Name: SIVANAGA SURYA VAMSI POPURI

ASU ID: 1217319207

EEE591 – Python for rapid engineering solutions

Project 1: Funny Money

Abstract: The goal of this project is to use machine learning to predict if a given currency note is counterfeit or not (Genuine or fake). Many data samples are used for training the model. Each sample consists of four features, and a class value, which represents the genuineness of the notes for that sample. Initially, a statistical analysis of the provided dataset is done to understand the importance of each of the four features in deciding the class values. Then, machine learning using six different learning methods is applied, and the performances of all six methods are compared and analyzed.

1 INTRODUCTION

Currency is being tampered with by criminals illegally, and they use it to make their own currency notes, and this compromises the validity and originality of the notes. If notes are processed and sent out without properly verifying their authenticities, the economy would be badly affected. In order to prevent this, it is important to accurately determine whether a given note is fake or not. Thus, machine learning can be used to solve this problem, by using state of the art classification algorithms to predict the authenticity of a note. In this project, this is exactly what is being done. Given a dataset, namely a txt file containing more than thousand training samples, the classifier should perform the prediction accurately. All training samples have four features, namely variance, skewness, kurtosis and entropy. All four features help in deciding if the note is fake (class = 0) or original (class = 1).

The rest of the report is organized as follows. Section 2 explains the methodology used obtaining data, analysis and training. Section 3 presents the results obtained and comparisons among learning algorithms. Section 4 concludes the report with suggestions of possible future work, and section 5 consists of the references used to implement this project.

2 METHODOLOGY AND SET UP

The implementation of the currency detection has been done on Spyder, which uses Python 3.7. Data to be trained was given as a notepad text file, which is read into the python console using the **pandas** [1] data analysis library. The task given is to extract the data and provide a statistical analysis of the features of the data, namely the **correlation** and **covariance** among the features and the class labels. Based on this, a conclusion is made as to which features are important in deciding the class of the label, and which are not. Plots and matrices are provided in the results section for visualization.

Next, the data is used to train the classifier. Training of the model based on the labels were done using **scikit-learn** [2] based classifiers, namely the following.

1. Logistic Regression
2. Decision Tree
3. Perceptron
4. Random Forest

5. Support Vector Machine
6. K-nearest neighbors.

All 6 learning algorithms are available as scikit-learn libraries for quick and easy implementation. After implementation, the accuracy of performance of each algorithm is obtained and plotted, and conclusions are made as to which algorithm performs best for the given dataset. Influence of the dataset features on the accuracy is also illustrated in the results section.

3 RESULTS

The data read into the python console using pandas was analyzed, and a statistical analysis is provided by the following figures and tables. Table 1 shows a general statistical analysis of the provided dataset, such as mean, standard deviation, quartiles, and the median.

	variance	skewness	kurtosis	entropy	class
count	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000
mean	0.433735	1.922353	1.397627	-1.191657	0.444606
std	2.842763	5.869047	4.310030	2.101013	0.497103
min	-7.042100	-13.773100	-5.286100	-8.548200	0.000000
25%	-1.773000	-1.708200	-1.574975	-2.413450	0.000000
50%	0.496180	2.319650	0.616630	-0.586650	0.000000
75%	2.821475	6.814625	3.179250	0.394810	1.000000
max	6.824800	12.951600	17.927400	2.449500	1.000000

Table 1: Descriptive analysis of dataset

Next, the correlation between the features and the class labels are given as follows. Table 2 shows in decreasing order the most highly correlated variables, and figure 1 shows the correlation matrix among all features and class labels.

Most Highly Correlated

	FirstVariable	SecondVariable	Correlation
0	skewness	kurtosis	-0.786895
1	variance	class	-0.724843
2	skewness	entropy	-0.526321
3	skewness	class	-0.444688
4	variance	kurtosis	-0.380850

Table 2: Most Highly Correlated variables.

From table 2 and figure 1, it can be concluded that the **variance and skewness** are highly correlated with the **class labels**, and thus *are most significant in determining the label value of the class*. Also, the most highly correlated features in the dataset are skewness and kurtosis. On the other hand, the entropy plays a least significant role in determining the class value due to its low correlation with the labels.

