THYROID DISEASE PREDICTION

A MINI PROJECT REPORT

Submitted by

ARVINDHKUMAR. K - 210420243008

AATHARSH. B - 210420243010

SHAQEEM SAMI MURTUZA. S - 210420243049

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CHENNAI INSTITUTE OF TECHNOLOGY,

PUDUPER, KANCHEEPURAM

ANNA UNIVERSITY:: CHENNAI 600 025

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “ **THYROID DISEASE PREDICTION** ” is the bonafide workof **“K. ARVINDH KUMAR, AATHARSH.B, SHAQEEM SAMI MURTUZA.S**” who carried out the project work under my supervision.

SIGNATURE

Dr. M. GEETHA, Ph.D.,

HEAD OF THE DEPARTMENT

Professor
Dept. of Artificial Intelligence and
Data Science Engineering
Chennai Institute of Technology
Kundrathur,Chennai-600069

SIGNATURE

Dr. M. GEETHA, Ph.D.,

HEAD OF THE DEPARTMENT

Professor
Dept. of Artificial Intelligence and
Data Science Engineering
Chennai Institute of Technology
Kundrathur,Chennai-600069

Submitted for the **ANNA UNIVERSITY** examination held on _______________ at Chennai Institute of Technology, Kundrathur**.**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I convey my profound thanks and gratitude to our honorable **Chairman Mr.P.Sriram**, **Chennai Institute of technology,** Chennai for providing me an excellent academic climate, which made this endeavor possible.

I also express my gratitude to our beloved **Principal Dr.A. Ramesh, Ph.D** who constantly nurtured our of education and devote their precious time for our needs.

I am deeply indebted to pay my sincere thanks to our respectable **Head of the Department Dr.M.GEETHA, Ph.D.,** Department of Artificial Intelligence and Data Science for showing some exuberant consent for my project and providing me with all the facilities in the department to complete the project.

I sincerely thank our project Guide, **Dr M.GEETHA, Ph.D.,** Department of Artificial Intelligence and Data Science for having provided us valuable guidance, resources and timely suggestions throughout our work.

I thank all my department teaching and non teaching staff, who have helped directly and indirectly to complete this project in time.

Last but not least I extend my deep gratitude to my beloved family members for their coordination, encouragement and financial support to carry out this project.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Thyroid illness expectation has arisen as a significant undertaking as of late. Regardless of existing methodologies for its determination, frequently the objective is double order, the utilized datasets are little measured and results are not approved by the same token. Overwhelmingly, existing methodologies center around model enhancement and the component designing part is less explored. To beat these restrictions, this study presents amethodology that researches highlight designing for AI and profound learning models. Forward include decision, in reverse element disposal, bidirectional component end, and AI based highlight determination utilizing additional tree classifiers are taken on. The proposed approach can anticipate Hashimoto's thyroiditis (essential hypothyroid), restricting protein (expanded restricting protein), immune system thyroiditis (repaid hypothyroid), and non-thyroidal condition (NTIS) (simultaneous non-thyroidal disease). Broad investigations show that the additional tree classifier-based chosen highlight yields the best outcomes with 0.99 precision and a F1 score when utilized with the arbitrary timberland classifier. Results propose that the AI models are a superior decision for thyroid illness recognition with respect to the gave exactness and the computational intricacy. K-overlap cross-approval and execution examination with existing investigations substantiate the predominant exhibition of the proposed approach.

CHAPTER – 1

INTRODUCTION

Thyroid Disease (TD) has become one of the most common endocrine disorders worldwide. At least a person out of ten is suffered from thyroid disease in India. For predicting Thyroid disease analyzing blood report is required to analyze and predict disease. Thyroid blood test data set analysis will be conducted using various supervised machine learning classifier techniques. Based on the accuracy of different algorithm, best accuracy algorithm will be chosen to fetch the result. Thyroid disease is a medical condition that affects the function of the thyroid gland. The thyroid organ is situated at the front of the neck and creates thyroid chemicals that movement through the blood to assist with managing numerous different organs, implying that it is an endocrine organ. These chemicals typically act in the body to control energy use, baby advancement, and adolescence improvement. High level machine science is utilized in the space of medical care. It expected information to be gathered for clinical sickness expectation. For beginning phase infection identification, different smart expectation calculations are utilized. The Clinical Data Framework is great with informational indexes, yet clever frameworks are not accessible for the quick finding of sicknesses. In the end, AI calculations play a vital situation in taking care of complex and non-direct issues during the production of expectation model. The qualities that can be chosen from the different informational indexes that can be utilized as portrayal in a solid patient as explicitly as conceivable are required in any sickness forecast models. If not, misclassification can bring about a decent persistent getting unseemly consideration. The truth of gauging any condition related with thyroid disease is likewise of the best cardinal number. Thyroid organ is endocrine in stomach. It is raised in brought down piece of human neck, under apple of Adam, and helps in emission of thyroid chemicals and which atlast influences digestion rate and protein union. To control body digestion, these chemicals depend on how rapidly heart beats and how rapidly calories consume. The organization of thyroid chemicals assists with controlling the body's digestion. These organs comprise of two mature levothyroxine (truncated T4) and triiodothyronine thyroid chemicals (condensed T3). These thyroid chemicals are fundamental for assembling and general development and guideline to manage internal heat level. T4 and T3 are only two enacted thyroid chemicals that typically make out of thyroid organs.

These chemicals are indispensable to the control of proteins; dissemination at internal heatlevel and energy-bearing and spread in all aspects of the body. With T3 and T4 hormones,iodine is essential structure block of thyroid organs and is prostrate in just a few novel issues, which are extremely pervasive. Deficient components of these chemicals to hypothyroidism and an unseemly part to hyperthyroidism. Hyperthyroidism and underactive thyroidism have various starting points. There are various medications. Thyroid medical procedure is feeble to ionizing radiation, constant thyroid non- abrasiveness, iodine lack, and loss of protein to create thyroid chemicals.

CHAPTER – 2

LITERATURE SURVEY

1. **Ankita Tyagi** and **Rikitha Mehra** proposed "Interactive Thyroid Disease Prediction System Using Machine Learning Techniques" 5th IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC-2018), 20-22 Dec, 2018, Solan, India.

2. **Ataide et al** proposed **soft computing techniques** for thyroid prediction. The results on the UCI data set showed that multilayer perceptron (MLP) yielded an accuracy of **97.4%.** However, after feature extraction the same classifier shown an accuracy **of 91.7%** which is less than previous results.

3. **Yasir Iqbal** and **Mir Dr. Sonu Mittal** proposed Thyroid Disease Prediction Using **Hybrid Machine Learning Techniques:** An Effective Framework International journal of scientific & technology volume 9, issue 02, February 2020, the features of the dataset We may achieve the highest accuracy of **92.07%** by J48 classifier.

