University of Strathclyde

Big Data Fundamentals CS989

Understanding the YouTube's Trending Section

Francesco Giuseppe Mascia

Word count [3222]

Contents

Executive Summary

This report analyses information related to the YouTube Trending section, collected between November 2017 and June 2018. Through the use of Python, the data has been primarily analysed using descriptive statistics. Graphic visualizations were used to outline the structure of the main features of interest, such as the trending lifetime, the likes/dislikes ratio and the most common tags. Moreover, unsupervised and supervised statistical methods have been used to mine more detail from the structure. The supervised method of multiple regression in particular, led to some notable results regarding a difference in the analytical behaviour, between different categories of video.

1. Introduction

This report aims to analyse and describe a dataset using Python, in order to give an overview of the different tools this software has to offer for data analytics tasks.

The version that will be utilized is Python 3.7.1, using Jupyter Notebooks, a very user-friendly tool, especially when it comes to data analysis and visualisation. More detailed information available in the Appendix (**A1**).

The analysis will focus on a dataset named *"Trending YouTube Videos Statistics"*, which includes a daily record of different attributes of the top trending YouTube videos, it can be found for free on *Kaggle.com*. According to the *YouTube Help* site "*Trending aims to surface videos that: are appealing to a wide range of viewers, are not misleading, clickbaity or sensational, capture the breadth of what's happening on YouTube and in the world, ideally, are surprising or novel*".

What exactly leads a video to go trending is not of public domain, but again according to *YouTube Help*, it is related to the view count, the rate of growth and the age of the video, with a dynamic process that reiterates approximately every 15 minutes.

Trending videos are usually different in different countries, the original file included various datasets from the main countries in the world. Only the one related to the United Kingdom (*YouTube.com/GB*), will be further analysed.

2. The Dataset

2.1. Description

The dataset reports information regarding the YouTube Trending section, recorded during a time frame of seven months, from November 14, 2017 to June 14, 2018. The dataset comprises a total of 38916 rows and 16 columns. Each row refers to a specific video and the columns report the attributes. A quick description of those included in the analysis is following provided:

- title: the title of the video;
- trending_date: the day when the data was recorded;
- channel_title: the channel that published the video;
- category_id: the category of the video (each number refers to a different category);
- tags: tags related to the video;
- views: how many views that video had at the day recorded;
- likes: how many likes;
- dislikes: how many dislikes;
- comment_count: how many comments in the comment section;
- comments_disabled: if the comments were disabled in that video;
- ratings_disabled: if likes and dislikes were disabled in that video;

A significant part of the information needed for the analysis is not yet included in the dataset, however, it can be gathered from what is already there. For these and other reasons it is essential to go through a cleansing and preparation procedure, that will lead to a thinner and more manageable dataset.

2.2. Data Cleansing and Preparation

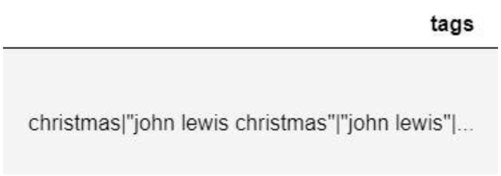
Before the proper beginning of the analysis, a crucial step is to clean and improve the data, in order to exploit it at the maximum potential. In this chapter the packages Numpy, Pandas, and Counter have been used to pursue this task.

The first problem to face is that all videos in the dataset are coming up for any single day they went trending, hence, most of them are showing more than once. A clever strategy to tackle this problem is by adding a column that shows the number of objects with the same title. This can be done combining the `groupby()` and the `transform()` functions (*StackOverflow.com*, 2015). The new column will be named 'count'.

Then again, to get rid of the duplicates is sufficient to aggregate the videos by title. In this case for each title, only the minimum values are kept, so to have only the information related to the first day the video went trending. The minimum values represent the features of a video for the first day it went trending, and the new column shows how many days it trended in total. This new information will be very useful for further analysis and will be explained into more detail in the next chapters.

The number of rows had now decreased to only 3369, so each video was showing on average 11.55 times (38916/3369).

A second possible issue can be addressed to the tags' column.



tags

christmas|"john lewis christmas"|"john lewis"|...

Table 1 Tags column, snapshot from Jupyter

In **Table 1** it is possible to see that the words are separated by special characters, this fact could make really difficult to analyse the text content in an appropriate way. The `str.replace()` function, can be helpful in these cases, allowing the user to discard any kind of inconvenient character from the data. In this case, the special characters "[**\"|',**]" will be replaced from a simple space.

3. Key Challenges

The YouTube dataset allows the user to range over I wide variety of analysis. In this report, they will be split into three phases across chapter **4**. Each phase will go into detail about a different aspect of the analysis, using different suitable methods for different purposes.

The first phase (**4.1.**) will focus on data mining, gathering a first wide knowledge of basic information. During this process a variety of problems will be addressed, i.e. “What are the videos that trended for the higher number of days?”, “What are the channels with most trending videos?” or “What are the most popular tags among most popular videos?”. Mostly descriptive statistics will be used to face these challenges.

