# A Comparative Analysis of Optimizing Classification Techniques on Fruits Images Dataset

Youssef George Fouad 19P9824, Kerollos Wageeh Youssef 19P3468, Jannah Ayman Amir 19P1728

CSE382 Data mining and Business Intelligence

## ABSTRACT

Data mining, also known as data dredging or data archeology, provides several techniques to extract new and interpretable information from given datasets. In this research, we compared the performance of several classification techniques, which are SVM, RBF, KNN, decision trees on pictures of 3 fruits (pineapple, cocos, and avocado) from fruits-360, an open-source dataset. The KNN classifier has given 97.9%, the highest accuracy score among the tested classifiers. In addition, we compare how principal component analysis (PCA), one of the most famous dimensionality reduction techniques, affects the performance of the classification tasks.

## INTRODUCTION

Data mining is mainly used to transform raw into useful, interpretable, and meaningful information. For instance, knowledge discovery in database (KDD) gives us methods and techniques to discover useful patterns and information from raw data. Figure 1 below gives an overview of the basic stages of the KDD process starting from the raw data, to selecting the target data we would work on, before preprocessing and transforming it to start the mining process that leads us to discovering some patterns, models, and useful information. This discovered information shall finally be evaluated and assessed before being used in business or further research use cases. Furthermore, it provides several techniques to extract new interpretable information from given datasets.

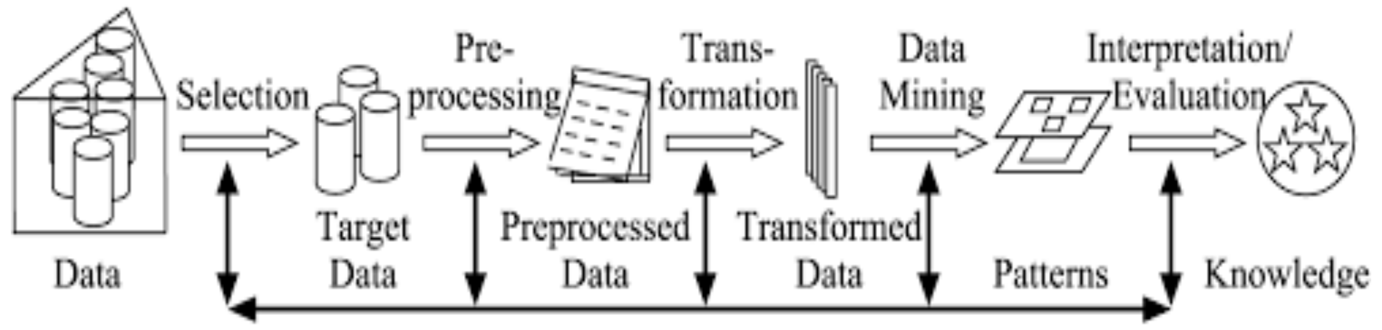

*Figure 1*

## APPROACH

Initially, we start data importing with selecting some fruits to work on their images, pineapple, cocos, and avocado images were chosen out of the 118 fruits and vegetables available in the fruits-360 dataset as they have a lot of similarities. Every image is represented by 100×100 pixels, and each pixel is represented by 3 color values, the red, the green, and the blue color channel, to form an RGB image.

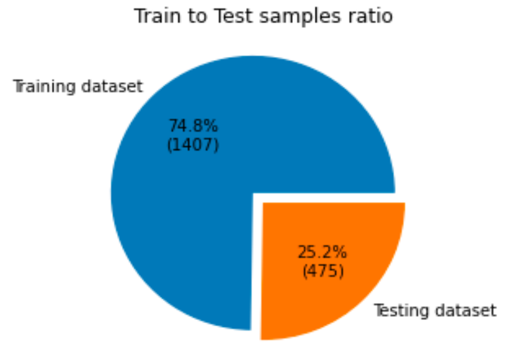

*Figure 2*

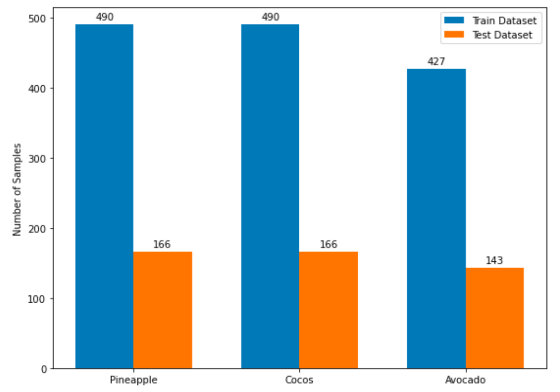

*Figure 3*

$$\text{RGB[A] to Gray:} \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

*Equation 1*

After importing the data, visualizations are made over the datasets to facilitate understanding. The pie chart shown in figure 1 illustrates the ratio percentage and count of the total of each of the training and testing datasets. As the graph suggests, there are 1407 training images and 475 testing samples. Digging deeper, the bar chart in figure 2 breaks down the previously mentioned numbers into the detailed distribution of fruits among training and testing sets.

