Detection of Sugar Syrup Adulteration in Honey Using Near Infrared Spectroscopy

A Dissertation for

Course code and Course Title: ELE-625 Project

Credits: 16

Submitted in partial fulfilment of Master's Degree
M.Sc. in Electronics
by

MS. VIOLA FERNANDES

Seat No.: 22P0360015

ABC ID: 342036959940

PRN:201905918

Under the Supervision of

DR. MARLON SEQUIERA

School of Physical and Applied Sciences
Electronics

GOA UNIVERSITY
MAY 2024

Examined by:

Seal of the School

DECLARATION BY STUDENT

I hereby declare that the data presented in this Dissertation report entitled, **Detection of Sugar Syrup Adulteration in Honey Using Near Infrared Spectroscopy** is based on the results of investigations carried out by me in the M.Sc. Electronics at the School of Physical and Applied Sciences, Goa University under the Supervision of Dr/Prof. Marlon Sequiera and the same has not been submitted elsewhere for the award of a degree or diploma by me. Further, I understand that Goa University or its authorities will be not responsible for the correctness of observations / experimental or other findings given the dissertation. I hereby authorize the University authorities to upload this dissertation on the dissertation repository or anywhere else as the UGC regulations demand and make it available to any one as needed.

Name of Student:
Viola Fernandes

Signature: Fernandes

Seat no: 22P0360015

Date: 14/05/2024

Place: Goa University

COMPLETION CERTIFICATE

This is to certify that the dissertation report **Detection of Sugar Syrup Adulteration in Honey Using Near Infrared Spectroscopy** is a bonafide work carried out by Ms. Viola Fernandes under my supervision in partial fulfilment of the requirements for the award of the degree of Masters in the Discipline Electronics at the School of Physical and Applied Sciences, Goa University.

Dr. Marlon Sequiera

Electronics Discipline

School of Physical and Applied Sciences

Date: 13/05/2024

Prof Ramesh V. Pai

School of Physical and Applied Sciences

Date:

Place: Goa University

School/Dept Stamp

Contents

Preface....6

Acknowledgements....7

Tables and figures....8

Abbreviations Used....9

Abstract....10

Chapter 1: Introduction....11

1.1: Background....*14*

1.2: Aim and Objectives....*15*

1.3: Hypotheses....*16*

1.4: Scope....*16*

Chapter 2: Literature Review....19

Chapter 3: Methodology....35

3.1: Samples....*35*

3.2: Spectra Acquisition....*37*

3.2.1: Operating Principle of Spectrophotometer: Beer Lambert Law....38

3.3: ML Regressors:....*41*

3.3.1: Support Vector Regressor....41

3.3.2: Partial Least Square Regressor....43

3.3.3: Decision Tree Regressor....45

3.3.4: Random Forest Regressor....48

3.3.5: K Nearest Neighbor Regressor....51

3.4: Pre-Processing....*53*

3.5: Principal Component Analysis....*55*

3.6: Data Partitioning *56*

Chapter 4: Analysis and Conclusions 57

4.1: Performance Analysis *57*

4.2: ***Conclusion*** *62*

References 63

Appendix 71

Preface

This dissertation represents the culmination of several months of research and hard work. It is with great pleasure that I present it to the academic community.

Throughout this journey, I have been fortunate to receive support and guidance from many individuals and the institution, to whom I owe immense gratitude.

The dissertation is organized as follows: Chapter 1 provides an introduction to the topic, including its significance and relevance. Chapter 2 reviews the existing literature on Honey adulteration and it's detection using various methods, synthesizing key findings and identifying gaps in the literature. Chapter 3 outlines the methodology employed in this study, detailing the research design, data collection methods, and analytical approach. Finally, Chapter 4 presents the empirical findings and analysis.

Acknowledgements

This dissertation would not have been completed without the help and guidance of several individuals who in one way or the other contributed and extended their valuable assistance in the preparation and completion of this study, it is a pleasure to thank those who made it possible.

I would like to express my gratitude to the faculty of the Electronics Department of Goa University for giving me this opportunity to learn throughout this process of doing my project.

The completion of this project could not have been possible without the expertise of Dr. Marlon Sequiera, our project guide.

I would like to thank the teaching faculty, Prof. Rajendra S. Gad, Dr. Jivan S. Parab, Dr. Narayan Vetrekar and Dr. Aniketh Gaonkar for their support throughout the completion of this project.

I would also like to thank Mr. Vishant Malik (Lab Technician) for his assistance and provision of necessary components.

A sincere thank you to Mrs. Ashwini and Sir Ramchandra for their help and support.

Last but not the least, I would like to thank my friends and family for always extending their helping hand and support.

Tables and figures

Figure 1: Flow diagram.....15

Figure 2: Weighed honey sample.....36

Figure 3: Samples 36

Figure 4: Sample in a quartz cuvette.....37

Figure 5: Interaction of Light with the Sample.....38

Figure 6: Block Diagram of Spectrophotometer.....40

Figure 7: Smoothened Spectra 54

Figure 8: PLS Regressor 59

Figure 9: SVR 59

Figure 10: DTR 60

Figure 11: RFR.....60

Figure 12: KNR.....61

Table 1: Summarised Literature.....25

Table 2: Parameters.....52

Table 3: Summarised Performance of ML Regressors 58

Abbreviations Used

NIR	Near Infrared
NIRS	Near Infrared Spectroscopy
SVR	Support Vector Regression
KNR	K Nearest Regression
PLSR	Partial Least Square Regression
PCA	Principal Component Analysis
SG	Savitsky Golay
RFR	Random Forest Regression
DTR	Decision Tree Regression
HFCS	High Fructose Corn Syrup
ML	Machine Learning
LDA	Linear discriminant Analysis

Detection of Invert Sugar Syrup Adulteration in Honey Using Near Infrared Spectroscopy

Viola Fernandes

MAY, 2024

Abstract

This work has developed a method for detection of Invert Sugar Syrup Adulteration in Honey using Near Infrared Spectroscopy with the side of machine learning algorithms. A comparative study is done on the efficacy of the different machine learning model to detect the adulteration. Honey adulteration is the deceptive practice of diluting or altering honey with other substances to increase volume or improve appearance while lowering production costs. Common adulterants include high-fructose corn syrup, sugar syrups, and water. Adulteration not only compromises the quality and flavour of honey but also poses health risks, as some adulterants may contain harmful substances or allergens. Detection of adulteration can be challenging, as sophisticated methods are often employed to mimic the chemical composition of pure honey. The achieved RMSE are 8.72, 2.54, 0.23, 0.33 and 0.92 for SVR, PLSR, DTR, RFR and KNR respectively.

Chapter 1: Introduction

In the food industry, there is a high chance that many foods and food items are adulterated. Expensive items (such extra virgin olive oil and vanilla) and those whose composition or production can change due to weather variations during the growing and harvest seasons (such as oranges and coffee), may be especially vulnerable to this method. [1]

A dishonest technique that is common in the food sector is honey adulteration, which is reducing the purity and nutritional value of honey by adding less expensive ingredients like corn syrup, sugar syrup, or water. In addition to lacking the healthy enzymes, antioxidants, and antibacterial qualities present in pure honey, adulterated honey also could have dangerous impurities. Since dishonest producers use advanced techniques, detecting adulteration is still difficult even with regulatory controls and quality control initiatives in place. To protect consumer health and preserve the reputation of this natural sweetener, it is crucial to guarantee the quality and traceability of honey from hive to table. Honey has been used in various fields such as healthcare and food industry due to its value and versatility as an ingredient. It is a supersaturated solution or semi-solid natural product synthesized from flower nectar by honey bees. Honey's chemical composition can vary depending on factors like floral source, region, and processing methods, but generally, it primarily consists of:

1. Carbohydrates: The main carbohydrate in honey is glucose (approximately 31-44%) and fructose (approximately 25-40%). These sugars give honey its sweet taste and provide energy.

2. Water: Honey contains varying amounts of water, typically ranging from 17-20%. The moisture content can affect the honey's shelf life and consistency.

3. Proteins: Honey contains small amounts of proteins, amino acids, and enzymes. These proteins and enzymes contribute to its nutritional value and can have various health benefits.

4. Minerals: Honey contains trace amounts of minerals such as potassium, calcium, magnesium, sodium, phosphorus, and small quantities of other minerals. The mineral content depends on the floral source and soil composition.

5. Vitamins: Honey contains small amounts of vitamins, including B vitamins (such as B6, niacin, riboflavin, and pantothenic acid) and vitamin C.

6. Acids: Honey contains several organic acids, including gluconic acid, citric acid, and acetic acid. These acids contribute to honey's flavour and help inhibit the growth of microorganisms.

7. Antioxidants: Honey contains various antioxidants, including flavonoids and phenolic compounds, which may help protect cells from damage caused by free radicals.

8. Other compounds: Honey may contain traces of other compounds, such as pollen grains, propolis, and bee-derived enzymes, depending on its source and processing.

The specific composition of honey can vary significantly depending on factors such as floral source, geographical location, and processing methods. This variability gives different types of honey their unique flavours, colours, and properties.

Since honey is vulnerable to adulteration or unethical mixing with cheaper or low-grade honey, sugars and other substances due to its recognition as a high-quality food product. There is a rising demand due to the population’s increased health concerns since it is proved to have healing properties. This unethical behaviour not only compromises the authenticity of the product but also poses serious health risks to consumers, especially those with allergies or sensitivities to certain ingredients.

There are several reasons why honey gets adulterated, including adding sugars to suit consumer preferences or combining inexpensive, low-quality honey with more costly honey to maximize yield.

Traditionally, honey has been used to heal bronchial phlegm, boost immunity, destroy bacteria, relieve colds and sore throats, and boost immunity.

Furthermore, research indicates that honey possesses a number of pharmacological qualities, including anti-inflammatory, antioxidant, and anti-cancer effects against osteosarcoma, prostate cancer, breast, and cervical cancer. Honey has the potential to improve human health through topical application or oral consumption. According to references, honey can be taken orally to cure a variety of conditions, including laryngitis, osteoporosis, gastrointestinal ulcers, anorexia, sleeplessness, constipation, and issues with the liver, heart, and digestive systems.

On the other hand, advantages of topical application of honey are prescribed for eczema, lip sores, sterile and infected wounds, genital lesions, burns, surgery scars, and athlete's foot. [2]

Honey is adulterated with various sugar substances such as corn syrup, sugar syrup, maple syrup, invert sugar syrup. Invert sugar is prepared by adding an acid to plain sugar syrup. Basically, table sugar is disaccharide. It's made of 2 sugar molecules called glucose and fructose. These two, bond together and to break up the bond between them a hydrolysis process is used (a chemical reaction where water is used). Addition of acid helps to split sucrose (table sugar) into glucose and fructose, thus converting it into invert sugar.

1.1: Background

Adulteration of honey is mainly done to increase quantity and sweetness of already available pure honey. After mixing it with invert sugar syrup, the sweetness of honey increases since it contains more amount of glucose and fructose instead of sucrose in a free state.
It can be difficult to determine whether honey has been tampered with because some adulterants are difficult to identify with the unaided eye. Producers of honey, regulators, and beekeepers have evaluated the authenticity and purity of honey using a variety of traditional techniques, including the Water Dissolution Test, Paper Towel Test, Flame Test, Density Test, and Crystallisation Pattern. A number of common laboratory assays identify honey adulteration. These traditional techniques can provide early indicators of potential adulteration, but they also call for more sophisticated laboratory oratory testing. Furthermore, the spectrum of particular pollutants that some advanced tests are capable of detecting may be limited, challenging because of their complexity, and requiring specialized knowledge. [3] To combat adulteration, regulatory bodies also establish quality standards, implement testing protocols, and enforce penalties for offenders. Additionally, consumer awareness and support for certified honey products can incentivize producers to maintain integrity in their practices.

Research shows that Near Infrared Spectroscopy (NIRS) shows tremendous promise for determining the chemical constituents and internal characteristics of food products. As far as food science is concerned, the use of the NIR region is of significant importance. NIRS can be employed for determining adulteration in honey as well. It is non-destructive and highly sensitive to chemical constituents present in any substance. NIRS uses the range from 780 to 2500 nm and is useful in the identification of a variety of adulteration problems using experimental and statistical methods.

1.2: Aim and Objectives

Honey is characterized as a natural and raw food that can be used not only as a sweetener but also as a medicine due to its therapeutic effect on human health.

The aim of this study is to determine the percentage of adulteration in honey using NIR Spectroscopy. The first step would be to adulterate honey using partially invert sugar syrup and mix the two solutions. Thereafter keeping the sample in a water bath and then sonicating the sample before spectra acquisition and further processing.

