DA 350

Final Exam

Eric Buehler

**Introduction:**

The purpose of this document is to report my investigation on the Product Sales Training data. First, I will present my findings about the underlying nature of the data set using unsupervised Machine Learning Methods. Then I will walk through my methodology for selecting the best regression based supervised method for predicting the number of product sales within the first month of release, and the best classification based supervised method for predicting Longevity, whether or not the product continues to be actively demanded 12 months later.

**Unsupervised Methods:**

**-PCA:**

The first unsupervised method I used was principal component analysis. Principal component analysis (PCA) creates Eigen Vectors, which are some linear combination of the data in the variables. Often, these Vectors reveal some interesting elements about the data. My first step was to pull out continuous variables. Binary variables don't make much sense for PCA. PCA looks at the distance between variables and distance can be a tricky metric for binary variables. I then ran the PCA on the continuous variables.

I found that the first principal seems to be products with a high number of similar products sold, magazines articles, ads, and advertising budget. Products high on this component are most likely highly competitive and popular products. Most variables show a positive correlation, but similar sales, and prior demand have the lowest coefficients.

The second component on the other hand seems to be positively correlated with prior demand, competitor products, advertising budget and ads, but negatively correlated with competitor ads and low awards. These are most likely cheap products that many competitors make, but our company dominates through advertising.

Looking at Figure 1, helps visualize this analysis. We see variables like number Competitor Ads, and Awards pointing strongly in the direction of the first component, capturing popular and high in demand products. On the other hand, we see prior demand and advertising budget pointing in the direction of the first component, capturing these cheaper, ad-dominated products.

Figure 1 also shows the presence of several outliers, which I threw out of the training set (see lower right corner).

Lastly, I checked to see how many principal components captured what percentage of variance in the data. I found that 7 or more components could capture 99% of the variance in the data.

**Figure 1: Principal Components of Data**



**-Clustering:**

The second unsupervised method I used was clustering. Clustering is a way of imposing labels upon data to better understand the groups within the data that might not be apparent from factorial variables. The problem with clustering is there is no way to determine the best number of clusters. We can approximate the “best” by looking at the within groups sum of squares, a way of measuring how close data points are to their clusters mean. Figure 2 shows these within cluster means, looking at the “elbow” of the graph I determined that eight seems to be an appropriate number of clusters.

**Figure 2: Within Groups Sum of Squares**



Visualizing these clusters proves to be quite messy but looking at their centers still reveals an interesting story about the data. The first cluster, holding 12 observation, seems capture a group similar to those high on the second principal component. These are cheaper products with a lot of competition in the form of competitor products and ads, but our company seems to dominate in terms of advertising budget.

The eighth cluster, holding 78 observations, seems to capture a group similar to those high on the first principal component. These are more expensive products with high advertising budgets, and a lot of competition competitor ads and competitor products. Most of the other clusters seem to fall in between these two groups.

Cluster 3, holding 168, seems a little unique as it is similar to cluster 8, but it seems that competitors are running a lot more adds. In other words, our company might be falling behind to competitors with products in this category.

**-Unsupervised Conclusions:**

There seem to be two extreme groups of products. Products that are cheap, and have many competitors, but our company dominates through advertising. And, Products that are expensive, popular, and highly competitive. Clustering also reveals an interesting group that are expensive, popular, and highly competitive, but our company might be falling behind in with advertising.

**Supervised Methods for Regression:**

My approach for testing supervised methods for regression was to try to achieve the best root mean squared error (RMSE) by trying the different algorithms and tweaking their parameters. After choosing a model, I estimated the model’s performance.

**-** **Linear Regression:**

To begin, I started with the simplest method of regression: linear regression. For training control, I set the number of cross-folds to ten, and the number of times to repeat to 3. Initially, I trained the model using all the variables in the training set, which resulted in an RMSE of 26317.

To see if I could improve the RMSE, I examined a summary of the variables significance. Similar sales, advertising budget, similar products, and innovation all were positively significant at a confidence of 99%. Competitor ads and number of magazines were negatively significant at slightly lower levels of confidence. This directionality gave me a better idea of how variables examined in principal component analysis such as advertising budget and similar products related to the variable of interest, Sales. It also provided me with an idea a subset of variables to use to potentially improve the model’s RMSE.

Unfortunately, re-training the linear model with the new subset of variables didn’t improve the RMSE.

Lastly, I tried training the model using the first eight principal components of the training data. This resulted an improved RMSE of 25915. I tried using this model to make Sales predictions. The results were strange with some large negative values. Hopefully, other models will result in a better RMSE and predictions.

