Explainable and Interpretable Dry Beans Classification using Soft Voting Classifier

Belayneh Dejene*[1], Gizachew Setegn[2], Selamawit Belay[1]

[1] *University of Gondar, Department of Information Science, Gondar, 196, Ethiopia*
[2] *Debark University, Department of Computer Science, Debark, 90. Ethiopia*

Abstract

Dry beans, integral to the Fabaceae family, boast global significance with their diverse genetic heritage tracing back to their dissemination from America centuries ago. This study endeavors to develop an explainable dry bean classification model using a soft voting classifier, juxtaposing its performance against classic and ensemble machine learning algorithms. Data preprocessing ensured suitability for classification algorithms, with feature selection employing information gain and variance inflation factors. The class imbalance was addressed via SMOTE + Tomek methods. Evaluation metrics encompassed accuracy, precision, recall, and F1-score. XGBoost led with 92.5065% accuracy, while soft voting classifiers (LGBM, XGB, CatBoost, RF, and DT) closely followed at 92.691%. The soft voting classifier proved optimal for dry bean classification, aiding in model interpretation and decision-making processes.

Keywords

Classification, Dry bean, Explainable, Machine learning, voting classifier.

1. Introduction

Dry beans belong to the diverse Fabaceae family, sometimes referred to as Leguminosae, and they are the most important and the most produced pulse in the world [1]. It originated in America, while there is a wide genetic diversity in the world since, in the 15th and 16th centuries, they were transported to Europe and Africa and quickly spread to the rest of the globe [1]. The selection of dry beans plays an important role in the economy of agriculture-based countries like Bangladesh, India, Pakistan, etc. throughout the winter season. Currently, Dry bean is a staple food for many regions of the world and processing enables the consumption and incorporation of this nutrient-dense food in daily diets. Dry beans are the most known source of protein. In addition, they are low in fat and a rich source of fiber and other important nutrients [2][3]. Dry beans are important for environmental and human health benefits, such as improved soil fertility, reduced risk of chronic disease, and improved or promoted glycemic control [1]. There are several genetic diversities of dry beans, and it is the most produced one among the edible legume crops in the world. According to the Turkish Standards Institution, dry beans are classified as Barbunya, Battal, Bombay, Calı, Dermason, Horoz, Tombul, Selanik, and Seker" based on their botanical characteristics [4][5][6]. Plants are sensitive to the effect of climatic changes and they have a variety of resistance. Finding high-quality seed is the primary challenge facing dry bean producers and distributors or marketers. Using a lower quality seed in production will induce to lower quantity even if all the cultivation conditions are provided.

A wide range of computational tools are available to regulate food and agricultural product quality. But most of them are done with the use of conventional techniques of the professionals. For example, different seed categorization is conducted based on human understanding, and determining the type of dry beans requires a skillful person to take a huge time manually, and passes a challenging process [6]. In particular, the color of various dry bean species varies, and geometrical data does not

Proceedings of the of DAAfrica'2024 Workshop, November 23, 2024, Bejaia, Algeria

[1]* B. Dejene.

belzman2011@gmail.com (B. Dejene); gizachewmulucs@gmail.com (G. Setegn); selamsun7@gmail.com (S. Belay)

0000-0002-5978-7691 (B. Dejene)

reveal this color variation. Due to this reason, it is vital in economically technical aspects to build an automated technique to detect as well as categorize seed features rapidly and repeatedly. Even, it is difficult for a human operator to understand or handle the seeds except for specific tools or automatic software procedures. The main problem dry bean producers and marketers face is in ascertaining good seed quality. Lower quality of seeds leads to lower quality of produce. Seed quality is the key to bean cultivation in terms of yield and disease. In today's world, the inspection of the quality of seeds, fruits, and vegetables along with the examination and categorization of seeds and grains have been performed worldwide to meet these demands with the help of machine learning and computer vision [4]. This is why we try to use a soft voting classifier and compare it with individual algorithms to classify dry beans. In recent years, machine learning algorithms have been used in the inspection, classification, prediction, and segmentation of food product quality. Classification techniques are becoming more popular in the fields of medicine, biostatistics, bioinformatics, agriculture, business, etc. as machine learning applications [7]. Machine learning is a subfield of artificial intelligence that enables computers to understand existing data and estimate the existence of unidentified targets. Seed quality is influential in crop production. Seed classification is important for both producers and marketers to provide the values of sustainable agricultural systems. By applying predictive analysis to agricultural data, significant decisions can be taken and classifications can be made.

Besides the classification model conducting explainability and interpretability of the classification model provide the professionals with insights into how the classifications are made, fostering trust in the model's decisions [8]. The explainable machine learning model impacts professionals more likely to trust and adopt understand and interpret the reasoning behind the model's recommendations by solving the black box nature of the algorithms. To handle this problem, several studies have been conducted to detect the quality of dry beans using various machine-learning techniques. For example, [4][5][6][7] conducted on dry bean classifications. The previous research on dry bean classification has largely neglected the crucial aspect of explainability and interpretability in their models. Instead, researchers predominantly focused on employing various algorithms without addressing the black box nature inherent in these methods. Classic machine learning approaches were commonly utilized, often with default parameter settings, despite evidence suggesting that optimizing these parameters could enhance classification performance [9]. Additionally, while some studies attempted to tackle class imbalance issues, they typically employed simplistic oversampling methods, which could lead to the generation of redundant data. Advanced techniques for addressing class imbalance were rarely explored. Furthermore, previous research overlooked feature selection methods, which could potentially improve model efficiency and interpretability. The absence of studies utilizing explainable techniques to handle black-box models, as well as the scarcity of research employing soft voting classifiers and tuned parameters, underscored the need for this study. Motivated by these gaps, this study endeavors to develop an explainable and interpretable classification model for dry beans. It seeks to utilize soft voting classifiers, a technique not extensively explored in previous research, and compare its performance with individual machine learning algorithms. By incorporating explainable and interpretable methods, this study aims to classify dry beans accurately while providing insights into the decision-making process, thus facilitating evidence-based policies and interventions in the selection of appropriate dry bean classes.

