Danmarks Tekniske Universitet

02450: Introduction to Machine Learning and Data Mining

Report 1
Data, feature extraction and visualisation

Lukas
Leindals
s183898

Gustav Gamst
Larsen
s180820

Gabriella
Morote
s170820

Contents

1 A description of our data

1.1 Problem of interest

YouTube is the biggest video sharing platform in the world, where everyone can share their opinion, promote themselves or publish homemade entertainment. From a business perspective, YouTube has also become a platform you can live off, with a few people becoming millionaires off of their content on youtube.com. With data on what the population of the 10 biggest countries (when looking at population) are watching on the site, we would be able to see where certain channels will be able to bloom, or what content is being watched the most, which could be great information for companies to see where they should spend their add money when coming to different countries or different types of videos that attracts the most viewers.

1.2 General Information about the data

The data we have chosen to look at is data regarding the trending videos uploaded to youtube.com. The data was last updated 3 months ago, and contains data in 16 columns such as category, day of upload, day the video started trending, likes and dislikes etc. The data was obtained at Kaggle.com with a usability score (from their own site) of 7.9/10. There are 10 different CSV files showing the trending videos in 10 different countries. The data has not been touched, and is therefore a fresh dataset.

Our datasets has the following variables:

- Video id
- Trending date
- Title
- channel title
- category
- publish time
- tags
- views
- likes
- dislikes
- comment count
- thumbnail link
- comments disabled
- ratings disabled
- video error or removed
- description

1.3 Primary modelling aim

We wish to classify and identify what makes a video trend fast. Because the trending can depend on many variables (as listed above) we are interested in knowing whether the video will trend faster if it gets a lot of likes? Or perhaps dislikes is better for a video to trend faster? We could also be interested in which video categories are trending the most in the in different countries.
Machine-learning and data-visualisation gives us a clear picture (literally) of all these things and it is our primary aim to use machine learning for exactly this purpose.

Because our interest in our data set is to classify how fast a video would trend depending on it's amount of likes, views etc. our main machine learning aim is to use the classification method.

When carrying out a classification we would like to try and predict either country or category of the data. This will be done by looking at the attributes likes, dislikes, views and comment count, depending on which variables gives the best classification result. This could be further analysed with the Principal Component Analysis of the attributes.

We could also use other machine learning methods with some of our attributes. For instance, another supervised learning method as regression could be done instead of classification because we with the information of a video, could make a prediction of how many days it would take the video to trend.

A regression will be done by looking at the four variables we also used for classification to see how they depend on each other.

When doing our clustering it would be fun to see if we could chose two of the previously four mentioned variables to cluster the videos that have about the same value of trending time together, as this could be used as a form of classification to see if these variables say something about what makes a video trend fast or slow. These variables could also be used to see if we could find a rule, that e.g. says how many likes a video must have to trend in the first 3 days or similar rules when doing association mining.

We could use anomaly detection in our YouTube data set in order to be able to detect whenever there is a trending video that differs in attributes from the other trending videos. It could be important to check which category this video is or check out what the video contains, as it might could be unpleasant and should be reported to YouTube and removed.

2 Explanation of our Attributes

2.1 Description of attributes

We choose to look at the variables we found soothing for the project and to make the best PCA. We have chosen the following Variables:

- Country
- Category
- Day of upload
- Day of trending
- Views
- Likes
- Dislikes
- Comment count

2.1.1 Country

We want to look at trends across country borders and to do this we compare some of the largest countries when looking at population numbers. In our data, we have categorised the countries after index numbers, therefore the variable is a discrete nominal attribute in our dataset.

Index Number	Country
1	Canada
2	Germany
3	France
4	Great Britain
5	India
6	Japan
7	South Korea
8	Mexico
9	Russia
10	United States

Table 1: A table showing the indexing of the countries in our dataset

2.1.2 Category

This attribute contains 30 different labels. We have changed it to only contain the six labels 'Music','Pets & Animals', 'Gaming', 'People & Blogs', 'Howto & Style' and 'Movies', when doing clustering and PCA. This is done to get a better overview.

The video category is described with a number but can be translated with a .json document that came with the dataset. This feature is a discrete nominal attribute to the dataset. There is not much to say about this attribute, all the datapoints is categorised by this attribute and when you upload a video, you choose the category that fits the video the most. The only flaw in this, could be if the one who uploaded the video, choose the wrong category but this would be obnoxious since a category tag probably would make the video easier to trend.

2.1.3 Trending Time

The day of upload and day of trending are both discrete interval attributes. We wish to be able to see how long it takes for certain video-categories to become trendy. Therefore we used these two variables to create our own variable called trending time. This a discrete interval variable which tells the difference in days between the day of upload and the day the video was considered trending.

