

Disease Prediction by Symptoms

*Project report submitted to
Government Engineering College, Bharatpur
in partial fulfilment of the requirement for the award of
the degree*

**Bachelor of Technology
In
Computer Science & Engineering
by
Mohit Agrawal (16EELCS020)**

Under the guidance of
Mr. Hemant Saxena



**Department of Computer Science and Engineering
Government Engineering College
Bharatpur 321001(India)
2019-2020**

ABSTRACT

I carried out my internship at Dzone Software Solution & Service Provider, Jaipur. Dzone Software Solution & Service Provider represents the connected word offering innovative and customer-centric information technology experiences, enabling Enterprises, Associates and the Society to Rise. Dzone Software Solution & Service Provider provides internship opportunity to the students in various emerging technologies.

The purpose of the program is to fulfil the core equipment for the award of a degree of Bachelor of Technology in Computer Science and Engineering to get a practical aspect of the theoretical work studied at the university and to understand the operation in the corporate sector and to enable students gain experience in different tasks.

During my internship period, I was assigned to the department of Machine Learning where I was assigned to make a Disease Prediction Software. There I interacted with many working professionals.

There I have gained the knowledge of the things actually work in an organization like the complete procedure of implementing a project which include the understanding of the problem, the cost estimation of project, the methodology and the final implementation of the project.

In conclusion, this was an opportunity to develop and enhance skills and competencies in my career field which I actually achieved.

ACKNOWLEDGEMENT

I would like to take the opportunity to thank and express my deep sense of gratitude to my corporate mentor **Mr. Hemant Saxena** and my faculty mentor **Prof. Arvind Singh Chaudhary**. I am greatly indebted to both of them for providing their valuable guidelines at all stages of the study, their advice, constructive suggestions, positive and supportive attitude and continuous encouragement, without which it would have not been possible to complete the project.

I am thankful to **Mr. Hemant Saxena** for giving me the opportunity to work with Dzone Software Solution & Service Provider.

I would also like to thank my supervisor **Ms. Surbhi Saxena**, who helped me a lot during my internship period in completing my machine learning project.

I owe my wholehearted thanks and appreciation to the entire staff of the company for their cooperation and assistance during the course of my project.

I would also like to thank my parents, who helped me a lot during my internship period in my project.

I hope that I can build upon the experience and knowledge that I have gained and make a valuable contribution towards this industry in coming future.

Mohit Agrawal

CERTIFICATE



Software Training & Solution Provider

Phone : 0141-4108506
Mobile : 098297 08506
email : dzonehemant@gmail.com
website : www.dzone.co.in
Bank Account info of DZONE:
Central Bank of India
A/c No. 3042317496
IFSC : CBIN0283747

PLOT NO. 258, KATEWA NAGAR, NEW SANGANER ROAD, JAIPUR-302019 (RAJ.)

Ref. :- HRD/July-2019/TR-03

Date : 01/07/2019

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. Mohit Agrawal** student of **Govt. Engg. College ,Bharatpur** has worked with us as a Trainee on **Python/ML** platform between 1st May 2019 to 30th June 2019. During this period he worked on our multiple Projects in company and successfully completed them. His conduct during Internship was excellent.

His work is satisfactory and there are no dues against him in our company.

We wish him all the best for his future endeavours.

O - ZONE
Software Training & Solution Provider

Director

Mr. Hemant Saxena

(Director)

Table of Content

S. No	Content	Page No.
1.	Abstract	I
2.	Acknowledgement	Ii
3.	Certificate	Iii
4.	Table of Content	Iv
5.	Introduction	1
6.	Solution and Services	2
7.	Machine Learning (Introduction)	3
8.	Machine Learning Architecture	6
9.	Machine Learning Algorithms	7
10.	ML Development Lifecycle	10
11.	Setup ML Codebase	12
12.	ML Testing & Modelling	13
13.	ML Project Structure	14
14.	TKINTER in Python	25
15.	Disease Prediction Prototype	28
16.	Conclusion	29
17.	Bibliography	30

INTRODUCTION

The Company

Dzone Software Solution & Service Provider represents the connected word offering innovative and customer-centric information technology experiences, enabling Enterprises, Associates and the Society to Rise.

Dzone Software Solution & Service Provider is providing its services in field of software solution for the application development sector and ERP design with accelerated growth over the last 10 years. Our mission is to provide to our customer cost effective state of the art product and services, to enable them to implement straight through processes to better serve and retain their clients. We employ highly trained specialized and motivated people to deliver outstanding consulting implementation and training services.

We believe “Innovate from inside” i.e. we offer innovative solutions to our valuable customers that enable them to realize their full potential; we anticipate future trends and demand by engaging in active dialogue with our customers. Our commitment to our customer satisfaction is only matched by a relentless quest for forming strategic alliances with world-class software vendors and business consultants that assist us to expand and improve our value proposition to the benefit of our customer

Vision

We will Rise™ to be among the top three leaders in each of our chosen market segments while fostering innovation and inclusion.

We will consistently achieve top quartile growth by contributing to our customers' success, by enabling our employees to realize their potential and by creating value for all our stakeholders.

History

Dzone Software Solution & Service Provider started in 2010 as a technology outsourcing.

SOLUTIONS AND SERVICES

➤ Next Gen Solutions

- Big Data
- Content Delivery Network
- Device Testing and Certification
- Digital Enterprise Services
- Green and Sustainability Solutions
- Internet of Things(IOT)
- Industrial Internet of Things (IIOT)
- Long term Evolution
- Smart Grid

➤ Python Programming

➤ Swift Programming

➤ JavaScript

➤ Java

➤ Infrastructure and Cloud Services

➤ Mobile App Development

➤ Customer Experience

➤ DevOps

➤ Enterprise Architecture

➤ Machine Learning

Machine Learning

Introduction

Machine learning (ML) is the **scientific study** of **algorithms** and **statistical models** that computer systems use to perform a specific task without using explicit instructions, relying on patterns and **inference** instead.

Machine learning algorithms build a **mathematical model** based on sample data, known as "**training data**", in order to make predictions or decisions without being explicitly programmed to perform the task.

The goal of Machine Learning is to discover patterns in your data and then make predictions based on often complex patterns to answer business questions, detect and analyses trends and help solve problems.

Learning Algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

- Supervised Learning
- Semi-Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Features Learning

Features of Machine Learning

The important features of machine learning are:

- 1) Develop computational models of human learning process
- 2) Explore new learning methods and develop general learning algorithms independent of applications.
- 3) Make the computers smarter, more intelligent.
- 4) Machine Learning is inherently a multi-disciplinary subject area.
- 5) ML will produce smarter computers capable of all the above intelligent behavior.

ML Applications

There are many machine learning applications in the market. The top categories are:

- Banking
- Financial Market Analysis
- Medical Diagnosis
- Natural Language Processing
- Sentiments Analysis
- Recommendation Systems
- Time Series Forecasting etc.

History

Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM.

