Classification of first weekend revenue of movies from its pre-release reviews

Author: Bibek Mishra

Abstract

The paper discusses about building model to classify the first weekend revenues of movies from critics' pre-screen reviews. This is considerably tough because our approach does not consider user reviews and metadata parameters to predict the revenue. Our research found out that SMO algorithm with default settings performed best for our problem at hand. I used error analysis and tuning procedures to evaluate different avenues to improve the performance. My final model's performance improved significantly after the error analysis and beat the baseline by a big margin.

Keywords

Support Vector Machine, Sequential Minimal Optimization, Feature selection, Linear, pre-release, movie review

Predicting Movie revenue from Text Reviews

Production of a movie requires a lot of money and considerable amount of time. Hence, the production houses and movie makers are eager to invest their money and time on projects that could fetch them high returns. Consequently, there are many folks in the movie industry who like to know the market value or predicted revenue of a movie before it is released.

The opening weekend revenue of a movie is good predictor of total revenue of a movie. In their paper, Simonnof and Sparrow found that opening revenue can account for almost 25% of the total movie revenue. Strength of the opening weekend of a release influences all major decisions pertaining to a film's ultimate financial destiny [4]. These decisions include number of weeks the movie is played at box office and number of screens it is released in international arena. Considering the importance of first weekend revenue, many researchers have spent considerable amount of time to find predictive models to forecast the same.

Prior Work

The prior work in this field tried to predict the movie revenue both from the metadata features and movie reviews. The metadata features that were mostly studied are genre, MPAA rating, star cast, number of screens, language, release timeline, awards etc. Both reviews from users and critics were studied in various research papers to predict the movie revenue.

Mestyan et.al tried to predict the box office success of movie based on Wikipedia activities [2]. They considered number of contributors for Wikipedia page, number of edits, number of page views and collaborative rigor to gauge popularity of 312 movies released in 2010[3].

Sharda and Delen devised a two hidden layer neural network to correctly classify movies in 9 categories according to their anticipated revenue[3]. Their method took metadata features as input to classify the movies in one of 9 categories (flop to blockbuster) according to predicted revenue. Their solution based on multi-layer perceptron and logistics regression could predict the revenue with 37% in the category and with 75% in +/- 1 category.

Joshi et. al tried to predict the opening weekend revenue from its metadata and reviews[1]. They used a linear regression using text (i.e. review) and non-text (i.e. metadata) features to predict gross revenue on opening weekend and averaged opening revenue per screen. Their study shows that a combination of metadata and text features achieve the best performance in terms of Mean Absolute Error (Mean Absolute Error) and Pearson's correlation Coefficient (r). They also devised a model from text-only features that got comparable results when compared to models built on metadata and text features.

Simonoff and Sparrows considered metadata features to predict the first weekend revenue of a movie[4]. The model used linear regression and could achieve R2 = 96.6% for movies release on more than 10 screens. As the revenues considered in model were right-tailed, their log values were used in research.

Dataset

I used the pre-screen movie reviews from the research work of Joshi et. al. The dataset consists of prescreen reviews of 1717 movies released in between 2005 to 2009, both years included. The dataset consists of two attributes – (i) review and (ii) revenue. The review attribute is the appended review for the film from different trusted sources (For example: Austin Chronicle, the Boston Globe, the LA Times, the Entertainment Weekly, the New York Times, Variety and the Village Voice. The class attribute is the opening weekend revenue for the movie.

Since the revenue values are skewed towards the higher side, I decided to consider the log of revenue in my dataset. Then, I categorized the revenue in 5 categories according to decreasing order of it. The categories were VERY HIGH (Blockbuster), HIGH, MEDIUM, LOW, VERY LOW (Flop) (Table 2).

