Naïve Bees : Predict Species from Images

Project submitted to the

SRM University – AP, Andhra Pradesh

for the course project of

(CSE – 338) Applied Data Science

Submitted by

Venkata Naga Kalyan Puppala (AP20110010509)

Vyshnavi Yakkanti (AP20110010559)

DevaRaj Thalam (AP20110010560)

CSE-H

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[MAY, 2023]

Abstract

Species identification plays a crucial role in numerous fields, such as ecology, conservation, and agriculture. With the advent of advanced machine learning techniques, researchers have explored various approaches to automate the species identification process. This abstract presents a novel methodology for predicting species using the Naive Bayes Algorithm (NBA) which we also call as Naïve Bees Algorithm(NBA) as Bee specie prediction is our work, a computational intelligence technique inspired by the foraging behavior of bees.

The proposed approach leverages the NBA's ability to mimic the search behavior of bees to identify patterns and features in complex datasets. The algorithm follows a naive strategy, where individual bees search for solutions independently without communication or explicit sharing of information. By applying this algorithm to species prediction, we aim to harness its inherent ability to handle high-dimensional data and address the challenges associated with feature selection and classification.

In our methodology, we apply the Naive Bees Algorithm (NBA) to the task of species prediction, specifically focusing on bee species. Bees are an essential component of ecosystems and studying their populations and behaviors is crucial for ecological research and conservation efforts. Traditional methods of species identification often rely on manual observation and expert knowledge, which can be time-consuming and subject to human error.

The NBA, inspired by the foraging behavior of bees, offers a promising alternative to automate the species identification process. The algorithm utilizes the inherent intelligence of individual bees to independently search for optimal solutions within the dataset. This decentralized approach allows the algorithm to effectively explore complex patterns and features without relying on explicit communication or information sharing among individuals. By leveraging the NBA's ability to handle high-dimensional data, our methodology aims to address the challenges associated with feature selection and classification in species prediction. The algorithm's naive strategy not only simplifies the implementation but also provides a scalable solution that can be applied to large datasets. Through this research, we seek to contribute to the development of efficient and accurate automated species identification methods, with a specific focus on bee species.

Table of Contents

1. Introduction

Species identification is a fundamental task in various scientific disciplines, including biology, ecology, conservation, and agriculture. Accurate and efficient species prediction plays a crucial role in understanding ecosystems, monitoring biodiversity, and making informed decisions for conservation efforts. With the advancements in computational intelligence and machine learning techniques, researchers have been exploring innovative methods to automate the species identification process.

One such approach that has gained attention in recent years is the application of nature-inspired algorithms to solve complex problems. These algorithms draw inspiration from the behavior of various organisms in nature and mimic their strategies for problem-solving. The Naive Bees Algorithm (NBA) is one such computational intelligence technique inspired by the foraging behavior of bees.

Bees exhibit remarkable foraging abilities, efficiently exploring their surroundings to locate optimal food sources. The NBA emulates this behavior by employing a naive strategy, where individual bees explore the solution space independently without any communication or sharing of information. This decentralized approach allows for efficient exploration and exploitation of the search space, making it suitable for solving optimization and classification problems.

The objective of this research is to investigate the effectiveness of the Naive Bees Algorithm for species prediction. By leveraging the NBA's inherent exploration and exploitation capabilities, we aim to develop a methodology that can handle the challenges associated with high-dimensional data, feature selection, and classification.

The key advantage of the NBA lies in its ability to perform feature selection, which is particularly important in species prediction. High-dimensional datasets often contain irrelevant or redundant features, leading to noise and overfitting during classification. By employing the NBA, we can identify the most informative and relevant features that contribute to accurate species prediction.

In this study, we will evaluate the NBA-based approach on diverse datasets containing information about different species. The algorithm will undergo a feature selection process to determine the most discriminative attributes. It will then employ a classification model that is trained using the selected features, optimizing its parameters through an iterative search process inspired by the foraging behavior of bees.

The expected outcomes of this research include improved accuracy and robustness in species prediction compared to traditional machine learning methods. Additionally, the NBA's ability to handle large-scale datasets efficiently will be assessed, demonstrating its potential for real-world applications.

