Python for data analysis : Project Avila

Stéphane LALISSE
Jules de SCHUTTELAERE
DIA2

Identify copyist of “Avila Bible” with Supervised Learning

Introduction:

- In this project we need to identify the different copyists of the “Avila Bible” by using different models. The Avila data set has been extracted from 800 images of the "Avila Bible", a giant Latin copy of the whole Bible.
- Some analysis of the manuscript has individuated the presence of 12 copyists. Each pattern contains 10 features, they are intercolumnar distance, upper margin, lower margin, exploitation, row number, modular ratio, interlinear spacing, weight, peak number, modular ratio/interlinear spacing.
- The prediction task consists in associating each pattern to one of the 12 copyists (labeled as: A, B, C, D, E, F, G, H, I, W, X, Y).
- We did some data visualization to understand the dataset and then use machine learning to solve the problem with scikit-learn and modelisation.

Data set

- Patterns extracted from pages of the "**Avila bible**", containing 10 features and corresponding to each copyist.
- The task is to predict the copyist label associated with each pattern.
- The training set *avila-tr* contains **10430** samples, and the test set *avila-ts* contains **10437** samples.
- The data has been **normalized**, by using the Z-normalization method.

- Class distribution:

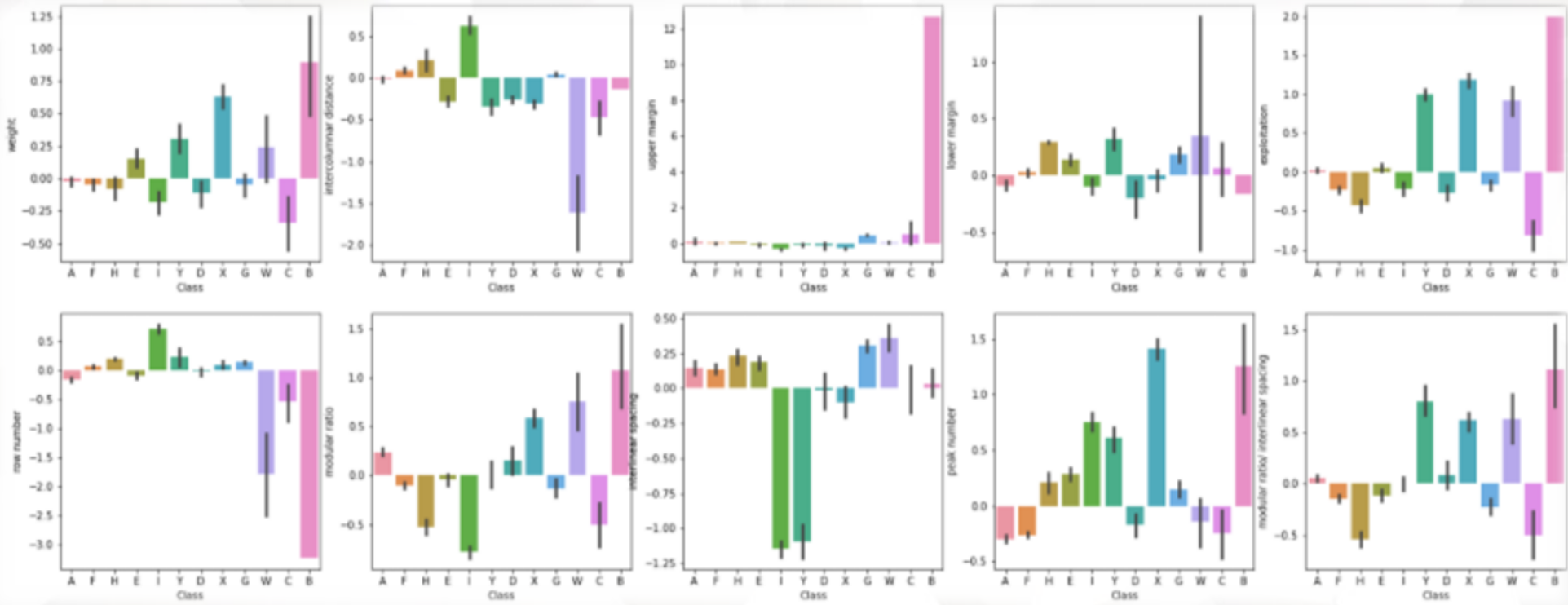
A	B	C	D	E	F	G	H	I	W	X	Y
4286	5	103	352	1095	1961	446	519	831	44	522	266

