Film Script Analyzer

Data Incubator Project Proposal

Luis Castro
November 12, 2016

I.I Definition

Project Overview

Mathematics provides a framework to understand the concepts and parameters by which a work of art is created. Literature, music & paint can be also understood through statistics, by finding underlying patterns and properties that are intuitively transmitted, but difficult to grasp independently.

The seventh art binds many facets of artistic production together, and thus is able to create marvels that humble our senses and lift our souls. It is also a billion dollars industry that has an enormous impact on our society and our culture.

The germ seed of a movie is the script. Scripts are the recipes, the structure for the director to bring his/her vision to life.

Professionals and amateurs alike write thousands of scripts; many worthy of recognition, many more destined to be forgotten.

From this immense source of natural language data meant for the silver screen is an opportunity to extract information that can help understand and automatically assess the overall characteristics of the text. A computer can go through millions of lines of text in an instant, and given the proper model, it is possible to create an algorithm that accurately extracts the most relevant features of a film script, describing and analyzing it.

It may seem frivolous or even arrogant to qualify art, but it is something we do all the time.

- *'That is not art, that is a piece of…'* **classification.**
- *'I guess it's good, but doesn't move me'* which is like 6 or 7, **regression.**

The project draws raw data primarily from 3 available sources; they were scrapped and I want to thank them here for making the information available.

- http://www.springfieldspringfield.co.uk/
- http://www.imsdb.com/
- http://www.dailyscript.com/

These webpages host the scripts that will be used to feel the machine-learning model.

FADE IN:

EXT. UPPER MISSOURI RIVER/1820'S - EVENING

ANGLE ON A SINGLE COTTONWOOD LEAF... brown and crisp... clinging to its empty branch... the solitary sign of life on an otherwise barren tree.

A gust of wind... the leaf breaks free... flutters down, landing in the slow current of the Missouri. The last leaf of the fall, taking its final journey south.

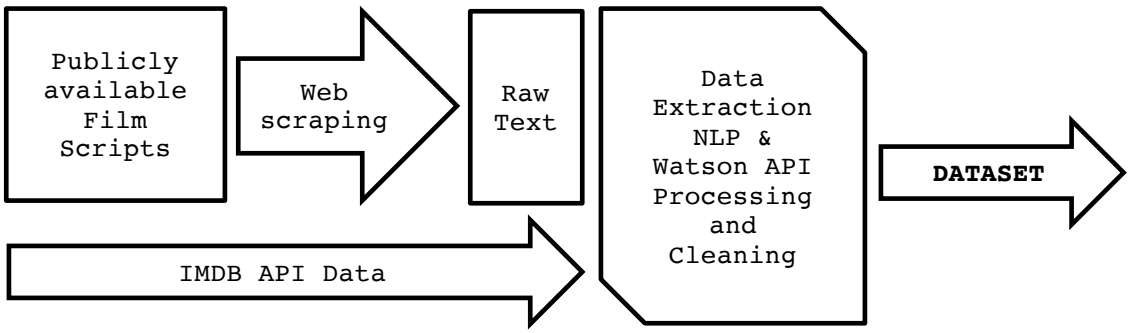
As it floats along the surface, rising and falling with the current, all we can hear is the river's gentle movement... the trickle of water... the splash of timid rapids... until DISTANT VOICES invade this world... soft at first, but growing louder... LAUGHTER... SINGING.

Glimpse into the structure of a script.

So far **2319 scripts** have been scrapped from these pages to acquire this texts, they are the raw material to construct the dataset that will later be used.

How the HTML code looks.

Additionally, in order to have a context for the script, information such as date, genre, actors, directors, public reception etc. the IMDB API is queried. **The model will assess the relationship between the information obtained from the raw script text, and the Meta data generated from the film.**



Overview of the process followed to create the dataset.

As seen previously, this is supervised learning, the targets being the characteristics or descriptions of the film. It can be classification by assigning a film a genre or regression, by assigning it a score. It uses Watson's API for analysis, web scrapping to collect the raw data, NLP for natural language statistics and IMDB for Meta data.

Project Statement

The problem, stated succinctly is twofold:

1) Determine which of the IMDB Meta data features can be extracted and predicted accurately from the script.
2) For all those that it is possible, develop an application that gathers all the relevant data from the web and transforms it into a dataset that is used by a machine-learning algorithm to have as an output such prediction.

The approach taken follows these steps:

1) Scrape the web for available film scripts.
2) Clean and format the raw texts to be used.
3) Request a Personality Insight of processed text via IBM Watson's API.
4) Request meta-data of films using IMDB's API.
5) Analyze raw text with the NLTK package (Python's NLP library), as it provides tools to perform numerical and syntactic analysis of text.
6) Perform exploratory analysis of dataset, uncovering most relevant features.
7) Use supervised regression learning algorithms to obtain cross-validated predictions the target features.

The following are ongoing steps to keep refining the model, making it more accurate:

1) **Keep increasing the dataset**
2) **Tune parameters of the algorithms.**
3) **Evaluate results.**

Metrics

For regression, two metrics will be used to determine the performance of the model: RMSE and MAE. They are used to measure difference between values predicted by a model against the values actually observed. Here are the formulas respectively.

$$\sqrt{\frac{\sum_{t=1}^{n}(x_{1,t}-x_{2,t})^2}{n}}$$

$$\frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

For classification, accuracy will be the main objective, that is the rate of correct classifications made. The metric for the model will be cross entropy.

$$-\sum_{x} p(x)\,\log q(x).$$

I.II) Data Gathering

Web Scrapping

Before analyzing the data, there needs to be data to be analyzed. The process to gather data follows the next steps.

1) Identify the sources for data, the websites previously mentioned.
2) Read the scripts data, presented in HTML format.
3) Parse HTML to raw text.

Once implemented, the application can continuously perform web scrapping from these sites, however in order not to be banned from them (happened before with the Gutenberg Project) the data was randomly sampled.

After retrieving the sample pages in HTML, it is parsed to text with the Beautiful Soup library and then saved to disk as text.

Preprocessing the raw text may have an impact on in its later use, particularly scripts include dialog and 'action' indications for the scenes, and the indications are usually on **UPPERCASE.** A hypothesis now is that removing them might have (either positive or negative) impact in the model later. So, additional files for each script will be processed and saved without these indications.

IMDB's API

The second step in gathering data is to query Meta data to IMDB through its API, this is done sending a request using the name of the movie, the returned data is stored and will be added later to the dataset. The method of how to do so is explained at OMDB.

http://www.omdbapi.com/?t=Kung+Fu+Panda&y=&plot=short&r=json

Response:

{"Title":"Kung Fu Panda","Year":"2008","Rated":"PG","Released":"06 Jun 2008","Runtime":"92 min","Genre":"Animation, Action, Adventure","Director":"Mark Osborne, John Stevenson","Writer":"Jonathan Aibel (screenplay), Glenn Berger (screenplay), Ethan Reiff (story), Cyrus Voris (story)","Actors":"Jack Black, Dustin Hoffman, Angelina Jolie, Ian McShane","Plot":"The Dragon Warrior has to clash against the savage Tai Lung as China's fate hangs in the balance: However, the Dragon Warrior mantle is supposedly mistaken to be bestowed upon an obese panda who is a tyro in martial arts.","Language":"English","Country":"USA","Awards":"Nominated for 1 Oscar. Another 14 wins & 38 nominations.","Poster":"https://images-na.ssl-images-amazon.com/images/M/MV5BMTkxOTY1NjUyN15BMl5BanBnXkFtZTcwMjMxMDk1MQ@@._V1_SX300.jpg","Metascore":"73","imdbRating":"7.6","imdbVotes":"311,115","imdbID":"tt0441773","Type":"movie","Response":"True"}

An example of the URL for request and the JSON code sent back.

