Discrimination and classification of fresh fruit & spoiled food using principal component analysis and support vector machine

Submitted in the partial fulfillment of the degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATIONS

Under

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

BY

DIGANTA SENGUPTA

University Roll no: 12019002002010

University Registration no: 204201900200165

UNDER THE GUIDANCE OF

PROF. UTTAM THAKUR

Electronics and Communications

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Discrimination and classification of fresh fruit & spoiled food using principal component analysis and support vector machine

UNIVERSITY OF ENGINEERING
&
MANAGEMENT, JAIPUR

Discrimination and classification of fresh fruit & spoiled food using principal component analysis and support vector machine

Submitted in the partial fulfillment of the degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATIONS

Under

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

BY

DIGANTA SENGUPTA

University Roll no: 12019002002010

University Registration no: 204201900200165

UNDER THE GUIDANCE OF

PROF. UTTAM THAKUR

Electronics and Communications

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Approval Certificate

This is to certify that the project report entitled "**Discrimination and classification of fresh fruit & spoiled food using principal component analysis and support vector machine**" submitted by (Roll:**12019002002010**) in partial fulfillment of the requirements of the degree of **Bachelor of Technology** in **Electronics and Communications** from University **of Engineering and Management, Jaipur** was carried out in a systematic and procedural manner to the best of our knowledge. It is a bona fide work of the candidate and was carried out under our supervision and guidance during the academic session of 2019-2023.

Prof. Uttam Thakur
Project Guide, Assistant Professor (ECE)
UEM, JAIPUR

Prof. (Dr.) Prashant Ranjan
HoD, Dept. of ECE, UEM Jaipur

Prof. (Dr.) A Mukherjee
Dean Academics, UEM, JAIPUR

ACKNOWLEDGEMENT

The endless thanks go to Lord Almighty for all the blessings he has showered onto me, which has enabled me to write this last note in my research work. During the period of my research, as in the rest of my life, I have been blessed by Almighty with some extraordinary people who have spun a web of support around me. Words can never be enough in expressing how grateful I am to those incredible people in my life who made this thesis possible. I would like an attempt to thank them for making my time during my research in the Institute a period I will treasure. I am deeply indebted to my research supervisor, Professor Uttam Thakur for such an interesting thesis topic. Each meeting with him added in valuable aspects to the implementation and broadened my perspective. He has guided me with his invaluable suggestions, lightened up the way in my darkest times and encouraged me a lot in the academic life.

Diganta Sengupta

ABSTRACT

Food wastage due to spoilage is one of the key global problems, the quantity of food loses and spoilage per year is 40-50% for root crops, fruits, and vegetables. The objective is to develop an affordable IoT & machine learning based approach for food quality monitoring. The need for this type of system is to correctly classify the vegetables and fruits from experimental data, in two categories fresh & spoiled. The proposed system is employed with Arduino Uno as the main processing unit, two Volatile Organic Compound gas sensors. To classify the vegetable and fruit, the sensed data (MOS Gas Sensors MQ3 & MQ7) were collected over a week & machine learning models —Principal Component Analysis (PCA), Linear Regression & Linear SVM Classifier, have been deployed with the resulting accuracy of 99.1%, 99.9%, 99.96% & 99.96% respectively have been used to classify the data. The uniqueness of the system is, it uses an affordable IoT system which can deploy at any possible environment to preserve any type of food items by assessing the spoilage levels from the real-time novel data collected.

Table of Contents

List of Figures

1

1. CHAPTER

LITERATURE REVIEW

A Raspberry Pi-based development of a machine learning and edge IoT-based non-destructive food quality monitoring system is proposed in the cited study [1]. This system is cost-effective, adaptable, and user-friendly thanks to mobile-based monitoring. Present day state of the art advances like edge IoT, AI, and Easy to understand GUI [6] utilizing python has been utilized all through the examination. Foods grown from the ground (tomato and banana) were held under observing chamber for seven days; Sensor data have been recorded for three labels: spoiled, fresh, and semi-fresh. Further, four different AI calculations [7] have been created over information to characterize the decay phase of the item.

The Raspberry Pi screen-based Python-based graphical user interface [9] that enables real-time sensor data monitoring and control and aids in providing alert messages has been developed. The framework[11] performs two assignments: discovery of deterioration phase of the product and cautioning the client about the natural circumstances for specific organic product/vegetables. Based on available information about favorable environmental conditions, a software expert system has been developed to accurately inform the user of current conditions [10]. For information storage, Raspberry Pi uploads processed data to a firebase cloud via Wi-Fi.

INTRODUCTION

1.1 Volatility of the Edibles

Aroma (odor) plays a crucial role in the creation of novel food products [1]. Thus, the primary objective of many studies is to characterize aroma. [7]. It is necessary to understand the mechanism of VOCs release from decomposition of fruit caused via factors such as weather, temperature, moisture, carbohydrate accumulation, and fungal infections; as well as the sources and functions of VOCs, which would contribute to the early detection and monitoring of fruit diseases after harvest and to reduce harm to consumers. The fragrant properties of banana are significant for its engaging quality as a new natural product. The flavor of bananas is pleasant, and they are widely consumed all over the world. The banana's typical flavor compounds are made in a short amount of time while it is ripening, and the aroma's composition changes at different stages, creating distinctive aroma signatures [11]. There have been a few studies on the banana aroma compositions during ripening [11,12], but most of them focused on the chemical aroma compositions during the mature period or during fruit processing. It has also been reported [13] that the composition of aromas shifts at various stages of fruit ripening, which aids in comprehending the formation of aromatic compounds. The distinct flavors of various banana cultivars are due to differences in the predominant volatile components. For instance, the Cavendish cultivar's predominant ester is the volatile component 3-methylbutyl butanoate [14]. Esters typically impart a fruity aroma [7]; For instance, isoamyl butanoate, 3-methylbutyl acetate, and isoamyl isovalerate all play a role in the fruity smell of the banana [15,16]. Although 246 volatile compounds, including 112 esters, 57 alcohols, 39 acids, 10 aldehydes, and 10 ketones, were found in banana fruit, only 12 compounds were found to be significant contributors to the aroma of bananas [16]. Fruit is an essential part of a healthy diet because it is high in vitamins, minerals, fiber, and other nutrients. Individuals smell natural product prior to choosing whether to purchase since organic product fragrance is a sign of natural product quality and development. Even though different fruits have different aroma profiles, the volatile organic compounds (VOCs) of newly harvested and mature fruit are mostly made up of esters, alcohols, aldehydes, ketones, lactones, and terpenoids.

1.2 Artificial Olfaction

The human nose has been utilized to assess scent and nourishment for quite a while [1]. The best way to determine whether food is fresh is to smell it. Even the best inspector's nose can be off after smelling for a few hours & other medical conditions might render this manual smelling ineffective. In circumstances like these, electronic noses are helpful. An electronic nose recognizes and automatically detects gases, vapors, and odors. These are not confined by human elements, for instance, shortcoming, receptiveness to toxic substances and frailty to recognize come compounds [1]. According to a definition, sensory analysis with trained panelists precedes the use of the electronic nose to prepare the sensory characteristics of food products. Along these lines, one might say that e-nose is an instrumental method for uncovering some smell segregates of food varieties. An electronic nose is an instrument planned to impersonate the human feeling of smell. electronic noses are being utilized for examinations of the examples' sweet-smelling profiles without earlier partition of the unstable portion into individual parts. An array of non- or partially selective gas sensors and a data processing and pattern recognition system that can identify even complex aromatic profiles make up electronic noses (e-noses). Pattern recognition gives the system more selectivity and reversibility, which opens up a wide range of possibilities. These reaches from the food and clinical businesses to ecological observing and process control. The sensor system responds differently to different aromas, and these responses produce a signal pattern that is unique to that aroma [35]. The PC framework perceives the example of signs and afterward can look at the personality of the fragrance of differed food sources' concentrates by design acknowledgment framework, e.g., fake brain organization. This counterfeit olfaction could be useful in the portrayal of many examples in a moderately brief time frame [13]. For instance, a company that uses coffees from all over the world can use e-nose to analyze coffee samples. Under the best conditions, the analysis of many coffee samples can produce a group of points for each sample, which can also be grouped. The differences between the groups ought to increase as the number of samples grows. By comparing its proximity to one of the known samples, mapping the sensor responses of an unknown sample on the same scheme might make it possible to identify it.

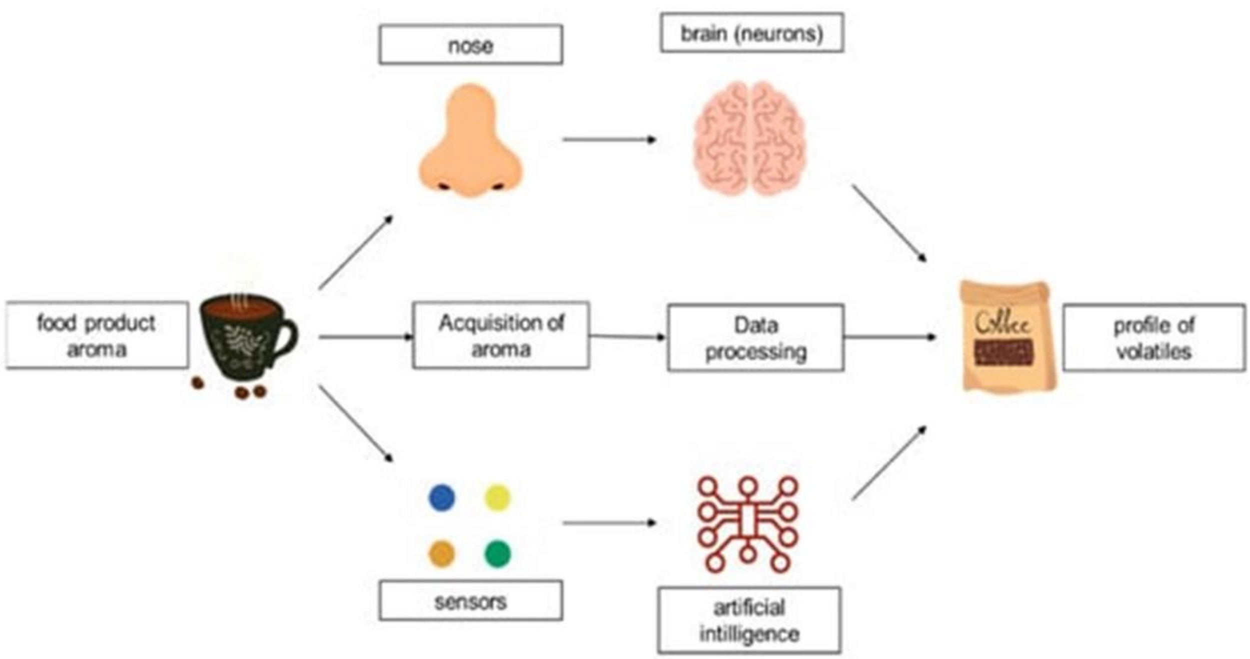

Figure 1The Analogy between Biological Olfactory System & Electronic Nose Technology

