A Report

on

"Youtube Data Analysis"

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

by

Neha Jaju 33224

Shyamal Khachane 33227

Manvi Pandya 33235

Gauri Nandkhedkar 33239

Under the guidance of

Mrs. D.D.Londhe

Department Of Information Technology

CERTIFICATE

This is to certify that the project report entitled

"Youtube Trending Data Analysis"

Submitted by

Neha Jaju 33224

Manvi Pandya 33235

Shyamal Khachane 33227

Gauri Nandkhedkar 33239

Is a bonafide work carried out by them under the supervision of Prof. D. D. Londhe and it is approved for the partial fulfilment of the requirement of Software Laboratory Course – 2015 for the award of the Degree of Bachelor Of Engineering (Information Technology)

Prof. D. D. Londhe
Internal Guide
Department of Information Technology

Dr. A. M. Bagade
Head of Department
Department of Information Technology

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this project. A special gratitude to our guide, Prof. D. D. Londhe whose contribution of stimulating suggestions and encouragement helped us to coordinate our project, “Youtube Trending Videos Analysis”. We would like to thank Prof. R. V. Kulkarni and Prof. D. D. Londhe for providing very valuable and timely suggestions and help in the entire review. We would also like the entire project staff team for providing valuable reviews and suggestions from time to time. Also, without the efforts of my team members, their amazing ideas for this project, their hard work, dedication and valuable support, this project wouldn’t be possible.

Neha Jaju

Manvi Pandya

Shyamal Khachane

Gauri Nandkhedkar

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

YouTube is the most popular and most used video platform in the world today. Here we will use Python with some packages like Pandas and Matplotlib to analyze datasets . We will analyze this data to get insights into YouTube trending videos, to see what is common between these videos. Those insights might also be used by people who want to increase the popularity of their videos on YouTube.

The dataset that we used is obtained from Kaggle. It contains data about trending videos for many countries. Here we will analyze a dataset containing 5 csv files and 5 json files(for 5 different countries), including various kinds of information like video titles, channels, video categories, publish time, number of views, number of likes and dislikes, etc.

1.2 Motivation behind project topic

Nowadays, an increasing number of people post their videos on Youtube, and it is interesting to know whether a video is popular is dependent on its category and the cultural background of viewers. We plan to use the acquired dataset to analyze the composition and popularity associated with different factors of online videos on YouTube. We'd like to dig in deeper to elaborate the relationship between them. To be more specific, some videos are highly controversial because of its content or types. Also, we want to show how the cultural divergence affects people's likes and the overall most popular video types to shed light on how YouTubers are supposed to refine their videos to get more subscribers, and recommend popular channels of particular video genres.

1.3 Aim and Objective(s) of the work

1.3.1 Project aims

The aim of this project is to make YouTubers refine their videos to get more subscribers by choosing the correct content which users want depending upon different parameters in different regions with different cultures.

1.3.2 Project objectives

- To sum up all views of all videos and sort in descending order to know what category of videos is popular.

- To analyze the published time of all videos and sort in 24 hours to understand when is popular hour to publish videos.
- To utilize the number of comments and views to calculate how people interact with the youtuber.
- To count the number of trending videos of different types at different times, with which we can see the fluctuation.
- To show the popularity of particular video types by examining the total views of different video types.
- To evaluate how these attributes are associated with cultures in different countries.
- To calculate the ratio of likes to dislikes for every channel to show the viewers' attitudes towards them

2. Literature Survey

In "International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 6, June 2018" by Dayananda Sagar University, Bangalore, India the analysis is done using users Sentiments features such as Views, Comments, Likes, and Dislikes.The paper deals with analysis of YouTube Data. This approach uses several steps to identify the optimized algorithm for classification of YouTube Data.

The following are listed below:

• Fetching YouTube Data Using Google API
• Storing Data to Hadoop Distributed File System (HDFS)
• Indexing Data
• Cleaning Data
• Creating Model for Data
• Analysis of Data
• Training Data
• Classification of Data by selecting best Fitting Algorithm for Social media Analysis: YouTube

RESULTS

- Views, Likes, Dislikes, Comments are correlated with 85% approximately.
- They also found Top 10 YouTube channels and Top 10 YouTube categories
- They used the Linear Regression classification approach to classify the YouTube Data. The results given are accurate which shows that it is good practice for the social business.

3. Design and Implementation

3.1 System Architecture

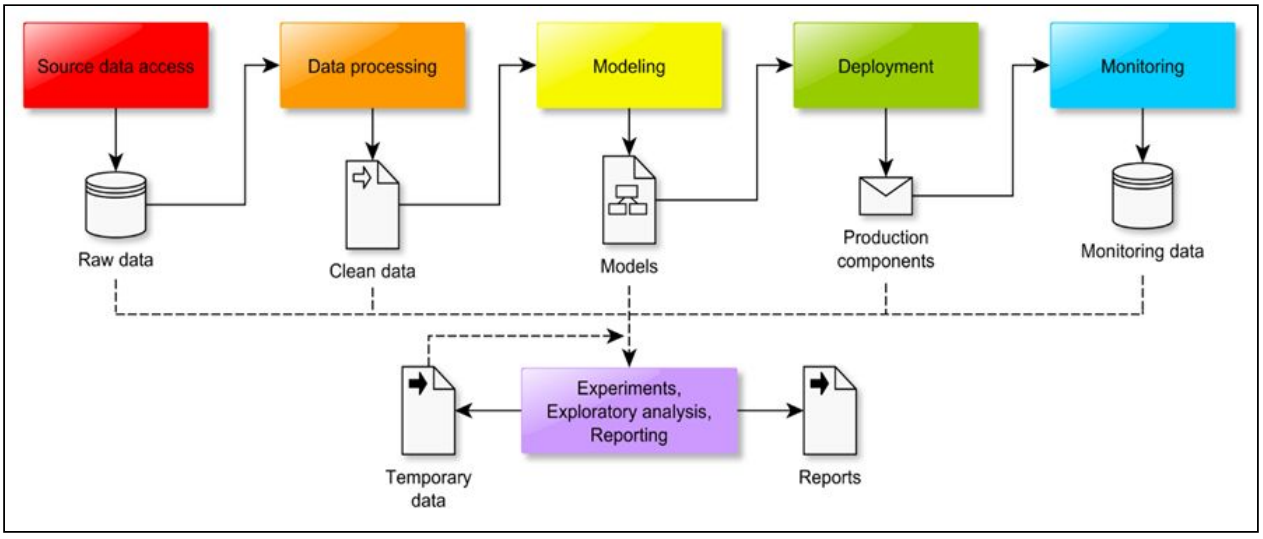


Figure 1: System Architecture

3.2 Dataset Description

The dataset that we have used is obtained from Kaggle.
It contains data about trending videos for many countries. Here we will analyze a dataset containing 5 csv files and 5 json files(for 5 different countries). The datasets that we have used are CAvideos.csv, DEvideos.csv, FRvideos.csv, GBvideos.csv, USvideos.csv and the corresponding json files too.

