UNC GREENSBORO

Evaluations among Machine Learning Techniques on Fruit Image Classification

Peng Chen

Supervisor: Dr. Shan Suthaharan

Master in Computer Science

Nov. 19, 2018

Evaluations among Machine Learning Techniques on Fruit Image Classification

Abstract

Image classification of fruits plays an important role in real life. Cashiers need to determine the appropriate price of the produce purchased by customers in supermarkets. It also makes an influence in the agriculture field, helping recognize fruit quality and disease problems. However, this issue has become very complex due to various properties of different types of fruits, not to mention some of them with similar colors and styles. Some researchers utilize support vector machine (SVM) algorithm to deal with, loaded images with grayscale, which cannot yield high accuracy. Some researchers try to use deep learning techniques, which consume significant amount of computing time, while training models. In our study, we utilize k-nearest neighbor (KNN), decision tree (DT), random forest (RF) and logistic regression (LR) algorithms to analyze different image datasets, loaded fruit images with grayscale and RGB, and make a comparison among these methods. All of these machine learning algorithms can reach high accuracy but behave differently among different datasets. Finally, the knowledge and experience that we gained can help us develop recommendations to solve future unexpected fruit classification problems and select suitable machine learning techniques.

Key Word: Image Classification; KNN; Decision Tree; Random Forest; Logistic Regression

1. Introduction

In real world development, big data analysis plays an important role in helping society grow and develop in many ways. As data grow bigger and bigger, it becomes quite significant to analyze big data with machine learning methods, either in scientific research fields or in real life society. By analyzing big data, useful information can be extracted to solve problems scientifically in real world. Business company can benefit from them, by predicting the price of their products or investing on advertisement. Even government also can benefit from data analysis quite a lot, like, they will know how to predict the time, strength and location of earthquakes, hurricanes and other natural disasters. In this paper, we provide an insightful method of how to analyze dataset and based on them, and then extract helpful information. Finally, we deployed tables and graphs to show the behaviors of these different machine learning techniques with different datasets and a detailed explanation was provided. More importantly, we gave a suggestion about how to choose an appropriate machine learning method for a specific dataset for higher accuracy.

2. Related Literature

Shan Suthaharan provides an insightful overview of machine learning technique with examples in Matlab language and R language, including how to analyze dataset and how to train machine learning techniques and even their cons and pros contained [1]. For feature extraction, Anderson et al. [2] present a new feature fusion method for fruit and vegetable images classification, which can lead to higher classification accuracy. S.Arivazhagan et al. propose a new feature fusion method using color and texture features [3]. Both of these two fusion methods can help increase classification accuracy, but they are very complex. For machine learning technique, O. Chapelle et al. illustrate the method of image classification using SVM and histogram. The paper shows that SVM can generalize well on difficult image classification problems where the only features are high dimensional histograms [4]. Hence, in our experiments, we tried to utilize both of the feature fusion methods and several techniques, including SVM. However, we found the two fusion methods improve the classification accuracy slightly, but increase the complexity significantly. Also, when we choose DT, RF, LR, KNN and SVM techniques, we found that running SVM can takes much long time than others. Thus, we do not use it in the final experiment results.

3. Materials and Methods

The whole data analysis process can be summarized as below.

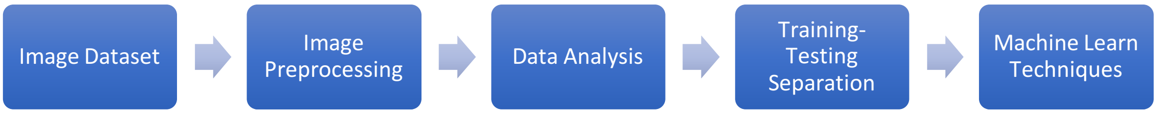

Figure 1: The Analysis Process

Figure 1 illustrates how the process performs in the master's project. There are 5 processes included and being executed in order. The first process is called 'Image Dataset', which illustrates the name, origin and the number, size and other features of images with click-links provided. Also, the way and reason of how to choose data from the downloaded dataset is explained as well. The second process is called 'Image Preprocessing', which is the process of reading images and converting images to number data. It includes two parts: BASIC and FURTHER. The third process is called 'Data Analysis', which includes analyzing the features and correlating of the data. To be specific, some of the characteristics are listed in section 3.3 with related figures shown to help explain the data. The fourth part is called 'Training-Testing Separation', which is the process about how to choose training data and testing datasets to train models. In this paper, we have four different methods provided and their specific numbers are shown in figures and then their differences resulting in precision are illustrated in the experiment section. Finally, the last process is called 'Machine Learning Techniques', which is the key part and the most important step in the experiment. In this paper, there are four different machine learning techniques offered: KNN (K-Nearest Neighbors), decision tree, random forest and logistic regression, which are all using labeled datasets, so we also call them supervised learning techniques. Their comparison among different datasets is also performed and then the summary part is also based on the technique performance.

3.1 Image Dataset

There are 3 datasets contained in the project. The first dataset is called ‘Fruits 360 dataset’ downloaded from https://www.kaggle.com/moltean/fruits, which includes 4 folders: Test, Training, papers and test-multiple_fruits. In this paper, we only use the Training folder file, which has 81 class fruits included with a total of 41322 images included. For this dataset, the great thing is all the images are the same size: 100*100 pixels, convenient for our experiment. The second dataset we use is called ‘supermarket produce dataset’ downloaded from http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz. This dataset has 14 classes fruits included and one onion class. The paper is only aiming at analyzing fruits, so we do not use onion class dataset. A total of 2558 images are included. Each image is sized with 1024*768 pixels. The third dataset is called ‘FIDS30’, which includes 971 images among 30 classes downloaded from http://www.vicos.si/Downloads/FIDS30. Unlike the first two datasets, this dataset seems to be more real, collected from hundreds of places with different sizes and noise added, like trees, leaves, plates and other backgrounds. For this experiment, we only use 963 of them as we resize the images, 8 of them do not work, so we delete them. Also, some images include a single fruit while others contain dozens. One thing to notice is for all the experiments except the comparison of different datasets, we only use the first dataset to analyze as it has the largest number of images, and more data to use.

3.2 Image Preprocessing

BASIC) As the very first step, image preprocessing plays an important role in the whole data analyzing process. First, we read all these images in three channels: R, G, B with OpenCV package. That means the images are converted real numbers based on their colors. Specifically, each image is read by its pixels and each pixel can produce 3 numbers: R value, G value, B value, which range from 0 to 255. In this way, each image can produce 100*100*3=30000 numbers. Among the 30000 features, features 1-3 stand for the R, G, B values of the first pixel, which is located at the upper left corner and then the second feature 4-6 stand for the second pixel, which is the right pixel, next to the first one. Once the 100 pixels of the first row are finished, the first pixel of the second row of the image will be read until the last pixel at the bottom right corner one has been read. While reading images, we create an array to store their labels based on their different names. After loading all the images, we obtain 2 arrays: images array, whose size is 41322*30000 (100*100*3); labels array, which has 41322 values with 81 kind of classes of image name. After that, these 81 different names in labels array can be converted to real number: 0, 1, 2, 3, …, 80.

FURTHER) We calculate the color histograms for different color channels. Each channel is divided into 8 groups (256/8), each group has a range of 32. The first group includes 0-31, the second include 32-63 and so on. Continuously, each channel will be finalized into 8 different groups. By combining these 3 channels together, we will have 8*8*8=256 numbers, each number standing for different combinations of R, G, B color. After this, we have converted each image into 256 real numbers.

Hence, until now, we have two different methods to analyze the color of images. The FURTHER is built on the BASIC.

3.3 Dataset Analysis

Dataset understanding is an essential part in the data analysis field. To understand data, we need to know some important features of the dataset, which play essential roles for the final results. We use scatters, plots and histograms to show the data features.

a. Unbalanced, accurate and complete

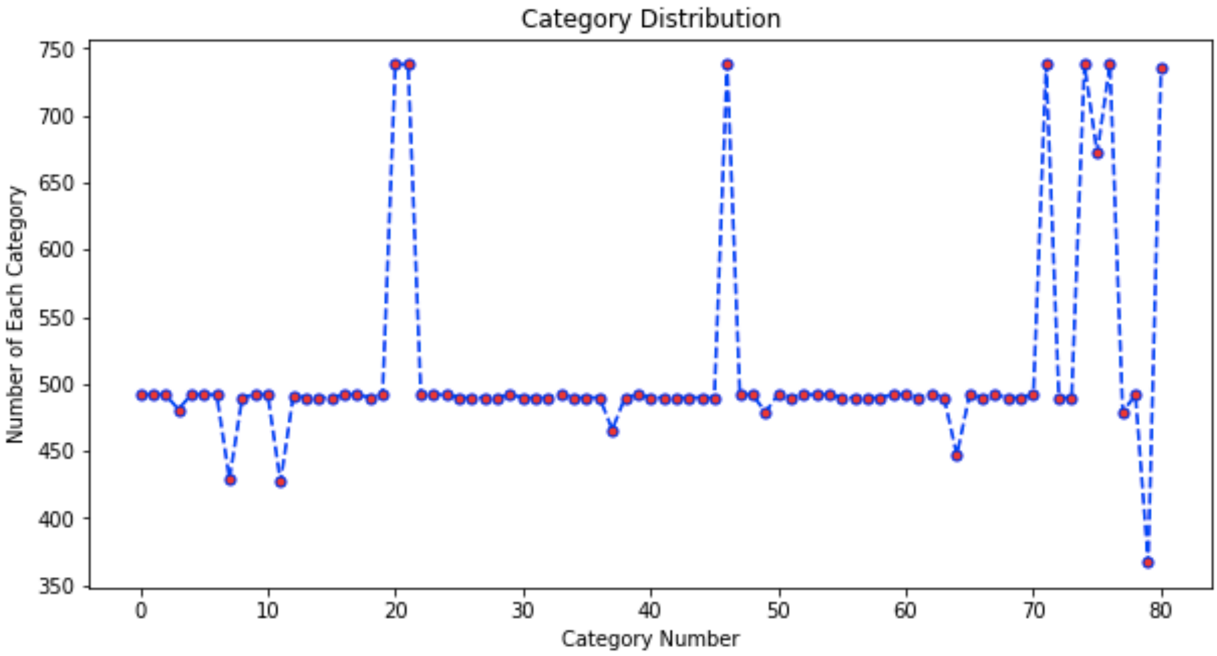

Figure 2: The Entire Dataset: Category Distribution

Figure 2 demonstrates a relationship between the data and observations in each class. It tells the category distribution of the dataset with the number of each class being shown. We can tell for most of the classes, their number is around 500, while some majority classes can reach almost 750, and some minority classes are around 350, with the ratio of minority to majority around 47%. Hence, from their different numbers, we know that the dataset is unbalanced, since balanced dataset has the same number of observations for each class. This leads to unbalanced data problem: some majority classes may be with higher prediction precision; some minority classes may be with lower prediction precision.