In Data Preprocessing, manipulation of data before mining tasks is essential for the speed and power needed to perform a task. Therefore, our input images undergo several types of data transformation. Starting with Data discretization, where The given input are images that needs to be discretized into numbers before processing them. Each image is discretized into a (100x100x3) array, where values vary from 0 to 255. Next, Data compression, where we reduce the complexity of the data by compressing the RGB picture to grayscale ones (100x100), where each pixel is represented by a single value. This can be done using python library OpenCV that uses formula 1 in figure 3 to calculate a single value from the three values. After that comes Data linearization, where each instance of the data is required to be in linear form, therefore, all instances are flattened from 2D form into 1D array of (10,000) values. The last step, Dimensionality reduction, by applying principal component analysis (PCA) on our dataset to reduce the number of features to n-features.

After data preprocessing is done, each image is represented as a 1D numeric array. Subsequently, we will train and test several classification models, which are KNN, SVM, SVM with RBF kernel, and decision tree using the training and testing datasets. Every classification algorithm is discussed in two ways, once with dimensionality reduction using principal components analysis (PCA) and once without PCA. Moreover, K-fold cross validation technique is used to minimize the error margin of the predictive classification model built throughout the training phase. As figure 4 suggests, the training set is split into five folds, where we chose to use K = 5, and on each of the five iterations, one-fold is used as validation set, while the other four are used as training set. Subsequently, Looking at equation the equation in figure 5, we find out how this technique helps in minimizing the error margin of the built model. Where Error_i is the error of the i-th iteration and the total error of the model is the average of the five error percentages calculated at the five iterations.

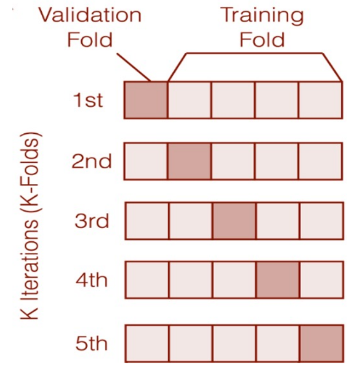

*Figure 4*

$$Error = \frac{1}{5}\sum_{i=1}^{5} Error_i$$

*Equation 2*

## RESULTS

Results were collected after the previous classification techniques were implemented (KNN, SVM, etc.) on both training and testing datasets. With the first technique, SVM, PCA is applied with 2 principal components as seen in figure 6 and took 63 mins and 45 secs to finish training and testing, producing 68% accuracy.

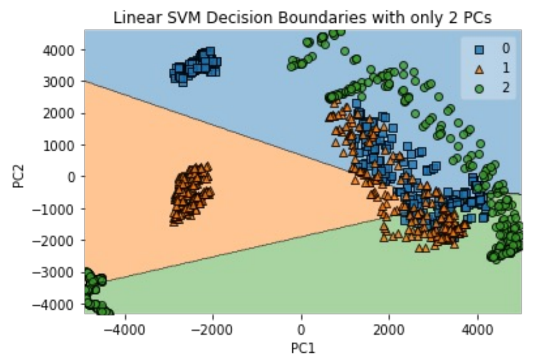

*Figure 5*

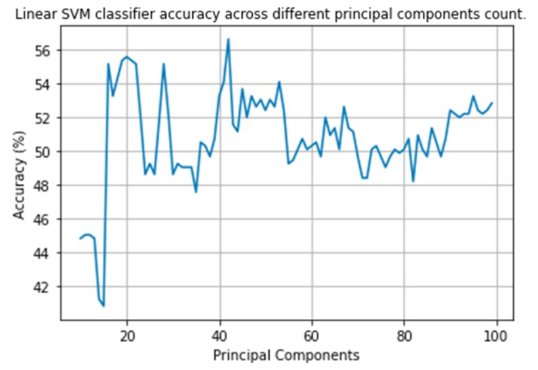

*Figure 6*

This led to using principal component values from 10-100 which resulted in 56.6% maximum accuracy at 42 principal components in 7 minutes and 47 seconds as shown in figure 7. Results without PCA are shown in table 1.