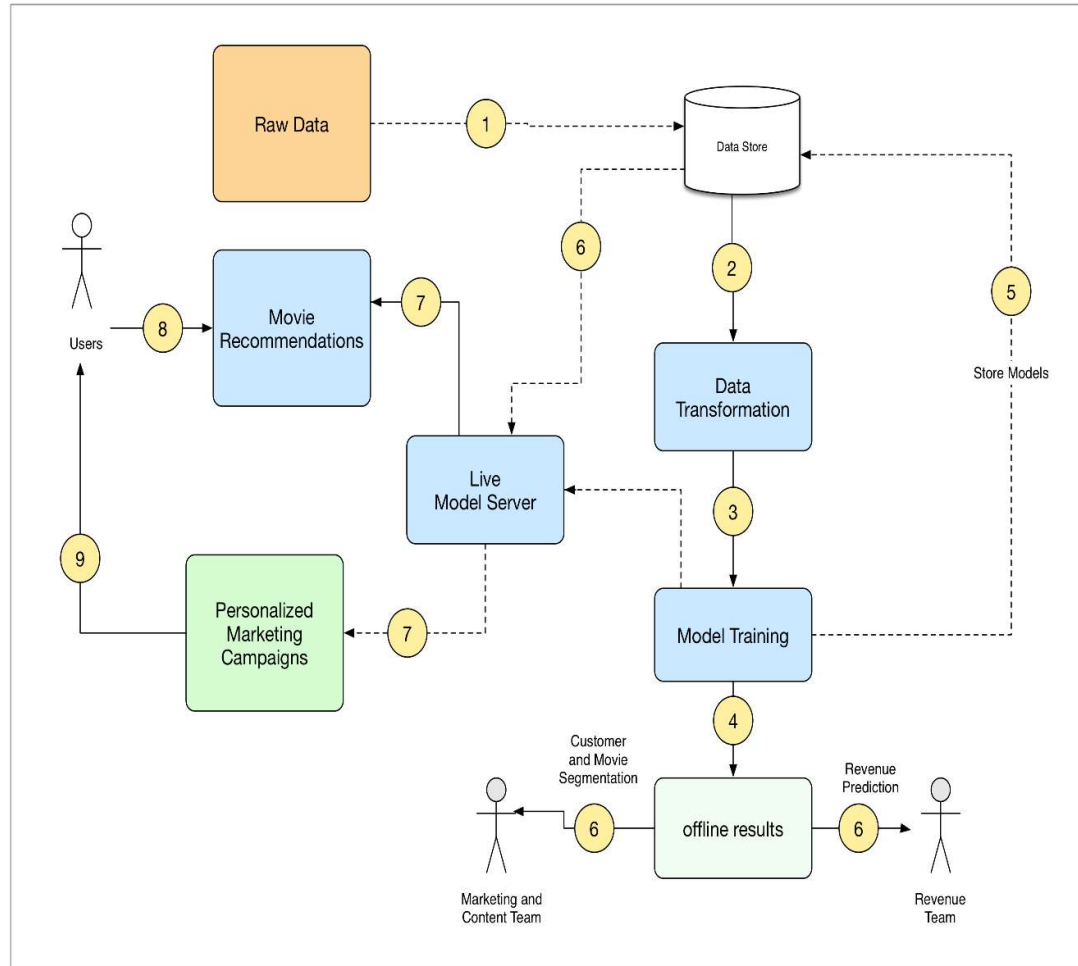
A representative book of the machine learning research during 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation.

HL vs ML

Dimension	Human Learning	Machine Learning
Speed	Slow	Fast
Ability to Transfer	No Copy mechanism	Easy to Copy
Required Repetition	Yes	Yes/No
Error-prone	Yes	Yes
Noise- tolerant	Yes	No

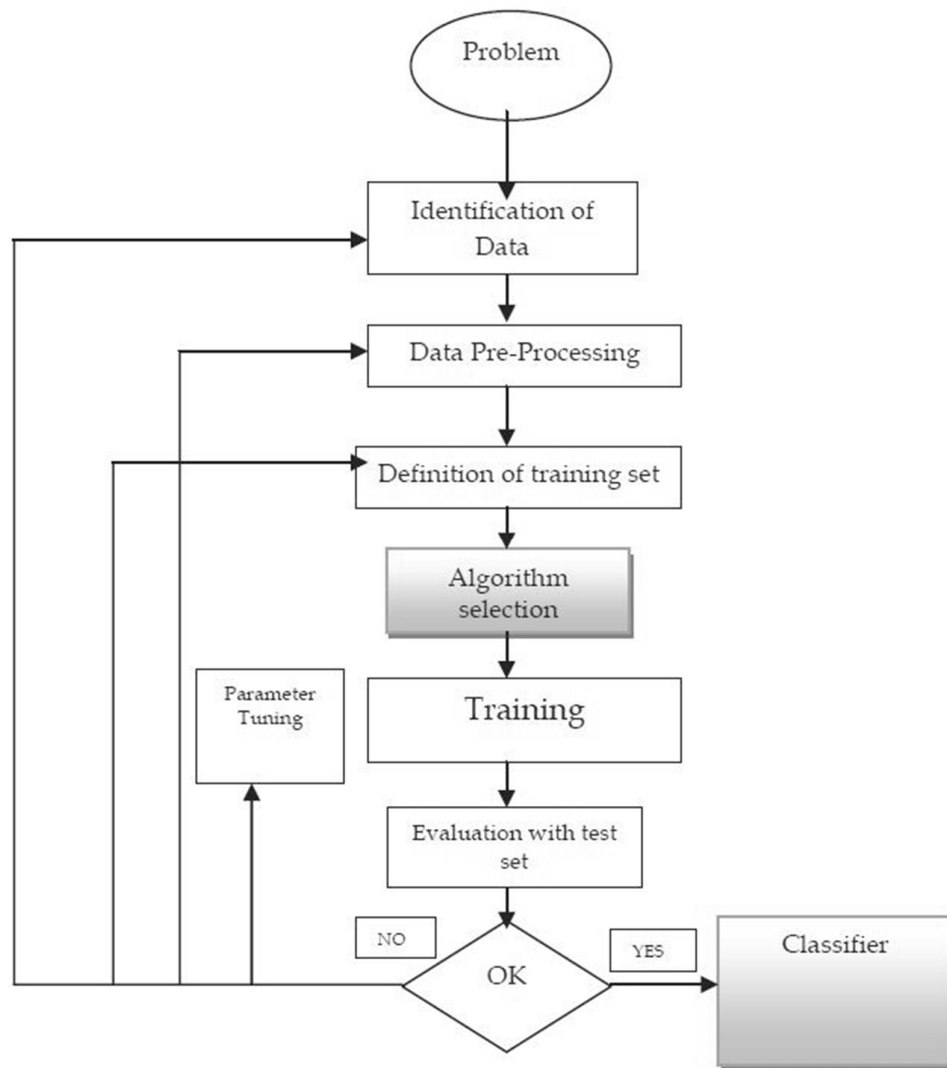
Machine Learning Architecture



Machine Learning Algorithms

Supervised Learning

Supervised learning is a machine learning technique for learning a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification).



