A Comparative Study of Machine Learning Frameworks for Demand Forecasting

Kalyan Mupparaju, Anurag Soni, Prasad Gujela, Matthew A Lanham
Purdue University Krannert School of Management
403 W. State St., Krannert Bldg. 466, West Lafayette, IN 47907
kmuppara@purdue.edu, soni16@purdue.edu, pgujela@purdue.edu, lanhamm@purdue.edu

ABSTRACT

We built various demand forecasting models to predict product demand for grocery items using Python's deep learning library. The purpose of these predictive models is to compare the performance of different open-source modeling techniques to predict a time-dependent demand at a store-sku level. These demand forecasting models were developed using Keras and scikit-learn packages and we made comparisons along the following dimensions: 1) predictive performance, 2) runtime, 3) scalability and 4) ease of use. The forecasting models explored in this study are Gradient Boosting, Factorization Machines, and three variations of Deep Neural Networks (DNN). We also explored the effectiveness of categorical embedding layers and sequence-to-sequence type architecture in reducing the errors in the demand forecasts. Our best neural network model is currently placed in the top 1.5 percentile of all the submissions in the Kaggle competition for forecasting retail demand.

INTRODUCTION

In today's world of extreme competition, cost reduction is of utmost importance for organizations, primarily in the retail and consumer product goods (CPG) industries. All the major players in these industries try to focus on cost-cutting and maintaining optimum inventory levels to gain a competitive edge. In addition to cost optimization, having just the right amount of inventory is also becoming important for consumer satisfaction especially in the perishable retail goods market. This is where demand forecasting helps these companies. Efficient and accurate demand forecasts enables organizations to anticipate demand and consequently allocate the optimal amount of resources to minimize stagnant inventory.

Gartner recently published a paper titled, *"Demand Forecasting Leads the List of Challenges Impacting Customer Service Across Industries* (Steutermann, 2016)*"*. As hinted by the title of the article, they conclude that accurate demand forecasts across all customer-facing industries is important for business. In the Forbes article, "*Ten Ways Big Data Is Revolutionizing Supply Chain Management* (Columbus, 2015)", demand forecasting is mentioned as the top 4 supply chain capabilities currently in use.

Despite the wide acceptance and usage of forecasting techniques, they have been limited to macro level forecasts. It is only recently that retail companies have started focusing on day level forecasts. A Wall Street Journal article, "*Retailers Rethink Inventory Strategies* (Ziobro, 2016)*"* mentions how Home Depot is trying to minimize its inventory at stores. We see that there is an increasing need for demand forecasting techniques that can accurately predict the demand for each item for each day for every store. This need is being fulfilled in some companies by using open source data science tools whereas few other firms use in-house commercial platforms.

In this paper, we try to evaluate the predictive model performance of models using open-source data science tools like R and Python to predict demand for thousands of products on a store level for a Kaggle competition dataset. Our performance metric is not just limited to model accuracy. We posit that the real value of a model to a business is a composite of (1) predictive model accuracy, (2) runtime, (3) scalability and (4) ease of use.

We structured this paper as follows. We performed a review of the literature to see what methodologies have found to be successful at understanding this problem. We discuss the data set used in our study. Next, we discuss the methodology/design we implemented and discuss the models we investigated. Lastly, we present our results, discuss our conclusions, and how we plan to extend this research.

LITERATURE REVIEW

We started our search for the optimal prediction model for forecasting by looking at past research done in demand forecasting using different machine learning algorithms. This exercise gave us an understanding of different machine learning models that could be used for forecasting. We also looked at measures frequently employed to compare their performances.

Previous research on demand forecasting has traditionally used a methodology called Autoregressive Integrated Moving Average (ARIMA). This methodology has been applied to studies of traffic flow (Williams B. M., 2003) and international travel demand (Lim, Time series forecasts of international

travel demand for Australia, 2002). Lim's paper analyzed stationary and non-stationary international tourism time-series data by formally testing for the presence of unit roots and seasonal unit roots prior to estimation, model selection, and forecasting. They used mean absolute percentage error (MAPE) and root mean squared error (RMSE) as measures of forecast accuracy. This paper showed that by comparing the RMSE's, lower post-sample forecast errors were obtained when time-series methods such as the Box–Jenkins ARIMA and seasonal ARIMA models were used.

Ching-Wu Chu et al. (Ching-Wu Chu, A comparative study of linear and nonlinear models for aggregate retail sales forecasting, 2003) compared the performances of linear (traditional ARIMA) models to non-linear models (Artificial Neural Networks) in forecasting aggregate retail sales. Here, neural networks with de-seasonalized data performed best overall, while ARIMA and neural networks modeled with original data perform about the same. After reviewing this research, we decided to try both ARIMA and neural networks in our analysis.

