CMPE 255 - Data Mining

YouTube Data Classification and Prediction

Team 10

SAN JOSÉ STATE
UNIVERSITY

Submitted to

Dr. David C. Anastasiu

on

08/01/2019

by

Nitish Joshi	013736320	nitish.joshi@sjsu.edu
Rohan Kamat	013759252	rohansantosh.kamat@sjsu.edu
Chaitanya Krishna Kasaraneni	013772642	chaitanyakrishna.kasaraneni@sjsu.edu

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Motivation

In today's world, the use of YouTube has increased a lot. People all over the world continuosly watch, upload and share videos on YouTube. It is a very popular website that shows videos ranging from all the genres and all the parts of the world.All kinds of data posses certain pattern, in this particular dataset , the patterns reflected user behaviour and ideology. For this reason, we were intrigued to work on a YouTube dataset and apply our knowledge of data mining.

1.2 Objectives

Our project objectives are focused on the classification and predictions of the YouTube videos based on the titles and description available in the dataset.
Below is the list of our objectives that we hope to achieve as part of our project implementation:

- Category prediction based on the title of a youtube video.
- Predicting the number of views (popularity) of a particular video given its title.
- Sentiment analysis of the description, tags and title.

Chapter 2: System Design and Architecture

2.1 Algorithms Selected

2.1.1 Prediction of Category based on the title of a youtube video:

1. Multinomial Naive Bayes Classifier
2. Support Vector Classifier
3. Random Forest Classifier
4. Decision Tree Classifier
5. K Neighbors Classifier

2.1.2 Predicting the popularity (views) of a particular video given its title:

1. Linear Regression
2. Random Forest Regressor
3. Gradient Boosting regressor
4. Ridge Regression
5. ElasticNet

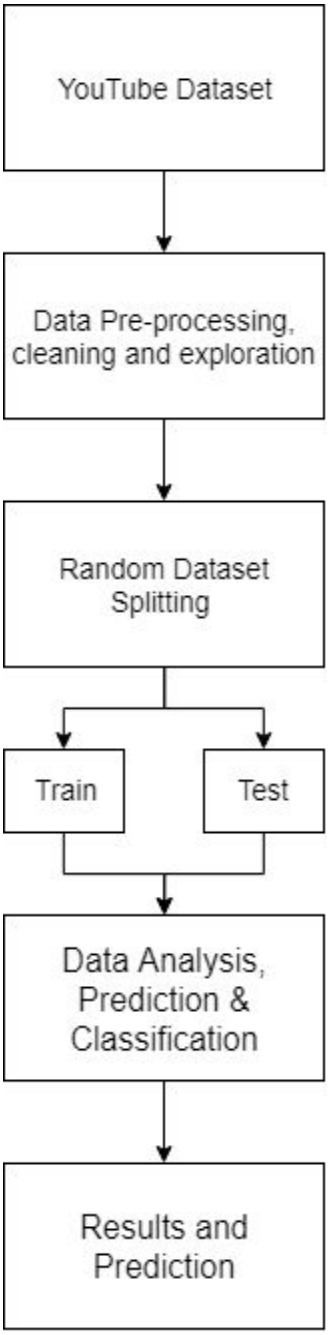
2.1.3 Sentiment Analysis of description, tags and title:

1. TextBlob
2. Support Vector Machine
3. Logistic Regression

2.2 Technologies,Tools and Libraries Used

1. Python 3
2. Jupyter Notebooks
3. Scikit-learn
4. NLTK
5. ML_Metrics(RMSE)
6. HPC
7. Pycharm

2.3 System Design and Architecture



Chapter 3: Experiments

3.1 Dataset

YouTube Video Dataset

The dataset includes 10 CSV files and 10 JSON files (200 MB approx).
This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, and FR regions (USA, Great Britain, Germany, Canada, and France, respectively), with up to 200 listed trending videos per day. The data also includes a category_id field, which varies between regions.