The second phase (**4.2.**) will be more focused on the discovery of hidden structures underlying the data, "Are there any patterns within the trending videos?". The engagement factors will be taken into account (views, likes, dislikes, comments), to determine how they can affect the trending life of a video. This aspect will be deepened using the unsupervised method of cluster analysis.

Furthermore, one last phase will conclude the analysis (**4.3.**). A regression model will be crafted, trying to gather useful insights about the videos' behaviour, and if this can be somehow predicted using the information available. This part is intended to highlight the key challenge of this paper: "Is it possible to predict the future duration of the trending life of a video using engagement factors?".

4. The Analysis

4.1. Exploratory Analysis and Descriptive Statistics

Python offers a discreate amount of open source packages, that can be used for data visualisation, such as Pandas, Matplotlib, Plotly and Seaborn, this topic is widely discussed in McKinney, W. (2009).
During this part of the analysis, a summary knowledge of the data will be assessed. This chapter will particularly focus on the trending period, analysing other attributes in the light of it.

As previously reported in chapter **2.2.**, the average video's trending life is 11.55 days. Anyhow, some of them stay in the trending section remarkably longer than others.

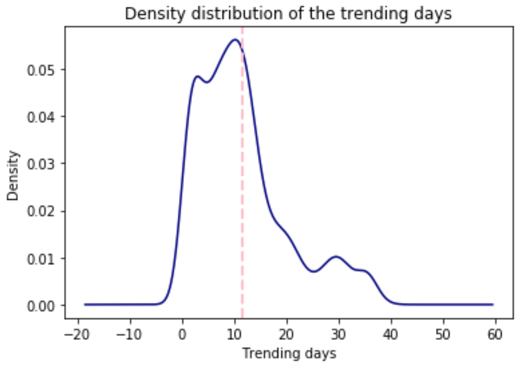


Figure 1 Density distribution for the number of days the videos went trending in the interval of seven months

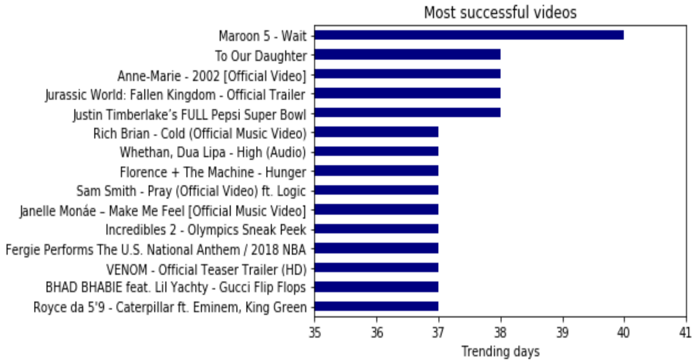


Figure 2 Top 15 video that trended for the highest number of days

The density ploy in **Figure 1** shows the trending days distribution in the dataset. It is visually clear, that a pike of density coincides with the mean (pink line). In general, most of the videos trend for a number of days between 1 and 17, and, according to the sample, is really difficult for a video to trend for more than 40 days in an interval of seven months.

Moreover, **Figure 2** displays more in detail which videos were the most successful. Confirming the results from **Figure 1**, none of them exceeds the upper limit of 40, which is the highest result achieved from the video "*Wait*" by the *Maroon 5*. Then there is a tie for second place, a cluster of four videos that totalised a score of 38. The remaining 13 places of the ranking are all taken by videos that had 37 trending days during the time frame considered.

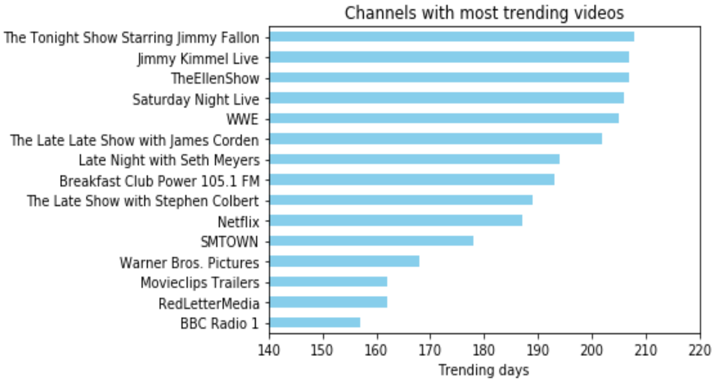


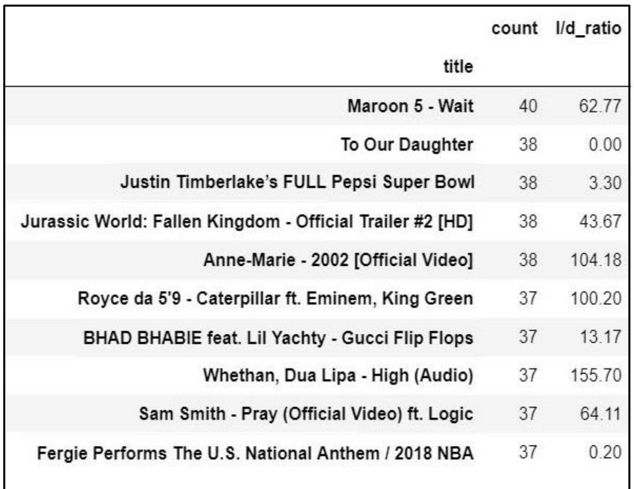
Figure 3 Top 15 channels with the highest sum of trending days for their videos