As for completeness, we can confidently tell the dataset is complete (no missing value) as we do counting the number of each feature among all the classes in the data matrix. However, the biggest problem is we cannot easily tell whether the dataset is accurate, as it is not separable well.

b. Big data VS Regular data

As for the scalability of the data set, we think about the controllers, which include three parts: features, observations and classes. Features determine size and dimension; observations determine size and volume; classes determine variety. When the rate of coming is high, observation also determines the velocity. As mentioned above, the number of features of the data set is 30000, the number of observations is 41322. In this dataset, each image stands for one observation. Also, it includes 81 classes. Thus, we can see this data set as a big data set considering the controllers. All these controllers are big, so there are enough reasons to convince us that this dataset is not a regular dataset, but a big dataset.

c. Separable VS Non-Separable

To tell if a dataset is separable or non-separable, we need to see its distribution among different classes. Are they overlapped or not? Among all these 30000 features, 81 classes, we cannot show all of them, but we choose some sets of features and classes to the best representation all of them. Some of them represent the color value from the background, some of them from the edges of the fruit, and others from the inside pixel of the edge. Here we have divided the whole image into two parts: body part and background part.

To represent all of the cases, the first set of features we choose: feature 1= 7575, feature 2=15000, feature 3=22725. In this case, feature 1 means the 2526th pixel blue value, which is located in the center of the upper left ¼ part of images; feature 2 means the 5001th pixel blue value, which is located in the center of the images; feature 3 means the 7576th pixel blue of the images, which is located at the center of the bottom right ¼ part of images. This set of three features can stand for 'Body part' for almost all the images of the dataset, which is what we need to classify images.

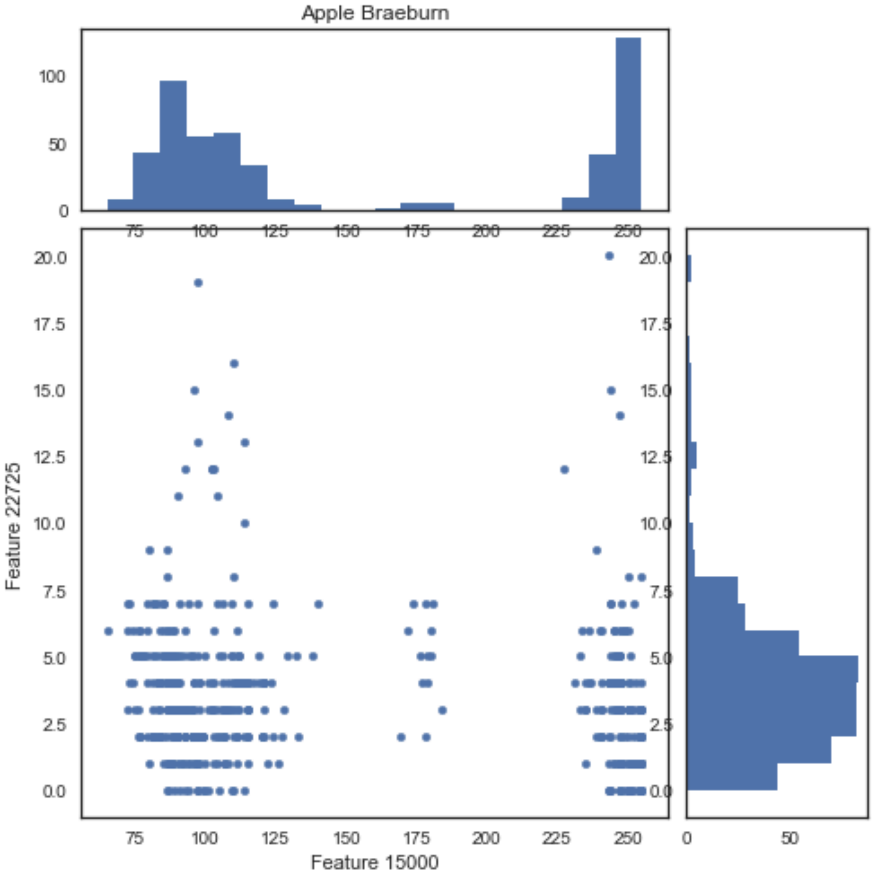

Figure 3: Apple Braeburn Distribution of Figure 15000 and Feature 22725

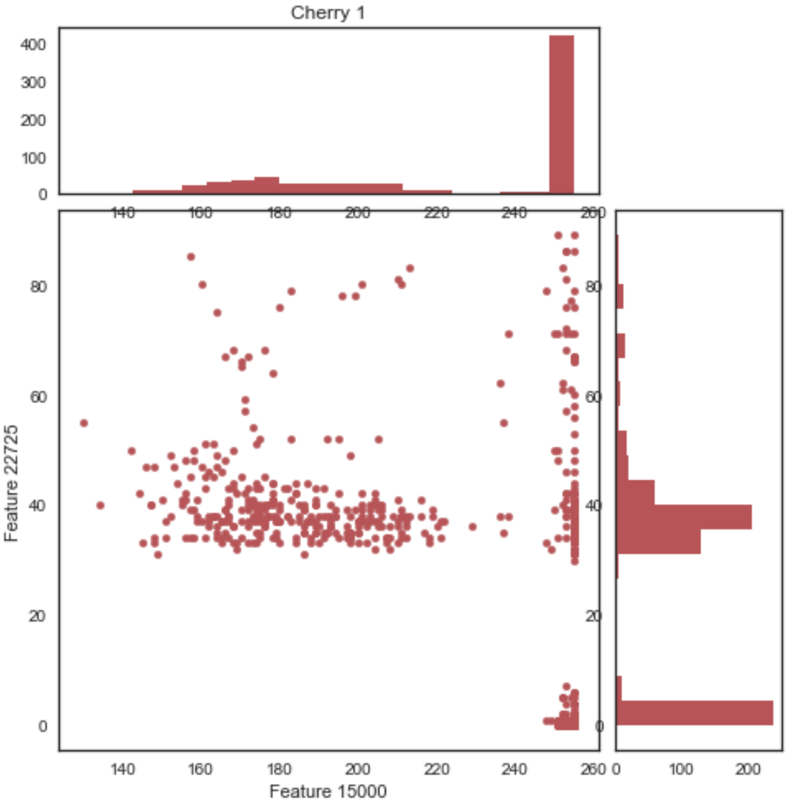

Figure 4: Cherry 1 Distribution of Figure 15000 and Feature 22725

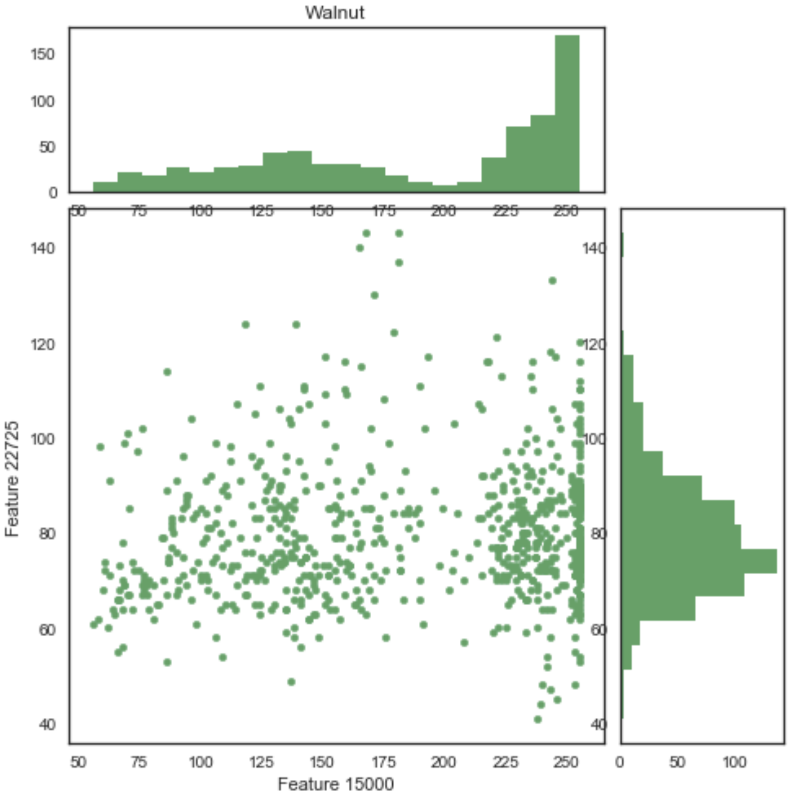

Figure 5: Walnut Distribution of Figure 15000 and Feature 22725

Figures 3-5 display the distribution of feature 15000 and feature 22725 among three different classes, where the classes are chosen randomly. From the histogram parts of these figures, we can tell the value from different classes are distributed differently. Let us take feature 22725, y-axis, to explain it. Feature 22725 from the Apple Braeburn class mainly range from 4-5, which means the blue value of 7576th pixel, the center of the bottom right ¼ part of images, are mostly at range 4-5, which is very low. From what we see from images, that pixel is with very little blue element from the color components. For class Cherry 1, the feature 22725 has two parts distribution, one is around 40, and the other part is about 0. This explains some of this class images are with 40 blue value for the pixel. The others of this class images are almost no blue element as for the 7576th pixel of images. For class walnut, we can see most of the feature 22725 value are distributed around 60-90, which is different from the other two classes.

To understand the GRB pixel value better, here we make an assumption: if all the R, G, B values of this pixel are about 0 (0,0,0), then we will see this pixel with 'BLACK' color, using a microscope or magnifying lens. Further, if a batch of pixels are with R, G, B values 0, then we can see a 'BLACK' part with our eyes possibly.

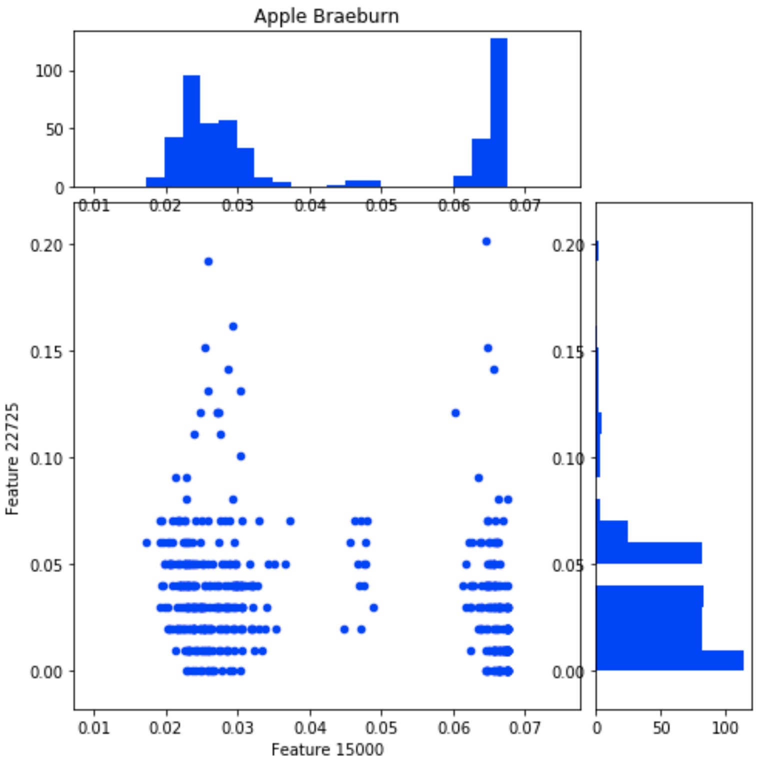

Figure 6: Normalized Apple Braeburn Distribution of Figure 15000 and Feature 22725