2.1.4 Views

The views are again a number, but it is a discrete ratio attribute. We know that 100 views are more than 90 views, and therefore we can order this type of data from highest to lowest and vice versa. This data is important since we can look at how popular a video is, and rank other trending videos against each other in views. This could also tells us something about how many views a trending video usually gets and which category gets the most views.

2.1.5 Likes and Dislikes

The likes and dislikes are also discrete ratio attributes much like the "views" variable. They do pretty much the same, but from this variable we can extract if a video is trending because its content is negative or positive. The views only gives us the arousal where as the feeling behind the arousal will be clear when taking these variables into account.

2.1.6 Comment count

A discrete ratio attribute as well. We have chosen to look at the comment count as well to see whether or not the amount of comments correlate with the trending time. Comments may in fact have an effect on the trendiness of a video, perhaps more comments mean faster trending time.

2.2 Data issues

Our dataset is complete, meaning we do not have any missing values or corrupted data. The only minor problem with the data, was the format of the dates, which was fixed by converting it to the format "yyyy-mm-dd". Another issue was that our variable trending time had most of its values in a the range [1,3] days, but also had a lot of observations ranging from [4,4215] days. This weird distribution meant that we could not use this a variable to predict anything from as even a standardisation or log-transfomation did not do anything good. Therefore we will consider using this as one of the variables we would like to predict the value of, when classifying and other things in the later reports. Another data issue is that the values of different attributes does not have the same dimensions, which is not suitable when making visualisations to understand the data. We have therefore chosen to standardise our data.

2.3 Summary statistics of the attributes

To get a better overview of our data, we have created a table giving us the summary statistics of each category to see if there is outliers or not.

	Category id	**Views**	**Likes**	**Dislikes**	**Comment count**	**Country**	**Trending time**
Count	375942	375942	375942	375942	375942	375942	375942
Mean	20.23	1326568	37884	2126	4254	5.46	7 days
Std	7.13	7098568	165413	22484	25459	2.982	97 days
Min	1	117	0	0	0	1	0 days
25 % qt	17	46978	669	41	109	3	1 days
50 % qt	23	177371	3446	179	511	5	1 days
75 % qt	24	647679	17477	749	2011	8	3 days
Max	44	424538912	5613827	1944971	1626501	10	4215 days

Table 2: Summary statistics

When looking at table 2 we observe that a lot of our means are higher than our 3rd quartile, meaning we have some outliers. These values does however seem to be the result of one outlier messing with the entire dataset, but rather a lot of outliers affecting the dataset. This causes some problems as our data will be very right-skewed.

3 Datavisualisation with PCA

3.1 Histograms

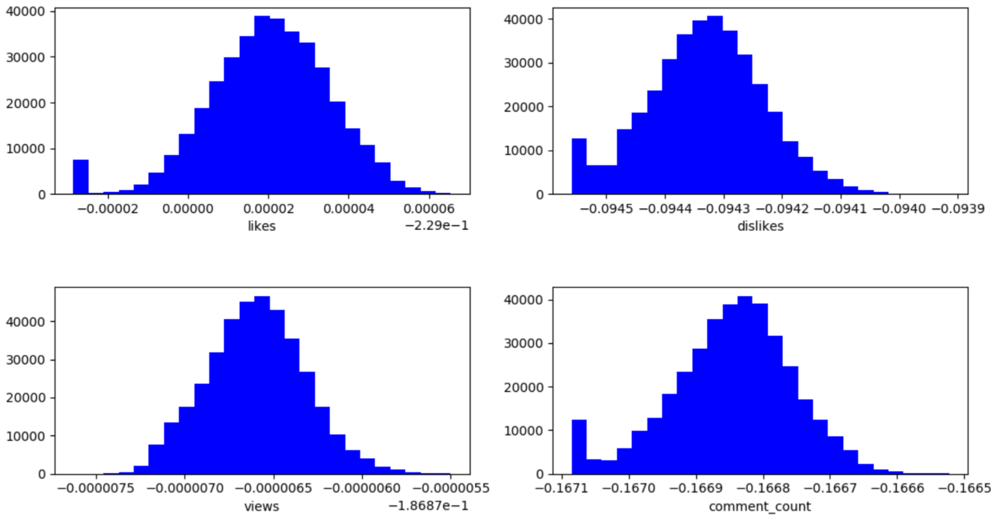

Figure 1: Histograms over variables after being log-transformed

In Figure 1 we see the histograms for each of our variables after applying log transformation. We performed log transformation instead of normalisation to make the data normally distributed, since it at first came out very right-skewed, which was indicated by table 2. We expected our data to be normal distributed from the beginning due to the central limit theorem, which says large amounts of data can be approximated with a normal distribution. This did however not seem to be the case until after we log-transformed the data, as seen in the histograms.