2. Related works

Several studies such as [4] [5][6] and [7], investigated the dry bean classifications using machine learning algorithms. However, most of the previous researchers didn't consider the explainability and the interpretability of the dry beans' classification model, most of these previous studies developed a classification model by handling the class imbalance problem on the whole data and

developing the classification model without tuning relevant parameters. These studies did not conduct any feature selection methods, they developed the classification model by using all the features in the dataset. M. Koklu and I. A. Ozkan [4] develop multi-class dry bean classifiers using MLP, SVM, kNN, and DT, classification models. The overall correct classification rates have been determined as 91.73%, 93.13%, 87.92%, and 92.52% for MLP, SVM, kNN, and DT, respectively. The SVM classification model has the highest performance with the accuracy of the Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira bean varieties 92.36%, 100.00%, 95.03%, 94.36%, 94.92%, 94.67%, and 86.84%, respectively. However, this researcher didn't consider the explainability and the interpretability of the dry beans' classification model. G. Słowiński [5] tried to classify dry beans using machine learning techniques: Multinomial Bayes, Support Vector Machines, Decision Trees, Random Forests, and Voting Classifier. The overall accuracies obtained were in the range: of 88.35 - 93.61%. However, this researcher didn't consider the explainability and the interpretability of the dry beans' classification model. M. Salauddin Khan et al [7] aimed to construct a multiclass dry bean classification model using the eight most popular classifiers and compare their performances. The algorithms they used, were LR, NB, KNN, DT, RF, XGB, SVM, and MLP with balanced and imbalanced classes. The XGB classifier performed better than other classifiers with the balanced and imbalanced dataset of dry beans within each class. It performed an accuracy of 93.0% and 95.4% in imbalanced and balanced classes respectively. The overall performance is better than the previous studies, however, the researchers didn't consider the explainability and the interpretability of the dry beans' classification model. The researcher develops the model without tuning the parameters and developing the model without those parameters faces overfitting. Not only this but also, the researcher handles the class imbalance problems on the whole dataset before splitting it, and evaluating the model using those fabricated datasets.

3. Materials and Methods

3.1. Data collection methods

To conduct this study, we have used the publicly available dataset in the Kaggle repository. The extracted datasets consist of a total of 13,611 grains of 7 different registered dry beans with a total of 17 features including the class level (see table 1 here below for the dataset descriptions)

Table 1. Dataset descriptions

No	Feature	Type	Description
1	Area	Integer	The area of a bean zone and the number of pixels within its boundaries
2	Perimeter	float	Bean circumference is defined as the length of its border
3	Major axis length	float	The distance between the ends of a dry bean can be drawn from a bean the longest line that
4	Minor axis length	float	The longest line that can be drawn from the bean while standing perpendicular to the main axis
5	Aspect ratio	float	Defines the relationship between L and l
6	Eccentricity	Real	The eccentricity of the ellipse having the same moments as the region
7	Convex area	Integer	Number of pixels in the smallest convex polygon that can contain the area of a bean seed
8	Equivalent diameter	float	The seed area diameter of a circle is the same area as a bean
9	Extent	float	The ratio of the pixels in the bounding box to the bean area
10	Solidity	float	The ratio of the pixels in the convex shell to those found in beans
11	Roundness	float	Calculated with the following formula
12	Compactness	float	Measures the roundness of an object
13	ShapeFactor1	float	Shape factor 1
14	ShapeFactor2	float	Shape factor 2
15	ShapeFactor3	float	Shape factor 3
16	ShapeFactor4	float	Shape factor 4
17	Class	Nominal	Target class of the dry bean

3.2. Data preprocessing methods

Data preparation involves data selection, data cleaning, data integration, feature selection, handling imbalances, and data transformation to make it available to extract value from those data [10][11]. In this subsection, we have detected the missing values, removed redundancies, detected outliers, and handled class imbalance problems from the dataset using statistical methods

3.2.1. Data cleaning

This is a way of removing noise, inconsistencies, redundancy, and missing values to carefully develop the model. Without cleaning the collected data, we can't get an accurate result [12][13]. In the dataset, there are no missing values, though we have not applied any methods to handle the missing values. From the data, we have removed 68 redundant records using drop redundant methods. Most of the variables have a higher proportion of outliers including Area, Perimeter, Minor Axis Length, Eccentricity, Convex Area, EquivDiameter, and ShapeFactor4. To handle this outlier, we have used interquartile range and boxplot methods.

3.2.2. Data transformation

Where data are transformed and consolidated into forms appropriate for extracting by performing summary or aggregation operations. The data are transformed into forms appropriate for mining [14][15]. In these datasets, only the class level needs to be transformed for mining purposes, but all the remaining features don't need to transform and we have used it as it is. To transform the class level, we have used the level encoding methods and encoded them into numeric values. We have encoded as 'DERMASON' = 0, 'SIRA' = 1, 'SEKER' = 2, 'HOROZ' = 3, 'CALI' = 4, 'BARBUNYA' = 5, and 'BOMBAY' = 6.

3.2.3. Feature selection

In this method, we have checked the importance of all the features by using information gain, (see Fig 1 here below), from the 16 features the last three, features (ShapeFactor4, Solidity, and Extent) were the least important, but it is not mean that they are not valuable for the model. We have checked the multicollinearity of the feature using the variance inflation factor, and the variance inflation factor shows that all of the features were significant to the model. Due to this, we have not dropped them for their usefulness and we used all of the 16 features for developing the classification model.

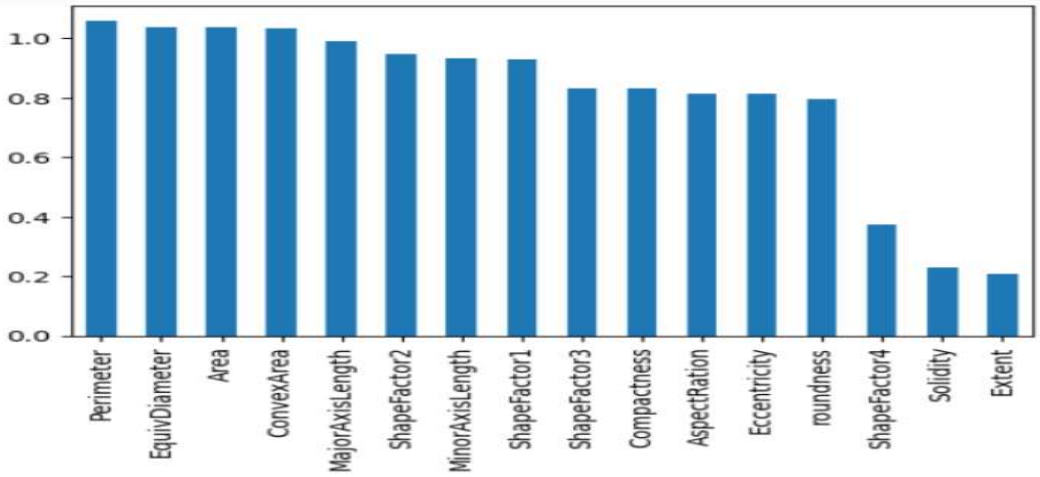

Fig. 1. Feature importance

3.2.4. Handling class imbalance

By nature, the class level of the collected data is imbalanced see Figure 2 here below. To overcome the imbalanced class distributions problem, we can add samples to or remove samples from the data set [16]. Sampling can be achieved in two ways, Under-sampling, randomly removing the majority