Features

	intercolumnar distance	upper margin	lower margin	exploitation	row number	modular ratio	interlinear spacing	weight	peak number	modular ratio/ interlinear spacing	Class
0	0.266074	-0.165620	0.320980	0.483299	0.172340	0.273364	0.371178	0.929823	0.251173	0.159345	A
1	0.130292	0.870736	-3.210528	0.062493	0.261718	1.436060	1.465940	0.636203	0.282354	0.515587	A
2	-0.116585	0.069915	0.068476	-0.783147	0.261718	0.439463	-0.081827	-0.888236	-0.123005	0.582939	A
3	0.031541	0.297600	-3.210528	-0.583590	-0.721442	-0.307984	0.710932	1.051693	0.594169	-0.533994	A
4	0.229043	0.807926	-0.052442	0.082634	0.261718	0.148790	0.635431	0.051062	0.032902	-0.086652	F

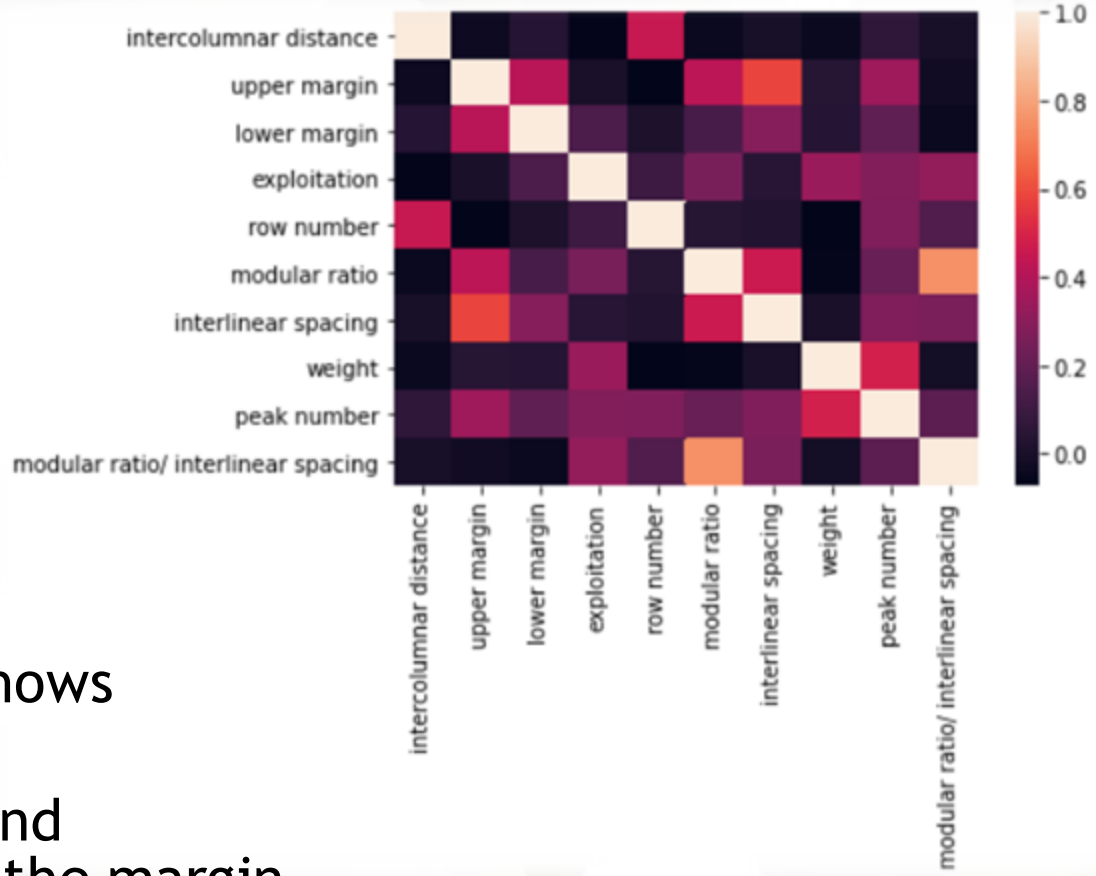
- *Intercolumnar distance*: distance between columns
- *Upper margin*: upside value
- *Lower margin*: downside value
- *Exploitation*: column exploitation coefficient (the number of black pixels/the total number of pixels in the column)
- *Row number*: number of rows in the column
- *Modular ratio*: estimates the dimension of handwriting characters
- *Interlinear spacing*: the distance in pixels between two rows
- *Weight*: a measure of how much a row is filled with ink
- *Modular ratio / Interlinear spacing*
- *Peak number*: the number of peaks in the horizontal projection histogram of rows

