
Forecasting Future Sales: A Time Series Analysis Utilizing Deep Neural Networks

Montanna Colburn

April 2019

In Collaboration With: Enterprise Training Solutions





Business Problem

Enterprise Training Solutions(ETS) is an e-learning company that sells government training programs for federal, state, local and educational institutions.

Though the company provides e-learning solutions, predicting future sales trends can be tremendously helpful for any business, tangible goods or not. Financial planning, product price points, marketing, hiring needs are just a few benefits from sales forecasting.



Research Questions

Can we predict future sales?

- What time series modeling will best suit our data?
 - How should we forecast, monthly/weekly/daily?
- What attributes can be used to aid in our prediction?
 - If using other attributes, how much of an impact do they have on the model?



Dataset

- The dataset is comprised of 9,977 observations by 23 columns **prior to cleaning*
- Attributes **not all 23 columns had inherent value and were removed off the bat*
 - Date (date of sale), Quantity (quantity sold), Type (type of sale), Name State (state where customer is located), Name (customer), Item (the product), Account (supplier), Class (customer type), Rep (sales representative), Sales Price (product price at sale), Amount (total sales amount), Account Type (payment account type), Balance (customers account balance)
- Target variable will be the Quantity of Sales



Pipeline

- Imports
- Data Cleaning/ Preprocessing
- Exploratory Data Analysis and Data Visualizations
- Time Series Analysis and Modeling
 - Monthly Aggregated Implementation
 - Daily Aggregated Implementation
- Deep Learning with RNN/ LSTM
 - Monthly Aggregated Implementation
 - Daily Aggregated Implementation
 - Daily Implementation with added Features
- Implementation of Gradient Boosted Decision Tree
 - Feature Analysis
- Final Outcomes
 - Interpretations and Limitations
 - Future Work

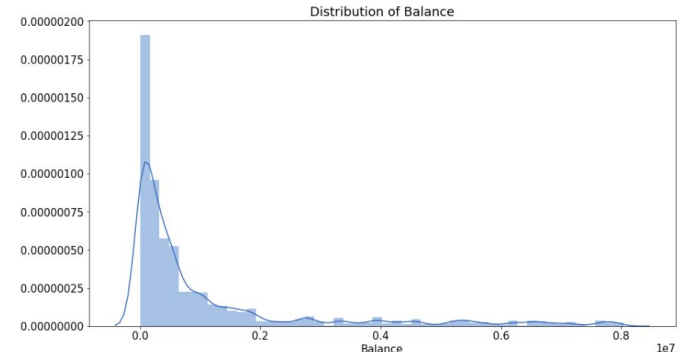
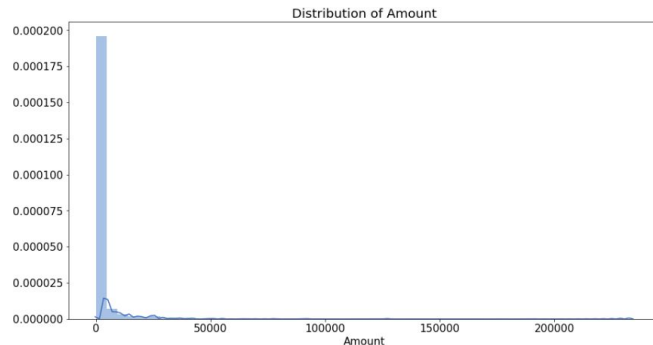
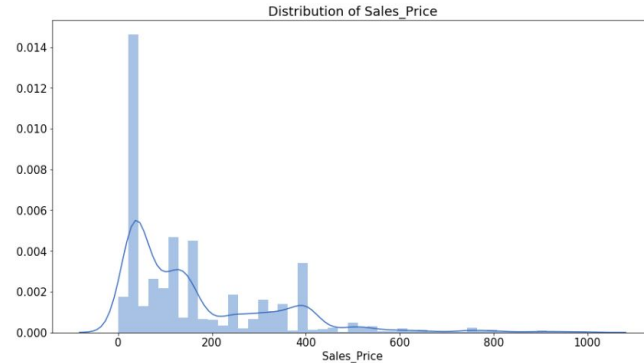
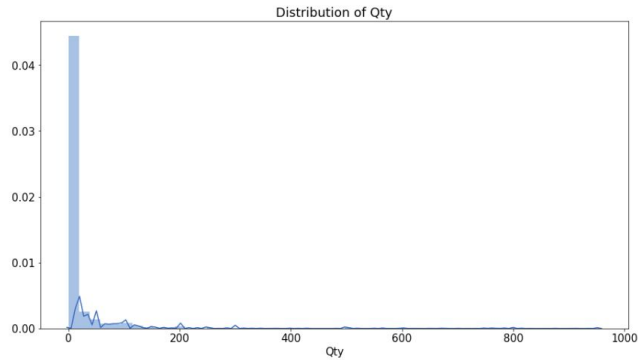


Post Data Cleaning

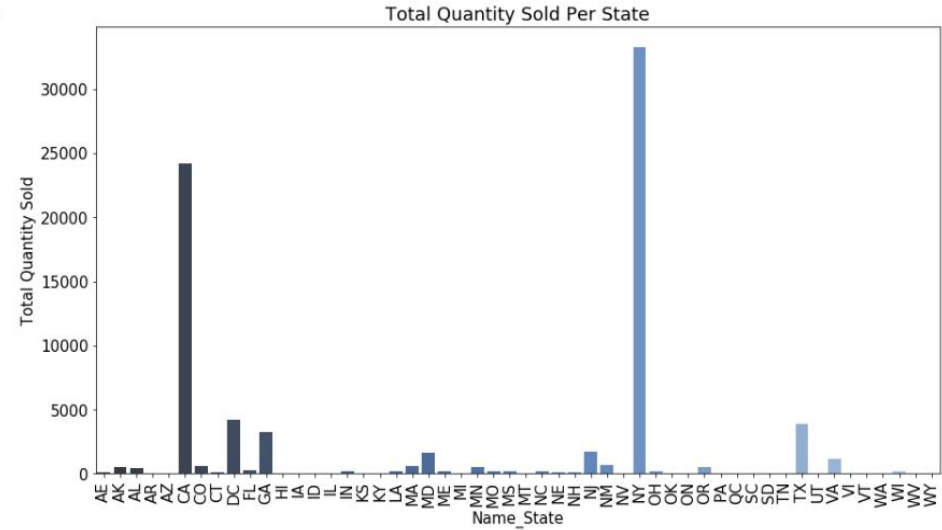
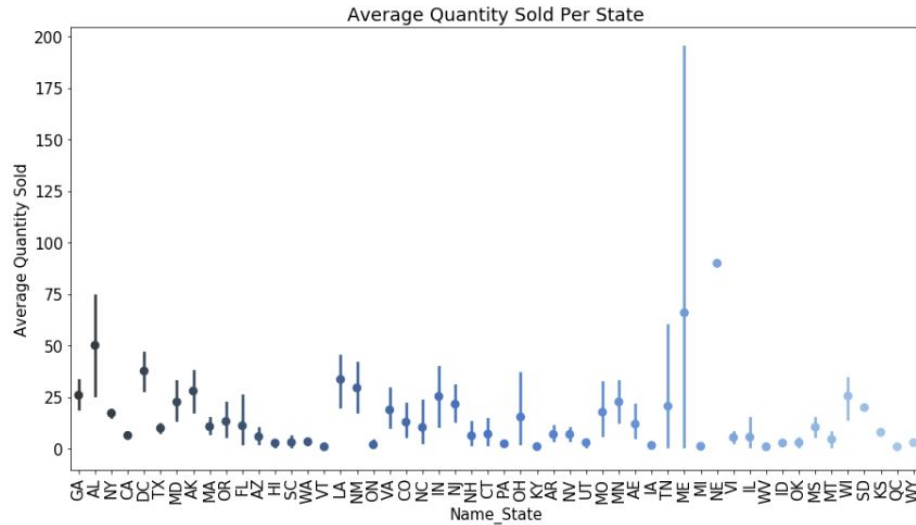
- 8,270 observations available for our Time Series Modeling
- 7,061 observations available for our multi-input Deep Neural Network and Gradient Boosted Model