(Guolin Ke, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, 2017) found that data instances with larger gradients play a more important role in the computation of information gain, GOSS can obtain quite accurate estimation of the information gain with a much smaller data size. We evaluated the performance of LGBM for time dependent sales data for prediction.

Neural Network models can be applied to a variety of domains and are useful for high dimensional data. However, there are several criticisms of neural networks. The cost-benefit of neural networks is limited in scenarios where the problem space does not have well-defined training examples to learn from. Further, the time required to train and tune the network increases as the number of nodes and connections increases. Finally, these networks can become "black boxes" as the explanation of how they arrived at a given result can technically dense for a non-technical audience (Tu, 1996).

Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. Truncating the gradient where this does not do harm, Long Short-Term Memory (LSTM) can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing constant error flow through "constant error carrousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is O(1). While (Graves, 2013) used LSTM to predict next sequence of text, we are using similar time dependent data of sales to predict the future sales.

Sutskever et al.(Sutskever, 2014) showed the effectiveness of an encoder decoder recurrent neural network structure for sequence-to-sequence prediction. A multilayered Long Short-Term Memory (LSTM) can be used to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. While Sutskever et al. used this approach in language translation, this method seems suitable for demand forecasting where we take the input sequence as the sales for the previous days, and predict the sequence of the sales in the future.

Cheng Guo and Felix Berkhahn (Cheng Guo, 2016) introduced the concept of categorical embedding in neural networks which can be used in building neural networks with categorical predictors. Embedding reduces memory and speeds up neural networks compared with one-hot encoding. It also captures the intrinsic relations between the categories by mapping each category to a Euclidean space.

This concept can be utilized while building neural network models to predict sales of a wide range of items which belong to different families.

Liu Yue, et al. in their 2007 paper (Liu Yue, 2007), have shown the effectiveness of another machine learning model in demand forecasting which is Support Vector Machine (SVM). In this research, the model of SVM is introduced into the retail industry for demand forecasting, and the experiment results show that the performance of SVM is superior to traditional statistical models and the traditional Radius Basis Function Neural Network (RBFNN). Liu Yue and colleagues also agree that the prediction accuracy of SVM can also be improved by using ensemble-learning techniques.

Factorization Machines can estimate interactions even in problems with huge sparsity (like recommender systems) where SVMs fail. (Rendle, 2011) shows that the model equation of FMs can be calculated in linear time and thus FMs can be optimized directly.

From our research of previous studies on demand forecasting, we have seen that a large variety of machine learning models like ARIMA, Exponential Smoothing, Neural Networks, and Support Vector Machines have been used. The accuracy values for forecasts are generally measured in RMSE or MAPE. Table 1 below is a summarization of the literature review.

Studies	**Motivation for the Research**	**Result of the Research**
Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering, 129(6), 664–672* (Williams B. M., 2003) Lim, C., & McAleer, M. (2002). Time series forecasts of international travel demand for Australia. *Tourism Management, 23(4), 389–396* (Lim, Time series forecasts of international travel demand for Australia, 2002)	To study the traditional methods of demand forecasting	• ARIMA models can give good accuracy • May cause problems in the initial model selection as they are based on heuristic selection of parameters • Can be time-consuming if many time series observations are to be analyzed • RMSE and MAPE are widely used to measure forecast performance
Ching-Wu Chu, & Guoqiang Peter Zhang. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics, 86(3), 217-231.*	To see if newer nonlinear machine learning models perform better than traditional methods	• Neural networks with deseasonalized data perform the best overall
Zell, A. (1994). *Simulation neuronaler netze* (Vol. 1). Addison-Wesley Bonn. (Zell, Simulation neuronaler netze (Vol. 1), 1994) Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications, 40*(16),	To explore other methods beyond Neural Nets and ARIMA	• SVM models have also been successful in giving good forecasts results for retail demand

