City, University of London
MSc in Data Science
Project Report 2016/17

TV show rating prediction by Machine Learning methods with quantisation of the review

Jaemin Jeremy Jung

Supervised by: Tillman E. Weyde
28 Sep. 2017

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that MSc Project Scheme 2016/17 –BCS accredited and allied courses 12 relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed:

ABSTRACT

- **Purpose**: To accurately predict TV show rating using Machine Learning methods with quantising review with sentiment words.

- **Methodology**:

1. **Quantisation of the review**: Review texts were vectorised using Google's word2vector prebuilt model. Additionally K-Means Clustering was conducted to the review vectors on a 300-dimensional space into 20 different clusters. Vectorised and clustered reviews were converted to a single code and prepared for inputting to following prediction model.

2. **Prediction**: Regressor Multi Layer Perceptron Neural Network model was used for predicting future TV ratings, input attribute with Quantised review, Genre, Network, Cast, Producers, Season, and Premiere Date.

3. **Result and Evaluation**: MSE (Mean Square Error), R^2, and MAD (Mean Absolute Deviation) were used to calculate the accuracy of the prediction. Result was : MSE (Mean Square Error)=0.5631, R^2=0.6509, and MAE (Mean Absolute Error) = 0.5953. Classification problem (targets with 'unpopular', 'moderate', and 'blockbuster') was also tried especially successful implementation with Decision Tree model. Gridsearch was conducted in order to optimise the model

4. **Discussion**: Neural Network model prediction was successful showing much better accuracy than the traditional statistical model. However, it showed some lower accuracy than recently published research. Data attributes can be supplemented in aired time and historical ratings in order to improve the accuracy.

- **Originality / value**: Two different sentiment analysis model were introduced in the article, with novel method. Both word2vec + clustering method and unigram scoring method are viable for text article analysis analysing and converting into quantised measure for further Machine Learning application in the future.

Acknowledgements

I would like to express my most thank to Tillman E. Weyde, my supervisor and leader in my journey of Data Science. It was his video clip leading me to journey of Data Science in this University and as well as finishing with his supervisory. I would like to thank my parent for supporting me to able to have this chance in London. I also express gratitude to Mr. Je-young Ko, for supporting me with his reference. Also I express thank for Revd. Thomas Soon-Han Choi and God offering valuable chance and life blessed in London.

Table of Contents

1 Introduction and Objectives

1.1 Prediction of TV show ratings: Background

TV show ratings are widely used for long time for pricing airtime for commercials by measuring how many target people are going to exposed to an advertisement on-air. It is quite important since billion dollar annual budgets are spent to trade airtime depended on this ratings. Broadcasters sell advertisement spot based on this predicted ratings and media buyers are using ratings for calculating the effectiveness projection of the budget spend for a campaign. Gap between predicted and actual ratings makes lower effectiveness on the advertiser side and less revenue maximisation on the Broadcaster side [2]. Thus improvement on accuracy of ratings predictions is not just matter of curiosity but is highly business related matter.

Thus, this project aims the beneficiaries as: advertiser, media buyer, broadcaster, and any third party interested in prediction of TV show ratings.

Currently Machine Learning methods are less introduced to predicting TV show ratings and only small amount of researches were conducted. There are many possibilities to explore on this topic as well as improve previous research results.

1.2 Objectives

The Research Question for the project is: "How can we accurately predict TV show rating using Machine Learning methods and quantisation of the critiques' review?"

To answer to this question, we have generated a set of objectives:

- For predicting TV show ratings, use Machine Learning methods with input elements with general show information as well as modified information.
- Implement several different models and parameters for improving efficiency of the forecast
- Quantise the reviews of the shows using two different methods: Vectorisation + clustering and Scoring with sentiment dataset.
- Implement Grid Search for discovering optimised parameter set.
- Compare predicted results with available similar researches and by grid factor in the model.
- Research what conditions are critical for the accuracy improvement.
- Discover further application expansion by using conducted method.
- Reflect what conditions can be improved by modification of the method.

1.3 Method

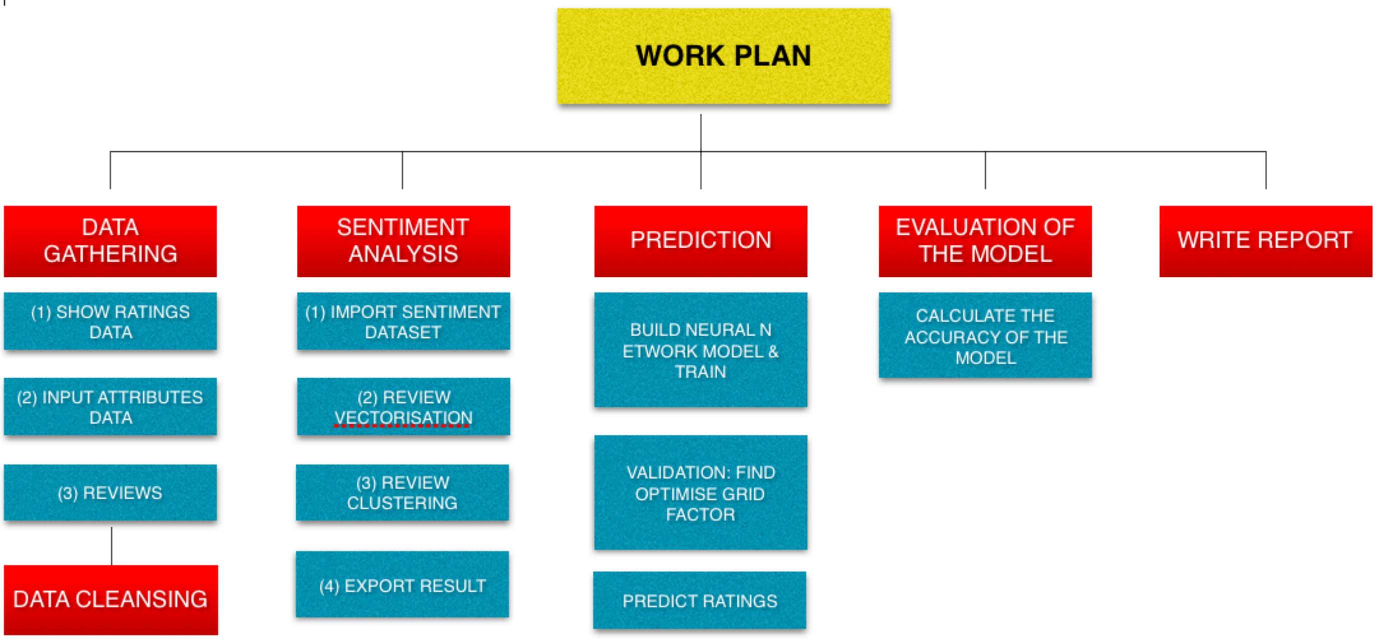


Figure 1Work plan breakdown

For the main prediction model, initially one of Tree models including Decision Tree, Random Forest, and Gradient Boosting Model, was considered for usage but later it has been replaced to Regressor Multilayer Perceptron (Neural Network) model. One of the reason for this change of model was combining sentiment analysis model and the main model as well as Neural Network model is expected to have higher accuracy and much convenient for the current set-up environment.

So Regressor Neural Network was used for the ratings calculation, with input attributes of Genre, Cast, Executive Producers, Network, Premiere Date and quantised review. Average ratings over episodes will be calculated since the objective is focused on using review quantisation, not the time axis analysis. 6 target ratings are going to be used as an output including A18-49 (All/Male and Female age from 18 to 49), M18-49 (Men age from 18 to 49), and W18-34 (Women age from 18 to 34)

For the sentiment analysis, initially Support Vector Machine (SVM) was selected for review quantisation model. But later word2vec was introduced in order to efficiently analyse reviews with pre-built model combined with K-means clustering skill. Alternatively pre-scored sentiment word set (unigram) was used also for predicting whether review is positive or negative output with continuous value.

Also classification problem was introduced for predicting three different target infered from the show ratings. For this classification problem, Decision Tree model was viable for conducting prediction in fast speed and simple implementation.

1.4 Structure of the report

Chapter 1 - [Introduction and Objectives] In chapter 1, we have explained why TV show ratings are eligible for deeper analysis as well as its commercial value. Research Question – 'How can we accurately predict TV show ratings with quantisation of the review?' and Project Objectives are listed as well as work breakdown chart and major method changes during the project work.

Chapter 2 – [Context] In chapter 2, previous relevant researches are referred compared to context of this research and its direction.
In section 2.1. published Machine Learning prediction for TV show ratings are presented and looked down how far they went currently.
In section 2.2. Star Ratings Predictions by Machine Learning methods were described compared with ratings predictions in 2.1. and contextualised how different they are and its context.
In section 2.3. Sentiment analysis topics were introduced including word2vec vectorisation, K-means clustering, sentiment word bag scoring and stopword removing.
In section 2.4. models are discussed for the prediction including Neural Network, Tree model and Support Vector Machine.
In section 2.5. Evaluation metrics are discussed to measure the accuracy and the results.

Chapter 3 – [Methods] Methods implemented in this research were described including 3.1. Data Importing, 3.2. Data Cleansing, 3.3 / 3.4 Quantisation of the Reviews, and 3.5. Prediction with Neural Network Model.
In section 3.1. detailed work of data importing and cleansing were described including application of external API and parsing (beautifulsoup4).
In section 3.2. Hashing and cleansing including detection of holiday were described. Preprocessing process was written here.
In section 3.3. and 3.4. Two different quantisation methods for review were explained. First one is implementing Google's word2vector prebuilt model for vectorising plus K-means clustering. Second one is implementing pre-built sentiment word dataset to score the whole text.
In section 3.5. modelling process were described. Neural Network model were implemented, Regressor and Classifier using Python library. Also segmentation of the target and tree model implementation was also conducted but this will be discussed in section 4.3.2 since it was added during the research process.
In section 3.6. evaluation result was presented with accuracy figures such as RMSE (Root Mean Square Error), R^2 (R-square) and MAD (Mean Absolute Deviation).

Chapter 4 – [Results] Result was analysed and critically evaluated for each task performed in sections in chapter 3. This includes, 4.1. Implementaiton 4.2. Compare accuracy between unigram and word2vec + clustering, 4.3. Classification to 3 different targets. Grid Search and different model trials are described.

