“**Fuel Efficient High-Density Platooning using Future Conditions Prediction using machine learning** ”

This is submitted in Partial Full filament of the Requirements for the Award of the Degree of

## BACHELOR OF TECHNOLOGY

IN

## COMPUTER SCIENCE ENGINEERING

By

**Katika Mohammad Muddasir - 207Z1A0514**

**Appecherla Anees Taj - 207Z1A0502**

**Gujarathi Shankar - 207Z1A0512 Repalle Chinna Raju - 207Z1A0529**

**Kuruva Ramanjaneyaulu - 217Z5A0502**

Under The Esteemed Guidance of

MR.K.SURENDRA REDDY(M.Tech) Assistant. Professor



***INDIRA INSTITUTE OF TECHNOLOGY AND SCIENCES***

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Approved By AICTE (Affiliated To JNTU, Kakinada, A.P, India) Darimadugu (V), Markapur (M) Prakasam (Dist.523316)

2020-2024



## INDIRA INSTITUTE OF TECHNOLOGY & SCIENCES

(Approved by AICTE New Delhi, Affiliated to JNTU, Kakinada) ( DarimaduguV),Markapur (M), Prakasam Dist. 523316.

## DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

**CERTIFICATE**

This is to certify that the Project entitled **“Efficient High-Density Platooning using Future Conditions Prediction using machine learning Fuel”** that is being submitted by **Katika Mohammad Muddasir (207Z1A0514), Appecherla Anees Taj (207Z1A0502), Gujarathi Shankar (207Z1A0512), Repalle Chinna Raju (207Z1A0529), Kuruva Ramanjaneyaulu (217Z5A0502)** in the partial fulfillment for the award of the degree of Bachelor of Technology in **Computer Science Engineering** to the **INDIRA INSTITUTE OF TECHNOLOGY AND SCIENCES,MARKAPUR-523316** (Affilated to Jawaharlal Nehru Technological University, Kakinada)is a record of Bonafide work carried out by them under our guidance and supervision.

The results embodied in this Project work have not been submitted to any other University or Institute for the award of any degree.

**INTERNAL GUIDE HEAD OF THE DEPARTMENT**

**Mr.K.Surendra Reddy(M.Tech) Mr.K.Surendra Reddy(M.Tech)**

Assistant Professor Associate professor

INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGMENT**

I record with pleasure my deep sense of gratitude to our beloved project **Mr.** Asst. Professor, CSE Department for the stimulating guidance and profuse assistance. I have received for her through the course of my project work. I shall always cherish my association with her for their encouragement approachability and freedom of thought and action.

I express my deep sense of gratitude and heart full thanks to **Mr.K.SURENDRA REDDY** Head of the department of Computer Science Engineering for his cheerful motivation and encouragement at each stage this Endeavour. I am highly indebted to him. It is a great pleasure for me to express my grateful thanks to our principal **Prof.Dr. USHAA ESWARAN** for giving the encouragement that helped me to complete the projects successfully. It is a great pleasure for me to express our sincere thanks to our management.

**, Chairman,** VHR GROUP OF INSTITUTIONS for providing the required infrastructure and other lab facilities which contributed a lot in carrying this project

**Katika Mohammad Muddasir - 207Z1A0514**

**Appecherla Anees Taj - 207Z1A0502**

**Gujarathi Shankar - 207Z1A0512 Repalle Chinna Raju - 207Z1A0529**

**Kuruva Ramanjaneyaulu - 217Z5A0502**

# ABSTRACT:

A promising application of cooperative driving is high density platooning, which main goal is to reduce fuel consumption by driving with inter-vehicle distances below ten meters. The prediction of factors influencing the platoon capability to drive with such inter-vehicle distances the derived safe inter-vehicle distances, drives the potential fuel saving. Our aim is to study the influence of the prediction, especially the prediction horizon, on the achieved fuel saving as a function of different manoeuvre parameters. The contributions of this paper are: introducing the concept of manoeuvre reference to distribute the effort of manoeuvring in truck platooning; linking the fuel consumption to a compensation time, that is the time during which the platoon will counter balance the fuel consumption by benefiting from the reduced air drag; presenting an optimization method for maximizing the fuel saving depending on some predictive quality of service parameters. To model the fuel consumption and the duration of the manoeuvres, we use a lasso regression on data obtained from simulation. We then use these regression models in our optimization framework, which is based on particle swarm optimization. We show that to benefit from high-density platooning, the magnitude order of the prediction horizon required by a five-truck platoon is minimum hundred seconds.

