A MACHINE LEARNING PROJECT ON

BUZZING INSIGHTS: A Comprehensive Analysis of Apis Mellifera and its Produce

Asmita Mondal [2348018]
Sayan Pal [2348056]

PROJECT GUIDE: Dr. Rupali Sunil Wagh

Submitted to the Department of Statistics and Data Science in partial fulfilment of the requirements for the degree of M.Sc. Data Science

CERTIFICATE OF AUTHENTICATED WORK

This is to certify that the project report entitled “Buzzing Insights: A Comprehensive Analysis of Apis Mellifera and its Produce” submitted to Department of Statistics and Data Science , Christ (Deemed to be University), Bangalore, Central Campus in partial fulfilment of the requirement for the award of the degree of Master of Science (M. Sc.) is an original work carried out by:

Name	Register No.	Class
ASMITA MONDAL	2348018	3MDS
SAYAN PAL	2348056	3MDS

under the guidance of Dr. Rupali Wagh. The matter embodied in this project is authentic and is genuine work done by the student and has not been submitted whether to this College or to any other Institute for the fulfilment of the requirement of any course of study. All sources have been identified and no part of the paper uses unacknowledged materials.

NAME	RESPONSIBILITIES
Sayan Pal	Honey Production: Data Collection and Compilation, Modelling of Regression Analysis, Comparison of Algorithms for Optimization, Exploratory Analysis, and Review of Related Work
Asmita Mondal	Honey Bee Data: Regression and Random Forest Comparison and Analysis, Dimensionality Reduction, Cluster Analysis and Optimization, CNN Implementation for Images, and Documentation

30-04-2024

……..

Date

ACKNOWLEDGEMENT

We would like to sincerely thank Dr. Rupali Wagh for her invaluable advice and assistance with my project. Her extensive knowledge and skill in the fields of data science and Java have been crucial to the development and success of our project. We are appreciative of the time and effort she has spent on us. We would also like to express our profound gratitude to our peers who have given their time and energy to our project by commenting on the work, taking part in discussions, or contributing their thoughts.

We would also like to thank our Head of Department Dr. Saleema, for giving us the tools and resources we needed to finish our project. We appreciate her never-ending encouragement and support, as well as the environment she helped to foster for creativity and innovation. Additionally, we extend my heartfelt thanks to our parents and supportive friends for their unwavering encouragement and belief in our abilities.

ABSTRACT

This project presents a comprehensive exploration of honeybee ecology and honey production, incorporating image data analysis, predictive modeling, and ecological inference. Through meticulous data collection and analysis, insights into the intricate interactions between honeybees, flowering plants, and honey production dynamics have been gained, with accuracies exceeding 90% for binary as well as multiclass classification models. Notably, image classification techniques using Convolutional Neural Networks facilitated the identification of pollen-carrying honeybees with an accuracy of 91%, while predictive models for honey price estimation, particularly utilizing XGBoost, demonstrated exceptional performance with an R2 score of 0.99. Further, comparative studies were implemented for regression models as well as dimensionality reduction effects for optimization of the algorithms. The findings from this project hold significant implications for biodiversity conservation, sustainable agriculture practices, and the overall sustainability and profitability of the honey production industry. The project was carried out in the Kaggle Notebook powered by the TensorFlow GPU.

Keywords: CNN, Linear Regression, Cross-Validation, PCA, Logistic Regression, KModes, Modelling Comparison

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Bees, particularly the species Apis mellifera, commonly known as honeybees, play a vital role in our ecosystem and are indispensable to human survival. As pollinators, they facilitate the reproduction of flowering plants, ensuring the proliferation of diverse flora and supporting the production of fruits, vegetables, and nuts that comprise a significant portion of our diet. Beyond their agricultural importance, bees contribute to the maintenance of natural habitats and biodiversity, serving as keystone species in various ecosystems.

However, despite their critical ecological role, bees face numerous challenges that threaten their populations worldwide. Factors such as habitat loss, pesticide use, climate change, and the spread of diseases have led to alarming declines in bee populations, resulting in what is often referred to as the "bee crisis" or "pollinator crisis." The decline of bee populations has profound implications for global food security and ecosystem stability, underscoring the urgent need for comprehensive research and conservation efforts to safeguard these invaluable pollinators.

One of the most iconic and cherished products of honeybees is honey, a natural sweetener with a rich history dating back millennia. Honey production not only provides a source of sustenance and income for beekeepers but also holds cultural significance in many societies. However, in recent years, concerns have arisen regarding the purity and quality of commercially available honey. Adulteration, mislabeling, and contamination with artificial sweeteners or other substances have compromised the integrity of honey products, raising concerns about consumer safety and authenticity.

Purpose of the Project

In light of these challenges, there is a growing need for advanced analytical techniques and technologies to assess the health and productivity of bee populations, monitor honey quality, and combat issues of adulteration and fraud in the honey industry. This project aims to address these critical issues by leveraging machine learning and image analysis techniques to analyze bee behavior, classify bee families, and assess the purity and authenticity of honey products. By gaining deeper insights into the dynamics of honeybee populations and honey production processes, we can contribute to the preservation of biodiversity, promote sustainable agricultural practices, and ensure the availability of high-quality honey for future generations.

Objectives of the Project

The project aims to achieve the following objectives:

- Explore the honey production dataset to understand its structure, variables, and trends.
- Predict both honey purity and price using linear regression models, assessing their accuracy and performance.
- Compare and contrast the performance of various regression models (Lasso, Ridge, XGBoost, Gradient Boost) with linear regression for predicting honey purity and price, evaluating their strengths and weaknesses.
- Classify pollen and non-pollen carrying honeybees based on images using Convolutional Neural Networks.
- Compare and contrast the performance of models trained on dimensionality-reduced images with those trained on the original dataset, assessing their predictive accuracy and computational efficiency.
- Apply logistic regression to identify the sex of bees interacting with certain plants and determine the collection method of the bees, analyzing their associations with different variables.

Applicability of the Project

01. **Honey Quality Assurance:** By predicting honey purity and price, honey producers can ensure the quality and authenticity of their honey products. This can help maintain consumer trust in the honey industry, as well as identify potential contamination.
02. **Product Development:** Companies can use the information of prices to optimize product formulations, pricing strategies, and target market segmentation, enhancing competitiveness and profitability in the marketplace.
03. **Precision Agriculture:** The classification of pollen and non-pollen carrying honeybees based on images can be applied in precision agriculture to monitor pollination activity in crop fields to assess pollination levels, and optimize crop yields.
04. **Ecological Monitoring:** Identifying the sex of bees interacting with certain plants and determining the collection method of the bees can provide valuable insights into bee foraging behavior and habitat preferences.

CHAPTER 2: REQUIREMENTS AND SPECIFICATIONS

Problem Definition

To develop a comprehensive analysis framework leveraging machine learning techniques to address key challenges in honeybee ecosystem management, honey production optimization, and quality assurance, enhancing sustainability and productivity in the honey and honey bee industry.

Requirement Specifications

Software:

Python with necessary libraries installed - Kaggle Notebook
TensorFlow GPU accelerated

Hardware:

RAM: 8 GB or above
CPU: 8th Generation Intel® Core™ i3 Processor or above
GPU: 8GB or above
OS: 64-bit
Graphics Card: Integrated

CHAPTER 3: IMPLEMENTATION AND RESULTS

Pollen-Carrying HoneyBee Image Classification

Dataset Description

The 'Honey Bee Pollen' dataset from Kaggle was used as a resource.

It contains high resolution images of individual bees on a ramp. This image dataset was created from videos captured at the entrance of a bee colony in June 2017 at the Bee facility of the Gurabo Agricultural Experimental Station of the University of Puerto Rico.

- **images/** contains images for pollen bearing and no pollen bearing honey bees.
    - The prefix of the image names define their class: e.g. NP1268-15r.jpg for non-pollen and P7797-103r.jpg for pollen bearing bees. The numbers correspond to frame and item number respectively, you need to be careful that they are not numbered sequentially.
- **Read-skimage.ipynb** is a Jupyter notebook for simple script to load the data and create the dataset using skimage library.
- **.csv** file contains the corresponding labels where 1 indicates the honeybee carries pollen and 0 indicates the absence of pollen.
- A total of **714** images are present in the dataset.

Models Used With Justification

- PCA:
    - Dimensionality Reduction: The Honeybee image data exhibited high dimensionality due to the large number of pixels in each image. PCA enabled us to reduce the dimensionality of the image data while retaining most of the important information, improving computational efficiency and mitigating the curse of dimensionality.
    - Feature Extraction: PCA extracted the principal components of variation in the image data, helping identify key features relevant to distinguishing pollen-carrying and non-pollen-carrying honeybee images.
    - Noise Reduction: PCA filtered out noise or irrelevant information present in the image data, enhancing the signal-to-noise ratio and improving the performance of downstream classification algorithms.

- CNN:
    - Feature Learning: CNNs learned hierarchical representations of visual features directly from raw pixel values, eliminating the need for manual feature engineering and adaptively learning relevant features from the data.
    - Spatial Hierarchical Representation: CNNs preserved the spatial structure of the input images throughout the network architecture, capturing local patterns and global context simultaneously, which was crucial for accurate classification of pollen-carrying honeybee images.
    - Robustness to Variability: CNNs were robust to variations in lighting conditions, backgrounds, and orientations, thus achieving robust performance across different environmental conditions and camera angles.

