Rossmann Store Sales

Problem

A forecasting problem featured at Kaggle competitions website is selected. 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.

Challenge is to predict 6 weeks of daily sales for 1,115 stores located across Germany. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation. A robust prediction model will help store managers stay focused on what's most important to them: their customers and their teams!

Data

We are provided with historical sales data for 1,115 Rossmann stores. The task is to forecast the "Sales" column for the test set.

Files

- **train.csv** - historical data including Sales
- **test.csv** - historical data excluding Sales
- **sample_submission.csv** - a sample submission file in the correct format
- **store.csv** - supplemental information about the stores

Data fields

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

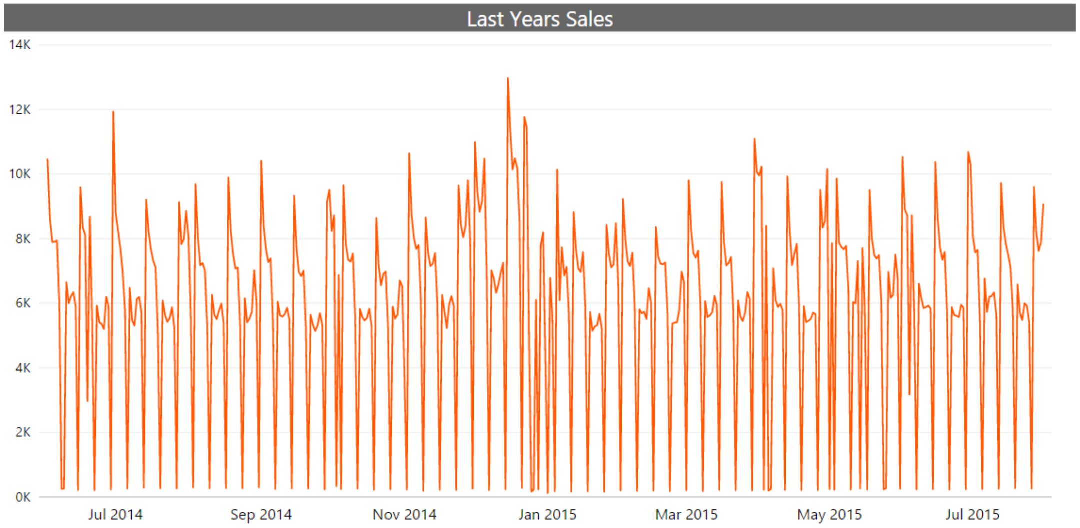
- **Id** - an Id that represents a (Store, Date) duple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was 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

Statistic	Value
Dataset size	1017209
Testing data size	41088
Total stores number	1115
Training data Time ranges	2013-01-01 to 2015-07-31
Testing data Time ranges	2015-08-01 to 2015-09-17

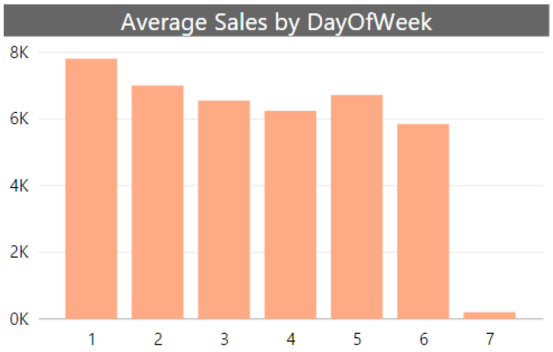
Data Exploration and Visualization in Power BI

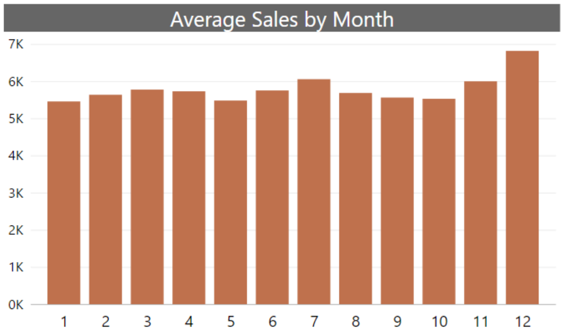
Statistic	Value
Global store sales average	5773.82
Max daily sales	41551
Min daily sales	46



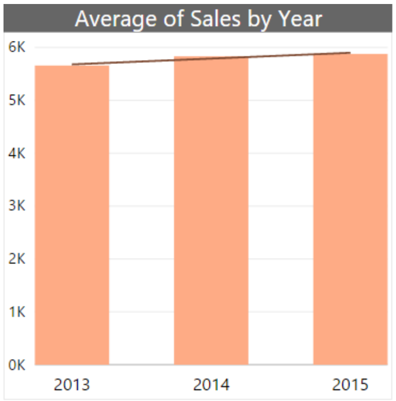
Plotting last year’s sales reveal a strong weekly trend with sharp dips on Sundays due to holiday and pointy peaks on Mondays. Peaks in December and January are distinctively higher than other peaks suggesting that Month would also be a valuable predictor of sales.

