Freshness of Food Detection Using Machine Learning and IOT

A MINI PROJECT REPORT

Submitted in partial fulfillment of the requirements

For the academic

Project

In

COMPUTER SCIENCE AND ENGINEERING

By

AASHRITH RACHERLA	**221710302001**
BVS MAHIDHAR	**221710302008**
MOTUPALLY SUNDARA CHARYA	**221710302038**
PRANAVI PULICHEARLA	**221710302050**

UNDER THE GUIDANCE OF

Mr. M. KIRAN SASTRY

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM (DEEMED TO BE UNIVERSITY)

HYDERABAD-502329

DECEMBER 2020

GITAM (Deemed to be university)
HYDERABAD

Rudraram village, Patancheru Mandal
Sangareddy Dist-502329, TS, INDIA.
Ph: 08455-220556/57; Fax: 08455-20046
Website: www.gitam.edu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project entitled **"Freshness of Food Detection"** is submitted by Aashrith Racherla (221710302001), BVS Mahidhar (221710302008), Motupally Sundara Charya (221710302038), Pranavi Pulichearla (221710302050) in partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering** at GITAM (Deemed to Be University), Hyderabad during the academic year 2020-21

It is the faithful record work carried out by them at the **Computer Science and Engineering Department**, GITAM University Hyderabad Campus under my guidance and supervision.

Mr. M.Kiran Sastry
Assistant Professor

Dr. S.Phani Kumar
Professor and HOD

Name and Signature of the project panel members.

1.
2.
3.

DECLARATION

We Aashrith Racherla (221710302001), BVS Mahidhar (221710302008), Motupally Sundara Charya (221710302038), Pranavi Pulichearla (221710302050) declare that the mini project report entitled "**Freshness of Food Detection**" under the guidance of **MR.M.KIRAN SASTRY**, Department of Computer Science and Engineering, School of Technology ,GITAM (Deemed to be University) , Hyderabad, has been submitted for mini project evaluation .

This is a record of bona fide work carried out by me and the content embodied in this project have not been reproduced /copied from any source. The content embodied in this project report has not been submitted to any other university or institute for the award of any other degree or diploma.

AASHRITH RACHERLA	**221710302001**
BVS MAHIDHAR	**221710302008**
MOTUPALLY SUNDARA CHARYA	**221710302038**
PRANAVI PULICHEARLA	**221710302050**

ACKNOWLEDGEMENT

Apart from our effort, the success of this mini project largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped us in the successful competition of this project.

We take this opportunity to place on record my sincere thanks to Honorable Pro-Vice Chancellor **Prof. N. SIVA PRASAD** GITAM School of Technology, GITAM Deemed to be University for his help and for providing all the required resources for completing mini project till stage.

We are thankful to **Dr. S.PHANI KUMAR**, The Head of the Department, Department of Computer Science and Engineering for the encouragement and support given to us throughout mini project.

We are grateful to the Internal Guide, **Mr. M.KIRAN SASTRY** and the Project Coordinator **Dr.S.APARNA**, for their valuable suggestions.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

ABSTRACT

In today's world, food spoilage is a crucial problem as consuming spoiled food is harmful for consumers. The food we consume provides nourishment and gives energy to our body which helps us to perform our day to day activities. A healthy and fresh diet is the most important way to keep ourselves fit. Our project aims at detecting spoiled food using appropriate sensors and monitoring gases released by the particular food item. The food quality tracking system based on the Internet of things is an integrated monitoring and management information system, which consists of intelligent database technology, radio frequency identification technology, food safety technology, network technology, as well as other practical high-tech techniques. A micro controller that senses this, issues an alert using internet of things, so that appropriate action can be taken. This has wide scale application in food industries where food detection is done manually. We plan on implementing machine learning to this model so we can estimate how likely a food is going to get spoiled and in what duration, if brought from a particular vendor. This will increase competition among retailers to sell more healthy and fresh food and create a safe world for all consumers alike.

Table of contents

1. INTRODUCTION 1

1.1 MOTIVATION 2
1.2 PROBLEM DEFINITION 2
1.3 OBJECTIVE 2
1.4 SCOPE OF THE PROJECT 3
1.5 APPLICATIONS 4

2. LITERATURE SURVEY 5

2.1 EXISTING SYSTEM 5
2.2 DISADVANTAGES OF EXISTING SYSTEM 7
2.3 PROPOSED SYSTEM 7
2.4 ADVANTAGES OF PROPOSED SYSTEM 8

3. ANALYSIS 9

3.1 INTRODUCTION 9
3.2 FEASIBILITY STUDY 9
3.2.1 TECHNICAL FEASIBILITY: 9
3.2.2 ECONOMIC FEASIBILITY: 10
3.3 REQUIREMENT ANALYSIS 10
3.4 SOFTWARE REQUIREMENT SPECIFICATION 11
3.4.1 USER REQUIREMENTS 11
3.4.2 SOFTWARE REQUIREMENTS 11
3.4.3 HARDWARE REQUIREMENTS 11

4. DESIGN 12

4.1 INTRODUCTION 12
4.2 ARCHITECTURE OF PROJECT 13
4.2.1 DATA COLLECTION 13
4.2.2 DATA MODELING 14

5. IMPLEMENTATION 16

5.1 DATA PREPARATION 16
5.1.1: GETTING THE DATASET: 16
5.1.2: IMPORTING THE LIBRARIES: 16

5.2 DATA SET 18
5.2.1 IMPORTING THE DATA-SET 18
5.3 DATA PREPROCESSING/FEATURE ENGINEERING 19
5.3.1: STATISTICAL ANALYSIS: 19
5.4: DATA TYPE CONVERSIONS 20
5.5: IDENTIFYING AND HANDLING MISSING VALUES 21
5.6: GENERATING PLOTS 22
5.6.1: VISUALIZE THE DATA BETWEEN TARGET AND THE FEATURES 22
5.7 ALGORITHMS USED 23
5.7.1 LOGISTIC REGRESSION 23
5.7.2 LINEAR REGRESSION 25
5.7.3 SVM (SUPPORT VECTOR MACHINE) 26

6. MODELLING 27
6.1 INTRODUCTION 27
6.2 LOGISTIC REGRESSION MODEL 29

7. TESTING AND VALIDATION 30
7.1 INTRODUCTION 30
7.2 EXPERIMENTATION 30
7.2.1TRAINING AND TESTING 31

8. RESULTS 34
8.1 LOGISTIC REGRESSION 35
8.2 SVM (SUPPORT VECTOR MACHINE) 36
8.3 MEAN SQUARED ERROR 37

9. CONCLUSION 38
9.1 PROJECT CONCLUSION 38
9.2 FUTURE SCOPE 38

10. REFERENCES 39

List of Figures:

Fig 3.3.1	Steps of Requirement Analysis	10
Fig 4.1.1	Design of Project	12
Fig 4.2.1	Steps in Project	13
Fig 4.2.1.1	Data Set	13
Fig 4.2.2.1	Kind of Data	14
Fig 5.1.2.1.1	Load Required Libraries	17
Fig 5.2.1.1	Loading dataset	18
Fig: 5.3.1.1	Data Description	20
Fig 5.4.1	Data types	20
Fig 5.5.1	Identifying and Handling Missing Values	21
Fig 5.6.1.1	Correlation	22
Fig 5.6.1.2	Correlation Graph	23
Fig 5.7.2.1	Graph	25
Fig 5.7.3.1	Support Vectors	28
Fig 7.2.1.1	Training Data	31
Fig 7.2.1.2	Confusion Matrix	32
Fig 7.2.1.3	Testing	32

1. INTRODUCTION

We are in the 21st century and food sector is very big part of our economy. The food we Consume provides nourishment and gives energy to our body which helps us to perform our day to day activities. A healthy and fresh diet is the most important way to keep ourselves fit. One of the biggest problems that it faces is food spoilage, i.e. food items, more specifically meat items or fruits and vegetable going stale. The bigger problem is these spoilt items going undetected and onto the hands of the consumer.