Unsupervised Learning -

Unsupervised learning is a type of machine learning where manual labels of inputs are not used. It is distinguished from supervised learning approaches which learn how to perform a task, such as classification or regression, using a set of human prepared examples.

Semi-supervised Learning -

Semi-supervised learning combines both labeled and unlabeled examples to generate an appropriate function or classifier.

Reinforcement Learning -

Reinforcement Learning where the algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.

Transduction -

Similar to supervised learning, but does not explicitly construct a function.

Learning to Learn -

Learning to learn where the algorithm learns its own inductive bias based on previous experience.

Algorithms Types

Linear Classifiers -

In machine learning, the goal of classification is to group items that have similar feature.

1. Fisher's Linear Discriminant
2. Naïve Bayes Classifier
3. Perception
4. Support Vector Machine

Decision Tree -

A decision tree is a hierarchical data structure implementing the divide-and-conquer strategy. It is an efficient nonparametric method, which can be used for both classification and regression. A decision tree is a hierarchical model for supervised learning whereby the local region is identified in a sequence of recursive splits in a smaller number of steps. A decision tree is composed of internal decision nodes and terminal leaves (see figure).

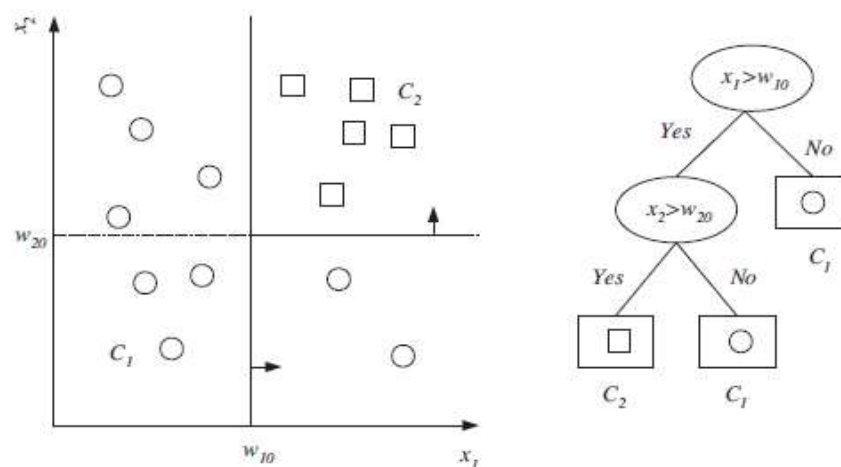
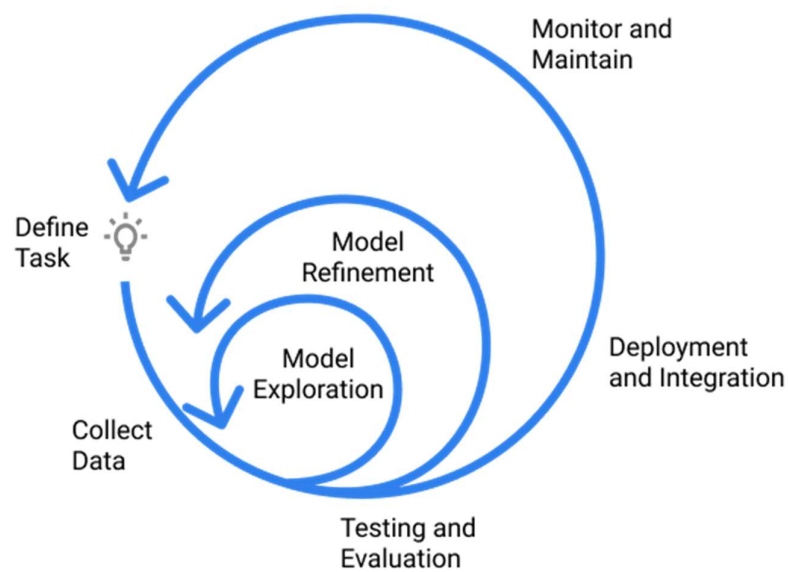


Figure 9.1 Example of a dataset and the corresponding decision tree. Oval nodes are the decision nodes and rectangles are leaf nodes. The univariate decision node splits along one axis, and successive splits are orthogonal to each other. After the first split, $\{x|x_1 < w_{10}\}$ is pure and is not split further.

Machine Learning Development Lifecycle

Machine learning projects are highly iterative; as you progress through the ML lifecycle, you'll find yourself iterating on a section until reaching a satisfactory level of performance, then proceeding forward to the next task (which may be circling back to an even earlier step).



Planning and project Setup-

- Define the task and scope out requirements.
- Determine project feasibility
- Discuss general model trade-offs (Accuracy vs Speed)
- Setup project codebase

Data Collection and labelling-

- Define ground truth (create labeling documentation)
- Build data ingestion pipeline
- Validate quality of data
- Revisit Step 1 and ensure data is sufficient for the task

Model Exploration-

- Establish baselines for model performance
- Start with a simple model using initial data pipeline
- Over fit simple model to training data
- Stay nimble and try many parallel (isolated) ideas during early stages

Model Refinement-

- Perform model-specific optimizations (i.e. hyper parameter tuning)
- Iteratively debug model as complexity is added
- Perform error analysis to uncover common failure modes
- Revisit Step 2 for targeted data collection of observed failures

Testing and Evaluation-

- Evaluate model on test distribution; understand differences between train and test set distributions (how is “data in the wild” different than what you trained on)
- Revisit model evaluation metric; ensure that this metric drives desirable downstream user behavior

Model Deployment-

- Expose model via a REST API
- Deploy new model to small subset of users to ensure everything goes smoothly, then roll out to all users
- Maintain the ability to roll back model to previous versions
- Monitor live data and model prediction distributions

Setting up a ML Codebase

```
data/  
docker/  
api/  
  app.py  
project_name/  
  models/  
    base.py  
    simple_baseline.py  
    cnn.py  
  configs/  
    baseline.yaml  
    latest.yaml  
  datasets.py  
  train.py  
  experiment.py  
scripts/
```

data/ provides a place to store raw and processed data for your project.

docker/ is a place to specify one or many Docker files for the project.

api/app.py exposes the model through a REST client for predictions.

