SHRINKING THE UNCERTAINTY IN ONLINE SALES PREDICTION WITH TIME SERIES ANALYSIS

Rashmi Ranjan Dhal[1] and B.V.A.N.S.S. Prabhakar Rao[2]
School of Computing Science and Engineering, VIT University, Chennai, India
Email: [1]rashmi.ranjan2013@vit.ac.in, [2]prabhakarrao@vit.ac.in

Abstract
In any production environment, processing is centered on the manufacture of products. It is important to get adequate volumes of orders for those products. However, merely getting orders is not enough for the long-term sustainability of multinationals. They need to know the demand for their products well in advance in order to compete and win in a highly competitive market. To assess the demand of a product we need to track its order behavior and predict the future response of customers depending on the present dataset as well as historical dataset. In this paper we propose a systematic, time-series based scheme to perform this task using the Hadoop framework and Holt-Winter prediction function in the R environment to show the sales forecast for forthcoming years.

Keywords:
Time Series Analysis, Sales Prediction, Hadoop Distributed File System, Holt-Winters Function, Seasonal and Level Variant

1. INTRODUCTION

In order to analyze the marketability of any particular product, the primary requirement is to assess what the current response of people on that product is. We can get the response of people from service outlets where products are sold or from the web where people comment freely on that product. Social media offer a very popular and powerful platform where manufacturers, retailers and customers can share their views and opinions. Social media is like a highly active news channel where each and every incident happening all over the world is published in real time [1]. It is a convenient platform where all involved can sit in their homes and discuss their personal experiences about various products. Moreover, they can also use a variety of devices like mobile, ipad, tab etc. to give their responses. From research point of view, we can tap the dataset collected from different social networks such as Twitter and use posts, comments, likes and other features for analyzing a particular product.

As the future is always unpredictable, it is a major challenge for business analysts to decide about the outcome after the launch of any particular product in the market. Our primary aim here is to track customers' behavior for a given product and based on an analysis of their response predict the future response of customers.

The guiding principle of soft computing is to exploit the imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low cost solutions. Indeed, the employment of soft computing for the solution to machine learning problems has led to high Machine Intelligence Quotient [2, 3, 4, 5].

In section 3, we discuss about the R programming environment, Hadoop system and JSON. In section 4 we briefly describe time series analysis and how it works on sales data. In section 5 we give the proposed model for prediction which is the complete work flow of the research. In section 6 we show the results of the experiment on the final prediction on sales data.

2. TECHNIQUES AND TOOLS

2.1 PREDICTIVE MODELING APPROACHES

Generally three techniques are deployed for predictive modeling. These are traditional techniques, data adaptive techniques and model dependent techniques. Fig.1 depicts these models diagrammatically.

Traditional approaches include linear regression and logistic regression. As depicted in Fig.1(a), they create a model which is designed to fit the underlying data. After model fitting, one applies predictor functions to analyze the possible response.

Data adaptive approaches find the most significant parameters which affect the prediction predominantly. As depicted in Fig.1(b), these approaches emphasize on the underlying hypothesis rather than performing a direct data analysis.

Model dependent approaches use analytical methods to generate data; predictor functions etc. (refer Fig.1(c)). These include mathematical modeling, Linear Programming and Operations Research.

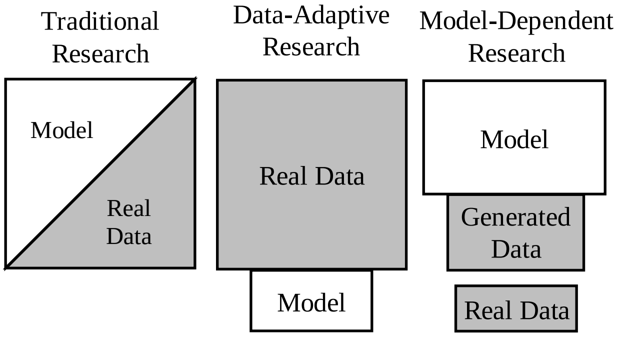

Fig.1.Types of Predictive Modeling: (a). Traditional approach, (b). Data adaptive approach, (c). Model dependent approach

Time series analysis is a data adaptive approach. The time series is a collection of observations of well-defined data items obtained through repeated measurements over time. For example, measuring the value of retail sales each month of the year comprises a time series. This is possible because sales revenue is well defined, and consistently measured at equally spaced intervals. For our research, we used sales data collected from time to time.