Why are these different? Simply our RNN and GBM will include more features, and in the case of dropped null values, our observations become less.

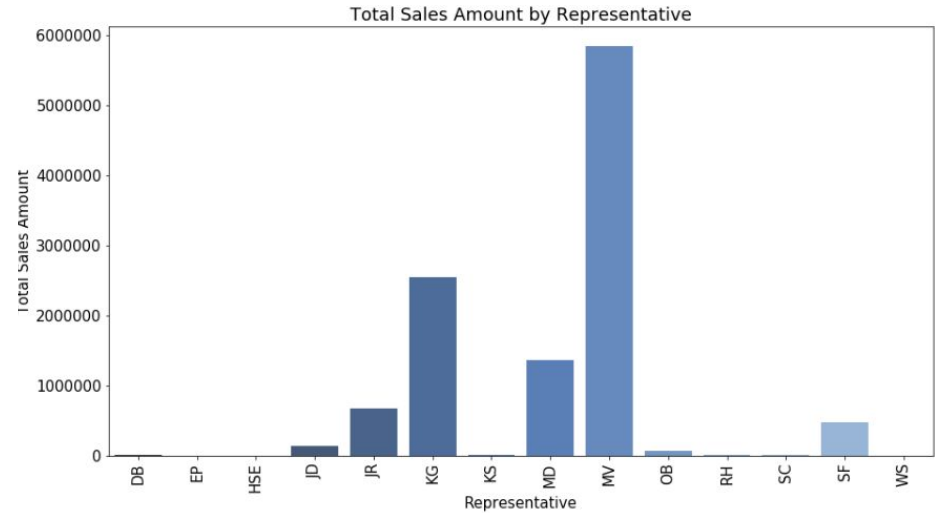
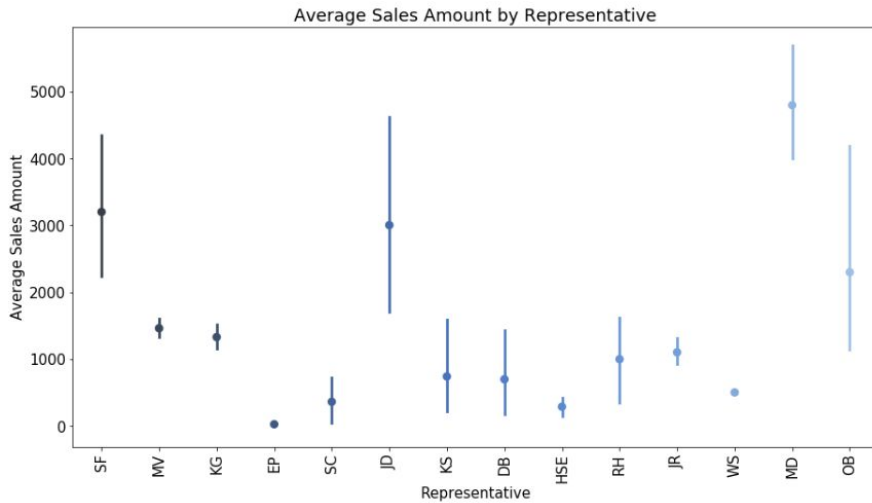
Data Visualization