Subsequently, is equally important to understand, what channels performed the better. **Figure 3** outlines the top 15 channels, which had their videos in the trending section more often. The YouTube channel *"The Tonight Show with Jimmy Fallon"* is top of the chart, with 208 days totally. Jimmy Kimmel, Ellen DeGeneres and the *"Saturday Night Live"* follow with a score above 200. A general tendency can be clearly noticed, out of the top ten channels, seven are related to a Night TV Show. It can be assumed that these types of channel, are those that most likely will publish trending videos.

Every video has a different amount of likes and dislikes, on average the likes/dislikes ratio is 23.16, it means that for every 23 likes the video gets 1 dislike. However, this ratio is remarkably different among different videos. In **Table 2,** this data can be observed in the light of the ranking previously examined in **Figure 2**.

The videos that stay trending for many days have a ratio quite distant from the mean, either a lot lower or a lot higher. While the first video has a ratio three times bigger than the mean, the third one is eight times less likeable than average. Based on this insight it might be assumed that a strong impact, either positive or negative, is good for a video to keep it trending. For further analysis a heatmap is computed, in order to analyse the correlation between the relevant quantitative variables in the dataset.

The heatmap in **Figure 4** shows that the only notable correlation is between comments and reactions (likes and dislikes). It is possible to go into more detail using a Python correlation function, `corr()`, which confirms the high interrelationship between comments and likes, almost 80%. So, people are usually inclined to leave a comment if they press like and vice-versa. Slightly minor correlation is spotted relatively to dislikes (72%).



title	count	l/d_ratio
Maroon 5 - Wait	40	62.77
To Our Daughter	38	0.00
Justin Timberlake's FULL Pepsi Super Bowl	38	3.30
Jurassic World: Fallen Kingdom - Official Trailer #2 [HD]	38	43.67
Anne-Marie - 2002 [Official Video]	38	104.18
Royce da 5'9 - Caterpillar ft. Eminem, King Green	37	100.20
BHAD BHABIE feat. Lil Yachty - Gucci Flip Flops	37	13.17
Whethan, Dua Lipa - High (Audio)	37	155.70
Sam Smith - Pray (Official Video) ft. Logic	37	64.11
Fergie Performs The U.S. National Anthem / 2018 NBA	37	0.20

Table 2 Likes/dislikes ratio for the Top 10 trending videos (0.00 means that the reactions were disabled)

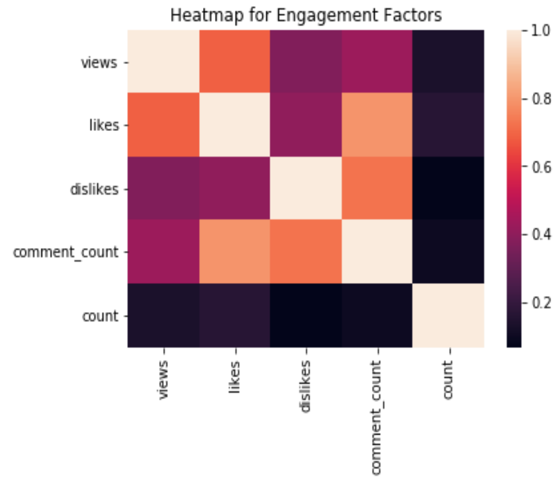
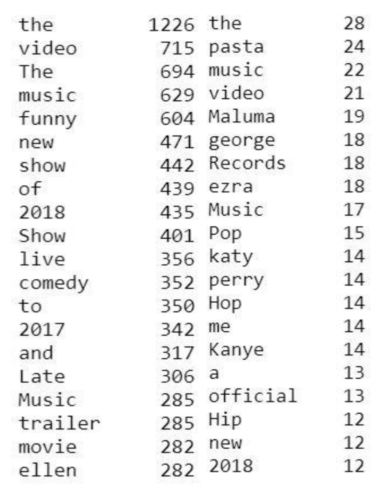


Figure 4 Heatmap showing the correlation between the most relevant quantitative variables



```
the          1226  the           28
video         715  pasta         24
The           694  music         22
music         629  video         21
funny         604  Maluma        19
new           471  george        18
show          442  Records       18
of            439  ezra          18
2018          435  Music         17
Show          401  Pop           15
live          356  katy          14
comedy        352  perry         14
to            350  Hop           14
2017          342  me            14
and           317  Kanye         14
Late          306  a             13
Music         285  official      13
trailer       285  Hip           12
movie         282  new           12
ellen         282  2018          12
```

Table 3 Top 20 used tags, overall (on the left), among Top 100 videos (on the right)

A really powerful tool that YouTube uses to sort and manage his videos are the tags. Using the `str.split()` and `stack()` functions combined it is possible to analyse the presence of single tag words across the videos (*StackOverflow.com*, 2017). In **Table 3** two different types of information are provided, on the left side the top 20 most popular tags overall, on the right side only those among the top 100 trending videos. The tags in the overall ranking confirm the Night Shows' popularity that already popped out in **Figure 3**. A curious fact is how popular the tag *'pasta'* is across the top 100 most trending videos.

