Fruit Classification Project

1st Garv Makkar
Computer Science with Artifial Intelligence
Indraprastha Institute of Information Technology, Delhi
garv21530@iiitd.ac.in

2nd Mohit
Computer Science with Artifial Intelligence
Indraprastha Institute of Information Technology, Delhi
mohit21542@iiitd.ac.in

***Abstract*—This document is a report on a statistical machine learning project. It is about a fruit classification problem. This report gives detailed information about data, algorithms and results.**

I. Introduction

The objective of this project is to predict labels of a given dataset. Kaggle provides the dataset, which contains various features. The project follows a systematic approach involving the application of clustering algorithms for label generation, dimensionality reduction algorithms for feature selection, and outlier detection algorithms for removing outliers. The ultimate goal is to develop a robust model using appropriate classification algorithms, ensemble methods, and k-fold cross-validation techniques for validation. This report provides a detailed description of the methods employed in the project.

II. About the Data

The dataset consists of 4096 features and a target feature with 20 classes. THere were 1215 rows in data. Each fruit has further two types, ripe or raw.

III. Methodology

The methodology used in this project involves the following steps:

A. Pre-Processing with Dimensionality Reduction

Two dimensionality reduction algorithms are used to select relevant features from the dataset. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were employed as dimensionality reduction algorithms. PCA is a commonly used technique for reducing the dimensionality of a dataset while preserving its important information. It identifies the directions along which the data varies the most and projects the data points onto these directions, known as principal components, which are orthogonal to each other. LDA, on the other hand, is a supervised dimensionality reduction technique that maximizes the separation between classes while minimizing the within-class variance. LDA is particularly useful when the goal is to improve the discrimination between different classes in the dataset, which is exactly what we require. We chose number of components of PCA = 375, and for LDA, we chose 19. These values were fitting best for our model.

B. Pre-Processing with Outlier Detection

An outlier detection algorithm based on the Local Outlier Factor (LOF) score was used in this project. LOF calculates the density of data points in their local neighborhood and compares it to the density of their neighbors' neighborhoods. Data points with significantly lower density than neighbors are potential outliers with higher LOF scores. A threshold was set based on the distribution of LOF scores, and data points with LOF scores above this threshold were considered outliers and removed from the dataset. This step was performed to improve the accuracy and reliability of the model by removing noisy data points that may negatively impact the performance of classification algorithms. LOF provides a robust approach to identifying and removing outliers, enhancing the data quality used for clustering, dimensionality reduction, and classification. Our parameters were neighbours = 50, contamination = 0.01 and we got 10 outliers.

C. Clustering and adding this new feature

A clustering algorithm was used to generate cluster labels as additional features for the dataset. We performed many types of clustering, like kmeans, meanshift, agglomerative, spectral, dbscan and birch. Out of these, K-means clustering was employed as a clustering algorithm to generate cluster labels for the dataset. K-means is an unsupervised learning algorithm that groups data points into clusters based on their similarities. It involves iteratively assigning data points to the closest centroid and updating centroids until convergence is reached. The optimal number of clusters was determined using silhouette analysis, which measures the quality of clustering results. The cluster labels generated by K-means were used as additional features for the dataset in subsequent project steps. Our parameters for kmeans were 4 clusters, 40K iterations and eklan algorithm.

D. Classification

Several classification algorithms were used to develop the classification model. Random Forest, Logistic Regression, Decision Trees and K-Neighbour ClassifierXGBoost were employed, which commonly used algorithms for classification tasks. Their parameters were decided by using loops and the ones giving best validation accuracy were chosen to be hyperparameters. Further, ensembling was done to improve them.

E. Ensembling

Ensemble methods were used to further improve the performance of the classification model. Adaboost and Bagging were employed as the ensembling techniques. Adaboost is an ensemble method that combines the predictions of multiple base models by assigning higher weights to misclassified samples, and then using these weighted samples to retrain the base models iteratively. The final prediction is made by combining the predictions of these base models based on their weighted accuracy. Bagging, on the other hand, is an ensemble method that creates multiple subsets of the original dataset by sampling with replacement, and then trains a base model on each of these subsets. The predictions of these base models are combined to make the final prediction, usually by taking a majority vote or averaging their outputs.

F. Checking our model

To check our model's performance validation accuracy was calculated. The k-fold cross-validation technique was used for validating the model. The dataset was randomly divided into k equal-sized folds, and the model was trained k times, with each fold used as the validation set once and the remaining k-1 folds used as the training set. The model's performance was evaluated using this validation accuracy, the average performance across all folds. We chose k has 5 to split it in 5 parts so that we get 80:20 ratio. We are getting 83 percent as our validation accuracy.

IV. Results

The model was evaluated using the k-fold cross-validation technique, with performance metrics calculated for each fold. The average performance across all folds was then calculated and reported as the final performance of the model. The model achieved a validation accuracy of 83 percent. Upon submission, the model received a public score of 86 percent and a private score of up to 82 percent, indicating a satisfactory overall performance in accurately classifying the labels in the dataset.

V. Conclusion

In this project, we developed a robust model for predicting labels in a given dataset by following a systematic approach involving clustering algorithms for label generation, dimensionality reduction algorithms for feature selection, outlier detection algorithms for removing outliers, and various classification algorithms and ensemble methods for model development. The model achieved a satisfactory validation accuracy of 82 percent and performed well in the Kaggle challenge with a public score of 86 percent and a private score of up to 83 percent. This project demonstrates the importance of careful data preprocessing, feature selection, and model validation techniques in developing accurate and reliable machine learning models.

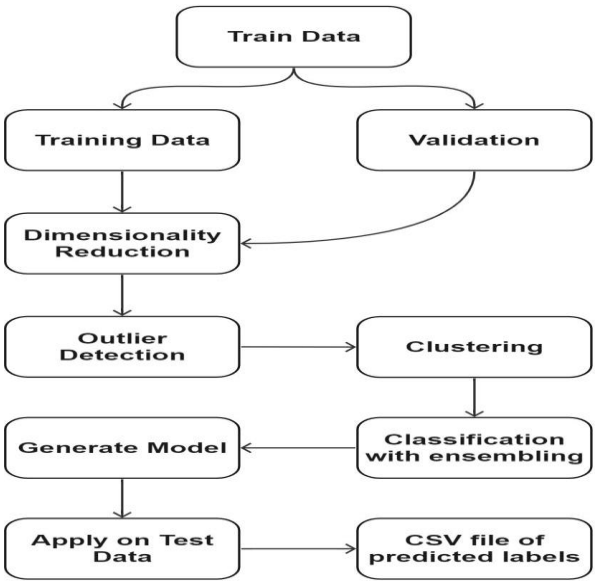

Fig. 1. Workflow

VI. Acknowledgments

We would like to express our sincere gratitude to Professor Koteshwar for his guidance and support throughout this project. We are thankful for his continuous encouragement and feedback, which greatly contributed to the successful completion of this project. We would like to express our gratitude to Kaggle for providing the dataset and hosting the challenge that motivated this project. We would also like to thank our mentors and advisors for their guidance and support throughout development.

VII. References

- Algorithms - https://scikit-learn.org/stable/
- PCA - http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- LDA - http://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html
- Kmeans - http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- LOF - http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html
- Logistic Regression - http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- Random Forest - http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- Decision Tree - http://scikit-learn.org/stable/modules/tree.html
- K Nearest Neighbour Classifier - http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- Adaboost - http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
- Bagging - http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html