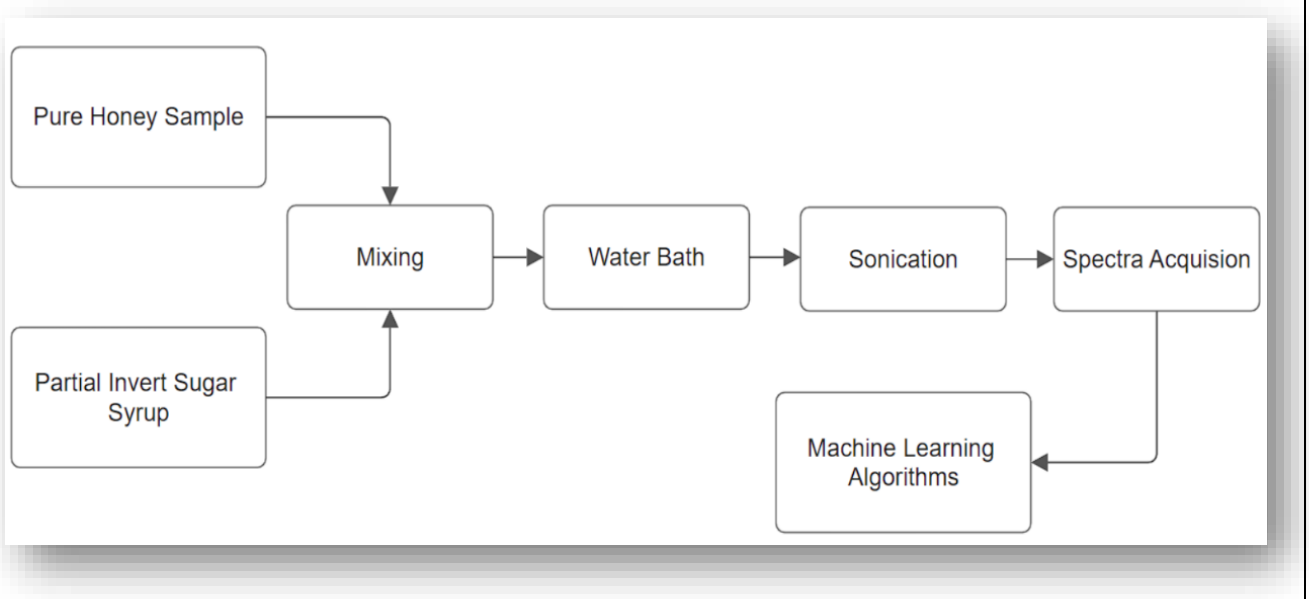

Figure 1: Flow diagram

1.3: Hypotheses

The invert sugar mixed in pure honey absorbs certain wavelengths of the NIR region of light. Aim is to isolate bands in the region that give the signature of the present adulterant. When light is incident to the sample, the sugars in the sample absorb certain wavelengths of that light thus showing high absorbance in the NIR region at certain wavelengths.

In this study, pure honey is adulterated with partial invert sugar syrup. Since we know that sugar syrup is produced by hydrolysis i.e. sugar breaking down into glucose and fructose thus having high individual absorbance.

1.4: Scope

Near-infrared spectroscopy (NIR) offers a promising scope for the detection of adulteration in honey due to its rapid, non-destructive, and cost-effective nature. NIR spectroscopy involves the measurement of the absorption of near-infrared light by organic molecules in the sample, providing information about its chemical composition. Here's the scope of using NIR spectroscopy for detecting adulteration in honey:

1. Chemical Composition Analysis: NIR spectroscopy can be used to analyse the chemical composition of honey, including sugar profiles, moisture content, and presence of additives or adulterants. By comparing the NIR spectra of authentic honey samples with those of adulterated or impure samples, deviations in the spectral patterns can be identified, indicating potential adulteration.

2. Quantitative Analysis: NIR spectroscopy allows for quantitative analysis of key parameters in honey, such as glucose, fructose, sucrose, and moisture content. Calibration models can be

developed using chemometric techniques to correlate NIR spectral data with reference analytical measurements, enabling accurate prediction of these parameters in unknown samples and detection of adulteration beyond threshold levels.

3. Isotopic and Geographical Origin Analysis: NIR spectroscopy coupled with chemometric modelling can be employed to analyse isotopic ratios and geographical origin markers in honey samples. By correlating NIR spectral data with reference isotopic databases or geographic information, authenticity verification and detection of adulteration based on origin can be achieved.

4. High Throughput Screening: NIR spectroscopy enables high throughput screening of large numbers of honey samples within a short time frame, making it suitable for routine quality control and surveillance purposes. Automated NIR instruments can rapidly analyse samples in batch mode, facilitating efficient detection of adulteration in honey production and distribution chains.

5. Non-Destructive Analysis: NIR spectroscopy is a non-destructive technique that requires minimal sample preparation, allowing honey samples to be analysed without altering their physical or chemical properties. This non-destructive nature makes NIR spectroscopy particularly suitable for in-line or at-line monitoring of honey processing operations and quality assessment during packaging and distribution.

6. Portable and Field Deployable: NIR spectroscopy instruments are available in portable and handheld configurations, offering the flexibility to perform on-site analysis of honey samples in field settings, such as apiaries, honey processing facilities, or customs checkpoints. This portability facilitates rapid screening of honey products and immediate decision-making regarding their authenticity and quality.

Overall, the scope of using NIR spectroscopy for detecting adulteration in honey lies in its ability to provide rapid, reliable, and cost-effective analysis of multiple quality parameters, enabling early detection and prevention of adulterated honey products from entering the market. Continued research and development in instrument technology, data analysis methods, and calibration strategies can further enhance the effectiveness and applicability of NIR spectroscopy for ensuring the integrity and authenticity of honey.

Chapter 2: Literature Review

In 1992, S. Kawano et. al. proposed an idea to develop a non-destructive sugar analyser for intact peaches, the potential of the near infrared (NIR) method with fibre optics in interactance mode was evaluated. In 'Shimizu Hakuto', a white peach cultivar, there were clear differences among the NIR second derivative spectra of high, medium, and low Brix values in the wavelength region around 906 nm. The author used linear regression and obtained a multiple correlation coefficient of 0.97 [4].

In 1993, T. Shilna proposed a study for determination of brix and acidity values of pineapple using near infrared spectroscopy. Spectra in the wavelength range of 700-2500 nm was obtained. Multiple regression analysis was used by the author and a multiple correlation coefficient of 0.88 was obtained[5].

In 2004, J. F. D. Kelly et. al. proposed a paper based on the initial study of Honey Adulteration by Sugar Solutions Using Mid infrared (MIR) Spectroscopy and Chemometrics. The author concluded that samples that were adulterated at levels more than 14% were more accurately classified. The authors used PLSR along with KNN and obtained an accuracy of 93% [6].

In 2006, J. D. Kelly et. al. implemented the application of Fourier Transform Mid Infrared Spectroscopy for the differentiation between Irish artisanal honey and honey adulterated with various sugar syrups. The authors used authentic honey and honey adulterated by beet sucrose, dextrose syrups, and partial invert corn syrup. The author used PLSR and obtained classification rates of 96.2% (pure honey), 97.5% (Beet sucrose),95.8% (dextrose syrups), and 91.7% (partial invert corn syrup) [7].

In 2008, M. M. Ali et. al. evaluated the potential application of near infrared (NIR) spectroscopy in the range of wavelength from 1000 to 2500 nm to non-destructively determine total soluble solids (Brix) and pH values of bananas. Their findings suggested that NIR

spectroscopy is applicable to predict sugar content in bananas non-destructively. The author used PLS and determined the coefficient of determination to be 0.81[8].

In 2020, M. E. Bahrami et. al. proposed a paper highlighting the unique application of near-infrared spectroscopy to prediction of sugar beet juice quality, and their results showed that NIR spectroscopy was capable of predicting the pol, sucrose concentration, °brix, and pH value (but slightly) of the juices in a wide field of concentrations leading to high R2 (0.984) and low error values[9].

L. Chen et al., 2011 employed NIR combined with discriminant partial least squares for identification of unadulterated and adulterated honey with HFCS. 46 adulterated honey samples and unadulterated honey were correctly classified and correct classification was achieved. It was observed that for adulterated honey better correct classification rates were achieved than for unadulterated honey[10].

In 2022, M. Benković et al. concluded that NIR coupled with ANN modelling can be considered an efficient tool for honey adulteration quantification. PLS models were created to predict the number of adulterants added, moisture content, and conductivity. The author achieved an R square value greater than 0.86 using PLS[11].

2019, M. J. A. González et al. employed (Vis-NIRS) spectroscopy combined with chemometrics as a screening technique for the identification and quantification of different adulterants (inverted sugar, rice syrup, brown cane sugar and fructose syrup) added to high-quality honey. The author achieved a complete discrimination between non-adulterated and adulterated samples. Honey samples were analysed in the range of 400–2500 nm with a spectral resolution of 0.5 nm. The author used Linear Discriminant Analysis and obtained R2 greater than 0.98[12].

In 2023, S. Ropciuc et. al. used Emission Excitation spectra combined with chemometrics to detect adulteration and authenticity of honey and did a comparative study of SVM and PLS-DA where SVM proved to be more accurate[13].

In 2020, X. Yang et. al., proposed a study suggesting the detection of syrup-blended adulteration of Manuka honey NIR spectroscopy combined with aquaphotomics. Manuka honey was adulterated with five different syrups (corn syrup, sucrose syrup, high fructose corn syrup, beet syrup, and rice syrup) in degrees from 10% to 50% for NIR spectra collection. The author used PLSR and PCA and predicted a constant of 0.9891[14].

In 2022, Raypah, Muna E. et al., suggested the application of NIRS combined with Aquaphotomics for detection of adulteration of Stingless Bee Honey. The author performed PCA analysis and PLSR obtaining an R2 value of 0.96[15].

In 2010, X. Zhu et. al., suggested the application of NIR spectroscopy combined with chemometrics to detect adulteration of honey samples. Sample set containing 135 spectra of authentic and adulterated honey samples. Spectral data were compressed using wavelet transformation (WT) and principal component analysis (PCA). The author concluded that WT proved more effective than PCA, as a means for variable selection. The author achieved an accuracy of 95.1%[16].

D. Valinger et. al. proposed the application of UV-VIS and NIR spectroscopy coupled with multivariate analysis for detection of honey adulteration. The author aimed to develop PLS and ANNs models for quantification of adulteration, as well as to describe physical and chemical properties of pure honey samples and prepared adulterations based on non-pre-processed UV-VIS and NIR spectra. The authors used PCA and PLSR and obtained a coefficient of correlation of 0.8[17].

In 2016, G. Bazar et. al. proposed the revelation of difference in water spectral pattern by detection of honey adulteration using NIR spectroscopy. The author applied PCR and PLSR and achieved an R2 of 0.91[18].

In 2017, A. Guelpa et. al. employed NIR Spectroscopy for verification of authenticity and fraud detection in South African Honey. Partial Least Square Discriminant Analysis was used by the author and an accuracy between 93.3% and 99.9% was achieved[19].

In 2010, S. Mishra et. al. employed near-infrared spectroscopy for detection of adulteration of jaggery syrup in honey. The authors used PLS and obtained an R2 of 0.81[20].

In 2002, S. Sivakesava et. al. successfully performed the classification of simple and complex sugar adulterants in honey using mid-infrared spectroscopy. The authors achieved optimum classification of 100% using linear discriminant analysis for honey samples adulterated with glucose, sucrose and fructose. Linear Discriminant analysis successfully classified beet and cane invert sugars in a single variety of honey (100% of the validation set) using PLS data compression with two PLS factors[21].

In 2021, A. Rust et. al. tested the application of ANOVA-simultaneous component analysis for quantification and characterization of effects of age, temperature, syrup adulteration and irradiation on near-infrared (NIR) spectral data of honey. The author used ANOVA Simultaneous Component Analysis[22].

In 2020, F. Huang et. al. employed NIR and ATR-FTIR spectral data fusion for detection of adulteration in Chinese honey. The authors used PCA with SVM and achieved 100% accuracy [23].

In 2012, S. Li et. al. successfully performed the detection of adulteration in honey by high fructose corn syrup and maltose syrup using Raman spectroscopy. The authors used a PCA and Linear Discriminant Analysis (LDA) model[24].

In 2018, Z. Bodor et.al. employed the application of Near Infrared Spectroscopy and Classical Analytical Methods for the Evaluation of Hungarian Honey[25].

In 2011, Z. Tu et. al. employed NIR Spectroscopy for detection of honey adulteration of honey. PCA with SVM was employed by the author and an accuracy of 100% was achieved. [26].

In 2006, J. D. Kelly et. al. evaluated the potential of near infrared transflectance spectroscopy for detection of adulteration of Irish honey by beet invert syrup and high fructose corn syrup. The authors used PLS and obtained a Correlation Coefficient between 0.72 to 0.79[27].

In 2022, W. Venesia et. al. employed NIRS for detection of adulteration of Kelukut honey with glucose syrup. A comparative study between Random Forest (RF) and K-Nearest Neighbour (KNN) was done by the author and the author achieved an accuracy of 90.2% for RF and 87% for KNN[28].

In 2022, W. Venesia et. al. employed NIRS for detection of adulteration of Kelukut honey with glucose syrup. A comparative study of KNN and Random Forest was done by the authors and an accuracy of 90.2% and 87% respectively was obtained[29].

Food analysis by portable NIR spectrometer by G. S. Folli consisted of a sub study of detection of adulteration in honey[30].

In 2023, D. D. Papazoglou et. al. suggested the application of UV–Vis spectroscopy for detecting adulteration in Mediterranean honeys. Comparative study of RF, PLS-DA and DD-SIMCA and RF proved to be more accurate giving an accuracy of 95% [31].

In 2015, Z. Gan et. al. used sensor and spectral analysis to classify botanical origin and determine adulteration of raw honey. PLSDA gave 96% accuracy[32].

In 2009, T. Gallardo et. al. tested the application of FTIR-HATR and multivariate analysis for quantification of adulterants in Mexican honey. The authors used PCA and achieved an R2 in the range of 0.97-0.99[33].

In 2022, W. Skaff et. al. successfully performed the detection of adulteration of honey using NIR Spectroscopy and Chemometrics. The authors also put light on the effects of honey adulteration on human health[34].

