PREDICTING MOVIE SUCCESS FROM SEARCH QUERY USING SUPPORT VECTOR REGRESSION METHOD

Chanseung Lee[1] and Mina Jung[2]

[1]Sheldon High School, Eugene, OR, USA
[2]Electrical Engineering and Computer Science, Syracuse University
Syracuse, NY, USA

ABSTRACT

Query data from search engines can provide many insights about the human behavior. Therefore, massive data resulting from human interactions may offer a new perspective on the behavior of the market. By analyzing Google query database for search terms, we present a method of analyzing large numbers of search queries to predict outcomes such as movie incomes. Our results illustrate the potential of combining extensive behavioral data sets that offer a better understanding of collective human behavior.

KEYWORDS

Prediction; Support Vector Regression; Linear Regression; Ridge Regression

1. INTRODUCTION

Analyzing search queries from the Internet users can provide extensive societal applications. For example, Ginsberg et al. [1] estimated the spread of influenza in the United States based on Google search logs. They processed search queries from Google search logs and generated a comprehensive model for use in influenza surveillance, with regional and state-level estimates of influenza-like illness activity in the United States.

Predicting the financial success of a movie is a challenging, open problem. Sharda and Delen [2] have trained a neural network to process pre-release data, such as quality and popularity variables, and classified movies into nine categories according to their anticipated income, from “flop” to “blockbuster”. With test samples, it is evident that the neural network classifies only 36.9% of the movies correctly, while 75.2% of the movies are at most one category away from the anticipated category.

Joshi et al. [3] have built a multivariate linear regression model that joined meta-data with text features from pre-release critiques to predict the revenue with a coefficient of determination.

Since predictions based on classic quality factors fail to reach a level of accuracy high enough for practical applications, the usage of user-generated data to predict the success of a movie becomes a very tempting approach.

Mestyan et al. [4] built a predictive model for the financial success of movies based on collective activity data of online users. They show that the popularity of a movie can be predicted before its

DOI : 10.5121/ijaia.2016.7101

release by measuring and analyzing the activity level of editors and viewers of the corresponding entry to the movie in Wikipedia.

We try to find out whether a statistical prediction of the box-office success of a movie can be made using worldwide searchers' queries along with other information such as rating, the quantity of theaters the movie was starred in, and the length of the title. Our proposed system builds an automated method of predicting movie incomes from search queries.

In [5], we used Linear Regression and Ridge Regression to predict movie incomes. However, since both Linear Regression and Ridge Regression are linear models, they fitted well only in linearly separable problems. To overcome this linearity problem, we improved the method by using a non-linear regression method with kernel function. The kernel function expands current input space into a higher dimension, and automatically finds an optimal model in the expanded dimension. Specifically, we use Support Vector Regression (SVR) method with Polykernel function. Our experimental results show that our kernel method SVR improves the performance of the system.

In the next section we describe the data sources used in the prediction, and the characteristics of query data and prediction process are described in Section 3. In Section 4, we explain predicting methods and experimental results. Section 5 concludes this paper and suggests our future directions.

2. DATA SOURCES

2.1 Data Sources

We collect the movie query data as the main variable from the Google Trends to predict a success of a movie. In order to improve the prediction, we use three additional information: Income(I), Rate(R), and number of theaters(T) from Box Office Mojo site [6]. While we collected movie data released in America from January to March in 2014 in the previous work [5], the number of movies are now doubled by adding data from April to June in 2014. The data variables are described in the following.

1) **Movie Query:** Google Trends [7] provides weekly search query frequencies of movie titles. Instead of using weekly query data, query results of four weeks are combined into one variable. From the query data, we extract four variables (M1, M2, M3, M4) where M1 means the sum of queries during 4 weeks before the release date, and M2 means queries of 5 to 8 weeks before the release date, and so on. Altogether, we are using query data of total 16 weeks as input. Our first hypothesis is that people who enter queries of movie title are interested in the movie, and thus likely to contribute to the movie's income.

   Sometimes the words within movie titles are so common (e.g. Highway) that the queries about the titles of the movies extensively overlap with unrelated queries. In these cases, other related queries including actors and directors are collected, and the average value of these query data are used.

2) **Rate (R):** We use the rating (G, PG, PG-13, R, NR) of a movie.

3) **Number of theaters (T):** The number of theaters the movie made it to.