1. CAvideos.csv - Dataset for Canada
2. DEvideos.csv - Dataset for Germany
3. FRvideos.csv - Dataset for France
4. GBvideos.csv - Dataset for Great Britain
5. USvideos.csv - Dataset for USA

Link for the dataset : https://www.kaggle.com/datasnaek/youtube-new#CAvideos.csv

The datasets consists of the following attributes:

Attribute	Description
video_id	id of the video
trending_date	trending date of the video
title	title of the video
channel_title	name of the channel that has published the video
category_id	id of the category video falls under
publish_time	time at which the video was published
tags	tags added for the video
views	number of views the video has
likes	number of likes the video has
dislikes	number of dislikes the video has
comment_count	number of comments the video has
thumbnail_link	links to the thumbnail in the video
comments_disabled	TRUE/FALSE indicating whether the comments are disabled
ratings_disabled	TRUE/FALSE indicating whether the ratings are disabled
video_error_or_removed	TRUE/FALSE indicating if the video has error/is removed
description	description of the video

Table 1 : Attribute Information

3.3 Phases

ETL : Extract Transform Load

This phase consists of the first interaction with data. ETL stands for Extract Transform and Load. It's a generic process in which data is firstly acquired, then changed or processed and is finally loaded into a data warehouse or databases or other files such as PDF, Excel. Data first needs to be extracted from source, and then transformed in a form which can be

processed by the computer. FInally, transformed data is loaded into the target data source or data warehouse.

Preprocessing

Preprocessing consists of the actions to be performed on loaded data, to make it usable and more meaningful, to further learning or analytical pipelines such as cleaning and error removal. Data may consist of errors , missing values generated by incorrect observations, incorrect storage. Inconsistent data or data with missing values may simply be excluded to prevent it from impacting learning pipelines. However, Random Forest, null values are not allowed. Also, when the size of the dataset is small, we need to replace values by averaging other observations.

Data Model Planning

Data model planning is the phase of choosing the right kind of data models which can well represent the features of the data set at hand. This requires statistical and mathematical expertise. Choosing the right kind of the data model ensures maximum efficiency and performance. Data model planning also involves interpreting the right data representation, suitable to the chosen model.

Feature Selection

Choosing the right features for a data model is essential. The right features help represent maximum relevant information from the data set. It helps the data model achieve maximum efficiency and performance. Various techniques like covariance and correlation matrix exist to identify the right features and their representation.

Data Model Building

Data model building of actually defining the appropriate data model, with specific structure. The defined model can then learn from the training data set and can be tested on the testing data set. This model can then be deployed for further use.

Visualization

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers of the images. This communication is achieved through the use of a systematic mapping between graphic marks and data values in the creation of the visualization.

3.4 ALGORITHMS

Regression algorithms attempt to estimate mapping function (f) from the input variable (x) to numerical or continuous output variable (y). Regression prediction problems are usually quantities or sizes. Hence our problem is a regression problem.

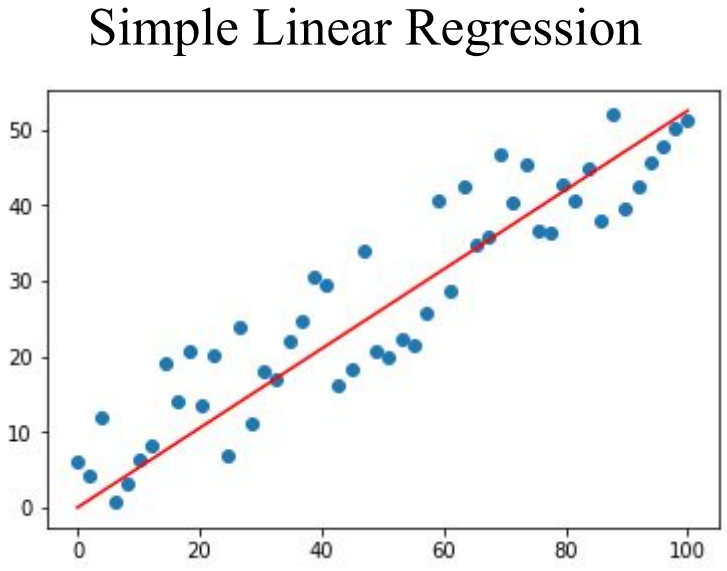


Figure 2 : Simple Linear Regression

- Simple linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable.
- The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.
- Using the training data, a regression line is obtained which will give minimum error. This linear equation is then used for any new data.

$$Y(pred) = b0 + b1*x$$

- The values b0 and b1 must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

Note : If we don't square the error, then positive and negative points will cancel out each other.

Random Forest Regression

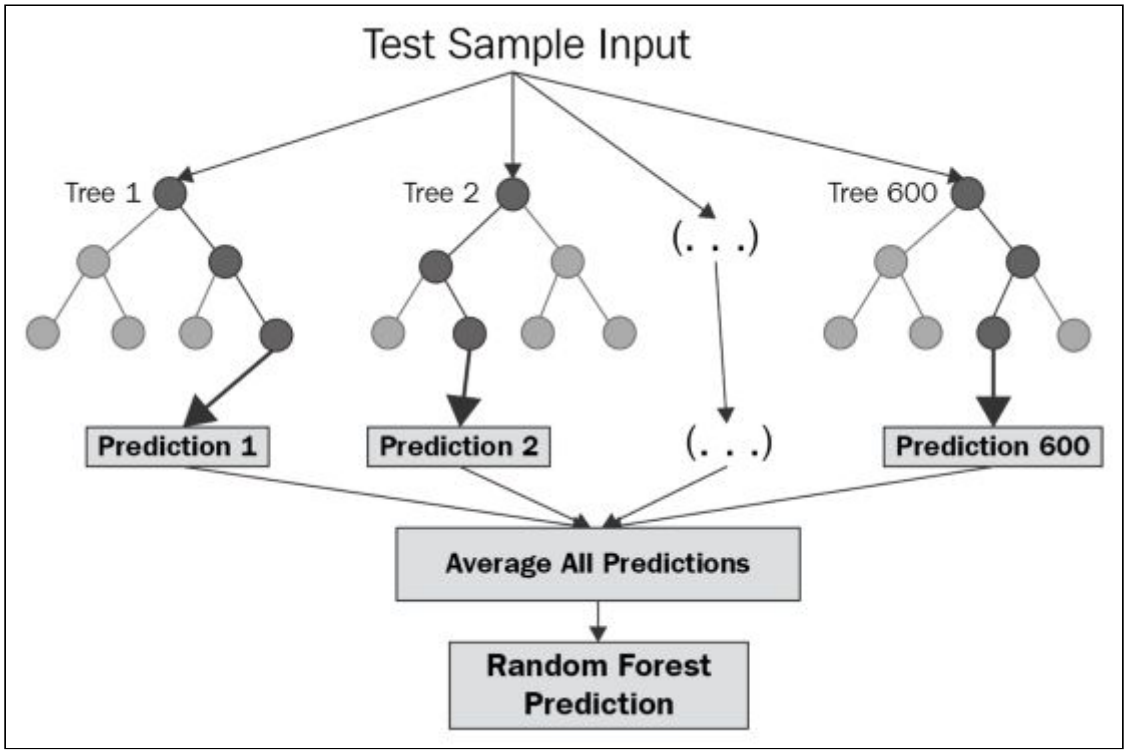


Figure 3 : Random Forest

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. It constructs decision trees and predicts outputs based on votes. A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which **aggregates many decision trees** with some modifications as :

1. The number of features that can be split on at each node is limited to some percentage of the total (which is known as the **hyperparameter**).

2. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents **overfitting**.

It better handles missing values and has the power to handle dataset with higher dimensions. It thus maintains accuracy of huge amounts of data and it predicts best in case of non-linear data.

4. RESULTS

4.1 VISUALIZATION:

4.1.1 Most Popular Video Category:

- On average,people irrespective of culture are highly interested in Music and then in Entertainment.
- On contrary when we see individually,following are the results:

1.Canada -> Entertainment
2.France -> Music
3.Germany -> Entertainment
4.Great British -> Music
5.USA ->Music

4.1.2 Fluctuating of publishing videos in 24 Hours

- This graph shows the best time to publish a video.
- On average,it is between 4pm to 5pm. While on the other hand individually, it is as follows:

1. Canada -> 4pm
2. France -> 5pm
3. Germany -> 4pm
4. Great British -> 5pm
5. USA -> 4pm

4.1.3 People Interaction with Youtubers

On average,Every 1 in 200 viewers writes down something under the video unless the video is related to Nonprofits & Activism ,the Comment/View Ratio of which is 2.5%.

4.1.4 Approval Rate of viewers on different categories

- Ratio of likes and dislikes as an indicator of audience approval rating for the video content.
- The approval rate varies among different genres for popular videos.
- Animal,Comedy and Education have the highest approval rate. Audience generally respond positively to these types of videos

4.1.5 Most controversial video categories.

- People tend to have divided opinions when viewing video of type News & Politics and Nonprofits & Activism
- Nonprofits & Activism, News & Politics and Entertainment are the 3 most controversial topics.

4.2 PREDICTION:

4.2.1 Views:

Model	Variance	Root mean square error
Linear Regression	0.75	1801874.0284891927
Random Forest	0.76	1777221.703304941

Table 2 : Views Result

Variance in case of views is high for Random Forest as it randomizes the data and hence can produce higher variance.

4.2.2 Likes:

Model	Variance	Root mean square error
Linear Regression	0.89	58052.46102613961
Random Forest	0.80	77261.00153942568

Table 3 : Likes Result

4.2.3 Count_comment:

Model	Variance	Root mean square error
Linear Regression	0.76	14297.715855479844
Random Forest	0.55	19781.825132439943

Table 4 : Comments Result

5. Conclusion

5.1 Tools Used

The various tools used for this project are as follows:

- Numpy
- Pandas
- Matplotlib
- Scikit Learn
- Seaborn
- Anaconda
- Python 3
- Python Virtual Environments

5.2 Conclusion

This project successfully implements supervised algorithms of Simple Linear Regression and Random Forest Regression on Trending Youtube Videos Data. Data was pre-processed followed by feature extraction and Visualization. We successfully visualized the data which answers the following questions (example : How many viewers tend to comment on videos, etc.) in the form of twinx graphs for better understanding. Prediction was performed for 20 percent of our dataset where regression was used and observations were recorded where we found that Random Forest worked better while predicting likes and comments as compared to linear regression, which performed better in case of prediction of views.

6. References

[1] International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 6, June 2018
[2] Kaggle
[3] https://towardsdatascience.com/
[4] Wikipedia

7. Appendix

Reading the Datasets

```
def __init__(self):
    '''
    Initialize a VideoManager instance.
    '''
    self.data_root = '../datasets/'
    self.countries = ['CA', 'DE', 'FR', 'GB', 'US']
    self.full_name = {'CA': 'Canada', 'DE': 'Germany',
                      'FR': 'France', 'GB': 'Great British', 'US': 'USA'}
    self.cat_dict = self.get_cat_dict()
```

```
    Load video data into self.dataframe and prepare for plotting.

    :return: None
    '''
    dataset = {}
    for country in self.countries:
        file_path = self.data_root + country + 'videos.csv'
        raw_df = pd.read_csv(file_path)
        dataset[country] = raw_df
```

Mapping categoryid from csv file to category_name in json file:

```
def get_cat_dict(self):
    '''
    Get the dict which maps category_id to category name.

    :return: dict
    '''
    cat_dict = {}
    for country in self.countries:
        with open(self.data_root + country + '_category_id.json', 'r') as file:
            json_obj = json.load(file)
        for item in json_obj['items']:
            cat_dict[int(item['id'])] = item['snippet']['title']
    return cat_dict
```

```
# Add one new column: `Categories`
result_view.insert(loc=1, column='Categories',
                   value=result_view.category_id.map(lambda x: self.cat_dict[x]))
```

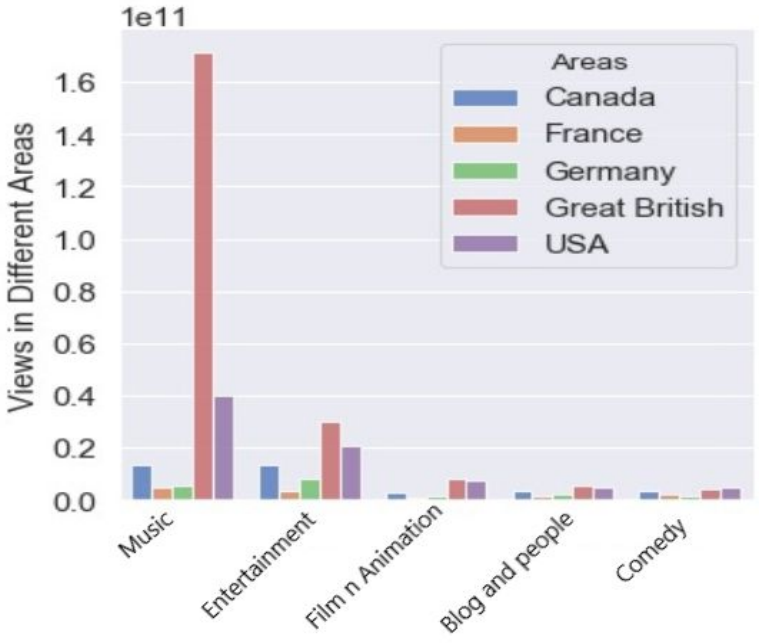
Finding Total Views in different countries in different categories

```
'''
Deal with views of different countries for top 5 categories.
'''
result_view = pd.DataFrame()
for area in self.countries:
    # read csv file and get specific column
    # Combine rows with same category_id
    df_view = pd.DataFrame(self.dataset[area][['category_id', 'views']]).groupby('category_id')[
        'views'].sum().rename('Views in Different Areas').reset_index()
    df_view['Areas'] = self.full_name[area]
    result_view = result_view.append(df_view)
```

```
# Sort total views in descending order.
result_view = result_view.sort_values(
    by='Total Views', ascending=False).reset_index(drop=True).reset_index()
```