4. **Gyanendra Chaubey** and team proposed Thyroid Disease Prediction Using **Machine Learning Approaches** Article in National Academy Science Letters · May 2020 with maximum accuracy of **98.89%** Hence, according to the dataset which is used in this work, the accuracy obtained is satisfactory.

Understanding: In these works, they use different classification **algorithms-Support**

Vector Machine, Artificial Neural Network, k-Nearest-Neighbor algorithm. They have analyzed accuracy of algorithms used and comparison is made to find best technique with high accuracy

CHAPTER- 3

EXISTING SYSTEM

- An effective thyroid disease prediction system helps users to predict the accuracy of the disease and draw conclusions from it.
- This system widely uses machine learning techniques and algorithms namely Support

  Vector Machines(SVM), decision trees, k-nearest neighbor (kNN) and other algorithms to predict and evaluate their performance in terms of accuracy.
- In existing systems, the symptoms are often less marked, thus the disease may be diagnosed several years after onset, once complications have already arisen.
- The problem with it is it fails to predict all possible conditions of the people and also the accuracy level of prediction for these systems is comparitively low
- Data Mining provides many classification techniques for the prediction of disease accuracy.

  The need of patient data collected from much health care organization is useful for the risk factors analysis for many diseases.
- The clinical decisions are usually based on the doctor's intuition. Therefore this may lead

  to disastrous consequences. Due to this there are many errors in the clinical decisions and it results in excessive medical costs.
- The researchers resorted to relying on machine learning techniques in this study to classify

  thyroid disease. Classification is used to characterize pre-defined data sets, is one of the most popular supervised learning data mining techniques.
- The classification is commonly used in the healthcare sector to aid in medical decision- making, diagnosis, and administration
- There are various algorithms of machine learning counting random forest, decision tree, naïve Bayes, SVM and ANN that are extensively used in the frequent diseases and in the prognostic problems.

- The supervised ML methods rely on labelled input data to gain knowledge about function which generates proper output while new-fangled data is given without label. Naive Bayes is a keen erudition classifier and certainly, it is hasty and well-known method for multi-class classification.

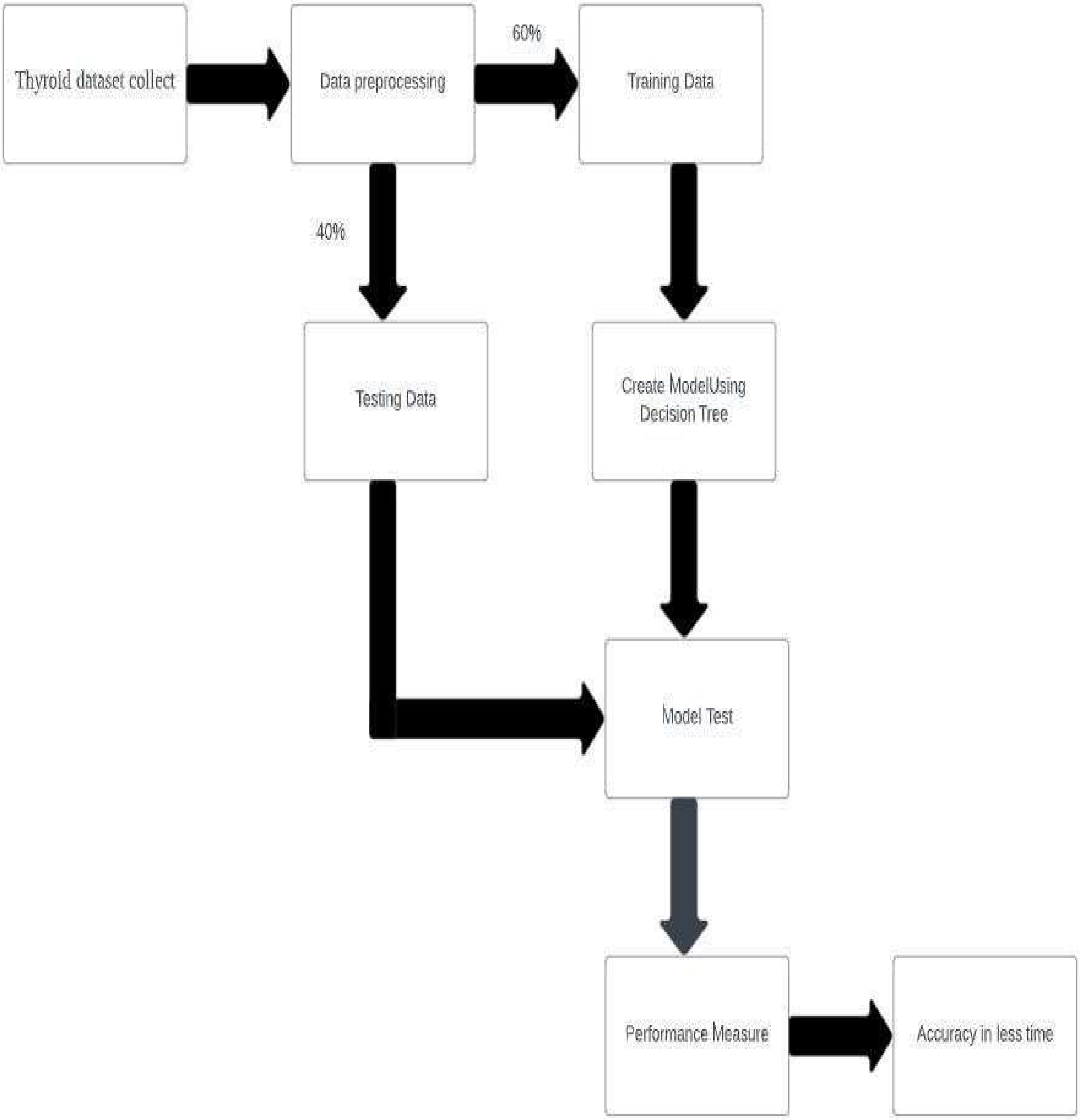
CHAPTER – 4

PROPOSED SYSTEM

- The proposed system is having many advantages over the existing system. It requires much less overhead and is very efficient. Machine Learning plays a very deciding role in disease prediction.
- Machine Learning algorithms, SVM - Support Vector Machine, Decision Tree, KNN - K-Nearest Neighbor's are used to predict the patient's risk of getting thyroid disease.
- Data from Kaggle is used to predict the type of disease.
- Our objective was to give society an efficient and precise way of machine learning which can be used in applications aiming to perform disease detection
- The algorithm extracts the features from different dataset to classify the data according to the labels. To check the accuracy of the prediction, test data is fed to the algorithm
- We also came up with the best model among three ML algorithms than can predict TD