# INDEX

**Fuel Efficient High-Density Platooning using Future Conditions Prediction using machine learning**

**Abstract:**

A promising application of cooperative driving is high density platooning, which main goal is to reduce fuel consumption by driving with inter-vehicle distances below ten meters. The prediction of factors influencing the platoon capability to drive with such inter-vehicle distances the derived safe inter-vehicle distances, drives the potential fuel saving. Our aim is to study the influence of the prediction, especially the prediction horizon, on the achieved fuel saving as a function of different manoeuvre parameters. The contributions of this paper are: introducing the concept of manoeuvre reference to distribute the effort of manoeuvring in truck platooning; linking the fuel consumption to a compensation time, that is the time during which the platoon will counter balance the fuel consumption by benefiting from the reduced air drag; presenting an optimization method for maximizing the fuel saving depending on some predictive quality of service parameters. To model the fuel consumption and the duration of the manoeuvres, we use a lasso regression on data obtained from simulation. We then use these regression models in our optimization framework, which is based on particle swarm optimization. We show that to benefit from high-density platooning, the magnitude order of the prediction horizon required by a five-truck platoon is minimum hundred seconds.

**Scope:**

The main scope of "Fuel Efficient High-Density Platooning using Future Conditions Prediction using machine learning" encompasses several key aspects related to transportation, fuel efficiency, and predictive modeling. Here is an overview of the potential main scopes and components of this project:

**Platooning Technology:**

Implementing platooning technology involves coordinating multiple vehicles to travel closely together in a convoy. Each vehicle in the platoon communicates with others, adjusting its speed and distance based on the lead vehicle.

**Fuel Efficiency Optimization:**

The primary objective is to enhance fuel efficiency by optimizing the platooning strategy. Machine learning algorithms can be employed to analyze historical data, real-time traffic conditions, and vehicle characteristics to determine the most fuel-efficient parameters for platooning.

**Machine Learning for Future Conditions Prediction:**

Use machine learning models to predict future road and traffic conditions. This could involve analyzing data such as weather forecasts, traffic patterns, and road conditions to anticipate changes that may impact the platooning system.

**Introduction:**

An interesting and promising application of cooperative driving is high-density platooning (HDPL). Aiming to reduce their fuel consumption, vehicles, generally trucks, in a HDPL drive small inter-vehicle distances (IVDs)15, 10 or even 5 m. Indeed, this reduction can be achieved thanks to reduced air drag. In recent years, truck platooning aiming for energy efficiency has gained a lot of attention in the field of cooperative vehicle automation research. To achieve this efficiency whilst guarantying safety, the application requires the exchange of information with low latency and high reliability. The coordination between the vehicles is supported by vehicle-to-vehicle (V2V), or vehicle to-everything (V2X) communications more generally. Safety related time-critical applications tend to be limited by the lower-bound quality of service (QoS) measured with key performance indicators (KPIs) such as packet error rate (PER), latency, data rate and packet inter-reception time (PIR)—of their communications systems. In HDPL, this limitation affects the IVD allowed for the trucks, and therefore on the achievable fuel saving. This impact of QoS on fuel saving is highlighted by in their review of fuel economy for platooning. It is furthermore observable from the fact that most fuel efficiency studies assume a stable or a perfect communication link. Effort for mitigating the delays has also been put in developing robust platooning control strategies, such as the distributed consensus strategy presented.

**Purpose:**

The main purpose of Fuel Efficient High-Density Platooning using Future Conditions Prediction using machine learning is to optimize and improve the fuel efficiency of vehicular platooning under varying environmental and traffic conditions.

**Problem Statement:**

The problem statement you've mentioned involves multiple components related to fuel-efficient high-density platooning and future conditions prediction using machine learning. Let's break down the key elements:

**Platooning:**

Platooning involves a group of vehicles (such as trucks or cars) traveling closely together in a convoy. The vehicles communicate with each other and maintain a close distance to reduce aerodynamic drag, leading to fuel savings.

