Store Sales Forecasting Final Project
CS 4262-01 Foundations of Machine Learning, Spring

Jovian Wang [1]

Abstract

We constructed and compared a number of regression-based and deep learning models for the prediction of sales data from a large grocery retailer in Ecuador. Our analysis of the models suggests that LSTM-based multivariate models have high accuracy and perform better than regression models.

1. Introduction

In the highly competitive retail industry, accurate sales forecasting is often crucial for effective business planning and decision-making. Retailers need to predict their future sales in order to manage inventory, plan marketing campaigns, optimize pricing strategies, and make other strategic decisions that can impact their bottom line. Accurate sales forecasting can also lead to greater competitiveness, enhanced channel relationships, and customer satisfaction (Moon, Mentzer, & Smith, 2003). However, sales forecasting is an especially complex and challenging task, as it depends on a wide range of internal and external factors such as seasonal trends, level and trend shifts (Fildes & Beard, 1992), promotions, economic conditions, and consumer behavior.

A frequent approach to this problem is to use regression models that leverage historical information to formulate causal sales-related models (Fildes et al, 2008). However, these methods may be inaccurate and typically require a high number of explanatory variables; for example, the PromoCast regression-bases system by Cooper et al (1999) was introduced with 67 variables.

In recent years, machine learning has emerged as a promising alternative approach to sales forecasting, offering the potential to generate more accurate predictions than traditional statistical methods. Deep learning models are generally effective at capture complex patterns and relationships in large data sets; when working with sequential or temporal data, models that use recurrent neural networks are especially so. These methods, when applied to sales and business data, may allow retailers to make better-informed decisions and respond more quickly to changing market conditions. These models may be particularly important in the retail industry, where the complexity of sales data often requires sophisticated modeling approaches.

In this project, we aim to develop and compare various regression and deep learning models for predicting store sales. We evaluate the performance of these models using aggregate store sales data from Corporación Favorita, a large Ecuadorian-based grocery retailer. Our project and analysis provides valuable insights for retailers looking to leverage machine learning-based sales forecasting models in their own organizations.

2. Methods

We constructed eight models using more than four years of historical data to forecast aggregate sales per day for fifteen days.

2.1. Data Set, Pre-processing, and Features

This project uses store sales data from Corporación Favorita, as well as data sets on oil price and holidays, which are all made available by Kaggle through their "Store Sales - Time Series Forecasting" competition. The sales data and oil price data are temporal and provide the corresponding information by day from 2013-01-01 to 2017-08-15.

The sales data originally provided information on the sales (`sales`) and number of products on promotion (`onpromotion`) for each product by store by day. For the purposes of this project, it was aggregated such that each observation corresponds to the arithmetic mean of all `sales` and all `onpromotion` by day.

The reasoning for the relevancy of oil prices to sales is that Ecuador's economic health has been historically vulnerable to shocks in oil prices. The original oil data set contained 525 missing values. Due to the large, dense, and complex nature of the data, we chose to impute these missing values

[1]Vanderbilt University, Nashville, Tennessee, United States of America. Correspondence to: Jovian Wang <jovian.l.wang@vanderbilt.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

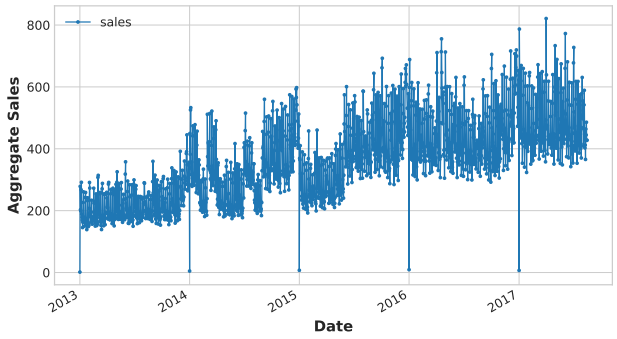

Figure 1. All aggregated sales by date.

using the respective queries to a locally weighted linear regression model with $\tau = 1$. Oil prices `oil_price` were then joined on (`sales`) and (`onpromotion`) by day.