In 2016, M. Bougrini et. al. successfully performed the classification of honey according to geographical and botanical origins and also performed detection of adulteration using Voltammetric Electronic Tongue. The author employed PCA, SVM and Hierarchical Clustering and achieved accuracies between 73.41 to 100%[35].

In 2006, S. Sivakesava et. al. proposed a paper performing a rapid spectroscopic technique to determine adulteration of honey with corn syrup. The author suggested the application of PLSR and obtained R^2 values of 0.996, 0.95, and 0.884 for glucose, fructose, and sucrose respectively[36].

Table 1: Summarised Literature

Sr. No.	Paper	Algorithm/ Method	Accuracy/R2
1.	In 1992, S. Kawano et. al. proposed an idea to develop a nondestructive sugar analyzer for intact peaches, the potential of the near infrared (NIR) method with fiber optics in interactance mode was evaluated.	Linear Regression	Multiple Correlation Coefficient = 0.97
2.	In 1993, T. Shilna proposed a study for determination of brix and acidity values of pineapple using near infrared spectroscopy.	Multiple Regression analysis	Multiple Correlation Coefficient = 0.88
3.	In 2004, J. F. D. Kelly et. al. proposed a paper based on the initial study of Honey Adulteration by Sugar Solutions Using Mid infrared (MIR) Spectroscopy and Chemometrics.	PLSR and KNN	93%
4.	In 2006, J. D. Kelly et. al. implemented the application of Fourier Transform Mid Infrared Spectroscopy for the differentiation between Irish	PLSR	Classification rates: 96.2% (pure honey), 97.5%(Beet sucrose),95.8%(dextrose syrups), and 91.7%(

	artisanal honey and honey adulterated with various sugar syrups.		partial invert corn syrup)
5.	In 2008, M. M. Ali et. al. evaluated the potential application of near infrared (NIR) spectroscopy in the range of wavelength from 1000 to 2500 nm to non-destructively determine total soluble solids (Brix) and pH values of bananas.	PLS	Coefficient of Determination =0.81
6.	In 2020, M. E. Bahrami et. al. proposed a paper highlighting the unique application of near-infrared spectroscopy to predict sugar beet juice quality and their results showed that NIR spectroscopy was capable of predicting the pol, sucrose concentration, °brix, and pH value (but slightly) of the juices in a wide field of concentrations leading to high R^2 and low error values.	PLS	R^2= 0.984
7.	L. Chen et al., 2011 employed NIR combined with discriminant partial least squares for identification of	PLS	97.9% adulterated and 95.8% unadulterated.

	unadulterated and adulterated honey with HFCS. 46 adulterated honey samples and unadulterated honey were correctly classified and correct classification was achieved.		
8.	In 2022, M. Benković et al. concluded that NIR coupled with ANN modelling can be considered an efficient tool for honey adulteration quantification.	PLS	R^2= < 0.86
9.	2019, M. J. A. González et al. employed (Vis-NIRS) spectroscopy combined with chemometrics as a screening technique for the identification and quantification of different adulterants (inverted sugar, rice syrup, brown cane sugar and fructose syrup) added to high-quality honey.	Linear Discriminant Analysis	R > 0.98–
10.	In 2023, S. Ropciuc et. al. used Emission Excitation spectra combined with chemometrics to detect adulteration and authenticity of honey	Comparative study of SVM and PLS-DA	SVM proved to be more accurate.

11.	In 2020, X. Yang et. al., proposed a study suggesting the detection of syrup-blended adulteration of Manuka honey NIR spectroscopy combined with aquaphotomics.	PCA analysis, and partial least square regression (PLSR)	$R_{Predicted}$ =0.9891
12.	In 2022, Raypah, Muna E. et al., suggested the application of NIRS combined with Aquaphotomics for detection of adulteration of Stingless Bee Honey.	PCA analysis, and partial least square regression (PLSR)	R^2= 0.96
13.	In 2010, X. Zhu et. al., suggested the application of NIR spectroscopy combined with chemometrics to detect adulteration of honey samples. Sample set containing 135 spectra of authentic and adulterated honey samples. Spectral data were compressed using wavelet transformation (WT) and principal component analysis (PCA).	WT-LS-SVMzs	95.1%
14.	D. Valinger et. al. proposed the application of UV-VIS and NIR spectroscopy coupled with	PCA analysis, and partial least square regression (PLSR)	Coefficient of Correlation = 0.8

	multivariate analysis for detection of honey adulteration.		
15.	In 2016, G. Bazar et. al. proposed the revelation of difference in water spectral pattern by detection of honey adulteration using NIR spectroscopy.	principal component regression (PCR) and partial least squares regression (PLSR)	R^2= 0.91
16.	In 2017, A. Guelpa et. al. employed NIR Spectroscopy for verification of authenticity and fraud detection in South African Honey.	Partial least squares discriminant analysis (PLS-DA)	Between 93.3% and 99.9%
17.	In 2010, S. Mishra et. al. employed near-infrared spectroscopy for detection of adulteration of jaggery syrup in honey.	PLS	R^2=0.81
18.	In 2002, S. Sivakesava et. al. successfully performed the classification of simple and complex sugar adulterants in honey using mid-infrared spectroscopy.	PLS	100%

19.	In 2021, A. Rust et. al. tested the application of ANOVA-simultaneous component analysis for quantification and characterization of effects of age, temperature, syrup adulteration and irradiation on near-infrared (NIR) spectral data of honey.	ANOVA-simultaneous component analysis (ASCA)	
20.	In 2020, F. Huang et. al. employed NIR and ATR-FTIR spectral data fusion for detection of adulteration in Chinese honey.	PCA with SVM	100%
21.	In 2012, S. Li et. al. successfully performed the detection of adulteration in honey by high fructose corn syrup and maltose syrup using Raman spectroscopy.	PLS-LDA	84.4%
22.	In 2018, Z. Bodor et.al. employed the application of Near Infrared Spectroscopy and Classical Analytical Methods for the Evaluation of Hungarian Honey.	Principal component analysis (PCA) and linear discriminant analysis (LDA) model.	81%, 75%

23.	In 2011, Z. Tu et. al. employed NIR Spectroscopy for detection of honey adulteration of honey.	PCA with SVM	100%
24.	In 2006, J. D. Kelly et. al. evaluated the potential of near infrared transflectance spectroscopy for detection of adulteration of Irish honey by beet invert syrup and high fructose corn syrup.	PLS	Correlation coefficient = 0.72 to 0.79
25.	In 2022, W. Venesia et. al. employed NIRS for detection of adulteration of Kelukut honey with glucose syrup.	Comparative study of Random Forest (RF) and KNN.	R_F= 90.2% and KNN= 87%
26.	In 2017, M. Oroian et al. employed Raman spectroscopy for detection of adulteration of honey.	PLS-PCR	90%
27.	Food analysis by portable NIR spectrometer by G. S. Folli consisted of a sub study of detection of adulteration in honey.	SVM	0.83-0.97
28.	In 2023, D. D. Papazoglou et. al. suggested the application of UV–Vis	Comparative study of RF,	R_F= 95%, PLS-DA=8%, DD-SIMCA=93%

	spectroscopy for detecting adulteration in Mediterranean honeys.	PLS-DA, and DD-SIMCA models achieved an accuracy of 95, 88, and 93%	
29.	In 2015, Z. Gan et. al. used sensor and spectral analysis to classify botanical origin and determine adulteration of raw honey.	PLSDA	96%
30.	In 2009, T. Gallardo et. al. tested the application of FTIR-HATR and multivariate analysis for quantification of adulterants in Mexican honey.	PCA	R^2 in the range of 0.97–0.99
31.	In 2022, W. Skaff et. al. successfully performed the detection of adulteration of honey using NIR Spectroscopy and Chemometrics.	PCA	
32.	In 2016, M. Bougrini et. al. successfully performed the classification of honey according to geographical and botanical origins and also performed detection of	PCA, SVM, HCA (Hierarchical Clustering)	73.41% to 100%

	adulteration using Voltammetric Electronic Tongue.		
33.	In 2006, S. Sivakesava et. al. proposed a paper performing a rapid spectroscopic technique to determine adulteration of honey with corn syrup.	PLSR	R^2 values of 0.996, 0.95, and 0.884 for glucose, fructose, and sucrose respectively.
34.	Adulteration of honey and available methods for detection – a review by Zábrodská, B., Vorlová, L. [37]		–
35.	In 1979, I. Kushnir proposed a paper titled Sensitive thin layer chromatographic detection of high fructose corn syrup and other adulterants in honey. [38]	Chromatography	–
36.	In 2016, S. Shafiee et. al. investigated the application of hyperspectral imaging system and data mining-based classifiers for honey adulteration detection. [39]	Hyperspectral Imaging	–
37.	In 2017, S. Amiry et. al. performed multivariate analysis to classify adulterated honey samples. [40]	Multivariate Analysis	

38.	In , D. Bertelli et. al. proposed a paper suggesting the detection of adulteration of honey by sugar syrup using one dimensional and two-dimensional high resolution Nuclear Magnetic Resonance. [41]	Nuclear Magnetic Resonance	
39.	In 2015, U. Kamboj et. al. proposed a paper suggesting the prediction of adulteration in honey using rheological parameters. [42]	Rheology	–
40.	Non-linear Correlation of Absorbance with Respect to Concentration of Sugar in Aqueous Solutions of High Purity Laboratory Chemical Reagent Sucrose and Ordinary Cane Sugar by Indimuli J. M et. al. [43]	HPLC	–
41.	Physicochemical Analysis of Several Natural Malaysian Honeys and Adulterated Honey by Puteri Nurul Syahirah Md Dan. [44]	Physicochemical analysis	–
42.	The Sugars of Honey - A Review by Doner, L.W. [45]	–	–

Chapter 3: Methodology

3.1: Samples

Authentic forest honey sample was procured from a local honey vendor and the sample was stored at room temperature prior to the adulteration and spectra acquisition process.

The invert sugar syrup sample was purchased from a local supermarket. Invert sugar is produced by hydrolysis of table sugar. Table sugar is essentially a disaccharide. It is composed of two sugar molecules, fructose and glucose. These two form a bond. And a hydrolysis—a chemical reaction in which water is used—is employed to sever the link between them. Acidic addition facilitates the breakdown of sucrose, or table sugar, into glucose and fructose. Invert sugar is said to increase sweetness and help preserve and increase the moisture.

The honey sample was adulterated with partial invert sugar syrup in the following concentrations: 0.5%, 1%, 1.5%, 2%, 2.5%, 3%, 3.5%, 4%, 4.5%, 5%, 8%, 10%, 15%, 20%, 25%, 30% and 40% (%w/w).

The pure honey sample was weighed using an analytical balance (a constant of 20g of sample was used for spectra recording of each concentration). The samples were stirred for 5 mins using a stirrer and then the samples were kept in a water bath of 40 degrees Celsius for 20 mins. Thereafter, the samples were kept at room temperature for 48hrs to get rid of the air bubbles. Later the samples were then sonicated for 20 mins to ensure homogeneity. This standard method was employed according to reviewed literature.

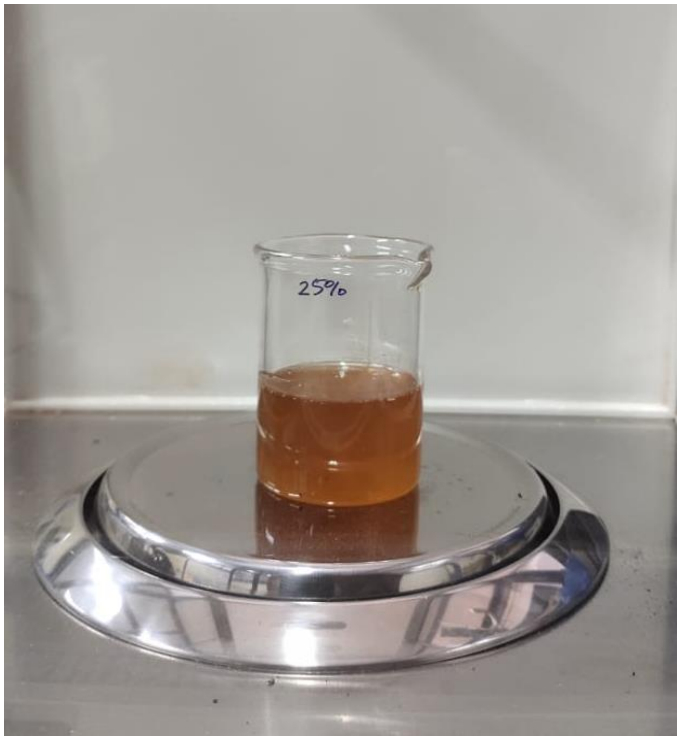

Figure 2: Weighed honey sample

Figure 3: Samples

3.2: Spectra Acquisition

The Jasco V-770 Spectrophotometer was used to acquire spectra of all samples. The samples were placed in a quartz cuvette of with pathlength 10mm, Volume 3.5ml, and with transmission range 190-2500 nm and then the cuvette was kept in the sample holder. The Near Infrared Spectra (NIR) of the honey samples were acquired in transmission mode. The NIR bandwidth and range was set to 20 nm and 550 nm to 1400 nm. Every sample was measured in parts of 2, where each part was measured 4 times which made up to 8 spectra per concentration and 144 spectra in total.

