# Dried Bean Variety Classification Model

Machine Learning 934G5

CandNo: 260774

May 12, 2023

UNIVERSITY
OF SUSSEX

# 1 Introduction

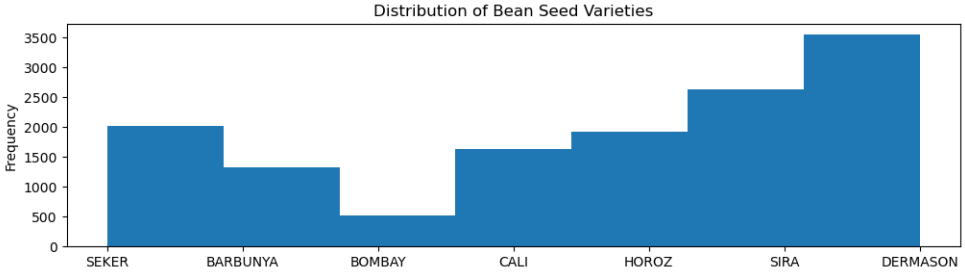
The objective of this report is to present a dry bean seed classification model using machine learning techniques. Beans are an important crop worldwide and bean classification is a crucial process that can affect their quality and value. In this project, a bean seed dataset was used to train and evaluate the classification model. The model is based on seed characteristics such as size, shape and texture, and machine learning algorithms were used to identify and classify different bean seed varieties. The results of the model will be presented and discussed in this report, as well as its potential applications.

# 2 Dataset

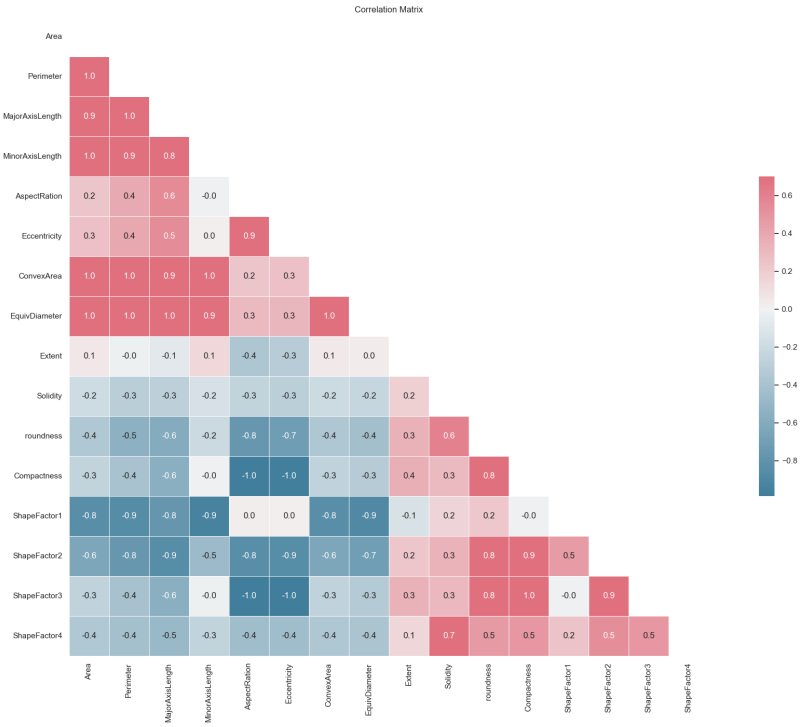
The model was trained and evaluated using a dataset of 13.611 dry bean seed samples from 7 different cultivars, collected in Turkey between 2016 and 2017. The dataset includes information on 16 physical and biochemical attributes of the seeds, such as weight, size, shape, color, and protein content [1].

While the dataset is a valuable resource for developing and testing dry bean seed classification models, it also has some limitations. For example, the dataset only includes samples from Turkey and a limited number of cultivars [1], which may not be representative of the global diversity of dry bean seeds. The dataset does not include information on the age or origin of the seeds, which could affect their quality and classification. In addition, as can be seen in the seed variety distribution graph, there is not a good amount of data for BOMBAY seed, which may lead to a bias of that classification in the model.

