Algorithm

Counterfeit Banknotes

Data Analyst Course - Project 6

Michael Orange

Mission

Create an algorithm for detecting counterfeit banknotes

Algorithm based on banknotes characteristics:

- Length (in mm)
- Height measured in the left side (in mm)
- Height measured in the right side (in mm)
- Margin between superior side and the print (in mm)
- Margin between inferior side and the print (in mm)
- Diagonal (in mm)

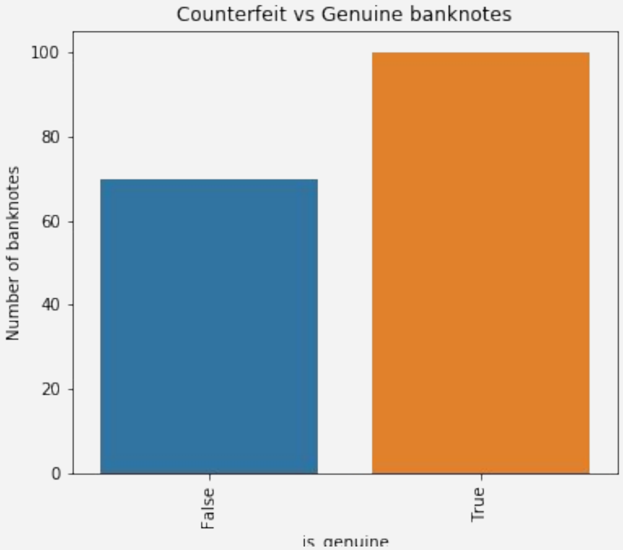
1. Data Exploration
2. Principal Component Analysis
3. Modelisation
4. API

Data Exploration

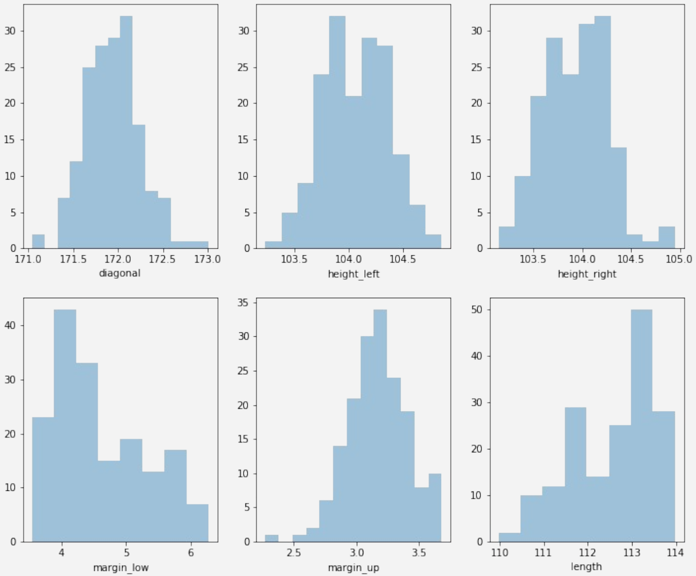
A dataset to train our model

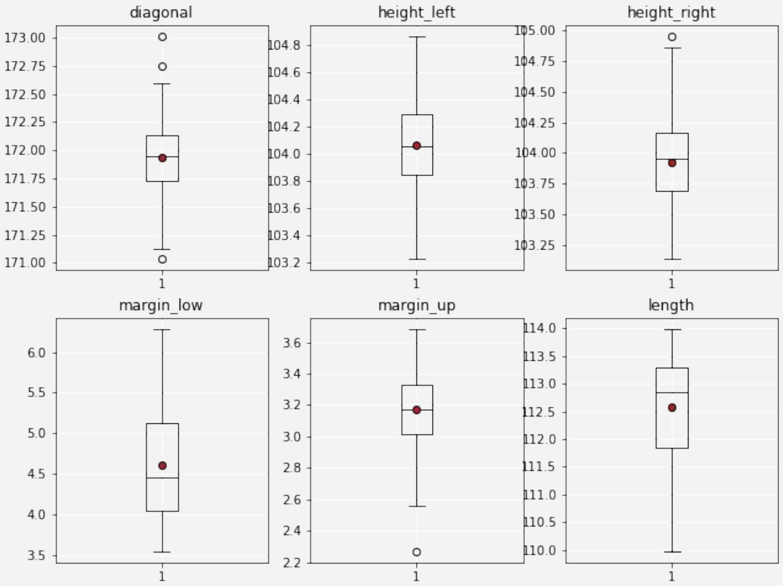
CSV file available to train our model with Genuine banknotes and Counterfeit (non Genuine) banknotes.

