

A PROJECT REPORT
ON
A DRIVING DECISION STRATEGY(DDS) BASED ON MACHINE
LEARNING FOR AN AUTONOMOUS VEHICLE

Submitted by

N.AMULYA	21781A3295
P.CHANDHINI	21781A32A0
S.PRADEEP KUMAR	21781A32D1
T.TULASI VISHNU PRIYA	21781A32E2
E.NIKHIL KUMAR	21781A32H2

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Under the guidance of

Mr. S KUMARAGURUBARAN

Assistant Professor



SRI VENKATESWARA COLLEGE OF ENGINEERING &
TECHNOLOGY(AUTONOMOUS), R.V.S. NAGAR, CHITTOOR – 517127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuram)

(Accredited by NBA, New Delhi & NAAC, Bengaluru)

(An ISO 9001:2000 Certified Institution)

APRIL 2025

**SRI VENKATESWARA COLLEGE OF ENGINEERING &
TECHNOLOGY (AUTONOMOUS), R.V.S NAGAR,
CHITTOOR – 517127. (A.P).**

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuram)

(Accredited by NBA, New Delhi & NAAC, Bengaluru)

(An ISO 9001:2000 Certified Institution)



BONAFIDE CERTIFICATE

This is to certify that, the project report entitled “**A DRIVING
DECISION STRATEGY (DDS) BASED ON MACHINE LEARNING FOR AN
AUTONOMOUS VEHICLE**” is the Bonafide work carried out by

N.AMULYA	21781A3295
P.CHANDINI	21781A32A0
S.PRADEEP KUMAR	21781A32D1
T.TULASI VISHNU PRIYA	21781A32E2
E.NIKHIL KUMAR	21781A32H2

Students of B.TECH., CSE (DATA SCIENCE), SVCET (AUTONOMOUS), during the academic year 2024-2025, in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE).

SIGNATURE

Mr. A LIBONCE

HEAD OF THE DEPARTMENT

Associate Professor

CSE (Data Science)

SVCET(A), Chittoor

SIGNATURE

Mr.S KUMARGURUBARAN

SUPERVISOR

Assistant Professor

CSE (Data Science)

SVCET(A), Chittoor

**SRI VENKATESWARA COLLEGE OF ENGINEERING &
TECHNOLOGY (AUTONOMOUS),
R.V.S NAGAR, CHITTOOR – 517 127. (A.P).
(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuram)
(Accredited by NBA, New Delhi & NAAC, Bengaluru)
(An ISO 9001:2000 Certified Institution)**



CERTIFICATE OF EVALUATION

COLLEGE NAME: SRI VENKATESWARA COLLEGE OF ENG., &TECH

BRANCH: COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

NAME OF STUDENTS:

N.AMULYA	21781A3295
P.CHANDINI	21781A32A0
S.PRADEEP KUMAR	21781A32D1
T.TULASI VISHNU PRIYA	21781A32E2
E.NIKHIL KUMAR	21781A32H2

PROJECT TITLE: A Driving Decision Strategy(DDS)Based on Machine Learning for an autonomous Vehicle
The project work report submitted in partial fulfillment for the award of CSE (DATA SCIENCE) degree in SVCET, CHITTOOR, is evaluated and confirmed to be report of the work done by the above student.

Viva-Voce Conducted on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

**SRI VENKATESWARA COLLEGE OF ENGINEERING &
TECHNOLOGY (AUTONOMOUS), R.V.S NAGAR,
CHITTOOR – 517 127. (A.P).**

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuram)

(Accredited by NBA, New Delhi & NAAC, Bengaluru)

(An ISO 9001:2000 Certified Institution)



DECLARATION

We hereby declare that the Project Report entitled “**A DRIVING DECISION STRATEGY (DDS) BASED ON MACHINE LEARNING FOR AN AUTONOMOUS VEHICLE**” under the guidance of **Mr. S.KUMARAGURUBARAN, Assistant Professor**, Sri Venkateswara College of Engineering & Technology (Autonomous), Chittoor is submitted in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE).

Signature of the student

N.AMULYA

21781A3295

P.CHANDINI

21781A32A0

S.PRADEEP KUMAR

21781A32D1

T.TULASI VISHNU PRIYA

21781A32E2

E.NIKHIL KUMAR

21781A32H2

ACKNOWLEDGEMENT

A grateful thanks to **Dr. R. VENKATASWAMY, chairman** of Sri Venkateswara College of Engineering and Technology for providing education in their esteemed institution.

We wish to record our deep sense of gratitude and profound thanks to our beloved **Vice Chairman, Sri R. V. SRINIVAS** for his valuable support throughout the course.

We express our sincere thanks to **Dr. M. MOHAN BABU**, our beloved principal for his encouragement and suggestion during the course of study.

With the deep sense of gratefulness, we acknowledge **Mr. A LIBONCE**, Head of the Department, Computer Science and Engineering (Data Science), for giving us his inspiring guidance in undertaking our project report.

We express our sincere thanks to the Project Guide **Mr. S KUMARAGURUBARAN, Assistant Professor**, Department of Computer Science and Engineering (Data Science), for his keen interest, stimulating guidance, constant encouragement with our work during all stages, to bring this project into fruition.

We wish to convey our gratitude and express our sincere thanks to all Project Review Committee members for their support and cooperation rendered for successful submission of our project work.

Finally, we would like to express our sincere thanks to all teaching, non- teaching faculty members, our parents, friends and for all those who have supported us to complete the project work successfully.

N.AMULYA	21781A3295
P.CHANDINI	21781A32A0
S.PRADEEP KUMAR	21781A32D1
T.TULASI VISHNU PRIYA	21781A32E2
E.NIKHIL KUMAR	21781A32H2

A Driving Decision Strategy (DDS) Based on Machine learning for an autonomous Vehicle

Abstract:

A modern-day self-sustaining car determines its driving method by means of thinking about solely exterior factors (Pedestrians, street conditions, etc.) barring thinking about the interior circumstance of the vehicle. To overcome above problems, in this paper author proposed a new strategy, i.e. "A Driving Decision Strategy (DDS) Based on Machine learning for an autonomous vehicle". Analysis of both external and internal factors determines the optimal strategy for an autonomous vehicle (consumable conditions, RPM levels etc.). To implement this, the project author has introduced an algorithm called DDS (Driving Decision Strategy) algorithm which is based on genetic algorithm to choose optimal gene values which helps in taking better decision or prediction. DDS algorithm obtained input from sensor and then passes to genetic algorithm to choose optimal value which helps in faster and efficient prediction. Propose DDS with genetic algorithm performance is comparing with existing machine learning algorithm such as Random Forest and MLP (multilayer perceptron algorithm). Propose DDS shows better prediction accuracy compare to random forest and MLP.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGENO.
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	2
	2.1. SURVEY OF DIAGNOSIS SYSTEM	
3.	SYSTEM ANALYSIS	3-4
	3.1. EXISTING SYSTEM	
	3.2. PROPOSED SYSTEM	
4.	SYSTEM DESIGN	5-14
	4.1. SYSTEM ARCHITECTURE	
	4.2. UML DIAGRAMS	
	4.3. SYSTEM STUDY	
5.	SYSTEM ENVIRONMENT	15-20
	5.1. INTRODUCTION TO PYTHON	
	5.2. ADVANTAGES	
	5.3. DISADVANTAGES	
	5.4. HISTORY	
6.	MACHINE LEARNING	21-29
	6.1. CATEGORIES	
	6.2. CHALLENGES	
	6.3. APPLICATIONS	
	6.4. PRO'S AND CON'S	
7.	IMPLEMENTATION	30-40
8.	SYSTEM	41-44
	8.1. TYPES OF TESTING	
	8.2. UNIT TESTING	
	8.3. INTEGRATION TESTING	
	8.4. FUNCTIONAL TEST	
	8.5. SYSTEM TEST	
	8.6. WHITE & BLACK BOX TESTING	