Data Visualization

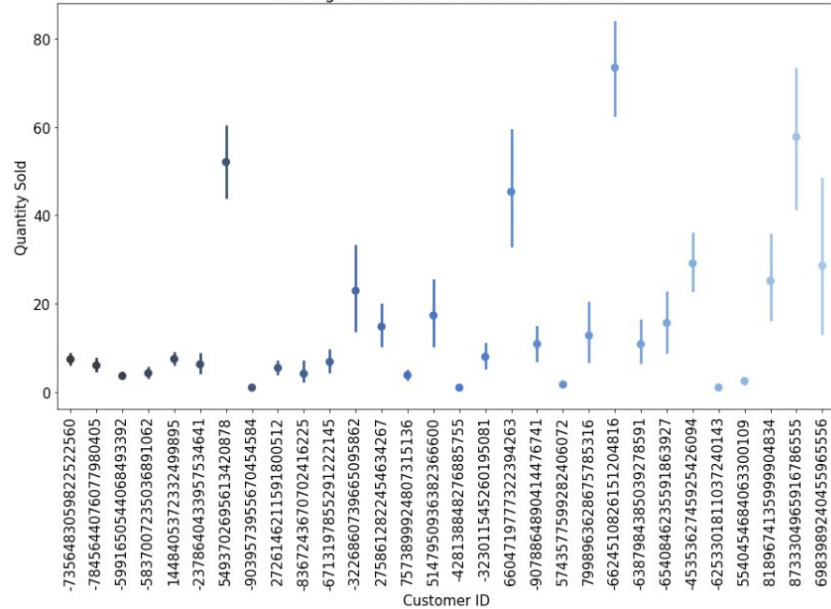


Data Visualization

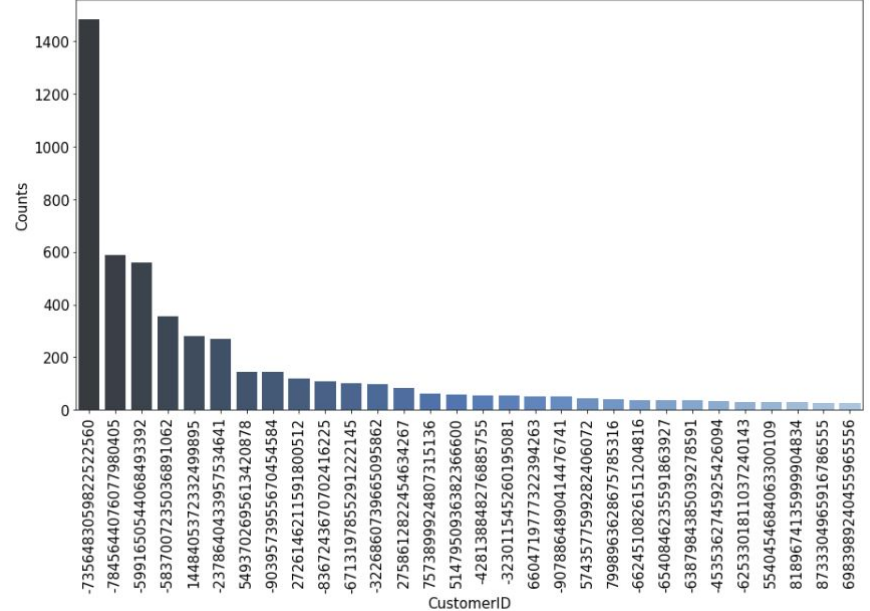


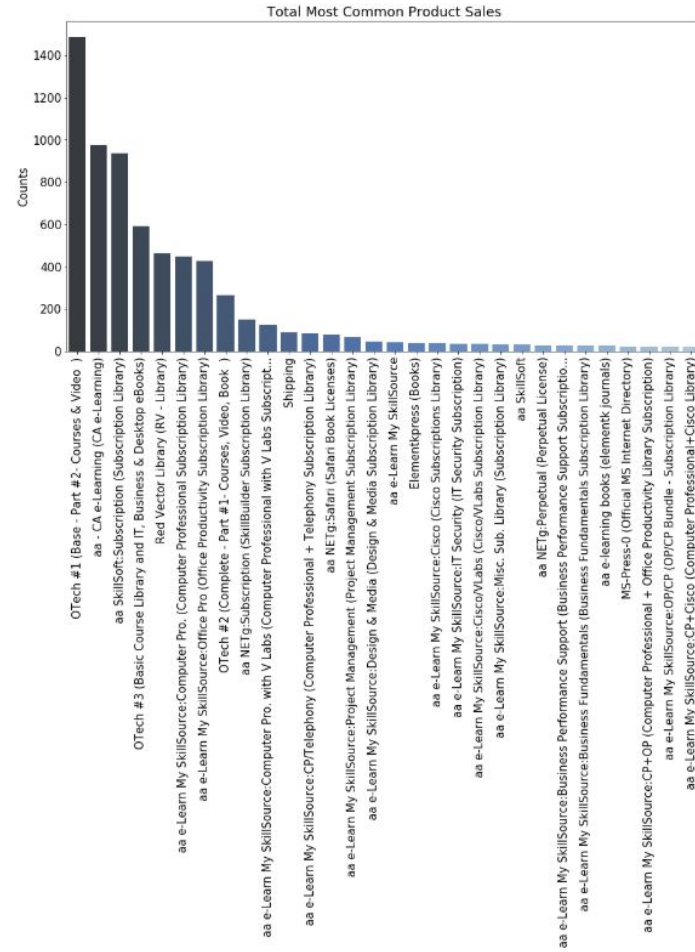
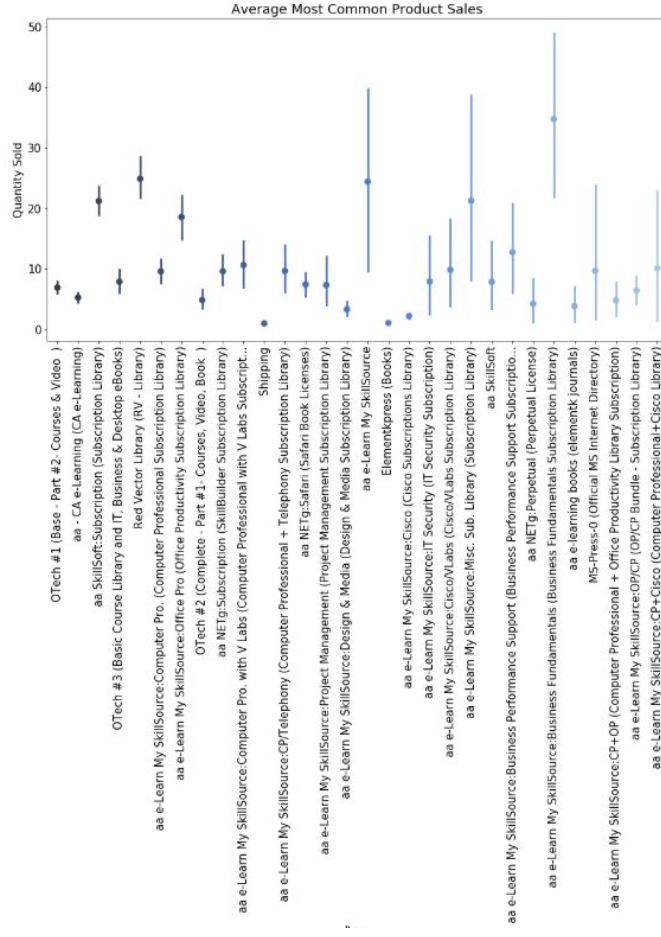
Data Visualization