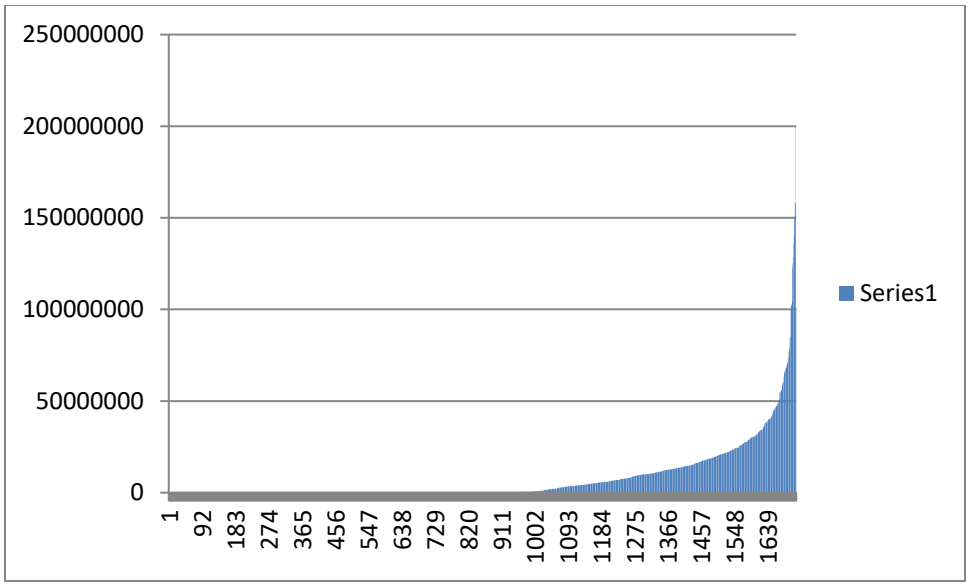

Figure 1: Distribution of Opening Weekend Revenue. The chart clearly shows that the dataset is skewed towards right.

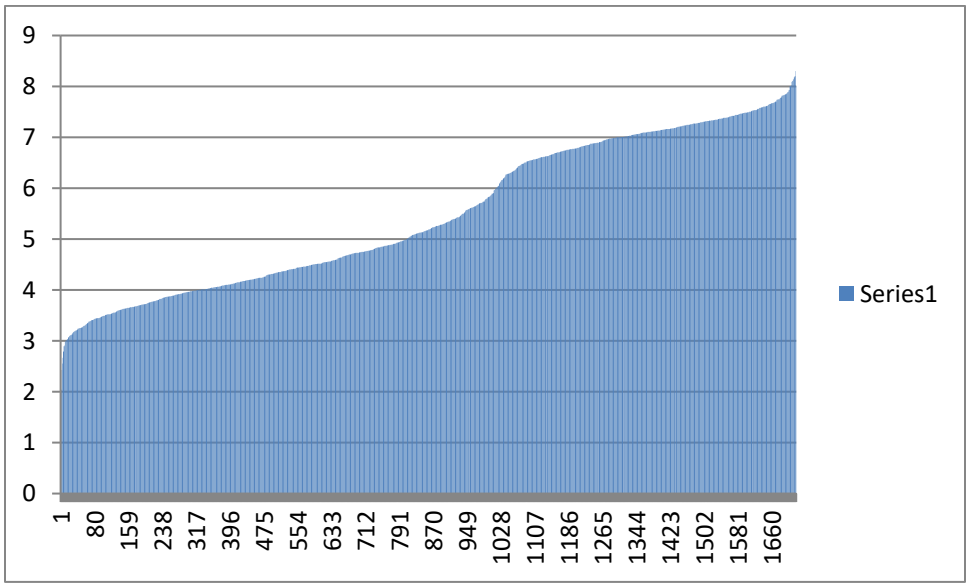

Figure 2: Distribution of Log (Revenue). The log(revenue) is more uniformly distributed across the dataset.

The entire dataset was divided into training, development and test sets. The training set contained reviews of movies released between 2005 to 2007. The development set and test set had reviews of movies released in 2008 and 2009 respectively (Table 1).

Dataset	Release Year	Number of Instances
Training Set	2005 to 2007	1146
Development Set	2008	317
Test Set	2009	254

Table 1: Distribution of all instances in training, development and test dataset.

Class	Very Low (Flop)	Low	Medium	High	Very High (Blockbuster)
Log (Revenue)	1.97 - 3.24	3.24 – 4.51	4.51 – 5.77	5.77 – 7.04	7.04 – 8.31
Revenue (In Dollars)	94.9-1747	1747-32140	32140 - 591163	591163 - 10873484	10873484 – 2E+08

Table 2: Classification of instances according to first weekend revenue.

My Approach

The research by Joshi et.al points out that a predictive model derived only from text reviews can achieve performance compared with a model built from text and metadata features. In my approach, I plan to predict first weekend revenue of a movie only from its pre-screen revenues. As first weekend revenue is representative of the entire revenue of a movie, usually we can have information about the full revenue of movie from it. Nonetheless, It is challenging because we are not using any metadata features or after-release cues in building our model. However, I believe that the critics' pre-release reviews contain enough information and features to predict first weekend revenue of a movie with accuracy.

Initially, I used the training dataset to train the model and validate its performance through cross-validation. Then, I used the development set as test set with the above mentioned model in order to do error analysis. Suggested improvements were implemented to obtain the intermediate model. Finally, I attempted to tune the model by considering different parameters of AttributeSelectedClassifier algorithm with SMO classifier. I tuned the following parameters: (1) number of features and (2) exponent of SMO classifier. In conclusion, I mentioned my results, compared model performance with the baseline performance and discuss on it.