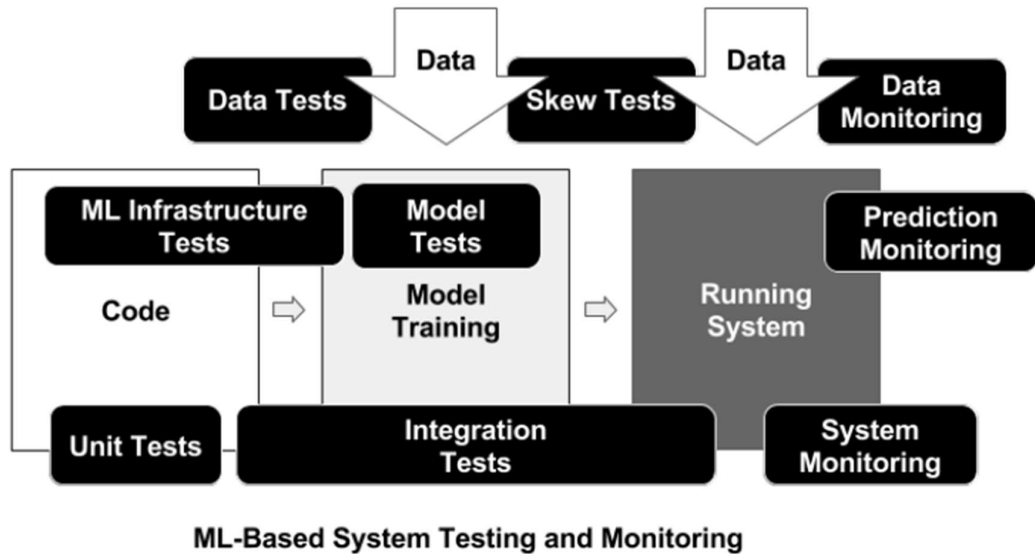
models/ defines a collection of machine learning models for the task, unified by a common API defined in **base.py**.

datasets.py manages construction of the dataset. Handles data pipelining/staging areas, shuffling, reading from disk.

experiment.py manages the experiment process of evaluating multiple models/ideas.

train.py defines the actual training loop for the model. This code interacts with the optimizer and handles logging during training.

ML-based System Testing and Monitoring-



Training system processes raw data, runs experiments, manages results, stores weights.

- Test the full training pipeline (from raw data to trained model) to ensure that changes haven't been made upstream with respect to how data from our application is stored. These tests should be run nightly/weekly.

Prediction system constructs the network, loads the stored weights, and makes predictions.

- Run inference on the validation data (already processed) and ensure model score does not degrade with new model/weights. This should be triggered every code push.

Serving system exposed to accept "real world" input and perform inference on production data. This system must be able to scale to demand.

Required monitoring:

- Alerts for downtime and errors
- Check for distribution shift in data

Machine Learning Project Structure-

Various businesses use machine learning to manage and improve operations. While ML projects vary in scale and complexity requiring different data science teams, their general structure is the same.

1. Strategy: Matching the problem with the solution-

In the first phase of an ML project realization, company representatives mostly outline strategic goals. They assume a solution to a problem, define a scope of work, and plan the development.

Disease Predication:

When a patient wants to consult to a doctor it may take much time or patient may be unable to consult to a doctor at that incident. Then there is a solution of the problem is that He can use Disease Prediction Software at primary level.

In this case, a user or patient can feed his symptoms to software, then machine learning model will predict the disease using some machine learning algorithms.

2. Dataset Preparation and Pre-processing –

Data is the foundation for any machine learning project. The second stage of project implementation is complex and involves data collection, selection, preprocessing, and transformation. Each of these phases can be split into several steps.

Data Collection

It's time for a data analyst to pick up the baton and lead the way to machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

Data Visualization

A large amount of information represented in graphic form is easier to understand and analyze. Some companies specify that a data analyst must know how to create slides, diagrams, charts, and templates.

Data Cleaning

This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with mean attributes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	itching	skin_rash	nodal_skin_itching	continuous_shivering	chills	joint_pain	stomach_acidity	ulcers_on_muscle	vomiting	burning_mouth	fatigue	weight_gain	anxiety	cold_hands_and_feet	swelling_of_joints	weight_loss	restlessness	lethargy	patches_in_skin				
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0
10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
17	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
19	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
28	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
29	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

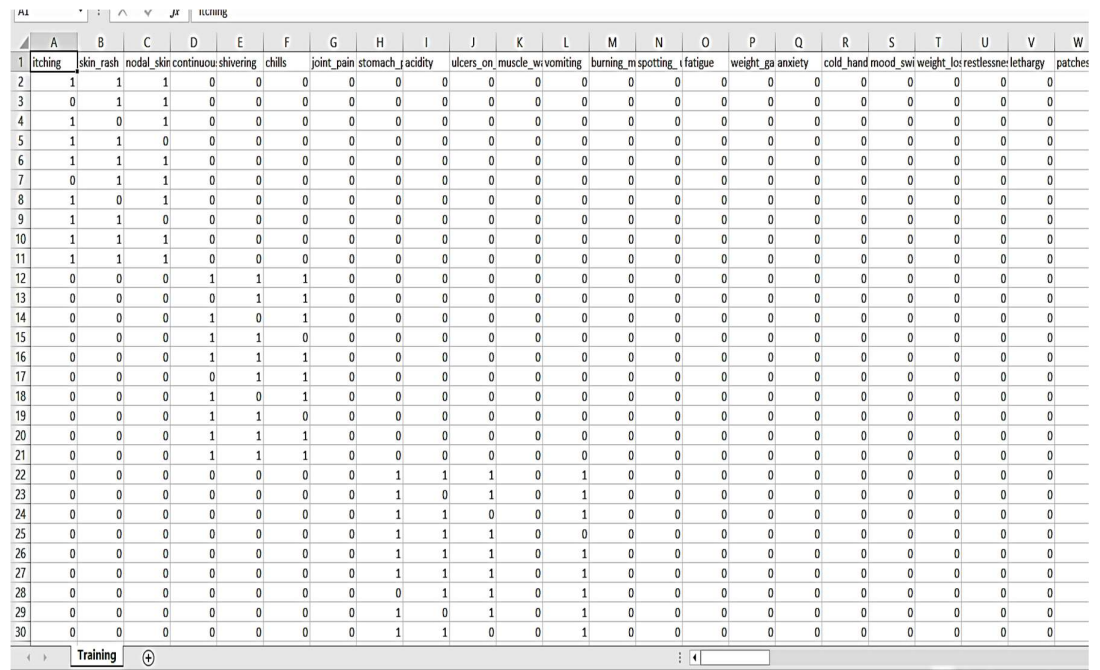
Fig. Clean Dataset

3. Dataset Splitting

A dataset used for machine learning should be partitioned into three subsets - training, test, and validation sets.

