VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka

PROJECT REPORT

ON

"LIVER DISEASE ANALYSIS"

Submitted in Partial Fulfilment of requirement for the award of the degree of

Bachelor of Engineering

in

Computer Science & Engineering

Submitted by :

MD IRFAN KHAN (3GN17CS400)

POOJA TANDLE (3GN16CS056)

PAWAR ARATHI (3GN16CS055)

Under the guidance of:

Prof. RAGEENA M

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE,

BIDAR - 585 403 KARNATAKA

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI
GURU NANAK DEV ENGINEERING COLLEGE,
BIDAR-585403, KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship work entitled **"LIVER DISEASE ANALYSIS"** carried out by **MD IRFAN KHAN (3GN17CS400), POOJA TANDLE (3GN16CS056), PAWAR ARATHI (3GN16CS055)**, a bonafide student of Guru Nanak Dev Engineering College in partial fulfilment for the reward of **Bachelor of Engineering** in **Computer Science and Engineering** under **Visvesvaraya Technological University**, **Belagavi** during the academic year **2019-2020** is true representation of Project work completed satisfactorily.

_____	_____	_____
Guide	**Co-Ordinator**	**HOD**

EXTERNAL VIVA

Examiners: 1) _____ 2) _____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without complementing those who made it possible and whose guidance and encouragement made our efforts successful. So, my sincere thanks to all those who have supported me in completing this Internship successfully.

I am highly intended my Project guide **Prof. RAGEENA M,** for guiding and giving me timely advices and suggestions in successful completion of project work.

My sincere thanks to **Dr. RAVINDRA EKLARKER**, Principal, GNDEC and **Dr. DAYANAND J**, Head of the Department of Computer Science and Engineering, GNDEC for their encouragement, support and guidance to the student community in all fields of education. I am grateful to our institution for providing us a congenial atmosphere to carry out the Internship successfully.

I extend my sincere thanks to my department faculty members of computer science and engineering and also non-teaching staff for supporting me directly or indirectly for the completion of this Internship.

MD IRFAN KHAN
(3GN17CS400)

Content Page No

CHAPTER-1

INTRODUCTION

The use of intelligence systems in medical diagnosis is increasing gradually. There is no doubt that evaluation of data taken from patients and the decisions of experts are the most important factors in diagnosis. But, expert systems and different artificial intelligence techniques for classification also help experts in a great deal.

Classification systems, helping possible errors that can be done because of a fatigued or inexperienced expert to be minimized, provide medical data to be examined in a shorter time and more detailed. Problems with liver patients are not easily discovered in an early stage as it will be functioning normally even when it is partially damaged. Early diagnosis of liver problems will increase the patient's survival rate. Liver disease can be diagnosed by analyzing the levels of enzymes in the blood.

Furthermore, nowadays mobile devices are widely used for the supervision of humans' body conditions. In addition, automatic classification algorithms are demanded. According to their implements for liver diseases (almost definitely mobile capable or web capable), that decline the patient queue at the liver experts such as endocrinologists. Liver disorders are also an important disease in medicine. Levels of enzymes combined to blood are analyzed in Liver Disorders diagnosis. It can be a lot of possible errors in this diagnosis due to the number of enzymes to be many as well as the effects of different taken alcohol rates to vary from one patient to the other. Medical Data Company includes 345 specimens consisting of six fields and two classes. Each sample is taken from a single man. Two hundred of these samples are of one class with the remaining 145 are possessed by the other. First Five attributes of the collected data samples are the outcomes of blood tests while the last attribute includes daily alcohol consumption.

1.1 Introduction to Machine Learning:

The journey of AI began in the 1950's when the computing power was a fraction of what it is today.

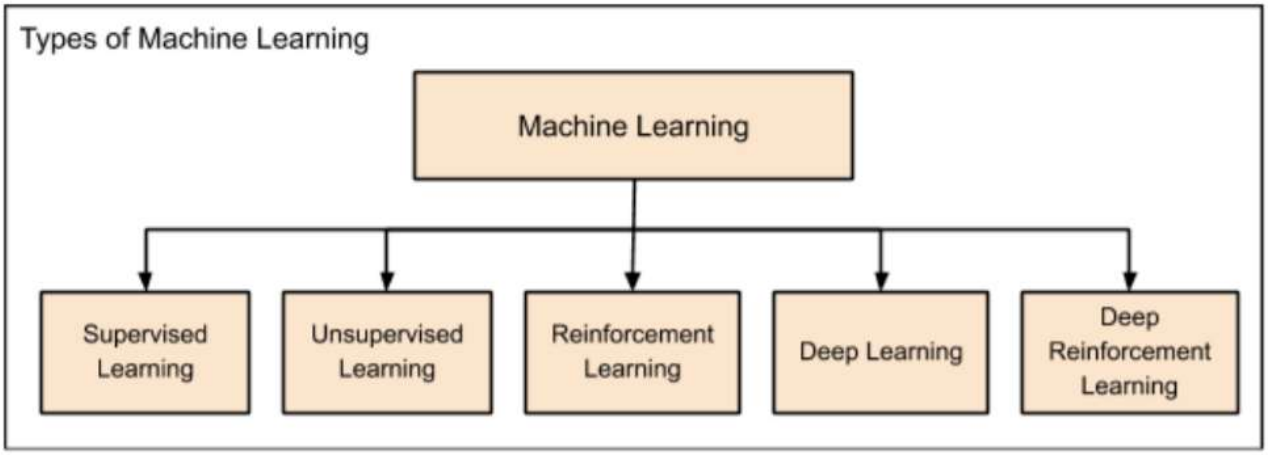
AI started out with the predictions made by the machine in a fashion a statistician does predictions using his calculator. Thus, the initial entire AI development was based mainly on statistical techniques.
Statistical Techniques The development of today's AI applications started with using the age-old traditional statistical techniques. There are several other such statistical techniques which are successfully applied in developing so-called AI programs. We say "so-called" because the AI programs that we have today are much more complex and use techniques far beyond the statistical techniques used by the early AI programs.
Some of the examples of statistical techniques that are used for developing AI applications in those days and are still in practice are listed here:

- Regression
- Classification
- Clustering
- Probability Theories
- Decision Trees

Here we have listed only some primary techniques that are enough to get you started on AI without scaring you of the vastness that AI demands. If you are developing AI applications based on limited data, you would be using these statistical techniques. However, today the data is abundant. To analyse the kind of huge data that we possess statistical techniques are of not much help as they have some limitations of their own. More advanced methods such as deep learning are hence developed to solve many complex problems. As we move ahead in this tutorial, we will understand what Machine Learning is and how it is used for developing such complex AI applications

1.2 Categories of Machine Learning:

Machine Learning is broadly categorized under the following headings:



Machine learning evolved from left to right as shown in the above diagram.

- Initially, researchers started out with Supervised Learning. This is the case of housing price prediction discussed earlier.
- This was followed by unsupervised learning, where the machine is made to learn on its own without any supervision.
- Scientists discovered further that it may be a good idea to reward the machine when it does the job the expected way and there came the Reinforcement Learning.
- Very soon, the data that is available these days has become so humongous that the conventional techniques developed so far failed to analyse the big data and provide us the predictions.
- Thus, came the deep learning where the human brain is simulated in the Artificial Neural Networks (ANN) created in our binary computers.
- The machine now learns on its own using the high computing power and huge memory resources that are available today.
- It is now observed that Deep Learning has solved many of the previously unsolvable problems.
- The technique is now further advanced by giving incentives to Deep Learning networks as awards and there finally comes Deep Reinforcement Learning

1.3 Supervised Learning:

Supervised learning is analogous to training a child to walk. You will hold the child's hand, show him how to take his foot forward, walk yourself for a demonstration and so on, until the child learns to walk on his own.

Regression:

In the case of supervised learning, you give concrete known examples to the computer. You say that for given feature value x1 the output is y1, for x2 it is y2, for x3 it is y3, and so on. Based on this data, you let the computer figure out an empirical relationship between x and y.

Once the machine is trained in this way with a sufficient number of data points, now you would ask the machine to predict Y for a given X. Assuming that you know the real value of Y for this given X, you will be able to deduce whether the machine's prediction is correct.

Thus, you will test whether the machine has learned by using the known test data. Once you are satisfied that the machine is able to do the predictions with a desired level of accuracy (say 80 to 90%) you can stop further training the machine.

Now, you can safely use the machine to do the predictions on unknown data points, or ask the machine to predict Y for a given X for which you do not know the real value of Y. This training comes under the regression that we talked about earlier.

Classification:

We may also use machine learning techniques for classification problems. In classification problems, you classify objects of similar nature into a single group. For example, in a set of 100 students say, you may like to group them into three groups based on their heights - short, medium and long. Measuring the height of each student, you will place them in a proper group.

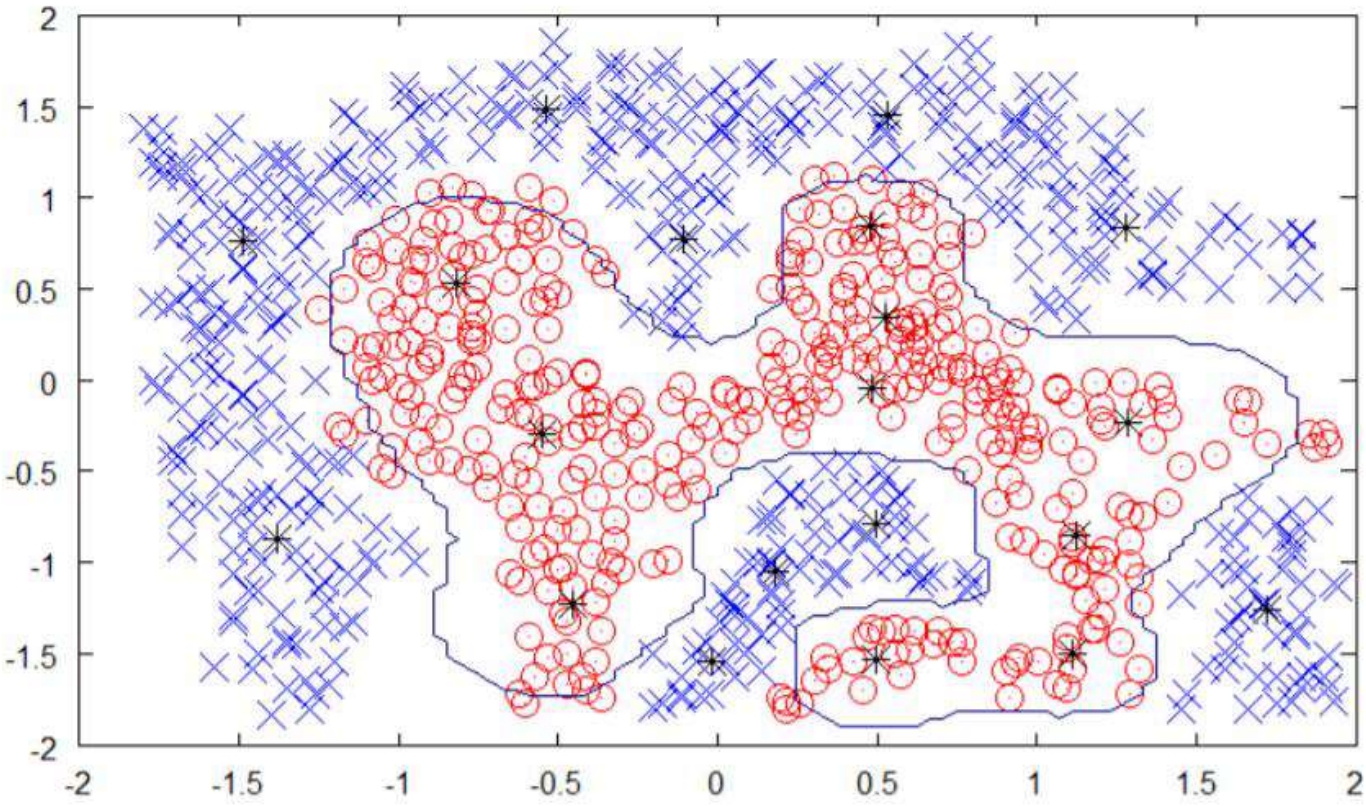
Now, when a new student comes in, you will put him in an appropriate group by measuring his height. By following the principles in regression training, you will train the machine to classify a student based on his feature – the height. When the machine learns how the groups are formed, it will be able to classify any unknown new student correctly. Once again, you would use the test data to verify that the machine has learned your technique of classification before putting the developed model in production.

