Predicting Sales: Rossmann Drug Company

ISE 5103: Intelligent Data Analytics

Forecasting Rossmann Drug Company sales across Germany using Data Analytic Techniques

Ivan Calderoni, Vanadda Sermpol, and Parish Kaleiwahea

EXECUTIVE SUMMARY:

Rossmann, a drug store from Germany operating over 3,000 stores across Europe, requires their store managers to predict their daily sales for up to six weeks in advance. Since the store sales are influenced by many different factors such as seasonality, location of the store, holidays, etc., store managers have predicted their sales based on their unique methods, reasoning and formulas that have caused the accuracy of results to vary and be inconsistent throughout the stores.

Rossmann provided historical data sales for 1,115 stores through Kaggle, which account for all of their stores located in Germany. The data is all found in three different documents: train, test, and store files. The task at hand is to forecast the "Sales" column for the test set.

Our project aims to create a few robust models and then compared them with each other. The first model was created with the use of Random Forest. It resulted with a Kaggle score of 0.13038. The Random Forest model will be used as a benchmark to compare the future models we create.

More conclusions will be drawn for the final draft of this project.

PROBLEM DESCRIPTION:

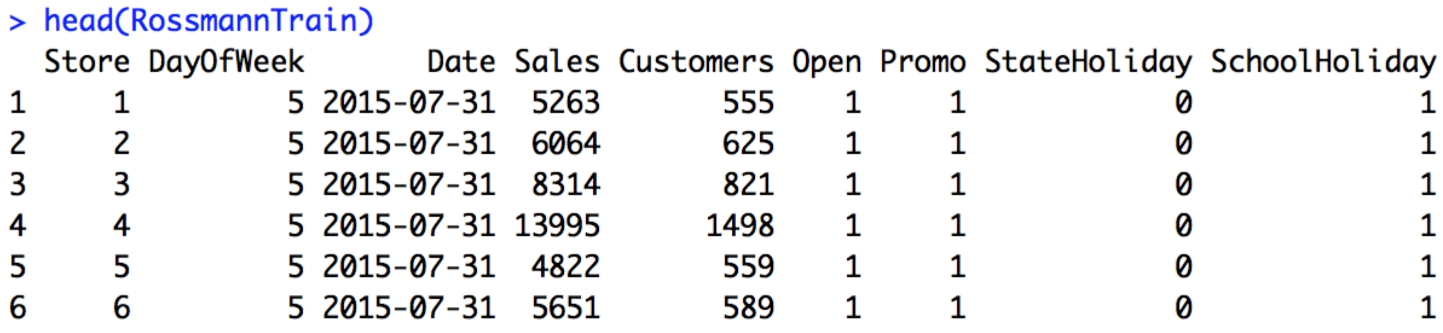
The dataset comes from a Kaggle competition hosted by Rossmann Stores to forecast sales for up to six weeks in advance. Sales forecasting is the process of estimating future sales and when this is done accurately, it enables companies to make informed business decisions. A few of the many benefits that arise from successfully looking ahead include: ensuring one can meet the customer's needs and demands; avoid missing sales opportunities; improved financial management; and adequate staffing. Any business or company would strongly benefit from having accurate and meaningful models that can guide and facilitate many of the hard business decisions that have to be made.

The data provided Rossmann can be found here:
https://www.kaggle.com/c/rossmann-stores-sales

It consists of a Training, Testing and Store datasets. The Training dataset contains 1,017,209 records and the Testing dataset contains 41,088 records. These two datasets provide historical sales data ranging from January 1, 2013 through July 23, 2015.

The Training data includes the following predictors: Store, DayOfWeek, Date, Sales, Customers, Open, Promo, StateHoliday, and SchoolHoliday. The Testing dataset contains the same predictors except for the Sales variable which is what the models in this project will predict.

