Project Report

Python for Data Analysis

Alix Petitcol – Marine Sublet ESILV Master 1 - S1 2021-2022

TABLE OF CONTENTS

I. Objectives of the python project

II. Presentation of the selected dataset and its problematic

III. Presentation of the study

IV. Predictions

V. Conclusion

I. Objective of the python project

From a set of data, carry out a complete study with visualization and machine learning algorithms in order to explain the different links existing between the variables of the dataset.

Perform your visualization study on a Jupyter notebook and offer a Flask API to visualize and create one of the best prediction models you will find, where a user can choose the parameters suitable for the model.

II. Presentation of the selected dataset and its problematic

For this project we had the choice between two sets of data, we have chosen the Facebook Comment Volume Dataset.

Due to the frequency of use of social networks today, it is interesting to study the behavior of consumers concerning these services. This work aims to study and model user activity from Facebook.

The main purpose here is to estimate the number of comments that a message should receive in the hours following its post. The analysis will be done first by studying through different graphs the user behaviors and trends that stand out the most. Secondly, we will implement some prediction algorithms using regression techniques.

Instances in this dataset contain 30 features directly extracted from facebook posts, and 24 derived features that are aggregated by page, by calculating min, max, average, median and standard deviation of essential features.

The dataset contains 5 variants of data, however for our prediction analysis we have used only one of its dataset due to its sufficient size.

III. Presentation of the study

To understand which parameters influenced the most the fluctuation of the target variable (number of comments under a post), we have chosen to make several graphics beginning with a visualization of the correlation between every variable available in the dataset.

Then we have focused our study using the most relevant features (such as the page category and the day of the publication during the week), by associating them with other features with some “pandas.dataframe” formulas such as "group by“.

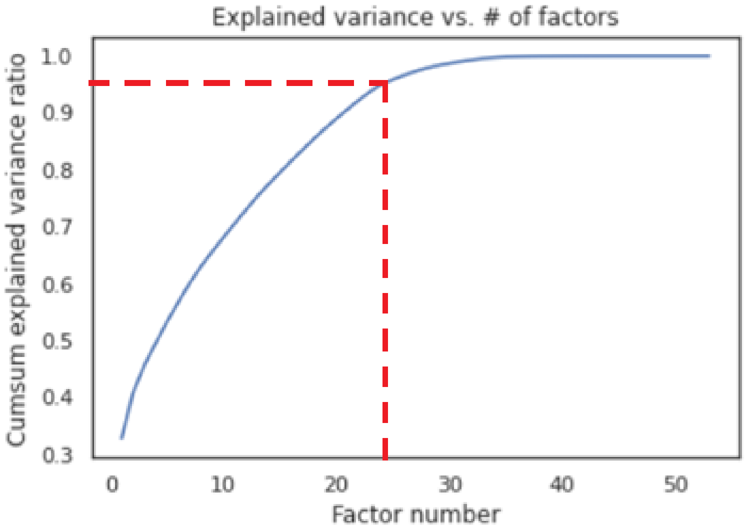
We did not need to create further variables in our dataset as they were already numerous, therefore we only changed the disposition of a few columns. For instance, for the convenience of graphic modelization, we regrouped all the weekday columns (explained by 0 or 1 according to the day) in a single column where the day appeared in form of its noun.

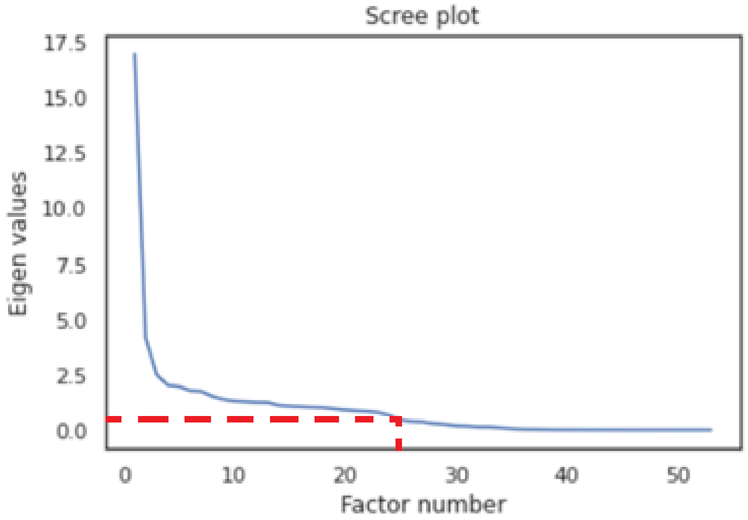
We have tried putting in interactive graphics to extend our knowledge about python visualization libraries. So, you will find graphics made with Seaborn, Matplotlib, but also Pygal or Plotly.