In section 4.1. initial implementation result and the accuracy were presented. RMSE was around 0.75 and R2 was around 0.65. This was Neural Network Regressor prediction.
In section 4.2. result on two different quantisation methods in sentiment analysis were presented, compared with each other.

In section 4.3. problem was converted into classifying to three different segmented target from continuous interval (ratings) prediction. Three different models, including Neural Network, Decision Tree and Random Forest were discussed.
In section 4.4. parameter selection / grid search result was discussed.
In section 4.5. model aggregation for six targets were discussed. In the data there were six targets different in age and sex. I.e., M18-49 means Men aged from 18 to 49 years old. There are six categories such as this and at once the whole six targets were predicted by Neural Network.

Chapter 5 – [Discussion]
In section 5.1. in this section the reason was discussed why the initial result's accuracy is not catching compared to Jinsong Cui's research [1]. Several reasons were discussed about this.
In section 5.2. it is discussed why unigram model is outperforming word2vec + clustering model. Sentiment word filtering was pointed out as one of the reason.
In section 5.3. classification problem was discussed implemented in section 4.3. Three different models, Neural Network classifier, Decision Tree and Random Forest were compared with the results.
In section 5.4. Stopword removing effect was spotlighted.
In section 5.5. Grid search / parameter choice result was discussed with regression/classification prediction and with different models.
In section 5.6. the effect of Decision Tree model was discussed.
In section 5.7. target aggregation for one model was discussed.

Chapter 6 – [Evaluation, Reflections, and Conclusions]
In section 6.1. the value of unigram approach sentiment analysis was discussed.
In section 6.2. the effectiveness of the tree models on the task were spotlighted.
In section 6.3. supplement on the time band data and possibility of the time series analysis were discussed.
In section 6.4. finding out the important factors that making difference in ratings, model analysis is discussed.
In section 6.5. some reflections, possible improvements if the research was re-done, were listed.
In section 6.6 transition on TV viewing behaviour and audience measurement, where the audience measurement and TV show rating are going, has been viewed.

2 Critical context

TV show rating had been used for setting price of advertisement spot on television broadcasting market, which is mostly major income of broadcasters. On the broadcaster side, it is very important to accurately predict the TV show ratings since the ratings are highly reflected to their total income. On the advertiser side, show rating is also extremely important to maximise their campaign delivery to targeted audience. Malpredicting of the ratings results loss of money or effciency either broadcaster or advertiser side.[2] There were number of scientific prediction research for TV show rating from decades ago. Mostly those were using statistical and regression modelling requiring very strict input as well as fixed method. Unless until a year ago, no Machine Learning method were conducted as published to general. As this is early stage, there is many possibility to conduct research as well as weaving recent technologies from computer science to them.

2.1 TV Rating Predictions with Machine Learning models

Traditionally TV ratings are being predicted by mathematical / probabilistic methods including linear / stochastic model until recently [2], fed with historical ratings and show attributes such as cast, genre, channel and historical ratings. Traditional methods usually require complex and enduring mathematical modelling / calculations which needs rigorous definitions and adapt for certain usage.
However less than a year ago, Machine Learning methods were first introduced to the TV rating prediction. In October 2016, Jinsong Cui et al. at Nielsen US [1], applied GBM (Gradient Boosting Machine) tree model for TV show rating prediction task, with input attributes of historical ratings, show attributes, social media ratings, and marketing spend, etc. Still it is very early stage TV show ratings prediction by machine learning method and above research is pioneer one. I expect there is much possibility of explore in TV show rating prediction with Machine Learning methods. Generally Tree Models are viable for ratings predictions reported from several research, but expansions are possible including using Neural Network, sentiment analysis, demographics, etc. One can adapt different models, different input factor, and other innovative approach for this ratings prediction.

2.2 Star Ratings Predictions by Machine Learning

Similarly but differently, star ratings prediction on movie by machine learning method is relatively widely used in the scientific world. Many movie database sites provide 5-star ratings for a movie and Tree models and Support Vector Machine models have been used for star rating predictions. [2]

The difference between star ratings and TV ratings are passivity of the rating people. Star ratings are produced by audiences by voting how subjectively they felt about the show but TV show ratings are generated by their viewing behaviour, not voting. So one is passive and one is active. Star ratings are not a measurement but a aggregated score measuring opinion of the public audience, for selectively representing who wanted to actively vote. In contrast, TV show ratings is measurement of percentage of households using television, which is behavioral statistical figure.

Usually the context of the star rating prediction is to 'recommend' proper movie to the user in different characteristics. 'Collaborative methods', which searches for similar user and similar movie based on past ratings are used combined with star ratings.[9]

There is some similarity between star ratings prediction and TV ratings predictions, since both use movie/show attributes and decision trees. Our research is not using star-ratings nor collaborative methods.

2.3 Sentiment analysis

Some Twitter sentiment analysis was conducted using machine learning techniques. Neethu S. et al.[5], conducted sentiment analysis in Twitter using machine learning techniques. Tags and words are classified as 'positive' or 'negative' by using pre-built word bag and classified either positive or negative by machine learning model (Naive Bayes, Maximum Entropy, and SVM). Bo Pang et al.,[5] also conducted similar classification but accuracy was not good.

Separately sentiment analysis using Google's word2vec were introduced. [4] Some simple research were introduced to sentiment analysis [7] but currently sentiment analysis on TV show reviews with Google's word2vec is innovative. Google shared pre-built word2vec to public which makes able to 'vectorise' a word on a multidimensional space. A word can be mapped to a point in a 300-D space. Similar words are plausibly having short distance between them. By converting every words in a review to vectors in a multidimensional space and by averaging it, we can have a expression of a review on a multidimensional space. Vectorised reviews can be clustered using K-means and each clustered groups highly possibly having similar characteristics. This 'groups' can be a input factor to a Neural Network model for rating prediction.

Separately, another different sentiment classification was considered which is using unigram scores. For each review, -10.0 ~ +10.0 scored 54,129 sentiment word bags (unigram) [6] were imported from National Research Council Canada. After moving stopwords such as 'a' and 'the', reviews were parsed into words and each sentiment words were converted to scores using unigrams. The aggregated 'point' represent the positivity / negativity of the review on the show. This has been used as an input vector on the Neural Network model.

2.4 Models

For the TV rating predictions, Jinsong Cui et al., indicated Tree models, especially Gradient Boosting Machine is viable when considering accuracy and scalability. [1] Neural Network model was not very much selected for TV rating predictions model since it requires relatively high computing resources. Initially we also planned to use tree model, but had a conclusion Multi-Layer Perceptron model is more convenient considering relatively small dataset and accuracy of the result. We have all quantised input meaning there is possibility for using NN model other than tree models, meaning improvement of the accuracy, as it might be bit small. If we avoid overfitting, we expect there shall be some improvement on the accuracy.

For the sentiment analysis model, adapting additional independent SVM (Support Vector Machine) model was considered for scoring reviews, but it was difficult to line up the clustered reviews from positive one to negative one before the main prediction. So we only used clustered group as non-hierarchical groups and used as an unsequencialised input factor.

2.5 Evaluation Method

As an error calculation measure, various error measuring methods were used in previous works. WAPE (Weighted Mean Absolute Percentage Error) [1] and AAD (Average Absolute Deviation) [3] was also adapted for evaluation for rating forecast. R-square was also used in [1].

For emotional analysis task, simple accuracy calculation was generally used for the estimation [5][8]. Cross Validation was also commonly used for the evaluation, testing different model for different fold of the dataset [1][3][5].

RMSE (Root-Mean Square Error) was used popularly such as in [9], and [3].

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}.$$

$$\mathrm{MAD} = \frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}|$$

$$\mathrm{WAPE} = \left(\frac{1}{n}\sum \frac{|Actual - Forecast|}{|Actual|}\right) * 100$$

Fig 1. Definitions of Error Measures

Confusion matrix was also used to show the relation between the actual and the predicted value. Accuracy was separately calculated.

3 Methods

IBM's Data Science Experience were used as a platform for pySpark notebook. Python 3.5 and Spark 2.1 were selected as standard. For Data Storage, both IBM Data Science Experience's Object storage and Amazon's S3 were both used. Almost all tasks were performed on this platform except some web crawling task using OWASP Zed Attack.

3.1 Data Importing

Data importing was conducted in order to secure relevant data for the research. Some different tools such as Google docs, Diffbot API and OWASP Zed Attack were used to smoothly extract data from the sites. Sample data was gathered from three different sources. 1. Word Association Lexicons Sentiment data (unigram) 2. Rating data from tvratingsguide.com 3. Show attributes and critics' reviews from rottentomatoes.com

3.1.1 Word Association Lexicons

'Lexicons' containing emotional scores of 54,129 terms from -10.0 to 10.0 were imported from National Research Council (NRC) Canada. Agreement were needed to download it.[6]

3.1.2 Ratings data

Ratings data were imported from tvratingsguide.com. On the site were uploaded ratings spreadsheets in forms of google docs excel file. I ran OWASP Zed Attack Proxy 2.6.0 and crawled whole directory of the website and obtained about 70 links indicating each google docs containing mostly specific TV show's rating by target and by episode. From the links google doc's document ID and sheet No.s were extracted and rebuilt to specific link in order to download it as .csv files. They've been imported to the Spark system. [Appendix B.1.]

3.1.3 Show Attributes and Reviews.

TV show titles and seasons were gathered from the ratings data (from above 3.1.2.) and extracted to a dataframe across two columns. Combining two columns with some rules resulted to obtain link of the show addresses in rottentomatoes.com. Such as 'https://www.rottentomatoes.com/tv/good_place/s02' indicates show 'Good Place' Season 02 is located at the above link. This made able us to download source html of the specific TV show attribute page. Source pages are downloaded by pySpark using urllib2 library and parsed by using Beautifulsoup4 library. Each informations including Genre, Cast, Executive Producers and critics' review were imported to a dataframe and saved to .csv file. [Appendix B.2.]

As the Reviews imported above were from different web sites, we used Diffbot API Toolkit to import only relevant texts from complex web pages and remove Ads and unnecessary informations. [Appendix B.3.]