class, oversampling the minority class, or by combining over and under-sampling techniques [16][17]. The extracted dataset class level has 7 values, from these values, some of them have the least values see Figure 2 here below. In the class distribution, the "BOMBAY" class has the least value when we compare it with other classes. To conduct this research, we used the synthetic minority over-sampling technique (SMOTE) + Tomek methods to handle the class imbalance of the class levels of the dataset. The main reason that we use SMOTE + Tomek is, it avoids the loss of valuable information [16][17]. In SMOTE + Tomek, the SMOTE combines the SMOTE ability to generate synthetic data for the minority class and the Tomek ability to remove the data that are identified as Tomek links from the majority class [18][19].

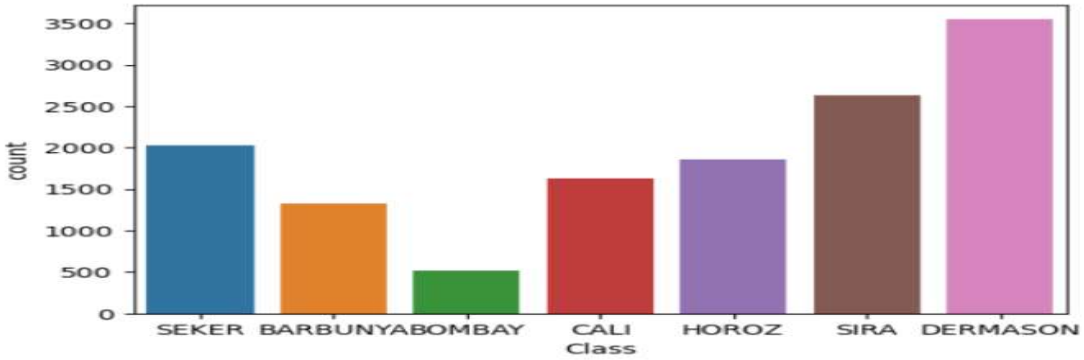

Fig. 2. Class imbalance

3.3. Train test split

In model building, the researcher needs to develop datasets for training and testing to learn and evaluate the machine appropriately [20][21]. To conduct this study, we used the stratified splitting technique to split the whole dataset to train and test data and split the dataset into 80:20 train test ratios.

3.4. Parameter tuning

In the process of machine learning and deep learning algorithms, the performance of the algorithm highly depends on the selection of hyperparameters, which has always been a crucial step in the process of machine learning [22][23][24]. To improve the performance rate for each algorithm a collection of hyperparameters has been tuned using grid search methods. Gird search is commonly used as an approach to hyper-parameter tuning that will methodologically build and evaluate a model for each combination of algorithm parameters specified in a grid [24]. Here, we used the grid-search with GridSearchCV for selecting tuning parameters for a homogeneous ensemble machine learning algorithm.

Table 2. Tuned parameters

No	Algorithms	Parameters
1	Soft voting classifier	Default parameters
2	LGBM classifier	Default parameters
3	Random Forest	criterion='entropy',max_features='sqrt',min_samples_split=3,n_estimators=500,random_state=0,max_depth=20, max_leaf_nodes=400, n_jobs=-1
4	Cat boost	random_state=42, learning_rate=0.1, l2_leaf_reg=4, iterations=600, depth= 6
5	xgboost	random_state=42, verbosity=0, min_child_weight=2, max_depth=4, learning_rate=0.15, gamma=0.22, colsample_bytree=0.5
6	Decision tree	max_depth=20, criterion='gini',max_features='sqrt',splitter='best', max_leaf_nodes=100 ,min_samples_split=3

3.5. Classification model

In this study, to construct a dry bean classification model we have used a soft voting classifier in both the balanced and the unbalanced dataset. To compare that the soft voting classifier can perform better than other machine learning algorithms, another model was developed using decision tree algorithms and other ensemble learning classifiers namely random forest, catboost XGBoost, and LGBM classifiers. To improve each algorithm's performance rate, a collection of hyperparameters has been tuned using grid search methods. The performance of each classification model was evaluated using accuracy, precision, recall, and F1- score.

3.6. Model explainability

To enhance the explainability of the classification model, we have employed various feature relevance explanation techniques like Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanation (SHAP) to highlight the most influential features and regions in the input data, and to explain the quality of the inner functioning of deep learning models and decisions by calculating the influence of each input variable and producing relevant scores. Global interpretability techniques, such as feature importance analysis or rule extraction, are employed to reveal the underlying patterns and decision rules learned by the model [8].

4. Result and discussion

Experiments have been carried out to develop a dry bean classification model by using a soft voting classifier and comparing it with other classic and ensemble machine learning algorithms. To construct a classification model for dry beans, we conducted two experiments on the imbalance data and the balanced data using a soft voting classifier, RF, cat boost, XGB, LGBM, and DT. Each experiment was conducted using 16 features and by using all the tuned parameters using grid search (see Table 2). This experiment is multiclass classification because the dataset by nature has seven class levels. In these experiments, we evaluated all the classification models using accuracy, precision, recall, and f1_score evaluation metrics. Finally, we have explained the model using LIME and SHAP feature relevancy explanation techniques.