	diagonal	height_left	height_right	margin_low	margin_up	length
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	171.976100	103.951500	103.775900	4.143500	3.055500	113.207200
std	0.307981	0.296251	0.292406	0.314509	0.197726	0.380476
min	171.040000	103.230000	103.140000	3.540000	2.270000	111.760000
25%	171.790000	103.740000	103.557500	3.900000	2.937500	112.995000
50%	172.005000	103.915000	103.760000	4.080000	3.070000	113.210000
75%	172.162500	104.145000	103.972500	4.382500	3.192500	113.505000
max	172.750000	104.860000	104.950000	5.040000	3.530000	113.980000

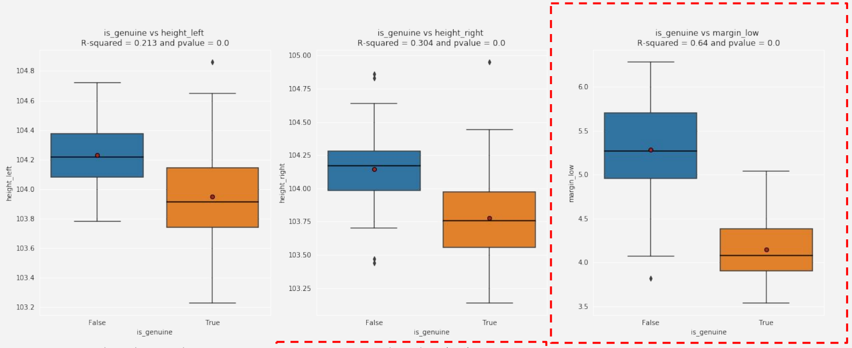


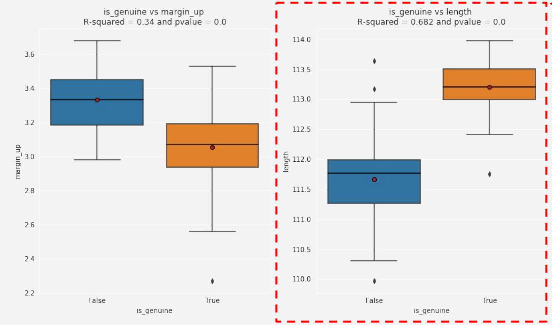
No significant outliers





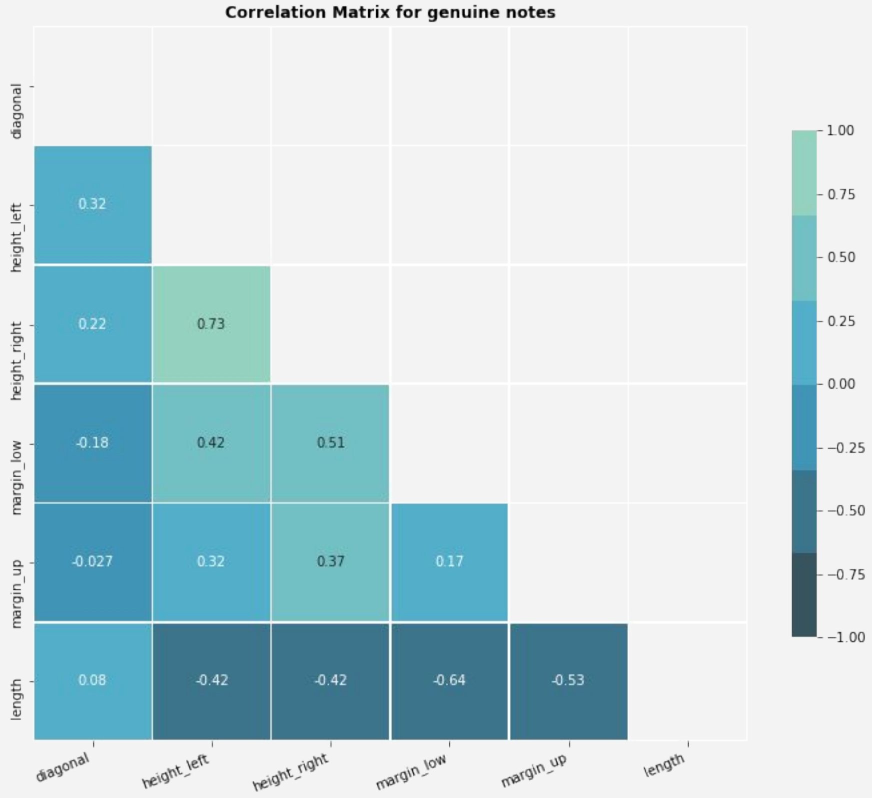
Significance of margin_low and length





At a significance level of 5%, the highest correlations with 'is_genuine' are observed for 'length' and 'margin_low'.

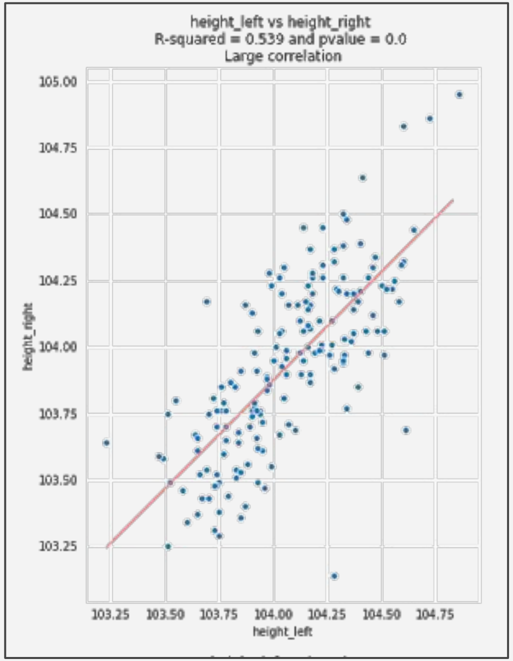
No numerical variables entirely correlated

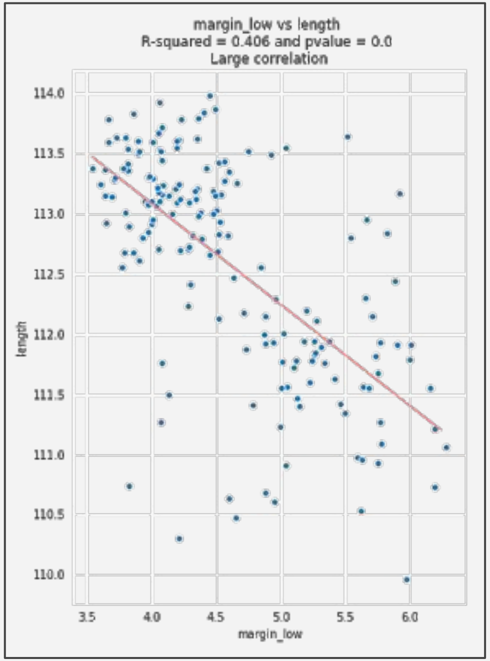


'height_right' and **'height_left'** have a significant correlation (0.73).

>> However no variables are redundant due to a very level of correlation.

