MALIGNANT TUMOR CELL DETECTION

Mini Project Report

Submitted in partial fulfillment of the Requirements for the award of degree

Of Bachelor of Technology (B.TECH)

in

COMPUTER SCIENCE AND ENGINEERING

By

P.SHIVA SINDHU (18AG1A0546)

Under the Esteemed Guidance of Mr. YOUNUS SHARIFF

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

ACE ENGINEERING COLLEGE

An Autonomous Institution

(NBAACCREDITEDB.TECHCOURSES:EEE,ECE,MECH,CIVIL&CSE, ACCORDEDNAAC'A'GRADE

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telengana)

Ghatkesar, Hyderabad-501 301

February 2022

ACE

Engineering College

An Autonomous Institution

(NBAACCREDITEDB.TECHCOURSES:EEE,ECE,MECH,CIVIL&CSE,ACCORDEDNAAC'A'GRADE)

Ghatkesar,Hyderabad5501301

(AffiliatedtoJawaharlalNehruTechnologicalUniversityHyderabad)

Website:www.aceec.ac.in E-mail:info@aceec.ac.in

CERTIFICATE

This is to certify that the Mini project entitled **"Malignant Tumor Cell Detection"** being submitted by **P.SHIVA SINDHU(18AG1A0546)** in partial fulfillment of the requirements for the award of the degree of the **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** to the Jawaharlal Nehru Technological University, Hyderabad during the Academic year 2021-22 is a record of bonafide work carried out by her/him under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any other degree or diploma.

Internal Guide
Mr. Younus Shariff
(Assistant Professor)

Head of the Department
Dr.M.V.VijayaSaradhi
Professor and Head
Dept of CSE

External Examiner

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who have helped me transform an idea into real time application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfil my dreams.

A special thanks to our secretary, **Prof.Y.V.GOPALA KRISHNA MURTHY,** for having founded such an esteemed institution. I am also greatful to our beloved principal, **Dr.B.L.RAJU** for permitting us to carry out this project.

I profoundly thank **Dr.M.V.VijayaSaradhi,** Head of the Department of Computer Science and Engineering, who has been an excellent guide and also a great source of inspiration to my work.

I extremely thank **Mr.K.Prem Kumar** Assistant Professor, Project coordinator, who helped us in all the way in fulfilling of all aspects in completion of our Mini-Project.

I am very thankful to my internal guide **Mr. Younus Shariff** , **Assistant Professor** of the Department of Computer Science and Engineering who has been an excellent and given continuous support for the Completion of my project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

P.Shiva Sindhu (18AG1A0546)

DECLARATION

I hereby declare that the project work entitled "**Malignant Tumor Cell Detection**" submitted to the **ACE ENGINEERING COLLEGE** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B.Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mr. Younus Shariff, Assistant Professor** and this project work have not been submitted to any other university for the award of any other degree or diploma.

P.Shiva Sindhu (18AG1A0546)

ABSTRACT

Cancer is the second leading cause of death globally and accounted for 8.8 million deaths in 2015. Most of the people fail to detect their disease before it becomes chronic. It leads to increase in death rate around the world. Breast cancer is one of the diseases that could be cured when the disease identified at earlier stages before it is spreading across all the parts of the body. The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients. For better clinical decisions, it is important to accurately distinguish between benign and malignant tumors.

Women are seriously threatened by breast cancer with high morbidity and mortality. The lack of robust prognosis models results in difficulty for doctors to prepare a treatment plan that may prolong patient survival time. Hence, the requirement of time is to develop the technique which gives minimum error to increase accuracy. Aim of research categorises in three domains. First domain is prediction of cancer before diagnosis, second domain is prediction of diagnosis and treatment and third domain focuses on outcome during treatment. The proposed work can be used to predict the outcome of different technique and suitable technique can be used depending upon requirement. This research is carried out to predict the accuracy. The future research can be carried out to predict the other different parameters and breast cancer research can be categorises on basis of other parameters.

CONTENTS

1. INTRODUCTION

Breast cancer occurs in breast cells, the fatty tissue or the fibrous connective tissue within the breast. A tumor can be benign or malignant. A tumor has the potential to spread to other parts of the body. Factors such as age and a family history of breast cancer can increase the risk of breast cancer.

Types and stages of breast cancer: The treatment options for breast cancer are based on cancer stage and type. The status of the cell surface receptors determines breast cancer classification**.**

Two types of tumors are identified:

Benign: this tumor type is not dangerous for a human body and rarely causes human death. In this type, the tumor grows in one part (spot) of the body and has limited growth.

Malignant: this tumor type is more dangerous and causes human death, it is called breast cancer. The malignant tumor develops when cells in the breast tissue abnormally grow. The main types of breast cancer include:

Ductal carcinoma in situ (DCIS): is the earliest form of breast cancer and is curable.

Invasive Ductal Carcinoma (IDC): begin in the milk duct and is the most common breast cancer.

Invasive Lobular Carcinoma (ILC): start in a lobule of the breast. It has the ability to spread fast to the lymph nodes and other areas of the body.

