Rossmann Store Sales Forecasting

Krishna Chaitanya Sanka
Mentor: Srdjan Santic

Agenda

- Problem Statement
- High Level Approach
- Data Exploration
- Feature Engineering
- Modelling Approach and Predictions
- Future Work

Problem Statement:

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

Task:
The task is to forecast 6 weeks of daily sales for 1,115 stores located across Germany using machine learning approach.

Business Impact:
- Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.
- By helping Rossmann create a robust prediction model, you will help store managers stay focused on what's most important to them: their customers and their teams!

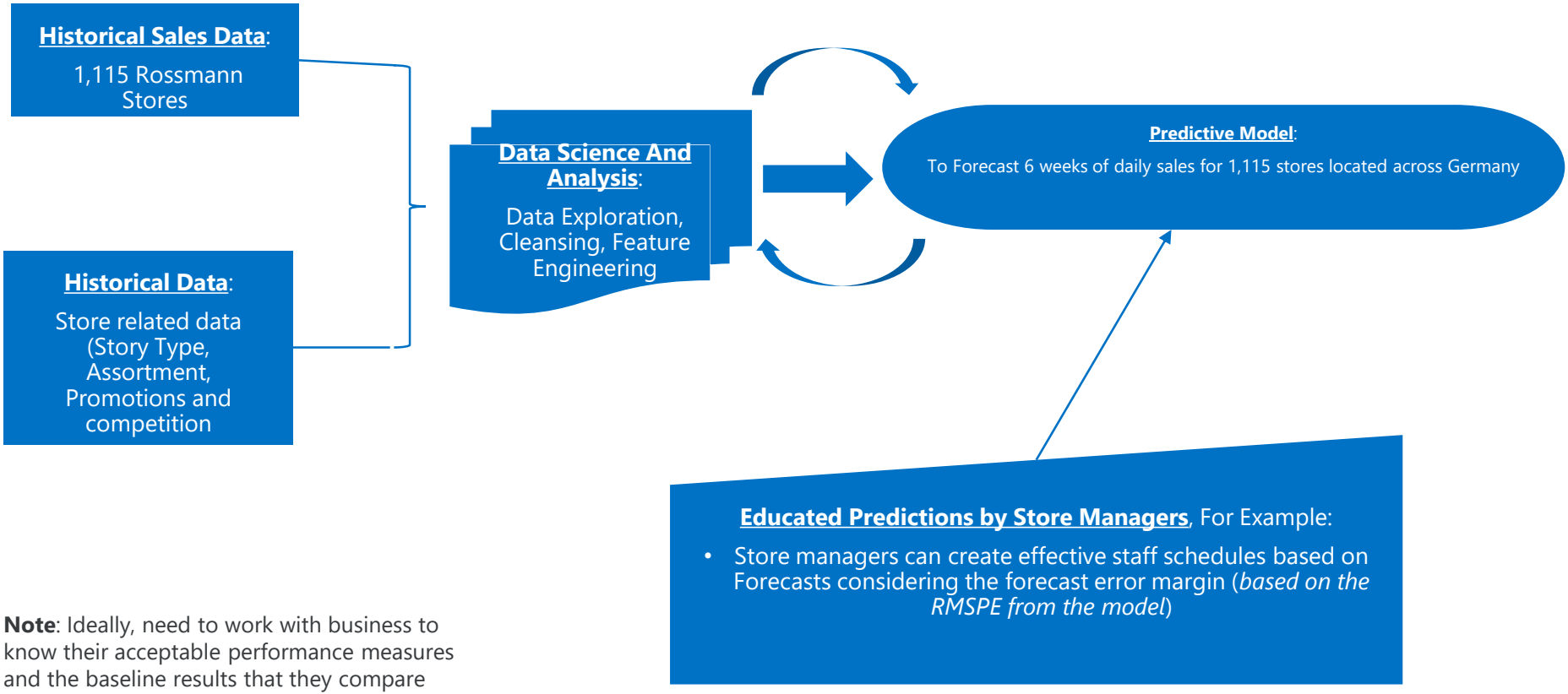
Dataset:
Historical Sales data for 1,115 Rossmann stores that includes store features, promotions and data related to competition

Description of the dataset

Most of the fields are self-explanatory. The following are descriptions for those that aren't.

- **Id** - an Id that represents a (Store, Date) duple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indicates if the (Store, Date) was affected by the closure of public schools
- **StoreType** - differentiates between 4 different store models: a, b, c, d
- **Assortment** - describes an assortment level: a = basic, b = extra, c = extended
- **CompetitionDistance** - distance in meters to the nearest competitor store
- **CompetitionOpenSince**[Month/Year] - gives the approximate year and month of the time the nearest competitor was opened
- **Promo** - indicates whether a store is running a promo on that day
- **Promo2** - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
- **Promo2Since[Year/Week]** - describes the year and calendar week when the store started participating in Promo2
- **PromoInterval** - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