Training Set:

A data scientist uses a training set to train a model and define its optimal parameters - parameters it has to learn from data.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	itching	skin_rash	nodal_skin_	continuous	shivering	chills	joint_pain	stomach_	acidity	ulcers_on	muscle_wi	vomiting	burning_m	spotting_	fatigue	weight_ga	anxiety	cold_hand	mood_swi	weight_lo	restlessne	lethargy	patches
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Fig. Training Dataset

Testing Set:

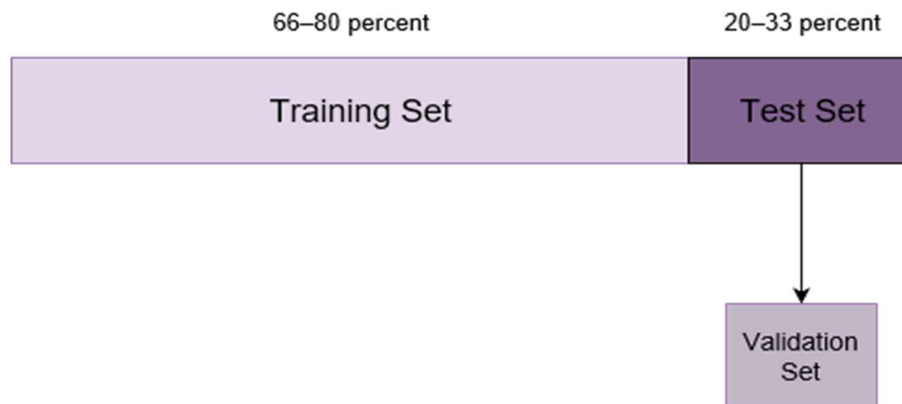
A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	itching	skin_rash	nodal_skin	continuous	shivering	chills	joint_pain	stomach_acidity	ulcers_on	muscle_wi	vomiting	burning_m	spotting	fatigue	weight_ga	anxiety	cold_hand	mood_swi	weight_lo	restlessness	lethargy	patches	
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0
17	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
19	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
20	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0
28	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
29	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. Testing Dataset

Validation Set:

The purpose of a validation set is to tweak a model's hyper parameters — higher-level structural settings that can't be directly learned from data. These settings can express, for instance, how complex a model is and how fast it finds patterns in data.



4. Modelling

During this stage, a *data scientist* trains numerous models to define which one of them provides the most accurate predictions.

Model training

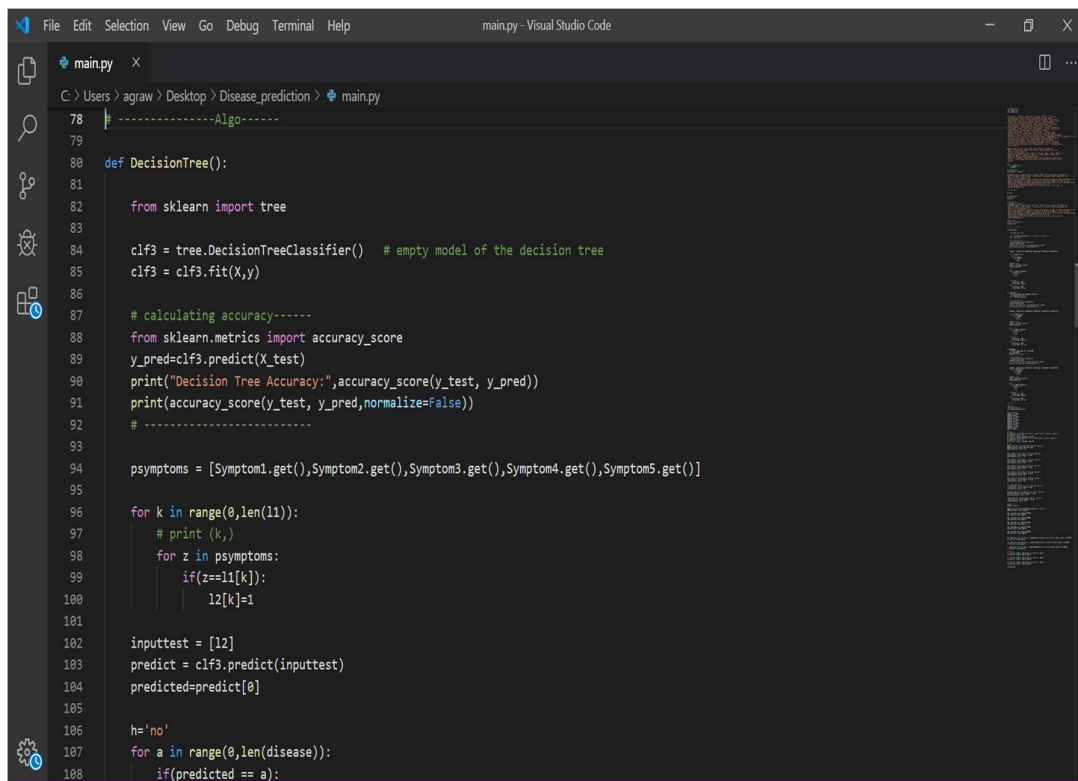
After a data scientist has preprocessed the collected data and split it into three subsets, he or she can proceed with a model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data — an answer you want to get with predictive analysis. The purpose of model training is to develop a model.

Supervised learning: Supervised learning allows for processing data with target attributes or labeled data. These attributes are mapped in historical data before the training begins. With supervised learning, a data scientist can solve classification and regression problems.

Unsupervised learning: During this training style, an algorithm analyzes unlabeled data. The goal of model training is to find hidden interconnections between data objects and structure objects by similarities or differences. Unsupervised learning aims at solving such problems as clustering, association rule learning, and dimensionality reduction. For instance, it can be applied at the data preprocessing stage to reduce data complexity.

