Prediction of Store Performance
Statistics 561
Sayan Mukherjee

Sophie Guo
Kevin Legein
Artur Shikhaleev
Justin (Zihao) Zhang

December 11, 2015

Abstract

The company Dirk Rossmann GmbH—usually known as just "Rossmann"—is Europe's second-largest drugstore chain. They made sales data from their stores available to `www.kaggle.com/`. In this paper, we explore how to use several business techniques—k-means clustering, decision trees, support vector machines, and hidden Markov models—to predict which stores perform, on the average, better than others and to try to see why.

1 Introduction

Store managers must be able to predict sales in some way: They need this to determine pricing, decide on inventory quantities, forecast demand, and, above all, to decide whether a store is performing well. Only when they know whether a store is up to scratch can they decide how to increase productivity.

Rossmann is Germany's second-largest drugstore chain; it operates in seven Euorpean countries. The competition that offered these data initially asked competitors to forecast sales; but, in light of this course's focus on classification—and the lack of continuous predictors—we decided to determine whether a store could be classified as "good" (*i.e.*, has sales that exceed the median) or "bad" (has sales that fall short of the median) in addition to predicting sales

These data includs 1,115 stores over about three years. It includes, most notably, the nearest distance to a competitor and the store's age. So, in this analysis, we seek to answer the question: How does a store's performance depend on its age and that distance? We also seek to decide whether we can predict a store's sales from the independent variables given.

1.1 Explanation of Data

The data included:

- Day of the week

- Number of customers on a given day
- Whether the store was open on that day
- Whether there was a holiday
- The type of store
- Its distance to the nearest competitor
- The date on which that competitor opened
- Whether the store is running a promotion
- When that promotion began

We wished to categorize a store as a failing enterprise, an average one, or a successful one; *i.e.*, tell the company whether a given store should remain open.

2 Analysis and Techniques

2.1 Method: k-Means Clustering

Introduction to the Method: Given a large number of observations, k-means clustering gives the solution with the least processing time required. k-means clustering aims to partition n observations into k clusters, where each observation belongs to the cluster with the nearest mean. Given a set of observations $\{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_n}\}$, where $\vec{x_i} \in \Re^p$ for all i, k-means clustering tries to separate observations into k sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (sum of distance functions of each point in the cluster to the k center).

88.7 percent of variation is explained by the variation among clusters. Hence, the stores can be divided into 3 categories: small distance, median distance, and big distance. Competition distance seems to be a very important parameter in predicting sales of a shop. This is not surprising and is confirmed by the analysis. Clusters are illustrated in a plot below 1. As the distance between the nearest competitor increases, the sales increase because a customer has fewer options.

.

Logged Sales	Weekday	Month	Promo	State Holiday	School Holiday	Distance
7.24	4.00	5.83	0.38	1.05	0.18	**17156.78**
7.20	4.00	5.78	0.38	1.05	0.18	**7284.79**
7.27	4.00	5.78	0.38	1.05	0.18	**1476.51**

2.2 Method: LASSO

Introduction to the Method: MLE results usually have high sampling variance, which leads to high expected errors. As a result, regularization is introduced to trade bias for reduced variance. LASSO regression adds a penalty term to shrink coefficients towards zero. Specifically, it takes the ℓ_1 norm, which causes the minimum $\beta \in B$ to occur on a vertex, which is to say where a number of terms are 0. So the minimization problem becomes

$$\hat{\beta} = \arg\min_{\beta \in B} \frac{1}{n} \sum (y_i - x_i^T \beta)^2 + \lambda ||\beta||_1.$$

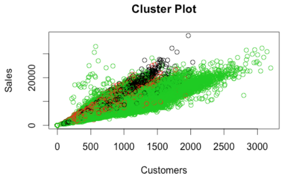


Figure 1: Cluster Plot

The figure 2 illustrates this principle:

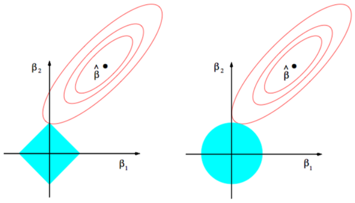
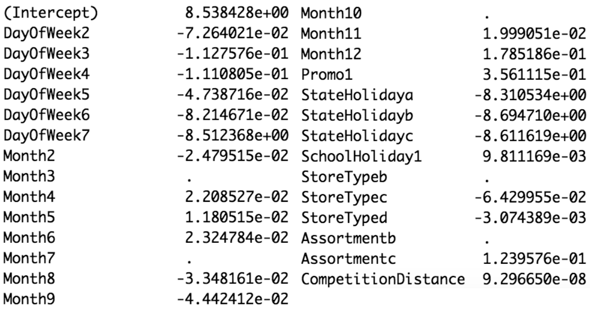