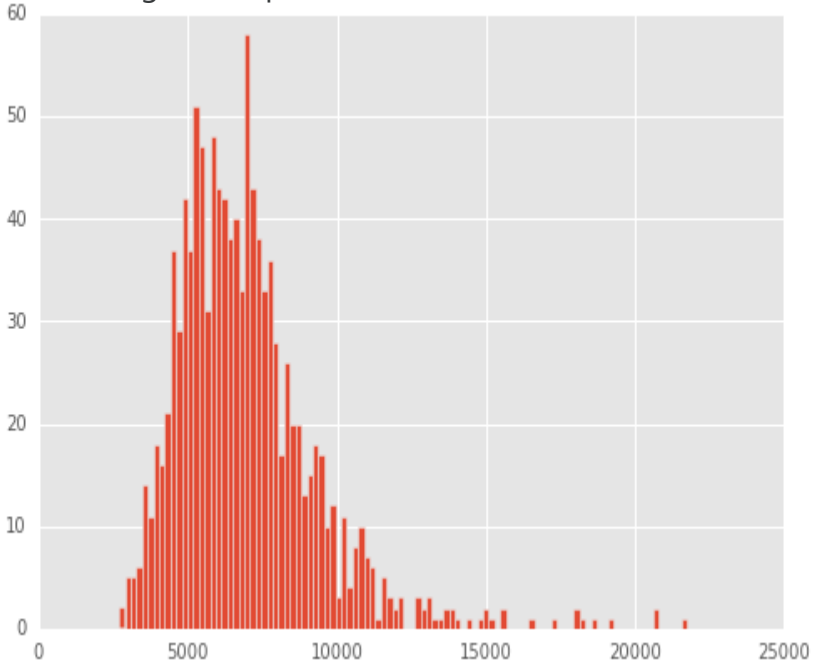
High Level Approach



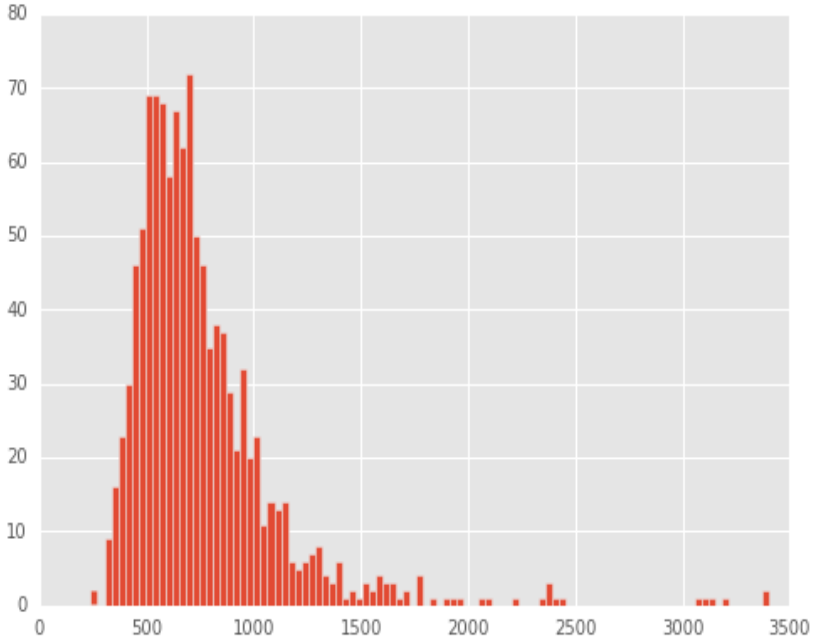
Note: Ideally, need to work with business to know their acceptable performance measures and the baseline results that they compare model output to.

Data Exploration

Average Sales per store when store is not closed



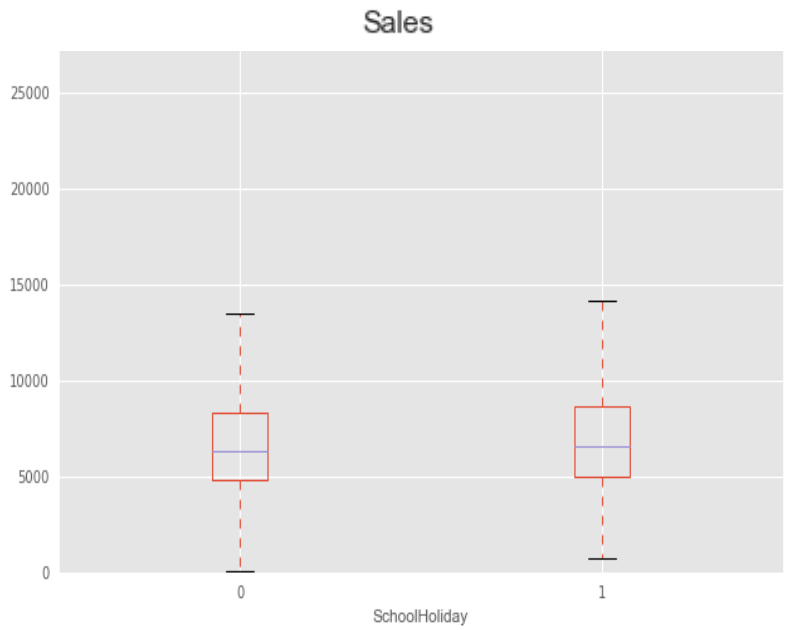
Average Customers per store when store is not closed

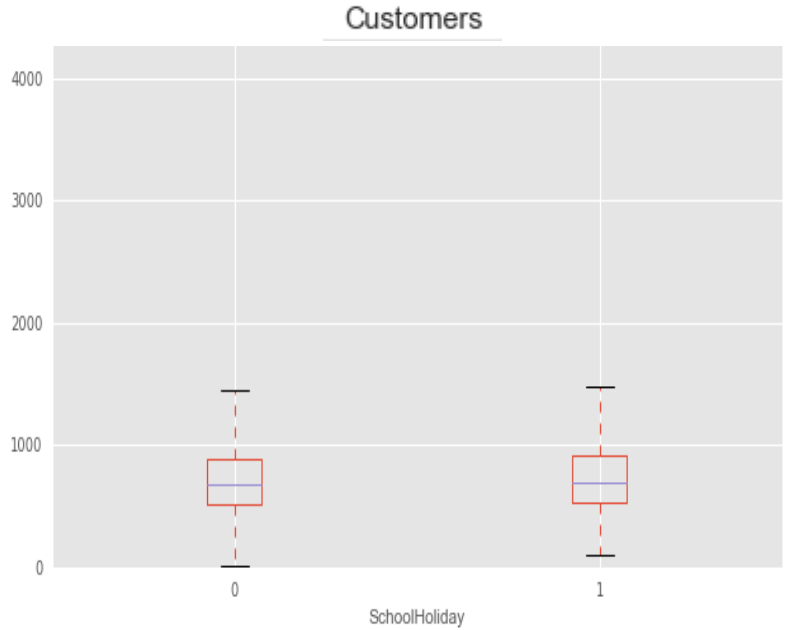


Insights:

- Given above has close to normal distribution, we can create additional features related to Average Sales per day per store, Average Customers per day per store, Average Sales per customer per day per store.

