IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Sales Forecasting Using Xgboost

M.Krishna Satya Varma
Department of Information Technology
S.R.K.R. Engineering College(A)
SRKR Marg, Bhimavaram.

N.Sai Durga Manikanta
Department of Information Technology
S.R.K.R. Engineering College(A)
SRKR Marg, Bhimavaram.

M.Hemanth
Department of Information Technology
S.R.K.R. Engineering College(A)
SRKR Marg, Bhimavaram.

M.Vinay
Department of Information Technology
S.R.K.R. Engineering College(A)
SRKR Marg, Bhimavaram.

K.Bharath Sri Sai Pradeep
Department of Information Technology
S.R.K.R. Engineering College(A)
SRKR Marg, Bhimavaram.

***Abstract*—A retailing market's ability to make informed business decisions and estimate long-term and non-permanent performance depends on accurate income forecasts. A retailer can estimate its anticipated future revenues for income earned over a specific period of time with the aid of sales forecasting. As a result, time is crucial in sales forecasting.**

With the aim of identifying patterns, cycles, and seasonal variances, the time sequence assessment makes the evaluation of observations as a series of statistics at specific intervals over a period of time. This will be a helpful resource in the predicting of future events. In this study, we used XGBoost to create a machine learning model to forecast the organization's future sales and revenue.

Keywords—Sales forecasting, XGBoost, Machine Learning, Sales Prediction

I. INTRODUCTION

The ability to predict future sales based on current month's sales is known as sales forecasting. Future sales, for instance, can be impacted by the introduction of new products, the availability of promotions, seasonality, and other variables that make it very challenging to forecast using historical habits.

Our system is meant to employ XGBoost, a particular version of a gradient boosting machine that uses more accurate approximations to identify the ideal model, in order to estimate sales in the event of product releases or market downturn. By enhancing the systems and altering the algorithms, it improves the gradient boosting machine architecture. Demand forecasting becomes incredibly precise and successful as a result.

II. LITERATURE SURVEY (RELATED WORK)

Sales forecasting is an important task for any business to make informed decisions regarding inventory, production, and financial planning. Literature survey for sales forecasting can help in understanding the existing methods and techniques used for sales forecasting. Here are some of the key research papers and articles on sales forecasting.

"A review of sales forecasting methods for short-life cycle products" by Kuo et al. (2019): This paper presents a comprehensive review of various sales forecasting methods for short-life cycle products. The authors review methods such as time series, regression, artificial neural networks, and support vector regression. "A comparative study of forecasting methods for retail sales" by Fildes and Petropoulos (2015): This article compares the accuracy of various forecasting methods, including judgmental forecasting, time series, and machine learning, for retail sales forecasting.

"Sales forecasting in the apparel industry using machine learning algorithms" by Kim and Hwang (2020): This study uses machine learning algorithms such as random forest and gradient boosting to forecast sales in the apparel industry. The authors found that machine learning methods outperformed traditional time series models.

"An evaluation of the forecasting performance of econometric models of US automobile sales" by Chatfield and Yar (2017): This paper evaluates the forecasting performance of various econometric models for US automobile sales. The authors compare models such as ARIMA, VAR, and VECM.

"Sales forecasting with artificial neural networks: A review" by Zhang and Qi (2017): This article provides an overview of sales forecasting using artificial neural networks. The authors discuss the advantages and limitations of using neural networks for sales forecasting.

"Forecasting sales in the pharmaceutical industry using time series models" by Chen et al. (2021): This study evaluates the performance of time series models such as ARIMA and SARIMA for sales forecasting in the pharmaceutical industry. The authors found that these models were effective in predicting sales.

These are just a few examples of the literature available on sales forecasting. There are many more studies and articles that discuss different aspects of sales forecasting, including data preprocessing, feature selection, and model selection.

III. SYSTEM IMPLEMENTATION (METHODOLOGY)

Sales forecasting using XGBoost (Extreme Gradient Boosting) can be performed using the following methodology:

1.Data Preprocessing: The first step is to collect historical sales data and prepare it for modeling. This includes removing any outliers, filling in missing values, and scaling or normalizing the data.

