**Machine Learning Project**

**Dexin Wang**

**Guangyi Tao**

1. **Abstract:**

The purpose of this project is to apply six preprocessing strategies and six algorithms to analyze the movies data collected from IMDB and Facebook, and published on Kaggle.com. The primary goal is to predict the rating for each movie from both nominal data and numerical data. As we completed the procedures and got the empirical results, we have learned how to apply different preprocessing strategies and algorithms and provide solutions for similar problems in the future to improve accuracy in several reasonable ways. We also found that, based on this project, the most efficient way to improve accuracy is Standard Score Normalization.

1. **Keywords:**

Multiclass Classification, Standard score normalization, Bag of words, Linear SVM, Stochastic Gradient Descent, Kernel Approximation, Nearest Centroid Classifier, Gaussian Naïve Bayes, Forests of randomized trees.

1. **Introduction:**

Being able to predict rating will be beneficial for film viewers and film studios. On one hand, choosing the most wonderful movies to watch in limited leisure time is the optimal goal for viewers. Failing to find a good movie or sitting through a bad movie that was expected to be good is frustrating for most viewers. Each year, tens of thousands of new movies are releases around the world. This massive amount of options also make the choice difficult. Predicting rating information will enable viewers to watch wonderful movies the first time without waiting on others’ comments and ratings on the movie and make the choosing process so much easier. On the other hand, all film studios want to create popular films from rampant movies to gain reputation and make profit. The general process is that film studios will invest money and resources to make films first. Then, the films will be released, and watched by viewers. Finally, the projection of the number of tickets sold could indicate whether the films are successful or not. The ratings are often an important indicator of the box office records. However, most ratings won’t be available until the movie is released and watched. As a result, the ratings won’t be much of a help to the success of the movie before its release. The model that we proposed, however, would enable the predicting power of the ratings to the movie even at the development and production stage. Hence, the accurately predicted rating may help producers avoid failures.

In this paper, there are two phases to use machine learning in IMDB dataset: Prediction and Preprocessing. In prediction phase, this paper will consider six algorithms: Linear SVM, Stochastic Gradient Descent, Kernel Approximation, Nearest Centroid Classifier, Gaussian Naive Bayes, Forests of randomized trees. In preprocessing phase, all algorithms will be used to evaluate the six preprocessing strategies. In this paper, these algorithms and preprocessing strategies are used to improve the predicted accuracy.

1. **Data:**
   1. **Raw Data**

The dataset was published on <https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset>. It only contains one table which contains 28 attributes and 5043 tuples.

* 1. **Table**

|  |  |  |
| --- | --- | --- |
| Features | Type | Content |
| Color | String | It only contains two values: ‘Color’ and ‘Black and White’. |
| Director\_name | String | The name of director in the movie. There are 2398 Directors. |
| Num\_critic\_for\_reviews | Numeric | The total number of characters of critic. |
| Duration | Numeric | The number of minutes for each move. |
| Director\_facebook\_like | Numeric | The total number of likes of the director on Facebook. |
| Actor\_3\_facebook\_like | Numeric | The total number of likes of the thirdly director on Facebook |
| Actor\_2\_name | String | The name of the secondary actor in the movie. There are 3032 actor names. |
| Actor\_1\_facebook\_like | Numeric | The total number of likes of the primary actor on Facebook. |
| Gross | Numeric | The total gross is from box office. |
| Genres | String | The types of the movies. (Multi-Valued). |
| Actor\_1\_name | String | The name of the primary actor in the movie. |
| Movie\_title | String | Movie titles as primary key |
| Num\_voted\_users | Numeric | The total number of voted users. |
| Cast\_total\_facebook\_likes | Numeric | The total number of likes of all actors and director on Facebook. |
| Actor\_3\_name | String | The name of the thirdly actor in the movie. |
| Facenumber\_in\_poster | Numeric | The total number of faces on the poster. |
| Plot\_keywords | String | The keys words in the movie. (Multi-Valued) |
| Movie\_imdb\_link | String | The link to connect the movie on IMDB |
| Num\_user\_for\_reviews | Numeric | The total number of reviews. |
| Language | String | The primary language in the movie. (47 languages) |
| Country | String | The country which releases the movie. (65 countries) |
| Content\_rating | String | The rating of the content. (18 ratings) |
| Budget | Numeric | The total number of budget. |
| Title\_year | Numeric | The year when the movie was released. (91 years) |
| Actor\_2\_facebook\_like | Numeric | The total number of likes of the Secondary actor on Facebook. |
| IMDB\_score | Numeric | Predicted feature (from 0 to 10) |
| Aspect\_ratio | Numeric | The ratio on screen. (22 types) |
| Movie\_facebook\_likes | Numeric | The total number of likes of the movie on Facebook. |