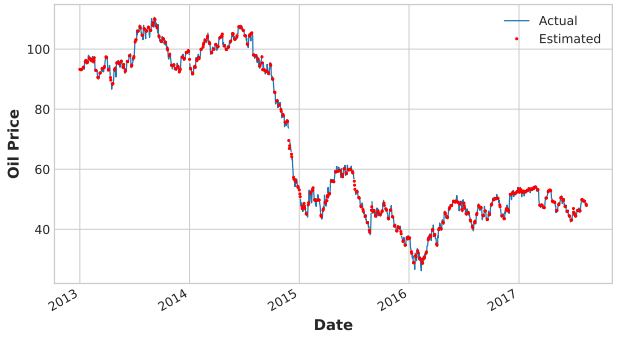

Figure 2. Final combination of actual and locally weighted linear regression-estimated oil prices by date.

The data set on holidays comprises observations for all festivities across Ecuador, encompassing local, state, and national holidays, which are further classified into several types including "holiday", "additional", "bridge", "work day", and "event". Information on national holidays was joined onto the rest of the data by their corresponding date. The features extracted thus far are `sales`, `onpromotion`, `oil_price`, and `holiday`.

A total of eight models were produced in this project; two use regression and six are neural networks. The models in each category differ from each other mostly in inputs. Each model is trained and validated on data between 2013-01-01 and 2017-07-31, and tested on sales between 2017-08-01 to 2017-08-15. There are a total of 1684 observations; 1669 in the training and validation set and 15 in the test set. Similar to the Kaggle competition, models in this project are evaluated using root mean squared logarithmic error (RMSLE). This error is calculated as:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(1+\hat{y}_i) - \log(1+y_i))^2}$$

where n is the number of observations, $\hat{y}_i$ is the predicted value of the target for instance i, and y_i is the actual value of the target for instance i.

2.2. Regression Approaches

Two regression models were constructed in this project and differ in input features.

2.2.1. Model One

The first regression model constructed in this project runs multivariate regression over the features described thus far, along with time step features that indicate the number of days passed. Due to its small number of levels, the nominal `holiday` data was encoded with one-hot encoding. This process produces new features `holiday_Additional`, `holiday_Bridge`, `holiday_Event`, `holiday_Holiday`, and `holiday_Work Day`, which contain values {0, 1}, where 1 indicates that the day corresponds with that type of holiday. We also considered polynomial time step features with $d = 3$; that is, the time step features that we considered for this model are `day` (the number of days since 2013-01-01), `day`^{2}, and `day`^{3}.

We used nested 5-fold cross validation, where each possible combination of features is cross-validated, to determine that the combination of features that results in the smallest overall RMSLE is {`day`, `day`^{3}, `holiday_Additional`, `holiday_Bridge`, `holiday_Event`, `holiday_Work Day`}. Our first regression model runs on these features.

2.2.2. Model Two

The second regression model was constructed to consider seasonality as well. We produced a periodogram that visualizes the strength of each length of frequencies in sales;

Table 1. Coefficients of Regression Model 1.

Feature	Coefficient
intercept	173.53471812
day	2.39480320e-01
day^3	-1.79018203e-08
holiday_Additional	1.54674479e+02
holiday_Bridge	2.45854407e+01
holiday_Event	3.63230673e+01
holiday_Work Day	7.33666134e+01

an inspection of this periodogram reveals some annual seasonality and great weekly seasonality. To account for the weekly sales seasonality, we used one-hot encoding to indicate day of the week, the same process as the one applied to holiday data, creating features `dayofweek_Tuesday` to `dayofweek_Sunday`. Note that we dropped one encoded feature `dayofweek_Monday` again to avoid multicollinearity. To account for the annual sales seasonality, we used a Fourier feature pair; this pair consists of a sine and a cosine curve, each with a frequency of one year. This creates features `sin` and `cos`. Our second regression model considers these new features in addition to those considered by the first model.