From the previous 2319 scripts, **1364** or about **59%** had assigned IMDB Meta data as requested with the name given by the script, in order for the application to run without intervention, the names submitted to OMDB are directly the names reported on the scrapped pages, which may not match.

Scripts are also filtered by:

1) Movies only.
2) English only.
3) With reported IMDB Rating.

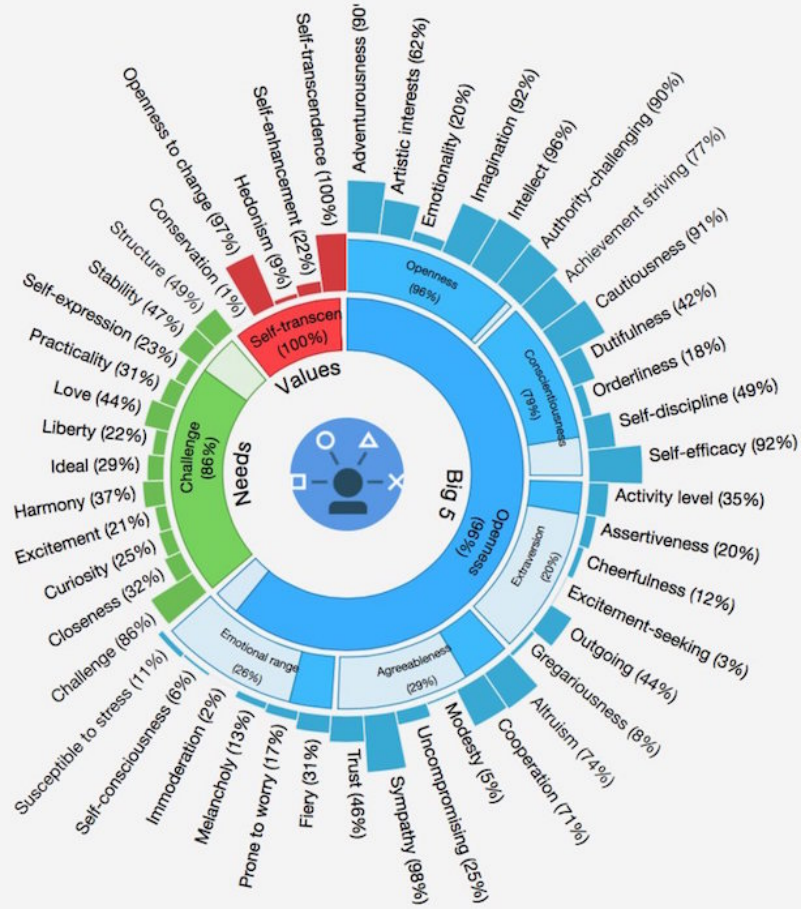
IBM Watson's API

The third step is to access IBM Watson's API, the Personality Insights part specifically. Here Watson analyses the text and returns a set of 30 different features for each text.

"IBM Watson is a technology platform that uses natural language processing and machine learning to reveal insights from large amounts of unstructured data"

This evaluates the text and assigns numerical values to it. To achieve this the following steps were followed:

- Create account
- Create instance of Personality Insights app.
- Authenticate with username/password.
- Submit text to be analyzed.
- Receive analysis of text: 30 descriptive features.



Graphic representation of features returned.

After receiving the data, it is processed to fit to a file whose elements look like this:

```
('Barbie and the Diamond Castle 2008.txt',
 {u'Achievement striving': 0.3261238981443512,
  u'Activity level': 0.2233213996760725,
  u'Adventurousness': 0.8402743701548514,
  u'Altruism': 0.9201917946717263,
  u'Anger': 0.053676946426321 7,
  u'Anxiety': 0.12273234973762553,
  u'Artistic interests': 0.9592269800623666,
  u'Assertiveness': 0.3044921447755847,
  u'Cautiousness': 0.8634964048074003,
  u'Cheerfulness': 0.7893284460516854,
  u'Cooperation': 0.9348984936714143,
  u'Depression': 0.454244557795795536,
```

This is the first entry in the flattened dictionary.

From the IBM Watson's Personality Insights basics page, we can understand the meaning behind this numbers:

Big Five *personality characteristics represent the most widely used model for generally describing how a person, the scriptwriter, engages with the world. The model includes five primary dimensions:*

- ***Agreeableness*** *is a person's tendency to be compassionate and cooperative toward others.*
- ***Conscientiousness*** *is a person's tendency to act in an organized or thoughtful way.*
- ***Extraversion*** *is a person's tendency to seek stimulation in the company of others.*
- ***Emotional Range,*** *also referred to as Neuroticism or Natural Reactions, is the extent to which a person's emotions are sensitive to the person's environment.*
- ***Openness*** *is the extent to which a person is open to experiencing a variety of activities.*

Each of these top-level dimensions has six facets that further characterize an individual according to the dimension.

Needs *describe which aspects of a product will resonate with a person. The model includes twelve characteristic needs: Excitement, Harmony, Curiosity, Ideal, Closeness, Self-expression, Liberty, Love, Practicality, Stability, Challenge, and Structure.*

Values *describe motivating factors that influence a person's decision making. The model includes five values: Self-transcendence/Helping others, Conservation/Tradition, Hedonism/Taking pleasure in life, Self-enhancement/Achieving success, and Open to change/Excitement.*

Watson can only perform the analysis with texts over 100 words, which may anyway be corrupt or incomplete files they are removed as well.

Natural Language Processing:

The fourth step is to apply NLP tools to the scripts and extract valuable statistics from them; this will be achieved with the NLTK library.

In specific 3 features will be created, while many others can and will be included, these provide basic insights of the text, they are:

- **# Of Words:** The total number of words in the script.
- **Diversity:** The number of unique words over the total number of words; this is a measure of the diversity of language used.
- **Word Length:** The average word length in the script, though with small dispersion can prove to show some insight in the script.
- **Parts of speech:** Such as noun, verb, adjective, adverb, numbers, etc. This is the normalized proportion they take of the whole text.

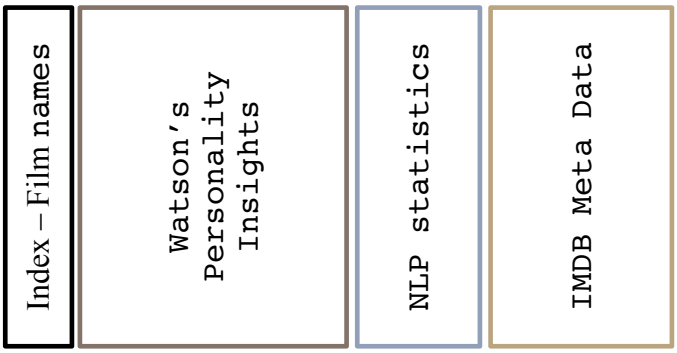
The first step is to have the scripts tokenized, that means, separate each word in the string, and then proceed to make calculations and gaining insights from the texts.

