Major Research Project

Walmart Sales Forecasting

Methodology and Experiments

Moeen Bagheri

July 2020

Methodology

Aim of Study

The data contains a time-series of the number of sales for 3049 products across 10 different stores. We aim to create several models and compare their performance on predicting the number of sales for all 30490 items for the next 28 days. Specifically, we aim to compare the performance of deep learning and machine learning methods. For deep learning methods, we will compare the performance of Recurrent Neural Networks (RNN) with Artificial Neural Networks (ANN). Moreover, the machine learning method was chosen to be a LightGBM model. Additionally, we will create hybrid models and check if it is able to improve our performance compared to the other models.

Selection of the Response Variable

All models will be trained using the Root Mean Square Error (RMSE) as the loss function. Moreover, the RMSE on the validation set will be used to optimize the hyperparameters of each model. On the other hand, in order to evaluate and compare the performances of the models, we will obtain the Root Mean Square Scaled Error (RMSSE) of the models on the test set. Compared to RMSE, RMSSE has the advantage of being scale independent, and so it can be used to compare the accuracy of forecasts across time-series with different scales. For example, an absolute error of 10 when approximating 100 should have a much higher forecast error than an absolute error of 10 when approximating 1,000,000. Moreover, since RMSSE is scale independent, it can easily be used to compare the performance of different models, regardless of the type or amount of normalization/scaling done on the data during

preprocessing. RMSSE is derived from the Mean Absolute Scaled Error (MASE), which is a common measure of accuracy for forecasting problems. In order to scale the error, the Mean Squared Error (MSE) of the model is compared to the MSE of a naive model, which predicts the sales at each time-step to be the same as the previous time-step. The original RMSSE uses the in-sample data (training set) for the naive model, since the out of sample data (test set) may not contain enough points to obtain a naive prediction when the forecasting horizon is too short: [1, 2]

$$RMSSE = \sqrt{\frac{MSE_{test,\,model}}{MSE_{train,\,naive}}}$$

However, since in our study we have a forecasting horizon of 28, we do not have to worry about this issue and hence for simplicity, we will use the out of sample data to obtain the MSE of the naive model:

$$RMSSE = \sqrt{\frac{MSE_{test,\,model}}{MSE_{test,\,naive}}}$$

Another advantage of RMSSE compared to RMSE is its interpretability. The value of RMSSE represents how well a model performs compared to a naive model. An $RMSSE > 1$ means that the model performs worse than the naive model and should be discarded, an $RMSSE = 1$ means the model performs just as well as the naive model, and an $RMSSE < 1$ means the model performs better than the naive model. The closer the value of RMSSE is to 0, the better the model performs.

In addition to RMSSE, the bias and variance of the models on the test set will be used in order to obtain an unbiased estimate of their predictive power.

Choice of Factors and Levels

A total of 5 models will be evaluated and compared in this project. The choice of the models are as follows: **(1)** Hybrid model of LSTM-ANN, **(2)** Hybrid model of LSTM-LGBM, **(3)** LSTM, **(4)** Artificial Neural Network, **(5)** LightGBM. For the LSTM model, the length of the sample series will be sampled from multiples of 50 with a maximum threshold of 300, the number of units for the layers of the LSTM and ANN models will be sampled from

the exponential space (32, 64, 128, ...), and the number of layers will be sampled to be from one to four. Moreover, the learning rate and regularization hyperparameters of all models will be optimized by sampling values from the log space.

Choice of Experimental Design

The time-series include a total of 1969 days of sales data, which will be split into training, validation, and test sets. The training set will include 1913 days of time-series data, which will be used to train the hybrid model. The validation set will include a time-series data for the next 28 days, from day 1914 to day 1941, which will be used for hyperparameter optimization. Finally, the test set will include the last 28 days of time-series data, from day 1942 to day 1969, which will be used to evaluate and compare the models and assess their generalization on unseen data.

There are two components in the dataset that must be considered when constructing the model's architecture. The first component is the time-series data and the second component is all other data, such as promotions and holidays, that are considered to be external factors and might have an influence on the time-series data. Since the dataset includes information on external factors, it is important to select a model architecture that is able to make use of this information.