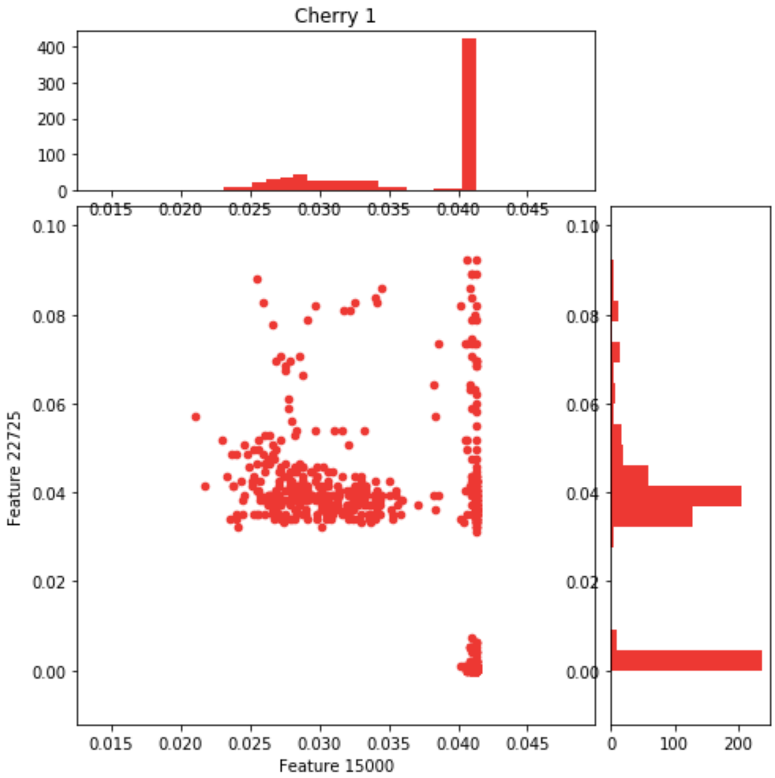

Figure 7: Normalized Cherry 1 Distribution of Figure 15000 and Feature 22725

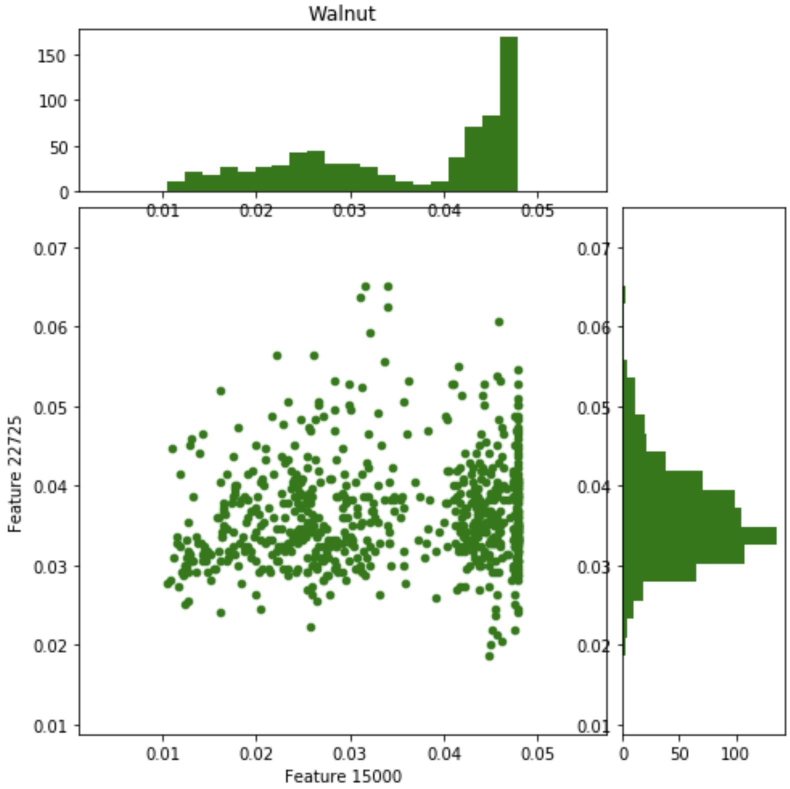

Figure 8: Normalized Walnut Distribution of Figure 15000 and Feature 22725

Figures 6-8 show the normalized value distribution of these three classes about feature 15000 and feature 22725. in the paper, we normalized all the RGB channels together and standardized all of them together in the following way. We can see all the normalized values are between 0 to 1. The normalized process makes data less sensitive to the scale of features, and that helps us find their coefficients. By comparing figure 5 and figure 8, we can tell the points distributed closer by normalization.

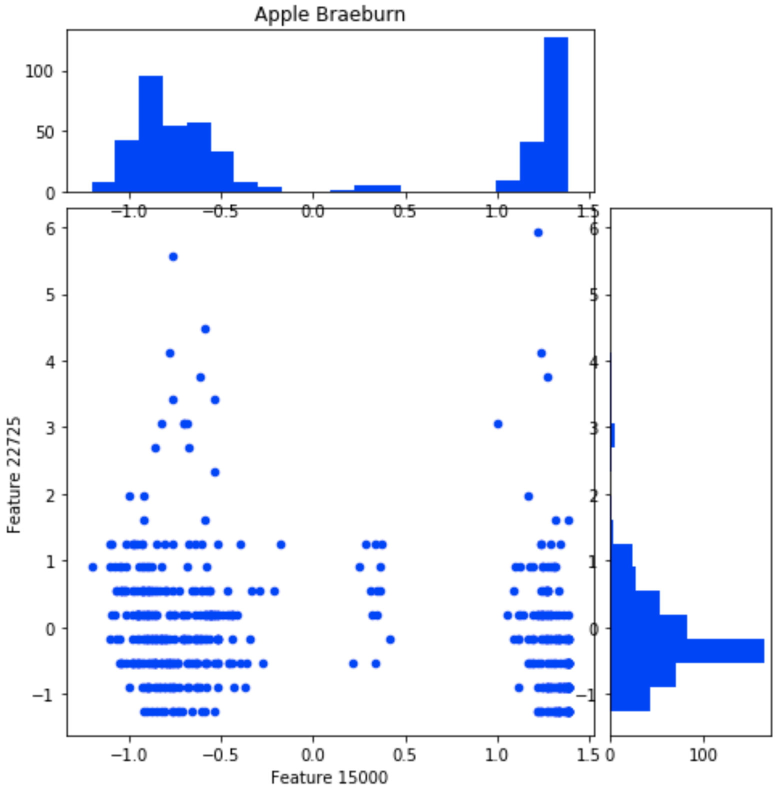

Figure 9: Standardized Apple Braeburn Distribution of Figure 15000 and Feature 22725

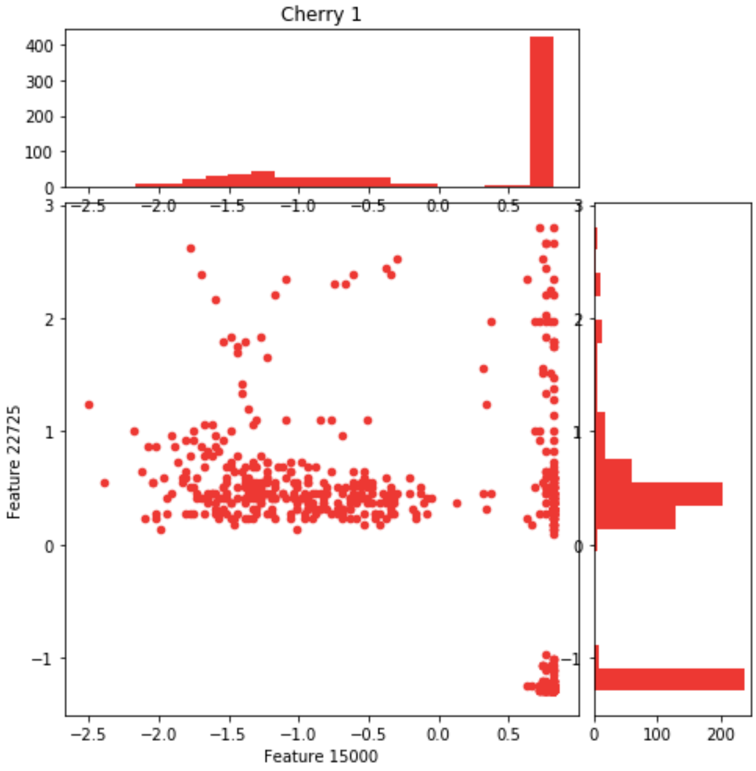

Figure 10: Standardized Cherry 1 Distribution of Figure 15000 and Feature 22725

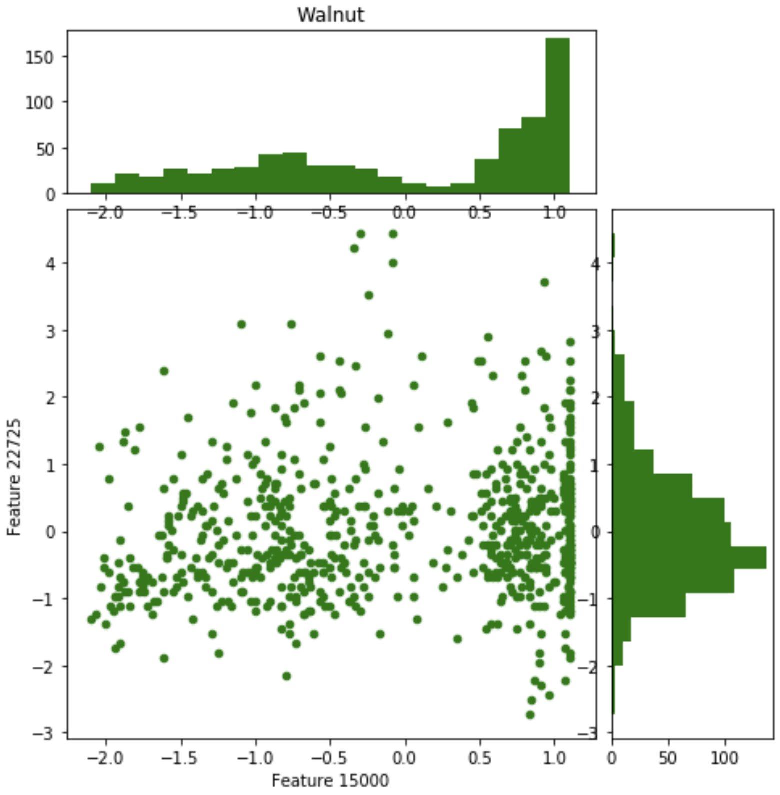

Figure 11: Standardized Walnut Distribution of Figure 15000 and Feature 22725

Figures 9-11 show the standardized distribution of feature 15000 and feature 22725. The standardized process helps us easily compare features that have different units or scales.

