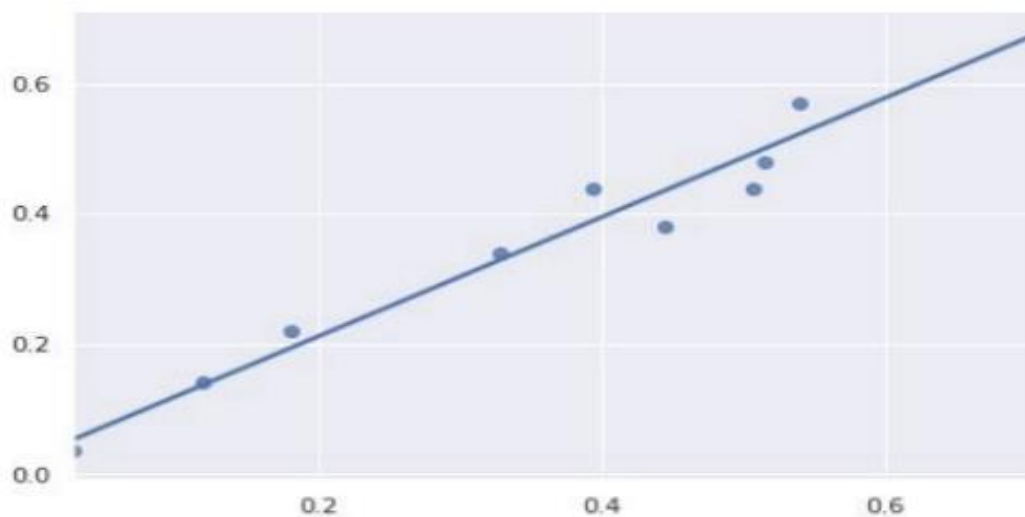


Fig 1.3.3- Linear regression scatter plot with regression line

2. Naïve Bayes is a supervised learning algorithm used for classification

problems, also called as Naïve Bayes Classifier. It assumes that features are independent of each other and there is no correlation between features. As assumption of features being uncorrelated is the reason for the name “naïve”.

Equation:

$$p(A|B) = \frac{p(A) \cdot p(B|A)}{p(B)} \quad (\text{Bayes' Theorem})$$

$p(A|B)$: Probability of event A given event B has already occurred

$p(B|A)$: Probability of event B given event A has already occurred

$p(A)$: Probability of event A

$p(B)$: Probability of event B

3. Logistic Regression is a supervised learning algorithm which is mostly used for binary classification problems. Even when regression contradicts with classification, here the spot is for logistic that refers to logistic function which does the classification task. It is simple but effective classification algorithms most commonly used for binary classification problems. Logistic function also known as sigmoid function.

Equation:

$$\text{Sigmoid Function: } y = \frac{1}{1 + e^{-x}}$$

Logistic regression takes linear equation as input and uses sigmoid function and logs odds to perform a binary problem. As result s shape graph will be the output

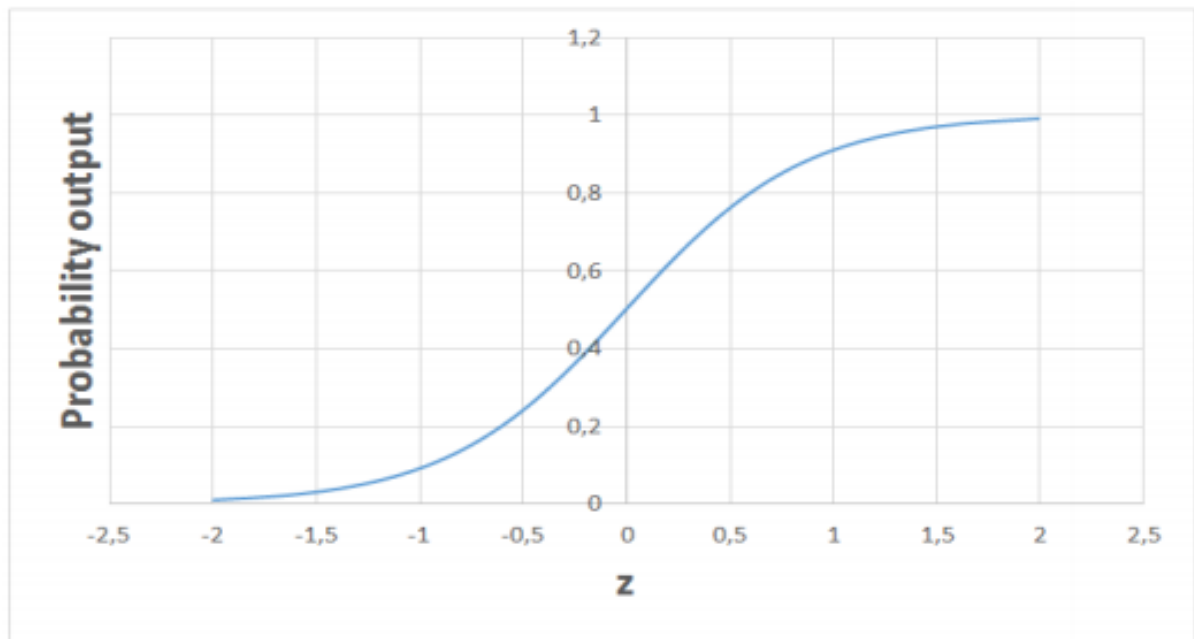


Fig 1.3.4 - Logistic regression with probability output in s shape

4. Decision Trees build upon continuously to partition the data. The aim of decision tree is to increase the predictiveness as much as possible at each stage so that the model keeps gaining information about the dataset.

Randomly splitting will not give us valuable insight into dataset. The purity of node is inversely proportional to distribution of different classes in that node.

Overfitting model would be too specific model and not be generalize well.

Though it achieves high accuracy with training set but poorly on new. The depth of the tree is controlled by `max_depth` parameter for decision tree algorithm in scikit-learn. Is also suitable to work on a mixture of feature data types.

Example

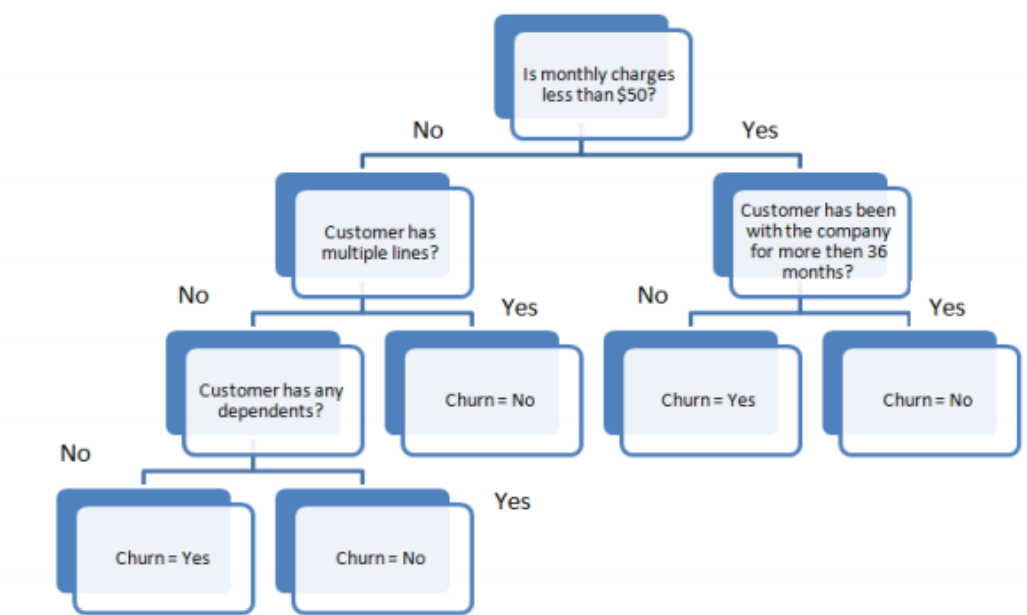


Fig 1.3.5 - Example for decision tree

5. Random Forest is an ensemble of many decision trees. They are built using a method called bagging where decision trees are used as parallel estimators. When used in classification problem, the result will be based on majority of vote received from each decision tree.

