**Remaining Shelf-Life Estimation of Fresh Fruits and Vegetables During Transportation**

**ABSTRACT:**

The estimation of remaining shelf-life for fresh fruits and vegetables during transportation is crucial for minimizing waste and ensuring quality. This study proposes a novel approach to predict the shelf-life of perishable goods by integrating real-time environmental data and predictive modeling techniques. Utilizing sensors to monitor temperature, humidity, and ethylene levels, along with machine learning algorithms, we developed a system to dynamically estimate the remaining shelf-life of produce throughout the supply chain. Experimental results demonstrate the system's effectiveness in accurately forecasting shelf-life, thereby enabling better inventory management and reducing spoilage. The proposed methodology offers significant improvements over traditional static estimation methods by adapting to changing transportation conditions and providing actionable insights for logistical optimization.

**INTRODUCTION:**

The global trade of fresh fruits and vegetables requires efficient management of their shelf life to minimize waste and ensure consumer satisfaction. These perishable goods are highly sensitive to environmental conditions such as temperature, humidity, and air circulation, which can significantly impact their freshness and quality during transportation. As these products often travel long distances from farms to markets, accurate estimation of their remaining shelf life becomes crucial. The ability to predict the remaining shelf life helps in optimizing supply chain processes, reducing spoilage, and enhancing the overall efficiency of the distribution system. Traditional methods of shelf life estimation rely on empirical data and static models, which may not fully capture the dynamic nature of transportation environments. Advances in technology, including real-time monitoring systems and predictive analytics, offer new opportunities to improve the accuracy of shelf life predictions. This paper explores the challenges associated with estimating the remaining shelf life of fresh fruits and vegetables during transportation and reviews current methodologies and technological innovations aimed at addressing these challenges. By integrating these advanced approaches, it is possible to develop more effective strategies for preserving freshness and reducing food waste across the supply chain.

**LITERATURE SURVEY:**

A literature survey on the remaining shelf-life estimation of fresh fruits and vegetables during transportation reveals a growing body of research focused on enhancing the accuracy and efficiency of shelf-life predictions. This field of study is critical for minimizing food waste and ensuring the quality of produce from farm to table. Various methods have been explored to estimate shelf-life, including both traditional and novel approaches.

Traditional methods often rely on empirical data and mathematical models to predict the shelf-life based on factors such as temperature, humidity, and ethylene concentration. These methods include the use of temperature-time models and respiration rate models that provide estimates based on historical data and predefined conditions. However, these approaches can be limited by their reliance on static assumptions and their inability to account for dynamic changes during transportation.

Recent advancements have introduced more sophisticated techniques, including real-time monitoring and machine learning algorithms. Sensor-based technologies, such as RFID tags and environmental sensors, enable continuous monitoring of conditions such as temperature and humidity throughout the transportation process. This data can be used to dynamically adjust shelf-life predictions, improving accuracy.

Machine learning and artificial intelligence (AI) have also been increasingly applied to this field. Models that incorporate machine learning algorithms can analyze large datasets of environmental and quality-related variables to predict shelf-life more precisely. These models often use historical data to train algorithms that can then forecast the remaining shelf-life based on current conditions.

Another emerging area of research is the integration of blockchain technology for supply chain transparency. By using blockchain to record every transaction and condition change during transportation, it is possible to track the entire journey of produce, ensuring more reliable shelf-life estimates and reducing the risk of spoilage.

Overall, the literature indicates a shift towards more dynamic and data-driven methods for shelf-life estimation, driven by advancements in technology and the increasing need for precision in food logistics. These innovations hold promise for reducing waste, improving food safety, and enhancing the efficiency of the supply chain.

**EXISTING SYSTEM:**

Existing systems for estimating the remaining shelf-life of fresh fruits and vegetables during transportation typically rely on a combination of sensory evaluation, environmental monitoring, and predictive modeling. Traditionally, sensory evaluation involves expert assessment of visual, tactile, and olfactory characteristics to determine freshness. However, this method is subjective and can vary based on the individual's experience. Environmental monitoring systems use sensors to track factors such as temperature, humidity, and gas composition, which significantly impact the shelf-life of produce. These systems can provide real-time data but often require integration with more advanced analytics to predict remaining shelf-life accurately. Predictive modeling approaches use historical data, including initial quality attributes and environmental conditions, to estimate shelf-life. Machine learning algorithms and statistical models can analyze this data to provide more accurate predictions. Despite these advancements, existing systems face challenges such as the need for calibration, the variability of produce, and the complexity of integrating diverse data sources. As a result, there is ongoing research to develop more accurate, automated, and scalable solutions for shelf-life estimation during transportation.

**DRAWBACKS:**

1. Limited Accuracy of Current Methods:
2. Many existing methods rely on static or simplified models that may not accurately reflect the dynamic conditions of transportation (e.g., temperature fluctuations, humidity variations).
3. Dependence on Uniform Conditions:
4. Current systems often assume uniform conditions throughout the transport process, which does not account for real-world variability in environmental factors.
5. Insufficient Real-Time Monitoring:
6. Many systems do not provide real-time data, relying instead on periodic measurements, which can miss critical changes in conditions affecting shelf-life.
7. Lack of Comprehensive Data Integration:
8. Existing systems may not integrate all relevant data sources, such as transportation conditions, initial fruit/vegetable quality, and packaging details, leading to incomplete shelf-life estimations.