Despite these limitations, the Dry Bean Seeds Classification Model we developed using machine learning algorithms achieved high accuracy and precision in classifying the seeds into their respective cultivars. We also demonstrate the potential of the model for real-world applications, such as identifying the origin and quality of dry bean seeds in the international trade market, or improving the breeding and selection of new dry bean cultivars with desired traits [1][2].




Furthermore, the correlation matrix in the following figure shows that all variables have a medium to high correlation with each other, except for Extent and Solidity which exhibit the lowest correlation values with the other variables. In any case, we will use all the features to train the model, since even though Solidity and Extent have a low correlation, there may be patterns inherent in these variables that are not explainable by correlation alone.




## 2.1 Pre-processing

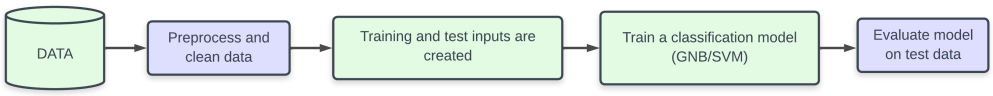
A pre-processing of the data was carried out involving cleaning of null, empty and duplicate values, then the data was normalised. Data normalisation is a process used to scale the values of variables in a specific range. The objective of normalisation is to make variables comparable to each other and to avoid a variable with larger or smaller values having more weight in the analysis than other variables.

To perform the normalisation, we use the MinMaxScaler function from sklearn.preprocessing, which contains tools for preparing and preprocessing data before it is used in machine learning algorithms. The MinMaxScaler function normalises the data to a range of [0, 1].

# 3 Proposed Methods

In this study, two Machine Learning models were used for bean seed classification: Gaussian Naïve Bayes (GNB) as primary model and Support Vector Machine (SVM) as an extra task.

For both models, the flow diagram is as follows:




## 3.1 Primary Model: Gaussian Naive Bayes (GNB)

The GNB model is a probabilistic classifier that uses Bayes' Theorem to predict the class of a seed instance. This model assumes that each feature is independent of each other, which makes it very computationally efficient and simple [3].

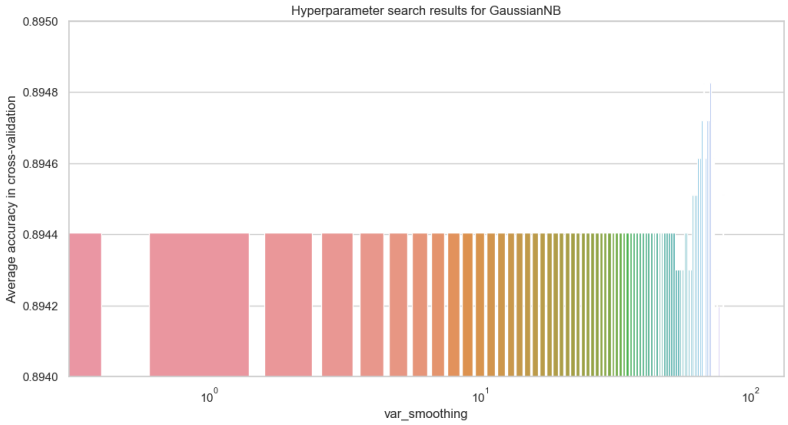
Because it requires less training data than other algorithms and is efficient in terms of processing time, it is chosen as the primary model.

### 3.1.1 Hyper-parameter Tuning

To adjust the hyperparameters, we use the grid search method which involves selecting specific values for each hyper-parameter and evaluating the model for each combination of values. The hyperparameters that can be modified in this code to obtain a better result are:

- var_smothing: It is used to smooth the estimated probability of each feature. This parameter is used to prevent the estimated probabilities from being zero, which can occur if a particular feature does not appear in the training set [4].