Today, in most of the hostel mess and government schools' kitchen everybody is getting affected by the food they consume. Milk, fruits like banana and other foods used in daily life, as all of them do not offer quality since their moisture harmful gases vary from time to time. To ensure food safety it should be monitored at every stage of the supply chain.

In all fruits and vegetables industries, the process of checking of quality of items is done manually, mostly by a person sitting across a conveyor belt as the items pass by. Hence, if an automated process is brought into place, it would not only increase the accuracy of spoilt food detection, but also reduce manual manpower required. To automate this process, we plan on using a collection of smart sensors with microcontroller like the Node Mcu. On detection of a spoilt or stale food item, a sound buzzer can be ringed to draw attention, moreover this data will be sent to the cloud, as an application of IoT. This enables appropriate authorities to view how often they get spoilt food items and create transparency.

1.1 MOTIVATION

Thirty students from Telangana Minority Residential School in Hyderabad's Asif Nagar were admitted to Niloufer Hospital after they fell sick eating lunch served at the mess on MONDAY, JULY 08, 2019. The students suffered from vomiting and severe abdominal pain. The students were within the age range of around 10 to 13. They ate the hostel food on Sunday afternoon and immediately presented with stomach pain and vomiting. They were delivered to the hospital on Monday morning and admitted. Each year, approximately 600 million people fall sick and 420,000 people die from the illness. Often, people consume spoiled foods because there's no significantly visible sign of food spoilage.

1.2 PROBLEM DEFINITION

In today's world, food spoilage is a crucial problem as consuming spoiled food is harmful for consumers. The food items kept at room temperature undergo rapid bacterial growth and chemical changes in food. The major problem is that these spoilt items going undetected and onto the hands of the consumer. Eating unhealthy food can cause several food borne diseases which may harm our health. To ensure food safety it should be monitored at every stage of the supply chain. In all fruits and vegetables industries, the process of checking of quality of items is done manually, mostly by a person sitting across a conveyor belt as the items pass by. Hence, if an automated process is brought into place, it would not only increase the accuracy of spoilt food detection, but also reduce manual manpower required.

1.3 OBJECTIVE

Our project aims at detecting spoiled food using appropriate sensors and monitoring gases released by the particular food item. We plan on implementing machine learning to this model so we can estimate how likely a food is going to get spoiled and in what duration, if brought from a particular vendor.

The detection of whether or not a food item is spoilt or not is made using the following two principles:

- **Oxygen Level Detection:** The underlying theory is that if food item, say fruits or meat, is inhabited by germs, the oxygen levels in the immediate surrounding is going to be lower than it normally is .The reason being that, the germs inside the food item are consuming the oxygen, and this change in the level is what we plan to detect.

- **Ammonia Gas:** Meat items like fish are known to release ammonia gases when they go stale. A gas
  Sensor captures the readings of the ammonia levels near the food item, and sends an alert to the Microcontroller when abnormal levels are detected.

1.4 SCOPE OF THE PROJECT

Our machine learning model is dependent upon the dataset we are using. As we gather more data, various extensions can be added as well like predicting how long a food item can last based on the food vendor it is brought from. This can greatly help food sellers manage the inventory and create a general awareness about food safety.

1.5 APPLICATIONS

Machine learning is the science of learning of data and making predictions. When applied to this project, it can enhance its productivity multiple times.

Following are the applications of machine learning in the given project:

- **Industrial Application:** After making an analysis of data of spoilt food items, it can successfully predict which part of the day, are most occurrences of spoilt food items occurring. It can also be analyzed which particular food items are getting spoilt more frequently. Accordingly, more people can be employed during the peak times and less during other times. This saves considerable amount of manpower.
- **Commercial Application:** When this concept is applied to retail stores and grocery shops, we will obtain a collective data of how frequently a food item gets spoilt if brought from a particular vendor. This pushes vendors to sell food items that have a longer shelf life, and creates consumer awareness to buy food with longer shelf life.
- **Consumer Application:** When incorporated into refrigerators, this smart sensing system can send email notifications directly user to whenever spoilt food is detected. This ensures consumers; especially children don't consume harmful and spoilt food.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

Paper 1: "EFresh – A Device to Detect Food Freshness"
September 2018

In this paper authors Naveed Shahzad, Usman Khalid used biosensor and electrical sensors to check out the freshness of food. A smart system that may sight the freshness of food like farm things, meat, and fruits. The identification and choice of hydrogen ion concentration device, moisture sensor, and the Gas sensor is used to develop a wise food freshness detector that ensures the freshness of food and tells whether or not to eat it or bin it. An android application is developed to select the type of food to be checked. The system ensures the quality of food, whether it is good for eating or not. It does not provide the facility to complain if the device does not provide accurate results. The feedback may recover the issues related to the device.

Paper 2: "Detection and classification of bacteria in common street foods using electronic nose and
Support vector machine"2017

Authors Jessie R. Balbin, Julius T. Sese, Crissa Vin R. Babaan focused on the classification of bacteria in street food. Street food features a major impact on the culture and however, as a result of the dearth of information on correct food preparation, the cleanliness and quality of street food are neglected. A bad microorganism that causes diarrheal diseases and it's exhausting to sight whether or not the microorganism exists, by using an electronic nose, and image processing.

This paper aims to design an electronic nose with gas sensors that will detect three common types of bacteria on street foods, namely Enterococcus faecalis, Escherichia coli and Staphylococcus aureus; and to classify if the said bacteria are present in the pre-cooking stage and the bacteria are still present after cooking. The electronic nose System detects the bacteria in the sample street food during the pre-cooking stage and Support Vector Machine detects the bacteria in the sample street food during the post cooking stage. This system lacks the detection of other parameters like moisture, gas level in food.

Paper 3: “Real-Time Milk Monitoring System”2018

Authors Prof. Kadam P. R, Miss. Shinde K. P. describes the scenario of smart city services that are provided to manage the city's assets by integrating information and communication technology (ICT) and the Internet of things (IoT). Different sensors, terminals with a variety of topologies and different application requires security for managing them. To make money day by day the quality of food decreases and it affects the health of people and this creates food safety problems. In this paper, the presented model detects the raw milk for spoilage detection. From the last decade, researches are coming up with different efficient methods for detecting spoilage of milk.