4) **Number of words in the movie title (W):** It is sometimes known that movies with simple titles succeed. We want to see the effect of the number of words in movie title.

5) Income of movie (I): This is the target variable we would like to predict. For the income of the movie, we used the income of the first weekend after the release date. For computational simplicity, logarithmic value of income is predicted in this paper.

2.2 Prediction Process

The process of entire prediction consists of three parts: *Data Collection, Data Preprocessing, and Prediction* .

(a) Data Collection: This process uses eight variables. Among the collected variables, rate, number of theaters the movie reached, first week income, and release date are collected from Box Office Mojo website, and entered into the Excel data file. For each movie, we entered the movie title into Google Trends, and downloaded the query data along with the rest in a csv form. The contents of these data are described in the following section.

(b) Data Preprocessing: We developed a Java program which reads data from the process of Data Collection, and processes and stores them in a one single master file that is ready for multivariable regression.

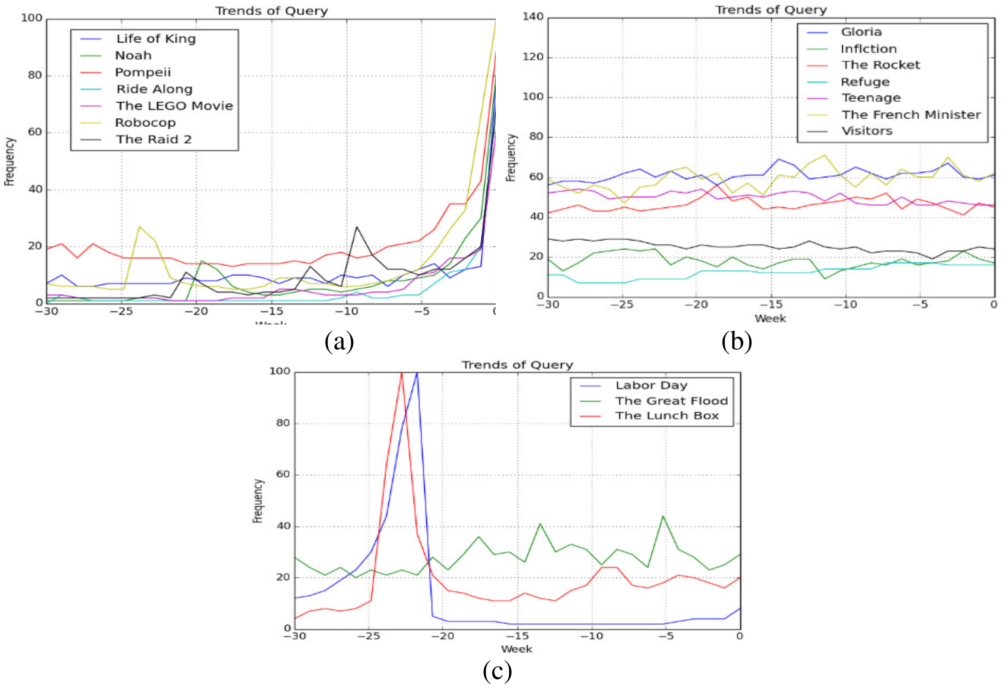
(c) Movie Prediction: We developed our income prediction system using Python. We applied several algorithms: Linear Regression, Regularized Linear Regression, and Support Vector Regression. These algorithms are from sklearn library [8] of Python language.

3 CHARACTERISTICS OF QUERY DATA

We could find some interesting characteristics of movie query data in terms of frequency trends. In most movies, as expected, their query frequencies drastically increase near the opening date. These movies include: *Life of a King, Noah, Pompeii, Ride Along, The LEGO Movie, RoboCop, The Raid 2, Veronica Mars, That Awkward Moment, Gang of Ghosts, Mistaken by Strangers, Muppets Most Wanted, Son of God, That Awkward Moment, The Monument Man, The Nut Job, etc*. As shown in Figure 1(a), the number of queries about these movies dramatically increases near the opening date of the movie. In this graph, the x axis is the number of weeks before a movie was released and the y axis is the frequency of the search queries of the movie.