Some variables are correlated linearly

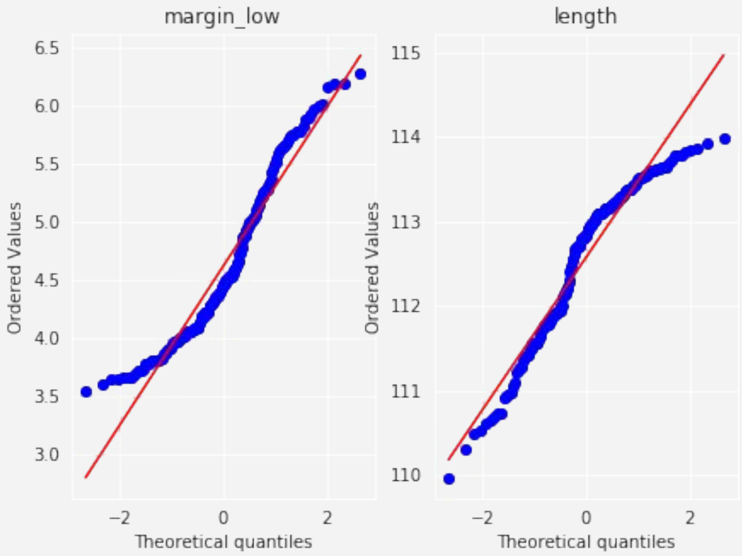




Gaussian-like distribution

Only 2 variables do not pass the Shapiro test at a significance level of 5%

However the normal distributions are still confirmed by the Henri lines for these 2 variables



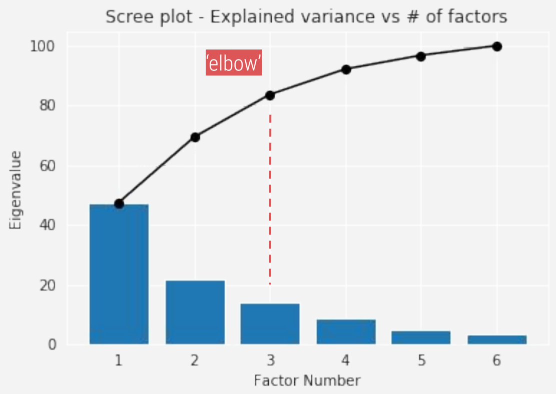
Principal Component Analysis

Preparation for PCA

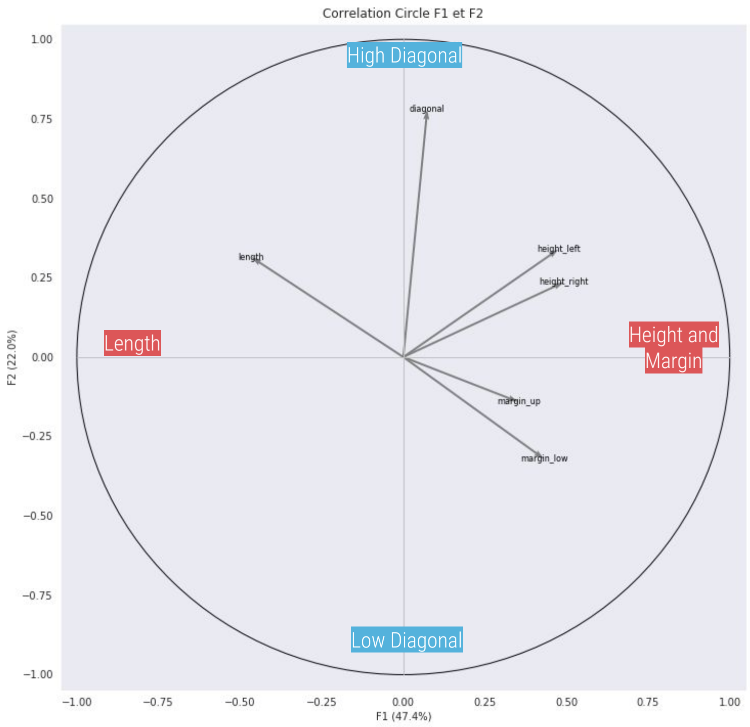
Mean centering - subtracting variable measured mean from each data value so that its empirical mean (average) is zero