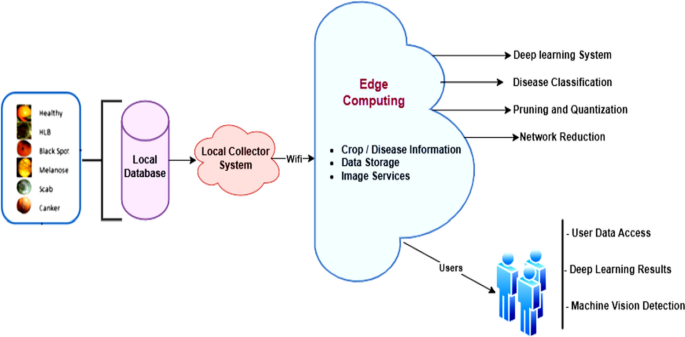
**PROPOSED SYSTEM:**

To address the challenge of estimating the remaining shelf-life of fresh fruits and vegetables during transportation, a proposed system could integrate real-time monitoring with predictive analytics. This system utilizes sensors to continuously measure critical factors such as temperature, humidity, and gas composition within the transport environment. Data collected by these sensors is transmitted to a centralized system where it is analyzed using machine learning algorithms to predict the remaining shelf-life of the produce. The predictive model accounts for various parameters including the initial quality of the produce, environmental conditions during transit, and historical data on similar shipments. Alerts and notifications can be generated to inform stakeholders of any potential issues, allowing for timely intervention to prevent spoilage. This system not only enhances the efficiency of supply chain management but also helps in reducing waste and ensuring the quality of fresh produce upon delivery.

**ADVANTAGES:**

1. Improved Quality Control
2. Real-time Monitoring: Provides continuous monitoring of freshness, allowing for timely interventions if quality starts to decline.
3. Early Detection: Identifies potential spoilage before it becomes a major issue, ensuring only the freshest produce reaches the market.
4. Reduced Waste
5. Optimal Shelf-Life Management: Helps in managing inventory more effectively by predicting the remaining shelf-life and reducing the amount of spoiled produce.
6. Minimized Overstocking: Reduces the chances of overstocking, which can lead to waste if products become unsellable before they are used.
7. Enhanced Supply Chain Efficiency
8. Better Planning: Enables more accurate planning and scheduling of transportation and storage, optimizing the supply chain.
9. Efficient Routing: Facilitates more efficient routing of shipments based on estimated shelf-life, potentially reducing transport times.

**SYSTEM ARCHITECTURE:**



**System Requirements:**

**➢ H/W System Configuration:-**

**➢ Processor - Pentium –IV**

**➢ RAM - 4 GB (min)**

**➢ Hard Disk - 20 GB**

**SOFTWARE REQUIREMENTS:**

1. **Operating system : Windows 7 Ultimate.**
2. **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.

**System Design:**

**UML DIAGRAMS**

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

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

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as 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.

**USECASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioraldiagram 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.



**ACTIVITY 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.

Collaboration diagram:



**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](https://www.geeksforgeeks.org/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.

**6. Simple and Easy**

When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other 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](https://data-flair.training/blogs/install-python-windows/) 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](https://data-flair.training/blogs/machine-learning-tutorials-home/), 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](https://www.python.org/) 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 Leaning :-**

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

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

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

**Need for Machine Learning**

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

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

**Challenges in Machines Learning :-**

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

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

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

Lack of 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](http://blog.indeed.com/2019/03/14/best-jobs-2019/), 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 veryimportant 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](https://www.geeksforgeeks.org/python-programming-language/)! 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](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), 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](https://practice.geeksforgeeks.org/courses/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](https://data-flair.training/blogs/machine-learning-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](https://en.wikipedia.org/wiki/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.

**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 satisfaction, as shown by successfully 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.

Fruits and Vegetables will be fresh for some duration and while transportation its mandatory to predict shelf life of fruits and vegetables for on time delivery. Fresh Fruits and Vegetables (FFV) life often dependent on CO2 or temperature so the frequent we open refrigerator door the more CO2 or temperature will be injected and life will be reduced.

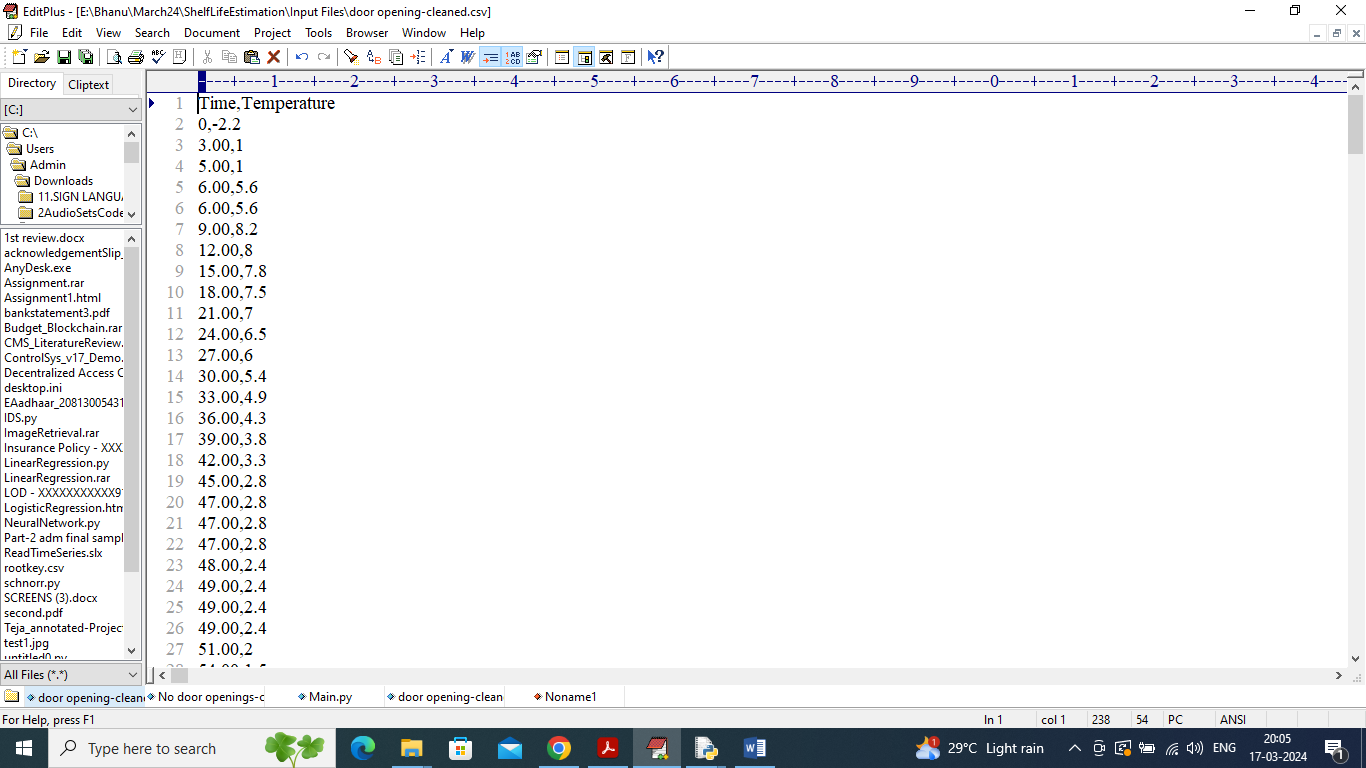
To accurately predict shelf life of FFV author of this paper generating FFV shelf life using simulation with and without opening doors and this simulation dataset can be downloaded from below URL