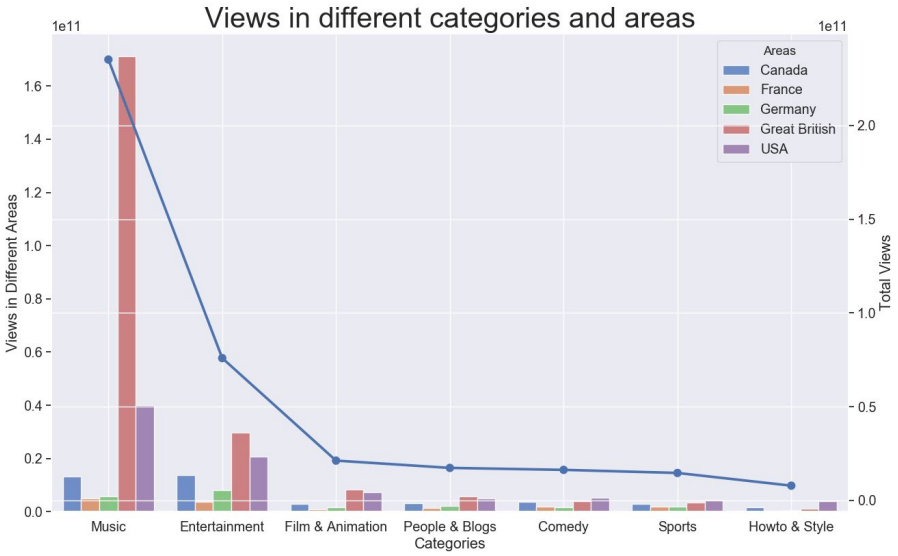
Histogram for views of different countries grouped by categories

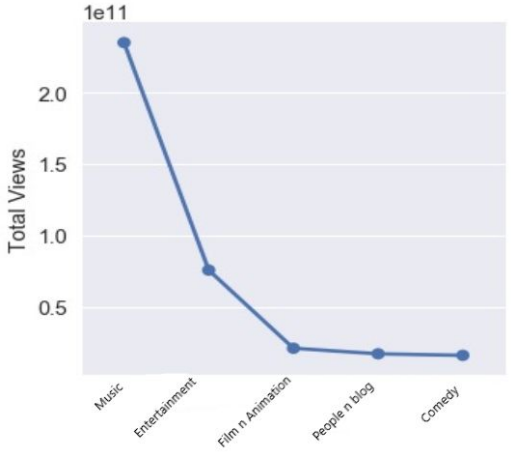
```
# Plot area views at sbplt_ax_view.
fig_view1 = sns.catplot(x="Categories", y="Views in Different Areas", data=result_view.head(25), hue="Areas",
                        hue_order=['Canada', 'France',
                                   'Germany', 'Great British', 'USA'],
                        kind="bar", palette="muted", edgecolor="1", alpha=0.85, legend_out=False,
                        ax=sbplt_ax_view)
```



Total Views grouped by categories (in form of point)

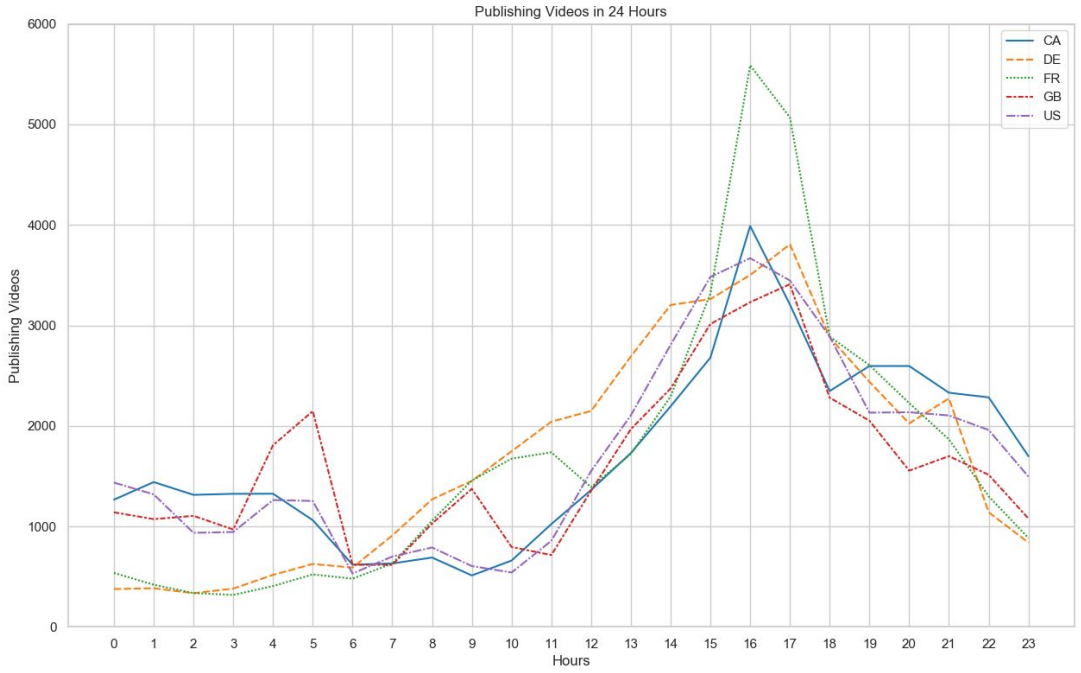
```
fig_view2 = sns.catplot(x="Categories", y="Total Views", data=result_view.head(25),
                        kind='point', color="b", ax=sbplt_ax_view2)
```





Show the fluctuation publishing videos in 24 Hours within 5 different area

```
'''
Show the fluctuation publishing videos in 24 Hours within 5 different area
'''
dic = {}
for e in self.dataset:
    # extract the hour from dataframe
    p_time = [int(x[11:13])
              for x in list(self.dataset[e].publish_time)]
    video_counter = [0] * 24
    for j in p_time:  # count the videos in each hour
        video_counter[j] += 1
    dic[e] = video_counter  # add the timetable into dictionary
frame = pd.DataFrame(dic)
sns.set(font_scale=1.5)
sns.set(style="whitegrid")
fig = sns.lineplot(data=frame, palette="tab10", linewidth=1.5)
fig.set_title('Publishing Videos in 24 Hours')
fig.set_xlabel('Hours')
fig.set_ylabel('Publishing Videos')
plt.xticks(np.arange(0, 24, 1))
plt.yticks(np.arange(0, 7000, 1000))
```



