Python Project

Antoine Gounot and Ciaran Gruffeille

The Project

This project constitutes the final project for our course on Data Analysis for Python. The instructions where straightforward, we were given a dataset and it was up to us to analyse it, visualise the data and apply machine learning methods to it.

We started off by looking at the different column names and trying to understand what each column meant and how we could interpret them together in the most interesting way. We gave ourself a to do list of potential graphs and axes of analysis before diving in.

We then implemented our ideas to better understand the data and make hypothesis on what the most important variables were.

We then Implemented various Machine Learning algorithms and tested their accuracies against each other so as to choose the best one for the Flask API that we created at the end.

The Question

The objective of our dataset was to analyse the popularity of different online news articles and determine what makes them popular.

Looking at the dataset we can see that there a total of 39644 instances, this is different to the 39797 indicated in the names file that came with the dataset.

There are a total of 61 attributes, 58 predictive attributes, 2 non-predictive and the goal field (shares).

The Variables

There are a total of 61 variables in the dataset, with 58 predictive attributes, 2 non-predictive and 1 goal field.

With our first analysis of the dataset we rapidly determined that we would need to add at least 4 new variables.

A Weekday variable that would have the name of the day on which the article was published, this allows us to avoid using the 7 bools that were in the dataset originally (Weekday_is_Monday, Weekday_is_Tuesday…etc). It also allows us to easily group by the days of the week for our visualisations.

Similarly we created a Article Type which indicates the category that the article belongs to. Once again we avoid using the 6 bools that were present beforehand and it eases the process of grouping and visualisation. All in all it's a much Cleaner Method.

We also created two variables corresponding to the rate of images per word and the rate of videos per word to establish the importance of the ratio between word and image or word and video to the amount of shares of an article.

We initially also wanted to add a fifth variable called Complexity which would be the indicator of the complexity of a text. The higher it was the harder an article would be to read and the more knowledge it would require. However after doing some searching and some further reading we fell on the Hapax Richness and indicator equivalent to the rate of unique tokens which was already present in the dataset. The Hapax Richness is equivalent to the total amount of unique words divided by the total amount of words.

The Context

This project fits perfectly into the context of our studies because it requires a complete analysis, sorting and cleansing of a dataset. This is required so as to have an overview of the data given to us.

We also needed to implement Machine Learning algorithms so as to establish which one would work best for our dataset.

These two points constitute what we had learnt during this first semester with an added difficulty of adding an API in either flask or Django able to predict using the machine learning algorithm with the highest accuracy established prior.

This is something we had not seen in class but our knowledge was more than sufficient to succeed and the challenge was more than welcome. Bringing a “real world” aspect to the project.

The Models

To find the most fitting model for our dataset, we set up 7 models which are: Logistic Regression, Decision Tree , Random Forests, Linear Discriminant, KNN, Naïve Bayes and Ada Boost.

For each model we tweaked the different hyper parameters using for loops so as to obtain the best possible model for each.

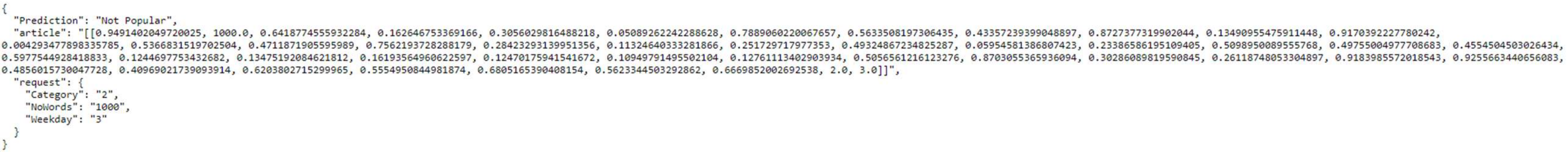
By training and testing them, we chose the Random Forests because it was the most accurate model.

The API

We created the API using the Flask Framework as it is a lot lighter than Django. The time constraints meant that we could not implement any html to create a more user friendly interface.

So for the Moment the API works by POST or GET request on the /API page. It is crude but it does the job. We decided to only request three variables, the day of the week, the category and the amount of words as asking for 47 variables seemed a bit much without a comfortable user interface.

The model is preloaded allowing for extremely fast response times to single queries. For example:



```
{
  "Prediction": "Not Popular",
  "article": "[[0.9491402049720025, 1000.0, 0.6418774555932284, 0.162646753369166, 0.3056029816488218, 0.05089262242288628, 0.7889060220067657, 0.5633508197306435, 0.43357239399048897, 0.8727377319902044, 0.13490955475911448, 0.9170392227780242,
0.004293477898335785, 0.5366831519702504, 0.4711871905595989, 0.7562193728288179, 0.28423293139951356, 0.11324640333281866, 0.251729717977353, 0.49324867234825287, 0.05954581386807423, 0.23386586195109405, 0.5098950089555768, 0.49755004977708683, 0.4554504503026434,
0.5977544928418833, 0.1244697753432682, 0.13475192084621812, 0.16193564960622597, 0.12470175941541672, 0.10949791495502104, 0.12761113402903934, 0.5056561216123276, 0.8703055365936094, 0.30286089819590845, 0.26118748053304897, 0.9183985572018543, 0.9255663440656083,
0.4856015730047728, 0.40969021739093914, 0.6203802715299965, 0.5554950844981874, 0.6805165390408154, 0.562334503292862, 0.6669852002692538, 2.0, 3.0]]",
  "request": {
    "Category": "2",
    "NoWords": "1000",
    "Weekday": "3"
  }
}
```

The response that we get in return is a json that has the prediction, the article variables as well as the request variables.

Conclusion

We have learnt a lot over the course of the project. It honed our skills as future Data Scientists. Namely in the different methods of visualisation but even more so in the thought process behind the analysis of a dataset. It was the first time that we saw so many different ways to analyse a dataset. Making the project extremely enticing and motivating.

We wish we could have had more information in the dataset so as the push our analysis of what makes an article popular even further. For now we have established that the complexity needs to compelling enough so as not to get bored but not too difficult so as to reach to a large audience. Objective and positive articles do better than subjective and negative ones. And the best categories are Social Media and Lifestyle as they are the most accessible. The best days are the weekend as this is when people have the most time to read the articles.

The Machine Learning process reinforced our knowledge on using Scikit-Learn.

The creation of an API was a fun challenge that added a taste of real world application to the project.

All in all we are very happy with the project and with our results.