CHAPTER - 5

SYSTEM ARCHITECTURE



CHAPTER - 6

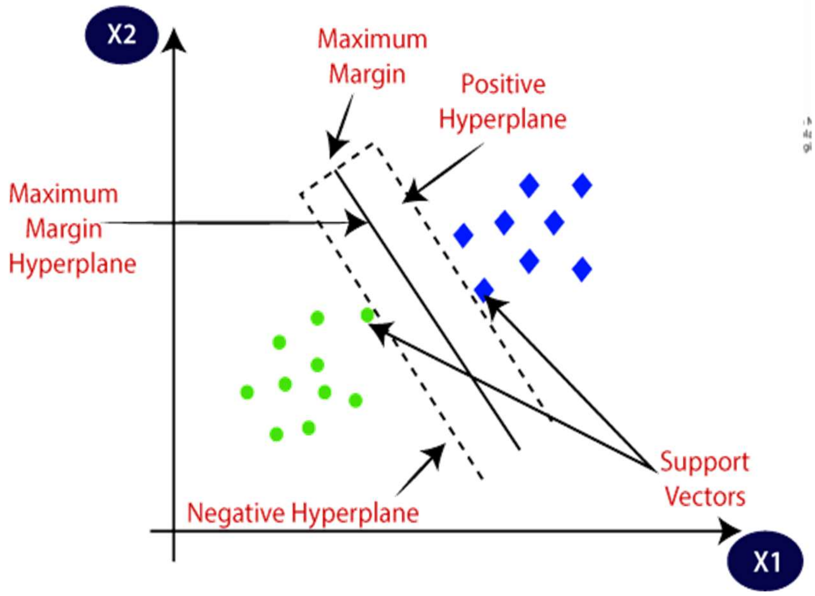
MACHINE LEARNING ALGORITHM

6.1 SVM

Support Vector Machine or SVM is one of the most famous Managed Learning calculations, which is utilized for Grouping as well as Relapse issues. In any case, essentially, it is utilized for Grouping issues in AI.

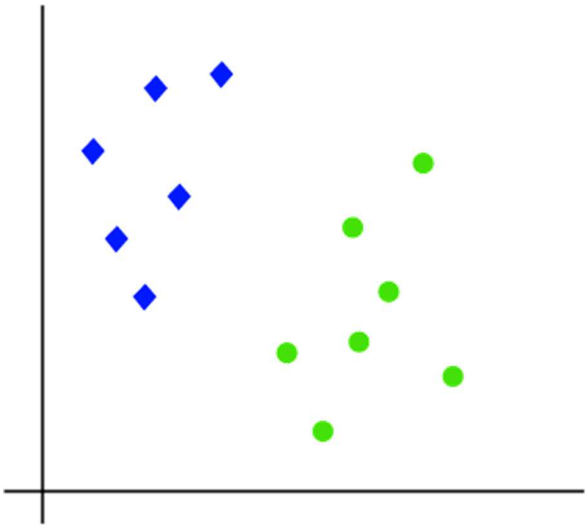
The objective of the SVM calculation is to make the best line or decision limit that can isolate n-layered space into classes so we can undoubtedly put the new data of interest in the right classification later on. This best decision limit is known as a hyperplane.

SVM picks the silly centers/vectors that help with making the hyperplane. These outrageous cases are called as help vectors, and subsequently calculation is named as HelpVector Machine. Consider the underneath graph in which there are two distinct classifications that are grouped utilizing a decision limit or hyperplane:

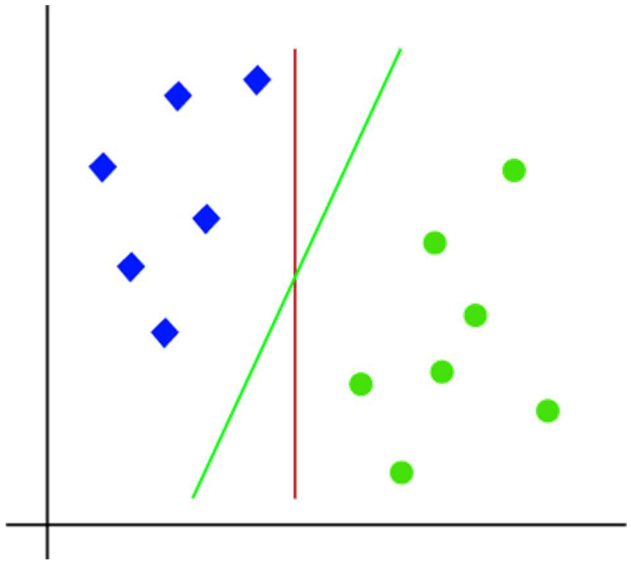


6.2 HOW DOES SVM WORKS

The working of the SVM calculation can be perceived by utilizing a model. Assume we have a dataset that has two labels (green and blue), and the dataset has two highlights x1 and x2. We want a classifier that can describe the pair(x1, x2) of headings in either green or blue. Consider the underneath picture:



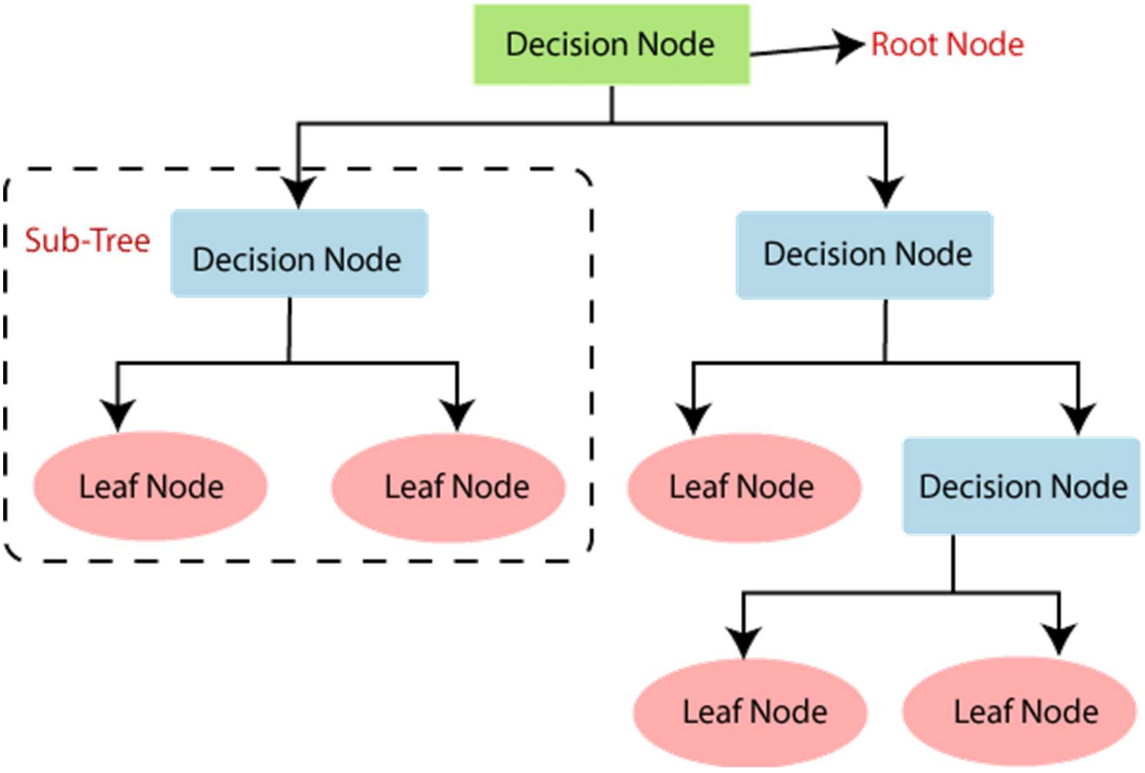
So as it is 2-d space so simply by utilizing a straight line, we can without much of a stretch separate these two classes. However, there can be numerous lines that can isolate these classes. Consider the underneath picture:



6.3 DECISION TREE

We really want a classifier that can describe the pair (x1, x2) of headings in either green or blue. The decision Tree is the most remarkable and famous device for characterization and expectation. A Decision tree is a flowchart-like tree structure, where each inside node indicates a test on a trait, each branch addresses a result of the test, and each leaf node (terminal node) holds a class name.

A Decision Tree is a supervised learning procedure that can be utilized for both classification and Regression issues, yet generally, it is liked for taking care of classification problem. It is a tree-organized classifier, where inside nodes address the elements of a dataset, branches address the decision principles and each cleaf node addresses the result. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. decision nodes are utilized to go with any decision and have numerous branches, though Leaf nodes are the result of those decsions and contain no further branches. The decisons or the test are performed based on elements of the given dataset. It is a graphical portrayal for getting every one of the potential answers for an problem/decision in view of given conditions. It is known as a decision tree on the grounds that, like a tree, it begins with the root node, which develops further branches and builds a tree-like design.



6.4 Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

6.5 How does the Decision Tree algorithm Work?

In a decision tree, for foreseeing the class of the given dataset, the calculation begins from the root node of the tree. This calculation contrasts the upsides of root property and the record (genuine dataset) characteristic and, in light of the examination, follows the branch and leaps to the following node.

For the following node, the calculation again contrasts the property estimation and the other sub- node and moves further. It proceeds with the cycle until it arrives at the leaf node of the tree. The total cycle can be better perceived by utilizing the underneath calculation:

- **Step-1:**Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively go with new decision trees utilizing the subsets of the dataset made in step - 3. Proceed with this interaction until a phase is reached where you can't further order the node and called the last node a leaf node.

6.7 KNN ALGORITHM

The k-Nearest-Neighbours (kNN) non-parametric classification approach is straightforward but frequently successful. In order to classify a data record t, its k nearest neighbours are obtained, creating a neighbourhood for t. a majority of those voting. With or without taking into account distance-based weighting, the categorization for t is typically determined using the data records in the neighbourhood. However, in order to use kNN, we must select a proper value for k, and the classification's outcome greatly depends on this number. The kNN approach is somewhat skewed by k. There are numerous methods for selecting the k value, but one straightforward one is to repeatedly run the algorithm with various k values and select the one that performs the best.

It was MacQueen who first proposed the K-means algorithm. Ball and Hall's ISODATA algorithm was an early but sophisticated iteration of k-means. The objects are divided into meaningful groups through clustering. Learning without supervision is clustering. Automatic document organization is known as document clustering.

We select K initial centroids in the K-means clustering technique, where K is the desired number of clusters. Each point is then assigned to the cluster's centroid, which has the closest mean. The centroid of each cluster is then updated based on the points assigned to the cluster. Up until the cluster, center doesn't change, we repeat the process (centroid). Last but not least, the objective of this algorithm is to minimize an objective function, in this case, a squared error function.

The objective function is given by,

$$J = \sum_{j=1}^{k} \sum_{j=1}^{n} \left\| x_i^{(j)} - C_j \right\|^2 \left\| x_i^{(j)} - c_j \right\|^2$$

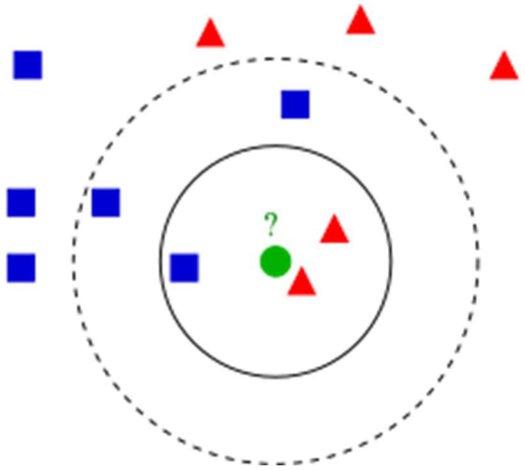
Where k is the number of clusters, n is the number of cases is a chosen distance measure between a data point and the cluster centre is an indicator of the distance of the n data points from their respective cluster centers.

One of the simplest machine learning algorithms, based on the supervised learning method, is K-Nearest Neighbour. The K-NN algorithm assumes that the new and existing cases are comparable and places the new instance in the category that is most like the existing categories. A new data point is classified using the K-NN algorithm based on similarity after storing all the existing data. This means that fresh data can be quickly and accurately sorted into a suitable category utilizing the K-NN method. Albeit the K-NN approach is most often utilized for arrangement issues, it can likewise be used for relapse. Since K-NN is a non-parametric strategy, it makes no presumptions about the basic information. It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to act when classifying data. The KNN method simply saves the information during the training phase, and when it receives new data, it categorizes it into a category that is quite similar to the new data.

The training examples are vectors with class labels in a multidimensional feature space. Only the feature vectors and class labels of the training samples are stored during the algorithm's training phase. The label that appears most frequently among the k training samples that are the closest to the unlabelled vector (a query or test point) during the classification phase, where k is a user-defined constant, is assigned to the query point.

Euclidean distance is a typical distance metric for continuous variables. Another metric, such as the overlap metric, can be used for discrete variables, such as text categorization (or Hamming distance). For instance, k-NN has been used with correlation coefficients, suchas Pearson and Spearman, as a metric in the context of gene expression micro array data. Frequently, the classification accuracy of k-NN can be significantly increased if the distance metric is learned with specialized algorithms, such as Large Margin Nearest Neighbour or Neighbourhood components analysis.

When the distribution of the classes is skewed, the fundamental "majority voting" categorization has a disadvantage. Because of their huge quantity, examples from a more frequent class tend to dominate predictions for new examples. One solution to this issue is to weight the classification by taking into consideration the distance between the test point and each of its k nearest neighbours. Each of the k closest points' class (or value, in regression problems) is multiplied by a weight that is proportional to the inverse of the distance separating it from the test point. Abstraction in data representation is a different strategy for dealing with skew. Depending on their density in the initial training data, each node in a self-organizing map (SOM) is a representative (a center) of a cluster of similar points. The SOM can then be treated using K-NN.



Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle)

6.8 KNN ALGORITHM PSEUDOCODE IMPLEMENTATION:

1. Load the desired data.
2. Choose the value of k.
3. For getting the class that is to be predicted, repeat starting from 1 to the total number of training points we have.
4. The subsequent stage is to work out the distance between the information point whose class is to be anticipated and all the preparation data of interest. Euclidean distance can be utilized here.
5. Arrange the distances in non-decreasing order.
6. Assume the positive value of k and filtering k lowest values from the sorted list.
7. We have top k top distances.
8. Let ka represent the points that belong to the ath class among k points.
9. If ka>kb then put x in the class.

CHAPTER – 7

IMPLEMENTATION

7.1 CODE :

```
import pandas as pd

df = pd.read_csv("thyroid.csv")

import numpy as np

df=df.replace({"?":np.NAN})

df=df.replace({"P":0,"N":1})

df=df.replace({"M":1,"F":2})

df=df.replace({"t":1,"f":0})

dele=["TBG","referral source",'TSH measured','T3 measured','TT4 measured','T4U measured','FTI measured',"TBG measured"]

for i in dele:

del df[i]

df.columns
```

```
Index(['age', 'sex', 'on thyroxine', 'query on thyroxine',

      'on antithyroid medication', 'sick', 'pregnant', 'thyroid surgery',

      'I131 treatment', 'query hypothyroid', 'query hyperthyroid', 'lithium',

      'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH', 'T3', 'TT4', 'T4U',

      ‘FTI’, binaryClass])
```

```
from sklearn.impute import SimpleImputera
= SimpleImputer(strategy='mean') df['TSH']
=  a.fit_transform(df[['TSH']])
 df['sex'] = a.fit_transform(df[['sex']])
 df['TT4'] = a.fit_transform(df[['TT4']])
 df['FTI'] = a.fit_transform(df[['FTI']])
 df['T4U'] = a.fit_transform(df[['T4U']])
 df['age'] = a.fit_transform(df[['age']])
 df['T3'] = a.fit_transform(df[['T3']])
 from sklearn.model_selection import train_test_split as tts
 y= df["binaryClass"]
 x= df[['age', 'sex', 'on thyroxine', 'query on thyroxine',
        'on antithyroid medication', 'sick', 'pregnant', 'thyroid surgery',
        'I131 treatment', 'query hypothyroid', 'query hyperthyroid', 'lithium',
        'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH', 'T3', 'TT4', 'T4U',
        'FTI']]
 x_train, x_test, y_train, y_test = tts(x, y, train_size=0.6, random_state=0)
 from sklearn.tree import DecisionTreeClassifier
 dtcmodel = DecisionTreeClassifier()
 dtc = dtcmodel.fit(x_train,y_train)
 dtcy_pred = dtcmodel.predict(x_test)
 from sklearn.metrics import accuracy_score as acs
 print("DecisionTreeClassifier Accuracy :",acs(dtcy_pred,y_test))
```

```
Decisiontreeclassifier  Accuracy  :  0.9973492379058979
```

```
from sklearn.neighbors import KNeighborsClassifier
knmodel = KNeighborsClassifier(n_neighbors=3)
knn = knmodel.fit(x_train,y_train)
knny_pred = knmodel.predict(x_test)
print("KNeighborsClassifier Accuracy :",acs(knny_pred,y_test))
```

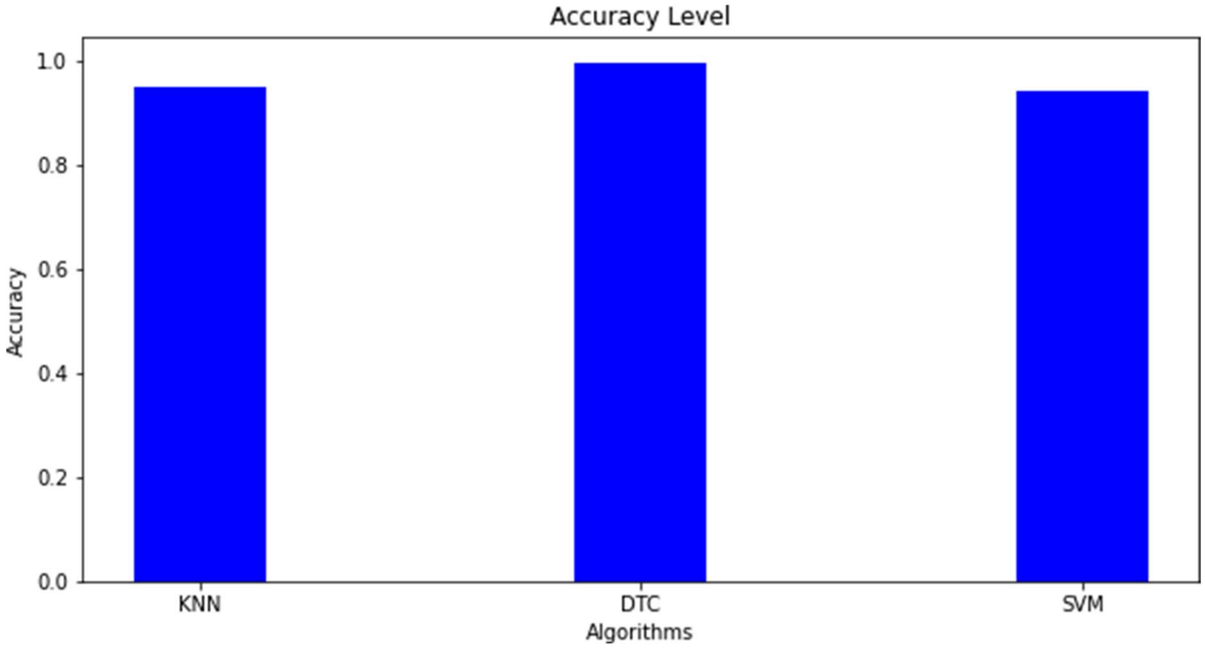
```
KNeighborsClassifier Accuracy : 0.950960901259112
```

```
from sklearn.svm import SVC
svcmodel = SVC()
svc = svcmodel.fit(x_train,y_train)
svcy_pred = svcmodel.predict(x_test)
print("SupportVectorMachine Accuracy :",acs(svcy_pred,y_test))
```

```
SupportVectorMachine Accuracy : 0.9443339960238568
```

```
import numpy as np
import matplotlib.pyplot as plt
data = {'KNN':0.950960901259112, 'DTC':0.9973492379058979, 'SVM':0.9443339960238568}
alg = list(data.keys())
val = list(data.values())
fig = plt.figure(figsize = (10, 5))
plt.bar(alg, val, color ='b')
plt.xlabel("Algorithms") plt.ylabel("Accuracy") plt.title("Accuracy Level”)

plt.show()
```



```
import matplotlib.pyplot as plt

from sklearn import metrics

confmatrix = metrics.confusion_matrix(y_test, dtcy_pred)

cmdisp = metrics.ConfusionMatrixDisplay(confusion_matrix = confmatrix, display_label

s = [False, True])

cmdisp.plot()

plt.show()

l1=[]

prediction = dtcmodel.predict(x_test)

l1.append(prediction)
```

```
for i in l1:
    for j in i:
        print(j)
 x = input().split(",")
 l=[]
 for i in x:
     l.append(float(i))
 pred = dtcmodel.predict([l])          .
 dic ={1:"Male",2:"Female"}          .
 perg = {1:"Pregnant", 2:"Not pregnant"}
 if l[1] == 1:
     print(dic[l[1]])
 else :
     print(dic[l[1]])
     if l[6] ==1:
         print(perg[1])
     else:
         print(perg[2])
 if pred == 1:
     print("Postive")
 else:
     print("Negative")
```