6266–6282. (Ghiassi, 2013)		
Guolin Ke, Qi Meng, Thomas Finley et. all. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree (Guolin Ke, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, 2017) Jie Zhu, Ying Shan, JC Mao, et. all. (2017). Deep Embedding Forest: Forest-based Serving with Deep Embedding Features (Jie Zhu, 2017)	To explore high performance GBM methods for data forecasting	• LGBM models can significantly outperform XGBoost and SGB in terms of computational speed and memory consumption
A. Graves (2013). In Arxiv preprint arXiv:1308.0850, Generating Sequences with Recurrent Neural Networks (Graves, 2013) S. Hochreiter and J. Schmidhuber. (1997). LSTM can solve hard long time-lag problems (Schmidhuber, 1997) Q.V. Le, M.A. Ranzato, R. Monga, et. all (ICML 2012). Building high-level features using large scale unsupervised learning. (Q.V. Le, 2012)	To understand different Recurrent Neural Network models	• Long Short-Term Memory (LSTM) is a very powerful technique to predict sequence • A simple, straightforward and relatively unoptimized approach can outperform a mature SMT system.
Sutskever, O. Vinyals, Q.V. Le, (2014). Sequence to Sequence Learning with Neural Networks. (Sutskever, 2014) Jean, S´ebastien, Cho, Kyunghyun, et. all, (ACL 2015). On using very large target vocabulary for neural machine translation. (Jean, 2015) Thrun, Sebastian. (NIPS, 1996) Is learning the n-th thing any easier than learning the first? (Thrun, 1996)	To study methods beyond LSTM for sales forecasting	• Sequence to sequence should perform well while predicting a sequence of time dependent demand
Steffen Rendle, (2011). Factorization Machines (Rendle, 2011) A. Toscher, M. Jahrer, and R. M. Bell., (2009). The BigChaos Solution to the Netflix Grand Prize (A. Toscher, 2009) M. Piotte, and M. Chabbert, (2009). The Pragmatic Theory Solution to the Netflix Grand Prize (M. Piotte, 2009)	To study methods that can predict demand for sparse dataset	• Factorization machines perform better than SVM for sparse dataset • Factorization machines can be used to predict demand effectively with many categorical variables

Xiangnan He, Tat-Seng Chua (2017). Neural Factorization Machines for Sparse Predictive Analytics (Xiangnan He, 2017)		
Cheng Guo, Felix Berkhahn (2016). Entity Embeddings of Categorical Variables (Cheng Guo, 2016) Yoshua Bengio and Samy Bengio, (NIPS 1999). Modeling high dimensional discrete data with multi-layer neural networks (Bengio, 1999) B. Yang, W. Yih, X. He, J. Gao, and Li Deng, arXiv preprint arXiv:1412.6575. (2014). Embedding entities and relations for learning and inference in knowledge bases (B. Yang, 2014)	To study the impact of Embedding for Categorical variables	• Entity embedding not only reduces memory usage and speeds up neural networks compared to one hot encoding • More importantly, it reveals the intrinsic properties of the categorical variables

Table 1: Literature review summary

DATA

The data used in this research is from the Kaggle competition which aims to forecast demand for millions of items at a store and day level for a South American grocery chain. (https://www.kaggle.com/c/favorita-grocery-sales-forecasting/data).

The data is provided in different tables named train, test, stores (store related data), items (merchandise data), transaction, oil (oil prices can be a good predictor of sales as Ecuador is an oil-dependent economy), and holidays events (holiday & major event related data). Table 2 provides a summary of all the important columns is given below along with the relations between each data table provided.

Variable	**Type**	**Description**
Id	Integer	Identifier defined at the date-store-item-promotion level
Unit_Sales	Numeric	Sales defined at the date-store-item-promotion level
Date	Date	Date of transaction for an item
Store_Nbr	Integer	Store identifier
Item_Nbr	Integer	Item identifier
Onpromotion	Boolean	Whether the item is on promotion
City	Text	City in which store is located
State	Text	State in which store is located
Type	Text	internal store categorization
Cluster	Integer	internal store clustering
Family	Text	The family of item
Class	Text	Class of items
Perishable	Boolean	Whether the item is perishable

Table 2: *Data used in study*

Figure 1 provides a data model of how each table and feature map to each other.