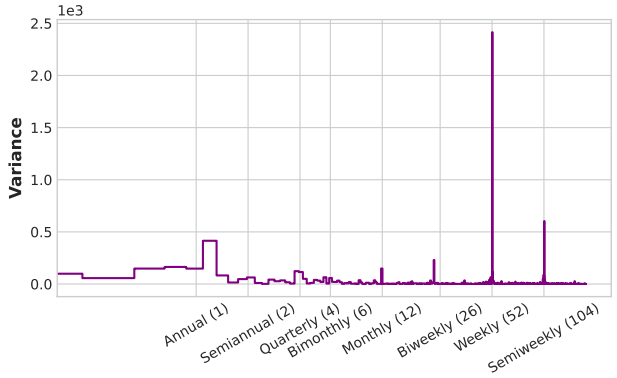

Figure 3. Periodogram of variance in sales over various frequencies. Spikes over annual and weekly frequencies indicate some annual seasonality and great weekly seasonality. Spikes over bimonthly and monthly frequencies are likely a result of the weekly seasonality.

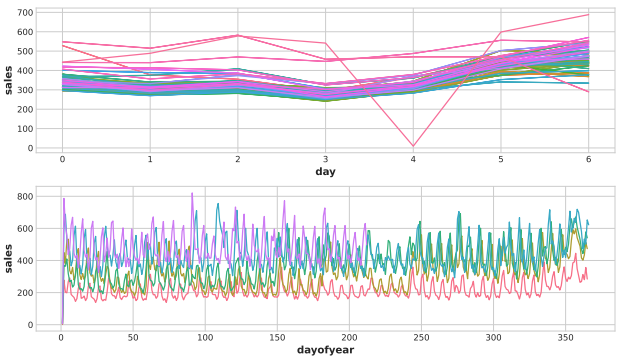

Figure 4. A seasonal plot of the sales per day of each week (top) and a seasonal plot of the sales per day of each year (bottom).

Table 2. Coefficients of Regression Model 2.

FEATURE	COEFFICIENT
INTERCEPT	167.07334342
DAY	2.28681418E-01
DAY^3	-1.33528775E-08
HOLIDAY_ADDITIONAL	1.36371408E+02
HOLIDAY_BRIDGE	3.86236004E+01
HOLIDAY_EVENT	3.57594909E+01
HOLIDAY_WORK DAY	-1.85224932E+01
DAYOFWEEK_TUESDAY	-2.62330751E+01
DAYOFWEEK_WEDNESDAY	-1.30823527E+01
DAYOFWEEK_THURSDAY	-6.18081386E+01
DAYOFWEEK_FRIDAY	-2.00654221E+01
DAYOFWEEK_SATURDAY	8.56584037E+01
DAYOFWEEK_SUNDAY	1.18093864E+02
SIN	-1.16258127E+01
COS	1.56286014E+01

2.3. Deep Learning Approaches

Each model hereafter is a long short-term memory network (LSTM) as described by Hochreiter and Schmidhuber (1997). We chose to use LSTMs for this case since it is a common recurring neural network (RNN). RNNs are neural networks with cyclic node connections and exhibit temporal dynamic behavior. They are used on sequential or, such as in this case, time series data.

2.3.1. MODEL THREE

For best results, we scaled each feature to be over [0, 1]. The time series was then transformed for supervised learning by making each observation include 29 lags per time series feature (feature values from 29 days ago to the current day) as well as the target feature 15 days after the current day. We decided to make the first LSTM univariate. This means the model input is composed of only `sales(t-29)` to `sales(t)` and the model's target is `sales(t+15)`, where `t` indicates the feature on that day. After dropping missing values, this results in a training and validation input set of 1625 observations, 1 time series feature, and 30 lags (or an input size of 1625 by 30 by 1).

Similar to all of the following models, model 3 has a 256-node LSTM layer followed by a series of densely-connected layers. It is run with a validation split of 0.2, Adam optimization, and a early stop callback that monitors for stagnation in validation loss with a patience of 10 epochs. In total, model 3 was trained with 76 epochs.