Description

1. CSV files - There are total 10 CSV files, with each file having data of a particular country. The data consists of video_id, title, category, views, likes, dislikes, description, etc. to name a few features.
2. JSON files - There are 10 JSON files having categories of each specific video.

3.2 Data Preprocessing

3.2.1 Data Preprocessing for Category Prediction

The data was read and only the relevant columns were kept for further processing, title and category_id in this case. After that, the data from the json file was read and a dictionary was created by mapping the category_id and the title and a dataframe was created to store it.

3.2.2 Data Preprocessing for Popularity Prediction

The data was first imported into a pandas dataframe a list was created for both title and views (through main dataframe).To make the code more interpretable, each preprocessing method was written as a definition and data was passed through it. The preprocessing steps involved were as follows in given particular order.

1. Lowercase: All the titles were first converted to lowercase strings.
2. Removing numeric values: The titles containing numeric values, were filtered out, this was done using regex [“\d”].
3. Removing Links: It was observed that there were some bizzare titles have links to other pages as well.A new regex[‘http[s]?://\S+'] for links was created to filter out such abnormalities.
4. Removing Punctuations and symbols: A string was generated which contained all the symbols and punctuations that were to be eliminated. The entire dataset was iterated through to check for the former.
5. Stemming:Porter stemmer was used to get the root words for the given dataset.

6. Stop Words: NLTK library was used to eliminate all the stop words from data, and all the words were tokenized.

3.2.3 Data Preprocessing for Sentiment Analysis

The data was first imported into pandas dataframe.

1. Removing redundant data: Since, a video could be in trending for several days, there might be multiple rows for a particular video. In order to calculate the total views, comments, likes, dislikes of a video, the videos were grouped by the video_id. To get the numbers of videos on which the 'Comments Disabled/ Rating Disabled/Video Error'. We need to remove the duplicates to get the correct numbers otherwise there will be redundancy.
2. Removing punctuation: re.sub('[^A-Za-z]+', ' ', a) was used to remove the punctuations in the dataset.

3.3 Methodology

3.3.1 Category Prediction

The csv data was read and only the necessary columns were kept (title and category in this case). The next step was to import the json data which consisted of the category of each video. This data was read and a new dictionary was created for mapping the category ID and the category of each video. Using this dictionary, a dataframe was created to store this data.

For training the model, the title was split into a string of words using CountVectorizer. After that, the classifier model was chosen for training the model by targeting the category values. The dataset was split randomly into 90/10 split to calculate the accuracy of the model.

We used five classifiers to predict the category of the video. Out of the five classifiers, we evaluated each of the classifier using the accuracy_score metric from sklearn.metrics. After evaluation, we got the best accuracy by using Random Forest and Decision Tree Classifiers, with an accuracy score of 98.7% and 98.8% respectively.

Finally, to test the prediction model, few hypothetical titles were created to predict their category. These titles were inserted into the classification model. The output was an array of numbers and we had to iterate through the category dictionary that was created earlier from the json file to find the title. The next step was to map these values to the titles that we wanted to predict and then converting the resulting dictionary into a dataframe.

3.3.2 Popularity Prediction

The first perspective was using description as the input parameter for predicting the views, the rmse value for the analyses was very high.The next approach was using titles, which gave a realization that titles played a very important role as the search query by a user hits the title. The first step was preprocessing which is described in the above section. A CSR matrix was generated using sklearn library which ensured minimal RAM is required for modelling phase.The approach for generating similarity matrix was same as done in the programming assignments, the index, value and pointer values were fed to sklearn csr library.
Evaluation:
Sklearn train_test_split library was used to split the data into test and train as (X_train,y_train) and (X_test,y_test), the test size was kept as 33% of the train, with a default random state of 42.Many different algorithms were tried such as Linear Regression, Lasso Regression, Ridge, ElasticNet, GradientBoostRegressor, RandomForest. The evaluation parameter chosen was RMSE and taken from ml_metrics library. The lowest RMSE was shown by linear regression 1.06321, rest all algorithms showed very high values ranging from 2-7000.