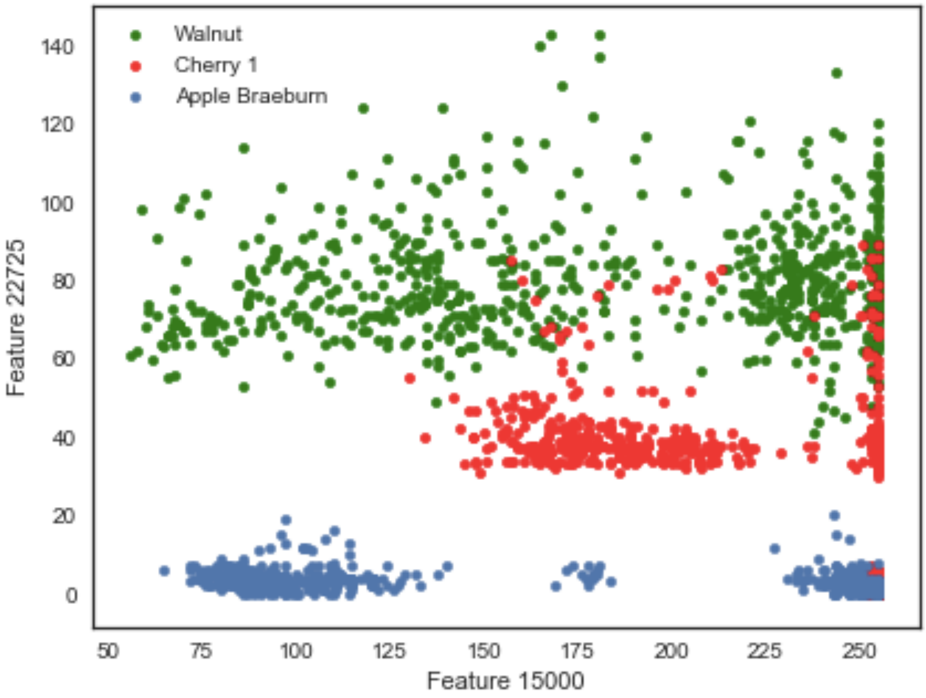

Figure 12: 2D-Sample Distribution of Figure 15000 and Feature 22725

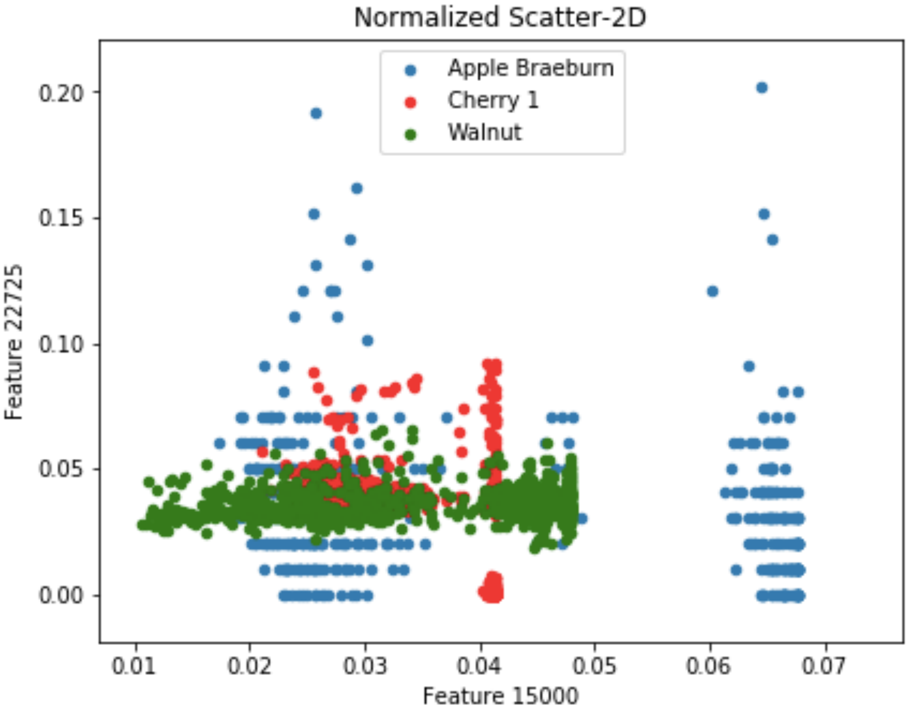

Figure 13: Normalized 2D-Sample Distribution of Figure 15000 and Feature 22725

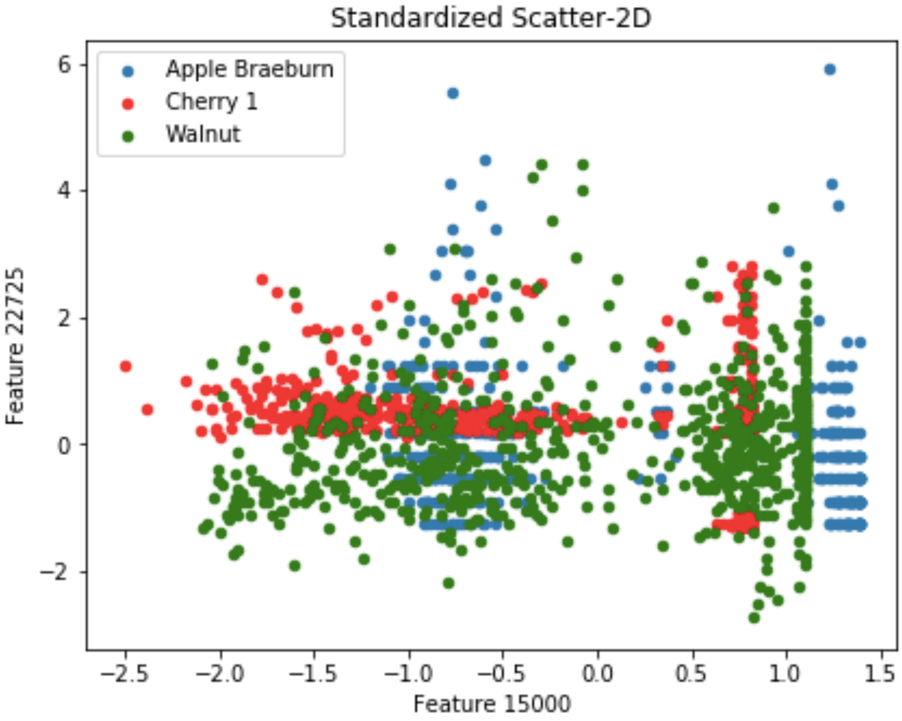

Figure 14: Standardized 2D-Sample Distribution of Figure 15000 and Feature 22725

Figures 12-14 show the original distribution, normalized distribution and standardized distribution of these two features among the three classes. From figure 12, we can tell these color points range differently as for y-axis, feature 22725. Roughly speaking, the green points are located in uptown; the red points are located in midtown; the blue points are located in downtown, like Manhattan Island in NYC. Hence, we can differentiate the three classes by feature 22725. That is to say the center of the bottom right ¼ part of images among these three classes is with different blue value, leading to different colors to see.

