ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse253952020.pdf

https://doi.org/10.30534/ijatcse/2020/253952020

A Comparative Study of Statistical Analysis on Big Mart using Data Mining Techniques

T K Thivakaran[1], Dr M Ramesh[2]

[1]Research Scholar, Annamalai University, Tamil Nadu, India.

[2]Professor, Annamalai University, Tamil Nadu, India.

ABSTRACT

In order to estimate sales revenue that is tangible and achievable, businesses involved in wholesales, manufacturing activities, marketing activities, retailing, logistics and supply chain activities need to use historical transaction data to forecast sales. In order to do this, there are several traditional data mining and statistical techniques that are used to identify trends, make predictive as well as descriptive analysis. The knowledge gained from such analysis is used in making business decisions. The data set in this study has been collected in the year 2013, and has 1559 products across 10 stores in different cities. First we conduct Exploratory Data Analysis to understand the nature of thAfter this, several traditional and novel data mining techniques have been applied on this data set, namely, linear regression, ridge regression, random forest regressor, decision tree regressor, XG Boost regressor and ARIMA. The cross-validation scores of all models are compared and inference as to which attributes and feature are given most weight during prediction of Item Outlet Sales attribute (target attribute) in the data set. Towards the end of the paper, the inferences and results are noted and discussed, hence completing the entire data analysis cycle.

Key words : Data Mining, Machine Learning, Gradient Descent, Gradient Boosting, Auto-regressive Integrated moving average, Time-series data, Sales Prediction.

1. INTRODUCTION

Sales forecasting is the primary importance to retail businesses. Forecasting and predictive analysis plays a very vital role in building and sustaining a vibrant business [1]. In order to develop the right business strategies, the decision-making executives of a corporation should have access to novel knowledge that can be gained from analysis of historical transaction data [2]. A traditional sale forecast generally looks very deeply into the certain situations and conditions that have previously occurred, and gains inferences from it [3]. These inferences could be regarding customer acquisitions, sales predictions, weak and strong products, outlets with better sales and even inadequacies in the product, supply, quality, customer care, etc [4]. These inferences could be considered while making important decisions such as setting a budget, making marketing plans for the next year, or target capital to products and services that sell extremely well and diverting capital from products that sell poorly. Therefore, predictive analysis of the future is done looking at past data. Decisions that are backed by data and novel in-depth knowledge gained from such analysis increases the likelihood of success [5]. Therefore, several papers have found that business that are digitally focused and use data to make data-driven decision are more profitable and last longer, and hence perform better compared to business that don’t.

In this paper, we consider the Big Mart data set. We start our analysis by taking an in-depth look into the nature of the data set and understand the distribution of the data points [6]. We try to look into the correlation between various features with the target attribute [7]. After this, we apply various data mining, machine learning and statistical models on this data set and compare the results of these models using various performance metrics such as Root Mean Square Error (RMSE). Towards the end of this paper, we highlight which method is best suited for similar data set, and give reasons for our selection based on our findings [8].

Hence, through this paper, we aim to make an in-depth comparative study of various data mining, machine learning and statistical models on a single data set, compare results using performance metrics and find the most suitable method for similar jobs [9]. This study compares various data mining techniques such as linear regression, ridge regression, random forest, decision tree, XGBoost and ARIMA. Most studies so far show that XGBoost provides the most accurate predictions of a time-series data set [10]. In this study, we aim to show that Auto-regressive moving average is still a much better choice compared to XGBoost since exploratory data analysis in business conditions have different goals that are not solely dependent on the accuracy of predictions and are dependent of various other factors.

2. EXPERIMENTAL SETUP

Experiment was conducted using different algorithms on Big Mart Dataset which consist of 14204 records and 13 attributes. Experimental Setup consists of two steps:

2.1 Data Preprocessing

In Data Preprocessing the first step was splitting of data into two parts that is train and test, in train the data consist of 8523 records, in test the data consist of 5681 records. Next step was to find the missing value and treat it, according to this step there were missing values in 3 attributes namely Item Weight with 2439 missing values, Outlet Size with 4016 missing values and Item Outlet Sales with 5061 missing values in which numerical column was treated with mean imputation and categorical column with mode imputation. After this step some of the categories in particular was combined into lesser category and after that the data categorical encoded and finally the data was ready for training.

2.2 Model Building

In order to test the model on the given data set, we have used the following tools: Python 3, R Programming Language, Jupyter Notebook for IPython, several open source machine learning package such as numpy, scipy, scikit-learn, xgboost etc. In order to plot the graphs for data visualization, we have used Matplotlib and graphViz for decision tree visualization.

2.2.1. Gradient Boosting

Gradient boosting is an ensemble machine learning technique used for solving classification and regression problems. It produces a predictive model in the form an ensemble of weak prediction models. The weak prediction models are typically decision trees. Like other boosting algorithms, gradient boosting also builds the model in a stage wise fashion. It generalizes the stages by allowing optimization of an arbitrary differentiable loss function. Gradient boosting was used to create the predictive model to predict BOD values.

