

Predicting E-commerce Sales & Inventory Management using Machine Learning

Divyansh Agnani^{1*}, Abhishek Bavkar¹, Samarth Salgar¹, and Sheetal Ahir¹

¹ Ramrao Adik Institute of Technology, Department of Computer Science, Mumbai, India

Abstract. Due to prevalent transition from visiting physical stores to online shopping, predicting customer behaviour in the e-commerce market is gaining increasing importance. Over the past few years, e-commerce marketplaces such as Flipkart and Amazon have observed a manifold increase in both sales and market share of retail products sold. During this period, many traditional retailers and wholesalers have set up their e-commerce portfolios to stay relevant in the marketplace. However, due to lack of technology penetration in the Medium, Small, & Micro Enterprises (MSMEs) sector, the technological tools required by them are largely absent. One key issue for small businesses in e-commerce has been to forecast how many products their business will sell, thereby introducing another big issue of predicting how much inventory of a product they need to hold, to match up with demand. This is partly due to the massive scale and opportunities opening for them by participating in an open, large marketplace such as e-commerce. The COVID-19 pandemic has also drastically changed consumer shopping trends and behaviours: consumers have changed their preference to buying online. Thus, Indian small businesses need to have the ability to forecast demand more accurately than ever before.

1 Introduction

Traditional (brick and mortar) small businesses in India have usually differentiated themselves against big retailers by offering products at good value (in terms of excess benefit over cost). This has been possible by a largely two factors: 1) knowing their customers well, helping them price their products competitively, and 2) successfully attaining customer's attention. However, this gap has closed with increasing relevance and participation of customers in the e-commerce marketplace. E-commerce websites such as Amazon and Flipkart have witnessed rapid growth in terms of sales and market share in the past few years. On the other hand, small businesses, termed as Medium, Small, and Micro Enterprises (MSMEs) have been unable to keep up with big retailers partly due to lack of technological tools available to them. Forecasting sales is a challenging, but important task in revitalizing the growth strategy for MSMEs by helping them make informed strategic decisions. It helps the businesses manage inventories, cash flows, and resources by optimizing their supply chain. The benefit of forecasting sales is dependent on how accurately a model predicts sales. If a model is inaccurate and is unable to help a business effectively maintain track of their inventories, it can lead to over-stocking of products, which results in excess overhang, or understocking, which results in potential notional losses or loss of opportunity.

Traditionally, economic forecasts have been used to help businesses predict sales. However, history is a

study of surprising events. For years, this reliance on economic forecasts have rendered small businesses inefficient in predicting demand and managing their inventories. This is because, economic forecasts are volatile in nature, changing with ever-coming stream of new information and numbers, giving little time for small businesses to streamline and adapt. Economic forecasts in India are also dated in their approach towards changing consumer trends and growing higher preference for online shopping.

This section of the report is followed by literature review in the subsequent section. Section 3 consists of methodology used to conduct our research, divided into two parts, first containing existing methodology, and second containing proposed changes to existing systems. The methodology also contains an evaluation and comparison of models used in our research. Section 4 involves concluding the paper, with a brief about our findings. The paper ends with references used in the research.

2 Literature Review

E-commerce market in India has dramatically grown in the past few years but is still lower compared to the growth of the market in developed nations, such as the United States (US) and China. Kumar, N. et al. in their research, titled 'Challenges and Opportunities of E-commerce in India' [9] noted that this is due to 1) poor

* Corresponding author: div.agn.rtl7@rait.ac.in

telecom infrastructure, 2) poor academic syllabus about e-commerce, and notably for our paper, 3) lack of positive attitude towards technology [9].

The authors recommend policy-driven changes, updated legal frameworks, and infrastructure development for e-commerce. Regarding technological infrastructure needed, Galhotra, B. et al. note in their research to survey and analyse digital platforms preference and changing consumer trends during the COVID-19 pandemic, that machine learning can be applied to predict the changing consumer preferences and trends in the e-commerce space [10].