3.2 Principal component analysis

When we perform PCA we we will create the principal components from all our transformed data, but only our 6 chosen categories will be projected onto the eigenvectors as this improves the apprehension of our plot. We made our PCA with the variables likes, dislikes, views and comment count, which were projected onto our principal components. After standardising and performing PCA the first time on our dataset, it became very clear that outliers caused some trouble as the data seemed clustered together with outliers extending the axis, which meant a lot of the data point came to lie on top of each other. As our data is right-skewed, we can improve our data by log-transforming our data to make it more normal distributed as this will improve our PCA as seen on figure 3. However it is still not very clear how the categories differs from one another through the axes as all the data points within the different categories quite follow the same positive linear shape. The biggest variation between the categories seems to be between 'Music' in the upper right corner (most positive values) and 'People&Blogs' in the lower left corner (most negative values). But the data points within the category still follows the same line as the other categories and this could imply a hard classification task for the categories or country classification.

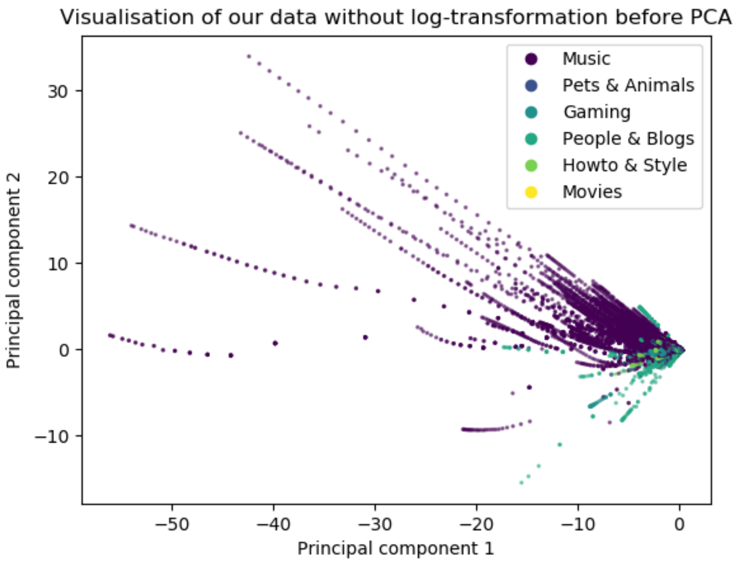

Figure 2

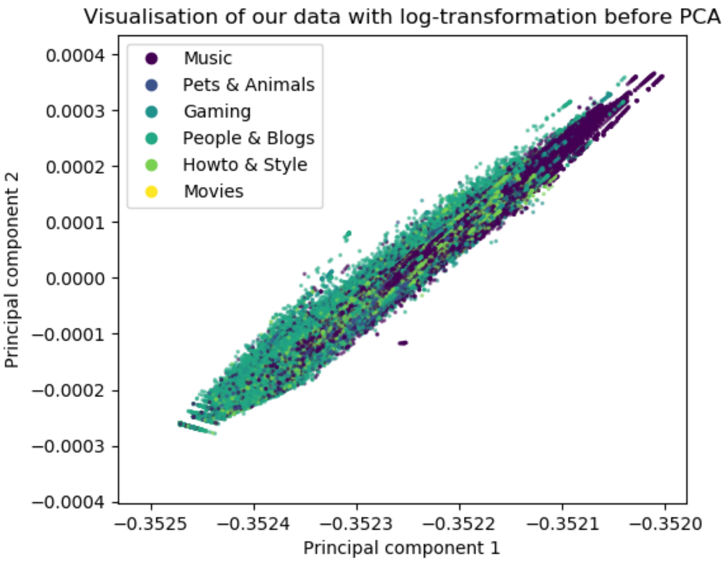

Figure 3

When looking at the variance explained in each case, the following results were obtained:

Principal component	Variance explained without log-transformed data	Variance explained with log transformed data
0	0.738275	0.311750
1	0.121569	0.609478
2	0.119819	0.077851
3	0.020336	0.000921

When looking at the explained variance, we see that the explained variance is better for the original data, when only looking at the first principal component, which means that if we were to describe our data with only one dimension, we might consider using the regularly standardised data. However when looking at the first two components, the sum of the explained is higher for our log-transformed data, meaning we keep more of the information about the data, when we reduce the dimensions and this is therefore preferred to the regular standardisation. We found an anomaly in the explained variance, that we could not find an explanation for, as the second principal component explains more of the variance than the first component, when looking at the explained variance of the log-transformed data. This does not matter as we plot the first two components, but is strange.