4.2. Unsupervised Method, K-Means Clustering

Unsupervised learning is a statistical method that allows the user to train a model, from data that has not been labelled or classified. The goal is the one to find the labels through the method itself, based on commonalities and patterns that are hidden in the data structure.

A particularly common unsupervised method is the cluster analysis. From a group of objects, an algorithm is applied in order to split the original set into two or more subsets (clusters). The main task of this method is to obtain clusters characterized by a high intra-cluster similarity and a high extra-cluster dissimilarity.

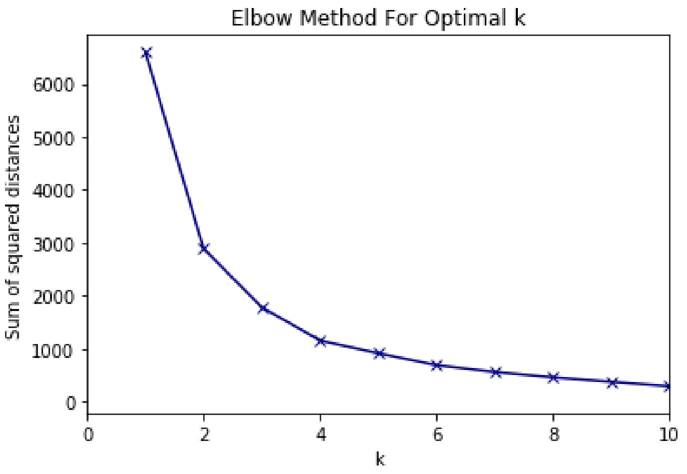


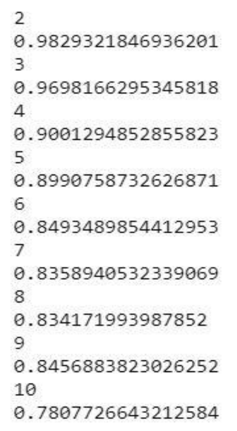
Figure 5 Elbow Method to determine the ideal number of clusters

The specific type of cluster analysis used in this chapter is the K-Means Clustering, this method gives the user the possibility to choose a priori the number of clusters in which to partition the data, contrariwise to other methods such as the Hierarchical Clustering. A deeper understanding of the topic can be found in Friedman, Tibshirani, Hastie (2009) and in Swamynathan M. (2017) for what concerns the Python applications. The last one will be the main reference for this chapter.

In order to perform a cluster analysis a few preparatory steps must come first. First of all, the null values must be dropped, in this case, those videos where reactions and comments have been disabled (69 of them were found). Secondly, it is necessary to choose what variables the analysis will be based on. The engagement factors will be used to research clusters in the dataset, so views, reactions (likes and dislikes) and comments. However, these four attributes have a remarkably different scale, i.e. the mean for the views is approximately 1.3 million, while for comments only slightly more than 6 thousand. In this case might then be appropriate to normalize the data, in order to put them all on the same scale. This way all the attributes will have the same weight when running the algorithm.

A crucial point is then to determine *k*, the number of clusters to train the algorithm. A variety of methods can be used. A first intuitive piece of information can be gathered using the Elbow Method, which measures the sum of squared distances, *"This method exists upon the idea that one should choose a number of clusters so that adding another cluster doesn't give much better modelling of the data"* (Bholowalia and Kumar, 2014).

As it can be seen in **Figure 5**, a first visual impact could suggest an ideal number of clusters between three and five. A further analysis can be carried out using the silhouette score, a measure of how good the model is, that considers the mean intra-cluster distance and the mean nearest-cluster distance (where 1 is the best result possible and 0 the lowest).



```
2
0.9829321846936201
3
0.9698166295345818
4
0.9001294852855823
5
0.8990758732626871
6
0.8493489854412953
7
0.8358940532339069
8
0.834171993987852
9
0.8456883823026252
10
0.7807726643212584
```

Table 4 Silhouette score for *k* in range [2-10]

Table 4 is showing the silhouette scores from two to ten clusters. An ideal *k* number is one the balances a high score with a high number of clusters. In this case, *k=5* could be considered an appropriate choice.