This paper states different studies that show that raw milk contains the bacteria which are harmful to human beings, so there is a need to develop one real-time system which will monitor the quality of milk distributed to the people or getting used for dairy products. The proposed system work with a set of different sensors which are connected to the Arduino board and in turn all data will get passed to the android app and according to the value, the system checks the quality of milk and user can easily identify the quality of milk, the user is getting. Along with milk, a system must check other items which will make the system more effective.

Paper 4: “The Vegetable Freshness Monitoring System Using RFID with Oxygen and Carbon Dioxide Sensor”2012

In this paper authors Ki Hwan Eom, Min Chul Kim proposed an oxygen and carbonic acid gas concentration observation system for freshness management, which supports radio frequency identification (RFID). Freshness may be checked by varying factors as well as wetness, temperature, oxygen, and carbonic acid gas. This paper focuses on oxygen and carbon dioxide. The concentrations of these two gases are related to freshness and affect the food.

This system uses a device for observation of gases and connects the device with the associate RFID tag. The RFID system is relatively easy to manage. With this combined system, it calculated the freshness of vegetables.

- **Artificial Intelligence Approach:**

An alternative approach is using artificial intelligence, on which some work is being done currently. The concept is basically using computer vision and analyzing the image of a food item to conclude if it is spoilt or not. However, the problem is, by capturing images we can only get hold of an idea of the exterior of the food item and not it's interior.

- **MIT Research:**

A MIT team has done research on this field by developing a sensor that detects spoilt meat items. But since it only detects a particular gas, it can have a lot of false negatives.

2.2 DISADVANTAGES OF EXISTING SYSTEM

- It only detects a particular gas, it can have a lot of false negatives.
- It does not provide the facility to complain if the device does not provide accurate results.
- By capturing images of food we can only get hold of an idea of the exterior of the food item and not it's interior.

2.3 PROPOSED SYSTEM

To maximize accuracy of detection, instead of relying on value of one sensor, we plan on using cumulative value of both oxygen and ammonia sensors to arrive at a decision. This reduces the possibilities of false negatives. Also, by incorporating an oxygen sensor, our smart sensor can be used to detect germ infection on practically any food item. Combing IoT and Machine learning only increases the inter-operability and application.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- The process of checking of quality of items is done manually, mostly by a person sitting across a conveyor belt as the items pass by. Hence, if an automated process is brought into place, it would not only increase the accuracy of spoilt food detection, but also reduce manual manpower required.

- Inside shelves and containers, our array of ammonia and oxygen sensors can be placed. As a spoilt food item is detected, management is alerted. Also, food items, having a shorter shelf life can be given a higher priority to sell first, greatly assisting the inventory management.

3. ANALYSIS

3.1 INTRODUCTION

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the fundamental specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is difficult, improper understanding of the present system can lead diversion from solution.

3.2 FEASIBILITY STUDY

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully.

3.2.1 Technical Feasibility:

A technical feasibility study can provide relevant context to the different aspects of your project and serve as a great planning tool by providing an overhead view of how your project can evolve during the course of its development, troubleshooting and tracking the progress of your project from concept to reality.

Technical resources/Tools required for our Project Development:

- Jupyter Notebook
- Python
- Python Libraries: Pandas, NumPy, Matplotlib, Scikit-learn.
- Dataset
- Machine Learning Model

3.2.2 Economic Feasibility:

- All the software and packages used in here are open source and hence free of cost. Thus under the open source license they are very economically available for usage and modification.

3.3 REQUIREMENT ANALYSIS

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements.

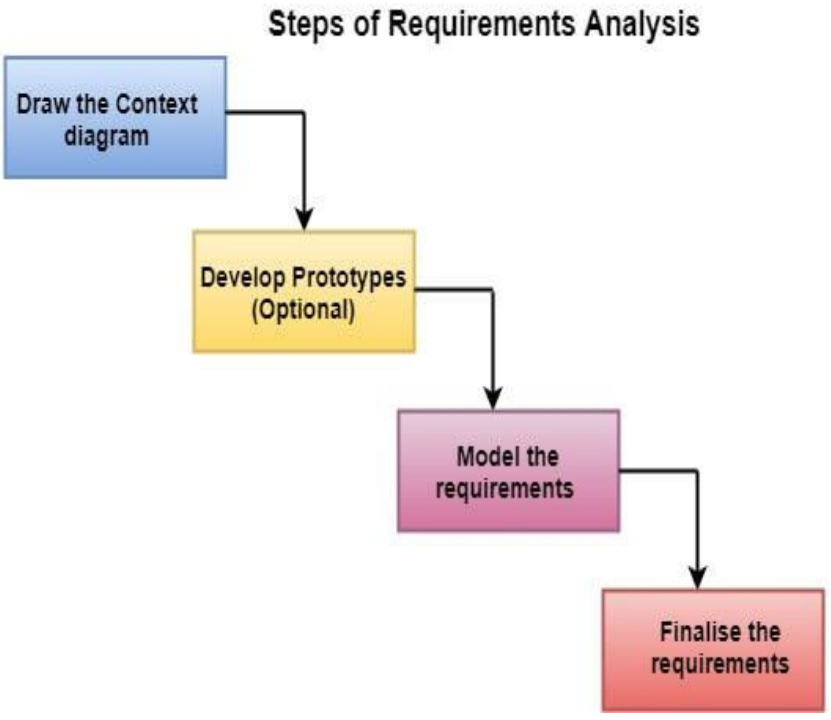


Fig 3.3.1 Steps of Requirement Analysis

3.4 SOFTWARE REQUIREMENT SPECIFICATION

3.4.1 User Requirements

Requirement Specification plays an important role to create quality software solution. Requirements are represented in a manner that ultimately leads to successful software implementation. The user needs to have a basic knowledge of installation of the application and running it.

3.4.2 Software Requirements

The software requirements specification is produced at the end of the analysis task. Software requirement is a difficult task, only decided after testing whether it fits the requirements.

- Operating System - Windows 10
- Tools and Technologies: Anaconda (Jupyter Notebook)
- Coding Language - Python

3.4.3 Hardware Requirements

This is an IoT project so hardware plays an important role. Selection of hardware also plays an important role in existence and performance of any software.

- Monitor
- Smart sensors- Node Mcu.
- Sound buzzer
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

4. DESIGN

4.1 INTRODUCTION

Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirements have been specified and analyzed, system design is the first of the three technical activities – design, code and test that is required to build and verify software.

During design, progressive refinement of data structure, program structure and procedural details are developed, reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design, procedural design.

