Data Science
(Classification of Cars)

Summer Internship Report Submitted in partial fulfillment of the requirement for undergraduate degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND

ENGINEERING submitted by

K. SANDEEP
221910306021

under the guidance of

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND

MANAGEMENT(GITAM) (Declared as Deemed-to-be-University u/s

3 of UGC Act 1956) HYDERABAD CAMPUS September 2021

DECLARATION

I submit this industrial training work entitled "**Classification of Cars**" to GITAM (Deemed to Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of "Bachelor of Technology" in "COMPUTER SCIENCE ENGINEERING". I declare that it was carried out independently by me under the guidance of_,
, GITAM (Deemed to Be University), Hyderabad, India. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree

Place: HYDERABAD Name: K Sandeep Date:27/07/2021

PinNo:221910306021

GITAM CERTIFICATE

This is to certify that the Industrial Training Report entitled **"DATA SCIENCE"** is being submitted by **K SANDEEP (221910306021)** in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science Engineering** at GITAM (Deemed to Be University), Hyderabad during the academic year 2019-2023

INTERNSHIP CERTIFICATE

ACKNOWLEDGEMENT

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who

have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Dr. N Seetha Ramaiah,** Principal, GITAM Hyderabad

I would like to thank Dr. **K. Manjunath Chari,** Head of the Department of Computer Science and Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present an internship report. It helped me a lot to realize what we study for.

I would like to thank the respected faculties who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well stacked till the end.

Student Name: K. SANDEEP

Student Pin no: 221910306021

ABSTRACT

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. The concepts of data science extend but are not limited to machine learning, deep learning and artificial intelligence. Through data science the unwanted data can be eliminated easily, which means clean data can be obtained very swiftly. The relation among parameters can also be found out easily with not much effort being applied. In the project of car classification evaluation, the concepts of data science play an important role as the main aim is to predict the "decision" parameter.

In the project every parameter is mapped with "decision" parameters in order to find the correlation among them. Also, the proportions of all parameters are distributed to get a clear understanding of density of values. This helps us approach a method to make predictions more accurate and suitable.

INDEX:

1. INFORMATION ABOUT DATA SCIENCE:

2. INFORMATION ABOUT MACHINE LEARNING

3. INFORMATION ABOUT PYTHON :

CHAPTER 1

1. Information About Data Science

1.1 What is Data Science

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, deep learning and big data.

1.2 Need of Data Science

Companies need to use data to run and grow their everyday business. The fundamental goal of data science is to help companies make quicker and better decisions,

which can take them to the top of their market, or at least – especially in the toughest red oceans – be a matter of long-term survival

Industries require data scientists to assist them in making a smarter decision. To predict the information everyone requires data scientists. Big Data and Data Science hold the key to the future. Data Science is important for better marketing.

1.3 Uses of Data Science

1. Internet Search

2. Digital Advertisements(Targeted Advertising and retargeting) 3.

Website Recommender Systems

4. Advanced Image Recognition

5. Speech Recognition

6. Gaming

7. Price Comparison Websites

8. Airline Route Planning

9. Fraud and Risk Detection

10. Delivery Logistics

CHAPTER 2

2. Information About Machine Learning

2.1 Introduction:

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

2.2 Importance of Machine Learning:

Consider some of the instances where machine learning is applied: the self driving Google car, cybersex fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and "more items to consider" and "get yourself a little something" on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today's data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that's in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

The process flow depicted here represents how machine learning works.

Fig 2.2.1 : The Process Flow

2.3 Uses of Machine Learning

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let's consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering,

network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data.

Traditionally, data analysis was always characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

2.4 Types of Machine Learning Algorithms

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.

2.4.1 Supervised Learning :

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labeled training data set – that is, a data set that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to "learn" how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multi class classification.

2.4.2 Unsupervised Learning :

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

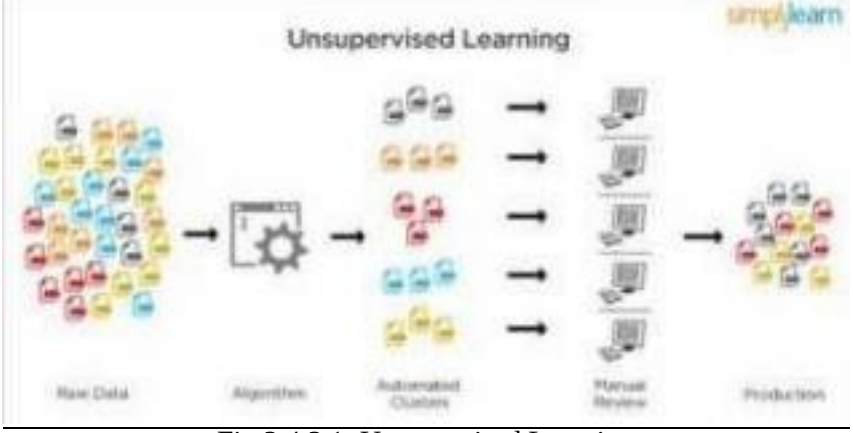

Fig 2.4.2.1: Unsupervised Learning

Popular techniques where unsupervised learning is used also include self organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