Decision Tree Algorithm

- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

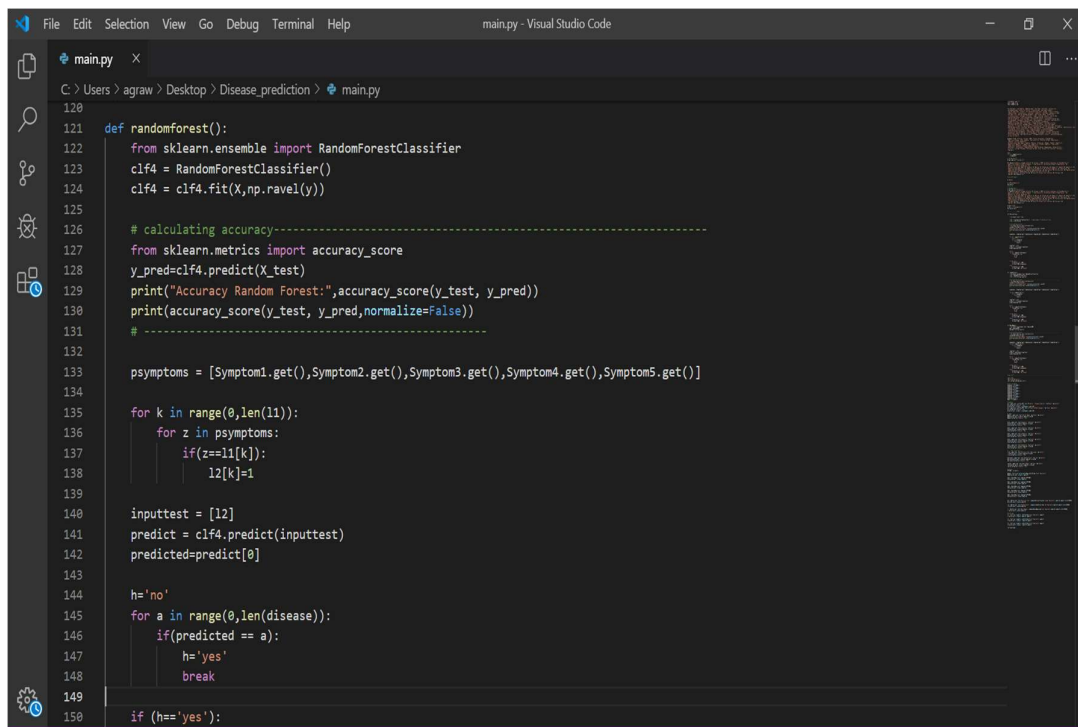


```
78 -----Algo-----
79
80 def DecisionTree():
81
82     from sklearn import tree
83
84     clf3 = tree.DecisionTreeClassifier() # empty model of the decision tree
85     clf3 = clf3.fit(X,y)
86
87     # calculating accuracy-----
88     from sklearn.metrics import accuracy_score
89     y_pred=clf3.predict(X_test)
90     print("Decision Tree Accuracy:",accuracy_score(y_test, y_pred))
91     print(accuracy_score(y_test, y_pred,normalize=False))
92     # -----
93
94     psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
95
96     for k in range(0,len(11)):
97         # print (k,)
98         for z in psymptoms:
99             if(z==11[k]):
100                 l2[k]=1
101
102     inputtest = [l2]
103     predict = clf3.predict(inputtest)
104     predicted=predict[0]
105
106     h='no'
107     for a in range(0,len(disease)):
108         if(predicted == a):
```

Fig. Decision Tree Diagram

Random Forest Algorithm

- A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as **bagging**.
- The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

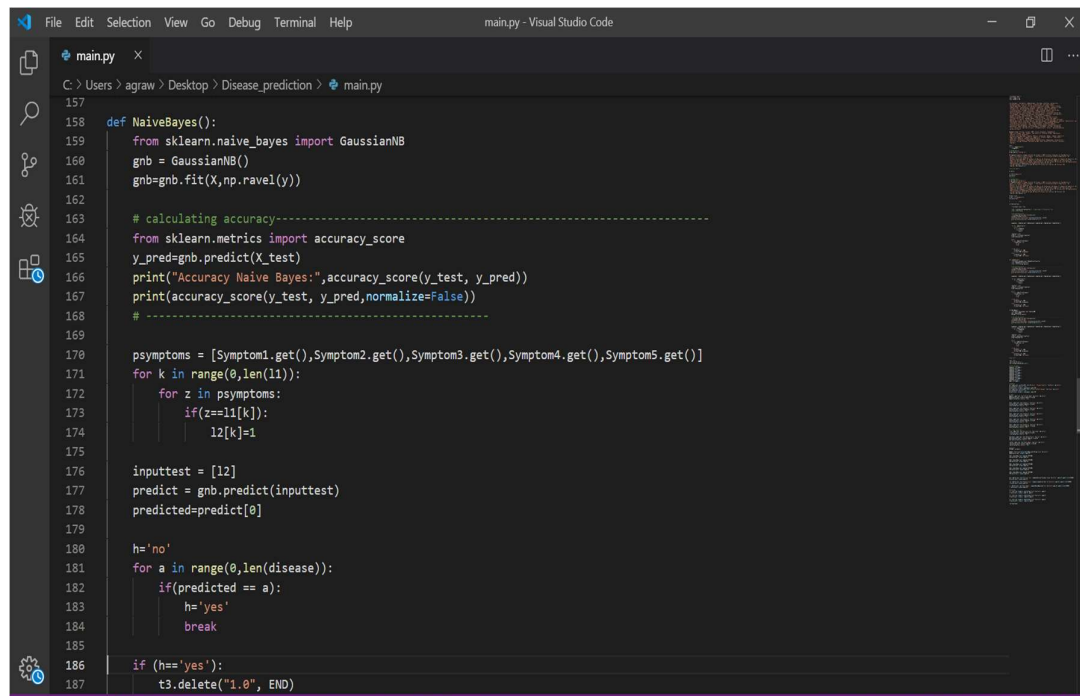


```
120
121 def randomforest():
122     from sklearn.ensemble import RandomForestClassifier
123     clf4 = RandomForestClassifier()
124     clf4 = clf4.fit(X,np.ravel(y))
125
126     # calculating accuracy-----
127     from sklearn.metrics import accuracy_score
128     y_pred=clf4.predict(X_test)
129     print("Accuracy Random Forest:",accuracy_score(y_test, y_pred))
130     print(accuracy_score(y_test, y_pred,normalize=False))
131     # -----
132
133     psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
134
135     for k in range(0,len(l1)):
136         for z in psymptoms:
137             if(z==l1[k]):
138                 l2[k]=1
139
140     inputtest = [l2]
141     predict = clf4.predict(inputtest)
142     predicted=predict[0]
143
144     h='no'
145     for a in range(0,len(disease)):
146         if(predicted == a):
147             h='yes'
148             break
149
150     if (h=='yes'):
```

Fig. Random Forest Algorithm

Naïve Bayer Algorithm

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.



```
157
158 def NaiveBayes():
159     from sklearn.naive_bayes import GaussianNB
160     gnb = GaussianNB()
161     gnb=gnb.fit(X,np.ravel(y))
162
163     # calculating accuracy-----
164     from sklearn.metrics import accuracy_score
165     y_pred=gnb.predict(X_test)
166     print("Accuracy Naive Bayes:",accuracy_score(y_test, y_pred))
167     print(accuracy_score(y_test, y_pred,normalize=False))
168     # -----
169
170     symptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
171     for k in range(0,len(11)):
172         for z in symptoms:
173             if(z==11[k]):
174                 l2[k]=1
175
176     inputtest = [l2]
177     predict = gnb.predict(inputtest)
178     predicted=predict[0]
179
180     h='no'
181     for a in range(0,len(disease)):
182         if(predicted == a):
183             h='yes'
184             break
185
186     if (h=='yes'):
187         t3.delete("1.0", END)
```