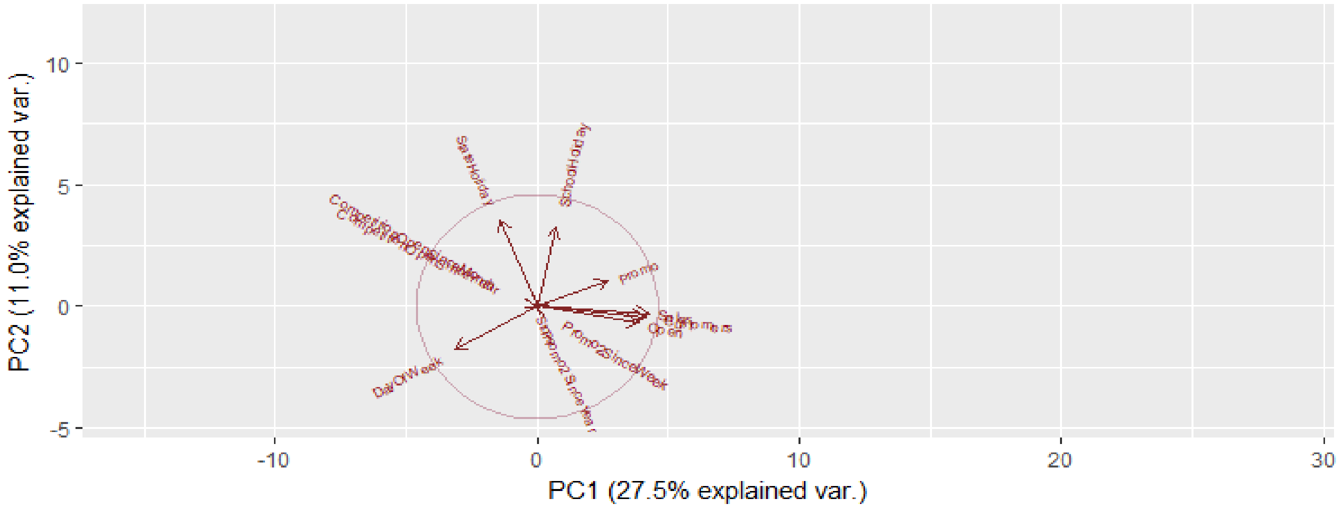
The Store dataset contains the following predictors: Store, StoreType, Assortment, Competition, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2, Promo2SinceWeek, Promo2SinceYear, and PromoInterval. This dataset is supplementary information that can be used along with the historical sales dataset. "Competition" here refers to a competitor store or business.



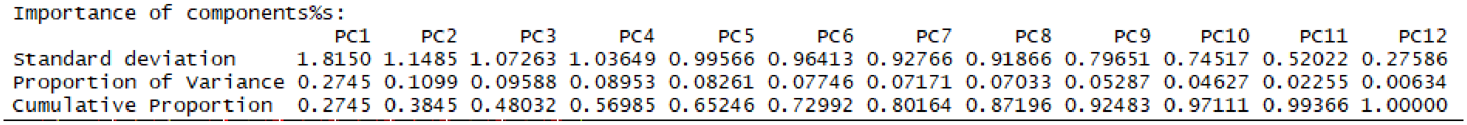
```
> head(RossmannTrain)
  Store DayOfWeek       Date Sales Customers Open Promo StateHoliday SchoolHoliday
1     1         5 2015-07-31  5263       555    1     1            0             1
2     2         5 2015-07-31  6064       625    1     1            0             1
3     3         5 2015-07-31  8314       821    1     1            0             1
4     4         5 2015-07-31 13995      1498    1     1            0             1
5     5         5 2015-07-31  4822       559    1     1            0             1
6     6         5 2015-07-31  5651       589    1     1            0             1
```

Figure 1: A snapshot of the training dataset.

FEATURES:



The PCA analysis given in the table and the graph was disappointing because it did not get the results we had hoped. We wanted to account for more variance in the first 3 components but it was less than 50% of the total variance. We will be cautious when using the PCA analysis, but some of the analysis requires fewer dimensions.