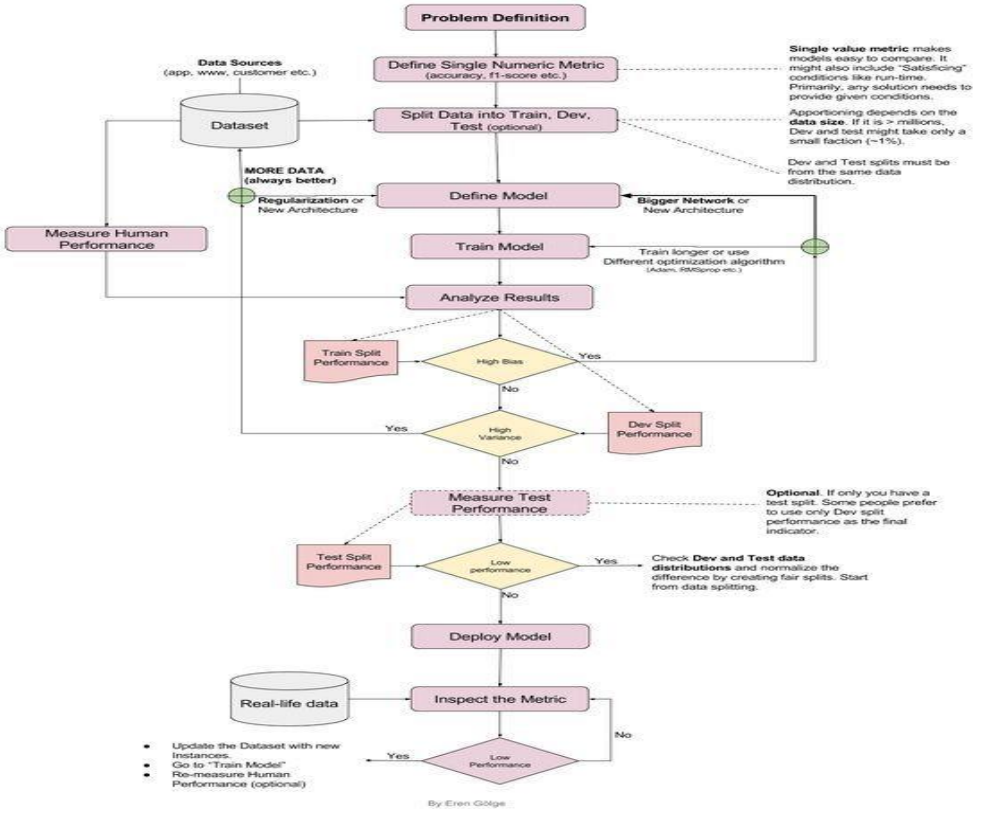


Fig 4.1.1 Design of Project

4.2 ARCHITECTURE OF PROJECT

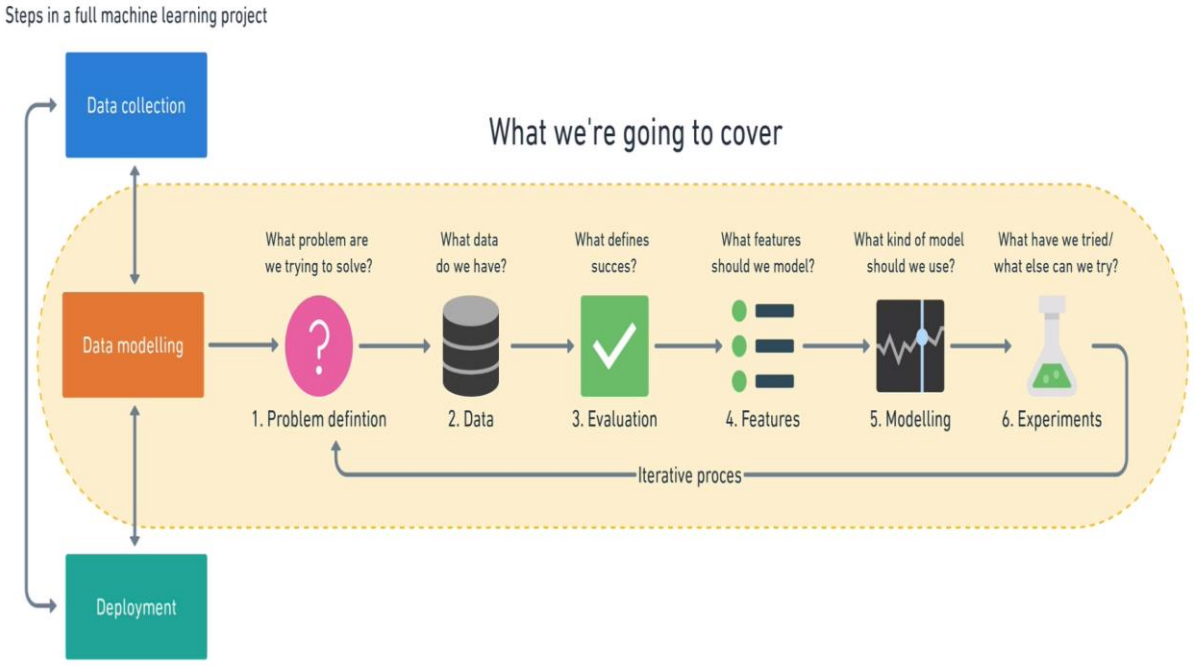


Fig 4.2.1 Steps in Project

4.2.1 Data Collection

We created a sample data set of some instances of foods that were spoilt and non-spoilt

Ammonia	Oxygen	Temperatu	Time_days	Condition_	Outcome
28	28	9	45	1	spoilt
2	3	13	14	1	spoilt
8	1	6	64	1	spoilt
25	18	22	6	1	spoilt
15	26	21	63	0	not spoilt
22	31	19	40	1	spoilt
39	29	4	60	1	spoilt
13	34	11	20	0	not spoilt
19	39	20	7	0	not spoilt
25	24	25	10	1	spoilt
22	12	9	6	1	spoilt
5	26	6	93	0	not spoilt
32	29	20	8	1	spoilt
14	20	5	99	1	spoilt

Fig 4.2.1.1 Data Set

We can run the data set in different machine learning models. The model which gives the best accurate result is considered as the best model to detect whether the food is spoilt or not.

4.2.2 Data Modeling

Data Modeling involves 6 steps:

- **Problem Definition**

  What business problem are we trying to solve? How can it be phrased as a machine learning problem?

  In our project we are trying to solve a **Supervised Learning Problem**

- **Data**

  If machine learning is getting insights out of data, what data we have? How does it match the problem definition? Is our data structured or unstructured? Static or streaming?

  In our project we are using Structured Data which is in the form CSV (comma separated value) file.

Fig 4.2.2.1 Kind of Data

- **Evaluation**

  What defines success? Is a 95% accurate machine learning model good enough?

  There are different evaluation metrics for classification, regression and recommendation problems. Which one we choose will depend on our goal

- **Features**

  What features does your data have and which can you use to build your model?

  The three main types of features are

  - Categorical features
  - Continuous (or numerical) features
  - Derived features

  Our data consists of Numerical features i.e. numbers.

- **Modeling**

  Which model should you choose? How can you improve it? How do you compare it with other models?