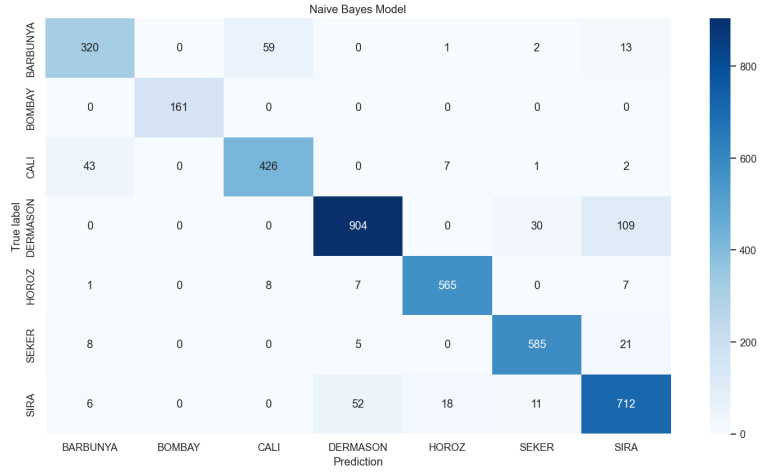
In order to find the best value of var_smothing, we create an array of 100 numbers that are uniformly distributed on a logarithmic scale between 10 and 10exp-9, then test the model with each value of this array. The results are shown in the graph below.




As the graph shows, the best Accuracy value (89.48%) is obtained with an approximate var_smothing value of 0.0023, therefore, this will be the value of the hyper parameter var_smothing that we will use in the final model.

### 3.1.2 Evaluation

Accuracy: 0.90 Precision: 0.90 Recall: 0.90 F1-score: 0.90




The results indicate that the model has an accuracy, recall and F1-score of 90%, suggesting that the model is very good at identifying dry bean seeds. Accuracy indicates the proportion of positives that are true positives, while recall indicates the proportion of true positives that were correctly identified. The F1-score is a harmonic mean between precision and recall, indicating the overall accuracy of the model. Overall, these results suggest that the model is highly effective in detecting dry bean seeds.

However, there are 52 records of SIRA that were incorrectly predicted as DERMASON and 109 records of DERMASON that were incorrectly predicted as SIRA. It follows that there is a similarity in the characteristics of these two classifications which may be misleading. Further improvements in the data quality could help avoid such misclassifications in the future.

## 3.2 Extra Task: Support Vector Machine

the SVM model is a classifier that seeks to find the optimal hyperplane that best separates the different seed classes. This model is especially useful in cases where classes are not linearly separable and more flexibility in separation is required [5].

### 3.2.1 Hyper-parameter Tuning

To adjust the hyperparameters, we use the grid search method which involves selecting specific values for each hyper-parameter and evaluating the model for each combination of values.

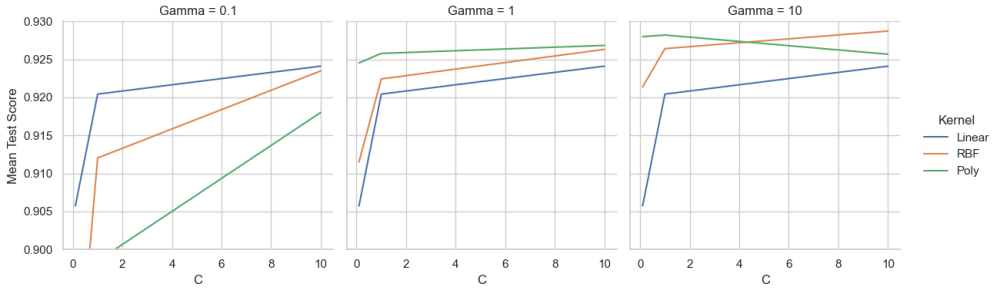
The hyperparameters that can be modified in this code to obtain a better result are:

- Kernel: Refers to the function used to transform the data into a higher dimensional space. The most common kernels are linear, polynomial and radial basis kernels (RBF). [6]
- C: It controls the trade-off between maximising the margin of the decision function and minimising the number of misclassified points in the training set. [7][8]
- Gamma It is used in the RBF kernel to define the size of the area of influence of a training point. Large values of gamma can cause overfitting of the model, while small values can cause underfitting. [7][8]