Treatment of breast cancer: Sometimes, patients treated with one of the treatments or combination of the treatments which are based on the woman's age, type and stage of cancer.

The main treatments for breast cancer are

Surgery: There are two main types of surgery for breast cancer.

1.1. MOTIVATION

Being the most frequently occurring cancer in women, breast cancer affects around 10% of women at some point in their life. An early detection of cancerous cells could help medical experts to identify appropriate treatment plan, thus saving many lives. The main motivation behind the project is to provide a model that results in an accurate prediction of malignancy of tumor cells, which would facilitate early treatment.

1.2. PROBLEM DEFINITION

As such, there is no prevention mechanism for Breast Cancer, but early detection can significantly improve the outcome. Further, this can also considerably reduce the costs of the treatment. This would help medical experts provide essential diagnosis for it. It is indispensable to employ mammograms and self-breast tests to detect any early irregularities before the tumor gets advanced.

1.3 OBJECTIVE OF PROJECT

The key objective of this project is to create a model for cancer detection that presents the highly accurate and efficient results. n followed by a similar artificial neural network with a complex network structure which has ‘n’ hidden layers, which can process the input data from the previous layer. The error rate of the input data will be consistently reduced by adjusting the weights of every node, which leads to achieve an accurate result.

2.LITERATURE SURVEY

2.1 Using Feature Selection Techniques to Improve the Accuracy of Breast Cancer Classification

Authors – Hajar Saoud, AbderrahimGhadi, M. Ghailani, BoudhirAnouarAbdelhakim

Classification is a data mining process that aims to divide data into classes to facilitate decision-making; it is therefore an important task in medical field. In this paper we will try to improve the accuracy of the classification of six machines learning algorithms: Bayes Network (BN), Support Vector Machine (SVM), k-nearest neighbors algorithm (Knn), Artificial Neural Network (ANN), Decision Tree (C4.5) and Logistic Regression using feature selection techniques, for breast cancer classification and diagnosis. We examined those methods of classification and techniques of feature selection in WEKA Tool (The Waikato Environment for Knowledge Analysis) using two databases, Wisconsin breast cancer datasets original (WBC) and diagnostic (WBCD) available in UCI machine learning repository.

2.2 Feature Selection Using Correlation Analysis and Principal Component Analysis

Authors – Sara Ibrahim, Saima Nazir, Sergio A. Velastin

Breast cancer is one of the leading causes of death among women, more so than all other cancers. The accurate diagnosis of breast cancer is very difficult due to the complexity of the disease, changing treatment procedures and different patient population samples. Diagnostic techniques with better performance are very important for personalized care and treatment and to reduce and control the recurrence of cancer. The main objective of this research was to select feature selection techniques using correlation analysis and variance of input features before passing these significant features to a classification method. We used an ensemble method to improve the classification of breast cancer. The proposed approach was evaluated using the public

WBCD dataset (Wisconsin Breast Cancer Dataset). Correlation analysis and principal component analysis were used for dimensionality reduction. Performance was evaluated for well-known machine learning classifiers, and the best seven classifiers were chosen

for the next step.Hyper-parameter tuning was performed to improve the performances of the classifiers. The best performing classification algorithms were combined with two different voting techniques. Hard voting predicts the class that gets the majority vote, whereas soft voting predicts the class based on highest probability. The proposed approach performed better than state-of-the-art work, achieving an accuracy of 98.24%, high precision (99.29%) and a recall value of 95.89%.

2.3 Breast Cancer Classification and Prediction using Machine Learning

Authors – Nikita Rane, Rucha Kanade, Jean Sunny, Prof. Sulochana Devi

Breast cancer is a dominant cancer in women worldwide and is increasing in developing countries where the majority of cases are diagnosed in late stages. The projects that have already been proposed show a comparison of machine learning algorithms with the help of different techniques like the ensemble methods, data mining algorithms or using blood analysis etc. This paper proposed now presents a comparison of six machine learning (ML) algorithms: Naive Bayes (NB), Random Forest (RT), Artificial Neural Networks (ANN), Nearest Neighbour (KNN), Support Vector Machine (SVM) and Decision Tree (DT) on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset which is extracted from a digitised image of an MRI. For the implementation of the ML algorithms, the dataset was partitioned into the training phase and the testing phase. The algorithm with the best results will be used as the backend to the website and the model will then classify the cancer as benign or malignant.

2.4 Deep Learning in Breast Cancer Detection and Classification

Authors - GhadaHamed , Mohammed Abd El-Rahman Marey, Safaa El-Sayed Amin, and Mohamed Fahmy Tolba

Breast cancer is considered one of the primary causes of mortality among women aged 20–59 worldwide. Early detection and treatment can allow patients to have proper treatment and consequently reduce rate of morbidity of breast cancer. Research indicates that most experienced physicians can diagnose cancer with 79% accuracy while 91% correct diagnosis is achieved using machine learning techniques. In this paper, we present the most recent breast cancer detection and classification models that are machine learning based models by analyzing them in the form of comparative study. Also, in this

paper, the datasets that are public for use and popular as well are listed in the recent work to facilitate any new experiments and comparisons. The comparative analysis shows that the recent highest accuracy models based on simple detection and the classification architectures are You Only Look Once (YOLO) and RetinaNet.

