Temporal Demand Forecasting Using Machine Learning Techniques: A Comparative-Study of Open-Source Versus Commercial Solution

Kalyan Mupparaju, Anurag Soni, Prasad Gujela, Matthew A. Lanham

Purdue University Krannert School of Management, 403 W. State Street, West Lafayette, IN 47907

kmuppara@purdue.edu, soni16@purdue.edu, pgujela@purdue.edu, lanhamm@purdue.edu

Abstract

In collaboration with a national consulting company, this study's objectives are twofold: (1) which machine learning approaches perform the best at predicting demand for grocery items? and (2) what is the performance one could expect to achieve using an open-source workflow versus using proprietary in-house machine learning software? The motivation behind this research is that consulting companies regularly help their retail clients try to understand demand as accurately as possible, but also in a scalable and efficient manner. Efficient and accurate demand forecast enables retailers to anticipate demand and plan better. In addition to delivering accurate results, data science teams must also continue to develop and improve their workflow so that experiments can be performed with greater easy and speed. We found that using open-source technologies such as scikit-learn, postgreSQL, and R, a decent performing workflow could be developed to train and score forecasts for thousands of products and stores accurately at various aggregated levels (e.g. day/week/month) level using deep-learning algorithms. However, the performance of our solution is yet to be compared to the data science team's commercial platform that we collaborated with and will be added soon. We have been able to learn how they have been able to achieve performance gains (in model accuracy and runtime), which made this collaboration a great learning experience.

Introduction

Having a data science workflow that is effective at identifying consumer demand is the holy grail problem task that many retail and consulting teams work on regularly. Some firms develop customized in-house machine learning solutions with highly specialized researchers, others focus on getting the most out of their expensive commercial platforms (e.g. SAS, IBM, Microsoft, etc.), and others are finding a balance among commercial and open-source. At the end of the day, all these teams seek to generate better decision-support for those decision-makers that they support.

In this study we try to find answers to following business questions related to demand forecasting for grocery items:

1) What predictive model performance can be expected from open-source data science tools to predict demand, which entails generating forecasts for millions of products?

2) How does the predictive performance compare to using a proprietary in-house data science tool of a major business applications company?

With the advent of a booming big data infrastructure market and new deep learning models, it is possible to excavate most of the underlying actionable information hidden inside sales data. Deep learning models are proven to generalize well with a large dataset, which is the case with national grocery chains. In this study, we intend to observe the applicability of a deep-learning based workflow to complex scenarios which retailers can face like adding new locations, new products, new tastes, and unsystematic external factors .

Methodology

Data Sources

Aggregated retail transactional data was provided at store, item, and date level For stores, its type, clusters, and location information was provided. For items, family, class, and perishability information was made available. Additionally, data for possible external factors like holidays and oil prices were also provided to test causality hypotheses. Provided dataset spanned from 01/01/2013 thru 08/15/2017 with required sales predictions within the 08/16/2017 to 08/31/2017 time window . We also created time variable windows by transforming that data into "cross-sectional" format (i.e. pivot dates as columns).

Exploratory Data Analysis

Visualization was utilized to observe the skewness and other anomalies in the data. For example, negatives sales and null values in the early years for promotional data persuaded us to exclude it. We further observed the usual peaks in weekends indicating cruciality of week of day.

Data Preparation

We filtered to include only the transactional data since Dec'15. To align with various models, data was pre-processed to create aggregation windows of various time horizons (i.e. 1, 3, 7, 14, 30, 60, and 140 days). These aggregations served as features for the model. As shown in the figure below, all numerical features were also scaled to fit between 0 and 1.