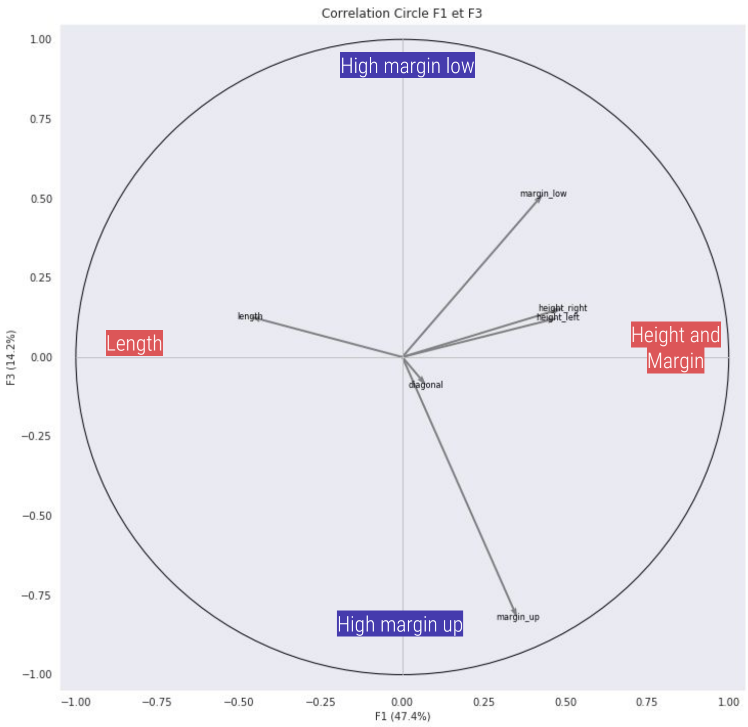
Variance standardization to 1 - after the mean centering, dividing 'centered' data by standard deviation.

3 components are selected
(capturing approx. 80% of the variance)