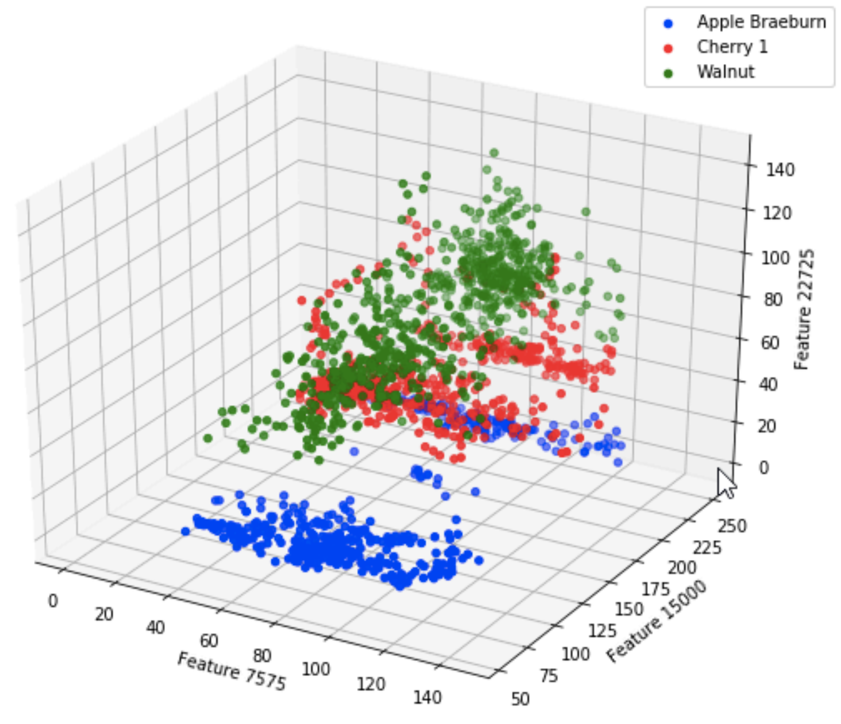

Figure 15: 3D-Sample Distribution of Figure 15000 and Feature 22725

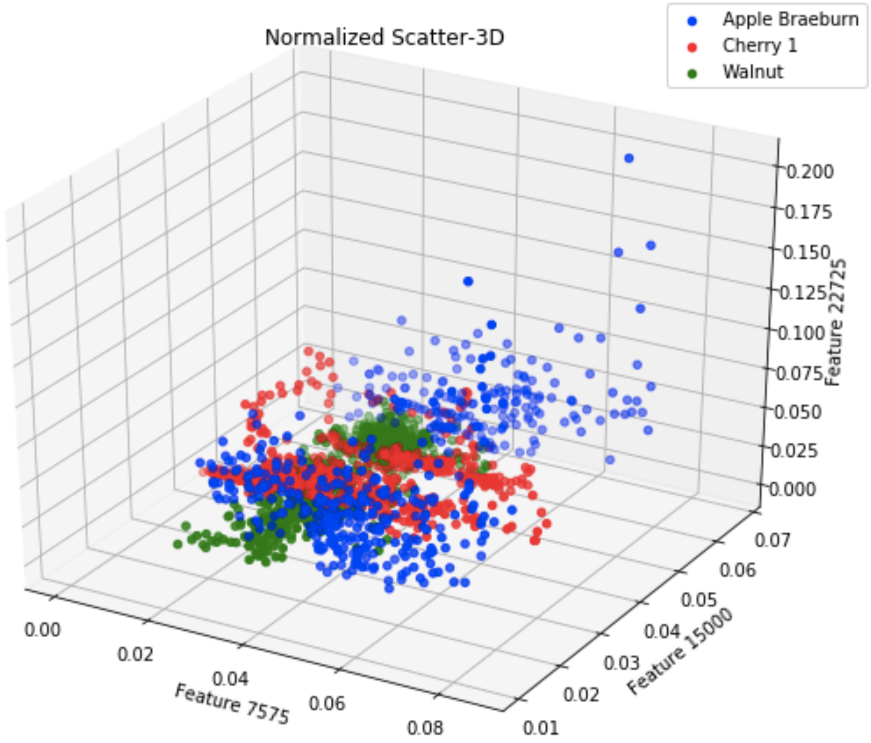

Figure 16: Normalized 3D-Sample Distribution of Figure 15000 and Feature 22725

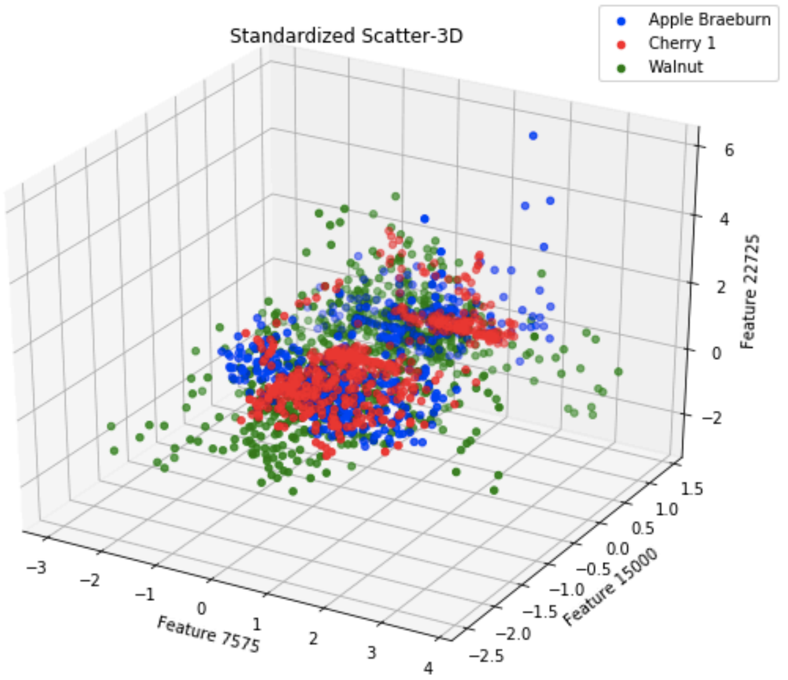

Figure 17: Standardized 3D-Sample Distribution of Figure 15000 and Feature 22725

Figures 15-17 show the original, normalized and standardized feature distribution of the three classes in 3-dimensional scatter plot individually. From figure 15, we can tell the dataset is not separable very well, but some part of them can be considered separable. Hence, we can consider this dataset as 'partially separable' for this set of features.

d. Scalability

Scalability of data refers to the capability of a system to handle a growing amount of data, which is quite significant. When we test the data on the regular laptop, the issue shows up, the system is broken down and stops working. Hence, there is enough reason to believe that it has a scalability problem.

e. High-Dimension

For high-dimensional problems, we can tell this data set is not high-dimensional, because the number of features (30000) is not higher than the number of observations (41322). However, to get results more efficiently, we need to choose the dominant features that classify them obviously as some features like, all of the background parts do not help at all in classifying different images. Hence, we make feature selection or feature extraction to reduce dimensions and reduce calculation. Thus, we will make a classification mostly based on the 'Body part'. Usually, there are two techniques to make dimension-reduction: feature hashing and principal component analysis(PCA). Feature hashing helps reduce the dimension of the data through hashing an infinite feature to a finite feature space. Similarly, PCA makes a dimension reduction by choosing correlated variables (features). The experiment section shows how we select the main features and how they perform as for classification accuracy.

3.4 Training-Test-Validation Dataset

The training and testing dataset separation plays an essential role for training machine learning models and making a prediction. In this section, we proposed four different methods of how to choose training datasets. Their distributions of numbers of each class are shown in figures 18-21.

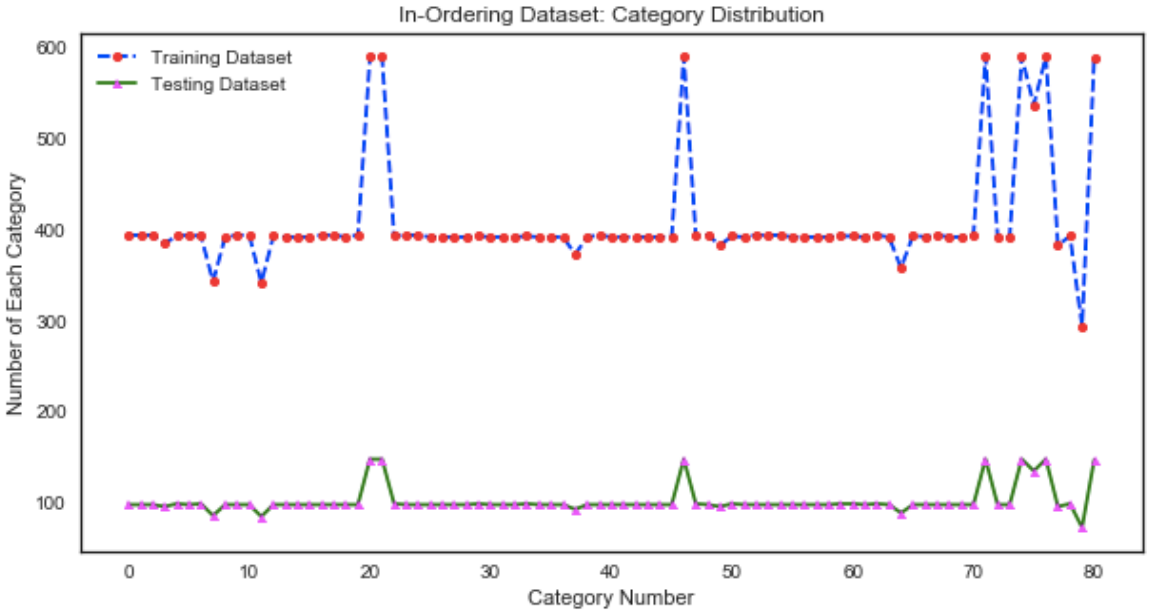

Figure 18: In-Ordering Dataset: Category Distribution-80:20

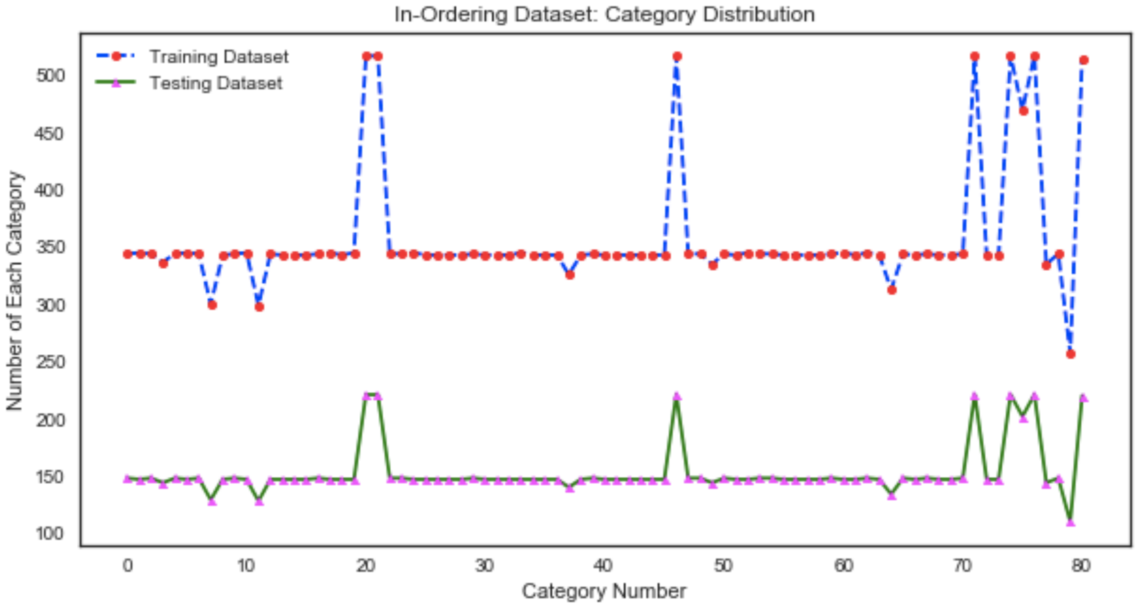

Figure 19: In-Ordering Dataset: Category Distribution-70:30

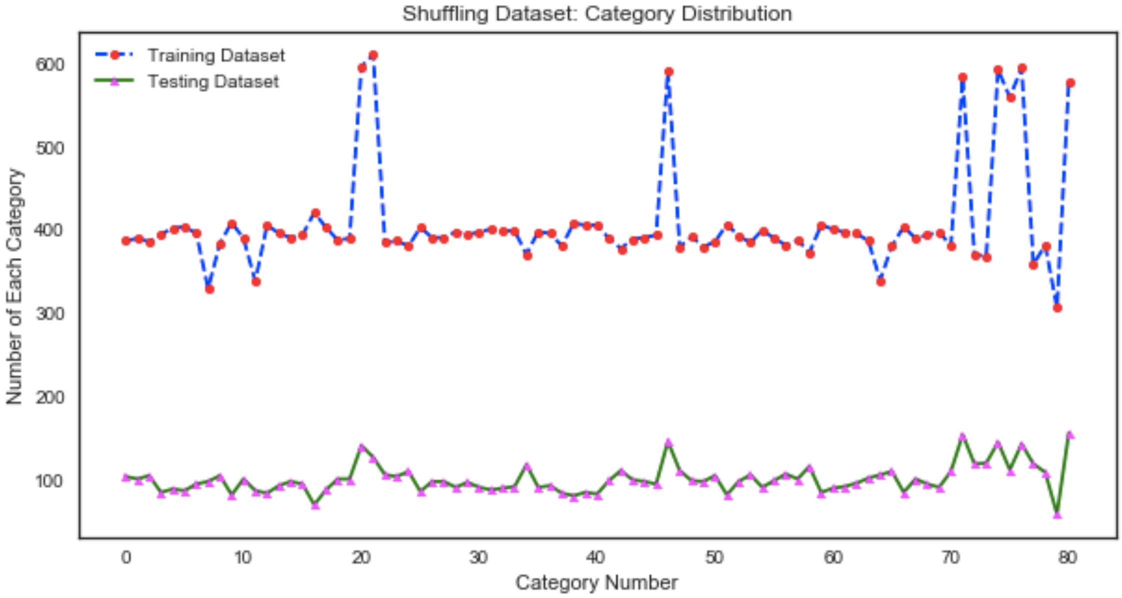

Figure 20: Shuffling Dataset: Category Distribution-80:20

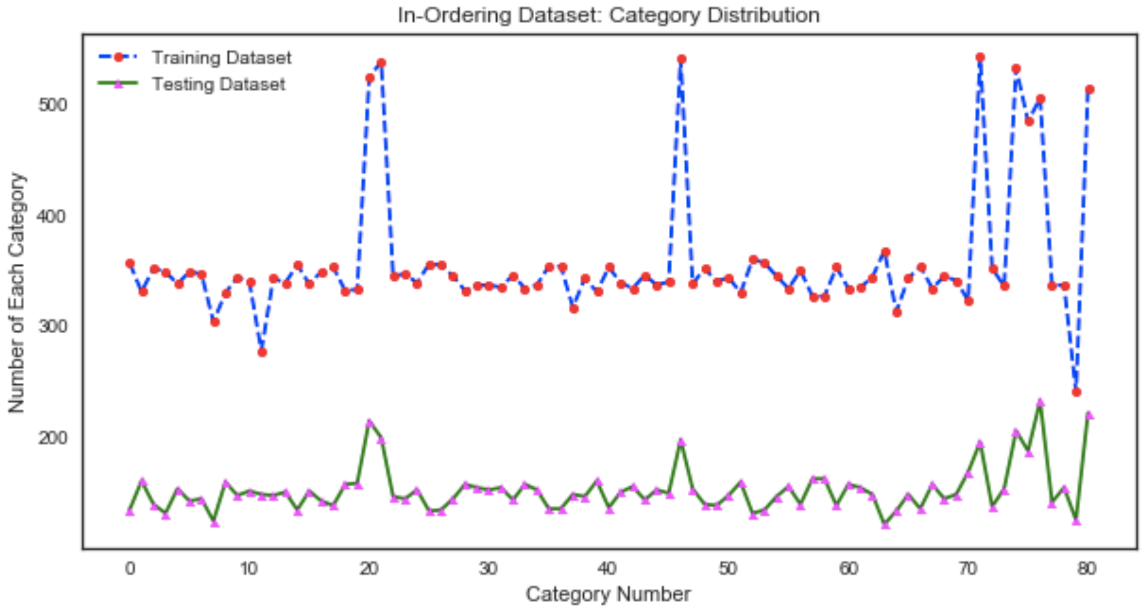

Figure 21: Shuffling Dataset: Category Distribution-70:30

Figure 18-figure 21 show the relationship between datasets and observations of every classes among the four ways. In each table, training datasets and testing datasets are deployed. Figure 18 shows each class number of separating the data with a 80:20 ratio in order, which means from 80 percent of each class images are chosen as training datasets and the rest 20 percent are seen as testing datasets. Figure 19 divides it by 70 percent and 30 percent as training and testing individually. Figure 20 shows each class number of separating the data with a 80:20 ratio in shuffling (out of order), which means 80 percent of total images are randomly chosen for training, and the rest 20 percent are used as testing. Noticeably, this separation does not guarantee the images from every class is 80 percent of that class, which explains the reason why each class number chosen is obviously different. Figure 21 shows identically as figure 20 except the ratio becomes 70:30.