Then follows to count them and analyze their syntactic role in the script.

Merging Data

After the previous gathering and generation of data, it was cleaned and ordered to best accommodate the needs of the application.

The data frame has the following structure at the end:



One variable per column, one instance per row, everything is fine with the world.

Noteworthy is to mention that after scrapping 2319 scripts, we have only 1364 instances so far. This is mostly due not finding the IMDB data with the film name, incomplete data found or outliers that need to be removed so the information is clean and relevant for the task.

II. Analysis

Data Exploration

Many of the individual characteristics of the data were discussed in the process of generating it. Now is time to explore the dataset as a whole, describe it and relevant statistics from it.

The dataset consists of 1364 instances and 65 features, as mentioned before these were created and are subject to change as more data is added and new features engineered.

```
In [2]: df.shape
Out[2]: (1364, 65)
```

There are **many missing values;** all of it is information not available in IMDB, but those features with the highest incidences are not the main features, will not hinder the analysis.

```
[755, 'Metascore']
[531, 'Awards']
[253, 'Rated']
[88, 'Poster']
[50, 'Writer']
[30, 'Released']
[24, 'Plot']
[23, 'Runtime']
[19, 'imdbVotes']
[19, 'imdbRating']
```

Data has **30 numeric features** that range from **0.0 to 1.0** (Watson), then **3 more** that were generated as statistics of the text as seen in the next image. Data should be explored for outliers, as there are some that might be corrupted or not useful. The categorical values are: **Genre, Actors, Language, Country, Director,** and **Rated.**

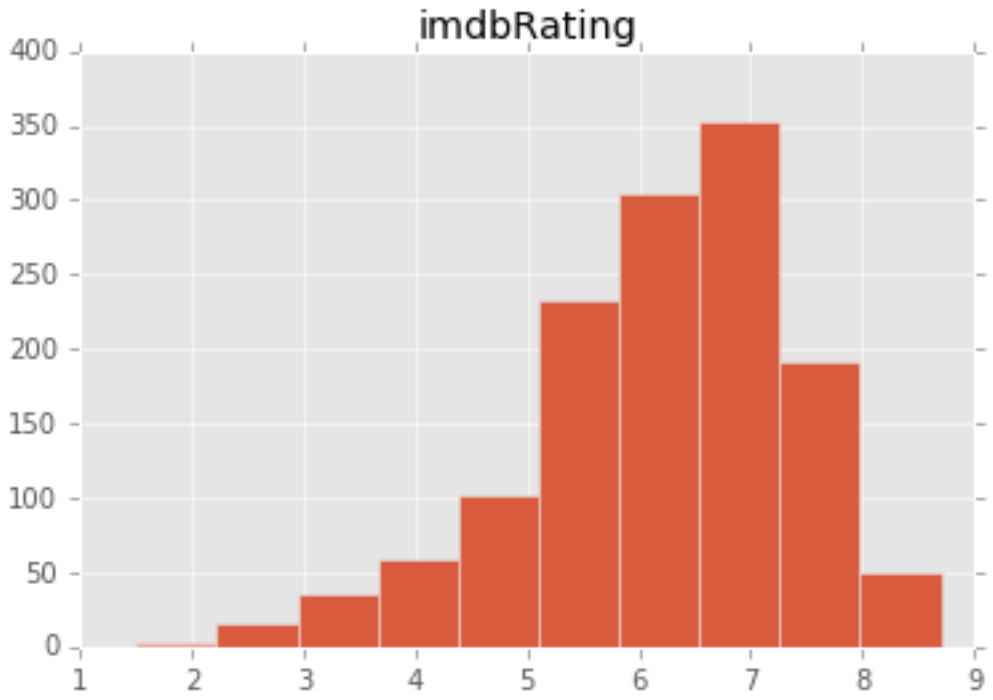
```
IQR['WORDS'][:5]
```

Flirting with Disaster 1996.txt	22652.0
Renaissance Man 1994.txt	20804.0
Silver Linings Playbook 2013.txt	21328.0
Z Channel A Magnificent Obsession 2004.txt	20431.0
Jimmy Carr Making People Laugh 2010.txt	27208.0

Checking for outliers, this example in the 'Words' feature, they seem to be valid data and will be kept.

The main interest is in English scripts, with IMDB ratings and Genre values. The non-English scripts are dropped, which account for 14% of the total.

Exploratory Visualization



Histogram of 'imdbRating', as it can be appreciated the distribution of scores approximates to a negatively skewed Gaussian distribution.

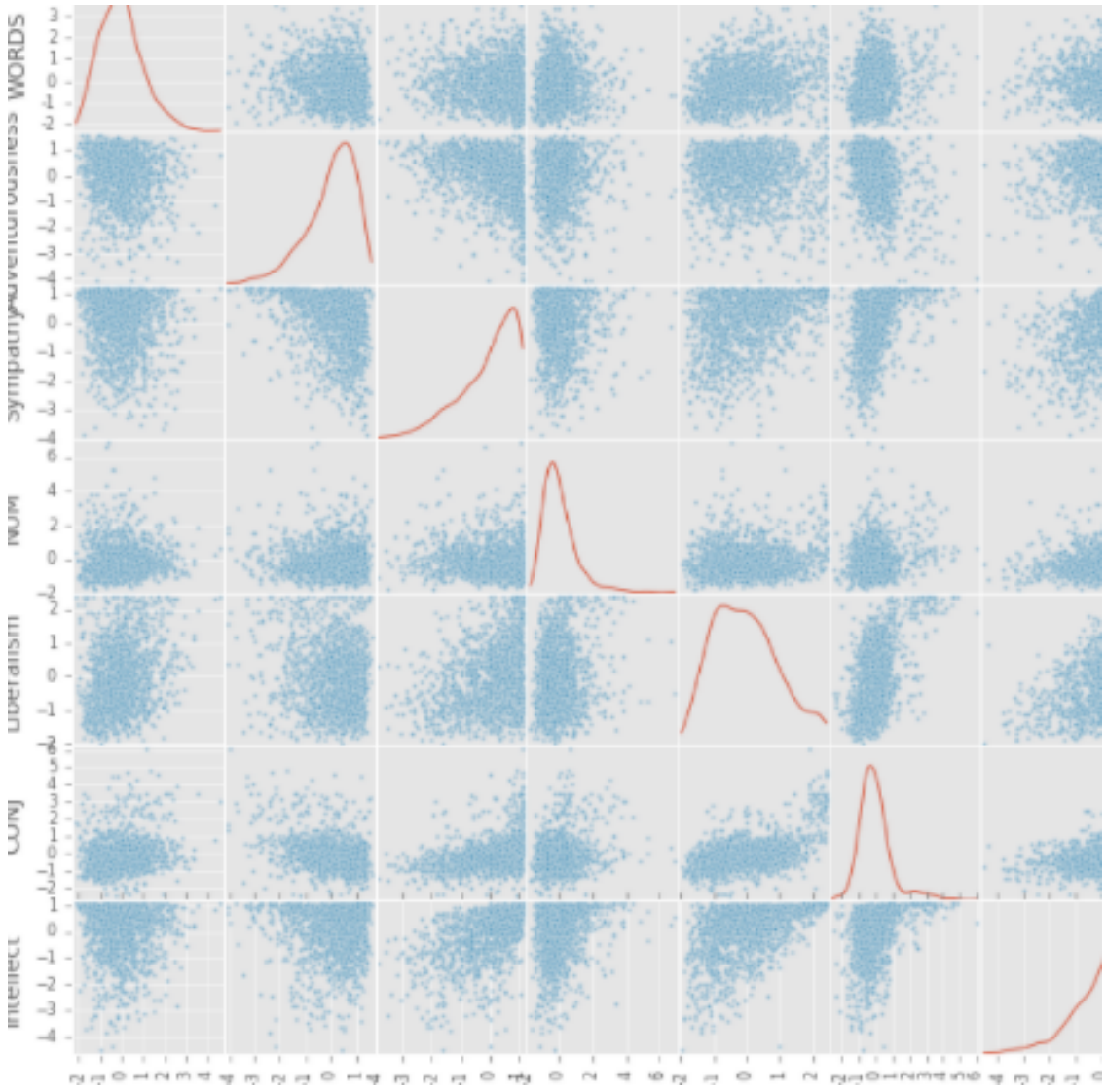
The most values are around 6 and 7 in the presented data, with a median of 6.4 and mean of 6.19. Values under 4 or above 8 are particularly rare.