Hence, our first choice of model will be a hybrid model with two components. The first component of the hybrid model will be an LSTM network that will learn from the time-series data. LSTM networks are able to handle both the linear and nonlinear demand variations, which eliminates the need of multiple methods for different demand variations [3]. The second component of the hybrid model will serve to learn from the external factors. Specifically, the second component will predict the difference in LSTM's predictions and actual demand based on the external factors. To elaborate, assume we want to predict the demand for a time period, $t \geq 1$, with the actual demand data, Y_t, known. If the LSTM model makes predictions, $\hat{y}_t$, for this time period, then the difference in the LSTM's

predictions and the actual demand is calculated as:

$$\Delta y_t = Y_t - \hat{y}_t$$

This difference, Δy_t will be used as the independent variable(s) in the second component of the hybrid model, along with the external factors as the dependant variables. The second component will then be trained to predict this difference, $\Delta \hat{y}_t$, based on the external factors. At last, the final forecast, $\hat{Y}_t$, will be obtained by aggregating the predictions of the two model components [3]:

$$\hat{Y}_t = \hat{y}_t + \Delta \hat{y}_t$$

The study done in [3] has already used a Random Forest (RF) network as the second component for forecasting demands of a multi-channel retail, and they were able to achieve very accurate predictions. In this study, we will try a LightGBM (LGBM) model as the second component. Similar to random forest, LGBM is a tree-based learning model. However, LGBM uses gradient boosting to enhance the performance of the model, whereas random forest uses bagging. In addition to LGBM, we will also evaluate the performance of an Artificial Neural Network (ANN) as the second component of the hybrid model.

On a separate note, since the dataset contains the sales data for 3049 products sold across ten different stores, we are dealing with a multi-channel problem. It is important to take into account the effects of aggregate demand on the sales. Hence, rather than training individual models for each item, a different model will be trained for each store, separately. Furthermore, since we want to forecast the daily sales for the next 28 days, we are dealing with a multi-step forecasting problem. There are three main methods for dealing with multi-step forecasting problems. In the first method, a separate model is trained for each step. However, this is not very practical, not only because of the number of models that will need to be trained, but also because normally the sales on each day are somehow dependent on the sales of the previous days, which is not considered in this method. In the second method, a single model predicts all steps at once. However, this method also lacks the ability to obtain any information from the previous steps of the forecasting horizon to predict future steps. Alternatively, one can use a single model to

forecast each step one at a time, where at each step, the previous forecasted values are used to forecast the sales of the current day. This method can be more consistent as it takes into account the effects of previous days when forecasting many days ahead. Hence, this is the method that will be used in this project to deal with multi-step forecasting.

For the other models, we will use the components of the hybrid model separately. Therefore, in addition to the hybrid model, we will evaluate the performance of an LSTM model, an ANN model, and a LightGBM model. For the ANN and LightGBM models, in order to help the models learn the seasonality from the time-series data and look more than one step in the past, we will introduce lag features in the dataset. To produce lag features, we will shift the sales data to the future by a fixed number of days. For example, the lag 7 of the sales at time-step t for an item will be its sales value at time-step $t - 7$. In addition to the lag features, we will introduce rolling means and rolling standard deviations of the sales with various sliding-window sizes in order to help the models learn from the time-series data.

All models will be created using `Python`. The LSTM and ANN models will be constructed using the package `keras` and LGBM will be implemented using the package `lightgbm`. Moreover, other packages, such as `Pandas` and `scikit-learn`, will be used for data preprocessing and model evaluation.

Bibliography

[1] "Mean absolute scaled error - mase." https://yardstick.tidymodels.org/reference/mase.html.

[2] "The m5 competition." https://mofc.unic.ac.cy/m5-competition/, 2020.

[3] S. Punia, K. Nikolopoulos, S. P. Singh, J. K. Madaan, and K. Litsiou, "Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail," *International Journal of Production Research*, pp. 1–16, 2020.