Average Most Common Customers Sales



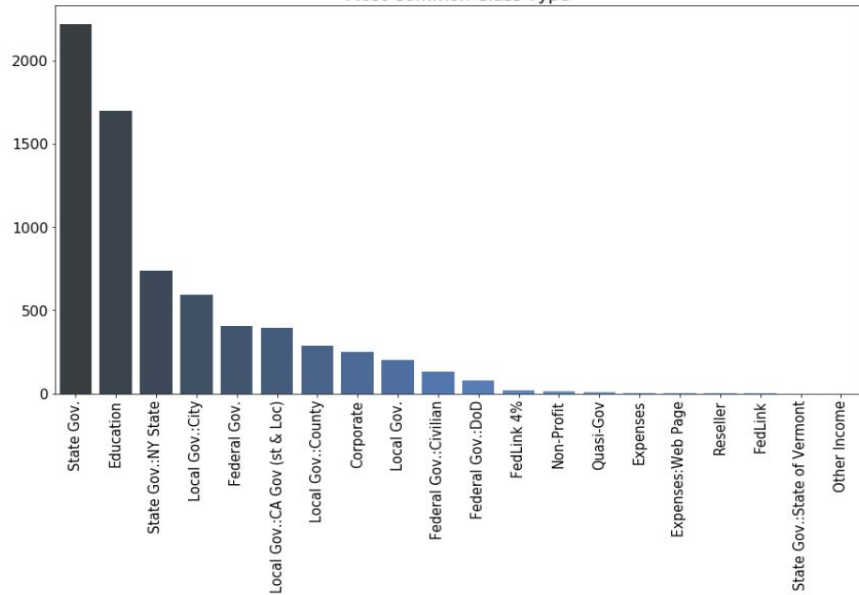
Total Most Common Customers Sales



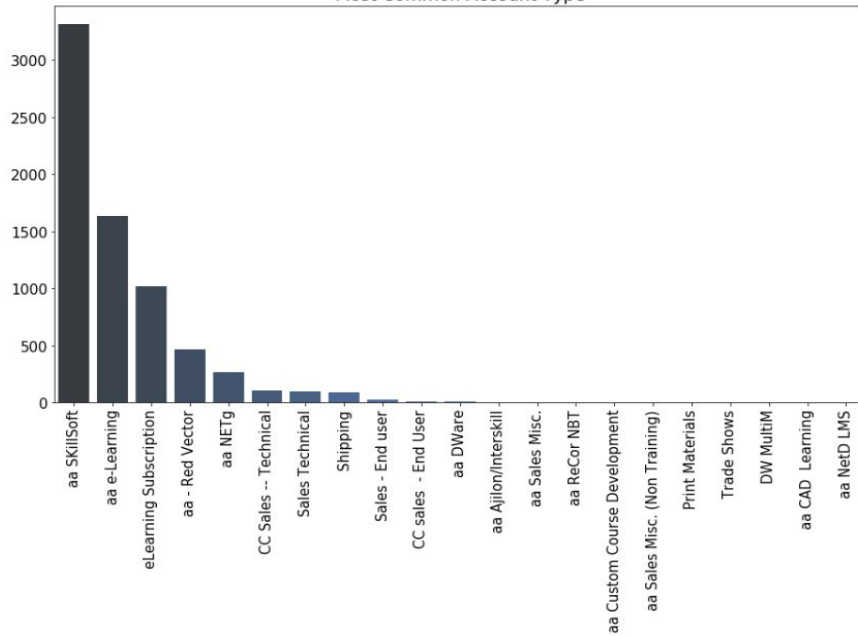


Data Visualization

Most Common Class Type



Most Common Account Type





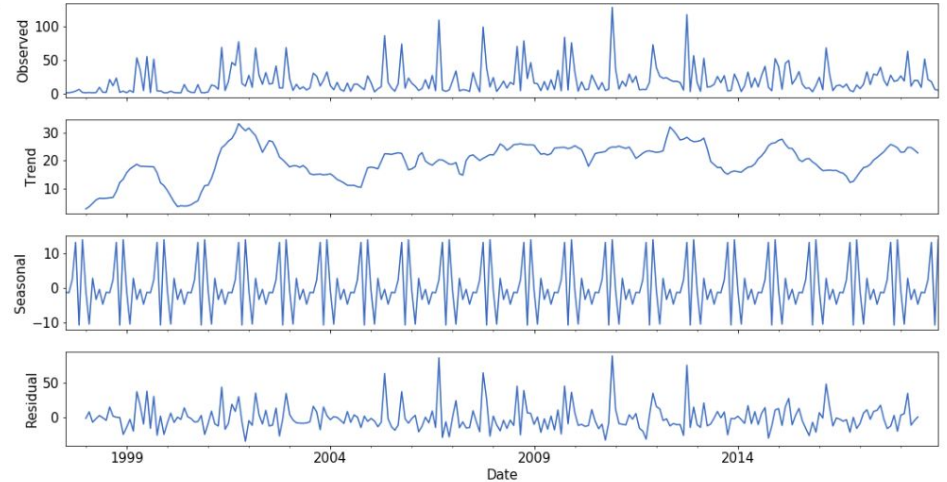
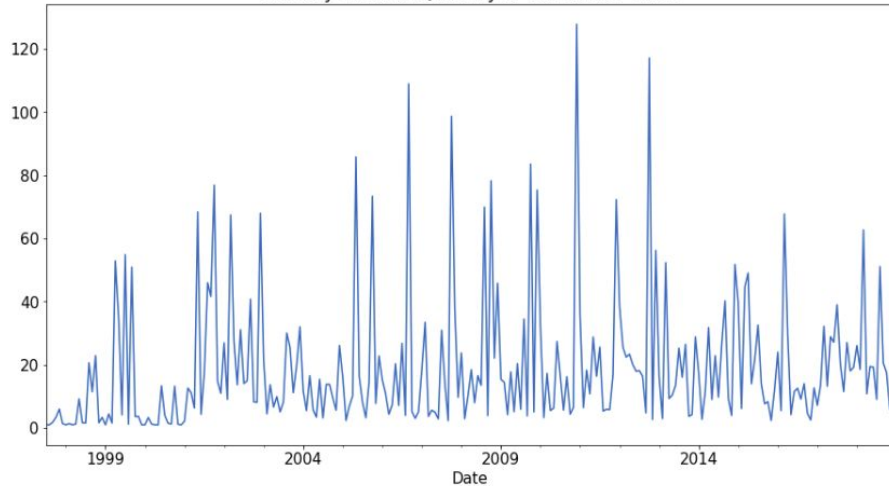
Time Series ARIMA Modeling

- Monthly Aggregated Implementation
- Daily Aggregated Implementation



Monthly ARIMA Prediction

Monthly Mean of Quantity of Sales Over Time

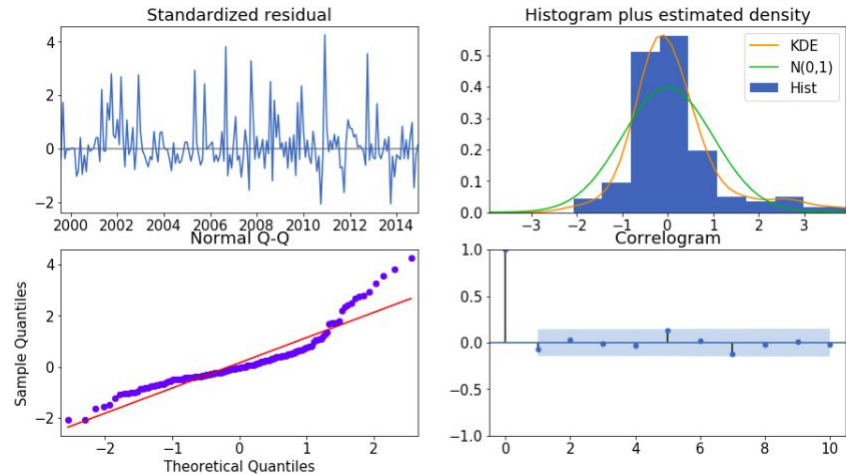


Monthly ARIMA Prediction

Second Best AIC Value: 2104.5863324367856.