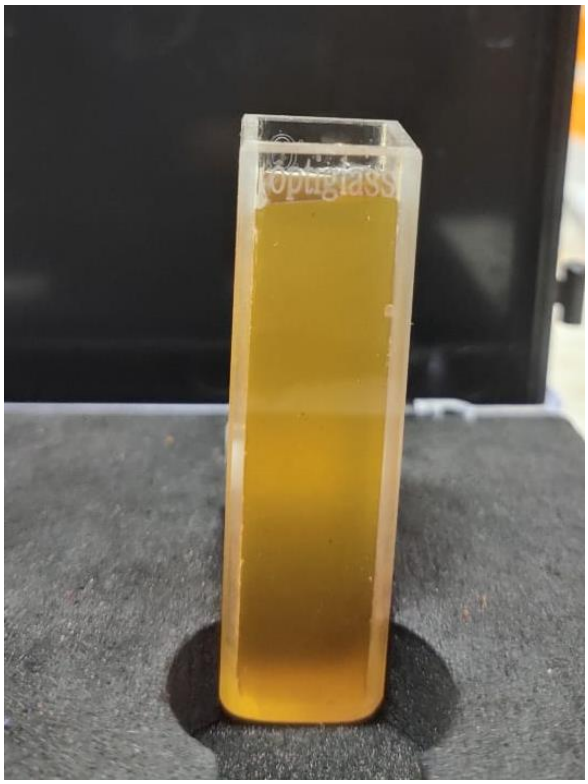

Figure 4: Sample in a quartz cuvette

3.2.1: Operating Principle of Spectrophotometer: Beer Lambert Law

Spectroscopy is based on the interaction between light and matter. When molecules absorb IR radiation, transitions occur from a ground vibrational state to an excited vibrational state. If the frequency of the light matches the frequency of the vibration of the bonds in the molecule, the molecule absorbs some of the light.

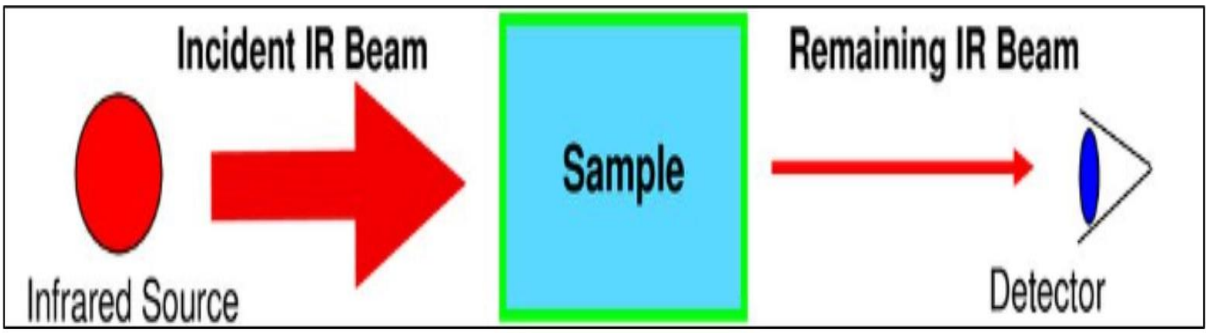

Figure 5: Interaction of Light with the Sample

For each wavelength of light passing through the spectrometer, the intensity of the light passing through the reference cell is measured. This is usually referred to as Io - that's I for Intensity.

The intensity of the light passing through the sample cell is also measured for that wavelength - given the symbol, I. If I is less than Io, then the sample has absorbed some of the light (neglecting reflection of light off the cuvette surface). A simple bit of math is then done in the computer to convert this into something called the absorbance of the sample - given the symbol, A. The absorbance of a transition depends on two external assumptions.

1. The absorbance is directly proportional to the concentration (c) of the solution of the sample used in the experiment.
2. The absorbance is directly proportional to the length of the light path (l), which is equal to the width of the cuvette.

The equation can be written as:

$$A \propto cl$$

This proportionality can be converted into an equality by including a proportionality constant (ϵ).

$$A = \epsilon cl$$

Where A is the value of absorbance, ϵ is the Molar absorbance coefficient, c is molar concentration and l is optical path length. [43]

This equation can be defined via the incident intensity Io and transmitted intensity I by

$$A = log10\left(\frac{Io}{I}\right)$$

The equations can also be written in terms of transmittance:

$$T = \frac{I}{Io}$$

However, it is more commonly expressed as a percentage transmittance:

$$T(\%) = \frac{100I}{Io}$$

The absorbance, *A*, of the solution is related to the transmittance and incident and transmitted intensities through the following relations:

$$A = log\frac{I}{Io}$$

$$\mathrm{A} = -\log \mathrm{T}$$

The absorbance has a logarithmic relationship to the transmittance; with an absorbance of 0 corresponding to a transmittance of 100% and an absorbance of 1 corresponding to 10% transmittance. Additional values of transmittance and absorbance pairings are given in Table 1. A visual demonstration of the effect that the absorbance of a solution has on the attenuation light passing through it is shown Figure 2, where a 510 nm laser is passed through three solutions of Rhodamine 6G with different absorbance. [44]

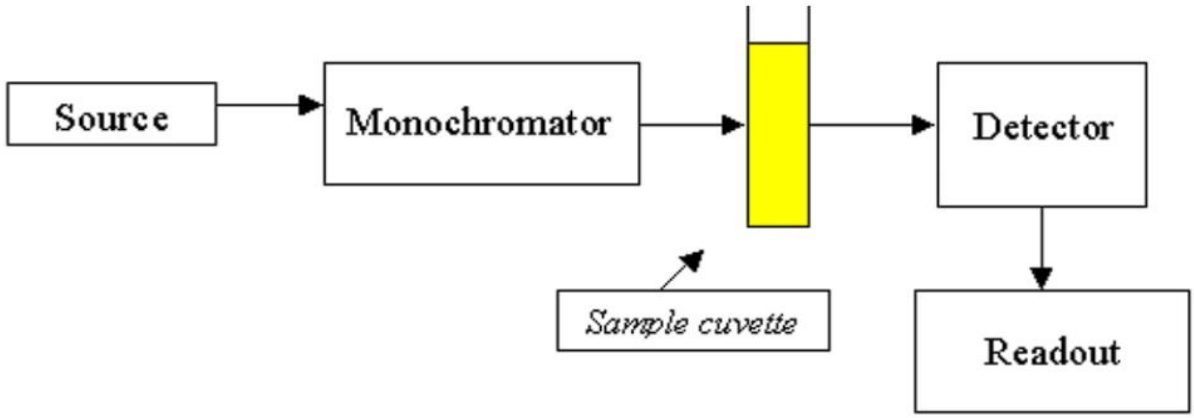

Figure 6: Block Diagram of Spectrophotometer

The block diagram (figure 6) of Spectrophotometer shows a light source that emits a broad spectrum of light covering the desired wavelength range. Common light sources include tungsten-halogen lamps, deuterium lamps (for the UV range), and xenon lamps (for the visible and near-infrared range), a monochromator which is a crucial component that separates the incoming polychromatic light into its constituent wavelengths, a sample cuvette for the sample and a detector measures the intensity of light transmitted through the sample as a function of wavelength.

3.3: ML Regressors:

Various ML regressors were used for the building of this model namely,

3.3.1: Support Vector Regressor

Support Vector Regression (SVR) is a supervised machine learning algorithm used for regression tasks. Unlike traditional regression models that aim to minimize prediction errors, SVR focuses on finding a function that best fits the data within a specified margin of error.

There are several concepts related to support vector regression (SVR) that you may want to understand in order to use it effectively. Here are a few of the most important ones:

Support vector machines (SVMs): SVR is a type of support vector machine (SVM), a supervised learning algorithm that can be used for classification or regression tasks. SVMs try to find the hyperplane in a high-dimensional space that maximally separates different classes or output values.

Kernels**:** SVR can use different types of kernels, which are functions that determine the similarity between input vectors. A linear kernel is a simple dot product between two input vectors, while a non-linear kernel is a more complex function that can capture more intricate patterns in the data. The choice of kernel depends on the data's characteristics and the task's complexity.

Hyperparameters**:** SVR has several hyperparameters that you can adjust to control the behaviour of the model. For example, the **'C'** parameter controls the trade-off between the insensitive loss and the sensitive loss. A larger value of **'C'** means that the model will try to minimize the insensitive loss more, while a smaller value of C means that the model will be more lenient in allowing larger errors.

Model evaluation: Like any machine learning model, it's important to evaluate the performance of an SVR model. One common way to do this is to split the data into a training set and a test set, and use the training set to fit the model and the test set to evaluate it. You can then use metrics like mean squared error (MSE) or mean absolute error (MAE) to measure the error between the predicted and true output values. [48]

Here's a brief overview of SVR:

1. Margin of Tolerance: SVR introduces the concept of a margin of tolerance around the predicted value. Instead of minimizing error directly, SVR seeks to ensure that most of the data points fall within this margin.

2. Support Vectors: SVR identifies a subset of training data points, known as support vectors, which are crucial for defining the regression function. These support vectors lie either on the margin boundaries or within the margin itself.

3. Kernel Trick: SVR often employs a kernel function to transform the input features into a higher-dimensional space, where the data might be more separable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels.

4. Regularization Parameter: SVR includes a regularization parameter (C) that controls the trade-off between maximizing the margin and minimizing the error on the training data. Higher values of C prioritize minimizing training error, while lower values prioritize maximizing the margin.

5. Epsilon Parameter: Another parameter in SVR is ε (epsilon), which defines the width of the margin of tolerance. Larger values of ε allow more data points to fall outside the margin, potentially resulting in a wider margin and a less complex model.

SVR is particularly effective for datasets with complex relationships between features and targets, as it can capture non-linear patterns by using appropriate kernel functions. However, SVR's performance heavily depends on choosing suitable kernel functions and tuning the hyperparameters effectively. Additionally, SVR might not be the best choice for very large datasets due to its computational complexity.

3.3.2: Partial Least Square Regressor

Partial least square regression is a Machine learning Algorithm used for modelling the relationship between independent and dependent variables. This is mainly used when there are many interrelated independent variables. It is more commonly used in regression and latent variable modelling. It finds the directions (latent variables) in the independent variable space, explaining the maximum variance in both dependent and independent variables. It iteratively extracts the latent variables to find the maximum covariance between dependent and independent variables. The article explores more PLS Regression and implementation using the Sklearn library.

Partial least squares regression (PLS regression) is a statistical technique that shares similarities with principal components regression. Instead of identifying hyperplanes of maximum variance between the response and independent variables, PLS regression constructs a linear regression model by projecting both the predicted and observable variables into a new space. This characteristic of projecting data to new spaces classifies PLS methods as bilinear factor models. Partial least squares discriminant analysis (PLS-DA) is a specific variant used when the response variable (Y) is categorical.

PLS is employed to uncover the underlying relationships between two matrices (X and Y). It takes a latent variable approach to model the covariance structures in these matrices. The objective of a PLS model is to identify a multidimensional direction in the X space that explains the maximum multidimensional variance direction in the Y space. PLS regression is particularly advantageous when the predictor matrix has more variables than observations and when there is multicollinearity among X values. This is in contrast to standard regression, which may struggle in these situations unless regularization is applied. [49]

Here's a concise overview of PLSR:

1. Dimensionality Reduction: PLSR aims to reduce the dimensionality of the predictor variables while still preserving their relationship with the response variable. It achieves this by extracting a small number of latent variables, or components, that explain the maximum covariance between the predictor variables and the response variable.

2. Iterative Process: PLSR iteratively constructs these latent variables by maximizing the covariance between the predictor variables and the response variable in each component. Unlike Principal Component Analysis (PCA), which focuses solely on explaining the variance of the predictor variables, PLSR considers both the variance and the covariance.

3. Simultaneous Modelling: PLSR builds the latent variables in a way that optimally predicts the response variable while also considering the predictor variables. This simultaneous modelling approach allows PLSR to handle multicollinearity effectively, making it suitable for situations where predictors are highly correlated.

4. Prediction and Interpretation: Once the latent variables are constructed, PLSR can be used for prediction by regressing the response variable on these latent variables. Additionally, PLSR provides insights into the relationships between the predictors and the response, making it valuable for interpretation.

5. Parameter Tuning: PLSR includes parameters such as the number of components to extract and the scaling method, which can affect the performance of the model. Cross-validation techniques are often used to determine the optimal number of components and other tuning parameters.

PLSR is widely used in fields such as chemometrics, spectroscopy, and genetics, where datasets often have high dimensionality and collinearity among predictors. It offers a balance between prediction accuracy and interpretability, making it a valuable tool for analysing complex data relationships. However, like any modelling technique, the effectiveness of PLSR depends on appropriate data preprocessing, model tuning, and validation procedures.

3.3.3: Decision Tree Regressor

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

- Conditions [Decision Nodes]
- Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

Discrete output example: A weather prediction model that predicts whether or not there'll be rain on a particular day.
Continuous output example: A profit prediction model that states the probable profit that can be generated from the sale of a product.
Here, continuous values are predicted with the help of a decision tree regression model. [50]

Here's a brief overview:

1. Tree Structure: The decision tree is constructed recursively by splitting the data into subsets based on the values of features. At each node of the tree, a decision is made regarding which feature to split on and what threshold to use for the split. This process continues until a stopping criterion is met, such as reaching a maximum tree depth or minimum number of samples per leaf.

2. Splitting Criteria: The algorithm selects the best feature and threshold for splitting the data at each node based on a splitting criterion, typically aiming to minimize variance or mean

squared error in the resulting subsets. Common splitting criteria include mean squared error, mean absolute error, and variance reduction.

3. Predictions: Once the decision tree is built, predictions for new data points are made by traversing the tree from the root node to a leaf node. The predicted value for a data point is typically the average (or another aggregation) of the target variable in the leaf node to which it belongs.