Supervised Learning is where the AI really began its journey. This technique was applied successfully in several cases. You have used this model while doing the hand-written recognition on your machine. Several algorithms have been developed for supervised learning.

1.4 Unsupervised Learning:

In unsupervised learning, we do not specify a target variable to the machine, rather we ask machine "What can you tell me about X?". More specifically, we may ask questions such as given a huge data set X, "What are the five best groups we can make out of X?" or "What features occur together most frequently in X?". To arrive at the answers to such questions, you can understand that the number of data points that the machine would require to deduce a strategy would be very large. In case of supervised learning, the machine can be trained with even about few thousands of data points. However, in case of unsupervised learning, the number of data points that is reasonably accepted for learning starts in a few millions. These days, the data is generally abundantly available. The data ideally requires curating. However, the amount of data that is continuously flowing in a social area network, in most cases data curation is an impossible task.

The following figure shows the boundary between the yellow and red dots as determined by unsupervised machine learning. You can see it clearly that the machine would be able to determine the class of each of the black dots with a fairly good accuracy.



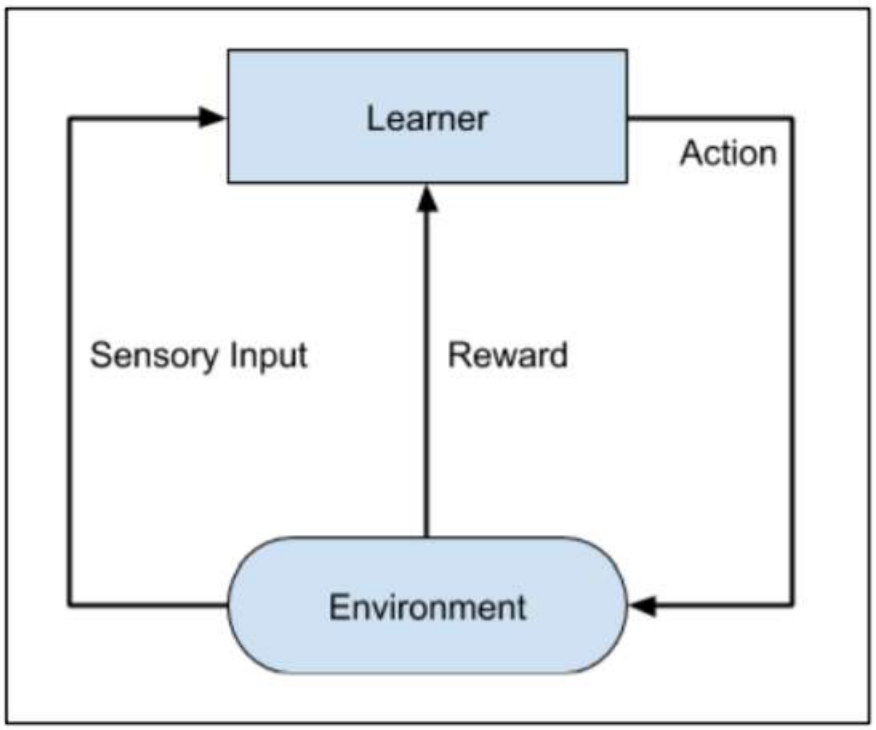
The unsupervised learning has shown a great success in many modern AI applications, such as face detection, object detection, and so on.

1.5 Reinforcement Learning:

Consider training a pet dog, we train our pet to bring a ball to us. We throw the ball at a certain distance and ask the dog to fetch it back to us. Every time the dog does this right, we reward the dog. Slowly, the dog learns that doing the job rightly gives him a reward and then the dog starts doing the job right way every time in future. Exactly, this concept is applied in "Reinforcement" type of learning. The technique was initially developed for machines to play games. The machine is given an algorithm to analyze all possible moves at each stage of the game. The machine may select one of the moves at random.

If the move is right, the machine is rewarded, otherwise it may be penalized. Slowly, the machine will start differentiating between right and wrong moves and after several iterations would learn to solve the game puzzle with a better accuracy. The accuracy of winning the game would improve as the machine plays more and more games.

The entire process may be depicted in the following diagram:



This technique of machine learning differs from the supervised learning in that you need not supply the labelled input/output pairs. The focus is on finding the balance between exploring the new solutions versus exploiting the learned solutions.

1.6 Deep Learning:

The deep learning is a model based on Artificial Neural Networks (ANN), more specifically Convolutional Neural Networks (CNN)s. There are several architectures used in deep learning such as deep neural networks, deep belief networks, recurrent neural networks, and convolutional neural networks.

These networks have been successfully applied in solving the problems of computer vision, speech recognition, natural language processing, bioinformatics, drug design, medical image analysis, and games. There are several other fields in which deep learning is proactively applied. The deep learning requires huge processing power and humongous data, which is generally easily available these days.

CHAPTER -2

Python Programming

2. Data types

Python has five standard Data Types:

- Numbers
- String
- List
- Tuple
- Dictionary

Python sets the variable type based on the value that is assigned to it. Unlike more riggers languages, Python will change the variable type if the variable value is set to another value. For example:

```
var = 123 # This will create a number integer assignment
var = 'john' # the `var` variable is now a string type.
```

2.1 Numbers

Python numbers variables are created by the standard Python method:

```
var = 382
```

Type	Format	Description
int	a = 10	Signed Integer
long	a = 345L	(L) Long integers, they can also be represented in octal and hexadecimal
float	a = 45.67	(.) Floating point real values
complex	a = 3.14J	(J) Contains integer in the range 0 to 255.

2.2 String

Most of the time Python will do variable conversion automatically. You can also use Python conversion functions (int(), long(), float(), complex()) to convert data from one type to another. In addition, the type function returns information about how your data is stored within a variable.

Create string variables by enclosing characters in quotes. Python uses single quotes ' double quotes " and triple quotes “”” to denote literal strings. Only the triple quoted strings “”” also will automatically continue across the end of line statement.

Strings can be accessed as a whole string, or a substring of the complete variable using brackets '[]'. Here are a couple examples:

Python can use a special syntax to format multiple strings and numbers. The string formatter is quickly covered here because it is seen often and it is important to recognize the syntax

```
print "The item {} is repeated {} times".format(element,count))
```

The {} are placeholders that are substituted by the variables element and count in the final string. This compact syntax is meant to keep the code more readable and compact.

Python is currently transitioning to the format syntax above, but python can use an older syntax, which is being phased out, but is still seen in some example code:

```
print "The item %i is repeated %i times"% (element,count)
```

```
firstName = 'john' lastName = "smith" message = """This is a string that will span across multiple lines. Using newline characters and no spaces for the next lines. The end of lines within this string also count as a newline when printed"""
```

```
var1 = 'Hello World!' var2 =
'RhinoPython'


print var1[0] # this will print the first character in the string an `H` print
var2[1:5] # this will print the substring 'hinoP`
```

2.3 List

Lists are a very useful variable type in Python. A list can contain a series of values. List variables are declared by using brackets [ ] following the variable name.

```
A = [ ] # This is a blank list variable
B = [1, 23, 45, 67] # this list creates an initial list of 4 numbers.
C = [2, 4, 'john'] # lists can contain different variable types.
```

All lists in Python are zero-based indexed. When referencing a member or the length of a list the number of list elements is always the number shown plus one.

```
mylist = ['Rhino', 'Grasshopper', 'Flamingo', 'Bongo']
B = len(mylist) # This will return the length of the list which is 3. The index is 0, 1, 2, 3.
print mylist[1] # This will return the value at index 1, which is 'Grasshopper' print mylist[0:2]
# This will return the first 3 elements in the list.
```

You can assign data to a specific element of the list using an index into the list. The list index starts at zero. Data can be assigned to the elements of an array as follows:

```
mylist = [0, 1, 2, 3] mylist[0]
= 'Rhino' mylist[1] =
'Grasshopper' mylist[2] =
'Flamingo' mylist[3] =
'Bongo' print mylist[1]
```

Lists aren't limited to a single dimension. Although most people can't comprehend more than three or four dimensions. You can declare multiple dimensions by separating an with commas. In the following example, the MyTable variable is a two-dimensional array :

```
MyTable = [[], []]
```

In a two-dimensional array, the first number is always the number of rows; the second number is the number of columns.

2.4 TUPLE

Tuples are a group of values like a list and are manipulated in similar ways. But, tuples are fixed in size once they are assigned. In Python the fixed size is considered immutable as compared to a list that is dynamic and mutable. Tuples are defined by parenthesis ().

```
myGroup = ('Rhino', 'Grasshopper', 'Flamingo', 'Bongo')
```

Here are some advantages of tuples over lists:

1. Elements to a tuple. Tuples have no append or extend method.
2. Elements cannot be removed from a tuple.
3. You can find elements in a tuple, since this doesn't change the tuple.
4. You can also use the in operator to check if an element exists in the tuple.
5. Tuples are faster than lists. If you're defining a constant set of values and all you're ever going to do with it is iterate through it, use a tuple instead of a list.
6. It makes your code safer if you "write-protect" data that does not need to be changed.

It seems tuples are very restrictive, so why are they useful? There are many data structures in Rhino that require a fixed set of values. For instance, a Rhino point is a list of 3 numbers [34.5, 45.7, 0]. If this is set

as tuple, then you can be assured the original 3 number structure stays as a point (34.5, 45.7, 0). There are other data structures such as lines, vectors, domains and other data in Rhino that also require a certain set of values that do not change. Tuples are great for this.

2.1.5 DICTIONARY

Dictionaries in Python are lists of Key: Value pairs. This is a very powerful datatype to hold a lot of related information that can be associated through keys. The main operation of a dictionary is to extract a value based on the key name. Unlike lists, where index numbers are used, dictionaries allow the use of a key to access its members. Dictionaries can also be used to sort, iterate and compare data.