Running the algorithm for *k=5*, the output does not outline significant patterns. Most of the objects, 3190 of a total of 3300, belongs to one single cluster. A visual representation of the clusters can be provided using labelled scatter plots. In **Figure 6** three different scatter plots are displayed. The three visualisations have the views in the x-axis, comparing the clusters among the other three engagement factors, likes, dislikes and comments. The five clusters are represented in five different colours, the colour purple indicates the most populous cluster, described previously. The other four clusters are basically only composed by outliers, characterised by an abnormal amount of views, reactions or comments

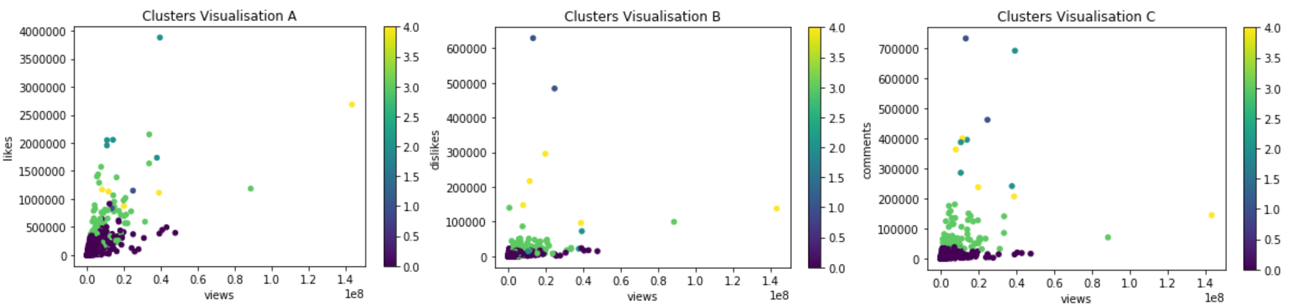


Figure 6 Three different scatter plots capturing the five clusters in different colours. Number of views in the x-axis, likes, dislikes and comment in the y-axis

Figure 7 gives a zoomed version of the previous graph. The points in the three scatter plots follow a similar pattern, most of the observations are chaotically herded in proximity of the mean. Beyond the area of maximum density, they gently spread across an extremely wide range, as it is clear in **Figure 6** as well.

According to this analysis, it could be assumed that trending YouTube videos cannot be grouped easily based only on engagement factors.

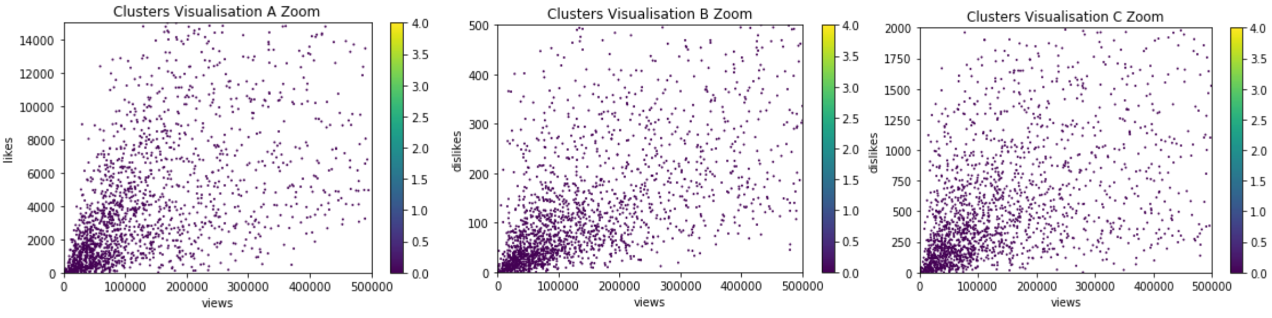
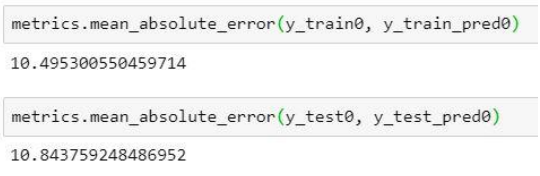


Figure 7 Clusters visualisation from Figure 6, zoomed in proximity of the mean

4.3. Supervised Method, Linear Regression

Supervised methods aim to train an algorithm using an input-output procedure, based on a pre-existent input-output scheme. This chapter will focus on Linear Regression, the most suitable method when dealing with numerical data as those currently under analysis. Usually, the data gets split into two subsets, a training set and a testing set. The first one has the specific task to train the model, the second one is helpful to ascertain if the model is efficiently achieving his purpose. The Regression based on one single predictor is called Simple Linear Regression, but as according to Swamynathan M. (2017) *"In most of the real-life use cases there will be more than one independent variable".* Indeed, Multivariate Regression will be the supervised method used to deepen the knowledge of how the videos' attributes are related to their own trending life. A wide range of possible applications are implemented in Python as again described in Swamynathan M. (2017).