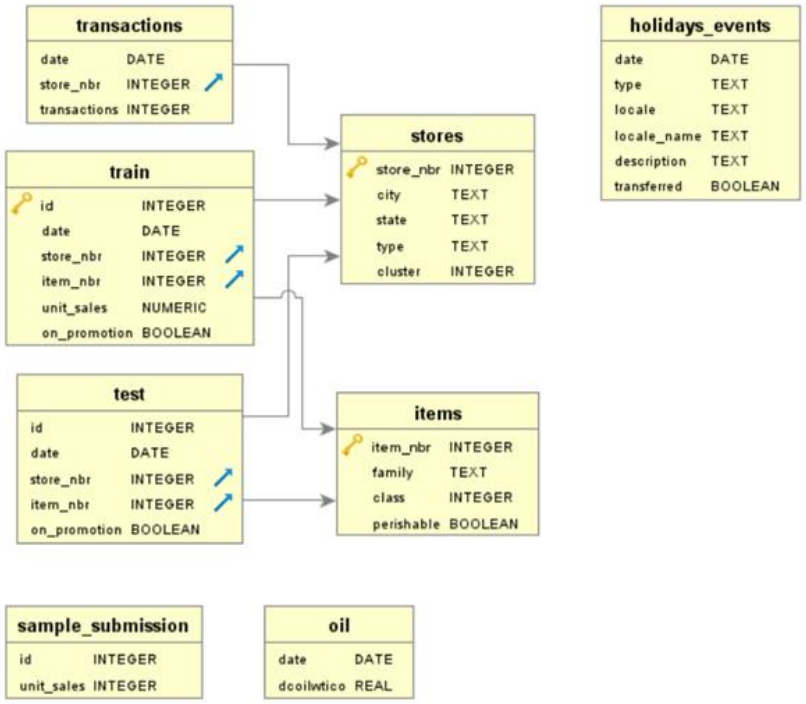

Figure 1: Sales: Data Model

METHODOLOGY

Figure 2 provides a detailed outline of the data mining workflow used in this study.

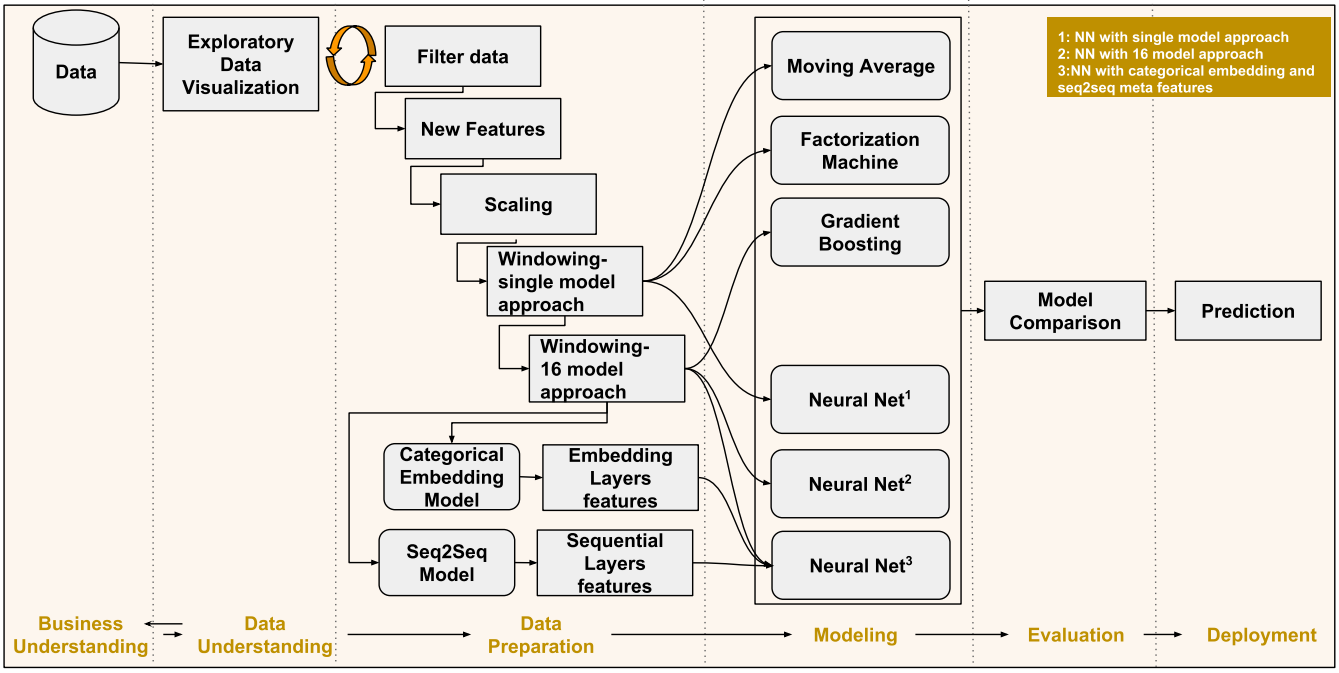

Figure 2: Data Mining Workflow

We set out to follow CRISP-DM (Cross-Industry Standard Process for Data Mining) process model to approach the given problem.

Business Understanding: We started with understanding the business's objectives of the problem and its application. We understood that in addition to the usual trend in shopping of items, there could be external factors (e.g. Oil Price, Holidays) which could affect the demand. Further, we understood the rationale of high weight given to perishable goods and its impact on the business. As we understood more, it was clear to use that it was a short-term (15^{th} days) demand forecasting required at a granular level. We prepared a preliminary strategy by starting our research more into various modeling technique applicable to time-series and best practices and tooling required in dealing with the Big-Data (100 million rows of training). Through Literature review, we were mindful of the fact that some model-output (e.g. PCA, Clustering) would serve as input for other models and for this reason, it was included in the strategy to try out various model (with moving average features).

