CSE342: SML Course Project Report

Divyajeet Singh (2021529)
Computer Science & Engineering Dept.
IIIT-Delhi, India
divyajeet21529@iiitd.ac.in

Siddhant Rai Viksit (2021565)
Computer Science & Engineering Dept.
IIIT-Delhi, India
siddhant21565@iiitd.ac.in

***Abstract*—This report presents the findings from the course project for *Statistical Machine Learning*, held as a Kaggle contest to build a machine learning model for fruit-classification. The goal was to achieve the highest accuracy across a leaderboard of 50+ teams. A Logistic Regression model achieved the best score, attaining an accuracy of 0.825+.**

***Index Terms*—(Multi-class) Classification, Dimensionality Reduction, Anomaly Detection, Logistic Regression**

I. Introduction

This document acts as a report for the Kaggle challenge (22 Mar - 18 Apr), held as the course project for *Statistical Machine Learning (Winter 2023)*. Given a limited dataset of 1216 images of 4096 dimensions each, the goal was to build a robust machine learning model to classify the images into one of 19 classes. Various approaches were discovered to tackle this multi-class classification problem. Appropriate advanced algorithms covered in the course were used to preprocess the data and build a classifier.

This report provides a detailed description of the methods used, discarded, and the results obtained during the study.

II. Literature Review

A. Classification (Statistical)

Statistical classification is the problem of identifying which of a set of categories/sub-populations an observation belongs to. In the domain of machine learning, classification is a supervised (usually frequentist) learning technique that, given a labeled set of training data, infers a function to assign new examples into one or more discrete categories.

The final submission of this project makes use of the multinomial variant of Logisitic Regression to solve the given problem.

1) Logistic Regression: A statistical method used to analyze and model between a (usually binary) dependent variable and one or more independent variables. This regression analysis technique aims to predict the probability of an unseen example belonging to a particular class.

B. Dimensionality Reduction

Dimensionality reduction is the transformation of data from a high-dimensional space onto a low-dimensional space in a way that retains the most meaningful and prominent properties of the original data, ideally close to its intrinsic dimension. Owing to the curse of dimensionality, dimensionality reduction is a common preprocessing step in machine learning.

This project makes use of the Principal Component Analysis and Linear Discriminant Analysis algorithms for projection of features onto a lower-dimensional space.

1) Principal Component Analysis (PCA): A statistical method that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The first principal component accounts for the highest variability in the data, and each succeeding component has the highest possible variance under the constraint that it is orthogonal to all preceding components.

2) Linear Discriminant Analysis (LDA): A supervised statistical procedure used for dimensionality reduction by separating the features in a way that maximizes inter-class and minimizes intra-class variance. The resulting features are linear combinations (weighted sums) called *discriminant functions* of the original features that best explain the dataset and are used to project the data onto a lower-dimensional space.

C. Anomaly (Outlier) Detection

In data analysis, anomaly detection refers to the identification of observations which deviate significantly from the majority of the data and do not conform to a well-defined notion of normal behaviour. Such examples may arouse suspicions of being generated by a different mechanism, or appear inconsistent with the remainder of the dataset.

The Local Outlier Factor procedure was used to detect outliers in the dataset.

1) Local Outlier Factor (LOF): A density-based statistical technique that computes the local density deviation of a given data point with respect to its neighbours to identify if it is an outlier. It is a measure of how isolated a data point is with respect to the surrounding neighborhood. The most anomalous observations are those that have a substantially lower density than their neighbours and are hence considered outliers.

III. Methodology

This section provides a detailed description of the methods used to preprocess the data and build the classifier.

CSE342: Statistical Machine Learning (Winter 2023), Dr Koteswar Rao Jerripothula, IIIT-Delhi.

A. Data Analysis and Sanity Check

The main analysis is done to identify the distribution of the classes. This is done by generating a a bar graph of frequencies of each class in the training dataset. The dataset was found to be very slighlty skewed in terms of class-distribution. The class `Leeche_Raw` appeared in 37 samples only, while the class `Banana_Ripe` had the highest number of samples at 86. An average of 60 samples were present in each class.

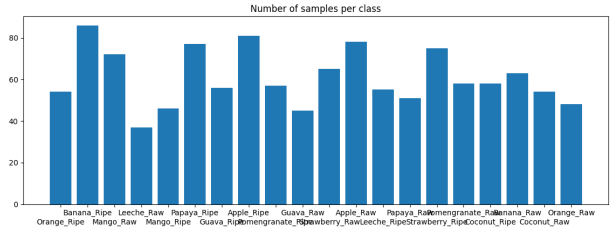

Fig. 1. Distribution of classes in the training dataset

The distribution of the classes shown in Fig. 1 was almost uniform.