In other cases, the movies' titles are so common that their queries do not show significant increase near opening date. These movies include *Gloria, Infliction, The Rocket, Refuge, Teenage, The French Minister, Visitors, Highway, etc*. When people type in these queries, the queries are not always directed towards the movie. For example, when a Google user enters query "Teenage", could mean the movie "Teenage" or teenager itself. For these movies, the queries of movie title are not a good indicator of movie income. Therefore, we use query data for the names of main actors and director of the movie. These queries are combined and their average is used as input. Figure 1(b) shows the frequency trends of using general terms in movie titles.

Queries of some movies are affected by social factors. These movies include *Labor Day* (at peak near Labor Day), *The Great Flood* (at peak when an actual flood strikes), *The Lunchbox* (launched in India first), *etc*. For instance, "Labor Day" might mean the actual Labor Day or movie. Figure 1(c) shows the patterns of queries about these movies. We used the same approach used in Figure 1(b) of movies with general/common terms. We used queries about main actor and director of the movie, and their average was used in the following experiments in Section 4.



(a)

(b)

(c)

Figure 1: Characteristics of movie query frequencies

4 PREDICTION METHODS AND EXPERIMENTS

We have used Linear Regression and Ridge Regression to predict movie incomes in ourearlier work [5]. However, since those methods fit well only in linearly separable problems, we apply a new non-linear regression method with kernel function. Specifically, we use Support Vector Regression (SVR) method with Polykernel function.

To collect data, we manually typed in movie titles at Google Trends and downloaded query data in csv form. We could collect as many as 298 movies released from January to June of 2014 from Google Trends and 8 movies were omitted due to lack of sufficient query data. We used this data for the following prediction methods.

4.1 Linear Regression Method

Predicting continuous outcome is a regression problem in statistics. Simple Multivariate Linear Regression [9] is the first method we chose to attempt. Multivariate Linear Regression creates a straight line among a scatterplot that minimizes residuals, i.e. sum of squared errors given as follows

$$\mathrm{E} = \min \sum_i (I_T - I_P)^2 \qquad (1)$$

where I_T and I_P mean the true income and the predicted income of the movies, respectively. We look for the following linear hyperplane which minimizes the above error E.

$$\mathrm{I_P} = \mathrm{w_R} \cdot R + w_{T.} \cdot T + w_W \cdot W + w_{M1} \cdot M1 + w_{M2} \cdot M2 + w_{M3} \cdot M3 + w_{M4} \cdot M4 + w_0 (2)$$

In this formula, R, T, W, M1, M2, M3, and M4 are the variables that we use for linear regression while *w* means the corresponding weight of each variable. The goal of learning is to correctly estimate the weight (*w's*) of each variable in regression line.

We developed a Phython program that uses Multivariate Linear Regression from sklearn library [8] of Python. Parameters of Multivariate Linear Regression were set to be the default values. We have a total of 298 movie data; 70% of them is used a training data and 30% is used for testing data. Figure 2 shows the relationship between *true Income* and *predicted Income* of the movies which was estimated by the input variables. As we can see in Figure 2, the result was promising. Many prediction points are gathered near/around diagonal line, and its mean square error (MSE) is 6.08. These results show that we are able to predict movie incomes by using query data with a reasonable accuracy.

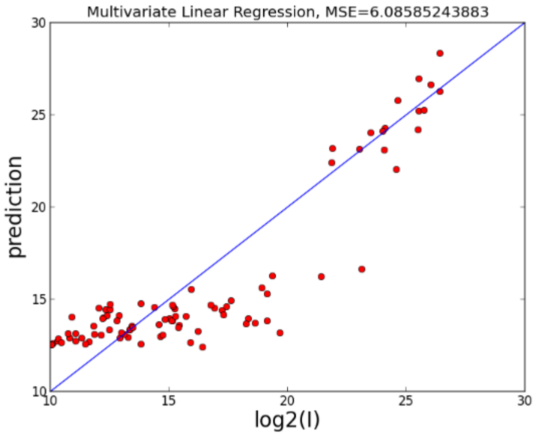


Figure 2: Result of multivariate linear regression

4.2 Ridge Linear Regression

The next question is whether we can improve the performance of linear regression model in predicting movie incomes. We carried out the second experiment using a different linear regression model, called Regularized Linear Regression (RLR). RLR is known to be more robust to noise and outliers in data [10]. When noises in data are expected, regularized regression is a powerful method to disregard noise and solve the problem. RLR is less sensitive to outliers by minimizing both the sum of error and *regularization term*, as given as in the following,

$$\mathrm{E} = \min \sum \ (I_T - I_P)^2 + \alpha \cdot w_i^2 \quad (3)$$

where $\mathrm{w_i}$ means the coefficient of input variables (R, T, W, M1-M4, etc). The regularization term is the square of each coefficient $\mathrm{w_i}$ and α signifies the regularization constant. The higher the value of $\mathrm{w_i}'s$ are, the higher the value of E is.