Experiment# 1: Imbalanced dataset

This experiment was conducted by using the imbalanced dataset or without applying any data imbalance handling methods. We have developed the model by using DT, RF, Catboost, XGB, LGBM, and a soft voting classifier. We have also evaluated those models' using accuracy, precision, recall, and f1_score (see Table 3 here below)

Table 3. Model performance using the imbalanced dataset

Algorithms	Metrics			
	Accuracy	Precision	Recall	F1_score
Decision tree	0.910668	0.920002	0.920301	0.920069
Random forest	0.923588	0.936018	0.933484	0.934621
Cat boost	0.927649	0.939536	0.939268	0.938353
XGBoost	0.928756	0.940888	0.938008	0.939343
LGBM classifier	0.92285	0.936619	0.935014	0.935765
Soft voting classier (LGBM, cat boost, XGB, RF, DT)	0.92691	0.940701	0.93913	0.939856
Soft voting classier (cat boost, XGB)	0.927649	0.939835	0.93874	0.93922
Soft voting classier (LGBM, cat boost)	0.925065	0.93902	0.936754	0.937815
Soft voting classier (RF, DT)	0.90993	0.920326	0.919321	0.919589

As we see from Table 3 above, the XGBoost algorithm outperforms the best result with accuracy precision, and f1_score of 0.928756%, 0.940888%, and 0.938008% respectively. But in the case of recall cat boost algorithm performs the best with 0.939268%. When we see the soft voting classifiers, the soft voting of the algorithms LGBM, cat boost, XGB, RF, and DT performs better than the soft voting with other algorithms. In the soft voting algorithms, the voting that contains cat boost and XGBoost algorithm performs a better result.

Experiment# 2: balanced dataset

This experiment is conducted by balancing the dataset using SOMTE + Tomek methods on the training set only and developing the model using DT, RF, Catboost, XGB, LGBM, and soft voting classifiers. We have also evaluated those models' using accuracy, precision, recall, and f1_score (see Table 4 below).

Table 4. Model performance using the balanced dataset

Algorithms	Metrics			
	Accuracy	Precision	Recall	F1_score
Decision tree	0.921004	0.932629	0.933324	0.932835
Random forest	0.921004	0.932629	0.933324	0.932835
Cat boost	0.925434	0.9367	0.938996	0.937794
XGBoost	0.925065	0.936134	0.93809	0.937017
LGBM classifier	0.924695	0.937783	0.937861	0.937776
Soft voting classier (LGBM, cat boost, XGB, RF, DT)	0.926541	0.936982	0.938738	0.937805
Soft voting classier (cat boost, XGB)	0.925065	0.936067	0. 93803	0.93695
Soft voting classier (LGBM, cat boost)	0.926541	0.938991	0.940395	0.939642
Soft voting classier (RF, DT)	0.906238	0.916778	0.919568	0.917991

Finally, in this experiment developing the model by handling the imbalance problem is not always a good solution to get a better performance.

4.1. Model comparison

As a result, the researcher compared the performance of algorithms to classify the dry bean using a soft voting classifier and other classic and ensemble machine learning algorithms using both imbalanced and balanced datasets. The dataset has seven classes. Then, the researcher used overall accuracy, precision, recall, and f1_score as an evaluation for classification model comparison. According to the overall performance, the classification algorithm that registered the highest performance is selected as the best algorithm for the classification model for the dry bean. As indicated in Table 3 and Table 4 above, the experiments are conducted on classification algorithms for classifying the dry bean. The XGB algorithms registered the highest accuracy of 92.8756% in the imbalanced dataset and the soft voting classifiers of the algorithm LGBM, cat boost, XGB, RF, and DT performed an accuracy of 92.6541% using the imbalanced datasets. The soft voting classifiers of LGBM, XGB, Cat boost, RF, and DT perform the best result next to XGBoost algorithms with overall accuracy, precision, recall, and f1_score of 92.691%, 94.0701%, 93.913%, and 93.986% respectively. The decision tree algorithm is registered with the lowest performance in both the imbalanced and the balanced datasets, see Table 3 and Table 4. Therefore, the XGBoost algorithm is selected as the best classifier as compared to other classic and ensemble machine learning algorithms, and the soft voting classifiers of LGBM, XGB, Cat boost, RF, and DT are selected as the best classifier where we compared with other voting classifiers.

4.2. Model explainability

To enhance the explainability of the classification model, we have employed various techniques. We have explained and interpreted the classification model developed with each algorithm to make the trust of how it achieves the result. The explainable AI approach with LIME and SHAP frameworks is implemented to understand how the model predicts the final results. To explain the model, we

have randomly selected the rows 100, 150, 200, 250, and 300 in the dataset. But this row was selected randomly and we can select any other rows in the dataset.

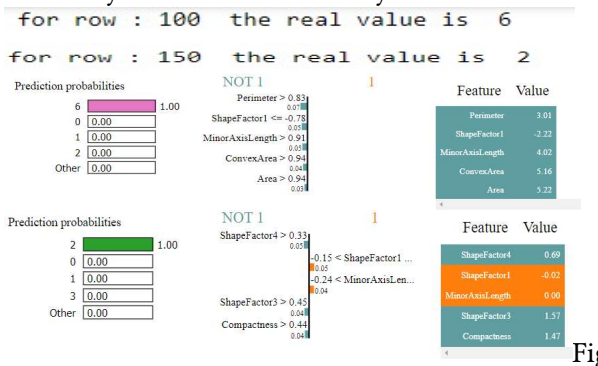

Fig. 3. Model Explanation with LIME for row 100

Fig.5. Model Explanation with LIME for row 150

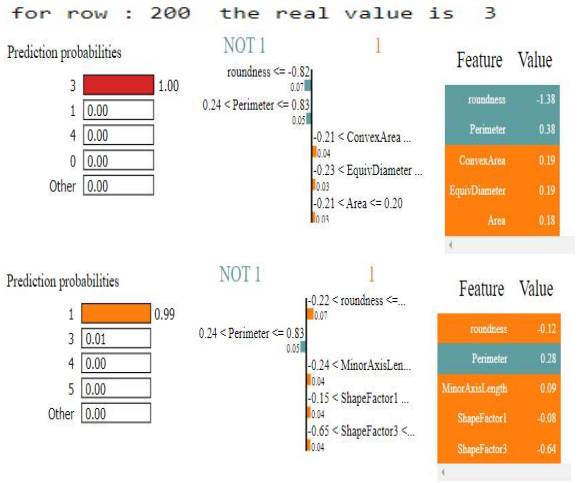

Fig. 4. Model Explanation with LIME for row 200

Fig.6. Model Explanation with LIME for row 250

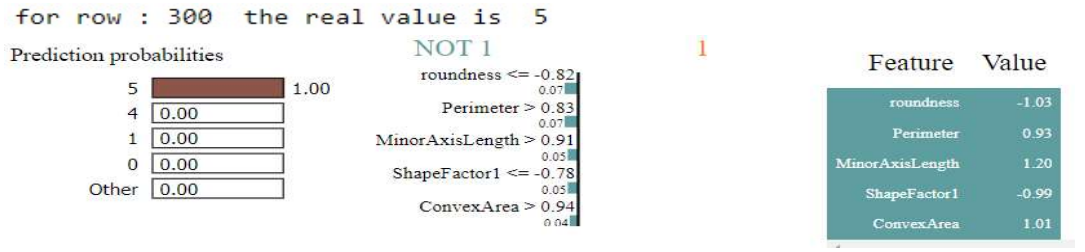

Fig. 7. Model Explanation with LIME for row 300

The figures 3, 4, 5, 6, and 7 above depict interpretations of an XGBoost model using the LIME explainable AI method for classifying specific types of dry beans. In each case, the model achieves 100% accuracy in classifying the beans into their respective classes. Here are the key findings from each interpretation:

Class 'BOMBAY' (Figure 3): The model identifies dry beans as 'BOMBAY' based on specific features such as perimeter, shape factors, minor axis length, convex area, and area. For instance, the beans are classified as 'BOMBAY' when perimeter > 0.83, ShapeFactor1 <= 0.78, MinorAxisLength > 0.91, Convex Area > 0.94, and Area > 0.94.

Class 'SEKER' (Figure 4): The model correctly classifies dry beans as 'SEKER' by considering features like shape factors, minor axis length, and compactness. For instance, beans are categorized as 'SEKER' when ShapeFactor4 > 0.33, ShapeFactor1 < -0.15, MinorAxisLength < -0.24, ShapeFactor3 > 0.45, and Compactness > 0.44.

Class 'HOROZ' (Figure 5): Dry beans are accurately classified as 'HOROZ' based on features such as roundness, perimeter, convex area, equivalent diameter, and area. For example, beans are classified as 'HOROZ' when roundness <= -0.82, Perimeter > 0.24 & <= 0.83, ConvexArea > -0.21 & <= 0.19, EquivDiameter > -0.23 & <= 0.19, and Area > -0.21 & <= 0.20.

Class 'SIRA' (Figure 6): The model identifies dry beans as 'SIRA' considering attributes like perimeter, roundness, minor axis length, shape factors, and shape factor 3. For instance, beans are classified as 'SIRA' when Perimeter > 0.24 & <= 0.83, roundness > -0.22 & <= -0.12, MinorAxisLength > -0.24 & <= 0.09, ShapeFactor1 > -0.15 & <= -0.08, and ShapeFactor3 > -0.65 & <= -0.64.
Class 'BARBUNYA' (Figure 7): Dry beans are correctly classified as 'BARBUNYA' based on features like roundness, perimeter, minor axis length, shape factor 1, and convex area. For example, beans are categorized as 'BARBUNYA' when roundness <= -0.82, Perimeter > 0.83, MinorAxisLength > 0.91, ShapeFactor1 <= -0.78, and ConvexArea > 0.94.
These interpretations provide insights into how the model makes its predictions, highlighting the specific features that are influential in classifying different types of dry beans.
Figures 8, 9, 10, 11, and 12 below show the decisions generated by the XGBoost model for the randomly selected rows of 100, 150, 200, 250, and 300 respectively. Based on the decisions generated by the XGBoost model, the class value for rows 100, 150, 200, 250, and 300 is 6, 2, 3, 1, and 5 respectively. to check the name of the class, see section 3.2.2.