3.3.3 Sentiment Analysis

The csv data was read and the redundant data was removed or grouped wherever required. Then the trends were plotted like which video was trending for most of the time, which video took long to become a trending video and categories that were mostly in the trending. Then the frequently used words in the titles, tags and descriptions were found separately and word clouds were built. Then the categorization of the tags, titles and description into positive negative and neutral was done using textblob. As there was no certain metric to evaluate the performance of textblob, it was set as the baseline and the SVM classifier was applied on the description, tags, title and the sentiment and it was evaluated using the classification_report in the scikit-learn library. Also used linear regression and accuracy was considered as the evaluation metric. Marginally, SVM performed better than Logistic Regression. Tried applying CNN and RNN on the dataset which were giving Memory Error on the local system.

Chapter 4: Discussions and Conclusions

4.1. Decisions Made

- The team discussed and researched about selecting the dataset that met all the criteria necessary for the project implementation.
- After finalizing the dataset, the team decided and narrowed down various tasks that were to be implemented during the project lifecycle.
- The team also made decisions regarding various preprocessing steps and data cleaning strategies to be used such as different data wrangling techniques, using stopwords, etc. in order to achieve the best possible results.
- Finally, the team discussed and made decisions about various algorithms to be used for each task ranging from various types of classifiers to be used for category prediction to achieve better accuracy, selecting different regression algorithms for popularity prediction, etc.

4.2. Difficulties Faced

- The dataset that the team selected was very big in size and hence, needed a lot of time and effort to perform data cleaning, transformation and preprocessing steps on it.
- Since the number of rows was very large, the amount of memory required for computation was huge and hence, the data had to be processed step by step in limited chunks.
- The team also faced difficulties in implementing algorithms and obtain the best possible results considering all the factors such as the complexity of data, the volume of data, a large number of features, etc.
- The code testing was done by using chunks, but evaluation of algorithms required entire data, as limited RAM was available locally and during HPC inaccessibility scenarios, all applications on local PC were shut and code was run on cmd.

4.3. Things that worked well

- The dataset selection was ideal for us in a way that since it was huge, we got to work and experience the complexity of handling such a huge dataset. Although initially, it looked like a daunting task, the team did their research and found out ways to handle and work with it.
- Selection of data cleaning and preprocessing tasks worked really well as a result of which we got the best out of our algorithms.

4.4 Things that didn't work well

- The first attempt was to use description as a relevant feature, this failed as it was giving very high RMSE value.
- Dimensionality reducing was tried out using PCA, but it deteriorated the RMSE
- There were no particular metrics to evaluate the sentiment analysis using textblob.
- Tried applying CNN and RNN on the dataset for sentiment analysis which were giving Memory Error on the local system.

4.5 Conclusion

- While implementing this project, we learned about a wide array of techniques, algorithms, and different preprocessing tasks involved in data analysis and prediction and how they affect the performance of the algorithms as a whole.
- We learned how to handle a large imbalance dataset.
- We learned how to apply various algorithms and use predictions models.
- We learned that the output of a model varies based on the requirement of the predictions.

Chapter 5: Project Plan/Task Distribution

Task	Responsibility
Dataset Selection	All
Data Exploration and Cleaning	All
Data Preprocessing	All
Research on Algorithms	All
Category prediction of a youtube video	Nitish
Predicting Views(Popularity) of a youtube video.	Rohan
Sentiment Analysis of Description, Tags and Titles	Chaitanya Krishna
Documentation and Report	All
PPT	All

References:

1. https://www.kaggle.com/datasnaek/youtube-new
2. https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
3. https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html#exercise-2-sentiment-analysis-on-movie-reviews
4. https://regexr.com/
5. https://www.kaggle.com/yanpapadakis/trending-youtube-video-metadata-analysis