8	OUTPUT RESULT SCREENSHOT	45-51
	7.1 HOME	
	7.2 DATA SET INSERTING	
	7.3 DATA SET LOADED	
	7.4 TRAIN THE MODEL	
	7.5 RANDOM FOREST RESULT	
	7.6 MLP RESULT	
	7.7 DDS RESULT	
	7.8 GRAPH	
	7.9 TEST DATA	
	7.10 RESULT	
9	CONCLUSION	52
10	REFERENCES	53

LIST OF FIGURES

SL.NO	NAME OF THE FIGURE	PAGE NO
1.	SYSTEM ARCHITECTURE	5
2.	USE CASE DIAGRAM	10
3.	CLASS DIAGRAM	11
4.	SEQUENCE DIAGRAM	12
5.	COLLABORATION DIAGRAM	13

INTRODUCTION :

Companies around the world are developing technologies for advanced autonomous vehicles, which are currently in the 4th stage of development. The principle of operation of selfdriving cars can be classified into three levels: recognition, judgement and control. As part of the recognition process, vehicles are equipped with various sensors, including GPS, cameras, and radar. As a result of this information, the judgement step determines a driving strategy. When the driving environment is identified, it is analysed and appropriate driving plans are developed and the objectives. Vehicle starts driving on its own after the control step has been completed. In order to reach its destination, an autonomous vehicle performs a series of actions, repeating on its own the steps of recognition, judgement and control.

as their performance improves. An increase in these sensors can lead to an overload of the vehicle's electrical system. In-vehicle computers compute data collected by sensors in self- driving vehicles. Due to overload, the speed of judgement and control decreases as the amount of computed data increases. These problems can jeopardise the vehicle's stability. As a means of preventing sensor overload, some studies have developed hardware that can perform deep- running operations inside a vehicle, while others use cloud computing to compute sensor data.

LITERATURE SURVEY :

Y. N. Jeong, S. R. Son, E.H. Jeong and B. K. Lee, "An Integrated Self- Diagnosis System for an Autonomous Vehicle Based on an IoT Gateway and Deep Learning" Applied Sciences, vol. 8, no. 7, July 2018.

This paper proposes "An Integrated Self-diagnosis System (ISS) for an Autonomous Vehicle based totally on an Internet of Things (IoT) Gateway and Deep Learning," which collects records from an independent vehicle's sensors, diagnoses itself and the have an impact on between its components the usage of Deep Learning, and notifies the driver of the results. Three modules make up the ISS. The first In-Car Gateway Module (In-VGM) takes facts from in-vehicle sensors, such as media records from a black box, riding radar, and car manipulate messages, and sends every piece of statistics over every Controller Area Network (CAN) to the on-board diagnostics (OBD) or actuators with the aid of the, FlexRay, and Media Oriented Systems Transport (MOST) protocols. The statistics from in-vehicle sensors is dispatched to the CAN or FlexRay protocol, whilst media facts acquired whilst using is dispatched to the MOST protocol. A vacation spot protocol message kind is created from various kinds of messages that have been transferred. The 2nd Optimized Deep Learning Module (ODLM) generates the Training Dataset the usage of information obtained from in-car sensors and calculates the chance of car components and consumables, as nicely as the threat of different components influenced through a faulty part. to enhance the self-diagnosis velocity and decrease the device overhead, whilst a V2X primarily based Accident Notification Service (VANS) informs the adjoining motors and infrastructures of the self- diagnosis result analyzed with the aid of the OBD. This paper improves upon the simultaneous message transmission effectivity via the In-VGM by way of 15.25% and diminishes the mastering error price of a Neural Network algorithm via the ODLM through about 5.5%.

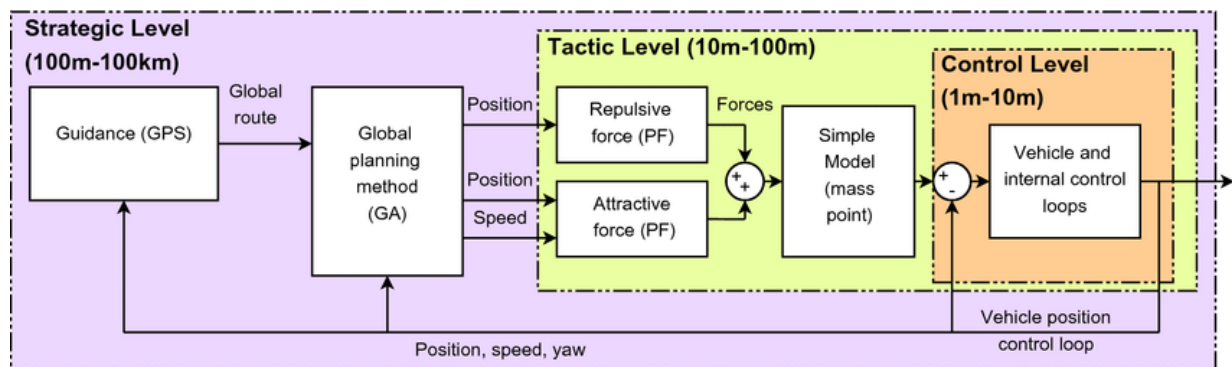
EXISTING SYSTEM:

Companies around the world are developing technologies for advanced autonomous vehicles, which are currently in the 4th stage of development. The principle of operation of self-driving cars can be classified into three levels: recognition, judgement and control. As part of the recognition process, vehicles are equipped with various sensors, including GPS, cameras, and radar. As a result of this information, the judgement step determines a driving strategy. When the driving environment is identified, it is analysed and appropriate driving plans are developed and the objectives. Vehicle starts driving on its own after the control step has been completed. In order to reach its destination, an autonomous vehicle performs a series of actions, repeating on its own the steps of recognition, judgement and control.

PROPOSED SYSTEM:

In this paper author is describing concept for driving decision strategy by observing vehicle internal data such as steering and RPM level to predict various classes such as speed (steering), changing lane etc. All existing technique was concentrate on external data such as road condition and pedestrians etc. but not concentrate on internal values. So to take efficient determination of steering condition and changing lane author is analyzing internal data. All internal data will be collected from sensor and then store on cloud and then application will read data from cloud and then apply machine learning algorithms to determine or predict steering condition or changing lane.

SYSTEM ARCHITECTURE :



HARDWARE & SOFTWARE REQUIREMENTS:

HARD REQUIRMENTS :

- System : i3 or above.
- Ram : 4 GB.
- Hard Disk : 40 GB

SOFTWARE REQUIRMENTS :

- Operating system : Windows8 or Above.
- Coding Language : python

SYSTEM STUDY FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM DESIGN :

4.1 .UML DIAGRAM'S :

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 for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