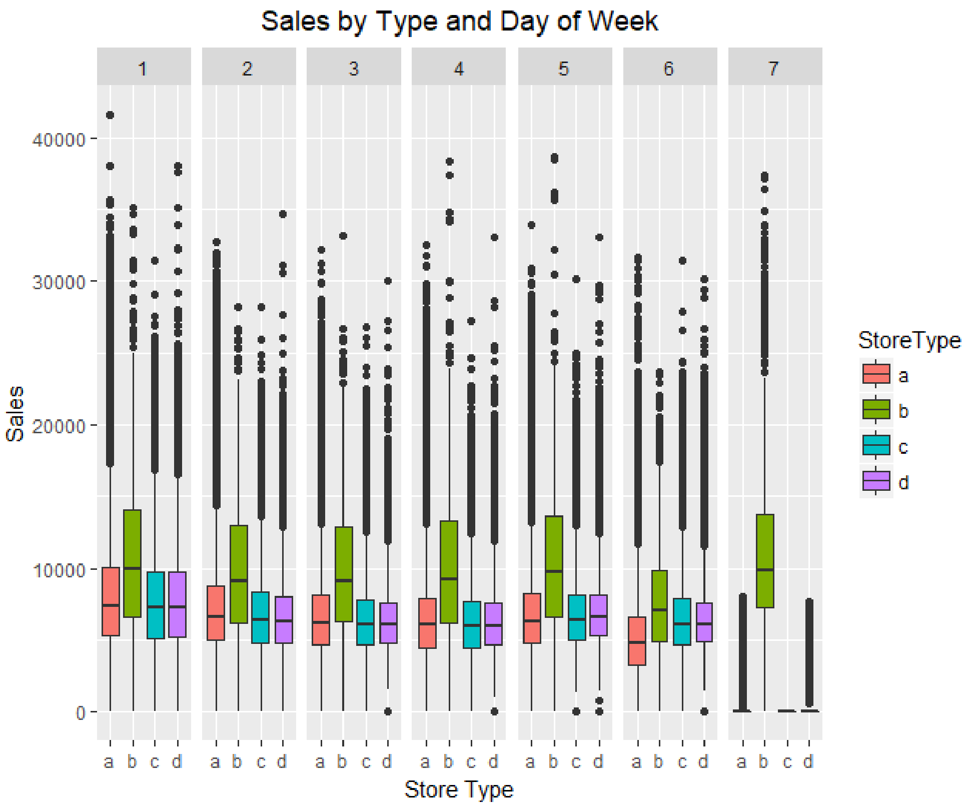
```
Importance of components%s:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12
Standard deviation     1.8150 1.1485 1.07263 1.03649 0.99566 0.96413 0.92766 0.91866 0.79651 0.74517 0.52022 0.27586
Proportion of Variance 0.2745 0.1099 0.09588 0.08953 0.08261 0.07746 0.07171 0.07033 0.05287 0.04627 0.02255 0.00634
Cumulative Proportion  0.2745 0.3845 0.48032 0.56985 0.65246 0.72992 0.80164 0.87196 0.92483 0.97111 0.99366 1.00000
```

We will explore more feature engineering as the analysis continues. One path forward is to add seasonality as a factor. From our research many retailers mention Christmas as a month they create. We know black Friday is the most important day for retailers and perhaps there could be a specialized model for that particular date.

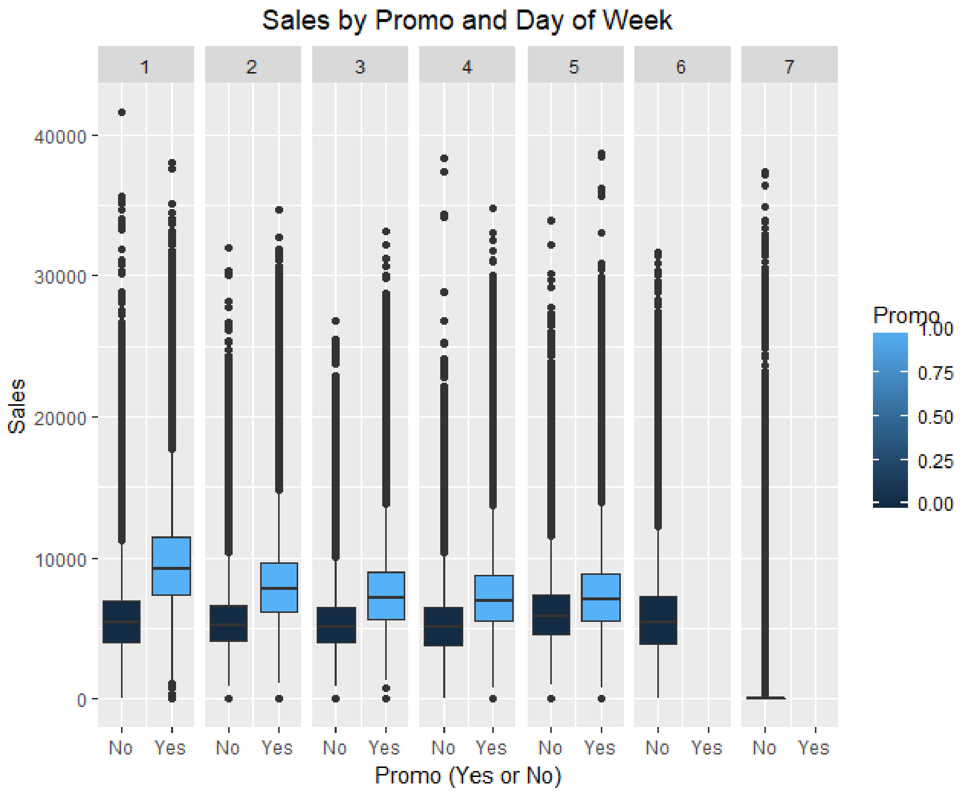
CLEANING:

The original data frame was very clean, despite having over 100,000 rows, however it only had 8 columns of data. Our data came with additional information about stores which decreased the columns in the training set. To get the information in the training data frame we needed to join the two data frames based on store code. It was critical that we got more data because we did not have enough for exploratory analysis. Joining the two data frames required more cleaning to be used in modeling. First we had to separate the factors using DPLYR and dummy code the variables. Once given its own rows we were able to increase our effectiveness of our model based on separating the store type and store assortment.

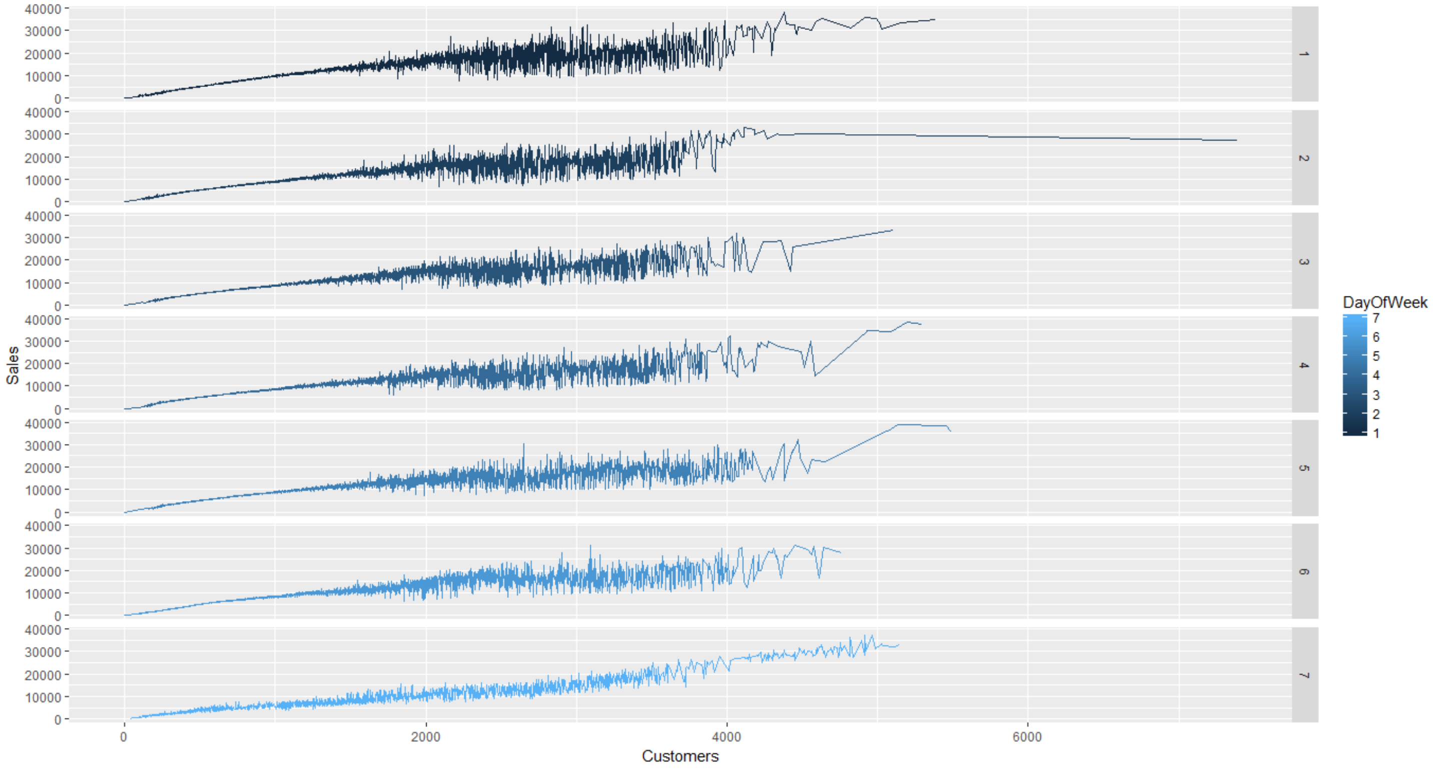
EXPLORATORY DATA ANALYSIS:



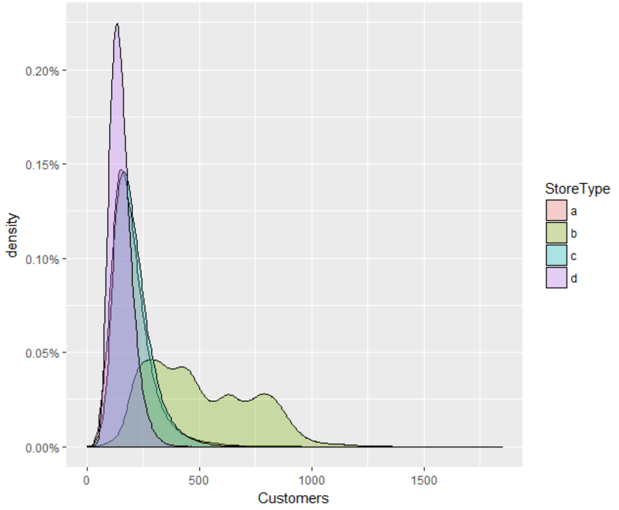
Graph 1: It is important to explore the data through further segmentation. In boxplot one it shows how store type b generates far more sales than the other type of stores. The median sales exceeded the third quartile of the other stores, however Store B doesn't share as many outliers as others.

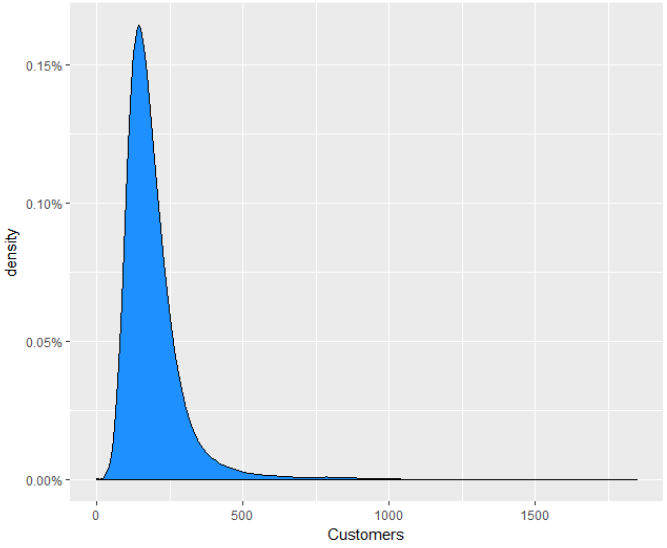


Graph 2: There data for the Rossman stores included when a store ran a promotion. We wanted to segment the data to discover if a promotion improved the sales. As expected there was a noticeable bump in sales when a store ran a promotion. Further segmenting the data we split the promotions by day of the week. Monday consistently outsold other days of the week by a higher rate than non promotional sales.



In the graph above it shows the linear relationship of customers to sales. It is obvious that the more customers you have the more sales you would get. One thing that is very interesting is that the relationship of customers to sales seem to fall apart at a certain point. ~1500 customers the linear scale gets very noisy, and it's unclear what the data suggestions about the relationship. To tease that out further the graph to the left shows the density of customers. It is an EXTREMELY rare occurrence to get greater than 1500 customers per day.