2.Feature Selection: The next step is to select the relevant features or variables that will be used to train the XGBoost model. These could include factors such as past sales data, marketing campaigns, seasonality, and economic indicators.

3.Model Building: The XGBoost model can be trained using the selected features and historical sales data. The model is optimized by tuning hyperparameters such as learning rate, maximum depth, and number of estimators.

4.Prediction: The created XGBoost model is subjected to various to numerous datasets to achieve maximum accuracy. The developed model is then used to predict the future sales of an organization. The model needs to be continuously evaluated in order to increase its accuracy in predicting the sales.

In conclusion, sales forecasting using XGBoost involves data preparation, feature selection, model training, model evaluation, model deployment, and continuous monitoring. This methodology can be customized based on the specific business needs and requirements.

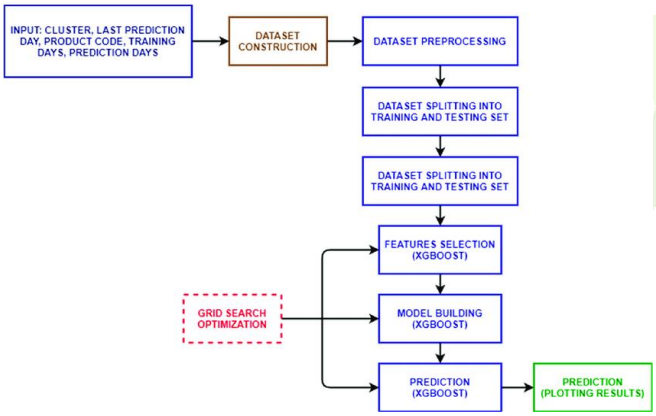

Fig-3.1: System Architecture

1. Data Preprocessing

Step-1: Cleaning missing values and Removing Outliers

The Dataset should be thoroughly checked to get all the missing fields, these missing fields can be filled with the Mean of the values of the column. The Obvious outliers/noisy data values can be removed by defining a threshold value.

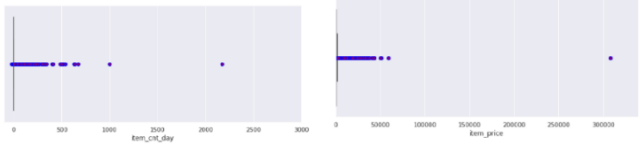

Fig-3.1.1: Removing Outliers

We defined Threshold as:

- Remove all data instances that sold more than 1500 items in a single day using item cnt day.
- Remove the item with a price higher than 150,000 using item price.

Step-2: Removing the negatively valued Data instances

All occurrences of data where the item price or item_cnt_day values are negative should be removed. These statistics could be related to the refund. So We choose to exclude those data points for the sake of simplicity.

Step 3 : Combining duplicates into a single feature value

Many of the stores resemble one another. This can be the result of store relocation or reopenings. After pre-processing, the shop names "!кутск рдоникиде, 56 ран" and "кутск рдоникиде, 56" are considered as the same shop.

Step-4: Splitting the whole Dataset into Training and Testing Data Sets

Currently, the cleaned dataset is separated into several groupings based on the user's interests. Two sets of data—a training set and a testing set—are created here. 90% of the dataset's records are found in the training set. The remaining records in the dataset are in the Testing dataset.

2. Feature Selection

- By averaging all "item cnt month" values for each shop and each item subcategory combination over the course of a month, "shopSubcategory_avg_ item_cnt_" is calculated.
- "shopCity_avg_item_cnt," which is calculated by average each city's "item cnt month" over the course of a month.
- "shop_City_Item_avg_item_cnt_" by averaging each city's and each item's "item cnt month" totals over the course of a month.
- "shop_avg_item_cnt_" is the sum of all "item cnt month" values for each shop over the course of a month.
- "shopItem_avg_item_cnt_," which is calculated by averaging all "item_cnt month" values for all item and shop combinations over the course of a month.
- To calculate "delta price," add side features with the names "globalAvg _tem price" and "shop_avg _tem_price." This "delta price" is meant to show how the average item price for the current month compares to the average item price worldwide.
- Build the "globalAvg shop revenue" and "shop avg revenue" side features to calculate "delta revenue."