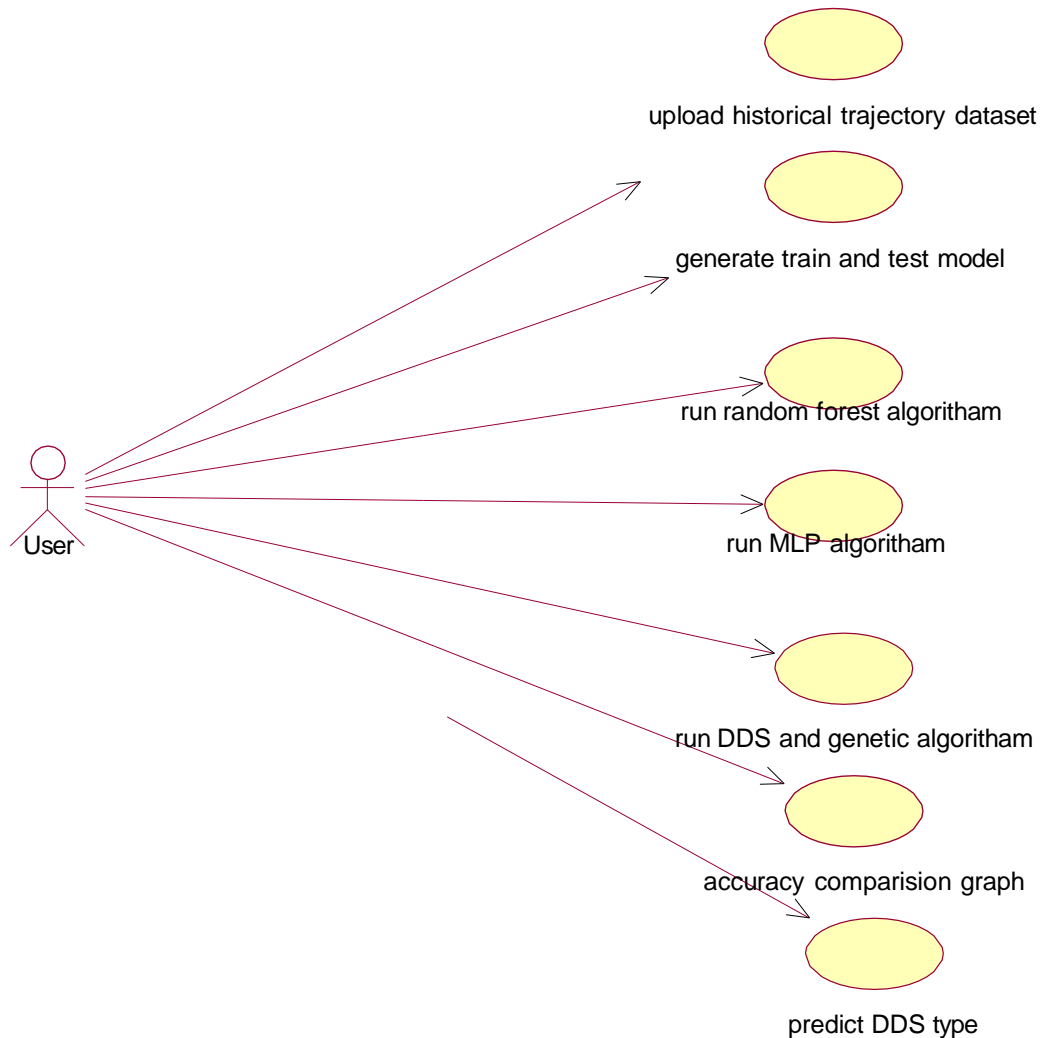
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

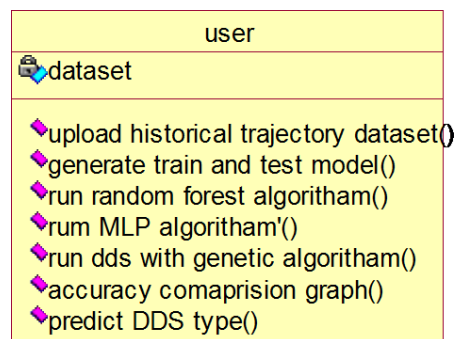
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a 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.



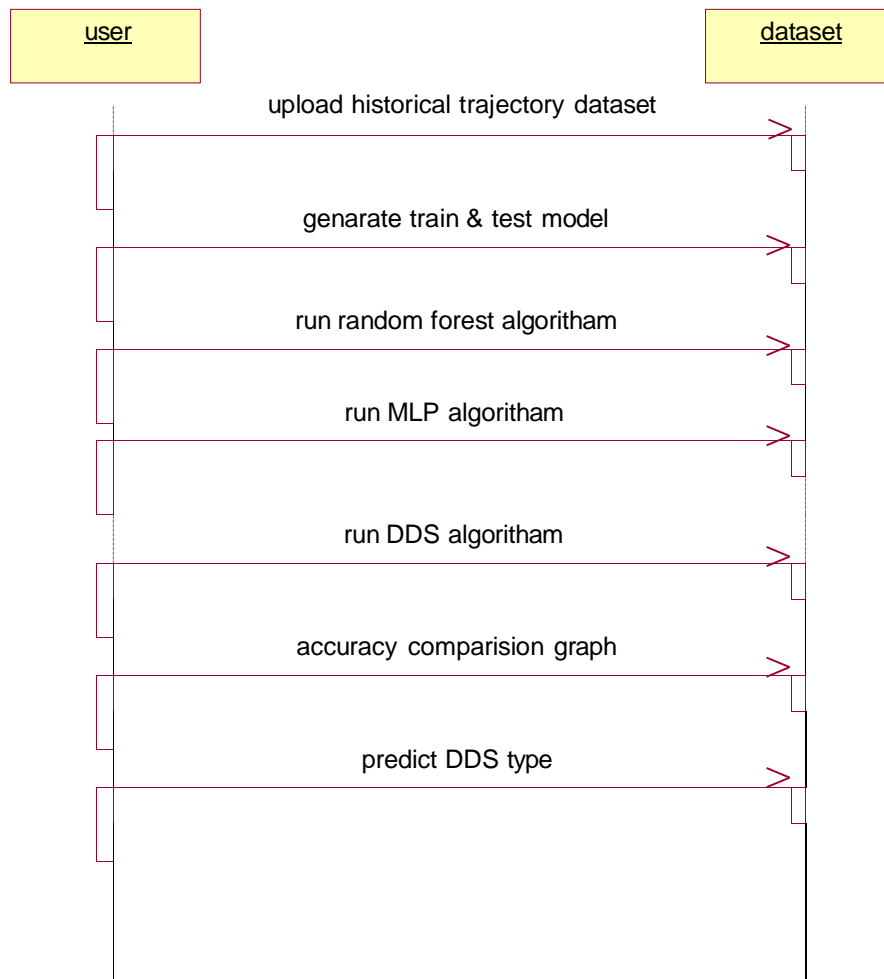
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



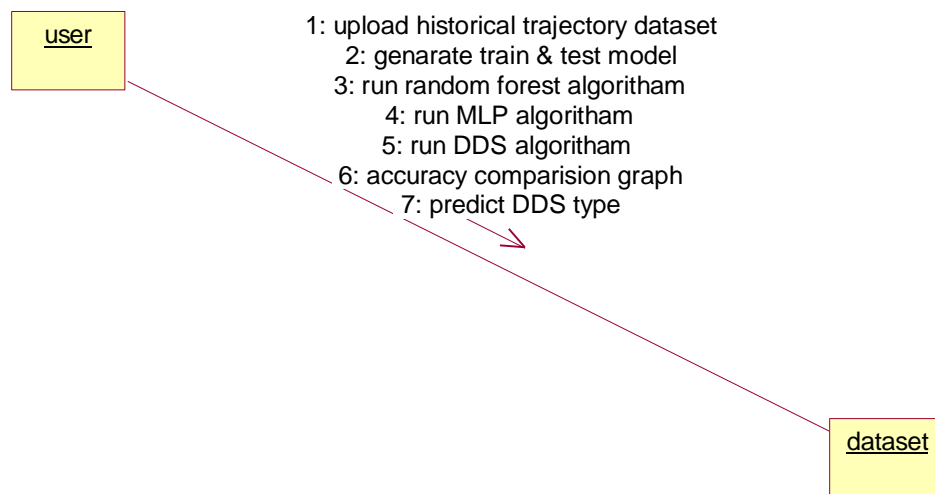
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



COLLABRATION DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



IMPLEMENTATION:

MODULES:

1. Upload Historical Trajectory Dataset : Upload Historical Trajectory Dataset' button and upload dataset.
2. Generate Train & Test Model :Generate Train & Test Model' button to read dataset and to split dataset into train and test part to generate machine learning train model
3. Run MLP Algorithm:Run MLP Algorithm' button to train MLP model and to calculate its accuracy.
4. Run DDS with Genetic Algorithm : Run DDS with Genetic Algorithm button to train DDS and to calculate its prediction accuracy.
5. Predict DDS Type :Predict DDS Type' button to predict test data

SOFTWARE ENVIRONMENT :

What is Python :

Below are some facts about 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, PyQt etc.)
- 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 *contain code 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++ do. Also, the fact that you need to 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.