Data Exploration – Continued

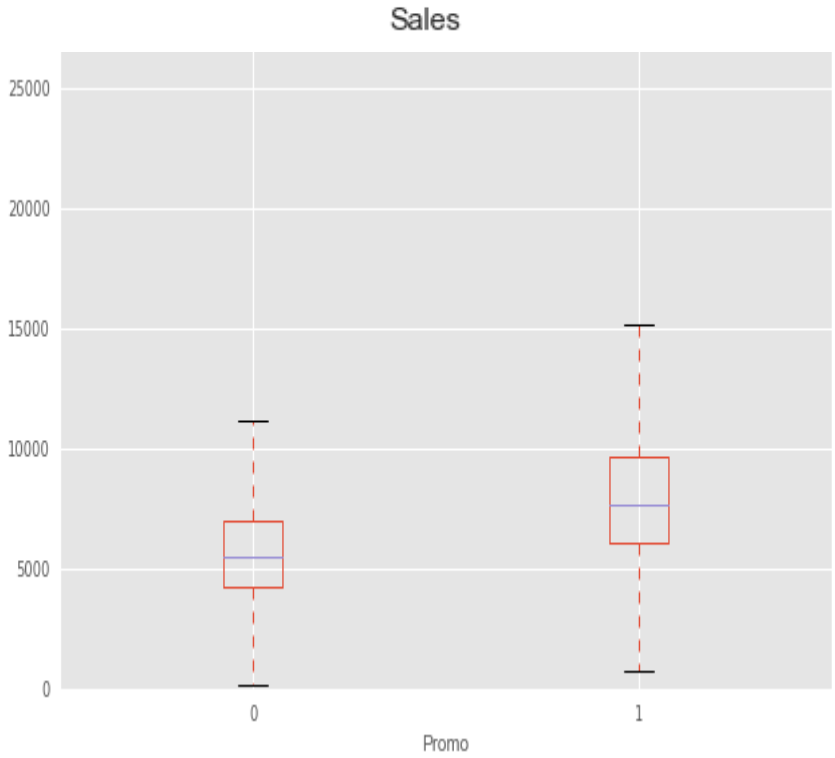


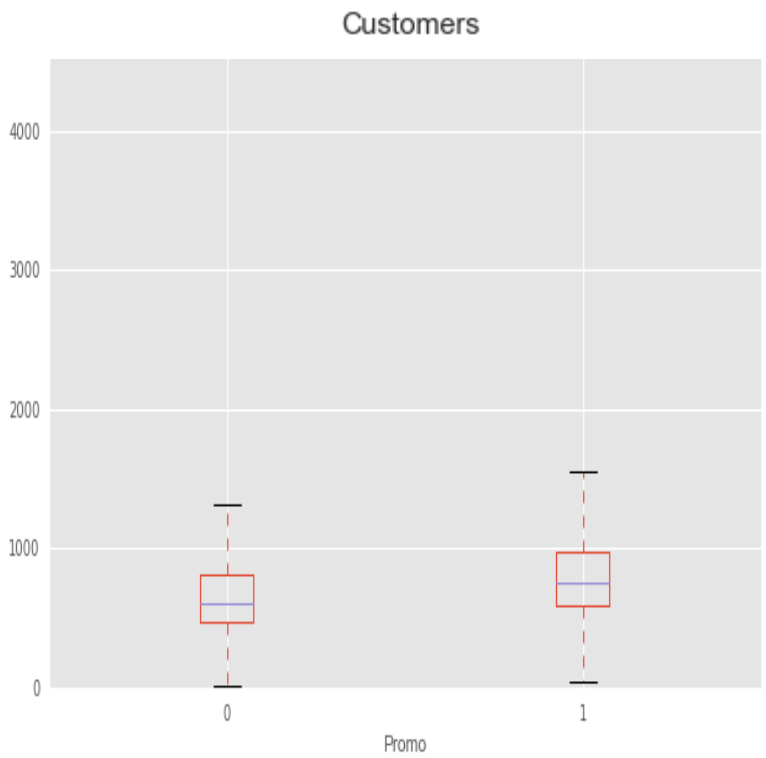


Insights: Is School holiday effecting my sales?
Yes, there is a slight increase in the Sales and the number of customers visited stores on a school holiday

Data Exploration – Continued

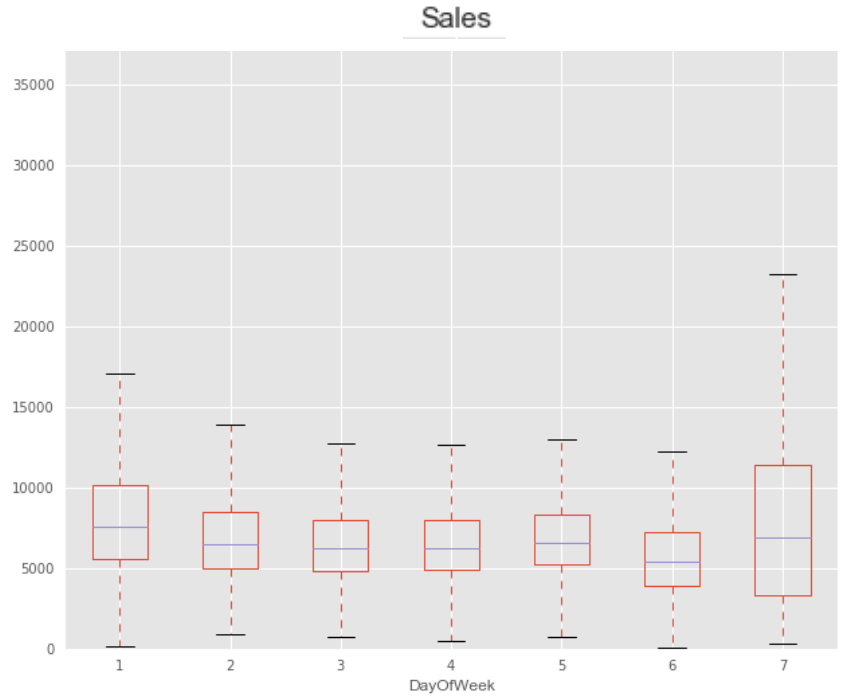
Insight: Clearly see the impact of promotions on the sales even though there is not a huge spike in customers visited.