As expected the day of the week shows clear segregation in Sales values. Weekdays generally have a causal relationship with people's behavior and work routines so they can be strong predictors of Sales on any given day.

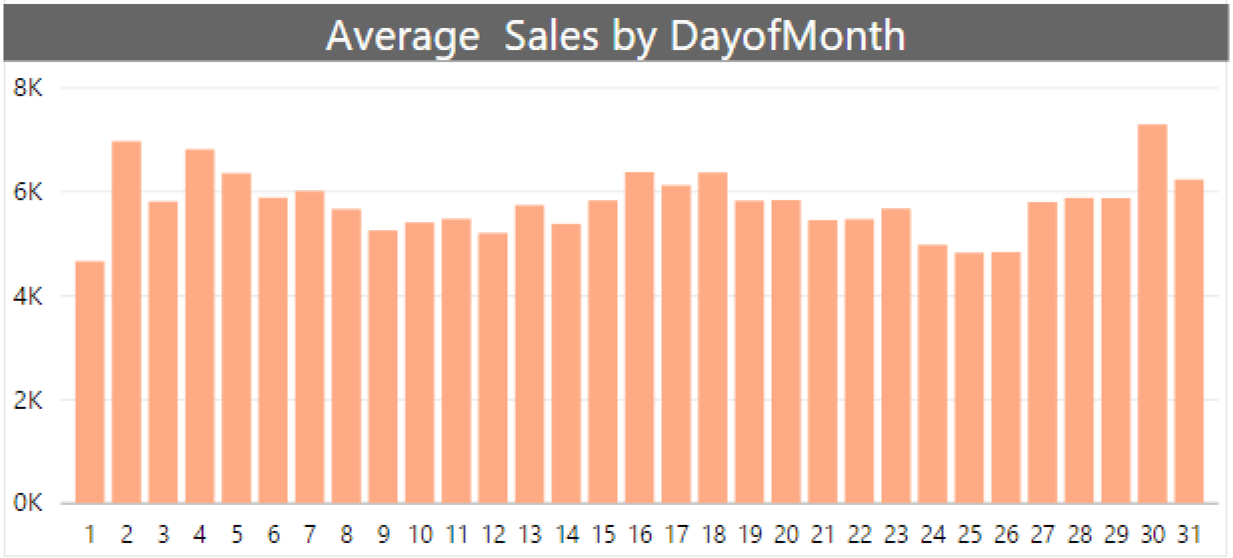




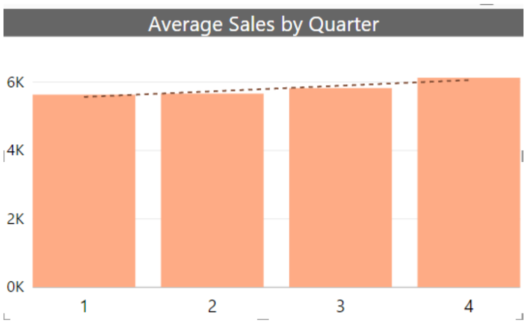
Average Sales by Month plot also shows reasonable difference in average daily sales among months, so this feature must also be utilized for training. Monthly difference may be a result of annual holidays, festivals and weather variations.



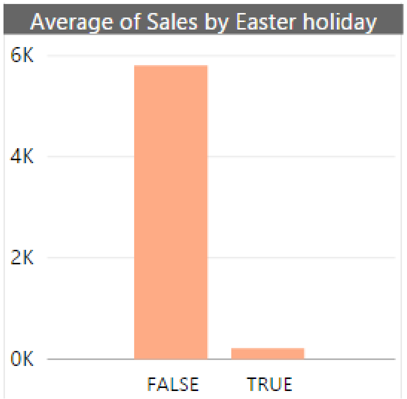
This plot shows a clear early trend, Sales are increasing with passing years. This can be due to increasing popularity of Rossmann Stores in Germany, or increased used of medication. But this causal explanation is not needed, for building a forecast model we only need to know that this variable effects the output, and our training algorithm will take care of the mathematics.