2.2 THE R, HADOOP, JSON ENVIRONMENT

R is a programming language for statistical data analysis and visualization. Data is imported with the CSV (Comma Separated

Value) or Semicolon Separated Value (SSV) formats for analysis using a variety of techniques such as linear / non-linear modeling, time series analysis, statistical analysis and clustering. The appropriate model is coded with the R syntax [6].

The Hadoop framework provides extensive and open support for storage and processing of large scale datasets. Here Hadoop is used for efficient storage of data and hive is used to create a table for the unstructured data. As the processing of data in companies such as eBay is complex and very large, Hadoop provides a convenient distributed platform for its storage. The Hadoop Distributed File System (HDFS) spreads data storage across multiple nodes. This file system has one parent node called Name node which has a directory tree of all files and multiple Data nodes which stores the actual data. The Job Tracker residing in the Name node creates jobs and allocates them to the Task trackers residing in Data nodes. The HADOOP High Availability distributed environment has adequate redundancy to tackle single point of failure and facilitate high availability. Hive is a query interface which can be used just as SQL to query data residing on HDFS. The query language called HQL (Hive Query Language) has a format similar to SQL [6], [7].

JSON (Java Script Object Notation) is a data interchange format. JSON is built on two structures:

1) A collection of name/value pairs. In various languages, this is realized as an object, record, dictionary, hash table, keyed list, or associative array.
2) An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

3. TIME SERIES ANALYSYS IN PRODUCTION ENVIRONMENT

3.1 PROCESSING ENVIRONMENT

The complete work flow of the product sales analysis is shown in Fig.2. The product sales datasets are readily available at different websites and social networking sites. For collecting data we created an API which streams real time data into our system. As the streaming input data is in random format, we convert it into JSON (Java Script Object Notation) format for further analysis.

A major drawback with our traditional RDBMS (Relational Data Base Management) is that it cannot handle unstructured data. To overcome this, we use the HADOOP eco system which includes Hive and HBase. The data is streamed into HDFS and then queried using Hive. The code can be written for data streaming from any particular website by an access token and a secret key provided by the server. Apache Flume can also be used to gather data. It works on a source-sink mechanism. In Flume, each data arrival is an event. The source produces data (event) and sends it through a secure channel to the sink. The sink then writes the data to predefined location i.e. HDFS files. Then the data is preprocessed to CSV format in matrix format. On this data frame we can forecast the future orders by using R.

The data that is captured using the eBay API is stored in Hadoop File System. Using Hive a table is created for the unstructured data.

3.2 SMOOTHING OF TIME SERIES AND PREDICTION

Traditional time series analysis describes the function as combination of three factors i.e. trend *T*, seasonal component *S* and residual *e*. We can evaluate the trend of time series by using non parametric regression techniques. For this we call a function st_1(.) which performs seasonal decomposition of the time series and determines the parameters *T*, *S* and *e*.

For predicting the next value of given time series X_t during any time period 'Δt', we require the weighted sum of past observations. The prediction period is set to 48 months. The more recent data is preferred as compared with previous data for prediction. Here we use exponential smoothing because it is suitable for a time series which has no symmetrical trend or seasonal component. Therefore, we use the Holt-Winters function for this type of problem.

We require three parameters i.e. level (α), trend (β), seasonal variation (γ). Holt Winters (data) function performs its action on time series't'. We can exclude the value of the trend parameter β, as $\beta = 0$, because the estimate of slope of trend component has not been updated over the time series.

Seasonal effect is a systematic and calendar related effect. Some examples include the sharp escalation in most retail series which occurs during festive seasons, or a marked increase in power consumption due to running air conditioners during summers.