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 more 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.

Any doubts till now in the advantages of Python? Mention in the comment section.

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

Compared 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

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

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 voor Wiskunde en 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.

What is 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 programming 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 specialist persons – 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!!!

How to start learning ML?

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.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

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 an e-tailer 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, dict, str 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 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

Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

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 with no Python background - without breaking.

Modules Used in Project :-

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.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

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 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 with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

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. First, 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 Cheatsheet here.](#) 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

Step 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>



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

Step 2: Click on the Download Tab.



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

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

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.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4ae77b9ab01b7079be	23017663	
XZ compressed source tarball	Source release		033e4aae6097051c3eca45ee3604803	17133432	
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa75b3daf1a442cbalce08e6	34898416	
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd905c38217a45773b95e4b36b243f	20982845	
Windows help file	Windows		063099573ab296b2ac58ade0b4f7cd2	8131761	
Windows x86 64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b093d7fd8e0d8d8e03184a0720a2	7504291	
Windows x86 64 executable installer	Windows	for AMD64/EM64T/x64	a702be0ad70d9bdc3543a7d3e563400	26882368	
Windows x86 64 web-based installer	Windows	for AMD64/EM64T/x64	29c3b1c9088bd73a8e651a3bd351b4b52	1362904	
Windows x86 embeddable zip file	Windows		9fab3bd18b41879fda94132574139d8	6741628	
Windows x86 executable installer	Windows		33c3822942d54446ac3b645147c394789	25663848	
Windows x86 web-based installer	Windows		1b670cfafed117d82c30983ea371d87c	1324608	

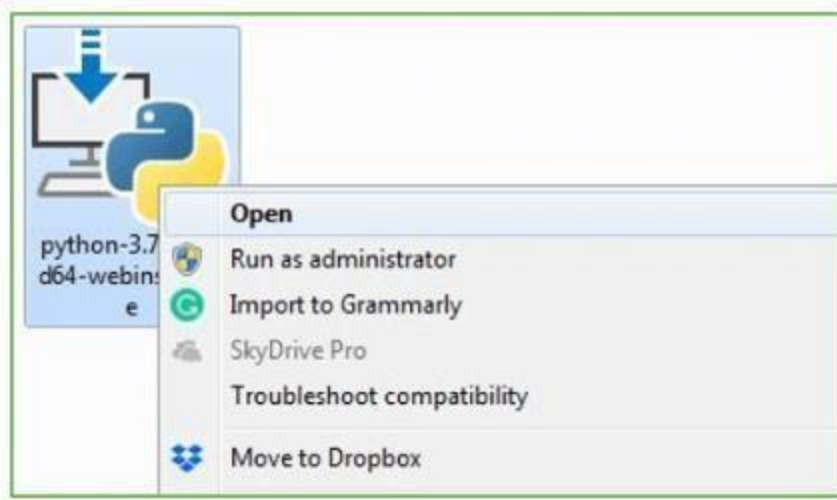
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, 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.



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



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



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

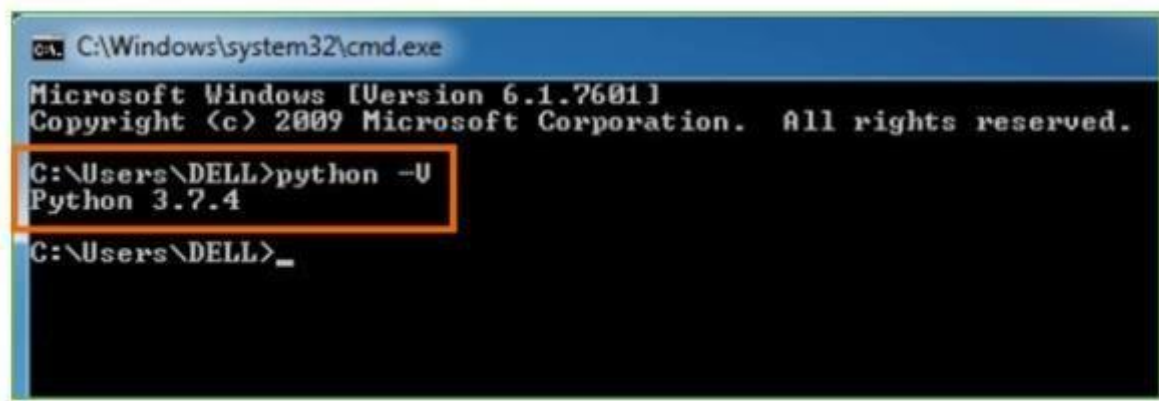
Step 1: Click on Start

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



Step 3: Open the Command prompt option.

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



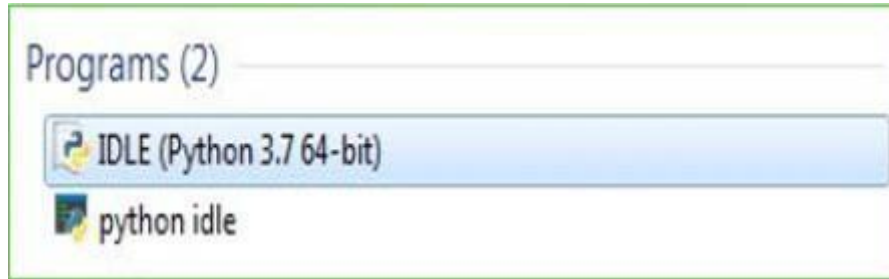
Step 5: You will get the answer as 3.7.4

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

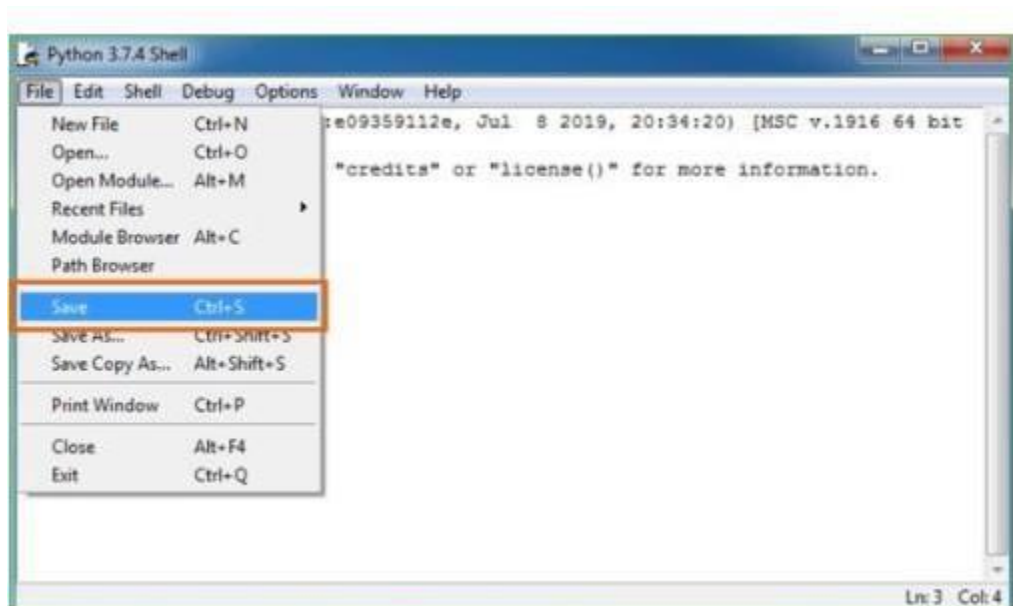
Step 1: Click on Start

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



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**



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**

SYSTEM TEST :

