SML Project Report

1st Nalish Jain
dept. CSE
Indraprastha Institute of Information Technology
Delhi, India
nalish21543@iiitd.ac.in

2nd Aditya Raj
dept. CSE
Indraprastha Institute of Information Technology
Delhi, India
aditya21512@iiitd.ac.in

I. Introduction

This document presents a succinct yet informative summary of the three machine learning models employed in our fruit classification project. By utilizing advanced classification techniques such as Logistic Regression, Random Forest Classifier and Convolutional Neural Networks (CNN) we were able to achieve highly accurate classification results for the given fruit dataset. This summary aims to provide a clear understanding of the models' capabilities and suitability for the task at hand.

II. Pre-Processing

Our fruit classification project involved the implementation of two widely used feature extraction techniques, namely Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

- Depending on the model being utilized, we also employed standardization techniques on the data-set to normalize the feature values and create a uniform distribution, thus reducing bias towards certain features that might contain drastically higher or lower values than others.
- We also utilized PCA to perform dimensionality reduction in an unsupervised manner. Subsequently, we employed LDA in a supervised fashion to further decrease the number of features to 19: the number of distinct classes - 1.
- By incorporating both supervised and unsupervised learning algorithms, we were able to effectively reduce the dimensionality of the "Fruit" dataset while preserving the pertinent information for accurate classification. This approach highlights the power of utilizing a combination of feature extraction methods to enhance the performance of machine learning models.
- As suggested in the project description, we explored the potential benefits of outlier removal and feature engineering through clustering algorithms; subsequently conducting extensive testing using various approaches, including Isolation Forests and LOF for outlier removal, and HDBSCAN combined with KMeans for clustering. However, our experimentation did not reveal any significant improvements in the classification results. In fact, in some cases, we even observed a decrease in accuracy. As a result, we ultimately decided not to incorporate these techniques into our final model.

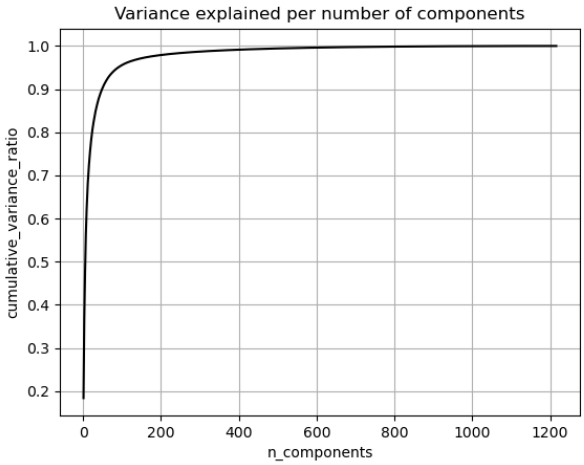

Fig. 1. Plotting variance explained per number of components in PCA

III. Hyperparameter Tuning

- To effectively tune the hyperparameters of our fruit classification model, we utilized *k-fold cross-validation* along with *Bayesian optimization*. The k-fold cross-validation technique allowed us to obtain a more reliable estimate of the model's performance by partitioning the dataset into k equal subsets, with each subset serving as the testing set once while the remaining k-1 subsets were used for training.
- The objective function, to be used for optimization, was defined as the negative of the mean of the validation accuracies obtained for each of the k partitions. We then minimized this function over a search space consisting of ranges of values for the hyper-parameters present in our model. This minimization was achieved using Bayesian optimization, which uses a probabilistic algorithm to determine the optimal range for the parameters being passed to its objective function.
- We also used various techniques to determine rough estimates for the ranges of our parameters. For instance, based on the information presented in Figure 1, it can be inferred that a cumulative variance of over 95% can be achieved for the dataset through the utilization of merely 200 components after performing PCA, while

nearly all of the variance can be captured by utilizing 1000 components. With these insights in mind, we set the range of consideration for PCA components from 200 to 1000. Post optimization, values obtained were 210, 375 and 450.
- For LDA, after testing with various values of `n_components`, it was concluded that a value of 19 yielded the best results. This finding aligns with theoretical expectations, as in this case, LDA is being utilized for dimensionality reduction purposes, while maintaining the essential class discrimination information.
- Performing cross validation in this way not only aided us in increasing our training accuracy, but it also ensured the model was not over-fitting to the training data and could generalize well to new samples.

IV. Classification Algorithims Used

A. Logistic Regression

Following thorough preprocessing and rigorous cross-validation, we employed the logistic regression algorithm to train our model using the complete training dataset with default parameters. We subsequently utilized the trained model to predict the labels of the test dataset. Upon submission of the resulting file, we achieved an impressive accuracy score of 85.5% on a 50% split of the test data set. These results validate the effectiveness of our feature extraction and hyperparameter tuning strategies, as well as the suitability of logistic regression for this fruit classification task.

B. Random Forests

- We utilized the Random Forest Classifier from the scikit-learn library for our classification task.
- In addition to the pre-processing steps, we included specific parameters of the classifier in our optimization search space.
- The `n_estimators` parameter determines the number of decision trees used by the algorithm, and we considered a wide range of values while optimizing this parameter. A higher value for n_estimators can improve training accuracy, but also increase the risk of overfitting.
- The `max_depth` parameter determines the distance of the primary predictor from the leaf nodes, and we optimized this parameter following similar guidelines as n_estimators.
- We also optimized the `min_sample_split` and `max_features` parameters, although they were found to be close to their default values, likely owing to the fact that we had performed dimensionality reduction prior to the use of the classifier.
- Our combination of classifier specific parameter optimization and appropriate pre-processing resulted in an accuracy of 82.608% on the first half of the testing data, which then generalized well to 82.692% on the private leaderboard **which was our best submitted accuracy**.

C. Convolutional Neural Network

- Constructed a basic Neural Network using a Keras Sequential model.
- Our main focus was on optimizing the number of hidden layers, the number of neurons in each respective layer, and evaluating the potential of utilizing dropout layers to improve performance.
- Due to the limited size of the dataset, we selected a low value to start the range for the number of neurons in the each layer. Owing to the same reason, dropout layers seemed to provide inconsistent results and even at best were not an improvement over the regular model.
- We also carefully considered the `epochs` parameter, selecting a small range of values to prevent overfitting, while also optimizing the `batch_size` parameter to reflect our choice of epochs.
- By optimizing these parameters along with appropriate pre-processing techniques, we achieved an accuracy of 83.091% on the public leaderboard.

V. Literature Review

A. Logistic Regression

1) Logistic regression is a versatile machine learning algorithm that can also be used for image classification tasks involving binary outcomes, such as distinguishing between images of cats and dogs. By training a logistic regression model on a dataset of labeled images, the algorithm can learn to recognize patterns and features such as color, texture, and shape that are indicative of each category. This makes logistic regression a powerful tool for automated image analysis and classification.
2) Logistic regression can be used in marketing analytics that can help businesses predict the probability of customers taking a specific action, such as making a purchase or subscribing to a service. By analyzing customer data, such as age, income, and browsing behavior, companies can customize their marketing strategies to increase conversion rates and maximize profitability.

B. Random Forest

1) Random forests are versatile machine learning models that can be applied to many practical scenarios. For instance, in finance and e-commerce, random forests can be used to detect fraudulent transactions by learning from historical data on customer behavior, transaction frequency, and purchase patterns. This approach can accurately identify suspicious activities while minimizing false positives.
2) In the medical field, random forests can assist doctors in making more accurate diagnoses by analyzing large datasets of medical records, including symptoms, lab results, and patient demographics. This method can help classify patients into different disease categories, identify high-risk patients, and suggest appropriate treatments.