Proceed to plot the most important features as listed by a Random Forest Regressor Feature Importance run over the 44 numeric features that are going to be used for the predictions.

	WORDS	Adventurousness	Sympathy	NUM	Liberalism
importance	0.086854	0.062778	0.040557	0.038585	0.032985

Sorted table of the feature importance in the dataset.

The plot is a grid showing the 7 most important features and their distribution against each other. The main diagonal shows the distribution of the values of the feature themselves. It is important to mention that this values have been scaled to have mean=0 and std=1. This is done so the data can be better appreciated, and also because many learning algorithms are susceptible to magnitude differences among features, assigning greater importance to features with greater magnitudes. Doing this we assure it is fair play for all features.



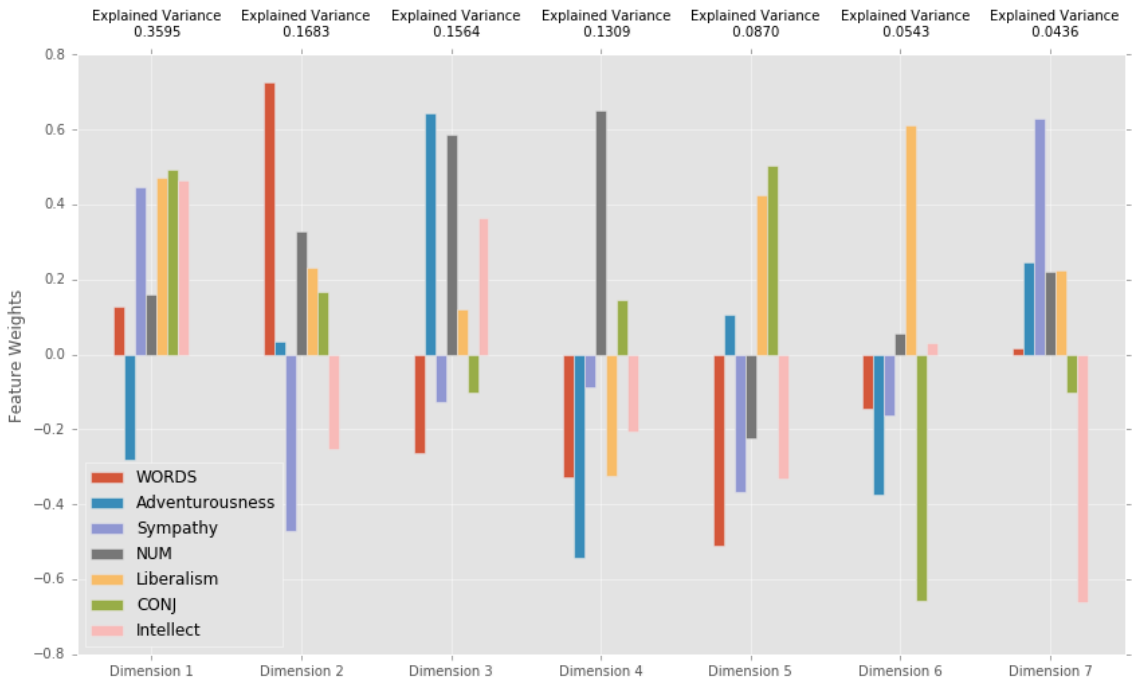
These are the 7 most important features according to RF feature importance.

It is difficult to appreciate a relationship among the presented features, even more to give a meaning to them, however weak trends can be speculated.

1) Liberalism and intellect appear to have a linear relationship.
2) Intellect and sympathy also appear to be related in a similar way.
3) Sympathy and adventurousness have a weak inverse linear relationship.

Some interesting questions follow:

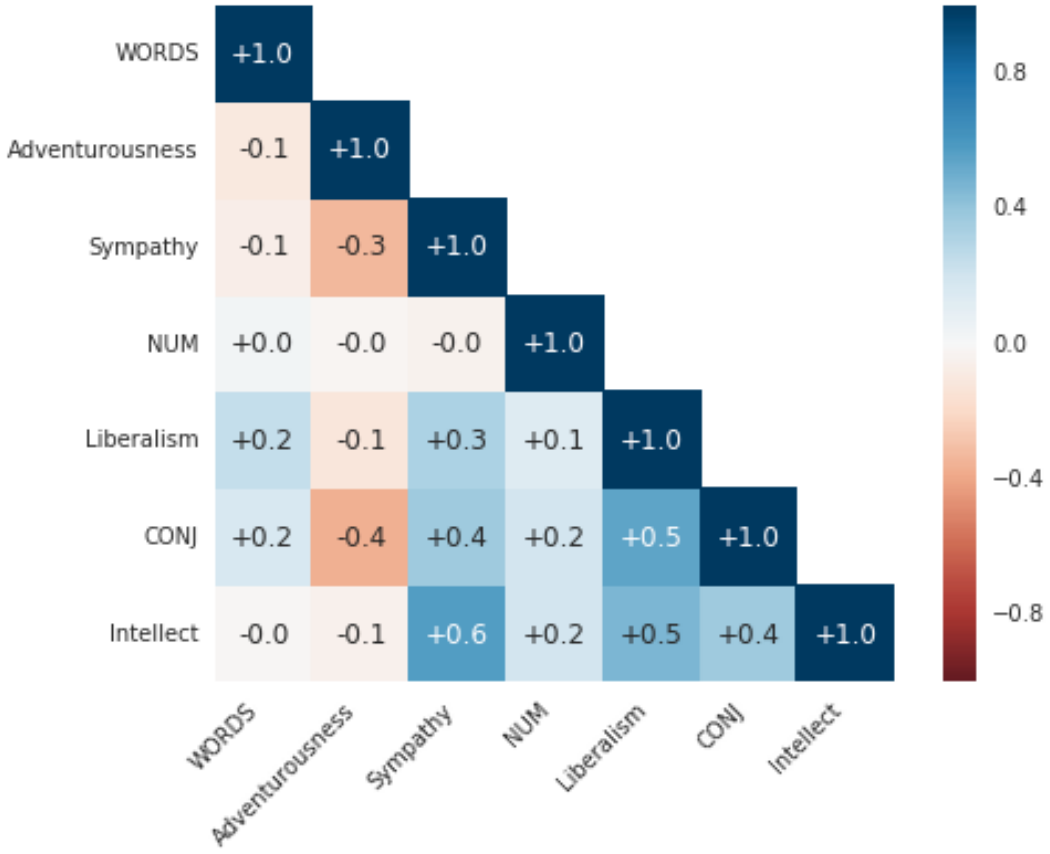
1) How is data structured?
2) How scripts relate with one another?
3) Are other features (not the most important) related?