2.5 Breast Cancer Classification using Deep Convolutional Neural Network

Authors – Muhammad Aqeel Aslam, Aslam and Daxiang Cui

Over the last decade, the demand for early diagnosis of breast cancer has resulted in new research avenues. According to the world health organization (WHO), a successful treatment plan can be provided to individuals suffering from breast cancer once the non-communicable disease is diagnosed at an early stage. An early diagnosis of cure disease can reduce mortality all over the world. Computer-Aided Diagnosis (CAD) tools are widely implemented to diagnose and detect different kinds of abnormalities. In the last few years, the use of the CAD system has become common to increase the accuracy in different research areas. The CAD systems have minimum human intervention and producing accurate results. In this study, we proposed a CAD technique for the diagnosis of breast cancer using a Deep Convolutional Neural Network followed by Softmax classifier. The proposed technique was tested on the Wisconsin Breast Cancer Datasets (WBCD). The proposed classifier produced an accuracy of 100% and 99.1% for two different datasets, which indicates effective diagnostic capabilities & promising results. Moreover, we test our proposed architecture with different train-test partitions.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Naïve Bayes Classifier produces ill results when the training data is not represented . The SVM classifier is unsuitable for large datasets and also not effective on high computer vision applications. When the data is imbalanced, Bi-clustering and Ada boost Techniques will lead to erroneous classification. RCNN takes more time to train the network.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- Its accuracy result is very less.
- It takes more time to load image dataset.

3.2 PROPOSED SYSTEM:

- We proposed breast cancer detection and classification system .Then, a comparison between the selected recent proposed approaches in the breast cancer detection and classification from the perspective of the data preparation done in each. Finally, a comparative analysis is proposed to show the detection and classification models used and their performance.
- The key objective of this paper is to propose a novice method to detect BC. This paper presents a detailed study of existing cancer detection models and presents the highly accurate and efficient results.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

The proposed method has produced highly accurate and efficient results when compared to the existing methods.

3.3 SYSTEM REQUIREMENTS:

3.3.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- Python IDLE 3.7 version (or)
- Anaconda 3.7 (or)
- Jupyter (or)
- Google Collab

HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- Operating system : Windows, Linux
- Processor : Intel i3 (minimum)
- RAM : 4 GB (minimum)
- Hard disk : 250 GB (minimum)

3.4 FUNCTIONALREQUIREMENTS

- Data collection
- Data preprocessing
- Training and Testing

- Model building
- Predicting

3.5 NON FUNCTIONALREQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are greater than 10000. Description of non-functional requirements is just as critical as a functionalrequirement.

- Usabilityrequirement
- Serviceabilityrequirement
- Manageabilityrequirement
- Recoverabilityrequirement
- Securityrequirement
- Data Integrity requirement
- Capacityrequirement
- Availabilityrequirement
- Scalabilityrequirement
- Interoperabilityrequirement
- Reliabilityrequirement
- Maintainabilityrequirement

