PREDICTING CUSTOMER BEHAVIOR ON RETAIL SALES: OPTIMIZING RETURNS BASED ON MARKDOWN SUCCESS

1.0 Introduction

In this project, we will be using datasets that report the weekly sales and other vital information for a retail giant in the United States. The dataset include data on holidays and select major events (markdowns) that come up once a year for each department in the retail store.

It is the objective of the project to analyse the data set so as to enable store executives make strategic decisions which would ultimately affect the bottom line. It is often suggested that markdowns do affect sales, so this project also sets out to test it as a hypothesis and subsequently predict which departments will be affected and to what extent.

1.1 Project Focus

This project shall focus on four distinct aspects:

- Evaluating the impact of holiday and major events that happen once a year (called markdown) on sales
- Predicting future performance based on these factors
- Optimizing sales in different departments based on these factors and
- Answering Environmental Questions

1.1.0 Environmental Questions

There environmental questions to be tackled by the project include the following:

What impact does the following have on customer behavior and sales return?

- Weather (Temperature)
- Customer's income potential (CPI)
- Unemployment
- Type of store
- Size of Store If the impact is significant, how could future sales performance be optimized based on these factors?

2.0 Data

The datasets contain historical sales data for 45 stores located in different regions in the United States. Each store contains a number of departments. The company also runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the five largest of which are the Super Bowl, Easter, Mother's Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks.

One of the major challenges of modeling retail data is the need to make decisions based on limited history.

The data sets are in three tranches. The first is a 8191 x 12 features data set. The 12 columns are as follows:

Contains additional data related to the store, department, and regional activity for the given dates.

- Store - The store number
- Date - The date for the day of the week
- Temperature - The average temperature in the region
- Fuel_Price - The cost of fuel in the region
- MarkDown1-5 - These are anonymized data related to promotional markdowns. Take note that MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- CPI - The consumer price index
- Unemployment - The unemployment rate
- IsHoliday - This is a boolean variable that determines whether the week is a special holiday week The next data set tranche is a 46 x 3 Stores data set. This data set contain anonymized information about the 45 stores, indicating the type and size of store.

The last data set is the 421571 x 5 Sales data set. This data set contains historical sales data, which covers weekly sales from 2010-02-05 to 2012-11-01. Within this data set we will find the following columns:

- Store - The store number
- Dept - The department number
- Date - The date for the day of the week
- Weekly_Sales - The weekly sales for the given department in the given store
- IsHoliday - This is a boolean variable that determines whether the week is a special holiday week

3.0 Study Strategy

3.1.1 Data Wrangling and Conversion

In this section, we cleaned up the data and perform data conversion so that we will have master data set that is suitable for the analysis.

The first step in creating a master data set for analysis was to merge the features data set and the stores data set. This created a data set that contains the store type and size as features. (See df_master1 (Line 5) - Codebase 1)

The next step was to prepare sales dataset for incorporation to the master data set. The sales data set contains 421571 rows which comprises weekly sales data for 98 departments per store number and for each week from January 2010. In order to make the data set consistent with the other data sets so that they could easily be merged to create a master data set, the departments need to be moved to the columns. The first step to achieve this is by ensuring that the Date column is converted to datetime (See df_master1 (Line 6) - Codebase 1). Then we used the unstack method to create a sales table with the departments as columns rather than rows. This creates a 6435 rows × 81 columns table. Finally, we created the Master Data Set for Analysis by merging df_sales2 and df_master1. This master data set shall be called df_master (See df_master1 (Line 13) - Codebase 1).

