Forged Banknotes Analysis (Report)

By Ryan Ng

Purpose

Forged banknotes have become commonplace for criminal activity and used especially in fraud cases. The purpose of this assignment try to automate the detection of forged banknotes.

Dataset

The original dataset was taken from the following website.
https://www.openml.org/d/1462

For the convenience of students, a simplified dataset containing 2 columns were provided for students (originally there were 5 columns).

V1. variance of Wavelet Transformed image (continuous)
V2. skewness of Wavelet Transformed image (continuous)

Each column has 1372 records (excluding the headers) and there are no null values in the dataset (so no need to remove null values).

Methods

The first step was to determine the statistical properties of the dataset (i.e. mean and standard deviation) using the numpy mean() and std() functions.


```
# Tabulate statistical properties
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

v1 = df['V1']
v2 = df['V2']

v1_mean = np.mean(v1)
v1_std = np.std(v1)

v2_mean = np.mean(v2)
v2_std = np.std(v2)
print(f"v1_mean: {v1_mean}, v1_std: {v1_std}, v2_mean: {v2_mean}, v2_std: {v2_std}")
```

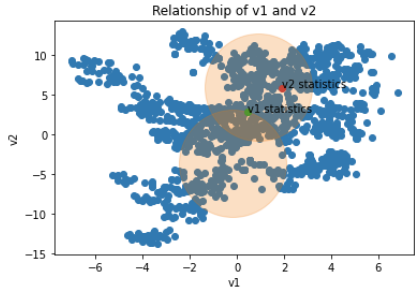
```
v1_mean: 0.43373525728862977, v1_std: 2.8417264052060993, v2_mean: 1.9223531209912554, v2_std: 5.866907488271995
```

Next, the initial 2 clusters were identified based on the cluster centres, which were determined by running the stacked columns of the dataframe through the .fit() function provided in the KMeans library. KMeans can be used to determine relationships between the data points (i.e. possible features that could be visualized but not seen in the original dataset).

```
plt.title('Relationship of v1 and v2')
v1 = df['V1']
v2 = df['V2']

plt.xlabel('v1')
plt.ylabel('v2')
plt.scatter(v1, v2)
plt.scatter(clusters[:,0], clusters[:,1], s = 10000, alpha = 0.25)
plt.scatter(v1_mean, v1_std) #labelled green
plt.text(v1_mean, v1_std, "v1 statistics")

plt.scatter(v2_mean, v2_std) #labelled red
plt.text(v2_mean, v2_std, "v2 statistics")
plt.show()
```

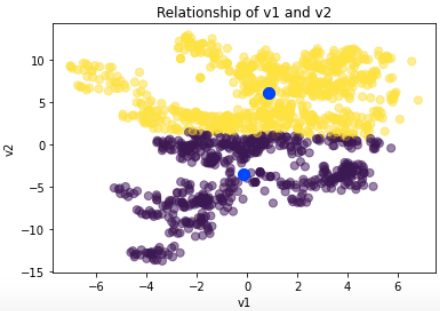


Thereafter, I proceeded on to use a separate predict() method in order to segregate the clusters according to the previously determined cluster centres.

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

v1_v2 = np.column_stack((v1, v2))
kmeans_result = KMeans(n_clusters = 2).fit(v1_v2)
pred = kmeans_result.predict(v1_v2)
```

```
clusters = kmeans_result.cluster_centers_
plt.title('Relationship of v1 and v2')
plt.xlabel('v1')
plt.ylabel('v2')
plt.scatter(v1, v2, s= 50, c = pred, alpha = 0.5)
plt.scatter(clusters[:,0], clusters[:,1], c = 'blue', s = 100)
plt.show()
```



As seen in the plot above, there are a few overlaps between the yellow and purple clusters, which is the portion that the algorithm may not have done very well in clustering.

Recommendations and Summary

Since a large part of the v1 values or the variance is between 4 and -4 while the skewness varies quite largely, while the range of skewness is broader, it is more likely the outlier data may affect how the clustered data are grouped. The overlaps also show that the algorithm is not likely to be able to achieve a very high accuracy, which is critical in helping to identify forged banknotes and catch criminals. The algorithm also has a high chance of identifying false positives with that being said. In my opinion, it would be better to use a supervised algorithm like neural networks so that there is a higher chance of correctly identifying forged banknotes as compared to having many false positives or true negatives identified by the algorithm.

In addition, perhaps a larger dataset should be used to provide more accurate results since there will be more training data to be trained on.