3.2 Data Cleansing

3.2.1 Hashing and vectorising the attributes

Since Neural Network was planned to use for modelling, all string values had to converted into float values. 1. Hashing and 2. Vectorising were used to make to convert strings to numerical values.
Hashing was needed to convert specific names or Genre or Network to numerical values. I.e., putting 'Alex Johnson' to hash() function returns '547' as a hash value indicating 'Alex Johnson' will always have hash value of '547', but not exclusively.
Vectorising was used to combining multiple attributes in a show such as many numbers of Executive Producers to one long vector. Mod N (remainder) of certain number of bucket N was used to make fixed length of the vector. For example, if hash value of Alex, Beckie and Charson was 51, 102 and 203 respectively, their mod 50 values shall be 1,2 and 3 respectively. So we construct 50-dimensional vector [1, 1, 1, 0, 0, 0, 0 … 0, 0] indicating Alex, Beckie and Charson is included in the cast. Overlapping can be happened since hash value 51 , 101, 151.. can be have same mod N value so will be expressed as same vector results. So determining the vector size N is quite considerable to do this job.
Eventually, 'Cast', 'Executive Producers', 'Genre' and 'Network' were hashed and vectorised.

3.2.2 Marking national holiday and day of the week.

By using air date, weekday() and USFederalHolidayCalendar() functions in pandas were used identify the day of the week and if the indicated day was national holiday. Two different columns were added to plot this.

3.2.3 Multiple reviews for one TV show

Since number of elements for input vector shall be remained constant, problem had risen when there is multiple reviews for one show. There was two ways: (1) build one input vector for each of the reviews in one show (2) aggregate multiple reviews to one input vector thus one input vector only representing single show.

We could not average the clustered results since 'Cluster 20' does not mean that it has positive interpretation than 'Cluster 1'. So e.g. there is three reviews clustered to 3, 10, and 13, we cannot average them simply to (3+10+13)/3 = 8.67 since cluster does not meaning score. There was one good method in order to reduce it to one fixed sized vector. We used 20 digit-long vector each digit representing number of the review corresponding to. E.g. if there is one review clustered to group 2 and three reviews clustered to group 5, then the vector shall be look like: [0 1 5 0 0 0 …. 0 0]. The review vector was directly used as an input element for prediction model.

3.3 Quantisation of the reviews

Initially quantisation of the review was discussed consideration of introducing two different models, one for review quantising and other for the show rate prediction. During the supervisor meeting, new Neural Network model combined two models were discussed, using word2vec for vectorising review and including clustering process after the vectorising. Also more sophisticated methods were discussed including PCA and dimension reduction. So we turned the direction to build one model + word2vector clustering.
In order to do this, word2vec model was used to 'quantise' parsed review for 298 elements and saved as a vector form. Google's word2vec model was built in order to map each word as a vector representation on a multidimensional space. Short distance between two words means high similarity between the words. [4] Gensim Library and Amazon's S3 storage was used in order to carry 3GB large model properly.
We can just 'average' the vectors used in one review, in order to find the 'centre point' of the words on the vector space. Stopword removing was conducted in order to 'cleanse' meaningless aspect such as 'the' and 'a'. After this, the centre point of the review will show the location or propensity of the review by its location on the multidimensional space. We gather this by clustering the 'centre points' to 20 different clusters by K-means method. This means review which are in same cluster, meaning similar property or opinion in high probability.
The number of element in input vector were issued since Neural Network model usually requires fixed size of the input vector. On the quantisation of the review, for single movie there were irregular number of movie reviews and when it was clustered, it could not be averaged since the clustered results are not numerical meaning value. So the clustered result will be built into one short input vector for the Neural Network model. Similarly vectorising will be conducted same as 3.2.1. And one 20-dimensional fixed-sized input vector will be constructed for an aspect. If one review existed and it belongs to 3rd cluster, the 20-D input vector shall be: [0 0 1 0… 0 0 0]. This will be one part of long input vector for the Neural Network model.

3.4 Quantisation of the reviews using sentiment word dataset

Two separate quantisation were proceeded, which the second one is quantisation of the review using sentiment word dataset (unigram). Unigram was downloaded included in 3.1.1. Word Associated Lexicons and have score of -10.0 ~ +10.0 for about 55,000 terms. By using this, each review were 'scored' by matching corresponding score to words after removing stopwords. The sum of scores were averaged by dividing number of terms thus results to one figure. This figure shows how positively (negatively) the review evaluate the show approximately since it calculates positivity / negativity of the whole article by summing each word's term score. Once more multiple reviews in one show were aggregated by averaging each review's scores and deducted to one figure for one show.

3.5 Prediction with Neural Network model

Regressor Neural Network (Multi-layer Perceptron) model is used for predicting final TV show ratings. Initially Support Vector Machine and Tree Models - such as GBM - are considered for the prediction where scoring of the review were considered as a process with a different model. After modifying quantisation to word2vec plus clustering, there were no need for using separate model for quantisation and much complex tool were needed to adapt to new input layer as well as become powerful method. Technically Scikit-learn's Multi-layer

Perceptron Regressor (sknn.mlp.Regressor) was used to train the model and to predict ratings. L2 regularisation technique was also used to avoid overfitting.

3.6 Evaluation

Evaluation metrics used for this research's implementation are: RMSE (Root Mean Square Error), R^2 (R-square) as well as MAD (Mean Absolute Deviation) [11]. As well as RMSE, MSE (Mean Squared Error) are also used in some context. RMSE and MAD are used for calculating how much errors has been occurred in the prediction. MAD is also used for comparing other researches, such as Jinsong et al.,[1]'s with WAPE (Weight Average Percentage Error :). R^2 is used to show how well prediction calculated along real value.
For classification used in 4.3.2. and 4.3.3., confusion matrix [10] is also used as well as MSE.

$$\mathrm{MAD} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

$$\mathrm{WAPE} = \left(\frac{1}{n}\sum\frac{|Actual - Forecast|}{|Actual|}\right) * 100$$

Fig. Definitions of Error Measures

Automated confusion matrix library was used for calculating and deriving f1-score and other relevant figrues. Also observing raw results were bi-conducted for eidetic interpretation.

4 Results

4.1 Implementation of Neural Network Model.

Using Spark Notebook, scikit-learn's Neural Network library was used to implement machine learning method. In sknn.mlp library, 'Regressor' model was used to predict the ratings by inputting 601 dimensional numerical input vector formed with DataFrame or List. (both is acceptable). Train and Test set ratio was set to 0.85 : 0.15 in order to secure minimum number of the test set. No. of hidden layers was set to 1300 and no. of iterations with 700. L2 Regularization was used in order to avoid overfitting. Running time was about 6 minutes and learning rate and weight decay (L2 figure) had to fit to specific range in order to run the model since it occurred error if it went outside of the range. Target (y) had 6 different targets, which is A18-49, M18-49, W18-49, A18-34, M18-34, and W18-34. So 6 different target ratings were tested with separate training and models.

Predicted results are shown in Figure 2, as well as resulting RMSE (Root Mean Square Error)=0.7504, R^2=0.6509, and MAE (Mean Absolute Error) = 0.5953. It is moderate-high value but there is some inaccurate result affecting lowering the accuracy. We will discuss about the result in Section 5.1.

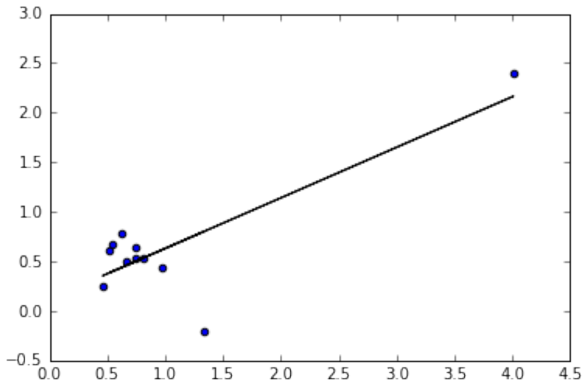


Figure 2 Word2vec prediction for A18-34 target. (x-axis is prediction, y-axis is actual ratings)

4.2 Compare accuracy between unigram and word2vec + clustering

Two different sentiment analysis model were adapted for this research. Word2vec vectorisation + clustering and unigram method. First, word2vec + clustering methods were implemented with same parameter settings with 4.1. Above. Second, similar Neural Network model with unigram (scoring review article using pre-scored sentiment word bag). By comparison, initially unigram's accuracy was little bit higher than word2vec + clustering method. (Appendix C.4.)

	Word2Vec + Clustering	Unigram
RMSE	0.7504	0.7113

R2	0.6509	0.6864
MAE	0.5953	0.5793

Table 1 Accuracy of two different sentiment analysis model

In detail, we acquired accuracy rate: RMSE (Mean Square Error)=0.7504, R^2=0.6509, and MAE (Mean Absolute Error) = 0.5953 for the second method. Compared to Jinsong et al.,[1]'s WAPE(Weight Mean Absolute Percentage Error) 0.083 (can be compared with MAE since it's similar derivation), it is still quite higher than the result.

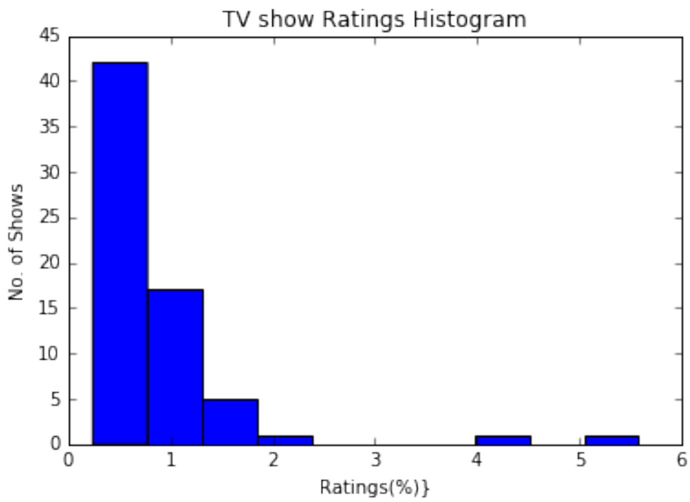
Secondly, unigram method was used for scoring the reviews. With parameters hidden layers=1300, learning rate=0.000005 and n_iter=1400. Error rate was: MSE = 0.5059, R^2=0.6864 , and MAE = 0.5793.

4.3 Classification to 3 different targets (blockbuster / moderate popular / unpopular) – how does the accuracy change

In this stage, predicting rating is converted to predicting segmented 'target' for seeing different aspect of classification. Ratings were segmented to 3 targets, 'blockbuster' / 'moderate popular' / 'unpopular', as shown in Appendix C.3. After that, Neural Network model, Decision Tree and Random Forest are implemented for classifying three different targets.