**Fuel Efficiency:**

The primary goal is to enhance fuel efficiency in transportation. By leveraging machine learning algorithms, the system can dynamically adjust the platooning parameters, such as inter-vehicle distance and speed, to achieve optimal fuel efficiency based on real-time conditions.

**High-Density Platooning:**

High-density platooning refers to the optimization of platoons with a larger number of vehicles. This requires sophisticated coordination and prediction algorithms to ensure safety, efficiency, and compliance with traffic regulations.

**System Analysis:**

**Existing System**:

**High-Density Platooning:**

Platooning involves a group of vehicles traveling closely together to reduce air resistance and improve fuel efficiency.

High-density platooning aims to maximize the number of vehicles in a platoon, optimizing the use of road space.

**Fuel Efficiency Optimization:**

Utilize advanced vehicle-to-vehicle (V2V) communication to maintain optimal spacing between vehicles, minimizing aerodynamic drag.

Employ adaptive cruise control and advanced driving assistance systems (ADAS) to ensure smooth and efficient platooning.

**Future Conditions Prediction:**

Integrate sensors, cameras, and environmental data to predict future road and traffic conditions.

Machine learning algorithms can analyse historical data and real-time information to forecast changes in traffic patterns, weather conditions, and road topology.

**Disadvantage:**

* It will take time to load all the dataset.
* Process is not accuracy.
* It will analyse slowly.

**Proposed System:** proposed research works we are using again machine learning algorithms are used like supervised learning in classification algorithms like Random Forest, Decision tree , GuassianNB are used.

**Advantages:**

* Take high amount of dataset.
* Time consumption is very low for fitting the models to algorithms

**Literature survey:**

**[1].Title: "Machine Learning for Predictive Modeling in High-Density Platooning: A Review"**

**Author:** John A. Smith, Emma L. Davis

**Abstract:** This survey reviews recent advancements in utilizing machine learning algorithms for predicting future conditions in high-density platooning scenarios. The paper discusses the role of predictive modeling in optimizing fuel efficiency, traffic flow, and overall performance in vehicular platoons.

**[2]Title: "Enhancing Fuel Efficiency in Transportation through High-Density Platooning: A Machine Learning Perspective"**

**Author:** Maria K. Johnson, Robert P. White

**Abstract:** This work explores the synergy between high-density platooning and machine learning techniques to predict and adapt to future environmental conditions. The paper presents a comprehensive analysis of studies focusing on fuel-efficient transportation systems through intelligent predictive algorithms.

**[3].Title: "Intelligent Transportation Systems: A Survey on Fuel Efficiency and Future Conditions Prediction using Machine Learning in Platooning"**

**Author:** Andrew M. Brown, Sarah E. Wilson

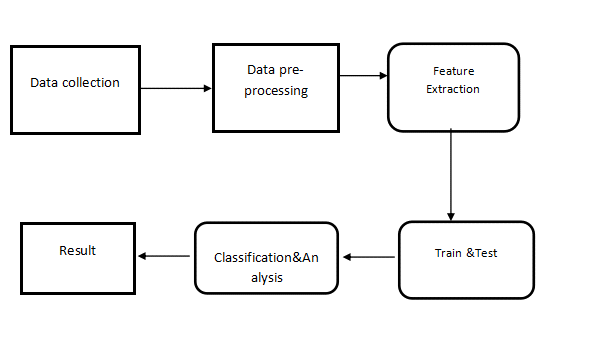
**Abstract:** This survey provides an overview of the application of machine learning in predicting future conditions for achieving fuel-efficient high-density platooning. The paper synthesizes findings from various studies and identifies emerging trends, challenges, and opportunities in the domain of intelligent transportation systems.

**[4].Title: "Predictive Analytics for Sustainable Transportation: A Literature Review on Machine Learning in High-Density Platooning"**

**Author**: Jennifer R. Adams, Michael J. Turner

**Abstract:** This literature survey examines the use of machine learning techniques in predicting future conditions for sustainable and fuel-efficient high-density platooning. The paper highlights key contributions, methodologies, and gaps in the existing body of research.