Baseline

First, I extracted only the unigram features from training dataset and built a baseline model from it. I used a 10 fold cross validation using the SVM Liblinear classifier.

The baseline performance is **0.08** and accuracy is **33%.**

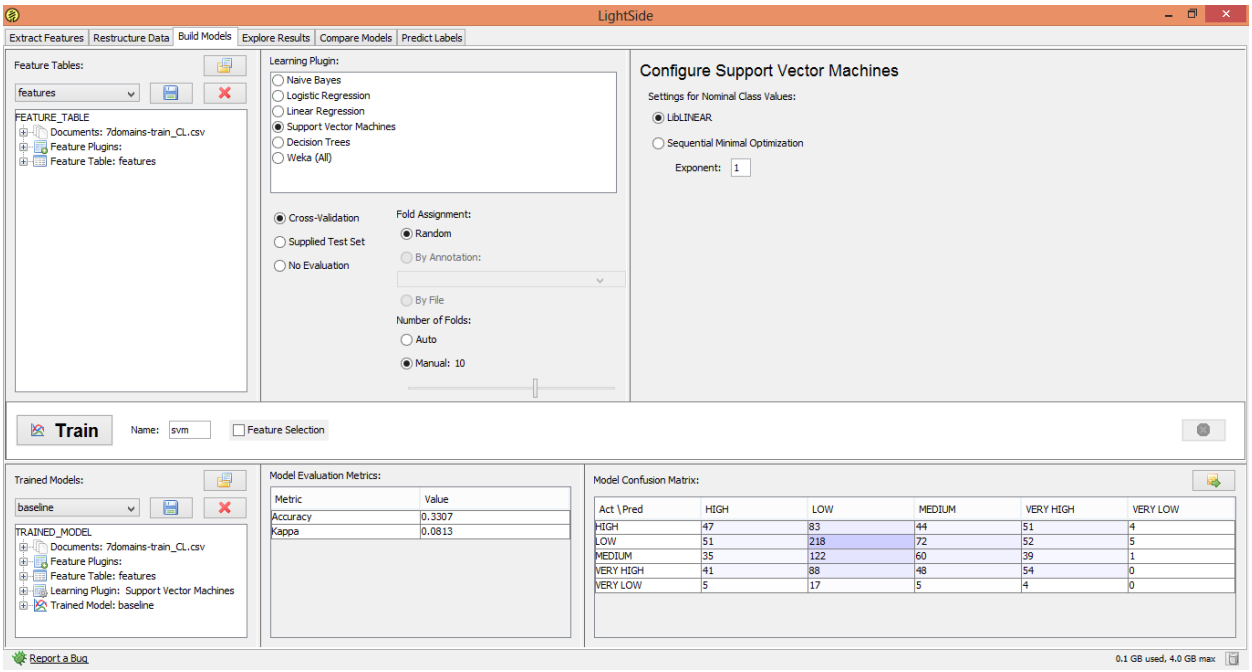

Figure 3. Building of baseline model using SVM Liblinear classifier.

Error Analysis

Then, I validate the development dataset using the baseline model to start my error analysis.

In Explore Results panel, I selected the recently created model. Looking at the confusion matrix in top/center of screen, I decided to analyze the cell labeled Actual Medium and Predicted Low. This had 43 errors.