3.1.3 Handling Missing Values

The missing values in the first few rows will be filled using the backfill method while the remainder will be filled using the interpolate method. Subsequently, interpolate method was used to fill the remaining missing values

4.0 Data Exploration

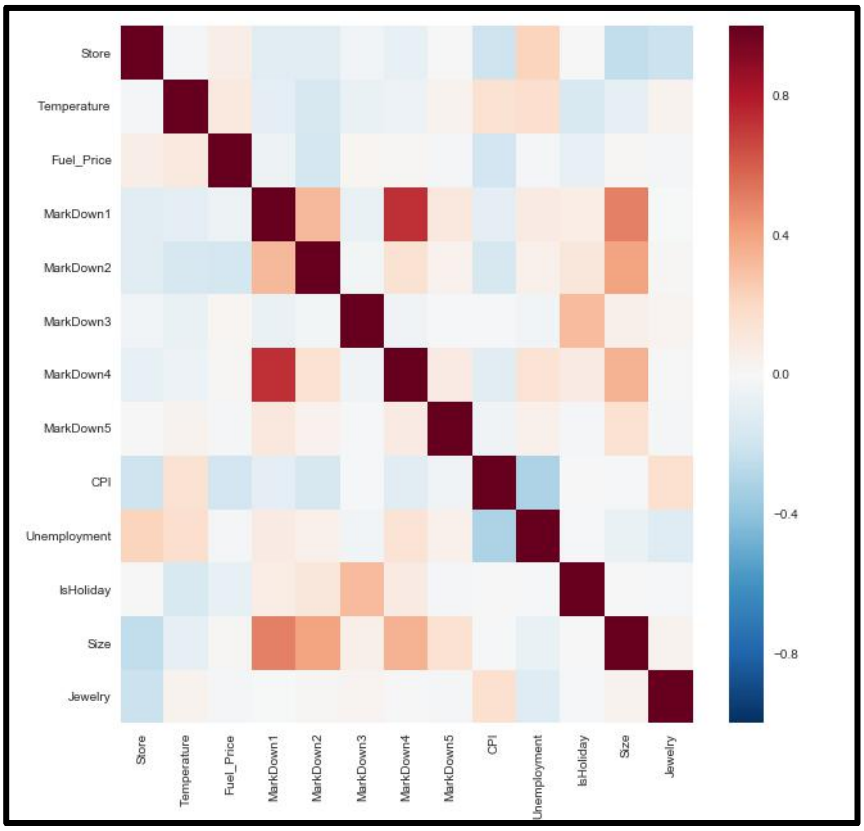
In order to gain insight into the data for this project, we will here carry out data exploration

4.1.1 Statistical Overview

The dataset has:

- About 8190 sales record and 95 features
- About 7% of the total period of sales are holidays
- Mean promotional sales are as follows: Markdown1 = 8887.617797, Markdown2 = 6107.224317, Markdown3 = 928.785220, Markdown4 = 3130.176556, Markdown5 = 4544.031686
- Promotions are generally more successful (more sales are recorded) during holidays than during non-holidays

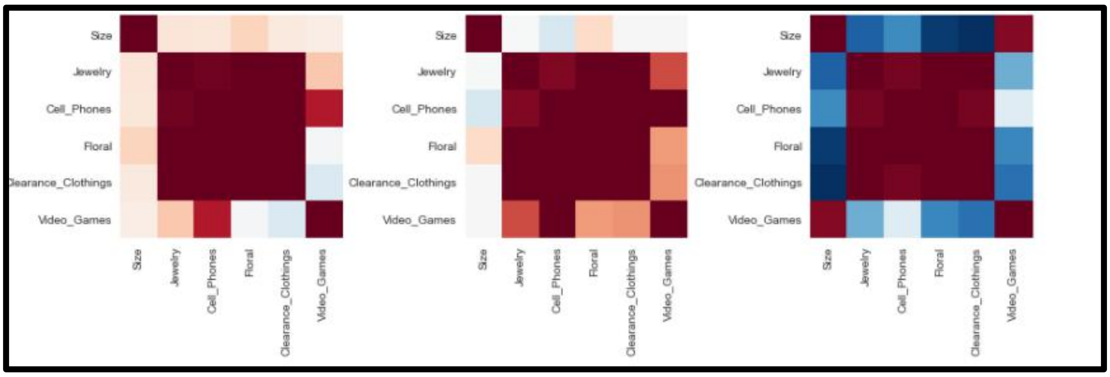
Below is the heatmap of few of the features and the markdowns:



From the heatmap, there is a positive(+) correlation between size and most of the markdowns. Which could mean that the bigger the store, there is more likelihood of success in the promotional events. There is also positive correlation between the promotional events which could suggest that each promotional event have impact on the other.