Fig. Naïve Bayer Algorithm

Module Evaluation and Testing

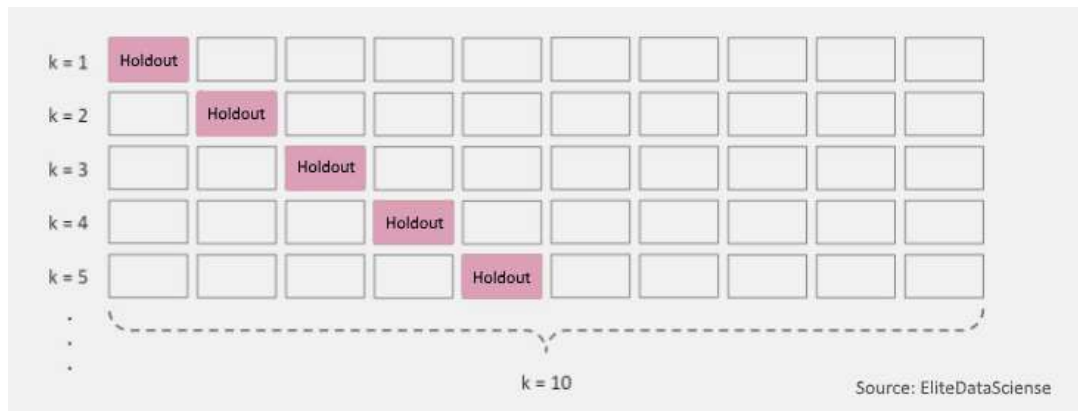
The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That's the optimization of model parameters to achieve an algorithm's best performance.

```
File Edit Selection View Go Debug Terminal Help main.py - Visual Studio Code
main.py x
C:\Users\agraw\Desktop> Disease_prediction > main.py > ...
32 'Common Cold','Pneumonia','Dimorphic hemorrhoids(piles)',
33 'Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthritis',
34 'Arthritis','(vertigo) Parosymal Positional Vertigo','Acne','Urinary tract infection','Psoriasis',
35 'Impetigo']
36
37 l2=[]
38 for x in range(0,len(11)):
39     l2.append(0)
40
41 # TESTING DATA df -----
42 df=pd.read_csv("Training.csv")
43
44 df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,
45 'Peptic ulcer disease':5,'AIDS':6,'Diabetes':7,'Gastroenteritis':8,'Bronchial Asthma':9,'Hypertension':10,
46 'Migraine':11,'Cervical spondylosis':12,
47 'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
48 'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':25,
49 'Common Cold':26,'Pneumonia':27,'Dimorphic hemorrhoids(piles)':28,'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,
50 'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthritis':34,'Arthritis':35,
51 '(vertigo) Parosymal Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Psoriasis':39,
52 'Impetigo':40}},inplace=True)
53
54 # print(df.head())
55
56 X= df[11]
57
58 y = df[["prognosis"]]
59 np.ravel(y)
60 # print(y)
61
62 # TRAINING DATA tr -----
```

```
File Edit Selection View Go Debug Terminal Help main.py - Visual Studio Code
main.py x
C:\Users\agraw\Desktop> Disease_prediction > main.py > ...
13 'swollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_tingling_lips',
14 'slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swelling_joints',
15 'movement_stiffness','spinning_movements','loss_of_balance','unsteadiness',
16 'weakness_of_one_body_side','loss_of_smell','bladder_discomfort','foul_smell_of_urine',
17 'continuous_feel_of_urine','passage_of_gases','internal_itching','toxic_look(typhos)',
18 'depression','irritability','muscle_pain','altered_sensorium','red_spots_over_body','belly_pain',
19 'abnormal_menstruation','dischromic_patches','watering_from_eyes','increased_appetite','polyuria','family_history','mucoid_sputum',
20 'rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_transfusion',
21 'receiving_unsterile_injections','coma','stomach_bleeding','distention_of_abdomen',
22 'history_of_alcohol_consumption','fluid_overload','blood_in_sputum','prominent_veins_on_calf',
23 'palpitations','painful_walking','pus_filled_pimples','blackheads','scurring','skin_peeling',
24 'silver_like_dusting','small_dents_in_nails','inflammatory_nails','blister','red_sore_around_nose',
25 'yellow_crust_ooze']
26
27 disease=['Fungal infection','Allergy','GERD','Chronic cholestasis','Drug Reaction',
28 'Peptic ulcer disease','AIDS','Diabetes','Gastroenteritis','Bronchial Asthma','Hypertension',
29 'Migraine','Cervical spondylosis',
30 'Paralysis (brain hemorrhage)','Jaundice','Malaria','Chicken pox','Dengue','Typhoid','hepatitis A',
31 'Hepatitis B','Hepatitis C','Hepatitis D','Hepatitis E','Alcoholic hepatitis','Tuberculosis',
32 'Common Cold','Pneumonia','Dimorphic hemorrhoids(piles)',
33 'Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthritis',
34 'Arthritis','(vertigo) Parosymal Positional Vertigo','Acne','Urinary tract infection','Psoriasis',
35 'Impetigo']
36
37 l2=[]
38 for x in range(0,len(11)):
39     l2.append(0)
40
41 # TESTING DATA df -----
42 df=pd.read_csv("Training.csv")
43
```

Cross-validation:

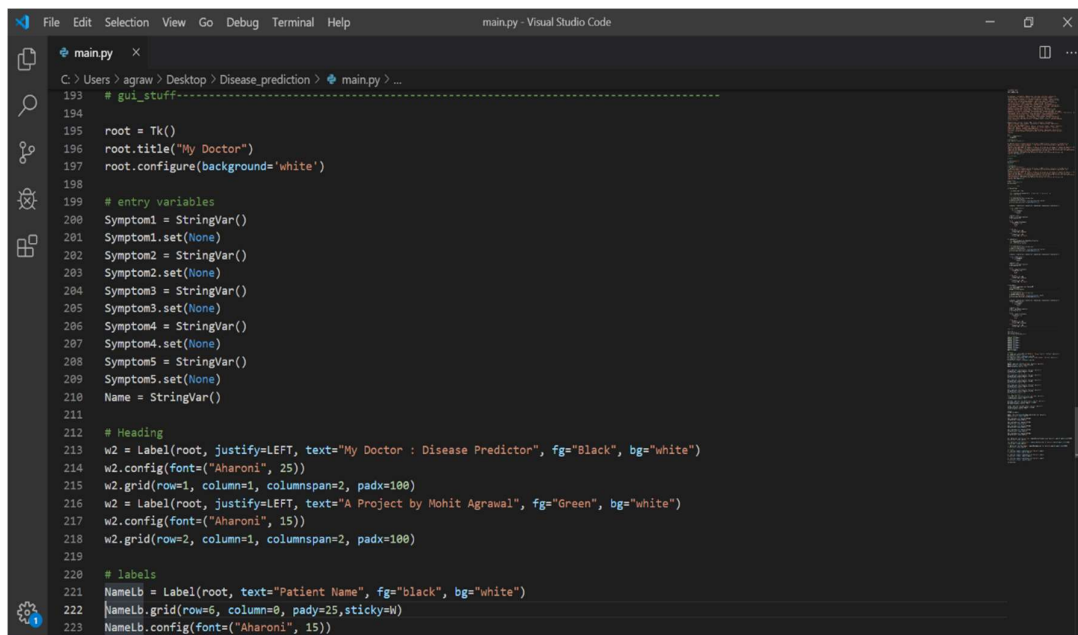
Cross-validation is the most commonly used tuning method. It entails splitting a training dataset into ten equal parts (folds). A given model is trained on only nine folds and then tested on the tenth one (the one previously left out). Training continues until every fold is left aside and used for testing. As a result of model performance measure, a specialist calculates a cross-validated score for each set of hyper parameters. A data scientist trains models with different sets of hyper parameters to define which model has the highest prediction accuracy. The cross-validated score indicates average model performance across ten hold-out folds.