Analyze ratio of comment_count to views of all categories and for all countries

```
dataframe = pd.DataFrame()
for country in self.countries:
    raw_df = self.dataset[country][[
        'category_id', 'views', 'comment_count']]
    df = raw_df.groupby('category_id')[
        ['views', 'comment_count']].sum().reset_index()
    df['Comment/View Ratio'] = df['comment_count'] / df['views']
    df['Country'] = self.full_name[country]
    dataframe = dataframe.append(df)
```

Analyze ratio of comment_count to views of different countries of all categories.

```
dataframe['Total Ratio'] = dataframe.groupby(
    'category_id')['Comment/View Ratio'].transform('sum')
```

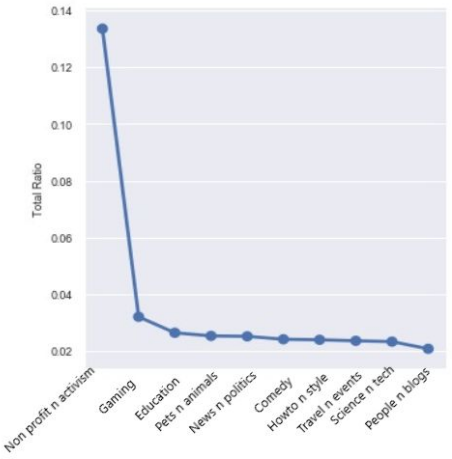
Histogram for Comment/View Ratio for different countries and for all categories

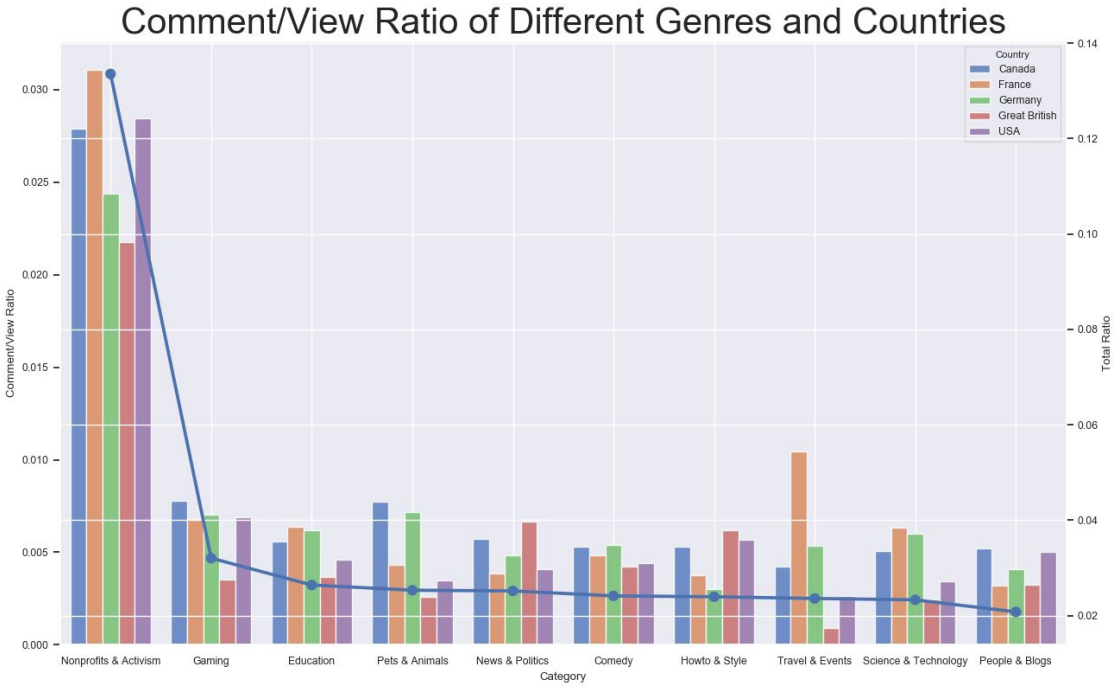
```
fig_area = sns.catplot(x="Category", y="Comment/View Ratio", data=dataframe.head(hist_count),
                       hue="Country",
                       hue_order=['Canada', 'France',
                                  'Germany', 'Great British', 'USA'],
                       kind="bar", palette="muted", edgecolor="1", alpha=0.85, legend_out=False, ax=ax1)
```

Point Representation of Total Ratio of Comment/View of different countries for all categories

```
fig_total = sns.catplot(x="Category", y="Total Ratio", data=dataframe.head(hist_count),
                        kind='point', color="b", ax=ax2)
```

```
dataframe = dataframe.sort_values(
    by='Total Ratio', ascending=False).reset_index(drop=True).reset_index()
```





To find Likes/Dislikes Ratio for all countries for all different countries

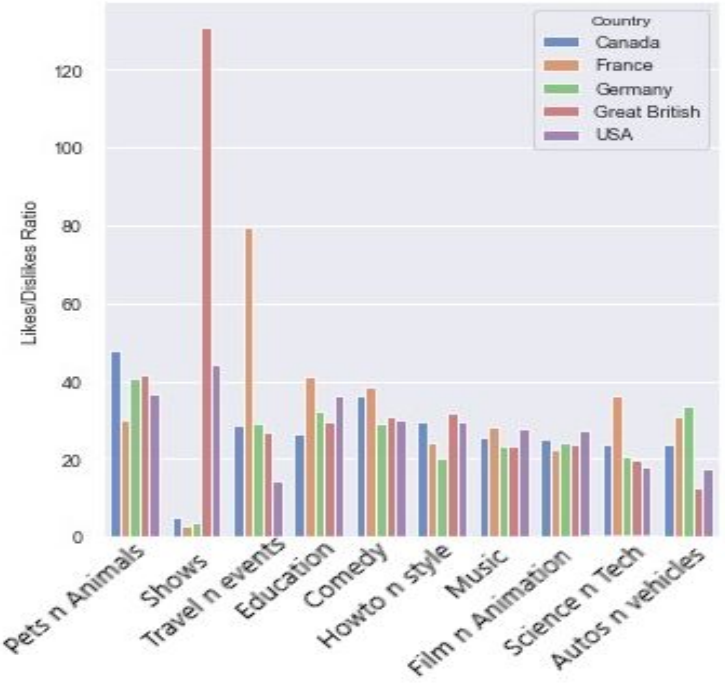
```
    '''
    Show video categories with the highest like/dislike ratio.
    '''
    dataframe = pd.DataFrame()
    for country in self.countries:
        raw_df = self.dataset[country][[
            'category_id', 'likes', 'dislikes']]
        df = raw_df.groupby('category_id')[
            ['likes', 'dislikes']].sum().reset_index()
        df['Likes/Dislikes Ratio'] = df['likes'] / df['dislikes']
        df['Country'] = self.full_name[country]
        dataframe = dataframe.append(df)
```