3.3 Correlations

	likes	dislikes	views	comment count	trending time
likes	1.000	0.454	0.778	0.794	0.004
dislikes	0.454	1.000	0.422	0.705	0.001
views	0.778	0.422	1.000	0.510	0.009
comment count	0.794	0.705	0.510	1.000	0.000
trending time	0.004	0.001	0.009	0.000	1.000

Table 3: Table of correlations

Table 3 above shows the correlations between the different attributes. It is the co-variance matrix but standardised with the standard deviations of the attributes, in order to measure how the different attributes vary together, even though they have different scales. From the table we see some significant correlations between the comment count and likes with a correlation coefficient on 0.794. Likes and views have a high coefficent as well. Trending time has as close to zero correlation with the other variables as you can get and it once again becomes clear that this variable is sort of an "outlier-variable". The coefficients shows us that the data contains linear correlations in two dimension. This can help us to interpret how the data is correlated between dimensions in the principal components, because some of the attributes in the correlations indicates that there is redundancy in the data.

3.4 Interpretation of directions

To interpret the directions of our principal components, we take a look at our V-matrix. Which is a matrix containing the eigenvectors of $S = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}}$ is our transformed data, meaning the first column of $\tilde{\mathbf{X}}$ represents the amount of likes each video had:

$$V = \begin{bmatrix} 0.650021 & -0.333162 & 0.167872 & -0.662039 \\ 0.267784 & 0.731654 & 0.624623 & 0.053112 \\ 0.530499 & -0.373962 & 0.147145 & 0.746372 \\ 0.473640 & 0.462426 & -0.748341 & 0.042578 \end{bmatrix}$$

The first column of V is the eigenvector with the largest eigenvalue and therefore corresponds to the subspace of the first principal component. As all the values are positive, it means 'likes' is the variable that has the biggest impact on the first principle component, followed by 'views'. For our second principal component, we see that likes and views must have a negative value after being transformed in order for them to have a positive outcome. What stands out most about this matrix, is that dislikes has the lowest impact on the first principal component and the largest impact on the second. Based on this explanation, we could interpret the visualisation of the data in Figure 3 as many elements in 'Music' both contains many likes but also dislikes.

Whereas there is more elements from 'People&Blogs' that contains lower likes and a small tendency of more dislikes as well.

4 Discussion

Through the visualisation of our data on the principal components (see figure 3) as well as the summary statistics we are able to understand more about our data. Since we had a quite large data set and a fairly amount of different categories to plot, we probably expected to be able to see a bigger distribution of the categories along the principal components in order to interpret any trends or extremes between the categories. Instead we get that they all spread out in the same way, though with a little tendency towards more data points in the category 'music' having more positive values than the rest of the categories.

So according to our visualisation, we are sceptical whether our primary machine learning method is feasible to our data set, as there is little seperation between the colors (categories) in our PCA visualisation. We do however hope that this little seperation will be enough to classify some of the videos and a solution might be to start by only using the categories which are far apart for a better result.

The reason to this distribution of data points could be that videos in general are trending because different compositions of the attributes, and they also trend of different reasons in-between the categories.

However, we have still learned a lot about our data set. One of our biggest problems was that our data followed a log-normal distribution. This may be due to some videos being published much earlier than others as youtube is being used more and more, making videos trend faster. In the next report this may be an interesting thing to explore, as the elder videos might cluster together or maybe there's another reason for the weird distribution of our data

5 Appendix

5.1 Contributions

Part 1	
Gabriella	30%
Gustav	40%
Lukas	30%
Part 2	
Gabriella	25%
Gustav	50%
Lukas	25%
Part 3	
Gabriella	20%
Gustav	20%
Lukas	60%
Discussion	
Gabriella	60%
Gustav	20%
Lukas	20%

Table 4: Contribution Table

Table 4 shows who made which contributions. This is however a rough estimate as we all collaberated to all the sections and most of the programming and writting was done with collaberation tools such as LiveShare in Visual Studio Code and overleaf.com

5.2 Code

The following scripts were used for the project. To see the entire repository go to `https://github.com/s183920/02450_intro_to_ML_project`

- Script to load data
- Script to find the category labels
- Script to clean the data set
- Script to make summary statistics and histograms
- Script to make PCA