Figure 2: The Principle of LASSO Regression

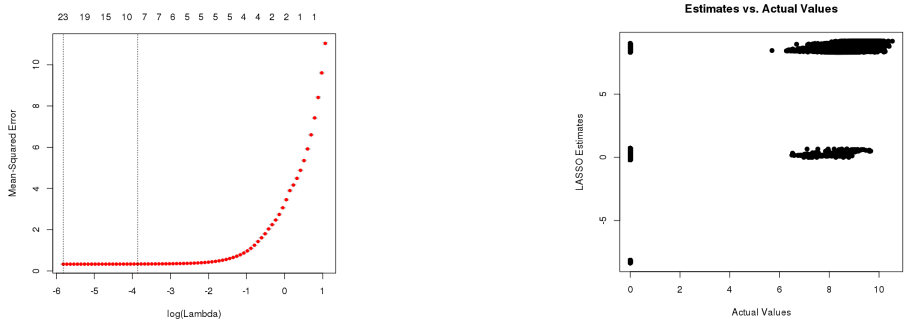
Results: The covariates included are DayofWeek, Month, Promo, StateHoliday, SchoolHoliday, StoreType, Assortment, Competition Distance and Promo2. Because the training data set is too large with more than 300,000 entries, we randomly select 20,000 samples from the training data set and train the regression model on these sampled data.

From the table 3 below, we can see that the majority of the coefficients in the trained LASSO model are non-zero. The covariates that have the largest effect on sales are state holidays and Sundays. Surprisingly, they both have negative effect on store sales—one would think people did more shopping on those days. The average store sales on any state holiday are about equal with those of regular days. The differences between sales on Monday, the baseline in the model, and sales on any other day of a week are negligible except for Sunday. The store sales on Sunday are on average. One possible explanation is that drug stores may not the top shopping destinations during holidays.



```
(Intercept)        8.538428e+00 Month10                .
DayOfWeek2        -7.264021e-02 Month11             1.999051e-02
DayOfWeek3        -1.127576e-01 Month12             1.785186e-01
DayOfWeek4        -1.110805e-01 Promo1              3.561115e-01
DayOfWeek5        -4.738716e-02 StateHolidaya      -8.310534e+00
DayOfWeek6        -8.214671e-02 StateHolidayb      -8.694710e+00
DayOfWeek7        -8.512368e+00 StateHolidayc      -8.611619e+00
Month2            -2.479515e-02 SchoolHoliday1      9.811169e-03
Month3             .            StoreTypeb           .
Month4             2.208527e-02 StoreTypec         -6.429955e-02
Month5             1.180515e-02 StoreTyped         -3.074389e-03
Month6             2.324784e-02 Assortmentb          .
Month7             .            Assortmentc         1.239576e-01
Month8            -3.348161e-02 CompetitionDistance 9.296650e-08
Month9            -4.442412e-02
```

Figure 3: LASSO Output



(a) Mean Square Error *versus* λ

(b) Estimates against Actual Values

Figure 4

Predictive Power: If the LASSO model predicts sales well, we would expect to see a straight line in the the plot of estimates against actual values 4b; but we see two clusters of points and the predicted values within each group lie almost on a straight horizontal line. This result is not surprising, since most of our variables are categorical with only one continuous covariate, CompetitionDistance. Unless CompetitionDistance is directly proportional to store sales, we will not be able to predict the actual value of sales with great accuracy. Therefore, we decide to turn this sales-prediction problem into a classification problem. Instead of predicting the actual sales, we create three different categories of stores - successful stores, average stores, and failing stores, and predicted which category the store will fall into on a particular day. Another direction we can explore in the future is to add a time series component to our model. By including the sales of the store on the previous days, we can establish a baseline for the sales and predict the sales more accurately.

2.3 Method: Decision Trees

Introduction to the Method: Classification and regression trees are machine-learning methods for constructing prediction models from data. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree.

Results: Before we shift to the classification problem, we also attempt the random forest regression. As with before, we sample 20,000 data from the training data set to train the random forest model. With the default 500 trees, the random forest model explains about 97 percent of the variation. This extremely high rate may be caused by the log scale of the response variable, but it also raises concern about the random forest regression model fitting noise. Therefore, we decide to plot the fitted values against the actual values. The `R` output is in the appendix (A1)

```
Call:
 randomForest(formula = log(Sales + 1) ~ DayOfWeek + Month + Promo +
                                        StateHoliday + SchoolHoliday +
                                        StoreType + Assortment +
                                        CompetitionDistance + Promo2,
            data = joined_sample)

            Type of random forest: regression
            Number of trees:  500
            No. of variables tried at each split:  3

                Mean of squared residuals: 0.2842065
                % Var explained: 97.47
```