IV. Predictions

To make our predictions, we first started by splitting our dataset into a test and a train dataset, with a test size of 0,33. Then we scalled our data in order to get better predictions.

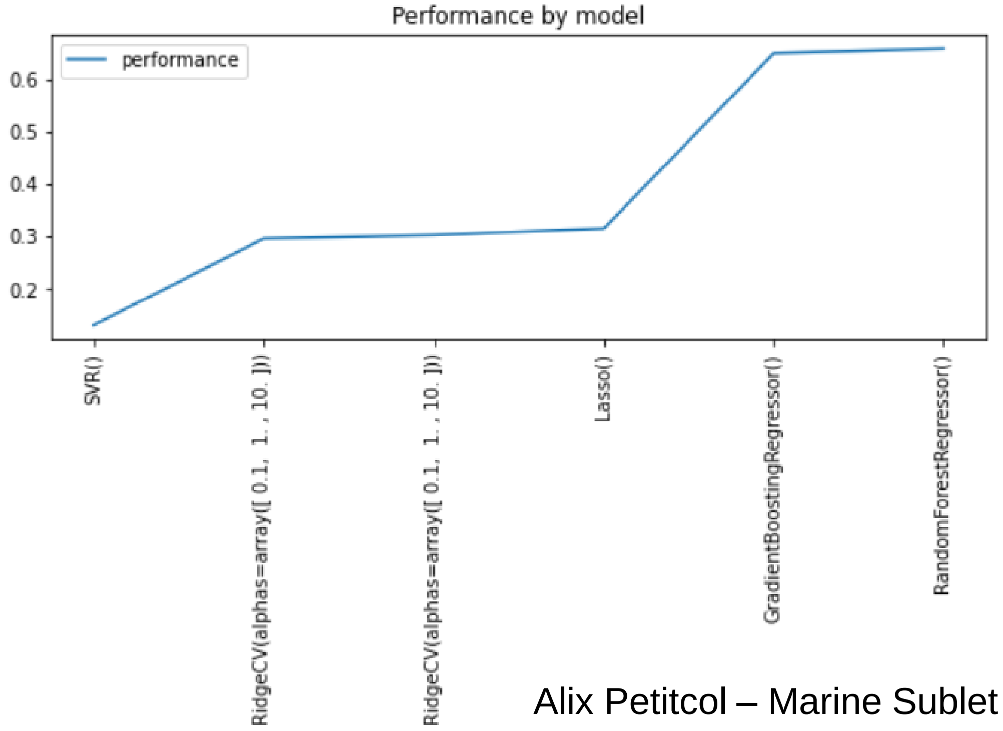
We also tried to do a PCA (Principal Component Analysis) in order to reduce the dimensionality of the datasets, increase its interpretability while minimizing the information loss.
We found out that with the PCA, the first 25 components explained 95% of the variance. We would therefore only need those 25 components, instead of our 54 features, which would make our predictions much faster. Unfortunatly, we obtained better results on the models without the PCA so we decided not to keep it.



