SML Project Report

Rahul Oberoi and Vishal Singh

***Abstract*—This study presents an analysis of various machine learning models for fruit classification. A dataset was provided that consisted of multiple features with their corresponding labels. The models that were tested included K-Nearest Neighbors (KNN), Linear Regression, Decision Trees, Random Forest, and Logistic Regression. The accuracy of the models was evaluated by comparing the predicted labels with the actual labels in the test dataset. The results showed that Logistic Regression had the highest accuracy for fruit classification. Further optimization of the Logistic Regression model was carried out using Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Outlier Factor (LOF), and K-Means clustering techniques. The hyperparameters of the model were tuned using Grid Search Cross Validation to achieve better results. Overall, the study demonstrates the effectiveness of machine learning models in accurately classifying fruits based on their features.**

I. Details of the method used:

A. This code performs the following steps:

- Imports necessary libraries.
- Reads in the training and testing data from CSV files.
- Removes any missing values from the training data and performs Principal Component Analysis (PCA) to reduce the number of dimensions in the data.
- Performs hyperparameter tuning for PCA using CrossK-Fold validation technique.
- Performs Linear Discriminant Analysis (LDA) to further reduce the number of dimensions in the data.
- Removes outliers using Local Outlier Factor (LOF) and KMeans clustering.
- Applies the trained model to the test data and generates predictions.
- Saves the predictions to a CSV file.

In order to find the best machine learning model for our dataset, we experimented with various models such as K-Nearest Neighbors (KNN), Linear Regression, Decision Trees, and Random Forest. After comparing their performance, we found that Logistic Regression provided the highest accuracy. Therefore, we decided to focus on this model and further optimized it.

We used Principal Component Analysis (PCA) to reduce the dimensionality of our input data. This technique helped us to reduce the number of input variables while retaining the most important information from the data. We reduced the number of input variables from 4096 to 372, while still retaining majority of the original variance.

To further separate our classes, we used Linear Discriminant Analysis (LDA) which finds a projection of the data that maximizes the separation between the classes. We reduced the number of input variables from 372 to 19 using LDA.

We also used Local Outlier Factor (LOF) to identify and remove outliers from our training set. Outliers can negatively affect the performance of machine learning models, so removing them can improve accuracy. We set the contamination parameter to 0.05, which means that we expect 5% of the data to be outliers.

Furthermore, we added a new feature to our input data by using K-Means clustering. This helped us to capture the underlying structure of the data and potentially identify any subgroups or clusters within the data. We set the number of clusters to 5, which was determined through trial and error.

After these optimizations, we decided to focus on tuning the hyperparameters of our Logistic Regression model. We used Grid Search Cross Validation to search for the best combination of hyperparameters. We varied the solver, penalty, and C parameters to find the best model.

Overall, we found that optimizing the Logistic Regression model using PCA, LDA, LOF, and K-Means clustering, along with tuning hyperparameters using Grid Search Cross Validation, significantly improved the performance of our model.

II. Brief description of different applications of the methods used:

Principal Component Analysis (PCA) is a dimensionality reduction technique used to identify patterns in data and to express the data in a reduced number of dimensions. It is a widely used technique in machine learning, statistics, and data analysis. PCA involves transforming the original data set into a new coordinate system in which the first coordinate (or principal component) has the largest possible variance, the second coordinate has the second largest variance, and so on. These principal components are linear combinations of the original variables that are uncorrelated with each other. The goal of PCA is to reduce the number of dimensions of the data while retaining as much of the original variance as possible. This can help to simplify the data, remove noise, and improve the performance of machine learning models by reducing the number of features they need to consider.

K-fold cross-validation is a method of evaluating the performance of a machine learning model by dividing the available data into k subsets, or "folds", of equal size. The process involves iteratively training the model on k-1 of the folds and using the remaining fold as a validation set. This is repeated k times, with each of the k folds used once as the validation set. The model's performance is then evaluated by averaging the results obtained from each of the k iterations. K-fold cross-validation is commonly used to estimate the performance of a model on new, unseen data, as it provides

a more reliable estimate of the model's performance than a single train-test split. It also helps to mitigate the risk of overfitting by using multiple validation sets.