  Modeling breaks into three parts,

  - Choosing a model
  - Improving a model
  - Comparing it with others

- **Experimentation**

  What else could we try? How do the other steps change based on what we've found? Does our deployed model do as we expected?

  This step involves all the other steps. Because machine learning is a highly iterative process, you'll want to make sure your experiments are actionable.

  Our biggest goal should be minimizing the time between offline experiments and online experiments.

5. IMPLEMENTATION

5.1 DATA PREPARATION

Preprocessing of the data actually involves the following steps:

5.1.1: GETTING THE DATASET:

We can get the data set from the database or we can get the data from the client.

5.1.2: IMPORTING THE LIBRARIES:

We have to import the libraries as per the requirement of the algorithm.

5.1.2.1: Packages used:

Numpy: In Python we have lists that serve the purpose of arrays, but they are slow to process. Numpy aims to provide an array object that is up to 50 x faster than traditional Python lists. The array object in Numpy is called ndata; it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Pandas: Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Seaborn: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes. Visit the installation page to see how you can download the package.

Matplotlib: Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of Numpy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also. Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by Math Works. Matplotlib along with Numpy can be considered as the open source equivalent of MATLAB.

Scikit-learn (Sklearn): is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python.

preprocessing of the data :

```
1 #importing the libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 %matplotlib inline
6 import sklearn
7 import seaborn as sns
```

Fig 5.1.2.1.1 Load Required Libraries

5.2 DATA SET

It is a multivariate data set containing:

- Ammonia
- Oxygen
- Temperature
- Time_days
- Condition_ripe
- Outcome

5.2.1 IMPORTING THE DATA-SET

Pandas in python provide an interesting method read_csv (). The read_csv function reads the entire dataset from a comma separated values file and we can assign it to a Data Frame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the data frame. Any missing value or NaN value has to be cleaned.

```
1 l=pd.read_csv("freshnessdataset.csv")
2 l
3
```

	Ammonia	Oxygen	Temperature	Time_days	Condition_ripe	Outcome
0	28	28	9	45	1	spoilt
1	2	3	13	14	1	spoilt
2	8	1	6	64	1	spoilt
3	25	18	22	6	1	spoilt
4	15	26	21	63	0	not spoilt
...	...	...	...	...	...	...
95	5	35	15	20	0	not spoilt
96	7	26	19	40	0	not spoilt
97	37	9	8	33	1	spoilt
98	10	34	15	84	0	not spoilt
99	6	40	23	25	0	not spoilt

100 rows × 6 columns

Fig 5.2.1.1 Loading dataset

5.3 DATA PREPROCESSING/FEATURE ENGINEERING

5.3.1: Statistical Analysis:

Pandas in python provide an interesting method read_csv (). The read_csv function reads the entire dataset from a comma separated values file and we can assign it to a Data Frame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the data frame. Any missing value or NaN value has to be cleaned.

Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN values. Analyzes numeric and object series, as well as Data Frame column sets of mixed data types. The output will vary depending on what is provided.

For numeric data, the result's index will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default, the lower percentile is 25 and the upper percentile is

75. The 50 percentile is the same as the median.

For object data (e.g. strings or timestamps), the result's index will include count, unique, top, and freq. The top is the most common value. The freq is the most common value's frequency. Timestamps also include the first and last items.

If multiple object values have the highest count, then the count and top results will be arbitrarily chosen from among those with the highest count.

For mixed data types provided via a Data Frame, the default is to return only an analysis of numeric columns. If the data frame consists only of object and categorical data without any numeric columns, the default is to return an analysis of both the object and categorical columns. If include='all' is provided as an option, the result will include a union of attributes of each type.

```
1 l.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Ammonia	100.0	21.33	11.111060	1.0	12.5	23.0	30.25	40.0
Oxygen	100.0	23.53	10.978818	1.0	14.0	26.0	33.00	40.0
Temperature	100.0	14.26	6.874958	1.0	8.0	15.0	20.00	25.0
Time_days	100.0	51.39	29.027641	3.0	26.0	53.0	81.00	99.0
Condition_ripe	100.0	0.63	0.485237	0.0	0.0	1.0	1.00	1.0

Fig: 5.3.1.1 Data Description

5.4: DATA TYPE CONVERSIONS

When doing data analysis, it is important to make sure you are using the correct data types; otherwise you may get unexpected results or errors. In the case of pandas, it will correctly infer data types in many cases and you can move on with your analysis without any further thought on the topic. Despite how well pandas work, at some point in your data analysis processes, you will likely need to explicitly convert data from one type to another. This article will discuss the basic pandas data types (aka dtypes), how they map to python and Numpy data types and the options for converting from one pandas type to another.

```
1 l.dtypes
```

```
Ammonia          int64
Oxygen           int64
Temperature      int64
Time_days        int64
Condition_ripe   int64
Outcome         object
dtype: object
```

Fig 5.4.1 Data types

5.5: IDENTIFYING AND HANDLING MISSING VALUES

There are a number of schemes that have been developed to indicate the presence of missing data in a table or Data Frame. Generally, they revolve around one of two strategies: using a mask that globally indicates missing values, or choosing a sentinel value that indicates a missing entry. In the masking approach, the mask might be an entirely separate Boolean array, or it may involve appropriation of one bit in the data representation to locally indicate the null status of a value. In the sentinel approach, the sentinel value could be some data-specific convention, such as indicating a missing integer value with -9999 or some rare bit pattern, or it could be a more global convention, such as indicating a missing floating-point value with NaN (Not a Number), a special value which is part of the IEEE floating-point specification.

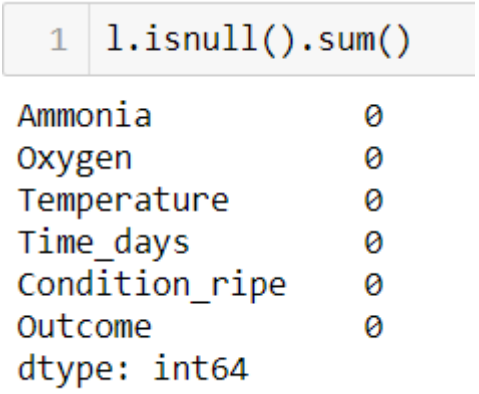


Fig 5.5.1: Identifying and Handling Missing Values

5.6: GENERATING PLOTS

5.6.1: Visualize the data between Target and the Features

- **Correlation:**

```
1 l.corr()
```

	Ammonia	Oxygen	Temperature	Time_days	Condition_ripe
Ammonia	1.000000	-0.278015	0.065643	-0.082175	0.706706
Oxygen	-0.278015	1.000000	-0.052296	0.024416	-0.544913
Temperature	0.065643	-0.052296	1.000000	-0.025315	0.038212
Time_days	-0.082175	0.024416	-0.025315	1.000000	-0.016903
Condition_ripe	0.706706	-0.544913	0.038212	-0.016903	1.000000

Fig 5.6.1.1 Correlation

- **Correlation graph:**