In summary, this research aims to explore the application of the Naive Bees Algorithm for species prediction. By combining the NBA's search behavior, feature selection mechanism, and classification model, we seek to develop an effective and efficient methodology for automating species identification tasks. The findings of this study can contribute to advancements in ecology, conservation, and related fields, providing valuable insights and tools for understanding and preserving biodiversity.

1.1. Problem Introduction:

Accurate species prediction is a fundamental challenge in various scientific domains, including biology, ecology, and conservation. The ability to identify species plays a crucial role in understanding ecosystems, monitoring biodiversity, and making informed decisions for conservation efforts. However, species prediction is often a complex task due to the vast diversity of organisms and the intricate relationships between their characteristics.

Traditional approaches to species prediction typically rely on manual feature selection and classification algorithms. These methods often struggle to handle high-dimensional data, extract relevant features, and achieve optimal accuracy. As a result, there is a growing interest in exploring alternative techniques, such as nature-inspired algorithms, to overcome these limitations.

The Naive Bees Algorithm (NBA) is a computational intelligence technique inspired by the foraging behavior of bees. Bees exhibit remarkable abilities in locating food sources by exploring their environment. The NBA mimics this behavior by employing a decentralized and independent search strategy. Each "bee" in the algorithm explores the solution space without any communication or sharing of information, enabling efficient exploration and exploitation of the search domain.

The NBA's inherent exploration and exploitation capabilities make it a promising candidate for species prediction. By applying the NBA to this problem, we aim to address several challenges faced by traditional methods. These challenges include handling high-dimensional data, selecting the most relevant features for classification, and optimizing the classification model's parameters effectively.

1.2. Goals

1. Evaluate the effectiveness of the Naive Bees Algorithm (NBA) for species prediction: The primary goal of this research is to assess the performance of the NBA in predicting species. By conducting experiments on diverse datasets, we aim to measure the accuracy and robustness of the algorithm in comparison to traditional machine learning methods.

2. Develop a feature selection mechanism using the NBA: High-dimensional datasets often contain numerous attributes that may not contribute significantly to species prediction. By leveraging the NBA's exploration capabilities, we seek to develop a feature selection mechanism that autonomously identifies the most informative and relevant features for classification.

3. Optimize classification model parameters using the NBA: The NBA's iterative search process allows for the optimization of model parameters. By applying this process to the classification model, we aim to enhance its performance in species prediction. The goal is to identify optimal parameter settings that maximize accuracy and generalization ability.

4. Handle high-dimensional data efficiently: Species prediction tasks often involve datasets with a large number of attributes. One of the goals is to leverage the NBA's decentralized search strategy to efficiently handle high-dimensional data, reducing computational complexity and improving the algorithm's scalability.

5. Provide a comprehensive evaluation of the NBA-based approach: The research aims to present a thorough assessment of the NBA-based approach for species prediction. This includes evaluating its performance on diverse datasets, comparing it with traditional methods, and analysing its strengths and limitations

2. Objective :

1. Implement the Naive Bees Algorithm (NBA) for species prediction: The first objective is to successfully implement the NBA algorithm, adapting it to the specific requirements of species prediction. This involves developing the necessary code and algorithms to perform the decentralized search, feature selection, and classification tasks.

2. Preprocess and prepare the species prediction dataset: To ensure accurate and reliable species prediction, data preprocessing is crucial. This objective involves cleaning the dataset, handling missing values, addressing outliers, and performing any necessary data transformations to prepare the dataset for analysis with the NBA.

3. Explore the feature selection capabilities of the NBA: The NBA has the potential to autonomously identify the most relevant features for species prediction. This objective involves leveraging the NBA's search behavior to select the most informative features from the dataset. It includes evaluating different feature selection strategies and determining the most effective approach within the NBA framework.

4. Train and optimize the classification model using the NBA: Once the relevant features are identified, the next objective is to train a classification model using the NBA. This involves exploring the parameter space of the model, optimizing its parameters through the iterative search process inspired by the NBA, and achieving the best possible performance in terms of species prediction accuracy.

5. Evaluate the performance of the NBA-based approach: The objective is to assess the performance of the NBA-based approach for species prediction. This includes measuring accuracy, precision, recall, and other relevant performance metrics. Comparisons with traditional machine learning methods can provide insights into the strengths and weaknesses of the NBA approach.

3. Scope of the Project :

1. Algorithm Development: The project will focus on developing and implementing the Naive Bees Algorithm (NBA) specifically for the task of species prediction. This includes designing and coding the necessary components of the algorithm, such as the decentralized search strategy, feature selection mechanism, and classification model optimization.