**-K-Nearest Neighbors for Regression:**

K-Nearest Neighbors (KNN) is a method that predicts values for observations by looking at the mean of those values k-nearest neighbors. The number of neighbors K is a parameter that must be specified. To start, I trained models for Ks 1 through 20 using all the variables in the training set. Figure 3 shows that after a K of 9 the RMSE seemed stop improving.

**Figure 3: RMSE Over Number of Neighbors**

****

To be sure, I looked closer at Ks 10 through 16. After a few runs, a K of 13 seemed like the best choice, resulting in a RMSE of 24795.

I then tried using this value of K to train the model on a few different subsets. The subset of significant variables determined during linear regression made the RMSE. However, training the model of the first eight principal components as in the linear regression improved the RMSE to 24620. A considerable improvement from linear Regression!

**-Random Forest for Regression:**

Is a method that constructs a multitude of decision trees and takes the mean prediction of this multitude of trees. In determining the best model, I looked at the performance between gini and information trees, which are two different ways of determining the best split at each step of the decision trees. Information performed better with its lowest RMSE being 27333 compared to gini’s lowest RMSE of 27591. Visualizing one of these trees showed Advertising, Magazine, Similar Product, Prior Demand seem to be some of the more important variables. Variables for the most part that had high positive significance for Sales in Linear Regression and played prominently in the first principal component.

**Figure 4: Visualization of Information Decision Tree**



Moving forward, I examined the parameter mtry, or the number of variables randomly sampled as candidates at each split. It turned out that 1 feature resulted in the best RMSE of 26248. Lastly, I tried training the information tree with the tuning parameter on smaller subset of variables and principal components. The principal training set resulted in the lowest RMSE of 25913.

I used a tree number of 10,000 for all these models. I tried increasing the number to 20,000 to no affect.

While this RMSE is competitive with the logistic model, it is far from better than KNN’s RMSE. It is worth mentioning that the predictions for the random forest looked more reasonable than the logistic model as there were no negative predictions.

**-Neural Network for Regression:**

Neural networks are typically organized in layers, which are made up of a number of interconnected nodes which contain an activation function, often logistical. The input layer takes in data, which communicates to one or more hidden layers where the processing is done by a system of weighted connections. The network outputs a prediction based on the weighted processing.

I had some trouble with dnn and nnet packages. DNN seemed to give the same predictions for every value. On the other hand, the nnet reported a RMSE competitive with other methods around, but the predictions looked very strange. I tweaked the decay and tune grid parameters a few times, but I have limited understanding of what they are doing. Overall, I don't see the neural networks out-doing KNN, or Random Forest.

**-Conclusion of Investigation for Regression:**

The K-Nearest Neighbor model trained on the principal components of the full training data gives the lowest RMSE of 24620 and has reasonable looking predictions. In second place is the Random Forest model trained on the principal components of the full training data with a low RMSE of 25913 and reasonable looking predictions. Linear regression and Neural Networks come up with RMSEs around 25000, but the predictions look very strange even after trying multiple different subsets of data and tweaking model parameters. Overall, K-Nearest Neighbor seems to be the best model for making Sales predictions.

**-Estimating Models Performance:**

To estimate the model’s performance, I split the training set into two smaller data sets each with 788 observations (which isn’t very different from the actual test set provided of 674 observations). One to be used as a training set, the other used as a test set. I re-trained the K-Nearest Neighbor model on the new training set, and predicted values for the test set. When comparing the predicted Sales, to the actual sales, the resulting MSE was 1378. Based on this, I expect my predictions for the actual test set Sales to be roughly 1300 dollars off.

**Supervised Methods for Classification:**

My approach for testing supervised methods for regression was to try to achieve the Accuracy by trying the different algorithms and tweaking their parameters. My biggest dilemma was deciding whether to use the rose balanced data or not. I waited to look at log loss until estimating the chosen model’s performance.

**-Logistic Regression:**

As with Regression, I started with the simplest form of classification, Logistical Regression. I first trained the model on the full training set, which resulted in a 91.24% accuracy. A pretty good Accuracy for the first method.

As with Linear Regression, I also made a subset of the significant variables for the Logistical model. Many of the significant variables were similar to the linear model. However, some variables such as competitor launch, style, and product necessity seem to play a bigger role in predicting Longevity. This intuitively makes sense as we would expect competition, and variables linked to ingenuity such as style and necessity to play a bigger role in a products longevity.

Lastly, I tried training the model using the smaller subset of significant variables, the eight PCAs of the full training data and the rose balanced data. The smaller subset slightly increased accuracy to 91.62%, but the PCA variables decreased accuracy didn’t increase accuracy any.