```
import seaborn as sns
fig = plt.subplots (figsize = (11, 11))
sns.heatmap(l.corr (), square = True, cbar = True, annot = True, cmap="GnBu", annot_kws = {'size': 8})
plt.title('Correlations between Attributes')
plt.show ()
```

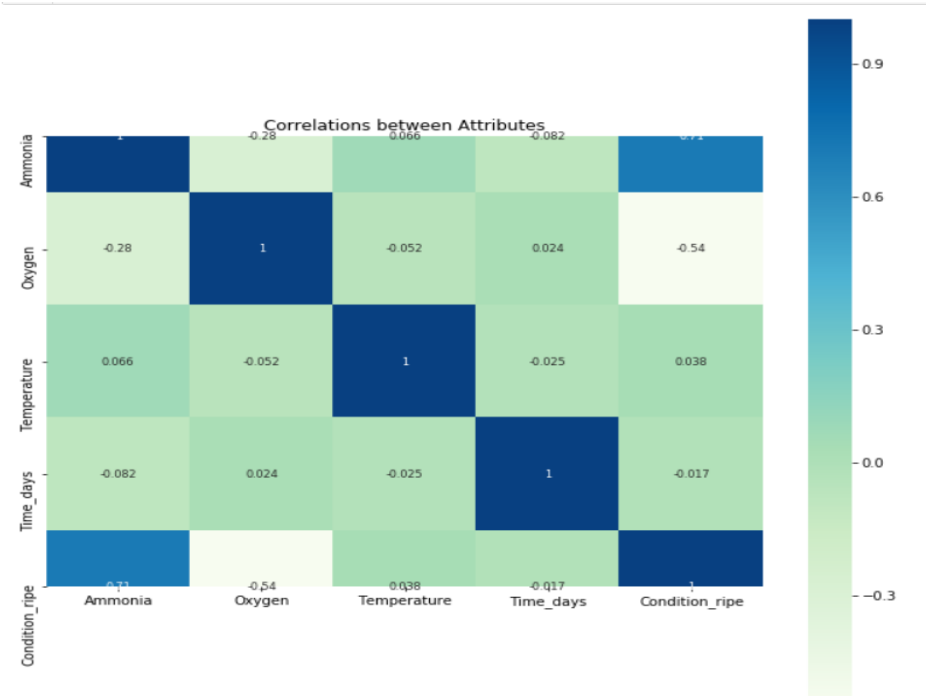


Fig 5.6.1.2 Correlation Graph

5.7 ALGORITHMS USED

5.7.1 Logistic Regression

Logistic Regression is used when the dependent variable (target) is categorical.

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis and it predicts the probability

Example: Yes or No, get a disease or not, pass or fail, defective or non-defective, etc.,

Also called a classification algorithm, because we are classifying the data. It predicts the probability associated with each dependent variable category.

Probability:

The probability in a logistic regression curve

$$p = \frac{e^y}{1 + e^y}$$

Where,
e is a real number constant, the base of natural logarithm and equals 2.7183
y is the response value for an observation

Methods:

There are three methods of Logistic Regression based on nature of the attribute data.

- Binary
- Nominal
- Ordinal

✓ Binary Logistic Regression

- ✓ Binary logistic Regression is performed on the Binary response variables. It has only two categories, such as presence or absence of disease, pass or fail, defective or non-defective products.

✓ Nominal Logistic Regression

- ✓ Nominal Logistic Regression is performed on the Nominal variables. These are categorical variables that have three or more possible categories with no natural ordering

Example: Food is crunchy, mushy and crispy

5.7.2 Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

Linear Regression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

1. Simple Linear Regression: With simple linear regression when we have a single input, we can use statistics to estimate the coefficients**.**

2. Ordinary Least Squares: When we have more than one input we can use Ordinary Least Squares to estimate the values of the coefficients.

3. Gradient Descent: When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.

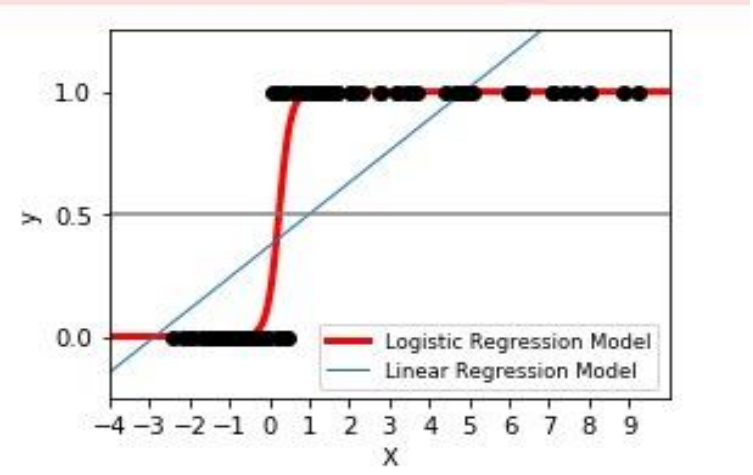


Fig 5.7.2.1 Graph

5.7.3 SVM (Support Vector Machine)

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyper planes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

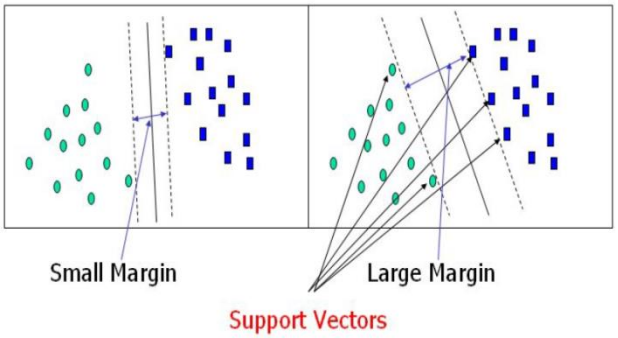


Fig 5.7.3.1 Support Vectors

Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper plane. These are the points that help us build our SVM.

6. MODELLING

6.1 INTRODUCTION

Modelling—which model should you choose? How can you improve it? How do you compare it with other models?

Once you've defined your problem, prepared your data, evaluation criteria and features it's time to model.

Modelling breaks into three parts, choosing a model, improving a model, comparing it with others.

1. Choosing a model

When choosing a model, you'll want to take into consideration, interpretability and ease to debug, amount of data, training and prediction limitations.

- **Interpretability and ease to debug**—why did a model make a decision it made? How can the errors be fixed?
- **Amount of data**—How much data do you have? Will this change?
- **Training and prediction limitations**—this ties in with the above, how much time and resources do you have for training and prediction?

Linear models such as logistic regression are usually easier to interpret, are very fast for training and predict faster than deeper models such as neural networks.

2. Tuning and improving a model

A model's first result isn't its last. Like tuning a car, machine learning models can be tuned to improve performance. Tuning a model involves changing hyper parameters such as learning rate or optimizer. Or model-specific architecture factors such as number of trees for random forests and number of and type of layers for neural networks.

The priority for tuning and improving models should be reproducibility and efficiency. Someone should be able to reproduce the steps you've taken to improve performance

3. Comparing models

Model 1, trained on data X, evaluated on data Y.
Model 2, trained on data X, evaluated on data Y.
Where model 1 and 2 can vary but not data X or data Y.

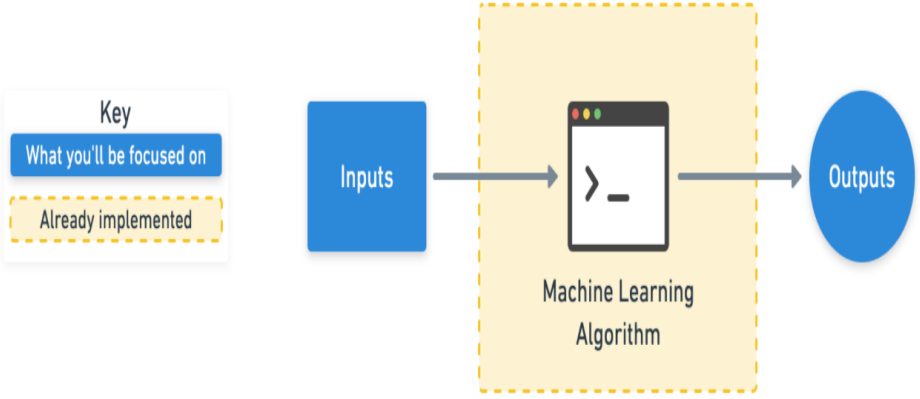


Fig 6.1.1 Modelling

To begin with, your main job will be making sure your inputs (data) lines up with how an existing machine learning algorithm expects them. Your next goal will be making sure the outputs are aligned with your problem definition and if they meet your evaluation metric.

In our project we took

- **Input:** Source of Food item
- **Output:** Food item is spoilt or not
- **Training Data:** Instances of spoilt/unspoilt food item, oxygen and ammonia concentrations for each sample.

6.2 Logistic Regression Model