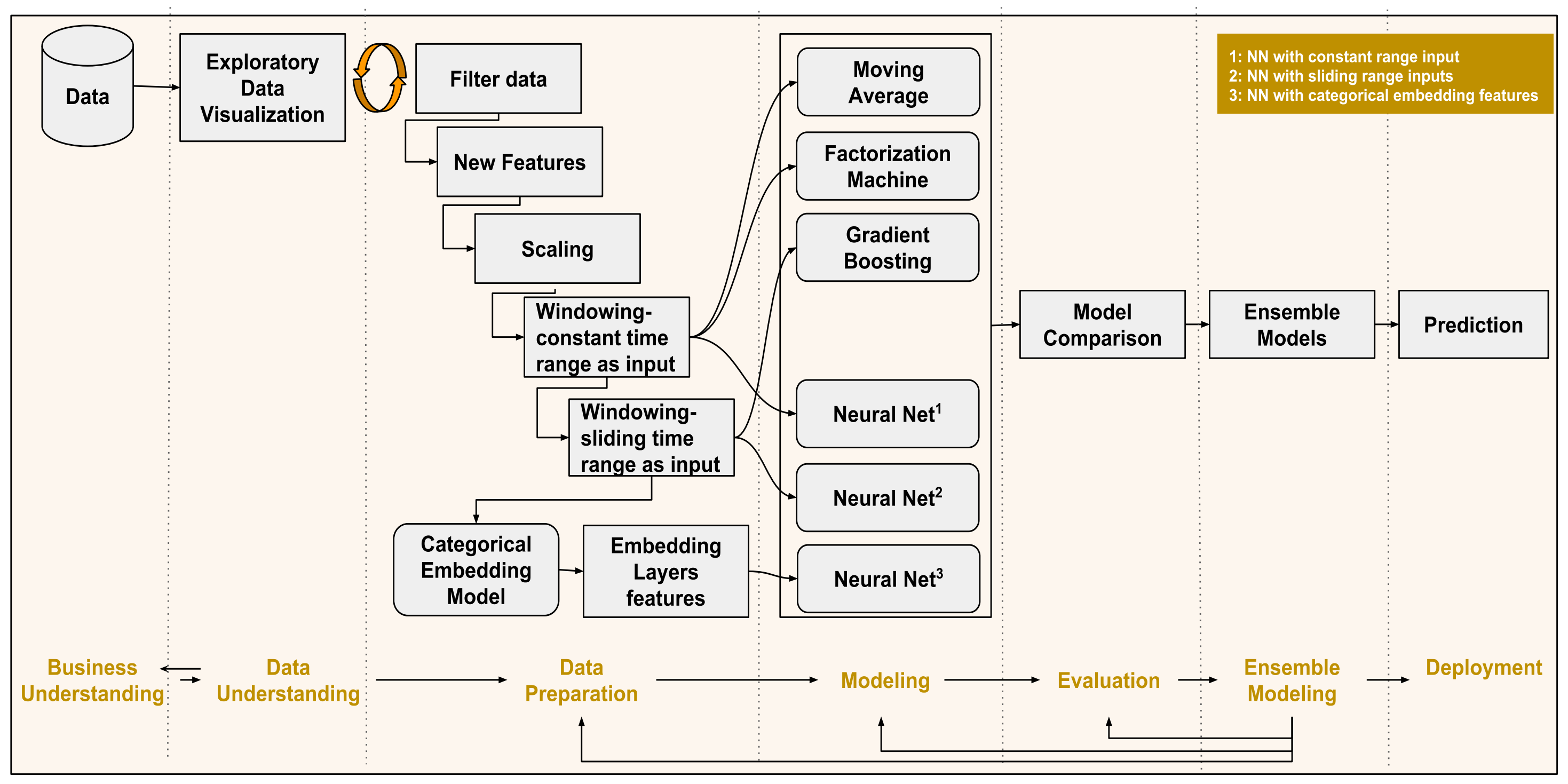

Figure 1: Analytics Workflow

For a few models, we created a set of inputs which had a fixed time range to predict for 16 testing days, while for other models we created inputs with changing sliding time ranges (changing time references). One important component of our architecture is '***Categorical Embedding***' – a technique to map categorical variables to weights which closely resemble the closeness of the categorical value. For instance, milk is closer to juice than an auto-parts category. The mapping is learned by a neural network and output weights are fed into the prediction model in a similar way as variables with one-hot encoding.

Cross-validation

Data was trained on 6 months of data with the last 15 days serving as validation.

Model Building and Comparison/Selection

Our baseline forecasting model was a simple moving average, which surprisingly performed better than expected. After the application of windowing to convert the time-series forecasting problem to a supervised learning problem, we applied a **gradient boosting** model, **factorization machine** (Rendle, 2010), and **neural networks** with and with out categorical embeddings to predict the sales for the test period of 16 days. Neural networks and gradient boosting were chosen as they have been proven to learn complex relations with ease. The factorization machine was chosen because of its ability to handle high dimensional sparse data with ease. We had to forecast the demand for both perishable and non-perishable items. We chose to have a higher weight for accuracy on perishable items as it made better business sense in the retail context. The model with the lowest weighted root mean square logarithmic error (RMSLE) was chosen to be the best performing model.

Ensemble Model

Based on the flexibility offered by different flavors of Neural Nets and Gradient boosting, we choose to combine the predictive performance by ensembling via stacking.