To find Ratio of Likes/Dislikes for different countries for all categories

```
dataframe['Average_Ratio'] = dataframe.groupby(
    'category_id')['Likes/Dislikes Ratio'].transform('sum') / 5
```

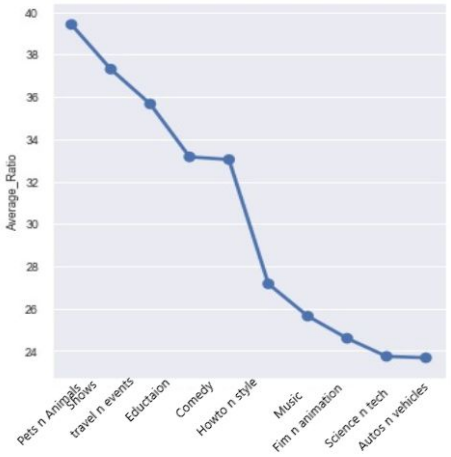
Histogram for Likes/Dislikes Ratio of different countries for all categories

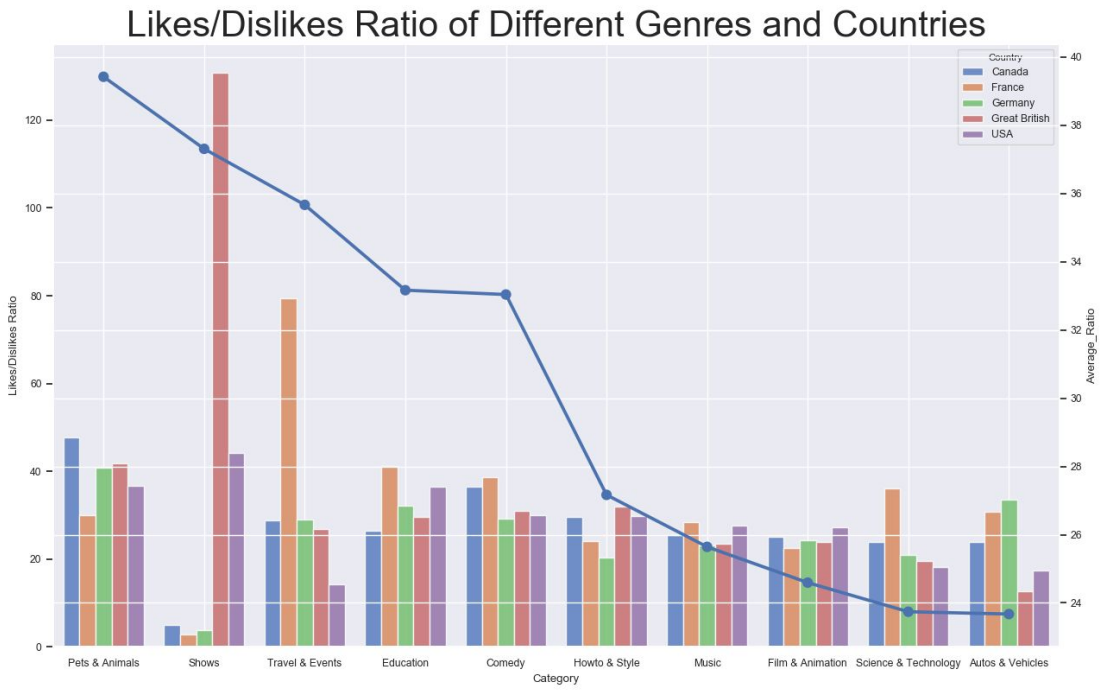
```
fig_area = sns.catplot(x="Category", y="Likes/Dislikes Ratio", data=dataframe.head(hist_count),
                       hue="Country",
                       hue_order=['Canada', 'France',
                                  'Germany', 'Great British', 'USA'],
                       kind="bar", palette="muted", edgecolor="1", alpha=0.85, legend_out=False, ax=ax1)
```



Point graph for Ratio of Likes/Dislikes for diff countries for all categories

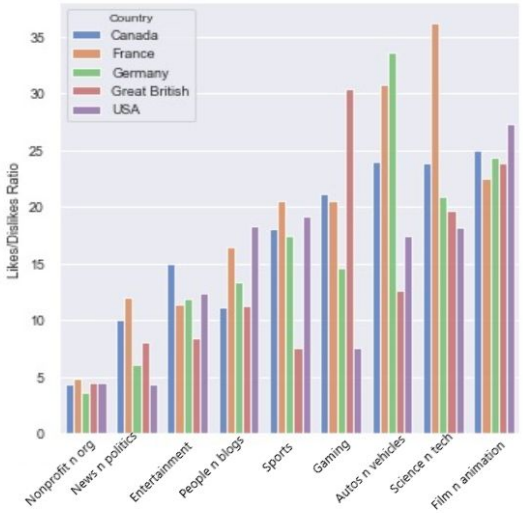
```
fig_total = sns.catplot(x="Category", y="Average_Ratio", data=dataframe.head(hist_count),
                        kind='point', color="b", ax=ax2)
```