4.3.1 Using Neural Network Model

In this section TV show ratings are grouped to three different targets, which is: 'unpopular', 'moderate popular', and 'blockbuster'. By investing the data visually by histogram below, three intervals are defined: [0, 1.5], [1.5, 3], and [3, ∞]. Each represents 'unpopular', 'moderate popular', and 'blockbuster'. Thus the problem has been converted from regression problem to classification problem in this section 4.3.



First we have continued to implementing Neural Network model, changing it to Classifier tool from Regressor tools. Definitely scikit-learn's Neural Network tool (sknn.mlp Classifier) was used.

One singularity in using Neural Network Classifier was much shorter running time and less using computational resource since the calculation was simplified changed from continuous target to discretised target. However, the accuracy of prediction was not so good in some reason. Despite of continuous research, it didn't improve as well as it was hard to find the reason, so we moved to decision tree model for this classification problem.

4.3.2 Using Decision Tree Model

Decision Tree model was also introduced for this classification problem. (Appendix C.4.) Also model was quicker than regressor model, as well as accompanying some accuracy increase. The prediction accuracy was 83% including MSE=0.0909. Model ran in almost instance compared to Neural Network Regression and classification. It was much suitable for detecting blockbusters. Usually blockbusters are well detected, but 'moderate' are sometimes classified as different category presuming since it is quite close interval to 'unpopular'. Decision Tree model was successful for classifying ratings to three categories having high probability of prediction. We have decided widen this to multiple Decision Trees, Random Forest.

	Precision	Recall	F1-score	Support
unpopular	0.90	1.00	0.95	9
moderate popular	0.00	0.00	0.00	1
blockbuster	1.00	1.00	1.00	1
avg / total	0.83	0.91	0.87	1

Table 2 Result for Decision Tree model classification for W18_49 target.

4.3.3 Using Random Forest Model

We expanded Decision Tree to Random Forest model in order to try different method. The result was slightly disappearing. Random Forest was less accurate in especially determining blockbusters and moderate populars. I presume this was because Random Forest's voting mechanism dilutes the signal from one specific decision tree catching blockbusters. Blockbuster properties are caught by small number of decision trees, but if there were more number of decision trees voting for 'not blockbuster', the 'signal' from the blockbuster created from certain decision trees may erased by overlapping in the voting phase. For example, if there was single Decision Tree, it might easily catch the 'blockbuster' by its own mechanism. However, in Random Forest, if two decision trees vote for 'blockbuster' and other three decision trees vote for 'not blockbuster' then the signal disappears. So we concluded that 'smoothing' sharp signal / results by aggregating multiple decision trees are having negative effect on the prediction.

4.4 Grid Search (Parameter Selection)

For the Neural Network Model, parameter selection is critical since it determines the accuracy of the result as well as efficiency of the model. Initially no hidden-layer with 1300 neurons were tried, which worked well. Also one hidden layer and two hidden layers were tried, but not much accuracy change were observed, so we decided to use model with one input layer and one output layer. Activation function was 'Rectified' for the input and hidden layer ($f(x) = x^+ = \max(0, x)$) and 'linear' for the output layer.

GridSearchCV function from scikit-learn but was less meaningful since number of layer should be adjusted manually and other attributes, (i.e. learning rate), had to be in accurate range in order to avoid error. So grid search also was conducted by manually, adjusting parameters small by small in order to optimise the model.

Also L2 regularisation was conducted in order to avoid overfitting. Simply setting 'L2' parameter and weight_decay value around $1/10^{th}$ of learning rate made it possible.

4.5 Model Aggregation

Model Aggregation was conducted for 6 different targets ‘A18-49’, ‘M18-49’, ‘W18-49’, ‘A18-34’, M18-34’, and ‘W18-34’. Simply 6 target columns were integrated to one DataFrame and for setup the output layer of the Nerual Network was set to ‘sigmoid function’ for activation function, as indicated in the technical manual. Aggregation was convenient for implementing 6 targets at once and using one model for multiple target predictions. This is important since six different prediction model can be aggregated one model, resulting different target gets different ratings.

5 Discussion

5.1 Accuracy of the Neural Network model

For evaluating how well the model had performed, initial Neural Network prediction model’s accuracies are showed as followed: MSE (Mean Square Error)=0.5631, R^2=0.6509, and MAE (Mean Absolute Error) = 0.5953 for ‘A18-49’ target. This is way higher accuracy than the result of Denny Meyer et al. [12], which might be classified as a traditional approach and showed MSE of 1.698~1.114. However, by comparing with some recent research such as Jinsong et al. [1]’s, it is not catching up to its R-score (R^2=0.95). We estimate this for opulent dataset originated from Nielsen’s corporate nature, as well as some set-up difference. Attributes such as time of air and marketing spend-on were unable to collect for us. Further research might include richer information for making higher figure.

5.2 Performance comparison of two sentiment analysis model

In this experiment, two different sentiment analysis models were implemented. Between those, unigram model had shown slightly higher accuracy than the word2vec + clustering model. We think this that the scoring on only the ‘sentiment’ word on a specific review was little bit better than vectorising ‘every’ words in a review. This means, filtering for sentiment words derived better effect than using general word bag, at least in this case, since it is predicting positivity / negativity of the movie, which is highly related to human’s emotion. So it should be worthwhile to filter and collect only sentiment words from the review and implement word2vec + clustering methods, which might perform more powerful ability. Sentiment analysis models were successfully implemented and showed possibility of extended adapt in many different fields, by quantising long text (more than hundred words of) article to clustered group or a score.

5.3 Classification and regression

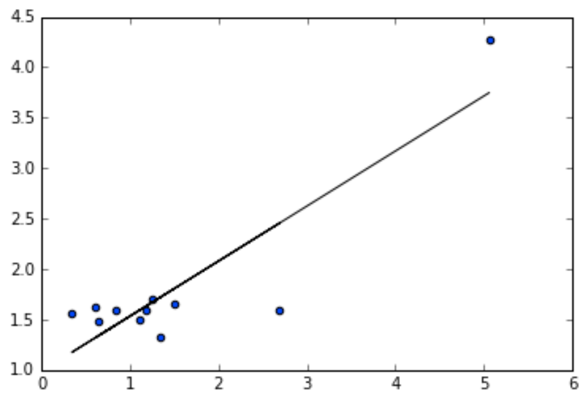
In section 4.3.2, ‘classification’ was conducted reducing the target from interval to three different fixed baskets. (‘unpopular’ / ‘moderate’ / ‘blockbuster’) In some reason, the Neural Network model had not well worked out for the classification. (Appendix C.2.) It had not detect ‘moderate’ or ‘blockbuster’ well. However, the tree model showed higher performance for the classification. Tree model well detected ‘blockbuster’ categorised movie as well as having simple code and fast running time. For the unavailability of the Neural Network model, we are not sure what the reason is yet. Maybe some parameter adjustment can find good combination for the perfect model. Or trying different python library might be work as well. Tree model was well detecting the critical elements from the input vector for blockbusters. We carefully presume this as Tree model accepts the value as discrete value, when NN model treats input element as a continuous value and sometimes multiplying weights for the input factor. This results tree model's processing more 'clear' for the discrete input factor such as in this research.

5.4 Stopword removing

Stopword filtering is discovered to be critical for the sentiment review analysis. Stopword usually indicates set of words usually removed in the state of preprocessing of Natural Language Processing (NLP), such as removing 'and', 'or', 'to', etc. [13] After the stopword removing, there was a significant accuracy improvement (0.1~0.2 of R^2 points) for the Neural Network model prediction in section 4.2. We also estimate that adapting filtering only to relevant word sets, such as unigram sets to word2vec model, will have viable impact for the achievement of the goal. Data pre-processing such as stopword removing is viable and inevitable for sentiment analysis.

5.5 Grid Search / Parameter choice result

Grid Search, which means looking for optimised parameters by implementing grid of parameters to the Machine Learning model, was applied throughout using both automatic library and manually. Dozens of times of experiment was applied for optimising the model. First, for Neural Network Model, number of hidden layer and number of units in each layer were critical parameter to be optimised. For the number of layer, from 0 to 2 layers are experimented and 0 hidden layers (meaning there only exists Input Layer and Output Layer) showed much accuracy. More important factor was no. of training, should be exceed some point (around 1,400) since the learning rate required from the model was quite small (0.000005) as well as L2 regularisation (0.0000003). Each layer needed around 1,300 units for successful prediction. Looking at the result chart (Fig), the model is surely detecting blockbuster (high ratings than usual) but underestimates its ratings. In the Fig. it shows the model estimated the highest ratings show (actual around 4.0) to 2.3%. This kind of underestimated predicted ratings blockbusters continued. I estimate this as blockbusters are able to predict, but how high they are going to is quite difficult to predict. In other words, it is possible that one show would show more than 2% of the ratings, but does not know if it would be 4% or 6%. It need to be more research on this for predicting blockbusters accurate ratings, as well as providing more input attributes relevant to them.



Unigram prediction for A18-49 target. (x-axis is prediction, y-axis is actual ratings)

5.6 Tree models

Tree models were tested during the 'classification' test in section 4.3.2. and 4.3.3 in order to compare accuracy and performance for the prediction. Accuracy were bit higher than Neural Network model, comparing RMSE. Tree model's RMSE was 0.40 and NN model's RMSE was 0.79. Less training times was consumed than Neural Network Classifier taking less than a few seconds. Blockbusters are well predicted but small portion of 'unpopular' ones were classified as 'moderate popular' presuming because of its interval closeness. For the classification task, tree model was efficient, accurate and even economical than the Neural Network model.

Expanding to Random Forest had some reverse effect in accuracy for predicting blockbusters. I carefully presume this because of the multi-tree structure of the random forest. The 'signal' detecting blockbuster generated from each trees are concealed by votes by multiple trees. For example, one tree successfully detected blockbusters, in Random Forest it reflected on the forecast directly. However, in Random Forest, even if one tree detected the blockbuster, if other trees denies to detect it as a blockbuster, the prediction goes wrong. In this reason, blockbuster detection and accurate classification is better in single Decision Tree than a Random Forest.