Trend is continuous and long term effect in a time series that is not related to calendar events and is devoid of irregular effects. It is a reflection of the cumulative effects of the underlying processes. It is influenced by several factors such as price inflation, general economic policies, population dynamics etc.

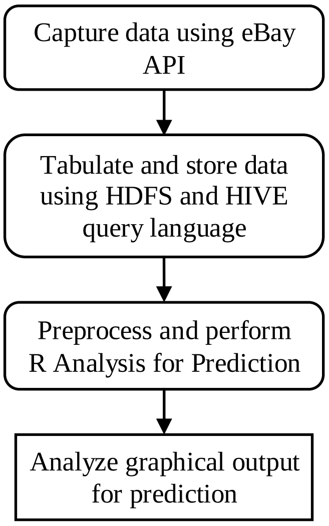
Capture data using eBay API
Tabulate and store data using HDFS and HIVE query language
Preprocess and perform R Analysis for Prediction
Analyze graphical output for prediction

Fig.2. Work-Flow diagram for time-series analysis

Let us examine the step-wise procedure for data analysis. The unstructured data is captured and stored into HDFS by translating it in the form of a table using the Hive query language. Next, the data is converted into csv format from JSON format. The following steps are required in R for further processing:

Step 1: Read and Convert input data:

The time series contains a lot of information other than the value of data, such as dates and frequency at which the time series was recorded [14],[22]. In order to encapsulate all information about a time series, the raw data must be converted to a time-series object by invoking the function *ts-object* (.). The command below loads dataset in CSV format and transforms it to ts-object.

data←*read.csv*(“*C:/data.csv*”, *header* = T, dec = “,”,sep = “;”)

data ← *ts*(*data*[,1], *start* = 2014, *freq* = 12)

In the above command, data [,1] reads the first column of the table the data arrivals are monthly as frequency is set to 12.

Step 2: Smoothen and Extract parameters:

We can fit a predictive model using the HoltWinters() function.

HoltWinters(*data*)

The above command performs the Holt-Winters function on the dataset and display its parameters namely, level, trend and seasonal component.

Step 3: Perform Prediction:

Here we fit the model with the data for next 12 months:

data.hw ← *HoltWinters* (*data*)

predict (*data.hw*, *n.ahead* =12)

For prediction we need to save the fitted model into an object named data.hw. The above commands predict the values of the data series for next 12 units of time.

Step 4: Represent Graphically:

We show the prediction for next 2 years graphically by invoking the following commands:

plot (*data*, *xlim* = *c*(2014, 2018))

lines (*predict*(*data.hw*, *n.ahead* = 48), col = 2)

The above command predicts and plots the data for 4 years between 2014 and 2018.

4. HOLT-WINTER PREDICTION MODEL

This model uses a process known as exponential smoothing. All data values in a series contribute to the calculation of the prediction model.

Exponential smoothing in its simplest form should only be used for non-seasonal time series exhibiting a constant trend (or what is known as a stationary time series). It seems a reasonable assumption to give more weight to the more recent data values and less weight to the data values from further in the past. An intuitive set of weights is the set of weights that decrease each time by a constant ratio. Strictly speaking this implies an infinite number of past observations but in practice there will be a finite number. Such a procedure is known as *exponential smoothing* since the weights lie on an exponential curve.

If the smoothed series is denoted by St, α denotes the smoothing parameter, the exponential smoothing constant, $0<\alpha<1$.The smoothed series is given by:

$$S_t = \alpha y_t + (1 - \alpha)S_{t-1}$$

where, $S_1 = y_1$. The smaller the value ofα, the smoother the resulting series. It can be shown that:

$$S_t = \alpha y_t + \alpha(1 - \alpha)y_{t-1} + \alpha(1 - \alpha)^2 y_{t-2} + \ldots + (1 - \alpha)_{t-1}\, y_1$$