1.3 Sensors and Chemicals

The olfactory framework can neither recognize nor distinguish fragrances without the acquiring of synthetics delivered by the items [23]. These synthetics can be tracked down in basic or complex designs. However, each chemical has its own distinct qualities and characteristics. As a result, chemical digital signatures are exclusive, and they will serve as the input data for both the EN instrument and the olfactory system. The programmed recognition process can't be accomplished without a gathered library (dataset) of the computerized marks of explicit smells. Targeted chemicals in a medium are detected by the sensor array. Each targeted aroma is detected by a distinct sensor, or each sensor is responsible for sensing a distinct aroma type. The medium's chemicals are detected by chemical sensors. These sensors fundamentally convert synthetic data into logical signs [24]. Combining a number of distinct sensors in the array is the only way to achieve the primary goal of an EN, which is to detect multiple chemicals with greater precision. While remembering that the sensors in the exhibit must be selected cautiously by considering the synthetics of interest [25], having a legitimate sensor cluster for explicit undertakings relies upon a few circumstances as follows [26]:

- low cost and reusability
- short response and regeneration times
- high resistance to a variety of media
- insensitivity to temperature and humidity
- high selectivity and sensitivity in relation to chemical compounds or substances that are included in gas mixtures
- high security for the chose application

1.4 Sensor types utilized in the EN technology

- **Metal Oxide Gas Sensor**

  MOS is the most commonly used sensor type in the EN instrument because of its suitability for wide range of gases [27]. The practical application areas of MOS based ENs are mainly related to quality control, monitoring, process, aging, geographical origin, adulteration, contamination and spoilage of food and beverages [28]. These sensors can operate at high temperatures but require high power consumption. On the other hand, oxide surface reactions are too slow at lower temperatures. The oxidative chemical reactions are constrained due to the low vapor pressure of water molecules at temperatures below 100 °C [29]. MOS are categorized into two main groups according to their responses to different gases [19]: 1) n-type and 2) p-type. The operational principle of n-type sensors is based on the reactions between the oxygen molecules in the air and the surface of these sensors. As a result of these reactions, free electrons on the surface are trapped resulting in potential barriers between grains which inhibit the carrier mobility that produces large resistance areas. The p-type sensors respond to oxidizing gases, remove electrons and produce holes. Their characteristic surface reactivity and oxygen absorption significantly increase the performance of the sensor, while enhancing the recovery speed, boosting the gas selectivity and reducing the humidity dependence of signals. These sensors are frequently chosen in many EN applications because of their high sensitivity and selectivity.

Figure 2 MQ family MOS Gas Sensor

- **Conducting polymer sensors**

Although there are some drawbacks of Conducting Polymer sensors such as the inability of detecting gases like trimethylamine in fish odor applications, conducting polymers are a reliable sensor type used in many EN instruments for medical, pharmaceutical, food and beverage industries, because of their low cost, fast response to odorants and resistance to sensor poisoning [19]. Since the active layers are crucial parts of sensors, there are many different manufacturing processes of conducting polymer films including vapor deposition polymerization, thermal evaporation, electrochemical deposition, layer-by-layer self-assembly technique, well-known Langmuir-Blodgett (LB) technique and more [31]. Several conducting polymers are used in sensors such as polypyrene, polyaniline and polythiophene [31]. When a change occurs in the material due to an interaction with an analyte, the resistance in the sensor changes which leads to the detection of gases.

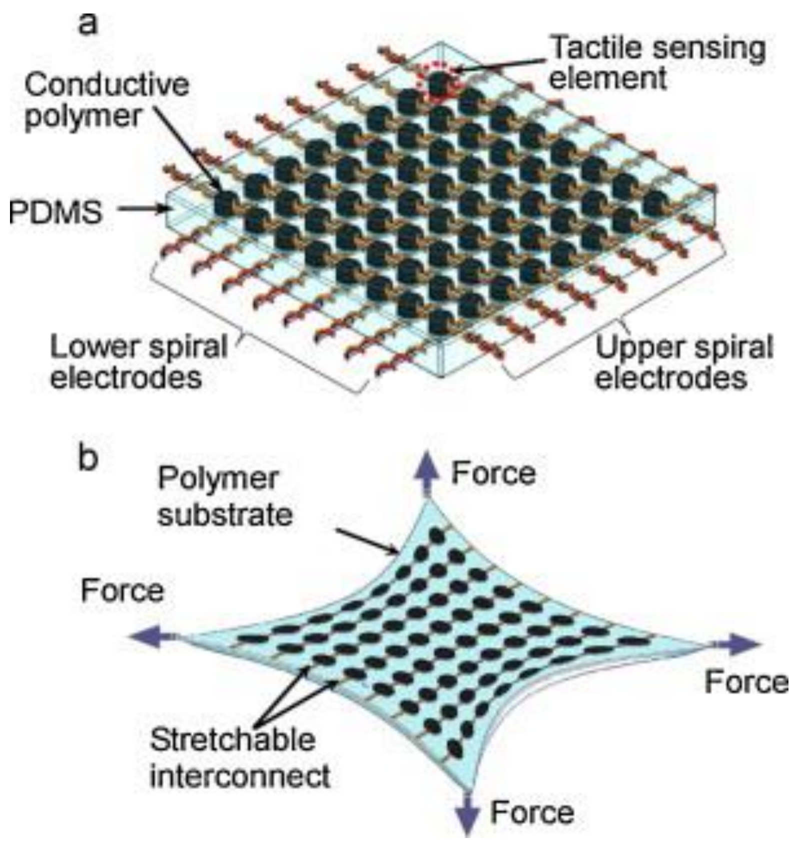

Figure 3 Conducting Polymer Sensor

- **Quartz Crystal Microbalance (QCM) sensors**

  QCM sensors are chosen as an EN component in several applications including medicine, environment monitoring, security and food safety because of their sensitivity, convenience, rapidness, stability and portability. In these sensors, the surfaces are covered with a sensitive coating [19]. A selective barrier on the crystal surface takes in the released gas from the environment, which then increases the total mass. Subsequently, frequency decreases because of the mass change on the gold surface of the QCM. Therefore, QCM sensors determine small variations on the sensor surface by measuring the frequency changes on the quartz crystal resonator. QCM sensors can operate in gas and liquid environments to determine the properties of chemicals [33]. Along with sensitive biosensors, they can detect even a nanogram of substance [34]. Importantly, QCM sensors are open for changes and improvements. The surface of these sensors can be modified so that they can detect entire cells or only one single molecular monolayer. Sensor enhancements can be actualized by an electrochemical approach, via immobilization of the silver electrode surface polyaniline film [35]. Furthermore, the biotoxicity of metallic electrodes caused by some bioactive coating films can be eliminated through employing electrodeless crystal configurations [29]. Additionally, molecularly imprinted polymers (MIPs) [28], polished gold films [33], biomimetic peptide-based sensing materials [39], multi-wall carbon nanotubes [34], acidized-multiwalled carbon nanotubes [32] and calixarenes [21] can be used in QCM sensors to coat surface layers. QCM based biosensors are commonly adopted to analyze odors because of their economic and easy manufacturing, and rapid analysis ability.

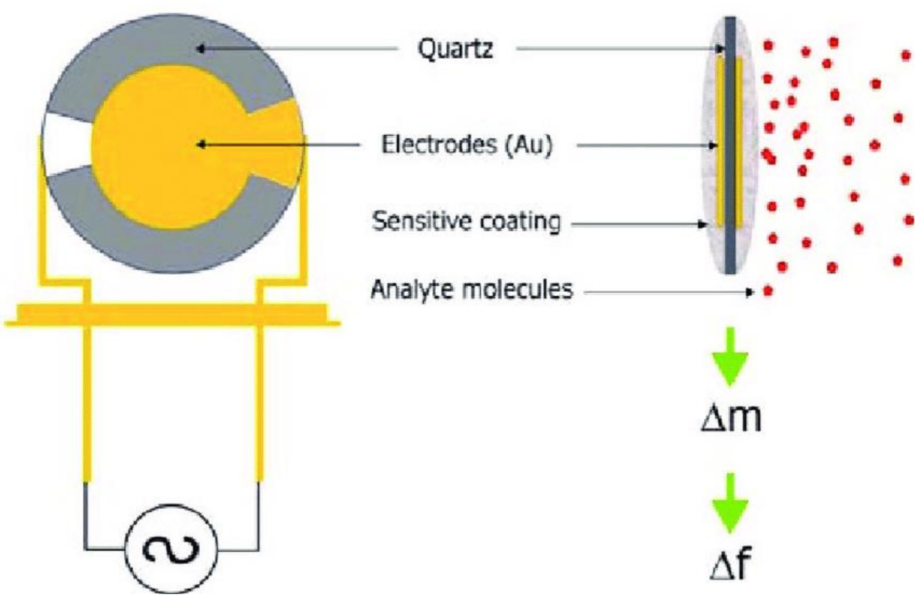

Figure 4 Quartz Crystal Microbalance Gas Sensor Working Principle

- **Acoustic Wave Sensors**

There are several acoustic wave sensor types, such as flexural plate wave device, fiber acoustic wave sensor, tube acoustic wave device, transverse wave device, bulk acoustic wave (BAW) and surface acoustic wave (SAW) sensors. BAW and SAW are frequently utilized in EN applications because of their small size, sensitivity, low cost and response to nearly all gases [45]. BAW sensors operate in the same way as QCM sensors, but they are less sensitive. They can measure extremely small frequency changes by means of their stability. Low cost, simplicity and robustness are other advantages of BAW sensors. However, the performance of these sensors in a liquid medium is inadequate. SAW sensors are derived from BAW devices. High sensitivity is achieved by operating at high frequencies. However, the signal-to-noise performance is poor in SAW sensors because of the high operating frequencies.

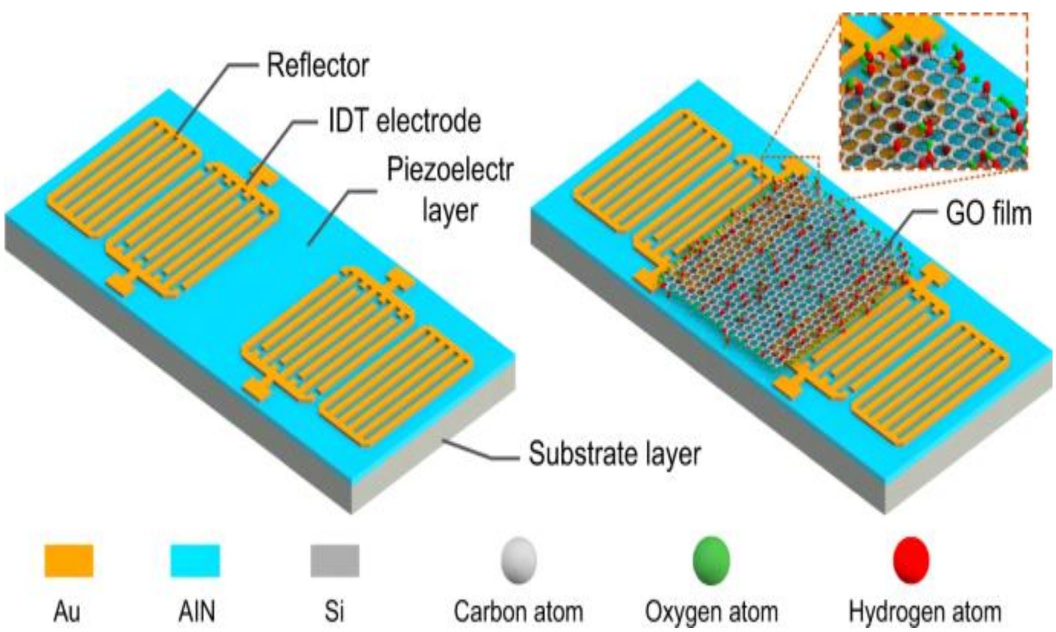

Figure 5 Acoustic Wave Sensor

- **Electrochemical (EC) Gas Sensor**

  Electrochemical (EC) gas sensors are devices that use chemical reactions to detect and measure the concentration of various gases in the environment. These sensors work by converting the gas molecules into an electric current, which can then be measured and analyzed. EC gas sensors typically consist of two electrodes separated by an electrolyte. One electrode is coated with a material that reacts with the gas being detected, while the other electrode remains inert. When the gas comes into contact with the reactive electrode, a chemical reaction occurs, which generates an electric current. The magnitude of the current is directly proportional to the concentration of the gas being detected. EC gas sensors are widely used in industrial, commercial, and residential settings to monitor and control the levels of toxic or flammable gases, such as carbon monoxide, methane, and hydrogen sulfide. EC gas sensors offer several advantages over other types of gas sensors, including high sensitivity, accuracy, and selectivity.