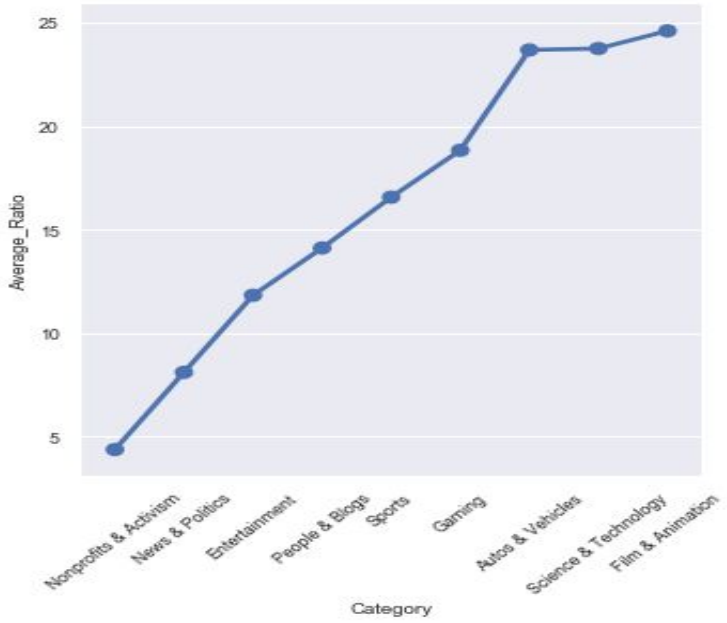


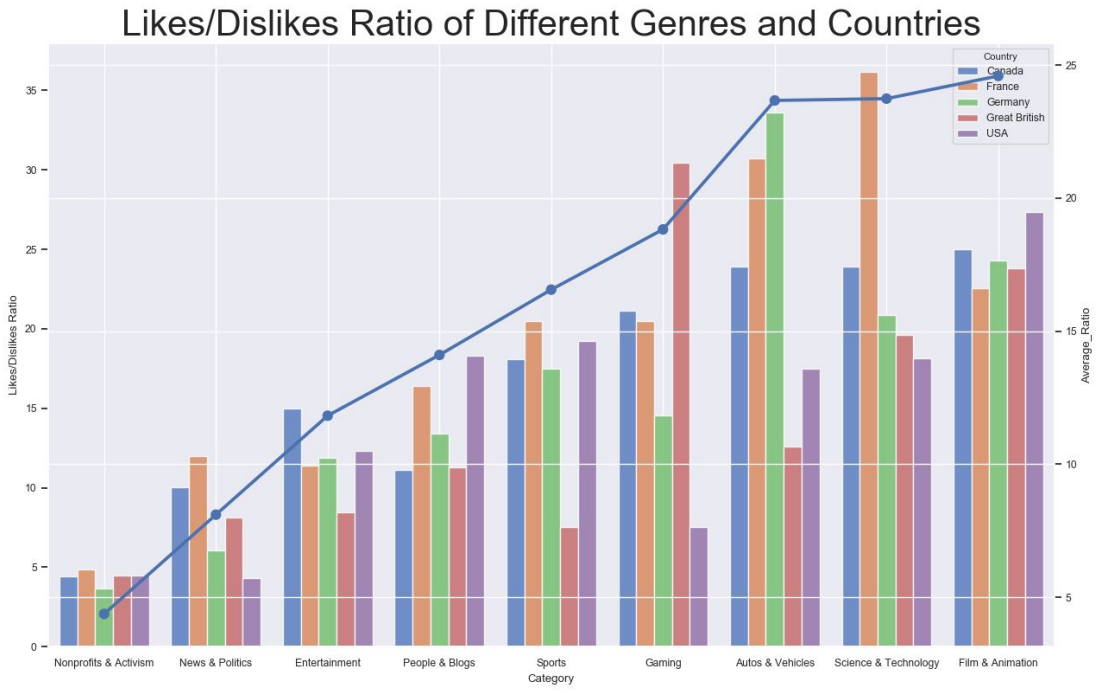


Show the most controversial video categories.

```
dataframe.insert(loc=1, column='Dif to One',
                 value=dataframe.Average_Ratio.map(lambda r: abs(r - 1)))
dataframe = dataframe.sort_values(
    by='Dif to One', ascending=True).reset_index(drop=True).reset_index()
```




25
20
15
10
5
Average_Ratio
Nonprofits & Activism
News & Politics
Entertainment
People & Blogs
Sports
Gaming
Autos & Vehicles
Science & Technology
Film & Animation
Category


Likes/Dislikes Ratio of Different Genres and Countries
Country
Canada
France
Germany
Great British
USA
Likes/Dislikes Ratio
Average_Ratio
Nonprofits & Activism
News & Politics
Entertainment
People & Blogs
Sports
Gaming
Autos & Vehicles
Science & Technology
Film & Animation
Category