4. Interpretability: Decision trees are highly interpretable models, as they can be visualized graphically, allowing users to understand the decision-making process and identify important features for prediction.

5. Overfitting: Decision trees are prone to overfitting, especially when the tree depth is not properly constrained. Regularization techniques such as pruning or limiting the maximum depth of the tree are commonly used to mitigate overfitting.

6. Ensemble Methods: Decision trees can be further improved by using ensemble methods such as Random Forests or Gradient Boosting, which train multiple decision trees and combine their predictions to achieve better performance and generalization.

Decision Tree Regressors are suitable for both small and large datasets and can handle numerical and categorical features without requiring feature scaling. However, they may not capture complex relationships in the data as effectively as some other algorithms, and their performance can vary depending on the quality of the data and the choice of hyperparameters.

3.3.4: Random Forest Regressor

A Random Forest Regressor is an ensemble learning method used for regression tasks, based on the Random Forest algorithm. It is a powerful and versatile machine learning technique that builds multiple decision trees and combines their predictions to produce more accurate and stable results.

Random Forest Regression is a versatile machine-learning technique for predicting numerical values. It combines the predictions of multiple decision trees to reduce overfitting and improve accuracy. Python's machine-learning libraries make it easy to implement and optimize this approach.

Ensemble Learning:

Ensemble learning is a machine learning technique that combines the predictions from multiple models to create a more accurate and stable prediction. It is an approach that leverages the collective intelligence of multiple models to improve the overall performance of the learning system.

Types of Ensemble Methods"

There are various types of ensemble learning methods, including:

Bagging (Bootstrap Aggregating): This method involves training multiple models on random subsets of the training data. The predictions from the individual models are then combined, typically by averaging.

Boosting: This method involves training a sequence of models, where each subsequent model focuses on the errors made by the previous model. The predictions are combined using a weighted voting scheme.

Stacking: This method involves using the predictions from one set of models as input features for another model. The final prediction is made by the second-level model.

A random forest is an ensemble learning method that combines the predictions from multiple decision trees to produce a more accurate and stable prediction. It is a type of supervised learning algorithm that can be used for both classification and regression tasks.

Every decision tree has high variance, but when we combine all of them in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data, and hence the output doesn't depend on one decision tree but on multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is called Aggregation.

Here's a concise overview:

1. Ensemble Learning: A Random Forest Regressor consists of a collection of decision trees, where each tree is trained independently on a random subset of the training data and a random subset of the features. This randomness helps to reduce overfitting and improves the model's robustness.

2. Bootstrap Aggregating (Bagging): Random Forest employs a technique called bootstrap aggregating or bagging, where each tree is trained on a bootstrapped sample of the original

training data. This sampling with replacement ensures diversity among the trees, leading to a more robust ensemble model.

3. Random Feature Selection: In addition to sampling data points, Random Forest also randomly selects a subset of features at each split in the decision tree. This feature randomness further enhances the diversity among the trees and reduces the correlation between them.

4. Prediction: Once the ensemble of decision trees is built, predictions for new data points are made by aggregating the predictions of individual trees. For regression tasks, the final prediction is often the average (or another aggregation) of the predictions from all the trees.

5.Interpretability and Versatility: While Random Forests may not be as interpretable as individual decision trees, they offer high predictive accuracy and can handle both numerical and categorical features without requiring feature scaling. They are also less sensitive to outliers and noise in the data.

6. Hyperparameter Tuning: Random Forests have several hyperparameters that can be tuned to optimize performance, such as the number of trees in the ensemble, the maximum depth of each tree, and the size of the random feature subsets.

7. Applications: Random Forest Regressors are widely used in various domains, including finance, healthcare, and ecology, for tasks such as predicting stock prices, estimating patient outcomes, and analysing ecological data.

Overall, Random Forest Regressors are robust, versatile, and effective models for regression tasks, offering a balance between predictive accuracy and interpretability. They are particularly well-suited for complex datasets with high-dimensional feature spaces and nonlinear relationships.

3.3.5: K Nearest Neighbor Regressor

The K-Nearest Neighbors (KNN) Regressor is a simple yet powerful supervised machine learning algorithm used for regression tasks. It predicts the value of a target variable by averaging the values of its K nearest neighbors in the feature space. Here's a brief overview:

1. Neighbor-based Approach: KNN Regressor operates on the principle that similar data points tend to have similar target variable values. It does not explicitly learn a model but rather memorizes the training data to make predictions.

2. Parameter K: The "K" in KNN refers to the number of nearest neighbors to consider when making a prediction. A higher value of K considers more neighbors, potentially resulting in smoother predictions but may overlook local patterns. Conversely, a lower value of K focuses on fewer neighbors, capturing more local variations but may be sensitive to noise.

3. Distance Metric: KNN Regressor typically uses a distance metric, such as Euclidean distance or Manhattan distance, to measure the similarity between data points in the feature space. The choice of distance metric can influence the algorithm's performance and should be selected based on the characteristics of the data.

4. Prediction: To make a prediction for a new data point, KNN Regressor identifies the K nearest neighbors based on the chosen distance metric and averages their target variable values to obtain the predicted value. In regression tasks, this average is often the mean value of the target variable among the nearest neighbors.

5. Non-parametric Nature: KNN Regressor is a non-parametric algorithm, meaning it does not assume any specific form for the underlying data distribution. Instead, it relies solely on the training data during prediction, making it flexible and adaptable to different types of data distributions.

6. Scalability and Efficiency: While KNN Regressor is conceptually simple and easy to implement, it can be computationally expensive, especially for large datasets or high-dimensional feature spaces. Techniques such as KD-trees or ball trees can be employed to improve the algorithm's efficiency.

7. Hyperparameter Tuning: The choice of K and the distance metric are critical hyperparameters in KNN Regressor. Cross-validation techniques can be used to select optimal values for these hyperparameters based on the performance on a validation set.

KNN Regressor is particularly useful for datasets with nonlinear relationships and where local variations are significant. However, it may not perform well in high-dimensional spaces or with datasets containing many noisy or irrelevant features. Additionally, it requires careful consideration of the choice of K and the distance metric to achieve optimal performance.

Table 2: Parameters

Regressor	Parameter
SVR	C=10.0, kernel= ‘rbf’, coef0= 0.01, degree=3, gamma= ‘scale’
PLSR	n_components=10
DT	random_state=0
RF	max_depth=2, random_state=0
KNR	n_neighbors=1, p=1, weights= ‘distance’, algorithm= ‘auto’

3.4: Pre-Processing

The Savitzky-Golay filter was applied which is a widely used digital smoothing filter, particularly in signal processing and data analysis. Unlike other smoothing techniques, such as moving averages, the Savitzky-Golay filter preserves important features of the signal, such as sharp peaks and step changes, while effectively removing noise. It achieves this by fitting a polynomial function to small subsets of adjacent data points, then using the coefficients of this polynomial to compute smoothed values for each data point. The filter provides a balance between noise reduction and preserving signal detail, making it suitable for a wide range of applications, including chromatography, spectroscopy, and time series analysis. The Savitzky-Golay filter is computationally efficient and can be easily adjusted by tuning parameters such as the window size and polynomial order to suit the characteristics of the data being processed. Overall, it's a valuable tool for enhancing the quality of noisy signals while retaining important information.

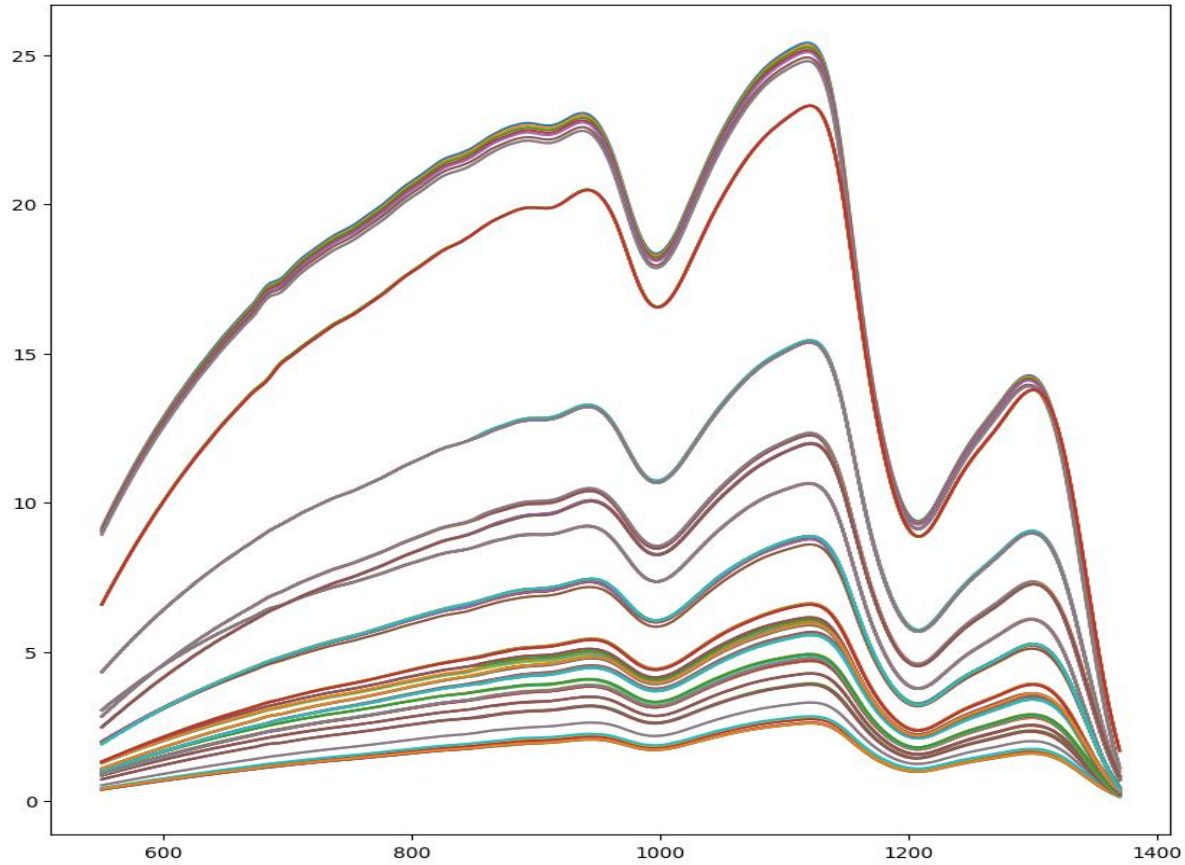

Figure 7: Smoothened Spectra

3.5: Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique used in data analysis and machine learning. Its primary goal is to simplify complex datasets while retaining the most important information.

In PCA, the algorithm identifies the directions (principal components) along which the data varies the most. These components are orthogonal to each other, meaning they are uncorrelated. The first principal component explains the largest amount of variance in the data, followed by the second component, and so on.

PCA is useful for several purposes:

1. Dimensionality Reduction: By representing data using a smaller number of principal components, PCA reduces the dimensionality of the dataset while preserving as much variance as possible.

2. Visualization: PCA can help visualize high-dimensional data in a lower-dimensional space, making it easier to explore and interpret.

3. Feature Extraction: PCA can be used to extract the most important features from a dataset, which can then be used for further analysis or modelling.

4. Noise Reduction: PCA can help remove noise and redundant information from the data, leading to better performance in downstream tasks.

3.6: Data Partitioning

K-fold cross-validation is a widely used technique in machine learning for model evaluation. It involves partitioning the dataset into K equally sized folds, where K is chosen to be 10. The model is trained and evaluated K times, each time using a different fold as the validation set and the remaining folds as the training set. This process allows for thorough testing of the model's performance across different subsets of the data.

During each iteration of K-fold cross-validation, the model is trained on K-1 folds and evaluated on the remaining fold. This ensures that every data point is used for both training and validation exactly once across the K iterations. The final performance metric, such as accuracy or mean squared error, is then calculated by averaging the results from the K iterations.

K-fold cross-validation provides several advantages, including more reliable estimates of the model's performance compared to a single train-test split, as it reduces the variance associated with a single partitioning of the data. It also allows for better utilization of the available data, particularly in cases where the dataset is small. However, it's worth noting that K-fold cross-validation can be computationally expensive, especially for large datasets or complex models. Additionally, while K-fold cross-validation provides a good estimate of the model's performance on the training data, it may not generalize well to unseen data, especially if the data is not representative or if there is significant variability in the data distribution.

Overall, K-fold cross-validation is a valuable tool for model evaluation and selection, providing a robust estimate of the model's performance while helping to mitigate the risk of overfitting.

Chapter 4: Analysis and Conclusions

In this chapter, we delve in the analysis and comparative study of various ML Regressor for prediction of Adulteration in honey using Near Infrared Spectroscopy. The NIR spectra for pure and adulterated honey is shown in Figure 7. Significant differences were observed for adulterated solutions of various concentrations. The main differences were observed in the range of 550 to 1400 nm.

4.1: Performance Analysis

In this study, RMSE value was used to determine the performance of all the models.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{Predicted - Actual}{n}}$$

As discussed in the literature review, SVR, KNR, DT, RF, PLSR were employed by the authors and they all gave distinct results and contributed to forthcoming studies. In this study, aim was to comparatively analyse all these regressors. Table 3 gives the RMSE of the all the regressors that were studied.

As shown in the table, KNR, PLSR, DT and SVR show low RMSE showing the models effectiveness.

The Google-Colab online platform was used for all of the studies and evaluations.