* 1. **Define Problem.**

According to Daumé (2017), “Regression: trying to predict a real value” (p.10). The regression is to predict the value of a continues valued function. Even though the range of IMDB score is from 0 to 10, it could not be defined as regression problem because IMDB rating only has the first decimal place. In order words, it is not continuing. Hence, this rating problem could be regarded as a multiclass classification with 100 label values. However, most film studios and viewers will focus on the digital number before the decimal point. Therefore, this multiclass classification could be converted as a multiclass classification with 10 label values. The conversion formula is:

]

After the transition, the label value is one from the set [2, 3, 4, 5, 6, 7, 8, 9, 10], because the original minimum label value is greater than 1.5, and the original maximum label value is smaller than 10.

According to the label distribution, we found that 1807 movies’ ratings are 7. In other words, almost 37% of movies’ ratings are 7. By considering the prediction with the most popular score,7, the accuracy will be 37%. Hence, our machine learning project should have at least 37% accuracy to prove that these preprocessing is useful.

Figure 1.

* 1. **Accuracy on raw data**

The comparison between the predicted accuracy on raw data and final accuracy will strongly indicate how the two phases work on the dataset. Hence, raw dataset should be trained and tested by the six algorithms with four measurements: accuracy, training time, testing time, and mean error. In this project, the testing time is not significant because the size of dataset is so small that could be neglected.

* The dataset is separated as training set (80%) and testing set (20%).
* All algorithms except Gaussian Naïve Bayes will only accept numerical type values. Then, the accuracy on raw data depends on data with numerical type values.
* The missing values are replaced by 0.
* Mean error is calculated by formula as:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 0.03% | 16s | 2800.5 |
| Stochastic Gradient Descent | 12.7%(Mean) (+/-10%) | 1.1s | 3553.1 |
| Kernel Approximation | 18.7%(Mean) (+/-5%) | 3.1s | 5402.1 |
| Nearest Centroid Classifier | 0% | 0.04s | 4113.7 |
| Gaussian Naive Bayes | 5.3% | 0.07s | 2845.3 |
| Forests of randomized trees | 29.6%(Mean) (+/-2%) | 0.07s | 2029.1 |

Some algorithms are designed by using random features or random parameters. Then, they will get the different accuracy each time. In the rest sections, these algorithms as unstable algorithms otherwise called stable algorithms. By conclusion of the table above, the best accuracy is predicted by the unstable algorithm, Forests of randomized trees, with almost 30% with numerical type value of raw data, but it could be improved.

1. **Preprocessing**

The raw dataset is the main reason to cause lower accuracy because of five reasons: duplicated tuples, feature scales, nominal features, unnecessary features, and excessive features. The goal of preprocessing is to clean and convert the dataset as acceptance dataset which could cause higher predicated accuracy. For each step, fixed dataset will apply the algorithms to show how each step of preprocessing affects the accuracy in reasonable ways.