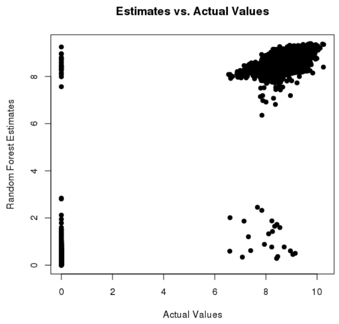
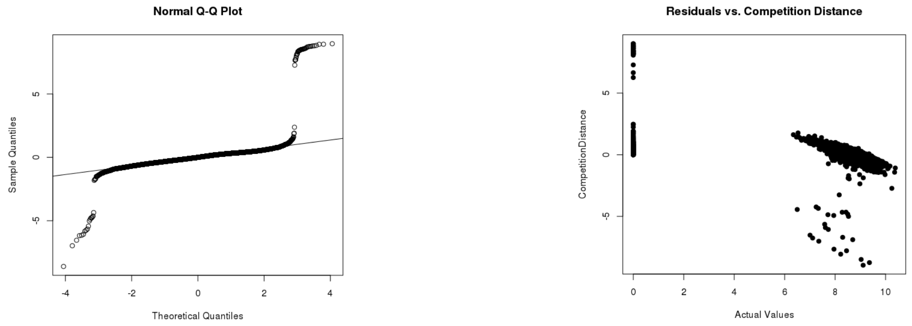


Figure 5: Random Forest Estimates against Actual Values

From the plot above, we can see that there exists some linear relationship between the estimated values and the actual values. In that respect, random forest regression performs better than LASSO in predicting the actual values. This improvement may be attributed to the fact that random forest regression first splits the data and performs regression separately on each group instead of fixing the coefficients for numerical variables for all groups as in LASSO.

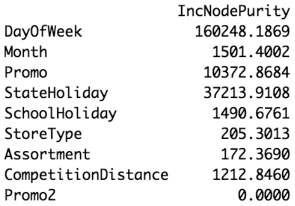
From the QQ-plot 6a, we can see that the majority of the points follows the normal line while the points at the two tails deviate significantly from the line. The residual plot confirms the existence



(a) Normal Quantile *versus* Quantile Plot

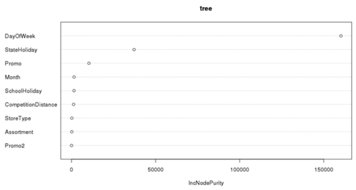
(b) Residuals Plotted against Competition Distance

Figure 6



```
                     IncNodePurity
DayOfWeek              160248.1869
Month                    1501.4002
Promo                   10372.8684
StateHoliday            37213.9108
SchoolHoliday            1490.6761
StoreType                 205.3013
Assortment                172.3690
CompetitionDistance      1212.8460
Promo2                      0.0000
```

(a) IncNodPurity Table



(b) IncNodePurity Graph

Figure 7

of outliers, which are mostly failing stores. Because we take out the stores that are not open from the training data set, the 0 sales are most likely caused by incorrectly entered data.

IncNodePurity gives a measure of the importance of a variable in random forest for regression. It is calculated by averaging over all trees the total decrease in node impurity from splitting on that variable. From the table and the plot in the plot above 6b we can see that DayOfWeek is the most important predictor in reducing node impurity in random forest followed by StateHoliday and Promo. This is consistent with our result from LASSO regression, which shows that Sunday and StateHoliday are the two most important predictors of sales.

2.4 Method: Support Vector Machines

Introduction to the Method: Support vector machine is a supervised learning method used widely in classification tasks. It is a linear classifier that, given data points in $\Re^p$, finds a hyperplane that has the greatest margin to the closest data point.

We are looking for a hyperplane $< \vec{w}, \vec{x} > +b = 0$ with the condition:

$$\arg\min_{w,b,\varepsilon} \left(\frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \varepsilon_i \right), \text{ where } \forall i, y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i$$

$\frac{1}{2}||w||^2$ comes from maximizing the margin while ϵ_i are slack variables are the penalties for misclassification.

Non-linear support vector machine is used in cases where the data itself is not linearly separable. It maps the data points into higher dimensional space Linear support vector machine and other linear classification methods rely on a linear kernel $K(\vec{x_i}, \vec{x}) = \vec{x_i}^T \vec{x}$, leading to a classifier of the form $f(\vec{x} = \text{sign}(\sum \alpha_i y_i K(\vec{x_i}, \vec{x}) + b)$. In the non linear support vector machine, the data points are first mapped to some high dimensional space and use a kernel function that corresponds to a dot product in that space.