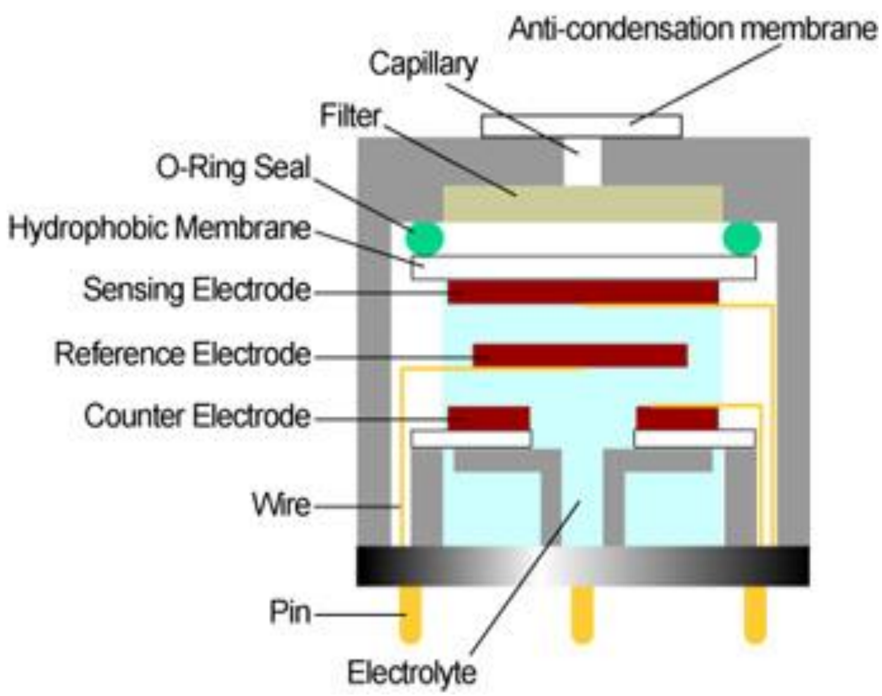

Figure 6 Electrochemical Gas Sensor

- **Catalytic Bead Sensor**

  Catalytic bead sensors are a type of gas sensor that are commonly used to detect the presence of combustible gases such as methane, propane, and hydrogen in industrial and domestic settings. The sensors consist of two small platinum wires coated with a catalyst such as palladium or platinum. When combustible gas is present, it reacts with the catalyst on the surface of the wires, causing the wires to heat up and increase in resistance. This change in resistance is then measured and used to determine the concentration of the gas in the surrounding environment. Catalytic bead sensors are highly sensitive and can detect combustible gases at very low concentrations, making them ideal for use in environments where gas leaks could be dangerous. They are also relatively inexpensive and easy to use, which has made them popular in a range of industries including oil and gas, chemical manufacturing, and mining.

Figure 7 Catalytic Bead Sensor

- **Optical Sensors**

Optical sensors are attractive for use in several EN applications, because of their compactness, immunity to electromagnetic interference and rapidity [19]. Fluorescence, optical layer thickness, colorimetric dye response, light polarization and absorbance are measured by optical sensors, and any of these optical changes are used to detect odors in the environment [7]. There are two special types of optical sensors, which detect subjects based on the differences of color, i.e., colorimetric sensors, and light, i.e., fluorescence sensors. Colorimetric sensors are composed of thin films made of chemically responsive dyes. Fluorescence sensors which are more sensitive than colorimetric sensors, identify fluorescent light emissions released by samples.

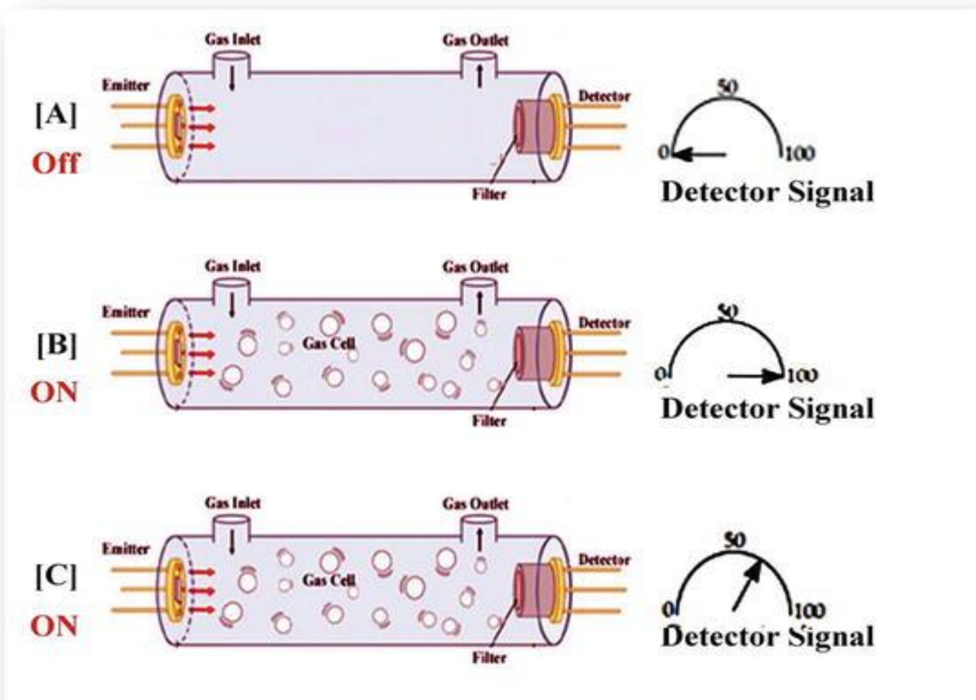

Figure 8 Optical Sensor

- **Photoionization detector Sensor**

A photoionization detector (PID) is a type of gas sensor that detects volatile organic compounds (VOCs) in the air by ionizing the gas molecules using ultraviolet (UV) light. The PID is commonly used in industrial and environmental applications, such as detecting hazardous gases in factories, monitoring air quality in cities, and testing for leaks in underground storage tanks. The PID works by emitting UV light onto the gas sample, which ionizes the gas molecules and produces positively charged ions and free electrons. The positively charged ions are attracted to the negatively charged electrode, while the free electrons are attracted to the positively charged electrode. The resulting electrical current is proportional to the concentration of the VOCs in the gas sample. One advantage of the PID is its high sensitivity to low levels of VOCs. The device is also fast and easy to use, requiring no preparation or calibration.

Figure 9 Photoionization Detector Sensor

2. CHAPTER

2.1 System Model

The Electronic Nose system comprises of (1) the sensing mechanism that will generate the sensor resistance values depending on the gases it will detect and (2) the Arduino Uno the main microcontroller board, that would transform the gas sensor array functional & store the analog values to a data acquisition card. The gases emitted from the samples is sensed by the array of sensors made up of MQ series: MQ3 for Alcohol and MQ 7 for Carbon Monoxide, a series of gas-sensitive sensors that detects volatile compounds concentrations. The concentration of the gases are sensed separately from the two stages of the fruit ripening – fresh & ripened. The gaseous concentration is determined by the ppm levels, along with the corresponding change in sensor voltage values obtained, when the sensor reacts to each of the respective gases desired to be measured. The data values are read to the Secure Digital (SD) Card, of 1 GB in a text format which has an extension of ".txt". The experiment is conducted purely at room temperature with the effect of change in room temperature on the level of spoilage of the target food being not in scope of concern for the experimental setup. The gas sensor array is mounted on the breadboard in a parallel fashion. The sensor array is supplied with an input voltage of +5V according to the optimum rating of the MQ series gas sensors along with the analog pin of MQ7 sensor connected to the analog pin (A5) & that of MQ3 gas sensor wired to the analog pin (A3) of the Arduino Uno board. A breadboard which has an internal connectivity of series & parallel connections, is used for interconnecting the gas sensors to the Arduino Uno board. The gas sensors are grounded to the ground pin (Gnd) of the Arduin Uno Board. In order to ensure, that the sensors detect the particular gas concentration in the aroma of the target food material in this experiment “banana”, the sensors were allowed to read the respective gas concentration in air, so as to stabilize the fluctuations of the sensor’s voltage level until, the sensor is heated up to a particular temperature. The data values obtained are converted into the comma separated format(.csv) from the text(.txt) to create the dataset in an excel file, the best suitable format to perform the applications of data science. Python is selected as the primary scripting language for the data science application & to script the Machine Learning Algorithms, owing to its popularity in the paradigm & wide range of libraries & functionalities. At first, the dataset is checked for missing or null values, which is tackled using the NumPy library of python & a Standard Scaling is performed onto the dataset. The principal components of the dataset that define classification between spoiled & fresh banana, are analyzed by Principal Component Analysis (PCA) which is a feature extractor & unsupervised learning clustering algorithm. A Linear relationship between the ppm level concentration & change in sensor voltage value for each of the target gas exists, which is to be by the linear regression analysis for the observed values of each of the target gases. Finally, Support Vector Machine (SVM) Classification, a supervised machine learning approach is used for identifying the distinct categories & classify the spoiled & fresh banana.

2.2 Hardware Components

- **Arduino Uno**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010.[2][3] The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.[4] It can be powered by a USB cable or a barrel connector that accepts voltages between 7 and 20 volts, such as a rectangular 9-volt battery. It is similar to the Arduino Nano and Leonardo.[5][6] .[7] The Uno board was the successor of the Duemilanove release and was the 9th version in a series of USB-based Arduino boards.[8] Version 1.0 of the Arduino IDE for the Arduino Uno board has now evolved to newer releases.[4] The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.[3]While the Uno communicates using the original STK500 protocol,[1] it differs from all preceding boards in that it does not use a FTDI USB-to-UART serial chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pin Mode (), digital Write (), and digital Read () functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analog Reference() function.[9]In addition, some pins have specialized functions:

- Serial / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
- External interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analog Write () function.
- SPI (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
- TWI (two-wire interface) / I²C: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
- AREF (analog reference): Reference voltage for the analog inputs.