Next up, SVM with RBF kernel, where the decision hyperplane is described with centers and standard deviations unlike the previous SVM. By applying PCA on the training dataset with 2 principal components, it results in 88% accuracy in 7.3 seconds, and the decision hyperplane concluded is shown in figure 8. Since results are less accurate that linear SVM, we try using principal components ranging 1-100, and calculate the accuracy for every principal component, which is plotted into a graph, seen in figure 9. Now removing PCA from the equation, the dataset results in 96.8% accuracy consuming 17 seconds for training and testing. Using one of them depends on whether your application is time critical or not.

Our third applied technique is KNN, which consumes most of its time in the testing phase, since it calculates the distance from the data to each of the training point, chooses the closest K points, and decides the class. With PCA applied, the maximum accuracy achieved is 93.05% with 9 principal components, and 13 nearest neighbors, which consume 0.5 seconds to train and test the datasets. For visualization, head to figure 10 which shows a sample decision boundary with 2 PCs and 2-NN. However, without using PCA, The KNN algorithm is trained by the entire dataset of 10,000 dimensions while changing the number of the nearest neighbors. As the number of nearest neighbors increase the accuracy decreases as figure 10 shows, this is due to overfitting, as the algorithm doesn't learn from the training dataset anymore. A range of the nearest neighbors is used from 1-150 to train and test. The maximum accuracy is 97.9% with 3 nearest neighbors. With 3 neighbors, it reaches the accuracy in only 0.4 seconds.

Lastly, the decision tree, which is an algorithms of low bias yet high variance, a small change in the input would change the output greatly. After applying PCA with a range of 1-100, we notice in figure 11 that using a greater number of principal components does not increase accuracy, however, it dramatically decreases it while reaching a maximum accuracy of 88.8% in 0.3 secs at 3 principal components and tree of 22 depth. On the other hand, when discarding the PCA, The maximum accuracy reached is 97.5% with maximum depth of 7 levels consuming 1.7 seconds. To sum up, by using PCA, the time consumed is on the depth of the decision tree, 22 levels, while without using PCA the time consumed is due to the large number of features, which are 10,000.

| Classifier | No PCA | | PCA | |
|---|---|---|---|---|
| | Accuracy | Time (s) | Accuracy | Time (s) |
| KNN | 97.90% | 0.4 | 93.00% | 0.5 |
| SVM | 96.00% | 8.9 | 68.00% | 3825 |
| RBF | 96.80% | 17 | 93.70% | 0.6 |
| DT | 93.70% | 1.7 | 88.80% | 0.3 |