A first hypothesis to be tested is that the amount of future trending days can be predicted using engagement factors. Hence, a multivariate regression can be computed, this first model will be named Model 1. The predictors will be: views, likes, dislikes and comments, while the target will be the *'*count*'* column (trending days). The summary statistics of this first model can be found in the Appendix (**A2**). To determine the precision of the model a few methods are available, one of them is the mean absolute error, which measures the average distance of the predicted values from the original target values. The target to be predicted has a range from 1 to 40 with a mean of 11.55. In **Table 5**, we can see that the model is not working has it should to predict the target, with an error remarkably too high to consider the prediction reliable.



```
metrics.mean_absolute_error(y_train0, y_train_pred0)

10.495300550459714

metrics.mean_absolute_error(y_test0, y_test_pred0)

10.843759248486952
```

Table 5 Model 1 MAE metrics, snapshot from Jupyter

Some attempt can be made to improve the model. For instance, a new set of variables that represents the relations between the various engagement factors could be added as predictor. Hence, 'likes/dislikes', 'views/likes', 'views/dislikes' and 'views/comments' ratios will now be included (Model 2). The results got slightly better due to this changing (summary statistics available in the Appendix, **A3**). The test MAE decreased to around 9.5, still a value too substantial to label the model as reliable. In **Figure 8** a visual representation of the model is displayed, the leverage measures the error of the prediction, the residuals measure the dissimilarity of an observation from the other ones. As it is clear from the scatter plot, it is possible that a few outliers are compromising the model, a better result could possibly be achieved leaving them out of the equation.

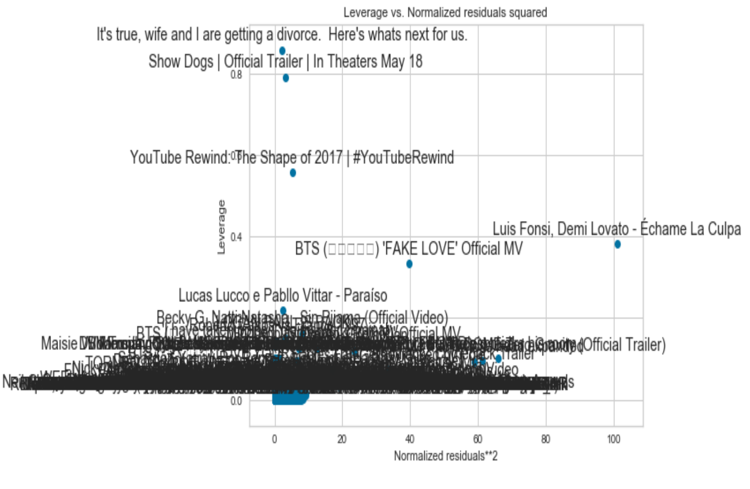


Figure 8 Leverage/Residuals scatter plot of Model 2

A further analysis can be now conducted considering the different category each video belongs to, fifteen categories are represented in the dataset (see **Figure 9**). The approach will be to create 9 subsets, one for each of the main categories, and fit 9 new different regressions (one for each category), using the same eight predictors mentioned in Model 2.

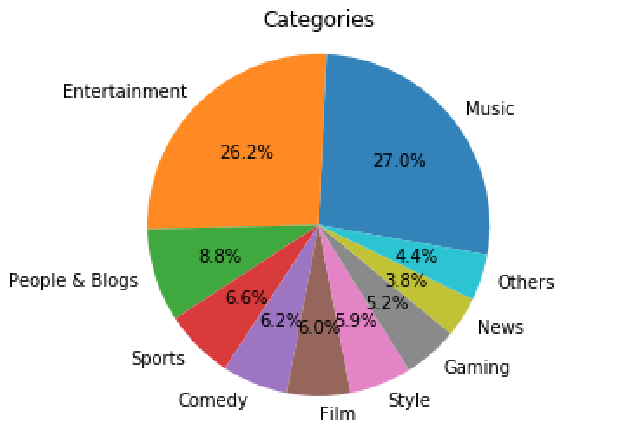
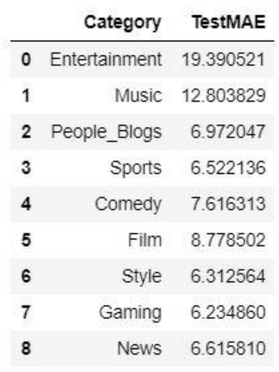


Figure 9 Pie chart for the percentage of videos from each category



	Category	TestMAE
0	Entertainment	19.390521
1	Music	12.803829
2	People_Blogs	6.972047
3	Sports	6.522136
4	Comedy	7.616313
5	Film	8.778502
6	Style	6.312564
7	Gaming	6.234860
8	News	6.615810

Table 6 MAE score for each model related to the specific category

In **Table 6** the insights regarding the new models are proposed. The regression works remarkably better for some categories rather than others. While *'*Entertainment*'* videos seem really difficult to predict, with an error of almost 20, *'*Gaming*'* and *'*Style*'* are more likely to coincide with the prediction value, since the MAE reported an error only slightly higher than 6.