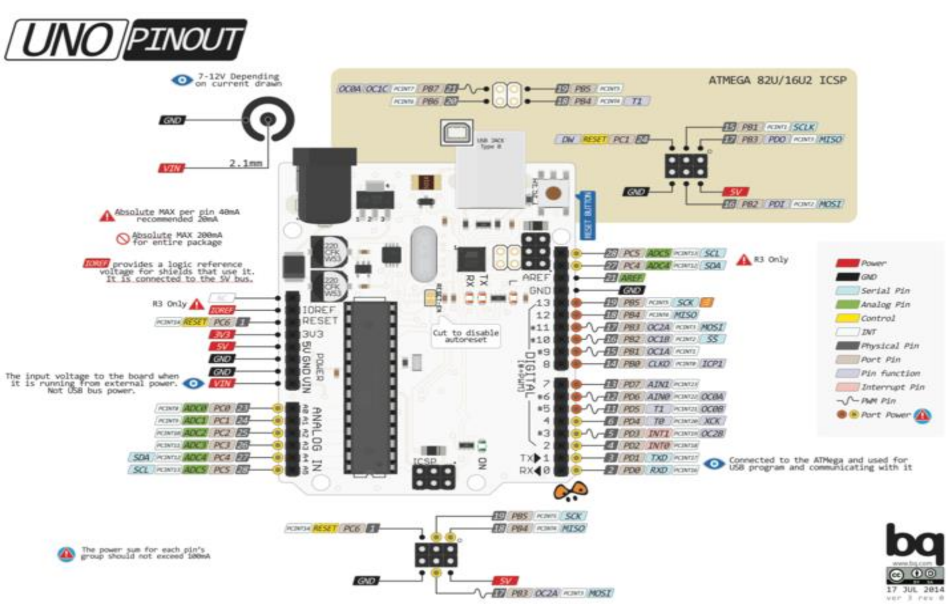

Figure 10 Arduino Uno Board

- **MQ3 Gas Sensor**

MQ-3 module is suitable for detecting Alcohol, Benzine, CH4, Hexane, LPG, CO. Sensitive material of MQ-3 gas sensor is SnO2, which with lower conductivity in clean air. When the target alcohol gas exists, the sensor's conductivity is higher along with the gas concentration rising. MQ-3 gas sensor has high sensitivity to Alcohol, and has good resistance to disturb of gasoline, smoke and vapor. This sensor provides an analog resistive output based on alcohol concentration. When the alcohol gas exists, the sensor's conductivity gets higher along with the gas concentration rising. There is a resistance across an A and B inside the sensor which varies on detection of alcohol. More the alcohol, the lower the resistance. The alcohol is measured by measuring this resistance. The sensor and load resistor form a voltage divider, and the lower the sensor resistance, the higher the voltage reading will be.

Features of MQ3:

- High sensitivity to alcohol and small sensitivity to Benzine.
- Fast response and High sensitivity
- Stable and long life
- Simple drive circuit

The enveloped MQ3 have 6 pins ,4 of them is used to fetch signals, and other 2 are used for providing heating current. Resistance value of MQ-3 is difference to various kinds and various concentration gases. So, when using this component, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 0.4mg/L (approximately 200ppm) of Alcohol concentration in air and use value of Load resistance that (RL) about 200 KΩ (100KΩ to 470 KΩ).

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

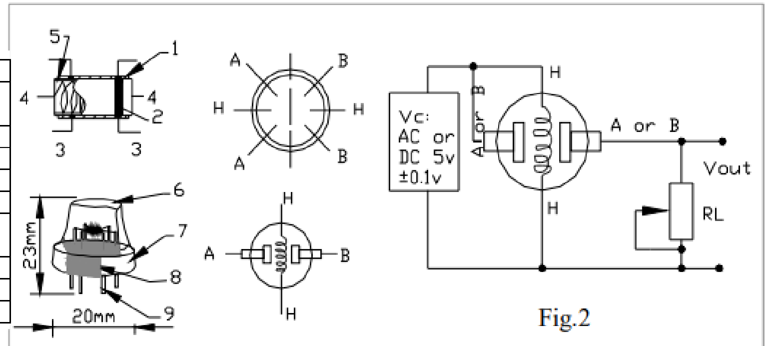

Fig. 1

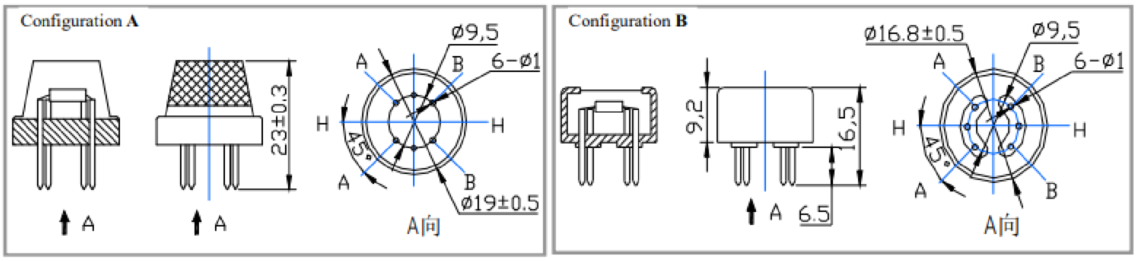

Figure 11 Structure & Configuration, Basic Measuring Circuitry

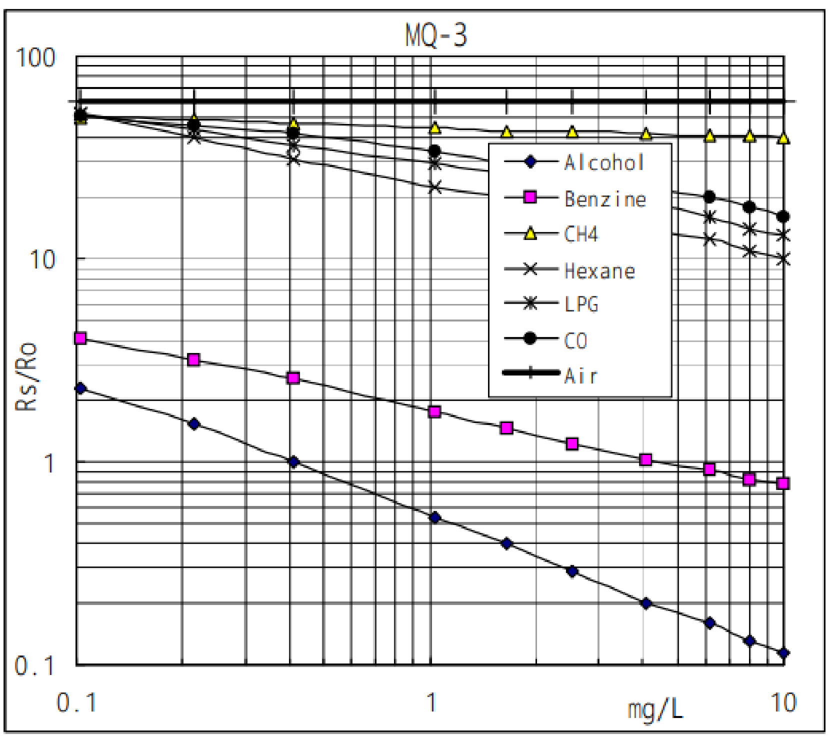

Figure 12 Sensitivity Characteristics of MQ3

- **MQ7 Gas Sensor**

MQ7 Gas sensor is another one of Metal Oxide Semiconductor (MOS) type Gas Sensor of MQ Gas Sensors family involving MQ 2, MQ 4, MQ 3, MQ 8, MQ 135, etc. It is mainly used to detect Carbon Monoxide. This sensor contains a sensing element, mainly aluminum-oxide based ceramic, coated with Tin dioxide (SnO_2), enclosed in a stainless-steel mesh. Whenever CO gas comes into contact with the sensing element, the resistivity of the element changes. The change is then measured to get the concentration of the gases present. The MQ7 Sensor has a small heating element present which is needed to preheat the sensor to get it in the working window. It can detect Carbon Monoxide Gas in the range of 20 PPM to 2000 PPM in the air.

Features of MQ3:

- High sensitivity to carbon monoxide
- Stable and long life

The enveloped MQ-7 have 6 pins ,4 of them is used to fetch signals, and other 2 are used for providing heating current.

The surface resistance of the sensor Rs is obtained through effected voltage signal output of the load resistance RL which series-wound. The relationship between them is described:

$$Rs\backslash RL = (Vc-VRL) / VRL$$

Sensitive layer of MQ-7 gas sensitive components is made of SnO_2 with stability, So, it has excellent long-term stability. Its service life can reach 5 years under using condition.

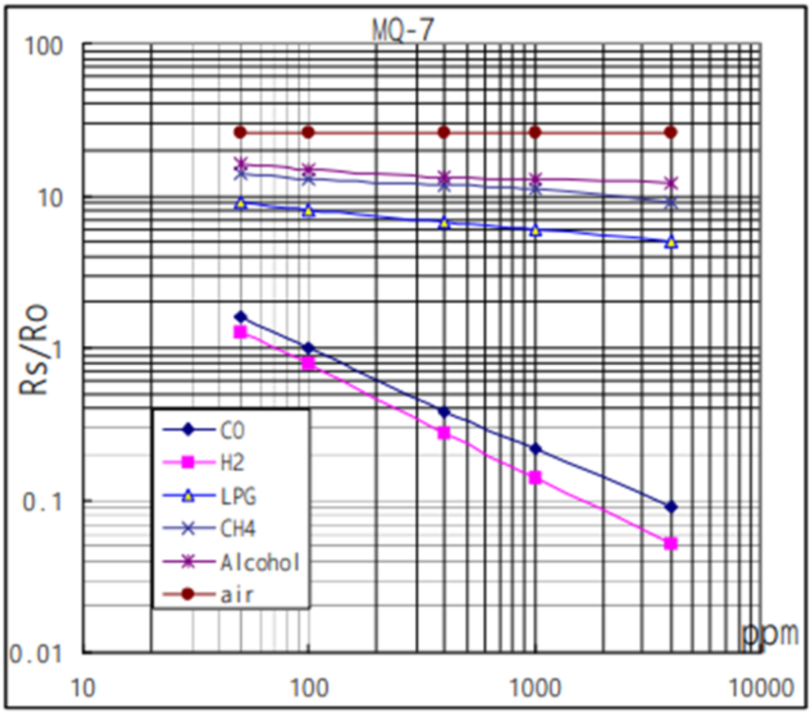

Fig.3 sensitivity characteristics of the MQ-7

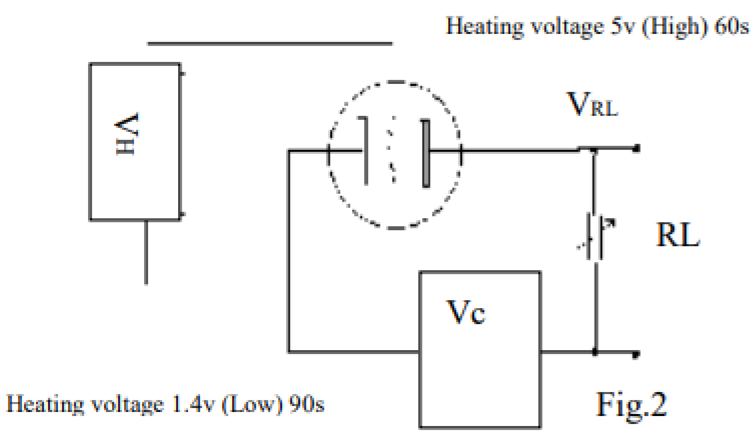

Figure 13 Standard Circuitry of MQ7

2.3 Software Components

- **Support vector machines (SVMs)**

Support Vector Machines (SVMs) are a popular type of machine learning algorithm used for classification and regression. SVMs work by finding a hyperplane that best separates the data points into different classes. The hyperplane is defined as a linear function that takes the form of f(x) = w ^ T x + b, where w is a weight vector, b is a bias term, and x is the input feature vector. The goal of the Linear SVM is to find the hyperplane that best separates the data points into two classes. A hyperplane is a linear function that divides the feature space into two regions. In two dimensions, a hyperplane is a line that separates the data points into two classes. In three dimensions, a hyperplane is a plane that separates the data points into two classes. To find the optimal hyperplane, the Linear SVM uses a technique called maximum margin classification. The margin is the distance between the hyperplane and the closest data points from each class. The Linear SVM aims to find the hyperplane that maximizes this margin. The distance between a point x_i and the hyperplane f(x) is given by:

$$\text{distance}\ (x\_\ I, f(x)) = |f\ (x\ \_\ i)\ |\ /\ ||w||$$

where ||w|| is the magnitude of the weight vector w. The goal of maximum margin classification is to find the hyperplane that maximizes the distance between the closest data points from each class. This distance is called the margin. The distance between the closest data points from each class is given by:

$$\text{margin} = (\text{distance}\ (x+, f(x)) + \text{distance}\ (x-, f(x)))\ /\ 2$$

where x+ is the closest data point from the positive class and x- is the closest data point from the negative class. The optimization problem for maximum margin classification can be formulated as:

$$\text{minimize: } 1/2\ ||w||\text{^}2$$

subject to:

$$y_i (w^T x_i + b) >= 1 \text{ for all } i$$

where y_i is the label for the ith sample, and x_i is its corresponding feature vector. The constraint y_i (w^T x_i + b) >= 1 ensures that all data points are correctly classified.

The optimization problem can be solved using a variety of techniques, such as quadratic programming or gradient descent. The resulting hyperplane is called the maximum margin hyperplane.

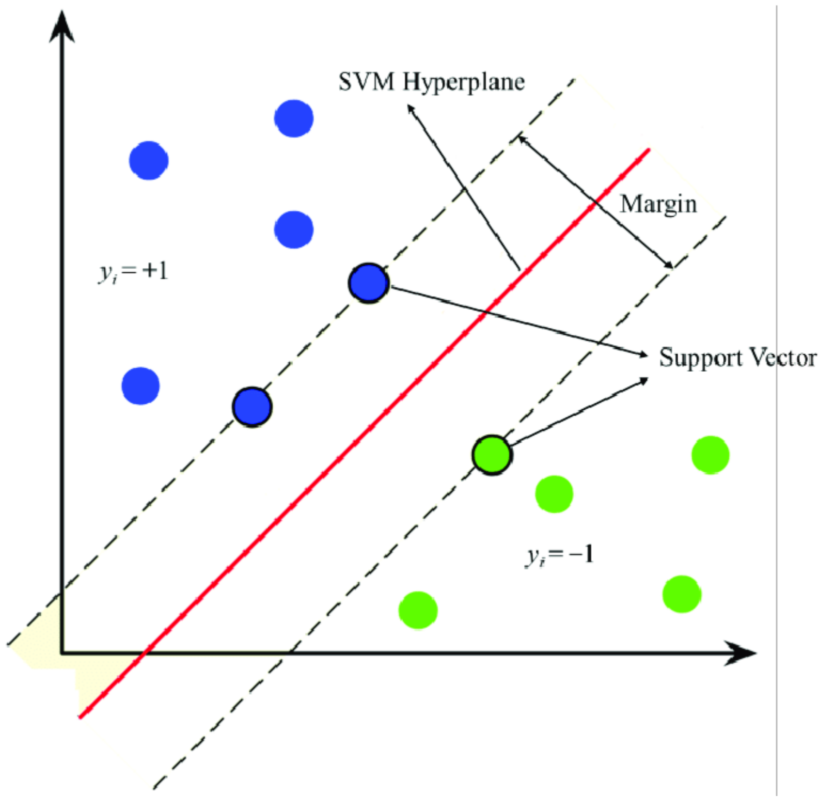

Figure 14 Linear Support Vector Machine

- **Principal Component Analysis**

PCA is a linear transformation that operates on the data by computing the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors of the covariance matrix represent the directions of maximal variance in the data, while the eigenvalues represent the magnitude of this variance. PCA works by computing the eigenvectors corresponding to the largest eigenvalues, PCA can reduce the dimensionality of the data while retaining the most important information. The covariance matrix of the eigenvectors & eigenvalues, is a square matrix that summarizes the relationship between each pair of variables in the data. The element in row i and column j of the covariance matrix represents the covariance between variables i and j.

The covariance between variables i and j can be computed as:

$$cov\ (i, j) = 1/(n-1)\ \Sigma\ (x\_i - \mu\_i)\ (x\_j - \mu\_j)$$

where μ_i is the mean of variable i across all samples.

The covariance matrix can be written as:

C = [cov (1,1) cov (1,2) ... cov (1, m)]

[cov (2,1) cov (2,2) ... cov (2, m)]

[... ... ...]

[cov(m,1) cov(m,2) ... cov (m, m)]

PCA computes the eigenvectors and eigenvalues of the covariance matrix using an algorithm called the Power Iteration method. The Power Iteration method starts with a random vector v and iteratively computes the product Cv until convergence. The resulting vector is an eigenvector of C with the largest eigenvalue. The process is repeated to find additional eigenvectors and eigenvalues. The eigenvectors of the covariance matrix represent the directions of maximal variance in the data. The eigenvalue represents the magnitude of this variance. The first principal component is the eigenvector with the largest eigenvalue. The second principal component is the eigenvector with the second largest eigenvalue, and so on. The principal components form an orthogonal basis for the new coordinate system. The principal components computed by PCA have several interpretations. One interpretation is that they represent the most important directions in the data, in the sense that they capture the most variance. Another interpretation is that they represent patterns in the data, such as correlations between features.

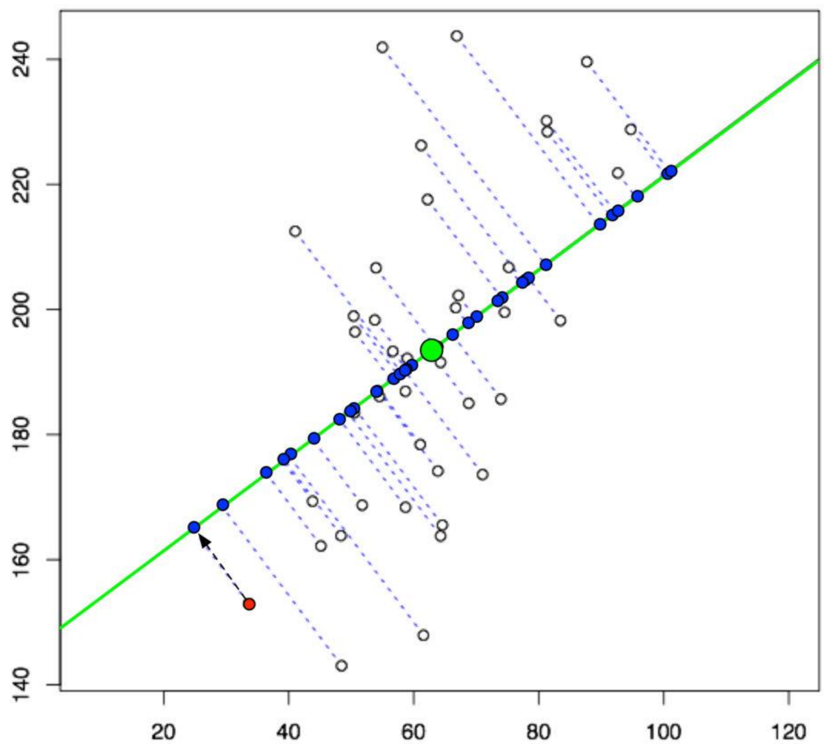

Figure 15 Interpretation of the Principal Components

- **Linear Regression**

Linear regression is a widely used technique in machine learning and data analysis for predicting a continuous output variable based on one or more input variables. It works by modeling the relationship between the input variables and the output variable as a linear function. In this article, we will discuss the basic concepts behind linear regression, including the computation of the regression coefficients, the interpretation of the regression model, and the applications of linear regression in data analysis. A linear function has the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m$$

where β_0 is the intercept, and β_1, β_2, ..., β_m are the regression coefficients for each input variable.

Linear regression works by estimating the regression coefficients that minimize the difference between the predicted values of y and the actual values of y in the training data. This is typically done by minimizing the sum of squared errors:

$$SSE = \Sigma (y_i - \hat{y}_i)^2$$

where y_i is the actual value of y for the i-th sample, and $\hat{y}_i$ is the predicted value of y based on the linear function. The regression coefficients can be computed using the ordinary least squares (OLS) method. The OLS method minimizes the sum of squared errors by finding the values of the regression coefficients that satisfy the following normal equations:

$$X^T X \beta = X^T y$$

where X is a matrix of size n x m, with each row representing a sample, and each column representing an input variable. The first column of X is a vector of ones representing the intercept. β is a vector of size m+1, with the first element representing the intercept, and the remaining elements representing the regression coefficients. y is a vector of size n representing the output variable.

The solution to the normal equations is:

$$\beta = (X^T X)^{-1} X^T y$$

Once we have computed the regression coefficients, we can use the linear function to predict the output variable for new input variables. The regression model computed by linear regression has several interpretations. One interpretation is that it represents the best linear approximation of the relationship between the input variables and the output variable. Another interpretation is that it represents the average effect of the input variables on the output variable, assuming that all other variables are held constant.