Data Exploration – Continued

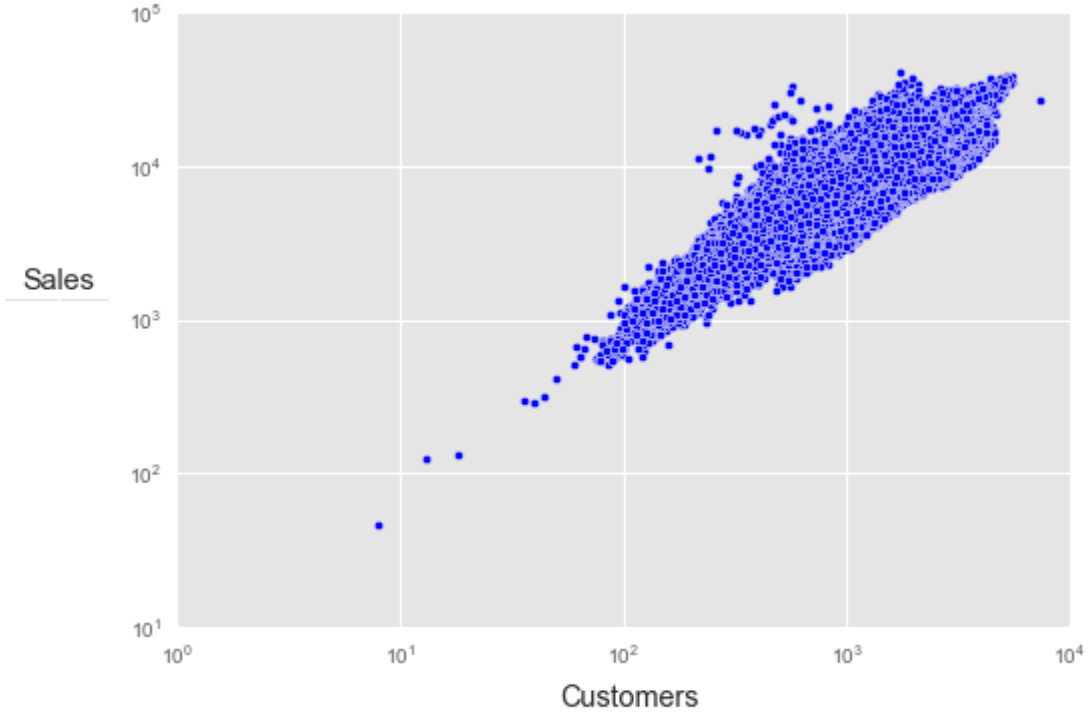
Sales vs DayOfWeek. Sunday has more sales than the rest of the days. This could be an important feature for the model.



Time series based features such as Day, Day of Week, Day of Year, Month Number will be important features given the sales are different on each day.

