Challenge One
Jovana Radinovic

Data Pretreatment

The dataset banknote authentication has 1372 rows and 5 columns.

	variance	skewness	curtosis	entropy	class
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

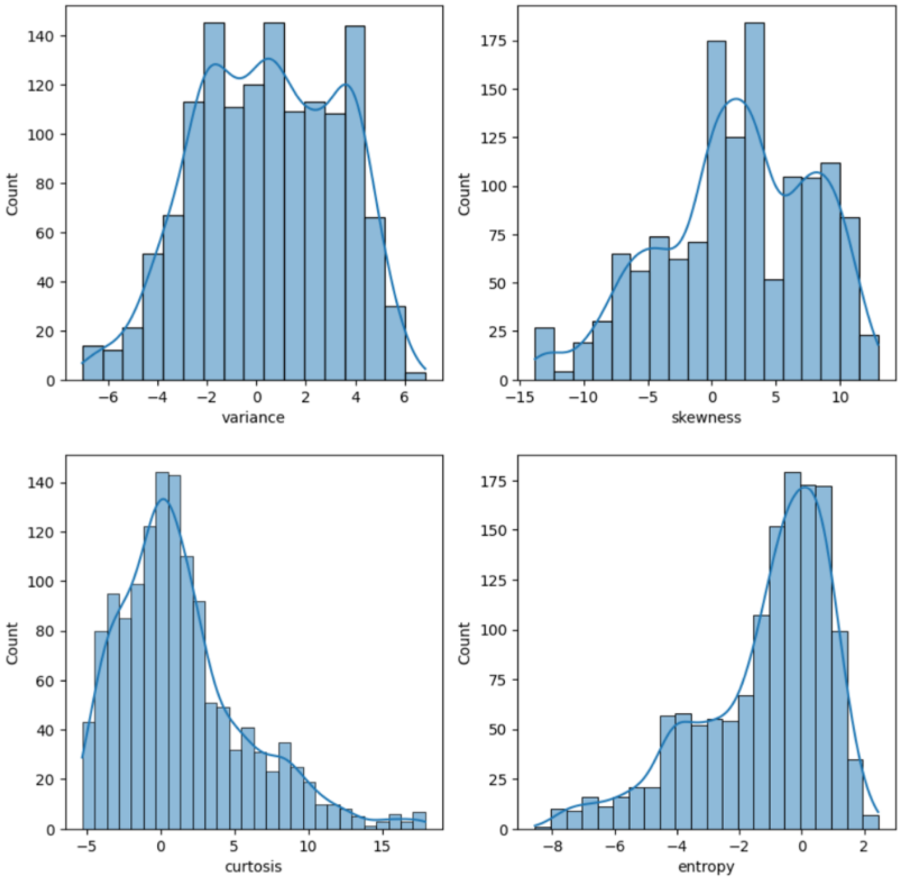
1372 rows × 5 columns

```
variance    float64
skewness    float64
curtosis    float64
entropy     float64
class         int64
dtype: object
```

The dataset is ordered in ascending by class so we need to do shuffling because without shuffling data the splitting into train set and test set could lead to wrong results, and the predictions wouldn’t be true.

To have an overview of class values I used barplot and the results are: 610 observation for class =1 and 762 for class = 0

For the continuous quantitative variables, I applied KDE to see if they follow a **Gaussian distribution:**



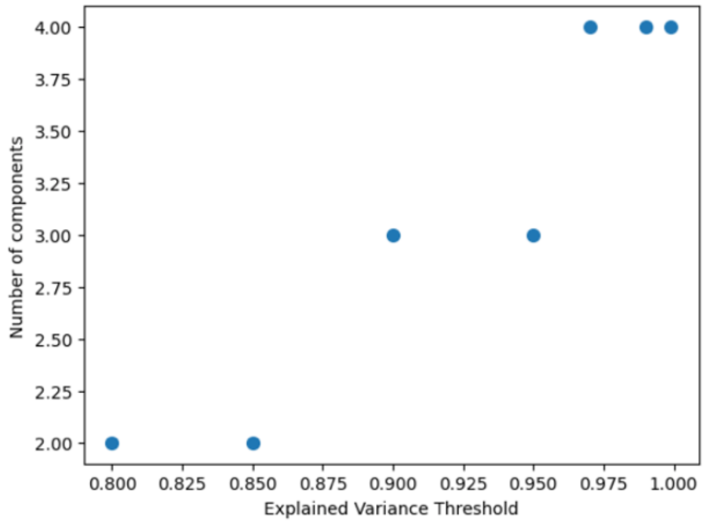
To evalute the dependency between the variables I used the **correlation matrix**.