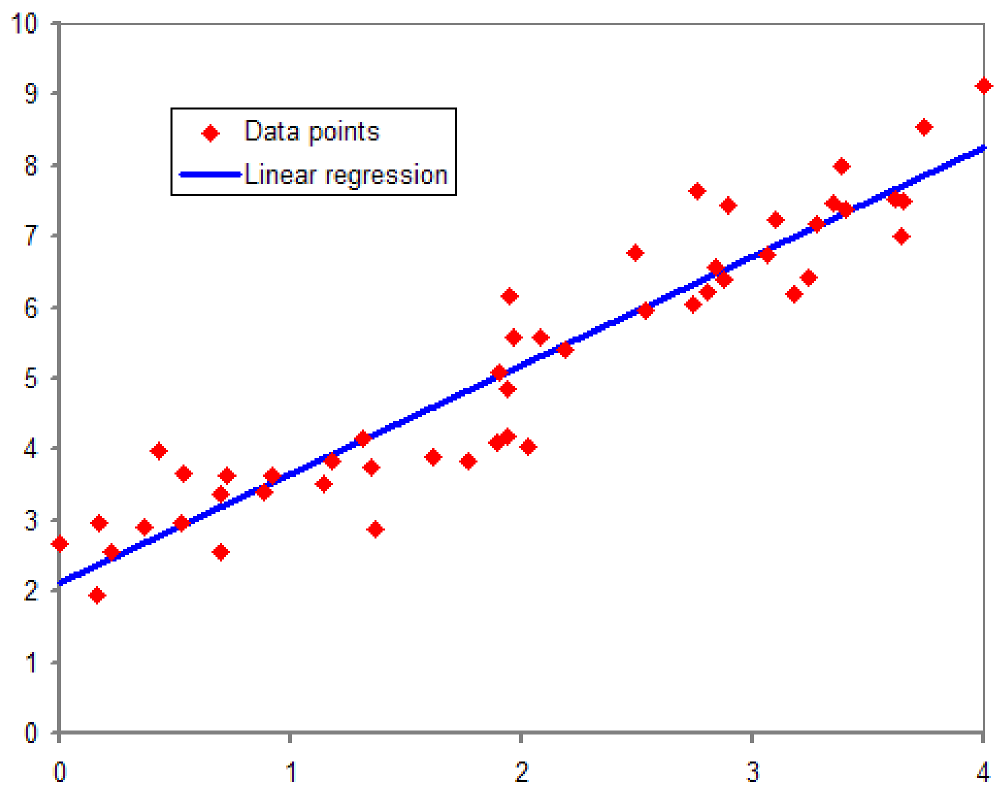

Figure 16 Visualization of the linear relationship with Linear Regression

3. CHAPTER

3.1 Results & Discussion

- **The Novel Dataset**

ppm level(CO)	sensor_voltage (CO)	ppm level (Alcohol)	sensor_voltage (Alcohol)	Class
55.09	3.85	0.4	2.12	1
53.69	3.85	0.5	2.13	1
53.69	3.83	0.51	2.13	1
52.35	3.8	0.51	2.13	1
51.91	3.79	0.51	2.13	1
52.79	3.8	0.51	2.13	1
18.75	3.15	0.51	0.36	0
19.12	3.17	0.02	0.4	0
18.87	3.16	0.03	0.42	0
13.96	2.92	0.03	0.45	0
13.62	2.9	0.03	0.47	0
16.51	3.05	0.04	0.47	0
18.28	3.13	0.04	0.52	0
19.37	3.18	0.04	0.53	0
19.37	3.18	0.04	0.45	0

Figure 17 Experimental Dataset

The experimental dataset collected is a labelled dataset, the accuracy & correctness of which is understood & well-defined by the specifications mentioned in the functioning of the hardware setup & the environmental conditions under which the data collection procedure is carried out. The values observed in the dataset has the recorded readings from the MQ7 for Carbon Monoxide (CO) & MQ3 for Alcohol mapped together in a single ".csv" format. The dataset contains four features among which, two- "ppm level (CO) & ppm level (Alcohol)" are dependent variables upon the “sensor_voltage value (CO)” & “sensor_voltage value (Alcohol)” which are the independent variables. The dataset contains 184 samples of spoiled & 180 samples of fresh banana. Upon observation of the dataset, it can be concluded as the sensor voltage value increases gradually the corresponding ppm levels increases which holds true for both of the observed gas concentrations & satisfies the working principle of the MOS gas sensors. The dataset has 363 null values hence data cleaning techniques need to be performed before analysis of the principal components. The simple overview of the dataset makes it evident enough that recorded ppm levels & the corresponding sensor voltage values for each of the gas concentrations in the aroma of the target food “banana”, provides a distinction between the two stages of the banana – “fresh & spoiled”. Hence, the dataset consists of two classes-“Class 1& 0”, with Class 1 belonging to the spoiled features while Class 0 denoting the fresh features in the dataset.

- **Principal Component Analysis**

Scikit-learn machine learning framework for implementation in Python is used which is the most popular among data scientists. The NumPy, pandas, matplotlib & seaborn frameworks are used to implement the proposed system. The dataset is divided into two categories, fresh & spoiled. The evaluation was performed on the independent variables of the dataset, which are namely “sensor_voltage value (CO) & sensor_voltage value (Alcohol)” of the novel dataset. The first principal component (PC1) captures 93% of the variance between the principal components & the second principal component captures remaining 7% of the dataset. The Principal Component Analysis plot is shown below, where the two clusters of the eigen values of the two principal components display a clear distinction between the two classes of classification, hence a good fit hyperplane can be plotted to classify the two categories.

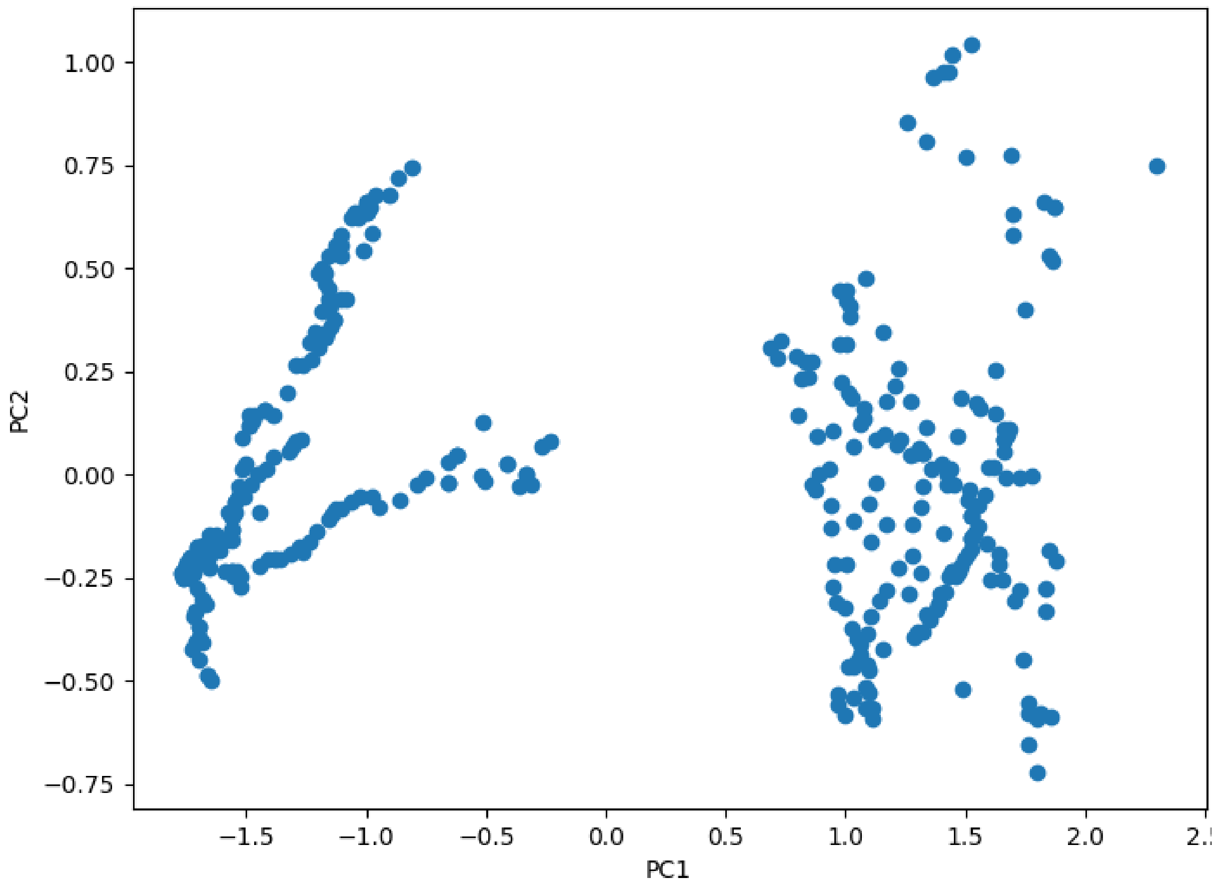

Figure 18 Scatter Plot for Principal Component Analysis

- **Linear Regression Analysis**

Scikit-learn machine learning framework for implementation in Python is used which is the most popular among data scientists. The NumPy, pandas, matplotlib & seaborn frameworks are used to implement the proposed system. The dataset is divided into two categories, fresh & spoiled. The evaluation was performed to observe the linear relationship between the sensor_voltage value & ppm levels of Carbon Monoxide & Alcohol respectively, the visualization of which are shown below. The Mean Square Error value for the Carbon Monoxide concentration values was found to be 0.0179 & the R^2 value was found to be 0.94, which defines an overall good fitting of the model. The Mean Square Error value for the Alcohol concentration values was found to be 1.3548 & the R^2 value was found to be 0.96, which defines an overall good fitting of the model.