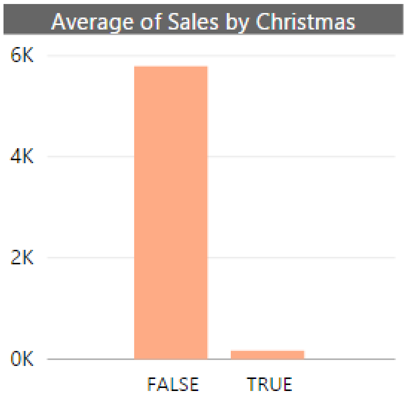


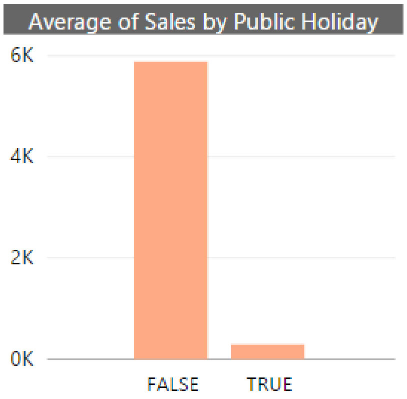
Average sales show a variation that is depends on the day of the month, start, end and middle of the month features high sales days. So this feature must also be engineered from date and fed to the training module.



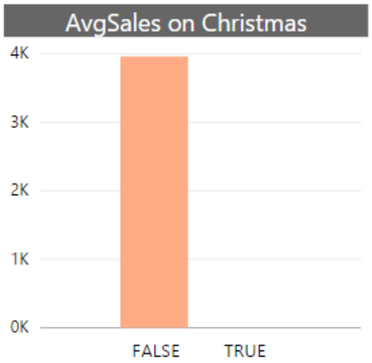
Average Daily Sales show a strong linear increase with the quarter of the year. So Quarter of the year can also be used as a predictor, but the quarterly trend may be captured by the month variable in a more fine grained fashion.

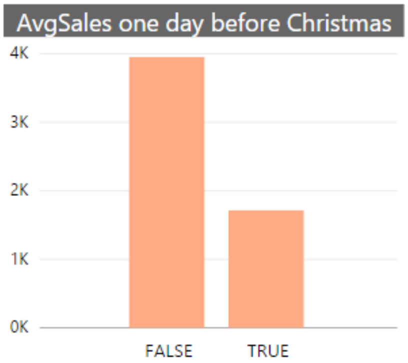


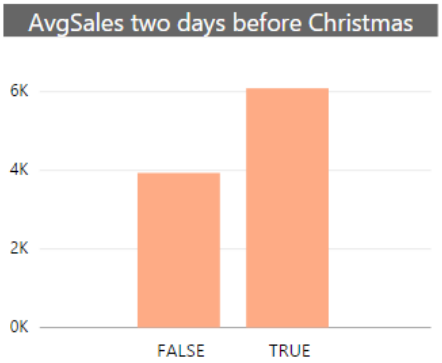




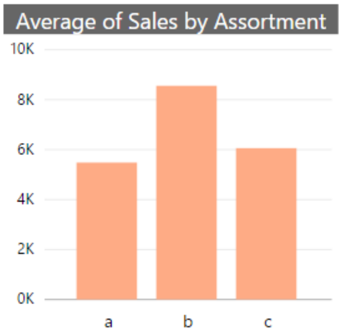
Average Daily Sales drop significantly on the public holidays, thus they must be used as predictor of daily Sales. But since holidays are relatively rare, they may not contribute a lot to the average accuracy measures. Since people do not like to visit medical stores amidst holiday celebrations they might want to buy their medicines ahead of these holidays, so let us examine daily sales in days preceding holidays.

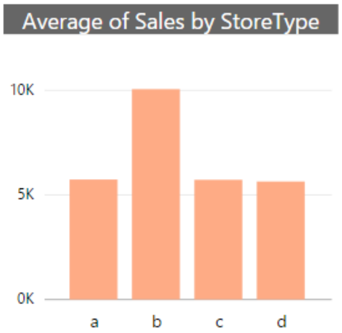






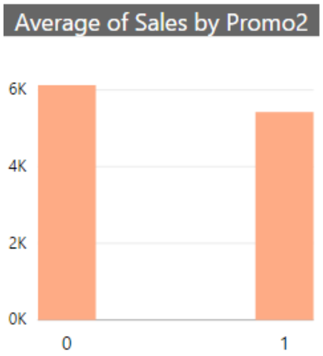
People don’t visit medical stores on Christmas, some do so one day before Christmas, and most like to refill their medical supplies two days before Christmas. So we can engineer features for days preceding the holidays and use them for prediction.

