Predictive Analysis Executive Summary

Part 1. Method

1. Linear regression:
    a. Backward stepwise selection (Using log for Y and Price)
    b. Forward stepwise selection (Using log for Y and Price)
    c. Lasso (with 100 lambda from 10^10 to 10^-2)
    d. Ridge (with 100 lambda from 10^10 to 10^-2)
2. Tree:
    a. Regression Tree after Pruning
    b. Bagging, Random forest (after optimizing mtry from 1 to 95)
    c. Boosted Tree (after optimizing shrinkage when n.tree set 1000 constant, then optimize number of trees)
3. KNN (K = 5)
4. MARS (degree = 2)
5. Neural Network (Hidden layer = 3)

Part 2. Best result (RMSE)

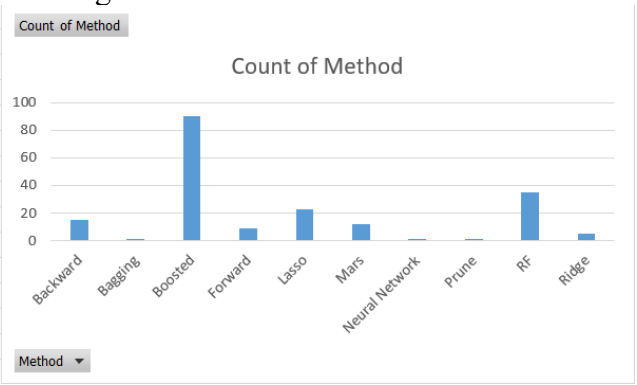
Best_RMSE	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	item_9	item_10	item_11	item_12	item_13	item_14	item_15	item_16	item_17	item_18	item_19	item_20	item_21	item_22	item_23	item_24
Store_1	462.772	15.6623	86.599	326.006	28.044	540.757	197.892	259.272	276.466	110.715	368.086	152.361	182.076	180.575	143.31	119	30.0085	697.21	213.23	88.1278	724.71	13.27	427.115	172.32
Store_2	351.651	13.8228	104.169	179.812	14.7675	349.282	118.923	177.01	136.428	87.2826	350.286	136.561	102.558	86.3517	134.897	54.1165	13.7193	394.575	139.803	70.8047	663.415	7.73361	100.966	71.335
Store_3	519.965	12.705	52.9796	208.9	18.0286	431.185	138.232	172.538	214.85	119.848	369.477	131	98.405	91.8231	81.4814	62.4581	14.8818	460.762	218.321	71.8477	556.428	8.29127	235.165	71.376
Store_4	406.323	7.65836	99.7753	173.401	18.4576	375.692	126.517	172.879	157.3	110.012	274.649	147.316	126.795	150.777	104.686	184.383	13.6287	447.434	209.971	83.1869	575.685	7.13982	221.578	113.081
Store_5	267.957	16.8838	94.7346	229.064	30.9689	527.033	177.529	149.36	172.338	76.6234	358.196	104.636	79.244	73.532	102.975	83.9778	22.1461	591.325	188.437	72.6017	718.676	12.3202	246.328	125.72
Store_6	418.995	15.0641	120.279	170.212	26.2744	555.229	120.819	206.956	174.136	142.829	496.651	142.927	113.198	98.4984	119.941	189.955	21.2182	389.326	172.699	68.3656	698.634	11.5629	148.899	127.434
Store_7	622.359	21.9247	101.302	287.945	30.1669	555.484	204.655	237.823	246.045	197.493	598.784	207.081	152.268	129.709	138.167	77.2976	22.6832	511.092	259.181	96.9594	922.889	13.5799	244.165	108.852
Store_8	498.079	7.16275	68.2793	181.025	22.4289	349.937	80.0104	120.096	152.253	110.856	201.564	134.569	240.7	193.096	64.4676	48.8253	17.7149	408.646	199.693	60.6653	349.834	5.43405	153.534	65.4237

Corresponding method: (RF: Random Forest)