3.Model Building

We decided to create a machine learning model using XGBoost Algorithm.

XGBoost:

XGBoost is an ensemble learning method that combines multiple weak learners to form a strong learner. Decision trees, which are binary trees that recursively divide the data into smaller subsets based on the values of the input characteristics, are the weak learners in XGBoost. By iteratively adding decision trees to the ensemble, the XGBoost technique trains decision trees using a gradient boosting framework, which entails minimizing a loss function. The loss function calculates the discrepancy between the target variable's actual values and its anticipated values. Decision trees are iteratively added to the ensemble as part of the gradient boosting technique, and each new tree corrects the flaws of the prior trees. The method creates a new decision tree at each iteration and fits it to the residuals of the prior trees. The disparities between the predicted values and the actual values are known as the residuals.

A regularized objective function is used by the XGBoost technique to reduce overfitting and boost the model's generalization capabilities. The objective function is made up

of two components: a regularization term that penalizes model complexity and a loss function that assesses the discrepancy between anticipated and actual values. The XGBoost algorithm uses a gradient descent optimization method to minimize the objective function. At each iteration, the algorithm calculates the gradients of the objective function with respect to the model parameters, and updates the parameters in the direction of the negative gradient. In order to use XGboost for evaluating the following Regularization parameters has to be followed:

1.Gamma: The more number of splits in decision trees are created with increase in values of Gamma.

2.alpha: L1 regularization on leaf weights. If alpha increases the regularization increases.

3.lambda: L2 regularization on leaf weights. It helps leaf weights to reduce smoothly to 0, rather than applying strong constraints on leaf weights like L1.

Steps to build XGBoost Tree:

1. Calculating the Similarity Score:

$$Similarity\ Score = \frac{(sum\ of\ residuals)^2}{Number\ of\ residuals + lambda} \quad (1)$$

2.Calculating Gain:

$$Gain = Left\ tree\ (similarity\ score) + Right(similarity\ score) - Root\ (similarity\ score) \quad (2)$$

3.The tree has to be pruned based on difference between Gain and Gamma.

$$Gain - gamma \quad (3)$$

4.Calculate outputs after pruning the tree.

$$Outputvalue = \frac{sum\ of\ residuals}{Number\ of\ residuals + lambda} \quad (4)$$

The regression is carried out using the output equation.

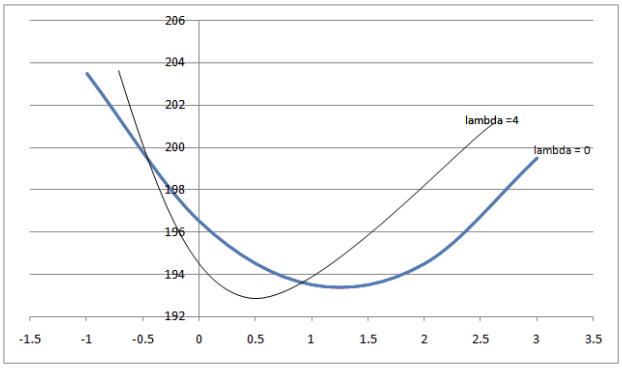

Fig-3.3.1: Example for outputs calculated using lambda=4 and lambda=0

To further improve the performance of the XGBoost algorithm, it includes several advanced features such as:

1.Tree pruning: The algorithm prunes the decision trees to prevent overfitting and improve the generalization performance of the model.

2.Weighted quantile sketch: The algorithm uses a weighted quantile sketch to approximate the distribution of the input features, which improves the efficiency of the tree building process.

3.Column subsampling: The algorithm randomly samples a subset of the input features at each iteration, which reduces overfitting and improves the generalization performance of the model.

4.Row subsampling: The algorithm randomly samples a subset of the training samples at each iteration, which reduces overfitting and improves the generalization performance of the model.

Here are the steps for training a sales forecasting model using XGBoost:

1.Load the data: Load the preprocessed data into memory using a data loading library like pandas. Ensure that the data is split into training and testing datasets.

2.Define the target variable: In this case, the target variable is the sales data.