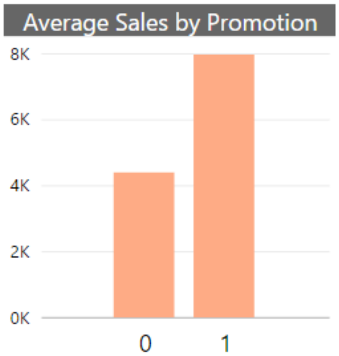


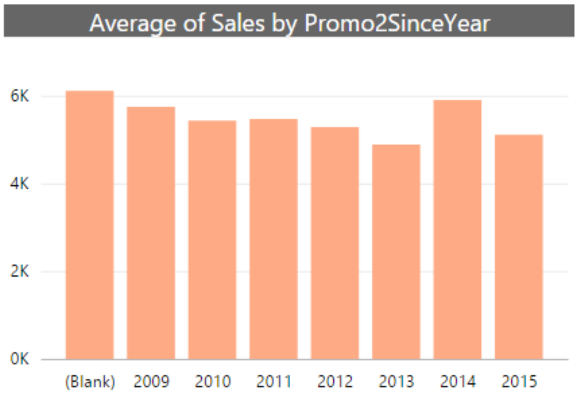


Store model and assortment level also impact the daily average sales and should be used for prediction.

As expected promotions increase the daily average sales so it is useful predictor. But the average daily sales for stores using Promo 2 is less than the others, this could be because Promo2 is only applied for weak performing stores. In any case these two features appear to be important predictors.

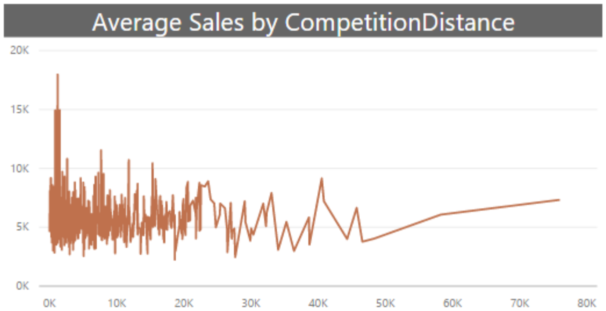




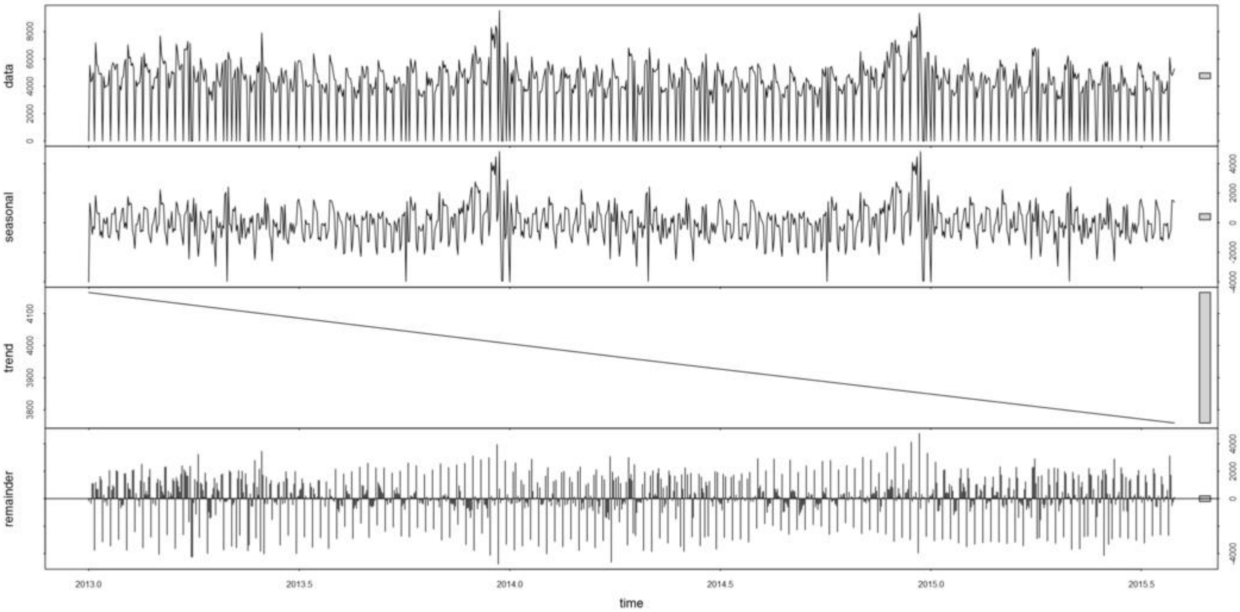


Blank denotes the stores with missing values, which in this case refers to the stores where Promo2 is never applied. The year in which promo2 was introduced appears to reasonably good predictor of Sales.