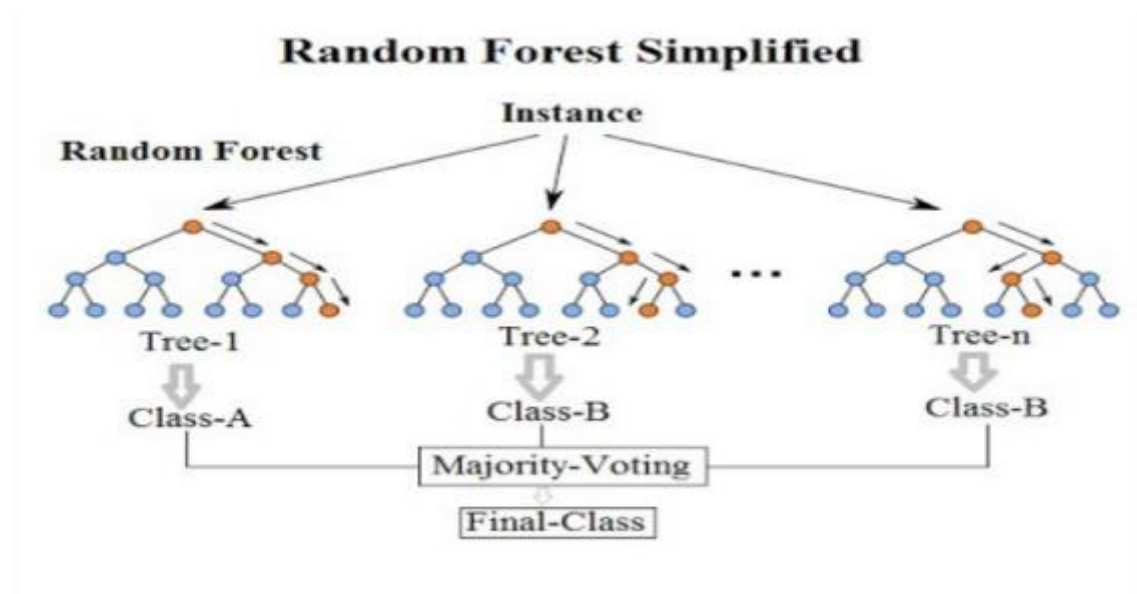


Fig 1.3.6 - Random forest outlier

6. K-means Clustering is a way to group of set of data points in a way that similar data points are together. Thus, they look for dissimilarities or similarities among data points. It is an unsupervised learning so there is no label associated with data points. They try to find the underlying structures of the data. Clustering is not Classification.

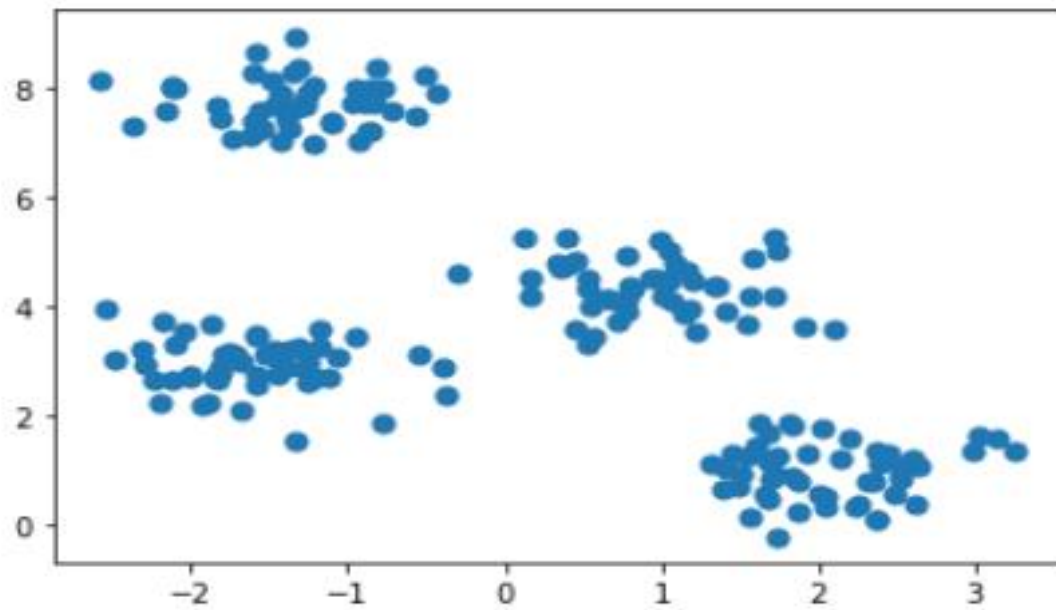


Fig 1.3.7 -scatter plot on K- means clustering

CHAPTER-2: AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM:

To predict the Class Attribute in the given Teaching Assistant Evaluation Data set using features from the given Data set

2.2 SCOPE:

The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. The scores were divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable.

In the given data set ,

1. Data set characteristics are Multivariate
2. Attribute characteristics are interger,real
3. Associated task – Classification
4. no.of instances – 151
5. no.of attributes – 5

2.3 ATTRIBUTES INFORMATION :

1. Whether of not the TA is a native English speaker (binary); 1=English speaker, 2=non-English speaker
2. Course instructor (categorical, 25 categories)
3. Course (categorical, 26 categories)
4. Summer or regular semester (binary) 1=Summer, 2=Regular
5. Class size (numerical)
6. Class attribute (categorical) 1=Low, 2=Medium, 3=High

2.4 DISTRIBUTION OF ATTRIBUTE NUMBER 17 : Class attribute

Value Number of instances with this value

```
-----  
3      52  
2      50  
1      49
```

Name: Class attribute, dtype: int64

2.5 DATA PREPARATION

In this project as a backbone tool python is used to carry out machine learning concepts. With the help of a software called Anaconda Navigator, a jupyter notebook is launched where it is already installed along with the navigator. jupyter notebook is an open-source web application that allows to create and share documents and has live code and also visualization.

After importing the required libraries the dataset will be read in the note book with help of data frame (two-dimensional labeled data structure with columns of potentially different types) and read_csv(desired file type).

2.6 READING DATASET

```
In [46]: pwd  
Out[46]: 'C:\\Users\\Jayaram Arava\\desktop'  
  
In [47]: cd desktop  
[WinError 2] The system cannot find the file specified: 'desktop'  
C:\\Users\\Jayaram Arava\\desktop  
  
In [48]: df=pd.read_csv("class.csv")
```

Fig 2.6.1- code for reading the dataset using pandas

The dataset provided cannot always be a fully valued set, in that case we need to prepare the data in such a way the machine understands what is the value that has been entered.

Check the values for null using info() function.

```
In [50]: df.head()
```

```
Out[50]:
```

	Native English Speaker	Course Instructor	Course	Summer or regular	Class size	Class attribute
0	1	23	3	1	19	3
1	2	15	3	1	17	3
2	1	23	3	2	49	3
3	1	5	2	2	33	3
4	2	7	11	2	55	3

```
In [51]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 151 entries, 0 to 150
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Native English Speaker 151 non-null    int64
1   Course Instructor      151 non-null    int64
2   Course                 151 non-null    int64
3   Summer or regular      151 non-null    int64
4   Class size             151 non-null    int64
5   Class attribute        151 non-null    int64
dtypes: int64(6)
memory usage: 7.2 KB
```

Fig 2.6.3- using info() function checking for null values

The dataset is now ready for creating a machine learning model.