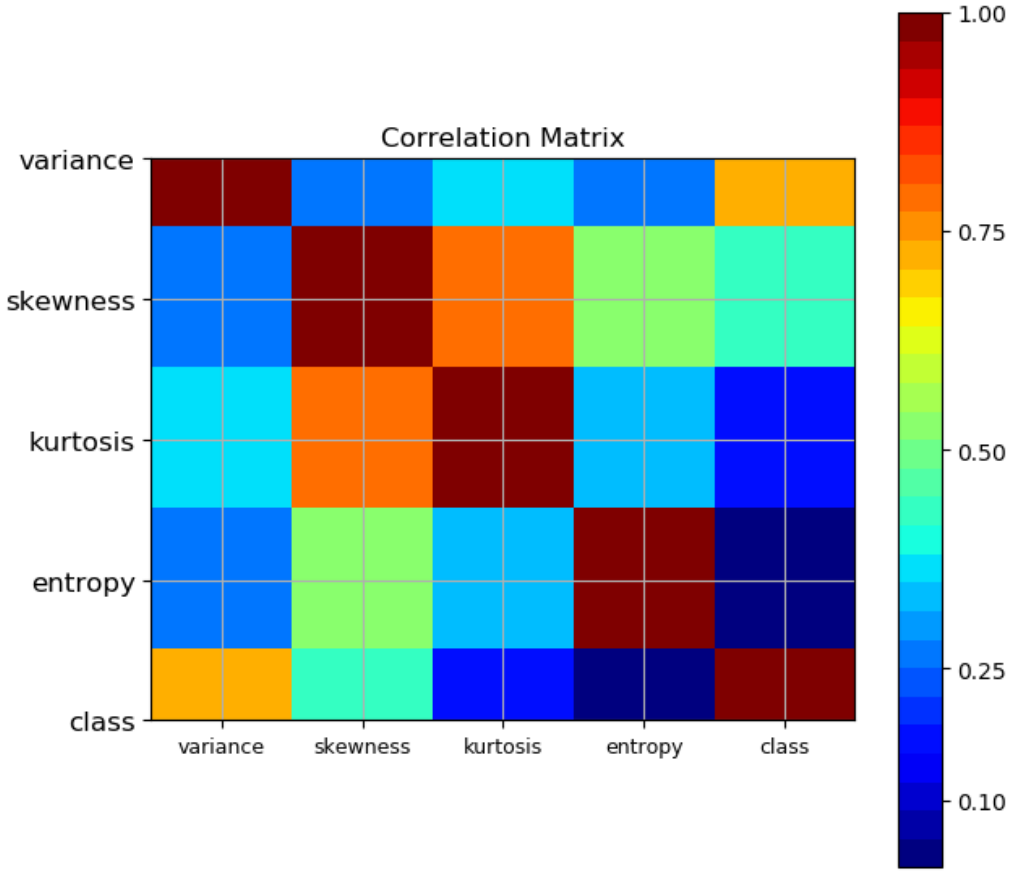

Figure 1: Correlation matrix for all features and class labels. One can infer that the entropy plays a minor role in determining the class value, and variance and skewness are crucial on the other hand.

The following table shows a cross covariance matrix obtained from the provided dataset. This tells us about the features that are dependent/independent on each other and on the class labels.

	variance	skewness	kurtosis	entropy	class
variance	8.081299	4.405083	-4.666323	1.653338	-1.024310
skewness	4.405083	34.445710	-19.905119	-6.490033	-1.297386
kurtosis	-4.666323	-19.905119	18.576359	2.887241	0.333985
entropy	1.653338	-6.490033	2.887241	4.414256	-0.024464
class	-1.024310	-1.297386	0.333985	-0.024464	0.247112

Table 3: Cross covariance matrix between features.

From the cross-co-variance matrix, similar to the inferences made from the correlation matrix, the skewness and kurtosis have a high cross covariance. Also, the entropy has the least cross-covariance with the class label value. We can thus conclude that the variance and skewness are best predictors of the money authenticity. Kurtosis plays a moderate role, and entropy plays a weak role.

Finally, a pair plot has been provided to support the inferences made from the covariance matrix and the correlation matrix and plots. Figure 2 does exactly the same.