Traditional sales forecasting methods have used time series analysis techniques, such as autoregressive models, integrated models, and moving average models [1], that forecast sales based on linear functions of historical sales data. Pavlyshenko, B. et al. have compared regression approach against the traditional time series forecasting approach for sales forecasting in their research [4].

Zhao, K. et al. compared autoregressive integrating moving average (ARIMA), a more recent model, with Deep Neural Network (DNN), Transfer Learning (TL), Convolutional Neural Network (CNN) and Weight Decay (WD) models in their research [1]. The author also assesses which model efficiently eliminates the need for manual feature engineering by extracting effective features [1].

Bohanec, M. et al analyse the use of black-box prediction models in a real-world case scenario of B2B sales forecasting. The authors identify the appropriate model based on business understanding by using intelligence and data-driven models [6].

Yu, J. et al. proposes and compares various forecasting approaches namely Winters' Exponential Smoothing, Time Series Decomposition, and ARIMA model [5]. Forecasting sales in e-commerce faces certain challenges regarding non-linear, irregular, and lumpy sales data. Yu, J. et al. in their work on forecasting sales for Amazon also describe necessary data to carry out forecasting as a major obstacle impeding the implementation of the model [5].

Bandara, K. et al. in their research [2] trains a Long Short-Term Memory network (LSTM) to predict sales given non-linear relationships in e-commerce. This inability to record effective data is also reiterated by Holmberg, M. et al. in which the authors recognize restaurants not having required education about forecasting sales and thus not maintaining sales data. The authors of this research have evaluated Extreme Gradient Boosted (EGB) and Long Short-Term Memory Boosted (LSTM-Boosted) models [7].

Another issue that arises from inadequate sales data is explored by Ferreira, KJ. et al. in their research on predicting demand for first exposure items (items that

have never been sold before) [8]. The authors have additionally worked on a price optimization model to help businesses maximize revenue on such products.

E-commerce consumer behaviour studies also showcase a cross-product relationship between different products. Li M. et al. identify different cross-product demand or sales pattern and evaluates their correlation with a pre-processing framework by comparing prediction errors of the NARNNN and ARIMA models [3].

3 Methodology

3.1 Existent Methodology

3.1.1 Data Pre-processing

Data pre-processing is a procedure that describes how the chosen data will be cleaned of any noise and outliers. This entails cleaning out data that has a lot of extra substance but is not relevant to our use case. For example, if the dataset contains missing values for sales and price values, we must manage them appropriately by replacing the missing value with the average value or using mean or median imputation to ensure that the data remains consistent. In this project, the following pre-processing mechanisms have been used.

3.1.1.1 Data Transformation

Data transformation is the process of changing data from one format to another to meet specific requirements. This procedure is also known as the ETL procedure, which stands for Extract, Transform, and Load. As the volume of data has grown exponentially, transformation has become a critical responsibility. As a result, reliable data transformation will allow users to concentrate on data that meets their business requirements. The same will be true for this project; all the data will be robustly transformed, with only the most significant data being integrated into a new data format.

3.1.1.2 Data Imputation

Some of the columns in our dataset have missing values where a default value of 0 should have been placed instead of NaN/undefined/null values. In such cases, we have filled the missing values with 0. In other, more notable cases, columns are missing values which are non-binary in nature and can't be misplaced with arbitrary values. Here, we have taken an approach to fill the missing values with the mode - or the most frequent values present in each column. It is observed that this approach does not consider the correlation between various features and can thus introduce bias in the data by undesirably assigning more values to a specific

category. Attention and caution have been undertaken to check for such discrepancies.

3.1.1.3 Feature Extraction

Feature extraction begins with a set of measured data and creates derived values (features) that are aimed to be insightful and non-redundant, facilitating the learning and heuristic steps resulting in better inference and usage by a machine learning model. For example, a feature that contains value count of just one, that is, it contains a unique value for all items in our data frame, it fails to be relevant to the model. Hence, it should be dropped. On the other hand, the unique value feature disposal needs to be dealt with caution. Sometimes, a feature with only two possible value counts, can be flagged as one with a unique value count, if one of the two possible values is filled NaN/undefined/null value. In other cases, features with values that frequently do not deviate from a 'popular' value again deem no relevance. In these cases, such features can be deemed to not be correlated to our target variable. Thus, such columns can be dropped. In some cases, one feature describes the listings in a data frame in a manner that makes other features redundant. This is often in the case of unique identifiers.