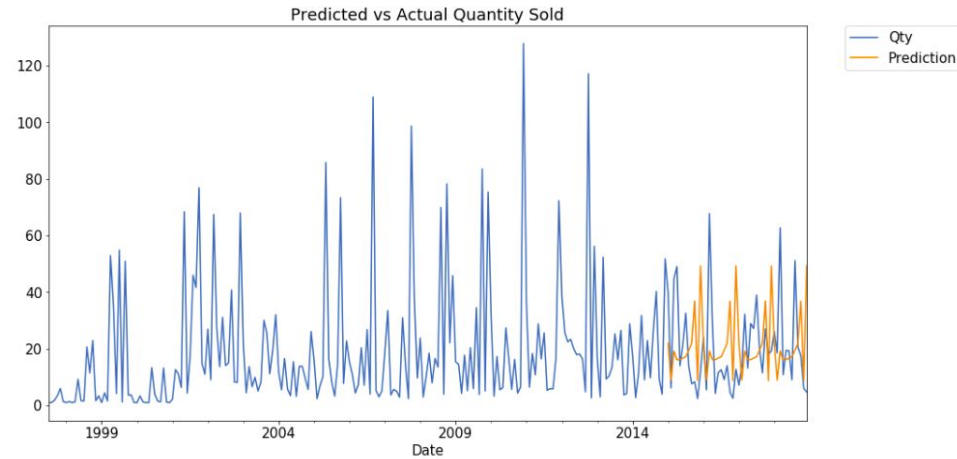
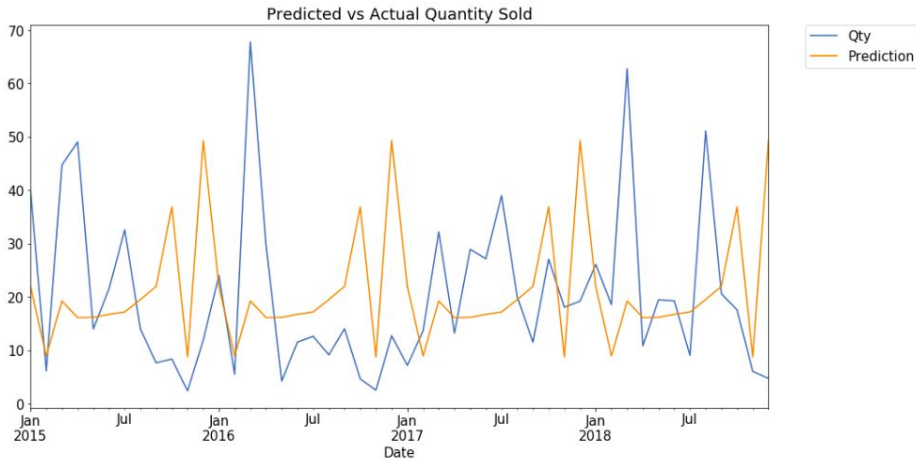
```
1 mod2 = sm.tsa.statespace.SARIMAX(train,
2                                     order=(0, 0, 0),
3                                     seasonal_order=(0, 1, 1, 12),
4                                     enforce_stationarity=False,
5                                     enforce_invertibility=False)
6 results2 = mod2.fit()
7 print(results2.summary().tables[1])
```

	coef	std err	z	P> z	[0.025	0.975]
ma.S.L12	-0.8299	0.040	-20.853	0.000	-0.908	-0.752
sigma2	511.2685	30.522	16.751	0.000	451.447	571.090



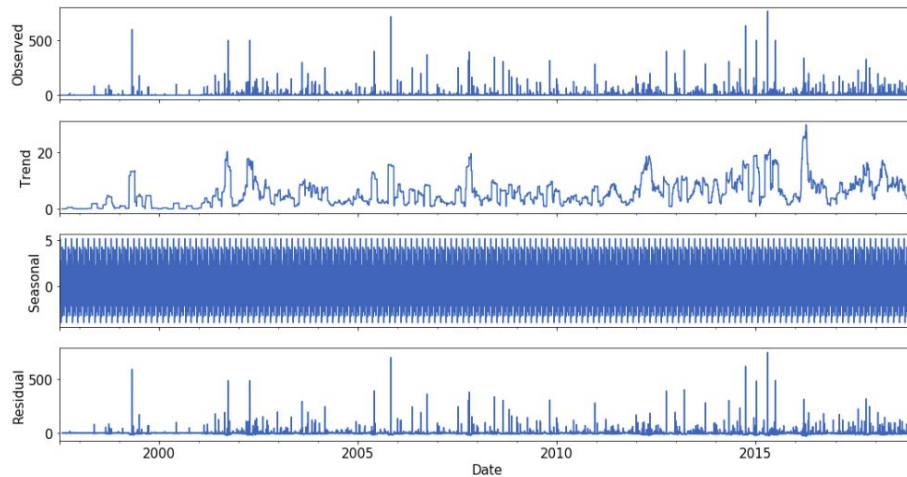
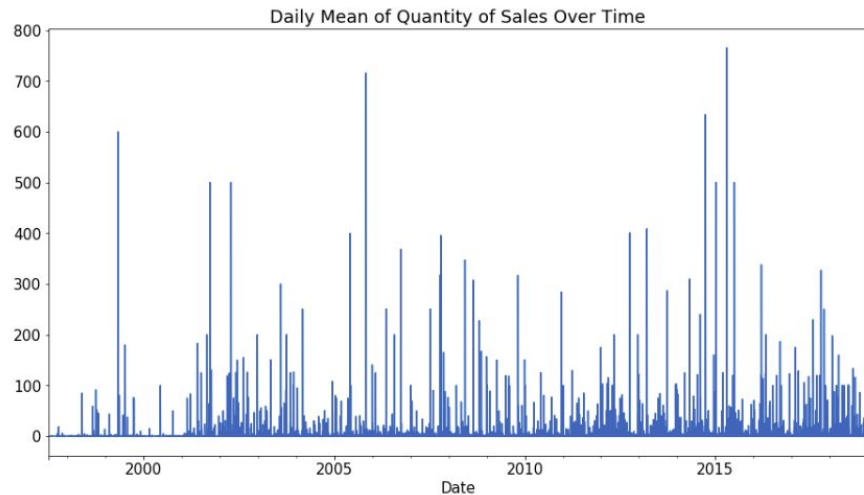
Monthly ARIMA Prediction