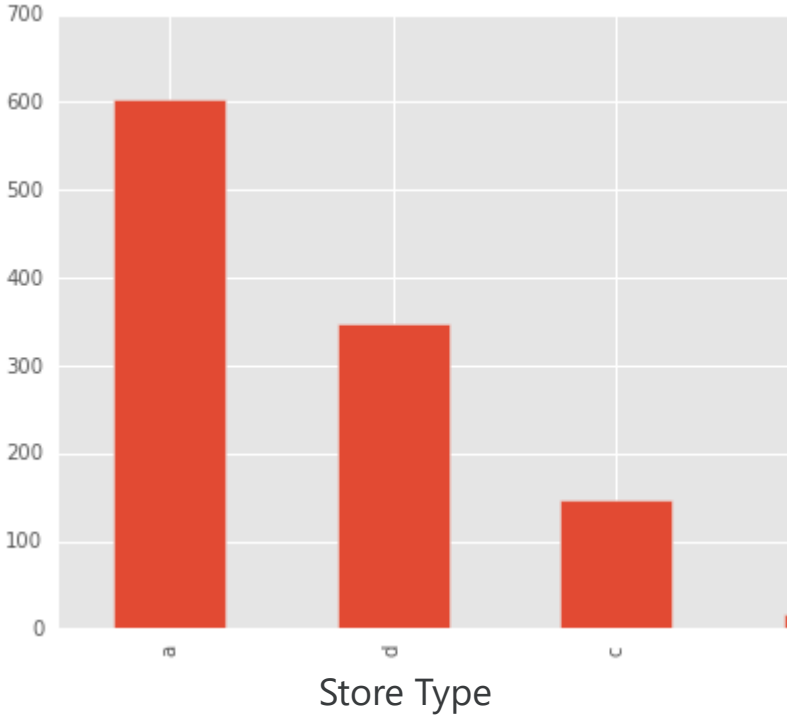
Data Exploration – Continued

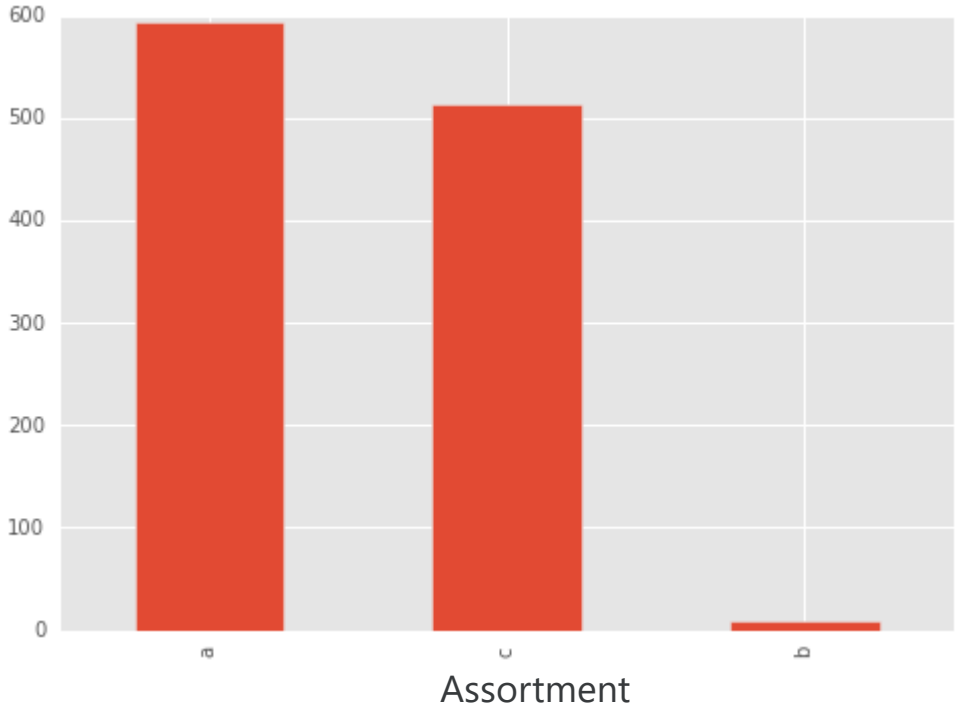
Customers vs Sales (log scale) – Linear relationship



Data Exploration – Continued

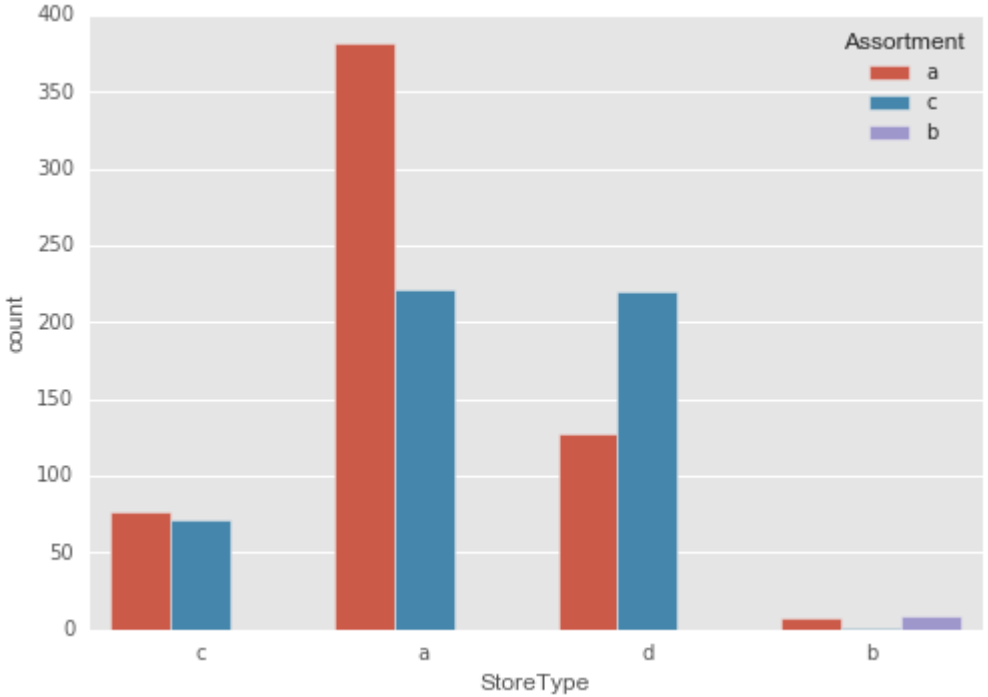
Number of Stores based on Store Type and Assortment. These features could be good ones for model as we see the distribution is not uniform