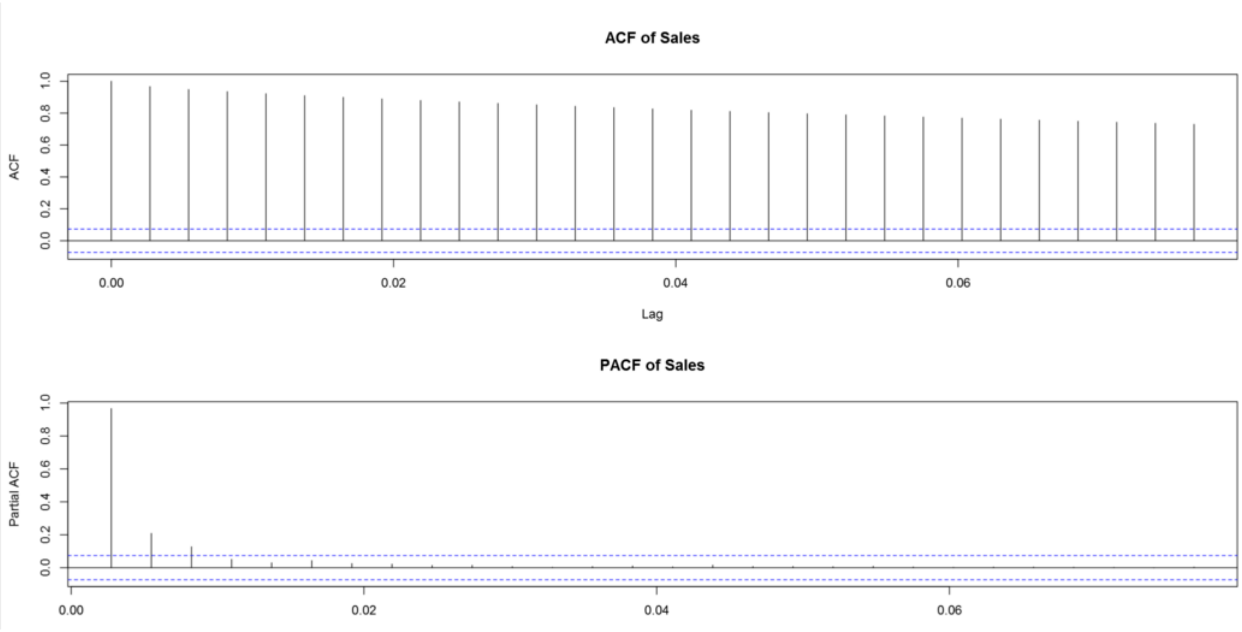
There is a lot of noise and no clear relationship between the average daily sales and distance to the competitor is visible. This added to the fact that a lot of stores have missing competitor's data makes this variable less useful.



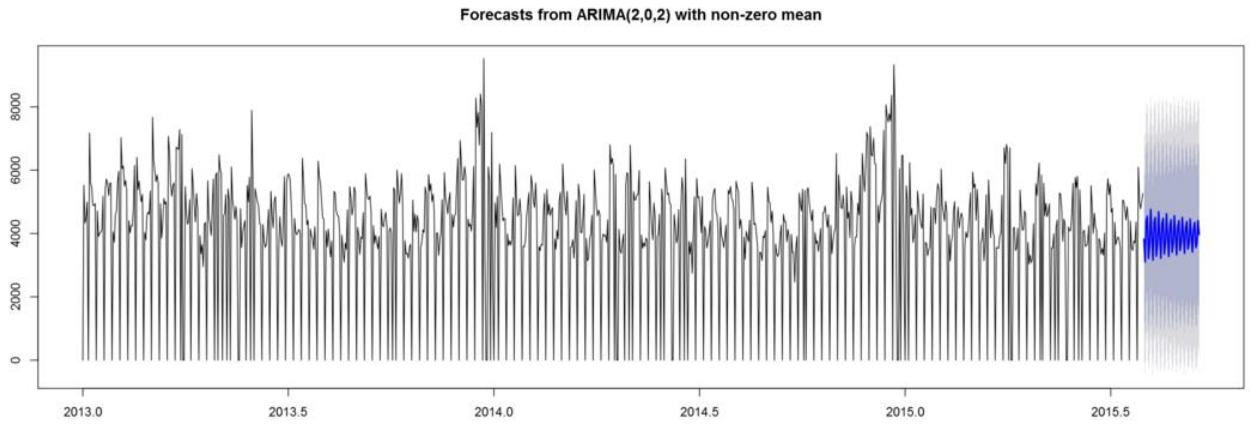
Decomposition of Sales Time Series



This is the decomposition of time series produced by an R visual in Power BI. The annual seasonal trend was separated and captures the peaks and dips which repeat annually. The trend part shows a gradual decrease over the time. Remainder appears to be stationary but the fluctuations are huge, that means that significant proportion of daily sales is not modelled in seasonal and trend components.