* 1. **Remove Duplicated Data**

Duplicated records in dataset may cause lower accuracy or higher accuracy, but both of influence will cause less understood and application dependent (Kołcz, 2003). One the hand, when both duplicated data sets are in testing set, it will cause overfitting for some algorithm because it affects data distribution. For example, perceptron algorithm will alter the weights and the biases when the predicted label is different from the real label in iterations. Hence, the duplicated data may update the weights and the bias twice. In order words, it will affect the speed of convergence to get accurate weights and the bias. The overfitting may happen when the number of iteration is not enough for linear separable dataset. On the other hand, duplicated data belong to training set and testing set may increase the accuracy. For instance, the model of decision tree will create subtree for all training data without considering pruning and depth limit. When the model meet the duplicated data, it will predict the exact label. However, the goal of the model is to predict new data that the model has never seen before. As a result, duplicated data should be removed even the accuracy becomes worse.

For IMDB dataset, it contains 126 duplicated data. After removing the duplicated data, two unstable algorithms had higher accuracy, and one unstable algorithm had lower accuracy. Also, two stable accuracy did not change accuracy, and one stable algorithm had higher accuracy with less mean error. Based on the changes on accuracy and mean error, the table could show that removing duplicated tuples could make prediction a little better.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 0.1% | 16s | 2656 |
| Stochastic Gradient Descent | 14.8% | 1.3s | 8029.7 |
| Kernel Approximation | 24.9% | 3.5s | 7806 |
| Nearest Centroid Classifier | 0% | 0.05s | 3583.5 |
| Gaussian Naive Bayes | 5.3% | 0.08s | 2977.5 |
| Forests of randomized trees | 28.3% | 1.2s | 2251.4 |

* 1. **Standard Score Normalization**

The motivation of Normalization is to make sure that algorithms work well and fast. Some algorithms may not work well with large values such as Linear SVM and KNN. Also, the normalized data will work fast such as evidence from the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 33.7% | 1.5s | 0.9 |
| Stochastic Gradient Descent | 29.3% | 0.02s | 1.2 |
| Kernel Approximation | 24.0% | 0.06s | 1.1 |
| Nearest Centroid Classifier | 8.2% | 0.007s | 2.8 |
| Gaussian Naive Bayes | 5.9% | 0.008s | 3.2 |
| Forests of randomized trees | 36.1% | 0.12s | 1.0 |

By comparison, all accuracy predicated by algorithms were increased especially the Linear SVM and Kernel Centroid Classifier. Also, the training times and mean error were extremely decreased.

* 1. **Bag-of-Words**

Until now, all algorithms just predict numerical values from dataset. It means these algorithms just use half of the data to predict labels, but another half nominal data is not nosey, and it can be converted as numerical features, which can be calculated by algorithms. In Wikipedia website, “The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier” (“Bag-of-words model” n.d.). After creating features, the dataset contains 29,142 features. Even though all accuracy is increased, and all mean error is decreased, but training time are increased extremely either. In order words, more features increase time cost.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 36.5% | 10.5s | 0.9 |
| Stochastic Gradient Descent | 33.2% | 17.4s | 0.9 |
| Kernel Approximation | 23.6% | 0.05s | 1.4 |
| Nearest Centroid Classifier | 11% | 10.7s | 2.5 |
| Gaussian Naive Bayes | 25.5% | 14.8s | 1.4 |
| Forests of randomized trees | 36.5% | 14s | 0.9 |

* 1. **Remove Unnecessary Features**

To evaluate machine learning algorithms, time cost also is an important parameter. Right now, the massive features cause longer time to process. In order to reduce the time cost, there are several ways to reduce the number of features which is created by “bag-of-words”. First, the primary key features could be deleted because they contain unit value for each tuple which is not useful to predict values. For IMDB dataset, “movie\_title” and “Move\_IMDB\_link” are primary key features. Without considering these features, the new dataset only has 19,306 features. The accuracy and mean error are neglected effect, and all time costs reduced more 30 percent.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 36.5% | 7.1s | 0.9 |
| Stochastic Gradient Descent | 34.8% | 14.9s | 1.0 |
| Kernel Approximation | 36.8% | 0.05s | 1.3 |
| Nearest Centroid Classifier | 11% | 7.0s | 2.5 |
| Gaussian Naive Bayes | 25.5% | 8.2s | 1.4 |
| Forests of randomized trees | 35.4% | 9.4s | 0.9 |