Regressor	Average RMSE
SVR	8.72
PLSR	2.54
DT	0.23
RF	0.33
KNR	0.92

Table 3: Summarised Performance of ML Regressors

It was observed that the transmittance of every adulterated sample varied while the structure of the spectra remained similar. Actual v/s Predicted graph of the results obtained by the algorithms that are PLSR, SVR, DTR, RFR and KNR are shown in figure 8, figure 9, figure 10, figure 11 and figure 12 respectively.

A summary of the results obtained by the algorithms are shown in Table 3. Average RMSE obtained after the 10 fold cross validation was be 8.72, 2.54, 0.23, 0.33, 0.92 for SVR, PLSR, DTR, RFR and KNR respectively.

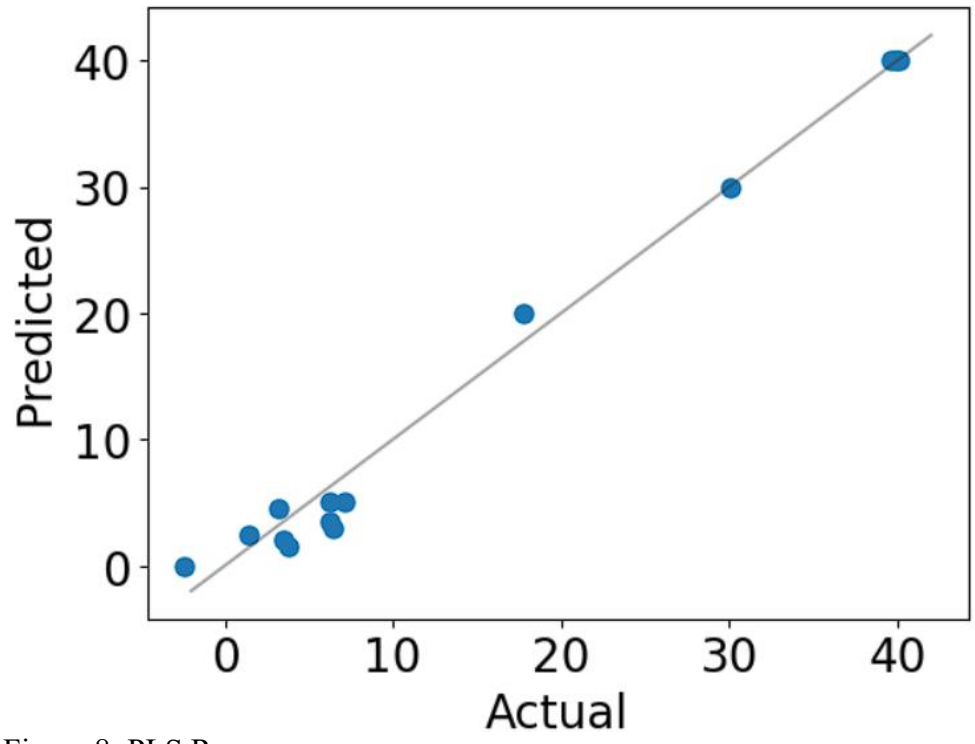

Figure 8: PLS Regressor

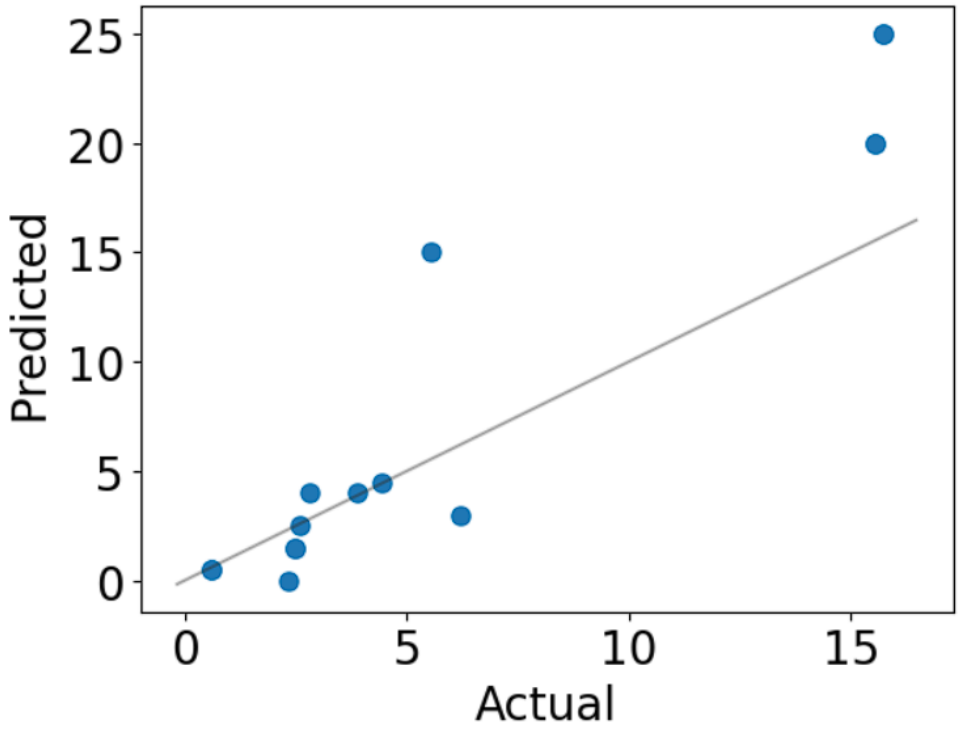

Figure 9: SVR

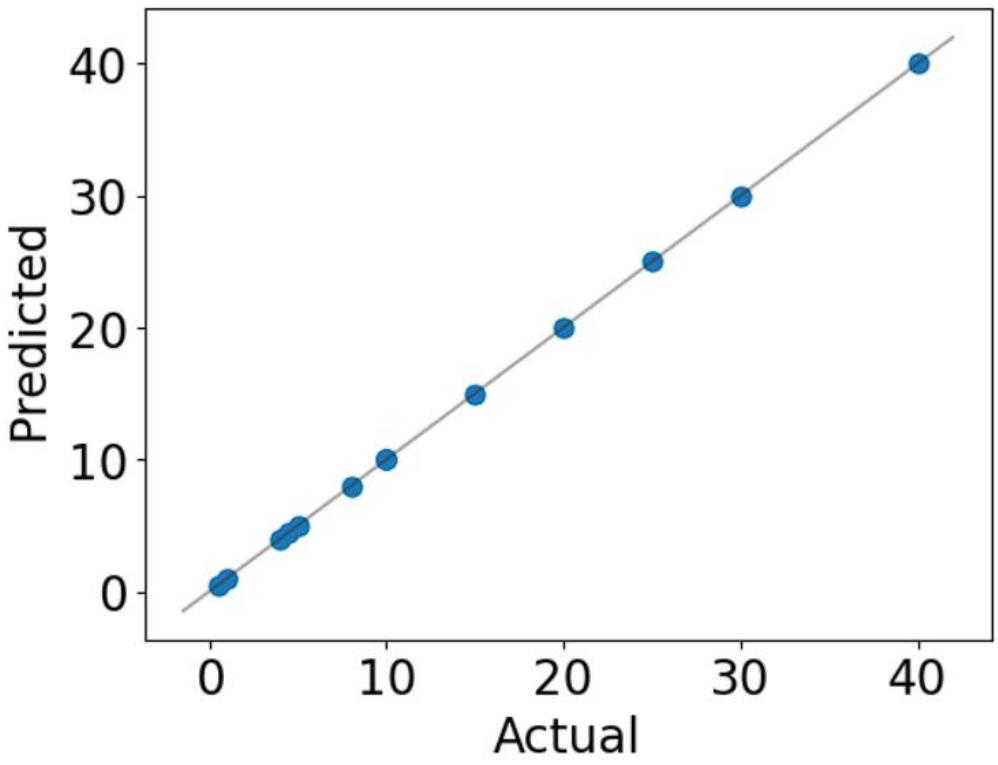

Figure 10: DTR

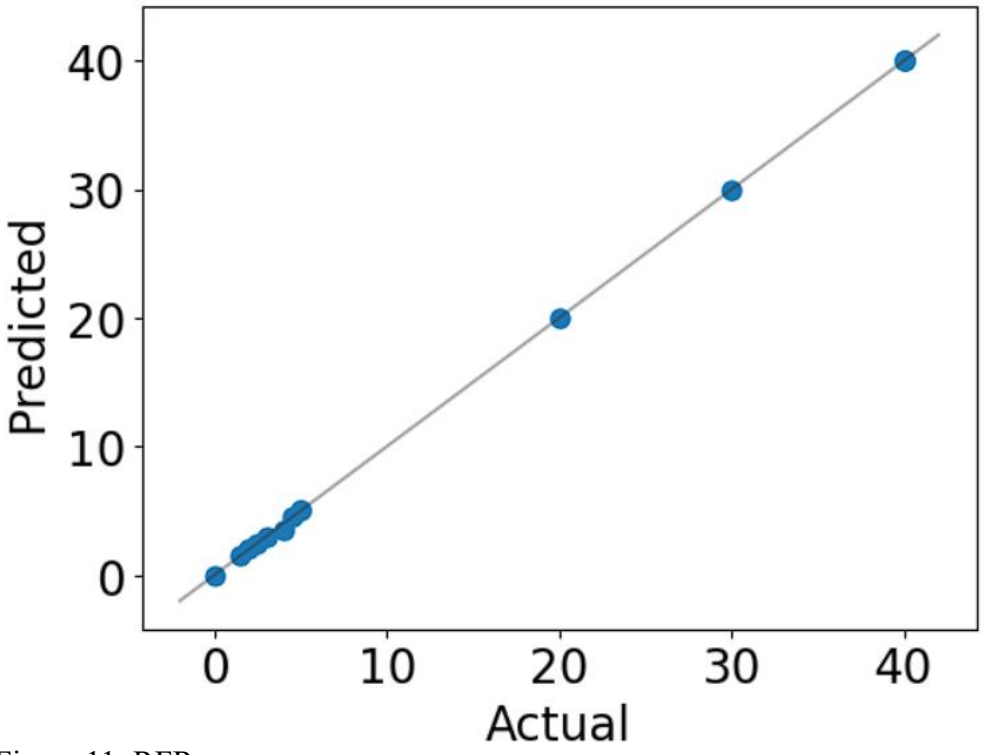

Figure 11: RFR

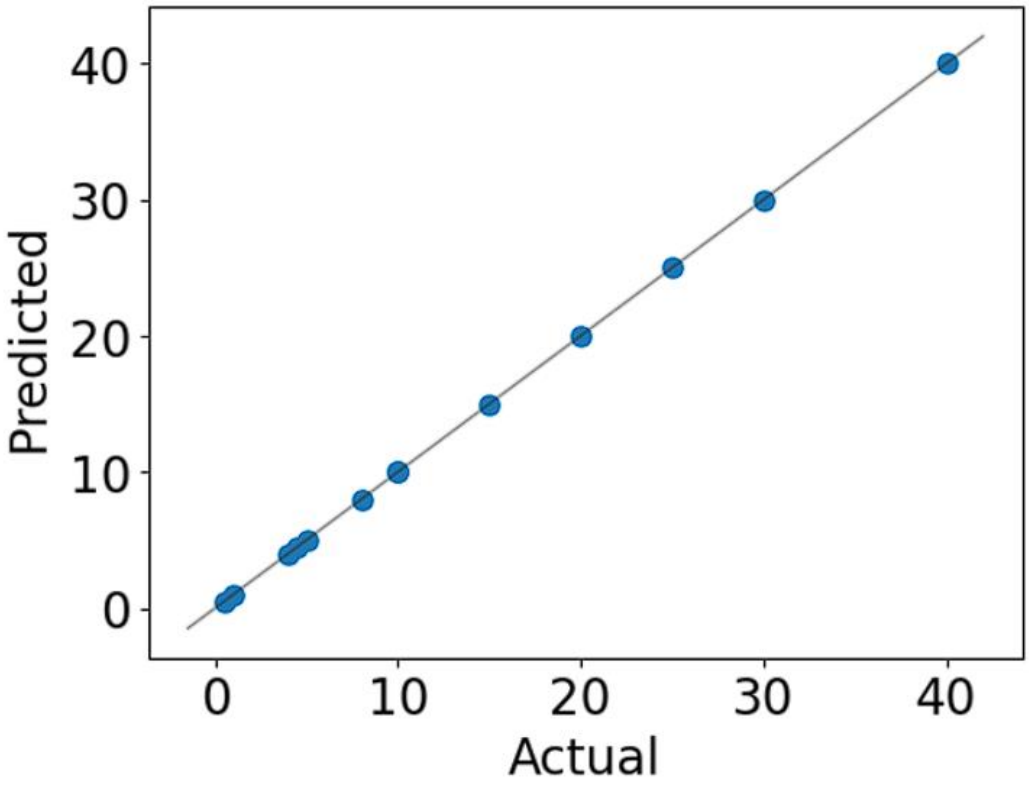

Figure 12: KNR

4.2: **Conclusion**

This work integrated dataset containing honey adulteration spectra obtained by NIR Spectroscopy, SG filter, PCA and ML Regressors to identify adulteration of sugar syrup in authentic honey. This study concluded that the application of ML regressors gave good results for identifying adulteration concentration in authentic honey. This work also provides a non-destructive way to determine any kind of adulteration in other food items as well. The application of DTR, RFR and KNR has shown more sensitivity to the minor sugar syrup concentrations in honey as compared to PLSR and SVR.

NIR Spectroscopy has shown a potential to be used as a screening technique for the identification of adulterants in honey samples and is significantly sensitive to a varying range of concentrations.

