Diagram

Description automatically generated

Course: CS634 - 002

Professor: Jason Wang

Student: Shanmuga Sankar Balamurugan

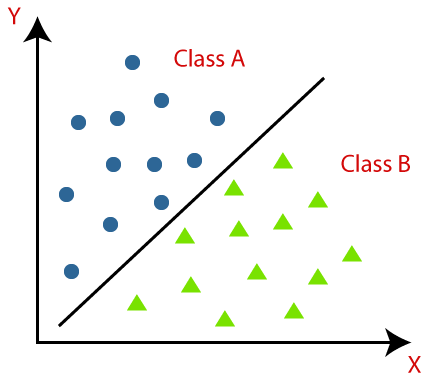
UCID: sb387

Email: [sb387@njit.edu](mailto:sb387@njit.edu)

Subject: Final Term Project Report

Project Option Number: Option 1 (Supervised Data Mining)

**SUPERVISED DATA MINING**



Contents

[1. Introduction 3](#_Toc101023297)

[2. Definition 3](#_Toc101023298)

[2.1. Linear SVM 3](#_Toc101023299)

[2.2. Random Forest 3](#_Toc101023300)

[2.3. Dataset 3](#_Toc101023301)

[3. Project Source Code 4](#_Toc101023302)

[3.1. Import Libraries 4](#_Toc101023303)

[3.2. Read and Display the Dataset 4](#_Toc101023304)

[3.3. Exploratory Data Analysis 4](#_Toc101023305)

[3.4. Pre-processing and PCA 6](#_Toc101023306)

[3.5. SVM Linear Classification (Category 1) 7](#_Toc101023307)

[3.6. Random Forest Classification (Category 2) 8](#_Toc101023308)

[4. Conclusion 9](#_Toc101023309)

# Introduction

The goal of the project is to perform supervised data mining (classification) algorithms on a dataset containing a binary target variable. For this project the two chosen techniques are linear SVM and Random Forest Classification.

# Definition

## Linear SVM

Support Vector Machine Linear Classification is when two classes are separated by a line. This line has two support vectors on both sides of the dividing line which are referred to as margin. Larger the margin greater the allowance for the classifier to predict correctly even when points break the area marked between the support vector and classification line.

## Random Forest

Random Forest Classification is when a group of decision trees are used together, and the mean output is used to determine the final output. The main characteristic as to how a random forest tree is grown is that attributes are chosen at random from around 60% of the data. The randomness of the algorithm makes it so that the results are not biased upon the data.

## Dataset

The dataset for the project is a raisin dataset, where we multiple feature which all describe raisin. The aim is to use these features and identify the target variable consisting of two classes Kecimen and Besni.

# Project Source Code

The code written for the executing Classification Models Linear SVM and Random Forest.

## Import Libraries

|  |
| --- |
| **import** pandas **as** pd  **import** numpy **as** np  **import** seaborn **as** sns  **import** matplotlib**.**pyplot **as** plt  **import** sklearn  **from** sklearn**.**svm **import** SVC  **from** sklearn**.**decomposition **import** PCA  **from** sklearn**.**ensemble **import** RandomForestClassifier  **from** sklearn**.**model\_selection **import** train\_test\_split  **from** sklearn**.**metrics **import** f1\_score  **from** sklearn**.**metrics **import** recall\_score  **from** sklearn**.**metrics **import** confusion\_matrix  **from** sklearn**.**metrics **import** accuracy\_score  **from** sklearn**.**metrics **import** precision\_score |

## Read and Display the Dataset

|  |
| --- |
| rasin\_dataset **=** pd**.**read\_csv**(**"C:\\Users\\Shank\\Desktop\\NJIT\\CourseMaterial\\Spring2022\\DataMining\\FinalTermProject\\Raisin\_Dataset.csv"**)**  rasin\_dataset |

Table

Description automatically generated

## Exploratory Data Analysis

Pair plots to show bi-variate relationship between all features w.r.t the class in the dataset

|  |
| --- |
| sns**.**pairplot**(**rasin\_dataset**,** hue**=**'Class'**,** height**=**2.5**);** |

Background pattern

Description automatically generated

Bar Graph to show the Class Distribution of the dataset; if class imbalance is found sampling techniques may be required before modelling.

|  |
| --- |
| rasin\_dataset**[**'Class'**].**value\_counts**().**plot**(**kind **=** 'bar'**,** color **=** **[**'orange'**,**'skyblue'**])**  plt**.**title**(**'Class Distribution'**)**  plt**.**xlabel**(**'Class'**)**  plt**.**ylabel**(**'Frequency'**)**  plt**.**ylim**([**0**,** 700**])**  plt**.**show**()** |