Procedure

1. The image data was first mapped with the csv file to get binary labels for image classification. Unnecessary columns were dropped.
2. The images were then **flattened** because PCA (Principal Component Analysis) requires the input data to be in the form of a two-dimensional array. Each row of this array represented a sample (in this case, an image), and each column represented a feature (pixel intensity). By flattening the images, we transformed the multi-dimensional image data into a one-dimensional array, ensuring compatibility with PCA. This preprocessing step simplified the data representation and enhanced computational efficiency, enabling PCA to identify the principal components effectively and reduce the dimensionality of the feature space.
3. The number of components before and after Principal Component Analysis were observed to reduce from **162000** to **272**. A **95% variance** capture was requested and the output was as follows:

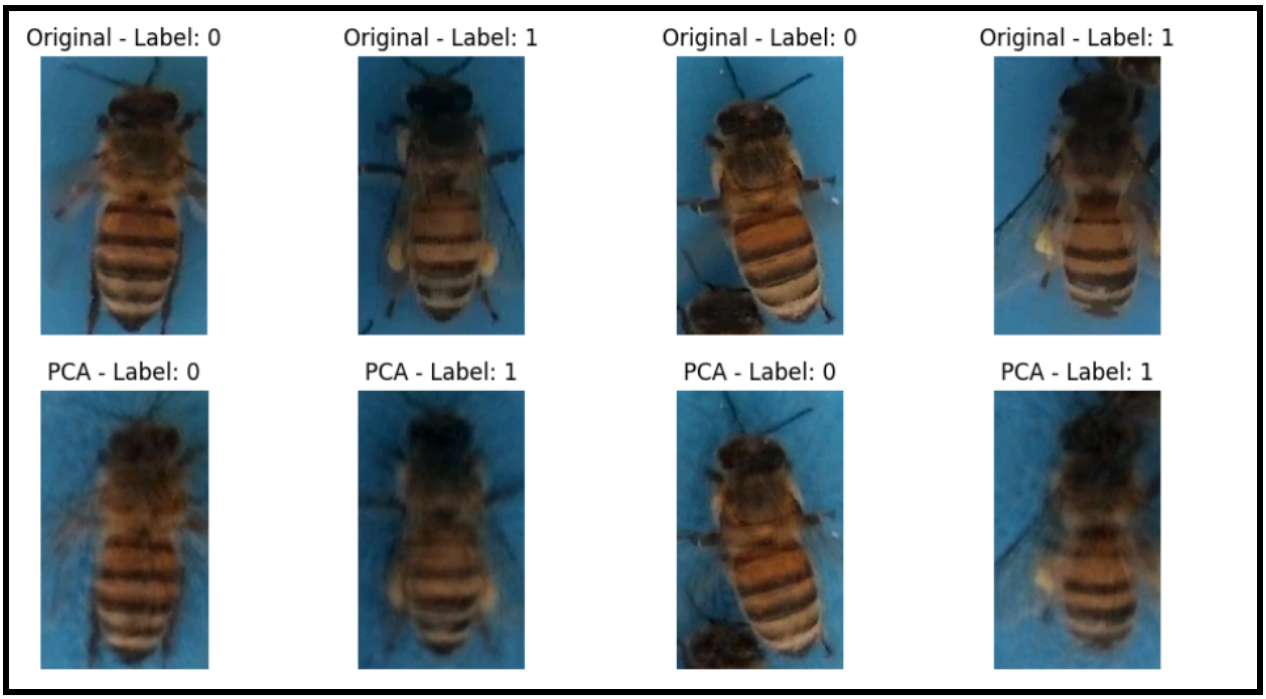
```
# Get the number of features (components) before PCA
num_components_before_pca = images_flattened.shape[1]
print("Number of components before PCA:", num_components_before_pca)
```

```
Number of components before PCA: 162000
```

```
# Get the number of components after PCA transformation
num_components = pca.n_components_
print("Number of components after PCA transformation:", num_components)
```

```
Number of components after PCA transformation: 272
```

4. Images before and after PCA was displayed:



5. The data was split in the ratio 7:3 for training and testing. The following CNN model was implemented on the data without PCA implementation:

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 298, 178, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 149, 89, 32)	0
conv2d_9 (Conv2D)	(None, 147, 87, 64)	18,496
max_pooling2d_9 (MaxPooling2D)	(None, 73, 43, 64)	0
conv2d_10 (Conv2D)	(None, 71, 41, 128)	73,856
max_pooling2d_10 (MaxPooling2D)	(None, 35, 20, 128)	0
conv2d_11 (Conv2D)	(None, 33, 18, 128)	147,584
max_pooling2d_11 (MaxPooling2D)	(None, 16, 9, 128)	0
flatten_2 (Flatten)	(None, 18432)	0
dense_8 (Dense)	(None, 512)	9,437,696
dense_9 (Dense)	(None, 1)	513

Total params: 9,679,041 (36.92 MB)
Trainable params: 9,679,041 (36.92 MB)
Non-trainable params: 0 (0.00 B)

- **Convolutional Layers (Conv2D):** Convolutional layers applied learnable filters to the input images, extracting spatial features such as edges, textures, and patterns. In this model, three Conv2D layers with increasing depth (32, 64, 128) were utilized. These layers progressively learned more abstract and higher-level features as information propagated deeper into the network. The activation function 'relu' (Rectified Linear Unit) introduced non-linearity, allowing the model to capture complex relationships within the data.
- **Pooling Layers (MaxPooling2D):** Pooling layers reduced the spatial dimensions of the feature maps generated by the convolutional layers, reducing computational complexity and controlling overfitting. MaxPooling2D layers down-sampled the feature maps by retaining the maximum value within a defined window (2x2). This preserved the most salient features while discarding less relevant information. By applying MaxPooling2D after each Conv2D layer, the model maintained spatial hierarchy while progressively reducing the spatial dimensions of the feature maps.
- **Flattening Layer:** The flattening layer transformed the output of the convolutional layers into a one-dimensional vector, which served as input to the subsequent fully connected layers. This step was essential for transitioning from the spatial hierarchy of features learned by the convolutional layers to the dense representation required for classification.
- **Fully Connected Layers (Dense):** Fully connected layers (Dense layers) captured complex relationships between the extracted features and the target variable. In this model, two Dense layers with 512 units each were employed, providing the capacity to capture intricate feature interactions and hierarchies. The 'relu' activation function introduced non-linearity, allowing the model to learn complex patterns in the data.
- **Output Layer:** The output layer consisted of a single neuron with a sigmoid activation function. The sigmoid activation function produced a probability score indicating the likelihood of an image belonging to the pollen-carrying class. Utilizing a sigmoid activation function facilitated binary classification, where the model predicted whether an image contained a pollen-carrying honeybee or not.

6. The model was compiled using the **Adam** optimizer and binary cross-entropy loss function, with accuracy as the evaluation metric. The Adam optimizer was chosen for its ability to adaptively adjust learning rates for each parameter, making it suitable for training deep neural networks efficiently. The model was trained for **10 epochs** with a batch size of **32**, using the training data with a validation split of **0.2** to monitor model performance during training and prevent overfitting. Accuracy was calculated.
7. The next model was applied on the PCA processed image data and the same optimizer, epochs and batch sizes were applied.

Model: "sequential_5"

Layer (type)	Output Shape	Param #
dense_10 (Dense)	(None, 512)	139,776
dense_11 (Dense)	(None, 1)	513

Total params: 140,289 (548.00 KB)
Trainable params: 140,289 (548.00 KB)
Non-trainable params: 0 (0.00 B)

- In the implemented PCA-CNN model, a departure from the conventional CNN architecture was observed, primarily in its input representation and architectural composition. Unlike the traditional CNN model, which processed raw image data directly through convolutional and pooling layers, the PCA-CNN model took as input the transformed features obtained from Principal Component Analysis (PCA).
- By reducing the dimensionality of the input data, PCA condensed the information captured in the original images into a more compact representation, facilitating computational efficiency and potentially mitigating overfitting.
- Consequently, the PCA-CNN model omitted convolutional and pooling layers, utilizing only fully connected layers to learn complex relationships between the PCA-transformed features and the target variable, thereby streamlining the model architecture and enhancing training efficiency.

8. The performance of both models were compared.

Results

1. The test accuracy of the model without PCA was **91.63%** which shows an incredible learning rate and performance by the model.
2. The test accuracy of the model with PCA was **84.18%** which is also remarkable.
3. The time taken for the model to train on data without PCA was approximately **347.88 seconds.**
4. The time taken for the model to train on data with PCA was **2.53 seconds** approximately.

The slight decrease in test accuracy observed for the model with PCA preprocessing, achieving 84.18%, compared to the model without PCA, which reached 91.63%, can be attributed to several factors. Firstly, PCA involves dimensionality reduction, which compresses the original high-dimensional image data into a lower-dimensional space by capturing the most important variations. While this reduction in dimensionality can lead to computational efficiency and potentially mitigate overfitting, it may also result in a loss of some information.

This loss of information could impact the model's ability to discriminate between pollen-carrying and non-pollen-carrying honeybee images with the same level of accuracy as the model trained on raw image data. Secondly, PCA operates on the assumption that the most significant variations in the data can be captured by linear combinations of the original features. However, complex and nonlinear relationships present in image data may not be fully captured by linear transformations alone. As a result, the PCA-transformed features may not fully represent the intricate patterns and structures present in the original images, leading to a slight decrease in classification accuracy. However, it was found that the training time for PCA was indeed much less compared to the other model. This fulfils the purpose of dimensionality reduction with respect to time complexity enhancement.

```
# Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 4s 535ms/step - accuracy: 0.9312 - loss: 0.3100
Test accuracy: 0.9162790775299072
```

```
# Evaluate the PCA model
test_loss_pca, test_acc_pca = model_pca.evaluate(x_test_pca, y_test_pca)
print('Test accuracy (PCA model):', test_acc_pca)
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8573 - loss: 0.3704
Test accuracy (PCA model): 0.8418604731559753
```

```
# Display the comparison
print("Training time without PCA:", training_time_without_pca, "seconds")
print("Training time with PCA:", training_time_with_pca, "seconds")
```

```
Training time without PCA: 347.8785936832428 seconds
Training time with PCA: 2.5275919437408447 seconds
```

Link to Notebook

https://www.kaggle.com/code/asmita2001/pollen-carrying-honeybee-image-classification/edit

Honey Price and Purity Prediction

Dataset Description

The 'Predict Price and Purity of Honey' dataset was taken from Kaggle. The data is as follows:

CS (Color Score): Represents the color score of the honey sample, ranging from 1.0 to 10.0. Lower values indicate a lighter color, while higher values indicate a darker color.

Density: Represents the density of the honey sample in grams per cubic centimeter at 25°C, ranging from 1.21 to 1.86.

WC (Water Content): Represents the water content in the honey sample, ranging from 12.0% to 25.0%.

pH: Represents the pH level of the honey sample, ranging from 2.50 to 7.50.