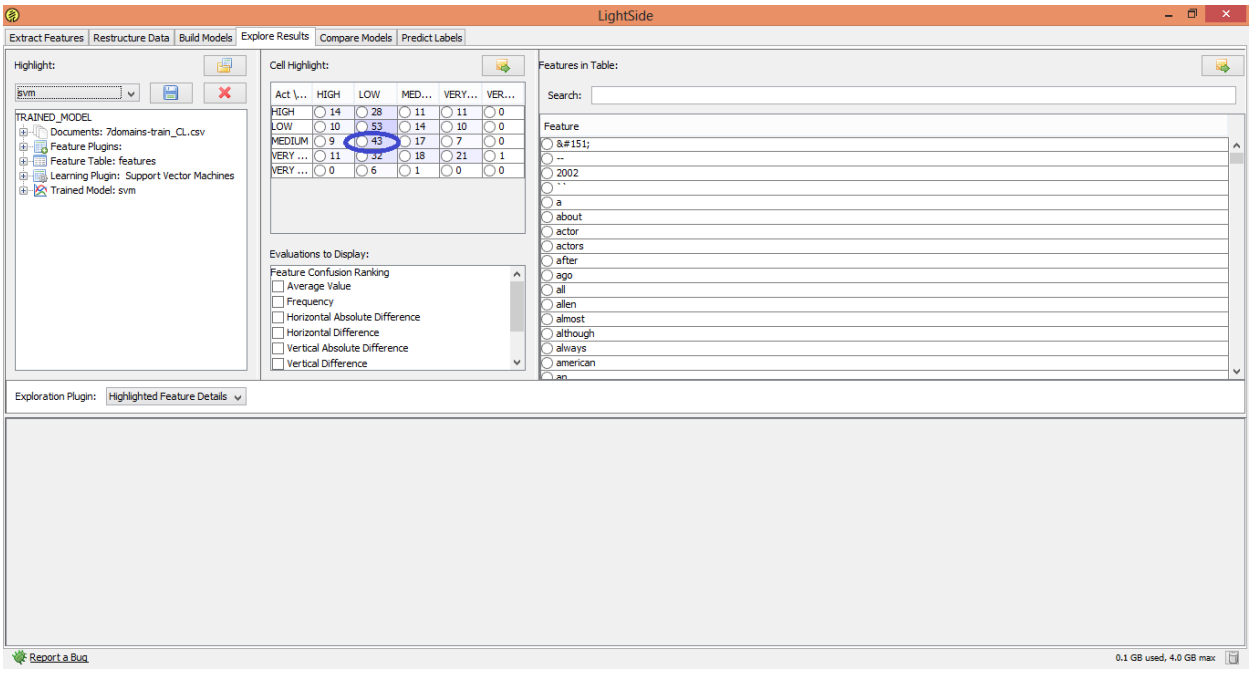

Figure 4: Error Analysis with Development dataset as supplied test-set.

In order to do a horizontal comparison, I clicked on Frequency, Horizontal Absolute Difference and Feature Weight. The entire range of feature weight was: [-1.4 to 1.4]. Then, I sorted the table by decreasing 'Horizontal Absolute Difference'.

Horizontal Comparison

Searching for a feature having high frequency, high horizontal difference and high weight, I decided to look at 'in' feature.

I found out that the word 'In' mostly appears before adjective, noun or article (a or an). In order to categorize words correctly with 'In', I plan to extract the Part Of Speech (POS) features from dataset. Moreover, there are many phrases in the reviews that can't be accounted for in unigrams. The meanings of these phrases will be cleared if we extract bigrams. Finally, the punctuations have rich meaning in most movie reviews. For example, comma followed by a but usually denotes a change in course of sentence.

I also found many features that had negligible absolute feature weight (Example: than, between, great, cast). I would like to try feature selection to check whether removing features from dataset will help.

Vertical Comparison

Then I did a vertical comparison, I clicked on Frequency, Vertical Absolute Difference and Feature Weight. The entire range of feature weight was: [-1.4 to 1.4]. Then, I sorted the table by increasing 'Vertical Absolute Difference'.

I found out some features (i.e movies, film) which mean the same thing but perceived differently. They had less vertical absolute difference, high feature weight and reasonable frequency.

Feature	Frequency	Vertical Absolute Difference	Feature Weight
film	2	0.094339623	0.060132276
movies	2	0.046511628	0.299197204

Table 3: Vertical Comparison

I decide to combine the features through a regular expression.

Next Step: Building model with New features

Next, I built the model on training data by extracting following features

i. Unigrams
ii. Bigrams
iii. POS Bigrams
iv. Punctuations
v. Build regular expression from movies and film

The extracted feature table had 92K features. Then, I used the AttributeEvaluationClassifier with SMO algorithm to find intermediate performance.