6 Evaluation, Reflections, and Conclusions

This project initially aimed the improvement of the accuracy for the prediction of the TV show ratings. In some aspect we have succeeded since we have achieved some accuracy improvement than some traditional statistical methods. However, there is some more possibility to increase the accuracy using more aspects of data. Some research[1]'s result showing bit better accuracy in this result in different situation. But sentiment analysis is novel and critical approach for evaluating how the opinion on the movie can be quantised. Both word2vec + clustering and unigram approach is valid for further application on similar project. It can be utilised for, i.e. quantising review in order to estimate how people value for the current stock price, etc. It is viable method for the prediction using Machine Learning methods.
We consider this result can be also useful for the commercial usage for the prediction of the ratings.

6.1 Sentiment analysis

Using unigram for estimating response of the viewers were never tried before especially for predicting show ratings. Some of the researches included social media ratings such as 'Facebook Likes' for their prediction model [2] but scoring whole articles linking with audience measurement were rarely conducted. This approach can be used other purpose such as: stock price prediction combined with social media quantisation, etc.

6.2 Tree model regression

Tree models have strong performance for predicting ratings, especially in classification. It shows much faster training period

As sentiment analysis results are compressed to 1-dimensional or 2-dimensional vector, tree models can be still good candidate for the prediction model. It takes less computational resources as well as its fast speed and giving more capacity on larger dataset. In depth, Multi-Layer Perceptron and Tree models have just difference in how they divide multidimensional planes (e.g. tree models divide multidimensional planes orthogonally, however Multi-Layer Perceptron doesn't), where tree models much reduce hardware resource usage.

6.3 Time band information and time-series prediction

Show ratings are highly depend on the time which it has been aired. Rating difference frequently occurs in evening prime time and morning time although it is completely same content. In this research dataset time band is missing so need more information about in what time band it has been aired in order to improve the performance.

Also it can be breakdown into prediction to each episode, using historical data and time series analysis. It will be more commercially attractive and close to business use. Quantised reviews can be still useful for predicting show's average ability to gather audiences. Also currently it is omitted that if the review was published before / or after the air, next step would be to include such factors for advanced level.

6.4 Model Analysis

Model analysis can be conducted for what attributes are how much affecting on the ratings. For example, the time of air or specific movie director, or combination of those can be critical for the blockbuster level show ratings. If certain criteria almost always 'hits' movie viewing, it can be considered in a business process as well as investment, as well as projection before buying certain show or copyright.

6.5 Improvements for the further plan

- TF-IDF can be used for the quantisation for analysing the review. TF-IDF itself can be an input vector learning the opinion from the review. It might be more detailed analysis on the article itself. Also it can be combined with word2vec clustering.
- Dimensionality Reduction can be conducted in order to make input vector more compact. Since input vector is 601 dimensions, it would be good to reduce it by PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) or other dimension reduction methods.

6.6 Transition on TV viewing behaviour and audience measurement

As current TV viewing behaviours are transition to diversely such as Netflix and online demand videos, there is high expansion of the target factor and even possibility of predicting ROI (Return On Investment). In this context, recommendation systems can be used for targeting intentionally selected customers and also this leads my model to develop from target ratings to ‘number of targeted customers’. Additionally, audience measurement is related with Return On Investments (ROI) and it is possible to expand this research to the direction. Other development of media consuming behaviour and pattern can be considered as a further expansion.

6.7 Conclusion

In conclusion, TV show ratings are accurately predicted but there are some points to improve such as time band data supplement and model adjustment. Accuracies are achieved and there is enough capacity adapted to the business environment. Further ways to develop is adjust to technical development and shifting media viewing behaviour and prediction and implementing relevant measures. ROIs can be one of the fine target to reach, as well as diverging targets up to specific needs for the advertisers and consumers. Machine Learning can conduct significant role in that case and digital world is coming. Mobile and future media will be detailed for the target segmentation and reaching to the audience. Neural Network and the more advanced technology will be applied to the prediction and the segmentation. It will be interesting and fulfilling to chase this development and work with it. This research is a basis for the future.

7 References

[1] Sereday, S. & Cui, J., 2017. Using Machine Learning to predict future TV ratings. Nielsen Journal of Measurement, 1(3).
[2] Roberto Pagano Politecnico et al., Prediction of TV ratings with dynamic models.
[3] Marovic, M., 2011. Automatic movie ratings prediction using machine learning. MIPRO.
[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.
[5] Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02.
[6] Kiritchenko, S., Zhu, X., Mohammad, S. (2014). Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research, 50:723-762, 2014.
[7] Haixia L., Sentiment Analysis of Citations Using Word2vec. School Of Computer Science, University of Nottingham Malaysia Campus, 2017.
[8] Neethu, M.S., Sentiment Analysis in Twitter using Machine Learning Techniques. IEEE, 31661.
[9] Spiegel, S. & Kunegis, J., 2009. Hydra: A Hybrid Recommender System. CNIKM.
[10] Sammut C, Webb GI. Encyclopedia of machine learning. New York ; London: Springer; 2010:1030.
[11] Mendenhall W, Scheaffer RL. Mathematical statistics with applications. North Scituate, Mass.: Duxbury Press; 1973:A88.
[12] Meyer D, Hyndman R. The accuracy of television network rating forecasts : The effects of data aggregation and alternative models. . 2006.
[13] Manning, C.D., Raghavan, P., Schütze Hinrich, et al. (2010) <Introduction to Information Retrieval>. Post & Telecom Press

I. Appendix A: Project Proposal for MSc in Data Science

Name: Jeremy Jaemin Jung
E-mail address: jaemin.jung@city.ac.uk
Contact Phone number: +44 7447 360854
Project Title: TV show ratings prediction by Machine Learning Methods with quantisation of the review.
Supervisor: Tillman E. Weyde

<TV Show rating prediction by Machine Learning methods with quantisation of the review> - Project Proposal

Jaemin Jeremy Jung.

1. Introduction

The purpose of this project is to 'accurately' predict TV show ratings using Machine Learning model and emotional analysis of the review. Machine Learning is in early stage of adapting in TV show ratings prediction and there is substantial potential to improve the accuracy of the forecast. It is very important to predict the ratings since the forecasted ratings are used for the set of price of ad-spot and calculation of the effectiveness of the media buying. The products will be generated model, generated predicted ratings, and knowledge on the rating prediction. The beneficiaries will be media buyers and sellers in the industry, since both are using TV rating as a measure of effectiveness of ad-execution, or as a criterion of ad-spot price setting. TV rating is such important that ad-spot prices are decided before the air, and later the effectiveness and profit are calculated by the actual ratings, in the industry.

As adapting Machine Learning technique on the TV show rating prediction, we would like to add emotional analysis in addition to previous models using the properties of the show. The critics' and audiences' review of the show will be analysed with emotional dataset with Machine Learning method and will be reflected to the prediction model.

The Research Question is: "How can we accurately predict TV show rating using Machine Learning method and user review emotional analysis?"

2. Critical Context

(a) Approach

For recommendation and rating prediction tasks, there is three different approaches: content based-method, collaborative method, and hybrid method [5]. Collaborative methods are using ratings of others, similar users [6]. As it is for individual user actions' prediction, it is not valid for view ratings on this research. We would like to use content based approach, which is using item's features and user's affinities. In this case user's affinities are calculated from the emotional analysis from reviews.

(b) Models

V. Rating prediction model

Tree models have been widely used in previous researches for recommendation and rating prediction problem, as well as been proven as significantly effective method for the prediction. Regression tree is used in M. Marovic et al [6], vectorising the properties of the movie to binary variables, e.g. if the movie is horror, the corresponding vector will have value 1, otherwise 0.
Gradient Boosting Machine (GBM) has been proven as the most effective among many models, among linear regression, penalised regression, support vector machines, and neural network, etc [2]. GBM is ensemble of many decision trees, using average rating calculation of many decision tree as final rating.
Thus we are going to pursue tree model including Decision Tree, Random Forest, and GBM for predicting show ratings. Regression model needed to be adapted since we are not encountering with multi-class problem. Also, the result from sentimental analysis from following paragraph B, will be adapted one of the element to input in current model.

B. Emotional analysis model.

Extracting emotional response from articles or reviews are conducted from several previous researches. Bo Pang et al. [7] compared Naive-Bayes, Maximum Entropy, and Support Vector Machines (SVM) and evaluation result shows generally SVM is more accurate than other two methods. M. Neethu [8] also reports that Naive-bayes is slightly underperforming than the two methods, seen it in binary classification. The source of the sentimental analysis was twitter and movie reviews.
We are selecting SVM as a sentimental analysing model provisionally, and it will explained in next chapter 3.(d)

(c) Input argument for the model

Program characteristics were primarily used as input attributes for model in [1] and [2]. Used input attributes in [6] are: (1) Year (2) Director (3) Writer (4) Editor (5) Cinematographer (6) Production Designer (7) Costume Designer (8) Country (9) Production Company (11) Color (12) Sound (13) Running Time (14) Language (15) Actor 1-5 (16) Genre (17) Budget (18) Box office gross (19) Rental Revenue and (20) Average User Rating. Jinsong Cui et al [2] has included similar attributes but more concentrated on historical ratings, since his working company, Nielsen had good accumulated ratings records. Various historical ratings such as commercial / live ratings, unique viewers (reach), average audiences(AA%) and persons or households using TV (PUT/HUT) were used for main input factor. , marketing spend, TV brand effect, and social media ratings.

(d) Evaluation Method

As an error calculation measure, RMSE (Root-Mean Square Error) was used popularly such as in [3], and [6].

WAPE (Weighted Mean Absolute Percentage Error) [2] and AAD (Average Absolute Deviation) [6] was also adapted for evaluation for rating forecast. R-square was also used in [2].

For emotional analysis task, simple accuracy calculation was generally used for the estimation [7][8]. Cross Validation was also commonly used for the evaluation, testing different model for different fold of the dataset [2][6][7].

(e) Recommendation system

One of the context related with this subject is recommendation system. Contrast to predicting whole number or percentage of the viewer in this research, recommendation system targets to forecast individual preference and the rating score. In this reason, collaborative methods are usually adapted for the recommendation system, generating (user) x (item) matrices for recording the scores written by the user. There is some methodological similarity between Recommendation system and Total Rating prediction, such as using dataset, models (sometimes), and evaluation method.