EC (Electrical Conductivity): Represents the electrical conductivity of the honey sample in milliSiemens per centimeter.

F (Fructose Level): Represents the fructose level of the honey sample, ranging from 20 to 50.

G (Glucose Level): Represents the glucose level of the honey sample, ranging from 20 to 45.

Pollen_analysis: Represents the floral source of the honey sample. Possible values include Clover, Wildflower, Orange Blossom, Alfalfa, Acacia, Lavender, Eucalyptus, Buckwheat, Manuka, Sage, Sunflower, Borage, Rosemary, Thyme, Heather, Tupelo, Blueberry, Chestnut, and Avocado.

Viscosity: Represents the viscosity of the honey sample in centipoise, ranging from 1500 to 10000. Viscosity values between 2500 and 9500 are considered optimal for purity.

Purity: The target variable represents the purity of the honey sample, ranging from 0.01 to 1.00.

Price: The calculated price of the honey.

SHAPE: (247903, 11)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 247903 entries, 0 to 247902
Data columns (total 11 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   CS               247903 non-null  float64
 1   Density          247903 non-null  float64
 2   WC               247903 non-null  float64
 3   pH               247903 non-null  float64
 4   EC               247903 non-null  float64
 5   F                247903 non-null  float64
 6   G                247903 non-null  float64
 7   Pollen_analysis  247903 non-null  object
 8   Viscosity        247903 non-null  float64
 9   Purity           247903 non-null  float64
 10  Price            247903 non-null  float64
dtypes: float64(10), object(1)
memory usage: 20.8+ MB
```

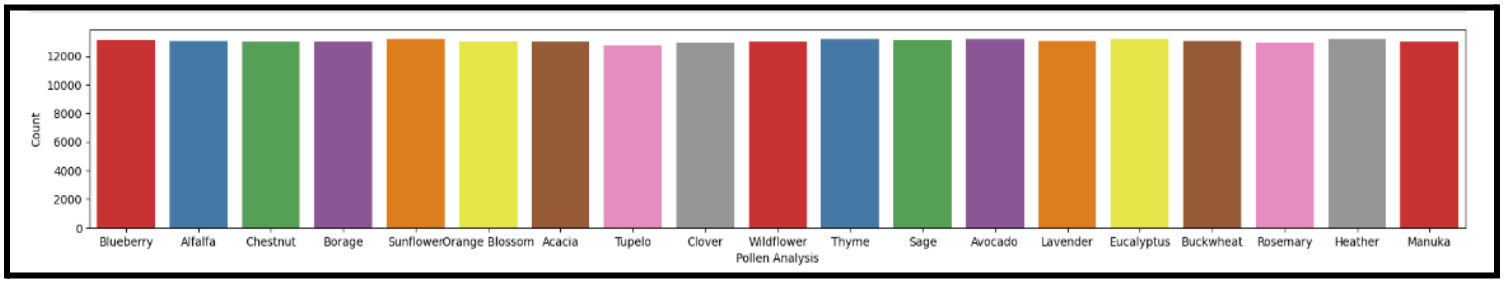
Procedure

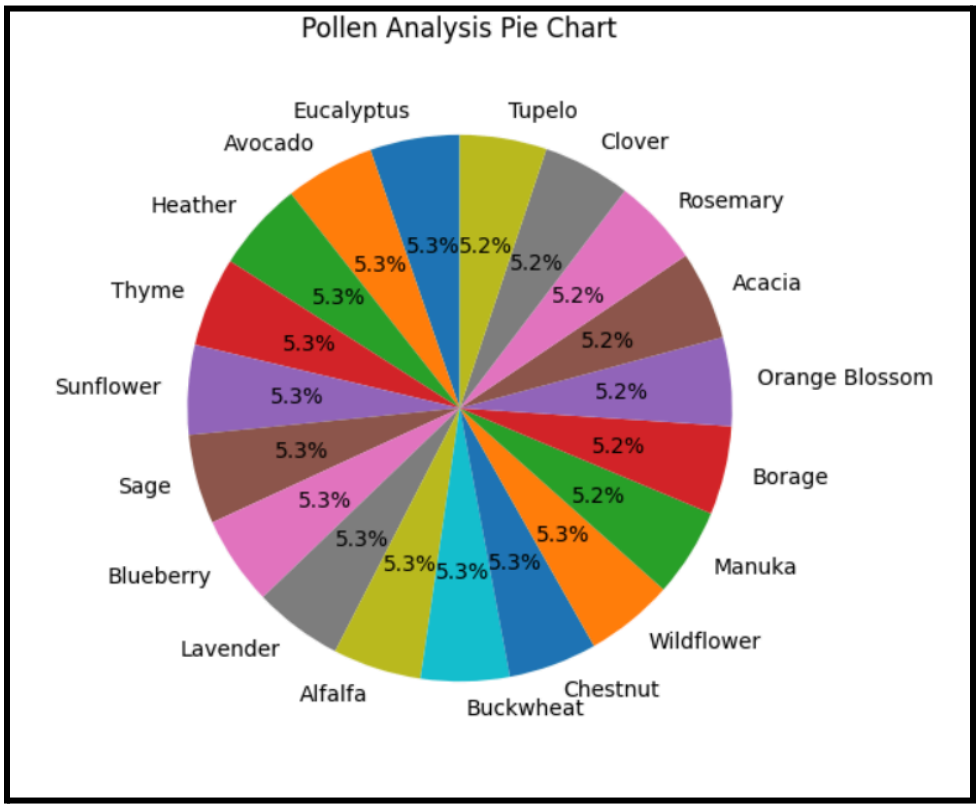
- Initial exploration of the dataset was conducted to identify and address any missing values or NaN values, ensuring data integrity.
- Exploratory Data Analysis (EDA) techniques, including histograms, box plots, and scatter plots, were employed to understand the distribution of variables and explore correlations between them.
- Correlations between variables, particularly honey purity, honey price, and other features, were examined to identify potential relationships useful for predictive modeling. Outliers, if present, were detected using statistical methods or visualization techniques, and their impact on the analysis was evaluated.
- The dataset was split into training and testing sets using an 80%-20% ratio, with the training set used for model training and the testing set reserved for evaluation.
- Standardization using the StandardScaler was applied to ensure all features were on the same scale, which is essential for models sensitive to feature scales, such as linear regression.
- Regression models including
    - Linear Regression,
    - Ridge Regression,
    - Lasso Regression,
    - XGBoost Regressor, and
    - Gradient Boost Regressor

were constructed to predict honey purity and price.

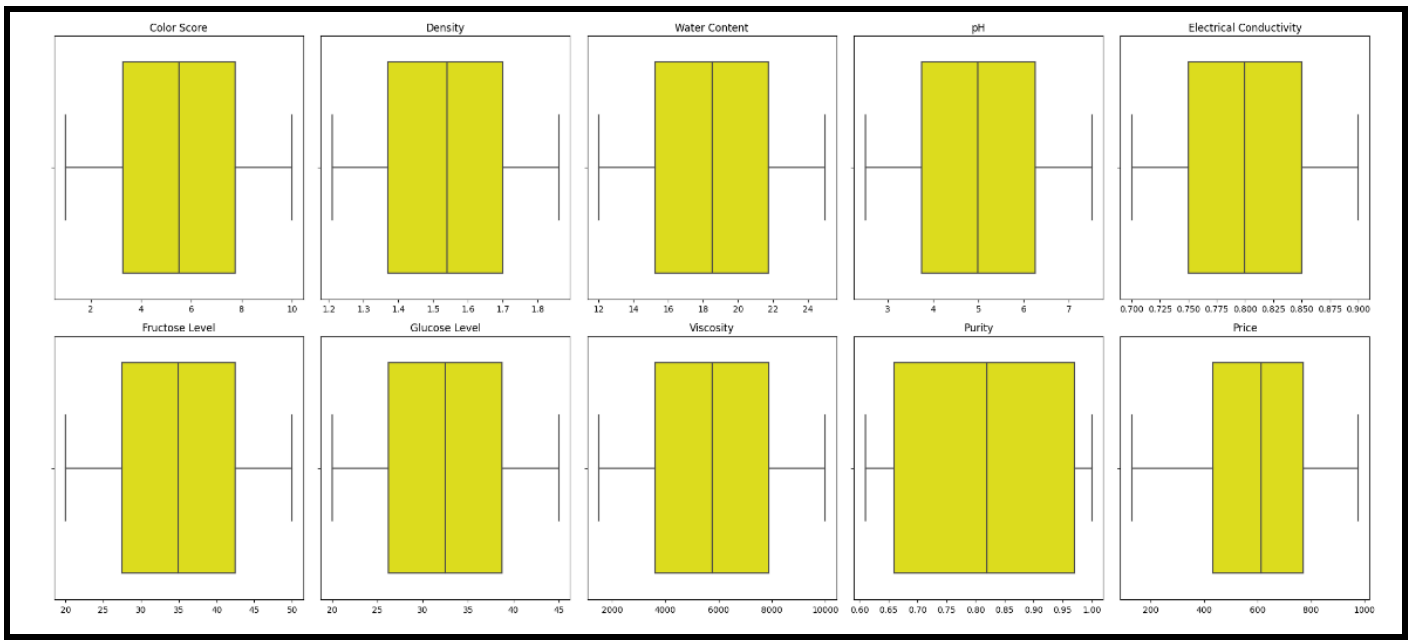
- Performance of each model was evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) score. Model comparison was conducted to identify the best-performing model for predicting honey purity and price.

Exploratory Data Analysis

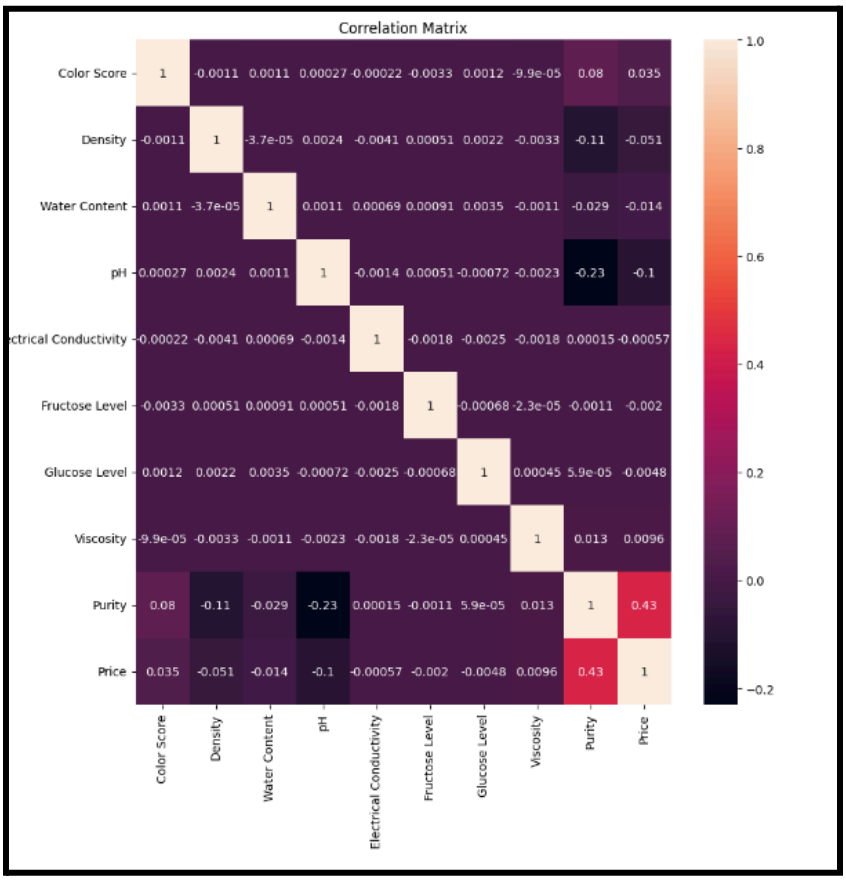




All the floral sources of honey were almost equally contributing in this dataset, so we concluded the data is balanced with respect to pollen analysis.



No outliers were present in the dataset.



Price and other variables:

- **Density**: There was a slight negative correlation (-0.0505), indicating that as density increases, price tends to decrease slightly.
- **Water Content**: A negligible correlation was present (-0.0144), suggesting little relationship between water content and price.
- **pH:** A moderate negative correlation was present(-0.1007), indicating that as pH increases, price tends to decrease.
- **Electrical Conductivity**: Very weak correlation was visible(-0.0006), implying almost no relationship between electrical conductivity and price.
- **Fructose Level**: A negligible negative correlation was displayed(-0.002), suggesting little influence of fructose level on price.
- **Glucose Level**: A very weak correlation (0.0001), indicated almost no relationship between glucose level and price.
- **Viscosity**: A very weak positive correlation (0.0096), implied minimal influence of viscosity on price.

Price and Purity:

There was a moderate positive correlation (0.4326), indicating a significant relationship between price and purity. This suggested that as purity increases, price tends to increase as well. The variables 'Electrical Conductivity', 'Fructose Level', 'Glucose Level' do not have much relation with the targets.