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

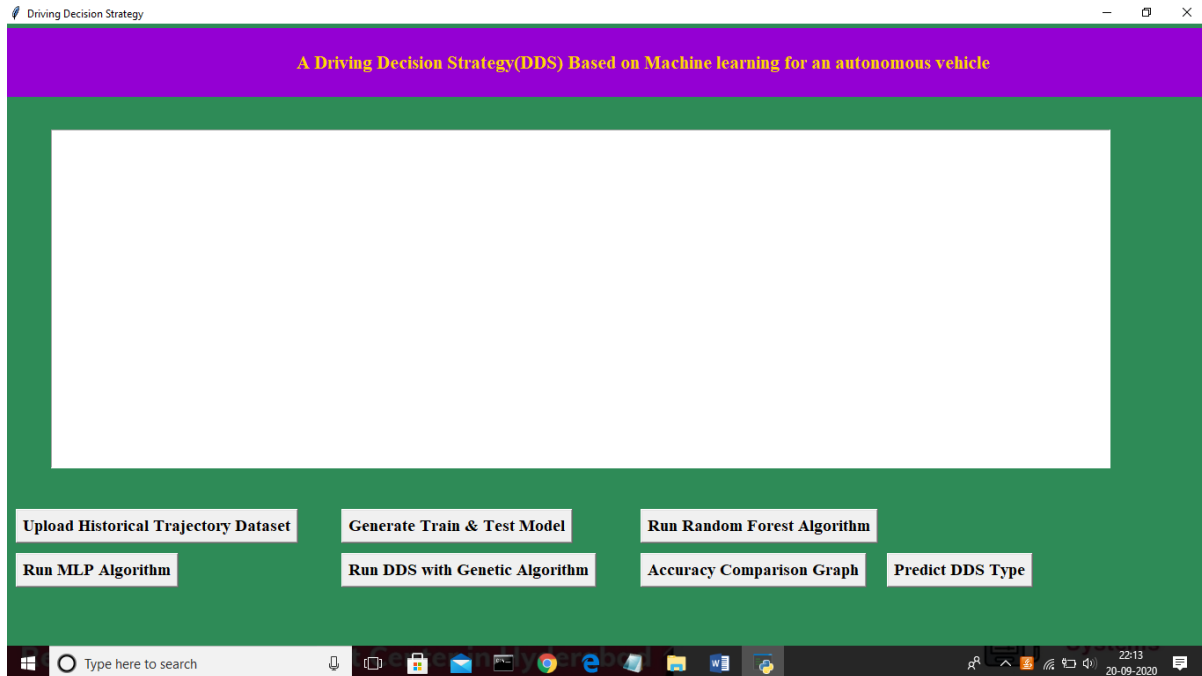
Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

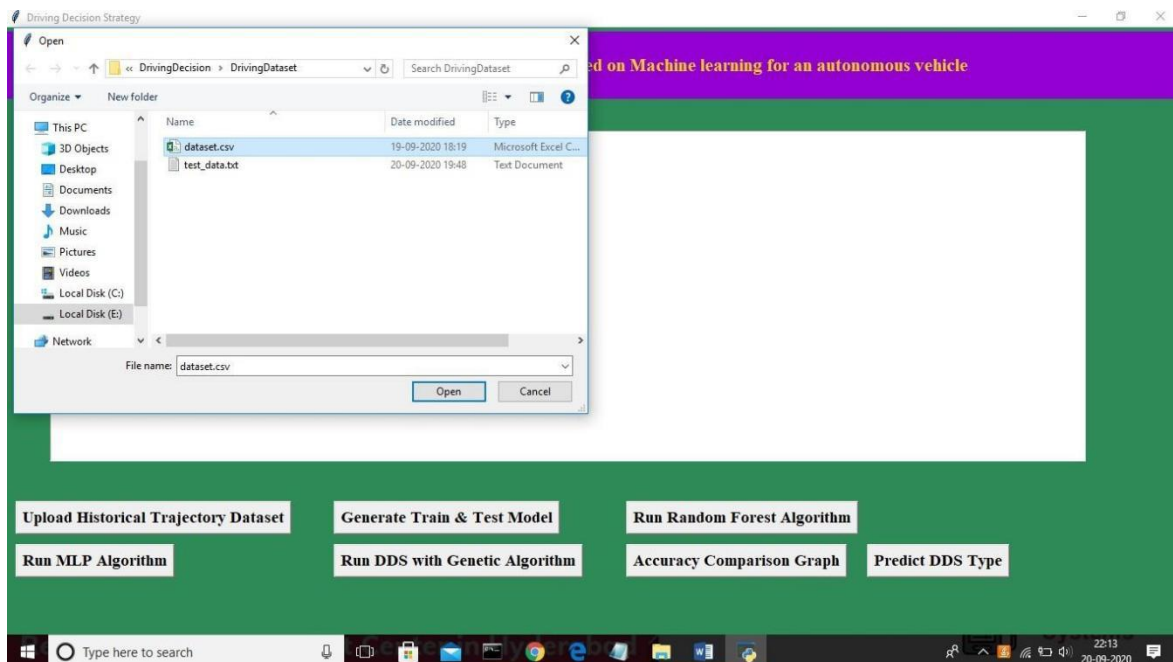
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

SCREENSHOTS :

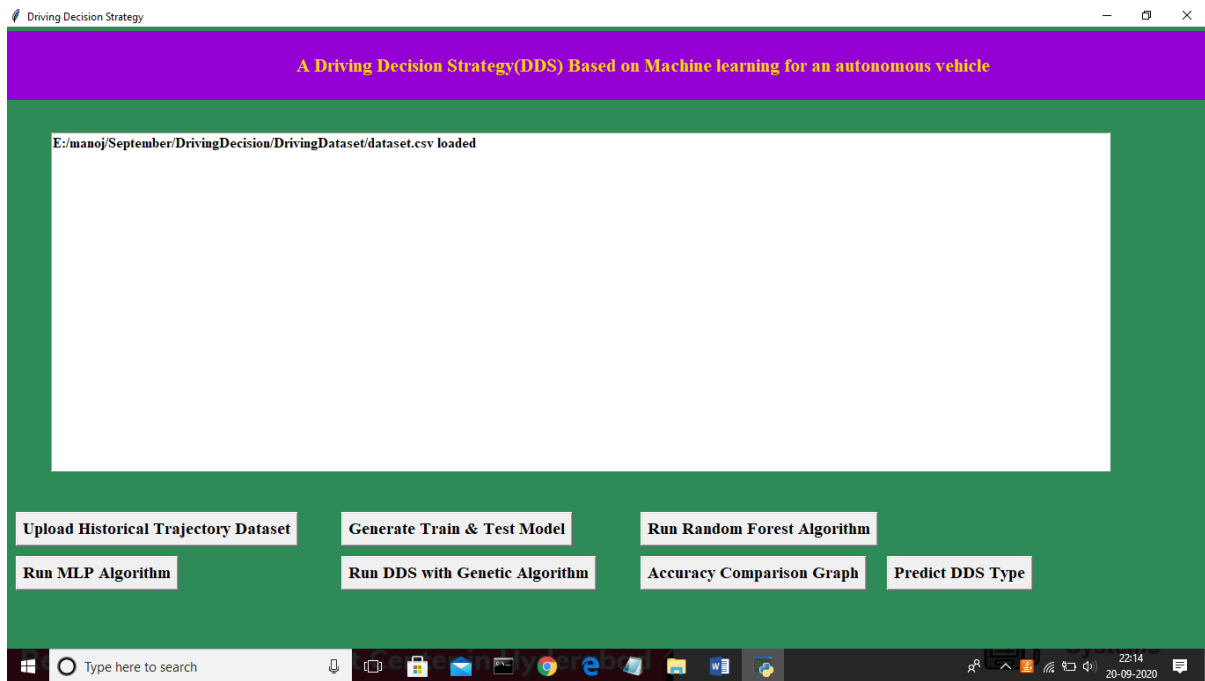
To run project double click on 'run.bat' file to get below screen