3.1.1.4 Feature Selection

Another important and widely used technique in data processing is feature selection, which is used to select appropriate features from noisy data. This method improves the speed of any data processing algorithm, which improves prediction accuracy and reduces the variance in predictions. While seemingly similar, the main distinction between feature selection and extraction is that feature selection retains a subset of the original features while feature extraction generates entirely new ones. Feature selection approaches can be employed if the original features must be preserved, as opposed to feature extraction techniques, which extract useful information from data in order to create a new feature subspace. When model explain ability is a top priority, feature selection strategies are applied.

3.1.1.5 Detecting and removing outliers

Outliers are data points that differ dramatically from the rest of the dataset's observations. It can happen as a result of measurement variability, misinterpretation when filling in data points, measurement errors (fault in measurement instrument), experimental errors, intentional dummy errors to test detection methods, sampling errors, or simply and often visibly, natural outliers. In our project, we have plotted column wise distribution of values in features to detect outliers visually. This plotted distribution can be seen in Figure 3.1 below. Other than that, we have used Grubbs Test function to detect outliers, which is a statistical tool used to detect outliers in a dataset assumed to come from a normally distributed population. This assumes normality. The data value with the highest absolute value is considered first by the algorithm. The

considered value is detected as an outlier and excluded from further analysis if the null hypothesis that it is not an outlier is rejected. The value with the second highest absolute value is then considered, and its quality is analysed using the Grubbs test once.

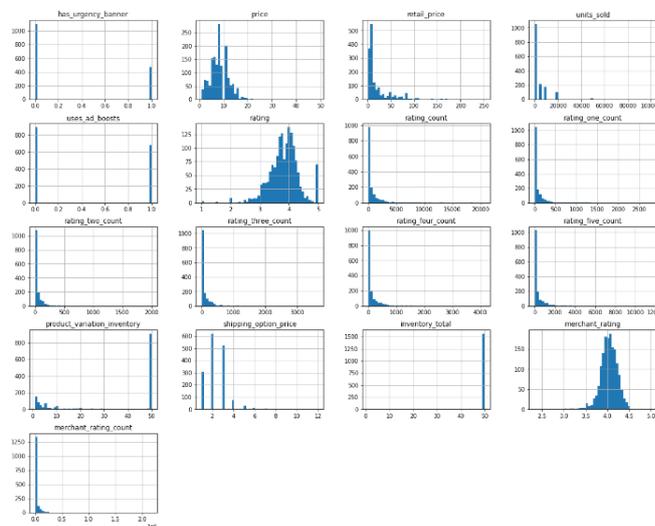


Fig. 3.1 Column wise Distribution of Values

3.1.2 Model Training

Model Training is the process of determining which algorithms will be used to solve a problem. We will be using four distinct regression algorithms for this study, which are Random Forest Regression, Extra Trees algorithm, Gradient Boosting Regression, and AdaBoost Regression.

Regression analysis is a statistical method for modelling the relationship between one or more independent variables and a dependent (target) variable. Regression analysis allows us to see how the value of the dependent variable changes in relation to an independent variable when the other independent variables are held constant.

Random Forest Regression: A random forest is a meta estimator that uses averaging to improve predictive accuracy and control over-fitting by fitting several classifying decision trees on various sub-samples of the dataset. As a result, it can be called a "Forest" of trees, hence the name "Random Forest." The name 'Random' comes from the fact that this algorithm is made up of a forest of 'Randomly constructed Decision Trees.' In other words, the average result of a series of decision trees is called a Random Forest regression. A decision tree is similar to a flow chart in that it asks a series of questions and then makes a prediction based on the answers.

Gradient Boosting: Gradient Boosted Decision Trees constructs an additive model in a forward phased manner, allowing for the optimization of any differentiable loss function. A regression tree is fitted on the negative gradient of the specified loss function at

each level. In other words, when a decision tree is the weak learner, the resulting algorithm is called gradient boosted tree. A weak hypothesis or weak learner is defined as one whose performance is at least slightly better than random chance.