Consider the following Time Series: 14, 24, 5, 18, 10, 17, 23, 17, 23,…. Using the formulae above, with an exponential smoothing constant, $\alpha = 0.1$, we get:

$$S_1 = y_1 = 14$$

$$S_2 = \alpha y_2 + (1 - \alpha)S_1 = 0.1(24) + 0.9(14) = 15$$

$$S_3 = \alpha y_3 + (1 - \alpha)S_2 = 0.1(5) + 0.9(15) = 14$$

$$S_4 = \alpha y_4 + (1 - \alpha)S_3 = 0.1(18) + 0.9(14) = 14.4 \text{ etc}$$

Thus the smoothed series depends on all previous values, with the most weight given to the most recent values. Exponential smoothing requires a large number of observations. Exponential smoothing is not appropriate for data that has a seasonal component, cycle or trend. However, modified methods of exponential smoothing are available to deal with data containing these components. The Holt-Winters model uses a modified form of exponential smoothing. It applies three exponential smoothing formulae to the series. Firstly, the level (or mean) is smoothed to give a local average value for the series. Secondly, the trend is smoothed and lastly each seasonal sub-series (i.e. all the January values, all the February value for monthly data) is smoothed separately to give a seasonal estimate for each of the seasons. A combination of these three series is used to calculate the predictions output.

The exponential smoothing formulae applied to a series with a trend and constant seasonal component using the Holt-Winters additive technique are:

$$a_t = \alpha(Y_t - S_{t-p}) + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$S_t = \gamma(Y_t - a_t) + (1-\gamma)S_{t-p}$$

where,

α, β and γ are the smoothing parameters

a_t is the smoothed level at time t

b_t is the change in the trend at time t

s_t is the seasonal smooth at time t

p is the number of seasons per year

The Holt-Winters algorithm requires starting (or initializing) values. Most commonly:

$$a_p = \frac{1}{p}(Y_1 + Y_2 + \ldots + Y_p)$$

$$b_p = \frac{1}{p}\left[\frac{Y_{p+1} - Y_1}{p} + \frac{Y_{p+2} - Y_2}{p} + \ldots + \frac{Y_{p+p} - Y_p}{p}\right]$$

$$s_1 = Y_1 - a_p, \quad s_2 = Y_2 - a_p, \quad \ldots, \quad s_p = Y_p - a_p$$

The Holt-Winters forecasts are then calculated using the latest estimates from the appropriate exponential smoothes that have been applied to the series. So we have our forecast for time period,

$T + \tau$:

$$\hat{y}_{T+\tau} = a_T + \tau\ b_T + s_T$$

where, a_T is the smoothed estimate of the level at time T, b_T is the smoothed estimate of the change in the trend value at time T and s_T is the smoothed estimate of the appropriate seasonal component at T.

As mentioned earlier the Holt-Winters model assumes that the seasonal pattern is relatively constant over the time period. Students would be expected to notice changes in the seasonal pattern and identify this as a potential problem with the model, particularly if long–term predictions are made. In practice this is dealt with by transforming the original data and modeling the transformed series or using a multiplicative model. Students are not expected to know this, but are required to identify a variable seasonal pattern as a potential problem. The exponential smoothing formulae applied to a series using Holt-Winters Multiplicative models are:

$$a_t = \alpha \frac{Y_t}{s_{t-p}} + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$s_t = \gamma \frac{Y_t}{a_t} + (1-\gamma)s_{t-p}$$

The initializing values are as for the additive model, except:

$$s_1 = \frac{Y_1}{a_p}, \quad s_2 = \frac{Y_2}{a_p}, \quad \ldots, \quad s_p = \frac{Y_p}{a_p}$$

So we have our prediction for time period $T+\tau$:

$$\hat{y}_{T+\tau} = (a_T + \tau \ b_T)s_T$$

5. EXPERIMENTAL RESULTS

For our experiments, we utilized the sales record available on the eBay web portal [22]. The eBay API was coded with java streaming code with an access token. The code was run on Net Beans platform and data was retrieved in real time manner.

The Fig.3 shows the sales info of a production environment. The data is the yearly men's clothes' sales record. It shows monthly sales info from 2007-2013. It is in CSV format. From the given data we plotted different graphs which are explained next.