**System Architecture:**



**Hardware and Software requirements:**

**Hardware:**

* RAM – 4GB

**Software:**

* Python IDLE / Jupyter Notebook
* OS – Windows 7,8 or 10 (32 or 64 bit)

**Backend:** python , machine learning

**Frontend:** Stream lit or flask

**Python Libraries:**

Numpy , pandas ,matplotlib ,seaborn ,sklearn

**Functional Requirements:**

**Platooning Control System:**

**Description:** The system should be able to control and coordinate the movement of vehicles in a platoon to achieve fuel efficiency.

**Features:**

Establish and maintain communication between vehicles in the platoon.

Adjust speed and distance between vehicles dynamically.

Initiate and terminate platooning based on traffic conditions.

**Machine Learning Prediction Module:**

**Description:** The system should utilize machine learning models to predict future traffic and environmental conditions.

**Features:**

Train and update machine learning models based on real-time and historical data.

Provide accurate predictions of traffic density, road conditions, and other relevant factors.

Integrate seamlessly with the platooning control system.

**Energy Efficiency Optimization:**

**Description:** Optimize fuel efficiency by leveraging machine learning predictions.

**Features:**

Adjust platooning parameters based on predicted traffic and road conditions.

Minimize fuel consumption by optimizing speed and following distance.

Provide real-time feedback to drivers on fuel-efficient driving behaviors.

**Safety Mechanisms:**

**Description:** Ensure the safety of the platooning system and its participants.

**Features:**

Implement collision detection and avoidance algorithms.

Monitor vehicle health and alert drivers or initiate safety protocols if issues arise.

Integrate with existing safety systems such as ABS (Anti-lock Braking System) and ESC (Electronic Stability Control).

**Non-Functional Requirements:**

**Performance:**

**Description:** The system should perform efficiently and respond quickly to changing conditions.

**Criteria:**

Platooning adjustments should be made within milliseconds.

Machine learning predictions should have a high accuracy rate.

**Reliability:**

**Description:** The system should be reliable and available under various conditions.

**Criteria:**

Minimal downtime for maintenance or updates.

The platooning system should have failover mechanisms in case of component failures.

**Scalability:**

**Description:** The system should be able to handle a variable number of vehicles in a platoon.

**Criteria:**

Support platoons of varying sizes, from a few vehicles to a large convoy.

**Security:**

**Description:** Protect the system from unauthorized access and ensure data integrity.

**Criteria:**

Implement secure communication protocols.

Regularly update and patch system components to address security vulnerabilities.

**Usability**:

**Description:** The system should be user-friendly for both drivers and maintenance personnel.

**Criteria:**

Intuitive user interfaces for drivers to engage and disengage platooning.

Clear error messages and diagnostic tools for maintenance personnel.

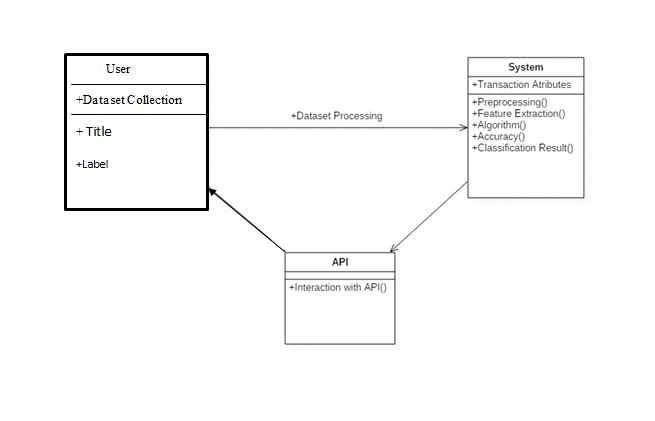
**Regulatory Compliance:**

**Description:** Ensure that the system complies with relevant regulations and standards.

**Criteria:**

Adherence to traffic safety regulations.

Compliance with data privacy and security standards.



**Modules:**

**1. DATA COLLECTION**

**2. DATA PRE-PROCESSING**

**3. FEATURE EXTRATION**

**4. EVALUATION MODE**

**DATA COLLECTION**

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into machine learning models.

Data used in this paper is a set of cervical cancer data with 15 features . This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called labelled data.