3. Approaches: Methods & Tools for Design, Analysis & Evaluation

(a) Data acquisition

Data usage permission should be gathered from IMDb.com and Nielsen. IMDb.com provides TV programme review and characteristics, while Nielsen generates National TV rating data. Both are critical for the research. Web-crawling might be required if the IMDb data were not able to provided with text files. XML or JSON can be used for the crawling task.

(b) Cleansing

Cleaning should be performed since the origins of the data are different. We need to only include the shows included in the both dataset. In sentimental analysis, reviews without star ratings shall be temporarily excluded for initial learning process.

(c) Tools

Apache Spark is going to be used as a main tool. For web-crawling and other Machine Learning pipeline application, Spark is one of the optimised solution for big data task. IBM's Data Science tool or Seaford server in City University shall be used as hardwares. Both accompanies pySpark tool with Notebook.

In detail, Support Vector Machine Machine Learning library shall be used for emotional analysis. Three different decision tree models (Decision Tree, Random Forest,and Gradient Boosting Machine) are going to be trained for the forecast, and the best performed model shall be chosen by evaluation.

(d) Emotional Analysis

Provisionally SVM is going to be selected as an emotional analysis model chosen by performance test in literature. However, it might be confronted to computer resource problem, for countermeasure, Naive-Bayes will not be excluded for the potential selection. Naive-Bayes also has been proven to be work well in sentimental ML classification task [9].

Support Vector Machine library from pySpark tool shall be utilised for the implementation. Emotional dataset will be imported from University of Florida website (http://csea.phhp.ufl.edu/). This analysis results shall be performed by two part, one for critics and the other for public audience.

(e) Rating prediction

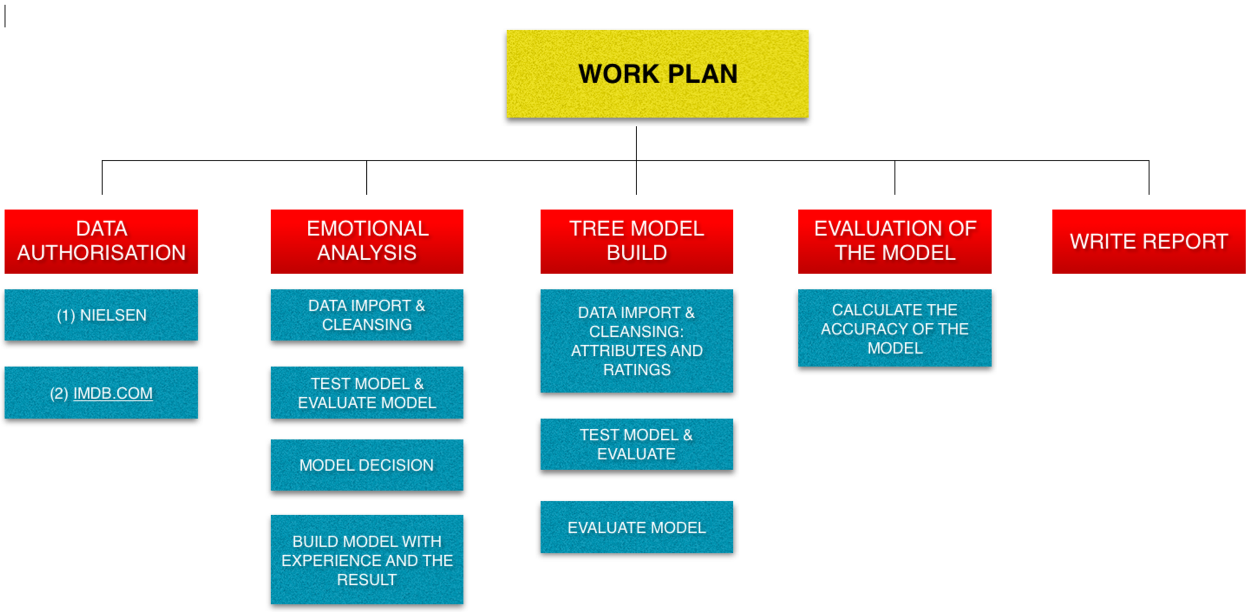
Approximately 40 properties will be input to the Tree models, including sentimental analysis results. The attributes shall be: actors' name, genre, director, historical ratings, air time / date / season, budget spend on making and most of the available multi-class or binary data, etc.

(f) Evaluation

Two separate evaluation methods shall be used for the task (d) Emotional Analysis and task (e) Rating prediction. Root Mean Square Error and simple accuracy calculation will be used for both tasks. ROC Curve will be drawn also. Cross validation is going to be applied only for task (e). WAPE (Weighted Mean Absolute Percentage Error) and AAD (Average Absolute Deviation) shall be additionally used for complement. R-square also will be calculated and plotted for both tasks.

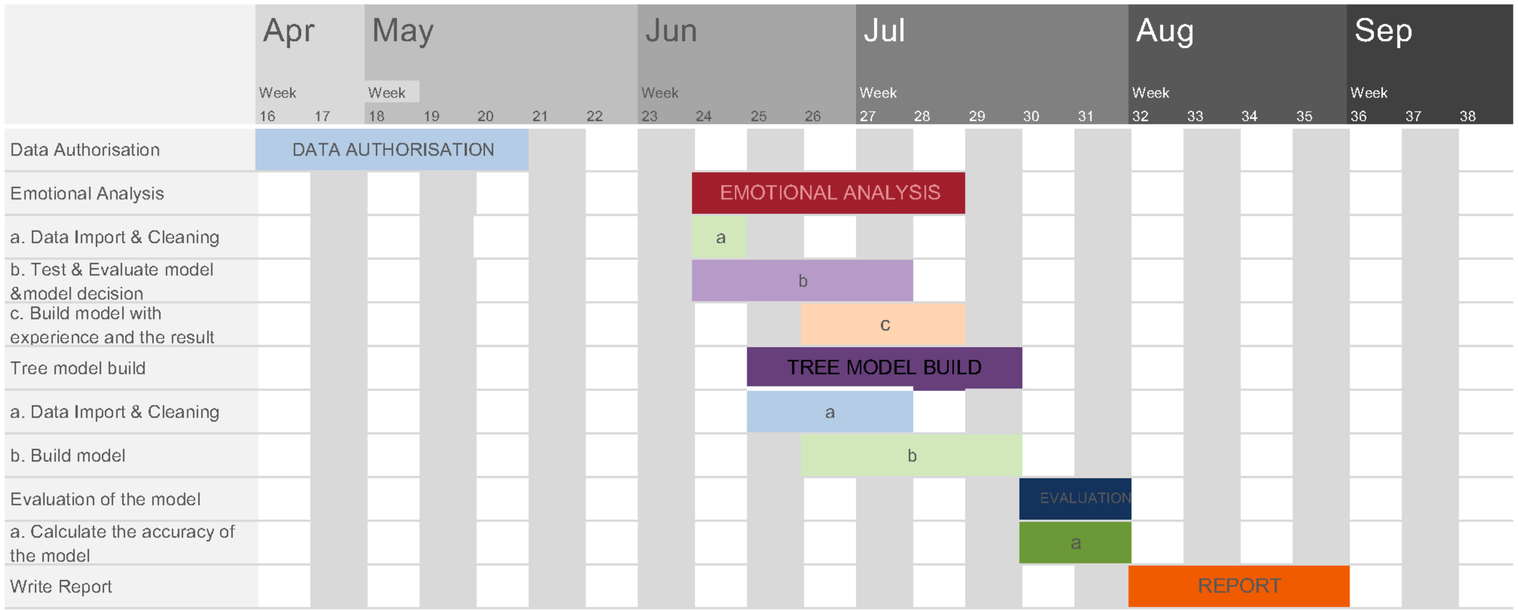
4. Work Plan

a. Request data usage permission to Nielsen US and IMDb.com. The content of the data shall be: 1. Historical ratings of the TV show from Nielsen including target ratings and 2. All text attributes provided IMDb.com in the each TV show information page, including critics' and audiences' reviews.
b. Build emotional review model with Spark combined with emotional word dataset. Data import & cleansing, model build, evaluation, and decision of the model will be conducted.
c. Build tree models for rating prediction with Machine Learning. Cross validation will be conducted. Input attributes is programme characteristics and result from task b.
d. Evaluation shall be performed with accuracy, RMSE, WAPE, AAD, and ROC curve, separately for task b. and c.
e. Write the report.



Project Timeline

Start Date: 4/17/17



5. Risks

Description	Likelihood (1-3)	Consequence (1-5)	Impact (L X C)	Mitigate
Data usage denial from Nielsen	3	5	15	Contact as early as possible. In case of the failure, change the country of Nielsen retry.
Data usage denial from IMDb.com	3	5	15	Contact as early as possible. In case of the failure, contact other similar site such as TV.com or thetvdb.com
Copyright - use of review from critics or users can be confronted with the copyright issue	1	3	3	Resolve issue with legal, data protection task.
Model plagiarism risk - There is similar research conducting on “Sentimental data analysis on Twitter - focused on demographics”.	2	2	4	Clarify the scope of the this article in order to avoid collision.
Model inactivity risk	1	4	4	Cleanse data properly, research model thoroughly, as well as mastering the Spark library.
Schedule delay risk	3	2	6	Check the timeline every moment and ensure where am I

References

[1] Armstrong, N. & Yoon, K., Movie Rating Prediction.
[2] Sereday, S. & Cui, J., 2017. Using Machine Learning to predict future TV ratings. *Nielsen Journal of Measurement*, 1(3).
[3] Spiegel, S. & Kunegis, J., 2009. Hydra: A Hybrid Recommender System. *CNIKM*.
[4] Li, P. & Yamada, S., A movie recommender system based on inductive learning. *IEEE Conference on Cybernetics and Intelligent Systems, 2004.*
[5] Adomavicius, G. & Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp.734–749.
[6] Marovic, M., 2011. Automatic movie ratings prediction using machine learning. *MIPRO*.
[7] Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*.
[8] Neethu, M.S., Sentiment Analysis in Twitter using Machine Learning Techniques. *IEEE*, 31661.
[9] Lewis, D.D., 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. *Machine Learning: ECML-98 Lecture Notes in Computer Science*, pp.4–15.

Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.
In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:
Part A: Ethics Checklist. All students must complete this part. The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.
Part B: Ethics Proportionate Review Form. Students who have answered "no" to questions 1 – 18 and "yes" to question 19 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in this case. The approval may be provisional: the student may need to seek additional approval from the supervisor as the project progresses.