Gradually decreasing autocorrelation function shows that the Sales values are highly correlated with the Sales on previous days. Both ACF and PACF graphs suggest that Autoregressive model might be a good fit for this time series.



This forecast is produced in Power BI by auto.arima function available in forecast library of R. The width of confidence interval as compared to actual values suggest that this model is not a great fit for the data and results in forecast with very high uncertainty.

Feature Engineering

- Extract Month, Year, Day of the Month and Week of the Year from the date. Exploratory analysis revealed that these temporal features could be useful predictors of daily average sales so they are extracted from date using R script.
- OpenLag and OpenLead
  These features show if store was closed the previous day or will be closed the next day. Since the days preceding the holidays showed increased Sales in exploratory analysis, same may apply to the day preceding and following the day the store is closed.
- An indicator variable each for the seven days preceding a state holiday.
- Calculate daily average sales, average customers and average Sales to customer ratio for each store : Since store ID is a categorical variable, there are more than a thousand stores, and algorithms do not handle such large number of categories very well. A good alternative is to use average sales and customer information for that store as a numerical predictor in regression.
- Calculate average Sales and average customers on each store on every weekday.
- Calculate average Sales and average customers on each store on every day of the month.
- Calculate Average Sales for each store type, where storetype is a unique combination of Assortment and Storetype.
- Calculate the months since which the competition has been opened.
- Remove Customers from the training set, since this information is missing from the test dataset because we cannot know the number of customers on any given day in advance.

Including 0 Sales Days	Score Private	Score Public	R2
Yes	0.12349	0.11873	0.957088
No	0.13085	0.12390	0.9576

Thus 0 Sales have some important information, which trains the algorithm for low sales days. This the 0 sales days should be used in training the model.

Top 15 Correlated features before (left) and after (right) Feature Engineering.

Feature	Corr Coefficient
Customers	0.823596732
Promo	0.368145259
DayOfWeek	0.178736358
Promo2	0.127595814
Promo2SinceWeek	0.095310509
weekOfYear	0.072140357
Date	0.062756509
DayofMonth	0.051848584
CompetitionOpenSinceMonth	0.043489324
SchoolHoliday	0.03861655
CompetitionDistance	0.036396089
year	0.036168645
Promo2SinceYear	0.034712812
OpenLead1	0.033294464

Feature	Corr Coefficient
Customers	0.823597
AvgSalesDOW	0.799343
AvgSalesDay	0.778105
AvgSalesMonth	0.768718
AvgSales	0.748835
AvgCustomerDOW	0.700744
AvgCustomerDay	0.667373
AvgCustomerMonth	0.65999
AvgCustomer	0.643572
Promo	0.368145
AvgSalesStoreType	0.219706
DayOfWeek	0.178736
AvgCustomerStoreType	0.164032
Promo2	0.127596

As we can see we have engineered features have a lot stronger correlation with the sales and will serve as effective predictors.

Feature importance as calculated by Permutation Feature importance

Feature	Score
AvgSalesDOW	0.976177
AvgSales	0.502096
AvgSalesDay	0.403695
Promo	0.252841
AvgSalesMonth	0.213291
AvgCustomer	0.120098
DayofMonth	0.063002
month	0.062051
DayOfWeek	0.040149

year	0.026632
AvgCustomerMonth	0.024499
AvgCustomerDOW	0.009504
AvgCustomerDay	0.009407
CompetitionDistance	0.007682
CompetitionSince	0.007588
AvgConversion	0.00393
SchoolHoliday	0.003364
AvgSalesStoreType	0.002333
StoreType	0.00216
promo2Active	0.001665
Assortment	0.001336
Promo2	0.001008
AvgCustomerStoreType	0.00054
IsStateHoliday	0.000484
StateHoliday	0.000453
OpenLag1	8.02E-05
IsStateHolidayLag2	7.91E-05
OpenLead1	4.05E-05
IsStateHolidayLag1	1.06E-05
IsStateHolidayLag4	7.55E-06
IsStateHolidayLag7	1.70E-06
IsStateHolidayLag3	4.31E-07
IsStateHolidayLag6	2.21E-07
IsStateHolidayLag5	1.49E-07
Open	0

Transformations

- Filter out the days on which store is closed because it will always result in 0 sales and can be handled separately. But adding these rows may bias the algorithm with respect to other predictors.
- Join the daily Sales data with the respective Store information, so that the store features can be used for prediction.
- Take log of competition distance to make the distribution more normal. Regression algorithms perform better with normally distributed feature set.
- Normalize numerical variables. Normalizing is very important in case of linear regression but with decision trees it can be skipped.
- Apply Ln+1 on all Sales and average Sales. To make the distribution more normal.