Accuracy of Model: 0.65
Kappa Statistics: 0.52

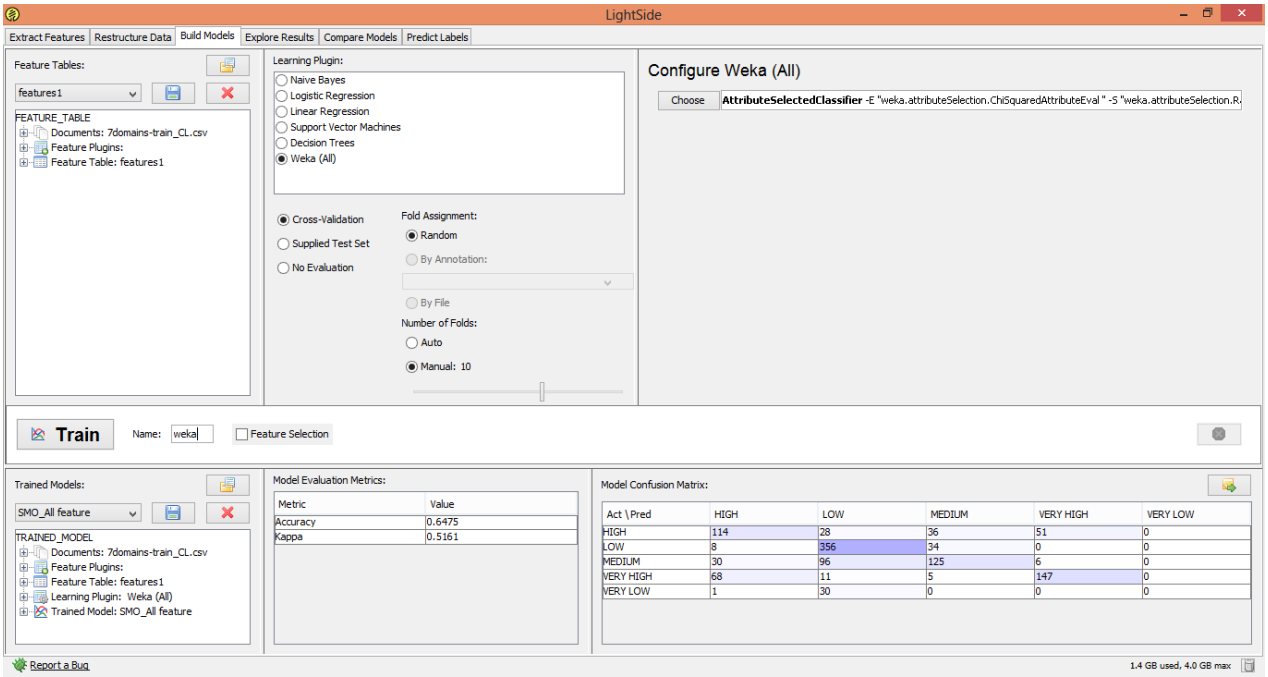

Figure 5: Building the intermediate model with AttributeSelectedClassifier and SMO classifier

Tuning

Then, I performed tuning. First, I create 5 sets of train/test pair by StratifiedRemoveFolds option from intermediate feature table (Figure 6). I choose Stratified option because it preserves the distribution of class values across folds and randomizes well.