The graph in Fig.4 depicts the sales records over a period of 7 years from 2007 to 2013 with the volumes of sales along Y-axis and months along X-axis. It can be observed that in the month of November and December the sales volumes increase sharply each year. Here there is a seasonal fluctuation and a random fluctuation that change with the time series.

In this case, it appears that an additive model is not appropriate for describing this time series, since the size of the seasonal fluctuations and random fluctuations seem to increase with the level of the time series.

An additive model computes the decomposition of the time series into its components, trend, seasonality, cyclical and error. It projects the identified parts to the future and sums the resulting projection to form the forecast. The model is assumed to be additive (that is all parts are summed up to give the forecast).

$$X'_t = T_t + S_t + C_t + \varepsilon_t$$

where, T is the trend, S is the seasonality, C is the cycle and ε is the error. Thus, we need to transform the time series in order to get a transformed time series that can be described using an additive model.

We apply logarithmic transformation of the time series by calculating the natural log of the original data. The transformed graph of logarithms of sales volumes versus time is shown in Fig.5. The Y axis shows that the sales data varies from 0-12000. We can observe that during the month of October-November the sales increased above 9000. This pattern repeats for each year end.

Here we can see that the size of the seasonal fluctuations and random fluctuations in the log-transformed time series seem to be roughly constant over time, and do not depend on the level of the time series. Thus, the log-transformed time series can be most conveniently described using an additive model. Because the Y scale is compressed between 0-20 as depicted in Fig.5, the size of the seasonal fluctuations and random fluctuations in the log-transformed time series seem to be roughly constant over time, and do not depend on the level of the time series. So the additive model can be used conveniently.

	Jan	**Feb**	**Mar**	**April**	**May**	**June**
2007	1664.81	2357.93	2840.71	3547.29	3752.96	3714.74
2008	2499.81	5198.24	7225.14	4806.03	5900.88	4951.34
2009	4717.02	5702.63	9957.58	5304.78	6492.43	6630.80
2010	5921.10	5814.58	12421.25	6369.77	7609.12	7224.75
2011	4826.64	6470.23	9638.77	8821.17	8722.37	10209.48
2012	7615.03	9849.69	14558.40	11587.33	9332.56	13082.09
2013	10243.24	11266.88	21826.84	17357.33	15997.79	18601.53
	July	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
2007	4349.61	3566.34	5021.82	6423.48	7600.60	19756.21
2008	6179.12	4752.15	5496.43	5835.10	12600.08	28541.72
2009	7349.62	8176.62	8573.17	9690.50	15151.84	34061.01
2010	8121.22	7979.25	8093.06	8476.70	17914.66	30114.41
2011	11276.55	12552.22	11637.39	13606.89	21822.11	45060.69
2012	16732.78	19888.61	23933.38	25391.35	36024.80	80721.71
2013	26155.15	28586.52	30505.41	30821.33	46634.38	104660.67

Fig.3. Data set of Sales Record

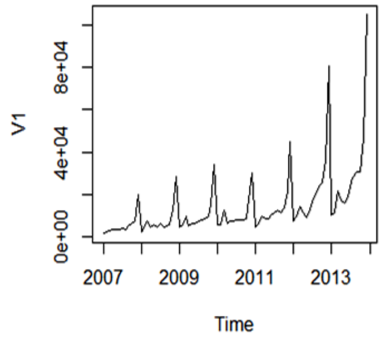

Fig.4. Chart Showing Sales Record

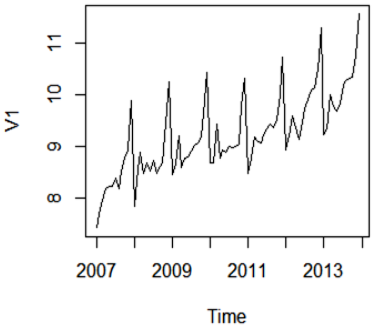

Fig.5. Log-transformed graph

Now the predictive model is fitted with the data set using Holt-Winter Function:

$$HoltWinters(x = logmydata)$$

Smoothing parameters: $\alpha = 0.413418$, $\beta = 0$, $\gamma = 0.9561275$.

