Bank Note Authentication using K-Means Clustering

Lim Jun Jie

School of Computing

Singapore Polytechnic

Singapore, Singapore

junjie.21@ichat.sp.edu.sg

***Abstract*— Authenticating bank notes have always been a big problem in the banking industry. As the ability to create near-original counterfeits become easier, better authentication methods should also be deployed to combat counterfeiters. A solution is to leverage machine learning models to detect counterfeits automatically. The objective of this paper is to train a K-Means Clustering model, an Unsupervised Learning algorithm to achieve segmentation between real and forged notes.**

Keywords— Machine Learning, Unsupervised Learning, K-Means Clustering, Counterfeit Bank Notes

I. INTRODUCTION

Counterfeit bank notes have many ill-effects on the economy. Mass circulation of fakes can spike inflation [1] as more money is circulated in the economy. The value of the currency will also decrease as a result. Counterfeit notes are so dangerous to a country's economy that harsh laws are set up around it, such as in Singapore, where those guilty of possessing counterfeit bank notes can face up to a jail term of 20 years and a fine [2]. Counterfeiting is also hard to combat against as the ability to create better counterfeit notes is prevalent due to advanced image processing and artwork software [3] that can be acquired at a cheap price.

As technology advances, the number of possible techniques that can be used to detect counterfeit bank notes also multiply. This paper will be focused on exploring the possibility of leveraging K-Means Clustering to detect counterfeit notes, as well as point out the strengths and limitations of using this solution. This model will be trained on image data taken from genuine and forged banknote specimens [4], where the model is supposed to achieve segmentation between counterfeit and original notes based on the image data.

II. RELATED WORKS

There is already some experimentation done on using K-Means Clustering for authentication of bank notes. Ragavi E [5] displayed that after normalizing the data, the K-Means Clustering model only passed 3 counterfeit notes as genuine out of 1,372 bank notes when setting the number of clusters k to 2.

Mudassir Khan and Mahtab Alam [6] suggested that setting the number of clusters k to 3 can help to create a cluster for a third group of banknotes that could be subjected to further analysis, an example scenario being that genuine banknotes that are in bad condition may be mistaken for a counterfeit one.

III. METHODOLOGY

In this paper, the following Python libraries will be used:

Pandas - Built on top of NumPy, it is a useful data science tool that can perform data wrangling, display numerical summaries and other data cleaning tasks [7].

Seaborn - A visualization library that is built on Matplotlib. It can generate many types of statistical graphs and works well with Pandas data structures [8].

Plotly - Another Python visualization library, Plotly can display interactive as well as 3-dimensional visualizations easily. This library is built on top of the Plotly JavaScript library [11].

sklearn - An extremely useful Python library for Machine Learning. This library contains various Machine Learning algorithms and statistical modeling tools [13].

A. Dataset

The data set used is taken from the University of California Irvine (UCI) Machine Learning Repository. The name for this dataset in the repository is the Banknote Authentication Dataset. It contains features of banknote images which were extracted using Wavelet Transform. An industrial camera usually used for print inspection was used to get the images [4].

The dataset shown in Figure 1 contains image data for both genuine and counterfeit banknote-like specimens. The dataset has 1372 rows and 5 attributes: variance, skewness and kurtosis of the Wavelet Transformed image, as well as the entropy of the image and the class [4], which was one hot encoded into 0 and 1 to represent genuine and counterfeit bank notes respectively [10].

	variance	skewness	curtosis	entropy	label
0	3.62160	8.66610	-2.8073	-0.44699	0
1	4.54590	8.16740	-2.4586	-1.46210	0
2	3.86600	-2.63830	1.9242	0.10645	0
3	3.45660	9.52280	-4.0112	-3.59440	0
4	0.32924	-4.45520	4.5718	-0.98880	0
...	...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949	1
1368	-1.38870	-4.87730	6.4774	0.34179	1
1369	-3.75030	-13.45860	17.5932	-2.77710	1
1370	-3.56370	-8.38270	12.3930	-1.28230	1
1371	-2.54190	-0.65804	2.6842	1.19520	1

Fig. 1. DataFrame of first and last 5 rows of dataset

B. Exploratory Data Analysis

To visualize and dive deeper into the dataset, Exploratory Data Analysis is required. The goal is to find out if there are any useful insights that can help with modeling the K-means Clustering algorithm.