Missing Data Treatment

There are a lot of missing values in Competition Open Since and Competition Distance. Different techniques were used to handle them and resulting models evaluated to come up with the best treatment.

Missing Competetion data	Score Private	Score Public	R2
Sentinel for OpenSince Median for Distance	0.12349	0.11873	0.957088
Sentinel for OpenSince Median for Distance	0.12491	0.11726	0.957035
Median for OpenSince Median for Distance with Indicator	0.12708	0.12038	0.95699
Median for OpenSince with Indicator Median for Distance with Indicator	0.12512	0.11815	0.956896
Median for OpenSince Median for Distance	0.12651	0.11923	0.957052

- Replace missing values for Competition Since with -50 sentinel value. Here missing values correspond to the store which has no competition or the stores for which competition is not yet opened. Competition open since month and year columns are missing in almost one third of the stores. No clear co-relation between missing values and other store features was observed through visualization. As imputation resulted in impractical values, sentinel was used instead.
- Replace missing values in competition distance with median.

Model Selection

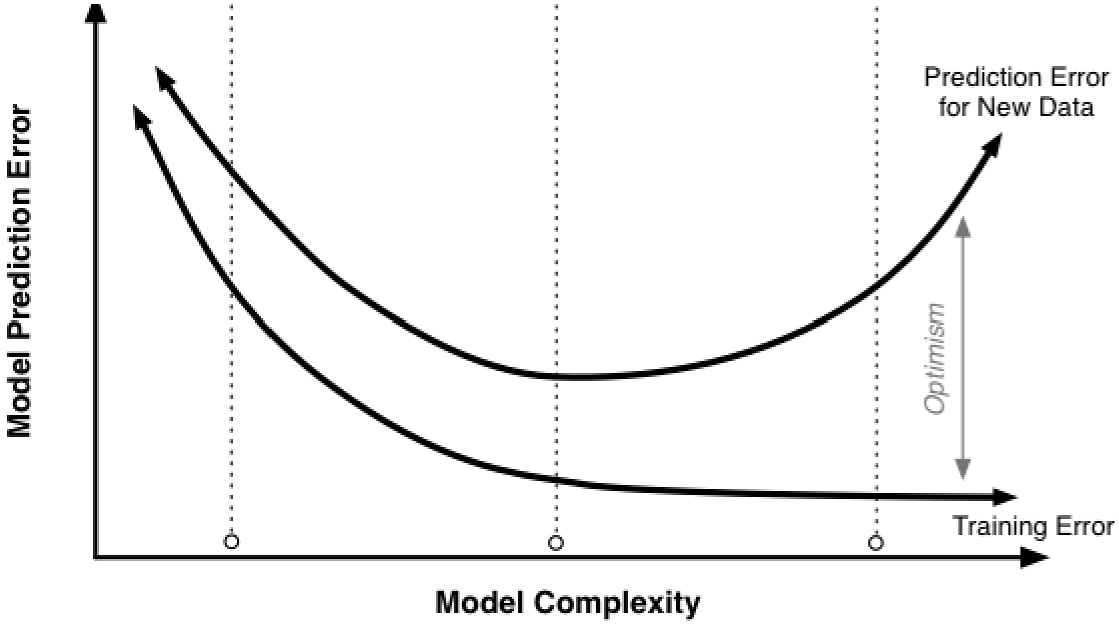
The most commonly used models for time series forecasting are ARIMA Models (Auto-regressive Integrated Moving-average). But they do not take other features into consideration, but solely rely on time series values, exploiting the temporal trends and relationships in the data. The data about the promotions, store-type and competitions was discovered to be important in exploratory analysis, but they will not be used in ARIMA model forecasting. Another approach is to turn the problem into a regression problem where all descriptive features about that day including the temporal features like month and day of the week are used to train a regression model, and the predictions about the future data will serve as a sales forecast. Regression model will also reveal the effect of individual features on Sales unlike ARIMA model which is like a black box. Thus regression approach is used.

Initially Linear regression model was used, but Boosted decision trees outperformed them with margin. There are two reasons. Firstly, decision trees can model non-linear relationships between features and label while linear regression cannot. Secondly, Boosted Decision Tree module in Azure ML is an ensemble model build by combining many trees which is usually better than an individual model. Comparison between model performances was made using the Model evaluation module, public scores and private scores. Boosted Decision Trees outperformed other algorithms.