Data Understanding: We sourced flat data files and made cosmetic changes to do EDA (Exploratory Data Analysis) exercise. Then, with a connection to Tableau, we collected the basic facts about the data and studied the distribution of all the key variables.

Data Preparation: Keras requires the predictor variables to be represented as the values in a dataset matrix with predicting values acting as one index and timestep as another dimension. These meant that we had to unstack time from rows to columns. This process is called windowing and is one of the most popular technique to convert time series forecasting into a supervised leaning problem. From a cutoff date in a dataset, we then calculated features of running sales mean at day 1, 3,7,14,30,60,140; running promo sums at 14, 60, and 140 days, running sales mean at 7 and 14 weeks. Further, we arranged the known sales amount of 16 testing days in the same matrix. For feature engineering, we added categorical embedding features of store/location characteristics and sales sequential prediction (from sequence to sequence method, serving as meta-model).

Modeling: Various models such as Moving Average, Factorization Machine, LGBM and Deep Neural Nets were trained and evaluated against each other. training and testing, we utilized the matrix in two different ways. One way is when all 16 testing days are considered as one dependent variable and trained on data before the cutoff. This is the single model approach for Neural Network(NN^1). Another way is when in which X horizon gets fixed but varying Yi, where i is one testing day. This led us to build 16 different neural networks for each of the days. This set of NN is called as 16 model approach of Neural Network(NN^2). Both single and 16 models approach datasets were also used to train other models as well. Finally, (NN^3) was developed as in improvement over NN^2 by including categorical features and sequence-to-sequence(seq2seq) metamodel.

Evaluation: Forecasting models are evaluated based on the statistical measures – Normalized Weighted Mean Squared Logarithmic Error(NWMSLE).

$$\text{NWMSLE} = \frac{\sum_{i=1}^{n} w_i (\ln(\hat{y_i} + 1) - \ln(y_i + 1))^2}{\sum_{i=1}^{n} w_i}$$

$\widehat{y_i}$ is the predicted sales at i SKU-Store level; y_i is the predicted sales at i SKU-Store level; and w_i is the weight given to SKU. Perishable items are given higher weights in evaluation. Other items' weights are kept equal to 1.

These are the standard measures to check performance in forecasting-based problems based on texts and literature.

FORECASTING MODELS

Moving Average Method

This is one of the oldest and most widely used methods of demand forecasting. In this method, the average sales of the previous 3 days, 7 days, 14 days, 28 days, 56 days, 112 days, & 180 days are used as the predictor for the sales of the next day. The predictions are multiplied by a factor that takes care of the difference in sales across the different days of the week. It is simple and gives good accuracy when done on a short-term horizon. However, it is not likely to predict well for a longer-duration span as it is not generalizing the trend mere following the past behavior with auto-regressive components.

LGBM

Light gradient Boosting Model(LGBM) is a fast variant method in the class of tree-based boosting algorithm. It is designed to be run for large data size where it provides the maximum time performance while achieving the same accuracy as other Decision Trees Boosting methods such as XGBoost and pGBRT. It runs by splitting the tree at a leaf rather than at a level. Specifically, it utilizes two novel techniques – Gradient-based One-Side Sampling(GOSS) and Exclusive Feature Bundling(EFB). GOSS samples out data having a smaller gradient to maximize the estimated information gain. This makes it possible to exclude a relatively large amount of data which doesn't contribute much to learning and expends a lot of processing. EFB is a dimensional reduction technique that optimally bundles highly uncorrelated features (i.e. they rarely take non-zero values simultaneously). It is one of the most popular machine learning algorithm in data competitions and speeds up the similar process by over 20 times.

Factorization Machine

Factorization Machine models are general predictor model (like SVM) that works well under very high-sparse data. It aims to learn a kernel by representing higher-order terms as the product of latent factor vectors. It essentially aims to reveal those latent factors that capture the interaction between factors. They are generic approach models that can mimic the superiority of factorization models like timeSVD++[Koren 2009b], FPMC[Rendle et al. 2010]. FMs model equation in the linear time leading to the fast computation of model. FMs allow for parameter estimation even under sparse data. Factorization of parameters allows for estimation of higher order interaction effects even if no observations for the interactions are available. As our dataset is sparse and the data size is large, we will not consider SVMs in our study but rather use Factorization Machines.

Deep Neural Networks

Neural network models have also been proposed as a means of predicting unit sales. There are several advantages to using neural networks for predictive analytics. First, neural networks are highly tunable and can reliably process large volumes of data. Second, these networks are less sensitive to outliers or extreme values than linear models.