PCA plot of the 7 principal features and how their variance is explained.

PCA or principal component analysis can help us discover underlying factors on the evident features, associations between factors that may come from common roots, by looking at this, we can associate groups of features that commonly act together and brings them together and in the darkness binds them.

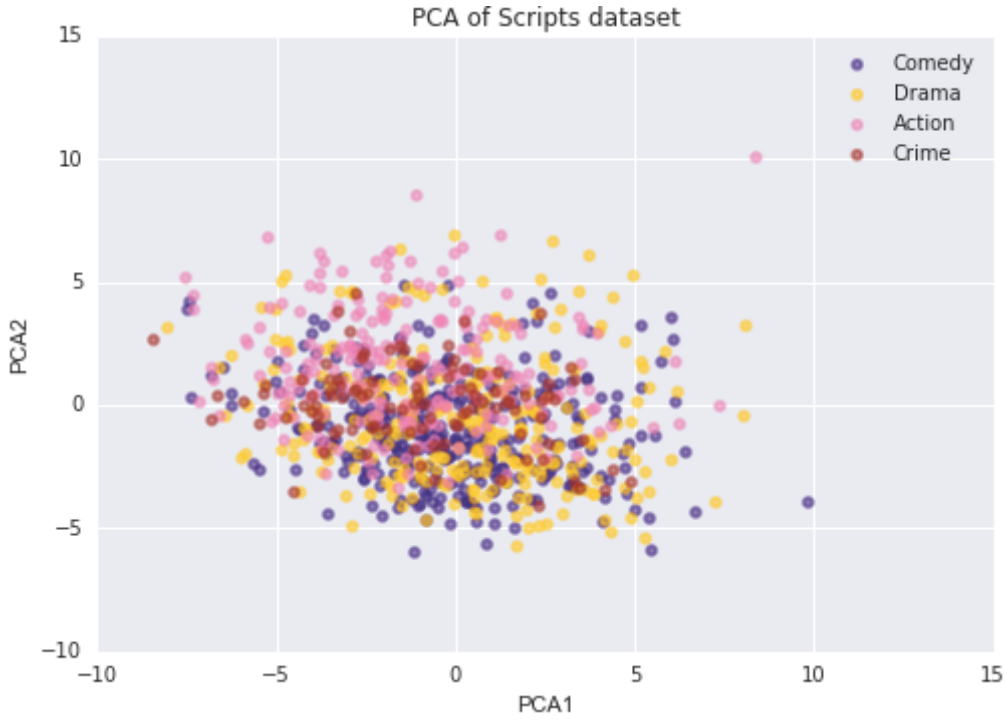
The PCA explains the variance of these 7 features only; it was plotted this way because it doesn't fit the screen to do so for all the 44, however for the next considerations all features will be used.



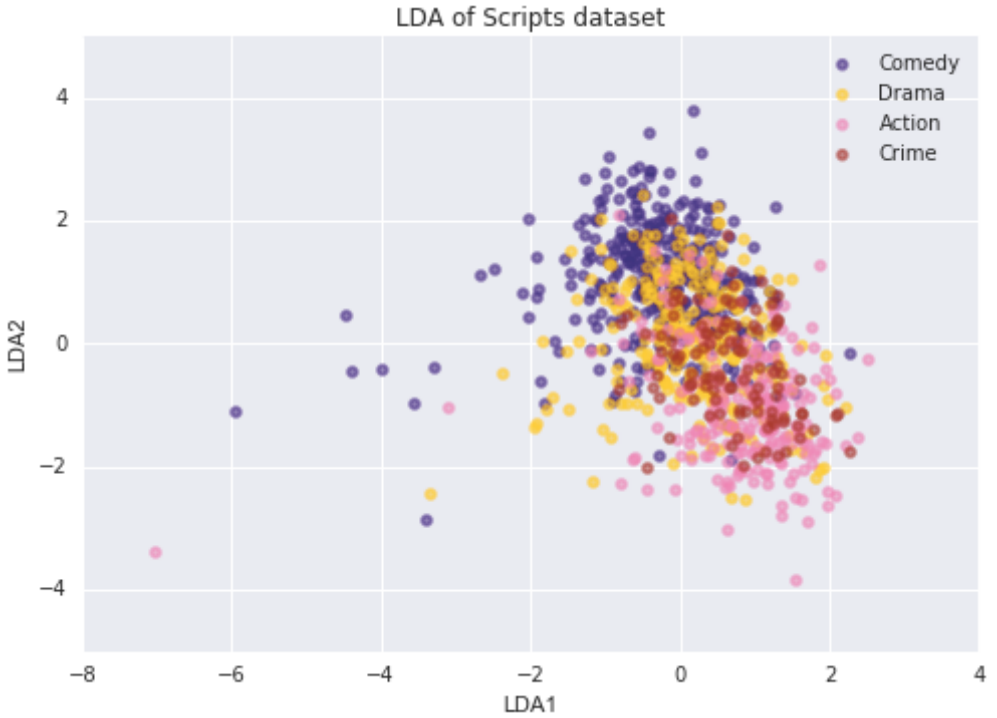
This is the correlation matrix of top 7 features.

What do we care about the relationship in the features, their correlation? It turns out it will give us clues about parameters we can not see directly in the data, like writing style, author, genre, common themes, etc. of the scripts. The correlation matrix confirms the speculations made previously with the scatter plot.

Proceed then to plot the first 2 PCA's and LDA's and see if we can find a structure to data.