Dictionaries are created by using braces ({}) with pairs separated by a comma (,) and the key values associated with a colon (:). In Dictionaries the Key must be unique. Here is a quick example on how dictionaries might be used:

```
room_num = {'john': 425, 'tom': 212} room_num['john'] = 645 # set the value associated with the
'john' key to 645 print (room_num['tom']) # print the value of the 'tom' key.
room_num['isaac'] = 345 # Add a new key 'isaac' with the associated value print (room_num.keys()) #
print out a list of keys in the dictionary print ('isaac' in room_num) # test to see if 'issac' is in the
dictionary. This returns true.
```

Dictionaries can be more complex to understand, but they are great to store data that is easy to access.

CHAPTER-3

Python Libraries

3.1 NUMPY

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

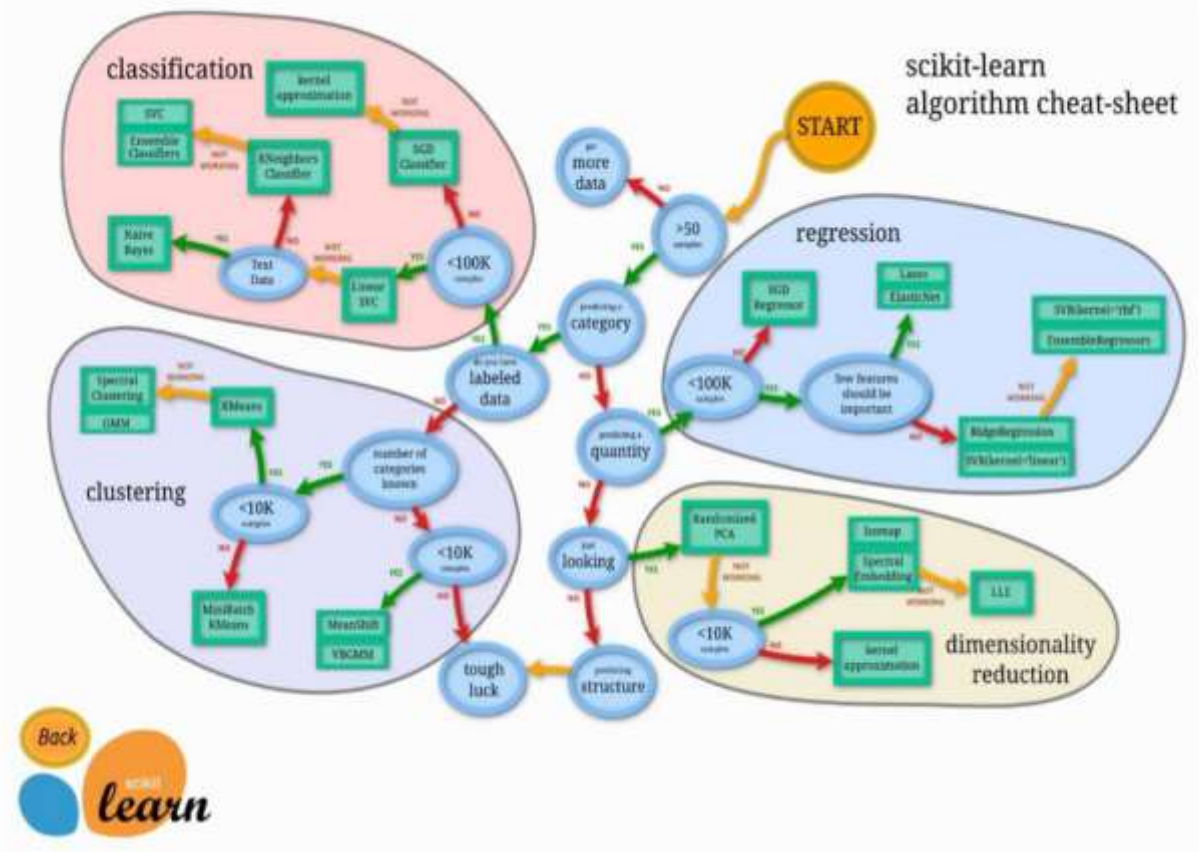
3.2 PANDAS

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem.
Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

3.3 MATPLOTLIB

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object-oriented interface or via a set of functions familiar to MATLAB users.

3.4 Sci-Kit Learn



This standard library has many pre-built classes of various Data-Science Algorithms which provide the ready-to-use implementation of these algorithms. The use of these algorithms is trivial and since these are well and field tested, you can safely use them in your AI applications. Most of these libraries are free to use even for commercial purposes.

3.5 FLASK

Flask stands out from other frameworks because it lets developers take the driver's seat and have full creative control of their applications. Maybe you have heard the phrase "fighting the framework" before. This happens with most frameworks when you decide to solve a problem with a solution that isn't the official one. It could be that you want to use a different database engine, or maybe a different method of authenticating users. Deviating from the path set by the framework's developers will give

Installing Python Packages with pip

Most Python packages are installed with the *pip* utility, which virtualenv automatically adds to all virtual environments upon creation. When a virtual environment is activated, the location of the pip utility is added to the PATH.
If you created the virtual environment with pyvenv under Python 3.3, then pip must be installed manually. Installation instructions are available on the pip website. Under Python 3.4, pyvenv installs pip automatically.

To install Flask into the virtual environment, use the following command:

```
>>>$ pip install flask
```

With this command, Flask and its dependencies are installed in the virtual environment.
You can verify that Flask was installed correctly by starting the Python interpreter and trying to import it:

```
>>> $ python
>>> import flask
>>>
```

If no errors appear, you can congratulate yourself: you are ready for the next chapter, where you will write your first web application.

Initialization

All Flask applications must create an *application instance*. The web server passes all requests it receives from clients to this object for handling, using a protocol called Web Server Gateway Interface (WSGI). The application instance is an object of class Flask,
usually created as follows:

```
from flask import Flask
app = Flask(__name__)
```

The only required argument to the Flask class constructor is the name of the main module or package of the application. For most applications, Python's __name__ variable is the correct value.

A Small Complete Application

In the previous sections, you learned about the different parts of a Flask web application, and now it is time to write one. The entire *hello.py* application script is nothing more than the three parts described earlier combined in a single file. The application is shown in Example 1.

Example 1. hello.py: A complete Flask application

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
return '<h1>Hello World!</h1>'
if __name__ == '__main__':
app.run(debug=True)
```

If you have cloned the application's Git repository on GitHub, you can now run git checkout 2a to check out this version of the application.
To run the application, make sure that the virtual environment you created earlier is activated and has Flask installed. Now open your web browser and type *http://127.0.0.1:5000/* in the address bar. Figure 1 shows the web browser after connecting to the application.

Rendering Templates

By default Flask looks for templates in a *templates* subfolder located inside the application folder. For the next version of *hello.py*, you need to store the templates defined earlier in a new *templates* folder as *index.html* and *user.html*.
The view functions in the application need to be modified to render these templates.
Example 2 shows these changes.
Example 2. hello.py: Rendering a template