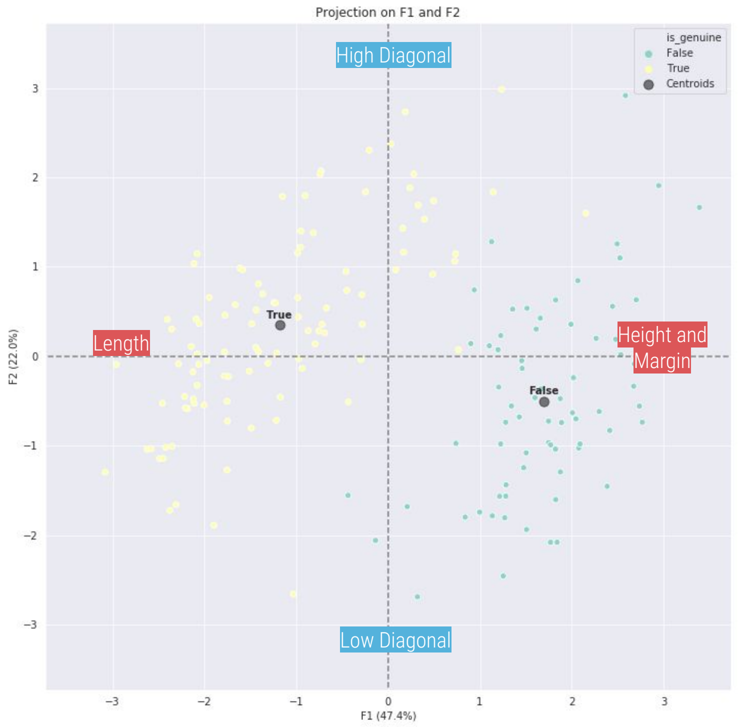


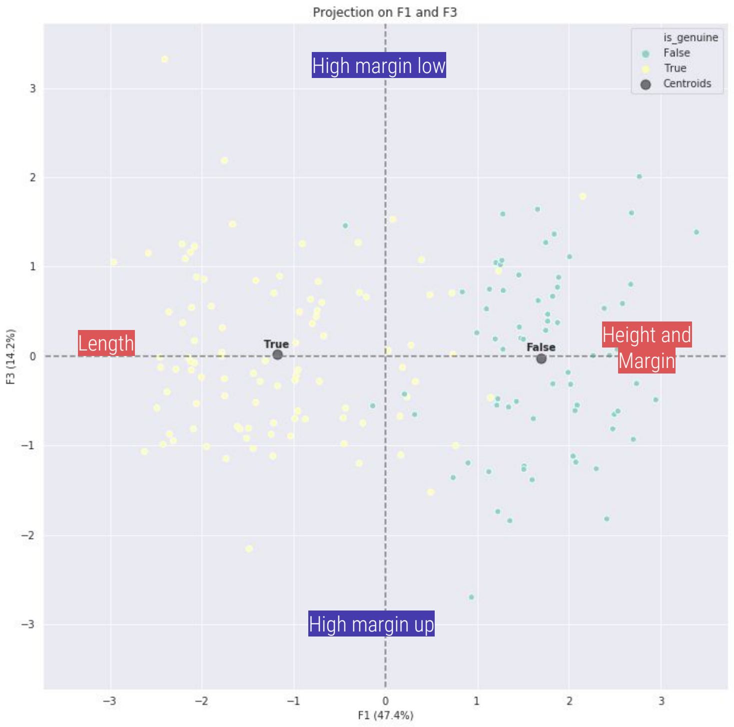
Correlation Circles



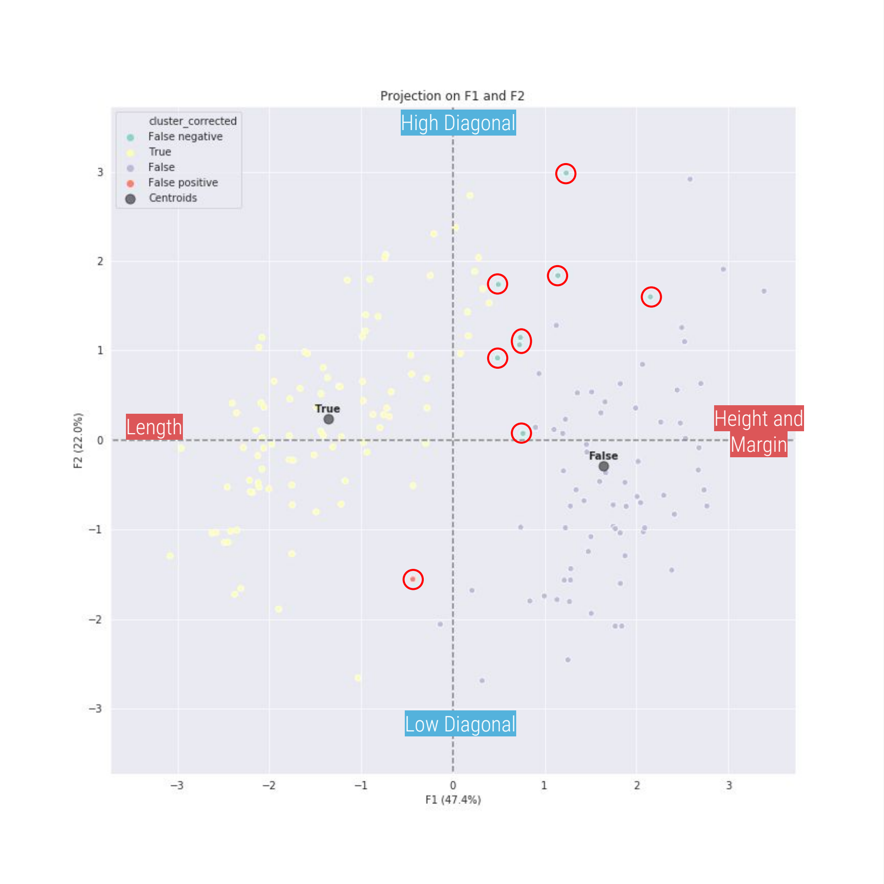


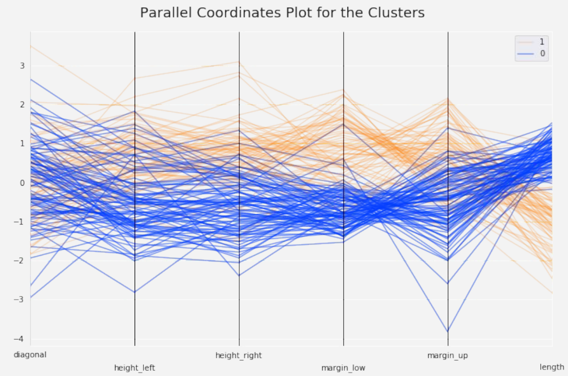
Projection on F1 and F2

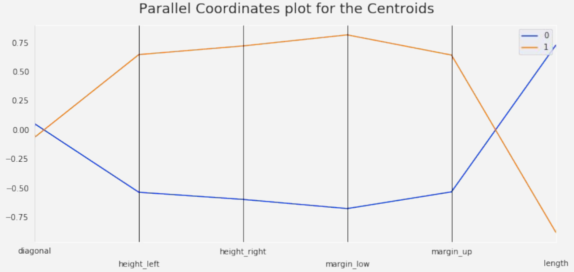




K-means Clustering







Modelisation

Logistic Regression

Methodology

Train, Test split : 80% to train the model, 20% to test the model

- **Statsmodel: Logit regression on scaled values**
- **Sklearn: LogisticRegression on scaled values**