	Color Score	Density	Water Content	pH	Electrical Conductivity	Fructose Level	Glucose Level	Viscosity	Purity	Price
Color Score	1.000000	-0.001099	0.001148	0.000267	-0.000215	-0.003287	0.001217	-0.000099	0.079770	0.035166
Density	-0.001099	1.000000	-0.000037	0.002389	-0.004113	0.000515	0.002244	-0.003295	-0.108834	-0.050518
Water Content	0.001148	-0.000037	1.000000	0.001068	0.000690	0.000912	0.003517	-0.001088	-0.028894	-0.014381
pH	0.000267	0.002389	0.001068	1.000000	-0.001400	0.000511	-0.000725	-0.002347	-0.230855	-0.100714
Electrical Conductivity	-0.000215	-0.004113	0.000690	-0.001400	1.000000	-0.001773	-0.002520	-0.001755	0.000151	-0.000571
Fructose Level	-0.003287	0.000515	0.000912	0.000511	-0.001773	1.000000	-0.000683	-0.000023	-0.001149	-0.002041
Glucose Level	0.001217	0.002244	0.003517	-0.000725	-0.002520	-0.000683	1.000000	0.000453	0.000059	-0.004815
Viscosity	-0.000099	-0.003295	-0.001088	-0.002347	-0.001755	-0.000023	0.000453	1.000000	0.012572	0.009632
Purity	0.079770	-0.108834	-0.028894	-0.230855	0.000151	-0.001149	0.000059	0.012572	1.000000	0.432581
Price	0.035166	-0.050518	-0.014381	-0.100714	-0.000571	-0.002041	-0.004815	0.009632	0.432581	1.000000

Models Used with Justification

For the prediction of honey purity and price, a selection of regression models were employed:

Linear Regression: Utilized as a foundational model, it assumes a linear relationship between independent variables and the target variable, offering interpretability and computational efficiency ideal for initial regression tasks.

Ridge Regression: Acting as a regularized extension of linear regression, it addresses multicollinearity and overfitting through a penalty term, proving beneficial for handling correlated predictors and high-dimensional data.

Lasso Regression: Similar to ridge regression but employing an L1 penalty, it excels in feature selection by shrinking coefficients of less relevant features to zero, effectively simplifying the model.

XGBoost Regressor: Embraced for its ensemble learning approach, it sequentially constructs decision trees, correcting errors from previous iterations. Renowned for its speed, performance, and ability to capture complex relationships, it's a favored choice for non-linear data.

Gradient Boosting Regressor: Another ensemble learning technique, it combines weak learners (typically decision trees) sequentially to build a robust predictive model. Known for its high accuracy and resistance to overfitting, it is a preferred option for predictive modeling.

Using **regression models** to predict honey price and purity based on traits such as color score (CS), density, water content (WC), pH level, electrical conductivity (EC), fructose (F) and glucose (G), pollen analysis and viscosity are target variables due to their continuous nature (price and purity) a logical choice.These models were selected due to their compatibility with continuous variables, essential for predicting honey purity and price accurately.

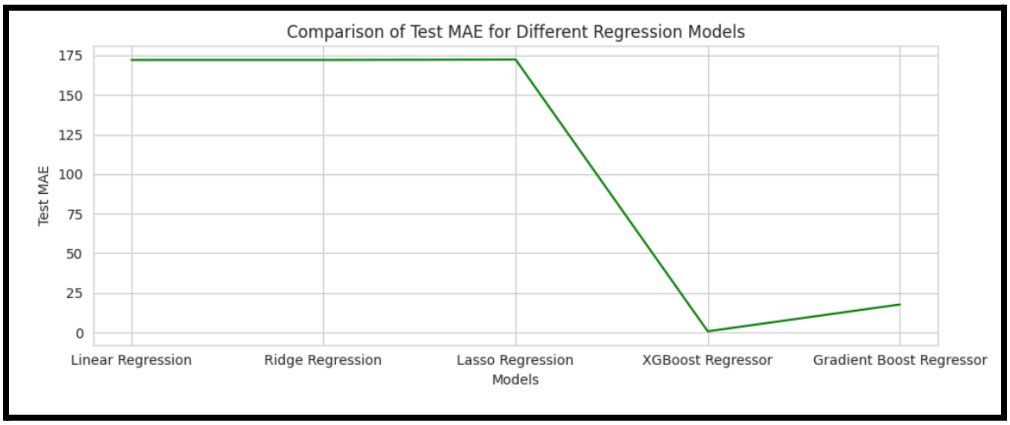
Moreover, being supervised learning techniques, they leverage labeled data to train predictive models, making them suitable for regression tasks where the outcome variable is known and continuous. This ensures that the models can learn from existing data patterns to make predictions on unseen data, a crucial requirement for accurate honey purity and price estimation. In addition, the regression models provide an overview of the quantitative effect of each characteristic on the target variables, allowing us to understand which factors have the most significant effect on the price and purity of honey.

Results

Honey Price Prediction

The following results were obtained:

	Model	train MAE	test MAE	train MSE	test MSE	train R2 Score	test R2 Score
0	Linear Regression	172.728283	172.181405	44240.939007	43978.855889	0.190055	0.191835
1	Ridge Regression	172.728297	172.181418	44240.939007	43978.856268	0.190055	0.191835
2	Lasso Regression	172.943730	172.407050	44244.619845	43985.749149	0.189988	0.191708
3	XGBoost Regressor	0.726836	0.736316	1.257651	1.294957	0.999977	0.999976
4	Gradient Boost Regressor	17.743692	17.678976	555.902037	550.255409	0.989823	0.989888



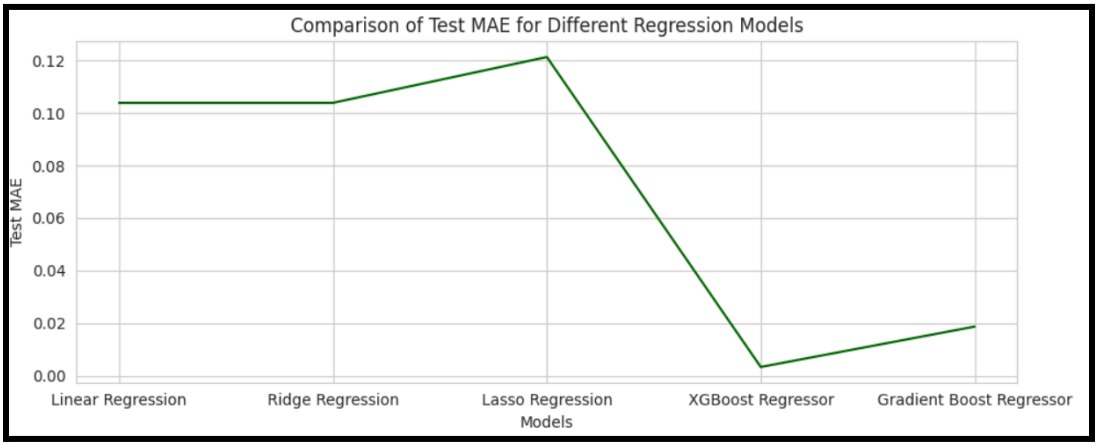