7.2 OUTPUT :

68.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,2.4,1.6,83.,0.89,93.

Male

Negative

7.3 IMPORTED PACKAGES

- import pandas as pd
- import numpy as pd
- import matplotlib.pyplot as plt
- from sklearn.impute import SimpleImputer
- from sklearn.model_selection import train_test_split as tts
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.neighbours import KNeighboursClassifier
- from sklearn.svm import SVC
- from sklearn.metrics import accuracy_score as acs

CHAPTER - 8
CONFUSION MATRIX :

- The confusion matrix drawn on the random selection of **test data** on the random selection of **training data**

- The confusion matrix explains about the how much the model **is accurate.**

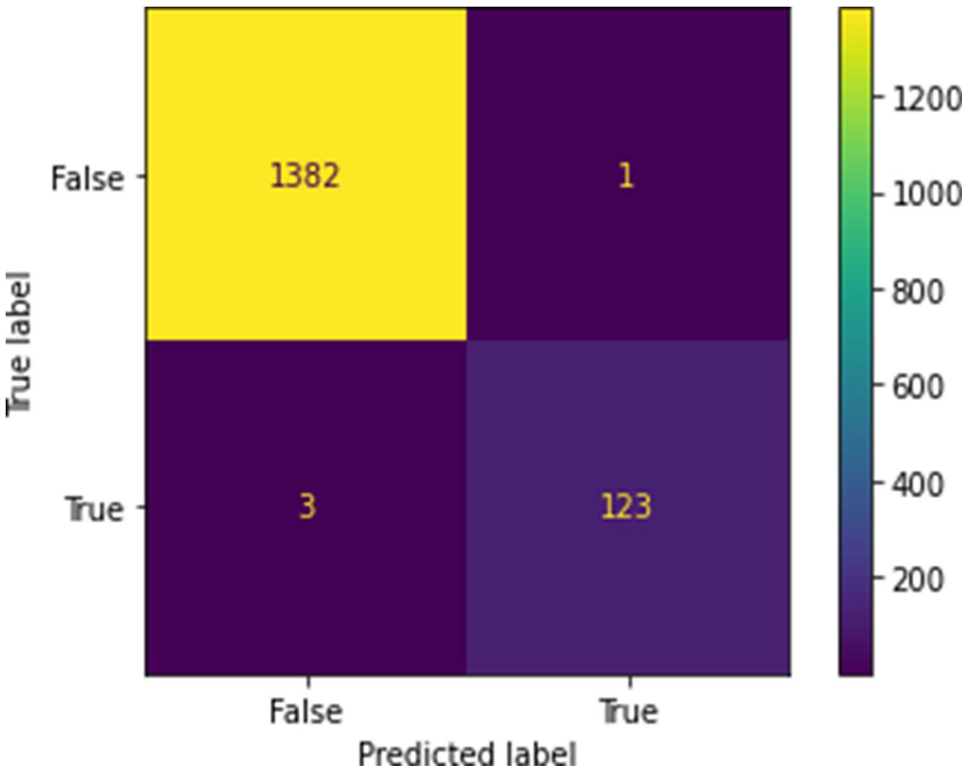
- The **formula** for the calculation of accuracy from the confusion matrix is

given as

$$\textbf{Accuracy} = \frac{\mathbf{TP + TN}}{\mathbf{(TP+FN) + (FP+TN)}}$$

where **TP** true positive, **FP** false positive, **FN** false negative,

TN true negative.



CHAPTER - 10

10.1 SOFTWARE REQUIREMENTS

S.No	Software	Version	URL
1.	Anaconda	2.3.1	https://www.anaconda.com/anaconda-distribution
2.	Python	3.9.10	https://www.python.org/
3.	Matplotlib	3.6.2	https://matplotlib.org/
4.	Sklearn	1.1.3	https://pypi.org/project/scikit-learn/
5.	Pandas	1.5.2	https://pandas.pydata.org/
6.	Numpy	1.23.0	https://numpy.org/

10.1.1 ANACONDA

For scientific computing, Anaconda is a free and open-source distribution of the Python and R programming languages (data science, machine learning applications, large-scale data processing. predictive analytics, etc.). It seeks to make deployment and package management simpler. Conda, a package management system, controls package versions. For Windows, Linux, and macOS, the Anaconda distribution provides data-science packages. More than 250 packages are pre-installed in the anaconda distribution, while more than 7,500 additional open-source packages, the anaconda package manager, and the virtual environment manager may all be downloaded from PyPI. In addition, Anaconda Navigator, a GUI, is provided as a graphical replacement for the command-line interface (CLI). Without using command-line commands, users can start applications and manage conda packages, environments, and channels with the help of Anaconda Navigator, a desktop GUI that is part of the Anaconda distribution. The packages can be installed in an environment, operated, and updated using Navigator. It can search for packages on Anaconda Cloud or in a local Anaconda Repository. It works with Windows, macOS, and Linux. Conda is a package manager and environment management system that instals, runs, and updates packages and their dependencies. It is open source, cross-platform, and language-agnostic. Although it was designed for Python scripts, it can package and distribute software for any language, including multi-language projects (for example, R). All versions of Anaconda, Miniconda, and Anaconda Repository include the conda package and environment manager.

10.1.2 PYTHON

A high-level, all-purpose programming language is Python. Code readability is emphasized in its design philosophy, which makes heavy use of indentation. Python utilizes trash assortment and has dynamic composing. It supports a variety of programming paradigms, including procedural, object-oriented, and structured programming. A multi-paradigm programming language is Python. Fully supported are structured and object-oriented programming and many of its capabilities also allow functional and aspect-oriented programming (including metaprogramming and metaobjects. Extensions are available for many other paradigms, such as design by contract and logic programming. Python's memory management system combines reference counting and a cycle-detecting garbage collector with dynamic typing. Method and variable names are bound via dynamic name resolution (late binding), which takes place while the program is running. Python aims for a more straightforward, uncluttered syntax and grammar while offering developersa selection of development paradigms.

