FUNDAMENTALS OF DATA ANALYTICS: IMDB DATASET

FINAL PROJECT:

# NAME: PREETI RACHEL JASPER

REG NO: 17BCS0003

**CONTENTS:**

* + Identify the sample size
  + EDA (exploratory data analysis)
  + Cleaning data
  + Perform PCA
  + K means
  + DBSCAN
  + Random Forest Technique

EXPLORATORY DATA ANALYSIS:

## Hundreds of thousands of movies have been released since the ship set sail with the first ever movie in the late 1800s. Predicting the success of a movie is a complex task as various factors influence its performance in the box office. Since a huge capital is invested in the production, marketing, promotion and distribution of the movies, it has been a topic of interest not only for the viewers, but also for the media and production houses and all others who are involved in these processes. For the study, a part of the IMDB dataset will be used and scrutinized to get valuable information. In this digital age, online publicity plays a major role in the success of a movie. This is hence a field that would benefit from a study that helps them predict/ analyze the likelihood of the success of a movie based on the following attributes.

IMDB rates movies according to true Bayesian estimate. WR = (v ÷ (v+m)) × R + (m ÷ (v+m)) × C (1) where:

R = Mean Rating

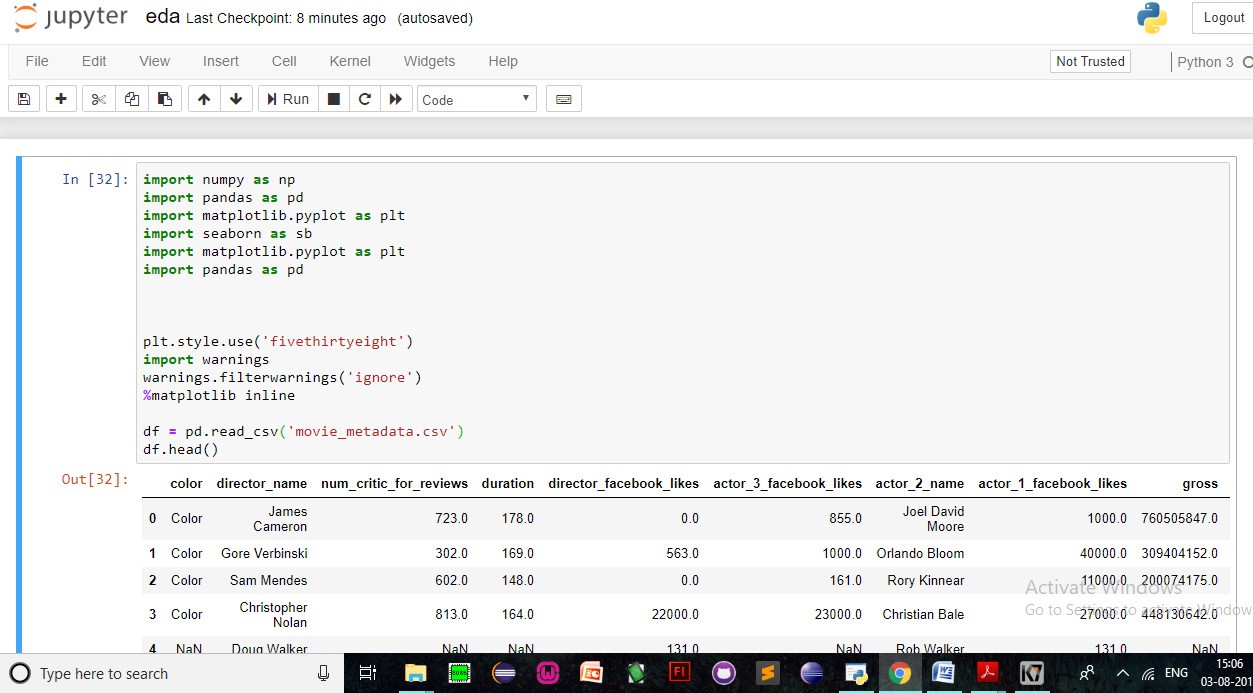
v = votes for the movie

m = minimum votes required to be listed in the Top 250 (currently 25000) C = the mean vote across the whole report (currently 7.0)

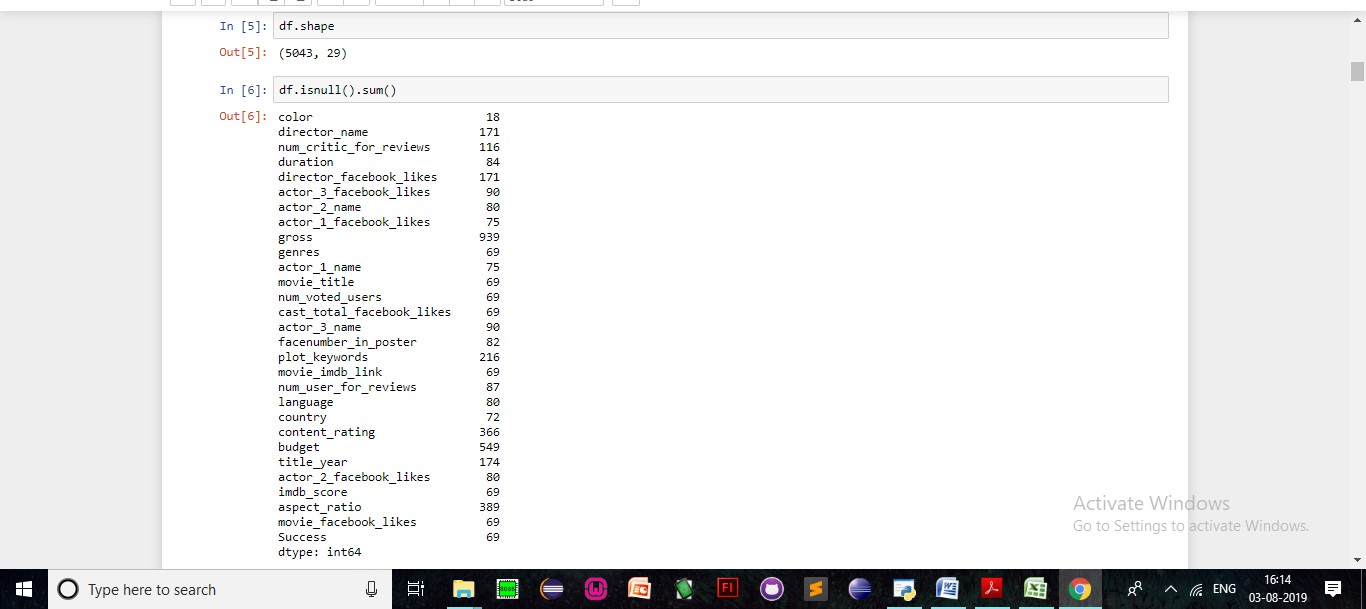
AVAILABLE ATTRIBUTES:

* color
* director\_name
* num\_critic\_for\_reviews
* duration
* director\_facebook\_likes
* actor\_3\_facebook\_likes
* actor\_2\_name
* actor\_1\_facebook\_likes
* gross
* genres
* actor\_1\_name
* movie\_title
* num\_voted\_users
* cast\_total\_facebook\_likes
* actor\_3\_name
* facenumber\_in\_poster
* plot\_keywords
* movie\_imdb\_link
* num\_user\_for\_reviews
* language
* country
* content\_rating
* budget
* title\_year
* actor\_2\_facebook\_likes
* imdb\_score
* aspect\_ratio
* movie\_facebook\_likes

# EDA using Python 3:

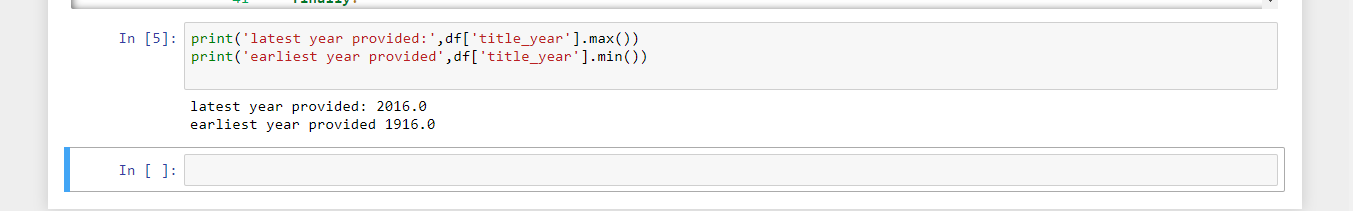


In order to normalize and clean the data, we must know the number of null values that have to be neglected or replaced, hence we apply this small piece of code:

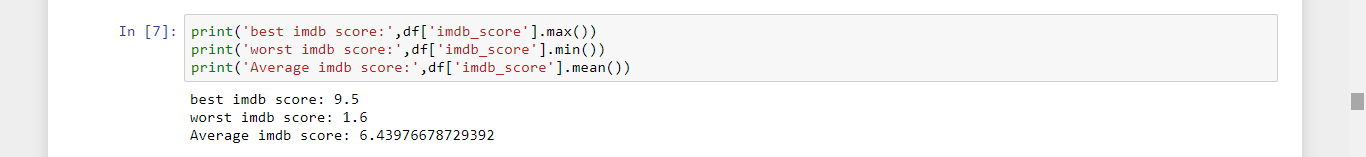


## AIM: To establish that certain attributes help the movie to do better.

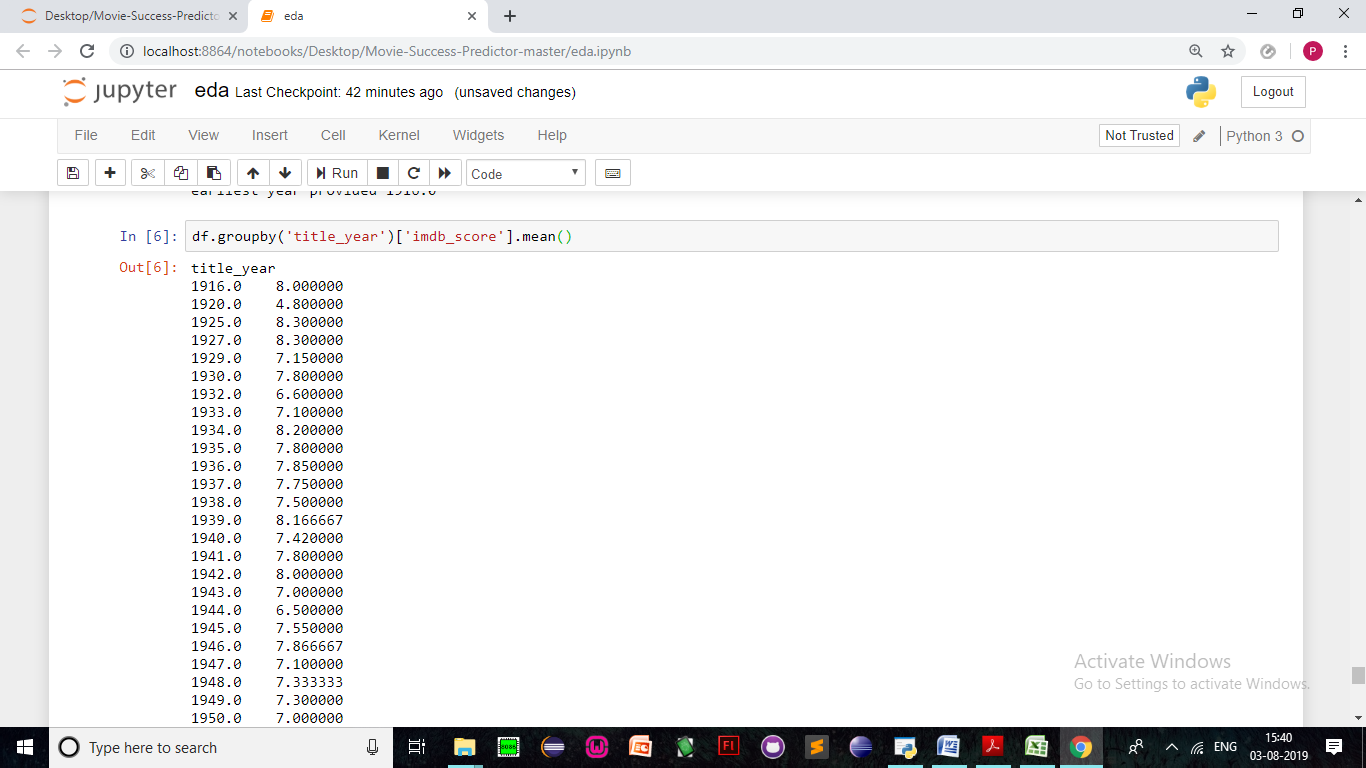
Let’s see the number of years of movies included in the list:

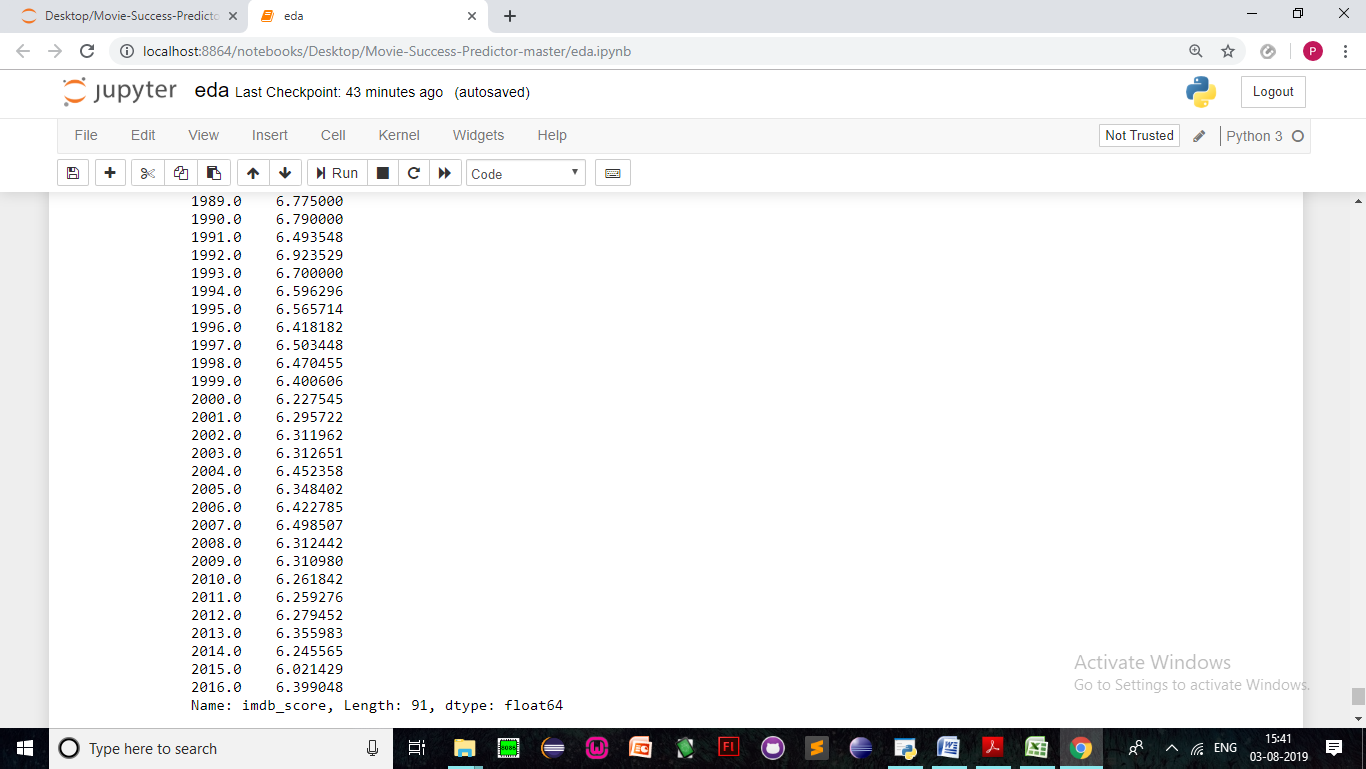


Let’s see the highest imdb score and the worst imdb score:



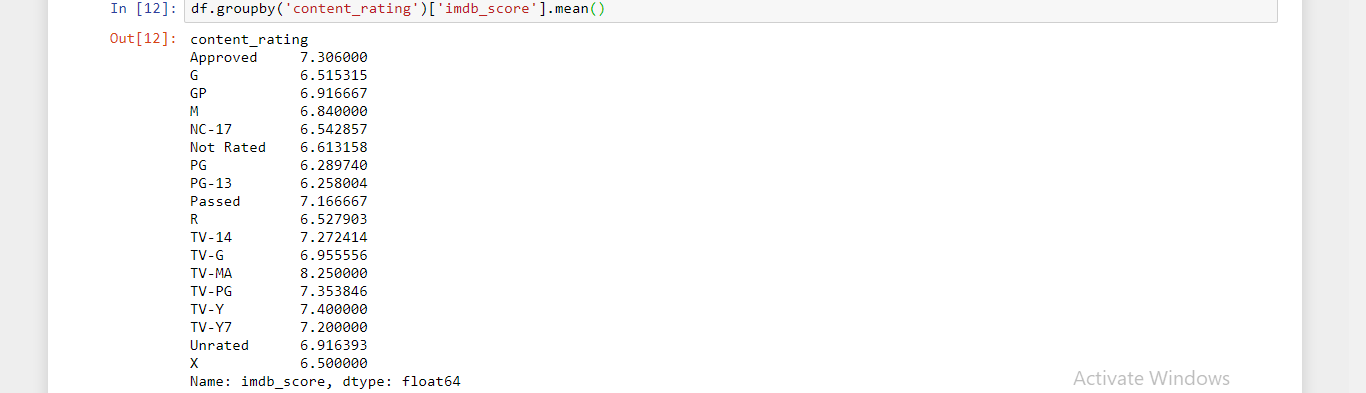
Now, lets see if theres any correlation between the title\_year and the corresponding imdb\_score :





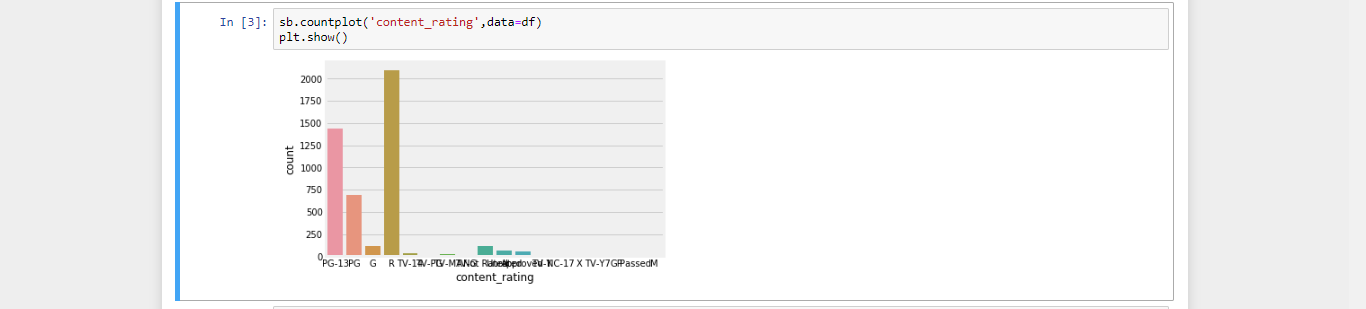
We can see that although the number of movies released at an earlier age is less compared to the number of movies released now, it is still apparent that the quality of movies has steadily but slowly gone down.