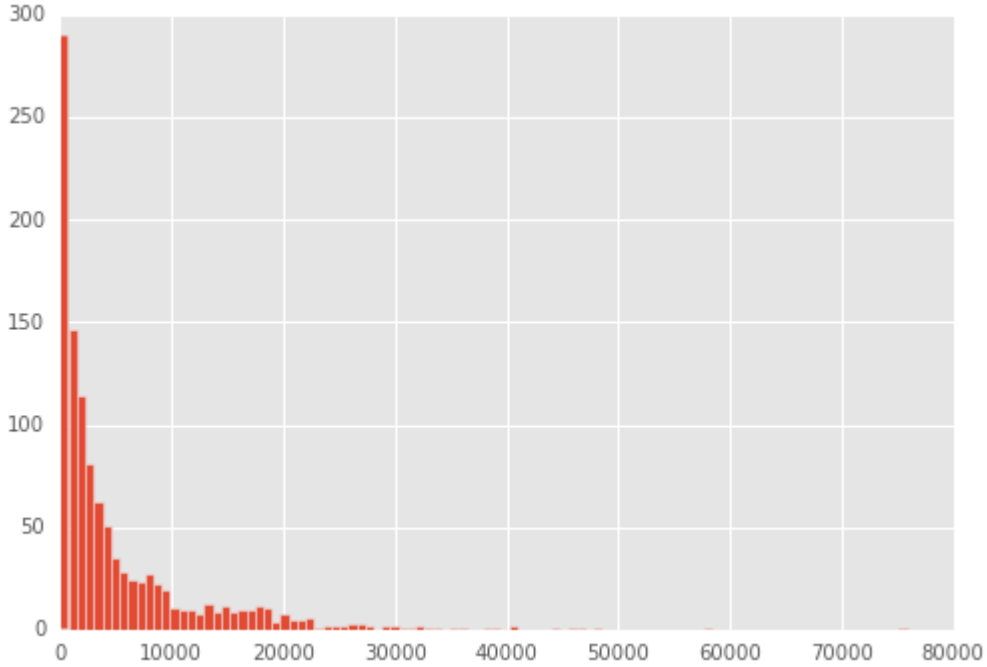
Data Exploration – Continued

Store data - Relationship between Assortment and Store Type



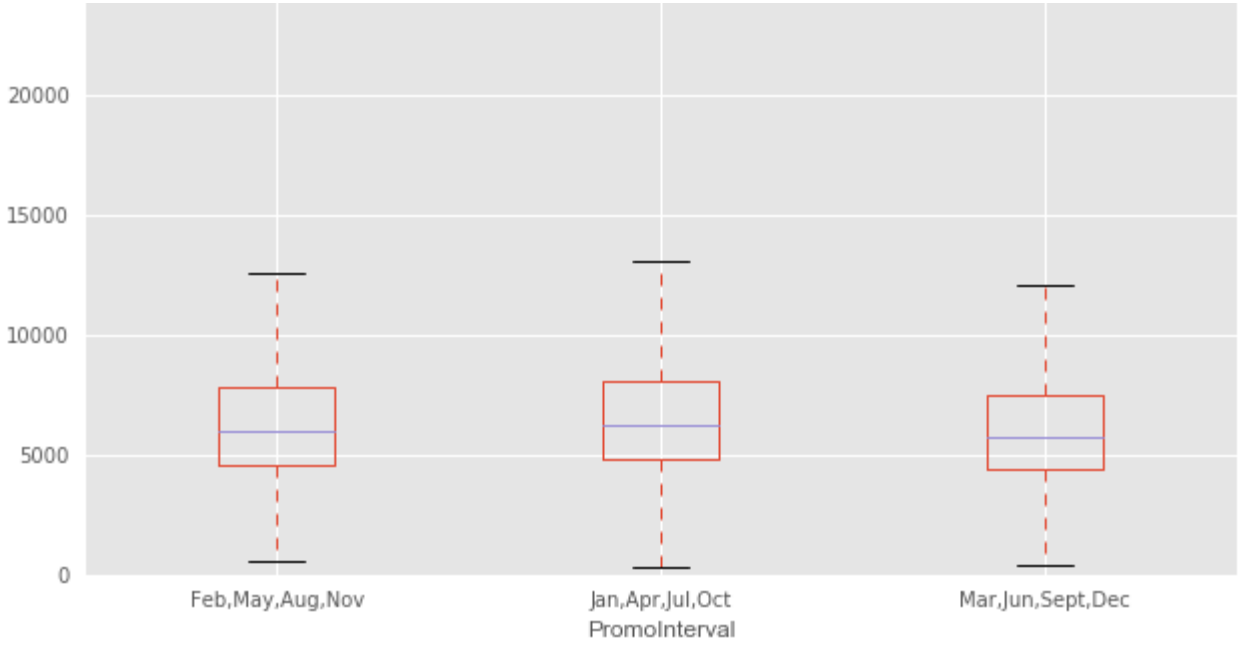
Data Exploration – Continued

Competition Distance – We can use log scale as part of feature transformation if we want to use this as a feature.