RMSE: 3.7986807331779495



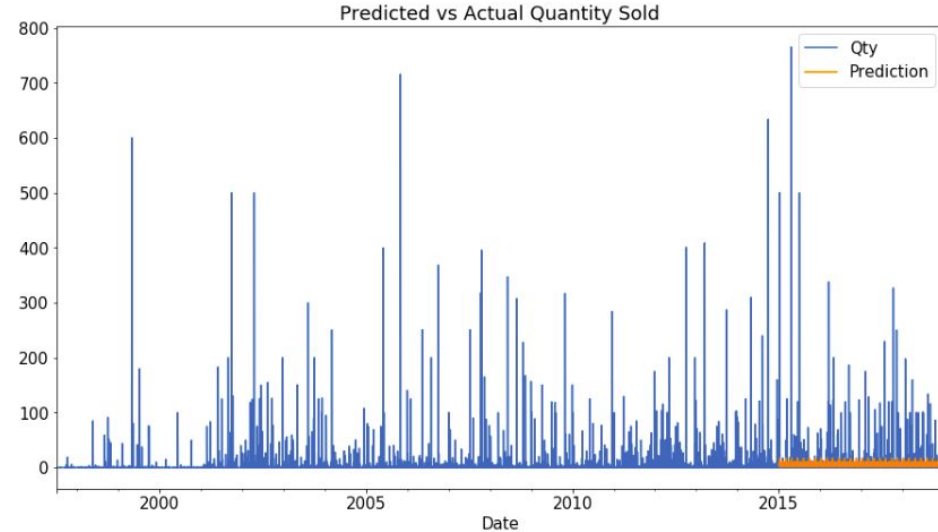
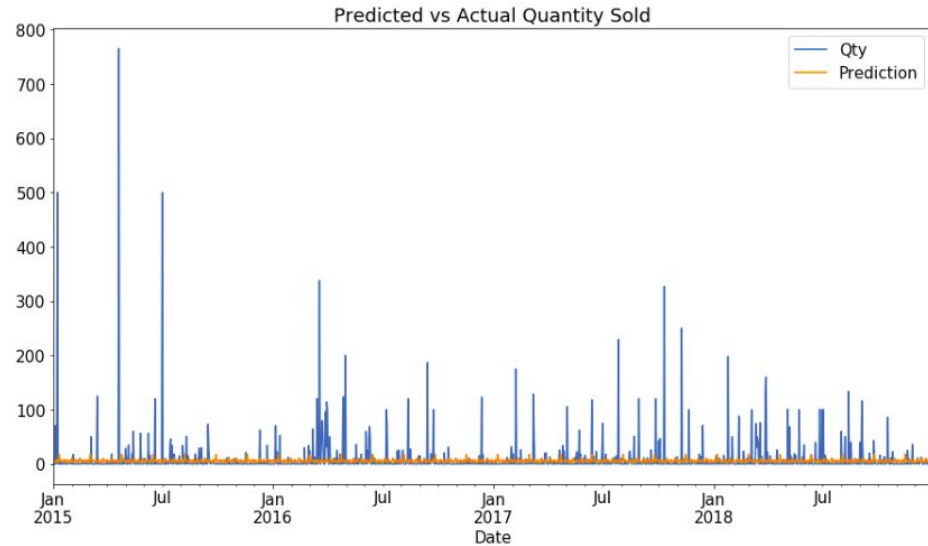
Daily ARIMA Prediction

- Downside of SARIMAX modeling is its incapability of memory storage to execute a 365 seasonality prediction. Instead, I used 52 for weekly seasonality, yet still ran into computation storage issues.



Daily ARIMA Prediction

RMSE: 3.39238159410763



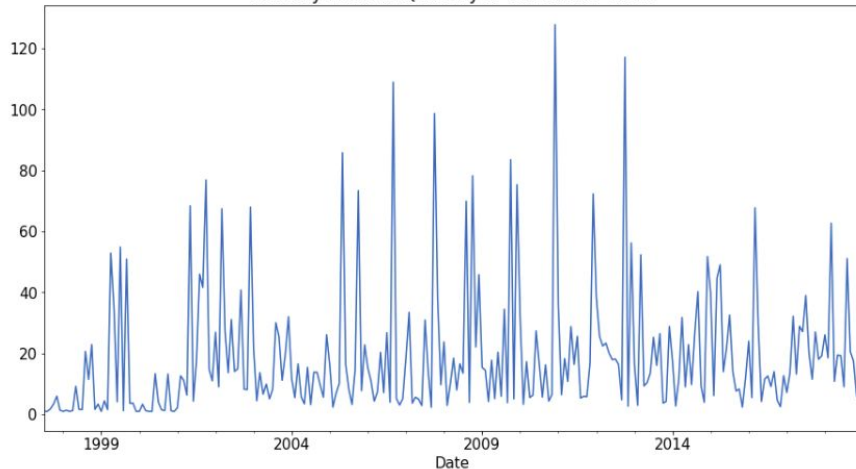


LSTM Modeling

- Monthly Aggregated Implementation
- Daily Aggregated Implementation
- Daily Implementation with added Features

Monthly LSTM Prediction

Monthly Mean of Quantity of Sales Over Time

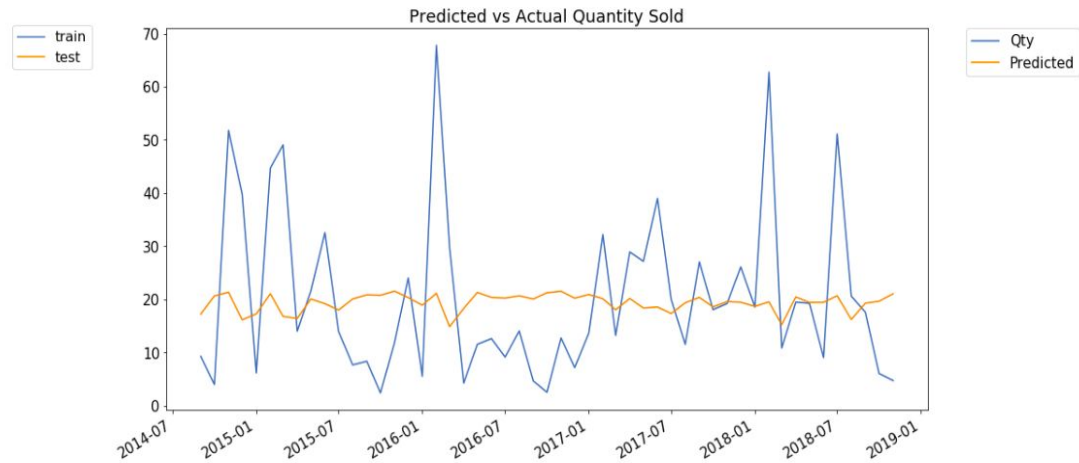
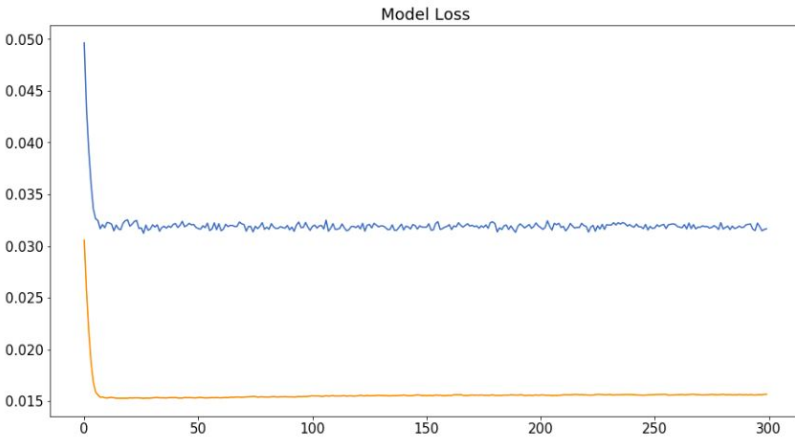


```
# The LSTM architecture
model = Sequential()
# First LSTM layer with Dropout regularization
model.add(LSTM(units=50, return_sequences=True, input_shape=(trainX.shape[1], trainX.shape[2])))
model.add(Dropout(0.2))
# Second LSTM layer
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
# Third LSTM layer
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
# Fourth LSTM layer
model.add(LSTM(units=50))
model.add(Dropout(0.2))
# The output layer
model.add(Dense(units=1))

# Compiling the RNN
model.compile(optimizer='rmsprop', loss='mean_squared_error')
# Fitting to the training set
history = model.fit(trainX, trainY, epochs=300, batch_size=100, validation_data=(testX, testY), verbose=0, shuffle=False)
```