Future scope of this work can be aimed to prediction of much broader range of concentrations thus to identify the limit of this method. Sugar syrups such as corn syrup, maple syrup can be identified.

References

[1] Initial Study of Honey Adulteration by Sugar Solutions Using Midinfrared (MIR) Spectroscopy and Chemometrics J. F. Daniel Kelly

DOI: 10.1021/jf034985q

[2] The Toxic Effect of Honey Adulteration- A Review

DOI: 10.3390/foods9111538

[3] Sugar detection in adulterated honey using hyper-spectral imaging with stacking generalization method by Madhusudan G. Lanjewar

DOI: https://doi.org/10.1016/j.foodchem.2024.139322

[4] S. Kawano et. al., Determination of sugar content in intact peaches by near infrared spectroscopy with fiber optics in interactance mode.

DOI: 10.17660/ActaHortic.1993.334.27

[5] Determination of Brix and Acidity in Pineapple Fruits by Near Infrared Spectroscopy

DOI: 10.17660/ActaHortic.1993.334.27

[6] J. Daniel Kelly et. al., Application of Fourier Transform Midinfrared Spectroscopy to the Discrimination between Irish Artisanal Honey and Such Honey Adulterated with Various Sugar Syrup

DOI: https://doi.org/10.1021/jf034985q

[7] J. Daniel Kelly et. al., Application of Fourier Transform Midinfrared Spectroscopy to the Discrimination between Irish Artisanal Honey and Such Honey Adulterated with Various Sugar Syrups

DOI: https://doi.org/10.1021/jf0613785

[8] M. M. Ali et. al., Prediction of total soluble solids and pH in banana using near infrared spectroscopy

[9] M. E. Bahrami et. al., Measurement of quality parameters of sugar beet juices using near-infrared spectroscopy and chemometrics

DOI:10.1016/j.jfoodeng.2019.109775

[10] Lanzhen Chen et. al., Determination of Chinese honey adulterated with high fructose corn syrup by near infrared spectroscopy

DOI:https://doi.org/10.1016/j.foodchem.2010.10.027

[11] M. Benkovic et. al., Qualitative and Quantitative Detection of Acacia Honey Adulteration with Glucose Syrup Using Near-Infrared Spectroscopy by

DOI: **https://doi.org/10.3390/separations9100312**

[12] Ma José Aliaño-González et. al.**, A** screening method based on Visible-NIR spectroscopy for the identification and quantification of different adulterants in high-quality honey

DOI:https://doi.org/10.1016/j.talanta.2019.05.067

[13] Sorina Ropciuc et. al., Honey authentication and adulteration detection using emission – excitation spectra combined with chemometrics

DOI:https://doi.org/10.1016/j.saa.2023.122459

[14] Xinhao Yang et. al., Manuka honey adulteration detection based on near-infrared spectroscopy combined with aquaphotomics
DOI: https://doi.org/10.1016/j.lwt.2020.109837

[15] Raypah, Muna E. et al., Identification of Stingless Bee Honey Adulteration Using Visible-Near Infrared Spectroscopy Combined with Aquaphotomics

DOI: 10.3390/molecules27072324

[16] Xiangrong Zhu et. al., Detection of adulterants such as sweeteners materials in honey using near-infrared spectroscopy and chemometrics
DOI: https://doi.org/10.1016/j.jfoodeng.2010.06.014

[17] D. Valinger et. al., Detection of honey adulteration – The potential of UV-VIS and NIR spectroscopy coupled with multivariate analysis
DOI: https://doi.org/10.1016/j.lwt.2021.111316

[18] NIR detection of honey adulteration reveals differences in water spectral pattern by György Bázár et. al.
DOI: https://doi.org/10.1016/j.foodchem.2015.08.092

[19] Verification of authenticity and fraud detection in South African honey using NIR spectroscopy By Anina Guelpa
DOI:https://doi.org/10.1016/j.foodcont.2016.11.002

[20] Detection of jaggery syrup in honey using near-infrared spectroscopy by Mishra, Sunita et. al.

DOI:10.3109/09637480903476415

[21] Classification of simple and complex sugar adulterants in honey by mid-infrared spectroscopy by Sivakesava, Sakhamuri et. al.

DOI:10.1046/j.1365-2621.2002.00573.x

[22] Application of ANOVA-simultaneous component analysis to quantify and characterize effects of age, temperature, syrup adulteration and irradiation on near-infrared (NIR) spectral data of honey by Alexandra Rust et. al.

DOI:https://doi.org/10.1016/j.saa.2021.119546

[23] Detection of adulteration in Chinese honey using NIR and ATR-FTIR spectral data fusion by Huang, Furong et. al.

DOI: 10.1016/j.saa.2020.118297

[24] Detection of honey adulteration by high fructose corn syrup and maltose syrup using Raman spectroscopy by Li, Shuifang et. al

DOI:10.1016/j.jfca.2012.07.006

[25] Application of near infrared spectroscopy and classical analytical methods for the evaluation of Hungarian honey by Z. Bodor et. al.

DOI:10.1556/446.14.2018.S1.2

[26] Adulteration detection of honey based on near-infrared spectroscopy by Tu, Zhenhua et. al.

DOI:10.3969/j.issn.1002-6819.2011.11.071

[27] Potential of near infrared transflectance spectroscopy to detect adulteration of Irish honey by beet invert syrup and high fructose corn syrup by Kelly, J. Daniel et. al.

DOI:10.1255/jnirs.599

[28] Physicochemical properties and detection of glucose syrup adulterated Kelulut (Heterotrigonaitama) honey using Near-Infrared spectroscopy by Woeng, Venecia et. al.

DOI: 10.1111/jfpp.16686

[29] Honey Adulteration Detection Using Raman Spectroscopy by M. Oroian et. al.

[30] Food analysis by portable NIR spectrometer by Gabriely S. Folli et. al.

[31] Application of UV–Vis spectroscopy for the detection of adulteration in Mediterranean honeys by Dafni D. et. al.

[32] Using sensor and spectral analysis to classify botanical origin and determine adulteration of raw honey by Zhilin Gan et. al.

[33] Application of FTIR-HATR spectroscopy and multivariate analysis to the quantification of adulterants in Mexican honeys by Tzayhri Gallardo-Velázquez et.al.

[34] Detection of adulteration in honey by infrared spectroscopy and chemometrics: Effect on human health by W. Skaff et. al.

[35] Classification of Honey According to Geographical and Botanical Origins and Detection of Its Adulteration Using Voltammetric Electronic Tongue by Bougrini et. al.

[36]A Rapid Spectroscopic Technique for Determining Honey Adulteration with Corn Syrup by S. Sivakesava et. al.

[37] Adulteration of honey and available methods for detection – a review by Zábrodská et. al.

[38] Kushnir I, Sensitive thin layer chromatographic detection of high fructose corn syrup and other adulterants in honey.

[39] S. Shafiee, Detection of Honey Adulteration using Hyperspectral Imaging

[40] S. Amiry et. al., Classification of adulterated honeys by multivariate analysis
DOI: https://doi.org/10.1016/j.foodchem.2016.12.025

[41] D. Bertelli et. al., Detection of Honey Adulteration by Sugar Syrups Using One-Dimensional and Two-Dimensional High-Resolution Nuclear Magnetic Resonance

DOI:https://doi.org/10.1021/jf101460t

[42] U. Kamboj et. al., Prediction of Adulteration in Honey Using Rheological Parameters

DOI:10.1080/10942912.2014.962656

[43] Indimuli A. M. et. al., Non-linear Correlation of Absorbance with Respect to Concentration of Sugar in Aqueous Solutions of High Purity Laboratory Chemical Reagent Sucrose and Ordinary Cane Sugar

[44] P. N. Syahirah et. al. Physicochemical Analysis of Several Natural Malaysian Honeys and Adulterated Honey

DOI: 10.1088/1757-899X/440/1/012049

[45]The Sugars of Honey - A Review by Doner, L.W.

[46] The Beer-Lambert Law

Link:

https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Spectroscopy/Electronic_Spectroscopy/Electronic_Spectroscopy_Basics/The_Beer-Lambert_Law

[47] Beer-Lambert Law | Transmittance & Absorbance

Link: https://www.edinst.com/blog/the-beer-lambert-law/

[48] https://www.geeksforgeeks.org/support-vector-regression-svr-using-linear-and-non-linear-kernels-in-scikit-learn/

[49] https://www.geeksforgeeks.org/partial-least-squares-regression-plsregression-using-sklearn/

[50] https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/

[51] https://www.geeksforgeeks.org/random-forest-regression-in-python/

Appendix

Features of Jasco- V-770 Spectrophotometer:

A wide range UV-Visible/Near Infrared Spectrophotometer with a unique optical design featuring a single monochromator and dual detectors for the wavelength range from 190 to 2700nm (3200nm option).

The V-770's single monochromator design provides for maximum light throughput with excellent absorbance linearity. A PMT detector is used for the UV to visible region and a Peltier-cooled PbS detector for the NIR region.

The V-770 UV-Visible/NIR spectrophotometer is operated using Spectra Manager™ Suite. This innovative cross-platform spectroscopy software is compatible with Windows 7 Pro (32- and 64-bit) and Windows 8.1 operating systems.
For simple operation, the handheld iRM has a great look and feel with a colour touch-sensitive screen. Data can also be downloaded to Spectra Analysis on a PC further PC data processing.
The V-700 Series has a growing list of software applications for both Spectra Manager™ and iRM. If you have an application which you don't see listed, please let us know as we may already have it or we can prepare an application designed specifically for your requirements.

Optical System	Czerny-Turner grating mount Single monochomator Fully symmetrical double beam
Light source	Halogen lamp, Deuterium lamp
Wavelength range	190 to 2700 nm (3200 nm option)
Wavelength accuracy	+/-0.3 nm (at 656.1 nm) +/-1.5 nm (at 1312.2 nm)
Wavelength repeatability	+/-0.05 nm (UV-Vis), +/-0.2 nm (NIR)
Spectral bandwidth (SBW)	UV-Visible: 0.1, 0.2, 0.5, 1, 2, 5, 10 nm L2, L5, L10 nm (low stray light mode) M1, M2 nm (micro cell mode) NIR: 0.4, 0.8, 1, 2, 4, 8, 20, 40 L8, L20, L40 nm (low stray light mode) M4, M8 nm (micro cell mode)
Stray light	1 % (198 nm KCL) 0.0005 % (220 nm NaI) 0.0005 % (340 nm NaNO2) 0.0005 % (370 nm NaNO2)

	SBW: L2 nm 0.04 % (1420 nm: H2O) 0.1 % (1690 nm: CH2Br2) SBW: L8 nm
Photometric range	UV-Visible: -4~4 Abs NIR: -3~3 Abs
Photometric accuracy	+/-0.0015 Abs (0 to 0.5 Abs) +/-0.0025 Abs (0.5 to 1 Abs) +/-0.3 %T Tested with NIST SRM 930D
Photometric repeatability	+/-0.0005 Abs (0 to 0.5 Abs) +/-0.0005 Abs (0.5 to 1 Abs) Tested with NIST SRM 930D
Scanning speed	10~4000 nm/min (8000 nm/min in preview mode)
Slew speed	UV-Vis: 12,000 nm/min NIR: 48,000 nm/min
RMS noise	0.00003 Abs (0 Abs, wavelength: 500 nm, measurement time: 60 sec, SBW: 2 nm)
Baseline stability	0.0003 Abs/hour (Wavelength: 250 nm, response: slow and SBW: 2 nm)
Baseline flatness	+/-0.0002 Abs (200 - 2500 nm)
Detector	PMT, Peltier cooled PbS
Standard functions	IQ accessories, Start button, Analog output
Standard programs	Abs/%T meter, Quantitative analysis, Spectrum measurement, Time course measurement, Fixed wavelength measurement, Validation, Daily check, Dual wavelength time course measurement
Dimensions and weight	460(W) x 602(D) x 268(H) mm, 29 kg
Power requirements	150 VA
Installation requirements	Room temperature: 15-30 Celsius, humidity: below 85%