Second, duplicated features could be deletes even though they are in different types. For the current dataset, it contains the number of likes of each actor or each director on Facebook. These numbers have more meaningful values than names. Hence, we assume that these name features are duplicated and could be deleted. To prove this assumption is correct, we apply algorithms with the dataset which has no features as “director\_name”, “actor\_1\_name”, “actor\_2\_name”, and “actor\_3\_name”.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 36.5% | 3.9s | 0.9 |
| Stochastic Gradient Descent | 38.4% | 5.07s | 1.0 |
| Kernel Approximation | 32.1% | 0.05s | 1.4 |
| Nearest Centroid Classifier | 11% | 3.1s | 2.5 |
| Gaussian Naive Bayes | 25.5% | 3.4s | 1.4 |
| Forests of randomized trees | 38% | 4.1s | 0.9 |

The new dataset has 8258 features. Most accuracy and mean error have no changes, and the time cost saves more than 50%. In sum, the assumption of duplicated features is correct and work well for this dataset.

* 1. **Checking Overfitting**

The above table shows the accuracies from different algorithms based on 8,258 features. However, Young (2007) mentions that bag-of-words may create too many features, and finally cause overfitting. Then, we will find the minimum number of features that would lead to the highest accuracy. The solution is a more complicated.

The first step is to define a number of iterations, K. For each nominal attribute, the solution will consider the most K popular values to create Kth features. If the total number of values, M, is less than K, the soliton will only create M features.

Next, all algorithm could evaluate the accuracy with K and cross-validation function. For this project, all cross-validation function will divide the training data into 5 chunks and get the mean accuracy with 5 chunks as a measurement to avoid overfitting problem. Also, the accuracy with testing set could be a measurement.

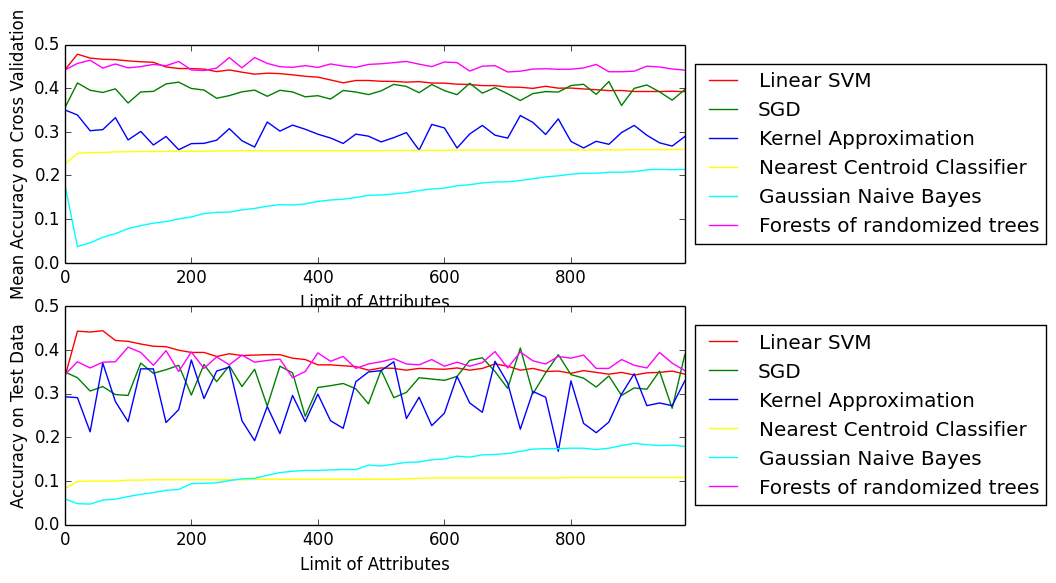
However, each iteration will cause at least one minute. K becomes bigger, the running times is longer for each iteration. Hence, we tried to use divide-and-conquer function to solve this problem with 8 partitions that each contain 1000s. 

