ESILV - Python for data analysis - project 2022

Thomas BAUDU and Paul CANAL

Introduction

This year our project in Python for data analysis was to analyse a dataset and create a model in Machine Learning.

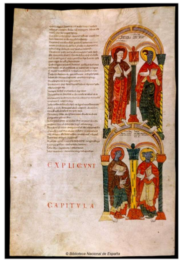
To analyse the dataset, we used several graphs like barplots, boxplots…

For the part in machine learning, we tried 4 different models that we will present you later.

To present our finished work we made an API using Flask, you can find more info on how to use it on our GitHub

Summary

Introduction



The Avila bible

Analyse

Model in Machine Learning

Flask API

Conclusion

The Avila bible

Italian Romanesque manuscript whose text and images were completed after his arrival in the Cathedral of Avila at the end of the 12th century.

The Avila Bible is a copy of the Latin Vulgate which, due to its large dimensions -842 pages of 580 x 390 mm in parchment and about 15Kg in weight- is considered the best exponent in the group of “Atlantic Bibles”large manuscripts for the liturgical use of monastic communities, created between the end of the 11th century and the end of the 12th century, in the midst of a Gregorian reform and in the geometric style very different from other European models, which was already common in Roman mural painting in the 12th century, and which is characterized in the miniature by the initials that start each Biblical book, with geometric frets and a very sophisticated decoration, which we also find in other Italian codices of that period, especially at the desks in Rome and Milan.

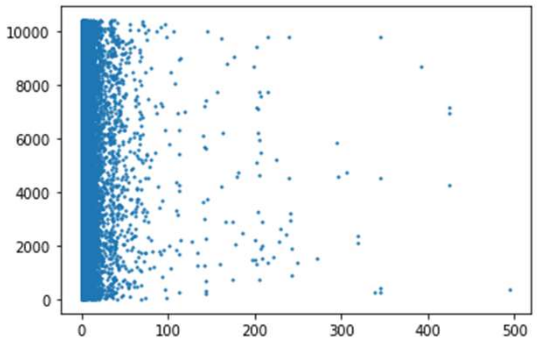
Although there is no record of its initial dating or origin, there is no doubt that its origin would be Italian from the middle of the twelfth century and that it would reach Avila around 1175 where the work was continued, in a style very different from that of the Italian part.

Source : https://www.turismo-prerromanico.com/en/manuscritos/biblia-de-vila-20131101184428/

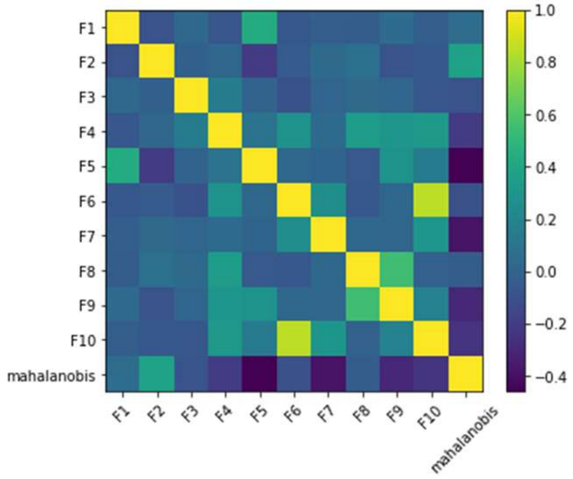
Analysing the dataset

The first graph represents the Mahalanobis Distance of the DataFrame. There is 10428 point on this graph. But only a hundred have a Mahalanobis distance > than 100. So, we decided to delete all the points with a distance > than 260 to delete the inconsistent points.

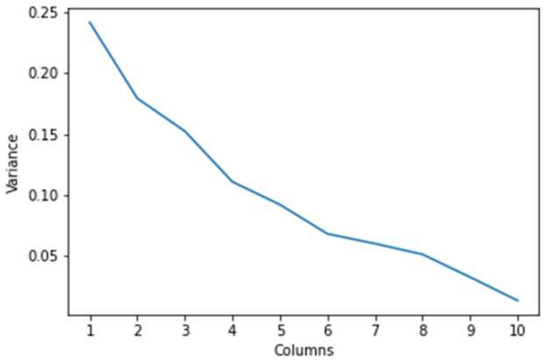
The second graph is a Correlation Matrix in the dataset. We can observe which columns are single and who help us to guess on the test set. The last one have the same goal, this is the PCA Method.