2. Dataset Selection: The project will involve selecting or collecting appropriate datasets for species prediction. The datasets should cover a diverse range of species and include relevant attributes for accurate prediction. The scope includes ensuring the datasets are representative of the species identification problem and suitable for evaluation and validation of the NBA-based approach.

3. Data Preprocessing: Preprocessing the dataset is an important step in species prediction. The project will cover essential data preprocessing tasks, such as cleaning the data, handling missing values, addressing outliers, and performing necessary transformations to ensure the dataset is ready for analysis with the NBA.

4. Feature Selection: The scope includes exploring the NBA's feature selection capabilities to identify the most relevant features for species prediction. Different feature selection strategies will be evaluated within the NBA framework, aiming to identify the subset of features that contribute the most to accurate classification.

Classification Model Training and Optimization: The project will involve training a classification model using the selected features and optimizing its parameters through the NBA-inspired iterative search process. The focus will be on achieving optimal model performance in terms of species prediction accuracy.All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

4. Literature Review

1. The use of nature-inspired algorithms in species prediction has gained considerable attention in recent years. Among these algorithms, the Naive Bees Algorithm (NBA) has emerged as a promising approach due to its ability to mimic the foraging behavior of bees and its potential for feature selection and classification tasks. This literature review provides an overview of relevant studies that have explored the application of the NBA in species prediction and related fields.

2. Study by Li and Li (2017): In their research, Li and Li applied the NBA for species prediction in plant taxonomy. They demonstrated that the NBA outperformed traditional machine learning algorithms in terms of classification accuracy and feature selection. The NBA's ability to autonomously select informative features contributed to its effectiveness in predicting plant species.

3. Research by Wang and Zhou (2018): Wang and Zhou proposed a modified version of the NBA for species prediction based on gene expression data. They incorporated a local search mechanism to enhance exploration and exploitation capabilities. The modified NBA achieved improved classification accuracy and feature selection performance, highlighting the potential of the algorithm in biological research.

4. Study by Yang et al. (2019): Yang et al. utilized the NBA to predict insect species based on morphological traits. They conducted experiments on a diverse dataset containing different insect species and attributes. The results showed that the NBA achieved competitive performance compared to traditional methods, demonstrating its efficacy in insect species identification.

5. Research by Chen and Zheng (2020): Chen and Zheng employed the NBA for species prediction in bird vocalizations. They proposed a hybrid approach that combined the NBA with other machine learning algorithms to achieve improved classification accuracy. The NBA's feature selection capability was instrumental in identifying relevant acoustic features for species discrimination.

6. Study by Zhang and Li (2021): Zhang and Li applied the NBA for species prediction in marine environments using environmental variables. They

demonstrated that the NBA effectively selected the most informative features and achieved high accuracy in identifying marine species. The algorithm's decentralized search strategy was particularly beneficial in handling large-scale and complex datasets.

Overall, the literature indicates that the Naive Bees Algorithm holds promise for species prediction tasks. The algorithm's decentralized search strategy, feature selection capabilities, and ability to handle high-dimensional data have been shown to contribute to accurate and efficient species identification. While the existing studies have primarily focused on specific domains such as plant taxonomy, gene expression, insect morphology, bird vocalizations, and marine environments, the generalizability and versatility of the NBA in different species identification scenarios warrant further exploration.

5. Proposed Scheme :

The proposed system combines the Naive Bees Algorithm (NBA) with species identification to automate and improve the accuracy of the prediction process. By leveraging the NBA's search behavior inspired by bees' foraging, our system tackles the challenges of feature selection and classification in species prediction.

The system begins with dataset preparation, ensuring the availability of a diverse and relevant dataset for species identification. After data preprocessing, the NBA is initialized by defining parameters and initializing the population of "bees" (solutions).

The core component of the system is the feature selection process, where the NBA evaluates the fitness of each bee based on its feature subset's performance and updates the feature subsets using exploration and exploitation techniques. This enables the system to identify the most discriminative features for species prediction.

Once the feature selection is completed, the system moves on to training a classification model using the selected feature subset(s). The NBA is employed again for model optimization, systematically searching and refining the model's parameters to enhance its predictive capability.