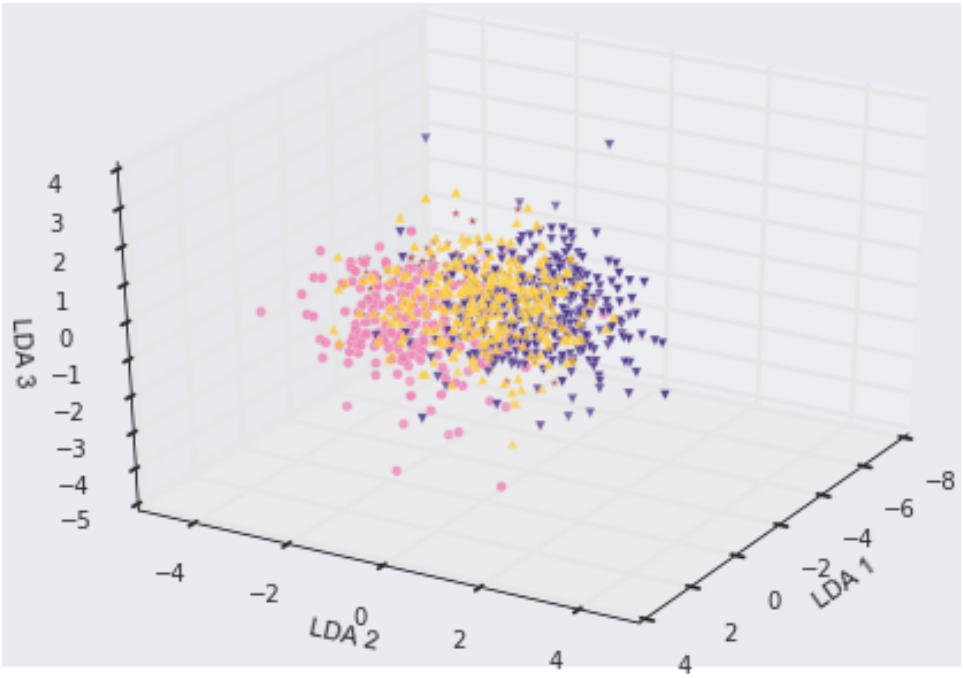


There is no clear clustering to be seen with the first 2 PCAs.



LDA provides a better segmentation, with Action and Crime opposite to Comedy on opposite corners and Drama occupying the whole center.

'Linear discriminant analysis (LDA) is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events.' - Wikipedia



This is the representation in 3D of the first three LDAs.

Similarity matrix / Recommendation dataset

The dataset created contains features that define the scripts with numerical values; in particular the 30 features created by Watson describe the 'personality' of the script. These features are used to create a similarity matrix, by using cosine similarity; it is possible to know how alike are a couple of scripts.

'Cosine similarity is a measure of similarity between two non zero vectors of an inner product space that measures the cosine of the angle between them.' - Wikipedia

The matrix will be a square matrix with side length equal to the number of scripts (could also be a triangle matrix as information is redundant).

```
recs['Removal 2010.txt']
```

```
[['Humboldt County 2008.txt', 0.99339255158462703],
 ['Flatliners 1990.txt', 0.99279872617400944],
 ['DEBS 2004.txt', 0.99106795768243194],
 ['Angus 1995.txt', 0.99036123319256542],
 ['Humoresque 1946.txt', 0.98934805237640233]]
```

These are top recommendations for the movie 'Removal'.

Recommendation

Recommendation systems work primarily in 2 ways.

1) Analyze customers' behavior and look at similarities among them. Ex. Cust1 likes 'A,B'. Cust2 likes 'A,C'. Cust1 may enjoy 'C'.
2) Look at things that have similar characteristics or that are associated. Ex. Cust1 likes A. A has 'a,b,c'. C has 'a,b,d'. Cust1 may enjoy 'C'.

Two interesting questions rise, how to know what customers like? What characteristics are important to create associations?

1) How to know what customers like? Primarily by votes 'likes', ratings '4 stars' or browsed items.
2) What characteristics are important to create associations? That needs more substantive expertise. Smurfs are blue, I like Smurfs but I may not enjoy blue scarfs. Fortunately since the comparison is just text we dwell in a single domain.

Summarization

'The target of the automatic text summarization is to reduce a textual document to a summary that retains the pivotal points of the original document. The research about text summarization is very active and during the last years many summarization algorithms have been proposed.' - Glowing Python

By computing a frequency map of words, they are filtered to ignore low and high frequencies. The sentences are ranked according to frequency of words and top sentences are returned for the summary.

Levenshtein distance

'Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t.' — LD in 3 flavors

LD is used to find the script names even if there are extra spaces, typos or a close but not exact memory of the name of the script, returning the most similar name found, the one with the smallest edit distance.

Combining everything we provide as input a close name of a script and as output is displayed:

```
fs = FrequencySummarizer()
fs.summarize('beat     spay    blove',5)
```

```
Title:  Eat Pray Love
Rated:  PG-13
Runtime:  133.0
Director:  Ryan Murphy
Actors:  Julia Roberts,  I. Gusti Ayu Puspawati
Genres:  Drama,  Romance
Rating:  5.7

Poster:  https://images-na.ssl-images-amazon.com/images/M/MV5BMTY5NDkyNzkyM15BMl5BanBnXkFtZTcwNDQyNDk0Mw@@._V1_SX300.
jpg

Summary:
I'm going to Italy and then l'm going to David's guru's ashram in lndia... ...and l'm going to end the year in Bali.
It's long, it's tedious, I can't keep up... ...and l get these insane anxieties about everything in my life... ...and
 l've lost my place.
And it was such a foreign concept to me, that l swear l almost began with: "l'm a big fan of your work."
-No, I don't even have my-- l-- You don't have your-- You don't-- You're so naked.
And l-- You know, I don't-- I don't know.


Similar Titles:
Me Again
In The French Style
Under the Greenwood Tree
Far from the Madding Crowd
Dames
```

This is the displayed information for 'Eat Pray Love', relevant Meta data, summary and recommendations.

Algorithms and Techniques

The answer to the question: 'what machine learning algorithm should I use?" is always "It depends.' It depends on the size, quality, and nature of the data. It depends what you want to do with the answer. It depends on how the math of the algorithm was translated into instructions for the computer you are using. And it depends on how much time you have. Even the most experienced data scientists can't tell which algorithm will perform best before trying them." — *Microsoft Azure*

The problem as discussed earlier is both **regression** and **classification.** Predicting continues value or discrete classes; the IMDB rating or the **script genre.** It is also a **Supervised Learning** problem, which is we know the correct assigned target values for each instances in the dataset.

Processing time or **memory** are no concerns, as it is a **small dataset** and is **not** analyzed in **real time. MAE** and **Accuracy** are the most important factors; the model should output a float number as close as possible to the original target or the exact class in most of the instances.

There are many algorithms that can be used for given a problem of these characteristics. Here a brief explanation of some of them.

Linear regression (regression): Calculates the target value as a linear combination of the parameters. It is fast and easy to understand, useful for linear data. Common tuning includes: selecting features, ridge and lasso.

Support vector machines: Calculates the maximum separation of features by creating hyper planes. By changing the kernel function, it can be adapted to non-linear data. Common tuning includes: selecting kernel, gamma and C.

Random Forest: An algorithm that consists of an ensemble of decision trees. The trees split the data by searching for maximum information gain on each split, good with non-linear data. Common tuning includes: tree depth, # of trees, column and row sub selection and learning rate.

Ensemble Bagging/Boosting: Using weak learners, this methods stack the learners. One creates different bags of samples of data to train, the other chooses with replacement from previous samples assigning higher

probability to high errors. Common tuning includes: Tree parameters, loss, and learning rate.

Clustering Methods: These methods opposite to the previous ones are 'lazy', meaning that they predict the result until it is requested. They do not take time on training, but have to keep all the data. Assign values by looking for closest data points and create an average of them. Common tuning includes: # of neighbors and weighted distances.

Neural Networks: Very robust and powerful non-linear algorithms. They are highly customizable but can take a long time to train, especially for deeper configurations. Common tuning includes: Depth, width, drop-out and activation function.

Along with the previous concerns, it is important to pay attention to over fitting vs. bias; since the dataset is small we can easily over fit and not be able to generalize well. Outliers can be dangerous here too, as they represent a higher percentage of the data. Cross validation is used to understand how the models behave with unseen data.