Extra Trees Regression: Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees. The Extra Trees approach uses the training dataset to generate a multitude of unpruned decision trees. In the case of regression, predictions are formed by averaging the prediction of the decision trees, whereas in the case of classification, majority voting is used.

Adaptive Boosting: AdaBoost for short, it was the first iteration of boosting algorithms that saw significant success. AdaBoost works by weighing the observations, giving more weight to instances that are difficult to predict and less to those that are already well-predicted. New weak learners are introduced one at a time, with the goal of concentrating their training on the increasingly challenging patterns. This means that difficult-to-predict samples are given increasingly greater weights until the algorithm finds a model that correctly classifies them.

3.1.2.1 *Splitting into training and testing subsets*

The creation of different samples for training and testing helps us evaluate model performance. Overfitting is a common problem while training a model. When a model performs exceptionally well on the data used to train it, but fails to generalise well to new, previously unseen data points, this phenomenon happens. This can happen for a variety of reasons, including data noise or the model learning to anticipate certain inputs rather than the predictive factors that could help it make more accurate predictions. In general, the more complicated a model is, the more likely it is to be overfitted. Underfitting, on the other hand, occurs when the model performs poorly even with the data that was used to train it. Underfitting happens most of the time when the model isn't appropriate for the problem you're trying to solve. This usually signifies that the model isn't as complicated as it must be in order to learn the parameters that have been shown to be predictive. The most typical strategy for identifying these difficulties is to create separate data samples for training and testing the model. In this way, we can use the training set to train our model and then use the testing set to assess if the model can generalise well to new, unseen data. We have split the data frame in the ratio of 70:30 for training and testing subsets respectively

3.1.2.2 *Training or Fitting the Model*

A training model is a dataset used to train a machine learning algorithm. It is made up of sample output data as well as the equivalent sets of input data that have an impact on the outcome. The training data model is used to process the input data via the algorithm in order to

compare the processed output to the sample output. The model is modified based on the results of this association. "Model fitting" is the term for this iterative procedure. The precision of the model is dependent on the accuracy of the training or validation dataset. In machine language, model training is the act of supplying data to an ML algorithm to help it find and learn suitable values for all the variables involved. There are various types of machine learning models, with supervised and unsupervised learning being the most common. When the training data comprises both the input and output values, supervised learning is performed. A supervisory signal is a set of data that contains both the inputs and the intended outcome. When the inputs are fed into the model, the training is based on the deviation of the processed output from the documented result. The process of unsupervised learning entails identifying patterns in data. After then, further data is used to fit patterns or clusters. This is basically an iterative procedure that improves accuracy by comparing the observed patterns or clusters to the expected patterns or clusters.

3.2 Proposed Methodology

We have employed hyperparameter tuning to enhance the performance and computational time of the models used. Parameters are inputs that describe the individual weights between nodes, computed by the algorithm used for training the model. Hyperparameter, on the other hand, are parameters that determine how the model learns, in contrast to the parameters learned by the model.

Hyperparameters are derived by setting different values for parameters, training models using combinations of these values, and evaluating which combination works best. A domain is specified from which these values are taken.

Specifically, we have used Randomised Search hyperparameter tuning for Random Forest and Extra trees models, which randomly transverses parameters from a specified domain, and chooses the optimal combination rand. The benefits of choosing randomised search over other alternatives is the computational times saved, due to its efficiency.

For boosting models, we've employed Grid Search hyperparameter tuning, which creates a grid of possible parameters from a domain and tries every combination to get optimal values. Unlike randomized search hyperparameter tuning, grid search algorithm doesn't transverse through the domain randomly.

3.3 Evaluation

The researchers have conducted an overall evaluation at the conclusion of this study to determine the algorithm which has the highest prediction accuracy, as well as evaluated the effects of hyper-parameter tuning in each of the models used, to arrive at the best

algorithm as per our motive. For each algorithm under consideration, the error percentage and prediction accuracy has been assessed. The evaluation metrics used to compare the various algorithms are based on the accuracy of prediction, and error rates. The mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) of the models used is given in Table 3.1.