CHAPTER- 3 : EXPERIMENTAL OR MATERIAL AND METHODS, ALGORITHMS USED

The given data set is in the form of Classification algorithm.so,we used classification types to predict the accuracy.

3.1 Types of classification algorithms used :

1. LogisticRegression Algorithm n

3.2 Logistic Regression Algorithm :

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1.**
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems.**
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

3.3 IMPORTED LIBRARIES

Libraries are collections of prewritten code that users can use to optimize tasks. In project as python is used for implementation tool, it has the most libraries as compared to other programming languages. More than of 60% machine learning developers use and goes for python as it is easy to learn. As python has comparatively large collection of libraries let's Ilook at the libraries that came in handy for mammographic dataset.

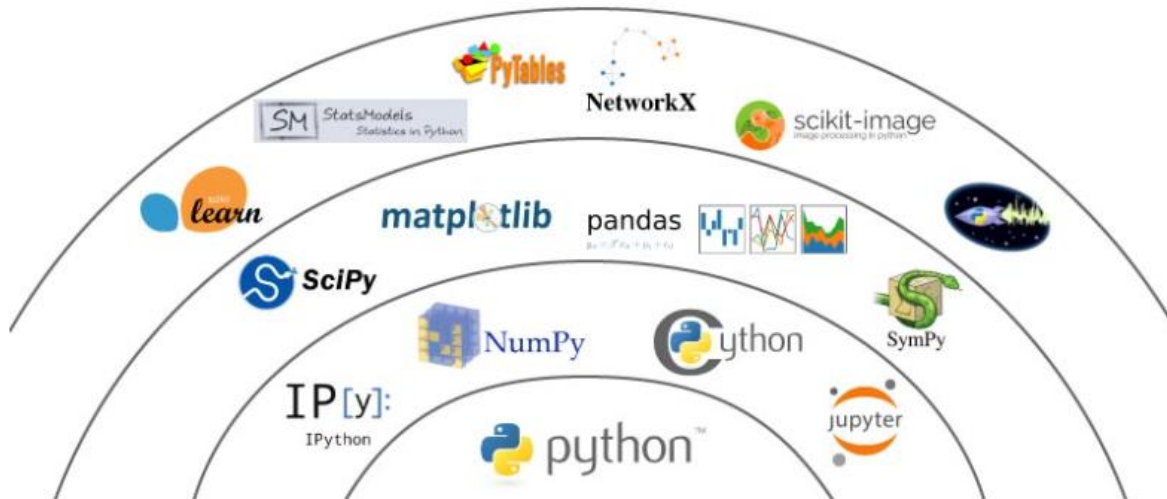


Fig 3.3.1- various python libraries for Machine Learning

LIBRARIES USED:

1. Pandas is a widely-used data analysis and manipulation library for python. It provides a lot of functions and methods that expedite the data analysis and preprocessing steps. IT also provides fast, flexible and expressive data structures working with relational or labeled or both easy and intuitive. Considered as fundamental high-level building block in performing practical, real-world data analysis in python. Has powerful tools like DataFrame and Series for analyzing

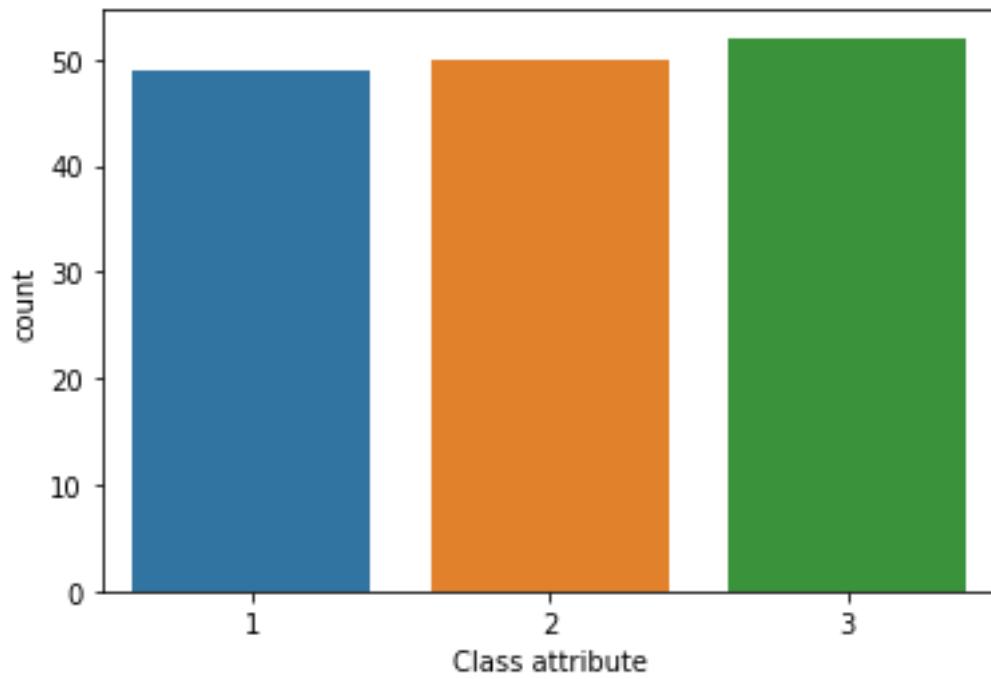
2. Numpy stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of countless of routines for processing those arrays. Using this mathematical and logical operations on arrays can be performed. The difference in using Numpy from pandas is, it works on numerical data whereas pandas on tabular data.

3. sklearn stands for Scikit-learn, a machine learning library. It is imported for various classification, regression and clustering algorithms including k-means, random forest, support vector machines, gradient boosting and DBSCAN. It is designed using libraries Numpy and Scipy.

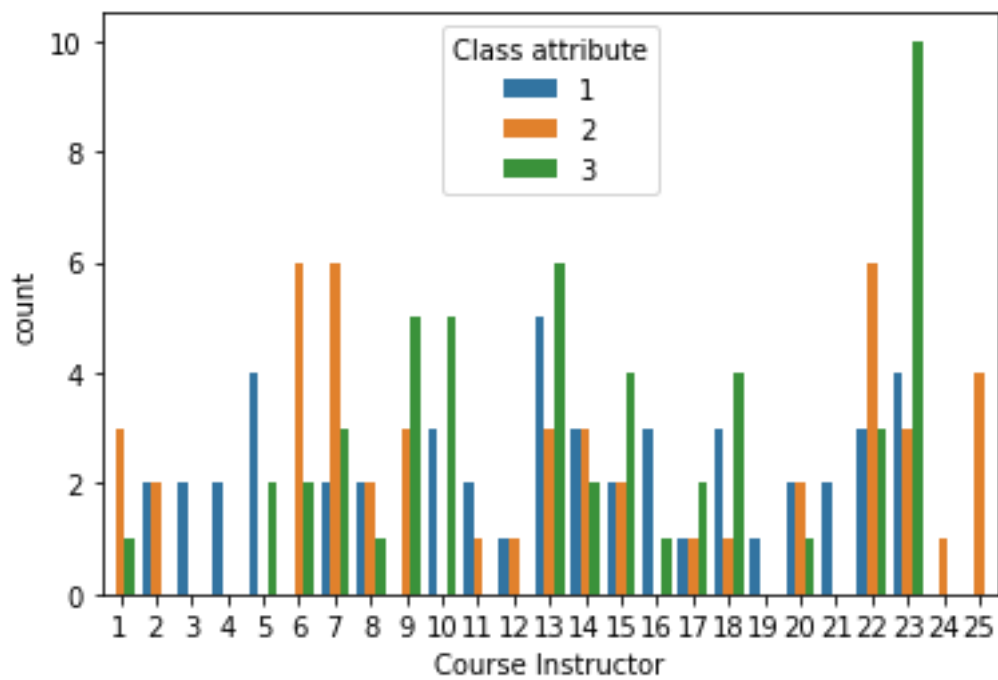
From sklearn another one called `model_selection` for training and testing the model imports `train_test_split`. It is a method setting a blueprint to analyze data and the using it to measure new data. Selecting a proper model allows to generate accurate results while making prediction. For proceeding, we need to train the model by using a specific dataset and test the model against another dataset.