Results: We randomly sample 20,000 data from the training data set and categorize each entry based on the sales values. If the sales of that store on that day is among the bottom 25 percent of all sales, we label it as "bottom"; if it is between the 25 percent– and 75 percent–quantiles, we label it as "average"; if it is among the top 25 percent, we label it as"top". Then we train the support vector machine on the sampled data. When we use this model to predict the category of the rest of the training data, the misclassification rate is around 27 percent. So with great accuracy we can tell which category each store falls into, failing, average or top performing.

2.5 Method : Hidden Markov Models

Introduction to the Method: This approach differs from others in its classification method. Here, we consider only "good" and "bad" states for each store: That is, whether it comes from a day with an average above or below its own median sales. We initially try three states, but the program begins to crash because of low probabilities when we include more than two levels. We note this limitation to the analysis.

In a hidden Markov model, it is assumed that the observed variable—here, sales—can come from a distribution $G_i(\cdot)$, each of which is associated with one a discrete state $i \in \{1, 2, \ldots, n\}, n \in \mathbb{N}$ (*viz.*, good and bad). But these states are unknowable. Materially, one specifies a prior belief about the initial probability of being in each state (we chose $p_1 = p_2 = 1/2$), a prior belief about the transition probabilities (we just choose $p(i, j) = 1/2$ for all $i, j \in \{1, 2\}$), and the parameters associated with being in each state. We think that the normal distribution was reasonable; the third quartile of the ratio makes a good *a priori* estimate for the good mean, and the first a good estimate for the bad—we specify the standard deviation to be half the mean in each case.

Then, one uses the Forward-Backward algorithm—which we discussd at length in class—to get a posterior estimate for the probabilities and successively maximizes the parameters of the model under those probabilities.

Results: We use the `HiddenMarkov` package in R to infer the stationary distributions for each store—that is, the average proportion of time it spends in the good state and in the bad. The results surprises us: Curiously, all the stores spend less than 35 percent of their time in the good state—this does not mean that it exceeded median sales only that often, but that it is usually in its bad state. Incidentally, we find that, when a store spends very little time in the bad state, the standard deviation in its sales is usually low (but the converse is false—having low standard deviation does not imply little time in the good state). But these probabilities have no correlation with distance from the competition, age of the competition, or promotional sales—these we checked with simple plots. If there is a predictor for that long-term fraction of time spent in the good state, it is more complex than any these data seem able to reveal.

Below is a histogram of the long-term proportion of time spent in the good state:

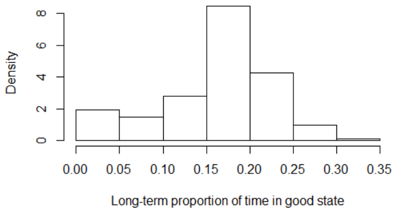


Figure 8: A histogram of the time spent in the good state

We learn: When stores do well, they tend to do very well. As a matter of fact, the sales appear at first to be log-normal distributed. But it would not make good sense to check the log of the sales—stores that are selling, for example, five times as much as most other stores are exceptional, and one cannot expect every corner drugstore to be exceptional, but only to do well inside a given range. Half the sales are between 3,700 and 7,800 Euros a day, but sales goes up to 41,000 on a given day. So, we concern ourselves more with the question of whether a store tended to exceed the median—about 5,700—and logging would collapse that right near the first and third quartiles.

We also look at the difference between the posterior difference in the good and bad means for each store. These tend to range between 2,000 and 4,000 Euros—which, not coincidentally, matches up well with the difference interquartile range of a given store's sales. So, assuming that the hidden Markov model is a valid approach here—and that there are not more states, which is note an unreasonable objection to make—we have found the unexciting conclusion that the markets fluctuate rather regularly. (We do note what might appear to be an inconsistency, though: LASSO regression suggested distance was important in predicting sales. This seems to contradict us here; but this analysis simply examines the fraction of time a store beats its own average, not whether it beats the overall average.)

3 Conclusions

The sales prediction model has certain limits: Although we find that the distance to the nearest competitor does affect sales, we have too few predictors in the model to get accurate data on the sales. A future analysis might incorporate a time-series analysis of sales; one could model sales on the basis of the last week and the same season in years past, for example. However, this seems not to agree with the general focus of this course.

The support vector machine approach does show us that classifying stores as good or bad is possible with fairly high accuracy—after all, the error rate is only about 20 percent.

The hidden Markov model suggests that a store will not necessarily have more good days than bad, despite its distance from a competitor. But "good" here does not mean report good sales records. The sales are assumed to be normally distributed, and so even a background bad day could yield an unexpectedly good ledger; for example, although a store can usually do poorly on Tuesdays, a particular Tuesday might yield spectacular sales. This is useful for a manager: Provided it is acceptable for the store to sell in its given range, the manager cannot infer that stores very close to their competition have just given up or are somehow systemically bad. They usually have good days about as often as any other store. Thus, there is no reason to infer it is burdensome to keep stores open near competitors.