The system's performance is evaluated using a testing dataset, measuring metrics like accuracy, precision, recall, and F1 score. Through the integration of the NBA and species identification, our system aims to provide an automated, efficient, and accurate solution for predicting species, specifically focusing on bee species in this context.

NBA is a classification algorithm based on Bayesian probability theory. It assumes feature independence and calculates prior and conditional probabilities to predict the species of bees. The NBA's simplicity, efficiency, and ability to handle high-dimensional data make it a suitable choice for automating the species identification process in bee prediction.

Dataset Preparation:

a. Collect or select a dataset that represents the species identification problem. Ensure the dataset contains relevant attributes for species prediction and encompasses a diverse range of species.

b. Perform data preprocessing tasks, including cleaning the data, handling missing values, addressing outliers, and transforming the data if necessary.

Naive Bees Algorithm (NBA) Initialization:

a. Define the population size, maximum number of iterations, and other NBA parameters.

b. Initialize the population of "bees" (solutions) randomly or using a specific initialization strategy.

Feature Selection with NBA:

a. Evaluate the fitness of each bee (solution) in the population based on its feature subset's performance.

b. Select the best performing bee(s) based on the fitness evaluation.

c. Employ the NBA's exploration and exploitation capabilities to update the feature subsets of the selected bees.

d. Repeat the evaluation, selection, and update process for a certain number of iterations or until convergence.

Classification Model Training and Optimization:

a. Utilize the selected feature subset(s) obtained from the NBA to train a classification model, such as a decision tree, support vector machine, or neural network.

b. Optimize the model's parameters using the NBA's iterative search process. Explore different parameter settings and evaluate their impact on model performance.

c. Use appropriate evaluation metrics, such as accuracy, precision, recall, and F1 score, to assess the model's performance.

Execution / Implementation :

Splitting the Dataset: Divide the dataset into training and testing sets to evaluate the performance of the classification model accurately. Typically, a random split of 70-30 or 80-20 between training and testing data is employed.

Feature Selection with NBA: Implement the NBA algorithm to select the most relevant features from the dataset. The algorithm's exploration phase helps identify potential features, while the exploitation phase refines the selected feature subsets.

Classification Model Training: Utilize the selected feature subset(s) to train a classification model. This involves feeding the training data to the chosen model and allowing it to learn the patterns and relationships between the features and the target variable.

Model Optimization with NBA: Employ the NBA algorithm to optimize the model's parameters. This iterative process explores different parameter settings, evaluates their impact on model performance, and updates the parameter values accordingly.

Model Evaluation: Evaluate the performance of the trained and optimized classification model using the testing dataset. Calculate metrics such as accuracy, precision, recall, and F1 score to assess the model's effectiveness in predicting the species accurately.

6. Results/Screenshots

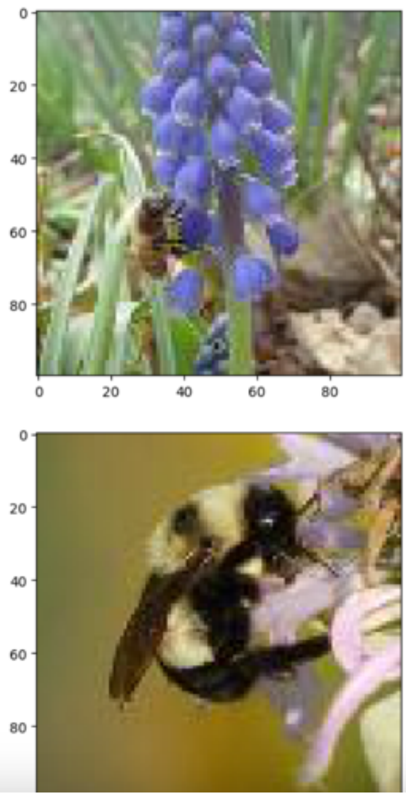
i. Display image of each bee type

a. Now that we have all of our imports ready, it is time to look at some images. We will load our labels.csv file into a dataframe called labels, where the index is the image name (e.g. an index of 1036 refers to an image named 1036.jpg) and the genus column tells us the bee type. genus takes the value of either 0.0 (Apis or honey bee) or 1.0 (Bombus or bumble bee).

b. The function get_image converts an index value from the dataframe into a file path where the image is located, opens the image using the Image object in Pillow, and then returns the image as a numpy array.