10.1.3 MATPLOTLIB

Matplotlib is a graphing library for the Python programming language and its NumPy numerical mathematics extension. For integrating charts into applications using all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK, it offers an object-oriented API. It is not recommended to use the procedural "pylab" interface, which is based on a state machine (similar to OpenGL) and was created to closely mimic the MATLAB interface. Matplotlib is used by SciPy. A Matplotlib package called Pyplot offers a MATLAB-like user interface. With the option to utilize Python, Matplotlib is meant to be as user-friendly as MATLAB.

10.1.4 SKLEARN

French data scientist David Cournapeau's scikits. learn Google Summer of Code project is where the scikit-learn project first began. Its name refers to the idea that it is a third-party modification to SciPy called a "SciKit" (SciPy Toolkit), which was independently created and distributed. Later, other developers rewrote the original codebase. The French Institute for Research in Computer Science and Automation in Saclay, France, led the project in 2010 under the direction of Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel. On February 1st of that year, the project's first public release was issued. In November 2012, scikit-learn and scikit-image were cited as examples of scikits that were "well-maintained and popular." One of the most well-liked machine learning libraries on GitHub is scikit-learn. Scikit-learn is a free machine learning package for the Python programming language (formerly known as scikits. learn and also referred to as sklearn). Support-vector machines, random forests, gradient boosting, k-means, and DBSCAN are just a few of the classification, regression, and clustering algorithms it offers. It is also built to work with Python's NumPy and SciPy scientific and numerical libraries.

10.1.5 PANDAS

To manipulate and analyse data, the Python programming language has a software package called pandas. It includes specific data structures and procedures for working with time series and mathematical tables. Pandas is mostly used for tabular data manipulation and analysis in Data Frames. Data can be imported into Pandas from a variety of file types, including Microsoft Excel, JSON, Parquet, SQL database tables, and comma-separated values. Pandas support many data manipulation tasks, including merging, reshaping, selecting, cleaning and wrangling data. Many of the same functionalities for working with Data Frames that were developed in the R programming language were brought into Python with the development of pandas. The NumPy library, which is focused on working well with arrays rather than the characteristics of working with Data Frames, is the foundation upon which the pandas' library is constructed.

10.1.6 NUMPY

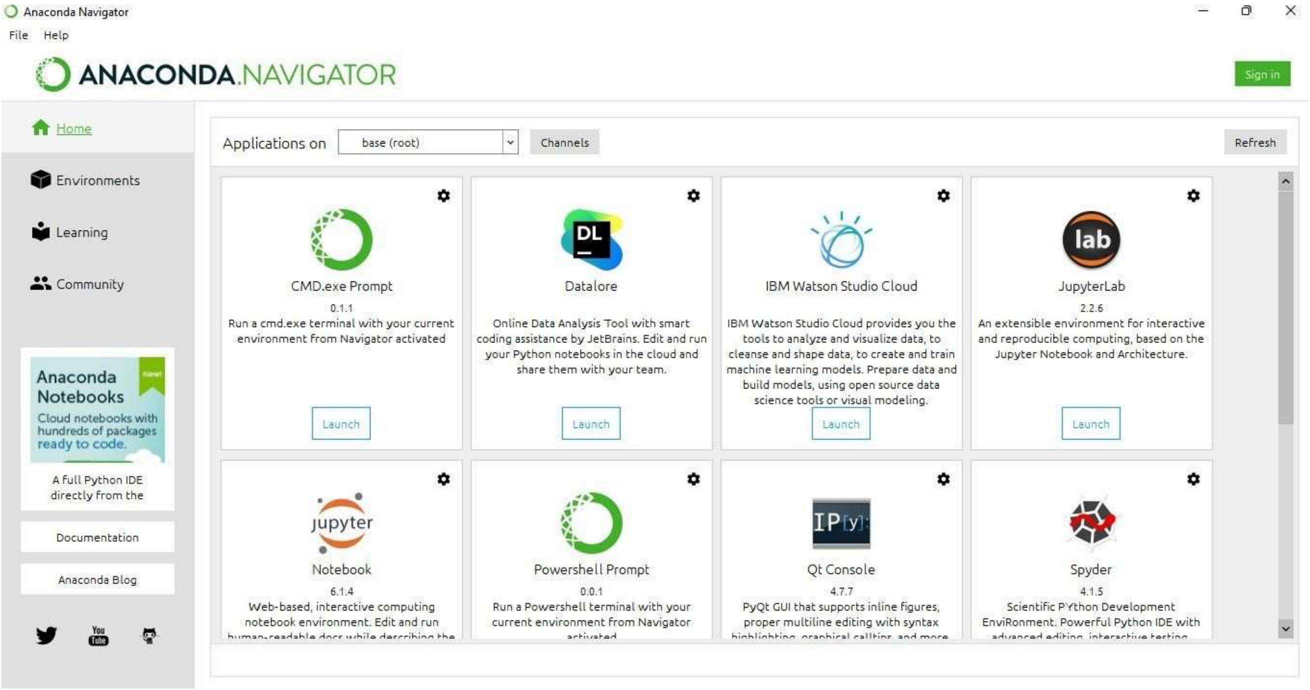
NumPy is a library for the Python programming language that considers more informationstockpiling with less memory. With a complex cluster and different assets, NumPy permitsPython developers to store numbers efficiently. Since Python is ostensibly the most broadly involved language in AI, NumPy addresses a basic center component of a specialist's tool stash for brain organizations and related AI programs. By using the library asset, softwareengineers can arrange all of this more significant level examination in a manner that advances proficiency. Different libraries and instruments, for example, SciPy are likewiseuseful toward this end, however NumPy explicitly addresses the requirement for huge complex clusters and lattice mathematical capacity.

10.2 INSTALLATION PROCEDURE

Step 1 - Install the dependencies for Windows

1. Download & install the Anaconda package 64-bit version [7] and choose Python 3.9. 10, version. This automatically installs Python and many popular data scientist/ML libraries (NumPy, Scikit-Learn, Pandas. R. Matplotlib...). tools (Jupyter Notebook, RStudio) and hundreds of other open-source packages for your future projects.

2. Install Word cloud and Lenskit. Word cloud is the package in python that is used for generating a little word cloud. Lenskit is another popular package which is used for recommendation systems.