By default, sklearn `train_test_split` will make random partitions for two subsets. We can also specify a random state for the operation. First, we need to split the dataset and then allocate the size for train and test. For this mammographic dataset we need train size as 70% (0.70) and test size as 30% (0.30) with the random state of 100.

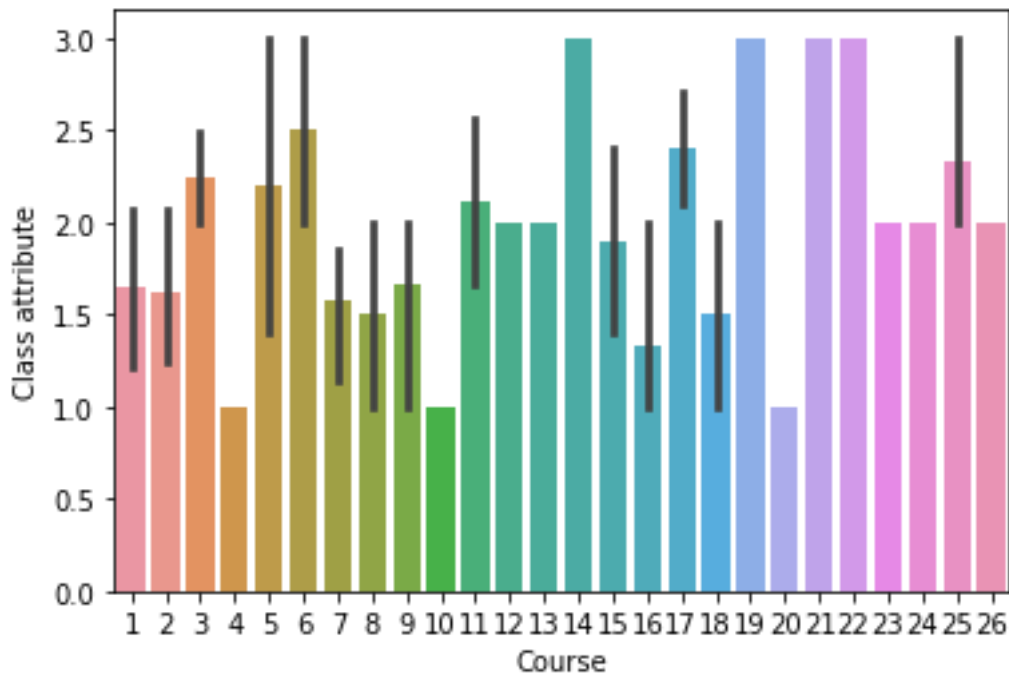
4. Seaborn is a library built on top of matplotlib. It used for data visualization and exploratory data analysis. They work easily with dataframes and pandas library. The graphs created can also be customized easily. It provides default styles and color palettes to make statistical plots more attractive. Also closely integrated to the data structures from pandas