PREDICTION

1. Prediction of views

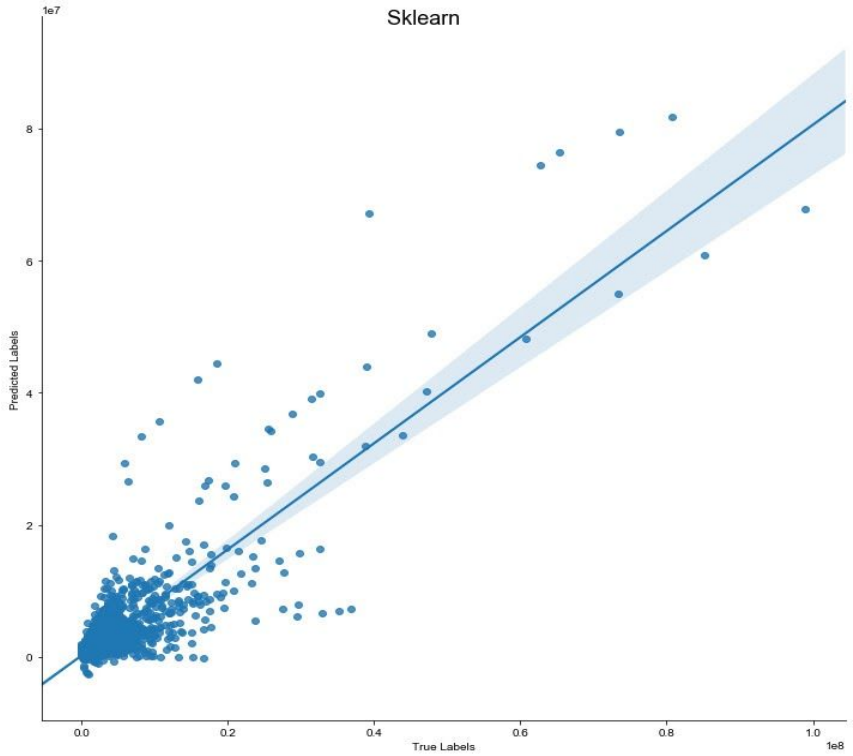


Fig. Views Linear Regression

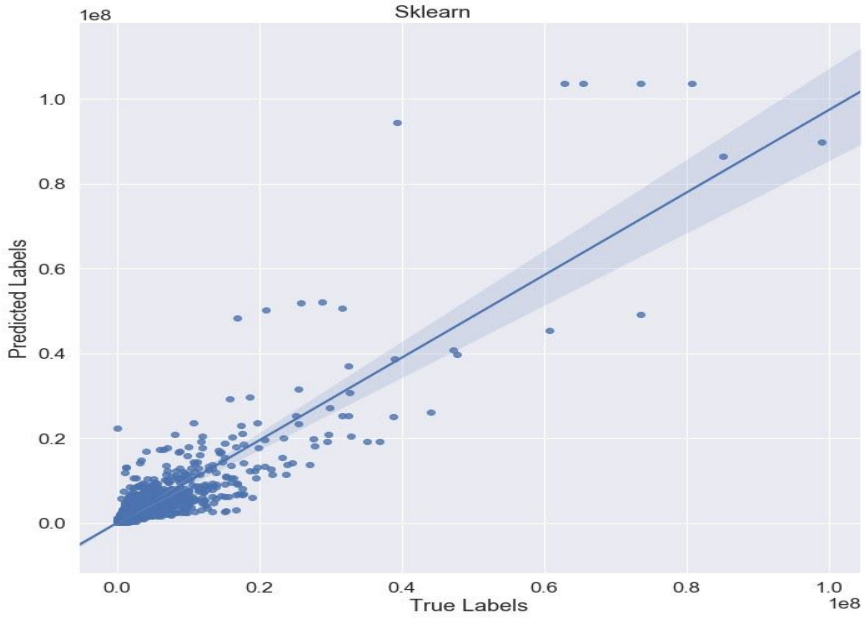


Fig. Views Random Forest

2. Prediction of Likes

Fig. Likes Linear Regression

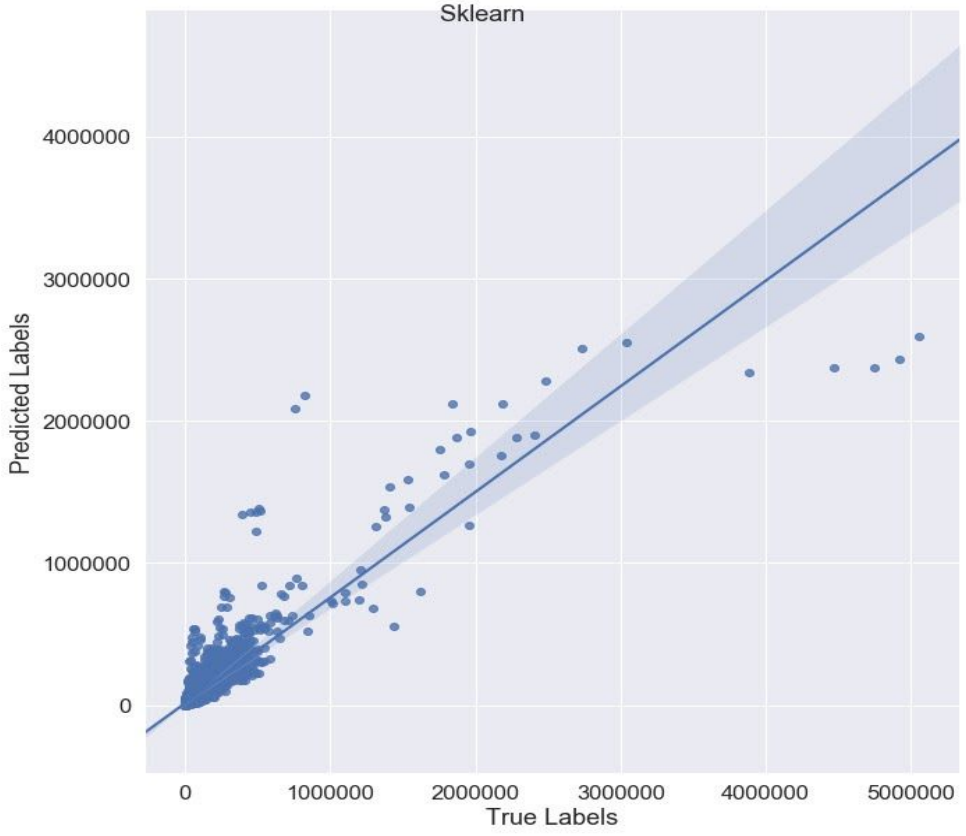


Fig. Likes Random Forest

3. Predictions of Comments

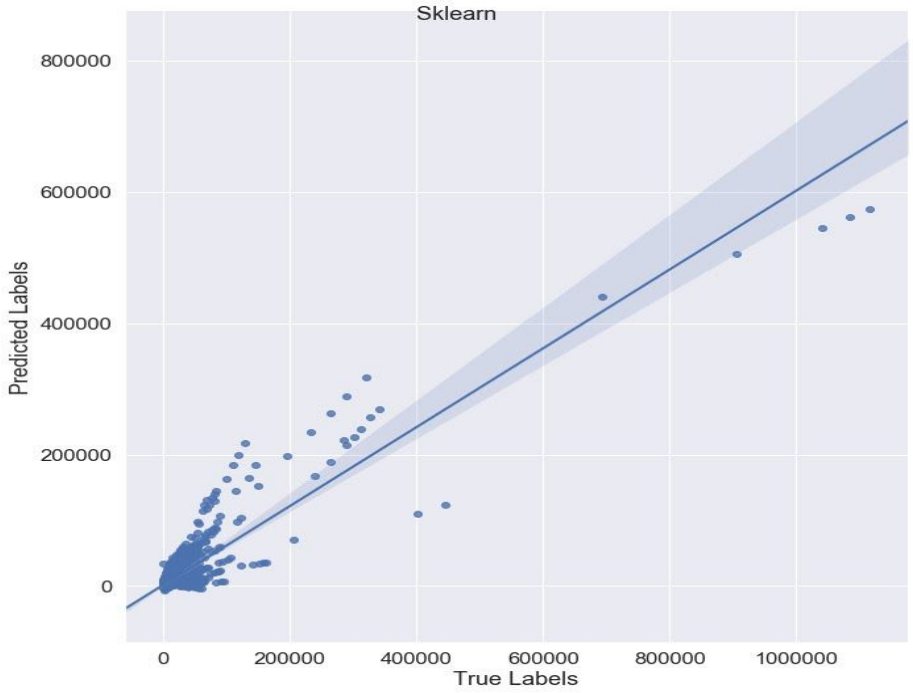


Fig. Comments Linear Regression

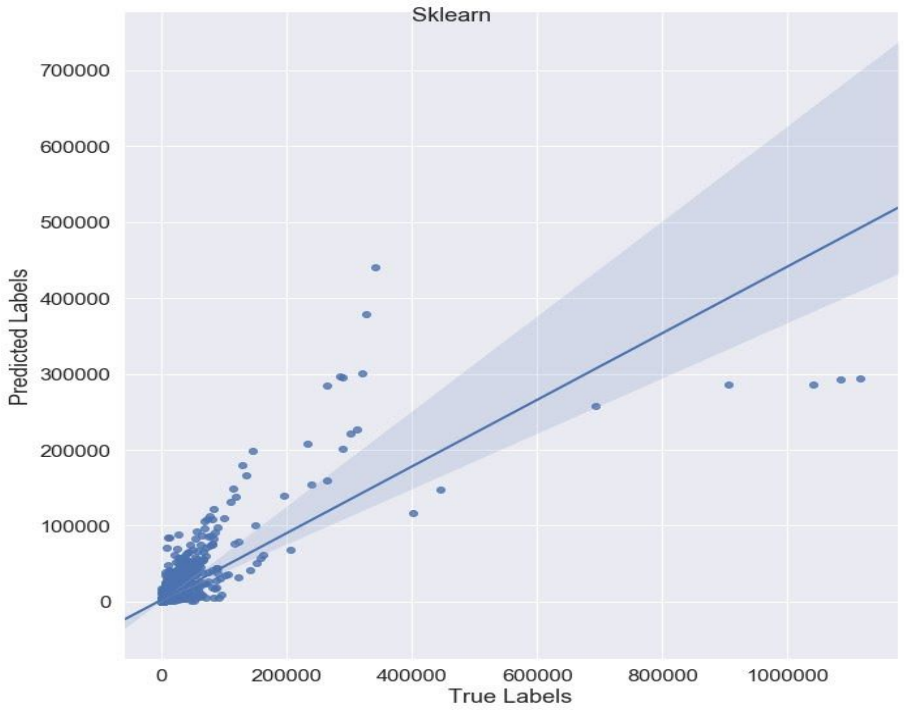


Fig. Comments Random Forest