However, these techniques are not without limitations. Once trained, neural networks are difficult to retrain. These networks require special software packages to handle time series data. Finally, these networks have a reputation for being "black boxes" and being difficult for non-technical audiences to interpret. Learning from ANN was enhanced with following methods:

Categorical Embedding: It is an advanced method (vs. one-hot encoding or dummy) to handle categorical data in machine learning model. Embedding maps categorical a feature to a continuous n-dimensional weight space based on a neural network algorithm which has a loss function defined on the target variable. This weights space exhibits the closeness within categorical values. For example, bigger cities would demonstrate similar weights indicating a closeness in the shopping distribution. This is in direct contrast to one-hot encoding where every category is given a weight of 1.

Sequence to Sequence Learning as metamodel: Sequential data poses a unique problem in form of a non-fixed dimensionality or in other words, whose length is not known apriori. An LSTM(Long Short-Term Memory) architecture can solve the problem by reading the sequence one step at a time and mapping it in to a large fixed-dimensional space, also called as Encoding, which can then be fed into another RNN(Recurrent Neural Network) model that learn and predicts the long-range dependencies, also called as Decoding. Thus, this model is also referred to as encoder-decoder architecture. Here, for every store-item level, sales are considered as sequential data and output were generated which acted as features for out DNN.

RESULTS

Descriptive Analysis

We first perform descriptive analytics on the sample of data we have taken. We plot the macro trends in the total sales of all items across all stores. The sales have been transformed using a log of (1+ sales) method to make sure that the variance of sales does not change greatly with time.

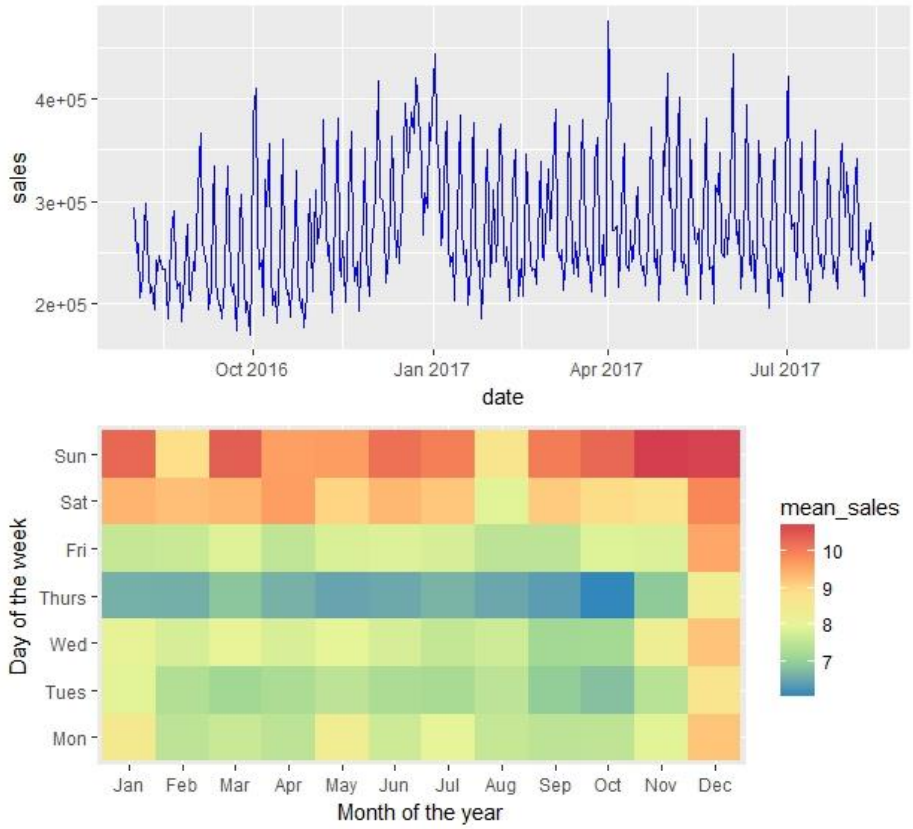

Figure 3: Exploratory analysis of Sales Data vs. Time

We see from the plots above that there is a clear seasonal (weekly) trend in the sales of grocery items. As one would expect, the sales are higher on the weekends and lower during the weekdays. Also, grocery sales seem to increase during the month of December. These results
As we have nearly 167000 store-item combinations, and the sales for each of these store-items is essentially a unique time series, visually inspecting all the features is not useful in this case.