What about content rating? Do children’s movies or restricted movies do well , lets look at the mean Imdb rating for each category:



From this it is clear that TV-MA does the most well. TV-MA is a rating assigned by the TV Parental Guidelines to a television program that was designed for mature audiences only. Using this rating, the TVPG warns viewers that the show's content contains foul language, graphic violence, graphic sexual activity or any combination of the same.

However, most films are categorized under PG,PG-13,R and G .

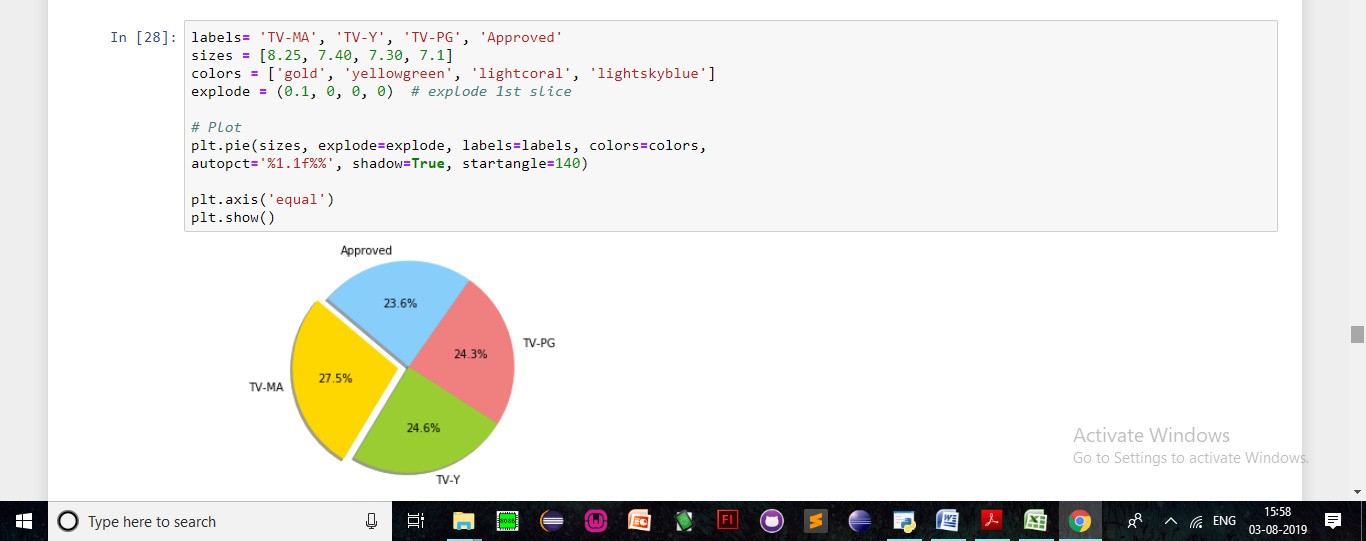


Therefore, from a wider perspective, directing a movie for a larger and wider audience will remain more profitable.

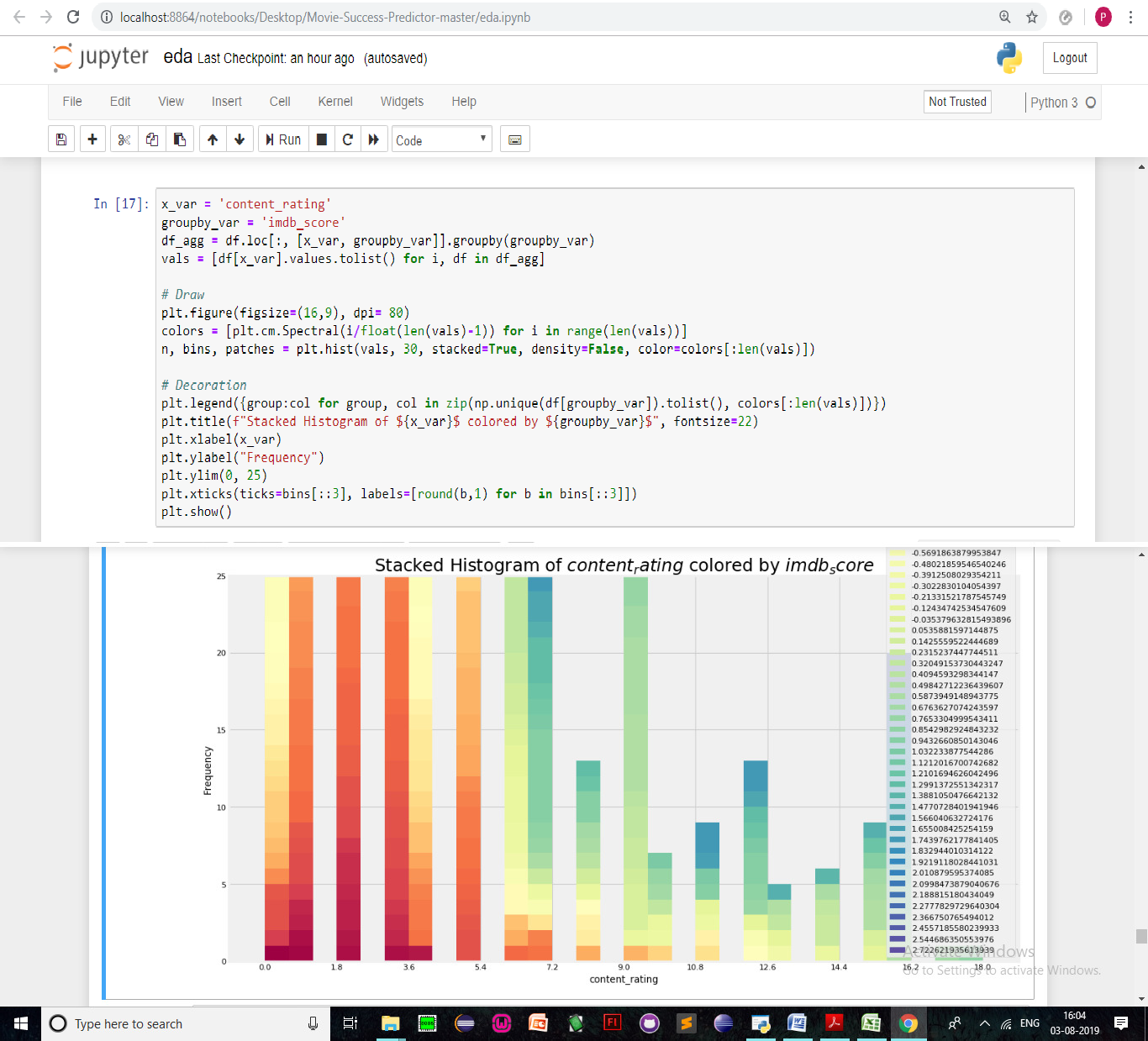
Provided below is a countsplot that showcases the same :



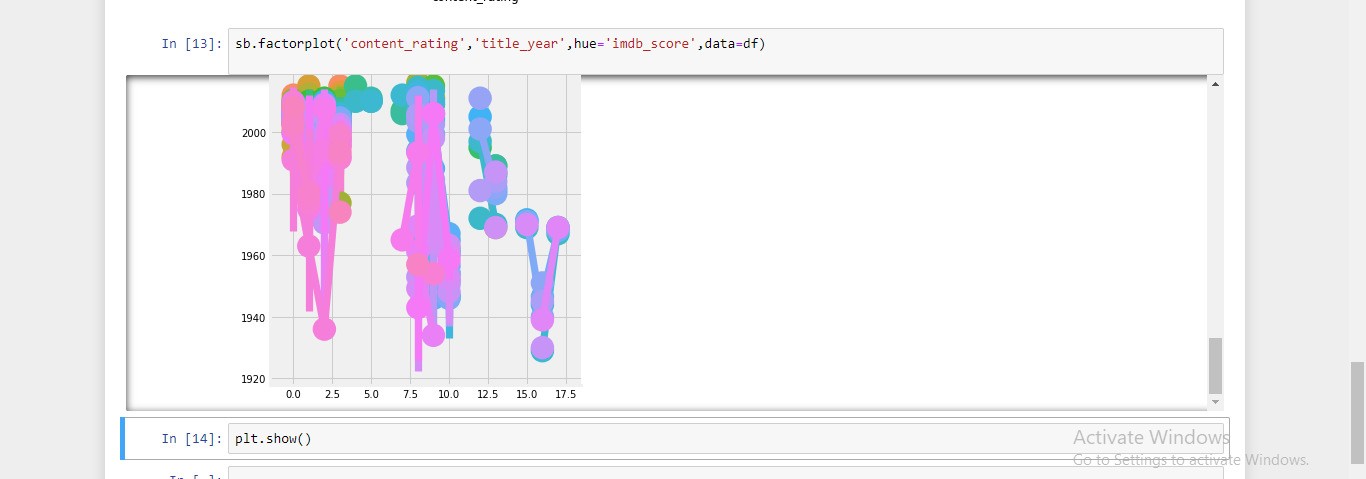
I have provided a piechart below that shows the average rating to the respective content rating :