```
from flask import Flask, render_template
@app.route('/index')
def index():
return render_template('index.html')
@app.route('/user/<name>')
def user(name):
return render_template('user.html', name=name)
```

The function render_template provided by Flask integrates the Jinja2 template engine with the application. This function takes the filename of the template as its first argument.
Any additional arguments are key/value pairs that represent actual values for variables referenced in the template. In this example, the second template is receiving a name variable.
Keyword arguments like name=name in the previous example are fairly common but may seem confusing and hard to understand if you are not used to them.

CHAPTER-4

Supervised Learning Algorithms:

Some of the widely used algorithms of supervised learning are as shown below:

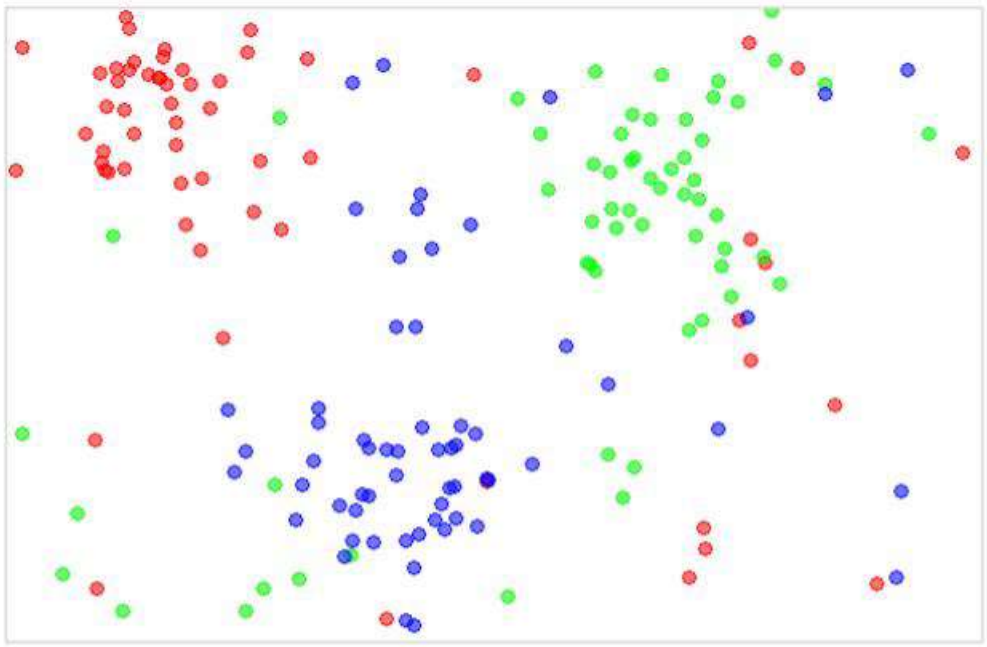
- k-Nearest Neighbours
- Decision Trees
- Naive Bayes
- Logistic Regression
- Support Vector Machines
- Linear/Multi-Linear Regression

4.1 k-Nearest Neighbours:

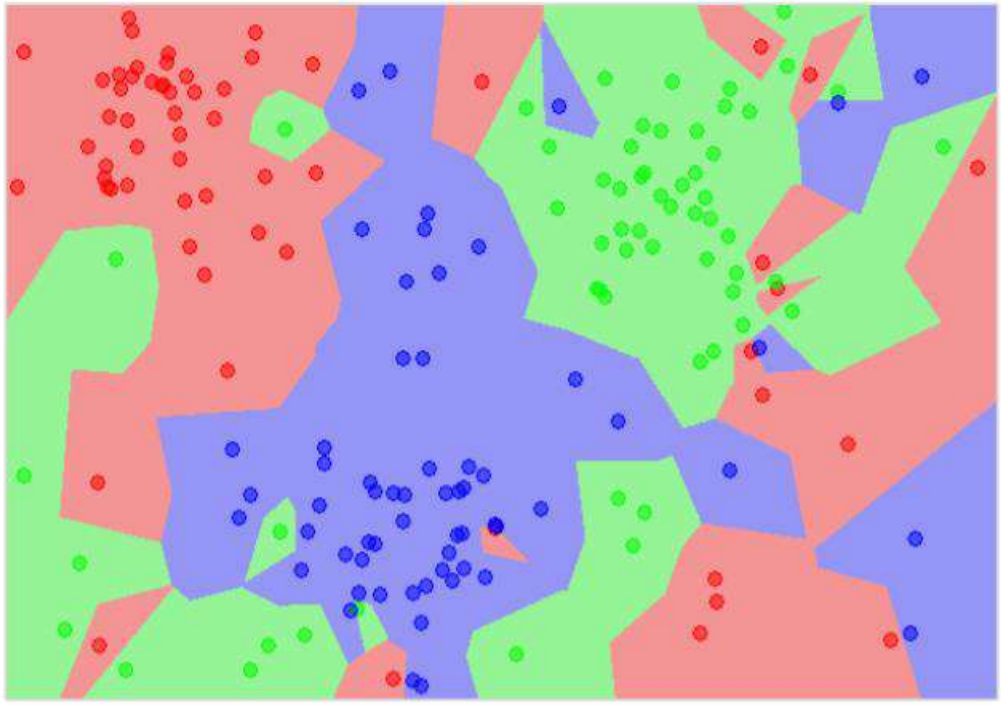
The k-Nearest Neighbours, which is simply called kNN is a statistical technique that can be used for solving for classification and regression problems.

Consider the distribution of objects as shown in the image given below:

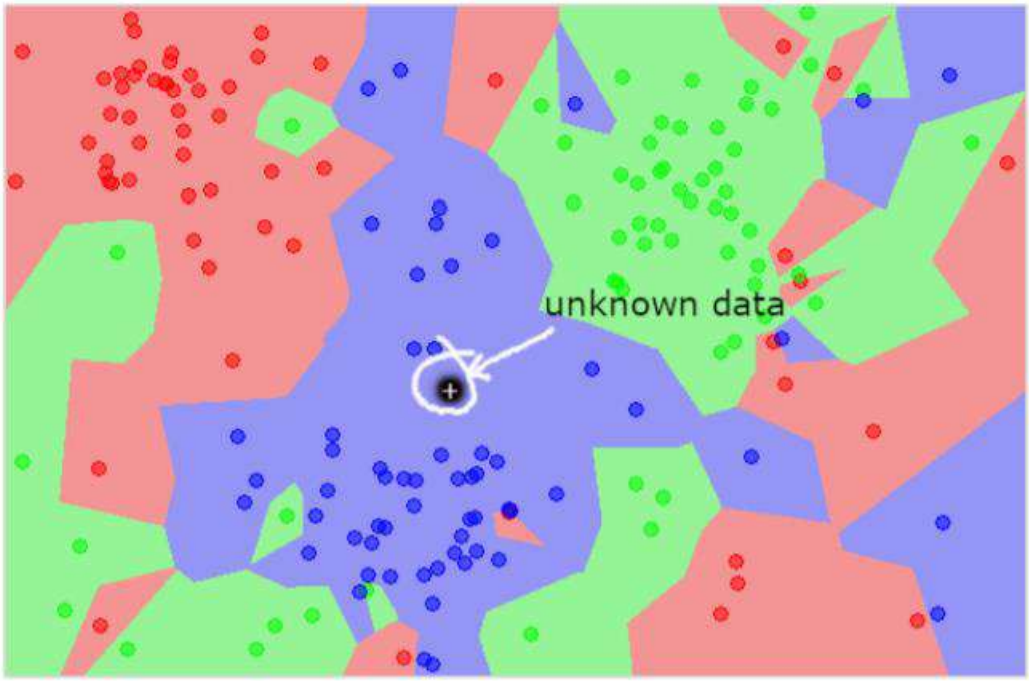
Source:



The diagram shows three types of objects, marked in red, blue and green colors.
When you run the KNN classifier on the above dataset, the boundaries for each type of object will be marked as shown below:



Now, consider a new unknown object that you want to classify as red, green or blue. This is depicted in the figure below.



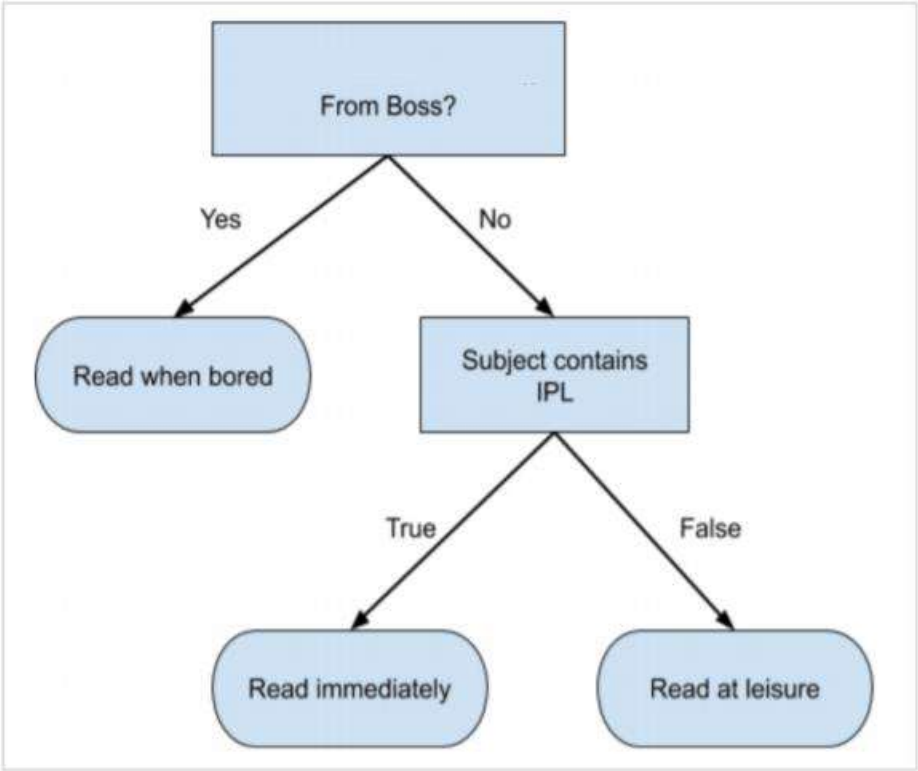
As you see it visually, the unknown data point belongs to a class of blue objects. Mathematically, this can be concluded by measuring the distance of this unknown point with every other point in the data set. When you do so, you will know that most of its neighbours are of blue color.
The average distance to red and green objects would be definitely more than the average distance to blue objects. Thus, this unknown object can be classified as belonging to blue class. The kNN algorithm can also be used for regression problems. The kNN algorithm is available as ready-to-use in most of the ML libraries like Sci-Kit Learn.

4.2 Decision Trees

A tree has many analogies in real life, and turns out that it has influenced a wide area of **machine learning**, covering both **classification and regression**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning

You would write a code to classify your input data based on this flowchart. The flowchart is self-explanatory and trivial. In this scenario, you are trying to classify an incoming email to decide when to read it.

In reality, the decision trees can be large and complex. There are several algorithms available to create and traverse these trees. As a Machine Learning enthusiast, you need to understand and master these techniques of creating and traversing decision trees.



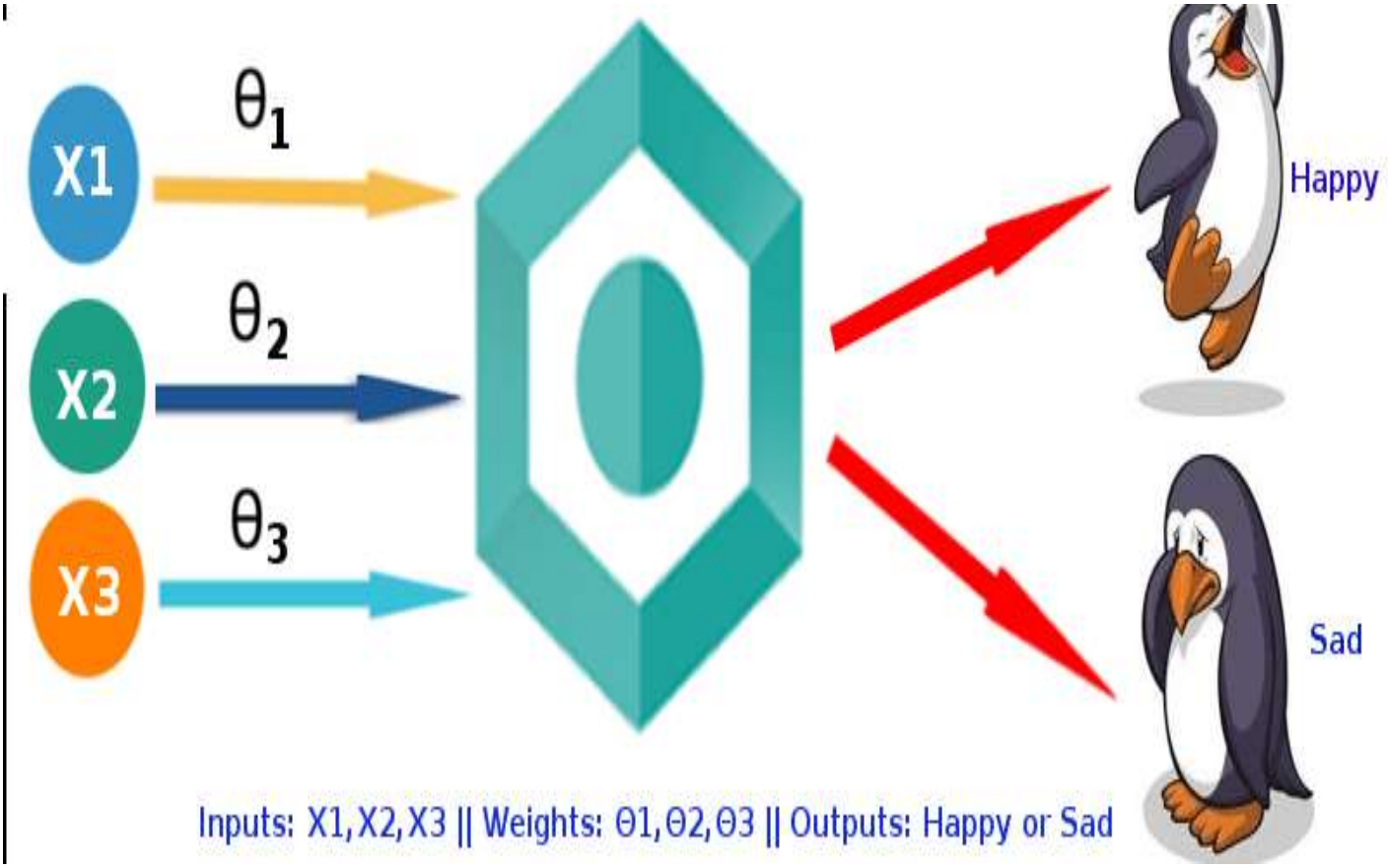
4.3 Naïve Bayes:

Naive Bayes is used for creating classifiers. Suppose you want to sort out (classify) fruits of different kinds from a fruit basket. You may use features such as color, size and shape of a fruit, for example, any fruit that is red in color, is round in shape and is about 10 cm in diameter may be considered as Apple.

So, to train the model, you would use these features and test the probability that a given feature matches the desired constraints. The probabilities of different features are then combined to arrive at a probability that a given fruit is an Apple. Naive Bayes generally requires a small number of training data for classification.

Also, it is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.

4.4 Logistic Regression:



Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example,

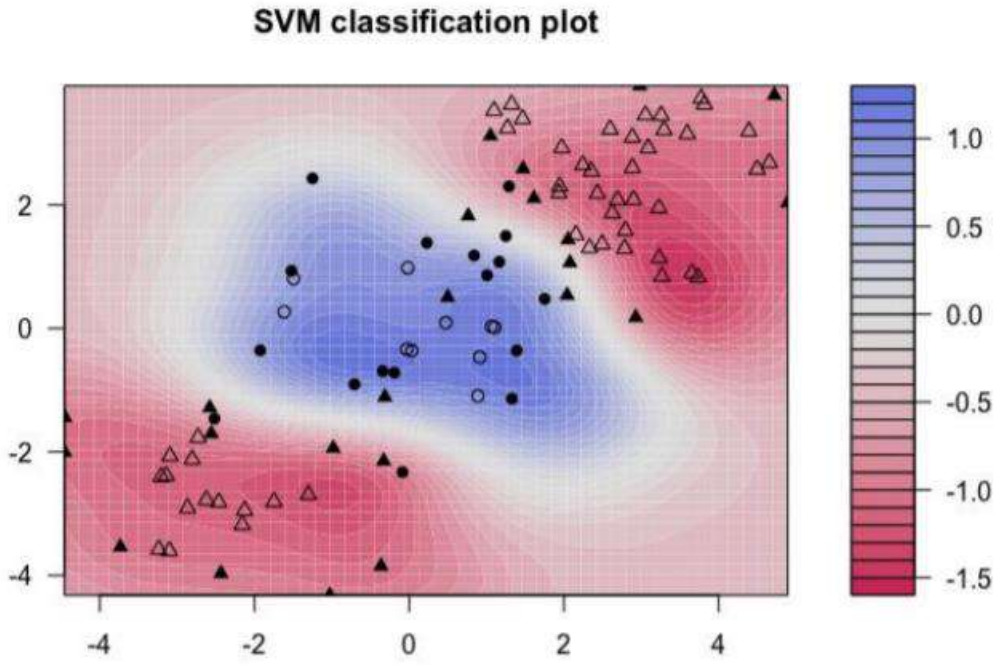
- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to1.

4.5 Support Vector Machines:

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. The distribution of data. Here the three classes of data cannot be linearly separated. The boundary curves are non-linear. In such a case, finding the equation of the curve becomes a complex job.



The Support Vector Machines (SVM) comes handy in determining the separation boundaries in such situations.

CHAPTER-5

PROJECT INTRODUCTION:

The use of intelligence systems in medical diagnosis is increasing gradually. There is no doubt that evaluation of data taken from patients and the decisions of experts are the most important factors in diagnosis. But, expert systems and different artificial intelligence techniques for classification also help experts in a great deal.

Classification systems, helping possible errors that can be done because of a fatigued or inexperienced expert to be minimized, provide medical data to be examined in a shorter time and more detailed. Problems with liver patients are not easily discovered in an early stage as it will be functioning normally even when it is partially damaged. Early diagnosis of liver problems will increase the patient's survival rate. Liver disease can be diagnosed by analyzing the levels of enzymes in the blood.