Forecasting models

The objective of all the machine learning models that were built was to forecast sales for about 167000 store-items for the period of 16-31th August 2017. First, a simple moving average model was built in Python. This model performed decently (Top 50%ile in the Kaggle competition presently) and gave a good baseline model to improve upon. Next, we built a neural network model and a gradient boosting model that utilize the moving average at different lags as the predictors As we had to predict the sales for a horizon of 16 days, we built 16 different models each one utilizing the data till time (t) and predicting the sales for time (t+1), (t+2), …, so on till (t+16). Both the 16-neural network(NN^2) and the 16-gradient boosting models approach stood in the top 2%ile of all solutions in the Kaggle competition. We then tried a single neural network(NN^1) that was trained with all (t+16) sales as the dependent, in order to predict the sales for the entire 16-day horizon using a single model. The results from this approach were worse than the 16-model approach. We also built a factorization machine model, but the 16-model neural network approach was still the best model we had at this point. After that, we tried a sequence-to-sequence approach to forecast the sequence of 16-day sales using the sequence of previous 50-day sales as input. We decided to use the output of this model as inputs to the 16-model neural network approach to improve the overall accuracy of the model. We also added

additional features to the 16-model approach using categorical embedding to create a final model(NN^3). This model reduced the model error slightly and is currently in the top 1.5%ile. The runtime comparisons below also show that all the models ran in a reasonable amount of time on a GPU supported machine.

Best Model Selection

In general, when retailers need forecasts of sales/demand, they are more interested in the forecasts for perishable goods than the nonperishable goods. Therefore, to have a higher accuracy in the forecasts of perishable items, we up-weighted the errors for perishable items by 25%. We used a normalized weighted mean square error metric to select the best one among all the models we built. As we transformed the sales using a log transform the actual metric we ended up using is normalized weighted mean square log error (NWMSLE).