2.3.2. MODEL FOUR THROUGH SEVEN

The following models are multivariate. We first considered the addition of `oil_price` as another time series feature to our fourth model, resulting in an input set of 2 time series

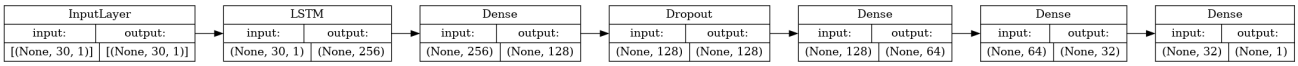

Figure 5. The structure of model 3. Models 4-7 are very similar but have different LSTM input shapes to accommodate multiple time-series features.

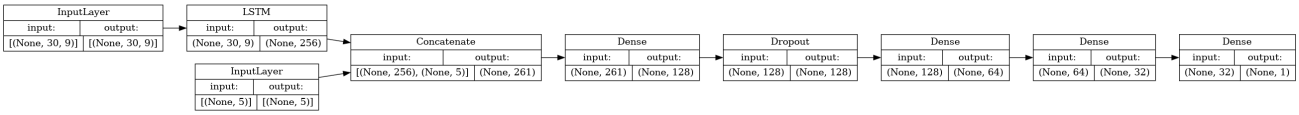

Figure 6. The structure of model 8. This model is similar to the other LSTM models but includes an auxiliary input that is concatenated to the result of the LSTM layer. This allows the model to consider non-sequential features, like whether or not the target date is a holiday, without changing the efficacy of the LSTM layer.

features and 30 lags (1625 by 30 by 2). This model is run with a validation split of 0.2, Adam optimization, and the same callback as before. In total it was trained with 14 epochs.

We found that model 4 resulted in a better training RMSLE than model 3. Therefore, we proceeded to add encoded day of the week as a time series feature on top of previous sales and oil price. This makes model 5's input set 8 time series features and 30 lags (1625 by 30 by 8). Model 5 trained with the same parameters, for a total 61 epochs.

Again, model 5 resulted in a better training RMSLE than model 4, and we proceeded to add `onpromotion` onto these features to produce model 6. This makes the input set 9 time series features and 30 lags (1625 by 30 by 9). Model 6 trained with 110 total epochs.

Model 6 had a better training RMSLE than model 5, so we added our annual time step features `sin` and `cos` to make model 7. The input set was 1625 by 30 by 11. Model 7 trained with 68 total epochs. We found that this model actually had a higher training RMSLE than previous models, so, according to the training and validation set, this model performed worse than model 6.

2.3.3. Model Eight

The final model constructed in this project considers past sales, `oil_price`, day of the week, and `onpromotion`, similar to model 6. However, we also added encoded holiday features as non-time series auxiliary features, which is put into a separate input than the time series features. For model 8, the time series features are processed by the LSTM layer, and then auxiliary holiday data, which describe whether or not the target date is a holiday, is concatenated to the LSTM output vector. The combination is thereafter fed into the densely connected layers. This model was ran with the same validation split, optimizer, and callback as before, and trained a total of 81 epochs.

3. Results

Learning curves, forecasts, and losses were visualized.

Table 3. Training and Test Loss of Each Model

Model Number	Training Loss (RMSLE)	Actual Test Loss (RMSLE)
1	0.319069	0.146351
2	0.274493	0.129491
3	0.312394	0.139087
4	0.312053	0.138109
5	0.256547	0.124266
6	0.250435	0.120183
7	0.254194	0.120615
8	0.248885	0.114729

4. Discussion

The first model, which uses multivariate regression on time step features and holiday data, yields a training RMSLE of 0.319059 and a test RMSLE of 0.146351. This means that, when measured on a logarithmic scale, the average error between the predicted and actual sales is about 31.9% and the average error between the forecasted sales and the actual test sales is about 14.6%. Despite using the features that yielded the best results according to the nested cross validation that we performed, a quick examination of the losses with those of the rest of the models suggests that this model is the least accurate of all eight; it has both the highest training loss as well as the highest actual test loss. The visualization of its predictions in figure 8 reveals that this model is able to capture the general upward trend of sales across the range of days as well as the occasional impact of holidays, but is unable to describe more of the

variance in day-to-day sales.