Results

Model comparison

Figure 2 shows two bar graphs highlighting the runtime and the chosen performance measure (RMSLE).

- Deep learning models show relatively better performance than other models.
- Neural nets observed better performance with more features/tweaking, which goes to suggest its methodical learning potential.
- Categorical Embedding is able to deliver on the last-mile improvement which is the most challenging to achieve in such practical scenarios.
- The Factorization Machine implementation we tested had the longest run time and performed the worst among the machine learning candidate models.
- The custom in-house machine learning runtime and performance with our data science partner is currenting underway for comparison and will be added soon.

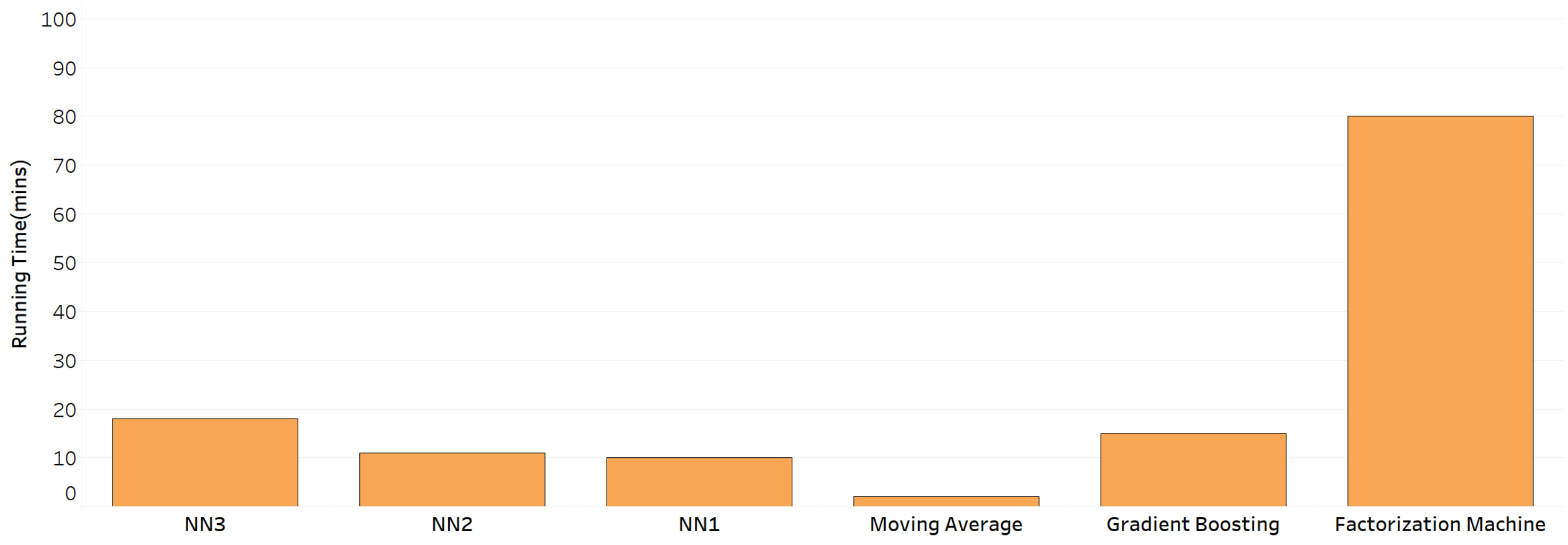

Figure 2: Run Time Summary

The pipeline we created using these open-source frameworks met acceptable running time constraints on a personal computer supported by GPU capability required for deep learning calculations. This provides some support to those teams considering developing their own customized workflows using open-source technologies, while bringing down the overall costs of alternative solutions.