Features distribution



There is some disparities between Classes features distribution. There is differences between each copyist's styles. This confirms the method chosen for this analysis based on supervised learning. Using features correlation to predict each sample's class.

Features Correlation Matrix



- The correlation matrix shows us the links between the variables
- We can see the link between them easier
- Correlation between the *Modular ratio* and *Modular ratio / Interlinear spacing* this shows the writing style of the copyist
- Correlation between the *Interlinear spacing* and the *Upper margin*, the more space there is in the margin, the tighter his lines will be between them when he writes.
- Correlation between *Weight* and *Peak number*, the more ink used in a line, the tighter his words are between them and therefore more likely to have peaks.
- Correlation between *Interlinear spacing* and *modular ratio*, logical because we use them in the variable *Modular ratio / Interlinear spacing*.

Methodologies

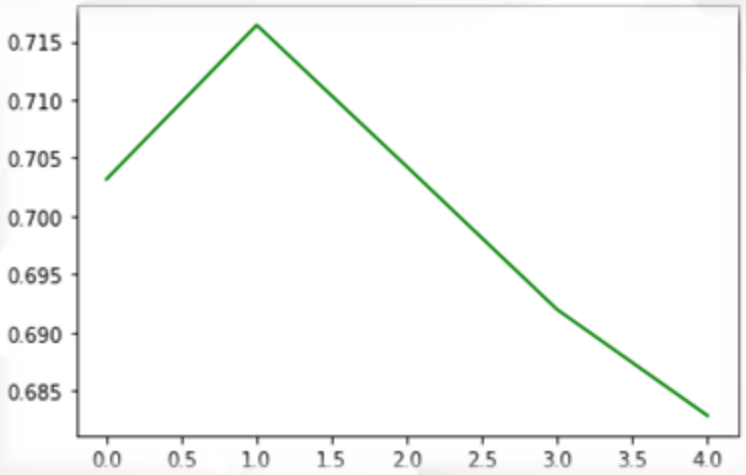
- **L**inear **D**iscriminant **A**nalysis
  - *Accuracy score*
  - *Cross-validation*
- **D**ecision **T**ree
  - *Accuracy score*
  - *Cross-validation*
- **R**andom **F**orest
  - *Accuracy score*
  - *Cross-validation*
- **G**rid Search: Random Forest
  - *Cross-validation*

! *We decided for this analysis to try out 3 classification models to find out the one maximize the prediction accuracy.*