Furthermore, nowadays mobile devices are widely used for the supervision of humans' body conditions. In addition, automatic classification algorithms are demanded. According to their implements for liver diseases (almost definitely mobile capable or web capable), that decline the patient queue at the liver experts such as endocrinologists. Liver disorders are also an important disease in medicine. Levels of enzymes combined to blood are analyzed in Liver Disorders diagnosis. It can be a lot of possible errors in this diagnosis due to the number of enzymes to be many as well as the effects of different taken alcohol rates to vary from one patient to the other. Medical Data Company includes 345 specimens consisting of six fields and two classes. Each sample is taken from a single man. Two hundred of these samples are of one class with the remaining 145 are possessed by the other. First Five attributes of the collected data samples are the outcomes of blood tests while the last attribute includes daily alcohol consumption.

The attributes are as follows:

1. Total_Bilirubin,
2. Alkaline phosphatase,
3. Alamine_Aminotransferase,
4. Aspartate_Aminotransferase,
5. Total_Protiens
6. Albumin
7. Albumin_and_Globulin_Ratio

5.1 PROBLEM STATEMENT:

The proposed system is built expert system which employs for the diagnosis of LDs is developed in an environment Neuroph and Crystal's reports were used for analysis. Approach for analyzing clusters to identify a meaningful pattern for determining whether a patient suffers from LD or not is presented. The system provides a guide for the diagnosis of LDs within the decision-making framework. The process for the medical diagnosis of LD starts when an individual consults a physician (doctor) and presents a set of complaints (symptoms). The physician then requests further information that will further aid in the proper diagnosis of the disease.
The doctor is likely to start with a health history and a thorough physical examination.

- **Blood tests.** A group of blood tests called liver function tests can be used to diagnose liver disease. Other blood tests can be done to look for specific liver problems or genetic conditions.
- **Imaging tests.** An ultrasound, CT scan and MRI can show liver damage.
- **Tissue analysis.** Removing a tissue sample (biopsy) from your liver may help diagnose liver disease and look for signs of liver damage. A liver biopsy is most often done using a long needle inserted through the skin to extract a tissue sample. It is then analyzed in a laboratory.

Strategy

This seems to be a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that , when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training is included in the Algorithms and Techniques' sub-section of the 'Analysis' part.

Metrics

In problems of disease classification like this one, simply comparing the accuracy, that is, the ratio of correct predictions to total predictions is not enough. This is because depending on the context like severity of disease, sometimes it is more important that an algorithm does not wrongly predict a disease as a non-disease, while predicting a healthy person as diseased will attract a comparatively less severe penalty.
Thus, here we will use **F-beta score** as a performance metric, which is basically the weighted harmonic mean of precision and recall. Precision and Recall are defined as:
Precision=TP/ (TP+FP), Recall=TP/ (TP+FN), where
TP=True Positive
FP=False Positive
FN=False Negative
In the same vein, F-beta score is:
F-beta score = $(1+\beta 2)$*precision*recall/((β 2*precision)+recall)
β = A number that decides relative weightage of precision and recall. In this case, a disease being classified as a non-disease will incur a high penalty. So, more emphasis is placed on recall.

5.2 DATA COLLECTION:

The initial step is gathering the informational index.
Datasets can be collected from UCI store where some benchmark datasets are accessible the imperative highlights in the dataset are identified by utilizing some least complex strategies, for example, previously or from specialists given data. The datasets contains boisterous information and highlight esteems, in this manner preprocessing methods are basic.

DATA MODELLING:

TASK 1: Convert data in excel file or CSV format

At first convert data from text to column in excel sheet before importing the data. Because, unstructured data were found in the dataset. Then, save the data in csv format.

TASK 2: Import data

Import data to data in csv format. No clean-up is required as the data has already been cleaned previously.

wine quality <- read.csv(file. Choose(), header=T, sep=",", check. names=TRUE)

TASK 3: Check characteristics and data relationship

- **Check data characteristics with following codes -**

head(liver disease analyser)
str(liver disease analyser)
summary(liver disease analyser)
class(liver disease analyser)
n row(liver disease analyser)
n col(liver disease analyser)

MODEL ACCURACY:

```
In [11]: from sklearn.metrics import r2_score
         accuracy=r2_score(y_test, y_pred)

In [12]: accuracy

Out[12]: 0.99995922724151454
```

This seems to be a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that , when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training is included in the 'Algorithms and Techniques' sub-section of the 'Analysis' part.

5.3 Data Table

	A	B	C	D	E	F	G	H	I	J	K
1	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
2	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	1
3	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
4	62	Male	7.3	4.1	490	60	68	7	3.3	0.89	1
5	58	Male	1	0.4	182	14	20	6.8	3.4	1	1
6	72	Male	3.9	2	195	27	59	7.3	2.4	0.4	1
7	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	1
8	26	Female	0.9	0.2	154	16	12	7	3.5	1	1
9	29	Female	0.9	0.3	202	14	11	6.7	3.6	1.1	1
10	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	2
11	55	Male	0.7	0.2	290	53	58	6.8	3.4	1	1
12	57	Male	0.6	0.1	210	51	59	5.9	2.7	0.8	1
13	72	Male	2.7	1.3	260	31	56	7.4	3	0.6	1
14	64	Male	0.9	0.3	310	61	58	7	3.4	0.9	2
15	74	Female	1.1	0.4	214	22	30	8.1	4.1	1	1
16	61	Male	0.7	0.2	145	53	41	5.8	2.7	0.87	1
17	25	Male	0.6	0.1	183	91	53	5.5	2.3	0.7	2
18	38	Male	1.8	0.8	342	168	441	7.6	4.4	1.3	1
19	33	Male	1.6	0.5	165	15	23	7.3	3.5	0.92	2
20	40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1

Context

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors.

Content

This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

Columns:

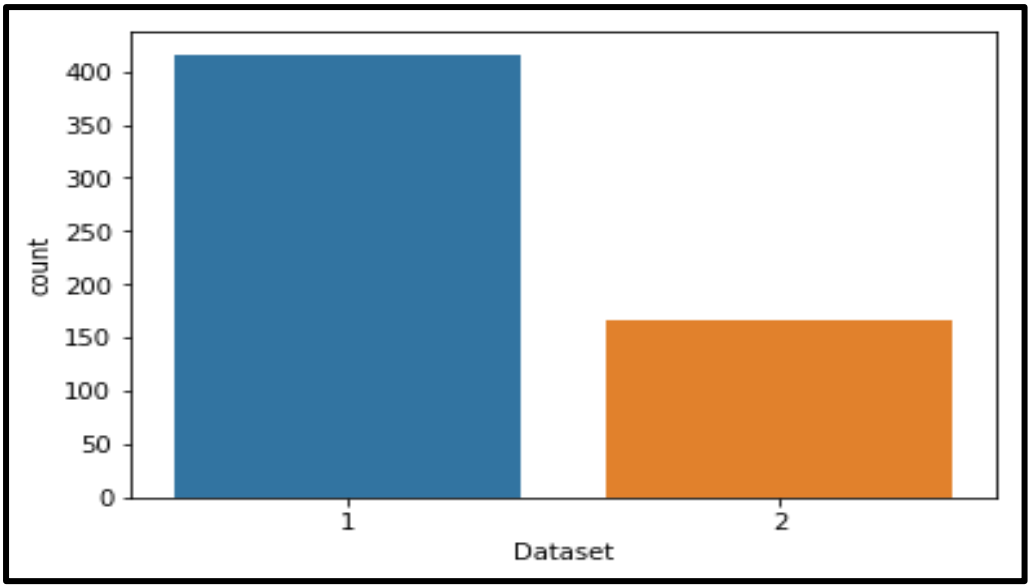
- Age of the patient
- Gender of the patient
- Total Bilirubin
- Direct Bilirubin
- Alkaline Phosphotase
- Alamine Aminotransferase
- Aspartate Aminotransferase
- Total Protiens
- Albumin
- Albumin and Globulin Ratio
- Dataset: field used to split the data into two sets (patient with liver disease, or no disease)

CHAPTER-6

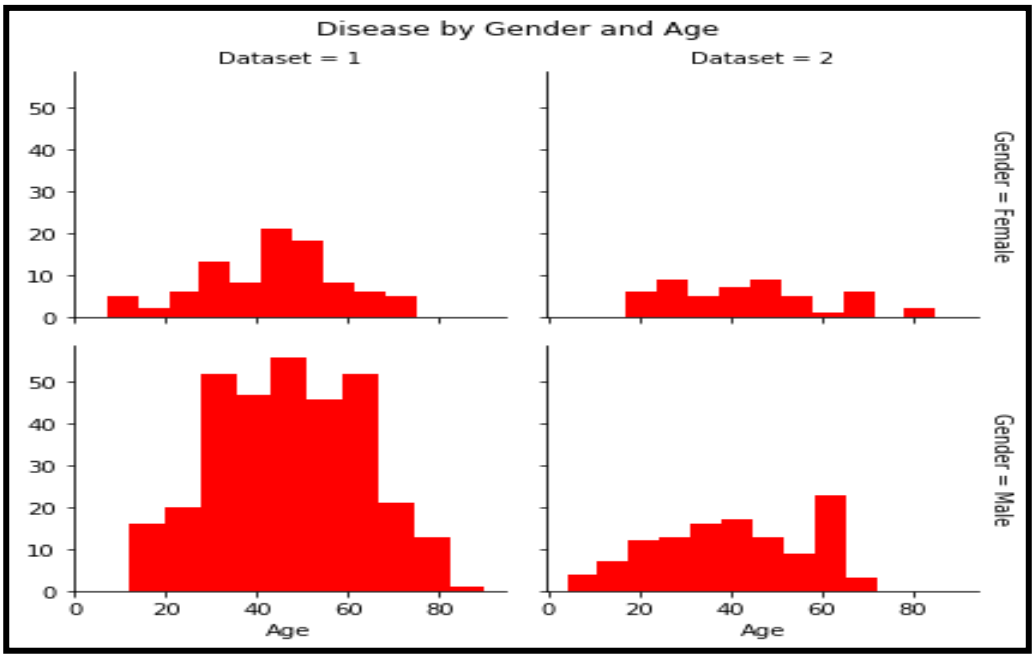
Data Analysis and Algorithm Accuracy

COUNT PLOT OF LIVER PATIENTS DIAGNOISED

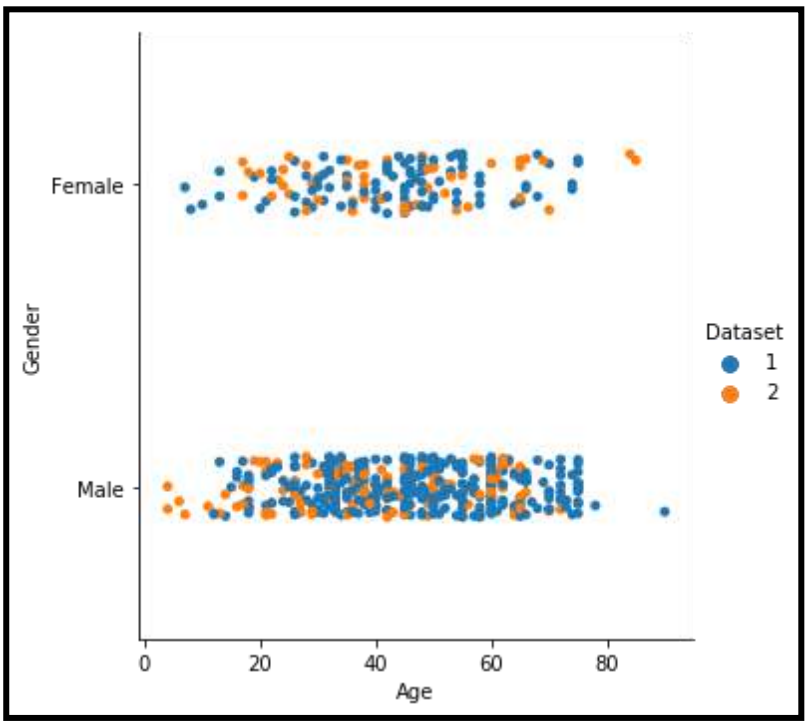
Number of patients diagnosed with liver disease: 416
Number of patients not diagnosed with liver disease: 167



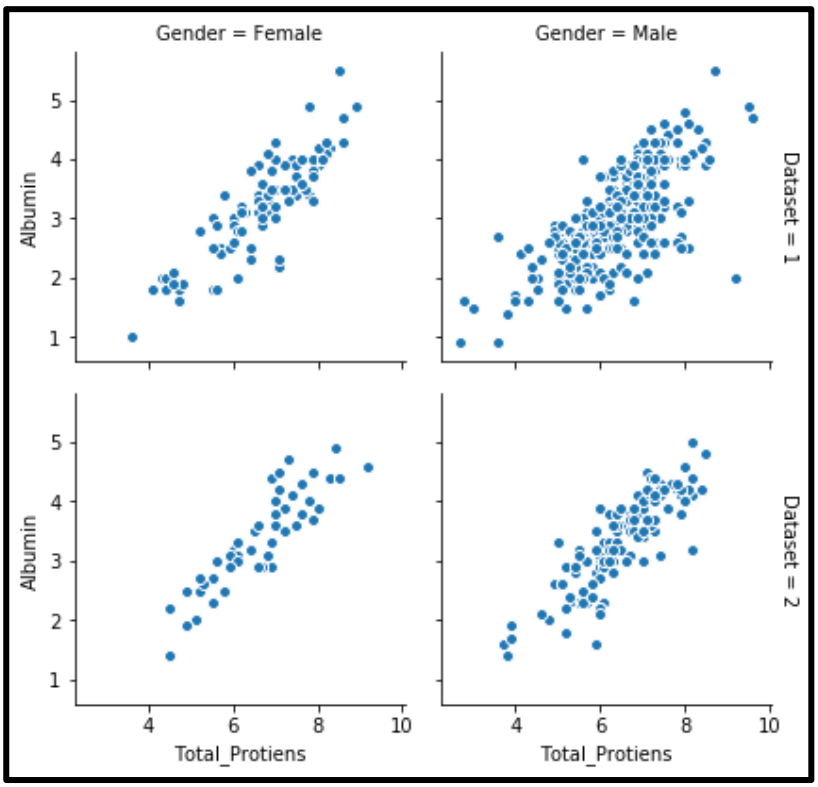
FACETGRID ON DISEASE BY GENDER AND AGE



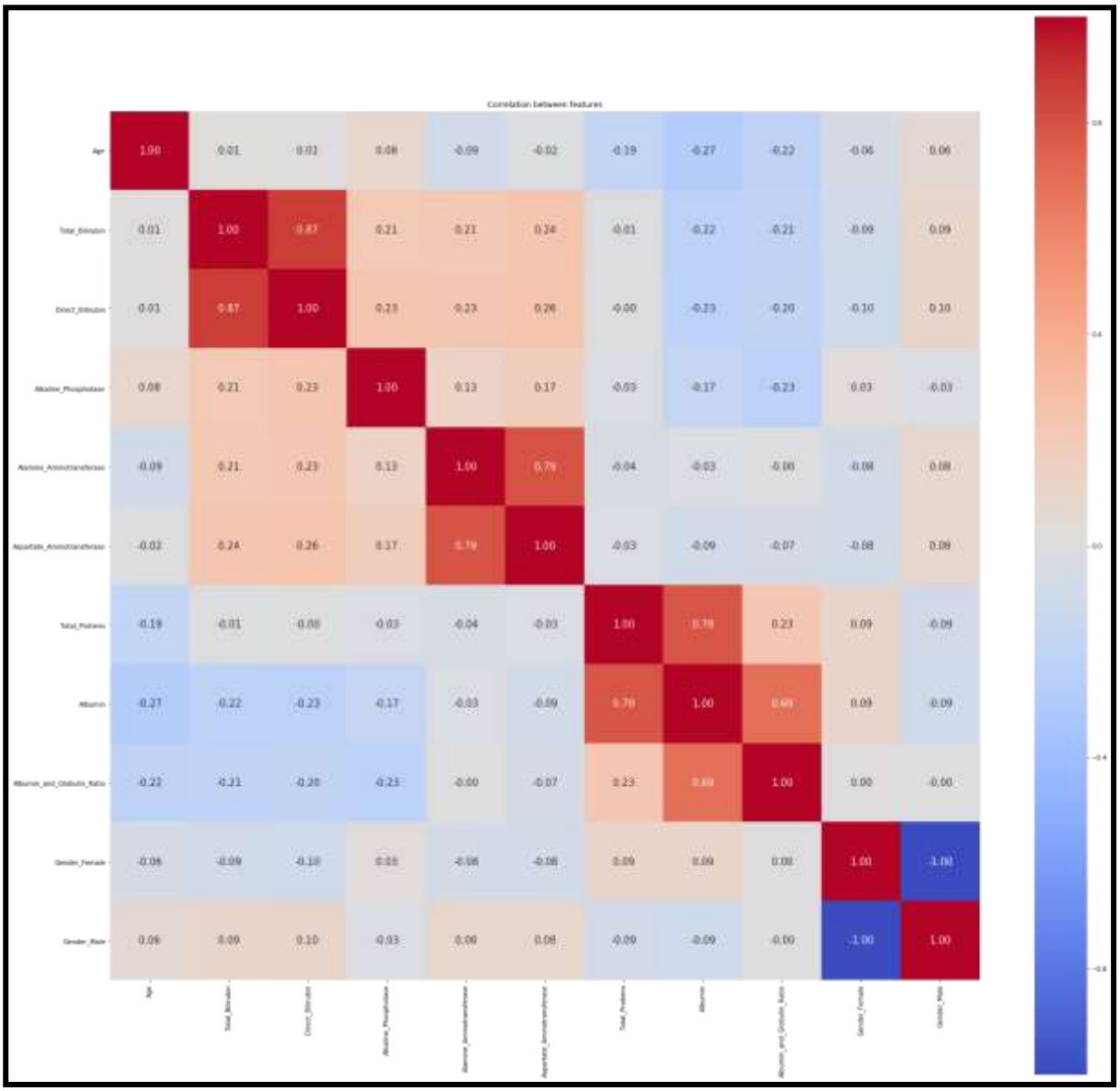
CATPLOT BASED ON AGE, GENDER & DATASET



FACETGRID ON DIRECT_BILIRUBIN &TOTAL_BILIRUBIN



CO-RELATION GRAPGH



Algorithms and Techniques

Three supervised learning approaches are selected for this problem. Care is taken that all these approaches are fundamentally different from each other, so that we can cover as wide an umbrella as possible in term of possible approaches. For example- We will not select Random Forest and Ada Boost together as they come from the same family of 'ensemble' approaches:

For each algorithm, we will try out different values of a few hyperparameters to arrive at the best possible classifier. This will be carried out with the help of grid search cross validation technique. The algorithms are described below:

1. Random Forest Classifier:

- n_estimators(number of trees in a forest)
- max_depth(maximum depth of one single tree)
- max_features(decides how many features are to be used)
- oob_score(decides whether to include out-of-bag or prediction error)

Accuracy scored: 0.68

2. Gaussian Naive Bayes Classifier