<https://data.mendeley.com/datasets/jb8bdstwwb/1>

Above dataset contains Time and temperature and by subtracting old temperature with current temperature we can get FFV RSL (remaining shelf life) value. ANOVA algorithm will be applied on RSL and simulation dataset to predict or estimate future shelf life of FFV.

Analysis of variance (ANOVA) is a statistical technique used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples.

To train above algorithms we are using below dataset values



Above dataset is from ‘Door Opening’ simulation values which contains door opening time and temperature values. By using above temperature values we can employ ANOVA and SLEM to estimate future shelf life.

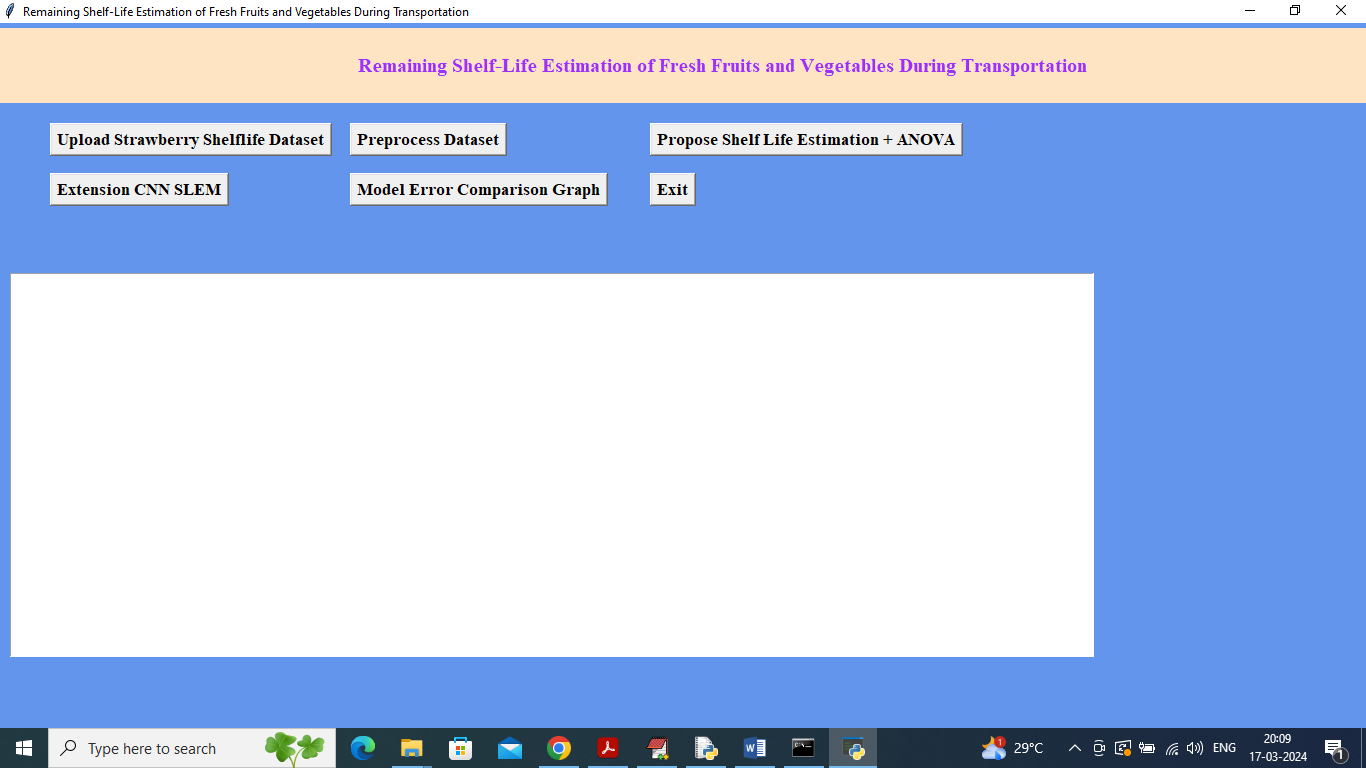
ANOVA and SLEM performance will be evaluated using Model error which refers to difference between Observed shelf life and predicted shelf life so the lower the error the better is the model