2.4.3 Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled data.

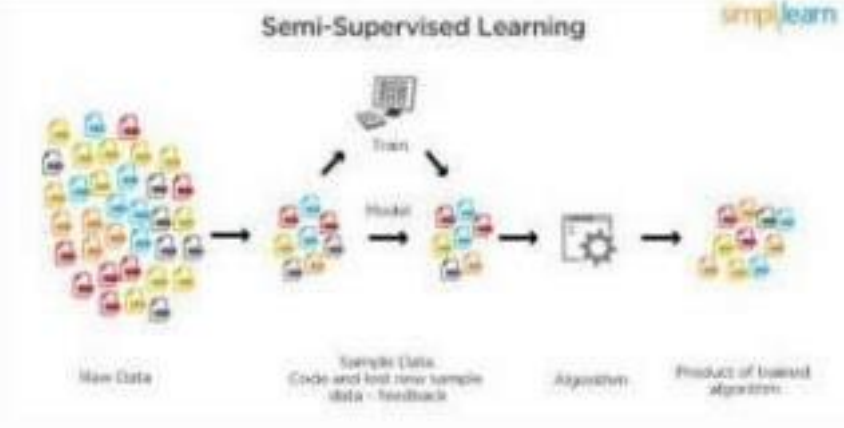

Fig2.4.3.1: Semi Supervised Learning

2.5 Relation between Data Mining, Machine Learning and Deep Learning

Machine learning and data mining use the same algorithms and techniques as data mining, except the kinds of predictions vary. While data mining discovered previously unknown patterns and knowledge, machine learning reproduces known patterns and knowledge—and further automatically applies that information to data, decision-making, and actions.

Deep learning, on the other hand, uses advanced computing power and special types of neural networks and applies them to large amounts of data to learn, understand, and identify complicated patterns. Automatic language translation and medical diagnoses are examples of deep learning.

CHAPTER 3
3.INFORMATION ABOUT PYTHON

Basic programming language used for machine learning is :

PYTHON **3.1 Introduction**

● Python is a high-level, interpreted, interactive and object-oriented scripting language.

● Python is a general purpose programming language that is often applied in scripting roles.

● Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.

● Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.

● Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

3.2 History of Python

● Python was developed by GUIDO VAN ROSSUM in early 1990's ●

Its latest version is 3.7 , it is generally called as python3

3.3 Features of Python

● Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax, This allows the student to pick up the language quickly.

● Easy-to-read: Python code is more clearly defined and visible to the eyes. ●

Easy-to-maintain: Python's source code is fairly easy-to-maintaining.

● A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.

● Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

● Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

● Databases: Python provides interfaces to all major commercial databases.

● GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

3.4 Python Setup

● Python is available on a wide variety of platforms including Linux and Mac OS X.

Let's understand how to set up our Python environment.

● The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

3.4.1 Installation(using python IDLE):

● Installing python is generally easy, and nowadays many Linux and Mac

OS distributions include a recent python.

● Download python from www.python.org

● When the download is completed, double click the file and follow the instructions to install it.

● When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.

Fig 3.4.1.1: Python download

3.4.2 Installation(using Anaconda):

● Python programs are also executed using Anaconda.

● Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.

● Conda is a package manager that quickly installs and manages

packages.

● In windows

● Step 1: Open Anaconda.com/downloads in web browser.

● Step 2: Download python 3.4 version for

(32-bits graphic installer/64 -bit graphic installer)

- Step 3: select installation type(all users)
- Step 4: Select path

(i.e. add anaconda to path & register anaconda as default python 3.4)

next click install and next click finish.

- Step 5: Open jupyter notebook (it opens in default browser)

Fig 3.4.2.1: Anaconda download

3.5 Python Variable Types

● Variables are nothing but reserved memory locations to store values.This means that when you create a variable you reserve some space in memory. ● Variables are nothing but reserved memory locations to store values. ● Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.

● Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.

● Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

- Python has five standard data types: –
- Numbers
- Strings
- Lists
- Tuples
- Dictionary

3.5.1 Python Numbers :

● Number data types store numeric values. Number objects are created when

you assign a value to them.

● Python supports four different numerical types – int (signed integers) long (long

integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

3.5.2 Python Strings:

● Strings in Python are identified as a contiguous set of characters represented in the quotation marks.

● Python allows for either pairs of single or double quotes.

● Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes

starting at 0 in the beginning of the string and working their way from -1 at the end.

● The plus (+) sign is the string concatenation operator and the asterisk (*) is

the repetition operator.

3.5.3 Python Lists:

● Lists are the most versatile of Python's compound data types.

● A list contains items separated by commas and enclosed within square brackets ([]).

● To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

● The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.

● The plus (+) sign is the list concatenation operator, and the asterisk (*) is

the repetition operator.

3.5.4 Python Tuples:

● A tuple is another sequence data type that is similar to the list.

● A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

● The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated.

● Tuples can be thought of as read-only lists.

● For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a

tuple. Tuples have no remove or pop method.

3.5.5 Python Dictionary:

● Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

● Dictionaries are enclosed by curly braces ({ }) and values can be assigned

and accessed using square braces ([]).

● You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you

understand that you can only use numbers to get items out of a list. ● What a dict does is

let you use anything, not just numbers. Yes, a dict associates one thing to another, no

matter what it is.

3.6 Python Functions

3.6.1 Defining a Function:

You can define functions to provide the required functionality. Here

are simple rules to define a function in Python. Function blocks begin with the

keyword

def followed by the function name and parentheses (i.e.()). Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses

The code block within every function starts with a colon (:) and is

indented. The statement returns [expression] exits a function, optionally passing back

an expression to the caller. A return statement with no arguments is the same as

return None.

3.6.2 Calling a Function:

Defining a function only gives it a name, specifies the parameters that are tobe included in the function and structures the blocks of code. Once the basic structureof a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

3.7 Oops Concepts

3.7.1 Class :

● Class: A user-defined prototype for an object that defines a set of attributes

that characterize any object of the class. The attributes are data members (class

variables and instance variables) and methods, accessed via dot notation.

● Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.

● Data member: A class variable or instance variable that holds data associated with a class and its objects.

● Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.

● Defining a Class: o We define a class in a very similar way to how we define a function. o Just like a function ,we use parentheses and a colon after the class name (i.e.():) when we define a class. Similarly, the body of our class is indented like a functional body is.

Fig 3.6.2.1: Defining a Class

3.7.2 init method in Class:

● The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.

● The init method has a special name that starts and ends with two underscores : init ().

4. Project Name : Classification of Cars

4.1 Project Requirements

4.1.1 Packages used.

✓ Pandas

✓ Matplotlib

✓ Numpy

✓Scikit-learn

✓ Category Encoder

4.1.2 Versions of the packages.

✓ Pandas : 0.25.1

✓ Matplotlib : 3.1.1

✓Numpy : 1.16.5

✓Scikit-learn:0.23.1

✓ Category Encoder 2.3

4.1.3 Algorithms used.

✓ Logistic Regression

✓ Decision Tree

✓ Random Forest

4.2 CASE STUDY

4.2.1 DATA SET:

The given data set consists of the following parameters:

A.buying

B.maint

C. doors

D. persons

E. lug_boot

F. safety

G.class

4.3 OBJECTIVE OF THE CASE STUDY:

To get a better understanding and chalking out a plan of action for solution of the client, we have adapted the viewpoint of looking at label categories and for further deep understanding of the problem, we have also removed column id as it is just a number and here based on the column label we classified tweets as positive or negative and taking input as tweets column. And output is label column.

CHAPTER 5 :MODEL BUILDING

5.1 PREPROCESSING OF THE DATA:

Pre-processing of the data actually involves the following steps:

5.1.1 GETTING THE DATASET: We can get the data set from the database or we can get the data from the client.

```
In [3]: #Load the dataset
        data = pd.read_csv("car_evaluation.csv",header=None)
        data.head()
```

Out[3]:

	0	1	2	3	4	5	6
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

5.1.2 IMPORTING THE LIBRARIES:

We have to import the libraries as per the requirement of the algorithm.

```
In [2]: import numpy as np #linear alegbra
        import pandas as pd #data processing CSV file I/O
        import matplotlib.pyplot as plt #data visualizaion
        import category_encoders as ce #data encoding
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.metrics import recall_score, precision_score, accuracy_score, plot_confusion_matrix, classification_report, f1_score
```

5.1.3 IMPORTING THE DATA-SET:

Pandas in python provide an interesting method read_csv(). The
The read_csv function reads the entire dataset from a comma separated values file and we can assign it to a Dataframe to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be accessed using the data frame. Any missing values or NaNvalues have to be cleaned.