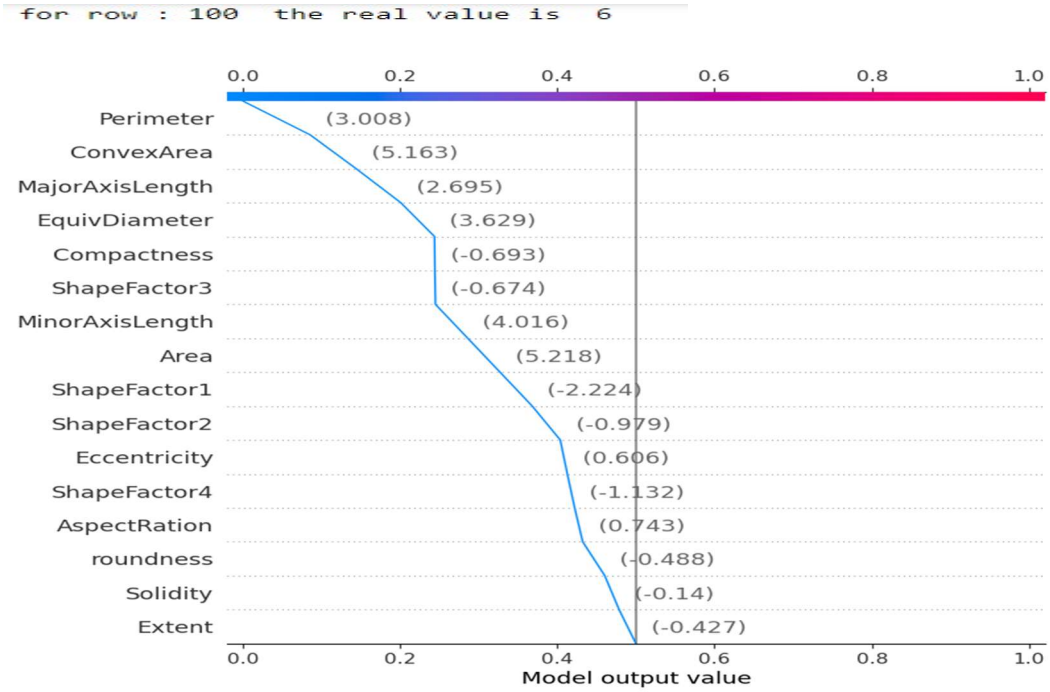

Fig.8. Decisions for row 100

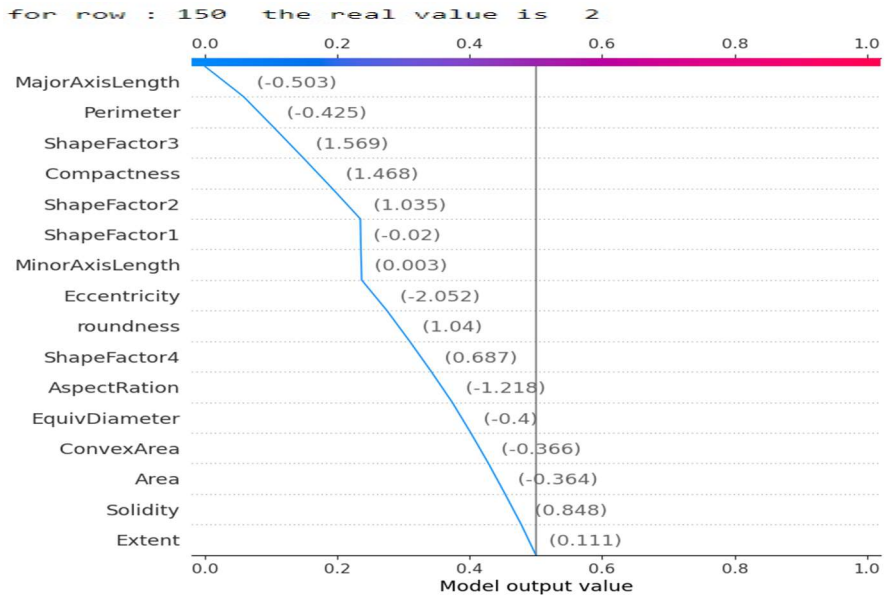

Fig. 9. Decisions for row 150

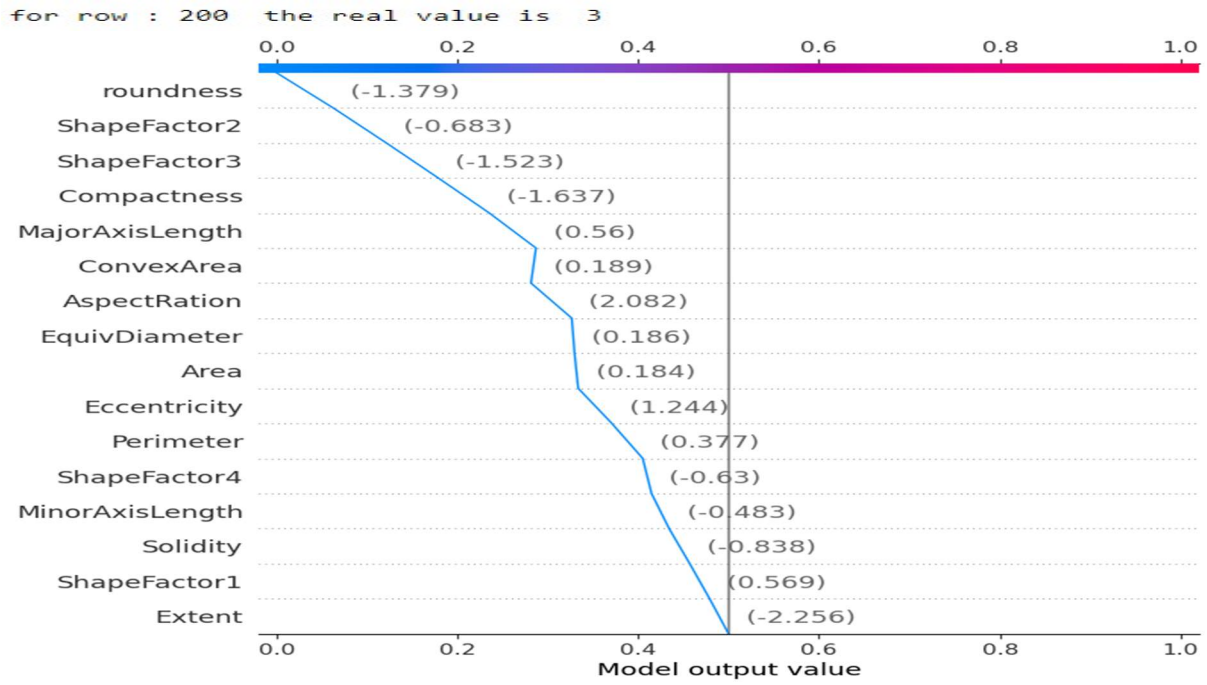

Fig. 10. Decisions for row 200

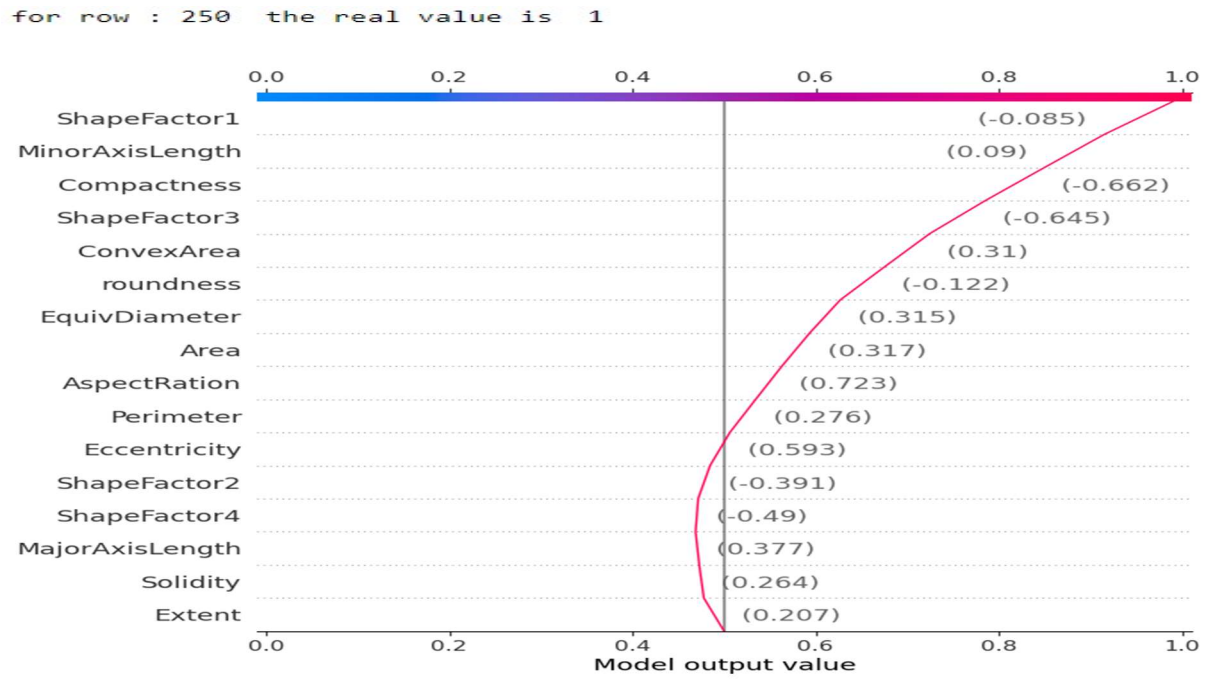

Fig. 11. Decisions for row 250

```
for row : 300  the real value is  5
```

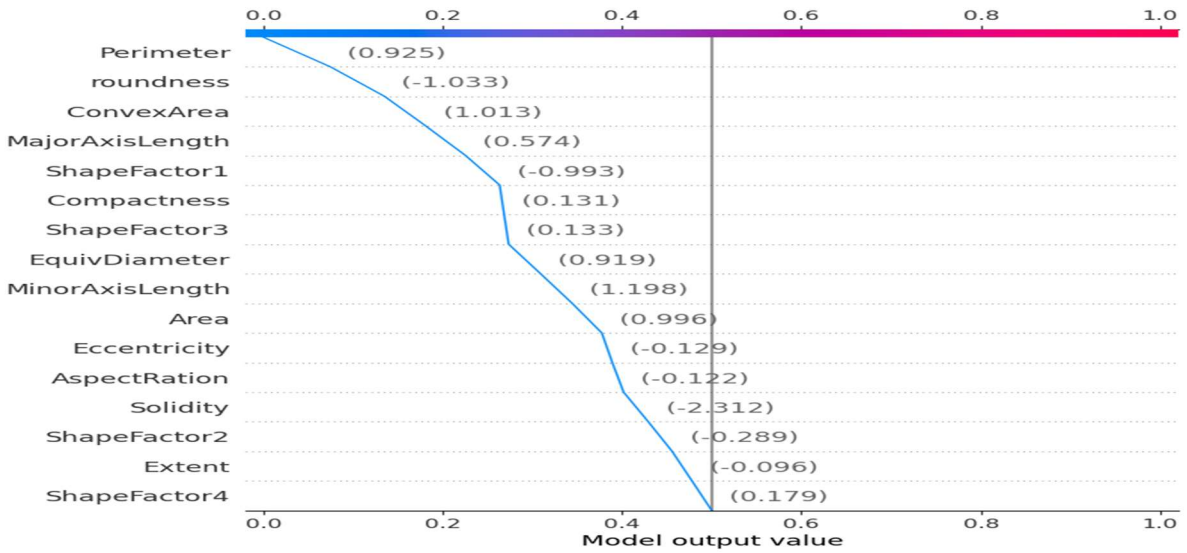

Fig. 12. Decisions for row 300

The figure 13 below shows the importance of each feature for each class in constructing the classification model. Based on the result above we have decided that XGBoost is the best classification model to classify the dry beans. So, we have explained the XGBoost model using SHAP explainable AI methods that explain the model using the feature relevancy in the model. As we see here below the figure shows the feature importance of each feature for each class.