In order to find the best combination of values, we create the following matrix and train the model with each possible combination:

```
parameters = {'kernel': ['linear', 'rbf', 'poly'],
              'C': [0.1, 1, 10],
              'gamma': [0.1, 1, 10]}
```

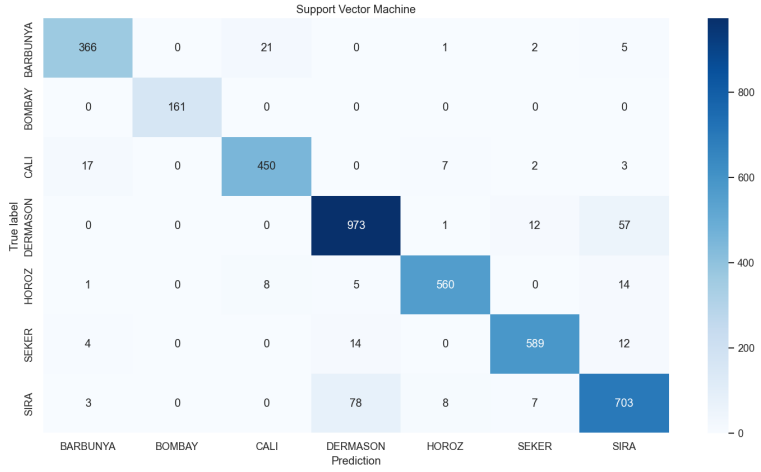
The results are shown in the graph below.




As the graph shows, the best Accuracy value (0.928) is obtained with the combination 'C': 10, 'gamma': 10, 'kernel': 'rbf', therefore, this will be the combination of the hyper parameters that we will use in the final model.

### 3.2.2 Evaluation

Accuracy: 0.93 Precision: 0.93 Recall: 0.93 F1-score: 0.93



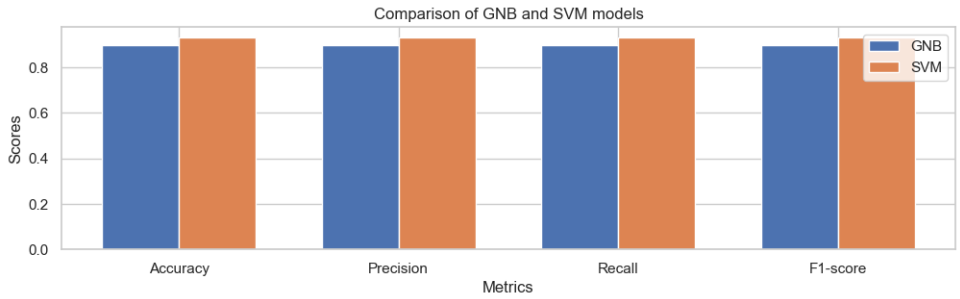
A Support Vector Machine (SVM) model with an Accuracy, Precision, Recall and F1-Score of 0.93 indicates that the model is highly accurate and reliable in detecting dry bean seeds.

Here we also observe erroneously predicted values between SIRA and DERMASON in the confusion matrix. In this case, there are 78 SIRA records that were wrongly predicted as DERMASON and 57 DERMASON records that were wrongly predicted as SIRA.

Overall, an SVM model with a score of 0.93 on these metrics suggests that the model is very effective at detecting dry bean seeds.

# 4 Comparison of both methods and K fold cross validation

By performing a single test set comparison of the two models, the Gaussian Naive Bayes model and the Support Vector machine model, we obtained very good results, both from 89% accuracy as shown in the figure below.

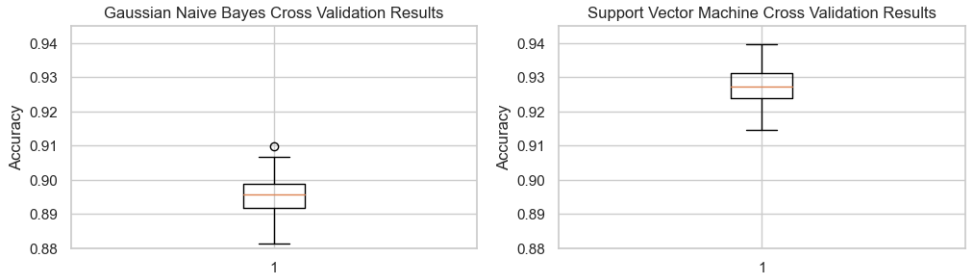



But once we identified the validations with the k-fold model, we were able to see the following:

The Gaussian model, with an accuracy of 89.86% on the test data without cross validation, gives us a mean cross validation of 89.5% with a standard deviation of 0.005. The SVM model, with an accuracy of 93.09% on the test data without cross validation, yields a mean cross validation of 92.75% with a standard deviation of 0.005.

Thus, the NB model obtains an accuracy of 89.86% when the test is run independently, but when iterating over the test subsets that are generated with k-fold, its prediction metric tends to decrease slightly. In the same way, the SVM model, obtains an accuracy of 93.09% and when iterating over the test subsets that are generated with k-fold, its prediction metric decreases slightly to 92.75%.

As a result, despite this slight decrease between the analysis with CV and without CV, both models have good stability and the results are independent of the test set when solving the dry seed classification problem. The following figure shows the cross validation results for both models.




# 5 Conclusion

Based on the evaluation of the Dry Bean Seeds Classification Model using machine learning algorithms, it can be concluded that the model is highly effective in detecting dry bean seeds, achieving high accuracy, precision, recall, and F1-score. The Gaussian Naive Bayes and Support Vector Machine algorithms were both used in the model and achieved high accuracy, precision, recall, and F1-score (90% and 93% respectively) and both were shown to be stable and independent of the dataset when performing cross validation tests. However, some misclassifications between similar classifications, such as SIRA and DERMASON, were observed, which highlights the need for improving data quality to avoid such errors in future models.

The results demonstrate the potential of the model for real-world applications, such as identifying the origin and quality of dry bean seeds in the international trade market or improving the breeding and selection of new dry bean

cultivars with desired traits. However, the dataset used for training and evaluation has some limitations, such as a limited number of cultivars, no information on the age or origin of the seeds and the fact that they are seeds from only one region. Further improvements in data quality could help avoid misclassifications in the future, including: increasing the number of BOMBAY seed samples to avoid bias in classification, including parameters to better differentiate SIRA seeds from DERMASON seeds to avoid mispredictions. In addition, future improvements to tada preprocessing are proposed, such as: selecting only those features that correlate strongly with each other and discarding features that do not correlate with others, and oversampling seeds with a small number of samples, such as BOMBAY, to balance the data.

# 6 References

[**1**] Murat Koklu et al. (2020) Multiclass classification of dry beans using computer vision and Machine Learning Techniques, Computers and Electronics in Agriculture.
Available at: https://www.sciencedirect.com/science/article/pii/S0168169919311573?via%3Dihub .

[**2**] Krishnan, S. et al. (2023) Identification of dry bean varieties based on multiple attributes using CatBoost machine learning algorithm, Scientific Programming. Available at: https://www.hindawi.com/journals/sp/2023/2556066/ .

[**3**] Majumder, P. (2020). Gaussian Naive Bayes. [online] OpenGenus IQ: Learn Computer Science. Available at: https://iq.opengenus.org/gaussian-naive-bayes/.

[**4**] kaggle.com. (n.d.). Naive Bayes with Hyperpameter Tuning. [online]
Available at: https://www.kaggle.com/code/akshaysharma001/naive-bayes-with-hyperpameter-tuning .

[**5**] Lujing Chen (2019). Support Vector Machine—Simply Explained. [online] Medium.
Available at: https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496.

[**6**] Scikit-learn.org. (2019). RBF SVM parameters — scikit-learn 0.21.3 documentation. [online] Available at: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

[**7**] Yıldırım, S. (2020). Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters. [online] Medium. Available at: https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167.

[**8**] Bhatt, B. (n.d.). svm-c-gamma-hyperparameter. [online] Deepnote. Available at: https://deepnote.com/@bhavesh-bhatt/svm-c-gamma-hyperparameter-ec7cdd4f-b499-4b4d-a320-f483e8099691 .