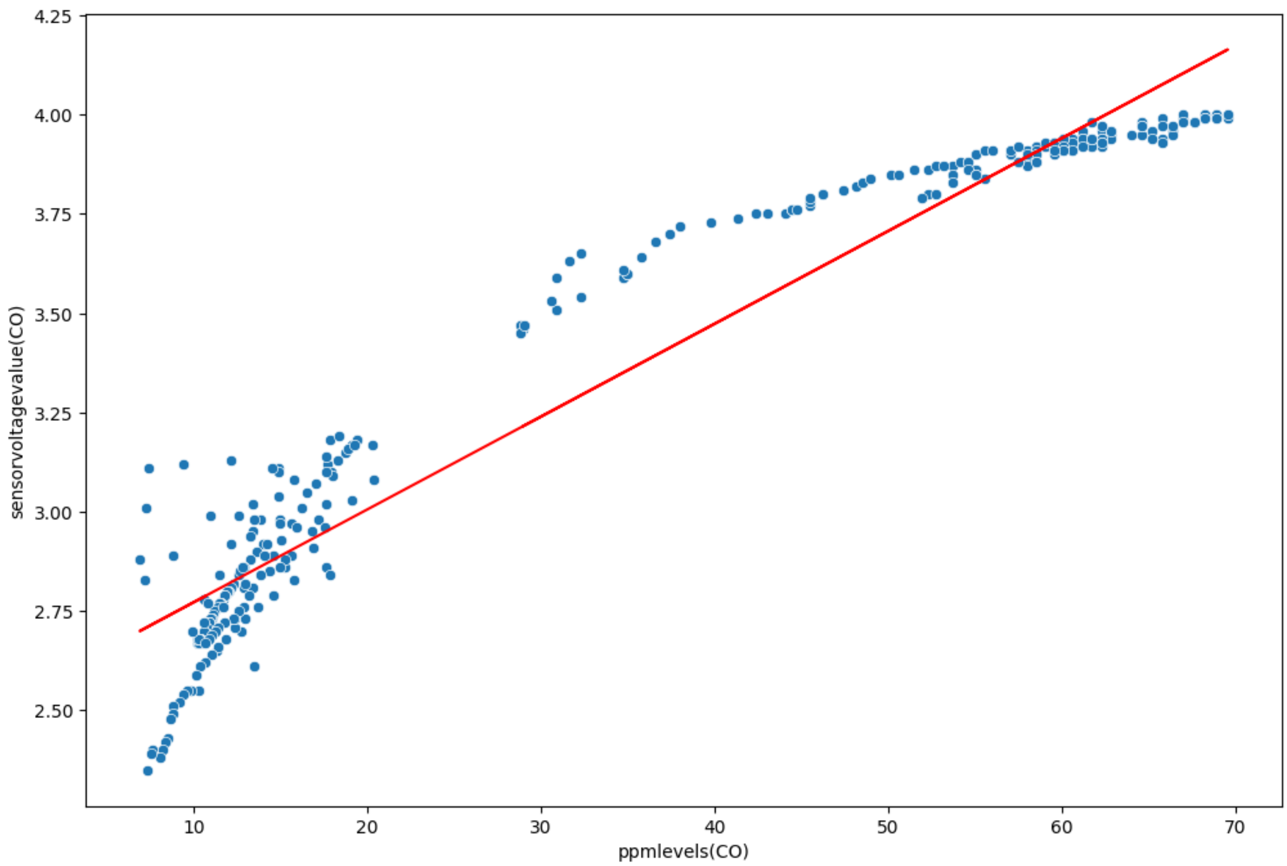

Figure 19 Scatter plot for Carbon Monoxide

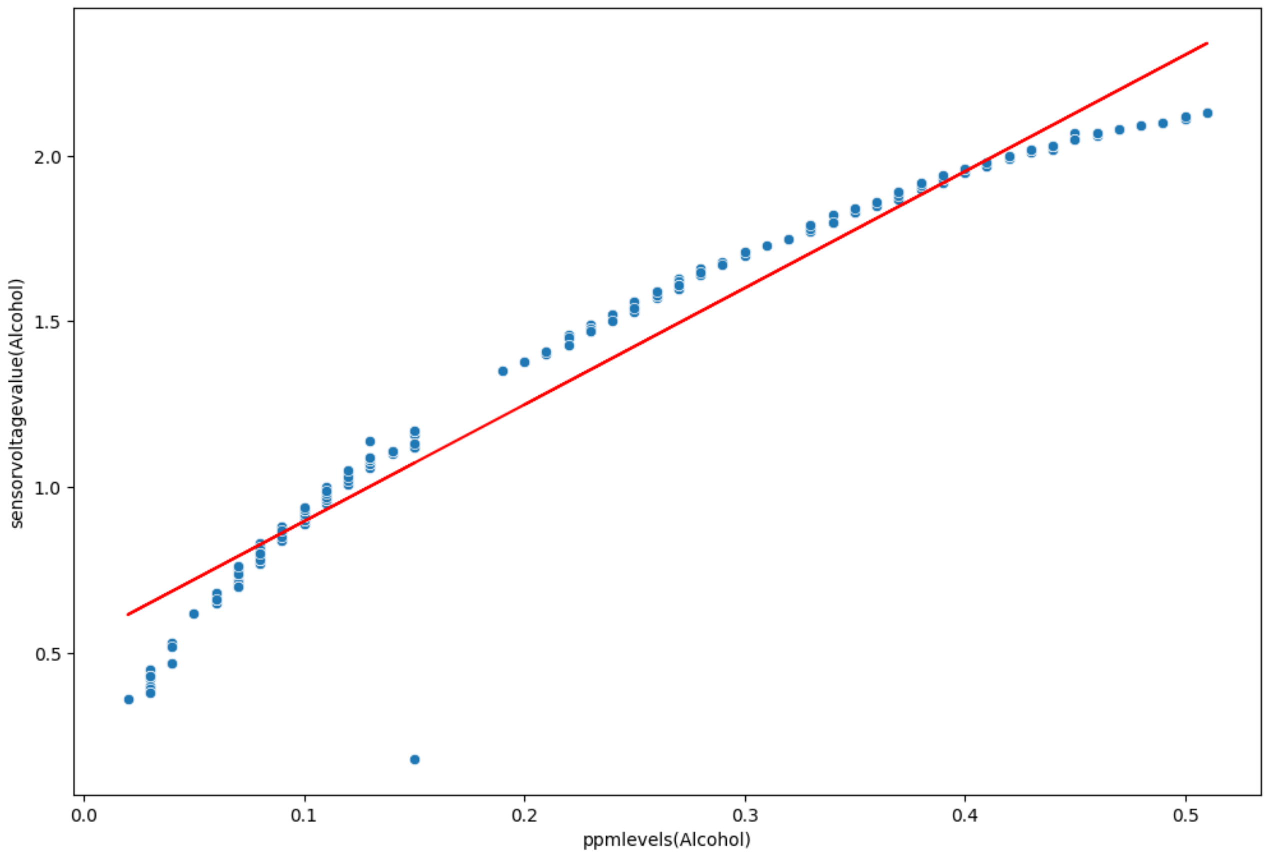

Figure 20 Scatter Plot for Alcohol

- **Support Vector Machine**

Scikit-learn machine learning framework for implementation in Python is used which is the most popular among data scientists. The NumPy, pandas, matplotlib & seaborn frameworks are used to implement the proposed system. The dataset is divided into two categories, fresh & spoiled. The evaluation was performed to classify between the sensor_voltage value of Carbon Monoxide & sensor_voltage value of Alcohol respectively, the visualization of which are shown below. The dataset is cleansed to remove any null or missing values using NumPy library functions & normalized with StandardScaler of the Scikit-learn framework. An accuracy of 96% is achieved by cross-validation which mitigated an overfitting issue with the model. The linear SVM model fits a hyperplane with a clear distinct classification of the two classes of fruit- spoiled belonging to class 1 & fresh to class 0.

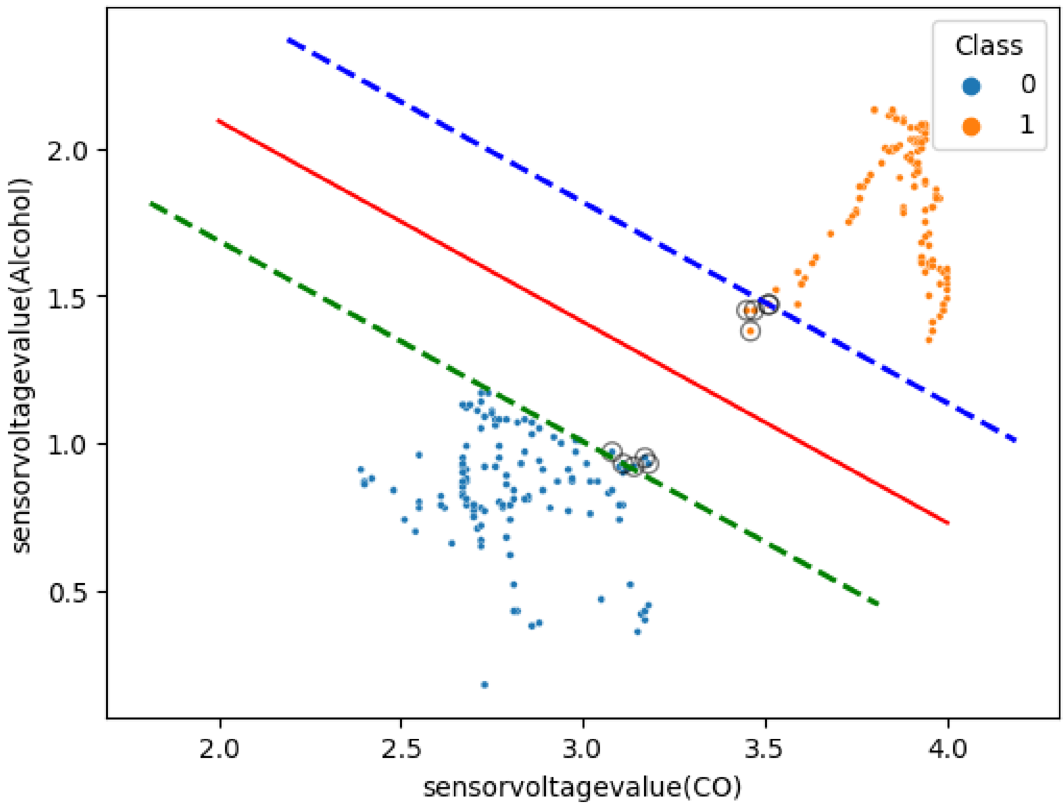

Figure 21 Scatter Plot of Linear Support Vector Machine

4 CHAPTER

4.1 CONCLUSION

The replication of human senses has been one of the most popular research topics for decades and the EN technology is a product of these long-term studies. Since the day of the first prototype EN was developed, it has become a useful device in several applications, in solving urgent problems. Especially in the food industry and medicine, traditional ways of distinguishing subjects are too slow, expensive and mostly subjective which may result in fatal errors. The EN technology instead offers a fast, sensitive, low cost and objective alternative. Moreover, the potential of this new way of sensing is still rapidly growing with the new developments in sensors and machine learning technologies.

It can be noticed that Principal Component Analysis takes about 15s to build its model & Linear Support Vector Machine which takes about 20 s to build its model with an accuracy of 96%, hence a much faster & efficient in distinguishing between which fruit is spoiled & which is still fresh. Hence, advancements in machine learning are a major breakthrough in deploying Electronic Nose Technology in the supply chains of the food industry.

APPENDIX

- **Arduino Script**

```
#include <SPI.h>

#include <SD.h>

const int chipSelect = 4;

int gas_sensor = A5; //Sensor pin

float m = -0.754896547; //Slope

float b = 1.30; //Y-Intercept

float R0 = 1.04;

void setup () {

  Serial. Begin (9600);

  while (! Serial) {

   ;

  }

  Serial. Print ("Initializing SD card...");

  if (! SD. begin(chipSelect)) {

    Serial.println("Card failed, or not present");

    while (1);

  }

  Serial.println("card initialized.");

  pin Mode (gas_sensor, INPUT);

}
```

```
void loop () {
  float sensor_volt;
  float RS_gas;
  float ratio;
  int sensorValue = analogRead(gas_sensor);
  sensor_volt = sensorValue*(5.0/1023.0);
  RS_gas = ((5.0*10.0)/sensor_volt)-10.0;
  ratio = RS_gas/R0;
  double ppm_log = (log10(ratio)-b)/m;
  double ppm = pow (10, ppm_log);
  double percentage = ppm/10000;
  String dataString = "";
  for (int analogPin = 0; analogPin < 3; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 2) {
      dataString += ",";
    }
  }
  File dataFile = SD. open ("datalog10.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.println(sensorValue);
    dataFile.println(",");
```

```
    dataFile.println(ppm);
    dataFile.close();
    Serial.println(sensorValue);
    Serial.println(ppm);
  }
  else {
    Serial.println("error opening datalog10.txt");
  }
}
```

- **Principal Component Analysis Model**

```
import pandas as pd
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv("/content/drive/MyDrive/VOC_collab.csv")
df.head()
X=df.iloc[:,[1,3]]
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
scalar.fit(X)
scaled_data = scalar.transform(X)
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
```

```
pca.fit(scaled_data)

x_pca = pca.transform(scaled_data)

x_pca.shape

plt.figure(figsize =(8, 6))

plt.scatter(x_pca[:,0],x_pca[:,1],cmap ='plasma')

plt. xlabel('PC1')

plt.ylabel('PC2')
```

- **Linear Regression Model for Carbon Monoxide**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

from google.colab import drive

drive.mount('/content/drive')

df = pd.read_csv("/content/drive/MyDrive/VOC_collab.csv")

X=df['ppmlevels(CO)'].values

y=df['sensorvoltagevalue(CO)'].values

plt.rcParams["figure.figsize"] = (12,8)

sns.scatterplot(x='ppmlevels(CO)', y='sensorvoltagevalue(CO)', data=df);

def loss(data, data_pred):