Since, it can be noted that theres a clear correlation between the rating and the imdb score, lets look at this bar graph :

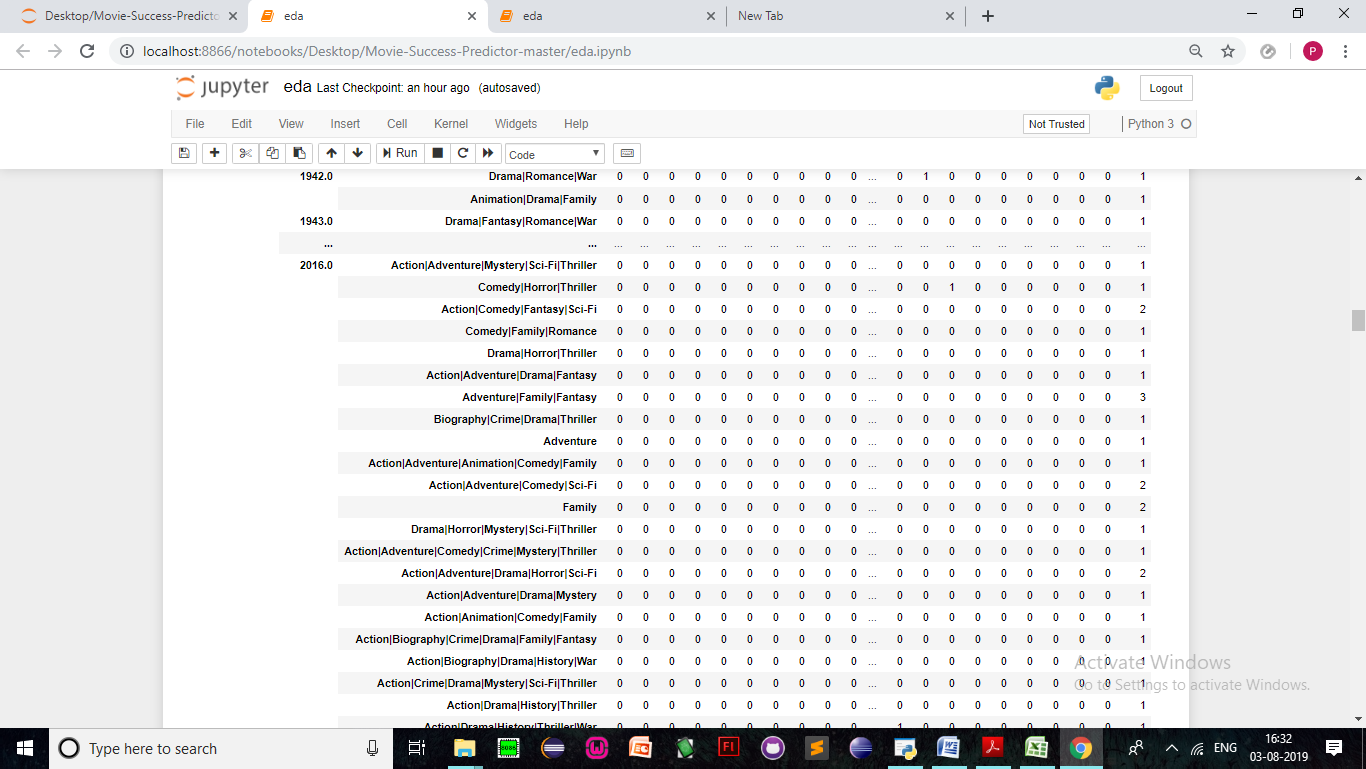


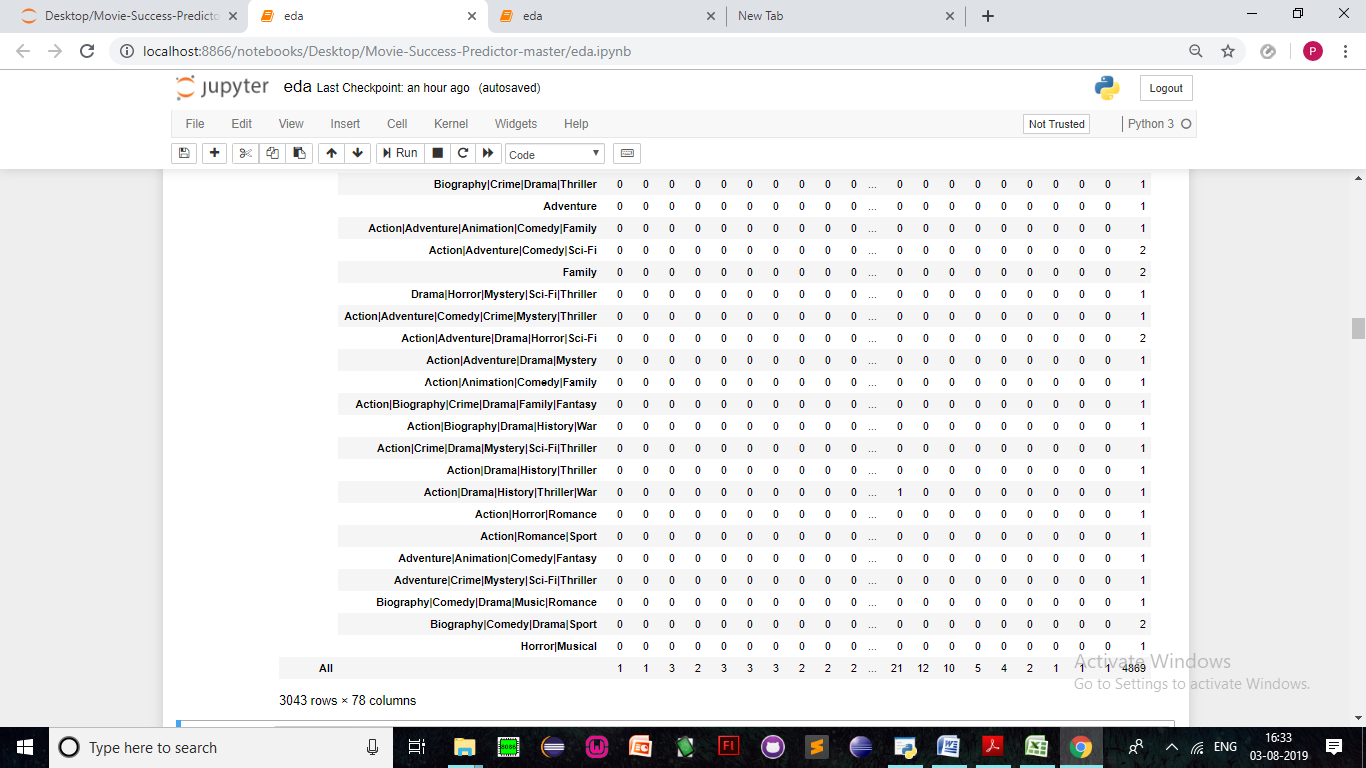
To sum up the above points graphically, I’ve included a factorplot that demonstrates the relationship between Content rating, title year and the imdb score:



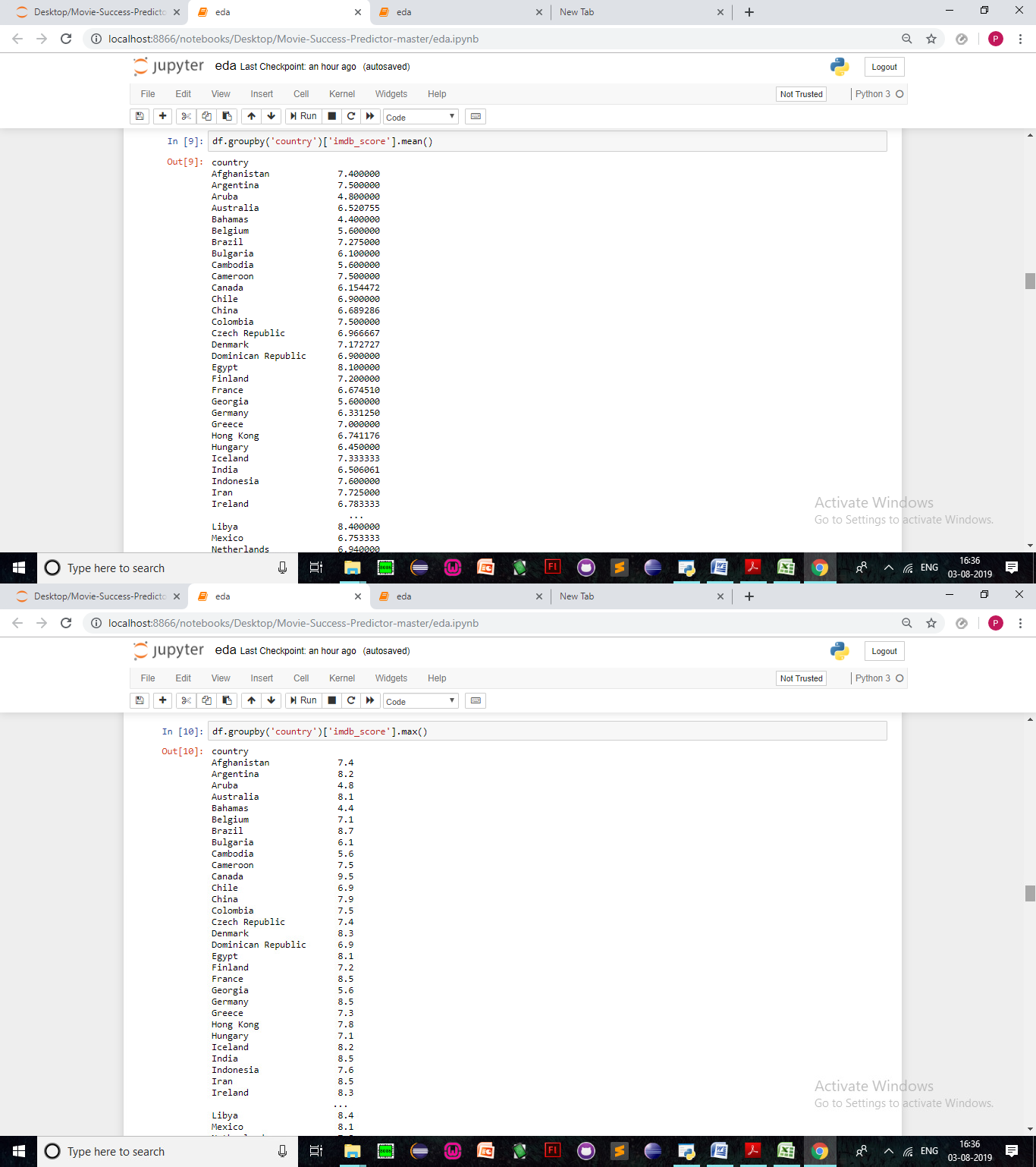
Provided below is a crosstab showing the relationship between title year , genres to the imdb scoring :







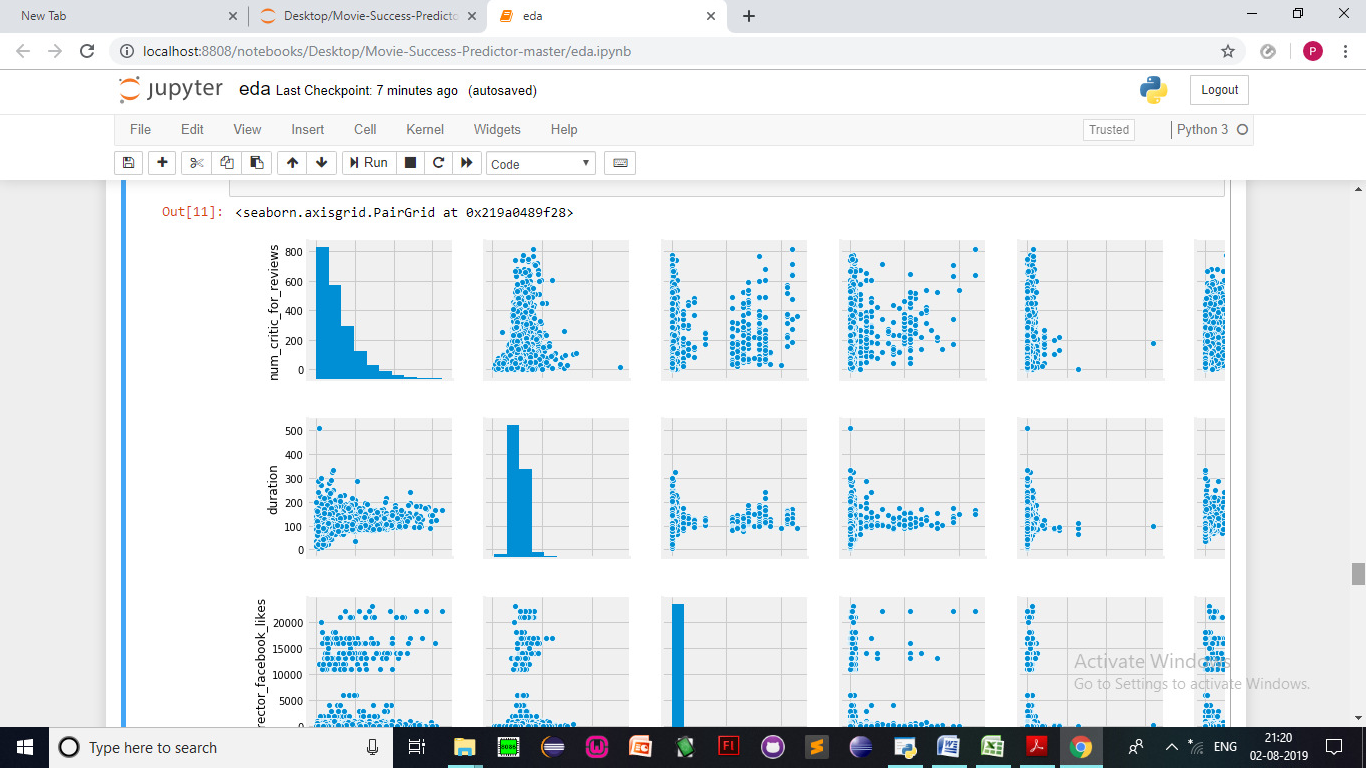
Before, we jump into the crux, let’s look at some interesting relationships between countries and their average imdb score to satisfy our curiosities:



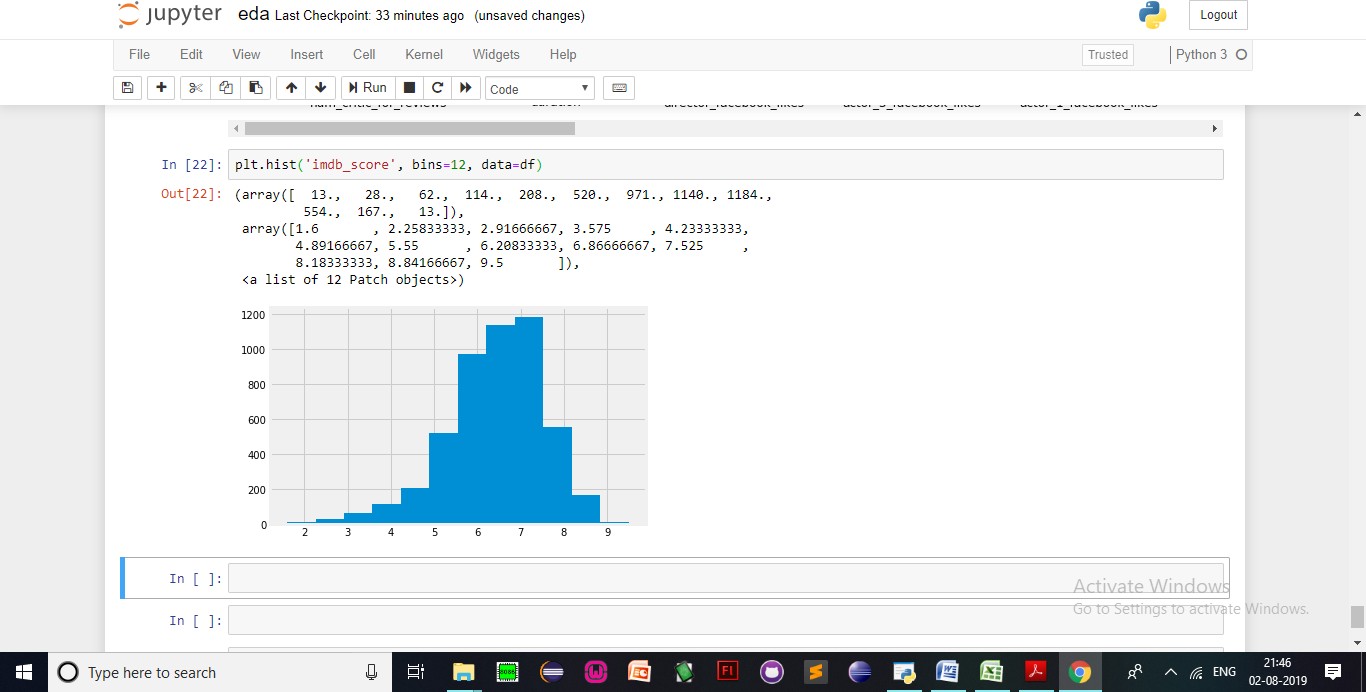
Although the highest mean belong to the least expected country Libya ( who has produced one of the least number of films when compared to the power nations like USA, UK and such ) the maximum score is from Canada. A country known for creating gems in the movie industry Last Night (I) (1998) , or the The Sweet Hereafter (1997) R .