As a sanity check on the dataset, 10 images selected at random were plotted in a 64×64 plot, but the results were not meaningful. On repeated testing, it was concluded that the features of the dataset must have been shuffled (or passed through a neural network) to prevent cheating in the contest by visualizing the data and submitting a manually labeled CSV file.

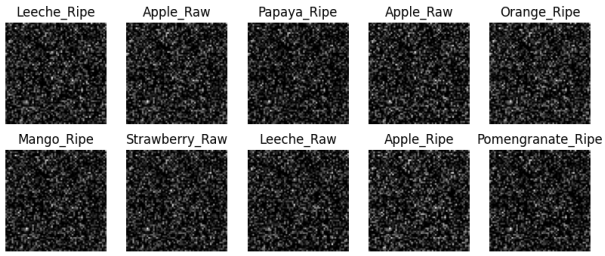

Fig. 2. Sanity check on the dataset

The sanity check (a trial shown in Fig. 2), hence, failed. Each selected sample did not conform to an image of a fruit. This was a clear indication that the dataset was not in its original form.

B. Data Preprocessing

The labeled dataset provided for the challenge consisted of 1216 images of 4096 (64×64) dimensions each. The images were of 19 different fruits, with each class having a different number of images. The data was preprocessed to increase cross-consistency using the following steps:

1) Outlier Detection: It was clear that the dataset was sparse in 4096 (flattened) dimensions. Since it had a large number of dimensions and only a few training samples, it was only reasonable to tune the Local Outlier Factor algorithm to detect only the most extreme outliers. This resulted in the removal of less than 16 images from the dataset.

TABLE I
FINAL HYPERPARAMETERS FOR LOF

Hyperparameter	Value
`n_neighbors`	10
`contamination`	0.01

Table I shows the final hyperparameters selected for outlier detection using LOF.

2) Standardization: Standardization is a common preprocessing step to build classifers, especially those classifying images. The images were standardized to have zero mean and unit variance to ensure that the features of the dataset were on the same scale and to prevent any feature from dominating the classifier.

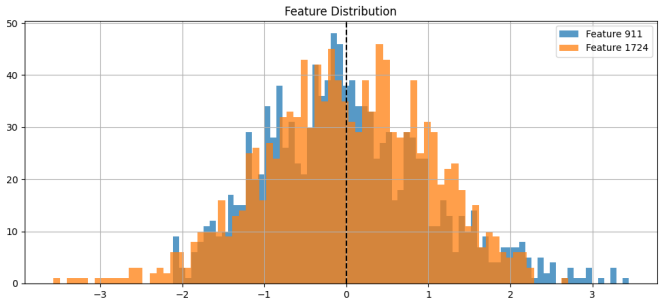

Fig. 3. Standardization of the dataset

As a check, two features were selected at random from the dataset and their distributions were verified. The results are shown in Fig. 3.

$$x_i^* = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}} \tag{1}$$

For each feature x_i for $i = 1, 2, \ldots, 4096$, x_i^* represents the standardized value of the feature and μ_{x_i} and σ_{x_i} are the mean and standard deviation of x_i. (1) was used to standardize the features.

It was noted that some features followed a gaussian distribution, while others were slightly skewed. No firm concluion about the distribution of individual features could be drawn. However, on multiple trials, it was clear that the features were on the same scale and the dataset was standardized.

3) Dimensionality Reduction: Owing to the sparsity of the dataset, dimensionality reduction was necessary to improve the performance of the classifier. The dimensionality reduction was performed using the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms.

With the features of these algorithms mentioned in Section II-B, it was concluded that an LDA after a PCA would be the best choice to extract the features that capture the most

variance in the dataset and project them onto an even lower-dimensional space to increase inter-class separation.

1) *PCA:* The dimensionality of the data was reduced to 256, to explain upto 98.5% of the variance in the dataset.