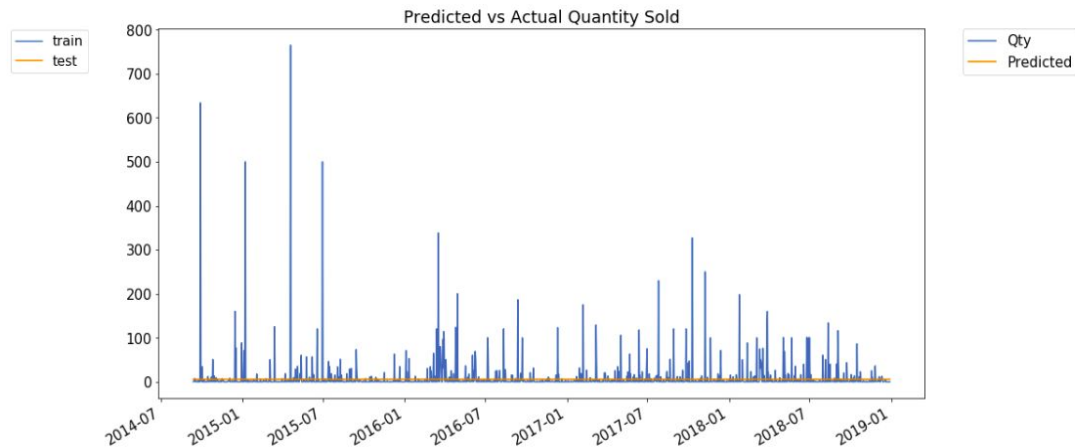
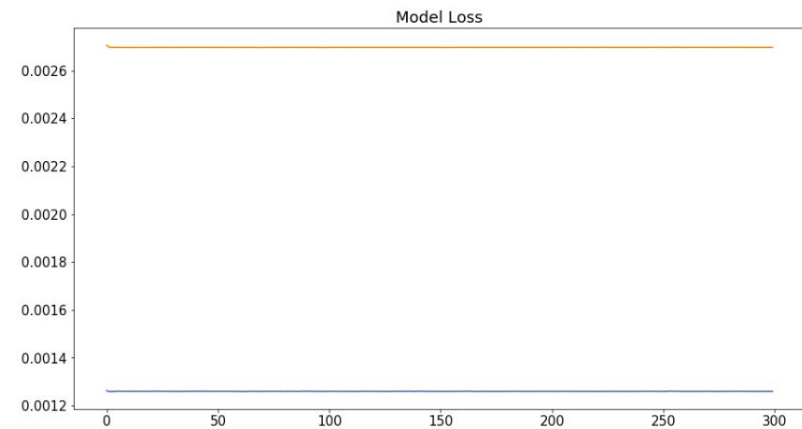
Monthly LSTM Prediction

RMSE: 15.835



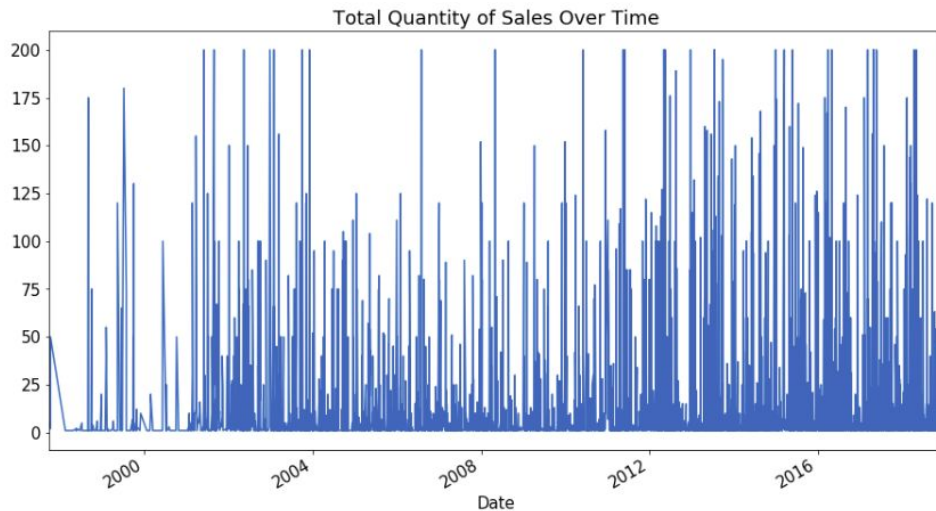
Daily LSTM Prediction

RMSE: 39.720



Daily LSTM Prediction: Added Features

- In our next series of predictions we incorporate known features, and thus, we're asking a new question; *can we predict the quantity sold in a given transaction?*



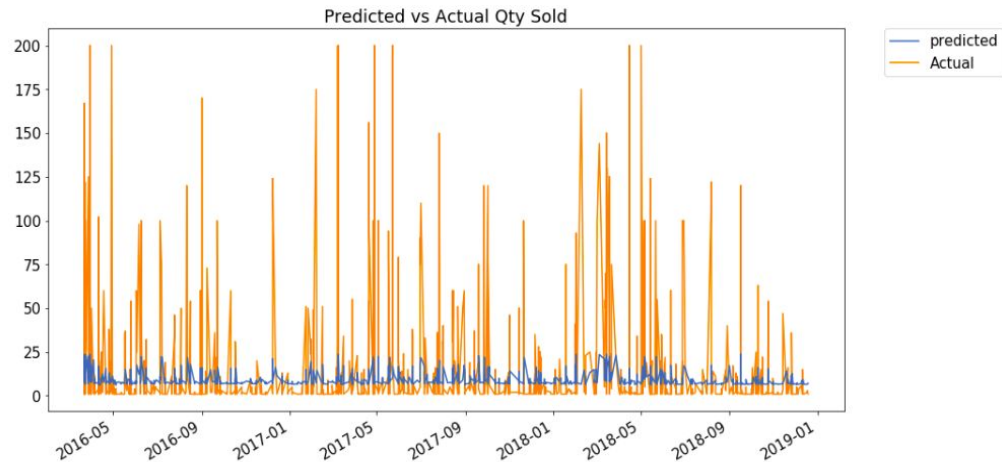
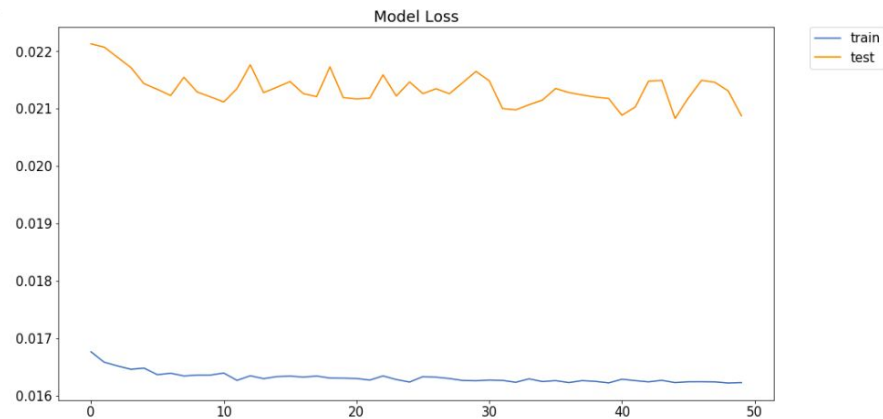