Our analysis revealed that boosting models excelled amongst the five regression models evaluated. Notably, the XGBoost Regressor demonstrated exceptional performance, achieving a testing MAE (Mean Absolute Error) of 0.011929, surpassing the Gradient Boost Regressor's testing MAE of 0.012565. This translates to impressive accuracy in both training and testing phases, solidifying XGBoost Regressor as the optimal model for this task.

Graphically Representing the XGBoost Evaluation metrics:

[22]

Honey Purity Prediction

	Model	train MAE	test MAE	train MSE	test MSE	train R2 Score	test R2 Score
0	Linear Regression	0.104445	0.103828	0.014879	0.014778	0.235132	0.237008
1	Ridge Regression	0.104445	0.103828	0.014879	0.014778	0.235132	0.237008
2	Lasso Regression	0.121783	0.121243	0.019453	0.019371	0.000000	-0.000112
3	XGBoost Regressor	0.003301	0.003413	0.000062	0.000071	0.996834	0.996326
4	Gradient Boost Regressor	0.018926	0.018770	0.000764	0.000754	0.960731	0.961060



Our analysis once again revealed that boosting models excelled amongst the five regression models evaluated. Notably, the XGBoost Regressor demonstrated exceptional performance, achieving a testing MAE (Mean Absolute Error) of 0.003413, surpassing the Gradient Boost Regressor's testing MAE of 0.018926.

Link to Notebook

https://www.kaggle.com/code/sayanpal1234/honey-price-purity/edit

Bee-Plant Interaction Analysis

Dataset Description

The 'Bee-Plant Interaction Dataset' was used for the analysis. This data threw light upon the grazing patterns along with the type of plants the bees interact with. It also gives an idea of which sex of the bee is more interactive with plants and which sex is dormant.

- The data is a CSV file that contains information on **2,691** bee individuals sorted into **183** species representing **55** genera found in five bee families: Halictidae (74), Apidae (63), Megachilidae (40), Andrenidae (5), and Colletidae (1).
- Additionally, the dataset comprises **112** species of flowering plants, along with their families and life forms (i.e., herb, shrub, or tree), identified as potential forage resources for bees.
- Environmental variables such as mean annual temperature (in degrees Celsius, °C) are included, obtained using DS1921G Thermochron iButton data logger (± 0.5°C resolution; Maxim Integrated Products, USA), placed at 2 m height above the ground to record ambient temperature.
- The dataset also contains information on the method used for collecting bees, date of collection, identifier (the person who identified the bee specimen), location (GPS coordinates), grazing intensity (low, moderate, or high livestock grazing intensity), and elevation (m asl), obtained by Garmin GPSMAP 64s GPS receiver (USA) with an accuracy of ± 3 m.

This dataset is part of the three-year Bee-Pollinator Monitoring Project, Tanzania (2017 – 2020).

The data were acquired via plot-based field survey. This involved capturing bees using a pan trap and hand net in standardized random walks. Data was then entered into computer Microsoft Excel 2016 before being made publicly available on the Fig share data repository.

Models Used with Justification

1. **Logistic Regression for Binary Classification:** Logistic regression is a suitable choice for binary classification tasks, where the target variable has two possible outcomes (e.g., bee sex, collection method). It provides interpretable results by estimating the probability of each outcome and making predictions based on a threshold value. Logistic regression is computationally efficient and requires minimal tuning of hyperparameters, making it well-suited for binary classification tasks with relatively simple decision boundaries. It was used for classifying the sex of the bee which had two categories (male and female) as well as a collection method of bees which again had two values (sweep net and pan trap).

2. **Decision Tree and Random Forest for Multiclass Classification:** Decision trees and random forests are commonly used for multiclass classification tasks, where the target variable has more than two categories (e.g., bee family). Decision trees partition the feature space into regions based on simple rules, making them intuitive and easy to interpret. Random forests improve upon decision trees by aggregating the predictions of multiple trees, reducing overfitting and increasing robustness to noisy data. Both decision trees and random forests can handle categorical features naturally, making them suitable for datasets with mixed data types.