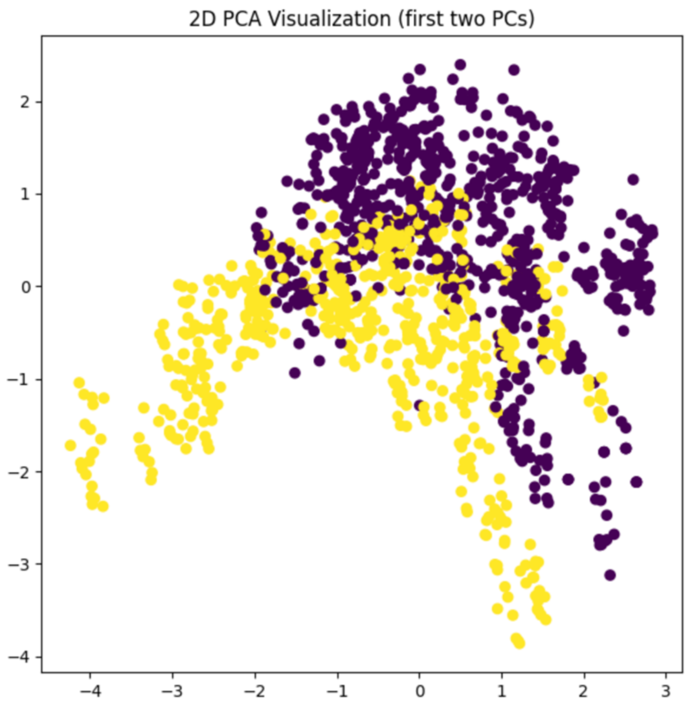
	variance	skewness	curtosis	entropy	class
variance	1.00	0.26	-0.38	0.28	-0.72
skewness	0.26	1.00	-0.79	-0.53	-0.44
curtosis	-0.38	-0.79	1.00	0.32	0.16
entropy	0.28	-0.53	0.32	1.00	-0.02
class	-0.72	-0.44	0.16	-0.02	1.00

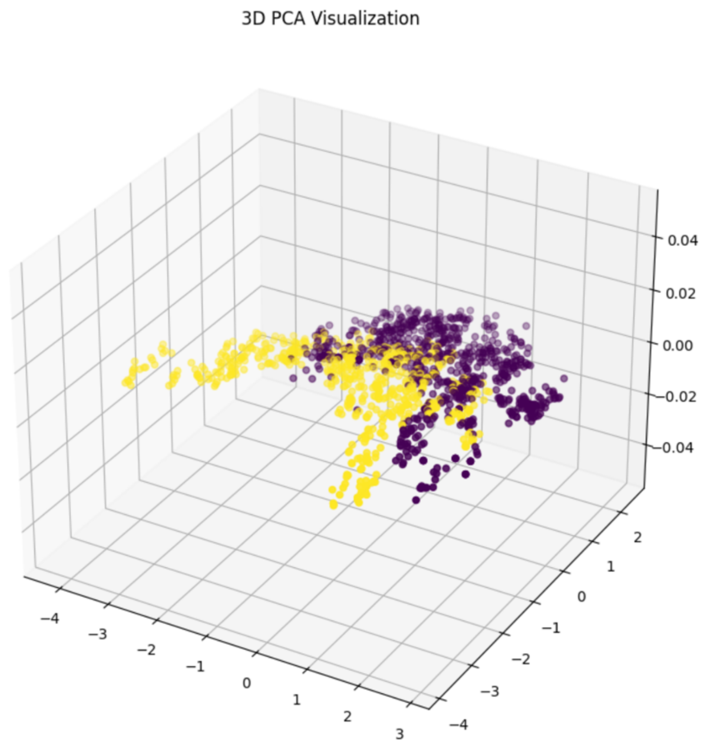
2.Unsupervised learning

PCA:

After comparing the different values of explained variance based on the number of components, I used PCs and obtained the following PCA:

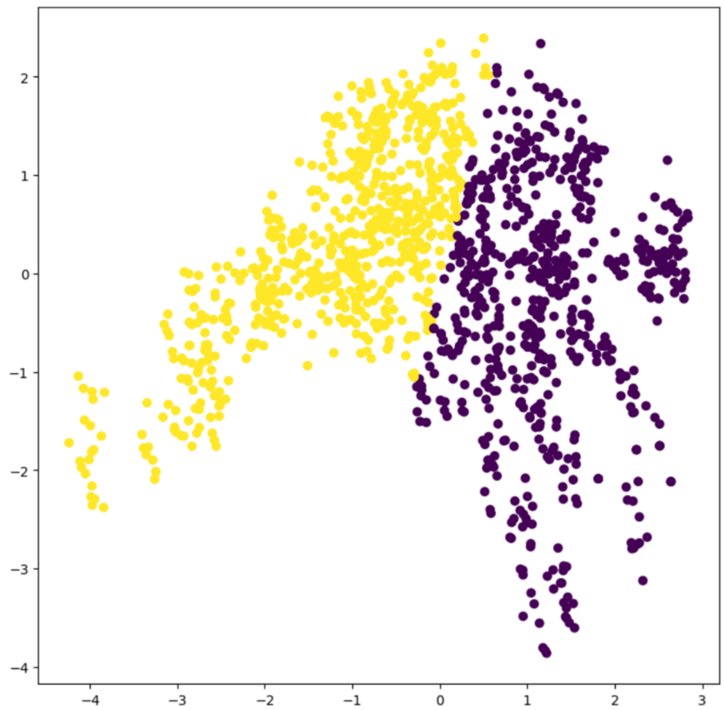


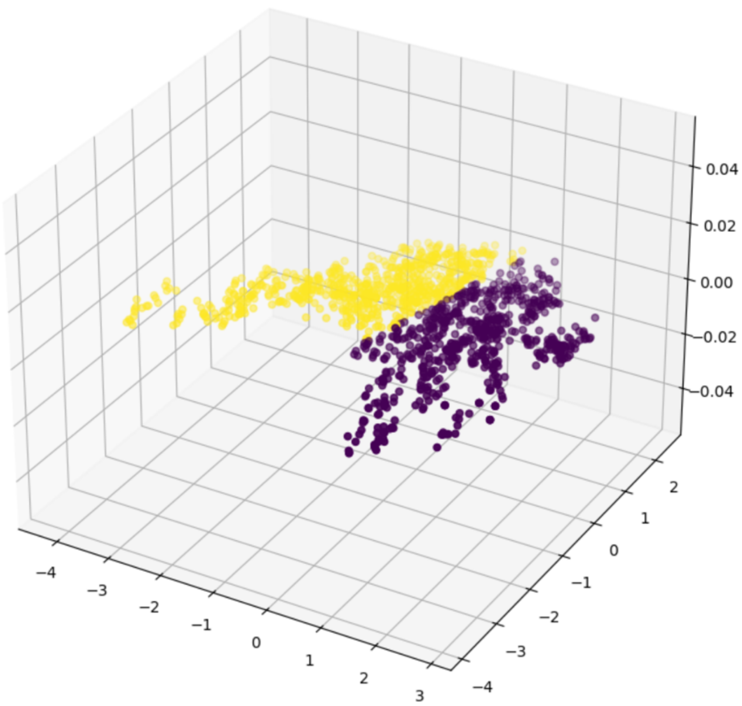




The classes aren't linearly separable.