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique used to find a linear combination of features that best separates or discriminates between two or more classes in a dataset. LDA works by projecting the original dataset onto a lower-dimensional space while maximizing the separation between the classes. This is achieved by finding a set of linear discriminants, which are vectors in the lower-dimensional space that maximize the between-class variance and minimize the within-class variance. The goal of LDA is to reduce the dimensionality of the data while preserving as much of the class discriminatory information as possible. This can help to improve the accuracy and efficiency of classification models, especially when dealing with high-dimensional data.

Local Outlier Factor (LOF) is an unsupervised machine learning algorithm used for outlier detection in datasets. The algorithm identifies data points that are significantly different from their neighboring points in the feature space. LOF works by first calculating the local density of each data point by determining the number of neighboring points within a specified distance (known as the radius of the neighborhood). The local density is then used to calculate the local reachability distance of each data point, which represents how isolated or reachable the point is within its local neighborhood. The LOF score of each data point is then calculated based on the ratio of its local reachability distance to that of its neighbors. Data points with LOF scores significantly greater than 1 are considered to be outliers, as they are much less reachable or isolated than their neighboring points. LOF is useful for identifying anomalies in datasets where the underlying distribution is not known or the data is highly dimensional. It is widely used in applications such as fraud detection, network intrusion detection, and anomaly detection in sensor data. LOF is a non-parametric and unsupervised algorithm, which means that it does not require any prior assumptions about the underlying distribution of the data and can be applied to a wide range of datasets.

KMeans clustering is a popular unsupervised machine learning algorithm used to group similar data points into clusters based on their features. The algorithm is widely used in data mining, image processing, and computer vision. KMeans clustering works by partitioning a set of data points into k clusters, where k is a user-defined parameter. The algorithm randomly selects k initial centroids, one for each cluster, and then assigns each data point to the nearest centroid based on the distance between the data point and the centroid. Once the initial assignment is made, the algorithm iteratively refines the cluster centroids by calculating the mean of all the data points assigned to each cluster. The data points are then reassigned to the nearest cluster based on the updated centroids, and the process continues until convergence (i.e., until the centroids no longer change or the change falls below a specified threshold). The KMeans algorithm is widely used for clustering in a variety of applications, such as customer segmentation, recommendation systems, and anomaly detection. However, the algorithm is sensitive to the initial placement of the centroids and can sometimes converge to suboptimal solutions. Therefore, it is common to run the algorithm multiple times with different initializations and select the best result based on a clustering evaluation metric such as the Silhouette coefficient.

Grid search cross-validation (GridSearchCV) is a technique used in machine learning to tune the hyperparameters of a model in order to optimize its performance. GridSearchCV works by exhaustively searching a predefined hyperparameter space, which is a set of values for each hyperparameter that are of interest. For each combination of hyperparameters, the algorithm evaluates the model performance using cross-validation, which involves splitting the data into training and validation sets and evaluating the performance of the model on multiple folds of the data.

The performance of the model is typically measured using a predefined evaluation metric such as accuracy, F1 score, or mean squared error. The hyperparameters that result in the best performance on the validation data are selected as the optimal hyperparameters for the model. GridSearchCV is a computationally expensive method since it evaluates the model on multiple hyperparameter combinations using cross-validation. However, it is widely used in practice to select the optimal hyperparameters for a variety of machine learning algorithms such as support vector machines, decision trees, and neural networks.

Logistic Regression is a popular supervised machine learning algorithm used for binary classification problems, where the goal is to predict the probability of an event occurring or not. It is widely used in various applications such as fraud detection, marketing analytics, and medical diagnosis. Logistic Regression works by modeling the relationship between a dependent variable (binary) and one or more independent variables (numeric or categorical) using the logistic function. The logistic function is an S-shaped curve that maps any input value to a probability between 0 and 1. During training, the algorithm learns the optimal values of the model parameters (coefficients and intercept) by minimizing a loss function, typically the log loss or binary cross-entropy, using an optimization algorithm such as gradient descent. Once the model is trained, it can be used to make predictions on new data by computing the probability of the event occurring based on the input features and a decision threshold.

ACKNOWLEDGMENT

Lecture Notes
Documentation of the sklearn library