Chart, bar chart

Description automatically generated

## Pre-processing and PCA

Replace the target variables to ‘1’ and ‘0’, followed by separating features from target to be treated as input to the models. We calculate the eigen vectors to understand the number components are required through PCA to accurately predict the result.

|  |
| --- |
| rasin\_dataset**[**'Class'**].**replace**([**'Kecimen'**,** 'Besni'**],[**0**,** 1**],** inplace**=True)**  X\_data **=** rasin\_dataset**.**iloc**[:,** **:-**1**].**values  y\_data **=** rasin\_dataset**.**iloc**[:,** 7**].**values  cov\_matrix **=** np**.**cov**(**X\_data**.**T**)**  eigen\_values**,** eigen\_vectors **=** np**.**linalg**.**eig**(**cov\_matrix**)**  eigen\_values **=** eigen\_values**[**np**.**argsort**(**eigen\_values**)[::-**1**]]**  **print(**eigen\_values**)** |

A picture containing text, orange

Description automatically generated

From the result shown in the image we can estimate that the top 5 components would capture most of the information in the data.

|  |
| --- |
| pca **=** PCA**(**n\_components **=** 5**)**  pca**.**fit**(**X\_data**)**  X\_data **=** pca**.**transform**(**X\_data**)**  X\_train**,** X\_test**,** y\_train**,** y\_test **=** sklearn**.**model\_selection**.**train\_test\_split**(**X\_data**,** y\_data**,** test\_size **=** 0.25**,** random\_state **=** 23**)** |

## 

## SVM Linear Classification (Category 1)

Train the SVM Linear Classification Model and use the model to predict on test data. The confusion matrix shows the distribution of the TP,TN,FP, and FN. These values are then used to calculate the accuracy, precision, recall and f1-score.

|  |
| --- |
| svm\_linear **=** SVC**(**kernel**=**'linear'**)**  svm\_linear**.**fit**(**X\_train**,** y\_train**)**    # replace the line below with the prediction of your model  y\_pred **=** svm\_linear**.**predict**(**X\_test**)**  mat **=** confusion\_matrix**(**y\_test**,** y\_pred**)**  sns**.**heatmap**(**mat**.**T**,** square**=True,** annot**=True,** fmt**=**'d'**,** cbar**=False)**  plt**.**xlabel**(**'true label'**)**  plt**.**ylabel**(**'predicted label'**)**  plt**.**title**(**"Confusion Matrix"**)** |

Chart

Description automatically generated

|  |
| --- |
| **print(**'Accuracy: %.3f' **%** accuracy\_score**(**y\_test**,** y\_pred**))**  **print(**'Precision: %.3f' **%**precision\_score**(**y\_test**,** y\_pred**))**  **print(**'Recall: %.3f' **%** recall\_score**(**y\_test**,** y\_pred**))**  **print(**'F1-Score: %.3f' **%** f1\_score**(**y\_test**,** y\_pred**))** |

Text

Description automatically generated

## Random Forest Classification (Category 2)

Train the Random Forest Classification Model and use the model to predict on test data. The confusion matrix shows the distribution of the TP,TN,FP, and FN. These values are then used to calculate the accuracy, precision, recall and f1-score.

|  |
| --- |
| rf **=** RandomForestClassifier**(**n\_estimators **=** 500**,** max\_depth **=** 5**,** random\_state**=**0**,** criterion **=** 'gini'**)**  rf**.**fit**(**X\_train**,** y\_train**)**    # replace the line below with the prediction of your model  y\_pred **=** rf**.**predict**(**X\_test**)**  mat **=** confusion\_matrix**(**y\_test**,** y\_pred**)**  sns**.**heatmap**(**mat**.**T**,** square**=True,** annot**=True,** fmt**=**'d'**,** cbar**=False)**  plt**.**xlabel**(**'true label'**)**  plt**.**ylabel**(**'predicted label'**)**  plt**.**title**(**"Confusion Matrix"**)** |

Chart

Description automatically generated

|  |
| --- |
| **print(**'Accuracy: %.3f' **%** accuracy\_score**(**y\_test**,** y\_pred**))**  **print(**'Precision: %.3f' **%**precision\_score**(**y\_test**,** y\_pred**))**  **print(**'Recall: %.3f' **%** recall\_score**(**y\_test**,** y\_pred**))**  **print(**'F1-Score: %.3f' **%** f1\_score**(**y\_test**,** y\_pred**))** |

Text

Description automatically generated

# Conclusion

From this exercise of applying two different classification algorithms on the dataset we find that both performed well. However, the random forest model outperformed the Support Vector Machine model on all metrics. In conclusion we can state with respect to the dataset that we should use a Random Forest model.