Mahalanobis Distance



Correlation Matrix

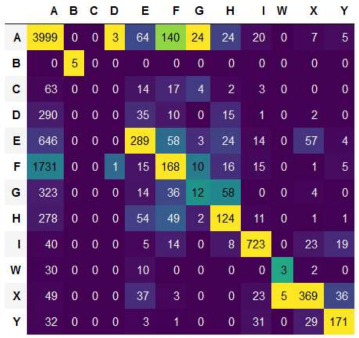


PCA representation

Model in Machine Learning

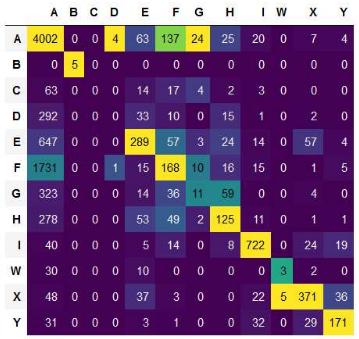
For our model in Machine Learning, we tried 5 different models.

The five graphs below are the confusion matrix of each model. We easily see that the two logistic regression's confusion matrix are not as good as the others. The best ones are from the Random Forest models, and the one without the PCA gives the best results.



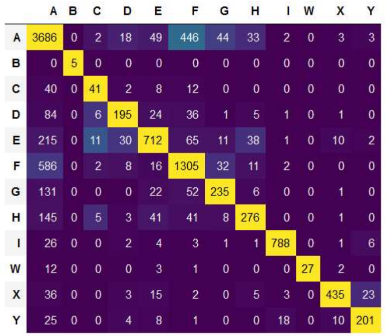
	A	B	C	D	E	F	G	H	I	W	X	Y
A	3999	0	0	3	64	140	24	24	20	0	7	5
B	0	5	0	0	0	0	0	0	0	0	0	0
C	63	0	0	0	14	17	4	2	3	0	0	0
D	290	0	0	0	35	10	0	15	1	0	2	0
E	646	0	0	0	289	58	3	24	14	0	57	4
F	1731	0	0	1	15	168	10	16	15	0	1	5
G	323	0	0	0	14	36	12	58	0	0	4	0
H	278	0	0	0	54	49	2	124	11	0	1	1
I	40	0	0	0	5	14	0	8	723	0	23	19
W	30	0	0	0	10	0	0	0	0	3	2	0
X	49	0	0	0	37	3	0	0	23	5	369	36
Y	32	0	0	0	3	1	0	0	31	0	29	171

Logistic regression



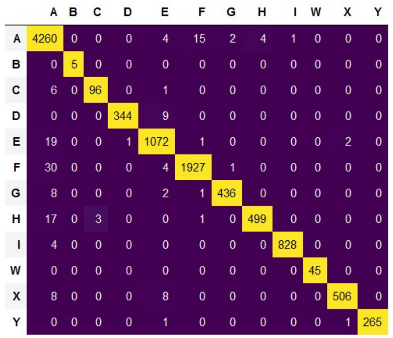
	A	B	C	D	E	F	G	H	I	W	X	Y
A	4002	0	0	4	63	137	24	25	20	0	7	4
B	0	5	0	0	0	0	0	0	0	0	0	0
C	63	0	0	0	14	17	4	2	3	0	0	0
D	292	0	0	0	33	10	0	15	1	0	2	0
E	647	0	0	0	289	57	3	24	14	0	57	4
F	1731	0	0	1	15	168	10	16	15	0	1	5
G	323	0	0	0	14	36	11	59	0	0	4	0
H	278	0	0	0	53	49	2	125	11	0	1	1
I	40	0	0	0	5	14	0	8	722	0	24	19
W	30	0	0	0	10	0	0	0	0	3	2	0
X	48	0	0	0	37	3	0	0	22	5	371	36
Y	31	0	0	0	3	1	0	0	32	0	29	171