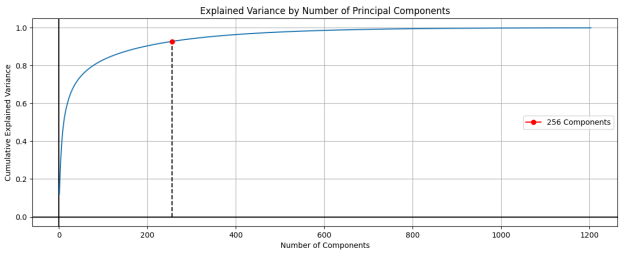

Fig. 4. Explained variance by Principal Components

Fig. 4 shows the explained variance by the principal components of the dataset. Attributing to the limited number of samples in the dataset, it was found that explaining any more variance led to overfitting.

2) *LDA:* A total of 18 features were extracted using LDA to minimize intra-class variance and increase the inter-class separation.
Fig. 5 shows the the inter-class separation. The figure consists of three subplots, each showing the final distribution of six individual classes.

4) Train-Test Split: The training data was finally split in an 80:20 ratio into training and validation sets. The training set was used to train the classifier and the validation set was used to tune the hyperparameters of the classifier.

C. Classification Model Selection

Various statistical classification techniques were covered in the course. However, due to the nature of the data, not all of them were suitable for the task. The following techniques were considered for the task:

1) k-Nearest Neighbours: A non-parametric technique that classifies a data point based on the class of its k-nearest neighbours. The algorithm, known for its simplicity, was found to be inadequate while testing to model the complexity of the dataset.

2) Gradient Boosting: A predictive technique that produces a model in the form of an ensemble of weaker prediction models, typically decision trees, where each subsequent learner tries to correct the errors made by its predecessors. Known for its invariability to overfitting, it was one of the first algorithms tested for the task.

However, the gradient boosting model was found to be dissatisfactory, as it was unable to generalize well on the validation set.

3) Random Forest: An ensemble learning method for classification that operates by constructing a multitude of decision trees and classifies using the majority vote of the trees. Although the dimensionality of the dataset was reduced, the random forest model was found to be overfitting on the training set and was unable to perform any better than the other models.

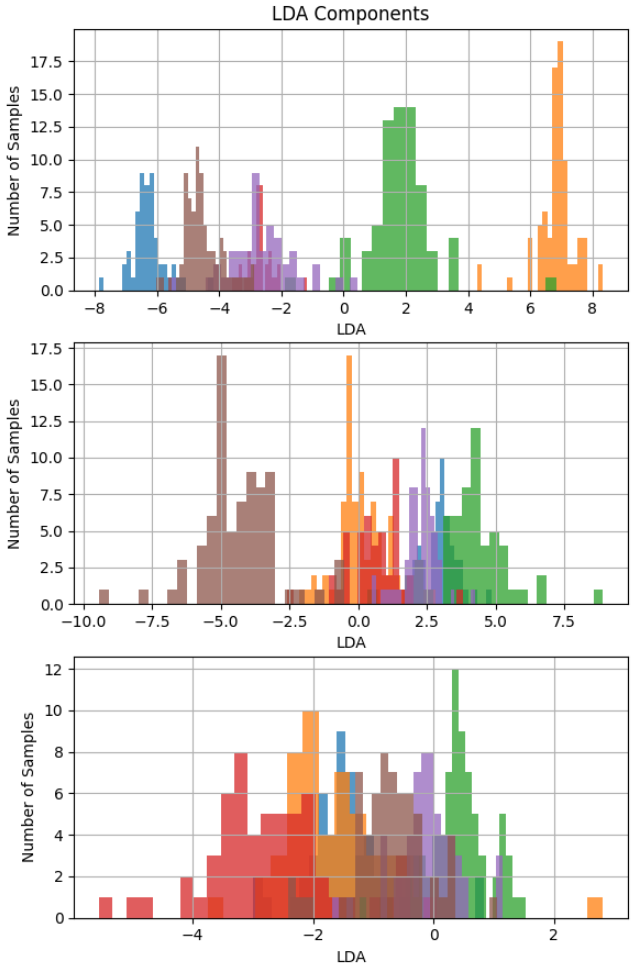

Fig. 5. Inter-class separation by LDA

4) (Gaussian) Naive Bayes (GNB): A probabilistic classifier based on Bayes' theorem with an assumption of independence between every pair of features. Some of the 18 final featues had slightly skewed dsitributions. Hence, an attempt was made to transform the dataset into a Gaussian distribution using the Box-Cox transformation to use GNB as it is known for its accurate performance on small datasets.

The model was proved to be unfit for the dataset as an appropriate conversion could not be achieved.