Tuning Model Parameters

Tune model hyper parameters module was used to get a rough estimate of parameter values and then manual selection was made by monitoring model performance.

While finding the optimal model parameters we must be very careful and strike the balance between model complexity and predictive power. Increasing the number of decision trees and other features that enhance the complexity of model may result in overfitting the test data and producing models that are biased by minor patterns in test data and perform poorly when exposed to new data.



As shown in the above figure, error in training data will continue to decrease with training data but after a point it will increase with model complexity. We need to find the optimum value that corresponds to the dip in curve of Prediction error. Thus we have used the scores computed on test data that was not seen by the algorithm. The highlighted parameters in the following tables yielded best performance, so they were used.

Boosted Decision Tree

# of Trees	Score Public	Score Private	R2
100	0.13601	0.12854	0.92
1000	0.12971	0.12352	0.94
5000	0.12790	0.12053	0.95
8000	0.12867	0.12069	0.958

Boosted Decision Tree with 1000 trees

Minimum Samples per Node	Score Public	Score Private	R2
20	0.12971	0.12352	0.94
40	0.12838	0.12269	0.950546
50	0.12987	0.12309	0.9503

Boosted Decision Tree with 1000 trees & Minimum Samples per Node: 40

Maximum Leaves per tree	Score Private	Score Public	R2
50	0.12838	0.12269	0.950546
80	0.12723	0.12155	0.952855
100	0.12619	0.11928	0.954127
120	0.12565	0.12006	0.954858
140	0.12605	0.11832	0.955405
180	0.12596	0.11708	0.95584
250	0.12349	0.11873	0.957088
300	0.12427	0.11906	0.9577
500	0.12671	0.11687	0.957906

Evaluation

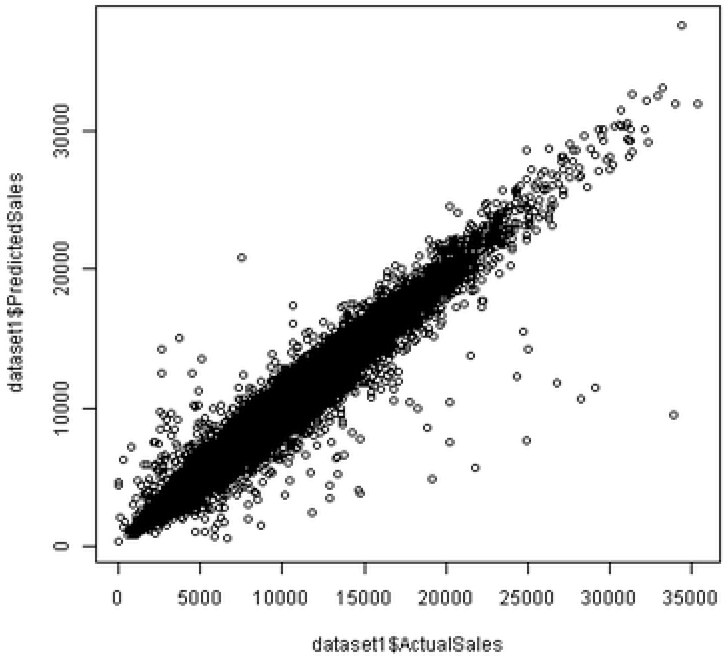
Submissions are evaluated on the Root Mean Square Percentage Error (RMSPE). The RMSPE is calculated as

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2},$$

where y_i denotes the sales of a single store on a single day and yhat_i denotes the corresponding prediction. Any day and store with 0 sales is ignored in scoring.

Forecasted results are uploaded to Kaggle which computes the public and private scores for us. Public scores are computed of a small hold out dataset while private scores are computed on a significantly larger dataset. Besides these scores we have also used the Coefficient of Determination (R2), which is computed by the Evaluate Model Module.

Our Model had an R2 of 0.957 which is pretty good for any regression problem. Our private score is 0.123, while the winner of the competition had a private score of 0.100. below is a plot of actual versus predicted sales of a hold-out dataset, consisting of 10% of dataset, randomly selected.



Further Improvements

The following things have a potential to improve the predictive power of our model and were utilized by people who performed the best on Kaggle competition. These were not pursued due to lack of time, the idea was to get acquainted with the application of forecasting to a real life retail problem.

- Using a richer feature set that uses the locality of store, demographic data and regional sales trends.
- Creating a hybrid technique that uses both the ARIMA model and Decision trees might result in a more stable prediction.
- If problem did not require us to predict 6 weeks into the future, recent sales average values like average daily sales in the last week could have been very useful predictors.