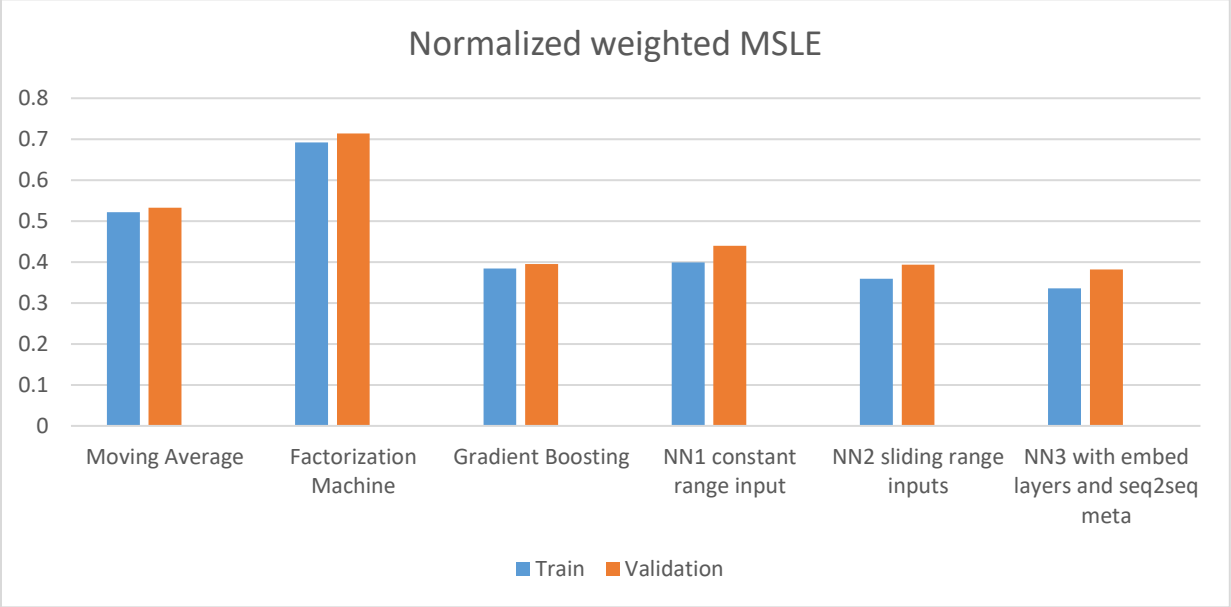

Figure 4: Comparison of Models built Normalized Weighted Mean Square Log error

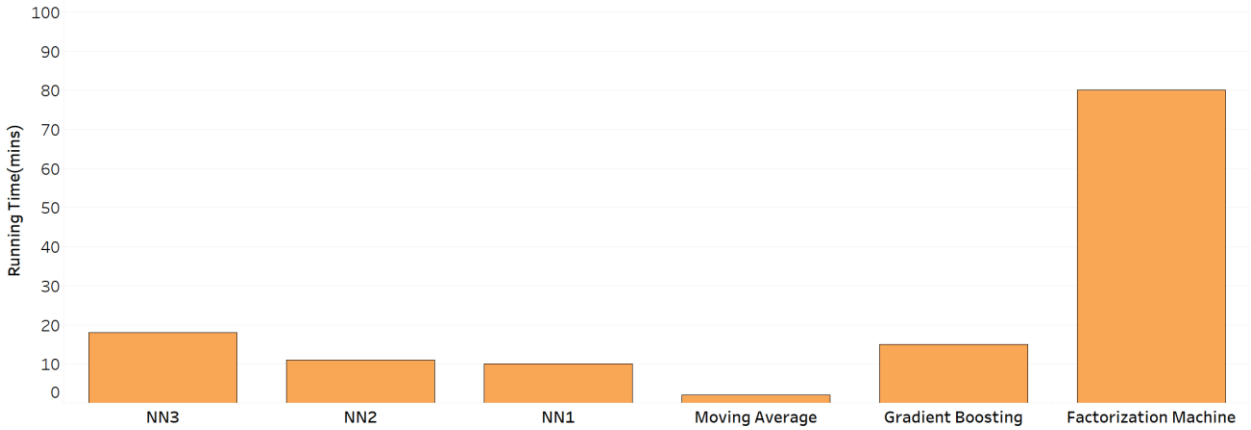

Figure 5: Running Time Comparison

CONCLUSION

Demand Forecasting at a granular level is a complicated analytical problem with multiple time-series all propagating in tandem and affected by external factors like Oil Price, Holidays etc.. Inventory Surplus and "Stock-outs" are key ground level issues in retail store management company affecting the bottom-line margins. A handy digital tool driven by this model could help the category manager to decide on the daily inventory stocks for millions of items, at various store locations. This model could also help them to estimate the demand for a new item without any historical shopping data. Though R/Python offers a basket of models to structure this complex problem to a manageable solution, infrastructure limitation constrained us to use a subset of data for training which could underrepresent the information contained in the model. The model also assumes the absence of catastrophic events (like earthquakes), in presence of which the variance would shoot up drastically here.

We could further improve our forecasts in future by attempting to use other features like item and store attributes in a sequence-to-sequence type neural network. It would also be interesting to look further into the macro inputs (other than Oil price) indicative of the health of economy like GDP/Inflation rate/Unemployment etc.

REFERENCES

A. Toscher, M. J. (2009). The BigChaos Solution to the Netflix Grand Prize.

B. Yang, W. Y. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Bengio, Y. B. (1999). Modeling high dimensional discrete data with multi-layer neural networks. *NIPS*.

Cheng Guo, F. B. (2016). Entity Embeddings of Categorical Variables.

Ching-Wu Chu, &. G. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics, 86(3), 217-231*.

Columbus, L. (2015). *Ten Ways Big Data Is Revolutionizing Supply Chain Management.* Forbes.

Ghiassi, M. S. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications, 40(16),* 6266-6282.

Graves, A. (2013). Generating Sequences with Recurrent Neural Networks. *Arxiv preprint arXiv:1308.0850*.

Guolin Ke, Q. M. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree.

Jean, S. C. (2015). On using very large target vocabulary for neural machine translation. *ACL*.

Jie Zhu, Y. S. (2017). Deep Embedding Forest: Forest-based Serving with Deep Embedding Features.

Lim, C. &. (2002). Time series forecasts of international travel demand for Australia. *Tourism Management, 23(4),* 389-396.

Liu Yue, Y. Y. (2007). Demand Forecasting by Using Support Vector Machines. *Third International Conference on Natural Computation*.

M. Piotte, a. M. (2009). The Pragmatic Theory Solution to the Netflix Grand Prize.

Q.V. Le, M. R. (2012). Building high-level features using large scale unsupervised learning. *ICML*.
Rendle, S. (2011). Factorization Machines.

Schmidhuber, S. H. (1997). LSTM can solve hard long time-lag problems.
sdas. (n.d.). dasd. *das*.

Steutermann, S. (2016). *Demand Forecasting Leads the List of Challenges Impacting Customer Service Across Industries.* Gartner.

Sutskever, O. V. (2014). Sequence to Sequence Learning with Neural Networks.
Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? *NIPS*.
Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 1225-1231.

Williams, B. M. (2003). *Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results* . Journal of Transportation Engineering, 129(6), 664-672.

Xiangnan He, T.-S. C. (2017). Neural Factorization Machines for Sparse Predictive Analytics.

Zell, A. (1994). Simulation neuronaler netze (Vol. 1). Addison-Wesley Bonn.

Ziobro, P. (2016). Retailers Rethink Inventory Strategies. *WSJ*.