- Regulatoryrequirement
- Environmentalrequirement

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

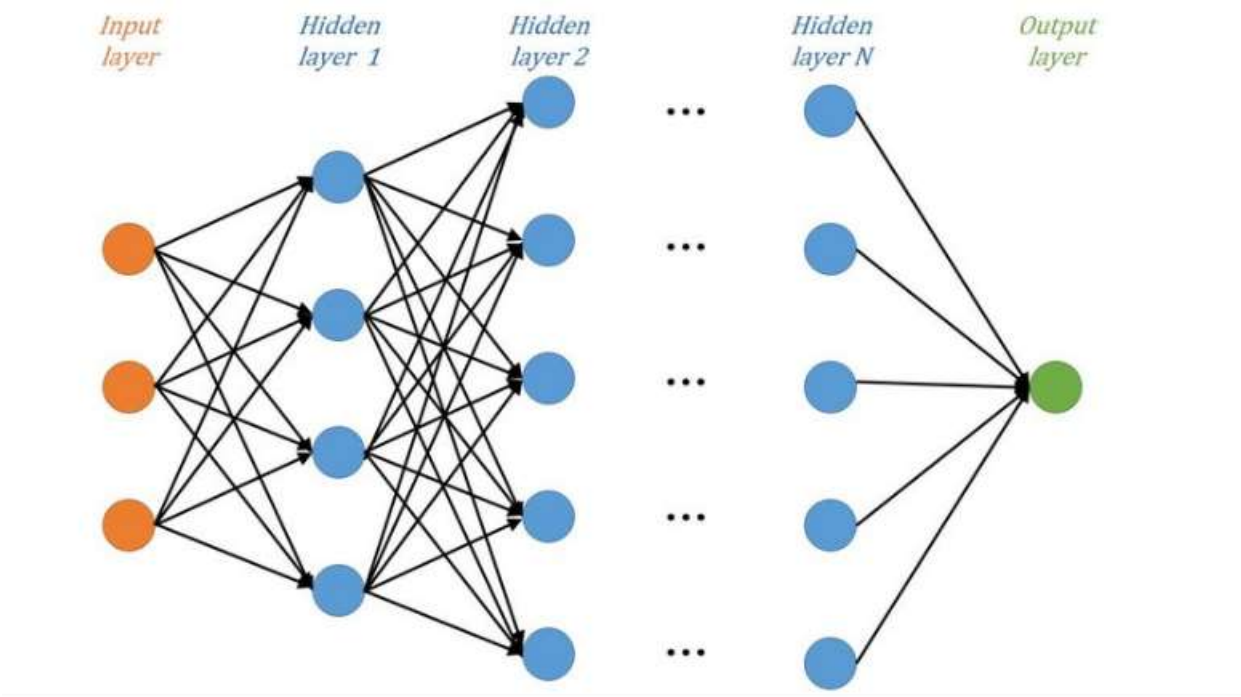


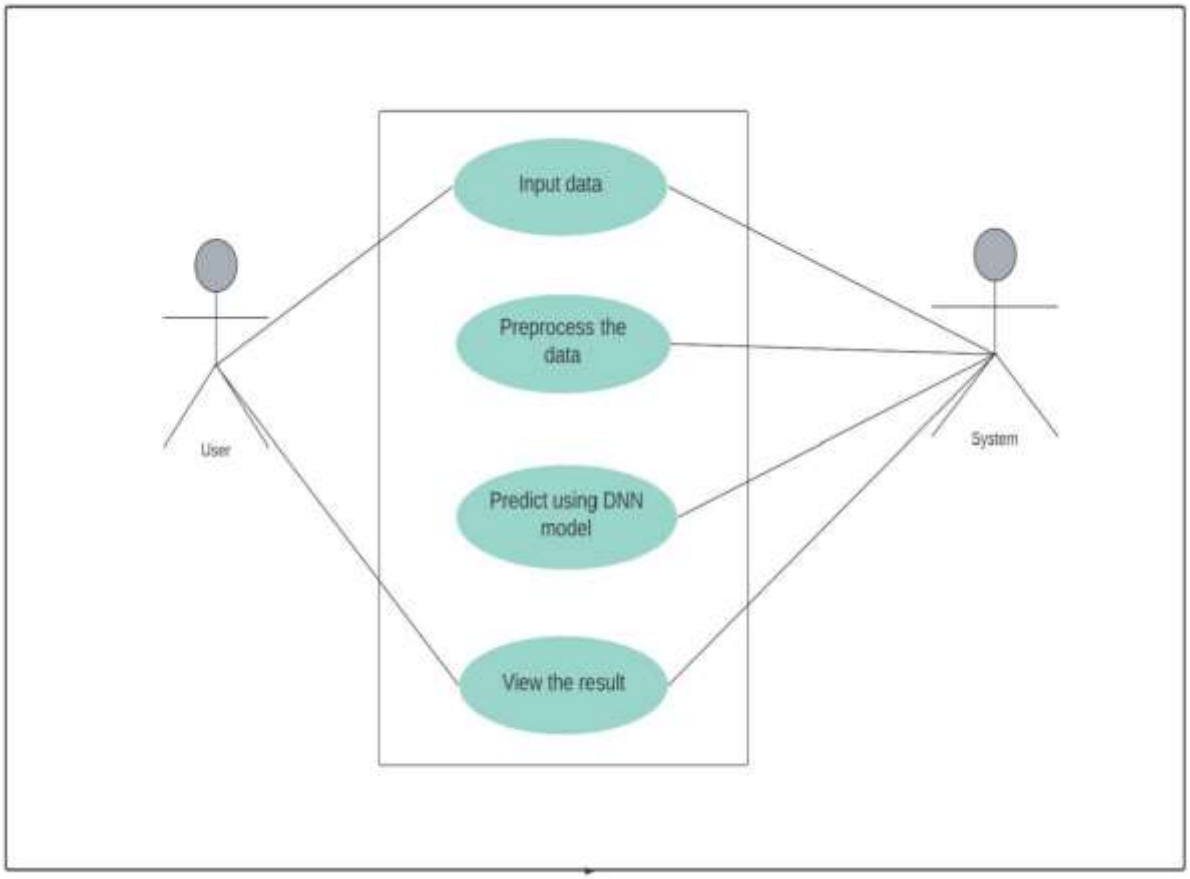
Figure 4.1.1

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

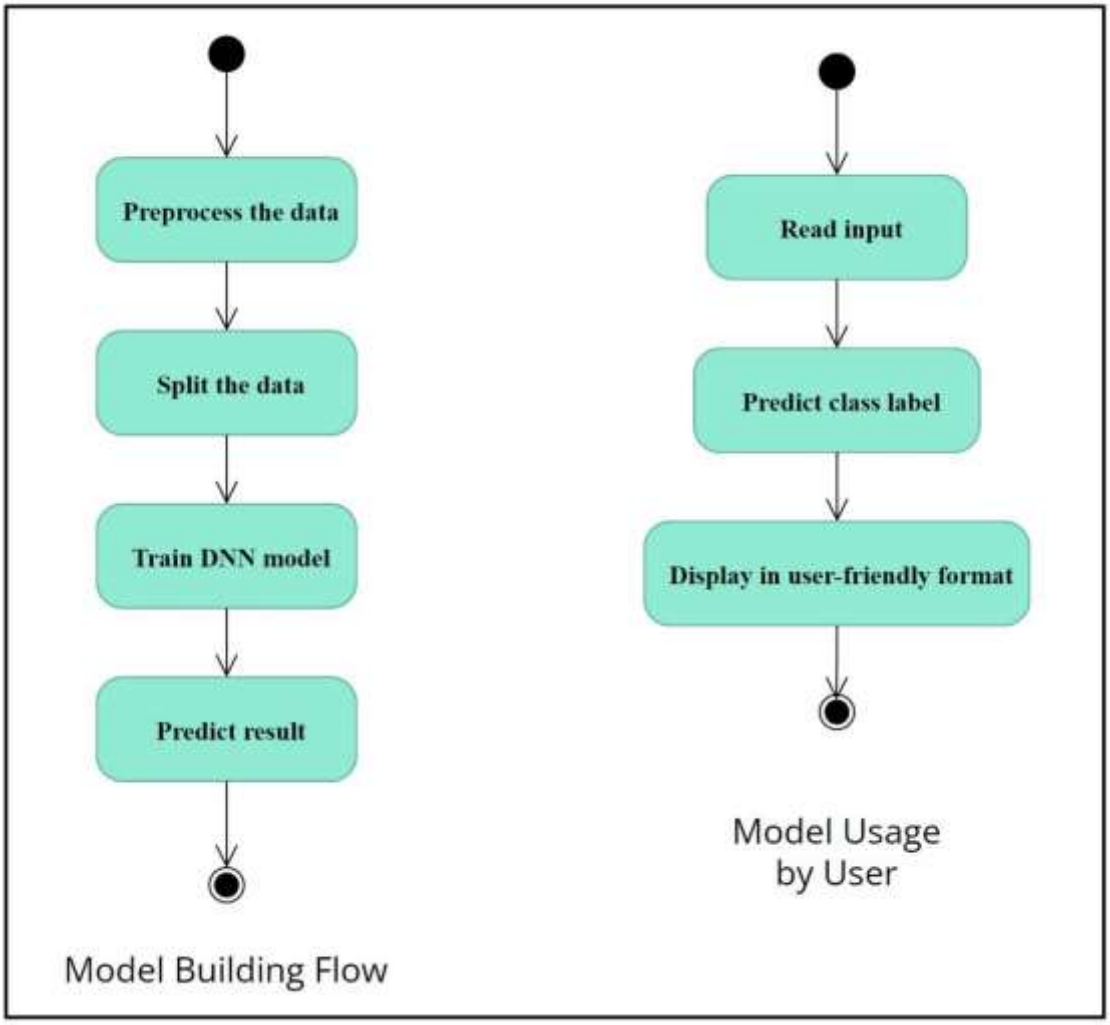
The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Activity diagram

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.


Preprocess the data
Split the data
Train DNN model
Predict result
Model Building Flow
Read input
Predict class label
Display in user-friendly format
Model Usage by User

5. IMPLEMENTATION

5.1 MODULES

- Load Data
- Data Preprocessing
- Spilt Data
- Build NN Model
- Upload Test Data
- Predict Result

5.2 MODULE DESCRIPTION

Python

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like –Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQtetc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contains libraries for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ and also the fact that you can write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Advantages of Python over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compare to more widely used technologies like JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

Python's simplicity can indeed be a problem. Its syntax is so simple that the verbosity of other languages like Java code seems unnecessary. It becomes tough for a python programmer to learn and get accustomed to other verbose languages if and when need

arises. This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde&Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voorWiskundeen Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Machine Learning

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories of Machine Learning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then

the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialists − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term “Machine Learning” in 1959 and defined it as a “Field of study that gives computers the capability to learn without being explicitly programmed”.

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine

Learning Engineer Is the Best Job of 2019 with a 344% growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So, this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

- Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

- Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

- Training – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

- Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

- Unsupervised Learning – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

- Semi-supervised Learning – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

- Reinforcement Learning – This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be a business owner or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dictionary, string and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

•Print is now a function

•Views and iterators instead of lists

- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.

•There is only one integer type left, i.e., int. long is int as well.

•The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.

•Text Vs. Data Instead of Unicode Vs. 8-bit

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people

5.3. INTRODUCTION OF TECHNOLOGIES USED

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

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.

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.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python Installation

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheethere.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

1. Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Figure 5.3.1

Now, check for the latest and the correct version for your operating system.