As this dataset contains no missing values [4] and it was loaded into a Pandas DataFrame from a text file with no errors, U can jump straight into visualizing the data for easier analysis.

1) Data Distribution

An interesting observation I have discovered is that the data has different distributions for variance, skewness, and kurtosis of the image between counterfeit and real banknotes as the histograms do not overlap completely as shown in Figure 2. The differing distributions between the 2 labels can help K-Means algorithm to segregate the 2 classes better, as the K-Means algorithm uses Euclidean Distance to calculate the distances from data points to the centroids to group every data point to a cluster [9]. This differing distribution may help to segregate the clusters better.

The entropy of the image has a similar distribution for both real and counterfeit banknotes as the histograms completely overlap each other. This may suggest that the entropy may not be useful for the clustering algorithm as the K-Means algorithm may not be able to segregate the banknotes correctly as the distributions do not have a difference. The mean, median and mode between the 2 classes are also roughly the same as shown in Figure 2.

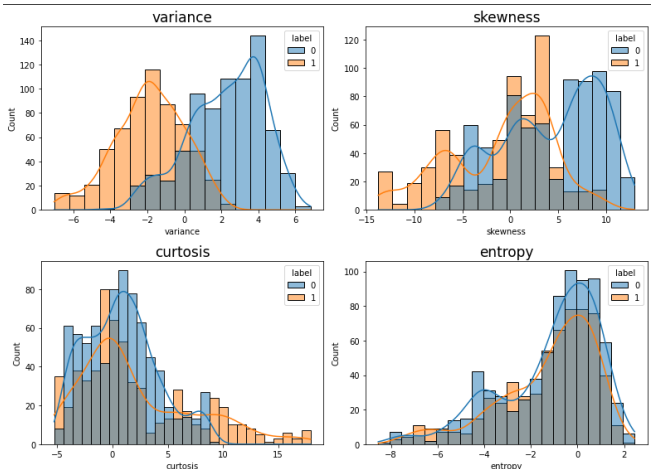

Fig. 2. Histograms for all features in the dataset

2) Outlier Analysis

Boxplots were also plotted to make visualizing the different distributions easier as well as to show outliers, as shown in Figure 3. From Figure 3, variance, skewness and curtosis do not have overlapping boxes. Outliers can be seen for the counterfeit (orange) boxplot for variance and curtosis, suggesting that the outliers can be useful in helping to create segmentation between real and counterfeit bank notes.

There is 1 outlier point in the real (blue) banknote boxplot of the variance feature. The outlier value is even lower than the mean of the counterfeit variance value. A hypothesis for this occurrence is that the banknote may be in a bad condition and therefore has a variance value similar to a counterfeit banknote, or that the camera could not capture that particular bank note properly.

The entropy feature shows multiple outliers at the same points for both counterfeit and real banknotes, as well as overlapping boxplots. The fences of both boxplots are also around the same points. This further suggests that entropy may not be useful in helping to clustering real and counterfeit bank notes.

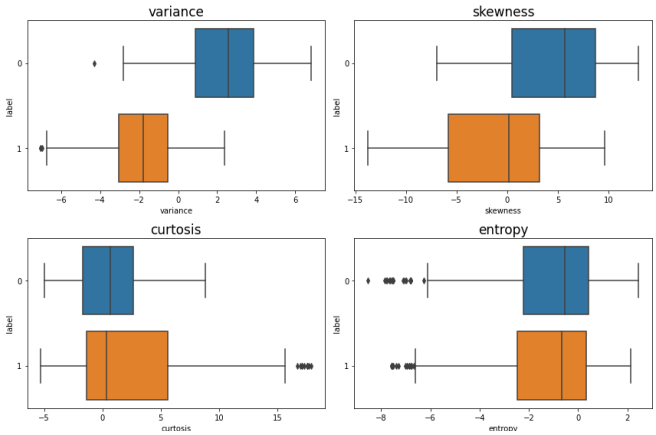

Fig. 3. Boxplots of all features separated by class

3) Simple Cluster Analysis

To find out if the different features can group the 2 classes, real and counterfeit bank notes in a bivariate scatterplot, I plotted a pair plot as shown in Figure 4.