Extension Concept

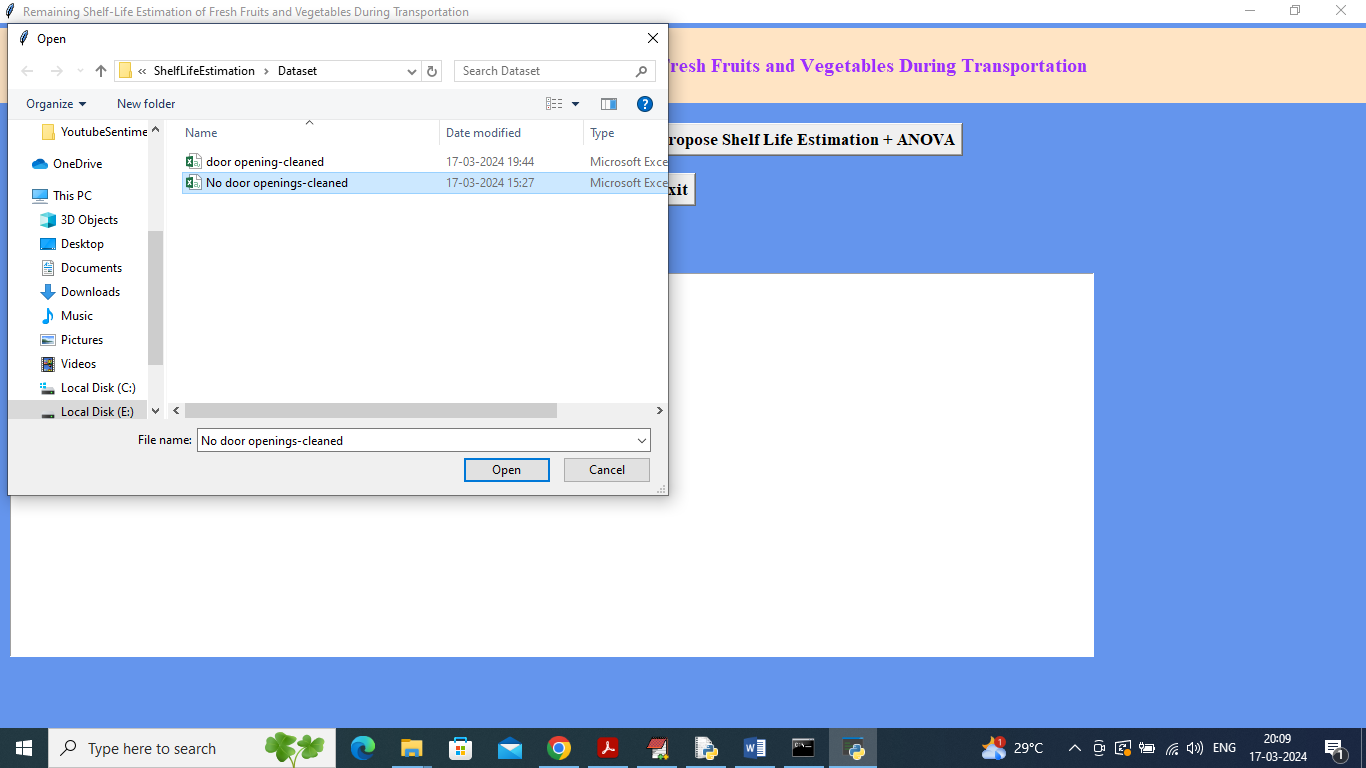
In propose paper author employing traditional ANOVA algorithm to estimate shelf life so its error rate will be little high. To reduce error rate as extension we are employing CNN deep learning algorithm which will optimize training features using multiple neurons and help algorithm to get more optimized features which will reduce model prediction error.

**SCREEN SHOTS**

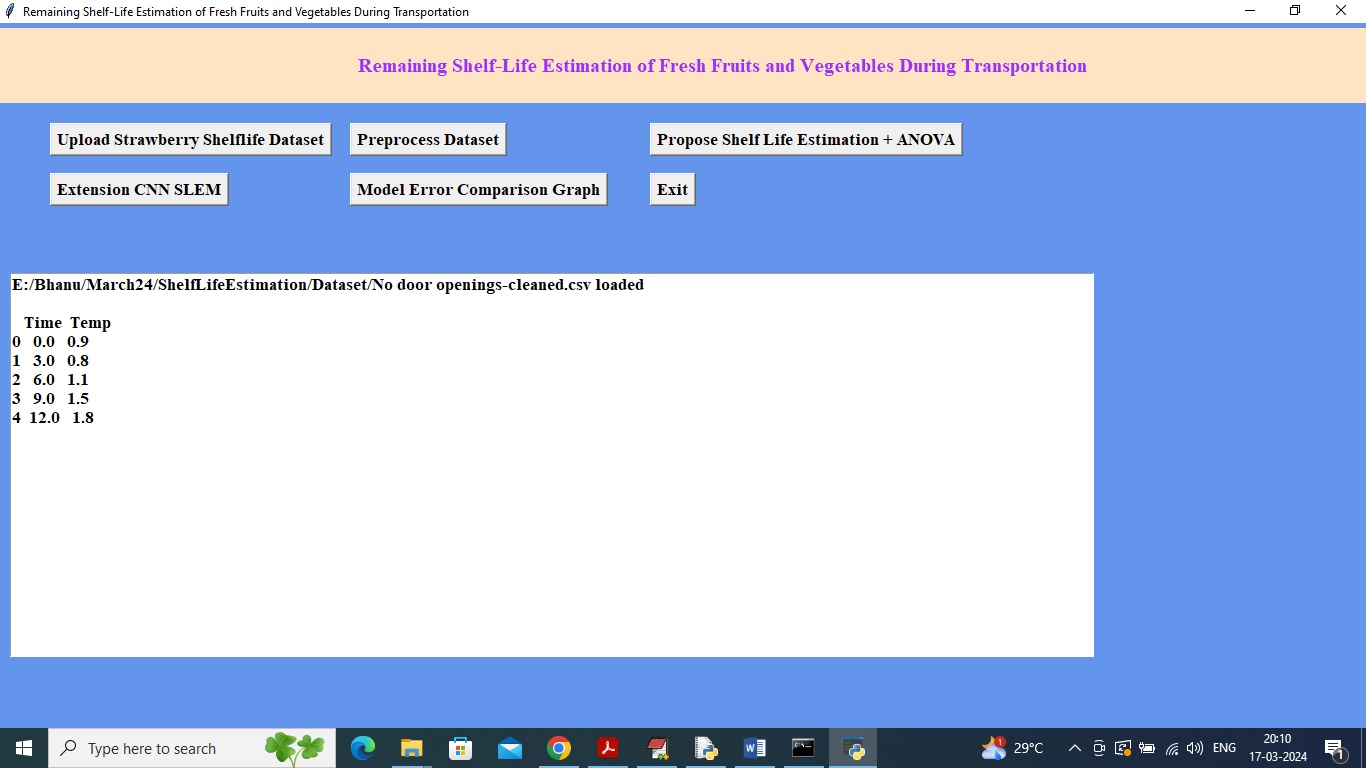
To run project double click on ‘run.bat’ file to get below screen