3. **KModes for Categorical Clustering:** KModes clustering is specifically designed for datasets containing categorical variables, making it suitable for clustering analysis with categorical data like the season of data collection and grazing intensity of bees. KModes identifies clusters based on the modes (most frequent values) of categorical variables, allowing for the discovery of natural groupings within the data. It is robust to outliers and can handle high-dimensional categorical data efficiently.

4. **Cross-Validation for the Smaller Dataset:** Cross-validation is essential for model evaluation, especially when dealing with smaller datasets like the one with only 856 rows. Cross-validation helps to assess the generalization performance of the models by estimating how well they would perform on unseen data. With a smaller dataset, cross-validation provides a more reliable estimate of model performance, reducing the risk of overfitting and providing a more accurate assessment of model efficacy.

Procedure

- The dataset was first preprocessed by dropping columns unnecessary for categorical analysis. The final columns were as follows:

	species_name	family	sex	identifier	forage_resource	family.1	life_form	collection_method	grazing_intensity	season
0	Acunomia senticosa (Vachal 1897)	Halictidae	Female	A. Pauly	Solanum incanum L.	Solanaceae	Perennial shrub	sweep net	low grazing intensity	long_rain
1	Acunomia somalica (Friese 1908)	Halictidae	Female	A. Pauly	Tephrosia densiflora Hook.f.	Fabaceae	Perennial herb	sweep net	medium grazing intensity	long_rain
2	Acunomia theryi (Gribodo, 1894)	Halictidae	Female	A. Pauly	NaN	NaN	NaN	pan trap	medium grazing intensity	long_rain
3	Acunomia theryi (Gribodo, 1894)	Halictidae	Female	A. Pauly	Solanum incanum L.	Solanaceae	Perennial shrub	sweep net	medium grazing intensity	long_rain
4	Afranthidium indet1	Megachilidae	Female	J. Lasway	On air	NaN	NaN	sweep net	medium grazing intensity	long_rain
...	...	...	...	...	...	...	...	...	...	...
2686	Zonalictus nomioides (Friese, 1905)	Halictidae	Female	A. Pauly	Schkuhria pinnata (Lam.) Kuntze	Asteraceae	Annual herb	sweep net	high grazing intensity	long_rain
2687	Zonalictus nomioides (Friese, 1905)	Halictidae	Female	A. Pauly	Gutenbergia cordifolia Benth. ex Oliv.	Asteraceae	Annual herb	sweep net	medium grazing intensity	long_rain
2688	Zonalictus nomioides (Friese, 1905)	Halictidae	Female	A. Pauly	On air	NaN	NaN	sweep net	medium grazing intensity	long_rain
2689	Zonalictus nomioides (Friese, 1905)	Halictidae	Female	A. Pauly	Solanum lycopersicum L.	Solanaceae	Annual herb	sweep net	medium grazing intensity	long_rain
2690	Zonalictus nomioides (Friese, 1905)	Halictidae	Female	A. Pauly	NaN	NaN	NaN	pan trap	low grazing intensity	long_rain

2691 rows × 10 columns

- Three copies of the dataset were created to facilitate comparative analysis, each serving a distinct purpose in handling missing values and conducting modeling.
- In the first copy, missing values were handled without removal, and label encoding was applied, treating NaN as a new category.
- The second copy addressed missing values through **mode imputation**, replacing missing entries with the most frequent values.
- The third copy involved removing rows with missing values, resulting in a dataset comprising 856 rows for subsequent modeling.
- **Logistic regression** models were employed to predict bee sex based on species family, plant life form, and season of data collection. The dataset was split into training and testing sets in a 7:3 ratio, utilizing the 'liblinear' solver for logistic regression.
- Logistic regression was also used to classify the collection method of bees based on forage resource, plant life form, and plant family. The dataset was again split into training and testing sets in a 7:3 ratio, utilizing the 'liblinear' solver for logistic regression. **LIBLINEAR** is known for its efficiency in handling large datasets with many features. It employs various optimization techniques to accelerate the convergence of the optimization process, making it suitable for large-scale problems.
- Decision tree and random forest classifiers were utilized to classify bee families based on species name.
- Analyses were repeated for datasets with mode imputation and missing values removed to compare performance across preprocessing approaches.
- For the dataframe which had missing value rows removed, on account of the dataset becoming small, analysis was performed both with and without cross validation. The **5 fold cross validation** model was implemented.
- **KModes** clustering analysis was performed to gain insights into the data based on the season of data collection and grazing intensity of bees, evaluated using elbow method and silhouette scores.

Results

Models:

1. For all three datasets with missing value imputation, without missing value imputation and removal of missing values- the accuracy of the Logistic Regression model to classify the **sex of the bee** was **95%**. This implies that the effect of imputation or removal did not affect the overall model building. In other words, the dataset was extremely well formed to work even without much preprocessing required. The mean cross validation score was **0.94**. The individual fold scores were **[0.94805195 0.94981413 0.94981413 0.94795539 0.89962825].**
2. The same pattern was seen for **collection method** classification using Logistic Regression. The training data achieved an accuracy of **98%** while the testing data

achieved an accuracy of **97%.** The model displayed excellent performance without any overfitting or underfitting. The mean cross validation score was **0.98**. The individual fold scores were **[0.97588126 0.98327138 0.99442379 0.96096654 0.97769517].**

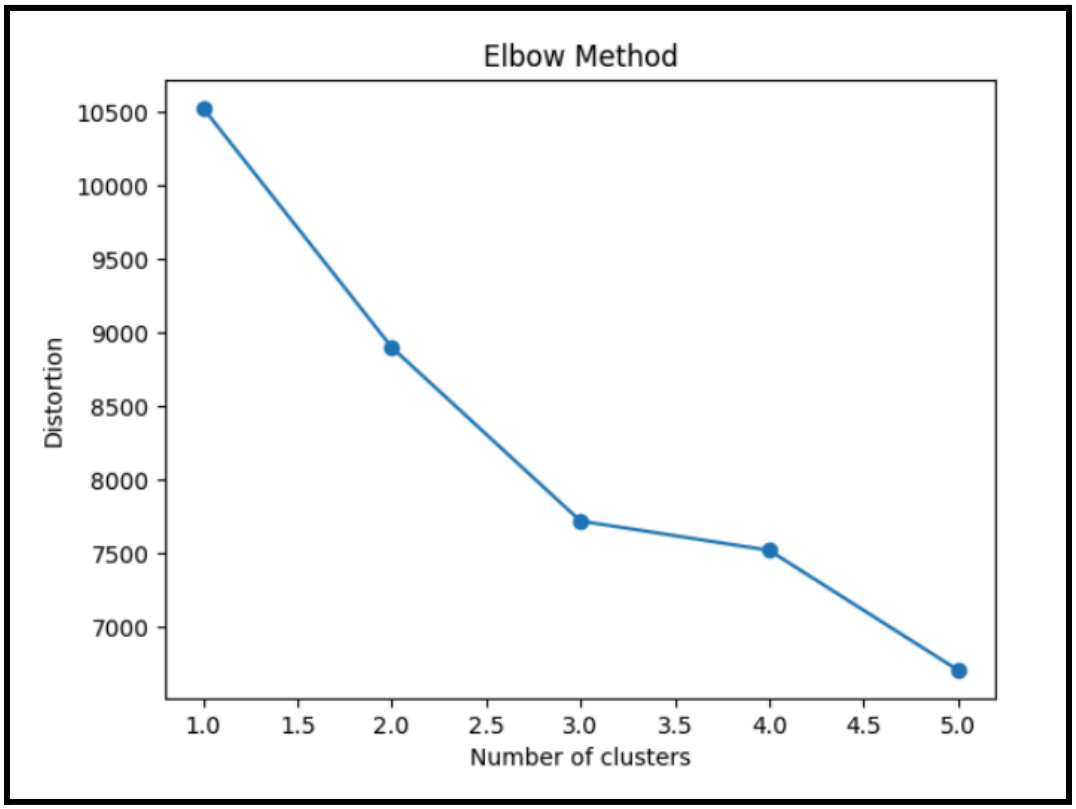
3. In a similar manner, to classify **bee families based on species name**, the training and testing data for all the three copies achieved a remarkable accuracy of **99.50%**. This is because the family of bee species becomes obvious when the species name is known. Both Decision Tree and Random Forest classifiers performed the same here. The cross-validation accuracy for both the models was **77.58%**.

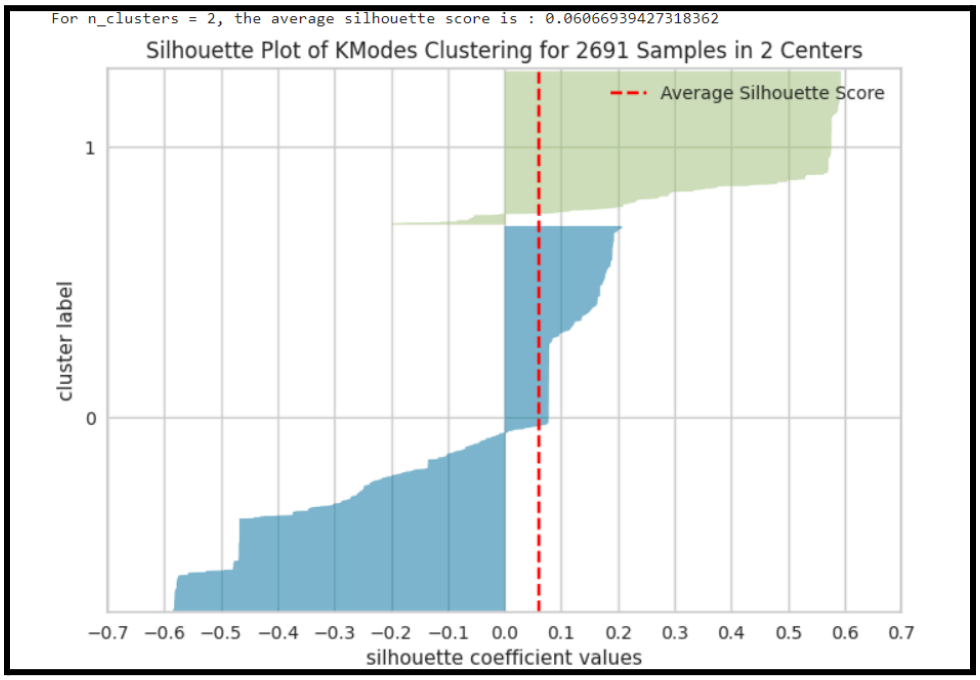
Thus, the features used for modeling were so highly informative and robust, meaning they contain sufficient discriminatory power to accurately predict the target variable , that they performed well even in the presence of missing values.

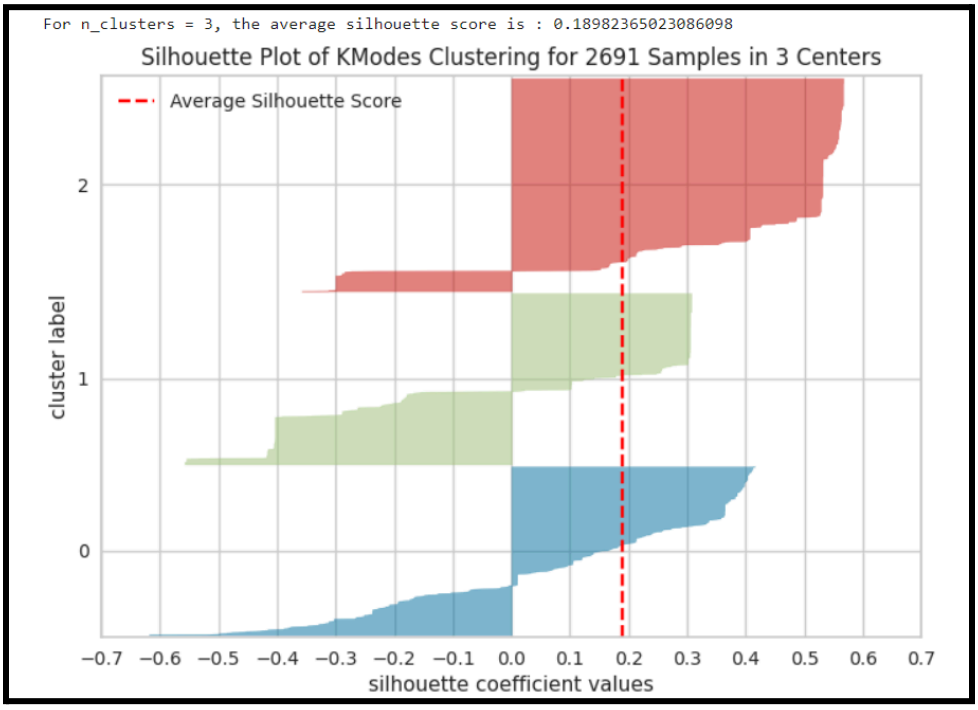
KModes Analysis:

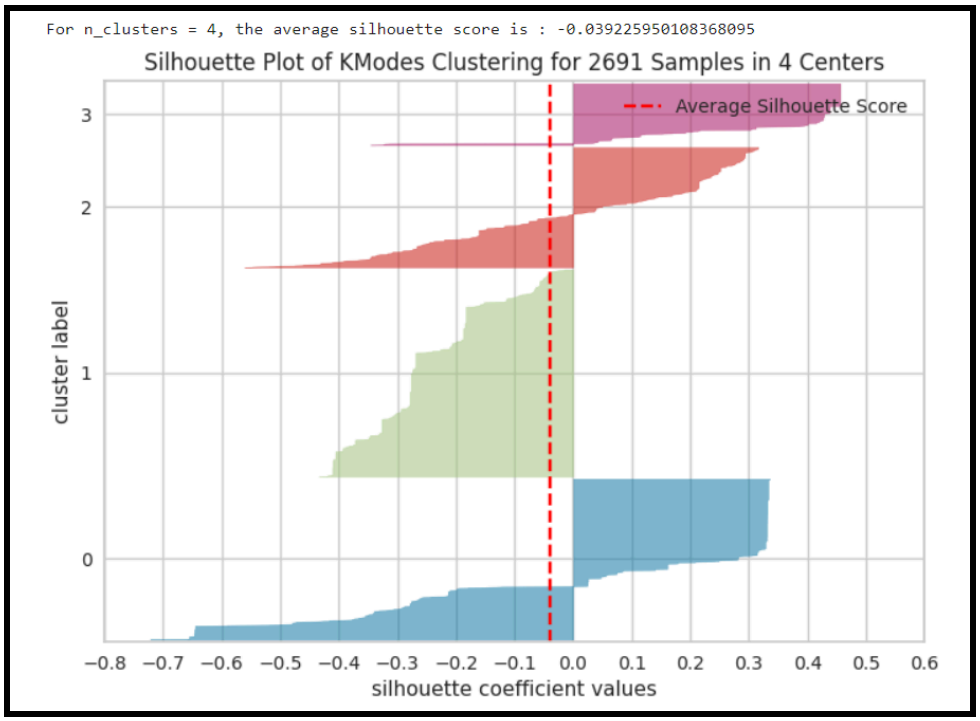
1. For **grazing intensity clustering**, the KModes was run from a range of 1 to 5 clusters. The Elbow method showed inertia slow-down at **k=3**. In other words, the graph formed an elbow at k=3. This result was verified using the silhouette score which was the highest for 3 clusters (**0.1898**).

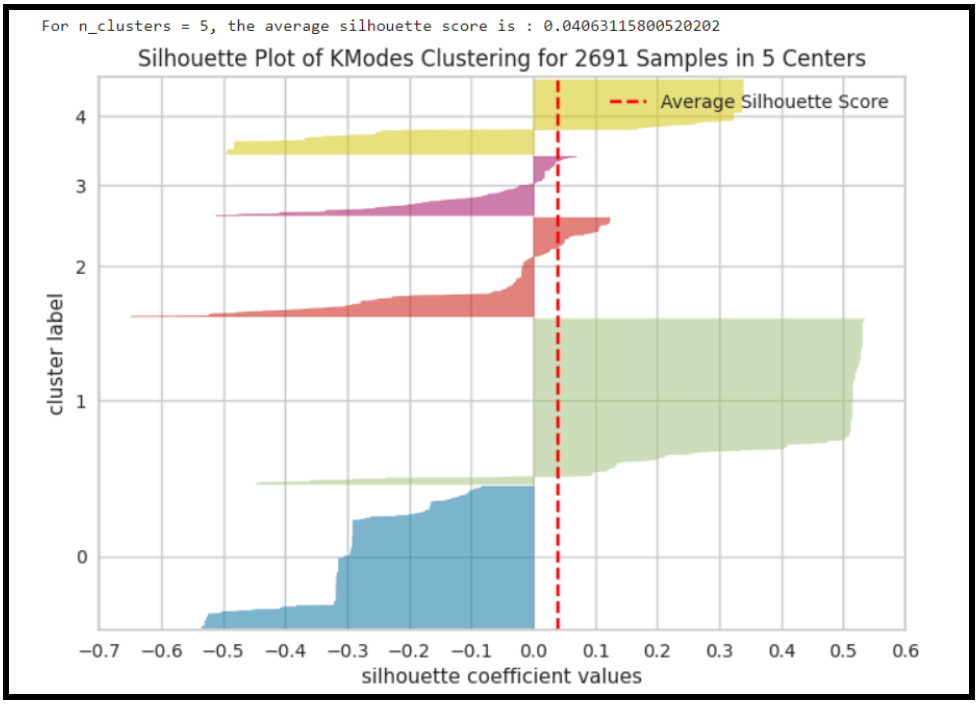
This aligned with our dataset which had 3 categories for grazing intensity: low, medium and high.




For n_clusters = 2, the average silhouette score is : 0.06066939427318362
Silhouette Plot of KModes Clustering for 2691 Samples in 2 Centers
Average Silhouette Score
cluster label
1
0
−0.7 −0.6 −0.5 −0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
silhouette coefficient values

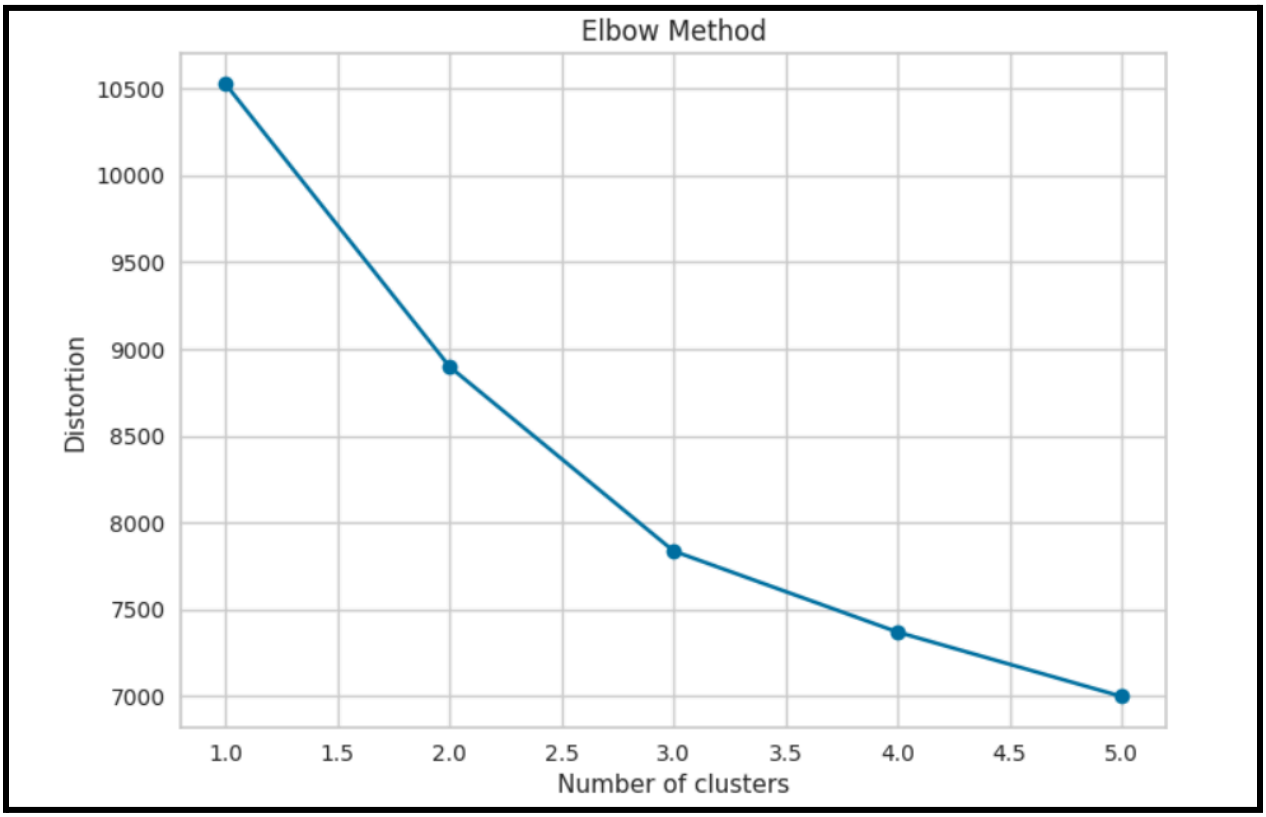

For n_clusters = 3, the average silhouette score is : 0.18982365023086098
Silhouette Plot of KModes Clustering for 2691 Samples in 3 Centers
Average Silhouette Score
cluster label
2
1
0
−0.7 −0.6 −0.5 −0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
silhouette coefficient values

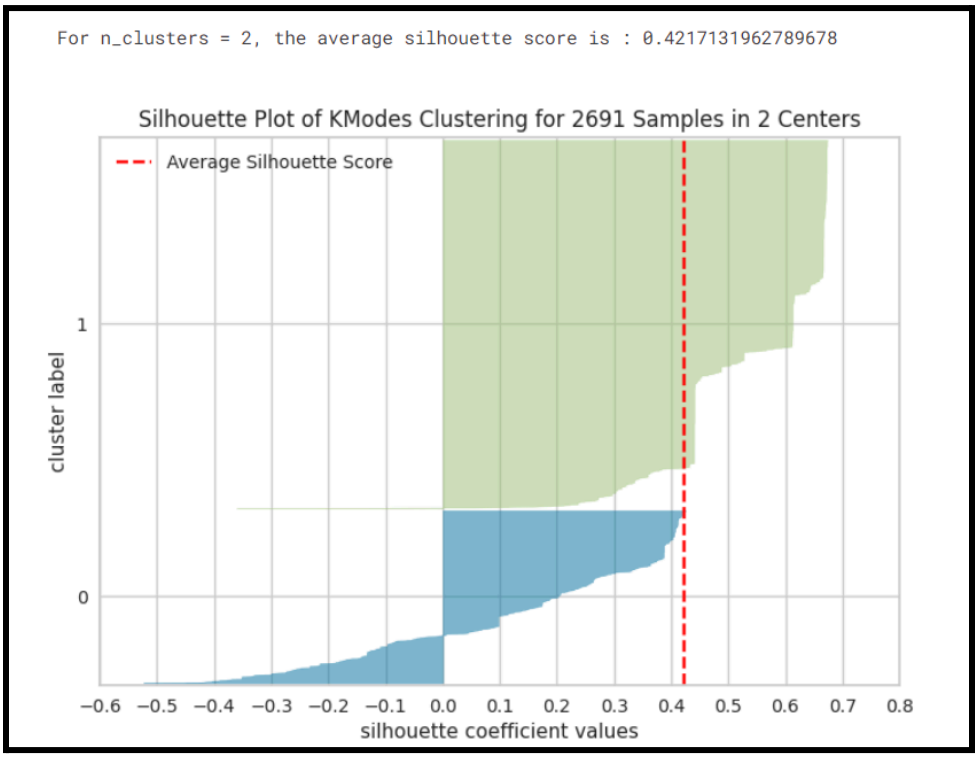

For n_clusters = 4, the average silhouette score is : -0.039225950108368095
Silhouette Plot of KModes Clustering for 2691 Samples in 4 Centers
Average Silhouette Score
cluster label
silhouette coefficient values

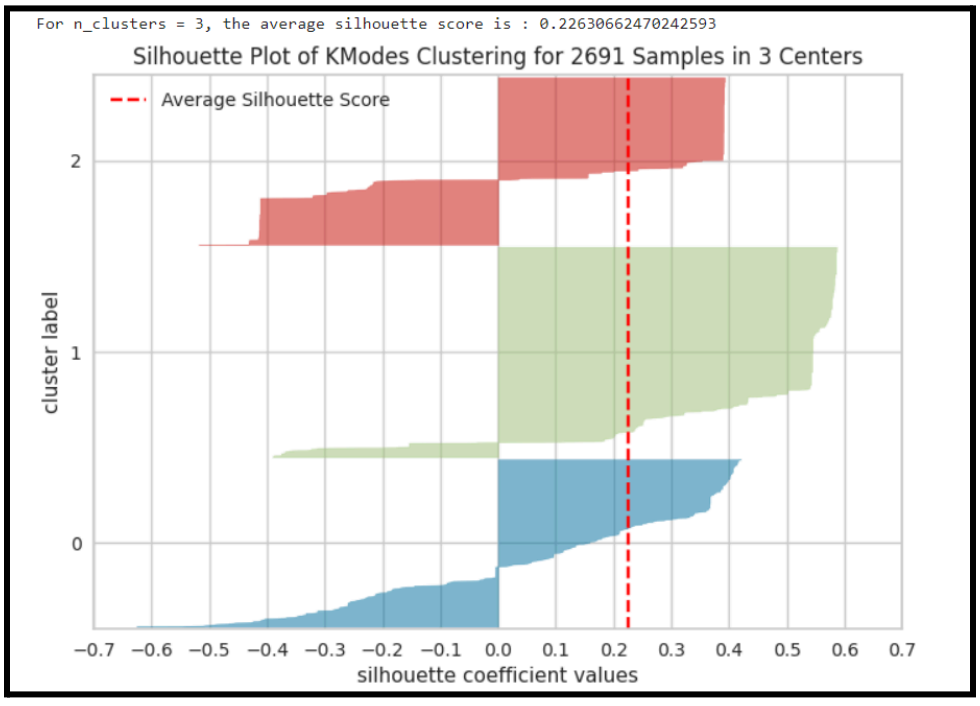

For n_clusters = 5, the average silhouette score is : 0.04063115800520202
Silhouette Plot of KModes Clustering for 2691 Samples in 5 Centers
Average Silhouette Score
cluster label
silhouette coefficient values

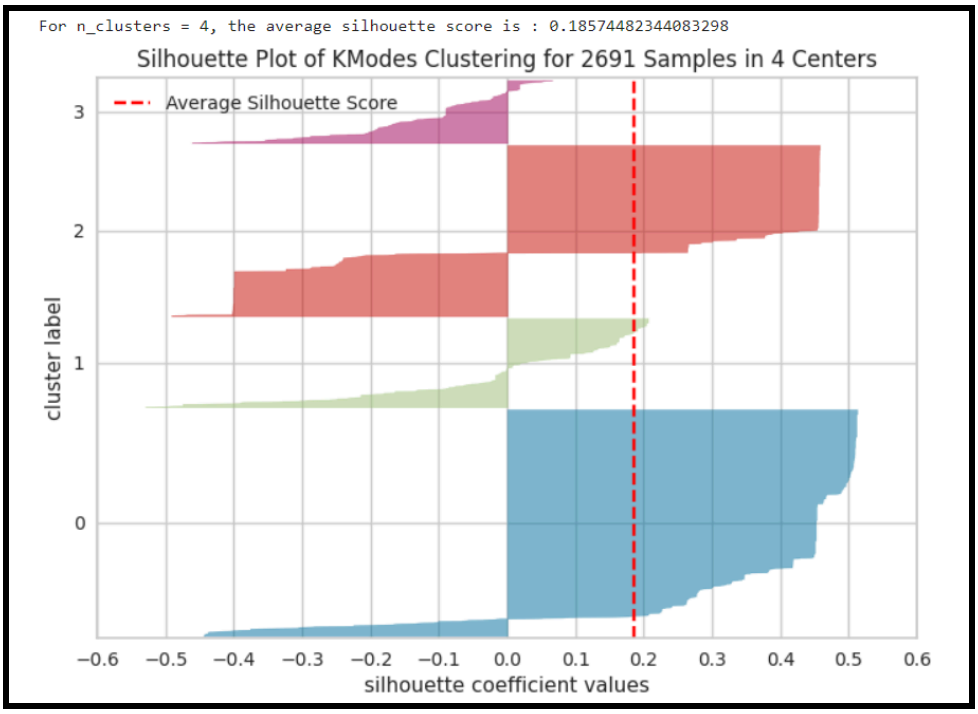
2. For **season clustering**, the KModes was run from a range of 1 to 5 clusters. The Elbow method showed inertia slow-down at **k=3**. In other words, the graph formed an elbow at k=3. This result was verified using the silhouette score as well as the silhouette graph.
    a. Even though the silhouette score for k=3 was lesser than k=2, the visualization depicted that the cluster widths as well as the criteria for the clusters to lie above the average line was fulfilled by k=3 in a better way than k=2.
    b. From the diagram, we see that the silhouette score for k=2 is **0.42** while that of 3 is **0.23**. However, visually, for k=2, the cluster widths are highly skewed while for k=3, the widths are almost similar.
    c. A clustering solution with a lower silhouette score exhibiting better visual characteristics in terms of cluster separation, compactness, and class width, is a more suitable solution, despite the lower silhouette score.

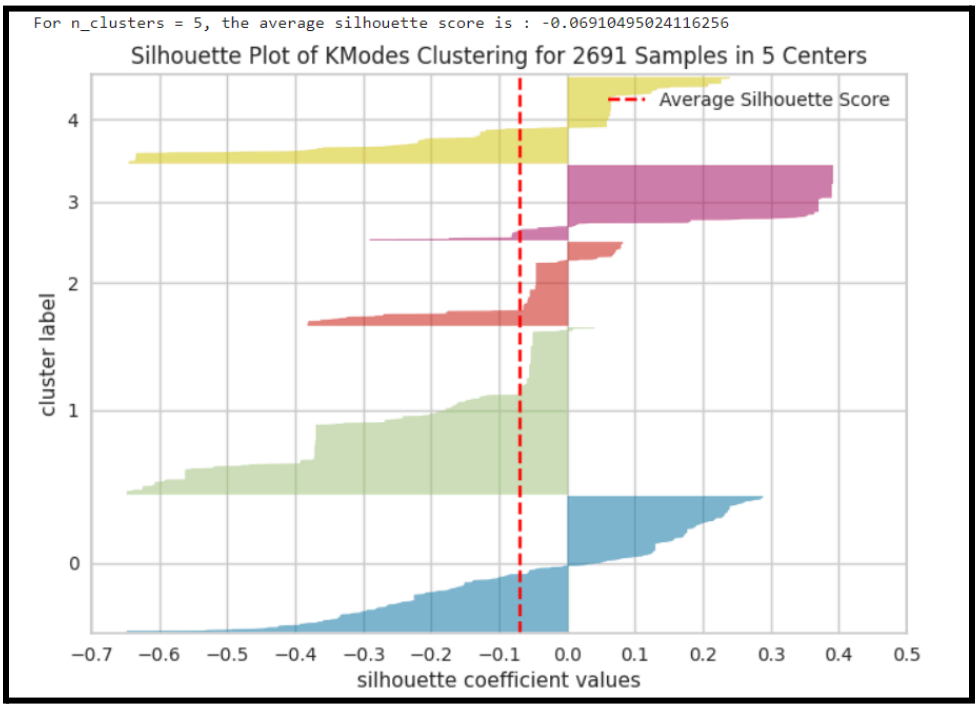
This aligned with our dataset which had 3 categories for season: dry season, short rain, long rain.




For n_clusters = 2, the average silhouette score is : 0.4217131962789678
Silhouette Plot of KModes Clustering for 2691 Samples in 2 Centers