From Figure 4, it is observed that variance, skewness and curtosis were able to segregate the labels into 2 distinct clusters when plotted against each other with minimal overlapping. However, when entropy was plotted with variance and skewness, the clusters were not as distinct as the clusters are clearly overlapping with each other. When entropy was plotted with curtosis, the clusters almost completely overlap each other. This suggests that we can do K-Means clustering on this dataset with only 3 features, without curtosis.

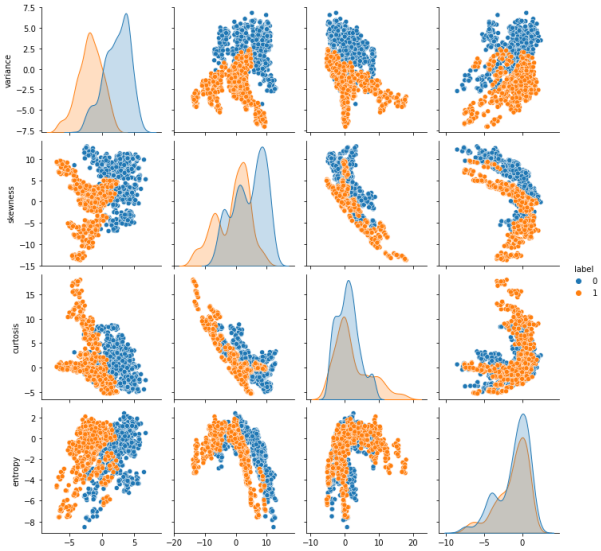

Fig. 4. Pair plot of all features colored by label

I then dropped the entropy feature and plotted a 3-dimensional scatterplot to find out if variance, skewness and curtosis data can segregate real and counterfeit bank notes, as shown in Figure 5. When the scatterplot is colored by the label feature, 2 distinct clusters can be seen in the 3-dimensional space, showing that using only curtosis, variance and skewness features can effectively create a segmentation between counterfeit and real bank notes.

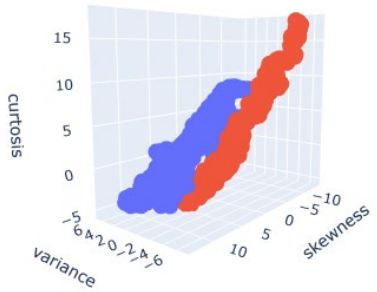

Fig. 5. 3-Dimensional scatterplot without entropy feature, colored by class

C. *Data Preprocessing*

1) *Dropping features*

Using the results from Exploratory Data Analysis, I have concluded that the entropy feature is not required for modeling the K-Means algorithm, thus I would be dropping the entire feature completely using the Pandas drop function, as shown in Figure 6. In addition, I have also decided not to drop the outliers and deal with the skewness of the distribution as the outliers and distribution of the data would be useful in helping to cluster real and counterfeit bank notes.

```
df_clean = df.drop(['entropy'], axis=1)
```

Fig. 6. Dropping entropy feature from dataset using Pandas

2) *Feature Scaling*

When using Pandas to generate a numerical summary on the features as shown in Figure 7, the 3 features have differing means and standard deviations. In this case, scaling the data to the same range is desired so that features with higher magnitudes, in this case skewness and curtosis would not weigh higher than variance and cause the algorithm to favor the 2 features over variance when clustering the data points.

	variance	skewness	curtosis
count	1372.000000	1372.000000	1372.000000
mean	0.433735	1.922353	1.397627
std	2.842763	5.869047	4.310030
min	-7.042100	-13.773100	-5.286100
25%	-1.773000	-1.708200	-1.574975
50%	0.496180	2.319650	0.616630
75%	2.821475	6.814625	3.179250
max	6.824800	12.951600	17.927400