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. The representation for naive Bayes is probabilities. A list of probabilities are stored to file for a learned naive Bayes model. This includes:

Class Probabilities: The probabilities of each class in the training dataset.
Conditional Probabilities: The conditional probabilities of each input value given each class value.

Accuracy scored: 0.5613

3. Logistic Regression:

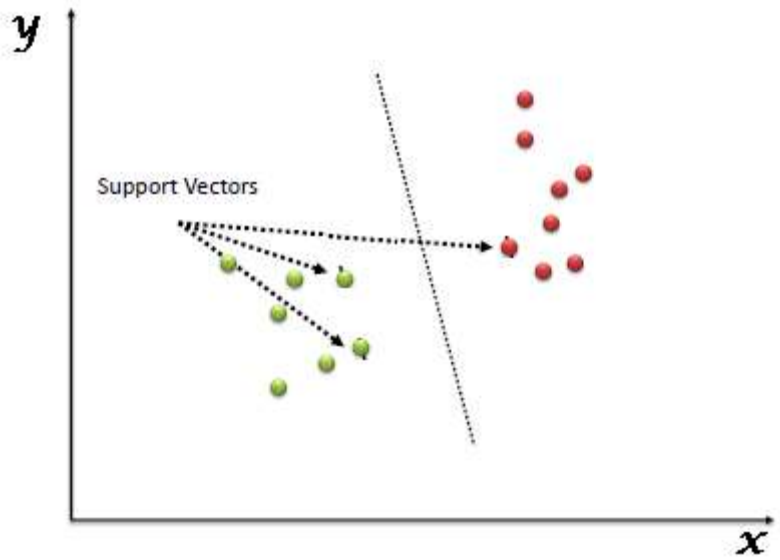
Since the outcome is binary and we have a reasonable number of examples at our disposal compared to number of features, this approach seems suitable. At the core of this method is a logistic or sigmoid function that quantifies the difference between each prediction and its corresponding true value. When presented with a number of inputs, it assigns different weights to features (based on their relative importance).

Since for this data it already knows the output beforehand, it continuously adjusts the weights such that when these weights summed up with their features are introduced in the logistic function, the results are as near as possible to the actual ones. Once presented with a test value, it again inserts the value into our logistic function and returns the output as a number between 0 and 1, which represents the probability of that test value being in a particular class.

Accuracy scored: 0.7143

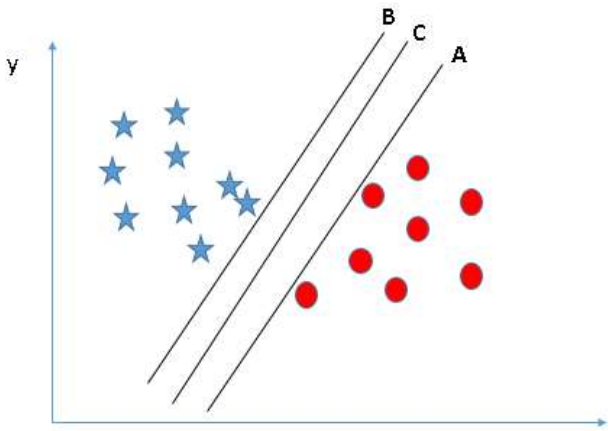
4. Support Vector Machine:

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

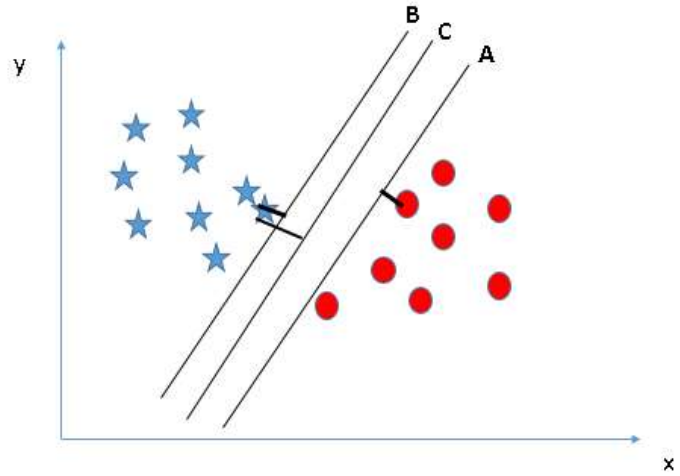


Let's understand:

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle. You need to remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.
- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

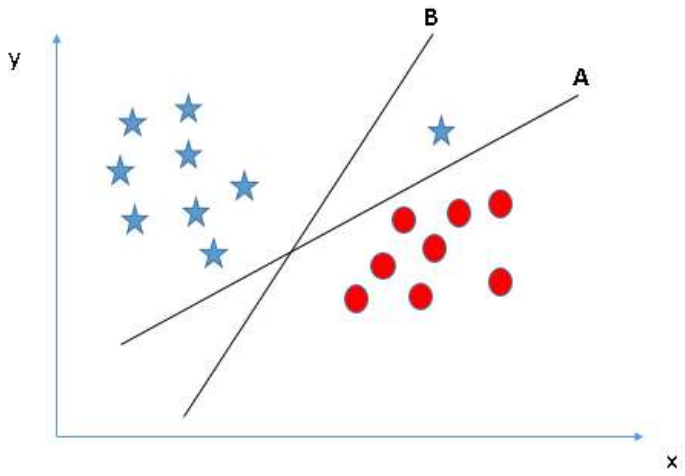


× Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:



Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

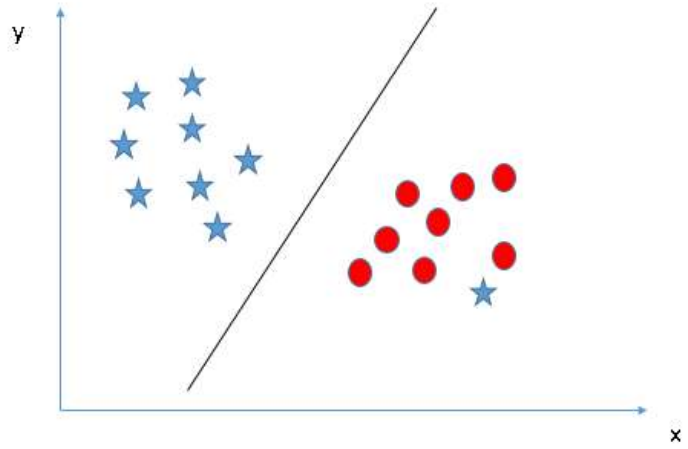
- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane



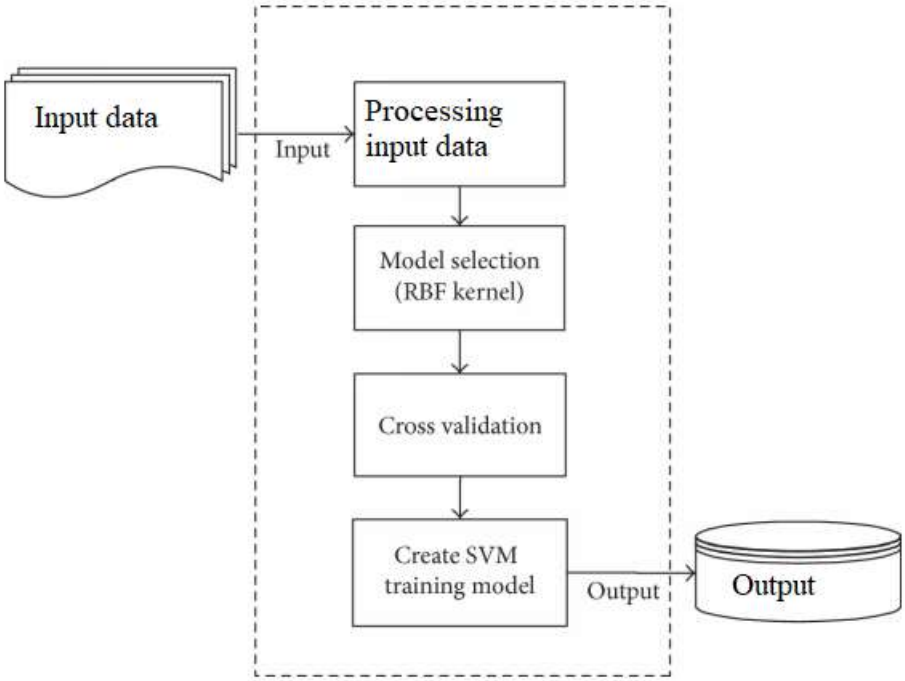
Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.

- As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



CHAPTER-7

FRONTEND:

7.1HTML

Okay, so this is the only bit of mandatory theory. In order to begin to write HTML, it helps if you know what you are writing.

HTML is the **language in which most websites are written**. HTML is used to create pages and make them functional.

The code used to make them visually appealing is known as CSS and we shall focus on this in a later tutorial. For now, we will focus on **teaching you how to build rather than design**.

The History of HTML

HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version is known as HTML5.

A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes**.

What are Tags and Attributes?

Tags and attributes are the basis of HTML.

They work together but perform different functions – it is worth investing 2 minutes in **differentiating the two**.

What Are HTML Tags?

Tags are used to **mark up the start of an HTML element** and they are usually enclosed in angle brackets. An example of a tag is: `<h1>`.

Most tags must be opened `<h1>` and closed `</h1>` in order to function.

What are HTML Attributes?

Attributes contain **additional pieces of information**. Attributes take the form of an opening tag and additional info is **placed inside**.

An example of an attribute is:

```
<img src="mydog.jpg" alt="A photo of my dog.">
```

In this instance, the image source (src) and the alt text (alt) are attributes of the `<img>` tag.

There are several common attributes that may appear in many elements :

- The `id` attribute provides a document-wide unique identifier for an element. This is used to identify the element so that stylesheets can alter its presentational properties, and scripts may alter, animate or delete its contents or presentation. Appended to the URL of the page, it provides a globally unique identifier for the element, typically a sub-section of the page. For example, the ID "Attributes" in `https://en.wikipedia.org/wiki/HTML#Attributes`.
- The `class` attribute provides a way of classifying similar elements. This can be used for semantic or presentation purposes. For example, an HTML document might semantically use the designation `<class="notation">` to indicate that all elements with this class value are subordinate to the

main text of the document. In presentation, such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source.

7.2 CSS

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the WWW, alongside HTML and JavaScript

CSS is designed to enable the separation of presentation and content, including layout, color, nad fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice. CSS also has rules for alternate formatting if the content is accessed on a mobile devices.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

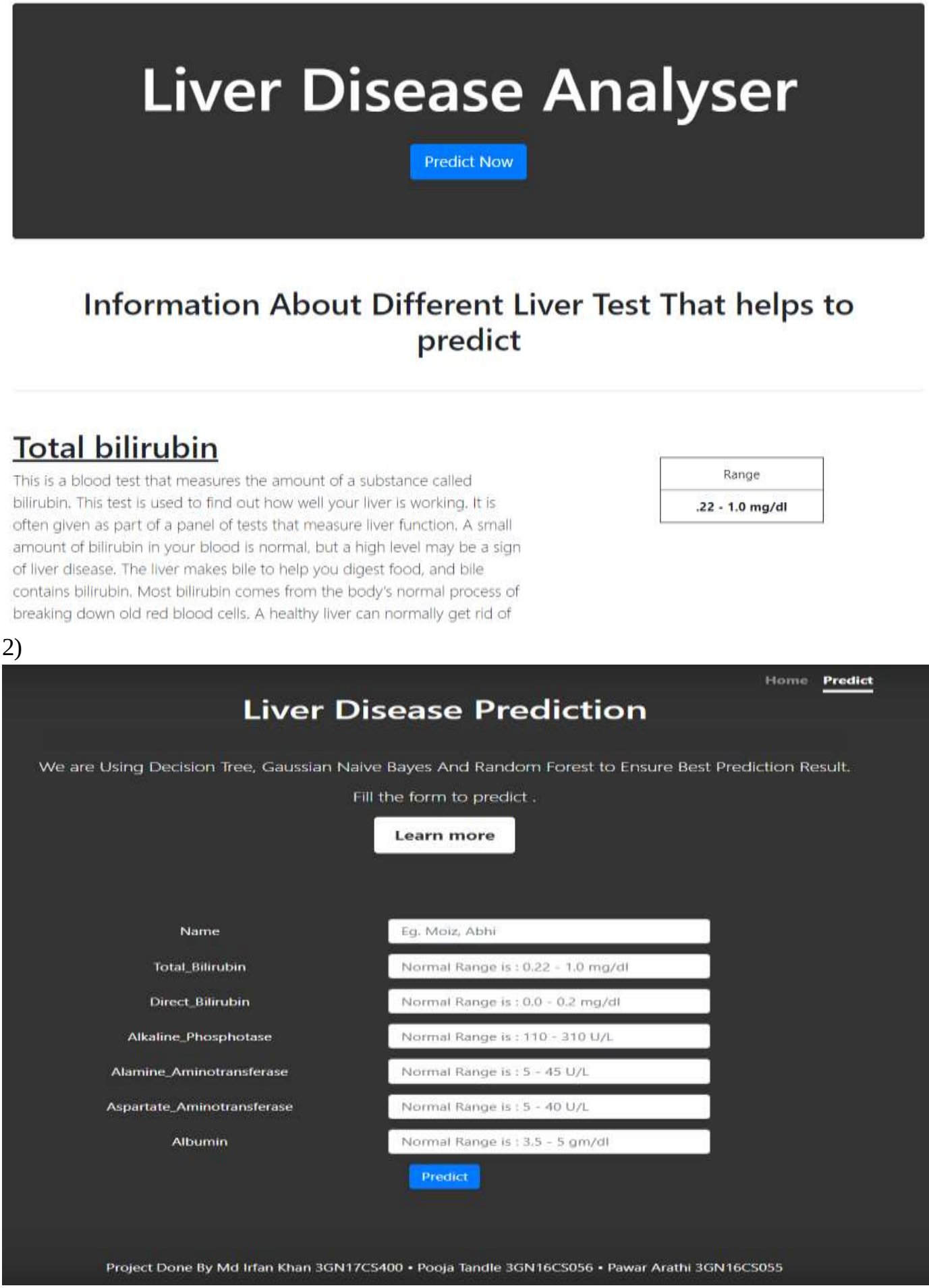
7.3 JavaScript

Alongside HTML and CSS, JavaScript is one of the core technologies of the WWW.[8] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.


Liver Disease Analyser
Predict Now
Information About Different Liver Test That helps to predict
Total bilirubin
This is a blood test that measures the amount of a substance called bilirubin. This test is used to find out how well your liver is working. It is often given as part of a panel of tests that measure liver function. A small amount of bilirubin in your blood is normal, but a high level may be a sign of liver disease. The liver makes bile to help you digest food, and bile contains bilirubin. Most bilirubin comes from the body's normal process of breaking down old red blood cells. A healthy liver can normally get rid of
Range
.22 - 1.0 mg/dl

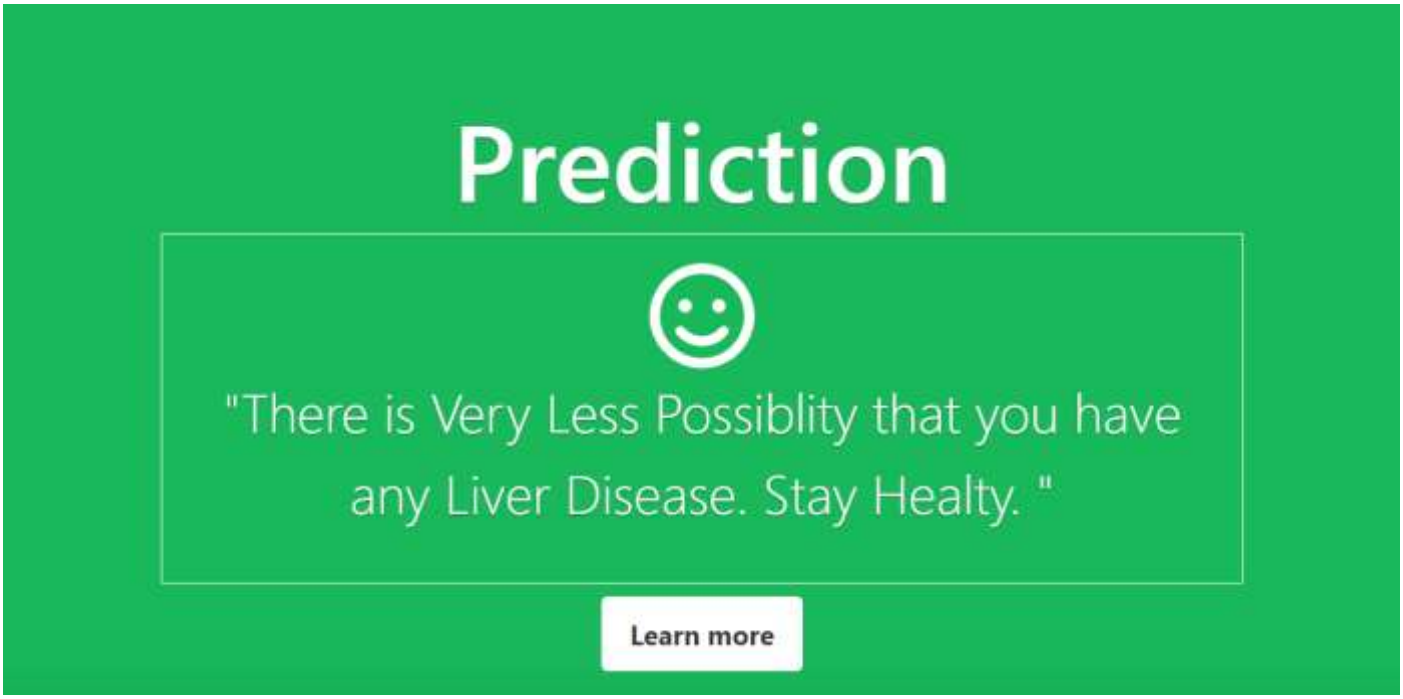
2)

Home
Predict
Liver Disease Prediction
We are Using Decision Tree, Gaussian Naive Bayes And Random Forest to Ensure Best Prediction Result.
Fill the form to predict .
Learn more
Name
Eg. Moiz, Abhi
Total_Bilirubin
Normal Range is : 0.22 - 1.0 mg/dl
Direct_Bilirubin
Normal Range is : 0.0 - 0.2 mg/dl
Alkaline_Phosphotase
Normal Range is : 110 - 310 U/L
Alamine_Aminotransferase
Normal Range is : 5 - 45 U/L
Aspartate_Aminotransferase
Normal Range is : 5 - 40 U/L
Albumin
Normal Range is : 3.5 - 5 gm/dl
Predict
Project Done By Md Irfan Khan 3GN17CS400 • Pooja Tandle 3GN16CS056 • Pawar Arathi 3GN16CS055

CHAPTER-8

OUTPUT:

Test is Negative:



Test is Positive:

CHAPTER-9

CONCLUSION

Initially, the dataset was explored and made ready to be fed into the classifiers. This was achieved by removing some rows containing null values, transforming some columns which were showing skewness and using appropriate methods (one-hot encoding) to convert the labels so that they can be useful for classification purposes. Performance metrics on which the models would be evaluated were decided. The dataset was then split into a training and testing set.

Firstly, a naive predictor and a benchmark model ('Logistic Regression') were run on the dataset to determine the benchmark value of accuracy. The greatest difficulty in the execution of this project was faced in two areas- determining the algorithms for training and choosing proper parameters for fine-tuning. Initially, I found it very vexing to decide upon 3 or 4 techniques out of the numerous options available in sklearn.

This exercise made me realize that parameter tuning is not only a very interesting but also a very important part of machine learning. I think this area can warrant further improvement, if we are willing to invest a greater amount of time as well as computing power.

This paper gives a thought of late machine learning calculations accessible for discovery and determination of liver illness. From the examine it very well may be unmistakably seen that diverse managed learning calculations K-Nearest Neighbor and Support Vector Machine give improved exactness on discovery of liver maladies

CHAPTER-9

REFERENCES

1. Nazmun Nahar and Ferdous Ara, "Liver disease prediction by using different decision tree techniques", International Journal of Data Mining & Knowledge Management Process (IJDKP),Vol.8, No.2, March 2018.

2. Divya, B, Kalaiselvi, R, Review on Confidentiality of the Outsourced Data, Research Journal of Science and Engineering Systems, vol.1, pp.1-7, 2017.

3. Hassoon, M, Kouhi, M S, Zomorodi-Moghadam, M, & Abdar, M, Rule optimization of boosted c5. 0 classification using genetic algorithm for liver diseaseprediction. In 2017 International Conference on Computer and Applications (ICCA), pp. 299-305, IEEE, 2017

4. M. Hassoon, M. S. Kouhi, M. Zomorodi-Moghadam and M. Abdar, "Rule Optimization of Boosted C5.0 Classification Using Genetic Algorithm for Liver disease Prediction," *2017 International Conference on Computer and Applications (ICCA)*, Doha, 2017, pp. 299-305. doi: 10.1109/COMAPP.2017.8079783

5. D.A. Saleh F. Shebl M. Abdel-Hamid et al. "Incidence and risk factors for hepatitis C infection in a cohort of women in rural Egypt"Trans. R. Soc. Trop. Med. Hyg. vol. 102 pp. 921928 2008. https://doi.org/10.1016/j.trstmh.2008.04.011

6. Performance analysis of classification algorithms on early detetion of Liver disease "Moloud Abdar, Mariam Zomorodi- Moghadam, Resul Das, I-Hsien Ting, doi: 10.1016/ j.eswa. 2016.08.065

7. A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis " Bendi Venkata Ramana1, Prof. M.Surendra Prasad Babu2, Prof. N. B. Venkateswarlu3 -(IJDMS), Vol.3, No.2, May 2012

8. S. B. Kotsiantis, Supervised Machine Learning: A Review of Clas-sification Techniques, Informatica (2007) 249-268 249.

9. A.S.Aneeshkumar and C.Jothi Venkateswaran, "Estimating the Surveillance of Liver Disorder using Classification Algorithms", International Journal of Computer Applications (095-8887), Volume 57-No.6, November 2012

10. T. Mitchell, Machine Learning. McGrawHill ,1997