2. Click on the Download Tab.

Figure 5.3.2

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

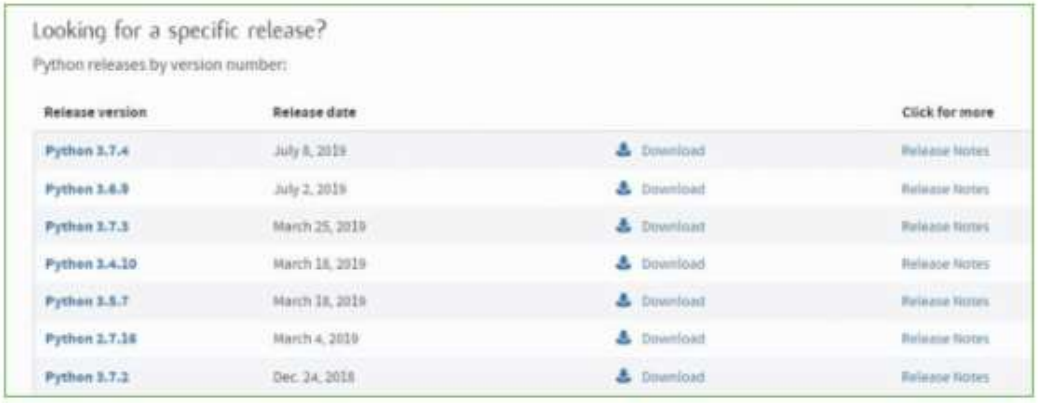


Fig 5.3.3

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Fig 5.3.4

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

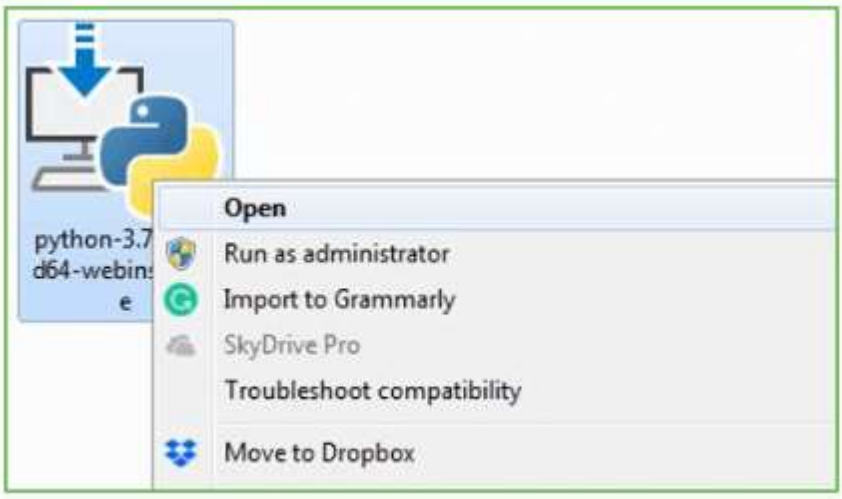


Fig 5.3.5

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

Fig 5.3.6

Step 3: Click on Install NOW After the installation is successful. Click on close.

Fig 5.3.7

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

Fig 3.7.8

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.