Linear Discriminant Analysis

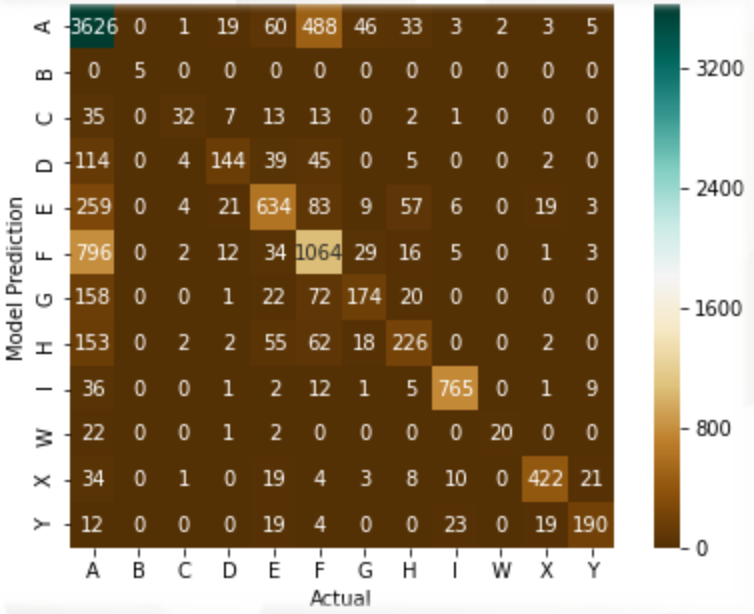
- A classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes rules.

Cross-validation results



We obtain an average prediction accuracy of 0.7 over the test set. I is not enough to considerate it as a good result.

Confusion matrix

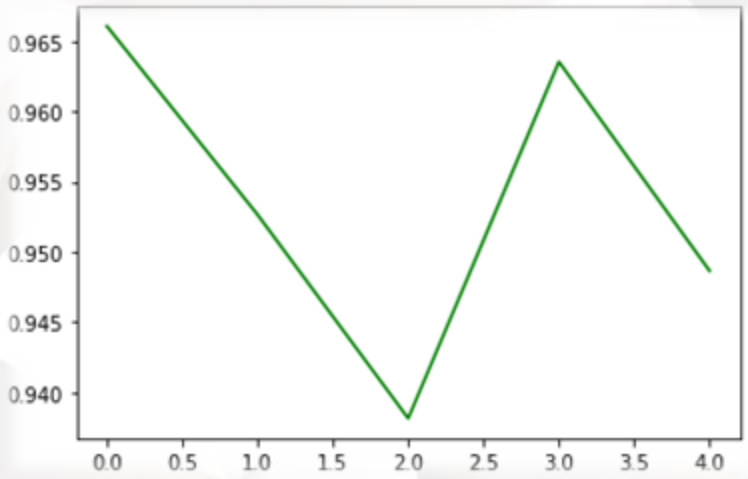



```
Out[25]: array([0.70315488, 0.71640363, 0.7042186 , 0.69193858, 0.6828215 ])
```

Decision Tree

- Decision Tree is a non-parametric supervised learning method used for classification and regression.

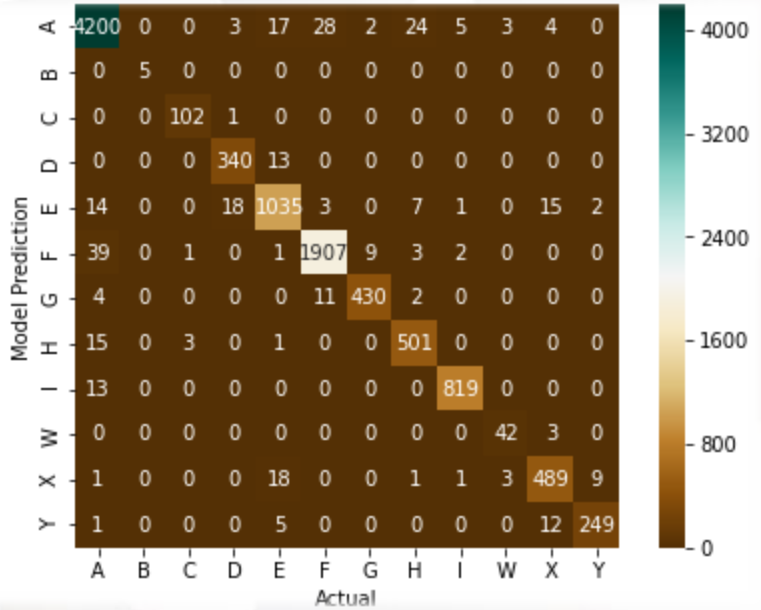
Cross-validation results




```
Out[13]: array([0.96606119, 0.95265423, 0.93815916, 0.96353167, 0.94865643])
```

We obtain an average prediction accuracy of 0.95 over the test set ! It is the best result so far.