```
1 from sklearn.linear_model import LogisticRegression
```

```
1 lr1 = LogisticRegression(multi_class = 'ovr', solver = 'sag',  max_iter = 5000)
```

```
1 lr1.fit(X_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=5000,
                   multi_class='ovr', n_jobs=None, penalty='l2',
                   random_state=None, solver='sag', tol=0.0001, verbose=0,
                   warm_start=False)
```

7. TESTING AND VALIDATION

7.1 INTRODUCTION

The goal of testing is to acquire errors. Testing is that the technique of trying to get each possible error or weakness in an extremely work product. Testing is done by carrying out different experiments on the tool, with different approaches in the scope. These experiments are generally cause-effect by demonstrating what-happens-when-what-is-done.

7.2 EXPERIMENTATION

What else could we try? How do the other steps change based on what we've found? Does our deployed model do as we expected?

This step involves all the other steps. Because machine learning is a highly iterative process, we need to make sure our experiments are actionable.

Our biggest goal should be minimizing the time between offline experiments and online experiments.

All experiments are conducted on different portions of our data.

- **Training data set**—we used this set for model training, 70–80% of our data is the standard.
- **Validation/development data set**—we used this set for model tuning, 10–15% of our data is the standard.
- **Test data set**—we used this set for model testing and comparison, 10–15% of your data is the standard.

7.2.1Training and Testing

1. Logistic Regression

Training Phase

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.3, random_state =101)
```

```
X_train
```

	Ammonia	Oxygen	Temperature	Time_days	Condition_ripe
86	7	35	8	53	0
14	39	35	6	21	1
23	23	25	19	81	1
26	16	22	21	54	0
22	25	17	11	89	1
...	...	...	...	...	...
63	29	16	4	54	1
70	7	7	19	15	1
81	25	10	20	94	1
11	5	26	6	93	0
95	5	35	15	20	0

70 rows × 5 columns

Fig 7.2.1.1 Training Data

```
y_train
```

```
86    0
14    1
23    1
26    0
22    1
     ..
63    1
70    1
81    1
11    0
95    0
Name: Condition_ripe, Length: 70, dtype: int64
```

Confusion Matrix

```
confusion_matrix = metrics.confusion_matrix(y_train, y_train_pred_lr) #train
confusion_matrix
```

```
array([[29,  1],
       [ 1, 39]], dtype=int64)
```

```
sns.heatmap(confusion_matrix,annot=True,annot_kws={"size": 16}, fmt='g') #train
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b7e8854708>
```

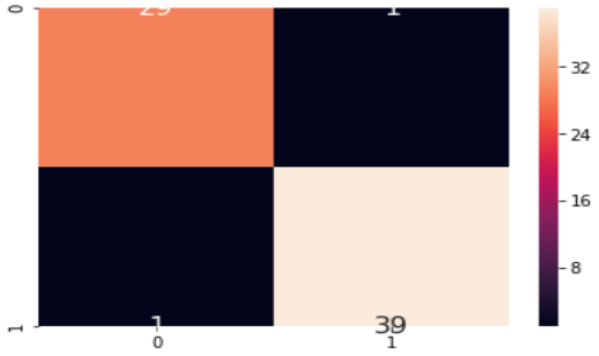


Fig 7.2.1.2 Confusion Matrix

Testing Phase

```
y_test_pred_lr = lr1.predict(X_test)
confusion_matrix = metrics.confusion_matrix(y_test, y_test_pred_lr)
confusion_matrix
```

```
array([[ 5,  2],
       [ 0, 23]], dtype=int64)
```

```
sns.heatmap(confusion_matrix,annot=True,annot_kws={"size": 16}, fmt='g') #test
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b7e88ec648>
```