3.5 Supervised Learning Techniques

a. KNN

KNN means the closest k classes are chosen when classifying. If the actual class is in the k classes, we can say it predicts accurately. Hence, with different K values, we can get different results. In order to choose the best k value for the KNN algorithm, we need to find out the best k value with the least error rate.

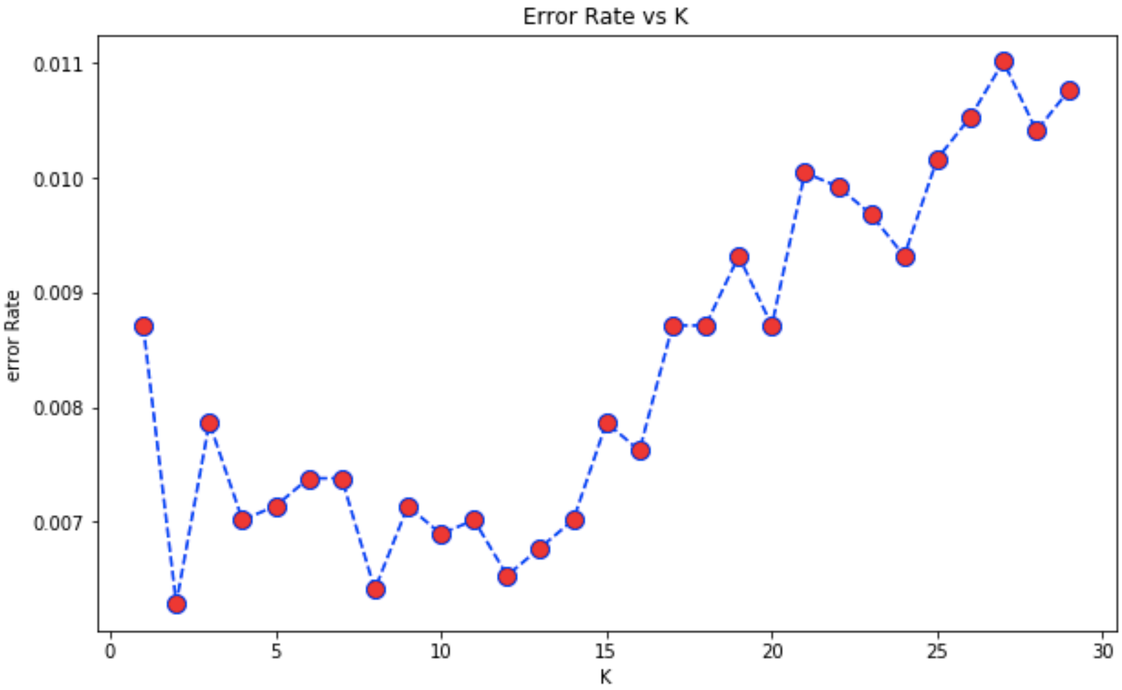

Figure 22: N Value VS Error Rate in KNN

Figure 22 shows the error rate for different K values in KNN technique. The total class number is 81. We only show the range of K from 1 to 30. From this figure, we can tell that error rate has the minimum value when K equals to 2. Based on this, to receive the best result of classification, we choose to set K value to 2 when using KNN model.

b. Decision Tree

There are two kinds of decision trees: Regression tree and Classification tree. If the response is continuous, then it is a regression tree. If the response is discrete, then it is a classification tree. In this paper, we only use classification tree. Decision tree makes a classification by cutting horizontally and vertically. The key of it is how to find the position to make a cut. The position to make a cut depends on the information gain, which is a quantitative measure. The split position and a feature can be seen as decision classifier.

In order to find the best position to split, entropy and information gain are introduced. While choosing a split position, each split information gain is calculated and the position with the maximal information gain will be chosen as the split position. If the data has multi features, it will calculate the best split position, find the value feature by feature until the best feature and split position are found, and the depth of tree are decided by cross-validation.

c. Random Forest

Random Forest uses decision tree model for parametrization, but a sampling technique, called the bootstrap is contained to optimize the model. A part of data domain is chosen as real domain data to build decision tree and then in this way, finally we have many decision trees built. Hence, compared with decision tree model, random forest has one more thing to think about, which subspace to choose, besides choosing features and choosing the best split position. Bootstrapping is used to help build multi decision tree at the train phase by randomly choosing subsets of domain data with replacement. Similarly, bagging plays an important role at test phase, which suggests the final result by choosing the majority of the class label at each decision tree. Random forest can improve classification accuracy by using multi decision trees at test phase.

d. Logistic Regression

Logistic Regression is built on probability, which utilizes a logistic function to estimate the parameters of a logistic model. The final result is the probability, and so it is a value between 0 and 1. After that, we set a threshold, usually 0.5. When the probability is larger than 0.5, then we can assume it will happen, labeling it with 1. When the probability is less than 0.5, we can assume it will not happen, labeling it with 0. LR is very popular in two-class classification problems.

4. Result

Before drawing conclusions, we should review the whole analysis process. Based on the analysis process, we can think about the following questions from dataset choosing perspective, data preprocessing perspective, data analysis perspective, machine learning model training perspective individually.

1. Which dataset to choose? As we have three totally different datasets in this paper.
2. Keep data matrix in order or shuffle?
3. Use data dimension-reduction or not?
4. Use normalization, standardization or not or both?
5. What is the partition of training dataset and testing dataset?
6. Which class to analyze? As we have multiple classes in the dataset.
7. Which machine learning technique to use?
8. What are the training dataset accuracy and testing dataset accuracy?

Question 1 is from dataset choosing perspective.

Question 2-4 are from data processing perspective.

Question 5-8 are from machine learning classification perspective.

These 8 questions cover every step of machine learning analysis process.

In this chapter, we will show the relationships between the classification accuracy and each of these perspectives among these four machine learning techniques.

4.1 In-Ordering & Shuffling and Partition

Figures 23-26 show the number of each class distribution based on different ordering and partitions, which are related to Question 2 and Question 5. To find out the result, we make a classification with decision tree technique among these 4 different options. One system problem here is when we run these 41322*30000 matrix, the system is broken down. Hence, we choose a part of them proportionally to see the classification accuracy in the following experiment.

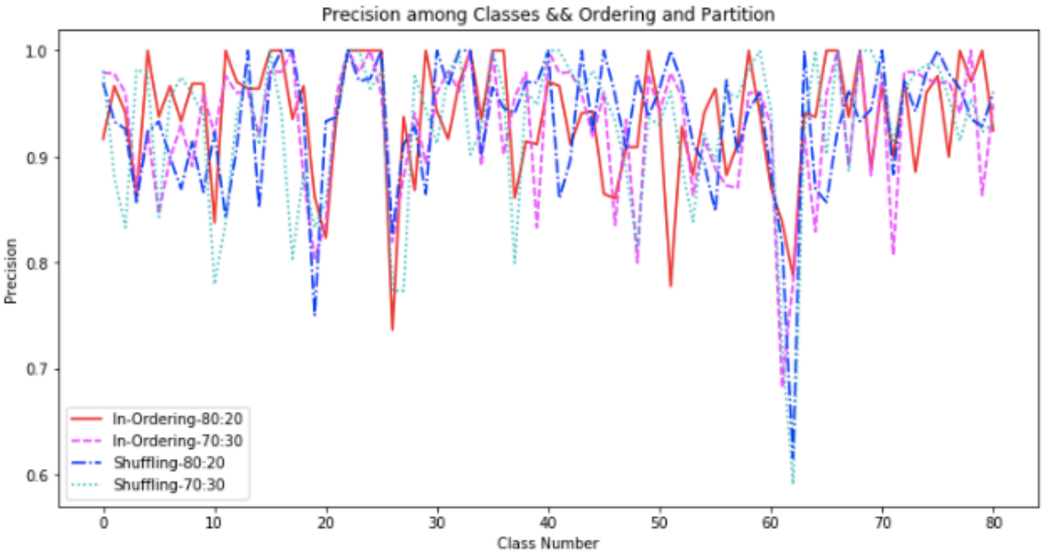

Figure 23: Precision among Different Order and Partition (1)

Figure 23 shows the classification precision of 81 classes of four different partition methods with decision tree technique, as mentioned above. From this figure, we can tell for most classes, the techniques behave similarly in general. To look into it, we have the next figure to show their performances individually.

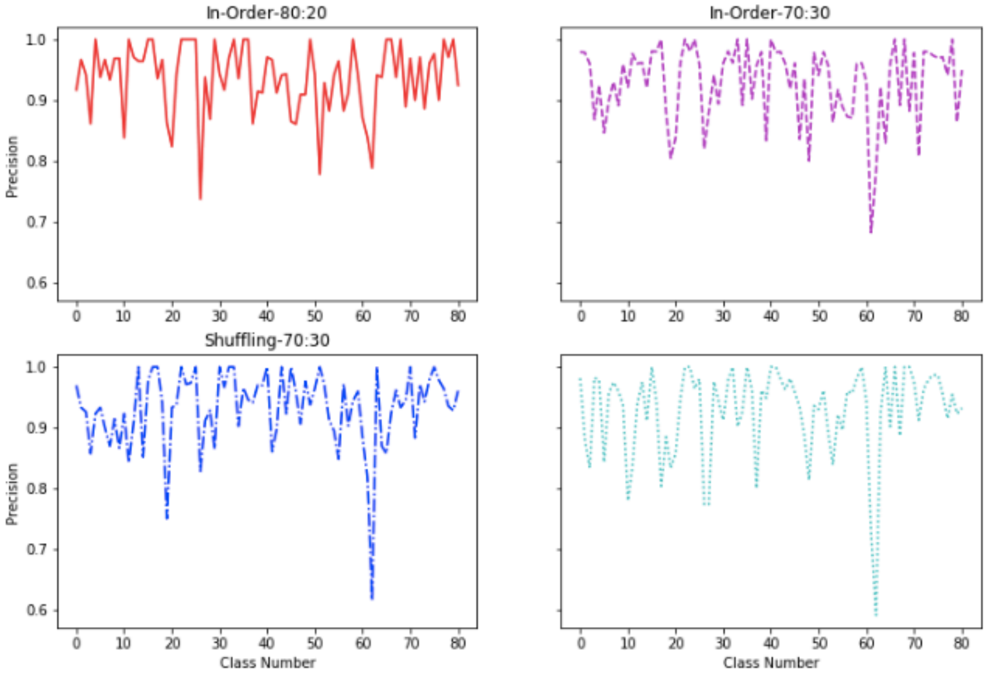

Figure 24: Precision among Different Order and Partition (2)

Figure 24 shows their performance clearly. The upper left one shows the result of separating it with a ratio 80:20 in order; The up right one shows the result of 70:30 in order; The bottom left one shows the result of shuffling 80:20; The bottom right one shows the result of shuffling 70:30.