- **Statsmodel: Logit regression on projected values**
- **Sklearn: LogisticRegression on projected values**

Cross-validation Kfold (cv = 5)

Selecting best model with scaled values or sklearn

Selection

Logit on Scaled values

```
                 Results: Logit
==================================================================
Model:            Logit         Pseudo R-squared: 0.954
Dependent Variable: is_genuine     AIC:           12.5605
Date:            2020-07-08 12:11 BIC:           18.3858
No. Observations:   136          Log-Likelihood:  -4.2802
Df Model:          1           LL-Null:        -92.139
Df Residuals:       134           LLR p-value:     4.1727e-40
Converged:         1.0000          Scale:          1.0000
No. Iterations:    14.0000
------------------------------------------------------------------
          Coef.  Std.Err.    z    P>|z|   [0.025    0.975]
------------------------------------------------------------------
margin_low  -9.0096   3.6243  -2.4859  0.0129  -16.1131  -1.9061
length       7.5277   3.0038   2.5061  0.0122    1.6404  13.4151
==================================================================
```

Prob = logistic(x) = 1 / (1 + $e^{(-x)}$)
with x = -9.0096 * margin_low + 7.5277 * length

Accuracy of the model: 97.8% **Selected**

Logit on ACP (F1, F2, F3)

```
                 Results: Logit
==================================================================
Model:            Logit         Pseudo R-squared: 0.848
Dependent Variable: is_genuine     AIC:           32.0556
Date:            2020-07-08 12:26 BIC:           37.8809
No. Observations:   136          Log-Likelihood:  -14.028
Df Model:          1           LL-Null:        -92.139
Df Residuals:       134           LLR p-value:     7.5678e-36
Converged:         1.0000          Scale:          1.0000
No. Iterations:    10.0000
------------------------------------------------------------------
     Coef.   Std.Err.    z      P>|z|    [0.025    0.975]
------------------------------------------------------------------
F1  -3.1539   0.6725   -4.6902   0.0000   -4.4719   -1.8360
F2   2.2678   0.6089    3.7247   0.0002    1.0745    3.4611
==================================================================
```

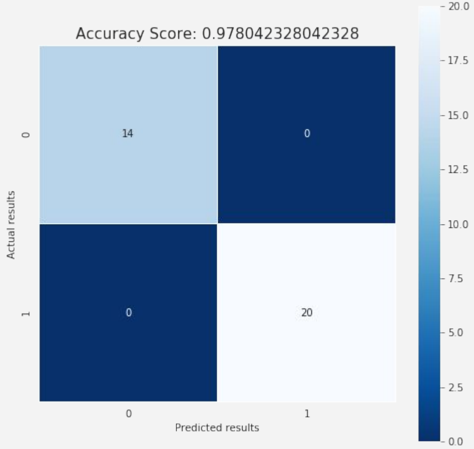
Prob = logistic(x) = 1 / (1 + $e^{(-x)}$)
with x = -3.1539 * F1 + 2.2678 * F2

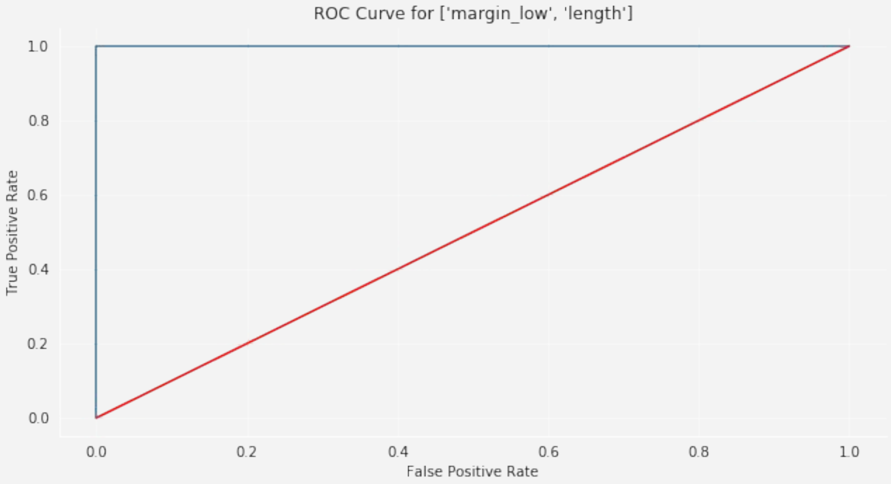
Accuracy of the model: 97.1% **Rejected**