In order to build the model using gradient boosting, XGBoost was used. XGBoost is an open source software library which provides a gradient boosting framework. It provides a portable, scalable and distributed gradient boosting library. XGBoost can run on a single machine as well distributed processing frameworks (such as Apache Hadoop, Apache Spark and Apache Flink).

2.2.2 Auto-regressive Integrated Moving Average (ARIMA)

When we consider a weak stationary process, such as the sales of a retail outlet, whose unconditional joint probability distribution does not change when shifted in time, auto-regressive moving average provides a very cheap and resourceful description in terms of two polynomials. These two polynomials are as follows:

1. Auto-regression (AR)
2. Moving Average (MA)

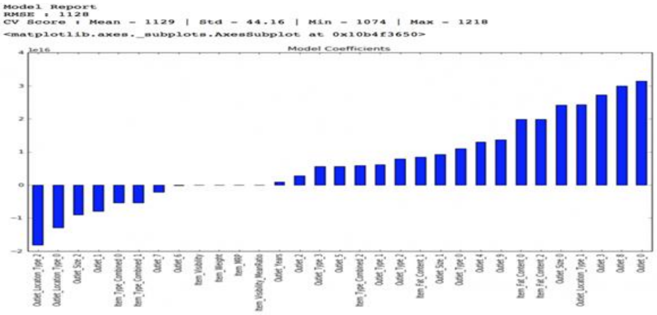
A general form of the ARMA model was first described in a thesis paper titled “Hypothesis testing in time series analysis” by Peter Whittle and published in 1951. In general, when we are given a series of data X, the ARMA model is a tool for not only understanding the nature of this data but also predict the future values in this series. In other words, ARIMA is a model that can predict any given time series as well as explain the behavior of the time series based on its own past values, its own lags and the lagged forecast errors. The AR part regresses the variable on its own past values whereas the MA part models the error terms which may occur at the same time as well as different times in the past, in a linear combination.

3. RESULT ANALYSIS

Result Analysis was done on various model namely, linear regression, ridge regression, random forest, decision tree, XG Boost and ARIMA. And the results for different models are as follow:

3.1 Linear Regression

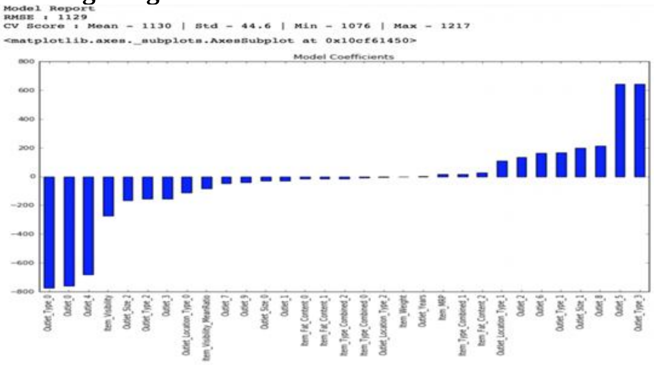
The Linear Regression model shown in figure 1 was trained with normalization True and obtained RMSE of 1128, mean of 1129, standard deviation of 44.16 and with min value of 1074 and maximum value 1218. After training the above bar plot was plotted which shows model coefficients values which indicates which are the attributes were contributing negative and which were contributing positive.



```
Model Report
RMSE : 1128
CV Score : Mean - 1129 | Std - 44.16 | Min - 1074 | Max - 1218
<matplotlib.axes._subplots.AxesSubplot at 0x10b4f3650>
```

Figure 1: Linear Regression Model Coefficients

3.2 Ridge Regression



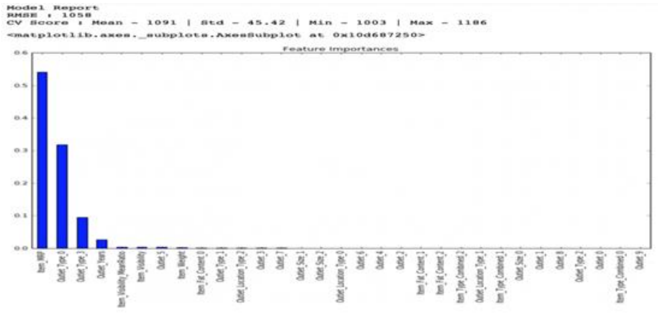
```
Model Report
RMSE : 1129
CV Score : Mean - 1130 | Std - 44.6 | Min - 1076 | Max - 1217
<matplotlib.axes._subplots.AxesSubplot at 0x10cf61450>
```

Figure 2: Ridge Regression Model Coefficients

The Ridge Regression model shown in figure 2 was trained with normalization True, alpha 0.05 and obtained RMSE of 1129, mean of 1130, standard deviation of 44.6 and with min value of 1076 and maximum value 1217. After training the above bar plot was plotted which shows model coefficients values which indicates which are the attributes were contributing negative and which were contributing positive.