Figure 3 shows the reason why Ridge Regression is robust to noise. When outliers (noise) are present as green dots within the green circle, the purple dashed line is the linear regression line, which is not correct one. In RLR, by adding a regularization term, RLR is less sensitive to outliers and makes a more reasonable line, the red line which is far from the outliers.

We run RLR function in sklearn library [8] with default parameter values. Figure 4 shows the relationship between *true Income* and *predicted Income* using RLR. Many predictions are gathered near the diagonal line, but were a bit more spread than linear regression model. The result was worse than Simple Linear Regression, and the RLR has a greater error of MSE 7.47. This result tells us that, in the proposed movie prediction system, most input data are not outliers nor noise. Therefore, the regularized linear regression actually degrades the accuracy of the prediction system.

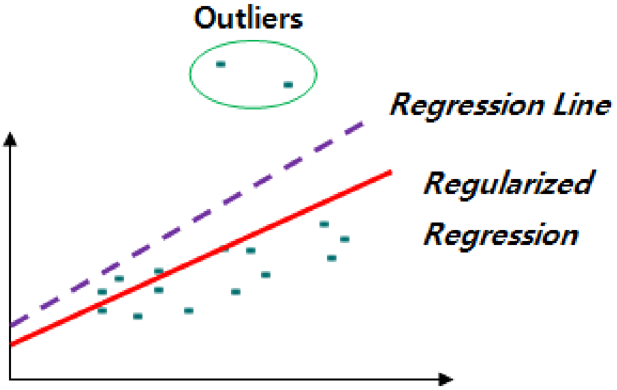


Figure 3: Regularized Linear Regression

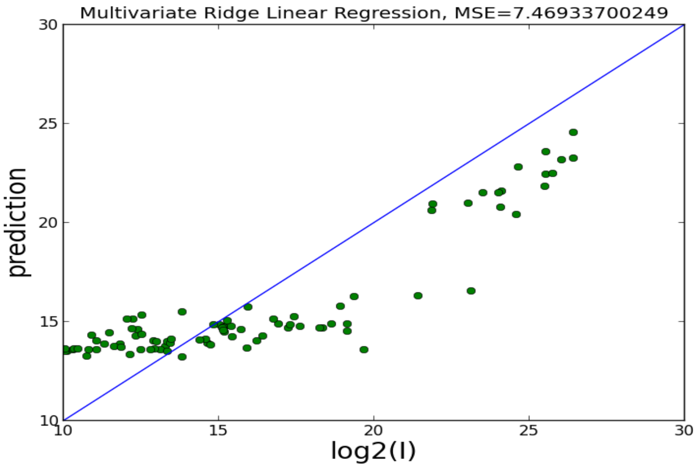
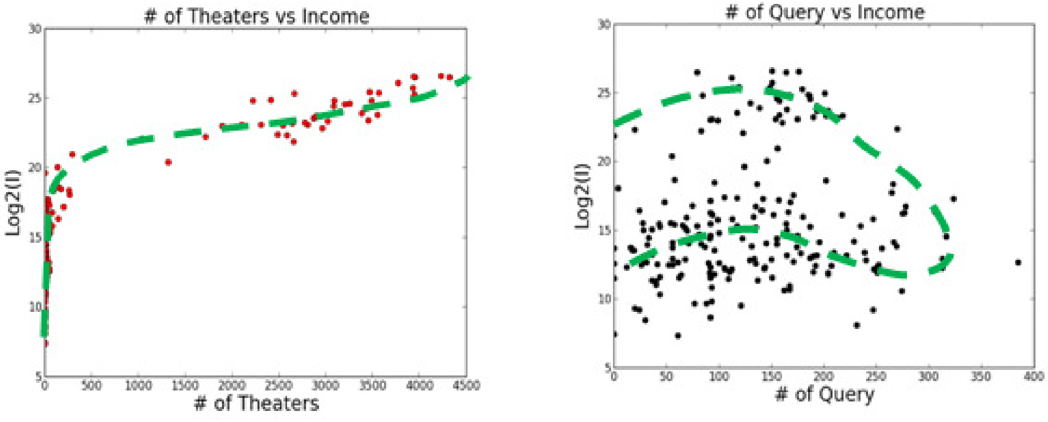