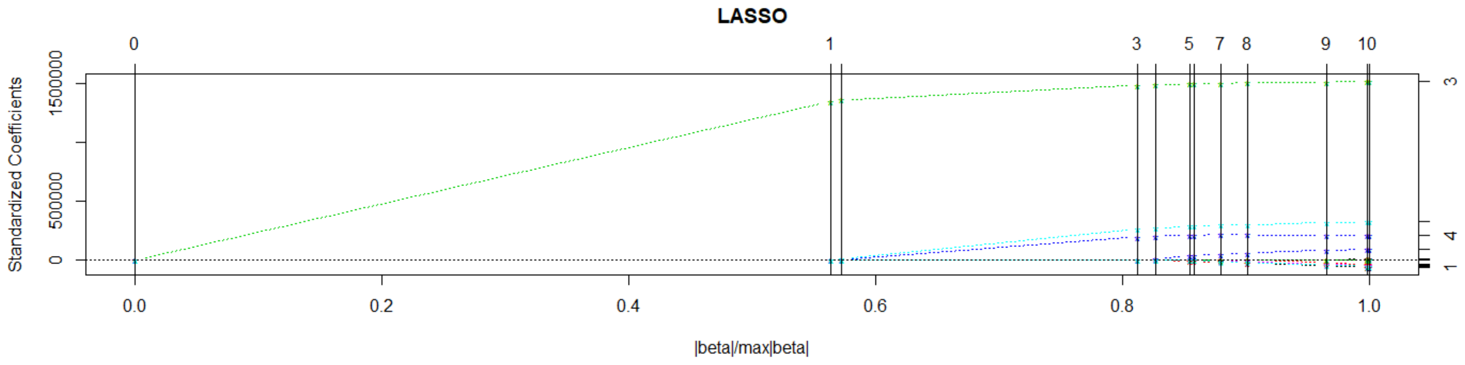


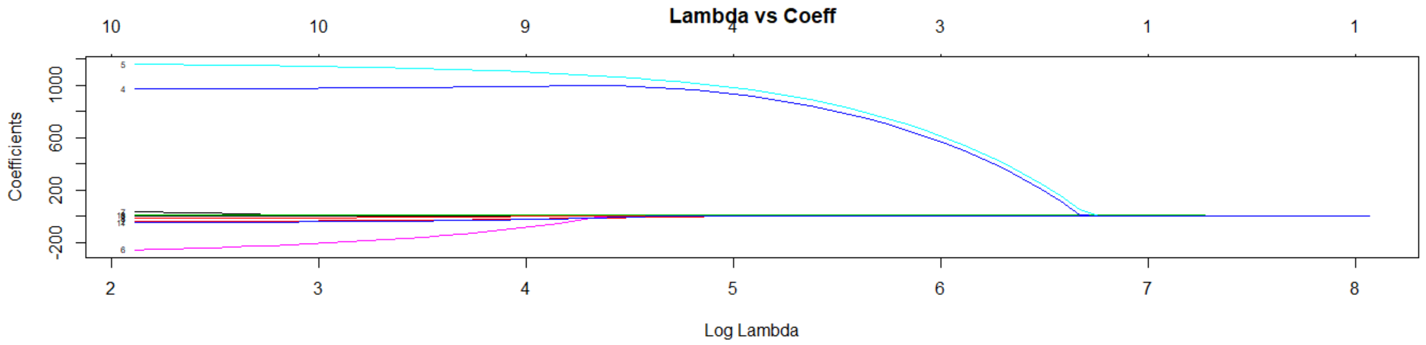


ANALYSIS PLAN:

LASSO:

Least absolute shrinkage and selection operator (LASSO) is one of the techniques used to develop a family of models. We identified LASSO as a technique that would help us minimize features and simplify the model. Feature engineering will be pivotal because there aren't a lot of variables to begin with and creating new features could be a good way forward to improve a different model technique besides LASSO.