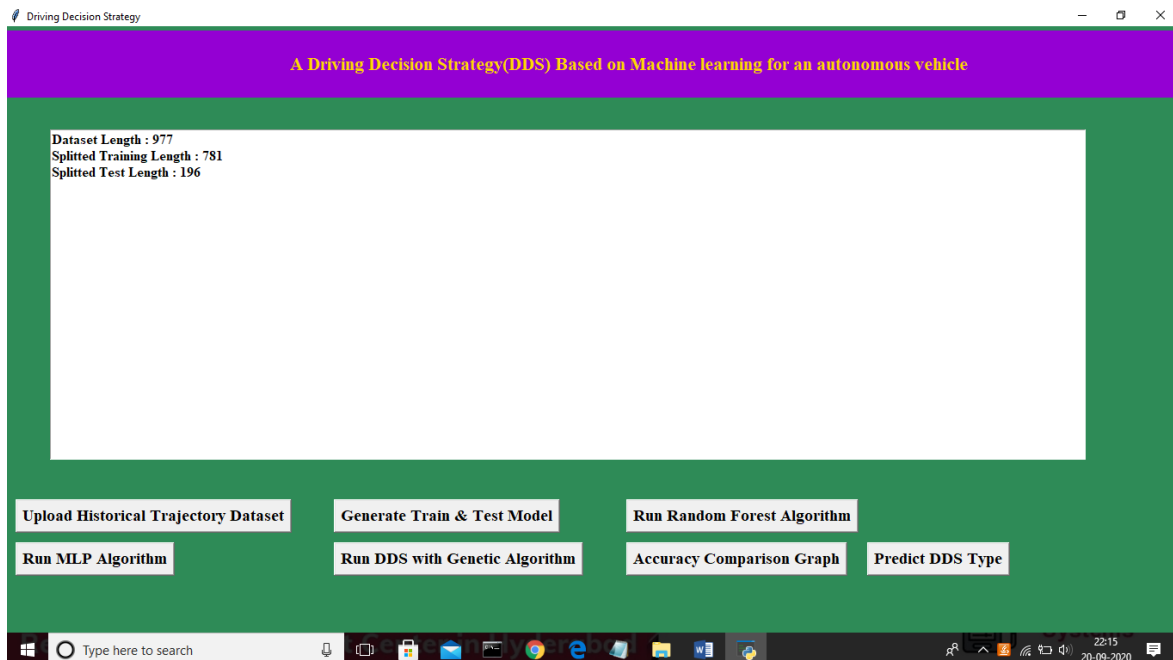
In above screen click on 'Upload Historical Trajectory Dataset' button and upload dataset



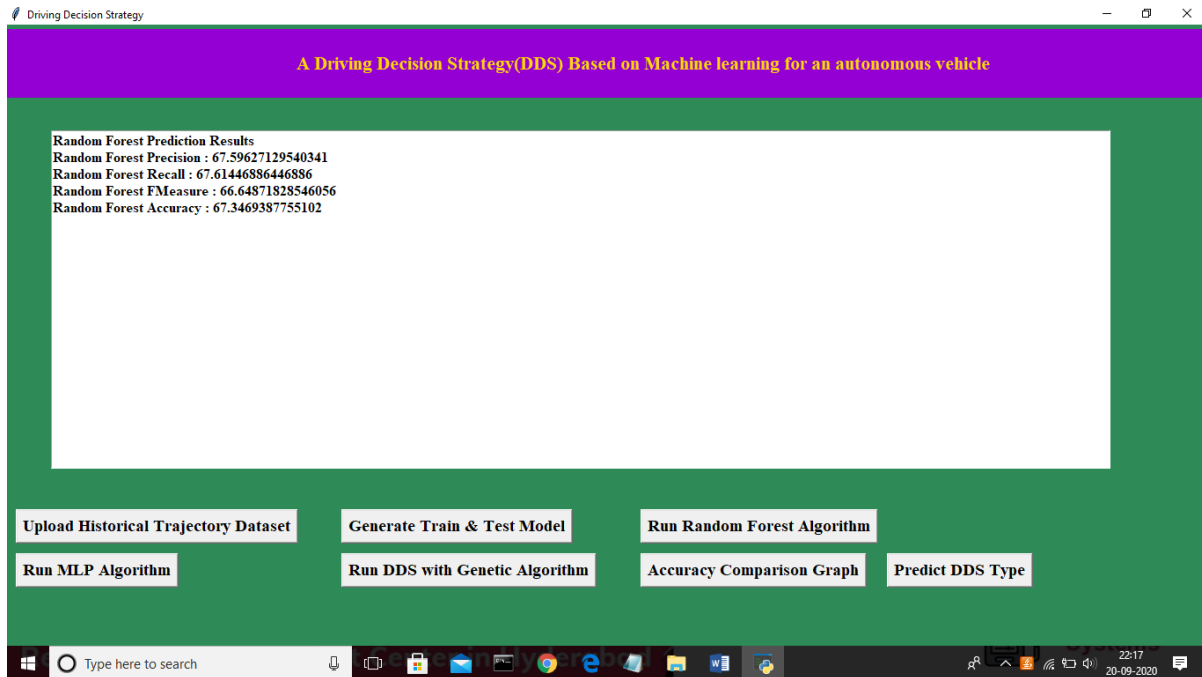
Now select 'dataset.csv' file and click on 'Open' button to load dataset and to get below screen



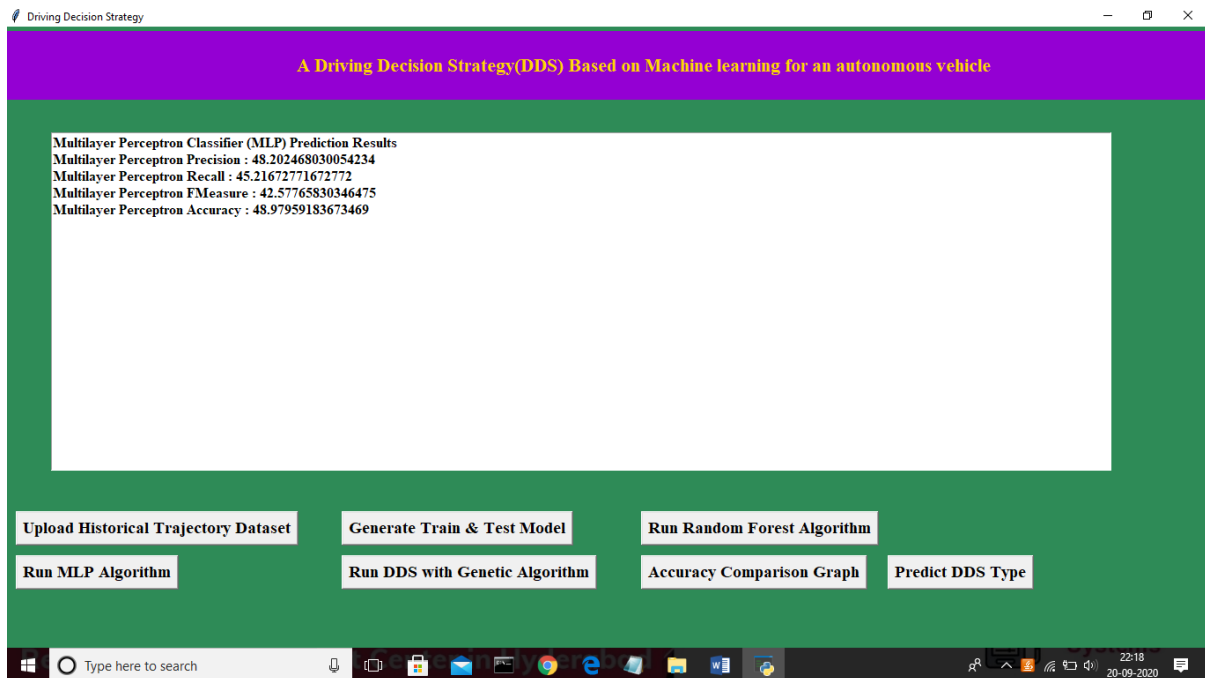
In above screen dataset is loaded and now click on 'Generate Train & Test Model' button to read dataset and to split dataset into train and test part to generate machine learning train model