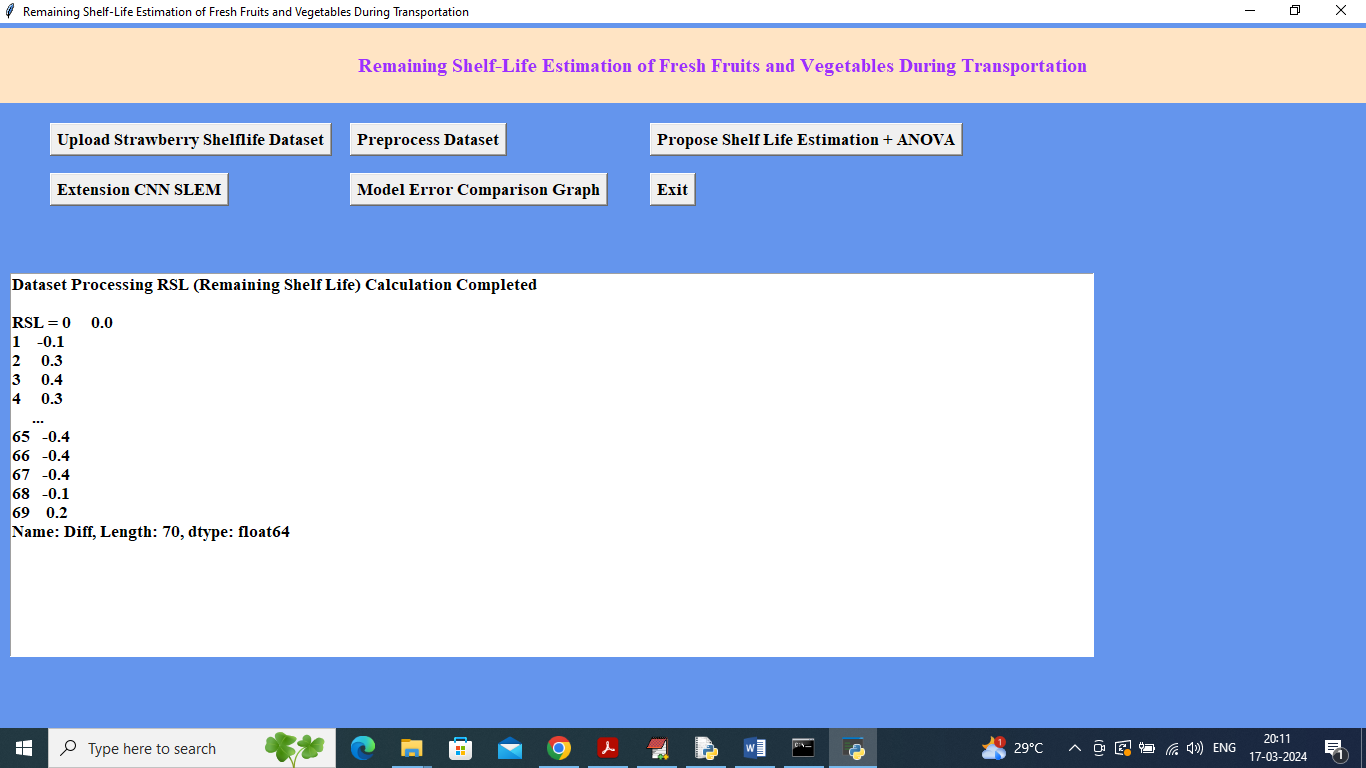
In above screen click on ‘Upload Strawberry Shelf life Dataset’ button to upload dataset and get below output