**DATA PRE-PROCESSING**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.

Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**FEATURE EXTRATION**

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

**EVALUATION MODEL**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance.

Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

4.3 UML Diagrams

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

l actors

l business processes

l (logical) components

l activities

l programming language statements

l database schemas, and

l Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems**.**

**Sequence Diagram:**

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

**Activity Diagrams-:**

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.

**Usecase diagram:**

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard.

UML stands for Unified Modeling Language.

UML is a pictorial language used to make software blue prints

**Class diagram**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase**.**

**Functional Requirements:**

**1. Data Collection and Integration:**

a. The system should collect and integrate relevant medical data, including vital signs, medical history, and any available data from monitoring devices.

b. Ensure compatibility with standard healthcare data formats and protocols.

**2. Feature Selection and Engineering:**

a. Identify key features influencing the early-stage prediction of cardiac arrest in new-borns, such as heart rate, oxygen saturation, and respiratory rate.

b. Implement feature engineering techniques to extract meaningful patterns and relationships from the selected features.

**3. Machine Learning Model Development:**

a. Develop a machine learning model capable of early-stage prediction of cardiac arrest in new-borns.

b. Choose appropriate algorithms, such as classification models or ensemble methods, considering the sensitivity and specificity required for medical predictions.

**4.Training and Testing:**

a. Implement a mechanism for training the machine learning model using a diverse dataset that includes cases of cardiac arrest and non-cardiac arrest scenarios in new-borns.

b. Set up a robust testing environment to evaluate the model's performance, including sensitivity, specificity, and accuracy.

**5. Real-time Monitoring:**

a. Enable real-time monitoring of vital signs and other relevant data to provide timely predictions.

b. Ensure the system can handle continuous data streams and update predictions dynamically.

**6. User Interface for Healthcare Professionals:**

a. Develop a user interface tailored for healthcare professionals to input patient data, view predictions, and receive alerts.

b. Include visualizations and trend analysis to aid in decision-making.

**7. Alerts and Notifications:**

a. Implement an alert system to notify healthcare professionals of potential cardiac arrest risks.

b. Allow customization of alert thresholds based on hospital protocols and individual patient conditions.

**Non Functional Requirements:**

**1. Performance:**

a. The system should be capable of processing and analyzing medical data rapidly, providing predictions within seconds.

b. It should handle a large volume of incoming data without significant degradation in performance.

**2. Reliability:**

a. The system must be highly reliable, with a minimal occurrence of false positives and false negatives in predicting cardiac arrest.

b. The prediction accuracy should meet or exceed medical standards for early stage recommendations.

**3. Availability:**

a. The system should be available 24/7 to ensure continuous monitoring and early detection of potential cardiac arrest cases in newborns.

b. Downtime for maintenance or updates should be scheduled during periods of low system usage.

**4. Scalability:**

a. The system should be designed to scale easily to accommodate an increasing number of new-borns and medical data.

b. It should be able to handle growth in both the number of users and the volume of historical and real-time data.

**5. Usability:**

a. The user interface should be intuitive and user-friendly, considering that healthcare professionals with varying technical expertise will use the system.

b. Training requirements for healthcare professionals to use the system should be minimal.

**6 . Interoperability:**

a. The system should integrate seamlessly with existing electronic health record (EHR) systems and other healthcare information systems.

b. It should adhere to relevant healthcare data exchange standards to ensure interoperability with various healthcare providers.

**7. Security:**

a. The system should comply with healthcare data security standards (e.g., HIPAA) to protect patient confidentiality and privacy.

b. Access to sensitive medical data should be restricted to authorized healthcare professionals.

**8. Maintainability:**

a. The system should be designed for easy maintenance and updates, with clear documentation for troubleshooting and support.

b. Code should be well-organized and well-commented to facilitate future enhancements or modifications.

**9. Regulatory Compliance:**

a. The system must comply with relevant healthcare regulations and standards governing the use of machine learning in medical diagnosis.

b. Regular audits should be conducted to ensure on-going compliance with evolving regulations.

**10. Ethical Considerations:**

a. The system should prioritize patient welfare, and its use should adhere to ethical guidelines and principles.

b. Considerations for transparency in the decision-making process of the machine learning model should be addressed.

**11. Training and Support:**

a. Provide training resources and documentation for healthcare professionals on how to interpret and act upon the system's predictions.

b. Establish a support system for addressing user inquiries, issues, and providing timely assistance.