Lets also look at some pairplots :

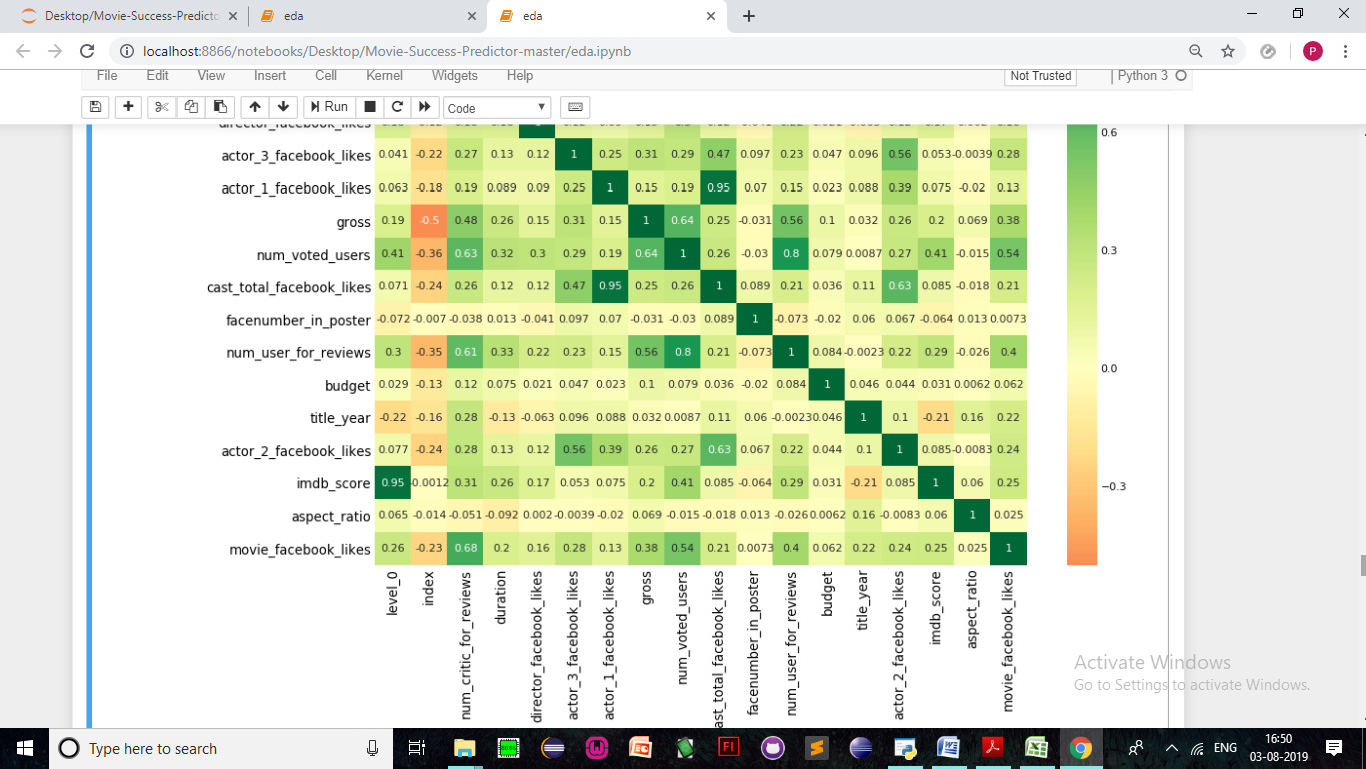
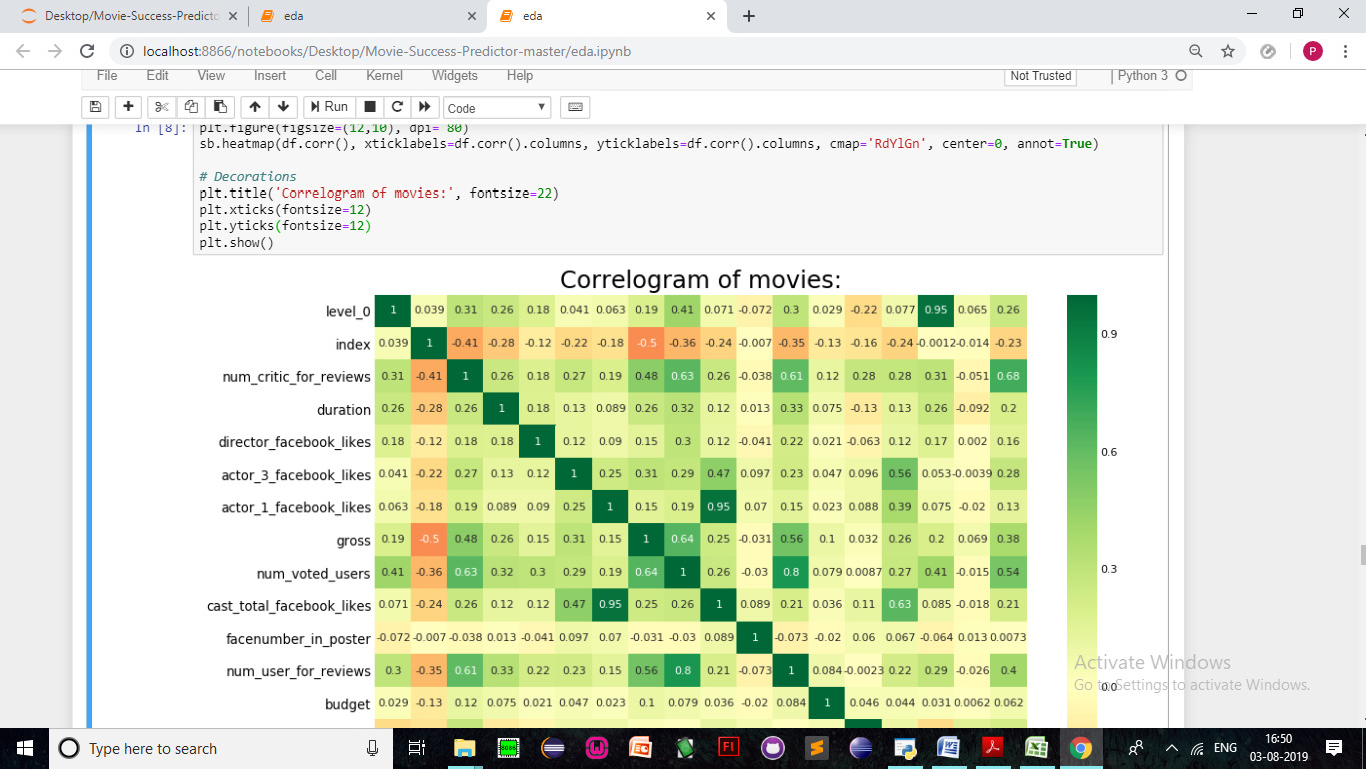
Here we compare each attribute to all attributes and to itself. This creates a diagonal of histograms when an attribute is compared to itself ( eg : attribute1 (X AXIS) vs attribute1 ( Y AXIS ) ).



Given below is a histogram that shows the imdb scores and their frequencies :

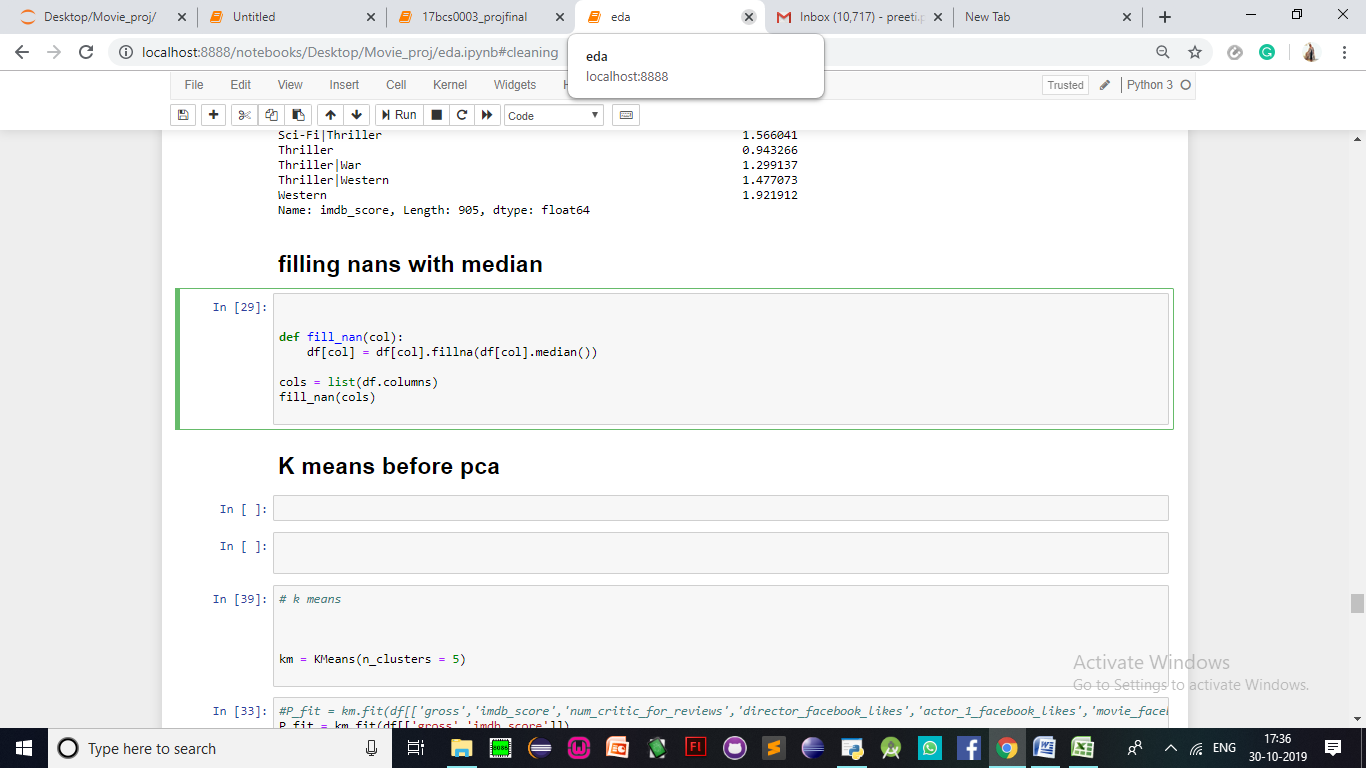


Finally, let’s look at a correlogram of each attribute to its own :