count plot implementation of Seaborn library



count plot implementation of Seaborn library



count plot implementation of Seaborn library

5. Matplotlib.pyplot is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting. It make changes to figures.

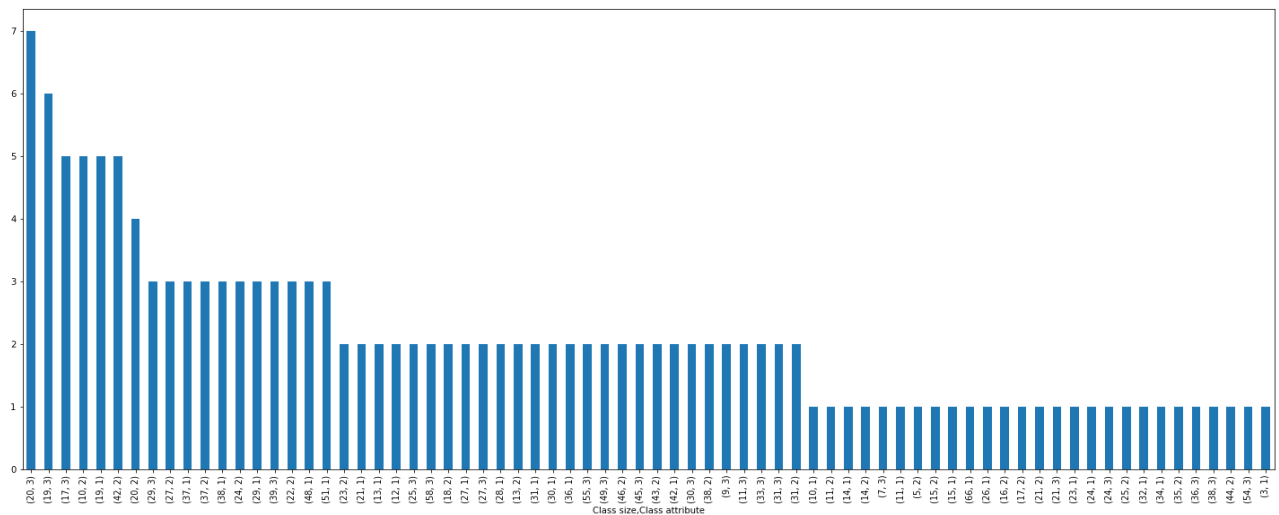


Fig 3.3.5- bar plot using matplotlib

```
In [45]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Fig 3.3.7- Imported libraries in jupyter notebook

3.4 LOGISTIC REGRESSION ALGORITHM

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1.**
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems.**

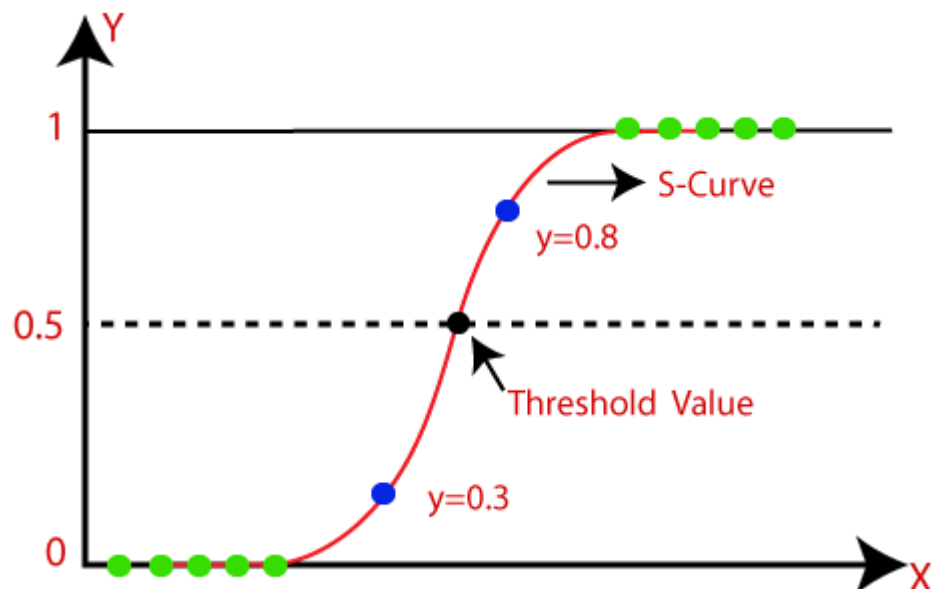
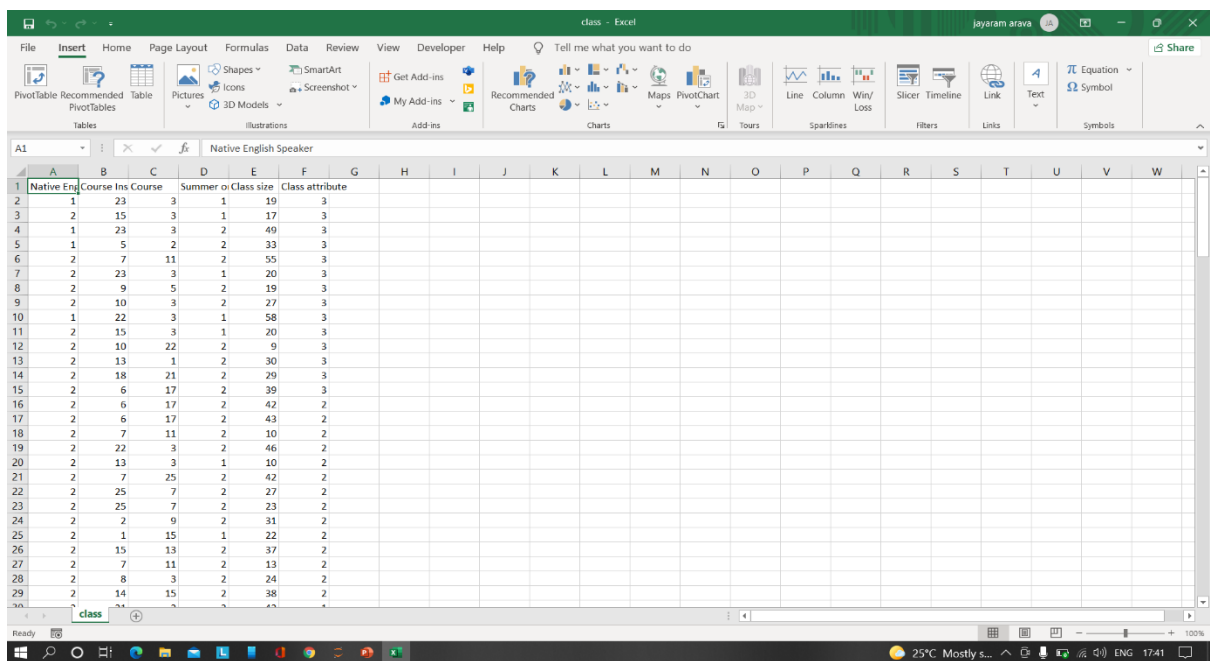


Fig 3.4.1 - Logistic Regression graph

3.5 Implementation of Logistic Regression :

1. Import the packages and classes you need.
2. Provide data to work with and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.
4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Native Eng	Course Ins	Course	Summer or	Class size	Class attribute																	
2	1	23	3	1	19	3																	
3	2	15	3	1	17	3																	
4	1	23	3	2	49	3																	
5	1	5	2	2	33	3																	
6	2	7	11	2	55	3																	
7	2	23	3	1	20	3																	
8	2	9	5	2	19	3																	
9	2	10	3	2	27	3																	
10	1	22	3	1	58	3																	
11	2	15	3	1	20	3																	
12	2	10	22	2	9	3																	
13	2	13	1	2	30	3																	
14	2	18	21	2	29	3																	
15	2	6	17	2	39	3																	
16	2	6	17	2	42	2																	
17	2	6	17	2	43	2																	
18	2	7	11	2	10	2																	
19	2	22	3	2	46	2																	
20	2	13	3	1	10	2																	
21	2	7	25	2	42	2																	
22	2	25	7	2	27	2																	
23	2	25	7	2	23	2																	
24	2	2	9	2	31	2																	
25	2	1	15	1	22	2																	
26	2	15	13	2	37	2																	
27	2	7	11	2	13	2																	
28	2	8	3	2	24	2																	
29	2	14	15	2	38	2																	

Fig 3.5.1 Given Dataset

```
In [58]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=45)
```

Fig 3.5.2 Train test split

```
In [61]: from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)

C:\Users\Jayaram Arava\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
  https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
  https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

Out[61]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)
```

Fig 3.5.2 Applying Logistic Regression

CHAPTER-4

RESULTS AND PERFORMANCE ANALYSIS

4.1 TRAINING AND ACCURACY (MODEL ANALYSIS)

Confusion matrix is to evaluate the accuracy of a classification. This visual metric plots the number of predictions made for each class for each possible class in a table, with each row corresponding to the actual labels and each column corresponding to a prediction. It is beneficial for detecting which actual classes are being detected the most, and what predicted classes are being misclassified as (Bhardwaj and Tiwari, 2015; Liu et al., 2009). To further highlight the misclassifications and compare predictions with other classifiers, the confusion matrices are normalized to show a percentage rather than a count.

The resulted confusion matrix has 45 incorrect prediction i.e., $(25 + 20 = 45)$.

From the matrix of y_test and y_pred with the help of metrics library the incorrect prediction is the outcome

		PREDICTIVE VALUES	
		POSITIVE (1)	NEGATIVE (0)
ACTUAL VALUES	POSITIVE (1)	TP	FN
	NEGATIVE (0)	FP	TN

Fig 4.1.1- outlier of confusion matrix

For the given dataset:

Confusion matrix

```
In [79]: from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
print('confusion matrix is',cm)

confusion matrix is [[8 6 0]
 [0 3 0]
 [1 8 5]]
```

Fig 4.1.2 - Confusion matrix

Accuracy is an evaluation metric would be misleading as it would not be representative of how well the classifier fitted the data. Additionally, in breast cancer detection, detecting FPs and FNs is primordial to avoid interpreting malignant as benign and vice versa, an interpretation which could harm the patient and eventually lead to their death.

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

For instance, if a dumb classifier that always classifies an image as “normal” is created, it would achieve 64.28% accuracy on the mini-MIAS dataset despite never

picking up abnormal cases. Therefore, a mixture of additional metrics should be used to assess how well the model learns the mammograms data and generalize to unseen cases

```
In [63]: from sklearn.metrics import accuracy_score  
print("The Accuracy is", accuracy_score(y_test, y_pred))
```

The Accuracy is 0.5161290322580645

Fig 4.1.4 Accuracy

Precision

corresponds to the number of correct positive predictions showing the model's ability to avoid labelling negative instances as positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

is the number of positive instances that are correctly predicted showing how well the model can find all positive instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score

 The F1 Score is the $2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$. It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

```
In [80]: from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1	0.89	0.57	0.70	14
2	0.18	1.00	0.30	3
3	1.00	0.36	0.53	14
accuracy			0.52	31
macro avg	0.69	0.64	0.51	31
weighted avg	0.87	0.52	0.58	31

Fig 4.1.5 For TESTING DATA

CHAPTER-5

5.1 SUMMARY AND CONCLUSIONS

- We have modelled the TEACHING ASSISTANT EVALUATION Data set using Logistic regression with the accuracy 51.621.
- In this Project the class attribute is mainly based on the course, course instructor ,native English speaker ,class size , summer or regular.
- In the future we can implement this project by doing certain changes in Data to improve accuracy. our team is interested in studying these methods and implement in another Machine learning Algorithms .