Coefficients: [,1], a = 10.37661961, b = 0.02996319, s_1 = -0.80952063, s_2 = -0.60576477, s_3 = 0.01103238, s_4 = -0.24160551, s_5 = -0.35933517, s_6 = -0.18076683, s_7 = 0.07788605, s_8 = 0.10147055, s_9 = 0.09649353, s_{10} = 0.05197826, s_{11} = 0.41793637, s_{12} = 1.18088423.

The parameters α, β and γ represent the level, trend and seasonal fluctuation respectively. The parameter "*a*" is the start value of level, "*b*" is start value of trend and <***s***> is a vector of start values for the seasonal component ($s_1 \ldots s_{12}$).

Note that the parameters α, β and γ all have values between 0 and 1. Values close to 0 means that relatively little weight is placed on the most recent observations when making forecasts of future values. So it can be concluded that the value of α is partially dependent on present values and more dependent on past values. As there is no trend component, β equals zero. From the value of γ, it can be said that γ is predominantly dependent on recent values.

In Fig.6 the original time series is indicated by black line. The red line is the forecasted value. As the red line and black lines approximately move in a similar fashion it can be concluded that Holt-Winters exponential method is very successful in predicting the seasonal peaks, which occur roughly in November every year.

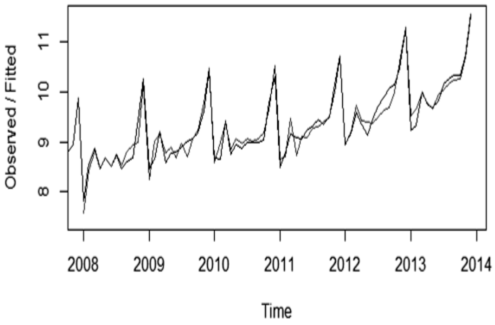

Fig.6. Holt Winter Filtering

To make forecasts for future times not included in the original time series, the "forecast.HoltWinters()" function in the "forecast" package is used.

> *data2* ← *forecast.HoltWinters* (*data*, *h* = 48)

> *plot.forecast*(*data*2)

The blue line in Fig.7 shows the estimated prediction and the prediction may vary in the range which is shown in blurring blue color area. The volume of data in 2014 varies like previous year. At the end of the year sales record is increased upto 12000. It also shows that in can fluctuate between 12500-11500 for 2014 year at end of year.

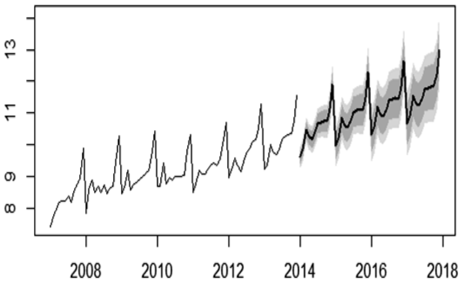

Fig.7. Holt-Winter Forecast

6. CONCLUSION AND FUTURE WORK

The above experiment illustrates how we can perform prediction analogy of manufacturing products and forecasting for realizing an efficient and cost saving business model. The sales data of eBay was forecasted for a span of next 4 years for which no time series data was available. The data is collected over API and stored at Hadoop Distributed file system in a table created by Hive. Then data is transformed for processing. Using Holt-Winter function from which the level and seasonal variation components factors can be determined. To get data in graphical format, data has to be gathered transformed to suitable object format. Is needed, transformations need to be applied to enable an additive prediction model. Finally, the model is applied on the data to get the prediction results graphically and the values of level, seasonal and s-components.

For our future work, we will perform data analysis by capturing the sentiments of users about a given product from social networking sites. Contents discussed in social media could help boost the performance of several applications by tapping the power of sentimental analysis, predictive analysis, and human computer analysis. For that, we plan to stream the data using Facebook Streaming API and Twitter Streaming API. After getting the data it will be stored either in MySql or in HDFS using Hive and HBase. When analysis is done in social media, our main aim will be to find the sentiments of collaborators in the social network including friends, followers, followees and their response to given products.