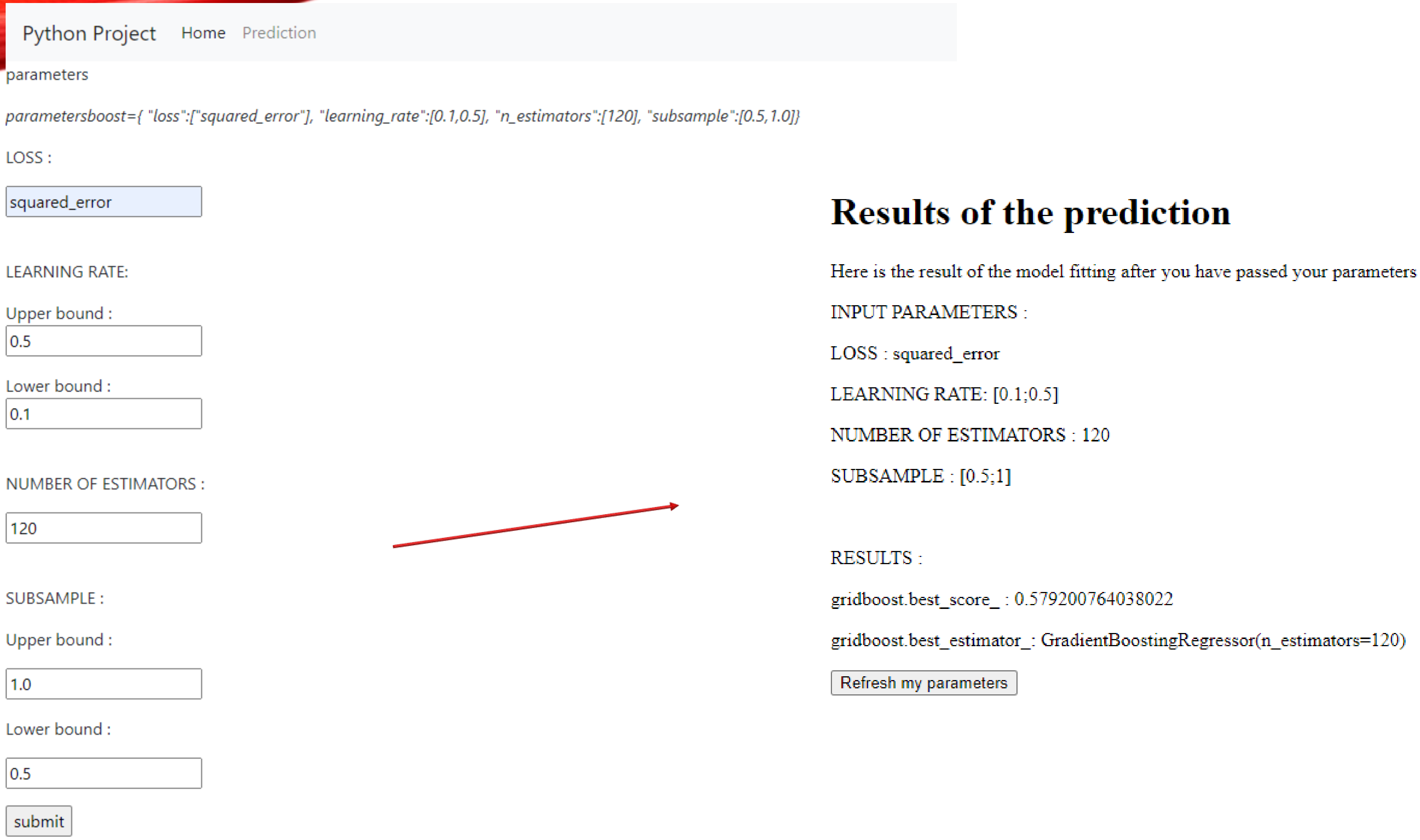


IV. Predictions

Since our target variable was the number of comments under a facebook post, so a numeric value, we tried many regression models such as SVM, Lasso, ElasticNetCV, Ridge, RandomForestRegressor or GradientBoostingRegressor. We applied each time the GridSearchCV in order to find the best parameters and our best result came with the RandomForestRegressor model. Nevertheless, the one we had implemented on our Flask API is the GradientBoostingRegressor.



IV. Predictions



parameters

parametersboost={ "loss":["squared_error"], "learning_rate":[0.1,0.5], "n_estimators":[120], "subsample":[0.5,1.0]}

LOSS :

squared_error

LEARNING RATE:

Upper bound : 0.5

Lower bound : 0.1

NUMBER OF ESTIMATORS :

120

SUBSAMPLE :

Upper bound : 1.0

Lower bound : 0.5

Results of the prediction

Here is the result of the model fitting after you have passed your parameters

INPUT PARAMETERS :

LOSS : squared_error

LEARNING RATE: [0.1;0.5]

NUMBER OF ESTIMATORS : 120

SUBSAMPLE : [0.5;1]

RESULTS :

gridboost.best_score_ : 0.579200764038022

gridboost.best_estimator_: GradientBoostingRegressor(n_estimators=120)

V. Conclusion

In conclusion, thanks to all our visualizations, we got to understand our dataset and the way its features impacted the target variable better.

This helped us to make better predictions with our models and interpret the results in a more accurate way.
We managed to get an accuracy of 70% on the test set, which is a good result given the target variable we had to predict.

We then created a Flask API to run one of our models and with which we can choose the parameters.