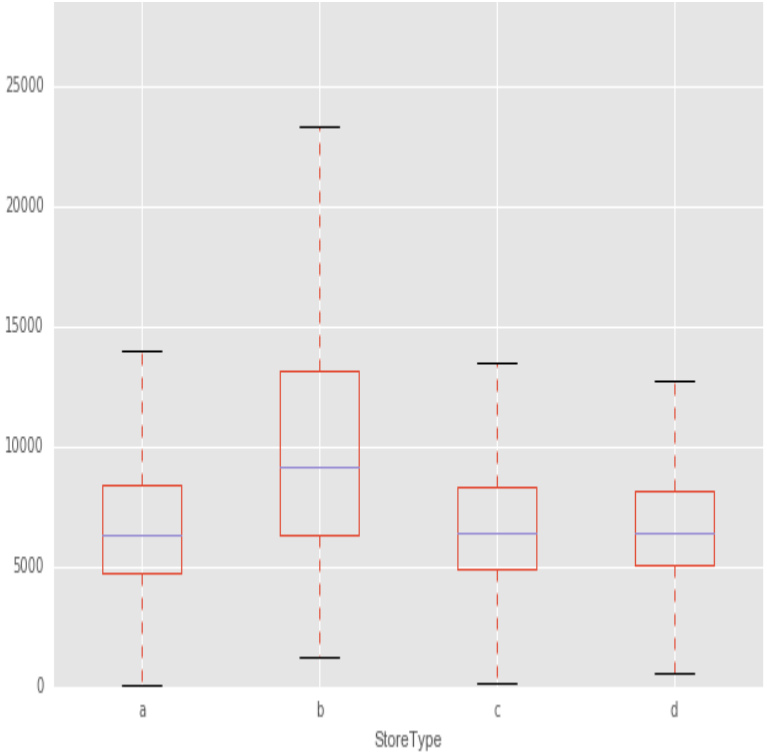
Data Exploration – Continued

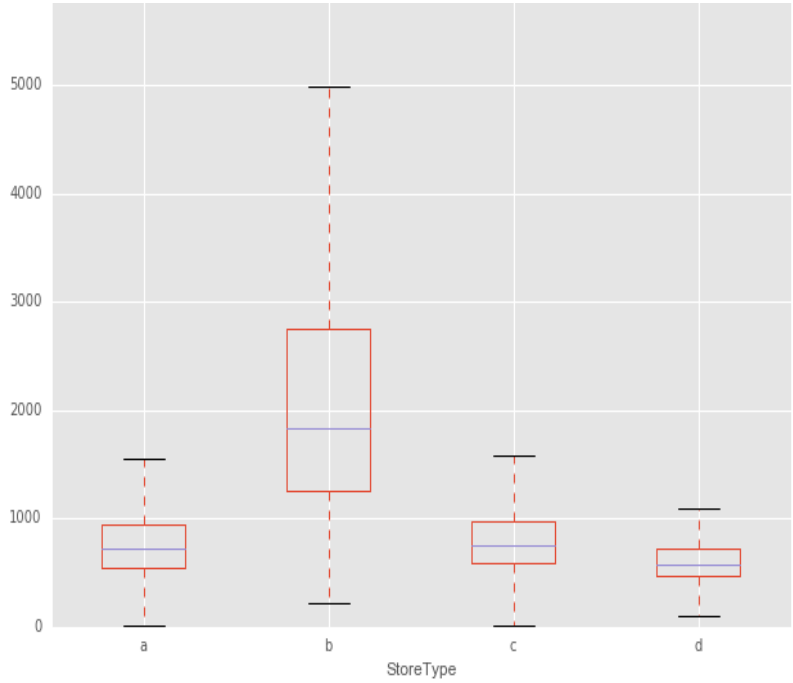
Sales by PromoInterval.



Data Exploration – Continued

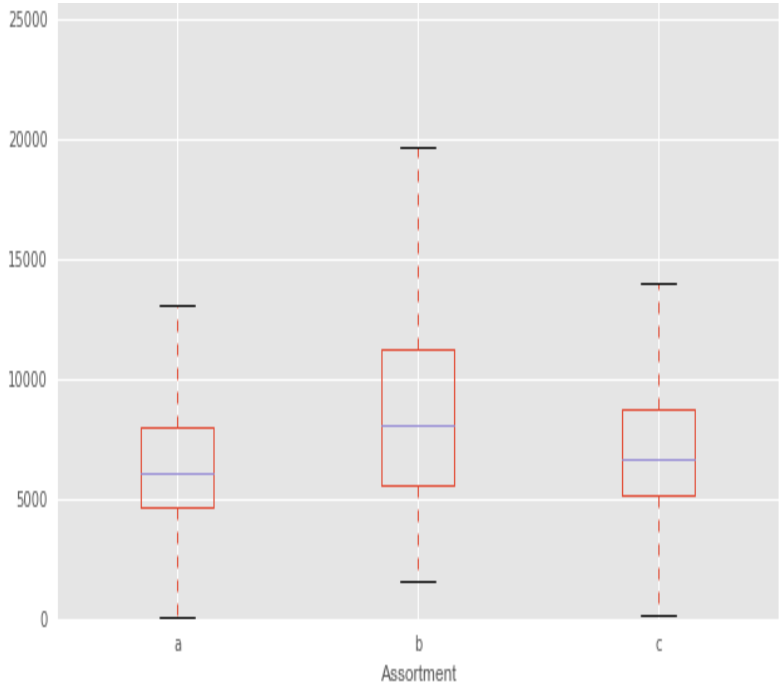
Sales and Customers by Store Type. Store Type b has more sales and customers

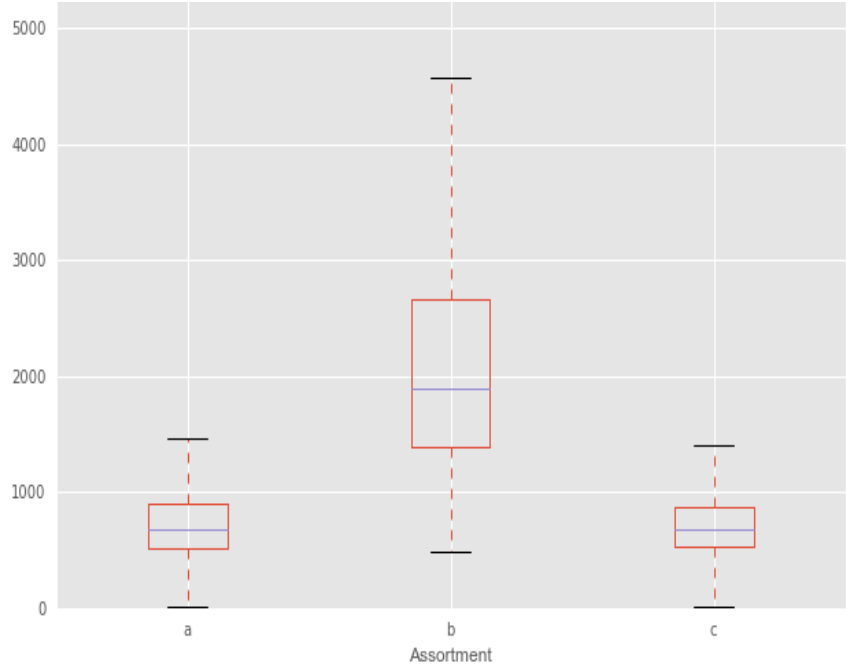




Data Exploration – Continued

Sales and Customers by Assortment – Assortment type b has more sales and customers.