5. Reflection on the Methods and Conclusion

During this paper a plethora of methods has been utilized, gaining knowledge from different angles.

Descriptive statistics in chapter **4.1.** proved to be a suitable tool to gather information from the dataset at hand. Due to Python visualisation libraries, some interesting insights were highlighted, especially regarding the channels and the tags related to trending videos.

In chapter **4.2.** the Cluster Analysis didn't report significant results, since no clear patterns were identified from the unsupervised algorithm. The output was represented using multiple scatter plots, which outlined a messy distribution, not easy to be divided into sensate groups. Possibly better results could be achieved using other video's attributes not included in the data.

The Multivariate Regression in chapter **4.3.** proved how difficult can be to predict internet trends, such as those related to YouTube videos. The Model 1 and 2 didn't give reliable predictions. Nonetheless, segmenting the dataset by category provided a good feedback, outlining remarkable differences among the different categories. Hence, it might be a good strategy to fit new models for even smaller categories of videos.

In conclusion, despite the high complexity of the YouTube trending mechanism, a marginal but encouraging understanding of the subject has been reached.

References

- Bholowalia P., Kuma A. (2014). *EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. International Journal of Computer Applications (0975 – 8887) Volume 105 – No. 9, November 2014*. [online] Available at: https://pdfs.semanticscholar.org/5771/aa21b2e151f3d93ba0a5f12d023a0bfcf28b.pdf [Accessed 1 Nov. 2018]

- Alade T. (2017). *Tutorial: How to determine the optimal number of clusters for k-means clustering*. Cambridge Spark. [online] Available at: https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering-14f27070048f [Accessed 26 Oct. 2018]

- Hastie T., Tibshirani R., Friedman J. (2009). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. 2nd ed. Springer. pp.501-530

- McKinney W. (2017). *Python for Data Analysis, Data Wrangling with Pandas, NumPy, and IPython*. 2nd ed. Sebastopol, US. O'Reilly Media, Inc. pp.253-316. [online] Available at: http://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/Python-for-Data-Analysis.pdf [Accessed 3 Nov. 2018]

- Mitchell J. (2018). *Trending YouTube Videos Statistics. Kaggle.com*. [online] Available at: https://www.kaggle.com/datasnaek/youtube-new/home [Accessed 1 Nov. 2018]

- *StackOverflow.com* (2015). *Python pandas: Add a column to my dataframe that counts a variable.* [online] Available at: https://stackoverflow.com/questions/29791785/python-pandas-add-a-column-to-my-dataframe-that-counts-a-variable [Accessed 1 Nov. 2018]

- *StackOverflow.com* (2017). *Counting the frequency of words in a Pandas data frame*. [online] Available at: https://stackoverflow.com/questions/46786211/counting-the-frequency-of-words-in-a-pandas-data-frame [Accessed 1 Nov. 2018]

- Swamynathan M. (2017). *Mastering Machine Learning with Python in Six Steps*. Berkely, US. Apress. [online] Available at: https://tanthiamhuat.files.wordpress.com/2018/04/mastering-machine-learning-with-python-in-six-steps.pdf [Accessed 2 Nov. 2018]

- *YouTube.com/GB*. [online] Available at: https://www.youtube.com/?gl=GB&hl=en-GB [Accessed 3 Nov. 2018]

- *YouTube.com. YouTube Help. Trending on YouTube.* [online] Available at: https://support.google.com/youtube/answer/7239739?hl=en [Accessed 1 Nov. 2018]

Appendix

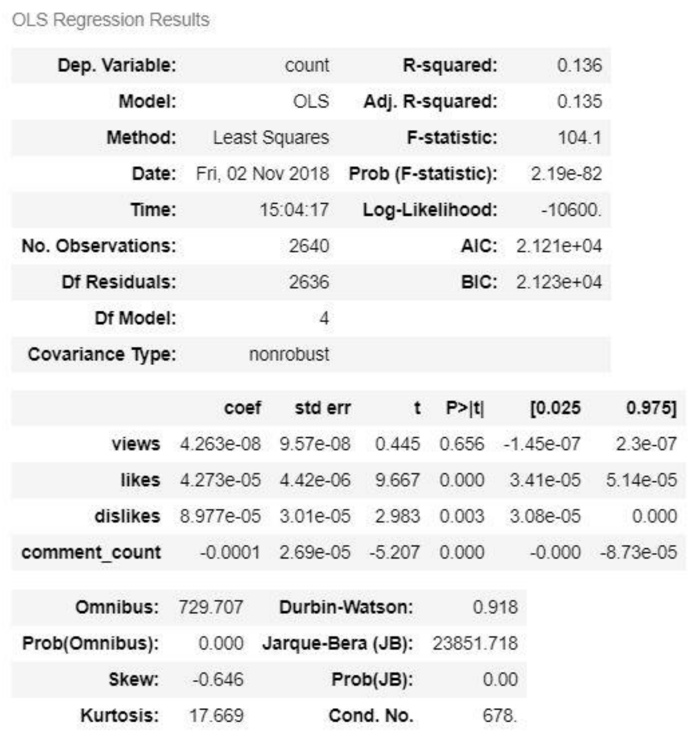
Appendix 1

Python version: Python 3.7.1 | Anaconda3-5.2.0 Windows-x86 (64 bit)