Figure 2.

Fortunately, we could decide to focus on the value of K from 0 to 70 based on the conditions from Figure 2.

1. Nearest Centroid Classifier always has the same accuracy which is around 20%
2. The accuracy of Gaussian Naïve Bayes is increasing with increasing K
3. The highest accuracy of Gaussian Naïve Bayes is never higher than Linear SVM and Forests of randomized trees even though there is no feature deletion.
4. SGD and Kernel Approximation are unstable algorithms.
5. Most time, the accuracy of SGD and Kernel Approximation are lower than Linear SVM and Forests of randomized trees.
6. The accuracy of Forests of randomized trees is neglected effect.
7. The Forests of randomized trees is unstable algorithms.
8. Linear SVM has lower values with increasing K, and it is eventually lower than Forest of randomized trees.
9. The highest accuracy with small when K is 20 for mean accuracy graph and 60 for testing accuracy.

According to (1), (2), (3), (4), (5), we could ignore Kernel Approximation, Nearest Centroid Classifier, Gaussian Naïve Bayes, and Forests of Randomized Trees because they do not work well to find highest accuracy. The remining conditions indicate that Linear SVM and Forests of Randomized trees have higher probability to get higher accuracy with minimum K. Hence, we can just consider range of K from 0 to 70 without gaps.

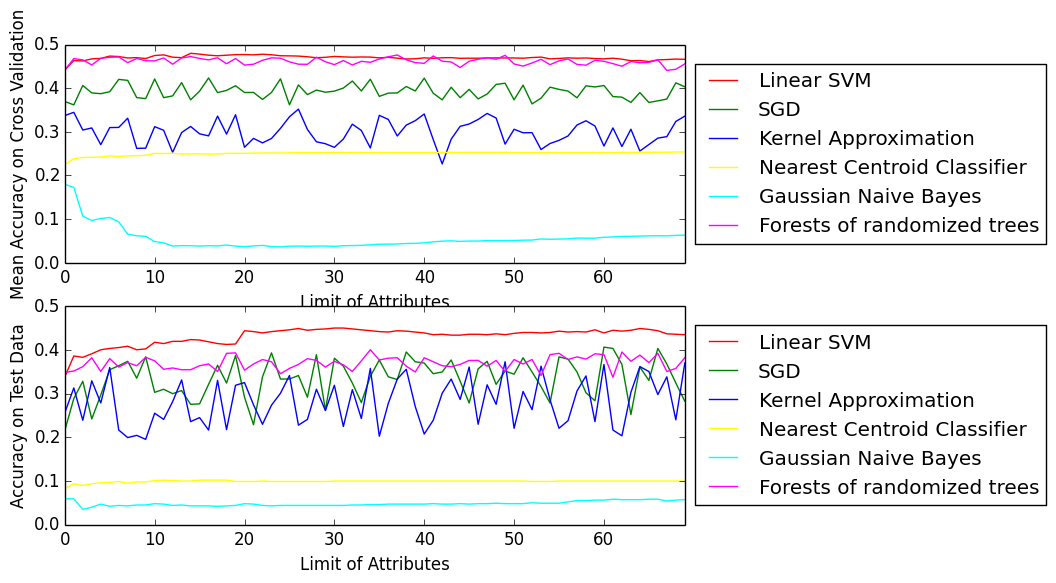


Figure 3.