REFERENCES

https://www.sas.com/en_in/insights/analytics/machine-learning.html

<https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/caac.21660>

<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>

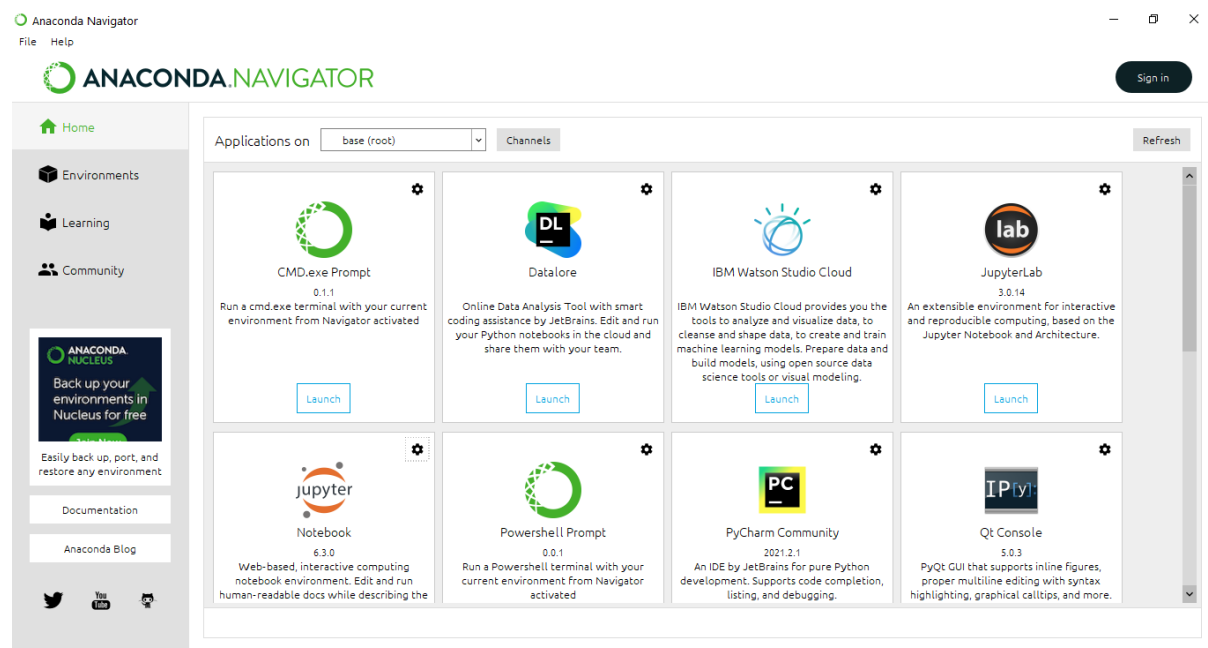
<https://docs.python.org/3/>

<https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition>

APPENDIX

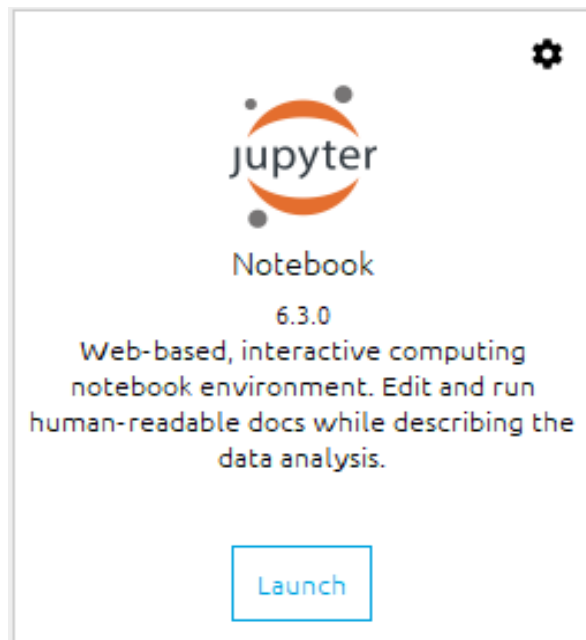
WORKING ENVIRONMENT

ANACONDA NAVIGATOR is desktop GUI used to launch applications and also manage packages in one place. Outlook

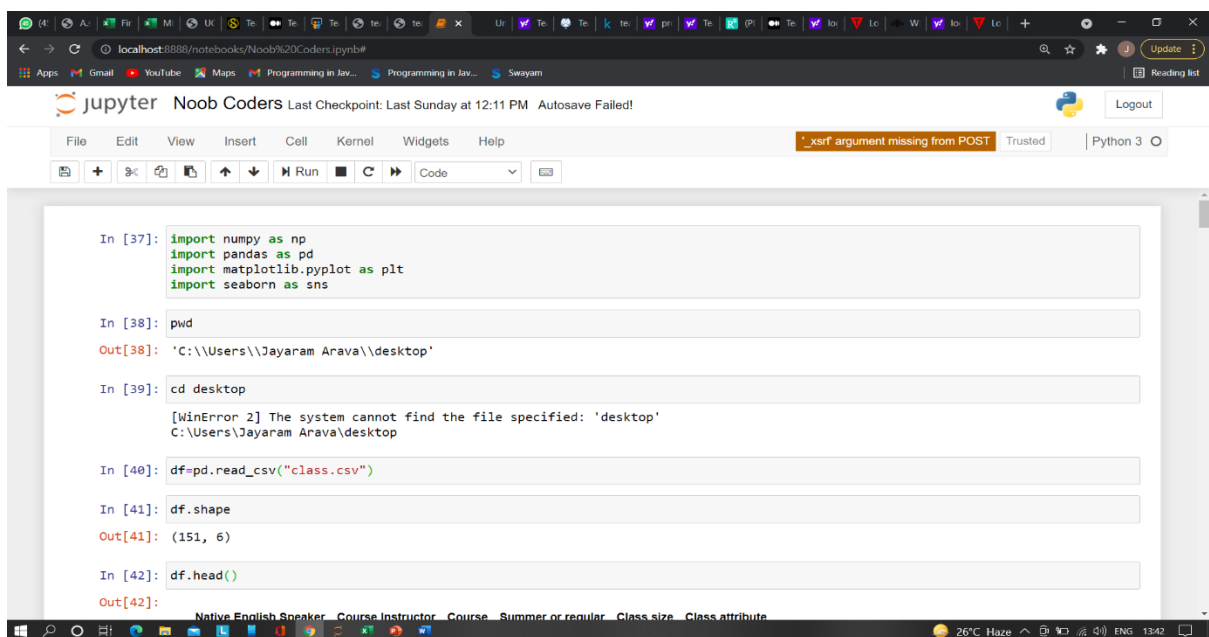


CODING ENVIRONMENT

Jupyter notebook from the anaconda navigator is launched along with all the preinstalled packages for python.