Development Environment: Jupyter Notebook
Copyright © 2018 Project Jupyter – Last updated Sun, Oct 21, 2018

Packages: Numpy, Pandas, Matplotlib, Plotly, Collection, Seaborn, Sklearn, Statsmodels.api, Statsmodels.graphics.regressionplots

Appendix 2



OLS Regression Results

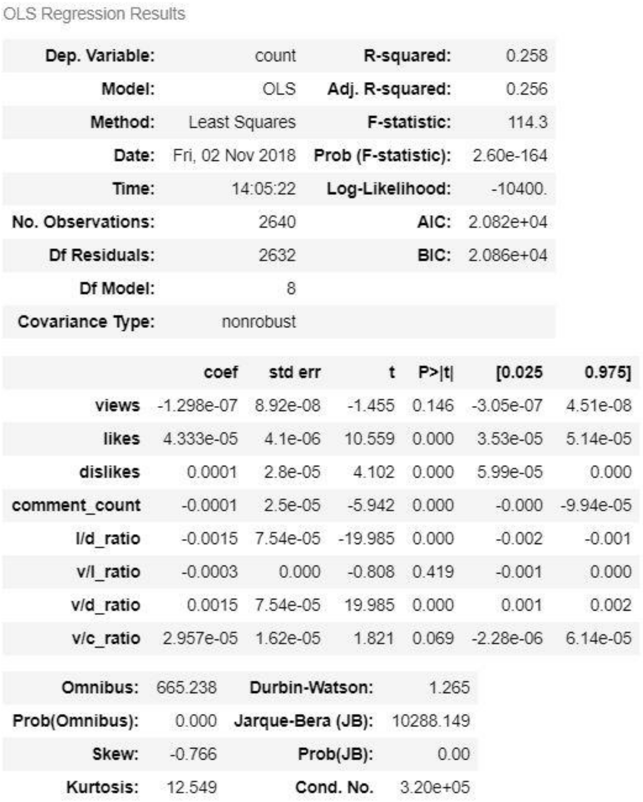
Dep. Variable:	count	**R-squared:**	0.136
Model:	OLS	**Adj. R-squared:**	0.135
Method:	Least Squares	**F-statistic:**	104.1
Date:	Fri, 02 Nov 2018	**Prob (F-statistic):**	2.19e-82
Time:	15:04:17	**Log-Likelihood:**	-10600.
No. Observations:	2640	**AIC:**	2.121e+04
Df Residuals:	2636	**BIC:**	2.123e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
views	4.263e-08	9.57e-08	0.445	0.656	-1.45e-07	2.3e-07
likes	4.273e-05	4.42e-06	9.667	0.000	3.41e-05	5.14e-05
dislikes	8.977e-05	3.01e-05	2.983	0.003	3.08e-05	0.000
comment_count	-0.0001	2.69e-05	-5.207	0.000	-0.000	-8.73e-05

Omnibus:	729.707	**Durbin-Watson:**	0.918
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	23851.718
Skew:	-0.646	**Prob(JB):**	0.00
Kurtosis:	17.669	**Cond. No.**	678.

A2 Summary statistics for Model 1

Appendix 3



OLS Regression Results

Dep. Variable:	count	**R-squared:**	0.258
Model:	OLS	**Adj. R-squared:**	0.256
Method:	Least Squares	**F-statistic:**	114.3
Date:	Fri, 02 Nov 2018	**Prob (F-statistic):**	2.60e-164
Time:	14:05:22	**Log-Likelihood:**	-10400.
No. Observations:	2640	**AIC:**	2.082e+04
Df Residuals:	2632	**BIC:**	2.086e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
views	-1.298e-07	8.92e-08	-1.455	0.146	-3.05e-07	4.51e-08
likes	4.333e-05	4.1e-06	10.559	0.000	3.53e-05	5.14e-05
dislikes	0.0001	2.8e-05	4.102	0.000	5.99e-05	0.000
comment_count	-0.0001	2.5e-05	-5.942	0.000	-0.000	-9.94e-05
l/d_ratio	-0.0015	7.54e-05	-19.985	0.000	-0.002	-0.001
v/l_ratio	-0.0003	0.000	-0.808	0.419	-0.001	0.000
v/d_ratio	0.0015	7.54e-05	19.985	0.000	0.001	0.002
v/c_ratio	2.957e-05	1.62e-05	1.821	0.069	-2.28e-06	6.14e-05

Omnibus:	665.238	**Durbin-Watson:**	1.265
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	10288.149
Skew:	-0.766	**Prob(JB):**	0.00
Kurtosis:	12.549	**Cond. No.**	3.20e+05

A3 Summary statistics for Model 2