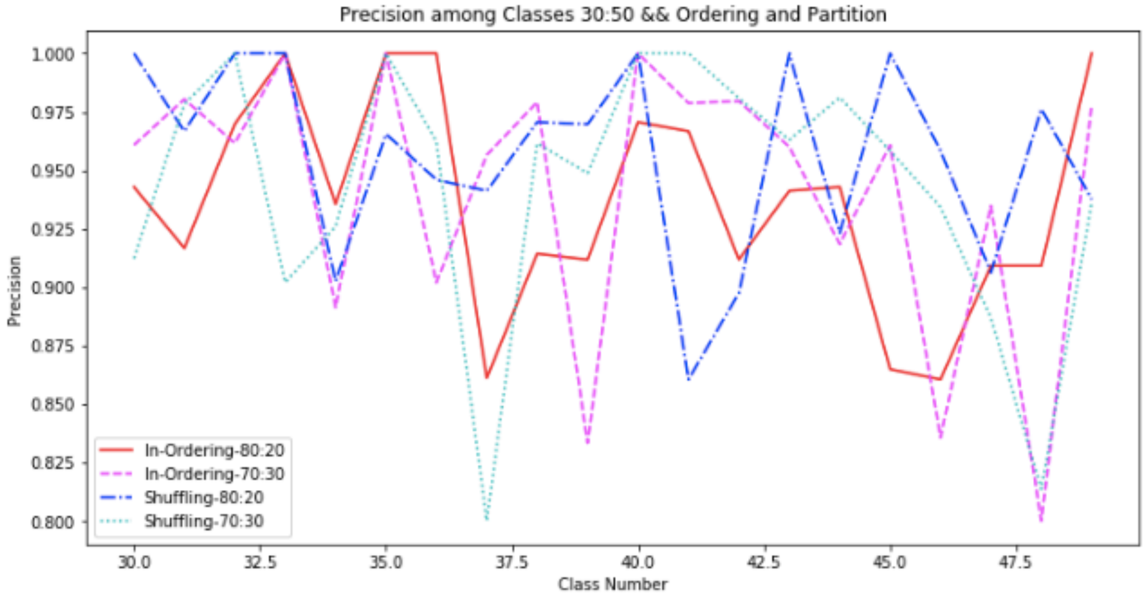

Figure 25: Precision among Different Order and Partition (3)

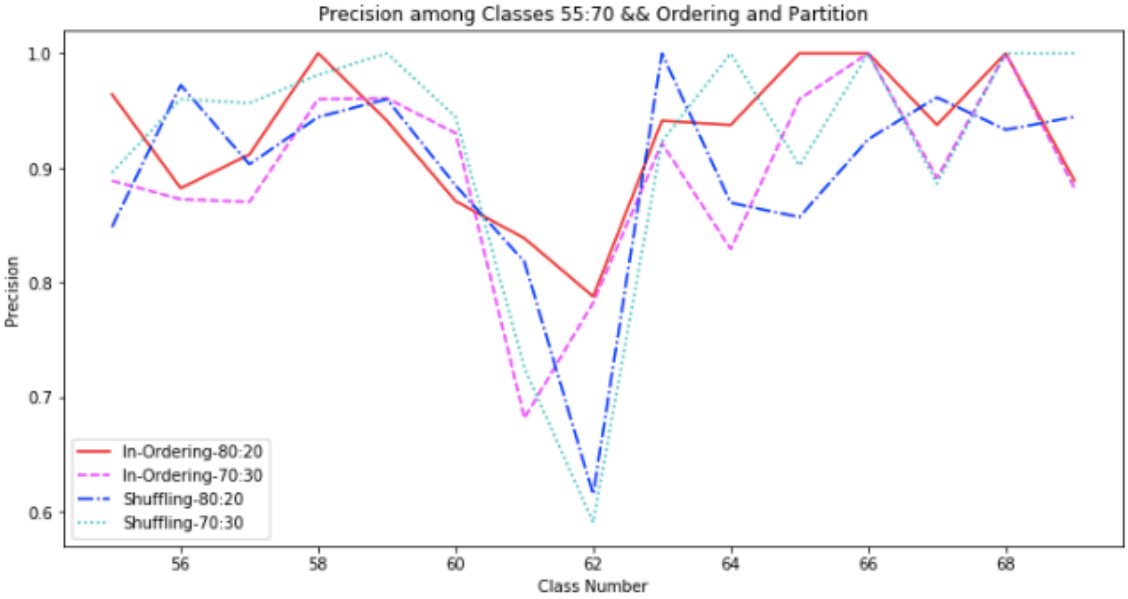

Figure 26: Precision among Different Order and Partition (4)

Figure 25 shows the performance of these four techniques among class 30 to class 50. In this figure, we can tell the red line (In-Order 80:20) reach the best result as it stays higher than the other three for majority of the class 30 to class 50. To make sure our result is more reasonable, let us see their behaviors in other ranges of classes.

Figure 26 shows the result of class 55 to class 75, where these four techniques are displayed. Obviously, we can notice that the red line stays at the highest position almost all the time. Especially for class 62, we can see that all of these four methods cannot get quite high precision, but the red line can still keep the best performance. Thus, there is enough reason to believe that the in-order 80:20 can get the best result. Based on this result, we use this separation method in the following experiment.

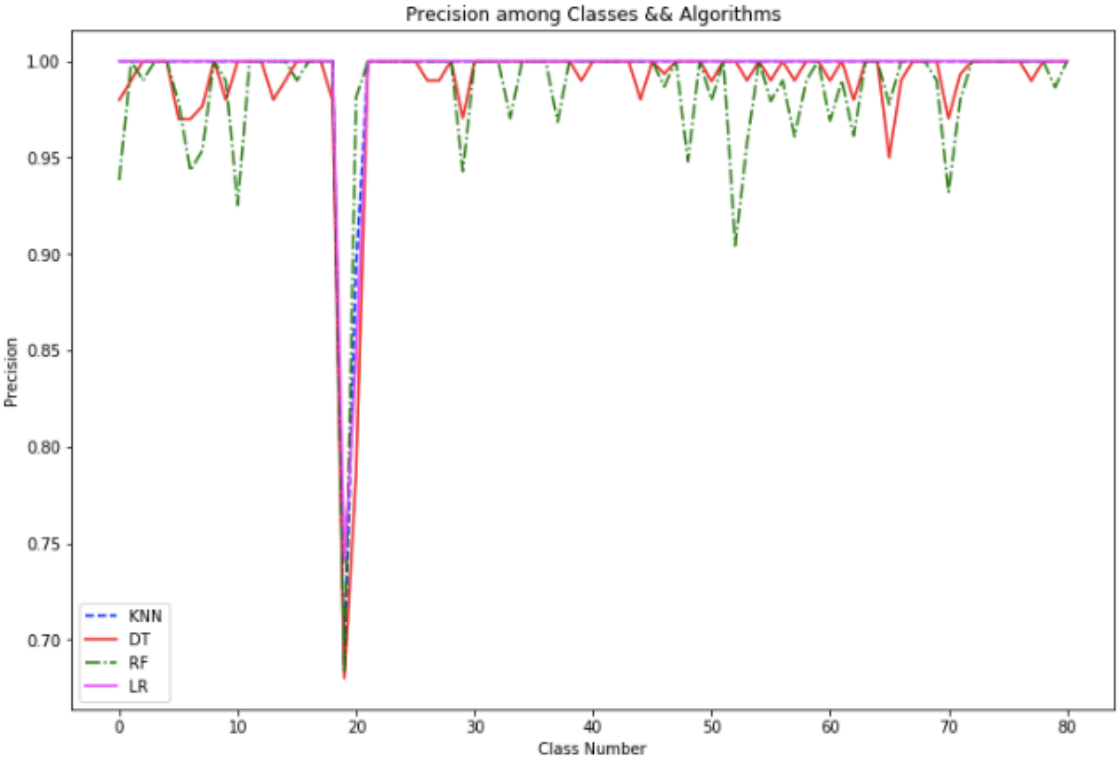

Figure 27: Comparison of classification accuracy among the four techniques (Related to Question 6)

4.2 Machine Learning Techniques

Figure 27 illustrates the classification precision among these 81 classes fruits with the four machine learning techniques. It shows how accurate the machine model makes predictions. From this figure, we can tell that all of these methods do not behave well for class 19 and class 20. That is because both of the classes are cherries, with same style, shape, or even similar color, but just different degree of the color, one if bright red, the other is dark red. This is the main reason to make the precision lower. Another thing we can also see from figure 9 is that the pink line (Logistic Regression) behaves constantly in classification accuracy, that is because this technique makes a classification based on probability. For instance, if the technique predicts the image as a 'watermelon' with 85% probability, then it will result in 'YES' for the final decision because it only results in 'YES' or 'NO'. For KNN (blue line), we can see it also behaves constantly, because is make a classification based on the closest class it belongs to, with 'YES' or 'NO' result, similar to Logistic Regression.

However, if we use this case with decision tree algorithm (red line), this technique will make a decision based on the built tree. This process was discussed above. If no further split is needed, this does not mean this subdomain is only with one class. Probably there is some data with other labels in it. Hence, no one decision tree can be a 'perfect' tree, that is why some errors exist. Then the random forest is a combination of multi decision trees, so it behaves similarly.

4.3 Original, Normalized and Standardized Dataset

To understand Question 4, we have three different ways dealing with the dataset and the final comparing result is shown as below.