This result aligns with the motivation for creating the second regression model. Model 2 results in an training RMSLE of 0.274493; not only does it have a smaller training loss than model 1, but it also actually has a smaller testing loss (0.129491). It follows that the average error of its forecasts is around 12.9% when measured on the logarithmic scale. This is already a huge improvement to model 1, and the visualization of its predictions shows that this is likely due to its ability to really leverage weekly and annual patterns in addition to the overall upward trend. It is also interesting to note that the coefficients related to day of the week and ordinal date are also magnitudes larger than those of time step features, showing the great amount of variance introduced by these seasonal patterns.

Model 3 introduces a deep learning approach and results in a training RMSLE of 0.312394 and an actual test loss of 0.139087. It appears to be capturing a decent portion of the patterns in sales, and despite being univariate and only requiring historical sales data, it is already outperforming model 1, which was trained using sales, time step features, and holiday data. In addition, it seems that this model weighs annual seasonality far less in its predictions than model 2.

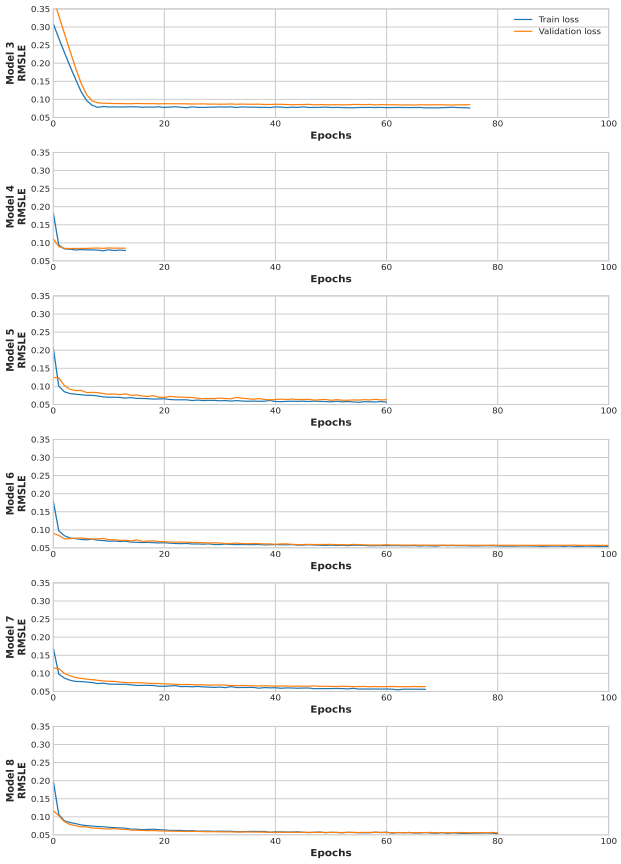

Figure 7. Learning curves for each LSTM model. Blue lines are training loss, yellow lines are validation loss. Models were trained for varying number of epochs since they were automatically stopped by a callback that monitored validation loss.

Model 4 introduces another time series feature, oil price, for the LSTM neural network to process. Its performance was only a very slight improvement in both training loss and actual test loss over that of model 3, but it trained for a total of 14 epochs, 62 epochs less than model 3. Overall, it appears that including this relevant time series feature not only improved LSTM performance, but also drastically improved training time for the model to create similarly meaningful outputs.

Model 5 re-adds weekly seasonality into consideration, which, similar to model 2, drastically improves performance, resulting in a training RMSLE of 0.256547 and a test loss of 0.124266; both losses are actually less than those achieved by model 2. From figure 9, it appears to also recognize and really leverage weekly patterns. Model 6 adds another time step feature and achieves another slight increase in accuracy. Annual seasonality was then re-added in model 7 via the Fourier curves, but we saw that performance actually decreased, so Fourier features were dropped from the next model.

Model 8 utilizes historical sales, oil price, promotion data, and weekly seasonality time series features like model 6, but is also able to consider holiday data. In return for a slightly more complex design, this model achieves a better training loss at 0.248885 RMSLE and a test loss at 0.114729; on a logarithmic scale, the average error of its forecasts from the actual test sales values are about just 11.4%. Furthermore, its predictions and forecasts visually appear to more closely match the changes in sales, as seen in figure 9. It also required 81 epochs to train, which, while comparatively on the higher side, is not entirely an unacceptable tradeoff for this increase in performance.