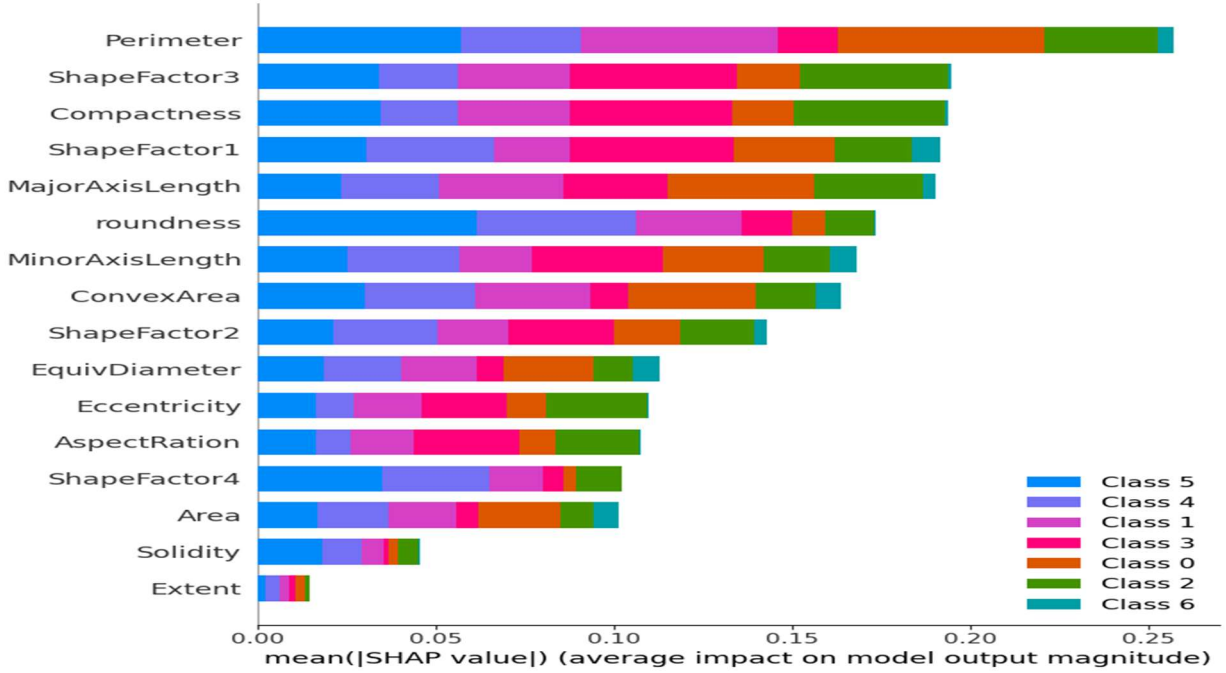

Fig. 13. Explainable AI with SHAP

5. Conclusion and Recommendation

Dry beans belong to the diverse Fabaceae family, sometimes referred to as Leguminosae, and they are the most important and the most produced pulse in the world. It is originally from America, while there is a wide genetic diversity in the world since, in the 15th and 16th centuries, they were transported to Europe and Africa and quickly spread to the rest of the globe. There are numerous genetic diversities of dry beans, and it is the most produced one among the edible legume crops in the world. According to the Turkish Standards Institution, dry beans are classified as Barbunya, Battal, Bombay, Calı, Dermason, Horoz, Tombul, Selanik, and Seker" based on their botanical characteristics. This study aimed to develop an explainable and interpretable classification model for