A.Screenshots and Outputs



localhost:8888/notebooks/Noob%20Coders.ipynb#

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help '_xsrftoken' argument missing from POST Trusted Python 3

Out[42]:

	Native English Speaker	Course Instructor	Course	Summer or regular	Class size	Class attribute
0	1	23	3	1	19	3
1	2	15	3	1	17	3
2	1	23	3	2	49	3
3	1	5	2	2	33	3
4	2	7	11	2	55	3

Checking for null values

In [43]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 151 entries, 0 to 150
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  --
0   Native English Speaker  151 non-null   int64
1   Course Instructor       151 non-null   int64
2   Course                  151 non-null   int64
3   Summer or regular       151 non-null   int64
4   Class size              151 non-null   int64
5   Class attribute         151 non-null   int64
dtypes: int64(6)
memory usage: 7.2 KB
```

In [44]: `#By above we can conclude that there are no null values`
`sns.heatmap(df.isnull(), cmap='viridis')`

Out[44]: `<matplotlib.axes._subplots.AxesSubplot at 0x1e9edf6bfc8>`

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed!

File Edit View Insert Cell Kernel Widgets Help

['_xsrf' argument missing from POST] Trusted Python 3

Analysing the data

```
In [45]: df['Class attribute'].value_counts()
Out[45]: 3    52
         2    50
         1    49
         Name: Class attribute, dtype: int64
```

```
In [46]: sns.countplot(x='Class attribute',data=df)
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x1e9edf40508>
```



```
In [47]: y=df['Class attribute']
In [48]: x=df.drop('Class attribute',axis=1)
In [49]: x.head(20)
```

```
Out[49]:
```

	Native English Speaker	Course Instructor	Course	Summer or regular	Class size
0	1	23	3	1	19
1	2	15	3	1	17
2	1	23	3	2	49
3	1	5	2	2	33
4	2	7	11	2	55
5	2	23	3	1	20
6	2	9	5	2	19
7	2	10	3	2	27
8	1	22	3	1	58

localhost:8888/notebooks/Noob%20Coders.ipynb#

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help

['_xsrftoken' argument missing from POST] Trusted Python 3

3	1	5	2	2	33
4	2	7	11	2	55
5	2	23	3	1	20
6	2	9	5	2	19
7	2	10	3	2	27
8	1	22	3	1	58
9	2	15	3	1	20
10	2	10	22	2	9
11	2	13	1	2	30
12	2	18	21	2	29
13	2	6	17	2	39
14	2	6	17	2	42
15	2	6	17	2	43
16	2	7	11	2	10
17	2	22	3	2	46
18	2	13	3	1	10
19	2	7	25	2	42

localhost:8888/notebooks/Noob%20Coders.ipynb#

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help

['_xsrftoken' argument missing from POST] Trusted Python 3

```
In [50]: y.head(20)
Out[50]:
```

0	3
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3
12	3
13	3
14	2
15	2
16	2
17	2
18	2
19	2

Name: Class attribute, dtype: int64

```
In [51]: sns.countplot(x='Course Instructor',hue='Class attribute',data=df)
```

localhost:8888/notebooks/Noob%20Coders.ipynb#

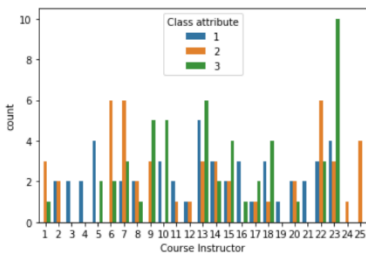
Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed!

File Edit View Insert Cell Kernel Widgets Help

xsrf argument missing from POST Trusted Python 3

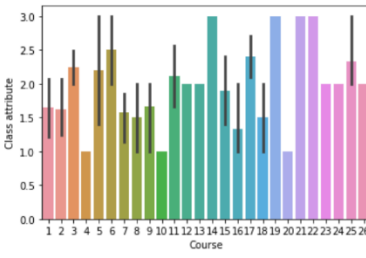
```
In [51]: sns.countplot(x='Course Instructor',hue='Class attribute',data=df)
```

```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1e9db27d08>
```



```
In [52]: sns.barplot(df['Course'],df['Class attribute'])
```

```
plt.show()
```

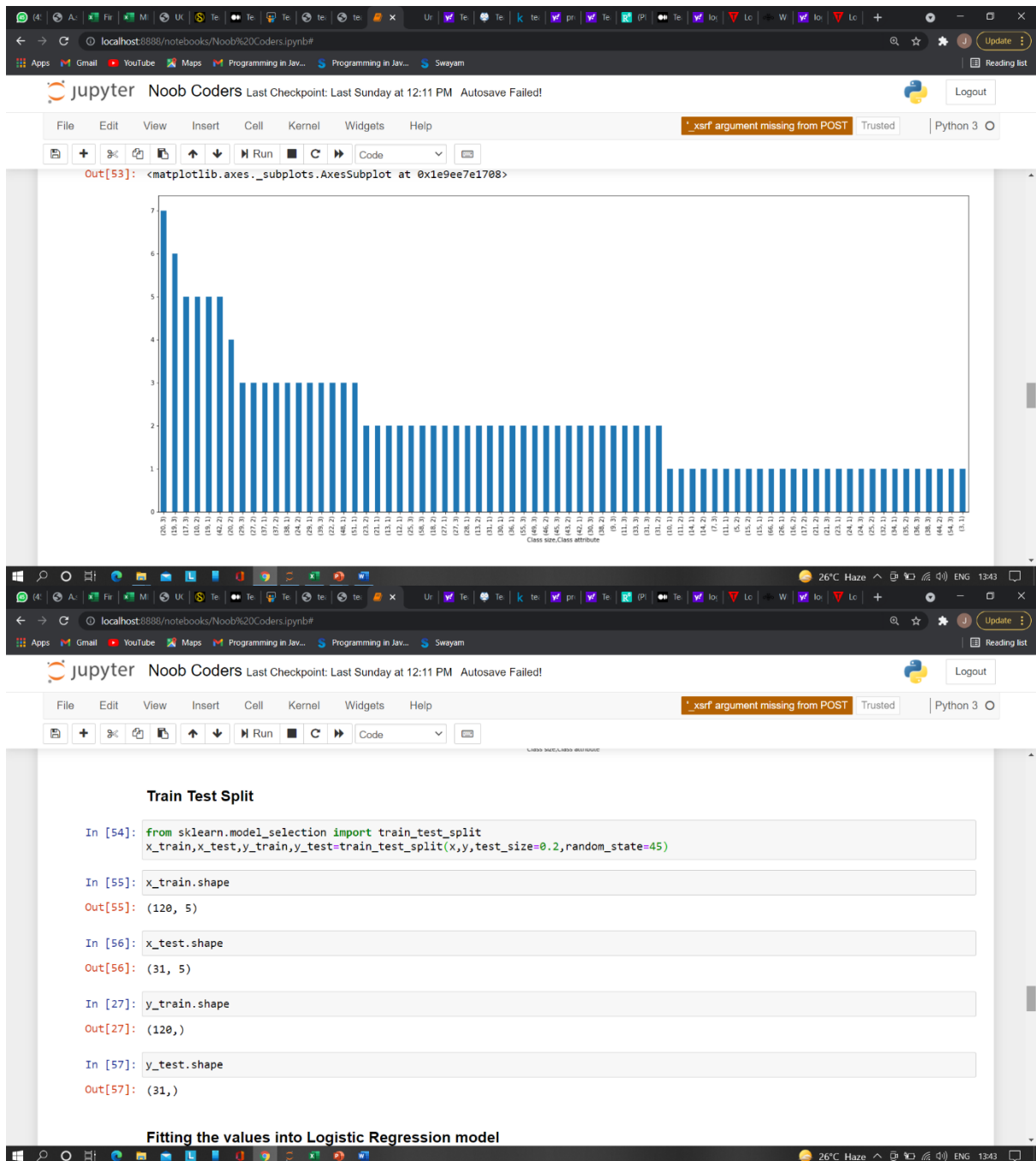


```
In [53]: plt.figure(figsize=(25,10))
```

```
df.groupby(['Class size'])['Class attribute'].value_counts().sort_values(ascending=False).plot(kind='bar')
```

```
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x1e9ee7e1708>
```





localhost:8888/notebooks/Noob%20Coders.ipynb#

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed!

File Edit View Insert Cell Kernel Widgets Help

Python 3

Fitting the values into Logistic Regression model