Benchmarks

As specified earlier in the metrics section, the aim is to minimize the RMSE, RAE or maximize the accuracy. To do that all, tests will be performed to the previous algorithms, to choose among them 1 or more to be tweaked and tuned to see which can be used for such purpose.

However after all is said and done (and more is said than done), what accounts for a good model? Unless we have 0 RAE or 100% accuracy for all training and testing and future data, we can't know for sure. However there is a benchmark to overcome, which is chance. As previously seen, the target values approximate to a Gaussian distribution. With known mean and std. If we draw from it at random, there is a good chance we will draw a number close to the mean (6.19 so far). So we need to check if the new distribution of values, the predicted values come from a statistically different distribution and not from chance. How much better is the model than just selecting the mean of the population? For classification will be easier to determine how the model behaves against chance, there are more than 18 different genres listed, only 18 will be used as for the rest there are not many instances.

III. Methodology

Data preprocessing

All the features to be used in the dataset are numeric features. However they have very different ranges.

1) **Personality Insight:** 30 features [0-1).
2) **Parts of speech:** 10 features [0-1).
3) **# of Words:** 1 feature [1k-20k+)
4) **Word Length:** Centered around 3.

Some learning algorithms have no problem accepting features with different ranges of values, however there are some that depend on creating hyper planes or calculating distances like SVM and KNN, it not taken care of it can result in poor performance.

The data is then preprocess, that is, all features will have mean = 0 and std = 1.

Many other steps of data preprocessing such as removing outliers, non relevant data, NANs, low or zero variance features, etc. where already done and explained in the data gathering section.

Implementation

Classification: The Genres feature had in itself a list of many genres describing a film. It has been split in 2 Genres1 and Genres2 containing only one genre each. The aim is to correctly assign a genre to each script.

From he created features; remove classes that contain less than 4 instances.

```
[u'Adult', u'Musical', u'History', u'Film-Noir', u'War', u'Reality-TV']
```

These are the classes with small representation in the dataset.

After some out of the box trials for the machine learning algorithms, the 3 chosen were support vector machines, k-nearest neighbors and random forest.

Tuning: Each has a particular set of tuning parameters as explained earlier, to automatize the search for the optimal parameters a **Grid Search** is performed, the model is evaluated with each tuning parameter changed so to return the optimum tuning parameters from the grid, they were:

SVR Tuning:

- Kernel: Form of the separating function.
  - C: cost can take a range of numbers.
    - Gamma: Defines influence of a single sample.

Tune KNN:

- Data = 'Preprocessed' or 'Unprocessed'
  - K = 1-30
    - Weights = 'uniform' or 'distance'

Random Forest Tuning:

- n_estimators: Number of trees to grow.
  - max_depth: Max number of nodes expanded.
    - max_features: Max number of features to consider.

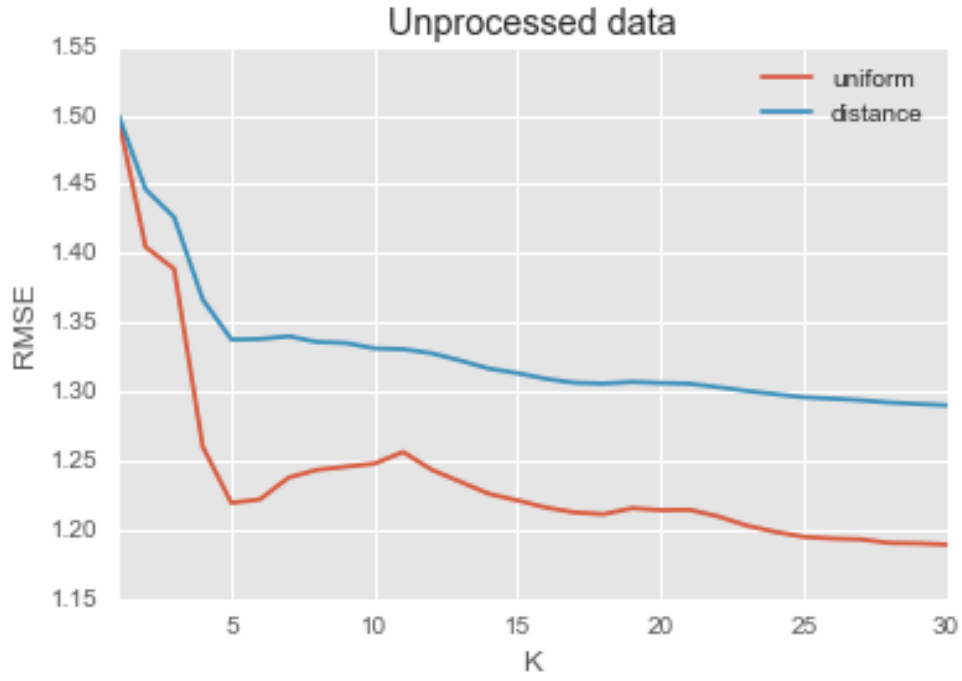
```
0.476900149031
{'kernel': 'rbf', 'C': 4.0, 'gamma': 0.01}
0.438934122872
{'n_estimators': 250, 'max_depth': 11}
0.430792005922
{'n_neighbors': 15, 'weights': 'distance'}
```

Respectively are: SVM, RF, KNN.

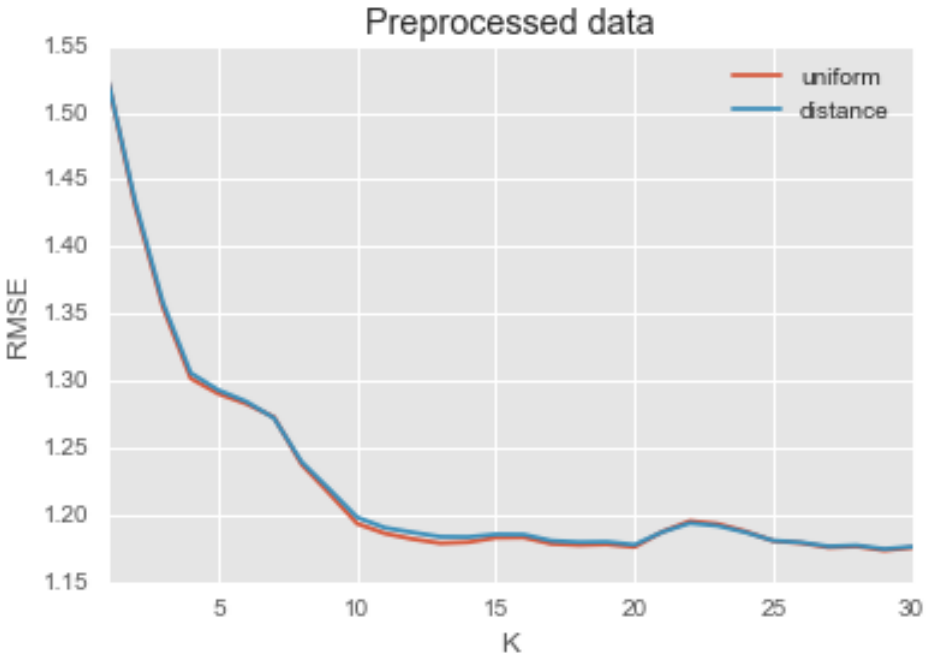
The first number is the cross-validated accuracy, next are the tuning parameters. The accuracy for trying to predict either the first or the second genre is about 57%.