5. Model Deployment

The model deployment stage covers putting a model into production use.

Once a *data scientist* has chosen a reliable model and specified its performance requirements, he or she delegates its deployment to a *data engineer* or *database administrator*. The distribution of roles depends on your organization's structure and the amount of data you store.



```
193 # gui_stuff-----
194
195 root = Tk()
196 root.title("My Doctor")
197 root.configure(background='white')
198
199 # entry variables
200 Symptom1 = StringVar()
201 Symptom1.set(None)
202 Symptom2 = StringVar()
203 Symptom2.set(None)
204 Symptom3 = StringVar()
205 Symptom3.set(None)
206 Symptom4 = StringVar()
207 Symptom4.set(None)
208 Symptom5 = StringVar()
209 Symptom5.set(None)
210 Name = StringVar()
211
212 # Heading
213 w2 = Label(root, justify=LEFT, text="My Doctor : Disease Predictor", fg="Black", bg="white")
214 w2.config(font=("Aharoni", 25))
215 w2.grid(row=1, column=1, columnspan=2, padx=100)
216 w2 = Label(root, justify=LEFT, text="A Project by Mohit Agrawal", fg="Green", bg="white")
217 w2.config(font=("Aharoni", 15))
218 w2.grid(row=2, column=1, columnspan=2, padx=100)
219
220 # labels
221 NameLb = Label(root, text="Patient Name", fg="black", bg="white")
222 NameLb.grid(row=6, column=0, pady=25, sticky=W)
223 NameLb.config(font=("Aharoni", 15))
```

TKINTER in Python

Create GUI Window

```
# gui_stuff-----  
-----  
  
root = Tk()  
root.title("My Doctor")  
root.configure(background='white')
```

Heading in Window

```
# Heading  
w2 = Label(root, justify=LEFT, text="My Doctor : Disease Predictor", fg="Black", bg="white")  
w2.config(font=("Aharoni", 25))  
w2.grid(row=1, column=1, columnspan=2, padx=100)  
w2 = Label(root, justify=LEFT, text="A Project by Mohit Agrawal", fg="Green", bg="white")  
w2.config(font=("Aharoni", 15))  
w2.grid(row=2, column=1, columnspan=2, padx=100)
```

Create Levels for Symptoms

```
# labels  
NameLb = Label(root, text="Patient Name", fg="black", bg="white")  
NameLb.grid(row=6, column=0, pady=25, sticky=W)  
NameLb.config(font=("Aharoni", 15))  
  
S1Lb = Label(root, text="Symptom 1", fg="black", bg="white")  
S1Lb.grid(row=7, column=0, pady=20, sticky=W)  
S1Lb.config(font=("Aharoni", 15))
```

```

S2Lb = Label(root, text="Symptom 2", fg="black", bg="white")
S2Lb.grid(row=8, column=0, pady=20, sticky=W)
S2Lb.config(font=("Aharoni", 15))

S3Lb = Label(root, text="Symptom 3", fg="black", bg="white")
S3Lb.grid(row=9, column=0, pady=20, sticky=W)
S3Lb.config(font=("Aharoni", 15))

S4Lb = Label(root, text="Symptom 4", fg="black", bg="white")
S4Lb.grid(row=10, column=0, pady=20, sticky=W)
S4Lb.config(font=("Aharoni", 15))

S5Lb = Label(root, text="Symptom 5", fg="black", bg="white")
S5Lb.grid(row=11, column=0, pady=20, sticky=W)
S5Lb.config(font=("Aharoni", 15))

```

List View

```

# entries
OPTIONS = sorted(11)

NameEn = Entry(root, textvariable=Name, width=50, bg="black", fg="white")
NameEn.grid(row=6, column=1, padx=10)

S1En = OptionMenu(root, Symptom1, *OPTIONS)
S1En.grid(row=7, column=1, padx=10)

S2En = OptionMenu(root, Symptom2, *OPTIONS)
S2En.grid(row=8, column=1, padx=10)

S3En = OptionMenu(root, Symptom3, *OPTIONS)
S3En.grid(row=9, column=1, padx=10)

S4En = OptionMenu(root, Symptom4, *OPTIONS)
S4En.grid(row=10, column=1, padx=10)

S5En = OptionMenu(root, Symptom5, *OPTIONS)
S5En.grid(row=11, column=1, padx=10)

```

Button

```
dst = Button(root, text="Decision Tree", command=DecisionTree,bg="orange",
fg="white", padx=10, pady=5,relief=RIDGE)
dst.grid(row=8, column=2,padx=10)

rnf = Button(root, text="Random Forest", command=randomforest,bg="red",fg=
"white",padx=10, pady=5,relief=RIDGE)
rnf.grid(row=9, column=2,padx=10)

lr = Button(root, text="Naive Bayes", command=NaiveBayes,bg="blue",fg="whi
te",padx=10, pady=5,relief=RIDGE)
lr.grid(row=10, column=2,padx=10)
```

Text Fields

```
#textfileds
t1 = Text(root, height=1, width=40,bg="black",fg="white", pady=5)
t1.grid(row=15, column=1, padx=10, pady=5)

t2 = Text(root, height=1, width=40,bg="black",fg="white", pady=5)
t2.grid(row=17, column=1 , padx=10, pady=5)

t3 = Text(root, height=1, width=40,bg="black",fg="white", pady=5)
t3.grid(row=19, column=1 , padx=10, pady=5)
```


Disease Predictor Prototype

- This Machine Learning project is used to predict the disease based on the symptoms given by the user. So, the output is accurate.
- The patient can fill up to 5 symptoms and based on these symptoms Machine Learning will predict disease.
- It predicts disease by using **three different machine learning algorithms**.
- It uses **tkinter** for GUI and **Numpy, Pandas** for data mining.

My Doctor

My Doctor : Disease Predictor

A Project by Mohit Agrawal

Patient Name

Symptom 1

Symptom 2

Symptom 3

Symptom 4

Symptom 5

Decision Tree

Random Forest

Naive Bayes

Conclusion

During my two months of summer internship at Dzone Software Solution and Service Provider, I have gained the exposure of the real working environment of a company and learned how the things work in real life. I have received the exposure of the company world.

As I have done my summer internship in Machine Learning with Python, I have learnt a lot about this technology and there is a lot more to be learned in this technology. There is a lot of stuff that can be done using this technology. In the training period, I have gone through an intermediate level of developing an android app and there is a lot more to be explored.

Bibliography

The various information is taken from the following sources

<https://www.javatpoint.com/>

<https://www.dzone.co.in>

<https://www.kaggle.com/datasets>

<https://www.wikipedia.org/>

<https://www.youtube.com/>