5. Conclusion

From our results, we conclude that an LSTM-based multivariate deep learning model that can account for non-time series auxiliary data is able to perform the best for forecasting sales time series data, and is able to do so with an acceptable training time. After comparison between the deep-learning models, we also conclude that, generally, including more of relevant time-series features may not only improve accuracy, but also improve training time, such as the case with model 4. However, this is not always the case, as with model 7, whose inclusion of Fourier features actually led to an increase in loss.

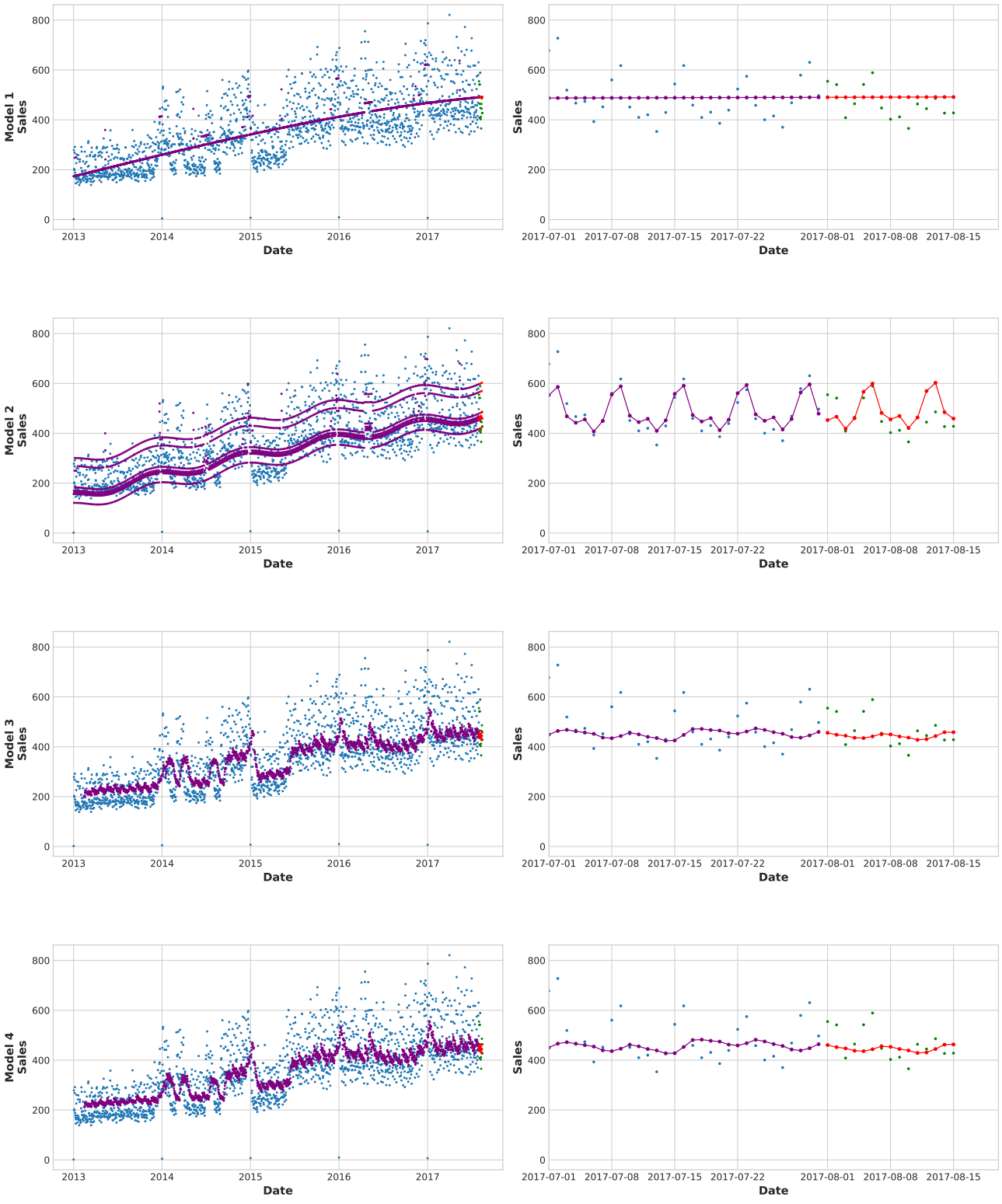

Figure 8. Predictions and forecasts of models 1 through 4. Blue dots represent actual aggregated sales per day in the training & validation set. Green dots represent actual aggregated sales per day in the test set. Purple dots represent predicted sales. Red dots represent forecasted sales onto the test dates. Predictions and forecasts over the entire dataset are visualized on the left, while a more detailed look at predictions and forecasts toward the end of the data are on the right.