Pseudocode:

# Module 1: Data Collection and Preprocessing

module DataCollectionAndPreprocessing:

# Function to collect and preprocess historical and real-time data

function collect\_and\_preprocess\_data():

# Collect historical and real-time data related to platooning and environmental conditions

historical\_data = collect\_historical\_data()

real\_time\_data = collect\_real\_time\_data()

# Preprocess the collected data (cleaning, normalization, feature engineering, etc.)

preprocessed\_data = preprocess\_data(historical\_data, real\_time\_data)

return preprocessed\_data

# Module 2: Feature Engineering

module FeatureEngineering:

# Function to extract relevant features

function extract\_features(data):

# Extract features from the preprocessed data

features = feature\_extraction(data)

return features

# Module 3: Future Conditions Prediction

module FutureConditionsPrediction:

# Function to train machine learning model for future conditions prediction

function train\_prediction\_model(features, future\_conditions):

# Train a machine learning model to predict future conditions based on historical and real-time data

prediction\_model = train\_model(features, future\_conditions)

return prediction\_model

# Module 4: Platooning Strategy Optimization

module PlatooningStrategyOptimization:

# Function to optimize platooning strategies based on predicted future conditions

function optimize\_platooning\_strategy(prediction\_model, current\_conditions):

# Use the prediction model to predict future conditions

predicted\_future\_conditions = predict\_future\_conditions(prediction\_model, current\_conditions)

# Optimize platooning strategies based on predicted future conditions

optimized\_strategy = optimize\_strategy(predicted\_future\_conditions)

return optimized\_strategy

# Main Program

# Collect and preprocess historical and real-time data

preprocessed\_data = DataCollectionAndPreprocessing.collect\_and\_preprocess\_data()

# Extract features from the preprocessed data

features = FeatureEngineering.extract\_features(preprocessed\_data)

# Train a machine learning model for future conditions prediction

prediction\_model = FutureConditionsPrediction.train\_prediction\_model(features, future\_conditions)

# Obtain current conditions

current\_conditions = get\_current\_conditions()

# Optimize platooning strategy based on predicted future conditions

optimized\_strategy = PlatooningStrategyOptimization.optimize\_platooning\_strategy(prediction\_model, current\_conditions

**Domain Specification:**

**MACHINE LEARNING**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

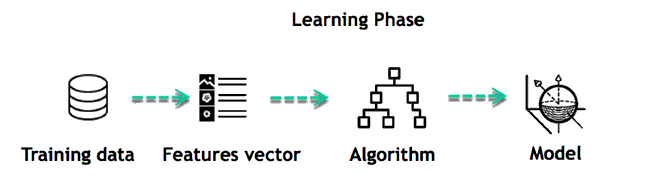
Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning**and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

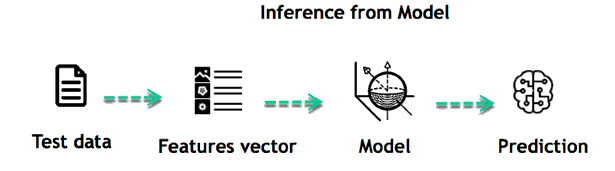
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