Regression: Using the same methods as before but for a regression problem, predict the imdbRating of a script. The models are tuned but some important processes need to be done for the models.

Preprocessing: How do results compare between processed and unprocessed data? The following plot shows the impact on KNN.

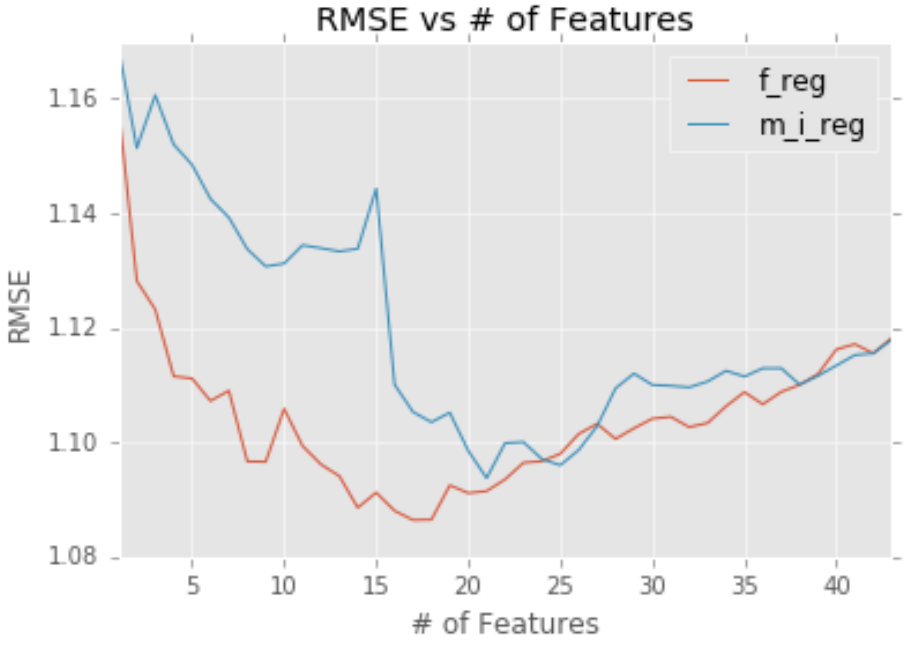


KNN regression tuning with unprocessed data.



Same tuning process with processed data, there is a clear impact on the models with and without preprocessing.

Feature selection: Another important task for improving the quality of the results, this is selecting the best features for prediction. To achieve it two methods for selecting best features are used and compared, f regression and mutual information regression.



Evaluating ML algorithm with different number of features.

```
          f_reg   m_i_reg
17  1.086448  1.105315
          f_reg   m_i_reg
21  1.091503  1.093763
```

Result matrix presents the best score depending on # of features and method used.

```
[u'Self-consciousness', u'Self-discipline', u'Sympathy', u'Modesty',
 u'Adventurousness', u'Gregariousness', u'Intellect', u'Imagination',
 u'Cheerfulness', u'Liberalism', u'WORDS', u'LENGTH', u'ADP', u'.',
 u'CONJ', u'NUM', u'X'],
```

These are the chosen 17 features, from f_reg.

F Regression: *'Univariate linear regression tests. Quick linear model for testing the effect of a single regressor, sequentially for many regressors.'*

Mutual Info Regression: *'Estimate mutual information for a continuous target variable.'*

Neural Networks (WIP): Adding to the previous methods, another powerful tools are Neural Networks, fortunately the Keras package makes it easy to set up and configure them.

Main considerations to have on a NN:

1) **Input size:** # of features for each row of the dataset, 44 in this case.
2) **Wide networks:** The number of nodes in the same layer of the network.
3) **Deep networks:** The number of layer a network has. Usually are the input layer, 1 or more hidden layers and an output layer that changes size depending on the expected result.
4) **Connections:** Are the layers fully connected among them, are there connections within the same layer? For this case full connections are used.
5) **Initialization:** The initial random values for the weights. Ex. uniform, normal, zero, etc.
6) **Activations:** The activation function for a node, the output of such node. Ex. tanh, softplus, sigmoid, etc.

7) **Loss:** This function tells about the performance of the network, it is the function to minimize or maximize. Ex. MSE, MAE, cross entropy.
8) **Optimizer:** It is the learning function, the one that propagates the knowledge learnt on each epoch.

Other important parameters are the batch size, the number of instances used on each training step and the epochs or the number of training steps to be had.

```
def reg_model(input_dim=44):
    model = Sequential()
    model.add(Dense(19,input_dim=input_dim,
                    init='normal',
                    activation='softplus'))
    model.add(Dense(1,init='normal'))
    model.compile(loss='mse',optimizer='Adamax')
    return model
```

Regression NN model example.

So far the best classification achieved for NN has been 26%, and regression of 1.16 RMSE and 0.79 MAE as this was just recently started.

IV Results:

Model Evaluation and Validation

The all the results are after model tuning, feature selection and data preprocessing. The reported results are from 5 fold cross validation, only the best results are reported here.

Accuracy in classification with SVM:

- 1st model Genre1: 48 %
- 1st model Genre1 or Genre2: 57%
- 1st and 2nd models Genre1 or Genre2: 74%

First model is model trained with Genre1, second model is trained with Genre2, combining them is the accuracy of either of them predicting either of the results.

- RMSE in regression with SVM: 1.064
- RMSE in regression benchmark: 1.191

The benchmark was set to be the RMSE of the population mean vs. the population. The model was better than that result, but there is still work to do.

Justification

Reflections on what has been learnt and seen.

- **Oranges and Apples:** The IMDB rating is the mean of votes for a film. A script is a subset of the film, but not vice versa. Films have music, visual effects, director's vision, talented actors etc. Since the information comes only from the script (no IMDB data was used for prediction). The leap from film score to script score is a big one.
- **Not Big Data:** Big data is showing time after time that obscure patterns emerge if you have big enough data. The subtleties of processes not immediately evident surface with great amounts of information. However for infrastructure, economic, technical (banned for requesting or scrapping too much data) and time restrictions, the dataset is a petite one and those patterns remain to be found if they are there.

- **(Wrong) Data:** Though the data generated and gathered for the dataset is relevant for the script and for some NLP tasks, further analysis is needed to remove features that do not add value and create new ones that do.
- **Try hard enough:** When solutions come easy, one doesn't try hard enough.

V Next Steps:

So far the results are promising but not conclusive, more data, more analysis and experimentation remain to be done.

1) **Gather more data:** Scripts analyzed now are in the order of thousands; make it be in the order of hundreds of thousands.
2) **Semantic features / word count:** Use word2vector to have another view or perspective of the scripts. Create word count of the top non stop words used by genre.
3) **Ensemble models:** Create ensemble classification and regression models, this way can help reduce the over fitting of a certain algorithm and bring closer to the real values or classes.
4) **Develop web interface:** Create an interface to display the results and statistics of the scripts, and receive new ones to be analyzed and compared with the others.
5) **Expansion:** Use the application infrastructure to provide similar uses for literature, books, poems, essays, etc.

All steps, calculations and relevant information and data are at: https://github.com/luisecastro/dataInc