Best_Method	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	item_9	item_10	item_11	item_12	item_13	item_14	item_15	item_16	item_17	item_18	item_19	item_20	item_21	item_22	item_23	item_24
Store_1	Boosted	Boosted	Boosted	Lasso	Lasso	Lasso	Boosted	Boosted	Lasso	Boosted	RF	Ridge	Backward	Boosted	Lasso	RF	Backward	Boosted	Boosted	Backward	RF	Backward	Backward	Boosted
Store_2	Boosted	Boosted	Boosted	Lasso	RF	Boosted	Lasso	Boosted	Boosted	Boosted	Boosted	Boosted	Mars	Boosted	RF	Ridge	RF	Boosted	Boosted	Lasso	Boosted	RF	Boosted	Boosted
Store_3	Forward	RF	RF	Lasso	Lasso	RF	Boosted	Boosted	Boosted	Boosted	Lasso	RF	RF	RF	Lasso	RF	RF	Boosted	Bagging	RF	Lasso	Prune	Boosted	Boosted
Store_4	RF	Boosted	Mars	Boosted	RF	Boosted	Backward	Boosted	Boosted	Boosted	Backward	Boosted	Forward	Backward	Boosted	Mars	RF	Boosted	Backward	Boosted	Forward	Backward	Backward	Boosted
Store_5	RF	RF	RF	Boosted	Boosted	Boosted	Boosted	Boosted	Mars	Boosted	Boosted	RF	RF	RF	Boosted	RF	Boosted	Boosted	Lasso	Boosted	Mars	Lasso	Boosted	Boosted
Store_6	Boosted	Boosted	Mars	Boosted	Mars	Boosted	Lasso	Boosted	Boosted	Boosted	Boosted	Boosted	Backward	Boosted	Boosted	Forward	RF	Boosted	Boosted	Lasso	Boosted	Lasso	Mars	Boosted
Store_7	Boosted	Boosted	Forward	Boosted	Backward	Lasso	Boosted	Backward	Forward	RF	Boosted	Forward	Forward	Boosted	Ridge	RF	Boosted	RF	RF	Lasso	Backward	RF	Forward	Lasso
Store_8	Mars	Boosted	RF	Boosted	Mars	Boosted	Boosted	Boosted	Ridge	Boosted	Lasso	Ridge	Mars	Mars	Boosted	RF	Neural Ne	Lasso	Boosted	Boosted	Boosted	RF	Boosted	Boosted

The count of each method being the best one

Method	Count of Method
Backward	15
Bagging	1
Boosted	90
Forward	9
Lasso	23
Mars	12
Neural Network	1
Prune	1
RF	35
Ridge	5
Total	**192**



Boosted Tree generates the best result generally.

Part 3. The setting of Boosted Tree and why it performs better

Boosted Tree: We ran a for-loop to find the optimal shrinkage range from 0.01 to 1 when setting n.tree = 1000 and interaction.depth = 6. Then we set the best shrinkage for each product in each store, and find the best number of trees from 500 to 5000 (distance is 100) using for loop as well. As a result, boosting method performs very well, becoming the winning model in our case. Compared to bagging and random forest, boosting fits a tree using the current residuals, rather than

the outcome Y. Small shrinkage and fewer leaves slows it's growing process, generating better results in general.

Part 4. Why some methods don't provide good prediction

KNN: This method does not apply to high dimension problem, since it is hard to find closer neighbor in a given range in high dimension situation.
Neural Network: This method mostly applies to situations where there is a large number of data, our dataset only contains around three hundred data points, which is far from a large dataset. The processing step does not get enough data to optimize each layer's coefficient, thus it does not provide good prediction.
Forward & Backward: We first try the model without taking log to any variables, we found that the RMSE is large. So, we decided to take log to Y(sales) and P(price), and the RMSE reduced significantly. We can safely conclude that the relationship between variable P and rest of the variable is not linear but rather exponential. When choosing the independent variables, we just tested BIC and use the minimum point to fit the total model. While there are other parameters we did not take into consideration, such as AIC, R2, adjusted R2 and so on. With only one training data set, this model cannot work well on the testing data set.
Mars: We set degree as 2, but in real life, there might be other interactive variables we haven't thought about. With no tracing, the model is also not the optimal one and if we set trace as 3 the result after pruning might be better.
Regression Trees: We first built regression trees and then used cross validation to find the best number of trees, using which pruned the tree. The tree itself may not produce the best results because it may overfit the training data. After pruning, the results slightly improved but it's still not the best model.
Bagging: We set seed 1 and then implement the random forest function setting mtry = p. The reason why it's not the winning model for most of the times is that bagging has the potential risk that the existence of very strong variables will lead to the similarity of different trees, which makes the average of variance not very reliable.
Random Forest: After complete Bagging, in each store and each product we loop through 1 to 95 to find the optimal mtry using for statement. As it turns out, random forest is an improvement over bagged trees. It makes the average of the resulting trees less variable and hence more reliable. That's why random forest tends to perform better than bagging for most of the times.

Part 5. Relevant Groceries

After plotting 192 regression trees, we find that the sales of cold cereal could be affected by the price of soup; Sales of diapers have a connect with the price of beer; Sales of frozen pizza relates to the features of beer; Sales of hot dogs and beer relates to the promotion of soup; Sales of margarine & butter relates to the display of mustard & ketchup; Sales of soup connects to the promotion of mustard & ketchup.