*Inferring*

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

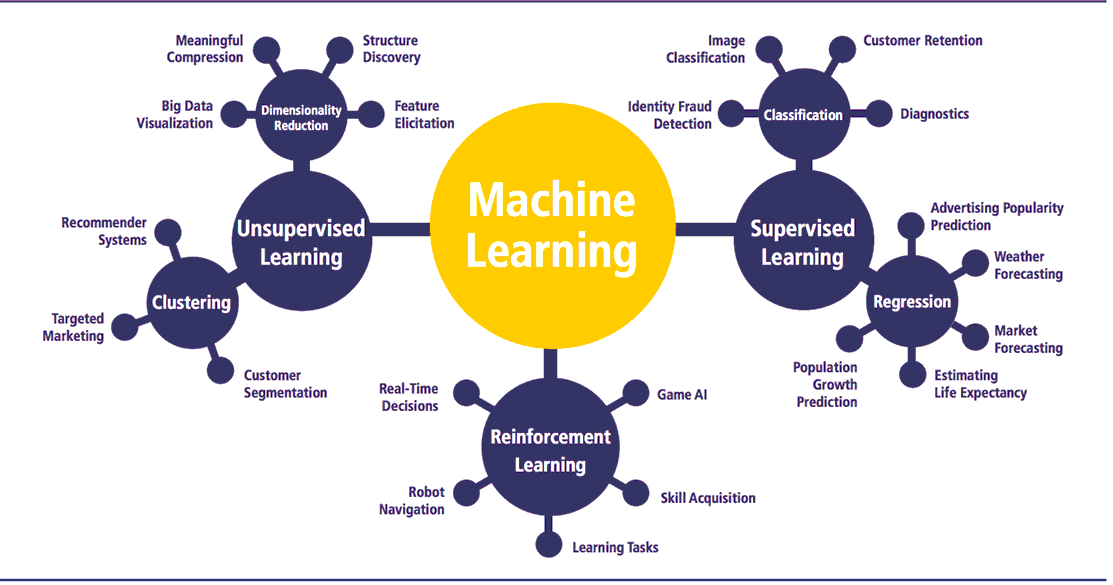


The life of Machine Learning programs is straightforward and can be summarized in the following points:

* Define a question
* Collect data
* Visualize data
* Train algorithm
* Test the Algorithm
* Collect feedback
* Refine the algorithm
* Loop 4-7 until the results are satisfying
* Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Machine learning Algorithms and where they are used?



Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

*Supervised learning*

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

* Classification task
* Regression task

***Classifications:***

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

***Regression***

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

*Unsupervised learning*

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Description** | **Type** |
| **K-means clustering** | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| **Gaussian mixture model** | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters | Clustering |
| **Hierarchical clustering** | Splits clusters along a hierarchical tree to form a classification system.  Can be used for Cluster loyalty-card customer | Clustering |
| **Recommender system** | Help to define the relevant data for making a recommendation. | Clustering |
| **PCA/T-SNE** | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

**Application of Machine learning**

**Augmentation**:

* Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

* Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

* Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

* The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

* Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

* Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:**The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

**AI in HR:**Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:**AI is a valuable tool for customer service management  
and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

Difference between Machine Learning and Deep Learning

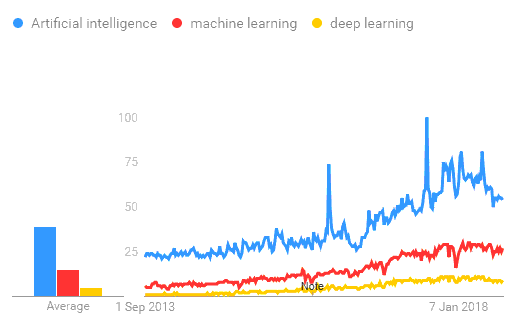
|  |  |  |
| --- | --- | --- |
|  | **Machine Learning** | **Deep Learning** |
| **Data Dependencies** | Excellent performances on a small/medium dataset | Excellent performance on a big dataset |
| **Hardware dependencies** | Work on a low-end machine. | Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication |
| **Feature engineering** | Need to understand the features that represent the data | No need to understand the best feature that represents the data |
| **Execution time** | From few minutes to hours | Up to weeks. Neural Network needs to compute a significant number of weights |
| **Interpretability** | Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost) | Difficult to impossible |

When to use ML or DL?

|  |  |  |
| --- | --- | --- |
|  | **Machine learning** | **Deep learning** |
| **Training dataset** | Small | Large |
| **Choose features** | Yes | No |
| **Number of algorithms** | Many | Few |
| **Training time** | Short | Long |

In the table below, we summarize the difference between machine learning and deep learning.

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.