3.Define the features: Identify the features that may influence sales, such as time, seasonality, economic indicators, and marketing campaigns. Ensure that the features are numerical and do not contain any missing values.

4.Create a DMatrix: Convert the training and testing datasets into a DMatrix, which is a data structure that XGBoost uses for training and prediction.

5.Set the hyperparameters: Set the hyperparameters of the XGBoost algorithm. The learning rate, the maximum depth of the trees, the number of trees, and the regularization parameters are some of the hyperparameters that can be adjusted.

6.Train the model: Train the XGBoost model using the training dataset. Use the xgb.train() function to train the model. Pass the hyperparameters and the DMatrix as arguments to the function.

7.Evaluate the model: Evaluate the performance of the model using the testing dataset. Use evaluation metrics such as MAPE, MAE, and RMSE to compare the performance of the XGBoost model with other models.

8.Tune the hyperparameters: Fine-tune the hyperparameters of the XGBoost algorithm to improve the performance of the model. Find the best hyperparameters using methods like grid search, random search, and Bayesian optimization.

9.Generate sales forecasts: Once the model is trained and optimized, use it to generate sales forecasts for future periods. Use the xgb.predict() function to generate forecasts based on new data.

10.Save the model: Save the trained model to disk so that it can be used for future predictions.

IV. EXPERIMENTS & RESULTS

The performance on the training set and validation set by varying values of Alpha in XGBRegressor (in RMSE)

Alpha=0.1 gives the best score

Using the value of the alpha the model is created. The model is subjected to data sets to train the model. After training the model the, the testing is completed and the results are stored in a .csv file. Based on the outputs, the features are selected to get maximum efficiency and accuracy of the model. Here is the importance of the features in the dataset based on the F score of the test.

	MSE(Train)	MSE (Validation)
alpha = 0.5	0.79921	0.89105
alpha = 0.3	0.79943	0.89729
alpha = 0.2	0.79550	0.89782
alpha = 0.1	**0.77927**	**0.88560**
alpha = 0.05	0.79821	0.89816

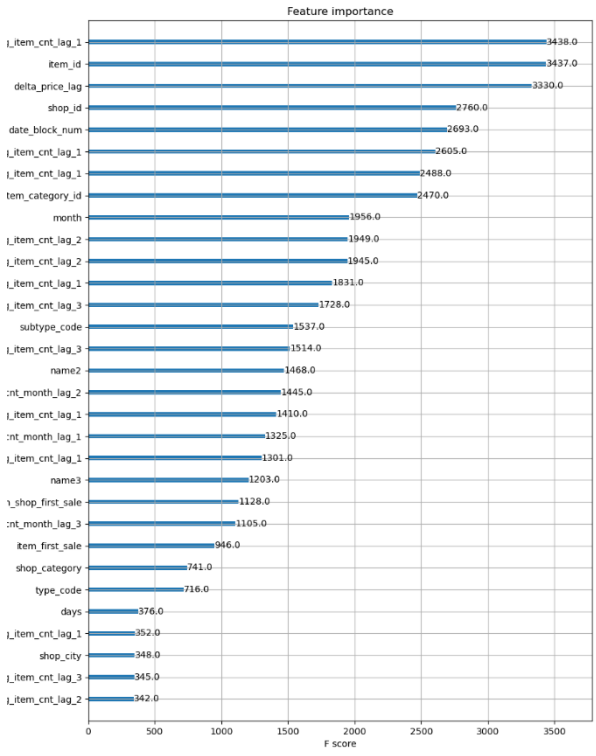

fig 4.1: F Scores of the Features

Here are the output values of the item_cnt_month feature based on their ID

ID	Item_cnt_month
0	0.441929
1	0.315621
2	0.850192
3	0.457466
4	2.875596
5	0.585607
6	0.441929

Likewise all the features are calculated and based on these results the predictions were performed. The final accuracy achieved after carefully tuning the hyperparameters for XGBoost is as follows:

Training Accuracy 1.000

Testing Accuracy 0.9834

V. CONCLUSION

Sales forecasting has advanced significantly over the years and is essential to many businesses' ability to compete. In this research, We've made an effort to apply several learning algorithms for sales forecasting. On this dataset, XGBoost fared the best at achieving our goal. This can be attributed to the effective feature engineering and the capability of optimizing with a wide variety of hyperparameters.