Figure 4: Results of Regularized Regression

4.3 Non Linearity Problem and Kernel Function

In Section 4.1 and 4.2, we used a multivariate linear regression model and ridge linear regression. In both of these methods, a straight line is described by a simple equation that calculates movie income from input variables. The purpose of these linear regression-like model is to find the optimal values for the slope that defines the linear line that comes closest to the data.

However, the biggest drawback of these models is that they can only solve linear problems. Linear problems are learning problems where the class values can be approximated from the input by linear hyperplanes. These linear algorithms cannot fit them in a non-linear problem environment. We know that vast majority of real world problems belong to non-linear problems, and this makes linear regression model obsolete. Like most of other real world problems, the learning problem for movie prediction is not a linear problem and does not necessarily follow a straight line.



(a) (b)

Figure 5: Non-linearity in input variables

Figure 5 shows the example of non-linearity relationships of the input variables. Figure 5(a) shows the relationship between number of theaters and movie income. It clearly shows that a straight line cannot effectively describe the relationship. Figure 5(b) shows even more complex fitting model between number of queries and movie income. We can also infer that the simple specific input variable, number of theaters keeps closer relationship with the movie income than the conventional input variable, number of queries.

In this paper, we adopt a kernel technique to solve this non-linearity problem. The basic idea of kernel method is to expand original dimension into higher dimension, and find a linear line in the expanded dimension. Figure 6 shows an example of kernel method where 2D input space is expanded to 3D feature space. We can see that there is a linear hyperplane in 3D feature space.

Suppose Φ is the mapping from original dimension to expanded dimension. If we transform our points to feature space, the scalar product form of two original data x1 and x2 looks like this:

$$<\Phi(x1), \Phi(x2)>$$

However, computing the value of Φ(x1) and Φ(x2) is very time-consuming and difficult. The key insight of kernel method is that there exists a class of functions called *kernel functions* that can be used to optimize the computation of this scalar product. Instead of computing Φ(x1) and Φ(x2), respectively, the whole term can be replaced by a kernel function. A kernel is a function K(x1, x2) that has the following property

$$K(x1, x2) =< \Phi(x1), \Phi(x2)>$$

for some mapping Φ. In other words, we can evaluate the scalar product of data in the low-dimensional data space without having to transform to the high-dimensional feature space.

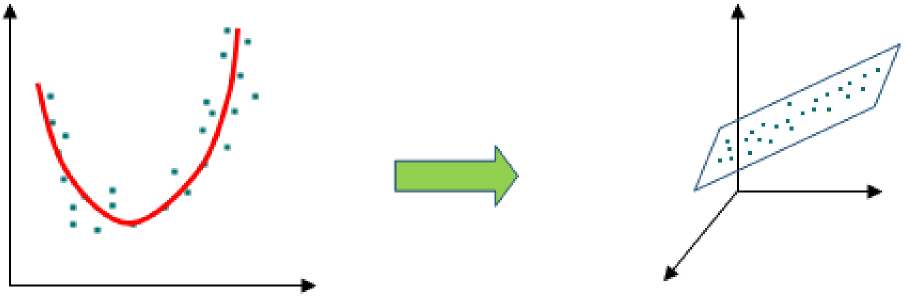


Figure6: Kernel Method

4.4 Support Vector Regression Method

Support Vector Regression (SVR) method [11] is the most popular kernel regression method, so we use SVR method as kernel algorithm. Kernel transforms current input dimension into an expanded dimension as described in Figure 6, and finds an optimal linear hyperplane in the expanded dimension. Many kernels are proposed in SVR method (e.g. Polynomial, RBF, sigmoid, etc) and we use polynomial kernel since it provides the importance of each variable as a result of learning. We used SOM module in Weka [12] software to run SVR algorithm. The result of applying SVR method is presented in Figure 7.