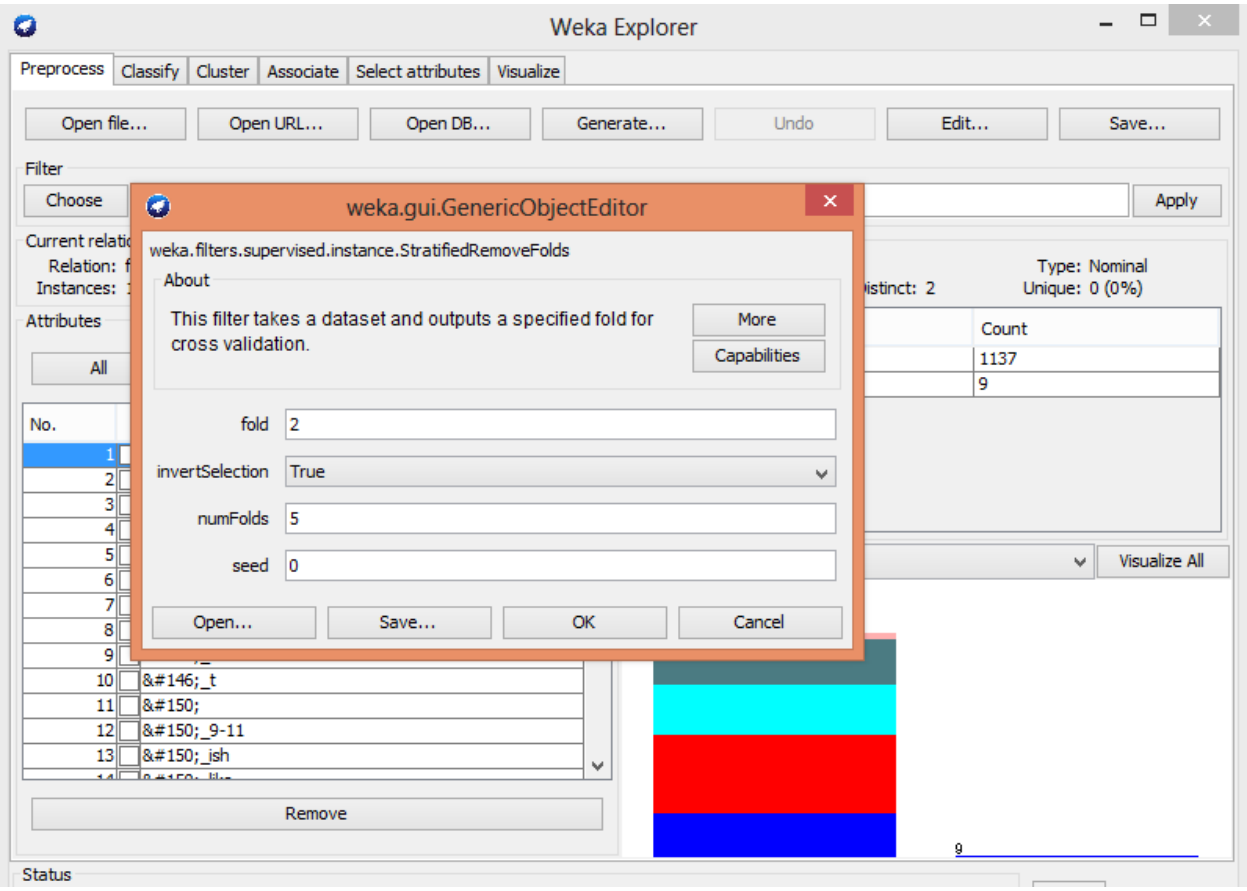

Figure 6: Creating train and test pairs from feature table.

Stage 1

I completed the stage 1 of Tuning in Experimenter. I found out the Kappa performance for all training sets and whole dataset.

I used the AttributeSelectedClassifier with SMO classifier because I wanted to evaluate effect of feature selection on tuning. Moreover, SMO is a simple but robust algorithm that works well on text datasets and prevents the model from over fitting.

I tried following features in tuning. The kappa values obtained were filled in tuning table (table 3).

Algorithm: AttributeClassifierSelector with SMO classifier

- Exponent of SMO Classifier: 1.0, Features Selected: ALL
- Exponent of SMO Classifier: 2.0, Features Selected: ALL
- Exponent of SMO Classifier: 1.0, Features Selected: Top 45K
- Exponent of SMO Classifier: 2.0, Features Selected: Top 45K

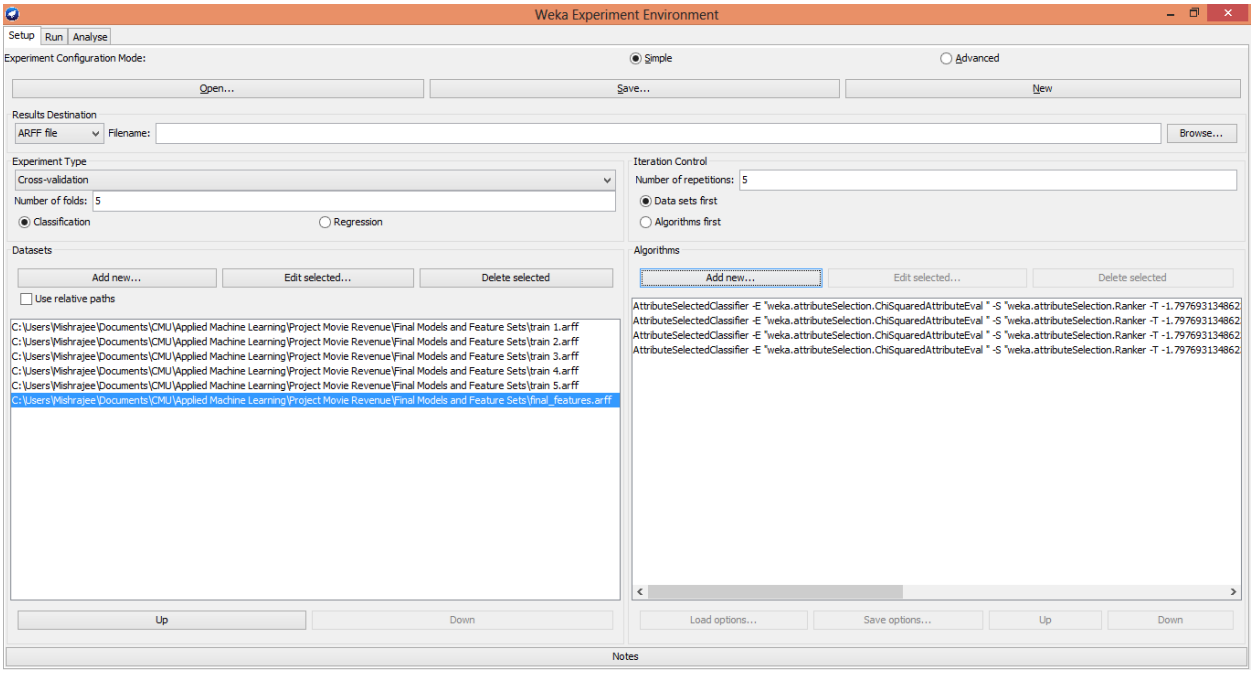

Figure 7: Screenshot of Experimenter. Five folds of training sets, whole dataset and four variations of algorithm selected.