    N=data.shape[0]

    loss=np.sum(np.square(y-y_coeff_model))/N

    re m=sum((X-np.mean(X)) * (y - np.mean(y)))/sum((X-np.mean(X))**2)
```

```
c=np.mean(y) - m*(np.mean(X))

y_coeff_model= m*X + c

loss(y,y_coeff_model)

return loss

plt.rcParams["figure.figsize"] = (12,8)

sns.scatterplot(x='ppmlevels(CO)', y='sensorvoltagevalue(CO)', data=df);

plt.plot(X, y_coeff_model,c='r');

plt.show()

print("The value of c is {} and m is {}".format(c, m))\

from sklearn.metrics import r2_score

r2 = r2_score(y,y_coeff_model )
```

- **Linear Regression Model for Alcohol**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

from google.colab import drive

drive.mount('/content/drive')

df = pd.read_csv("/content/drive/MyDrive/VOC_collab.csv")

X=df['ppmlevels(Alcohol)'].values

y=df['sensorvoltagevalue(Alcohol)'].values

plt.rcParams["figure.figsize"] = (12,8)

sns.scatterplot(x='ppmlevels(Alcohol)', y='sensorvoltagevalue(Alcohol)', data=df);
```

```
def loss(data, data_pred):
    N=data.shape[0]
    loss=np.sum(np.square(y-y_coeff_model))/N
    re m=sum((X-np.mean(X)) * (y - np.mean(y)))/sum((X-np.mean(X))**2)
c=np.mean(y) - m*(np.mean(X))
y_coeff_model= m*X + c
loss(y,y_coeff_model)
return loss
plt.rcParams["figure.figsize"] = (12,8)
sns.scatterplot(x='ppmlevels(CO)', y='sensorvoltagevalue(CO)', data=df);
plt.plot(X, y_coeff_model,c='r');
plt.show()
print("The value of c is {} and m is {}".format(c, m))\
from sklearn.metrics import r2_score
r2 = r2_score(y,y_coeff_model )
```

- **Support Vector Machine Model**

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv("/content/drive/MyDrive/VOC_collab.csv")
```

```
df.columns

df.info()

df.describe().transpose()

df.isnull().sum()

X=df.iloc[:,[1,3]]

Y=df.iloc[:,4]

from sklearn.svm import SVC

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, confusion_matrix

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,stratify=Y,random_state=99)

svc_model = SVC(C=.1, kernel='linear', gamma=1)

svc_model.fit(X_train,Y_train)

prediction = svc_model .predict(X_test)

print(svc_model.score(X_train, Y_train))

print(svc_model.score(X_test, Y_test))

print("Confusion Matrix:\n",

    confusion_matrix(prediction, Y_test))

sns.scatterplot(x=X_train.iloc[:,0],y=X_train.iloc[:,1],hue=Y_train,s=8);

w=svc_model.coef_[0]

b=svc_model.intercept_[0]

x_points=np.linspace(2,4)

y_points=-(w[0]/w[1])*x_points -b/w[1]

plt.plot(x_points,y_points,c='r')
```

```
plt.scatter(svc_model.support_vectors_[:,0],svc_model.support_vectors_[:,1],s=50,facecolors='n
one',edgecolors='k',alpha=0.5);
w_hat = svc_model. coef_ [0] / (np. sqrt(np.sum(svc_model. coef_[0] ** 2)))
margin = 1 / np. sqrt(np.sum(svc_model. coef_ [0] ** 2))
decision_boundary_points = np. array (list (zip (x_points, y_points)))
points_of_line_above = decision_boundary_points + w_hat * margin
points_of_line_below = decision_boundary_points - w_hat * margin
plt.plot(points_of_line_above[:, 0], points_of_line_above[:, 1], 'b--', linewidth=2)
plt. plot (points_of_line_below[:, 0], points_of_line_below[:, 1], 'g--',linewidth=2)
```

BIBLIOGRAPHY

1. T. M. Dymerski, T. M. Chmiel, W. Wardencki. Invited review article: An odor-sensing system-powerful technique for foodstuff studies. Review of Scientific Instruments, vol.82, no.11, Article number 111101, 2011. DOI: 10.1063/1.3660805

2. K. Mori. The Olfactory System: From Odor Molecules to Motivational Behaviors, Tokyo, Japan: Springer, 2014. DOI: 10.1007/978-4-431-54376-3.

3. B. Szulczynski, K. Arminski, J. Namiesnik, J. Gebicki. Determination of odour interactions in gaseous mixtures using electronic nose methods with artificial neural networks. Sensors, vol.18, no.2, Article number 519, 2018. DOI: 10.3390/s18020519

4. Jenny Gustavsson, Christel Cederberg, Ulf Sonesson, "Global food losses and food waste", Swedish Institute for Food and Biotechnology (SIK) Gothenburg, Sweden, Accessed on: March, 31, 2020. [Online]. Wu, Yujiao, et al. "Air Quality Monitoring for Vulnerable Groups in Residential Environments Using a Multiple Hazard Gas Detector." Sensors 19.2 (2019): 362

5. Tiwari, Abhai, Shashikant Sadistap, and S. K. Mahajan. "Development of Environment Monitoring System Using the Internet of Things." Ambient Communications and Computer Systems. Springer, Singapore, 2018. 403-412.

6. Hassan, Najmul, et al. "The role of edge computing in the internet of things." IEEE Communications Magazine 56.11 (2018): 110-115

7. Sharma, Shree Krishna, and Xianbin Wang. "Live data analytics with collaborative edge and cloud processing in wireless IoT networks." IEEE Access 5 (2017): 4621-4635.

8. Chongthanaphisut, Phunvira, Thara Seesaard, and Teerakiat Kerdcharoen. "Monitoring of microbial canned food spoilage and contamination based on e-nose for smart home." 2015 12th

International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI CON). IEEE, 2015.

9. Green, Geoffrey C., Adrian DC Chan, and Rafik A. Goubran. "Monitoring of food spoilage with electronic nose: potential applications for smart homes." 2009 3rd International Conference on Pervasive Computing Technologies for Healthcare. IEEE, 2009

10. Wijaya, Dedy Rahman, et al. "Development of mobile electronic nose for beef quality monitoring." Procedia Computer Science 124 (2017): 728-735.

11. Alfian, Ganjar, et al. "Improving efficiency of RFID-based traceability system for perishable food by utilizing IoT sensors and machine learning model." Food Control 110 (2020): 107016.

12. Aung, Myo Min, and Yoon Seok Chang. "Temperature management for the quality assurance of a perishable food supply chain." Food Control 40 (2014): 198-207.

13. Zhang, Zhongheng. "Missing data imputation: focusing on single imputation." Annals of translational medicine 4.1 (2016).

14. Aggarwal, Neha, and Kirti Aggarwal. "A mid-point-based k-mean clustering algorithm for data mining." International Journal on Computer Science and Engineering 4.6 (2012): 1174-1180.

15. Kambhatla, Nandakishore, and Todd K. Leen. "Dimension reduction by local principal component analysis." Neural computation 9.7 (1997): 1493-1516.

16. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019.

17. scikit-learn (0.22.2) documentation, Accessed on: March. 26, 2020. [Online]. Available: https://scikit learn.org/stable/modules/generated/sklearn.linear_model. LogisticReg ression.html

18. Tolesa, G. N., and T. S. Workneh. "Influence of storage environment, maturity stage and pre-storage disinfection treatments on tomato fruit quality during winter in KwaZulu-Natal, South Africa." Journal of food science and technology 54.10 (2017): 3230-3242

19. M. Falasconi, M. Pardo, M. D. Torre, A. Bresciani, and G. Sberveglieri, "The novel EOS835 electronic nose and data analysis for evaluating coffee ripening," Sensors Actuat. B, vol. 110, no. 1, pp. 73–80, Sep. 2005.

20. M. G. Bremer, N. G. E. Smits, and W. Haasnoot, "Biosensor immunoassay for traces of hazelnut protein in olive oil," Anal. Bioanal. Chem., vol. 395, no. 1, pp. 119–126, Sep. 2009.

21. L. García-Gonzalez and R. Aparicio, "Detection of defective virgin olive oils by metal-oxide sensors," Eur. Food Res. Technol., vol. 215, no. 2, pp. 118–123, Aug. 2002

21. D. Wilson, M. Baietto. Applications and advances in electronic-nose technologies. Sensors, vol.9, no.7, pp.5099–5148, 2009. DOI: 10.3390/s9070/23

22. Loutfi, S. Coradeschi, G. K. Mani, P. Shankar, J. B. B. Rayappan. Electronic noses for food quality: A review. Journal of Food Engineering, vol.144, pp.103–111, 2015. DOI: 10.1016/j.jfoodeng.2014.07.019.

23. Thakral, S. Gupta. Artificial neural networks for e NOSE: A review. International Journal of Applied Engineering Research, vol.7, no.11, pp.1601–1604, 2012.

24. R. Pallas-Areny, J. G. Webster. Sensors and Signal Conditioning, 2nd ed., New York, USA: Wiley, 2000.

25. Koshets, Z. I. Kazantseva, Y. M. Shirshov, S. A. Cherenok, V. I. Kalchenko. Calixarene films as sensitive coatings for QCM-based gas sensors. Sensors and Actuators B: Chemical, vol.106, no.1, pp.177–181, 2005. DOI: 10.1016/j.snb.2004.05.054.

26. Berna. Metal oxide sensors for electronic noses and their application to food analysis. Sensors, vol.10, no.4, pp.3882–3910, 2010. DOI: 10.3390/s100403882

27. R. Mohamed, R. Yaacob, M. Mohamed, T. Dir and F. Rahim (2018). Food Freshness Using Electronic Nose and Its Classification Method: A Review. International Journal of Engineering & Technology, Vol. 7, No. 3.28, pp. 49-53

28. L. Capelli, S. Sironi and R. Del Rosso (2014). Electronic Noses for Environmental Monitoring Applications. Sensors, 14, 19979-20007; doi:10.3390/s141119979

29. I. Tazi, N. Isnaini, M. Mutmainnah and A. Ainur (2018). Principal Component Analysis (PCA) Method for Classification of Beef and Pork Aroma Based on Electronic Nose. Indonesian Journal of Halal Research, Vol. 1, No. 1, pp. 5-8.

30. N. Linsangan, H. Abubo, R. Matanguihan and M. Natividad (2013). Meat Authentication by Carbon Dioxide and Nitrogen Detection Using Electronic Nose System. Mapua University, Manila, Philippines

31. A. Shahid, J. Choi, A. Rana, H. Kim (2018). Least Squares Neural Network-Based Wireless E-Nose System Using an SNO2 Sensor Array. Sensors (Basel), 18(5): 1446. Doi: 10.3390/s18051446.

32. A. Amari, N. El Bari and B. Bouchikhi (2007). Electronic Nose for Anchovy Freshness Monitoring Based on Sensor Array and Pattern Recognition Methods: Principal Component Analysis, Linear Discriminant Analysis and SupportVectorMachine. Computing, Vol. 6, No.3, pp. 61-67.

33. L. Capelli, S. Sironi and R. Del Rosso (2014). Electronic Noses for Environmental Monitoring Applications. Sensors, 14, 19979-20007; doi:10.3390/s141119979

34. R. Sarno, D. Wijaya (2019). Recent Development in Electronic Nose Data Processing for Beef Assessment. Telkomnika, Vol. 17, No.1, pp. 337-348.

35. M. Kabe, S. Deshmukh, T. Mujawar, V. Bachuwar, L. Deshmukh and A. Shaligram (2015). An Electronic Nose with LabVIEW using SNO2 Based Gas Sensors: Application to Test Freshness of the Fruits. International Journal of Scientific & Engineering Research, Vol. 6, No. 4