Logistic regression PCA



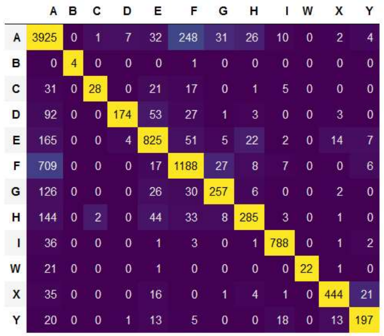
	A	B	C	D	E	F	G	H	I	W	X	Y
A	3686	0	2	18	49	446	44	33	2	0	3	3
B	0	5	0	0	0	0	0	0	0	0	0	0
C	40	0	41	2	8	12	0	0	0	0	0	0
D	84	0	6	195	24	36	1	5	1	0	1	0
E	215	0	11	30	712	65	11	38	1	0	10	2
F	586	0	2	8	16	1305	32	11	2	0	0	0
G	131	0	0	0	22	52	235	6	0	0	1	0
H	145	0	5	3	41	41	8	276	0	0	1	0
I	26	0	0	2	4	3	1	1	788	0	1	6
W	12	0	0	0	3	1	0	0	0	27	2	0
X	36	0	0	3	15	2	0	5	3	0	435	23
Y	25	0	0	4	8	1	0	0	18	0	10	201

K-Nearest Neighbors



	A	B	C	D	E	F	G	H	I	W	X	Y
A	4260	0	0	0	4	15	2	4	1	0	0	0
B	0	5	0	0	0	0	0	0	0	0	0	0
C	6	0	96	0	1	0	0	0	0	0	0	0
D	0	0	0	344	9	0	0	0	0	0	0	0
E	19	0	0	1	1072	1	0	0	0	0	2	0
F	30	0	0	0	4	1927	1	0	0	0	0	0
G	8	0	0	0	2	1	436	0	0	0	0	0
H	17	0	3	0	0	1	0	499	0	0	0	0
I	4	0	0	0	0	0	0	0	828	0	0	0
W	0	0	0	0	0	0	0	0	0	45	0	0
X	8	0	0	0	8	0	0	0	0	0	506	0
Y	0	0	0	0	1	0	0	0	0	0	1	265

Random Forest

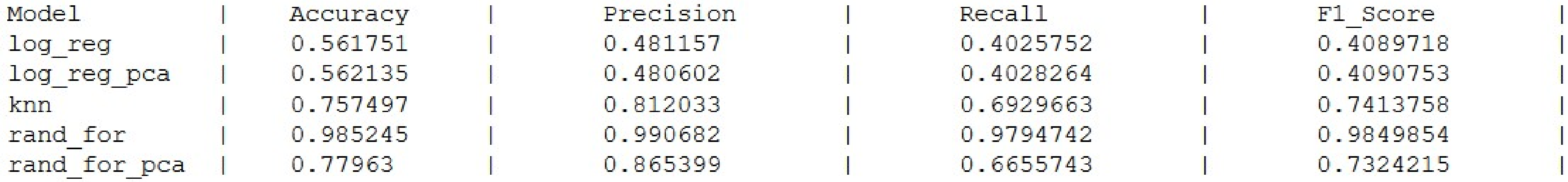


	A	B	C	D	E	F	G	H	I	W	X	Y
A	3925	0	1	7	32	248	31	26	10	0	2	4
B	0	4	0	0	0	1	0	0	0	0	0	0
C	31	0	28	0	21	17	0	1	5	0	0	0
D	92	0	0	174	53	27	1	3	0	0	3	0
E	165	0	0	4	825	51	5	22	2	0	14	7
F	709	0	0	0	17	1188	27	8	7	0	0	6
G	126	0	0	0	26	30	257	6	0	0	2	0
H	144	0	2	0	44	33	8	285	3	0	1	0
I	36	0	0	0	1	3	0	1	788	0	1	2
W	21	0	0	0	1	0	0	0	0	22	1	0
X	35	0	0	0	16	0	1	4	1	0	444	21
Y	20	0	0	1	13	5	0	0	18	0	13	197

Random Forest PCA

Model in Machine Learning

Finally, here are some important rates for each model :



```
Model          |    Accuracy     |        Precision        |        Recall          |        F1_Score        |
log_reg        |    0.561751     |        0.481157         |        0.4025752       |        0.4089718       |
log_reg_pca    |    0.562135     |        0.480602         |        0.4028264       |        0.4090753       |
knn            |    0.757497     |        0.812033         |        0.6929663       |        0.7413758       |
rand_for       |    0.985245     |        0.990682         |        0.9794742       |        0.9849854       |
rand_for_pca   |    0.77963      |        0.865399         |        0.6655743       |        0.7324215       |
```