In [3]:
```
#Importing Data Set
data = pd.read_csv("car_evaluation.csv",header=None)
data.head()
```

Out[3]:

	0	1	2	3	4	5	6
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

5.2 DATA SHAPE:

We use data shape to know how many rows and columns are present in our dataset.

In [3]:
```
data = pd.read_csv("car_evaluation.csv",header=None)
data.head()
```

Out[3]:

	0	1	2	3	4	5	6
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

5.2.1 Changing column names

In [4]:
```
# Changing column names for betterment
col_names = ['buying','maint','doors','persons','lug_boot','safety','class']
data.columns = col_names
data.head()
```

Out[4]:

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

5.2.2 Data Exploration :

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   buying     1728 non-null   object
 1   maint      1728 non-null   object
 2   doors      1728 non-null   object
 3   persons    1728 non-null   object
 4   lug_boot   1728 non-null   object
 5   safety     1728 non-null   object
 6   class      1728 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```

5.3 Inference:

```
In [6]: def show(data):
          for i in data.columns[1:]:
            print("Feature: {} with {} Levels".format(i,data[i].unique()))

        show(data)
```

```
Feature: maint with ['vhigh' 'high' 'med' 'low'] Levels
Feature: doors with ['2' '3' '4' '5more'] Levels
Feature: persons with ['2' '4' 'more'] Levels
Feature: lug_boot with ['small' 'med' 'big'] Levels
Feature: safety with ['low' 'med' 'high'] Levels
Feature: class with ['unacc' 'acc' 'vgood' 'good'] Levels
```

5.3.1 Feature Engineering:

Feature Engineering

```
In [8]: data.dtypes

Out[8]: buying      object
        maint       object
        doors       object
        persons     object
        lug_boot    object
        safety      object
        class       object
        dtype: object
```

5.3.2 Transforming the data:

```
In [9]: encoder = ce.OrdinalEncoder(cols = ['buying','maint','doors','persons','lug_boot','safety','class'])
        data = encoder.fit_transform(data)
        data.head()
```

Out[9]:

	buying	maint	doors	persons	lug_boot	safety	class
0	1	1	1	1	1	1	1
1	1	1	1	1	1	2	1
2	1	1	1	1	1	3	1
3	1	1	1	1	2	1	1
4	1	1	1	1	2	2	1

5.4 Train and Test Method:-

Splitting Data into Train Test

```
In [10]: x = data.drop(['class'], axis = 1)
         y = data['class']

         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)
         print("X_train: {}".format(x_train.shape))
         print("X_test: {}".format(x_test.shape))
         print("Y_train: {}".format(y_train.shape))
         print("Y_test: {}".format(y_test.shape))

         X_train: (1209, 6)
         X_test: (519, 6)
         Y_train: (1209,)
         Y_test: (519,)
```

6. ALGORITHMS USED:

The techniques used are Logistic regression, Decision Tree and Naïve Bayes Algorithm.

6.1 LOGISTIC REGRESSION:

Logistic regression is a statistical model that in its basic form uses a Logistic function to model a binary dependent variable, although many more Complex extensions exist. In regression analysis,**logistic regression** (or **logit Regression)** is estimating the parameters of a logistic model (a form of binary Regression). Mathematically, a binary logistic model has a dependent variable with Two possible values, such as pass/fail which is represented by an indicator variable

6.1.2 LOGISTIC REGRESSION ALGORITHM:

6.1.3Importing the algorithm:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