Fig. 7. Numerical Summary using Pandas

To scale this dataset, sk-learn's MinMaxScaler will be used. MinMaxScaler normalizes all features in this dataset to a minimum value of 0 and maximum value of 1 with the formula being shown in Figure 8, and the implementation of MinMaxScaler on the dataset being shown in Figure 9. After scaling, the minimum and maximum values for all 3 features are 0 and 1 respectively, as shown in Figure 10, using the Pandas describe function to generate a numerical summary.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Fig. 8. Formula for MinMaxScaler

```
features_to_scale = ['variance', 'skewness', 'curtosis']
scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(df_clean[features_to_scale])
scaled_df = pd.DataFrame(scaled_features, columns=features_to_scale)
```

Fig. 9. Implementation of MinMaxScaler using sklearn

	variance	skewness	curtosis
count	1372.000000	1372.000000	1372.000000
mean	0.539114	0.587301	0.287924
std	0.205003	0.219611	0.185669
min	0.000000	0.000000	0.000000
25%	0.379977	0.451451	0.159869
50%	0.543617	0.602168	0.254280
75%	0.711304	0.770363	0.364674
max	1.000000	1.000000	1.000000

Fig. 10. Numerical Summary of scaled features using Pandas

D. *Modeling and Analysis*

I am now going to apply the K-Means clustering algorithm preprocessed dataset. The sklearn has a K-Means model function which can be used to fit to the dataset. Then after that I am going to evaluate the performance of K-Means clustering using inertia and

1) *K-Means Clustering*

K-Means clustering has a parameter, k, which represents the number of clusters to create. The algorithm randomly initializes k centroids at random locations then iteratively repositions the centroids by calculating the mean of data points in the cluster [14], in order to minimize an objective function called inertia. Inertia is calculated by squaring the Euclidean distance between each data point and its centroid,

then summing the squares together for that cluster [15]. Lower inertia would mean that the clusters are denser, with data points being close to the centroid of that cluster. The iteration process will go on until the maximum specified iterations have been reached or when the centroids of clusters stop changing or change only by a little [16]. The formula for inertia is shown in Figure 11.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Fig. 11. Formula for Inertia for K-Means Clustering

2) *Visualization of Cluster Results*

The preprocessed dataset was fitted on sklearn's K-Means algorithm, setting the number of clusters k to 2 as well as setting the random state to a number, to ensure the same results every time. The code for implementation is shown in Figure 12. After fitting the K-means model on the dataset, I proceeded to plot a 3-dimensional plot using color-coded by the cluster labels using plotly as shown in Figure 13. From Figure 13, K-Means did a good job at creating 2 distinct clusters for this 3-Dimensional dataset. The centroids which are in orange lie in the middle of each cluster when visualized in a 3-dimensional space.

```
kmeans = KMeans(random_state=48, max_iter=1000, n_clusters=2)
kmeans.fit(kmeans_data)
```

Fig. 12. K-Means Clustering using sklearn

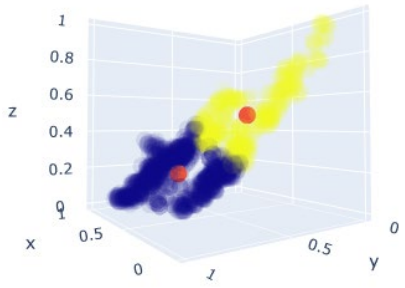

Fig. 13. Visualizing Clustering Result of K-Means

I proceeded to plot each combination of 2 features in a scatter plot to visualize how the scatters look when using 2 features only. When the clusters are visualized using variance and skewness features as shown in Figure 14, there is minimal overlap between the 2 clusters, The centroids in red are roughly in the middle for both clusters. This suggests that looking at variance and skewness of the wavelet transformed image data alone, the K-Means algorithm can achieve a good segmentation of 2 clusters.

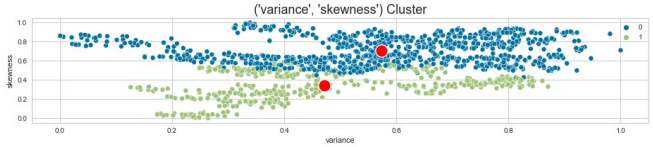

Fig. 14. 2-Dimensional Cluster Visualization using variance and skewness

2-dimensional scatter plots were also plotted for variance with curtosis, and skewness with curtosis as shown in Figure 15 and 16 respectively.