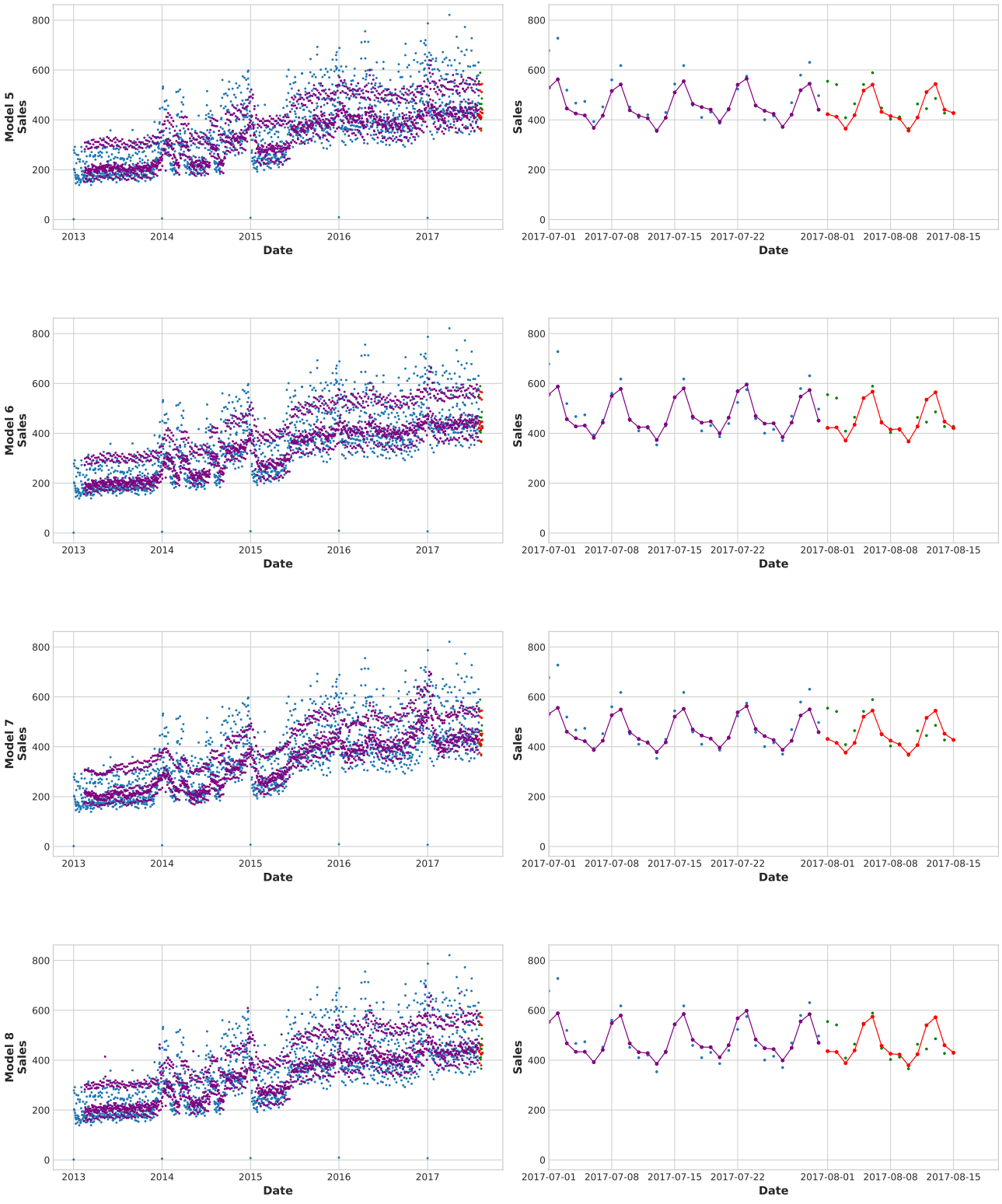

Figure 9. Predictions and forecasts of models 5 through 8. Blue dots represent actual aggregated sales per day in the training & validation set. Green dots represent actual aggregated sales per day in the test set. Purple dots represent predicted sales. Red dots represent forecasted sales onto the test dates. Predictions and forecasts over the entire dataset are visualized on the left, while a more detailed look at predictions and forecasts toward the end of the data are on the right.

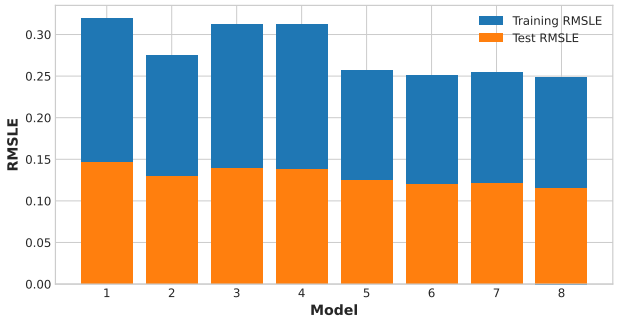

Figure 10. Losses per model.

If we only consider regression models, then regression models that not only account for overall trends with time step features but also account for seasonal patterns with period encodings and Fourier curves are able to perform substantially better than those that do not. These regression models that account for seasonality actually are also able to have adequate performance overall, even when compared to the deep learning models; for example, model 2 had an actual test RMSLE that was only about 0.015 greater than the RMSLE of model 8.

When comparing between the two model types, not only did the best LSTM model perform better than the best regression model, the worst LSTM model performed also better than the worst regression model. Model 3, which only accounted for past sales, still had slightly smaller errors than model 1. Therefore, overall, we would argue that deep learning/LSTM approaches are better. However, it is important to note that neural networks do have slightly longer training times than regressors, and that the overall model for neural networks are less interpret-able due to their hidden states.