Sklearn LogisticRegression

- Features: **length** and **margin low**
- **Cross-validation (kfold with 5 splits):**
  [0.96428571 0.92592593 1. 1. 1.]
  Accuracy of the model: **97.8 %**

With a AUC of 1, the model is excellent





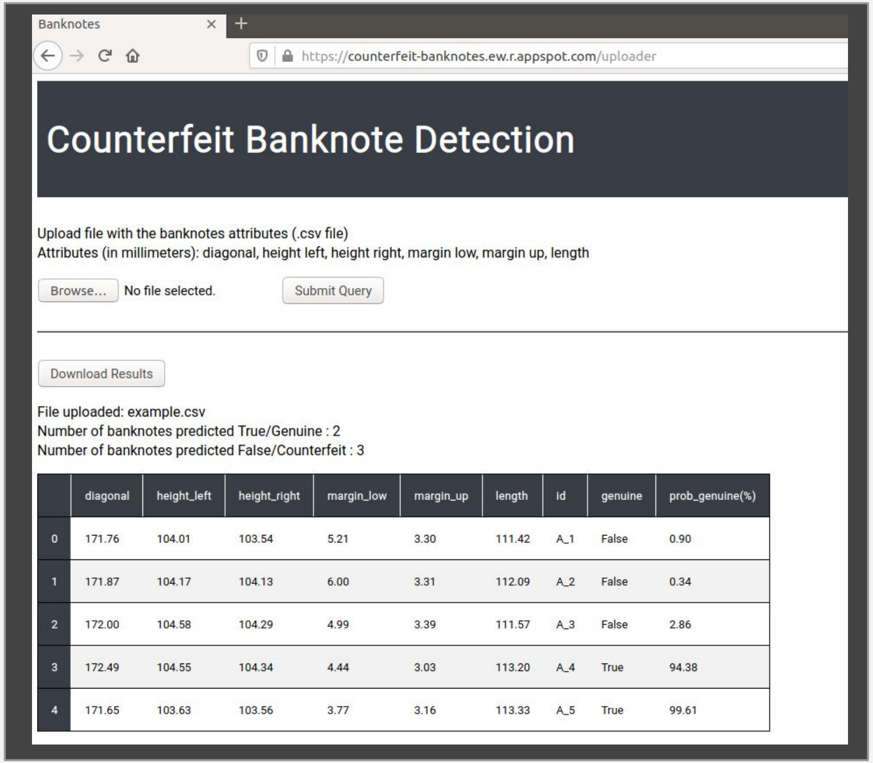
API

API - Google Cloud Platform

Python API hosted with Google App Engine

https://counterfeit-banknotes.ew.r.appspot.com

- Upload SKLearn LogisticRegression and StandardScaler trained models
- Request CSV file with ‘margin_low’ and ‘length’ attributes (in mm)
- Scale values of the CSV file (mean centering and variance standardization)
- Perform a SKLearn LogisticRegression on scaled values.
- Return a table with prediction and probability of the prediction.



	diagonal	height_left	height_right	margin_low	margin_up	length	id	genuine	prob_genuine(%)
0	171.76	104.01	103.54	5.21	3.30	111.42	A_1	False	0.90
1	171.87	104.17	104.13	6.00	3.31	112.09	A_2	False	0.34
2	172.00	104.58	104.29	4.99	3.39	111.57	A_3	False	2.86
3	172.49	104.55	104.34	4.44	3.03	113.20	A_4	True	94.38
4	171.65	103.63	103.56	3.77	3.16	113.33	A_5	True	99.61

THANK YOU!

Do you have any questions?

Code available on
https://github.com/Michael-Orange/algorithm_banknotes

CREDITS: This presentation template was inspired from **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Pixabay**