The Figure 3 indicates that Linear SVM will have the highest accuracy when k is 14 for mean accuracy graph, and when k is 60 for testing accuracy graph. For the most machine learning problem, algorithms will not peek test when they are creating models. Hence, we just try to focus on mean accuracy. When K is 14, we get the result as below table.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 42.1% | 1.4s | 0.8 |
| Stochastic Gradient Descent | 26.5% | 0.04s | 0.9 |
| Kernel Approximation | 27.4% | 0.05s | 1.2 |
| Nearest Centroid Classifier | 10.2% | 0.02s | 2.7 |
| Gaussian Naive Bayes | 4.3% | 0.02s | 3.5 |
| Forests of randomized trees | 36.9% | 0.11s | 0.9 |

* 1. **Algorithms**

This paper addresses how to increase predicted accuracy with preprocessing and algorithms. Hence, we will focus on the algorithm formulas to improve accuracy instead of considering algorithm principles in this section. The general solution is to compare the mean accuracy or testing accuracy from trying different penalty values, different loss functions, and different penalty functions. However, these unstable algorithms will have different accuracy with the same penalty values, different loss functions, and different penalty functions, and Gaussian Naïve Bayes has no parameter to improve accuracy, we just consider the these stable algorithms: Linear SVM and Nearest Centroid Classifier.

* + 1. **Linear SVM**

Learning an SVM has been formulated as a constrained optimization problem over w and ξ:

For this formula, three parameters could be edited to improve accuracy.

* C: The penalty parameter of the error term.
* : Specifies the loss function with ‘hinge’ or ‘square of the hinge lose’.

Even though the green line is flat, the accuracy has low coefficient of variation almost 0.1%. The data of the below image indicates the best accuracy will be caused by the lost function, ‘squared\_hinge’, with ‘0.8’ penalty from cross-validation functions.

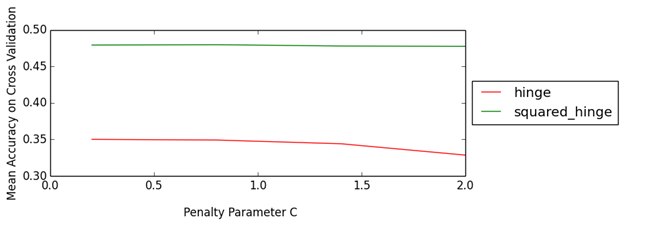


Figure 4.

* + 1. **Nearest Centroid Classifier**

For Nearest centroid classifier algorithm, “Shrink\_threshod” parameter will reduce the value of feature. For taking the best “Shrink\_threshod”, the Nearest Centroid Classifier will run time 10 times with values of “Shrink\_threshod” parameter from 0 to 2. However, the result shows that the best with values of “Shrink\_threshod” is 0 for this problem.

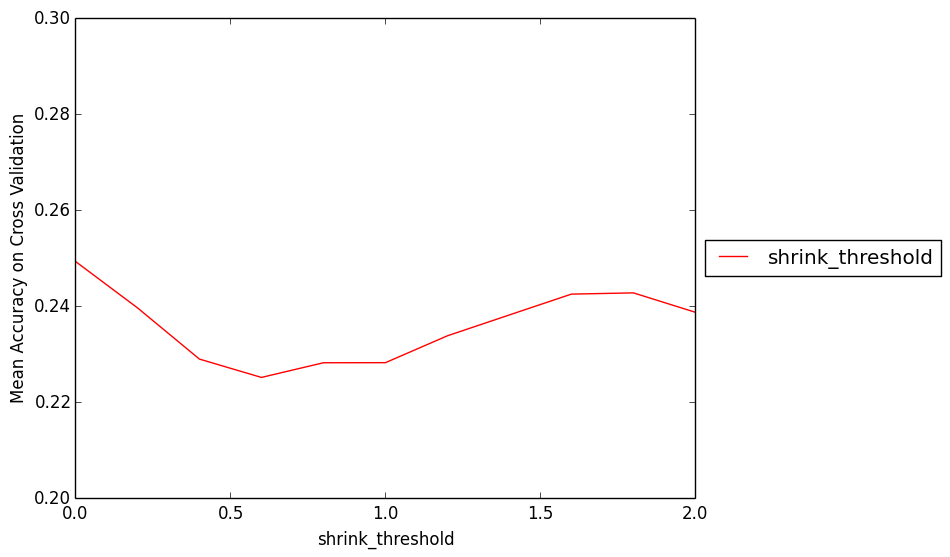


Figure 5.