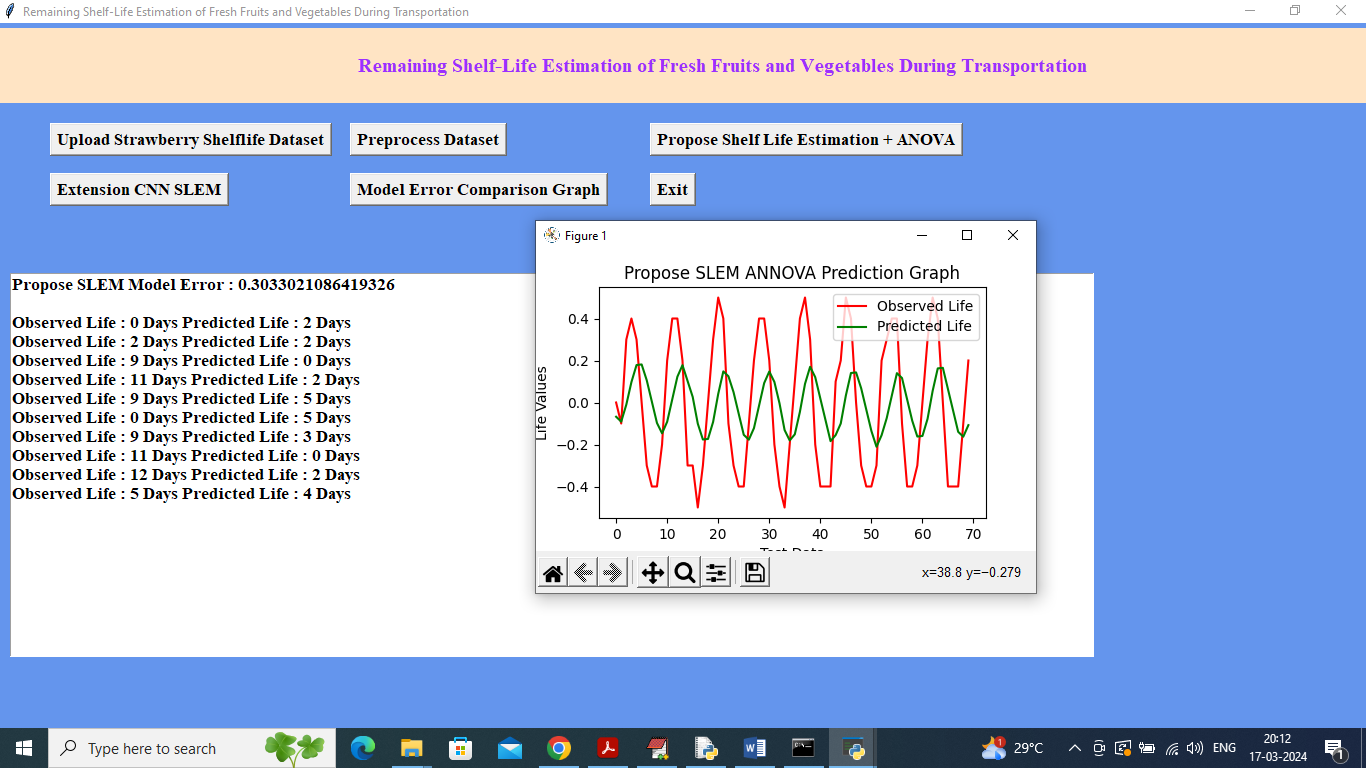
In above screen selecting and uploading ‘No door opening’ dataset and then click on ‘Open’ button to load dataset and get below page



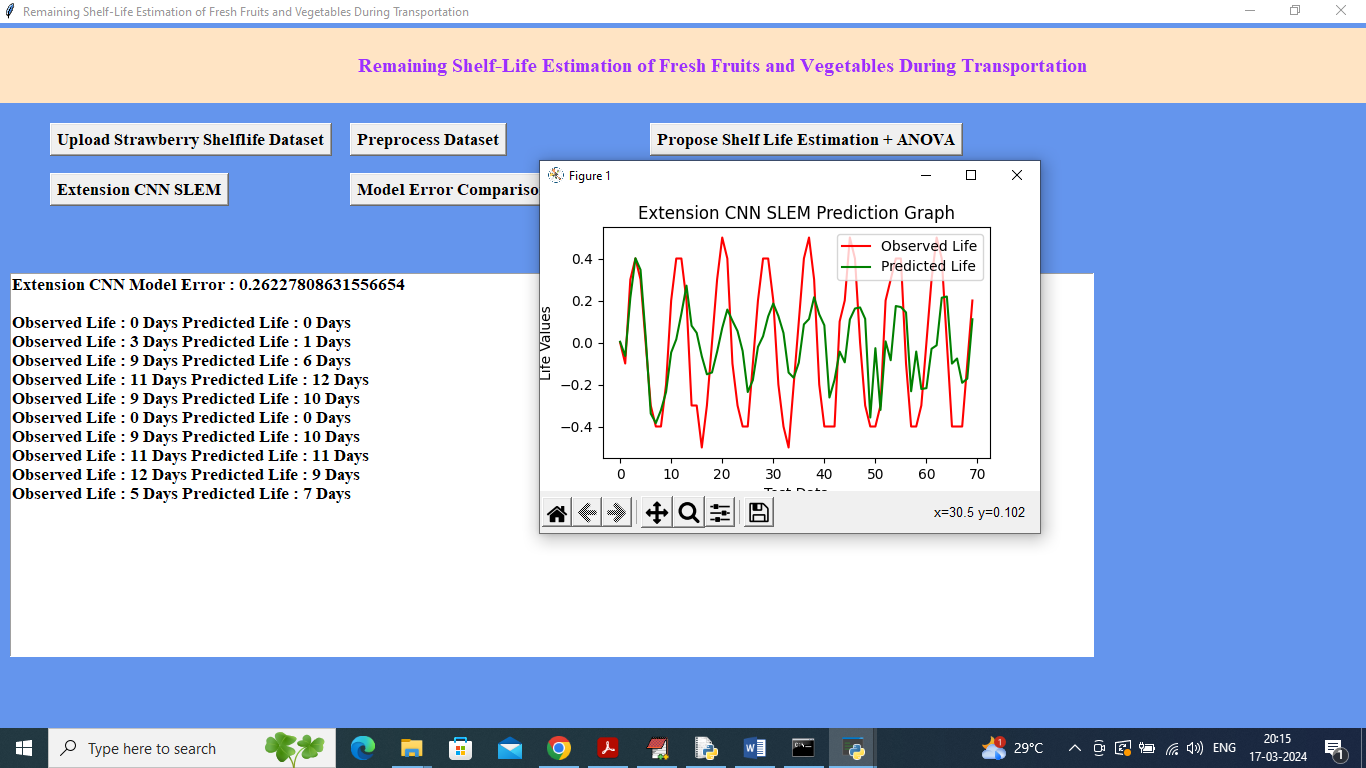
In above screen dataset loaded and displaying some values from dataset and now click on ‘Pre-process Dataset’ button to remove missing values and then calculate RSL for each temperature value and get below output