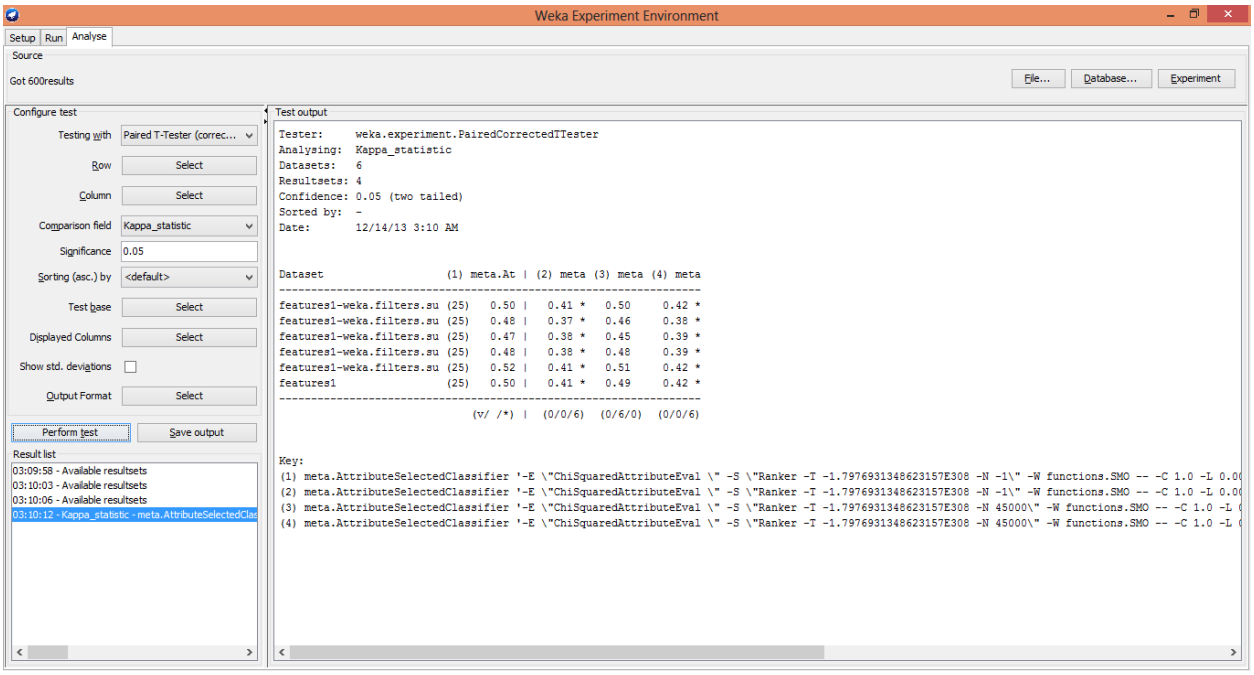

Figure 8: Result of tuning

Fold	**Setting 1** Exponent: 1.0 Features: All	**Setting 2** Exponent: 1.0 Features: All	**Setting 3** Exponent: 2.0 Features: Top 45K	**Setting 4** Exponent: 2.0 Features: Top 45K	**Optimal Setting**	**Test set Performance**
1	0.5	0.41	0.50	0.42	Exponent: 1.0, Features: ALL	0.34
2	0.48	0.37	0.46	0.38	Exponent: 1.0, Features: ALL	0.43
3	0.47	0.38	0.45	0.39	Exponent: 1.0, Features: ALL	0.41
4	0.48	0.38	0.48	0.39	Exponent: 1.0, Features: ALL	0.44
5	0.52	0.41	0.51	0.42	Exponent: 1.0, Features: ALL	0.43
Average for Setting	0.49	0.39	0.48	0.4		

Table 4: Tuning table

Average across test set performance: 0.41

Stage 2

In Stage 2, We build the optimized model over the whole dataset using the best setting from Stage 1. It's evident that the algorithm offered best performance using the default setting (Exponent: 1.0, Features: ALL). Hence, model will be built using the default setting.

Kappa Statistics for Optimized model = 0.52

Stage 3

Then, we tried to find the best setting observed on each fold and computed the test set performance by training a model using the optimal setting over the training data on the fold and then applying that model to the test data for that fold. All Kappa statistics values from each fold were mentioned in the test-set performance column in above tuning table.