* + 1. **Update accuracy**

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Accuracy | Training Time | Mean Error |
| Linear SVM | 42.7% | 1.4s | 0.8 |
| Stochastic Gradient Descent | 26.5% | 0.04s | 0.9 |
| Kernel Approximation | 27.4% | 0.05s | 1.2 |
| Nearest Centroid Classifier | 10.2% | 0.02s | 2.7 |
| Gaussian Naive Bayes | 4.3% | 0.02s | 3.5 |
| Forests of randomized trees | 36.9% | 0.11s | 0.9 |

1. **Model Ensembles**

For the current situation, the Linear SVM algorithm gets the highest accuracy. It does not mean that others algorithms are not useful. Base on the theory of Ensemble method, we consider to use stacking method to combine all algorithm predictions and predict final values with weight vote like Figure 6. 

Figure 6.

We could only get about 40% accuracy with 6 algorithms for 10 times. Then, we assume that Kernel Approximation, NCC, GNB have lower accuracy and? cause worse accuracy with stacking method. Then, we tried to use stacking method with the first 3 highest algorithms. Then, the accuracy is 42% for 10 times. The accuracy is higher than Linear SVM with 20% times. We conclude that stacking method may not work well than the single algorithm (Džeroski, 2004).

1. **Evaluation**

In this section, we will show three diagrams to explain all algorithms in each step of preprocessing. The first diagram indicates the accuracy. We could conclude that:

* The best step to improve accuracy is Normalization. Bag-of-Words is secondary.
* When the features values are so large, some algorithms may not work well such as Linear SVM and Nearest Centroid Classifier.
* The duplicated features and primary features should be deleted to increase accuracy.
* The excessive features may not cause overfitting with Gaussian Naïve Bayes.
* Changing penalty functions?, loss functions, and penalty parameters may not work or neglected work.

Figure 7.

The figure below relates to training time.

* When the features values, some algorithms will have higher time cost.
* The normalization will reduce time cost extremely.
* The Bag-of-Words function will increase time cost
* Limit the number of attributes could lower time cost.

Figure 8.

For the last diagram, it shows that Normalization will tremendously reduce the mean error. In other words, it is the best way to increase accuracy.

Figure 9.

1. **Conclusion**

In this paper, we used IMDB dataset to predict the rating value with six algorithms. After each preprocessing strategies, all algorithms make models for testing data to get accuracy, mean error, and training time. For the final accuracy, the accuracy is 42.7% which is higher than 37%. It shows that the project is meaningful. According to the evaluation section, we conclude that Normalization is the best strategy to increase accuracy and decrease time cost. Also, we believe that Forests of Randomized Trees and Kernel Approximation, and Stochastic Gradient Descent could be used by complicated dataset widely. Linear SVM algorithm could get the highest accuracy on edited data. However, we believe that the final accuracy could be improved to certain degree with Checking Overfitting step because the K popular values of each feature could be different; but, taking best K will cost time exponentially. Furthermore, reducing the number of values in label feature will also increase ?improve? accuracy.

**Citation**

Bag-of-words model. (2017, March 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 09:05, March 16, 2017, from <https://en.wikipedia.org/w/index.php?title=Bag-of-words_model&oldid=769511144>

Daumé, H. (2017) *A Course in Machine Learning*. Retrieved from <http://ciml.info/>

Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one?. *Machine learning*, *54*(3), 255-273.

Kołcz, A., Chowdhury, A., & Alspector, J. (2003). Data duplication: An imbalance problem?.

Yang, J., Jiang, Y. G., Hauptmann, A. G., & Ngo, C. W. (2007, September). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval* (pp. 197-206). ACM.