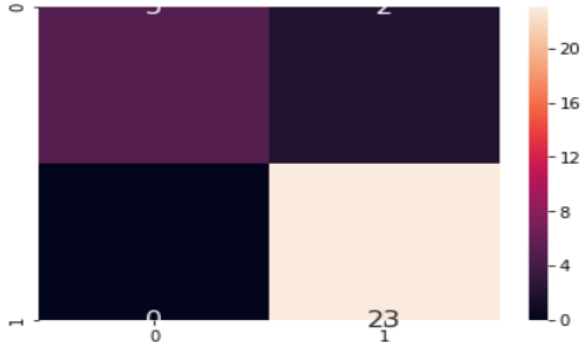


Fig 7.2.1.3 Testing

2. SVM (Support Vector Machine)

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=4)
```

```
X_train.shape
```

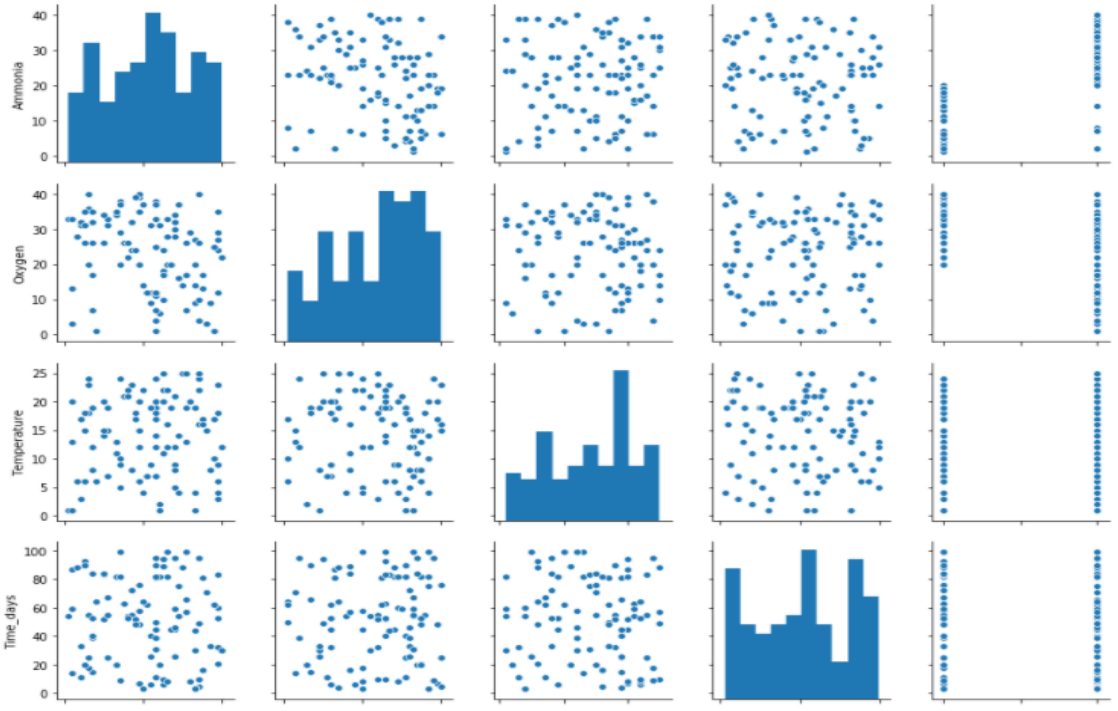
```
(80, 4)
```

```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(80, 4)
(20, 4)
(80,)
(20,)
```

```
sns.pairplot(l)
```

```
<seaborn.axisgrid.PairGrid at 0x1b7e85b9408>
```



8. RESULTS

Accuracy: Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset.

Precision: Precision is a metric that quantifies the number of correct positive predictions made. Precision, therefore, calculates the accuracy for the minority class. It is calculated as the ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted.

Recall: Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions. In this way, recall provides some notion of the coverage of the positive class. Recall is calculated as the number of true positives divided by the total number of true positives and false negatives.

F-Measure: F-Measure provides a way to combine both precision and recall into a single measure that captures both properties. Once precision and recall have been calculated for a binary or multiclass classification problem, the two scores can be combined into the calculation of the F-Measure.

8.1 LOGISTIC REGRESSION

```
In [208]:
1
2 a=classification_report(y_train, y_train_pred_lr)
3 print(a)
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	30
1	0.97	0.97	0.97	40
accuracy			0.97	70
macro avg	0.97	0.97	0.97	70
weighted avg	0.97	0.97	0.97	70

Fig 8.1.1 Training Score

```
In [211]:
1 k=classification_report(y_test, y_test_pred_lr)
2 print(k)
```

	precision	recall	f1-score	support
0	1.00	0.71	0.83	7
1	0.92	1.00	0.96	23
accuracy			0.93	30
macro avg	0.96	0.86	0.90	30
weighted avg	0.94	0.93	0.93	30

Fig 8.1.2Testing Score

8.2 SVM (SUPPORT VECTOR MACHINE)

```
In [239]: 1 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
          2 print(classification_report(y_train,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.85      0.94      0.89        31
           1       0.96      0.90      0.93        49

    accuracy                           0.91        80
   macro avg       0.90      0.92      0.91        80
weighted avg       0.92      0.91      0.91        80
```

Fig 8.2.1 Training Score

```
1 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
2 print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.80      0.67      0.73         6
           1       0.87      0.93      0.90        14

    accuracy                           0.85        20
   macro avg       0.83      0.80      0.81        20
weighted avg       0.85      0.85      0.85        20
```

Fig 8.2.2 Testing Score

8.3 MEAN SQUARED ERROR

The mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

We ran several machine learning models on our data set with the following results:

Logistic Regression

```
import math
from sklearn.metrics import mean_squared_error
RMSE=math.sqrt(mean_squared_error(y_true=y_test,y_pred=y_prediction))
```

```
print(RMSE)
```

```
0.2581988897471611
```

Fig 8.3.1 MSE for Logistic Regression

Linear Regression

```
import math
from sklearn.metrics import mean_squared_error,r2_score

RMSE=math.sqrt(mean_squared_error(y_true=y_test,y_pred=y_prediction))
```

```
print(RMSE)
```

```
7.704761855311984e-16
```

Fig 8.3.2 MSE for Linear Regression

Since Logistic Regression performs best in the given dataset, we chose that model. We take input of ammonia and oxygen gas values from the sensor and the machine learning model tells us if the food is spoilt or not.

9. CONCLUSION

9.1 PROJECT CONCLUSION

An exhaustive research has led us to conclude that the food industry can be revolutionized by a simple combination of sensors, IoT and machine learning. Food poisoning has been the source of innumerable diseases, to reduce and avoid illness; we use biosensors and electrical sensors that determine the freshness of household food items like dairy items, fruits, and foods. After integration, this model will create a competition between food manufacturers to sell more healthy food and create awareness among consumer to purchase more healthy food.

9.2 FUTURE SCOPE

Our machine learning model is dependent upon the dataset we are using. As we gather more data, various extensions can be added as well like predicting how long a food item can last based on the food vendor it is brought from. This can greatly help food sellers manage the inventory and create a general awareness about food safety.

10. REFERENCES

Fatima Mustafa and Silvana Andreescu, “Chemical and Biological Sensors for Food-Quality Monitoring and Smart,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, 16 October 2018.

1. Neethirajan, S.; Jayas, D.S. Nanotechnology for the food and bioprocessing industries. Food Bioprocess Technol. 2011, 4, 39–47. I.S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

Eissa, S.; Zourob, M. ‘In vitro selection of DNA aptamers targeting - lacto globulin and their integration in graphene-based biosensor for the detection of milk allergen’ Biosens. Bioelectron. 2017, 91, 169–174.

Archer DL, ‘freezing: An underutilized food safety technology?’ International Journal of Food Microbiology, 2004

Gunders D, “Wasted: How America is Losing 40% Percent of its Food from Farm to Fork to Lanfill”, Naturals Resources Defense Council, 2012

eFresh – A Device to Detect Food Freshness

A Review Paper on Electronic Sensor-Based System to identify undesired ingredients present in Liquid Smita A. Nagtode1, Dr. N.K. Choudhari2 1Assistant Professor, Datta Meghe Institute of Engg. Technology & Research, Wardha 2 Principal, Smt, Bhagwati Chaturvedi College of Engineering, Nagpur 2013.