5) Neural Networks: A model having interconnected layers of mathematical functions (neurons) to apply activation functions on inputs from the previous layers to produce subsequent outputs. Deep networks were found to be overfitting the small dataset. A shallow neural network was found to be one of the most suitable models for the task, as it was able to learn well on the limited dataset available.

Surprisingly, the final score of the neural network model dropped when the full testing dataset was used. Hence, this model was not considered for the final evaluation.

6) Multinomial Logistic Regression: (Described in Section II-A1)
An extension of logistic regression performed substantially better than all other models, except for the neural network, which was a close second. It performed well on multiple trials, each involving a different subset as the training set and the validation set. This model was also able to generalize well on the testing dataset.

D. Hyperparameter Tuning

Having finally decided on a pipeline, the validation set was used to tune the hyperparameters of the classifier. A grid search was performed to exhaustively search for the optimal hyperparameters for the different steps of the final model.

A pipeline was constructed having PCA, LDA, and Logistic Regression as the steps. Invariably, they were tested with different lists of hyperparameters and cross validated using k-fold cross validation technique.

TABLE II
FINAL HYPERPARAMETERS FOUND BY GRID SEARCH

Algorithm	Hyperparameter	Value
PCA	`n_components`	256
LDA	`n_components`	18
Logistic Regression	`max_iter`	10,000
	`C`	0.1
	`solver`	sag

Table II shows the final hyperparameters selected for the final model. The results from the grid search were obtained after 5-fold cross validation on each combination of possible hyperparameters. The hyperparameters were selected based on the highest average accuracy obtained on the validation set.

IV. THE FINAL MODEL

The inferred hyperparameters for all algorithms were used to train the final model on the entire dataset. The model was trained using a pipeline, which was then used to predict the labels of the testing dataset.

Fig. 6 shows the flow of final model used for the classification, including all the steps performed on the dataset. The steps included in the enclosing box indicate the final pipeline.

A CSV file following the given format was generated as the prediction from the pipeline and submitted to the Kaggle competition for evaluation.

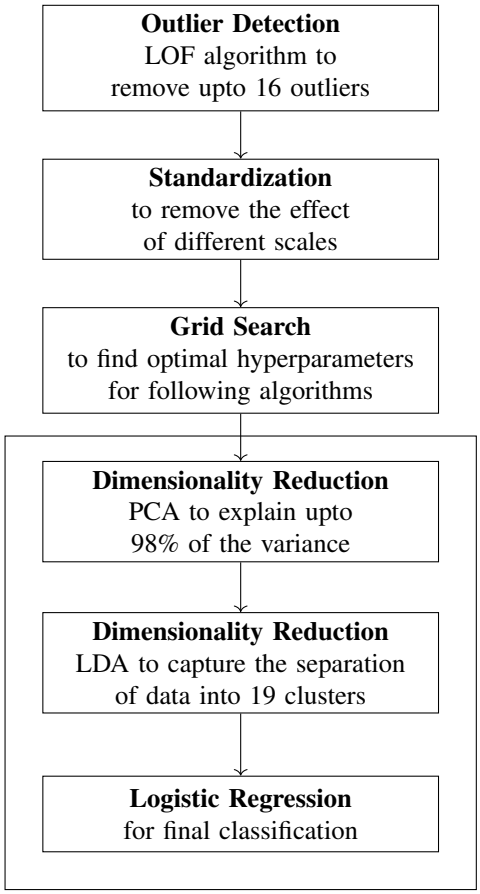

Fig. 6. Pipeline used for the final model

V. CONCLUSION

The final model was able to achieve a score of 0.82692 on the testing dataset, leading to a rank of 5 out of 57 teams on the leaderboard.

ACKNOWLEDGMENT

The authors would like to extend their sincerest gratitude to Dr Koteswar Rao Jerripothula *(Computer Science & Engineering Dept., IIIT-Delhi)* for their invaluable guidance throughout the project. Their insightful feedback and expertise have been instrumental in shaping this project into its final form.

The authors would also like to thank Niranjan Sundararajan (2020090), Vibhu Dubey (2020150), and Raghav Sahni (2020533), the teaching assistants of the course *Statistical Machine Learning (Winter 2023)*, for their unwavering support.

REFERENCES

[1] Dr Koteswar Rao Jerripothula, lecture notes, Statistical Machine Learning (Winter 2023), IIIT-Delhi.
[2] scikit-learn, https://scikit-learn.org/stable/