Looking at the clusters when plotted using variance and curtosis features in Figure 15, the clusters are slightly overlapping each other. The same occurrence can be seen in Figure 16, when the clusters are plotted with skewness and curtosis, the clusters also seem to overlap. These 2 observations suggest that the curtosis feature is not as useful as the variance and skewness when using K-Means to cluster the data points.

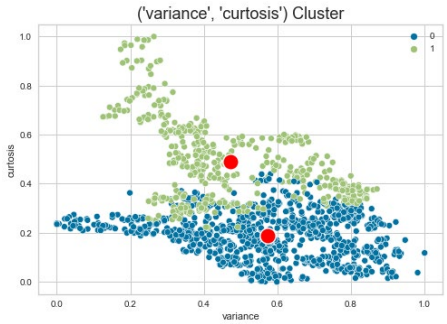

Fig. 15. 2-Dimensional Cluster Visualization using variance and curtosis

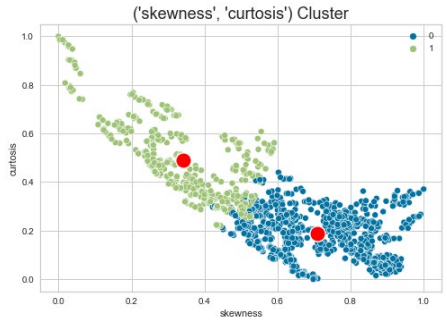

Fig. 15. 2-Dimensional Cluster Visualization using skewness and curtosis

3) *Evaluation of Clustering results*

As the ground truth labels are known, using intrinsic measures such as Silhouette Score or the Calinski-Harabasz Index [17] to assess the clustering results would not be appropriate.

A way to evaluate the clusters for this paper is to compare the 3D scatterplots color coded by the original labels and the cluster labels as shown in Figure 16 and 17 respectively. Comparing the clusters of both 3-Dimensional scatter plots, the segmentation that K-Means Clustering does not have the same segmentation compared to the segmentation created by the ground truth labels.

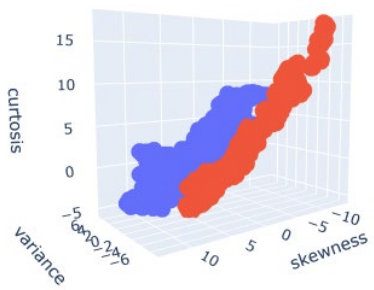

Fig. 16. 3-Dimensional Scatter Plot colored by original labels

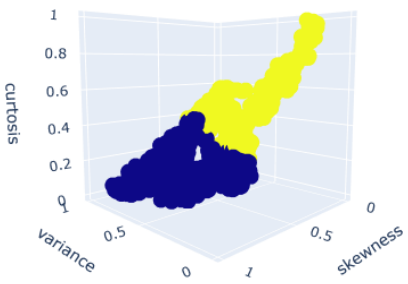

Fig. 17. 3-Dimensional Scatter Plot colored by clustered labels

IV. Results and Discussion

From this simple experimentation on using K-Means Clustering to cluster real and counterfeit bank notes, K-Means clustering was not able to correctly identify the 2 clusters, as the segmentation was wrong when comparing to the scatter plot that was segmented by ground truth labels. This may be due to inertia favoring clusters where data points are closest to the centroids as the objective of K-Means Clustering is to find the smallest within-cluster-sum-of-squares. The segmentation using ground truth labels shows that both clusters have high variance due to the data points having a wide spread as shown in Figure 16, and the K-Means algorithm may not have been able to replicate the same clusters due to the true clusters being widely spread out, and the inertia would therefore be high. Spectral Clustering or density-based clustering such as DBSCAN may be possible models that can be tried out on this problem of identifying counterfeit bank notes using Unsupervised Learning.

Another hypothesis for the inaccurate clustering using K-Means is due to the curtosis feature. The curtosis feature when plotted against the other 2 features separately in a bivariate scatterplot, overlapping clusters can be seen. Removing the curtosis feature may improve the quality of the clusters generated by the K-Means algorithm.