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

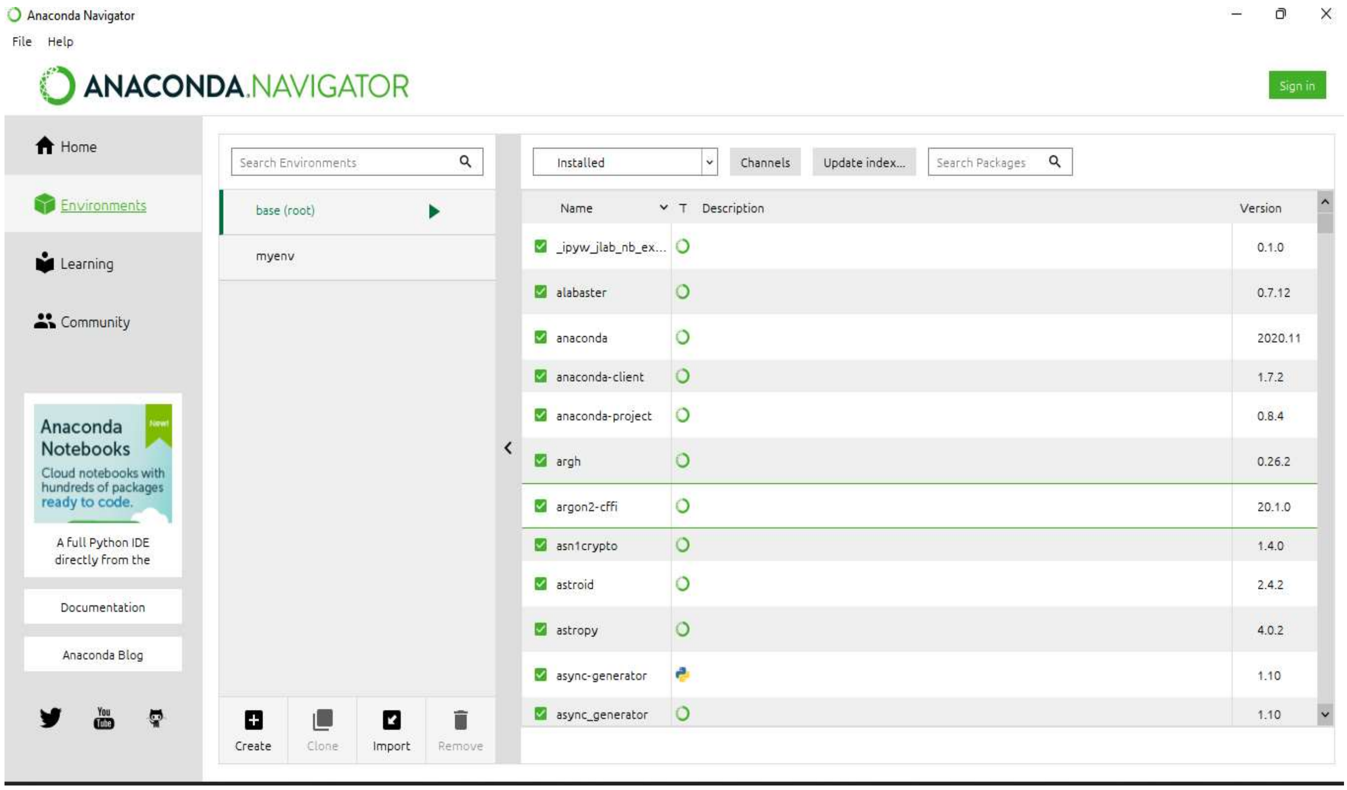
JupyterLab
2.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.1.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
4.7.7
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing



base (root)

myenv

	Name	T	Description	Version
☑	_ipyw_jlab_nb_ex...	○		0.1.0
☑	alabaster	○		0.7.12
☑	anaconda	○		2020.11
☑	anaconda-client	○		1.7.2
☑	anaconda-project	○		0.8.4
☑	argh	○		0.26.2
☑	argon2-cffi	○		20.1.0
☑	asn1crypto	○		1.4.0
☑	astroid	○		2.4.2
☑	astropy	○		4.0.2
☑	async-generator			1.10
☑	async_generator	○		1.10

10.3 HARDWARE REQUIREMENTS

1. Processor: Intel (R) Pentium(R)

2. Speed: 1.6 GHz and Above.

3. RAM: 4 GB and Above.

4. Hard Disk: 120 GB.

5. Monitor: 15" LED SVGA

6. Input Devices: Keyboard, Mouse.

10.4 SOFTWARE REQUIREMENTS

2. Operating system: Windows 7/8/8.1/10.

3. Coding Language: PYTHON

4. Python Version: Python 3.9

CHAPTER - 11

CONCLUSION

Problems with the thyroid include a variety of disorders that can result in the gland producing too little thyroid hormone (hypothyroidism) or too much (hyperthyroidism). Thyroid disorders can affect heart rate, mood, energy level, metabolism, bone health, pregnancy, and many other functions. The thyroid disease can then be easily identified based on the symptoms in the patient's history. Currently, models are evaluated using accuracy metrics on a validation dataset that is accessible. Here Decision tree algorithm provides the high accuracy of 99.7 %.

REFRENCE

1. L. Ozyılmaz and T. Yıldırım, “Diagnosis of thyroid disease using artificial neural network methods,” in: Proceedings of ICONIP’02 9th international conference on neural information processing (Singapore: Orchid Country Club, 2002) pp. 2033–2036.

2. K. Polat, S. Sahan and S. Gunes, “A novel hybrid method based on artificial immune recognition system (AIRS) with fuzzy weighted pre-processing for thyroid disease diagnosis,” Expert Systems with Applications, vol. 32, 2007, pp. 1141-1147.

3. F. Saiti, A. A. Naini, M. A. Shoorehdeli, and M. Teshnehlab, “Thyroid Disease Diagnosis Based on Genetic Algorithms Using PNN and SVM,” in 3rd International Conference on Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009.

4. F. Saiti, A. A. Naini, M. A. Shoorehdeli, and M. Teshnehlab, “Thyroid Disease Diagnosis Based on Genetic Algorithms Using PNN and SVM,” in 3rd International Conference on Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009.

5. V. Vapnik, Estimation of Dependences Based on Empirical Data, Springer, New York, 2012.

6. Obermeyer Z, Emanuel EJ. Predicting the future— big data, machine learning, and clinical medicine. N Engl J Med. 2016; 375:12161219.

7. Ehrenstein V, Nielsen H, Pedersen AB, Johnsen SP, Pedersen L. Clinical epidemiology in the era of big data: new opportunities, familiar challenges. Clin Epidemiol. 2017; 9:245-250.

8. Ghahramani Z. Probabilistic machine learning and artificial intelligence. Nature. 2015; 521: 452-459.

9. Azimi P, Mohammadi HR, Benzel EC, Shahzadi S, Azhari S, Montazeri A. Artificial neural networks in neurosurgery. J Neurol Neurosurg Psychiatry. 2015; 86:251-256.

10. P.C. Austin, J.V. Tu, J.E. Ho, D. Levy, D.S. Lee, Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes, J. Clin. Epidemiol. 66 (4) (2013)398–407.

11. L. Verma, S. Srivastava, P.C. Negi, A hybrid data mining model to predict coronary artery disease cases using noninvasive clinical data, J. Med. Syst. 40 (7) (2016)