**CLEANING DATA**

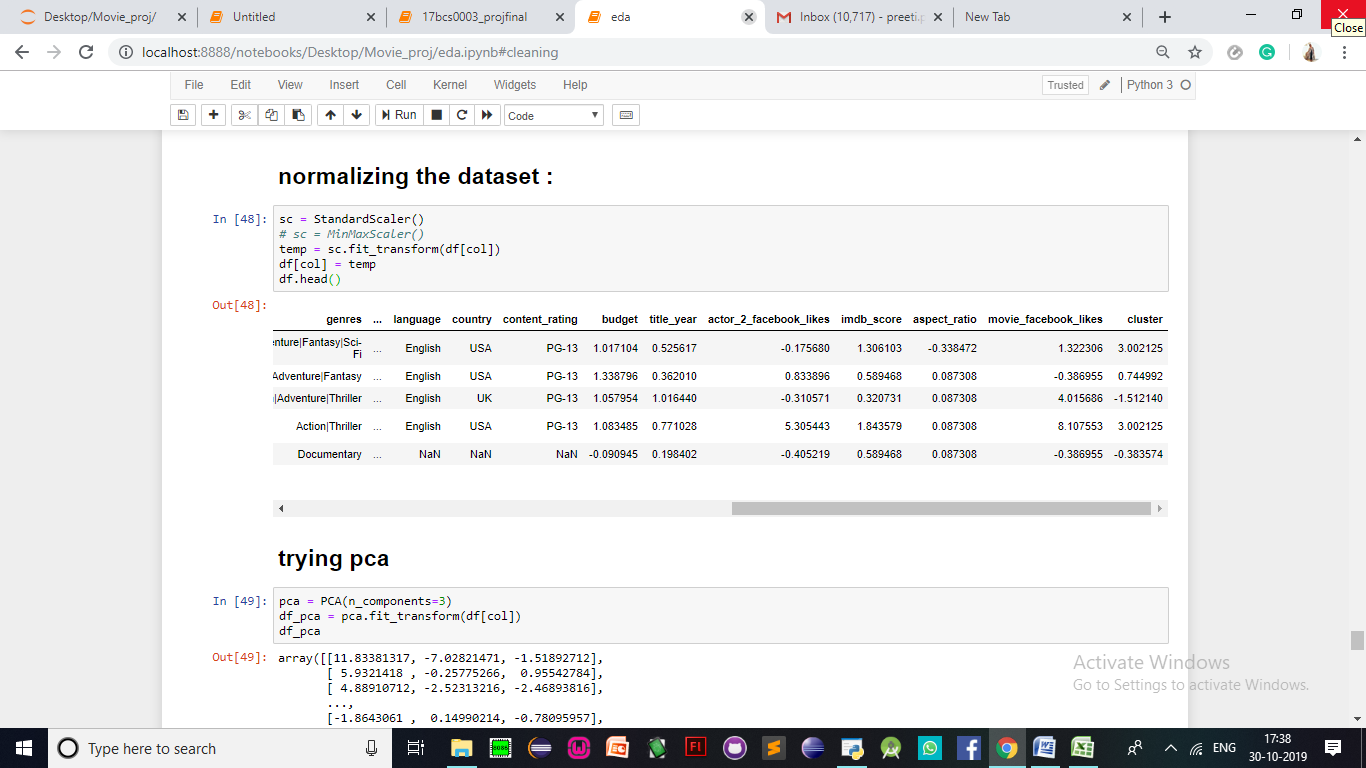
Having missing values in a dataset can cause errors with some machine learning algorithms. Missing values can cause incorrect analysis. Hence I have used df[col].median() to replace the null values with the variable medians.



**NORMALIZING THE DATA SET :**

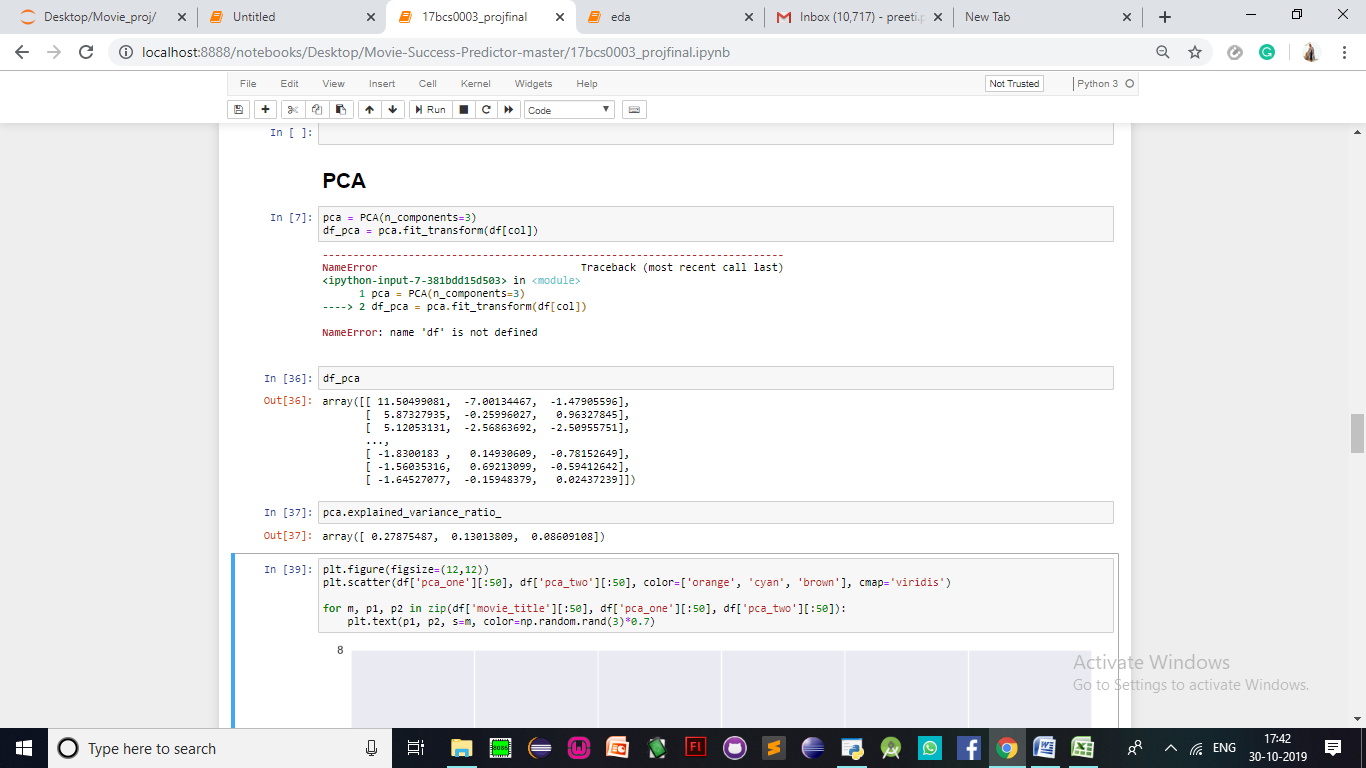
Normalization makes training less sensitive to the scale of features, so we can better solve for coefficients.