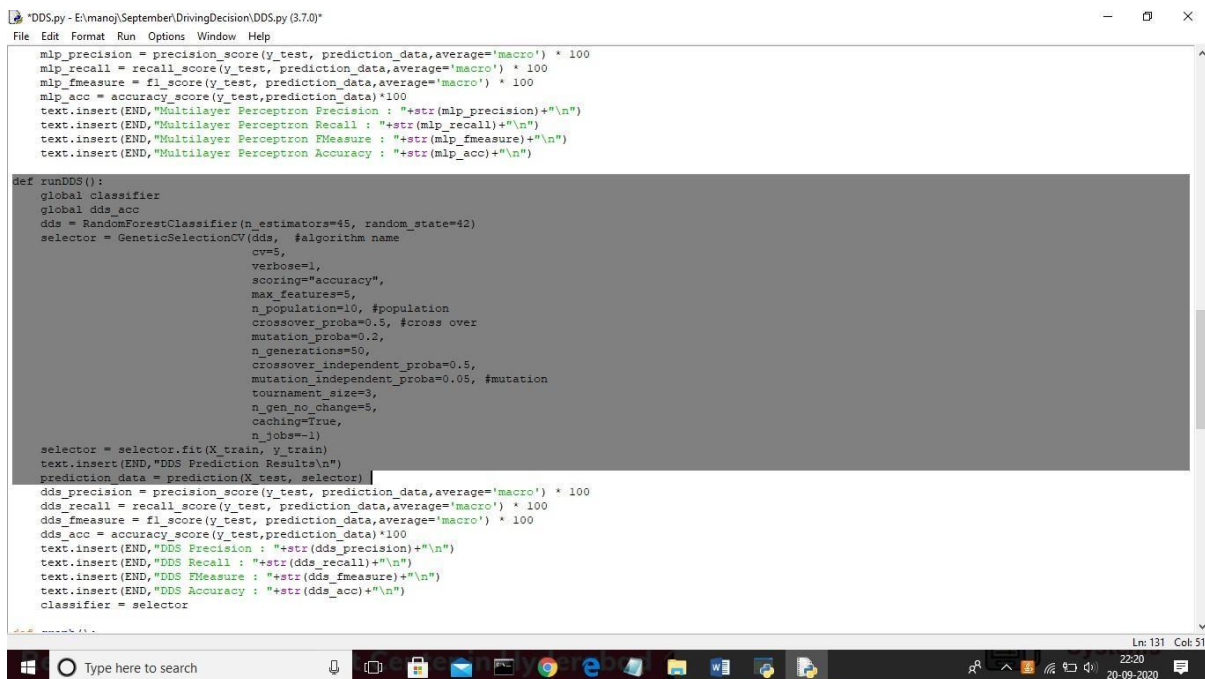
In above screen dataset contains 977 total trajectory records and application using 781 (80% of dataset) records for training and 196 (20% of dataset) for testing. Now both training and testing data is ready and now click on 'Run Random Forest Algorithm' button to train random forest classifier and to calculate its prediction accuracy on 20% test data



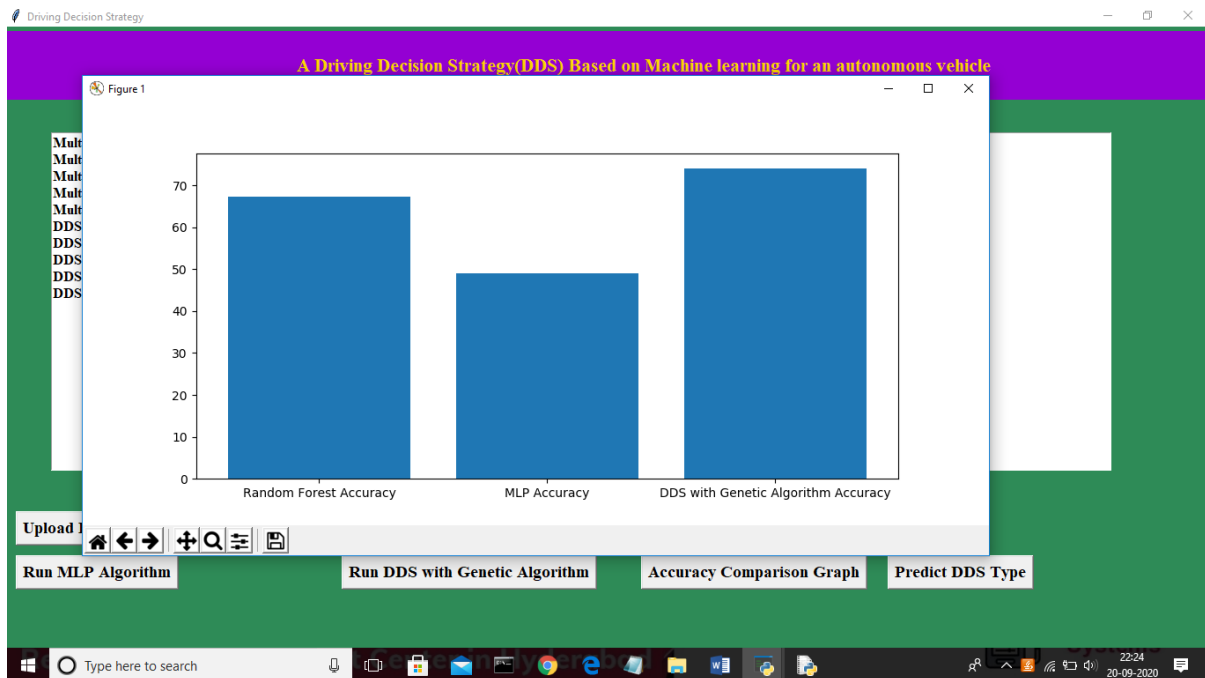
In above screen we calculated random forest accuracy, precision, recall and fmeasure and random forest got 67% prediction accuracy. Now click on 'Run MLP Algorithm' button to train MLP model and to calculate its accuracy