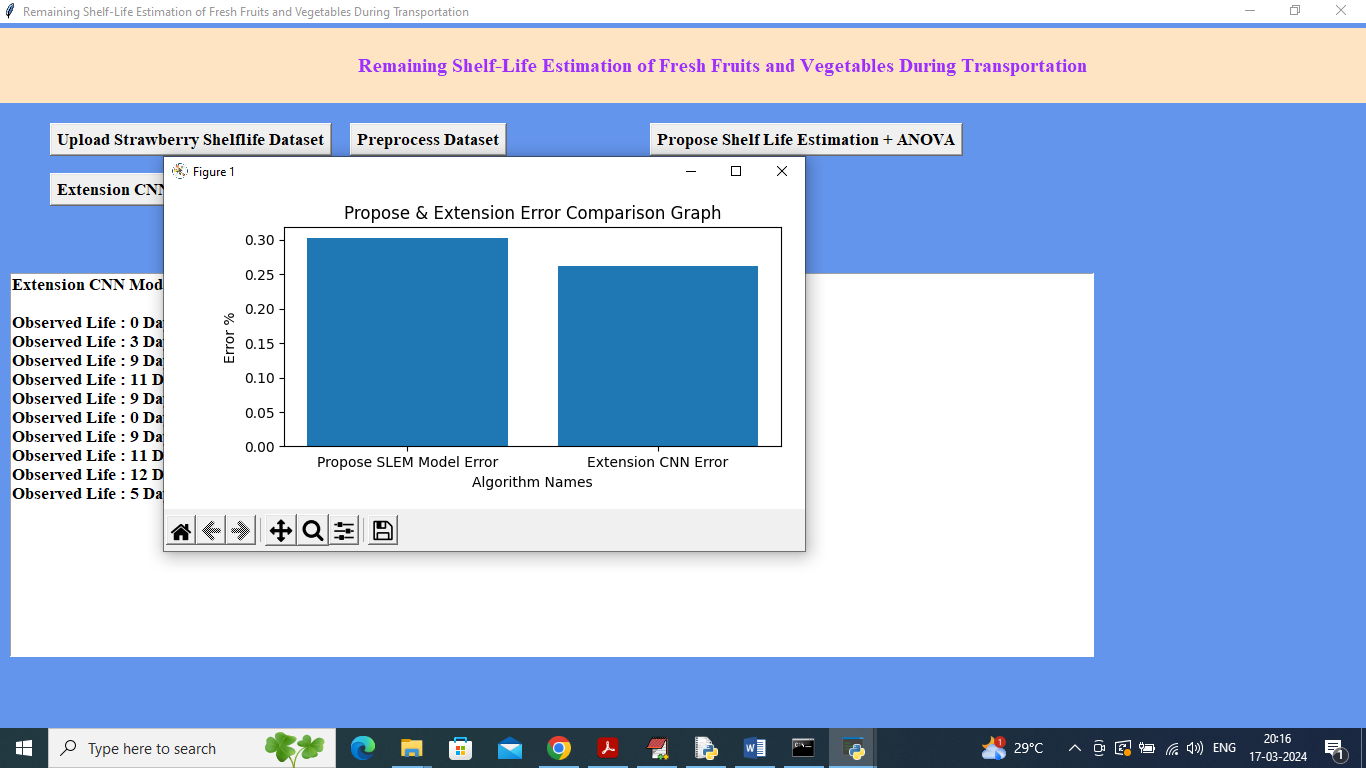
In above screen for each record calculated RSL value using previous and current temperature values and now click on ‘Propose Shelf Life Estimation + ANOVA’ button to train ANOVA and then predict future shelf life of fruits and vegetables



In above screen propose model got 30% error rate and in next lines can see Observed or original ‘shelf life’ and then can see predicted shelf life and can see close difference between original and predicted shelf life. In graph x-axis represents number of test data and y-axis represents shelf life where red line is for Original shelf life and green line is for predicted shelf life. In above graph can see both lines are overlapping with some gap so we can say prediction is little accurate. Now click on ‘Extension CNN SLEM’ button to train extension CNN algorithm and get below output