We note that in this project, the deep learning approaches that we used only included LSTMs. Extending this study to other recurrent neural network structures, tree-based models, or other deep learning approaches in general may be beneficial. In addition, further research may include more data samples, such as sales data from multiple other retail corporations, or sales data per store, in order to provide a more general understanding in the performances of these forecasting approaches.

6. Acknowledgments

The author produced this work independently. However, shout-out to Kobe Guo, who was once part of the team a long, long time ago...

7. References

Cooper, L. G., Baron, P., Levy, W., Swisher, M., & Gogos, P. (1999). PromoCast™: A new forecasting method for promotion planning. *Marketing Science, 18*(3), 301–316. https://doi.org/10.1287/mksc.18.3.301

Fildes, R., & Beard, C. (1992). Forecasting systems for production and inventory control. *International Journal of Operations & Production Management, 12*(5), 4–27. https://doi.org/10.1108/01443579210011381

Fildes, R., Nikolopoulos, K., Crone, S. F., & Syntetos, A. A. (2008). Forecasting and operational research: A Review. *Journal of the Operational Research Society, 59*(9), 1150–1172. https://doi.org/10.1057/palgrave.jors.2602597

Hochreiter, S., & Schmidhuber, J. (1996). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kaggle. n.d. *Store Sales - Time Series Forecasting*. https://www.kaggle.com/competitions/store-sales-time-series-forecasting/

Moon, M. A., Mentzer, J. T., & Smith, C. D. (2003). Conducting a sales forecasting audit. *International Journal of Forecasting, 19*(1), 5–25. https://doi.org/10.1016/s0169-2070(02)00032-8