Data Cleansing/Preprocessing

- Transformed the Date column to Datetime object and ordered the data by Date. This helped us to plot the time series to see if there is any discontinuity in the data and also helped to create “time series” related features

- Transformed the Sales label column into to log(sales). This helps in the modelling part to not sensitive to outliers

- Label encoding the State Holiday (sales data), Store Type and Assortment features (store data)

- Removed rows where store is open but no sales

Feature Engineering

Created the following Time Series based Features from Date Column (datetime object)

- Day
- Week
- Month
- Year
- DayofYear

Created Derived columns – Individual columns do not make sense. Hence combining them for interpretability

- CompetitionOpenSince - Appended CompetitionOpenSInceYear + CompetitionOpenSinceMonth
- Promo2Since – Appended Promo2SinceYear + Promo2SinceWeek columns
- PromoInterval ~ Created 4 features – one for each promointerval

Store based features (at store level per day) - This helps us to use customers feature as part of the model. These metrics are store specific and might help for predicting store sales

- SalesPerDay – Average number of sales per day at store level
- CustomersPerDay – Average number of customer per day at store level
- SalesPerCustomersPerDay - Average sales per customer per day

Modelling Approach

- Divided the train dataset into two sets (90% and 10%)

- Final Algorithm to train the model was XGBoost because
- It is an implementation of Gradient boosted decision trees designed for speed and performance.
- It is sparse aware (implementation with automatic handling of missing data values
- It has block structure (to support the parallelization of tree construction)
- It does Continued training (so that you can further boost an already fitted model on new data)
- It has the ability to optimize for a custom evaluation metric

```
num_boost_round = 5000, early_stopping_rounds=200
params = {"objective": "reg:linear",
          "booster" : "gbtree",
          "eta": 0.1,
          "max_depth": 10,
          "subsample": 0.85,
          "colsample_bytree": 0.4,
          "min_child_weight": 6,
          "silent": 1,
          "thread": 4,
          "seed": seed
          }
```

- Other Algorithms tried: Decision Trees and RandomForest

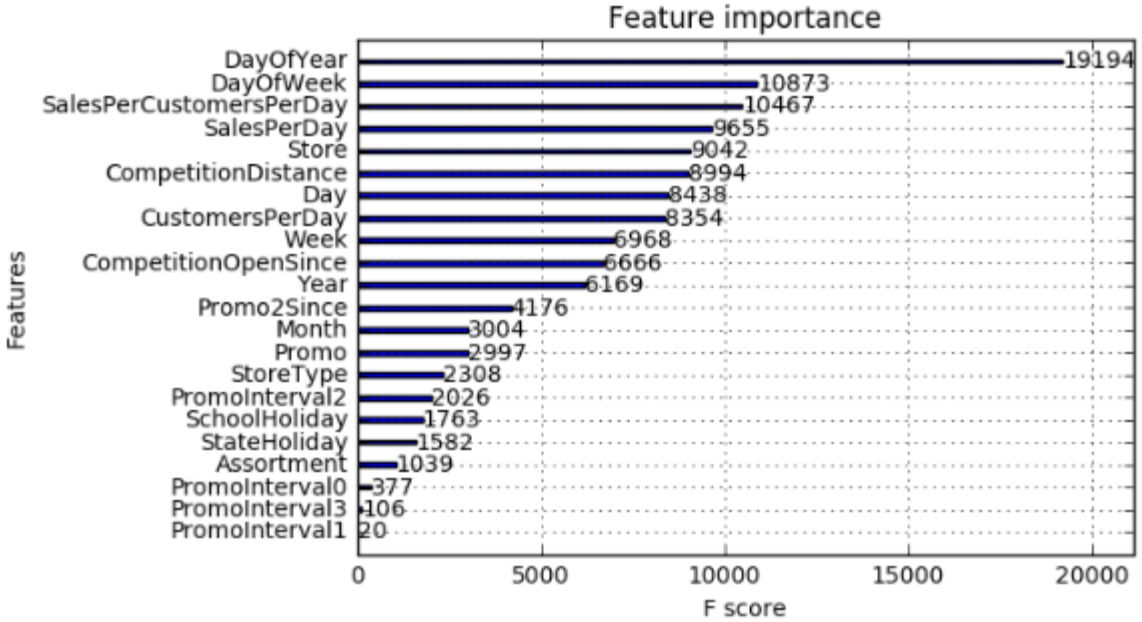
Predictions

- Trainset performance (90% of train set) : RMSPE: 0.092814
- Test Set Performance (10% of train set) : RMSPE: 0.107456

- Finally ran the model on the blind test set Rossmann provided and submitted the results in Kaggle and RMSPE is ~ 0.119 (top 20% in Kaggle for this dataset – Private Leaderboard)

Feature Importance

- DayofYear is the most important feature of the model and has a higher relative importance than other features and this is a derived feature.
- Most of the top features that are important for sales predictions at store level are derived features



Some of the Promotional Intervals did not come in the top feature list (PromoInterval1 and PromoInterval3)

Future Work

- Do a Grid Search of Parameters and Feature Selection - It is very computational Intensive but we can do it once to get the best parameters and features combination to get the model with lowest RMSPE.

- Check if there are any outliers in each store and check the methodologies to correct and test the above approach

- So far, the algorithms tried are Decision tree (for interpretability), randomForest and XGBoost. XGBoost is fast as well as it gave best results. I would like to try neural networks as well to see how it performs on this dataset.

- Solve this problem using time series approach instead of machine learning approach as we have enough data to capture trend and seasonality.

How this can be implemented in production?

Ideal way to approach this is *(need to check with business to see if we need to include any of their constraints in implementing the below approach)*

- Check with business on how frequently do we need to refresh this model from production standpoint.
- Develop an end to end pipeline that takes the consolidated sales data from all 1,115 stores and do the data pre-processing, feature engineering and then train the model (using the cross validation approach) and output the predictions based on the refresh frequency
- The pipeline should enable a continuous integration of new data (every day/week) and help forecast to be as accurate as possible as and when the model is trained including new data.
- A report should be send to each store manager for his specific store forecasts for next 6 weeks