A.1 If your answer to any of the following questions (1 – 3) is YES, you must apply to an appropriate external ethics committee for approval.		*Delete as appropriate*
1.	Does your project require approval from the National Research Ethics Service (NRES)? For example, because you are recruiting current NHS patients or staff? If you are unsure, please check at http://www.hra.nhs.uk/research-community/before-you-apply/determine-which-review-body-approvals-are-required/.	**No**
2.	Does your project involve participants who are covered by the Mental Capacity Act? If so, you will need approval from an external ethics committee such as NRES or the Social Care Research Ethics Committee http://www.scie.org.uk/research/ethics-committee/.	**No**
3.	Does your project involve participants who are currently under the auspices of the Criminal Justice System? For example, but not limited to, people on remand, prisoners and those on probation? If so, you will need approval from the ethics approval system of the National Offender Management Service.	**No**

A.2 If your answer to any of the following questions (4 – 11) is YES, you must apply to the City University Senate Research Ethics Committee (SREC) for approval (unless you are applying to an external ethics committee).		*Delete as appropriate*
4.	Does your project involve participants who are unable to give informed consent? For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf?	**No**
5.	Is there a risk that your project might lead to disclosures from participants concerning their involvement in illegal activities?	**No**
6.	Is there a risk that obscene and or illegal material may need to be accessed for your project (including online content and other material)?	**No**
7.	Does your project involve participants disclosing information about sensitive subjects? For example, but not limited to, health status, sexual behaviour, political behaviour, domestic violence.	**No**
8.	Does your project involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning? (See http://www.fco.gov.uk/en/)	**No**
9.	Does your project involve physically invasive or intrusive procedures? For example, these may include, but are not limited to, electrical stimulation, heat, cold or bruising.	**No**
10.	Does your project involve animals?	**No**
11.	Does your project involve the administration of drugs, placebos or other substances to study participants?	**No**

A.3 If your answer to any of the following questions (12 – 18) is YES, you must submit a full application to the Computer Science Research Ethics Committee (CSREC) for approval (unless you are applying to an external ethics committee or the Senate Research Ethics Committee). Your application may be referred to the Senate Research Ethics Committee.		*Delete as appropriate*
12.	Does your project involve participants who are under the age of 18?	**No**
13.	Does your project involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.	**No**
14.	Does your project involve participants who are recruited because they are staff or students of City University London? For example, students studying on a specific course or module. (If yes, approval is also required from the Head of Department or Programme Director.)	**No**
15.	Does your project involve intentional deception of participants?	**No**
16.	Does your project involve participants taking part without their informed consent?	**No**
17.	Does your project pose a risk to participants or other individuals greater than that in normal working life?	**No**
18.	Does your project pose a risk to you, the researcher, greater than that in normal working life?	**No**

A.4 If your answer to the following question (19) is YES and your answer to all questions 1 – 18 is NO, you must complete part B of this form.		
19.	Does your project involve human participants or their identifiable personal data? For example, as interviewees, respondents to a survey or participants in testing.	**No**

Appendices: Any material that would interrupt reading the report should be presented here. The appendices must contain all evidence that allows the markers to assess the extent of your success. Appendix A MUST be a copy of your project proposal in its original format; other material might include: interview records; questionnaires and questionnaire replies; routine design documentation; source code; test data; output listings; displays, etc.; if you produced software for general use, ready-to-install software, installation guide & user guide; data; wireframes and prototypes; annotated computer code If you ran experiments, we would expect the appendices to contain raw data and documentation of the analyses performed on the data; etc. Some kinds of appendix material can or should be submitted in electronic format only: e.g. massive lists of data, audio/video recordings, long source code listings. As you write up your report, your supervisor (referring if necessary to the module leaders) can answer specific doubts about what should be included and formats.

II. Appendix B. Source Data

B.1. Ratings Data

```
+------+--------------------+-------+------+------+------+------+------+------+
|showID|               Title|Episode|A18-49|M18-49|W18-49|A18-34|M18-34|W18-34|
+------+--------------------+-------+------+------+------+------+------+------+
|     1|Code Black Rating...|      1|   1.2|   0.9|   1.5|   0.7|   0.6|   0.8|
|     1|Code Black Rating...|      2|     1|   0.7|   1.4|   0.7|   0.4|   0.9|
|     1|Code Black Rating...|      3|   1.1|   0.8|   1.3|   0.6|   0.5|   0.8|
|     1|Code Black Rating...|      4|     1|   0.8|   1.2|   0.5|   0.5|   0.6|
|     1|Code Black Rating...|      5|   0.7|   0.5|     1|   0.4|   0.3|   0.5|
|     1|Code Black Rating...|      6|   0.9|   0.6|   1.3|   0.5|   0.4|   0.7|
|     1|Code Black Rating...|      7|   0.9|   0.7|   1.2|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|      8|   0.9|   0.7|   1.1|   0.6|   0.5|   0.6|
|     1|Code Black Rating...|      9|     1|   0.8|   1.3|   0.6|   0.5|   0.7|
|     1|Code Black Rating...|     10|   1.1|   0.7|   1.4|   0.6|   0.4|   0.7|
|     1|Code Black Rating...|     11|   0.9|   0.7|   1.1|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|     12|     1|   0.7|   1.2|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|     13|   0.9|   0.7|   1.1|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|     14|   0.9|   0.7|   1.1|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|     15|   1.1|   0.8|   1.3|   0.5|   0.6|   0.5|
|     1|Code Black Rating...|     16|   0.9|   0.7|   1.2|   0.5|   0.4|   0.6|
|     1|Code Black Rating...|   null|  null|  null|  null|  null|  null|  null|
|     1|Code Black Rating...|    AVG|  0.97|  0.72|  1.23|  0.54|  0.44|  0.65|
|     2|Criminal Minds Ra...|      1|   1.9|   1.5|   2.2|   1.1|   0.9|   1.4|
|     2|Criminal Minds Ra...|      2|   1.4|   1.1|   1.8|   0.8|   0.6|     1|
+------+--------------------+-------+------+------+------+------+------+------+
```

B.2. Show Attributes Data

```
  showID                                   Title  \
0      1        Code Black Ratings--Season 2 (2016-17)
1     10     Hell's Kitchen Ratings--Season 16 (2016-17)
2     11        The Exorcist Ratings--Season 1 (2016-17)
3     12      Dateline NBC Ratings--Season 25 (2016-17)
4     13       The Blacklist Ratings--Season 4 (2016-17)
5     14          Chicago Med Ratings--Season 2 (2016-17)


             Title1  Season              Title2  \
0        Code Black       2          code_black
1    Hell's Kitchen      16      hell_s_kitchen
2      The Exorcist       1        the_exorcist
3      Dateline NBC      25        dateline_nbc
4     The Blacklist       4       the_blacklist
5       Chicago Med       2         chicago_med

  Season1                                    url
0      02  https://www.rottentomatoes.com/tv/code_black/s02
1      16  https://www.rottentomatoes.com/tv/hell_s_kitch...
2      01  https://www.rottentomatoes.com/tv/the_exorcist...
3      25  https://www.rottentomatoes.com/tv/dateline_nbc...
4      04  https://www.rottentomatoes.com/tv/the_blacklis...
5      02  https://www.rottentomatoes.com/tv/chicago_med/s02

              Genre            Network  PremiereDate  \
0             Drama                CBS  Sep 28, 2016
1  Special Interest                FOX  Sep 23, 2016
2            Horror                FOX  Sep 23, 2016
3                                  NBC  Sep 18, 2015
4             Drama                NBC  Sep 22, 2016
5             Drama                NBC  Sep 22, 2016

                              execProducers  \
0  [Michael Seitzman, Marti Noxon, Linda Goldstei...
```

```
1  [Kent Weed, Arthur Smith, Gordon Ramsay]
2  [Rolin Jones, David Robinson, Barbara Wall, Ru...
3
4  [John Bokenkamp, John Eisendrath, John Davis, ...
5  [Dick Wolf, Michael Brandt, Derek Haas, Andrew...

                                                Cast
0  [Marcia Gay Harden, Rob Lowe, Boris Kodjoe, Me...
1  [Gordon Ramsay, Andi Van Willigan, Marino Monf...
2  [Geena Davis, Alfonso Herrera, Ben Daniels, Br...
3  [Lester Holt, Keith Morrison, Josh Mankiewicz,...
4  [James Spader, Megan Boone, Ryan Eggold, Diego...
5  [Oliver Platt, S. Epatha Merkerson, Yaya DaCos...

[67 rows x 10 columns]
```

B.3. Review Data

```
   showID reviewID                                  Title  \
0      11     11-0    The Exorcist Ratings--Season 1 (2016-17)
1      11     11-1    The Exorcist Ratings--Season 1 (2016-17)
2      11     11-2    The Exorcist Ratings--Season 1 (2016-17)
3      11     11-3    The Exorcist Ratings--Season 1 (2016-17)
4      11     11-4    The Exorcist Ratings--Season 1 (2016-17)
5      11     11-5    The Exorcist Ratings--Season 1 (2016-17)
6      11     11-6    The Exorcist Ratings--Season 1 (2016-17)
7      11     11-7    The Exorcist Ratings--Season 1 (2016-17)
8      11     11-8    The Exorcist Ratings--Season 1 (2016-17)
9      11     11-9    The Exorcist Ratings--Season 1 (2016-17)
10     11    11-10    The Exorcist Ratings--Season 1 (2016-17)

   Season              reviewer  \
0       1        Carissa Pavlica
1       1        Graeme Blundell
2       1           Hank Stuever
3       1         Rebecca Hawkes
4       1  Katherine McLaughlin
5       1        Chris Nashawaty
6       1            Edie Nugent
7       1           Tim Grierson
8       1         Scott D. Pierce
9       1          Robert Bianco
10      1            Ben Travers

                                       review_url  \
0   https://www.tvfanatic.com/2016/09/the-exorcist...
1   http://www.theaustralian.com.au/arts/review/th...
2   https://www.washingtonpost.com/entertainment/t...
3   http://www.telegraph.co.uk/tv/2016/10/19/the-e...
4   https://www.scifinow.co.uk/reviews/the-exorcis...
5   http://www.ew.com/article/2016/10/03/the-exorc...
6   https://www.commonsensemedia.org/tv-reviews/th...
7   http://www.thewrap.com/the-exorcist-review-dem...
8   http://www.sltrib.com/blogs/tv/4385405-155/tv-...
9   http://www.usatoday.com/story/life/tv/2016/09/...
10  http://www.indiewire.com/2016/09/the-exorcist-...

                                    r_description  \
0   If the quality of the story and the special ef...
1     Much anticipated and ferociously well realised.
2   From the first episode, Slater's version disti...
3   Taking place in the present day, the series is...
4   Screenwriter Jeremy Slater does decent work in...
5   This is a deadly serious drama that wants to a...
6   Geena Davis and Alan Ruck are both solid actor...
7   Judged by its pilot, The Exorcist looks to gro...
8   If you're a horror fan, this series shows some...
```

```
9   Despite the presence of some fine actors, a sc...
10   If it can find a way for this slow-burn openin...

             review_text
0   As many times as I've seen The Exorcist over t...
1   Much anticipated and ferociously well realised...
2
3
4   There’s something sinister and intriguing awai...
5
6   Common Sense Media's unbiased ratings are cond...
7   It’s been 43 years since “The Exorcist” first ...
8   In the premiere, MacGyver and his team try to ...
9   Horrifying.\nOh, not Fox’s The Exorcist (Frida...
10   Fox's TV adaptation of "The Exorcist" got off ...
```