REFERENCES

[1] Jure Leskovec, Lada A. Adamic and Bernardo A. Huberman, "The dynamics of viral marketing", *Proceedings of the 7th ACM Conference on Electronic Commerce*, 2006.

[2] S. Rajasekaran and G. A. Vijaylakshmi Pai "*Neural Networks, Fuzzy Logic and Genetic Algorithms Synthesis and Application*", PHI Learning Pvt. Ltd., 2009.

[3] Lotfi A. Zadeh, “Soft computing and fuzzy logic”, Software, *IEEE Software Journal*, Vol. 11, No. 6, pp. 48-56,1994.

[4] Lotfi A. Zadeh, “A theory of approximate reasoning”, *Journal on Machine Intelligence*, Vol. 9, pp. 149-194, 1979.

[5] Lotfi A. Zadeh, “Is there a need for fuzzy logic?”, *Information Sciences*, Vol. 178, No. 13, pp. 2751-2779, 2008.

[6] W. N. Venables, D. M. Smith and the R Core Team “An Introduction to R- Notes on R: A Programming Environment for Data Analysis and Graphics”, Project manual, 2014.

[7] Vignesh Prajapati, “*Big Data Analytics with R and Hadoop*”, PACKT Publishing, 2013.

[8] Munmun De Choudhury, Michael Gamon, Scott Counts and Eric Horvitz, “Predicting Depression via Social Media”, *Microsoft Research, Association for the Advancement of Artificial Intelligence*, 2013.

[9] Kay-Yut Chen, Leslie R. Fine and Bernardo A. Huberman, “Predicting the Future”, *Journal on Information Systems Frontiers*, Vol. 5, No. 1, pp. 47-61, 2003.

[10] Wenbin Zhang and Steven Skiena, “Improving movie gross prediction through news analysis”, *IEEE/WIC/ACM International Joint Conference on Web Intelligent Agent Technology*, pp. 301-304, 2009.

[11] Ramesh Sharda and Dursun Delen, “Predicting box office success of motion pictures with neural networks”, *Expert Systems with Applications*, Vol. 30, pp. 243-254, 2006.

[12] Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak and Andrew Tomkins, “The predictive power of online chatter”, *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 78-87, 2005.

[13] Mahesh Joshi, Dipanjan Das, Kevin Gimpel and Noah A. Smith, “Movie Reviews and Revenues: An Experiment in Text Regression”, *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies*, pp. 293-296, 2010.

[14] Rion Snow, Brendan O’Connor, Daniel Jurafsky and Andrew Y. Ng, “Cheap and Fast - But is it Good? “Evaluating Non-Expert Annotations for Natural Language Tasks”, *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 254-263, 2008.

[15] Fang Wu, Dennis M. Wilkinson and Bernardo A. Huberman, “Feeback Loops of Attention in Peer Production”, *Proceedings of the 2009 International Conference on Computational Science and Engineering*, Vol. 04, pp. 409-415, 2009.

[16] Bo Pang and Lillian Lee, “Opinion Mining and Sentiment Analysis”, *Foundations and Trends in Information Retrieval*, Vol. 2, No. 1-2, pp. 1-135, 2008.

[17] Namrata Godbole, Manjunath Srinivasaiah and Steven Skiena. “Large-Scale Sentiment Analysis for News and Blogs”, *Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[18] G. Mishne and N. Glance, “Predicting movie sales from blogger sentiment”, *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, pp. 155-158, 2006.

[19] Bin Fu, Eugene Fink and Jamie G. Carbonell, “Analysis of Uncertain Data: Tools for Representation and Processing”, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3256-3260, 2008.

[20] David M. Pennock, S. Lawrence, C. L. Giles and Nielsen, “The real power of artificial markets”, *Journal of Science*, Vol. 291, No. 5506, pp. 987-988, 2001.

[21] C. R. Cloninger, D. M. Svrakic and T. R. Przybeck, “Can personality assessment predict future depression?”, *Journal of affective disorders*, Vol. 92, No. 1, pp. 35-44, 2006.

[22] developer.ebay.com