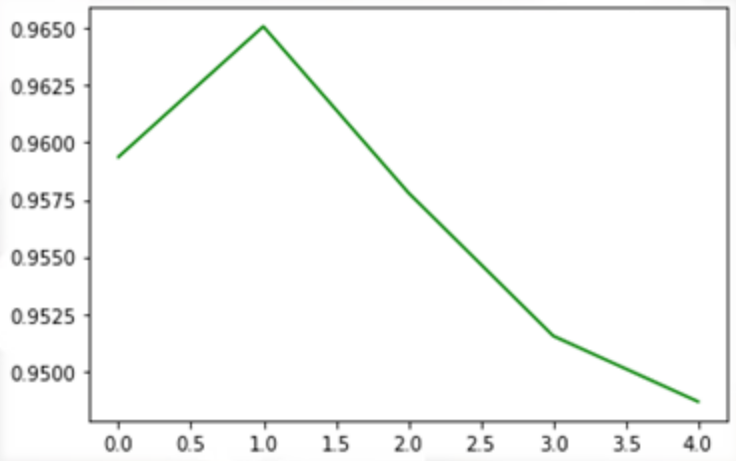
Confusion matrix



Random Forest

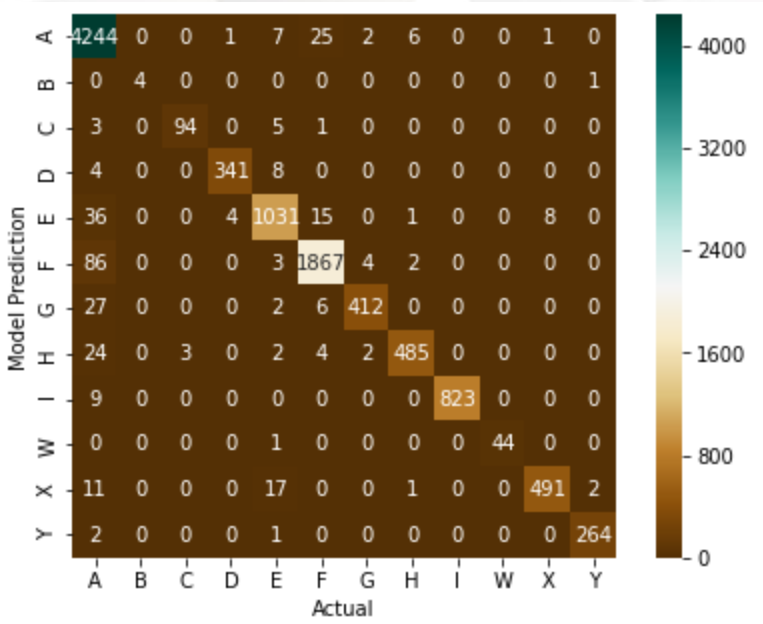
- A random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Cross-validation results



We obtain an average prediction accuracy of 0.95 over the test set ! It is the same results as the Decision tree method.

Confusion matrix

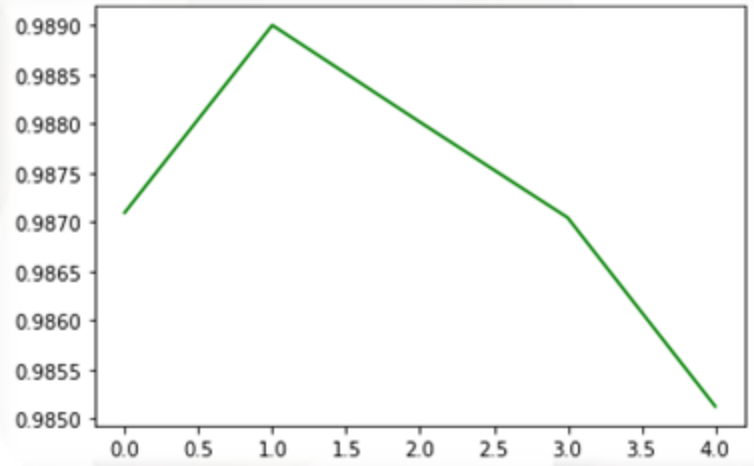



```
Out[16]: array([0.95936902, 0.96508847, 0.957814  , 0.95153551, 0.94865643])
```