However, I kept the rose models in consideration in case I found class imbalance to be a huge issue later on as there is an imbalance of 236 negatives to 1340 positives.

**-K-Nearest Neighbors Classification:**

As with Regression for KNN, I started by examining the best number of neighbors. Figure 5 shows that the best number of neighbors was 1 with an Accuracy of 98%.

I then tried training the model with the first 8 principal components of the full training set, the smaller subset of significant variables and the Rose Training set. None improved the Accuracy of the model.

**Figure 5: Best Number of Neighbors for Classification**



**-Random Forest for Classification:**

As with Regression, I started by determining whether Gini or Information were better measures of cut off, and information once again proved to be the better choice achieving an overall highest accuracy of 99% accuracy compared to Gini’s 98% accuracy.

Looking at the information decision in Figure 6, we see that similar products, and competitor products are important decision variables. This further demonstrates how predicting a products longevity is a different problem from sales. Factors concerning market saturation play a larger role in a product’s longevity.

**Figure 6: Visualization of Information Decision Tree for Classification**



After looking at 1 through 15 for the mtry parameter, I determined 12 to be the best number of features, resulting in an accuracy of 99%. Once again, I tried increasing the number of trees to little affect. This was the best accuracy so far. Honestly, I wouldn’t trust a model that claimed any better.

I was able to achieve an accuracy of 90% for the rose trained model with an mtry set to 1, and 20,000 trees.

**-Neural Network Classification:**

As with Regression for the dnn package, I was getting the same predictions for all node layers. However, this time, I was able to get the package working by using class balanced data. Figure 6 shows the accuracy for 1 through 5 nodes at each of the three layers dnn allows. The accuracy was around 50% for all them, overall pretty bad.

**Figure 6: Visualization of Information Decision Tree for Classification:**



Since this was a smaller data set, I decided to only focus on increasing the number nodes at the first layer. By doing this I was able to achieve an accuracy of 83% with 11 nodes at the first layer. Nowhere near the accuracy of random forest, but I was happy to get the network predicting with a decent accuracy.

**-Linear Discriminant Analysis:**

Linear Discriminant Analysis (LDA) finds a linear combination of features that separates two or more classes of objects in data, making it useful for classification problems. In LDA there aren’t as many parameters to measure so I focused on training the model using different subsets of data.

I tried the smaller subset of significant variables, the rose training set, and the full training set. The full training set worked best, which resulted in an Accuracy of 91%. A pretty good accuracy, but nowhere close to random forest.

The rose trained model achieved an accuracy of 83%.

**-Conclusion of Investigation for Classification:**

Random Forest gives the highest Accuracy at 98%. LDA and logistic regression also prove to be solid models with Accuracies of 91%, but not as good as random forest. Neural Networks once again proved to be somewhat troublesome, however after using a balanced data set I was able to improve the accuracy to 83%.

In my investigation, I tried a variety of data sets on my models. I ran into a variety of issues with each of these attempts. The rose trained models almost always performed worse with the exception of Neural Networks. However, this worse accuracy may be preferable, because class balanced trained models may give a better tradeoff between specificity, and sensitivity. In either case, Random Forest still performed the best of all the models investigated above.

**-Estimating Classification Models Performance:**

As with Regression I split the training set into two data sets. One to be used as a training set, the other used as a test set. I also made another smaller training set with rose training data as well.

First, I re-trained the random forest model on the non-balanced set. This model achieved an Accuracy of 98%, and a log loss of 9.992007e-16. At a cut of .324 the model was able to accurately classify all test data points as shown in table 1. This is great but seems problematic as I find it hard to believe the performance could be this good.

With this in mind I also looked at a balanced model. This model achieved an Accuracy of 90%, and a log loss of 3.8. At a cut of 0.579 the model had had 60 false positives, and 36 false negatives as shown in table 2.

**Table 1: Performance of Random Forest trained on Unbalanced Data**

|  |  |  |
| --- | --- | --- |
| **Column1** | **Negative** | **Positive** |
| FALSE | 121 | 0 |
| TRUE | 0 | 667 |

**Table 2: Performance of Random Forest trained on Balanced Data**

|  |  |  |
| --- | --- | --- |
| **Column1** | **Negative** | **Positive** |
| FALSE | 108 | 60 |
| TRUE | 36 | 771 |

Ultimately, I resolved to use the model trained on the non-balanced data. I doubt my predictions will be perfect as my estimate of performance asserts, but I do expect them to be very close to the true values.

**Conclusion:**

Please find attached my commented R-script, the csv of regression predictions made using a KNN model, and the csv of classification and probability predictions made using a Random Forest model.