```
In [58]: from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
```

C:\Users\Jayaram Arava\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:940: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
extra_warning_msg=LOGISTIC_SOLVER_CONVERGENCE_MSG)

```
Out[58]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)
```

```
In [59]: y_pred=model.predict(x_test)
y_pred
```

```
Out[59]: array([3, 2, 2, 3, 1, 3, 2, 1, 3, 2, 1, 1, 1, 1, 2, 2, 1, 1, 3, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 1], dtype=int64)
```

Accuracy

```
In [30]: from sklearn.metrics import accuracy_score
print("The Accuracy is",accuracy_score(y_test,y_pred))
```

The Accuracy is 0.5161290322580645

Confusion matrix

```
In [31]: from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
print('confusion matrix is',cm)
```

confusion matrix is [[8 6 0]
[0 3 0]
[1 8 5]]

Precision and Recall

```
In [32]: from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

localhost:8888/notebooks/Noob%20Coders.ipynb#

Jupyter Noob Coders Last Checkpoint: Last Sunday at 12:11 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help '_xsrftoken' argument missing from POST Trusted Python 3

Run Code

Precision and Recall

```
In [32]: from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1	0.89	0.57	0.70	14
2	0.18	1.00	0.30	3
3	1.00	0.36	0.53	14
accuracy			0.52	31
macro avg	0.69	0.64	0.51	31
weighted avg	0.87	0.52	0.58	31

```
In [33]: df.corr()
```

```
Out[33]:
```

	Native English Speaker	Course Instructor	Course	Summer or regular	Class size	Class attribute
Native English Speaker	1.000000	-0.250510	0.134660	0.214417	-0.151547	-0.255466
Course Instructor	-0.250510	1.000000	-0.236010	-0.168573	-0.035765	0.075095
Course	0.134660	-0.236010	1.000000	0.224927	-0.033485	0.143103

accuracy 0.52 31
macro avg 0.51 31
weighted avg 0.58 31

```
In [33]: df.corr()
```

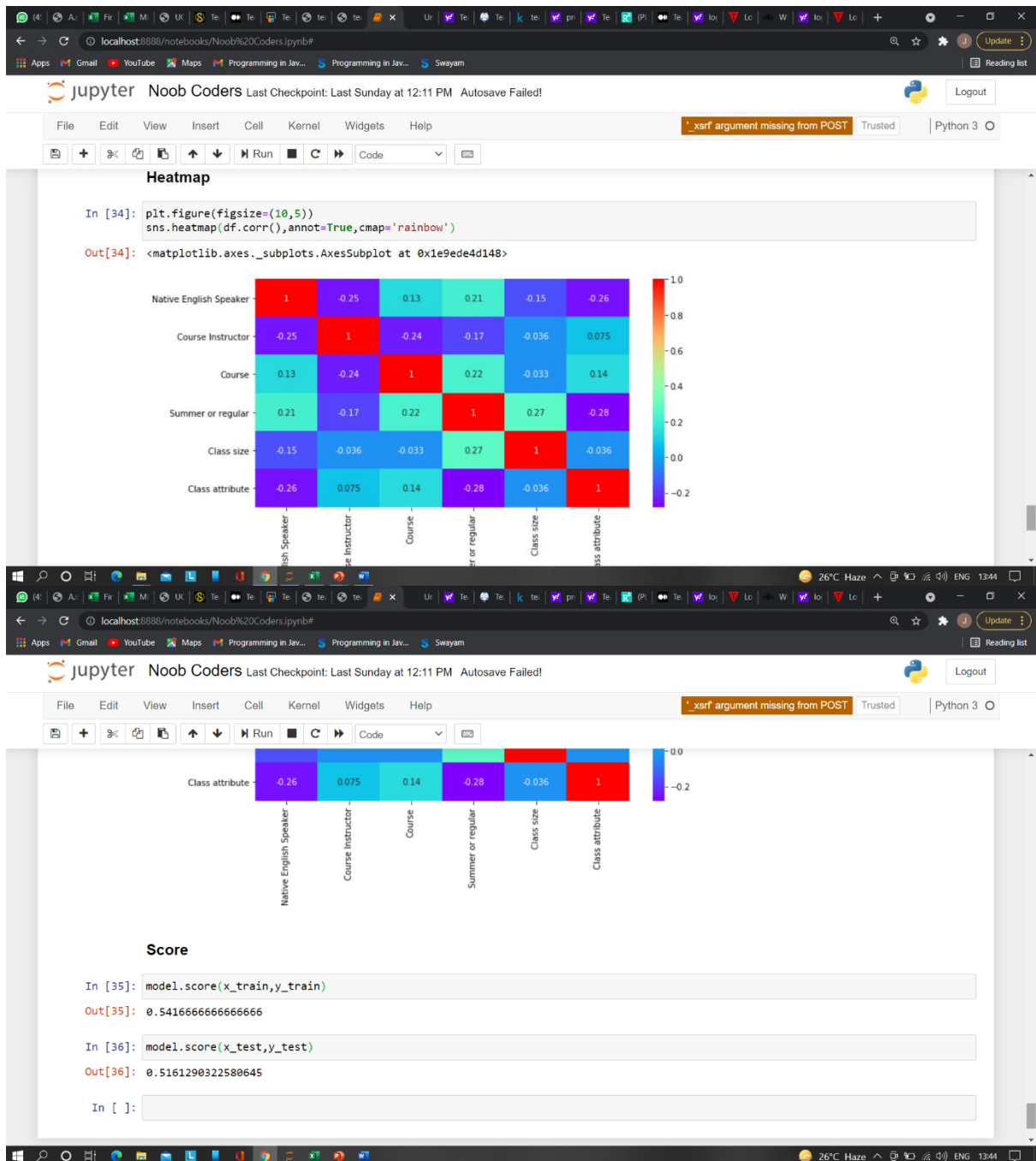
```
Out[33]:
```

	Native English Speaker	Course Instructor	Course	Summer or regular	Class size	Class attribute
Native English Speaker	1.000000	-0.250510	0.134660	0.214417	-0.151547	-0.255466
Course Instructor	-0.250510	1.000000	-0.236010	-0.168573	-0.035765	0.075095
Course	0.134660	-0.236010	1.000000	0.224927	-0.033485	0.143103
Summer or regular	0.214417	-0.168573	0.224927	1.000000	0.269554	-0.282742
Class size	-0.151547	-0.035765	-0.033485	0.269554	1.000000	-0.035676
Class attribute	-0.255466	0.075095	0.143103	-0.282742	-0.035676	1.000000

Heatmap

```
In [34]: plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='rainbow')
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1e9ede4d148>
```



SOURCE CODE :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
pwd
cd desktop
df=pd.read_csv("class.csv")
df.shape
df.head()
df.info()
#By above we can conclude that there are no null values
sns.heatmap(df.isnull(),cmap='viridis')
df['Class attribute'].value_counts()
sns.countplot(x='Class attribute',data=df)
y=df['Class attribute']
x=df.drop('Class attribute',axis=1)
x.head(20)
y.head(20)
sns.countplot(x='Course Instructor',hue='Class
attribute',data=df)
sns.barplot(df['Course'],df['Class attribute'])
plt.show()
plt.figure(figsize=(25,10))
df.groupby(['Class size'])['Class
attribute'].value_counts().sort_values(ascending=False).plot(kin
d='bar')
```

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,ra
ndom_state=45)
x_train.shape
x_test.shape
y_train.shape
y_test.shape
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
from sklearn.metrics import accuracy_score
print("The Accuracy is",accuracy_score(y_test,y_pred))
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
print('confusion matrix is',cm)
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
df.corr()
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='rainbow')
model.score(x_train,y_train)
model.score(x_test,y_test)

```