Grid Search: Random Forest

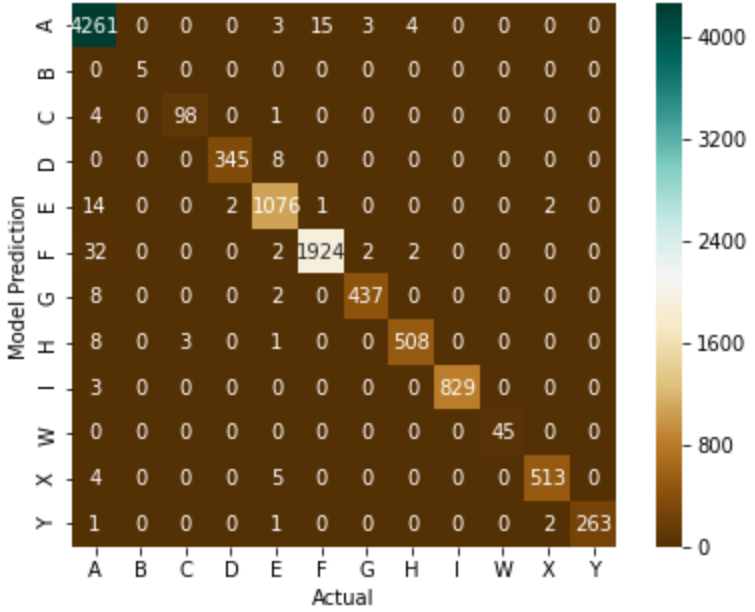
- Grid Search implements a “fit” and a “score” method. The parameters of the estimator used to apply these methods are optimized by the grid-search over a parameter grid.
- We obtain an average prediction accuracy of 0.98. It is most accurate prediction method.

Cross-validation results

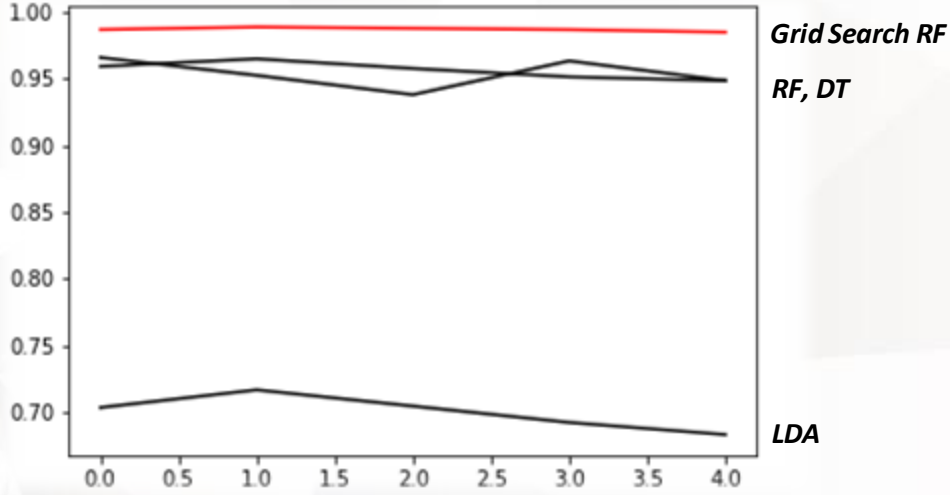



```
Out[23]: array([0.98709369, 0.98900048, 0.98801534,
       0.98704415, 0.98512476])
```

Confusion matrix



Results comparison



Conclusion

- The data visualized allowed us to better understand the data and the links between the variables, which permitted to use supervised learning to find out which copyist matches the pattern.
- From the 4 different methods we chose the Random Forest algorithm optimized with grid search. The sorting parameters were tuned to get an average accuracy of 0.98 out of an original accuracy of 0.95 with the default ones.
- A way for improving accuracy could be to use grid search applied to the decision tree method or other methods unused in this work.