Code:

```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
dataset=pd.read_csv('/content/drive/MyDrive/dataset1.csv')
x = dataset
x = dataset.drop(['Concentration'], axis=1).to_numpy()
wavelenght = x[0,:]
x= x[1:8676,:]
ttt = x[0,:]

for i in range(144):

  ttt = x[i,:]

  offset = 0.0

  temp = 0

  for h in range((len(ttt)-1),0,-1):



    if ( abs(ttt[h] - ttt[h-1])> 0.4):



      offset = abs(ttt[h] - ttt[h-1])
```

```
      temp = h
      break
  for k in range(temp):
    ttt[k]  = ttt[k] - offset


  x[i,:] = ttt


for i in range(len(wavelenght)):
  if (wavelenght[i] == 450):
    print(i)
# importing modules
import numpy
from matplotlib import pyplot


plt.figure(figsize=(16,10))


# plotting the signal
pyplot.plot(wavelenght, x.T)
```

```
pyplot.xlabel(' Wavelength')

pyplot.ylabel('%T')

pyplot.title("Spectra")



pyplot.show()

z = x[:,130:950]



wavelenghtz = wavelenght[130:950]

z.shape

# importing modules

import numpy

from matplotlib import pyplot



plt.figure(figsize=(16,10))



# plotting the signal

pyplot.plot(wavelenghtz, z.T)



pyplot.xlabel(' Wavelength')
```

```
pyplot.ylabel('%T')

pyplot.title("Spectra")



#pyplot.plot(wavelenghtz, gg.T)

pyplot.title("Spectra")

pyplot.show()



y= dataset.Concentration.to_numpy()

y = y[1:8676]

y

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt



x_smooth = signal.savgol_filter(z, window_length=23, polyorder=2) # , , deriv =

1,mode='mirror'

f = pyplot.figure()

f.set_figwidth(10)

f.set_figheight(10)
```

```
print("Plot after re-sizing: ")

pyplot.plot(wavelenghtz, x_smooth.T)

pyplot.show()

# Import necessary libraries

from sklearn.decomposition import PCA


'''

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test  = train_test_split(x_smooth, y, test_size=0.1,

random_state=0)'''


# Apply PCA

pca = PCA(n_components =  4)

X_pca = pca.fit_transform(x_smooth)

# Explained variance

explained_variance = pca.explained_variance_

total_explained_variance = explained_variance.sum()
```

```
# Print results

print(f"Explained Variance:\n{explained_variance}")

print(f"Total Explained Variance: {total_explained_variance:.4f}")

# Explained variance ratio

explained_variance_ratio = pca.explained_variance_ratio_

total_explained_variance_ratio = explained_variance_ratio.sum()



# Print results

print(f"\nExplained Variance Ratio:\n{explained_variance_ratio}")

print(f"Total Explained Variance Ratio: {total_explained_variance_ratio:.4f}")

# Import necessary libraries

import numpy as np

import matplotlib.pyplot as plt



# Plot explained variance ratio

cumulative_variance_ratio = np.cumsum(explained_variance_ratio)

f = plt.figure()

f.set_figwidth(10)

f.set_figheight(10)
```

```
plt.plot(cumulative_variance_ratio, marker='o')

plt.xlabel('Number of Principal Components')

plt.ylabel('Cumulative Explained Variance Ratio')

plt.title('Cumulative Explained Variance Ratio by Principal Components')

plt.show()
```

KNR:

```
import matplotlib

import numpy as np

from sklearn.model_selection import KFold

from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean_squared_error




# Apply PCA

pca = PCA(n_components =  2)

x_pca = pca.fit_transform(x_smooth)




rmse = 0
```

```
kf = KFold(n_splits=10, shuffle=True, random_state=42)


for i, (train_index, test_index) in enumerate(kf.split(x_pca)):
    print(f"Fold {i}:")
    #print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = x_pca[train_index], x_pca[test_index]
    y_train, y_test = y[train_index], y[test_index]


    X_train = X_train.astype(np.float64)
    X_test = X_test.astype(np.float64)
    y_test = y_test.astype(np.float64)
    y_train = y_train.astype(np.float64)




    neigh = KNeighborsRegressor( n_neighbors = 1, p = 1, weights =
'distance',  algorithm = 'auto' )
```

```
    neigh.fit(X_train, y_train)



    y_pred=neigh.predict(X_test)

    y_pred=np.array(y_pred).flatten()

    qq1=np.array(y_test).flatten()



    print( "*****************************")

    #plt.plot(qq1,y_pred)

    mat_plot(qq1,y_pred)



    rmse = rmse + np.sqrt(mean_squared_error(qq1,y_pred))



    print( "Result for each fold " + str(i))

    print( "*****************************")




rmse = rmse /10

print("average RMSE",rmse)
```

```
from sklearn.model_selection import cross_val_score

from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean_squared_error

# Import necessary libraries

from sklearn.decomposition import PCA



# Apply PCA

pca = PCA(n_components =  4)

x_pca = pca.fit_transform(x_smooth)



'''

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test  = train_test_split(x_pca, y, test_size=0.1,

random_state=42, shuffle= True)'''



neigh = KNeighborsRegressor()



x_pca = x_pca.astype(np.float64)
```

```
y = y.astype(np.float64)

'''

y_train = y_train.astype(np.float64)

y_test = y_test.astype(np.float64)

'''



from sklearn.model_selection import KFold

from sklearn.model_selection import GridSearchCV



parameters = {'n_neighbors':list(range(1, 5)) , 'p':list(range(1, 5)) ,

'weights':['distance'],  'algorithm': ['auto']}



from sklearn.model_selection import GridSearchCV

kf = KFold(n_splits=10, shuffle=True, random_state=42)

clf = GridSearchCV(neigh, parameters, cv = kf , return_train_score=False, scoring=

'neg_mean_absolute_error')  # scoring='neg_mean_squared_error

clf.fit(x_pca, y)

'''
```

```
#clf.cv_results_

ypred = clf.predict(x_test)

ypred=np.array(ypred).flatten()



aa=np.array(y_test).flatten()

mat_plot(aa,ypred)

'''

kf = KFold(n_splits=10, shuffle=True, random_state=42)

rmse = 0

for i, (train_index, test_index) in enumerate(kf.split(x_pca)):

    print(f"Fold {i}:")

    #print("TRAIN:", train_index, "TEST:", test_index)

    X_train, X_test = x_pca[train_index], x_pca[test_index]

    y_train, y_test = y[train_index], y[test_index]



    X_train = X_train.astype(np.float64)

    X_test = X_test.astype(np.float64)

    y_test = y_test.astype(np.float64)

    y_train = y_train.astype(np.float64)
```

```
    clf.fit(X_train, y_train)
    ypred = clf.predict(X_test)
    ypred=np.array(ypred).flatten()

    aa=np.array(y_test).flatten()
    #plt.plot(qq1,y_pred)
    mat_plot(aa,ypred)
    rmse = rmse + np.sqrt(mean_squared_error(aa,ypred))
df = pd.DataFrame(clf.cv_results_)
df

rmse = rmse /10
print("average RMSE",rmse)

from sklearn.model_selection import cross_val_score

print(cross_val_score(clf, x_pca, y,  cv=10, scoring ='r2'))
```

```
clf.best_params_

clf.best_score_

def mat_plot(a,b): # qq1 is the actual readings andthe b is the predictd

  from sklearn.metrics import mean_absolute_error

  from sklearn.metrics import mean_squared_error

  from sklearn.metrics import r2_score

  import numpy as np

  import matplotlib.pyplot as plt



  print("The R2 ", (r2_score( a, b)))

  print("RMSE:", np.sqrt(mean_squared_error(  a, b)))

  #print("MAPE%:", mean_absolute_percentage_error(  a, b))

  print("MAE",mean_absolute_error(a,b))

  print("MSE",mean_squared_error(a,b))

  print("RMSE",np.sqrt(mean_squared_error(a,b)))



  fig, ax = plt.subplots()
```

```
  ax.plot(b, a, linewidth=0,  marker="o", color='C0', markersize=8)


  #plot(x, y, color='green', linestyle='dashed', marker='o', markerfacecolor='blue',
markersize=12).
  low_x, high_x = ax.get_xlim()
  low_y, high_y = ax.get_ylim()
  low = max(low_x, low_y)
  high = min(high_x, high_y)
  ax.plot([low, high], [low, high], ls="-", c=".2", alpha=.4)
  #ax.set_title('R2 score')
  plt.rcParams.update({'font.size': 20})
  ax.set_xlabel("Actual")
  ax.set_ylabel("Predicted ")
  plt.show()
y_pred
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=0)
from sklearn.preprocessing import StandardScaler
x_sc = StandardScaler()
```

```
x_train = x_sc.fit_transform(x_train)

x_test = x_sc.transform(x_test)

mod =  SVR(C=10.0, kernel= 'rbf', coef0 = 0.01, degree = 3, gamma= 'scale')

mod.fit(x_train, y_train)

y_pred = mod.predict(x_test)

y_train.shape

# plotting the signal

from sklearn.metrics import r2_score

pyplot.scatter(y_test, y_pred)

m , b = np.polyfit(y_test, y_pred, 1)



#add linear regression line to scatterplot

plt.plot(y_test, m*y_test+b)

print("R2: ")

r = r2_score(y_test, y_pred)

print(r)

pyplot.xlabel(' Actual')

pyplot.ylabel('Predicted')
```

```
pyplot.title("Prediction")

pyplot.show()

y_pred = mod.predict(x_test)

svrmse= mean_squared_error(y_test, y_pred)

print(svrmse)

rmsesvr=np.sqrt(svrmse)

print(rmsesvr)

from sklearn.cross_decomposition import PLSRegression



pls2 = PLSRegression(n_components=10)

pls2.fit(x_train, y_train)

PLSRegression()

Y_pred = pls2.predict(x_test)

fig,ax = plt.subplots(1)



# plot the data
```

```
ax.scatter(y_test, Y_pred,color="blue", marker="o",)

m, b = np.polyfit(y_test, Y_pred, 1)



#add linear regression line to scatterplot

plt.plot(y_test, m*y_test+b)



from sklearn.metrics import r2_score

pyplot.scatter(y_test, Y_pred)

print("R2: ")

r = r2_score(y_test, Y_pred)

print(r)

meanse= mean_squared_error(y_test, Y_pred)

print(meanse)

rmseplsr=np.sqrt(meanse)

print(rmseplsr)
```

```
from sklearn.ensemble import RandomForestRegressor

from sklearn.datasets import make_regression

make_regression(n_features=4, n_informative=2,random_state=0, shuffle=False)

clf = PLSRegression(n_components=4)



kf = KFold(n_splits=10, shuffle=True, random_state=42)

rmse = 0

for i, (train_index, test_index) in enumerate(kf.split(x_pca)):

    print(f"Fold {i}:")

    #print("TRAIN:", train_index, "TEST:", test_index)

    X_train, X_test = x_pca[train_index], x_pca[test_index]

    y_train, y_test = y[train_index], y[test_index]



    X_train = X_train.astype(np.float64)

    X_test = X_test.astype(np.float64)

    y_test = y_test.astype(np.float64)

    y_train = y_train.astype(np.float64)



    clf.fit(X_train, y_train)
```

```
    ypred = clf.predict(X_test)
    ypred=np.array(ypred).flatten()


    aa=np.array(y_test).flatten()
    mat_plot(aa,ypred)
    rmse = rmse + np.sqrt(mean_squared_error(aa,ypred))
#df = pd.DataFrame(clf.cv_results_)
#df


rmse = rmse /10
print("average RMSE",rmse)
import matplotlib.pyplot as plt
import numpy as np
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor
cf = tree.DecisionTreeRegressor()
from sklearn.metrics import r2_score
r1 = r2_score(y_test, D_pred)
print(r1)
```

```
dtmse=mean_squared_error(y_test, D_pred)

rmsedt=np.sqrt(dtmse)

print(rmsedt)

fig,ax = plt.subplots(1)



# plot the data

ax.scatter(y_test, D_pred,color="blue", marker="o",)

m , b = np.polyfit(y_test, D_pred, 1)



#add linear regression line to scatterplot

plt.plot(y_test, m*y_test+b)

from sklearn.ensemble import RandomForestRegressor

from sklearn.datasets import make_regression

make_regression(n_features=4, n_informative=2,random_state=0, shuffle=False)

clf = DecisionTreeRegressor(random_state=0)



kf = KFold(n_splits=10, shuffle=True, random_state=42)

rmse = 0

for i, (train_index, test_index) in enumerate(kf.split(x_pca)):
```

```
    print(f"Fold {i}:")
    #print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = x_pca[train_index], x_pca[test_index]
    y_train, y_test = y[train_index], y[test_index]

    X_train = X_train.astype(np.float64)
    X_test = X_test.astype(np.float64)
    y_test = y_test.astype(np.float64)
    y_train = y_train.astype(np.float64)

    clf.fit(X_train, y_train)
    ypred = clf.predict(X_test)
    ypred=np.array(ypred).flatten()

    aa=np.array(y_test).flatten()
    mat_plot(aa,ypred)
    rmse = rmse + np.sqrt(mean_squared_error(aa,ypred))
#df = pd.DataFrame(clf.cv_results_)
#df
```

```
rmse = rmse /10

print("average RMSE",rmse)

from sklearn.ensemble import RandomForestRegressor

from sklearn.datasets import make_regression

make_regression(n_features=4, n_informative=2,random_state=0, shuffle=False)

regr = RandomForestRegressor(max_depth=2, random_state=0)

regr.fit(x_train, y_train)

R_pred=regr.predict(x_test)

from sklearn.metrics import r2_score

r = r2_score(y_test, R_pred)

print(r)

fig,ax = plt.subplots(1)



# plot the data

ax.scatter(y_test, R_pred,color="blue", marker="o",)

m, b = np.polyfit(y_test, R_pred, 1)



#add linear regression line to scatterplot
```

```
plt.plot(y_test, m*y_test+b)

from sklearn.ensemble import RandomForestRegressor

from sklearn.datasets import make_regression

make_regression(n_features=4, n_informative=2,random_state=0, shuffle=False)

clf = RandomForestRegressor(max_depth=2, random_state=0)



kf = KFold(n_splits=10, shuffle=True, random_state=42)

rmse = 0

for i, (train_index, test_index) in enumerate(kf.split(x_pca)):

    print(f"Fold {i}:")

    #print("TRAIN:", train_index, "TEST:", test_index)

    X_train, X_test = x_pca[train_index], x_pca[test_index]

    y_train, y_test = y[train_index], y[test_index]



    X_train = X_train.astype(np.float64)

    X_test = X_test.astype(np.float64)

    y_test = y_test.astype(np.float64)

    y_train = y_train.astype(np.float64)
```

```
    clf.fit(X_train, y_train)

    ypred = clf.predict(X_test)

    ypred=np.array(ypred).flatten()



    aa=np.array(y_test).flatten()

    mat_plot(aa,ypred)

    rmse = rmse + np.sqrt(mean_squared_error(aa,ypred))

#df = pd.DataFrame(clf.cv_results_)

#df



rmse = rmse /10

print("average RMSE",rmse)
```