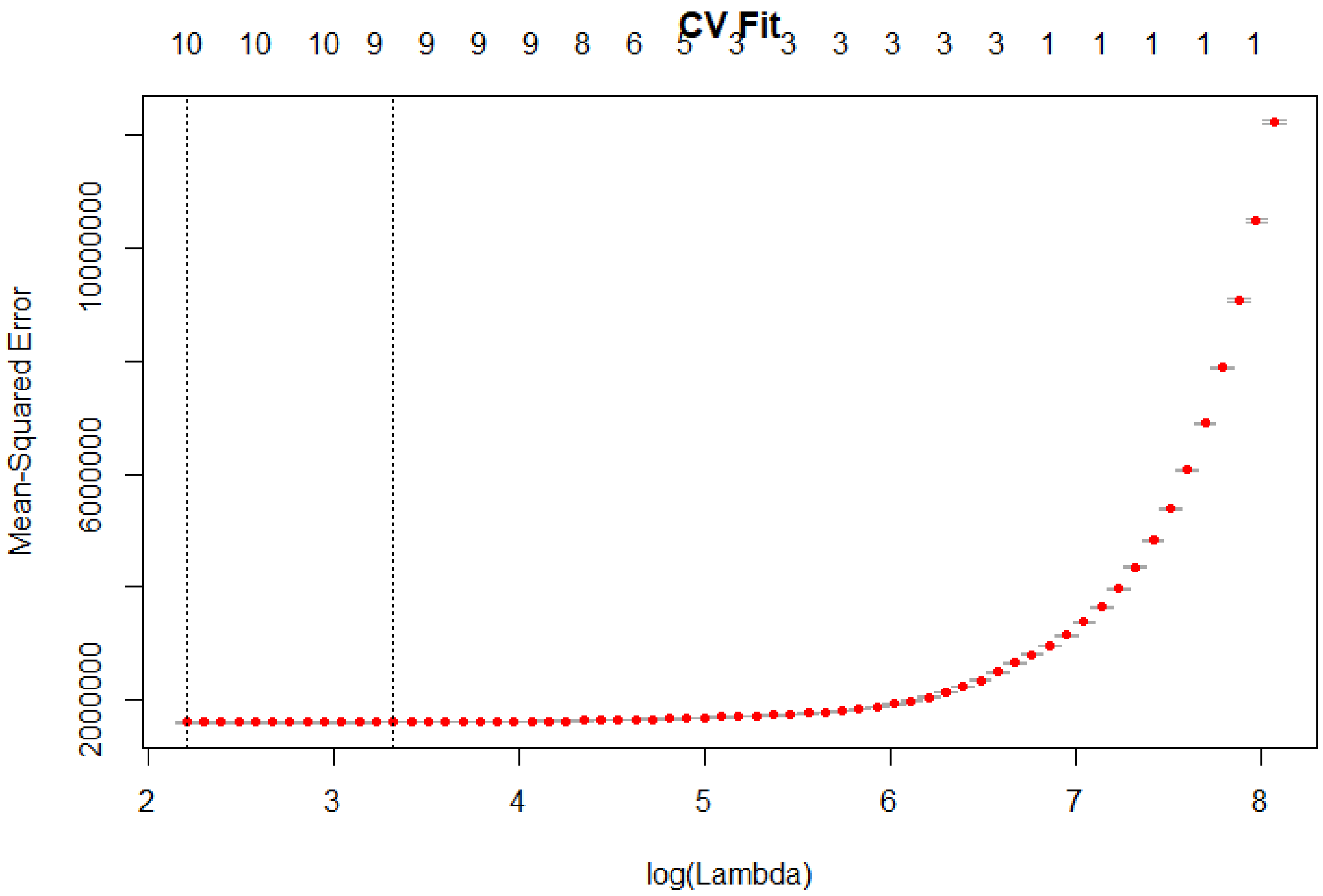




Our first step with LASSO was figuring out the ideal Lambda. The model created a selection of many different choices as pictured graphically above. Notice how 3 there were 2 very prominent features that were positive, and many that were bunched near a coefficient of zero. This was a signal that there were a few features that were carrying most of the weight for the model. I tentatively assume a lambda around log(6) but wanted to apply cross validation to verify my initial thoughts.

K Folds Cross Validation

K fold cross validation was our technique used to validate the dataset. The family of models produced interesting results best summarized in the graph below.



The minimum lambda was 9.082 and the maximum was 3188.719. We didn’t want to go overboard picking the model that had the best results because we did not want to overfit the model. As we suspected earlier there was not a lot of mean square error gains between 3 variables and 10. Based on the results we selected a lambda of 178.3 for our LASSO model.