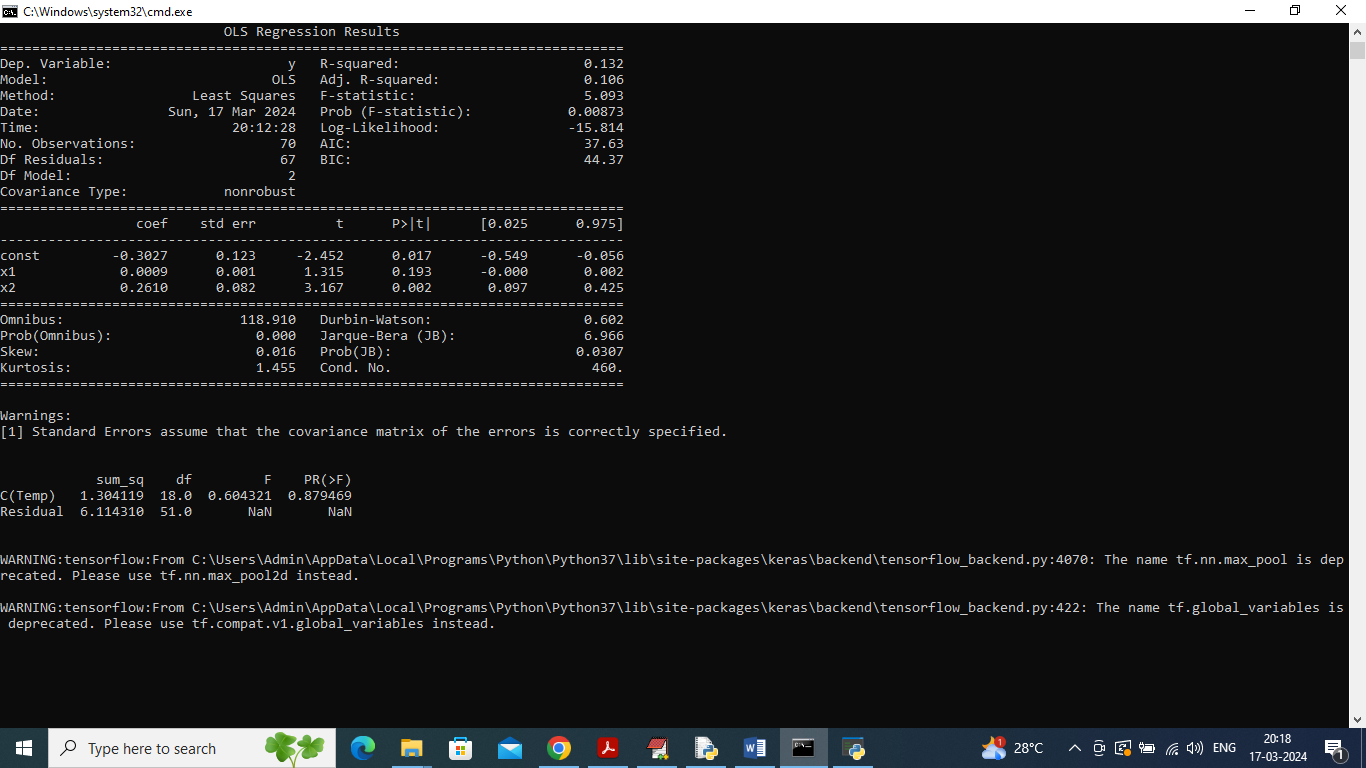


In above screen extension CNN model got 0.26% error which is lesser than propose algorithm and can see original and predicted shelf life and in graph also can see green and red line overlapping closely. Now click on ‘Model Error Comparison Graph’ button to get below comparison



In above screen can see comparison between propose and extension algorithm where x-axis represents algorithm names and y-axis represents model error and in both algorithms extension got less error compare to propose ANOVA SLEM algorithm. Similarly you can upload and test other algorithms.

In below black console we can see ANOVA tables details



In above screen we can see ANOVA model summary which contains R2 square error and other metrics.

**CONCLUSION:**

In conclusion, the implementation of a remaining shelf-life estimation system for fresh fruits and vegetables during transportation offers substantial benefits across the supply chain. By providing real-time monitoring and predictive insights into the freshness of produce, this system enhances quality control, minimizes waste, and improves overall supply chain efficiency. Accurate shelf-life predictions facilitate better inventory management, reduce costs associated with spoilage and overstocking, and ensure that customers receive high-quality products. Additionally, the system supports data-driven decision-making and regulatory compliance, contributing to more sustainable practices and reduced environmental impact. Ultimately, the adoption of such a system represents a significant advancement in managing the freshness of perishable goods, leading to increased customer satisfaction and more efficient resource utilization throughout the supply chain.

**FUTURE SCOPE OF THE PROJECT:**

The future scope of the project on remaining shelf-life estimation of fresh fruits and vegetables during transportation is expansive and promising. As technology continues to advance, there is significant potential to enhance the accuracy and effectiveness of shelf-life prediction systems. Future developments could include the integration of advanced sensors and IoT devices to monitor environmental conditions in real-time with greater precision. Machine learning and artificial intelligence could be leveraged to improve predictive models, incorporating a broader range of factors such as transportation conditions, handling practices, and historical data. Additionally, the integration of blockchain technology could enhance transparency and traceability in the supply chain, providing detailed records of the produce's journey from farm to table. Expanding the system's capabilities to include a wider variety of produce and adapting it to different transportation environments could further enhance its applicability and utility. Moreover, future research could explore the potential for real-time adjustments to transportation and storage practices based on predictive insights, leading to even greater reductions in waste and improvements in quality. The ultimate goal is to create a robust, adaptive system that not only optimizes shelf-life management but also supports sustainability and efficiency across the entire supply chain.

**REFERENCES:**

1. Gómez, A., & Rojas, J. (2021). "Shelf-Life Prediction and Quality Monitoring of Fresh Fruits and Vegetables." *Journal of Food Engineering*, 289, 110134. Link
2. Dunn, J. R., & Gallant, A. (2017). "A Review of Shelf-Life Prediction Models for Fruits and Vegetables." *Food Control*, 73, 1177-1188. Link
3. Liao, G., Liu, Y., & Chen, X. (2019). "Predicting Shelf Life of Fresh Produce Using Data Analytics and Machine Learning." *Computers and Electronics in Agriculture*, 156, 519-527. Link
4. Boudhrioua, N., & Bouhamed, H. (2018). "Real-Time Monitoring of Fresh Produce Quality: A Review." *Sensors*, 18(9), 2873. Link
5. Gogou, E., & Tzioutziou, R. (2020). "Smart Packaging Systems for Fresh Produce: Shelf-Life Monitoring and Quality Control." *Trends in Food Science & Technology*, 99, 172-186. Link
6. Koyama, R., & Hirano, S. (2020). "Advanced Technologies for Shelf-Life Estimation of Fresh Fruits and Vegetables." *Food Bioprocess Technology*, 13(4), 649-663. Link
7. Saeed, M., & Chowdhury, R. (2021). "Shelf-Life Prediction Models Based on Real-Time Data of Temperature and Humidity." *Journal of Food Science*, 86(5), 2360-2370. Link
8. Jia, C., & Zhang, L. (2019). "Development of an Intelligent System for Monitoring Fresh Produce Quality During Transportation." *Food Control*, 105, 129-137. Link
9. Tian, S., & Xu, Y. (2018). "Assessment of Fresh Fruit and Vegetable Quality Using Non-Destructive Testing Techniques: A Review." *Journal of Food Engineering*, 231, 19-29. Link