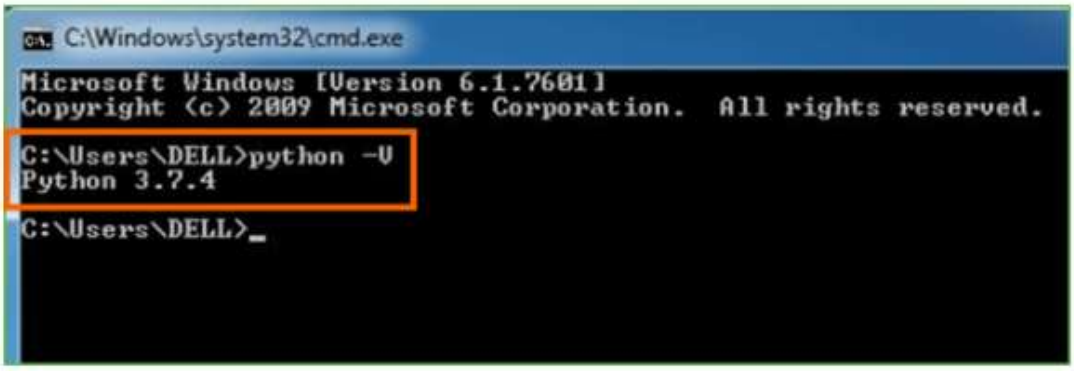


Fig 5.3.9

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

Fig 5.3.10

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save

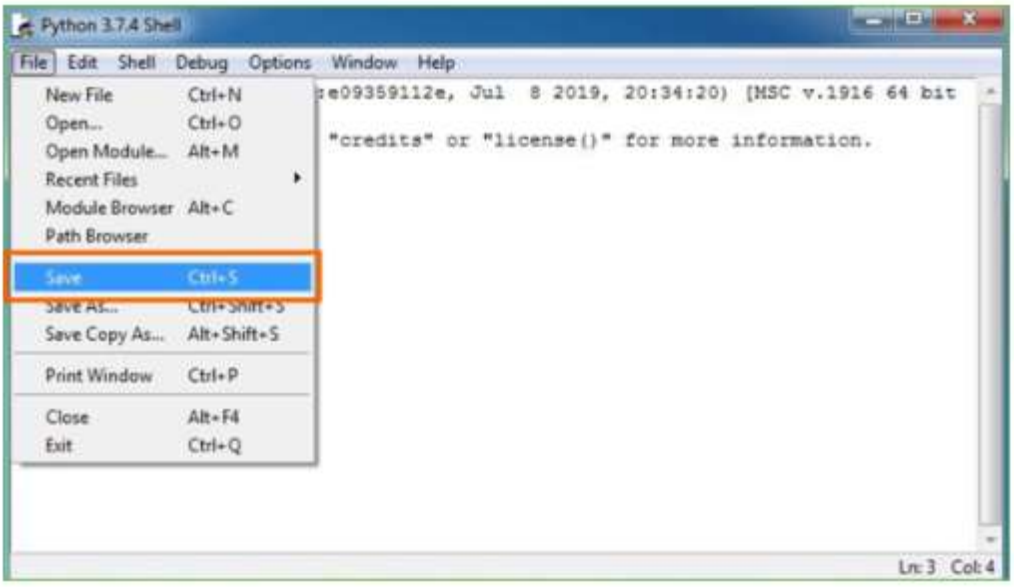


Fig 5.3.11

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g., enter print

5.4 SAMPLE CODE

```
[301] #LOAD AND PREPROCESS THE DATA

cancer = load_breast_cancer()

cancerdf = pd.concat([pd.DataFrame(cancer.data), pd.DataFrame(cancer.target)], axis=1)
cancerdf.columns = np.append(cancer.feature_names,'target')

X = cancerdf.drop('target', axis = 1) # All columns but the last one.
X = normalize(X)
drop_list = ['mean perimeter','mean radius','mean compactness','mean concave points','radius error','perimeter error','worst radius',
             'worst perimeter','worst compactness','worst concave points','compactness error','concave points error','worst texture','worst area']
X = X.drop(drop_list, axis = 1)

y = cancerdf['target'] # Just the last column.
```

```
#SPLIT THE DATA

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
print(X_train.head(5))
#cancerdf.columns
#X.describe()
```

```
     mean texture  mean area  ...  worst symmetry  worst fractal dimension
338     -0.409122  -0.977764  ...       -0.010920                -0.404502
427      0.625512  -0.838242  ...        0.103842                -0.405609
406     -1.029903   0.412347  ...       -0.198417                -0.765495
96      -0.337046  -0.579088  ...       -1.089031                -0.563959
490      0.732463  -0.535327  ...        0.322050                -0.106073

[5 rows x 16 columns]
```

Figure 5.4.1

```
[241] #DEFINE THE DNN ARCHITECTURE
      reset_random_seeds()
      model = keras.Sequential([
          keras.layers.Flatten(input_shape=(16,)),  # input layer
          keras.layers.Dense(10, activation='relu'),  # hidden layer (1)
          keras.layers.Dense(20, activation='relu'),  # hidden layer (2)
          keras.layers.Dense(10, activation='relu'),  # hidden layer (3)
          keras.layers.Dense(1, activation='sigmoid') # output layer
      ])
```

```
#CREATE THE MODEL
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
[243] #TRAIN THE MODEL
      model.fit(X_train, y_train, epochs=86)

      Epoch 1/86
      15/15 [==============================] - 1s 2ms/step - loss: 0.6882 - accuracy: 0.5890
      Epoch 2/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.6412 - accuracy: 0.7824
      Epoch 3/86
      15/15 [==============================] - 0s 3ms/step - loss: 0.6003 - accuracy: 0.8352
      Epoch 4/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.5568 - accuracy: 0.8615
      Epoch 5/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.5064 - accuracy: 0.8879
      Epoch 6/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.4517 - accuracy: 0.8989
      Epoch 7/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.3957 - accuracy: 0.9121
      Epoch 8/86
      15/15 [==============================] - 0s 2ms/step - loss: 0.3418 - accuracy: 0.9231
```

Figure 5.4.2

```
Epoch 85/86
15/15 [==============================] - 0s 2ms/step - loss: 0.0322 - accuracy: 0.9934
Epoch 86/86
15/15 [==============================] - 0s 2ms/step - loss: 0.0320 - accuracy: 0.9912
<keras.callbacks.History at 0x7f9d80dcc250>
```

```
[ ] #EVALUATE ON TEST DATA
    test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)

    4/4 [==============================] - 0s 3ms/step - loss: 0.0460 - accuracy: 0.9825
    Test accuracy: 0.9824561476707458
```

```
print(cancerdf.head(1))
#TAKE NEW INPUT FROM USER
ip_data = read_input(len(X.columns))

#PREPROCESS THE DATA
ip_data = transform(ip_data)

#print(ip_data)

#MAKE PREDICTION ON NEW DATA
y_new = model.predict(ip_data)
#print(y_new)

#DISPLAY PREDICTION TO USER
class_label(y_new)
```