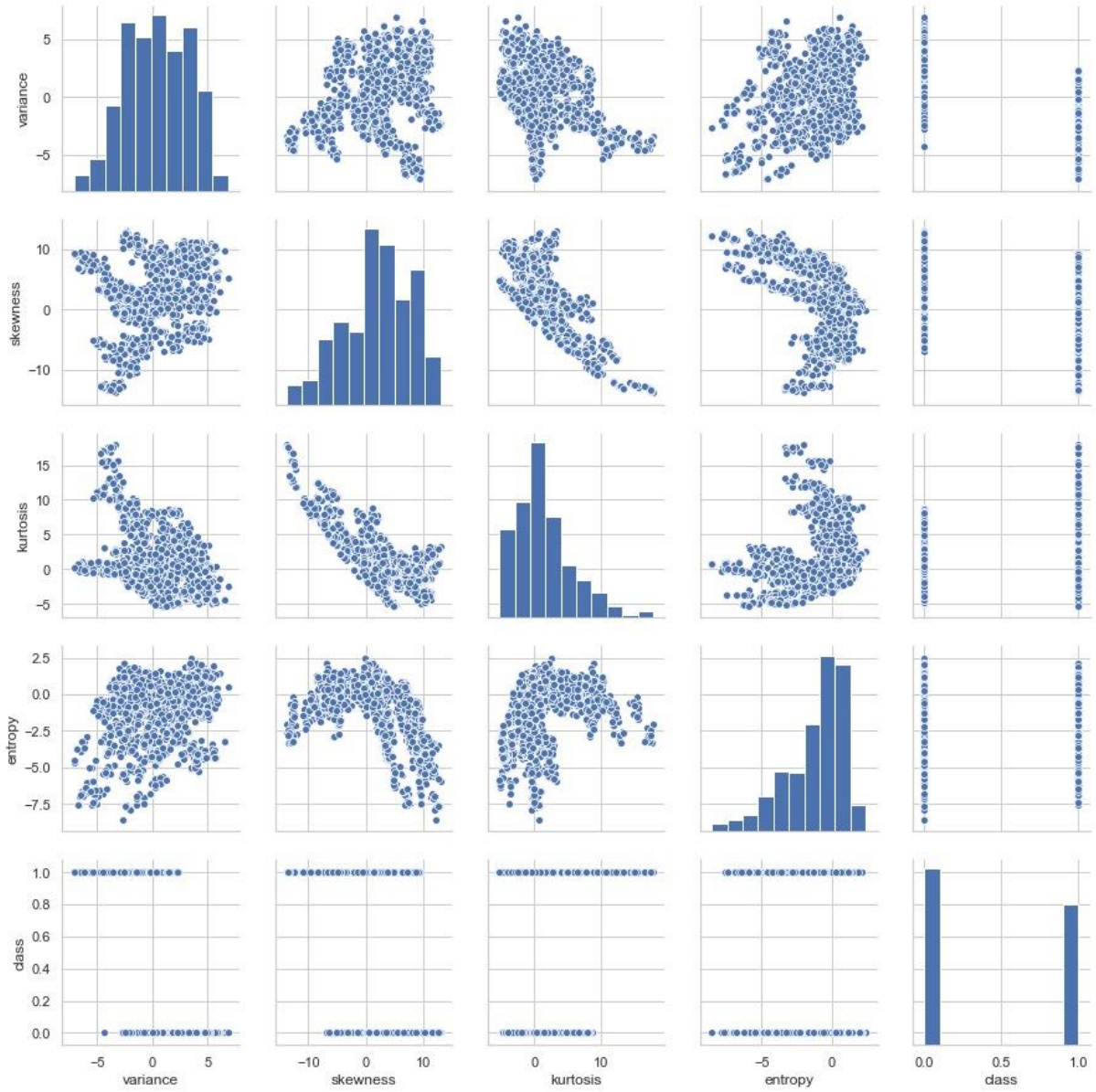

Figure 2: Pair plot for the provided dataset.

Now, it's time for implementing machine learning algorithms for performing classification. As mentioned before, six different learning algorithms were used to classify the currency note. The following plot describes the accuracies of all six algorithms, done on the testing set.

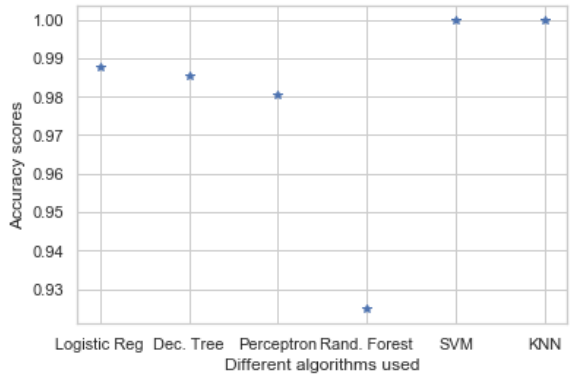

Figure 3: Plot describing percentage accuracies for each learning model

Table 4 presents the exact scores for each algorithm. From the table, one can infer that the **K-nearest neighbors** (for K = 3) and the **Support Vector Machine** learning models are 100% accurate in predicting the class labels. The least score is given by the Random Forest classifier.

Name of the Algorithm	Accuracy Score (Percentage)
Logistic Regression	98.78%
Decision Tree	98.54%
Perceptron	98.05%
Random Forest	92.48%
Support Vector Machine	100%
K-nearest-neighbors (K = 2)	100%

Table 4: Accuracies of each learning algorithm.

Since the table says that K-nearest neighbors (at K = 3) and the Support Vector Machine models have the maximum accuracies, these methods would be the best choice for classifying fake and original notes, reliably and accurately.

4 CONSLUSION

Thus, in this project, a classifier for predicting counterfeit money was implemented using machine learning algorithms. Data given in a text file was read using pandas and analyzed. Then, the data was split into training and testing datasets in order to train and test the performance of six different classifiers, and pick the best performing classifier for this dataset. And so, this can be used in future to prevent illegally tampered currency notes from spreading to the world.

In future, principle component analysis can be done to reduce features that don't contribute much (such as entropy) to determining the class of the currency note. This helps a lot in reducing computational complexity, whilst managing to maintain accuracy scores as much as possible.

5 REFERENCES

[1] Pandas official documentation: https://pandas.pydata.org/docs/

[2] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.