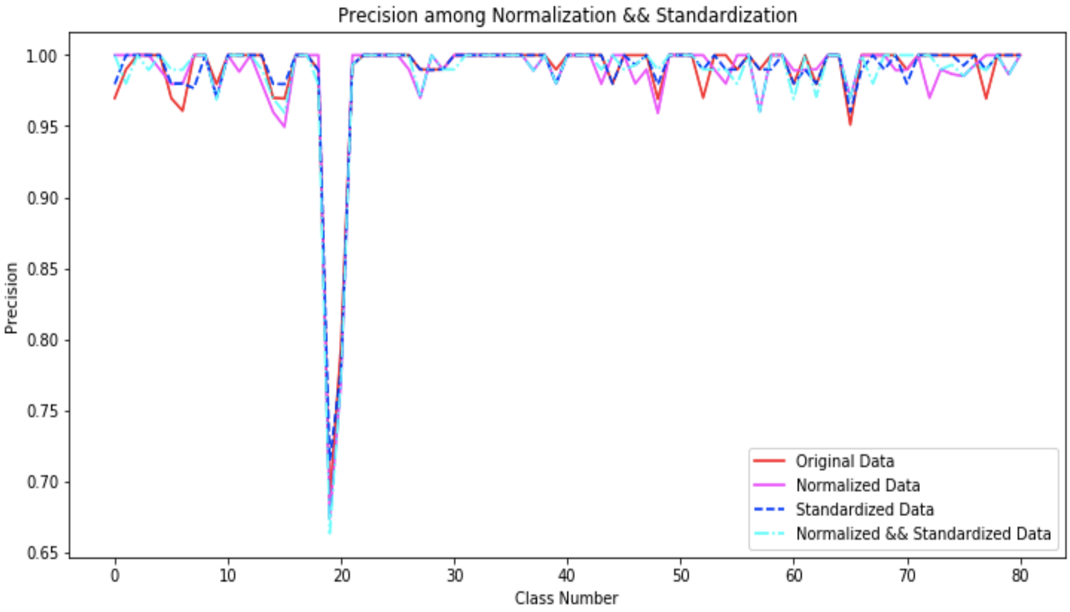

Figure 28: Comparison of classification accuracy among the four techniques (Related to Question 4)

Figure 28 shows the effect of normalization and standardization of the dataset as for the prediction precision. In this figure, four different methods of change of data have been deployed: Original data, normalized data, standardized data, normalized and standardized data. For their performance, we can tell all of these four-colored lines drop dramatically in around class 20, which means all of them result in low accuracy as for the class. However, for other parts of the lines, we know they also behave very similarly, but generally the blue dotted line stays higher than the other three lines, which means the standardized data can produce higher accuracy.

4.4 Dataset & Algorithm

To understand Question 1, three different datasets are given in this paper. The dataset 2 includes 15 classes of fruits, the total number is 2633. All of the images are with the same sizes. Dataset 3 contains 9 classes, the total number 273, but they are with different sizes(pixels). Hence, we resize all of the images into 32*32 pixels before reading them. With these four machine learning techniques, the final results are shown below in table 1.

Table 1: Precision among Different Techniques & Datasets

Precision	KNN	Decision Tree	Random Forest	Logistic Regression
Dataset 1	0.994	0.987	0.985	0.994
Dataset 2	0.921	0.916	0.994	0.979
Dataset 3	0.368	0.335	0.556	0.398

Table 1 shows the classification precision among the three datasets with four different techniques. For dataset 1, we can tell these four techniques perform quite well, where decision tree and random forest have a slightly low accuracy. For dataset 2, we can tell that random forest performs the best, or even significantly higher than KNN and decision tree as for accuracy. Continuously, logistic regression performs a little lower than random forest. For dataset 3, random forest receives a much higher classification accuracy than the other three and logistic regression behaves better than KNN and decision tree. Hence, there is enough reason to convince that random forest performs much better than others, and logistic regression behaves the second best, but it takes a longer time than others. In conclusion, we believe random forest performs well for the three datasets.

4.5 Dimension-Reduction (PCA)

To understand Question 3, PCA (Principal Component Analysis) is provided to reduce dimensions. PCA is a dimension reduction method to make datasets easier and speed up the analysis process significantly by extracting the major features. The figure 29 shows the results of PCA influence.

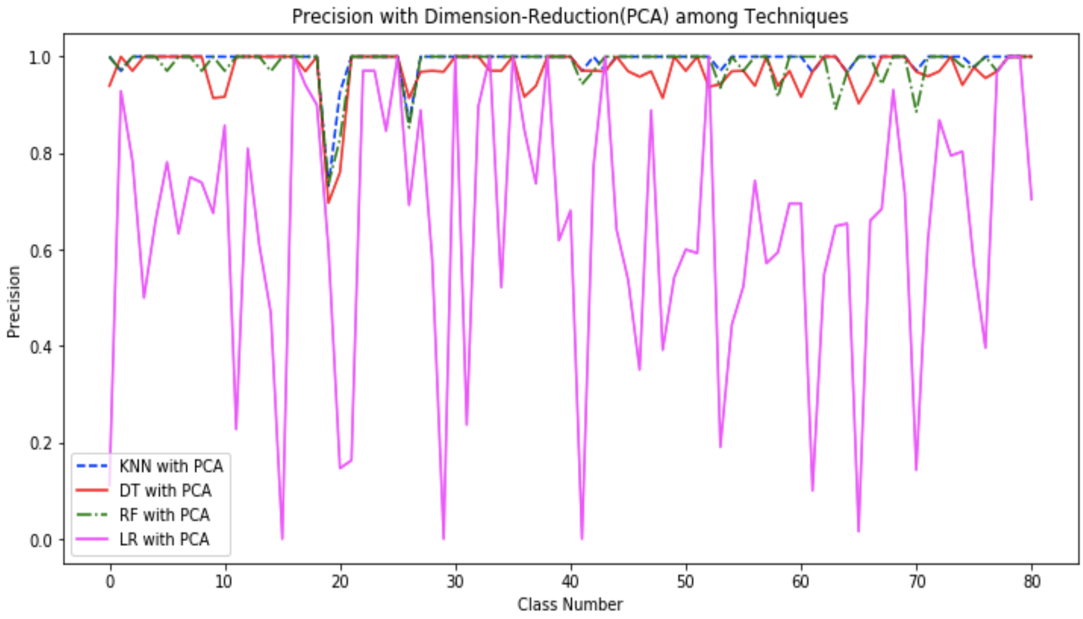

Figure 29: Precision among Different Techniques with PCA and without PCA (N=5)

Compared with figure 27, we can tell that decision tree and logistic regression behave worse than before. For KNN and random forest, it seems there is no great difference in precision.

4.6 Training Dataset VS Testing Dataset Accuracy (Confusion Matrix)

To understand Question 8, we have to look into the validation process, where the performance of classification models is studied by using a confusion matrix:

Table 2: Confusion Matrix

	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

The measure in training phase is called quantitative measure; the measure in testing phase is called qualitative measure. In the handout approach, the errors are calculated in the training, and the model with minimum error will be selected as the best model, and tested to obtain its accuracy. In order to get the best model, we use MSE, RMSE and entropy to do quantitative measure.

We measure training model by irregularity-based way: true positive, false positive, true negative, false negative.

Also, some related qualitative measures are as follows:

Accuracy=(TP+TN)/(TP+TN+FP+FN)

Sensitivity=(TP)/(TP+FN)

Precision=(TP)/(TP+FP)

Specificity=(TN)/(TN+FP)

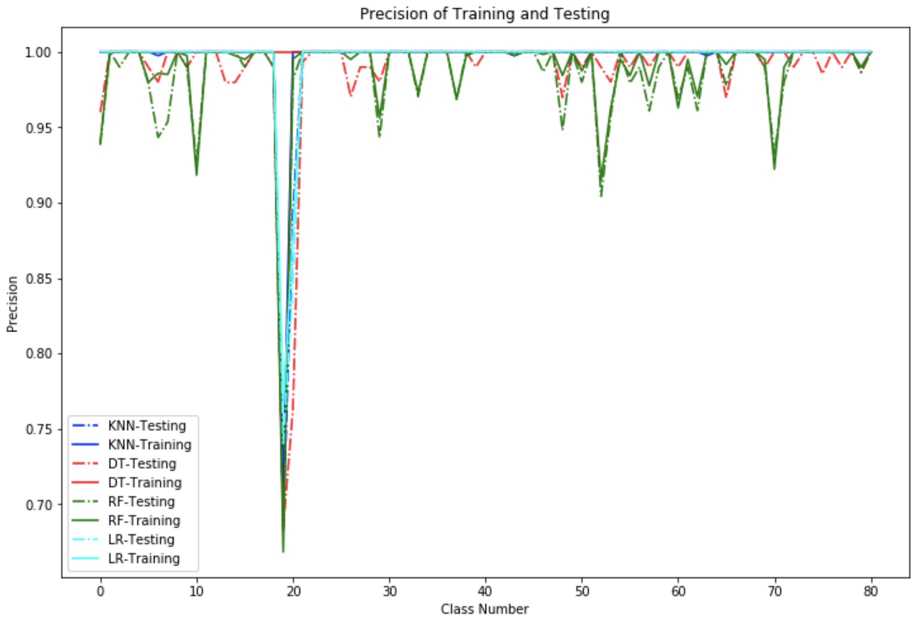

Figure 30: Precision of Training Dataset and Testing Dataset among Different Techniques

Figure 30 shows the precision of both training and testing phase among different techniques. The solid red line stays at the highest line, which means the trained decision tree model in the training phase behaves perfectly for the training dataset and all the data are classified correctly. Another thing we can notice is that the trained decision tree model does not behave perfectly for the testing phase (the red dotted line) as it does not stay at the highest position. KNN and logistic regression behave the same for both training dataset and testing dataset. Random forest technique behaves the worst among these four, but one thing to notice is the trained random forest behaves a little better in testing phase than training phase.

4.7 Additional Note

All of the results are based on the sklearn metrics. In the matrix, the precision=tp/(tp+fp), which is intuitively the ability of the classifier not to label as positive a sample that is negative; Recall=tp/(tp+fn), which is intuitively the ability of the classifier to find all the positive samples; F1 score means the weighted harmonic mean of the precision and recall; recall and precision are equally important by default; The support is the number of occurrences of each class in y_true.

5. Conclusion

For the final result, we break down the analysis process.

1. Dataset choice
2. Standardization or Normalization
3. Ordering of the data (shuffling or not)
4. Dimension reduction (PCA)
5. Partition of training and testing dataset
6. How well is the model trained?
7. Machine learning techniques

Based on what we analyzed in the experiment, to reach the best result, we need as much data as possible as, and then before doing any other processes of the data, we can standardize them. For training and testing partition, we recommend divide them 80:20 (in order), which means the 80% of each class of entire data are chosen as training data. For choice of machine learning techniques, we recommend random forest with PCA or logistic regression without PCA, but one thing to notice about logistic regression is that is takes longer time when running real big datasets. Also, we recommend KNN, with PCA or not, because it behaves excellently in either case. However, for decision tree, we can get the perfect trained model in training phase, but when testing it with testing data, it does behave as we expect.

Hence, the best combination is to standardize the dataset before dividing the dataset with a ratio of 80:20 in order with big enough dataset trained by RF technique.

REFERENCES

[1] Shan Suthaharan, 2016, Machine Learning Models and Algorithms for Big Data Classification, Integrated Series in Information Systems, Volume 36

[2] Rocha, A., Hauagge, D. C., Wainer, J., Goldenstein S., 2010, Automatic fruit and vegetable classification from images, Computer and Electronics in Agriculture 70(2010)96-104

[3] S.Arivazhagan, R.Newlin Shebiah, S.Selva Nidhyanandhan, L.Ganesan, 2010, Fruit Recognition using Color and Texture Features, Journal of Emerging Trends in Computing and Information Sciences, VOL. 1, NO. 2

[4] O. Chapelle ; P. Haffner ; V.N. Vapnik, Support vector machines for histogram-based image classification, IEEE Transactions on Neural Networks (Volume: 10 , Issue: 5 , Sep 1999),Page: 1055-1064