As a result of tuning, the performance of model over unseen data is : 0.41

In our case, the default settings of AttributeSelectedClassifier algorithm with SMO classifier performed the best. The performance of tuned model over unseen data was considerably lower than the model built over whole data in stage 2. There was no need to perform the ttest here since the performance did not improve through the tuning process.

Results & Discussion

From the above research, it is evident that error analysis helped us uncover many features that helped us to improve the performance of model. The model built after error analysis (Kappa:0.52) showed a great performance improvement over the baseline (Kappa=0.08) (Figure 9).

However, tuning did not help much in improving the performance. The model built over the default settings already provided the best possible performance.

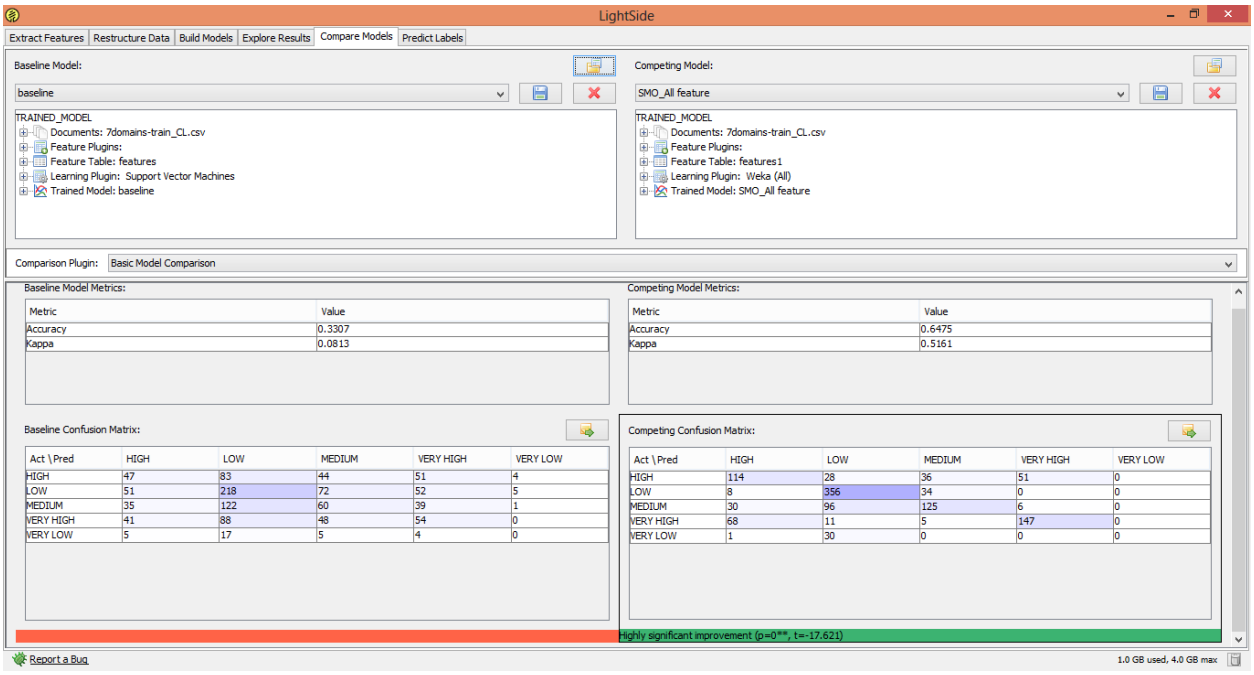

Figure 9: Final model showed significant improvement over the baseline performance.

Conclusion

It is possible to predict the first weekend revenue from the movie reviews with great accuracy (Kappa: 0.52). Considering both feature selection and exponents of algorithms, I found out that the linear version of SMO with all features provided the best possible performance. It would be an interesting next step to put effort behind predicting the entire movie revenue from its pre-screen reviews.

References

[1] Mahesh Joshi et.al, “Movie Reviews and Revenues: An Experiment in Text Regression”, HLT ’10 Human Language Technologies: The 2010 Annual Conference for Computational Linguistics, Pages 293-296

[2] Mestyan M, Yasseri T, Kertesz J, “Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data”, PLoS ONE 8(8): e71226. doi: 10.1371/journal.pone.0071226

[3] Sharda R, Delen D, “Predicting box office success of motion pictures with Neural Networks”, Expert Systems with Applications 30 (2006), Page: 243-254

[4] Simonoff J S, Sparrow I R, “Predicting movie grosses: Winners and losers, blockbusters and sleepers,” Chance, 13(3), 15-24 (Summer 2000)