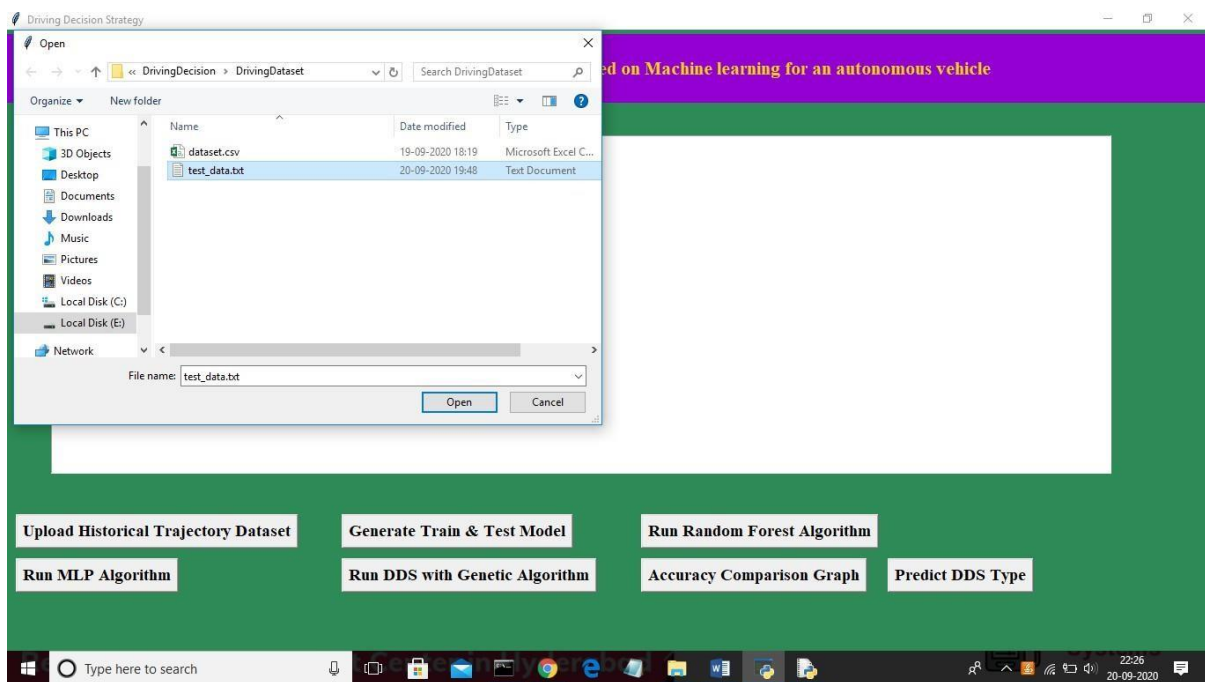
In above screen MLP got 48% prediction accuracy and in below screen we can see genetic algorithm code used for building propose DDS algorithm



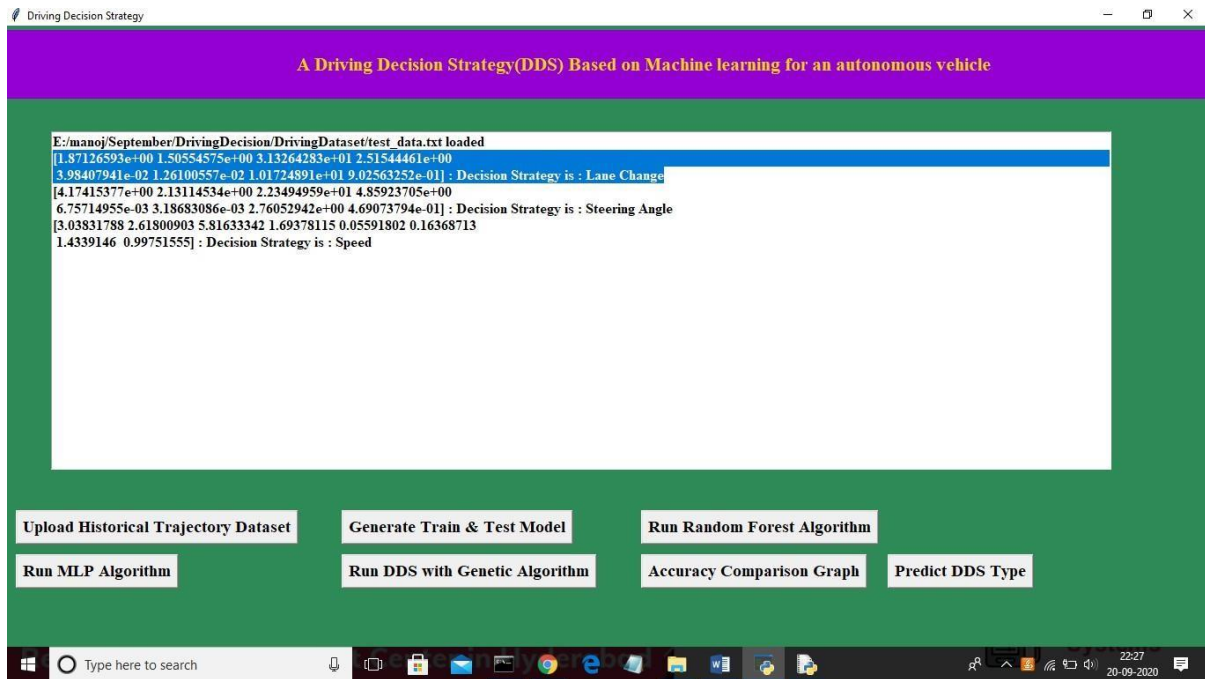
In above screen we can see genetic algorithm code used in DDS algorithm and now click on 'Run DDS with Genetic Algorithm' button to train DDS and to calculate its prediction accuracy



In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that DDS is performing well compare to other two algorithms. Now click on ‘Predict DDS Type’ button to predict test data



In above screen uploading ‘test_data.txt’ file and click on ‘Open’ button to predict driving decision



In above screen in selected first record we can see decision is Lane Change and for second record values we got decision as 'steering angle' and for third test record we got predicted value as vehicle is in speed mode.

CONCLUSION :

A Driving Decision Strategy was proposed in this paper. It uses a genetic algorithm based on gathered data to establish the vehicle's ideal driving strategy based on the slope and curve of the road it is travelling on, and it visualizes the autonomous vehicle's driving and consumables circumstances to provide drivers. To demonstrate the validity of the DDS, experiments were conducted to determine the optimal driving strategy by evaluating data from an autonomous vehicle. The DDS finds the best driving strategy 40 percent faster than the MLP, despite having similar accuracy. DDS also has a 22 percent higher accuracy than RF and calculates the best driving strategy 20 percent faster than the RF system. When accuracy and real-time are required, the DDS is the best choice. The DDS sends only the data needed to identify the vehicle's optimal driving strategy to the cloud, and analyses it using a genetic algorithm, it is faster than other methods. These tests were carried out in a virtual environment using PCs, which had inadequate visualization capabilities. A real-world test of DDS should be conducted in the future. Expert designers should also improve the visualization components.

REFERENCES :

- [1] Y. N. Jeong, S. R. Son, E.H. Jeong and B. K. Lee, “An Integrated SelfDiagnosis System for an Autonomous Vehicle Based on an IoT Gateway and Deep Learning,” Applied Sciences, vol. 8, no. 7, July 2018.
- [2] Yukiko Kenmochi, LilianBuzer, Akihiro Sugimoto, Ikuko Shimizu, “Discrete plane segmentation and estimation from a point cloud using local geometric patterns,” International Journal of Automation and Computing, vol. 5, no. 3, pp. 246-256, 2008.
- [3] Ning Ye, Yingya Zhang, Ruchuan Wang, Reza Malekian, “Vehicle trajectory prediction based on Hidden Markov Model,” The KSII Transactions on Internet and Information Systems, vol. 10, no. 7, 2017.
- [4] Li-Jie Zhao, Tian-You Chai, De-Cheng Yuan, “Selective ensemble extreme learning machine modeling of effluent quality in wastewater treatment plants,” International Journal of Automation and Computing, vol. 9, no.6, 2012.