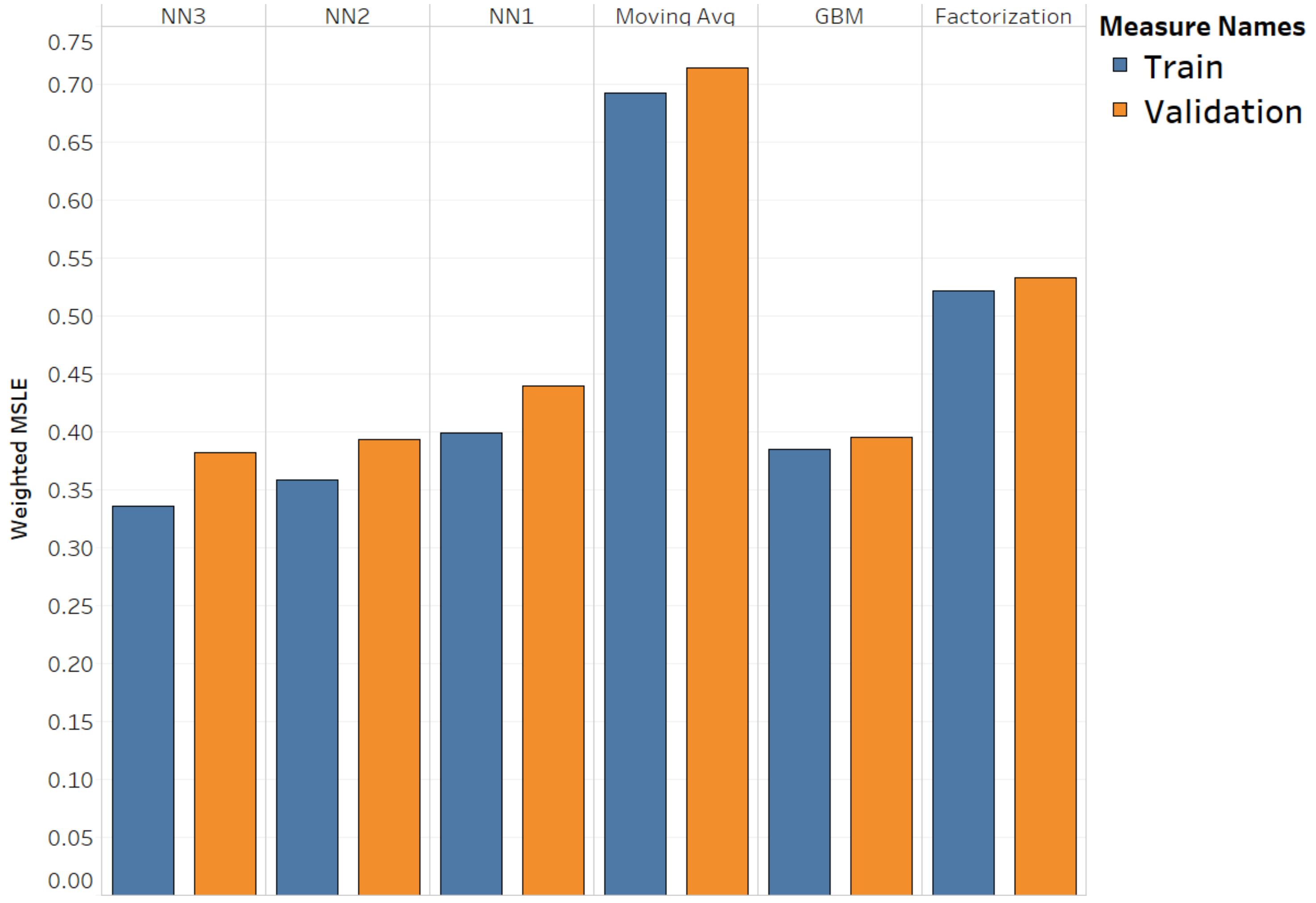

Figure 3: Model Selection Summary

Conclusions

Understanding product demand is the holy grail problem for many consulting companies and the retail clients they serve. Data science team members are often prototyping different experiments that might yield increased predictive performance. While this is important, it must also be performed with an eye on deploying such a model (or ensemble of models) that can efficiently score, often millions of products for thousands of stores. Many commercial platforms such as SAS Enterprise Miner for example, provide a nice workbench for analytics professionals to prototype and deploy off-the-shelf algorithms for prediction problems. However, these platforms can be costly. The data science team we collaborated with has created their own propriety in-house forecasting solution that often leads to them to be able to provide their clients highly accurate forecasts but also quickly. This study provided us a chance to build our own workflow from scratch on a problem they would be tasked with, but by using open-source technologies only, and compare how our solution performed versus theirs. We learned much in this collaboration and have many ideas for next steps. We believe it is not an unreasonable decision to create your own analytics solution going completely open-source if you have the right talent to maintain it.

Acknowledgments

As aspiring data scientists, we would like to thank our industry partner and professor for providing us continuous guidance, support, and feedback as we learned more about designing and developing our own data science workflow for forecasting problems. This was a fantastic experience to collaborate with industry leading professionals, and get ideas on how to not only achieve better predictive performance, but also learn to prototype with the end goal deploying the solution as well. This project was partially funded by the Purdue BIAC.