We clearly see that the Random Forest model (without the PCA) gives the best results for every rate

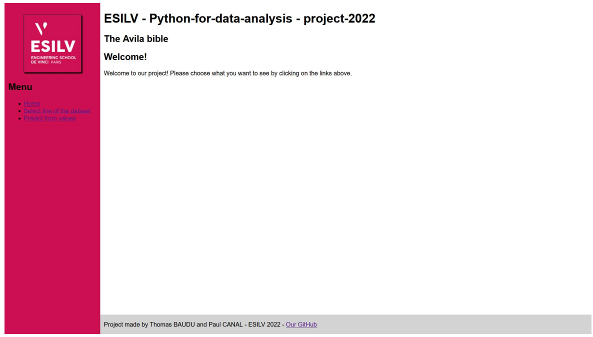
PCA seems to not have a lot of influence for the logistic regression model, however it is the case for the Random Forest model and we did not except the model without the PCA to be better.

Flask API

To represent our model, we made an API using Flask. It consists of a Python application and some HTML pages, along with CSS files.

In the Python application, we define functions to be called when we access a web page.
The first page is index.html, where we just present the project.
On the /avila_bible path (show_bible.html), we made a function to show the specific row of the dataset the user asked to see.
On the /predictions path (predict.html), we made a function to use the model we made: the user enters specific values and we predict which copyist would have had those values.

Each HTML page is an addition to the base.html file, to keep some coherence between all pages.



ESILV - Python-for-data-analysis - project-2022

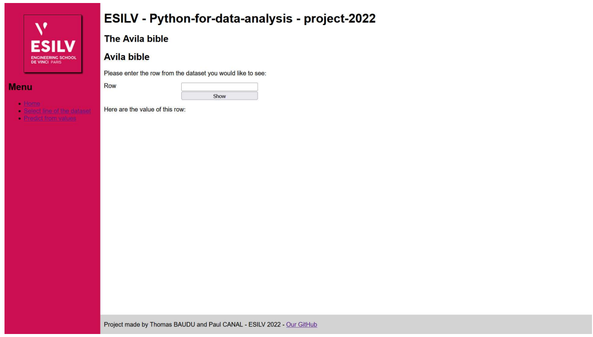
The Avila bible

Welcome!

Welcome to our project! Please choose what you want to see by clicking on the links above.

Project made by Thomas BAUDU and Paul CANAL - ESILV 2022 - Our GitHub

index.html



ESILV - Python-for-data-analysis - project-2022

The Avila bible

Avila bible

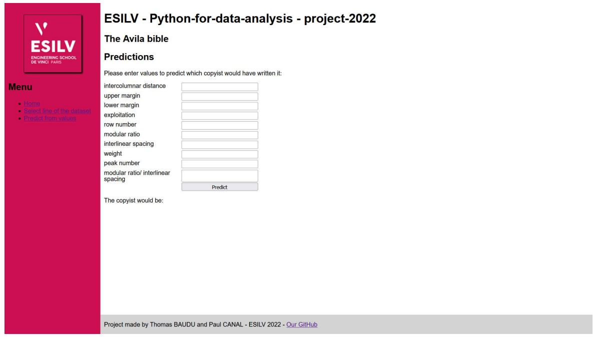
Please enter the row from the dataset you would like to see:

Row

Here are the value of this row:

Project made by Thomas BAUDU and Paul CANAL - ESILV 2022 - Our GitHub

show_bible.html



ESILV - Python-for-data-analysis - project-2022

The Avila bible

Predictions

Please enter values to predict which copyist would have written it:

intercolumnar distance
upper margin
lower margin
exploitation
row number
modular ratio
interlinear spacing
weight
peak number
modular ratio/ interlinear spacing

The copyist would be:

Project made by Thomas BAUDU and Paul CANAL - ESILV 2022 - Our GitHub

predict.html

Conclusion

The dataset we choose was interesting to analyse and also original compared to the other datasets (like detecting spam or drug consumption).

With around 0.985 of accuracy, we have a really good model. This result was achieved quite easily.

Flask was new to both of us so it was a new thing to learn, we also never used API so it was a good thing to start. It is also not too complex and we can made a lot of things with it.