Figure 5.4.3

6. TEST CASES

Figure 6.1

Figure 6.2

Figure 6.3

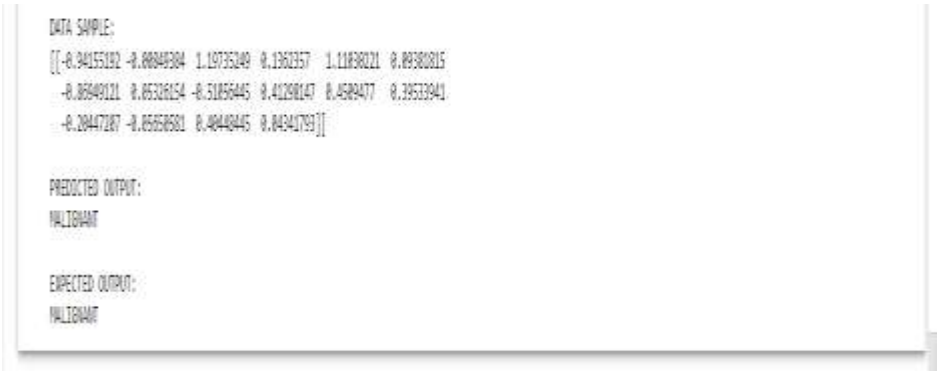


Figure 6.4

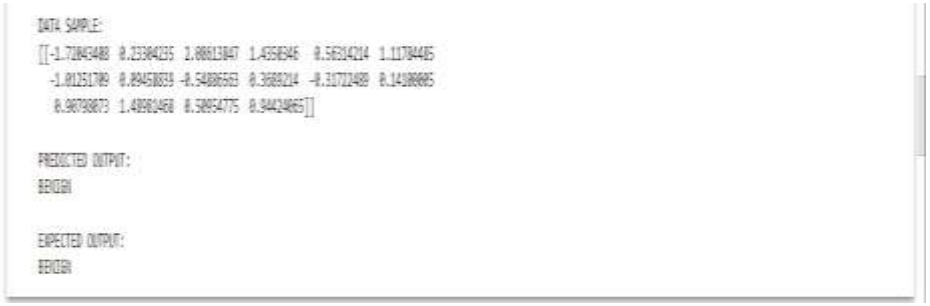


Figure 6.5

7. CONCLUSION

As breast cancer is one of the leading causes of death in women with high mortality rate, early detection can immensely help give a better prognosis for patients by helping medical professionals place more emphasis on early care and better treatment plans instead of on diagnosis/detection of cancer. Experimental results proved that the proposed NN is quite better than the existing methods. It is ensured that the proposed algorithm is advantageous in terms of accuracy and efficiency.

8. FUTURE ENHANCEMENTS

The proposed system uses 16 features to detect malignancy of a tumour cell. Further work can be done to reduce the number of features required to make such a prediction while retaining the same high accuracy for the model. This could lead to more efficiency in model building as well as in prediction once the model is deployed.

BIBLIOGRAPHY

[1] http://www.breastcancer.org/symptoms/understand_bc/what_is_bc.

[2] Y.S. Hotko, Male breast cancer: clinical presentation, diagnosis, treatment, Exp. Oncol. 35 (4) (2013) 303–310.

[3]https://www.biospectrumindia.com/views/21/15300/statistical-analysisof-breast-cancer-in-india.html.

[4] S. Malvia, S.A. Bagadi, U.S. Dubey, S. Saxena, Epidemiology of breast cancer in Indian women, Asia Pac. J. Clin. Oncol. 13 (4) (2017) 289–295.

[5] V. Anji Reddy, Badal Soni, Breast cancer identification and diagnosis techniques, in: Machine Learning for Intelligent Decision Making, Springer, 2020.

[6] Qiao Pan, Yuanyuan Zhang, Dehua Chen, Guangwei Xu, CharacterBased Convolutional Grid Neural Network for Breast Cancer Classification, IEEE, 2017, p. 31.

[7] SanaUllah Khan, Naveed Islam, Zahoor Jan, Ikram Ud Din, Joel J.P.C. Rodrigues, A novel deep learning based framework for the detection and classification of breast cancer using transfer learning, in: Pattern Recognition Letters, Elsevier, 2019.

[8] Qinghua Huang, Yongdong Chen, Longzhong Liu, Dacheng Tao, Xuelong Li, On combining biclustering mining and adaboost for breast tumor classification, IEEE Trans. Knowl. Data Eng. 32 (4) (2020) 728–738.

[9] Shweta Kharya, SunitaSoni, Weighted naive bayes classifier: A predictive model for breast cancer detection, Int. J. Comput. Appl. 133 (9) (2016) 32–37.

[10] R.D.H. Devi, M.I. Devi, Outlier detection algorithm combined with decision tree classifier for early diagnosis of breast cancer, Int. J. Adv. Engg. Tech./Vol. VII/Issue II/April-June 93 (2016) 98.