V. Conclusions

In conclusion, using K-Means clustering may not be the best choice for authentication of bank notes, however further analysis would have to be done such as testing without the curtosis feature. In the future, if there are more sophisticated methods for image processing of bank notes, more detailed data will be available to model using K-Means, which may also improve the results.

Although this paper is focused on using K-Means algorithm for this particular use case, other models can also be feasible as well, such as density-based clustering techniques or even Deep Learning models can come into play.

References

[1] Wikipedia Contributors (2019). Counterfeit money. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Counterfeit_money.

[2] CNA. (n.d.). *2 men charged with possessing fake S$10,000 notes*. [online] Available at: https://www.channelnewsasia.com/singapore/two-men-charged-fake-10000-notes-bank-2586316 [Accessed 11 Aug. 2022].

[3] Read 'Is That Real?: Identification and Assessment of the Counterfeiting Threat for U.S. Banknotes' at NAP.edu. (n.d.). [online] *nap.nationalacademies.org*. Available at: https://nap.nationalacademies.org/read/11638/chapter/4#16 [Accessed 11 Aug. 2022].

[4] archive.ics.uci.edu. (n.d.). *UCI Machine Learning Repository: banknote authentication Data Set*. [online] Available at: https://archive.ics.uci.edu/ml/datasets/banknote+authentication#.

[5] Banknote Authentication Analysis Using Python K-Means Clustering. (n.d.). [online] Available at: https://www.ijisrt.com/assets/upload/files/IJISRT20OCT060.pdf [Accessed 11 Aug. 2022].

[6] Khan, M. and Alam, M. (2021). Big Data Analytics to Authenticate Bank Notes Using K-Means Clustering. *Helix - The Scientific Explorer | Peer Reviewed Bimonthly International Journal*, [online] 11(3), pp.1–6. Available at: https://www.helixscientific.pub/index.php/home/article/view/361/300 [Accessed 11 Aug. 2022].

[7] ActiveState. (n.d.). *What Is Pandas in Python? Everything You Need to Know*. [online] Available at: https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/#

[8] Pydata.org. (2012). *An introduction to seaborn — seaborn 0.9.0 documentation*. [online] Available at: https://seaborn.pydata.org/introduction.html.

[9] gajawada (2019). *K-Means Clustering for beginners*. [online] Medium. Available at: https://towardsdatascience.com/k-means-clustering-for-beginners-2dc7b2994a4# [Accessed 11 Aug. 2022].

[10] jamesdmccaffrey (2020). *In the Banknote Authentication Dataset, Class 0 is Genuine / Authentic*. [online] James D. McCaffrey. Available at: https://jamesmccaffrey.wordpress.com/2020/08/18/in-the-banknote-authentication-dataset-class-0-is-genuine-authentic/ [Accessed 11 Aug. 2022].

[11] plotly.com. (n.d.). *Getting Started with Plotly*. [online] Available at: https://plotly.com/python/getting-started/.

[12] Data Science Stack Exchange. (n.d.). *machine learning - Do Clustering algorithms need feature scaling in the pre-processing stage?* [online] Available at: https://datascience.stackexchange.com/questions/22795/do-clustering-algorithms-need-feature-scaling-in-the-pre-processing-stage [Accessed 11 Aug. 2022].

[13] scikit-learn (2019). *scikit-learn: machine learning in Python*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/.

[14] StatQuest: K-means clustering. (2018). *YouTube*. Available at: https://www.youtube.com/watch?v=4b5d3muPQmA.

[15] Henrique, A. (2021). *Clustering with K-means: simple yet powerful*. [online] Medium. Available at: https://medium.com/@alexandre.hsd/everything-you-need-to-know-about-clustering-with-k-means-722f743ef1c4# [Accessed 11 Aug. 2022].

[16] Analytics Vidhya. (2019). *K Means Clustering | K Means Clustering Algorithm in Python*. [online] Available at: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/# [Accessed 11 Aug. 2022].

[17] Manimaran (2019). *Clustering Evaluation strategies*. [online] Medium. Available at: https://towardsdatascience.com/clustering-evaluation-strategies-98a4006fcfc.