TensorFlow

the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

* Researchers
* Data scientists
* Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

**TensorFlow Architecture**

Tensor flow architecture works in three parts:

* Pre-processing the data
* Build the model
* Train and estimate the model

It is called Tensor flow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can Tensor flow run?**

TensorFlow can hardware, and software requirements can be classified into

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done TensorFlow can be run on many different platforms. You can run it on

* Desktop running Windows, macOS or Linux
* Cloud as a web service
* Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of Tensor Flow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

* Linear regression: tf. estimator .Linear Regressor
* Classification :tf. Estimator .Linear Classifier
* Deep learning classification: tf. estimator. DNN Classifier
* Booster tree regression: tf.estimator.BoostedTreesRegressor
* Boosted tree classification: tf.estimator.BoostedTreesClassifier

**PYTHON OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **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 interactwith the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Features**

Python's features include:

**Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.

**Easy-to-read:** Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

**A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.

**Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.

**Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.

**Extendable:** You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.

**Databases:** Python provides interfaces to all major commercial databases.

**GUI Programming:** Python supports GUI applications that can be created andported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable:** Python provides a better structure and support for large programsthan shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* IT supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

**WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR**?

The following applications are available by default in Navigator:

* Jupyter Lab
* Jupyter Notebook
* QT Console
* Spyder
* VS Code
* Glue viz
* Orange 3 App
* Rodeo
* RStudio

Advanced conda users can also build your own Navigator applications

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

What’s new in 1.9?

* Add support for **Offline Mode** for all environment related actions.
* Add support for custom configuration of main windows links.
* Numerous bug fixes and performance enhancements.

**TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

* Meets the business and technical requirements that guided its design and Development.
* Works as expected and can be implemented with the same characteristics.

**TESTING METHODS**

* **Functional Testing**

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:

* Functions: Identified functions must be exercised.
* Output: Identified classes of software outputs must be exercised.
* Systems/Procedures: system should work properly

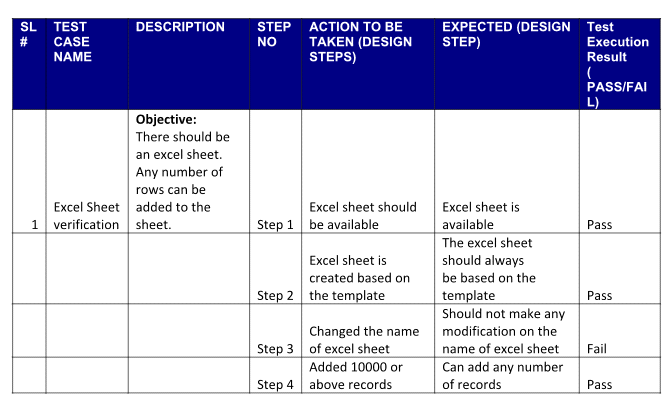
**Integration Testing**

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.

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

Test Case 1 :



**REFERENCES:**

[1] A. Al Alam, A. Gattami, and K. H. Johansson, “An experimental

study on the fuel reduction potential of heavy duty vehicle platoon-

ing,” in Proc. 13th Int. IEEE Conf. Intell. Transp. Syst. (ITSC), 2010,

pp. 306–311.

[2] S. Tsugawa, S. Jeschke, and S. E. Shladover, “A review of truck pla-

tooning projects for energy savings,” IEEE Trans. Intell. Veh., vol. 1,

no. 1, pp. 68–77, Mar. 2016.

[3] A. Kousaridas et al., “QoS prediction for 5G connected and automated

driving,” IEEE Commun. Mag., vol. 59, no. 9, pp. 58–64, Sep. 2021.

[4] L. Zhang, F. Chen, X. Ma, and X. Pan, “Fuel economy in truck

platooning: A literature overview and directions for future research,”

J. Adv. Transp., vol. 2020, Jan. 2020, Art. no. 2604012. [Online].

Available: <https://www.hindawi.com/journals/jat/2020/2604012/>

[5] B. McAuliffe, M. Lammert, X.-Y. Lu, S. Shladover, M.-D. Surcel,

and A. Kailas, “Influences on energy savings of heavy trucks using

cooperative adaptive cruise control,” SAE, Warrendale, PA, USA,

Rep. 2018-01-1181, 2018.

[6] S. Torabi and M. Wahde, “Fuel-efficient driving strategies for

heavy-duty vehicles: A platooning approach based on speed profile

optimization,” J. Adv. Transp., vol. 2018, Sep. 2018, Art. no. 4290763.

[7] M. Di Bernardo, A. Salvi, and S. Santini, “Distributed consensus

strategy for platooning of vehicles in the presence of time-varying het-

erogeneous communication delays,” IEEE Trans. Intell. Transp. Syst.,

vol. 16, no. 1, pp. 102–112, Feb. 2015.