K-means



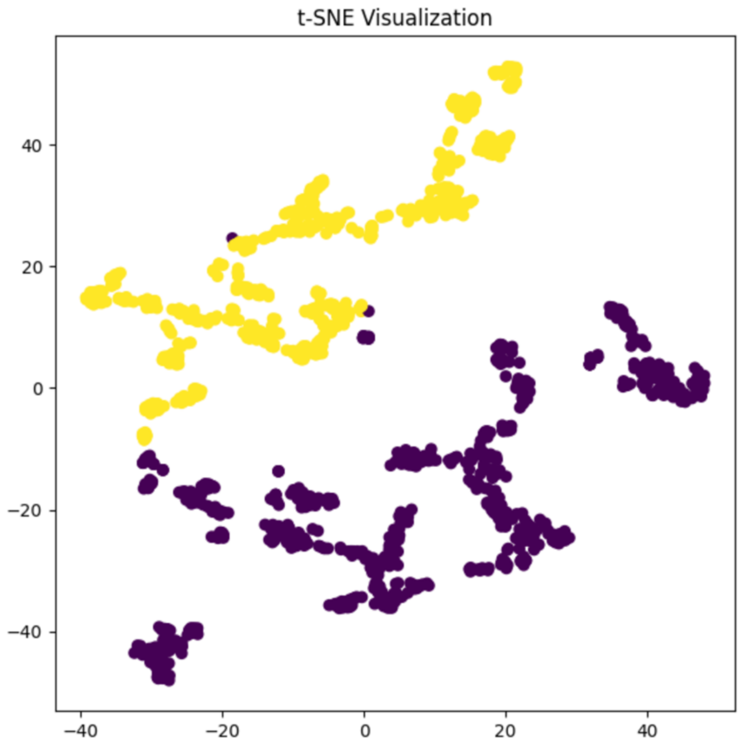


The minimum loss is 3453. Using **NMI** to scale the. Results between 0 (no mutual information) and 1 (perfect correlation)

NMI for k-means gave me the result: 0.01099, so we can assert that the agreement of the label assigned by k-means with the true labels is low.

t-SNE

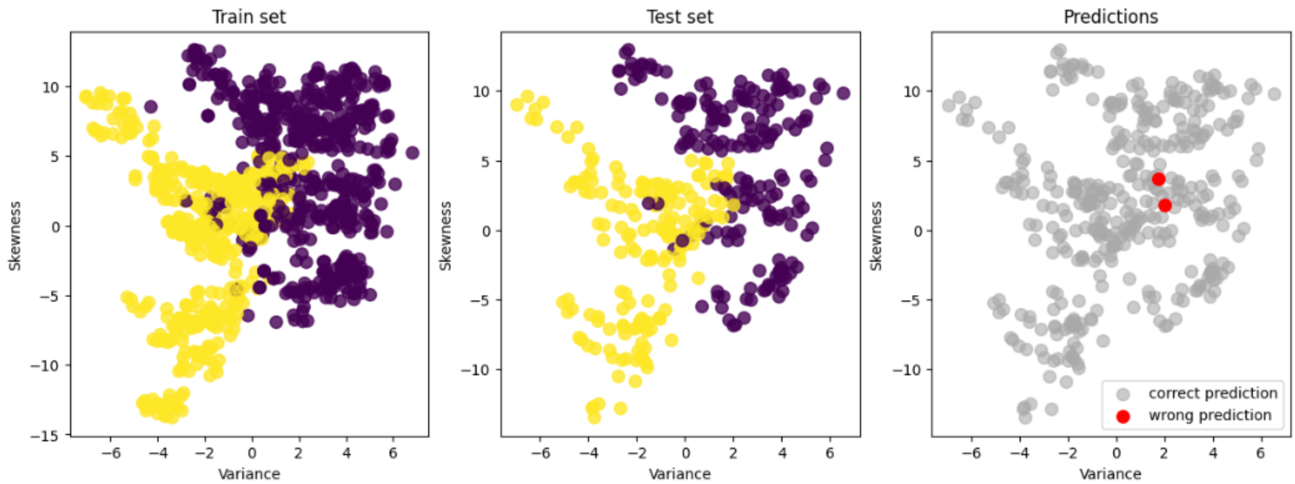
When applying t-SNE in the clustering process I obtained two distinct clusters:



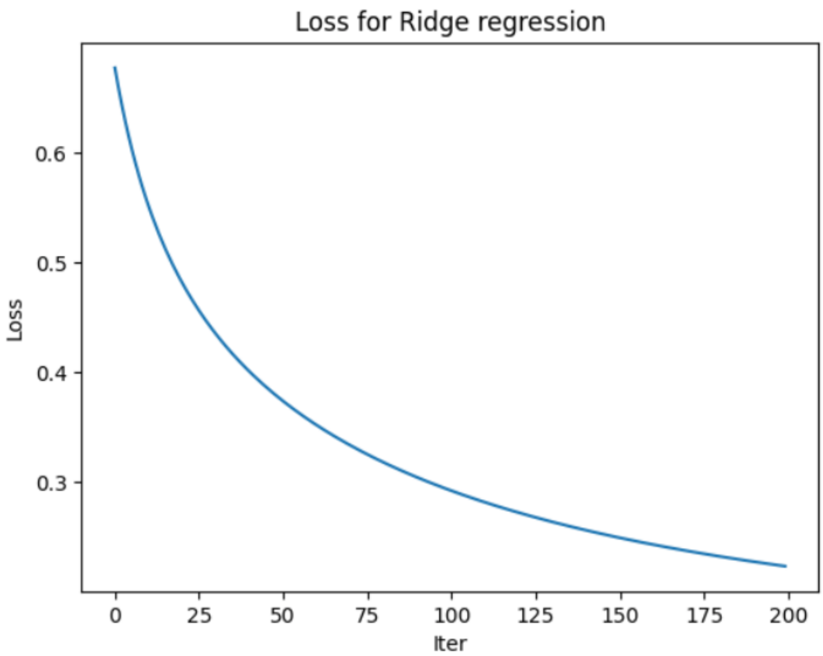
3.Supervised learning

Logistic regression

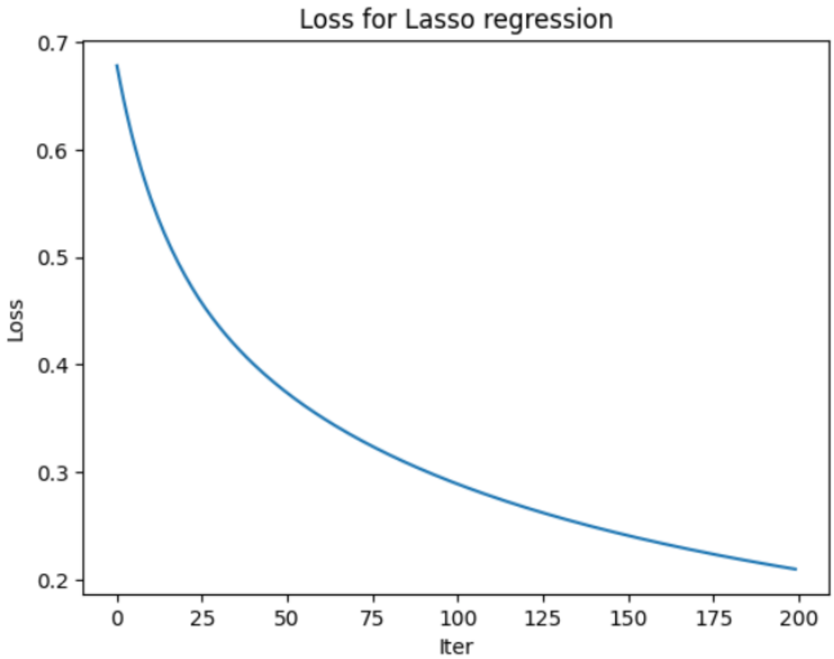
Using L1 and L2 regularizations I obtained the following results:



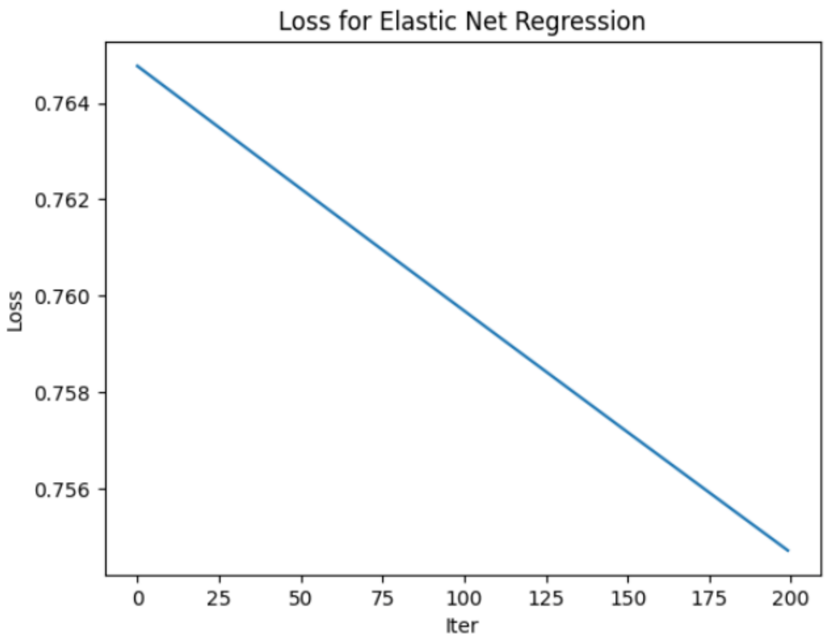
Ridge regression:



Lasso regression:



We can see that the loss function with L1 is the same I got with L2 but when I combined L1 and L2 regularization I obtained worse results for the loss function



Accuracy:

L2 regularization (Ridge):
Train set accuracy:97.10%
Test set accuracy:99.19%

L2 regularization (Lasso):
Train set accuracy:56.80%
Test set accuracy:52.15%

L2 and L1 regularization (Elastic Net):
Train set accuracy:56.80%
Test set accuracy:52.15%

We can conclude that the Ridge regularization is the one that produces. The best outcome in terms of correctness of the predictions compared to the ground truth labels.

Model assessment:

With L2 regularization (Ridge)

	precision	recall	f1-score	support
1	1.00	0.98	0.99	194
0	0.98	1.00	0.99	178
accuracy			0.99	372
macro avg	0.99	0.99	0.99	372
weighted avg	0.99	0.99	0.99	372

With L1 regularization (Lasso)

	precision	recall	f1-score	support
1	0.52	1.00	0.69	194
0	0.00	0.00	0.00	178
accuracy			0.52	372
macro avg	0.26	0.50	0.34	372
weighted avg	0.27	0.52	0.36	372

With L2 and L1 regularization (Elastic Net)

	precision	recall	f1-score	support
1	0.52	1.00	0.69	194
0	0.00	0.00	0.00	178
accuracy			0.52	372
macro avg	0.26	0.50	0.34	372
weighted avg	0.27	0.52	0.36	372

Decision tree

Using ID3 algorithm the accuracy I got is 0.954 and that means that decision tree is making predictions very precisely on the test set.

Naïve Bayes

With Naïve Bayes the accuracy is 0.5188 so we can say that only one label every two labels are predicted correctly.
Applying Gaussian Naïve Bayes, the results I obtained were lower (0.177). This can happen when the continuous variables are not distributed like Gaussian and the kernel density estimation made in the data pretreatment confirms that.

K-Nearest Neighbors

The accuracy achieved with KNN is 1.0 which indicates. That the KNN model predicts correctly all the instances of the class and is the best result obtained so far.