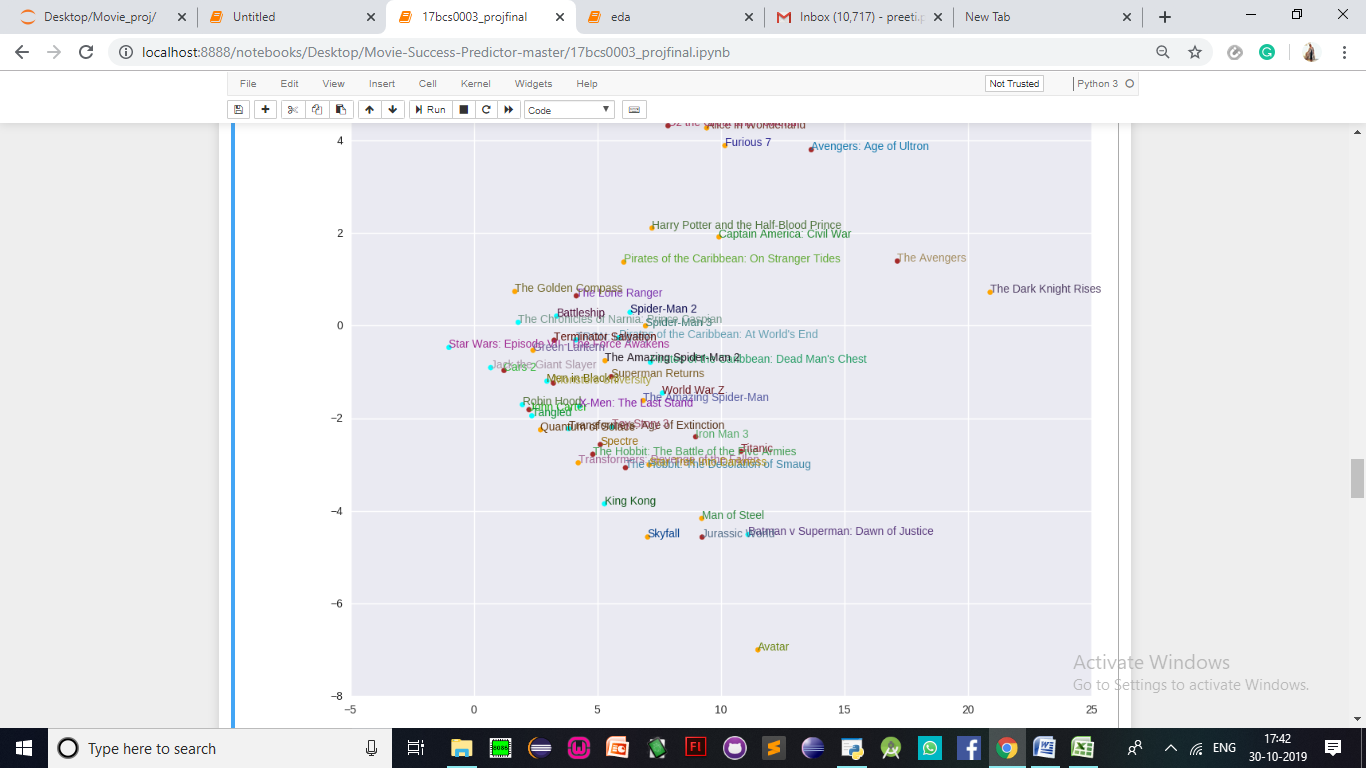
Standardizing tends to make the training process well behaved because the numerical condition of the optimization problems is improved.



**TRYING PCA:**

PCA is a good technique to reduce the high dimensionality of big data sets to fewer dimensions that are easier for humans to comprehend and visualize.





**KMEANS:**

K-means clustering is a clustering algorithm that aims to partition n observations into k clusters.

There are 3 steps:

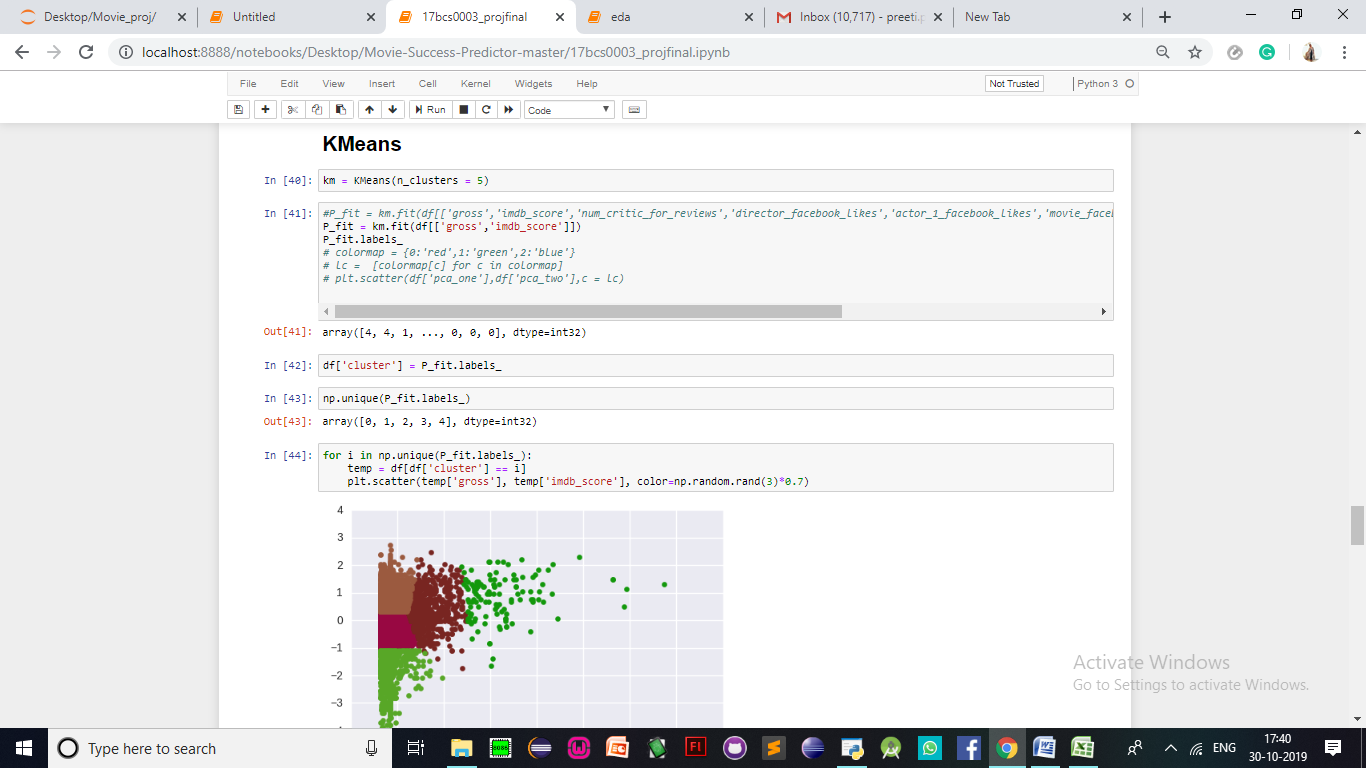
Initialisation – K initial “means” (centroids) are generated at random

Assignment – K clusters are created by associating each observation with the nearest centroid

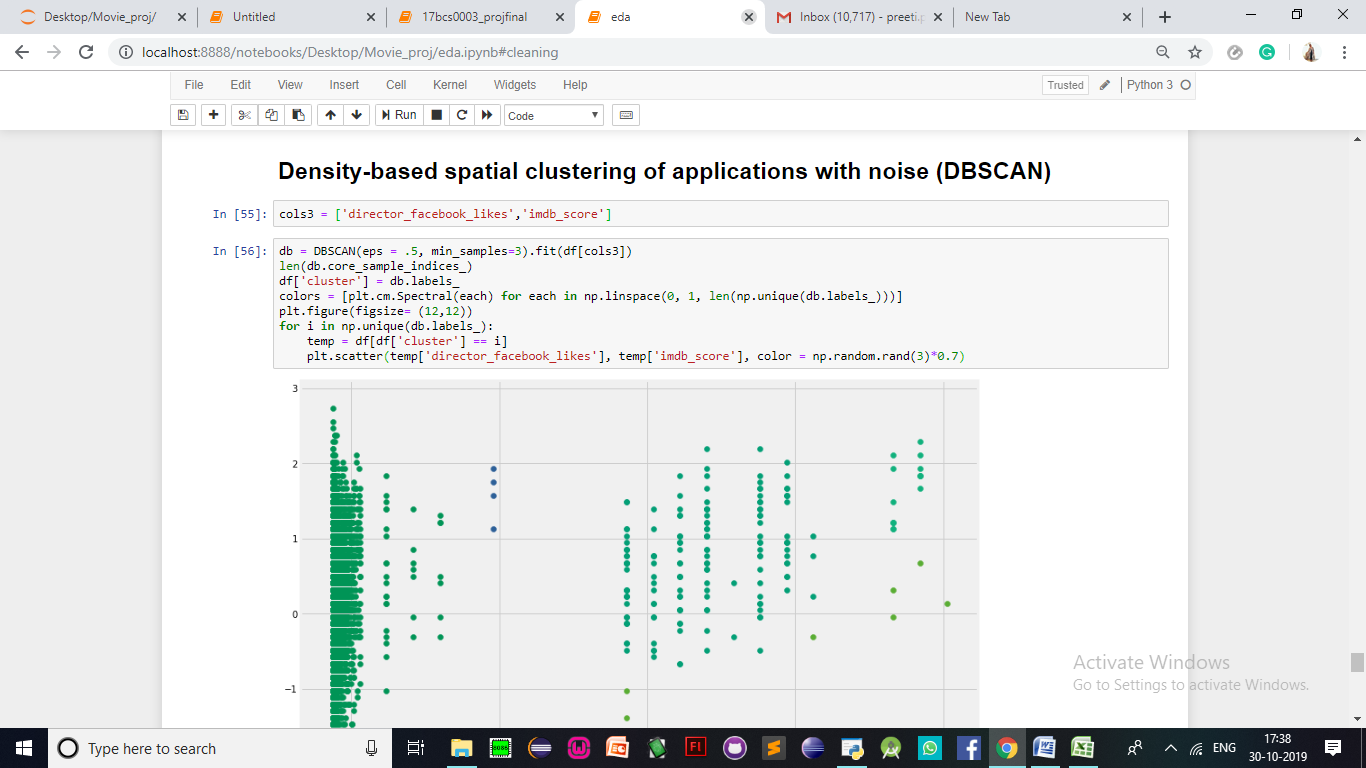
Update – The centroid of the clusters becomes the new mean

Assignment and Update are repeated iteratively until convergence

The end result is that the sum of squared errors is minimised between points and their respective centroids.



**DBSCAN**





THANK YOU!