[299 rows x 8 columns]

III. Appendix C. Source Code

C.1 Decision Tree Classifier

```
#Decision Tree Classifier
from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf=clf.fit(X_train_M18_49, y_train_M18_49)
Y_M18_49=clf.predict(X_test_M18_49)

print(Y_M18_49)

print(y_test_M18_49)

from sklearn.metrics import confusion_matrix,mean_squared_error,classification_report
print(confusion_matrix(y_test_M18_49, Y_M18_49))
print("MSE = %.4f" %mean_squared_error(y_test_M18_49,Y_M18_49))

target_names = ['unpopular', 'moderate popular', 'blockbuster']
print(classification_report(y_test_M18_49, Y_M18_49, target_names=target_names))
```

C.2 Neural Network Classifier

```
from sknn.mlp import Layer, Classifier, Regressor
from sklearn import datasets
from sklearn.model_selection import GridSearchCV

nn_M18_49 = Classifier(
   layers=[
            Layer("Rectifier", units=1300),
            Layer("Softmax")],
   learning_rate=0.01,
   regularize="L2",
   weight_decay=0.003,
   n_iter=700)


# nn_A18_49.fit(X, y)



nn_M18_49.fit(X_train_M18_49, y_train_M18_49)

Y_M18_49=nn_M18_49.predict(X_test_M18_49)


print(Y_M18_49)

print(y_test_M18_49)

from sklearn.metrics import confusion_matrix,mean_squared_error
print(confusion_matrix(y_test_M18_49, Y_M18_49))
print("MSE = %.4f" %mean_squared_error(y_test_M18_49,Y_M18_49))
Result:

[[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]]
[1 1 1 1 2 1 1 1 1 1 3]
[[9 0 0]
 [1 0 0]
 [1 0 0]]
MSE = 0.4545
```

C.3 Segmenting ratings to 3 different targets (‘unpopular’, ‘moderate’, ‘blockbuster’)

Plot a histogra to 'segment' ratings to three diffrent targets

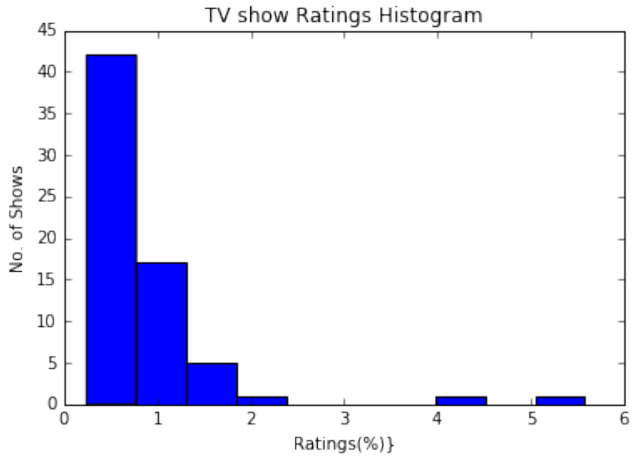
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

np.histogram(y_A18_49,bins=10)

plt.hist(y_A18_34)
plt.title("TV show Ratings Histogram")
plt.xlabel("Ratings(%)}")
plt.ylabel("No. of Shows")

fig = plt.gcf()



C.4 Using Decision Tree model for three different target constructed in C.3

#Decision Tree Classifier for 3 different targets - W18_49

from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf=clf.fit(X_train_W18_49, y_train_W18_49)
Y_W18_49=clf.predict(X_test_W18_49)

print(Y_W18_49)
print(y_test_W18_49)

from sklearn.metrics import confusion_matrix,mean_squared_error,classification_report
print(confusion_matrix(y_test_W18_49, Y_W18_49))
print("MSE = %.4f" %mean_squared_error(y_test_W18_49,Y_W18_49))

target_names = ['unpopular', 'moderate popular', 'blockbuster']
print(classification_report(y_test_W18_49, Y_W18_49, target_names=target_names))

<Classification Result>

```
Predicted(Y) = [2 2 2 2 1 1 2 1 1 1 3]
Actual(y) = [2 1 1 1 3 1 2 1 1 1 3]

Confusion Matrix
[[4 3 0]
 [0 2 0]
 [1 0 1]]

MSE = 0.6364

                  precision    recall  f1-score   support

       unpopular       0.80      0.57      0.67         7
moderate popular       0.40      1.00      0.57         2
     blockbuster       1.00      0.50      0.67         2

     avg / total       0.76      0.64      0.65        11
```

C.5

Neural Network Model predicting three classified target.
Training model usually takes over 5 min.

from sknn.mlp import Layer, Classifier
from sklearn import datasets
from sklearn.model_selection import GridSearchCV

```
nn = Classifier(
    layers=[
            Layer("Rectifier", units=1300),
            Layer("Sigmoid") #output funciton's activation function
          ],
    learning_rate=0.0005,
    regularize="L2",
    weight_decay=0.00003,
    n_iter=1400)

nn.fit(X_train, y_train)

Y=nn.predict(X_test)

# 6 targets are aggregated here for implementing one model

from sklearn.metrics import confusion_matrix,mean_squared_error,classification_report
print(confusion_matrix(y_test[:,0], Y[:,0]))
print(confusion_matrix(y_test[:,1], Y[:,1]))
print(confusion_matrix(y_test[:,2], Y[:,2]))
print(confusion_matrix(y_test[:,3], Y[:,3]))
print(confusion_matrix(y_test[:,4], Y[:,4]))
print(confusion_matrix(y_test[:,5], Y[:,5]))

print("MSE = %.4f" %mean_squared_error(y_test,Y))

target_names = ['unpopular', 'moderate popular', 'blockbuster']
print(classification_report(y_test[:,0], Y[:,0], target_names=target_names))
print(classification_report(y_test[:,1], Y[:,1], target_names=target_names))
print(classification_report(y_test[:,2], Y[:,2], target_names=target_names))
print(classification_report(y_test[:,3], Y[:,3], target_names=target_names))
print(classification_report(y_test[:,4], Y[:,4], target_names=target_names))
print(classification_report(y_test[:,5], Y[:,5], target_names=target_names))
```

IV. Appendix D. Installation Guide

Installation Guidance

1. Upload every Python Notebook file (.ipyne) to jupyter notebook server.
   1-1. Run from the smallest number. From 3.1.2. Gathering Ratings Data. Each number attached in front of the file corresponds to the section on the report. I.e., 3.3-1 means the file is related to the first part of the section 3.3.
2. For the data, it is already uploaded to the cloud server. It is distributed to IBM Object Storage and Amazon S3 storage (word2vec model). In order to use downloaded files, it should be uploaded to HDFS cloud storage and links in the code need to be modified to the directory.

Reminders

3. Working environment was Python 3.5 and Spark 2.1 with notebook on IBM DataScience Experience. I recommend you to use similar environment with Python 3.5 with Spark 2.1.

4. All files are stored in the /Data in this USB as well as copies are pre-uploaded to IBM DataScience Object Storage and Amazon S3. In order to use downloaded data in this USB, relevant codes need to be modified.
   4.1. GoogleNews-vectors-negative300.bin is only saved in the USB stick, and not uploaded to the moodle because of its large size.

5. The file lists are same as below

* IBM DataScience Object Storage
-tvratingsguide.com.txt
-ratings13.csv
-review_0.csv
-review_score_1.csv
-show_attr_2.csv
-show_link_2.csv
-review_vector_final_4.csv
* Amazon S3
-GoogleNews-vectors-negative300.bin

6. Credentials are inserted in the Python Notebook code and will directly open/save files to Object Storage and S3. Credentials are also attached below, as well as modified in the code already.

- IBM DataScience Object Storage

```
credentials = {
  'auth_url':'https://identity.open.softlayer.com',
  'project':'object_storage_2f7e552b_8b03_4878_982a_9b6087f1782d',
```

```
  'project_id':'77fe390c5b07479890c43f40c65c4d39',
  'region':'dallas',
  'user_id':'bfd036a7023247c2b368d099225ca5d4',
  'domain_id':'9f5a5ca449194843a3214cb7ef42aa9a',
  'domain_name':'1303613',
  'username':'member_9f2985250d92a7f3335e64f6dbf3ceba54aff88d',
  'password':"""jl}ImF]uu#0azB{4""",
  'container':'Project',
  'tenantId':'undefined',
}
```

- Amazon S3 (Google word2vec model file)
  https://s3.eu-west-2.amazonaws.com/word2vec-jaemin-dissertation-cityu/GoogleNews-vectors-negative300.bin

7. It might be required installing some modules including NLTK, boto (S3), urllib, beautifulsoup4, scikit-neuralnetwork and other libraries. If you got any of the errors because of these libraries, then simply run '!pip install <toolkit name>' .i.e., '!pip install scikit-neuralnetwork'
8. For Diffbot use in 'CollecReview.ipynb', expiration date of the token '736048adbb132aa735c5cf6f303315b0' is 10/10/2017. It need to be renewed after that 10/10/2017 by exchanging the token. Please be contacted if you need to swap it.