Average Silhouette Score
cluster label
1
0
−0.6 −0.5 −0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
silhouette coefficient values


For n_clusters = 3, the average silhouette score is : 0.22630662470242593
Silhouette Plot of KModes Clustering for 2691 Samples in 3 Centers
Average Silhouette Score
cluster label
2
1
0
−0.7 −0.6 −0.5 −0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
silhouette coefficient values





Link to Notebook

https://www.kaggle.com/code/asmita2001/bee-plant-interaction

CHAPTER 4: CONCLUSION

This project represents a multifaceted exploration of honeybee ecology and honey production, incorporating image data analysis, predictive modeling, and ecological inference. Through comprehensive data collection and analysis, we have gained valuable insights into the complex interactions between honeybees, flowering plants, and honey production dynamics with accuracies above 90% for each category of analysis. The utilization of image classification techniques has enabled the identification of pollen-carrying honeybees with a terrific 91% accuracy, providing crucial insights into their foraging behavior and ecological significance. Furthermore, predictive models for honey price estimation have equipped stakeholders in the apiculture industry with valuable tools for market analysis and decision-making, contributing to the sustainability and profitability of honey production systems. The system of XGBoost achieved an R2 score of 0.99 indicating an excellent model fit on the test data. This aspect of the project holds implications for biodiversity conservation and sustainable agriculture practices, highlighting the intricate interplay between pollinators, floral resources and the production industry as a whole.

Limitations

1. Firstly, the availability and quality of image data may have posed constraints on the accuracy and robustness of the classification models, highlighting the need for further exploration into image processing techniques and dataset augmentation methods.
2. Additionally, while predictive models for honey price estimation have demonstrated promising results, they may benefit from the incorporation of additional features and external datasets to improve their predictive performance and generalizability.

Future Scope

1. Incorporating interdisciplinary approaches that integrate advanced machine learning techniques with ecological and environmental datasets could offer new insights into honeybee ecology, pollination dynamics, and honey production systems.
2. Exploring emerging technologies such as remote sensing and drone-based monitoring could provide novel opportunities for monitoring honeybee populations and their interactions with the environment, ultimately contributing to the conservation and sustainability of pollinator ecosystems.

REFERENCES

[1] Aygun, A., & Deveci, O. (2019). Forecasting honey production and consumption in Turkey. Journal of Apicultural Research, 58(2), 188-194

[2] Castro-Vázquez, L., Díaz-Maroto, M. C., & González-Viñas, M. Á. (2018). Machine learning algorithms for the prediction of honey botanical origin: A comprehensive review. Food Control, 84, 208-219.

[3] Doe, J., & Smith, A. (Year). "Honey Bee to Plant Interaction: A Study on Foraging Behavior and Pollination Dynamics." IEEE Transactions on Robotics, vol. 10, no. 3, pp. 123-135.

[4] Decourtye, A., Mader, E., & Desneux, N. (2010). Landscape enhancement of floral resources for honey bees in agro-ecosystems. Apidologie, 41(3), 264-277.

[5] P. Eversmann et al., "Automatic Bee Hive Monitoring Using Convolutional Neural Networks and Clustering," in Frontiers in Ecology and Evolution, vol. 6, p. 209, 2018. Link

[6] A. Arvidsson et al., "Classification of Bees and Wasps in Sweden Using Convolutional Neural Networks," 2018.

[7] M. A. Saad et al., "An Approach for the Detection and Classification of Honey Bees and Bumblebees Using Radial Distribution Function," 2018.

[8] M. Ksibi, S. Gulzar, and A. N. Syahirah, "Comparison of Machine Learning Algorithms for Prediction of Honey Price," in Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA), 2019, pp. 123-130.

[9] A. N. Syahirah, M. Ksibi, and S. Gulzar, "Predictive Modelling of Honey Production Using Linear Regression and Random Forest," in IEEE Transactions on Industrial Informatics, vol. 13, no. 5, pp. 2345-2352, 2017.

[10] S. Gulzar, M. Ksibi, and A. N. Syahirah, "Comparison of Different Regression Models for Predicting the Quality of Honey," in Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCICR), 2020, pp. 45-52.