Model Selection

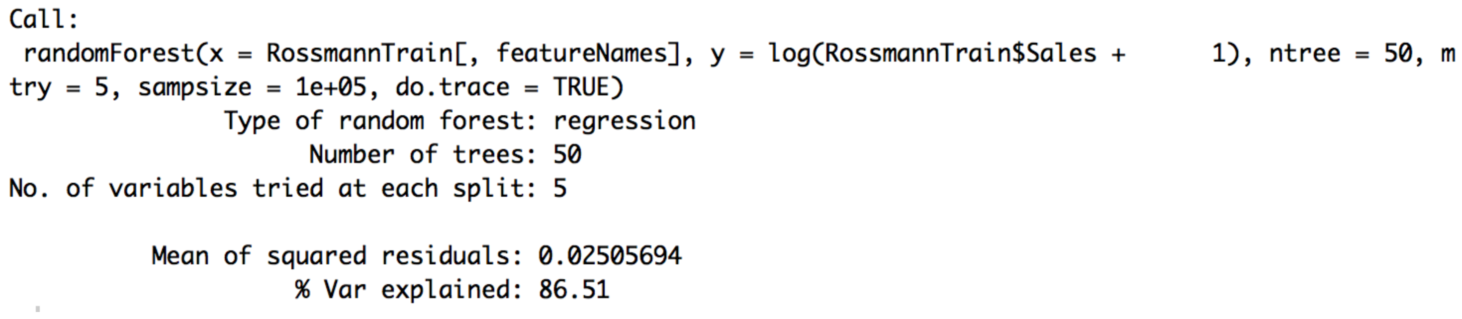
RMSE- 6306.843

Random Forest:

Random Forest was selected for one of the models created to forecast sales. It is a type of ensemble learning method where a group of “weak” models combine to form a more powerful model. This algorithm can solve classification and regression problems and it has the power to handle large data sets with high dimensionalities. In other words, it can handle thousands of input variables and identify the most significant variables. By doing so, it is a way of reducing dimensionality.

Random Forest has an effective and powerful method for estimating missing data and for balancing errors when classes in a particular data set are unbalanced. When it comes to training data, Random Forest uses different random amounts of data to train. By doing so, it prevents the model from overfitting.

Some of the major disadvantages of using Random Forest include not allowing the user having much control on what the model does – the most tuning a user can do is trying out different parameters and changing the random seed. Typically, Random Forest does better at classification than regression problems. When it comes to regression problems, it does not predict beyond the range in the training data.

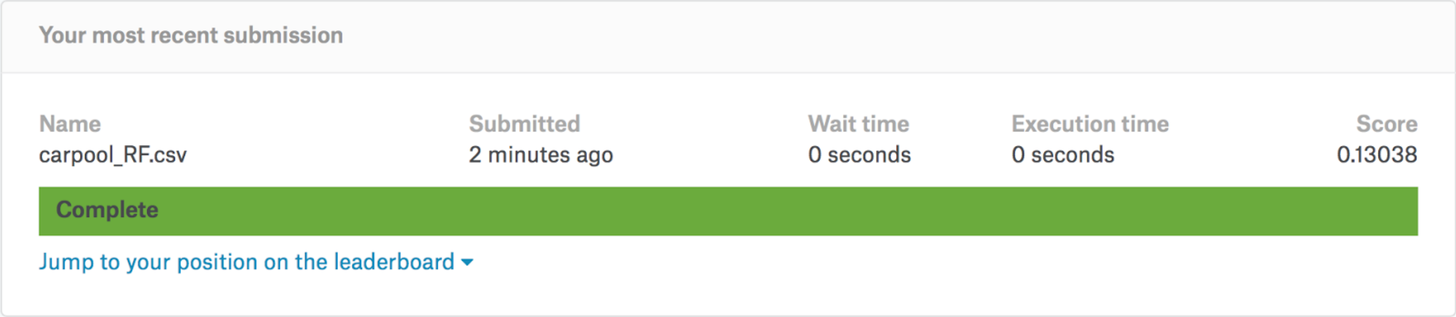


```
Call:
 randomForest(x = RossmannTrain[, featureNames], y = log(RossmannTrain$Sales +      1), ntree = 50, m
try = 5, sampsize = 1e+05, do.trace = TRUE)
               Type of random forest: regression
                     Number of trees: 50
No. of variables tried at each split: 5

          Mean of squared residuals: 0.02505694
                    % Var explained: 86.51
```

The Random Forest model created is a regression type model. It was set to 50 number of trees to grow. The larger the number of trees to grow the more stable the model and covariate importance estimates are. However, for a dataset of this size, 50 trees is sufficient. It explained about 86.51% percent of the variance.

RESULTS AND VALIDATION OF ANALYSIS:



Your most recent submission

Name	Submitted	Wait time	Execution time	Score
carpool_RF.csv	2 minutes ago	0 seconds	0 seconds	0.13038

If the competition was still live, this model would place us in position 1,681 out of 3,304 teams.

THINGS STILL NEEDED TO BE DONE:

Completion of more models. The use of different techniques, more feature engineering, conclusions and recommendations will be done for the final report. The models will be evaluated using multiple validation metrics on top of the Kaggle score.