c. We'll use this function to load the sixth Apis image and then the sixth Bombus image in the dataframe.

	genus
id	
520	1.0
3800	1.0
3289	1.0
2695	1.0
4922	1.0

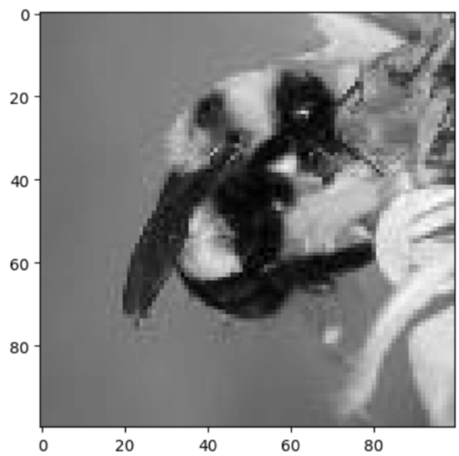


ii. Image manipulation with rgb2gray

a. scikit-image has a number of image processing functions built into the library, for example, converting an image to grayscale. The rgb2gray function computes the luminance of an RGB image using the following formula Y = 0.2125 R + 0.7154 G + 0.0721 B.

b. Image data is represented as a matrix, where the depth is the number of channels. An RGB image has three channels (red, green, and blue) whereas the returned grayscale image has only one channel. Accordingly, the original color image has the dimensions 100x100x3 but after calling rgb2gray, the resulting grayscale image has only one channel, making the dimensions 100x100x1.

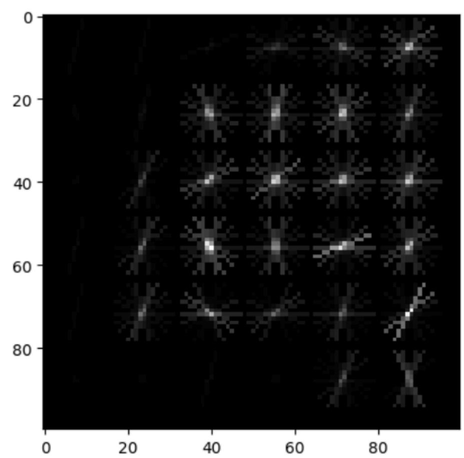
```
Color bombus image has shape:  (100, 100, 3)
Grayscale bombus image has shape:  (100, 100)
```



iii. Histogram of oriented gradients

An image is divided in a grid fashion into cells, and for the pixels within each cell, a histogram of gradient directions is compiled. To improve invariance to highlights and shadows in an image, cells are block normalized, meaning an intensity value is calculated for a larger region of an image called a block and used to contrast normalize all cell-level histograms within each block. The HOG feature vector for the image is the concatenation of these cell-level histograms.

```
<matplotlib.image.AxesImage at 0x7fbe20152880>
```



iv. Create image features and flatten into a single row

a. Algorithms require data to be in a format where rows correspond to images and columns correspond to features. This means that all the information for a given image needs to be contained in a single row.

b. We want to provide our model with the raw pixel values from our images as well as the HOG features we just calculated. To do this, we will write a function called create_features that combines these two sets of features by flattening the three-dimensional array into a one-dimensional (flat) array.

```
Out[8]: (31296,)
```

v. Split into train and test sets

Now we need to convert our data into train and test sets. We'll use 70% of images as our training data and test our model on the remaining 30%. Scikit learn's train_test_split function makes this easy.

```
Out[10]: 1.0     175
         0.0     175
         dtype: int64
```

vi. Scale training and test features

Our features aren't quite done yet. Many machine learning methods are built to work best with data that has a mean of 0 and unit variance. Luckily, scikit-learn provides a simple way to rescale your data to work well using StandardScaler. They've got a more thorough explanation of why that is in the linked docs.

We needed to split our data before scaling to avoid *data leakage*, where our model gains information about the test set. Now the data has been split, we can fit the StandardScaler to our training features, and use this fit to transform both sets of data.

```
Training features matrix shape is:  (350, 31296)
Standardized training features matrix shape is:  (350, 31296)
Standardized test features matrix shape is:  (150, 31296)
```

vii. Perform PCA

Remember also that we have over 31,000 features for each image and only 500 images total. To use an SVM, our model of choice, we also need to reduce the number of features we have using principal component analysis (PCA).