As we've shown, choosing the best parameters for a neural network architecture can frequently distinguish between average and cutting-edge performance. Hence, the tailored LSTM-based architecture may be tuned using a more exacting method for choosing hyperparameters. Lastly, since this might possibly minimize both bias and variation in our output predictions, we would also like to apply ensemble approaches in the future to aggregate predictions from the many models we tried.

VI. FUTURE WORK

Here are some potential areas for future work in sales forecasting using XGBoost:

1. Incorporating external factors: While historical sales data is essential for forecasting, incorporating external factors such as weather, social media sentiment, and economic indicators could further improve the accuracy of the XGBoost model.

2. Time-series analysis: XGBoost can be further optimized for time-series data by incorporating techniques such as lagged variables, differencing, and seasonal decomposition. These techniques can help capture seasonality, trends, and other temporal patterns in the data.

3. Ensemble learning: Ensemble learning techniques such as stacking or blending could be used to combine the predictions of multiple XGBoost models with different hyperparameters or feature subsets. This could help further improve the accuracy of the sales forecasts.

4. Transfer learning: Transfer learning, which involves reusing pre-trained models for related tasks, could be used to transfer knowledge from sales forecasting in one domain to another. For example, a pre-trained XGBoost model for forecasting sales in the retail industry could be fine-tuned for sales forecasting in the automotive industry.

5. Uncertainty quantification: Uncertainty quantification techniques such as Monte Carlo simulations or Bayesian inference could be used to estimate the uncertainty of the XGBoost model's forecasts. This could help decision-makers to make informed decisions under uncertainty.

6. Interpretability: XGBoost is often considered a "black-box" model, meaning that it can be challenging to interpret its internal workings. Future work could focus on developing techniques to make XGBoost more interpretable, such as feature importance analysis or partial dependence plots.

In conclusion, there are several areas for future work in sales forecasting using XGBoost, ranging from incorporating external factors to improving interpretability. These areas can help further improve the accuracy and usefulness of XGBoost models for sales forecasting.

ACKNOWLEDGMENT

We would like to acknowledge the researchers, data scientists, and practitioners who have contributed to the development and application of XGBoost for sales forecasting. Their efforts have helped to advance the field of machine learning and enabled businesses to make more accurate predictions of future sales. Additionally, We acknowledge the open-source community for their contribution in making XGBoost widely available to the public, enabling more researchers and practitioners to utilize this powerful tool in their work.

REFERENCES

[1]. Xinjie Li; Jiakai Du; Yang Wang; Yuanm Cao; Automatic Sales forecasting using LSTM Networks, IEEE Internanational Workshop, Shangai,China,20-22 November 2020.21

[2]. Alsem, K. J., P. S. H. and Reuyl, J. C. (1990) "The accuracy of market share models," International Journal of Research in Marketing 6: 183-99.

[3]. Brodie, R. J., Danaher, V. and Leeflang, P. (2002) "Market shares forecasting," in J. S. Norwell, MA: Kluwer Academic Publishers.

[4]. Y. Wang, X. Chen, Y. Zhao et al., "Mobile recommendation system based on gradient boosting of decision trees," Int. Joint Conf., Vancouver, BC, Canada, 2016.

[5]. Der Gooijeer JG, Hyndman RJ (2006) 15 years of time series forecast. Int J Forecasting.

[6]. Y. Guo and X. Li, "Application of an improved algorithm of a mobile e-commerce system," Industrial Management and Data Systems, vol. 118, no. 2, pp. 297–303, 2017.

[7].Zixuan Huo, Sales forecasting using Machine Learning, In Proceedings of IEEE International Workshop, Hangzhou, China,5-7 March 2021.

[8]. Sunitha Chriyan; Shaniba Ibrahim; Saju Mahanan; Susan Treesa; In proceedings of IEEE International Workshop, South end,UK,16-17 August 2018.

[9]. Walmart Sales Forecasting using RMSE algorithm and Feature engineering | IEEE Conference Publication | IEEE Xplore