3.3 Decision Tree Regressor

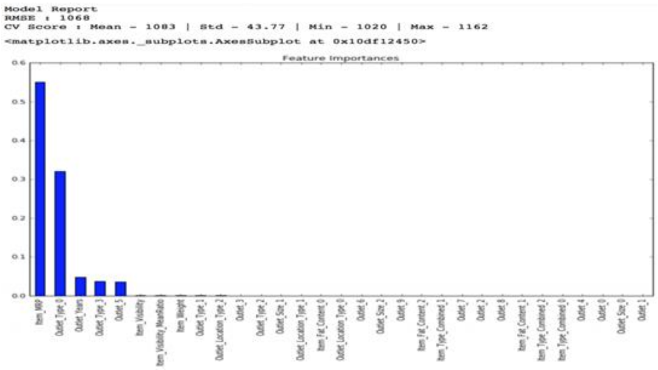
The Decision Tree Regressor model shown in figure 3 was trained with maximum depth of 3 and minimum sample leaf of 100 and obtained RMSE of 1058, mean of 1091, standard deviation of 45.42 and with min value of 1003 and max value of 1186. After training the above bar plot was plotted which shows import features contributing to the target variable.



```
Model Report
RMSE : 1058
CV Score : Mean - 1091 | Std - 45.42 | Min - 1003 | Max - 1186
<matplotlib.axes._subplots.AxesSubplot at 0x10d687250>
```

Figure 3: Decision Tree Regressor Feature Importance

3.4 Random Forest Regressor



```
Model Report
RMSE : 1068
CV Score : Mean - 1083 | Std - 43.77 | Min - 1020 | Max - 1162
<matplotlib.axes._subplots.AxesSubplot at 0x10df12450>
```

Figure 4: Random Forest Regressor Feature Importance

The Random Forest Regressor model shown in figure 4 was trained with estimator 400, max depth of 6 and minimum sample leaf of 100 and obtained RMSE of 1068, mean of 1083, standard deviation of 43.77 and with min value of 1020 and max value of 1162. After training the above bar plot was plotted which shows import features contributing to the target variable. The results are shown in Table 1.

Table 1: RMSE Score for different Model

Model Name	**RMSE Scores**
Linear Regression (scipy)	1127
Decision Tree (scikit-learn)	1058
Ridge Regression	1129
XGBoost	1052
ARIMA	1056

4. CONCLUSION

First We found out that Item_MRP is highly correlated with Item_Outlet_Sales. For this we have used Pearson's Correlation Coefficient available in Python's scipy library. In the above equation, Item_MRP is x and Item_Outlet_Sales is y. Similarly, each attribute was taken as x and correlation coefficient with respect to the target y was calculated. Second we found out that the data is sparse, and has no real trend. Hence, for time series forecasting in this particular data set, we can use both XGBoost and ARIMA. Here, for more accurate predictions, XGBoost showed better results than ARIMA. XGBoost showed better results on this dataset than ARIMA because of 3 reasons:

1. XGBoost has the exclusive feature of being sparse aware. XGBoost library in python automatically handles all missing values.
2. XGBoost supports parellelism in tree construction.
3. XGBoost also has continued training. Even when new dummy data is added, the model is able to re-train in boost existing model.

Even after all this reason, in this paper, we argue that for time-series forecasting, especially sales forecasting, ARIMA is still the better choice compared to XGBoost, because Exploratory Data Analysis have different goals. Even though XGBoost may give us better prediction of target variable, ARIMA will give us better understanding of how each variable is interacting with each other and also how each variable is interacting with the target variable and ARIMA is built specifically for time-series data analysis.

REFERENCES

1. Behera, Gopal & Nain, Neeta, “A Comparative Study of Big Mart Sales Prediction”, Computer Vision and Image Processing, Communications in Computer and Information Science, vol. 1147, (2019).
2. S. Beheshti-Kashi, H. R. Karimi, K. D. Thoben, M. Lutjen, M. Teucke, “A survey on retail sales forecasting and prediction in fashion markets”, Systems Science & Control Engineering 3(1), (2015), pp. 154–161.
3. P. Das, S. Chaudhury, “Prediction of retail sales of footwear using feedforward and recurrent neural networks”, Neural Computing and Applications 16(4-5), (2007), pp. 491–502.
4. P. M. Domingos, “A few useful things to know about machine learning”. Communication of the ACM, 55(10), (2012), pp. 78–87.
5. W. Y. Loh, “Classification and regression trees”, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1), (2011), pp. 14–23.
6. I. Bose, R. K. Mahapatra, “Business data mining-a machine learning perspective”, Inf. Manage, 39(3), (2001), pp. 211–225.

7. C. W. Chu, G.P Zhang, “A comparative study of linear and nonlinear models for aggregate retail sales forecasting”, Int. J. Prod. Econ, 86(3), (2003), pp. 217–231.
8. M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin, “Combing content-based and collaborative filters in an online newspaper” (1999).
9. K. Punam, R. Pamula, P. K. Jain, “A two-level statistical model for big mart sales prediction”, In International Conference on Computing, Power and Communication Technologies (GUCON), IEEE, (2018), pp. 617–620.
10. M. Xia, W. K. Wong, “A seasonal discrete grey forecasting model for fashion retailing”, Knowl.-Based Syst, (2014), 57, pp. 119–126.