6.1.4 Applying the algorithm:

1. Logistics Regression

```
In [12]: lr = LogisticRegression(max_iter = 1000,random_state = 48)
         lr.fit(x_train,y_train)

         yp_train = lr.predict(x_train)
         yp_test = lr.predict(x_test)

         evaluation_parametrics(y_train,yp_train,y_test,yp_test)
```

```
Classification Report for Train Data
              precision    recall  f1-score   support

           1       0.88      0.93      0.90       852
           2       0.66      0.58      0.62       266
           3       0.79      0.63      0.70        41
           4       0.53      0.38      0.44        50

    accuracy                           0.82      1209
   macro avg       0.71      0.63      0.67      1209
weighted avg       0.81      0.82      0.82      1209

Classification Report for Test Data
              precision    recall  f1-score   support

           1       0.87      0.93      0.90       358
           2       0.66      0.58      0.62       118
           3       0.75      0.75      0.75        24
           4       0.55      0.32      0.40        19

    accuracy                           0.82       519
   macro avg       0.71      0.64      0.67       519
weighted avg       0.81      0.82      0.81       519
```

```
Accuracy on Train Data is: 0.82
Accuracy on Test Data is: 0.82
Precision on Train Data is: 0.81
Precision on Test Data is: 0.81
Recall on Train Data is: 0.82
Recall on Test Data is: 0.82
F1 Score on Train Data is: 0.82
F1 Score on Test Data is: 0.81
```

6.2 Decision Tree

6.2.1 Importing the algorithm:

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
```

6.2.2 Applying the algorithm:

```
In [13]: dt = DecisionTreeClassifier(max_depth = 7,random_state = 48) # Keeping max_depth = 7 to avoid overfitting
         dt.fit(x_train,y_train)

         yp_train = dt.predict(x_train)
         yp_test = dt.predict(x_test)

         evaluation_parametrics(y_train,yp_train,y_test,yp_test)
```

```
--------------------------------------------------------------------------
Classification Report for Train Data
              precision    recall  f1-score   support

           1       0.98      0.97      0.98       852
           2       0.86      0.94      0.90       266
           3       0.93      0.68      0.79        41
           4       0.81      0.76      0.78        50

    accuracy                           0.95      1209
   macro avg       0.90      0.84      0.86      1209
weighted avg       0.95      0.95      0.95      1209

Classification Report for Test Data
              precision    recall  f1-score   support

           1       0.97      0.98      0.97       358
           2       0.86      0.81      0.83       118
           3       0.77      0.83      0.80        24
           4       0.52      0.58      0.55        19

    accuracy                           0.92       519
   macro avg       0.78      0.80      0.79       519
weighted avg       0.92      0.92      0.92       519

--------------------------------------------------------------------------
Accuracy on Train Data is: 0.95
Accuracy on Test Data is: 0.92
--------------------------------------------------------------------------
Precision on Train Data is: 0.95
Precision on Test Data is: 0.92
--------------------------------------------------------------------------
Recall on Train Data is: 0.95
Recall on Test Data is: 0.92
--------------------------------------------------------------------------
F1 Score on Train Data is: 0.95
F1 Score on Test Data is: 0.92
--------------------------------------------------------------------------
```

6.3 Random Forest

6.3.1 Importing the algorithm:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

6.3.2 Applying the algorithm:

In [14]:
```
rf = RandomForestClassifier(max_depth = 7,random_state = 48) # Keeping max_depth = 7 same as DT
rf.fit(x_train,y_train)

yp_train = rf.predict(x_train)
yp_test = rf.predict(x_test)

evaluation_parametrics(y_train,yp_train,y_test,yp_test)
```

```
------------------------------------------------------------------------
Classification Report for Train Data
              precision    recall  f1-score   support

           1       0.99      0.99      0.99       852
           2       0.88      0.98      0.93       266
           3       0.97      0.76      0.85        41
           4       0.97      0.74      0.84        50

    accuracy                           0.97      1209
   macro avg       0.96      0.86      0.90      1209
weighted avg       0.97      0.97      0.97      1209
```

```
Classification Report for Test Data
              precision    recall  f1-score   support

           1       0.98      0.97      0.98       358
           2       0.85      0.89      0.87       118
           3       0.88      0.88      0.88        24
           4       0.69      0.58      0.63        19

    accuracy                           0.94       519
   macro avg       0.85      0.83      0.84       519
weighted avg       0.94      0.94      0.94       519

------------------------------------------------------------------------
Accuracy on Train Data is: 0.97
Accuracy on Test Data is: 0.94
------------------------------------------------------------------------
Precision on Train Data is: 0.97
Precision on Test Data is: 0.94
------------------------------------------------------------------------
Recall on Train Data is: 0.97
Recall on Test Data is: 0.94
------------------------------------------------------------------------
F1 Score on Train Data is: 0.97
F1 Score on Test Data is: 0.94
------------------------------------------------------------------------
```

CONCLUSION

1. Implemented Logistic Regression, Decision Tree & Random Forest models.
2. Evaluated Evaluation Metrics (Confusion Matrix, Precision, Recall, Accuracy & F1-Score).
3. Random Forest performed slightly better out of the 3 models.