*Table 1*

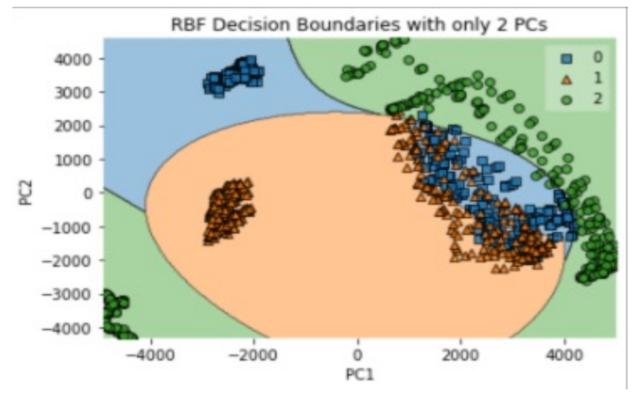

*Figure 7*

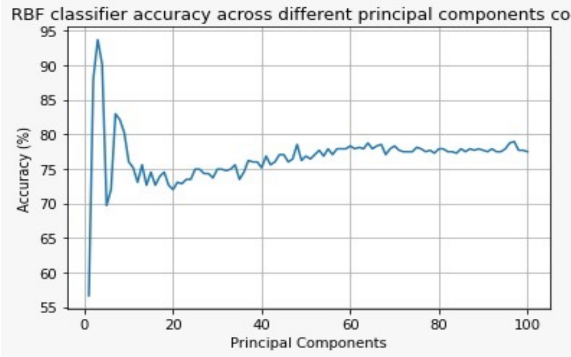

*Figure 8*

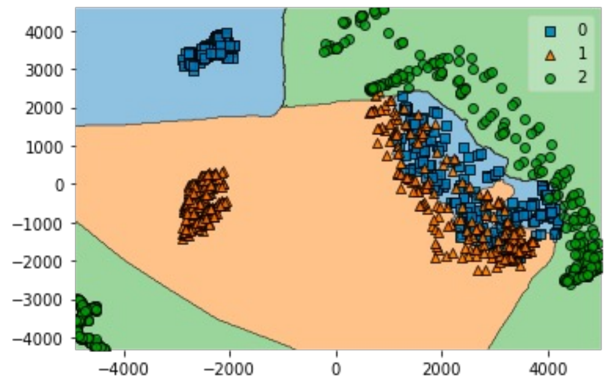

*Figure 9*

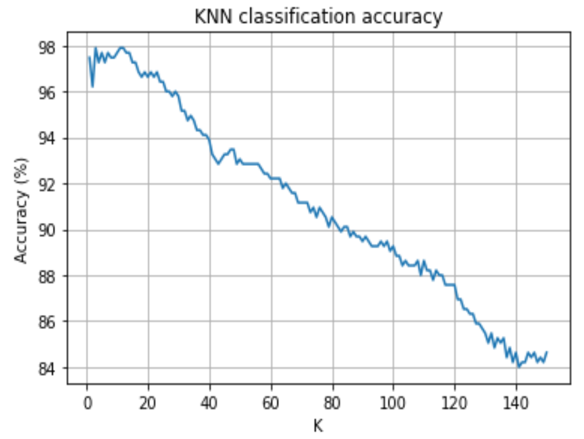

*Figure 10*

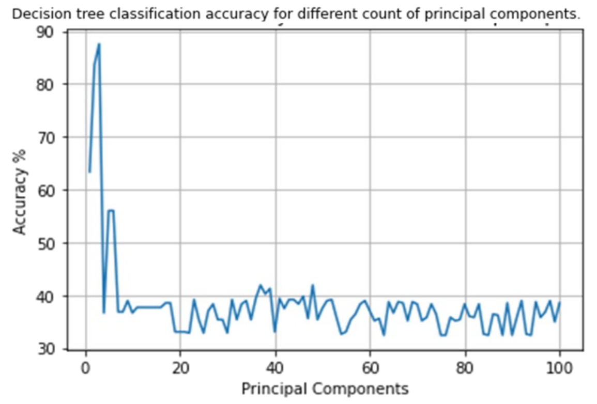

*Figure 11*

## CONCLUSION

To conclude, pineapple, cocos, and avocado were chosen to be studied from fruits-360 dataset. A new dataset was constructed using the tree fruits. Each pixel of every image of the new dataset is converted into three values, RGB, then each pixel is converted to grayscale and linearized to be of 10,000 dimensions. Dimensionality reduction was applied using PCA technique. The dataset with PCA and without PCA was given for 4 classifiers, linear SVM, SVM with RBF kernel, KNN and Decision tree, to test their accuracy. Some results of the 4 classifiers were gathered and shown in figure 12.

If it is decided not to use PCA before training the algorithms, this would be the order of preference of the classification algorithms: KNN, DT, RBF, SVM, with KNN to be the most appropriate algorithm. Otherwise, the order would be RBF, KNN, DT, SVM, with RBF is the most prioritized algorithm over the others.

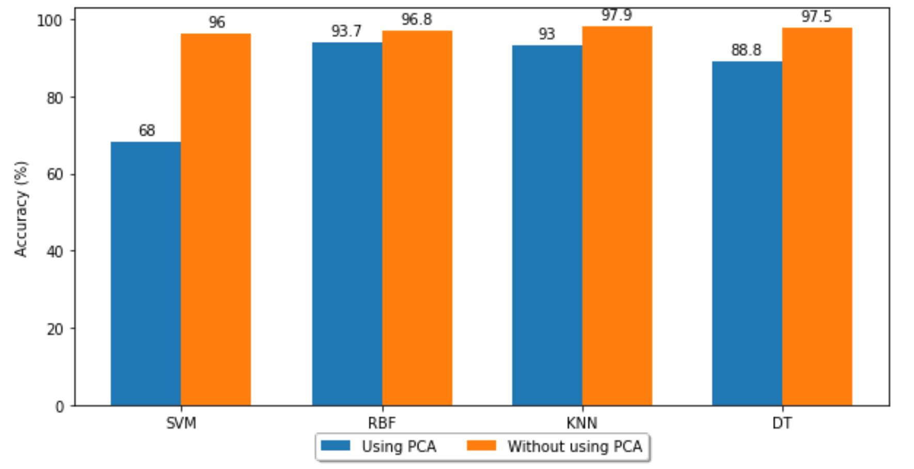

*Figure 12*

In case of critical testing time Decision tree without PCA is preferred with accuracy 97.5%, else the KNN without PCA takes precedence over the other classification algorithms achieving 97.9% accuracy.

## REFERENCES

[1] Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

[2] Y Mihai Oltean. (2018 February). Fruit-360, [2020.05.18.0]. Retrieved 2022.12.04 from kaggle.com/datasets/moltean/fruits

## ACKNOWLEDGMENT

We thank Dr. Nourhan Mohamed Zayed (Ain Shams University) and Eng. Mahmoud Soheil (Ain Shams University) fo their guidance and assistance in understanding essential data mining techniques.