SVR provides the best accuracy compared to the previous regression methods. Its MSE is 4.85, which is a significant improvement than the two other regression methods. Polynomial kernel generates the important/weight of each input variable. Table 1 shows the importance of variables from SVR with polynomial kernel. As we can see, in predicting movie incomes, the number of theater is the most important factor and the query of the 1st and 2ndmonths are the second most important factors. The oldest query (M4) and number of words are least influence factors in the prediction.

We carried out another experiment for SVR prediction model. The original variable set contains four query variables (M1, M2, M3, and M4). Among them, we select only two most recent query (M1, M2) and run the program. The result is shown in Figure 8. The MSE(=4.91) of this model is very close to that of Figure 7, the original SVR. Therefore, when predicting movie incomes, we can conclude that two month of recent query is enough to do the job. These experiments clearly show that query data are very important indicators of movie incomes, and we can predict the outcomes quite accurately.

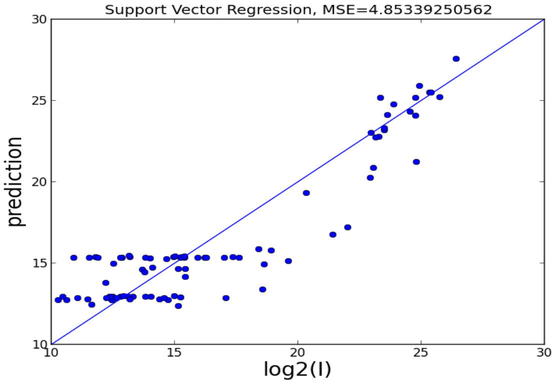


Table1: Importance of each variable

Attribute	**Weight**
Rate	0.1076
# of Theater	**0.5613**
# of Words	0.0285
1st Month query (M1)	**0.1503**
2nd Month query (M2)	**0.2972**
3rd Month query (M3)	0.0976
4th Month query (M4)	0.0643

4.5 Discussions

We have applied a number of methods, including linear regression, ridge linear regression, and support vector regression, to the problem of predicting movie income. Among them, support vector regression method showed the best performance with MSE=4.85, while MSE of linear regression and ridge regression are, 6.08 and 7.47, respectively.

Since the problem we attack has the characteristic of non-linearity, as shown in Figure 5, support vector regression, which can effectively process non-linear problems,showed best prediction results.

5 CONCLUSIONS

We developed a software program that can predict the future incomes of movies using Google search query data and some additional information about movies. We found that a movie's success (total income) is largely based on the number of theaters and searchers' queries. It is also slightly affected by its length of title and rating.

We tried several regression methods including Simple Linear Regression, Ridge Linear Regression and Support Vector Regression. Among them Support Vector Regression using Polynomial kernel shows the best performance in predicting movie incomes.

In the future, we will include some other important variables about query and/or movie such as trend of query, regional data, and so on. By using search query data, we will be able to predict many things in society. We will apply our prediction system to other interesting domains and predict various phenomena using query data.

REFERENCES

[1] Jeremy Ginsberg et al. “Detecting influenza epidemics using search engine query data” Nature,45(9), 2009.

[2] Ramesh Shardaand Dursun Delen. “Predicting box-office success of motion pictures with neural networks” Expert Systems with Applications 30(2), 2006

[3] Mahesh Joshi, Dipanjan Das, Kevin Gimpel and Noah A. Smith. “Movie reviews and revenues: An experiment in text regression” In Proceedings of NAACL-HLT 2010, 2010.

[4] Marton Mestyan, Taha Yasseri, and Janos Kertesz. “Early prediction of movie box office success based on wikipedia activity” PLoS ONE, 8(8), 2013.

[5] Chanseung Lee and Mina Jung “Predicting Movie Incomes Using Search Engine Query Data”International Conference on Artificial Intelligence and Pattern Recognition (AIPR), 2014

[6] Box office Mojo. http://www.boxofficemojo.com.

[7] Google trends. http://www.google.com/trends.

[8] Scikit-learn: Machine learning in python. http://scikit-learn.org.

[9] Wikipedia. http://en.wikipedia.org/wiki/Linear_regression

[10] Wikipedia https://en.wikipedia.org/wiki/Ridge_regression

[11] H. Drucker, Kaufman Burges, and V. Smola. Support Vector Regression Machines, NIPS, 1996

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H.Witten; “The WEKA Data Mining Software: An Update” SIGKDD Explorations, Volume 11, Issue 1, 2009