Table 3.1 Error rates

Model Used	MAE	MSE	RMSE
Random Forest	0.7434	1.2921	1.1367
Random Forest (Hyper-parameter tuned)	0.7428	1.2554	1.1204
Extra Trees	0.7286	1.2605	1.1227
Extra Trees (Hyper-parameter tuned)	0.7183	1.2275	1.1079
Gradient boosting	0.8146	1.3222	1.1499
Gradient boosting (Hyper-parameter tuned)	0.7730	1.2776	1.1303
Adaptive Boosting	0.9761	1.7546	1.3246
Adaptive Boosting (Hyper-parameter tuned)	0.9627	1.6380	1.2798

The R-squared metric, used to determine the accuracy of the models, are given in Table 3.2.

Table 3.2 Accuracy metrics

Model Used	R-squared
Random Forest	0.9142
Random Forest (Hyper-parameter tuned)	0.9167
Extra Trees	0.9166
Extra Trees (Hyper-parameter tuned)	0.9185
Gradient Boosting	0.9122
Gradient Boosting (Hyper-parameter tuned)	0.9152
Adaptive Boosting	0.8836
Adaptive Boosting (Hyper-parameter tuned)	0.8913

4 Conclusion

We have implemented and evaluated different models to solve one problem of these small enterprises: forecasting sales. This helps them take more strategic decisions when it comes to investments, supply chain management, and inventory management. While working on this project of using machine learning to forecast ecommerce sales, it was observed that while there are many different methods for forecasting ecommerce platform sales, the researcher was able to focus on four regression algorithms that are commonly used when forecasting future sales. All the selected machine learning models were able to be built and tested by the researcher. Furthermore, the researcher were able to hyperparameter tune the models and evaluate the subsequent effects on all four algorithms for comparison. The best algorithm is chosen as the model

with the best prediction range, where the predicted value and the actual value are almost identical.

Through surveying existing literature and work, sales predicting is found to be a regression problem, instead of a time-series forecasting problem as it has traditionally been studied. Thus, we have chosen to implement and compare various regression models on algorithms namely Random Forest Regression, Extra Trees algorithm, Gradient Boosting Regression, and AdaBoost Regression. Next, we have tested the mentioned regression models with hyperparameter tuning. Finally, we evaluate and compare the various models and the effects of hyperparameter tuning through R- squared (R2) metric and various error rates.

References

- [1] 1. J. Yu, X. Le et al. 'Sales Forecast for Amazon Sales Based on Different Statistics Methodologies'(2016)
- [2] 2. K.J. Ferreira, B.H.A. Lee, D. Simchi-Levi et al. 'Analytics for an Online Retailer: Demand Forecasting and Price Optimization' (2016)
- [3] 3. M. Bohanec, M.K. Borstnar, M. Robnik-Sikonja et al. 'Explaining machine learning models in sales prediction' (2017)
- [4] 4. Zhao, K. and Wang, C. (2017) 'Sales Forecast in E-commerce using Convolutional Neural Network', (August 2017).
- [5] 5. Li, M., Ji, S. and Liu, G. (2018) 'Forecasting of Chinese E-commerce Sales: An Empirical Comparison of ARIMA, Nonlinear Autoregressive Neural Network, and a Combined ARIMA-NARNN Model'
- [6] 6. M. Holmberg, P. Hallden et al. 'Machine Learning for Restaurant Sales Forecast' (2018)
- [7] 7. Bandara, K. et al. (2019) 'Sales Demand Forecast in E-commerce using a Long Short- Term Memory Neural Network Methodology'
- [8] 8. N. Kumar, A. Jhensanam, Md A. Hossin, B. Chanthamith et al. 'Challenges and Opportunities of E-commerce in India' (2018)
- [9] 9. B. Pavlyshenko et al. 'Machine-Learning Models for Sales Time Series Forecasting' (2019)
- [10] Bhavna Galhotra and Ayushi Dewan, 'Impact of COVID-19 on digital platforms and change in E-commerce shopping trends' (2021)