dry beans using a soft voting classifier and compare the performance with other classic and ensemble machine learning algorithms. The data source for this research is publicly available datasets on Kaggle. After applying the data preprocessing task, out of 13611 instances with 16 features and one class level, 13543 instances with 16 features were used for developing the classification model, and after handling class imbalance using SMOTE + Tomek, 7655 instances were used for the model. We checked the multicollinearity of each feature using variance inflation factors to check the significance of each feature, and we concluded that all the features were significant. The proposed model was constructed using soft voting classifiers, decision trees, random forests, extreme gradient boosting, cat boost, and LGBM algorithms using the balanced and unbalanced dataset. To conduct this study, we have done a total of twelve experiments. The performances of the models are evaluated using accuracy, precision, recall, and f1_score evaluation metrics. We have also explained the classification model using LIME and SHAP feature relevancy explanation techniques, to enhance the explainability and interpretability of the classification model by solving the black-box nature of the algorithms. In this study, the best classification model is identified using the accuracy of the developed classification model. Then, XGBoost is selected as the best algorithm that classifies the dry bean using the balanced dataset with 92.5065% accuracy. At the end of this conclusion, the researcher recommended that other researchers do: A dry bean classification model by including additional features of the dry bean like 3D features or the suture axis of the bean. The future researcher can also conduct a dry bean classification model using any other advanced algorithms to improve the performances and develop a mobile application.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] A. Desole, “Dry Bean Dataset Analysis,” Math. Mach. Learn., 2022.

[2] S. K. Sathe, “Dry bean protein functionality,” Crit. Rev. Biotechnol., vol. 22, no. 2, pp. 175–223, 2002, doi: 10.1080/07388550290789487.

[3] Y. Long, A. Bassett, K. Cichy, A. Thompson, and D. Morris, “Bean split ratio for dry bean canning quality and variety analysis,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2019-June, pp. 2665–2668, 2019, doi: 10.1109/CVPRW.2019.00323.

[4] M. Koklu and I. A. Ozkan, “Multiclass classification of dry beans using computer vision and machine learning techniques,” Comput. Electron. Agric., vol. 174, no. May, p. 105507, 2020, doi: 10.1016/j.compag.2020.105507.

[5] G. Słowiński, “Dry beans classification using machine learning,” CEUR Workshop Proc., vol. 2951, pp. 166–173, 2021.

[6] M. Moshinsky, “Dry Bean Classification,” Nucl. Phys., vol. 13, no. 1, pp. 104–116, 1959.

[7] M. Salauddin Khan et al., “Comparison of multiclass classification techniques using dry bean dataset,” Int. J. Cogn. Comput. Eng., vol. 4, no. March 2022, pp. 6–20, 2023, doi: 10.1016/j.ijcce.2023.01.002.

[8] P. E. D. Love, W. Fang, J. Matthews, S. Porter, H. Luo, and L. Ding, “Explainable Artificial Intelligence (XAI): Precepts , Methods , and Opportunities for Research in Construction Explainable Artificial Intelligence (XAI): Precepts , Methods , and Opportunities for Research in Construction,” pp. 1–58, 2022.

[9] B. E. Dejene, T. M. Abuhay, and D. S. Bogale, “Predicting the level of anemia among Ethiopian pregnant women using homogeneous ensemble machine learning algorithm,” BMC Med. Inform. Decis. Mak., vol. 22, no. 1, pp. 1–11, 2022, doi: 10.1186/s12911-022-01992-6.

[10] Anynomous, “Data Preprocessing Techniques for Data Mining,” Science (80-.)., p. 6, 2011.

[11] A. M. Dymond, R. W. Coger, and E. A. Serafetinides, “Data preprocessing applied to human average visual evoked potential P100-N140 amplitude, latency, and slope,” Psychiatry Res., vol. 3, no. 3, pp. 315–322, 1980, doi: 10.1016/0165-1781(80)90061-X.

[12] N. H. Son, “Data cleaning and Data preprocessing,” 2011, [Online]. Available: http://www.mimuw.edu.pl/~son/datamining/DM/4-preprocess.pdf

[13] S. B. Kotsiantis and D. Kanellopoulos, “Data preprocessing for supervised leaning,” Int. J. …, vol. 1, no. 2, pp. 1–7, 2006, doi: 10.1080/02331931003692557.
[14] S. Manikandan, “Data transformation,” J. Pharmacol. Pharmacother., vol. 1, no. 2, p. 126, 2010, doi: 10.4103/0976-500x.72373.
[15] J. W. Osborne, “Notes on the use of data transformations,” Pract. Assessment, Res. Eval., vol. 8, no. 6, 2003.
[16] I. Journal and C. Science, “Class Imbalance Problem in Data Mining : Review,” vol. 2, no. 1, 2013.
[17] R. P. Ribeiro, “SMOTE for Regression,” no. October 2015, 2013, doi: 10.1007/978-3-642-40669-0.
[18] E. F. Swana, W. Doorsamy, and P. Bokoro, “Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset,” Sensors, vol. 22, no. 9, 2022, doi: 10.3390/s22093246.
[19] “Imbalanced Classification in Python: SMOTE-Tomek Links Method | by Raden Aurelius Andhika Viadinugroho | Towards Data Science.” Accessed: Mar. 30, 2023. [Online]. Available: https://towardsdatascience.com/imbalanced-classification-in-python-smote-tomek-links-method-6e48dfe69bbc
[20] “Training, Validation and Testing Data Explained | Applause.” Accessed: Aug. 16, 2021. [Online]. Available: https://www.applause.com/blog/training-data-validation-data-vs-test-data
[21] M. K. Uçar, M. Nour, H. Sindi, and K. Polat, “The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets,” Math. Probl. Eng., vol. 2020, 2020, doi: 10.1155/2020/2836236.
[22] M. J. Healy, “Statistics from the inside. 15. Multiple regression (1).,” Arch. Dis. Child., vol. 73, no. 2, pp. 177–181, 1995, doi: 10.1136/adc.73.2.177.
[23] R. G. Mantovani, A. L. D. Rossi, E. Alcobaça, J. C. Gertrudes, S. B. Junior, and A. C. P. de L. F. de Carvalho, “Rethinking Defaults Values: a Low Cost and Efficient Strategy to Define Hyperparameters,” 2020, [Online]. Available: http://arxiv.org/abs/2008.00025
[24] B. H. Shekar and G. Dagnew, “Grid search-based hyperparameter tuning and classification of microarray cancer data,” 2019 2nd Int. Conf. Adv. Comput. Commun. Paradig. ICACCP 2019, no. February, pp. 1–8, 2019, doi: 10.1109/ICACCP.2019.8882943.