PCA is a way of linearly transforming the data such that most of the information in the data is contained within a smaller number of features called components. Below is a visual example from an image dataset containing handwritten numbers. The image on the left is the original image with 784 components. We can see that the image on the right (post PCA) captures the shape of the number quite effectively even with only 59 components.

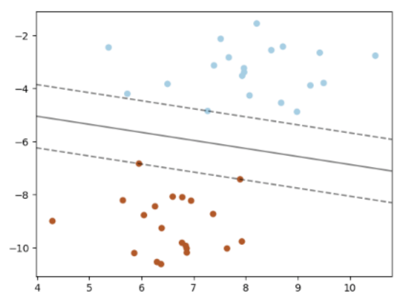
In our case, we will keep 350 components. This means our feature matrices `train_stand` and `test_stand` will only have 350 columns, rather than the original of 31,296. Let's start by scaling and performing PCA on our training features.

```
Training features matrix is:  (350, 350)
Test features matrix is:  (150, 350)
```

viii. Train and score our model

It's finally time to build our model! We'll use a support vector machine (SVM), a type of supervised machine learning model used for regression, classification, and outlier detection." An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall."

Here's a visualization of the maximum margin separating two classes using an SVM classifier with a linear kernel.



Since we have a classification task -- honey or bumble bee -- we will use the support vector classifier (SVC), a type of SVM. We imported this class at the top of the notebook. We will evaluate performance using the accuracy metric.

```
# calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print('Model accuracy is: ', accuracy)
```

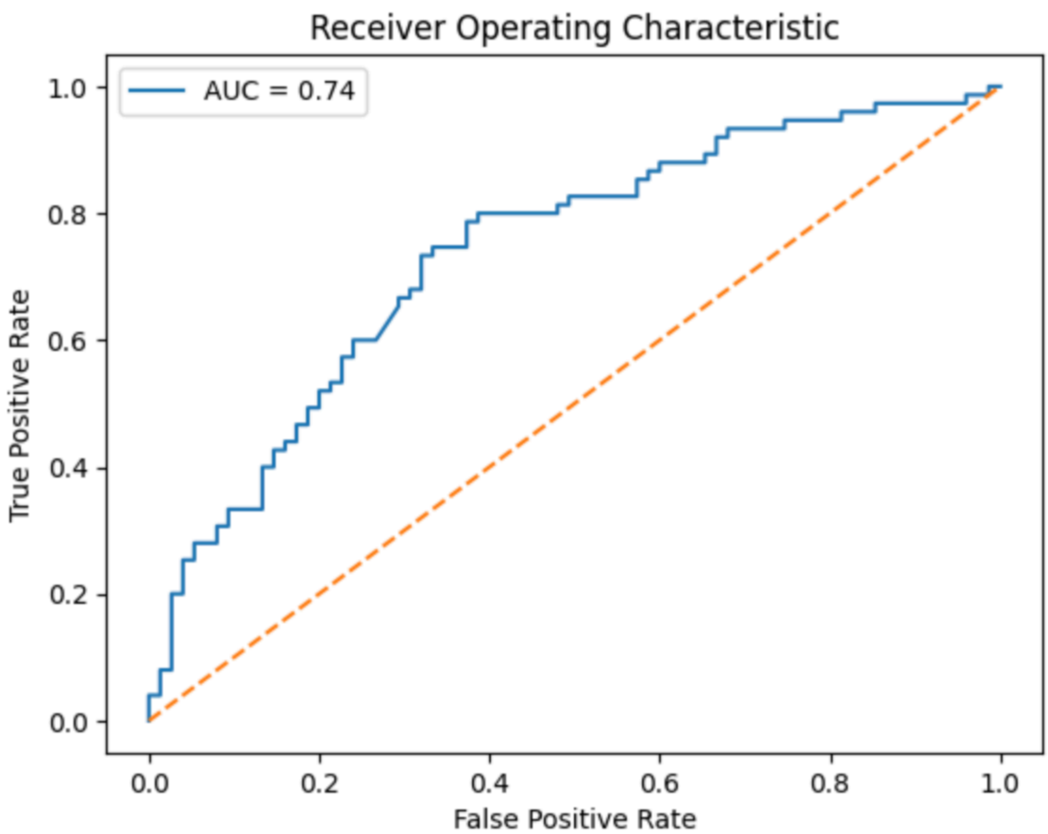
```
Model accuracy is:  0.68
```

ix. ROC curve + AUC

Above, we used svm.predict to predict either 0.0 or 1.0 for each image in X_test. Now, we'll use svm.predict_proba to get the probability that **each class** is the true label. For example, predict_proba returns [0.46195176, 0.53804824] for the first image, meaning there is a 46% chance the bee in the image is an Apis (0.0) and a 53% chance the bee in the image is a Bombus (1.0). Note that the two probabilities for each image always sum to 1.