Daily LSTM Prediction: Added Features

- Label encode categorical features
 - Type: 2
 - Name State: 52
 - Item: 183
 - Account: 21
 - Class: 20
 - Rep: 14
 - Account Type: 3
- Lag variable for previous days quantity

Daily LSTM Prediction: Added Features

RMSE: 28.753





XGBoost Modeling

- Daily Implementation
- Feature Analysis



Feature Engineering

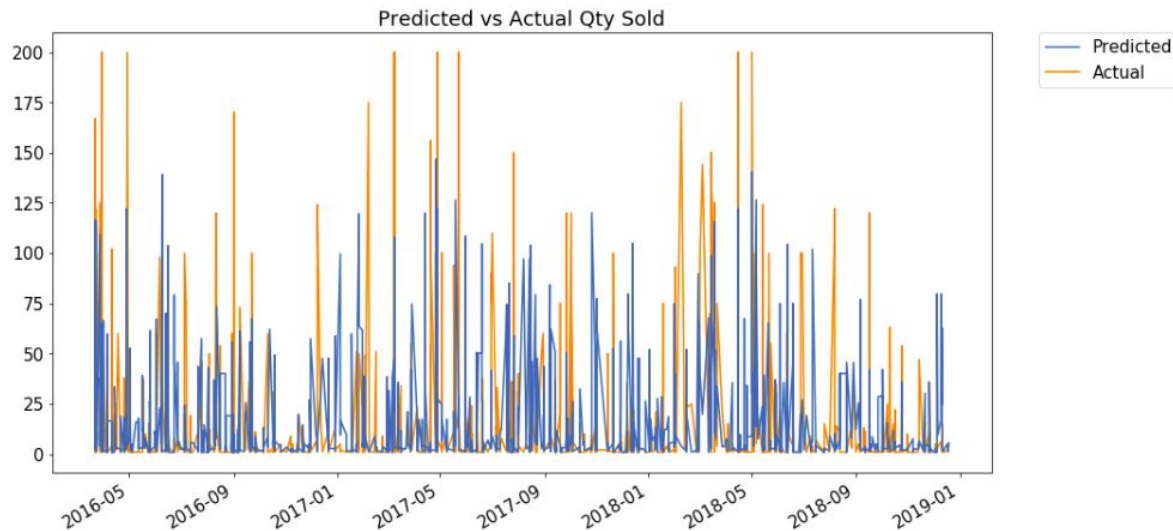
Same as LSTM daily prediction with added features

- Label encode categorical features
 - Type: 2
 - Name State: 52
 - Item: 183
 - Account: 21
 - Class: 20
 - Rep: 14
 - Account Type: 3
- Lag variable for previous days quantity

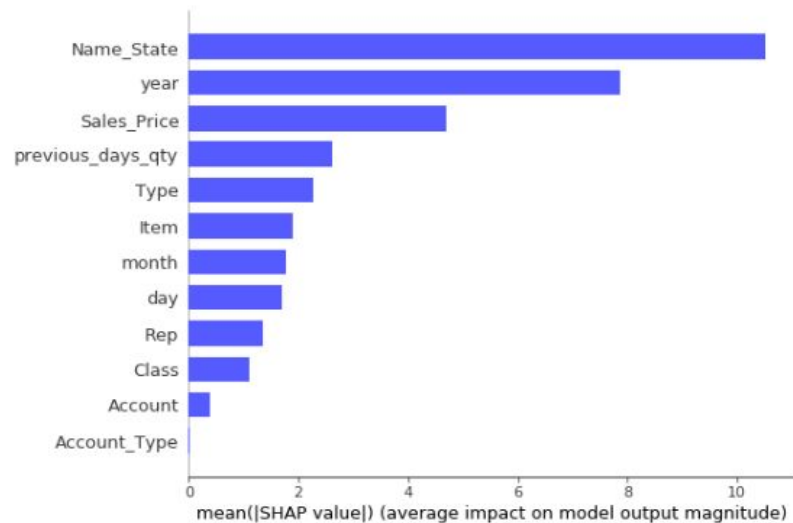
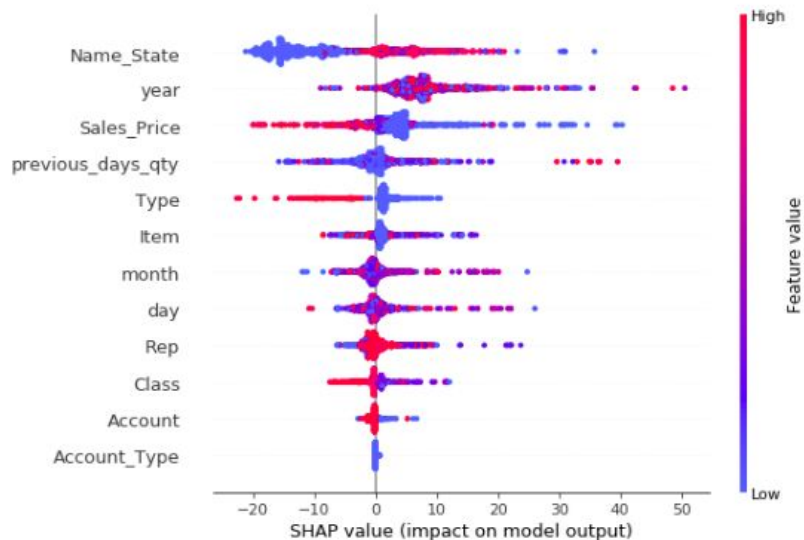
Daily Prediction

Tuned Parameters: 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 20

RMSE: 29.910



Feature Importance





Final Outcomes

Monthly SARIMAX Prediction: Test RMSE: 3.798

Daily SARIMAX Prediction: Test RMSE: 3.392

Monthly LSTM Prediction: Test RMSE: 15.835

Daily LSTM Prediction: RMSE: 39.720

Daily LSTM Prediction with Added Features: Test RMSE: 28.753

Implementation of Gradient Boosted Decision Tree: Test RMSE: 29.910



Further Exploration

- Incorporate more lagged features for the RNN and GBM; i.e previous 3 or 7 day quantity rolling average or other variables shifted values
- Research walk forward validation for LSTMs to see if it'd perform better
- Test out further architectures for LSTM model

Thanks!

Special thanks to ETS for their willingness to give a random samaritan their data, Max Sop for being a great mentor and resource, and the Thinkful community.