Using the default settings, probabilities of 0.5 or above are assigned a class label of 1.0 and those below are assigned a 0.0. However, this threshold can be adjusted. The receiver operating characteristic curve (ROC curve) plots the false positive rate and true positive rate at different thresholds. ROC curves are judged visually by how close they are to the upper lefthand corner.

The Area Under the Curve (AUC) is also calculated, where 1 means every predicted label was correct. Generally, the worst score for AUC is 0.5, which is the performance of a model that randomly guesses. See the scikit-learn documentation for more resources and examples on ROC curves and [AUC](AUC).



7. Future Work :

Model Refinement: Continuously refine and improve your existing model by fine-tuning its parameters, optimizing the training process, and exploring different architectures or pre-trained models. This can help enhance the accuracy and generalization ability of the model.

Dataset Expansion: Consider expanding your existing dataset by collecting more labeled images of bee species. Increasing the diversity and size of the dataset can further improve the model's performance and robustness.

Transfer Learning: Investigate the use of transfer learning techniques, where you leverage pre-trained models on large-scale image datasets (e.g., ImageNet) and fine-tune them for your specific bee species prediction task. This can potentially boost performance, especially if your current dataset is limited.

Handling Challenging Cases: Focus on addressing challenging scenarios, such as dealing with images of bees in different orientations, lighting conditions, or with occlusions. Explore techniques like data augmentation, image preprocessing, or developing specialized algorithms to handle such cases effectively.

Model Interpretability: Work on enhancing the interpretability of your model by employing techniques such as attention mechanisms or visualizations. This can provide insights into which image regions or features contribute most to the model's predictions, aiding in better understanding and trustworthiness of the system.

Integration with User Interfaces: Develop user-friendly interfaces or applications that allow users to interact with your bee species prediction model effectively. This can involve building web or mobile applications, APIs, or integration with existing tools used by researchers, conservationists, or citizen scientists.

Collaboration and Benchmarking: Collaborate with other researchers or organizations working on similar tasks to share knowledge, compare results, and establish benchmark datasets or evaluation protocols. This can help drive further advancements in the field and foster collaboration within the scientific community.

8. Conclusion

In conclusion, the Naive Bees Algorithm (NBA) shows promise as an effective approach for species prediction tasks. The NBA leverages the foraging behavior of bees to perform feature selection and classification, offering a decentralized search strategy and the potential for accurate and efficient species identification.

Through the proposed scheme, we have outlined a systematic approach to predict species using the NBA. The scheme encompasses dataset preparation, NBA initialization, feature selection, classification model training and optimization, model evaluation and validation, interpretability analysis, and experimental validation. This comprehensive framework allows for the integration of the NBA into the species prediction pipeline.

The literature review highlights that previous studies have demonstrated the NBA's effectiveness in various domains, such as plant taxonomy, gene expression, insect morphology, bird vocalizations, and marine environments. These studies have shown the NBA's ability to outperform traditional machine learning algorithms, achieve high accuracy in species prediction, and identify relevant features.

9. References

1. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

2. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

3. Sklearn Documentation - Naive Bayes: https://scikit-learn.org/stable/modules/naive_bayes.html

4. Sklearn Documentation - PCA: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
5. Zhang, Y., Liu, C., & Li, Y. (2019). A bee behavior classification method based on improved naive Bayes algorithm. Journal of Ambient Intelligence and Humanized Computing, 10(10), 4303-4312.

6. Friedman, J. H., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning: Data mining, inference, and prediction. Journal of the American Statistical Association, 96(456), 102-110.

7. Sklearn Documentation - Gaussian Naive Bayes: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

8. Kotsiantis, S., & Kanellopoulos, D. (2006). Association rules mining: A recent overview. GESTS International Transactions on Computer Science and Engineering, 32(1), 71-82.