For the negative(-) relationships, CPI and Unemployment are highly correlated. This suggests that unemployment affects the rate at which people spend on products in the retail stores.

Generating the correlation matrix grouped by type shows that the behavior of customers differ from one store type to the other depending on the size of the store. For store type A, the size of the store is moderately correlated to floral sales and has relatively weak positive correlation to sales of other gift items. This is an indication of this store type being a wholesale store where there are some sale activities but the activities are not as pronounced as in Store Type C.



In Store Type B, size has a moderate negative correlation with cell phone sales, moderate positive correlation with floral sales and very weak or no correlation with sales of other items. This indicates that this store type is likely a neighborhood store that sells only groceries and few household items. The lack of sales of clothings and jewelry more so gives credence to this assertion.

In Store Type C, there is an indication of strong negative correlation between size and sales. The high activity indicates that this is a superstore type and the negative correlation with size indicates that customers are likely to purchase things in more compact superstores than in the bigger superstores.

The behavior described above also gives us early indication of clustering with respect to type which will be useful in prediction and classification.

5.0 Hypothesis Testing

9 hypotheses tested

- Store size has impact on Markdown Success
- Missing values significantly affect result of tests
- The difference between sales recorded on holidays and those recorded on non-holiday weeks is statistically significant
- The difference between markdown sales on holidays and those recorded on non-holiday weeks is statistically significant
- The difference in Markdowns for each Store Type is statistically significant

- The difference in Sales of gift items for each store Type is statistically significant
- The difference in Store Size for each Store Type is statistically significant
- Store size has impact on Sales of Gift Items
- Store size has impact on MarkDowns

Two major results include:

1. Create composite function for Markdown success and test the hypothesis that there is significant difference in the means of size of the stores who had high Markdown success and those who had low markdown success. The null hypothesis is that there is no difference in store size between stores which had high Markdown success and those with low markdown success and the Alternate Hypothesis: (HA: pSS != pMS) is that there is a difference in store size between stores who had high Markdown success and those who had low markdown success. The result shows that we reject the null hypothesis
2. Whether there is significant difference in the means of the filled missing values and those of the general population.
   a) Null Hypothesis: (H0: pSS = pMS) There is no difference in the result by filling the missing values with numbers using the back fill/ interpolation methods.
   b) Alternate Hypothesis: (HA: pSS != pMS) there is a difference in the result by filling the missing values with numbers using the back fill / Interpolation methods

   In this case, the null hypothesis is accepted.

6.0 Unsupervised Learning - Anomaly Detection

The following algorithm has been implemented in this Unsupervised Anomaly Detection on the retail stores dataset

- Cluster based anomaly detection (K-mean)
- Re-partition of data into categories then Gaussian/Elliptic Envelope on each categories separately
- Markov Chain
- Isolation Forest
- One class SVM
- RNN (comparison between prediction and reality)

6.1.1 Test of Multicollinearity - Variance Inflation Factor

In this section, we will be exploring the degree of correlation between the predictor variables in our retail sales dataset. That is, we will be testing the models collinearity. For two variables that are collinear, we expect that they should contain similar information about the variance within the given dataset.

To detect the collinearity, we will create a correlation matrix and find variables with large absolute values.

6.1.1.1 Multicollinearity

Detection of multicollinearity is follows a more complicated procedure. This is because it emerges when three or more variables which are highly correlated are included in the model. It can also emerge when isolated pairs of variables are not collinear.

In order to test multicollinearity we will use Variance Inflation Factor (VIF).

6.1.1.2 Variance Inflation Factor

The variance inflation factor (VIF) is a measure of colinearity among predictor variables within a multiple regression. It is calculated by taking the ration of the variance of all a given model's beta if it were fit alone.

6.1.1.3 Steps for Implementing VIF

- Run a multiple regression
- Calculate the VIF Factors
- Inspect the factors for each predictor variable: If the VIF is between 5-10, multicollinearity is likely present and we would consider dropping the variable.

For details of the result see (VIF)

Out of 92 Features, 29 ~ 30% have VIF < 5.
Group features with VIF > 5 into 8 Categories to create new target features

NEW TARGET FEATURES

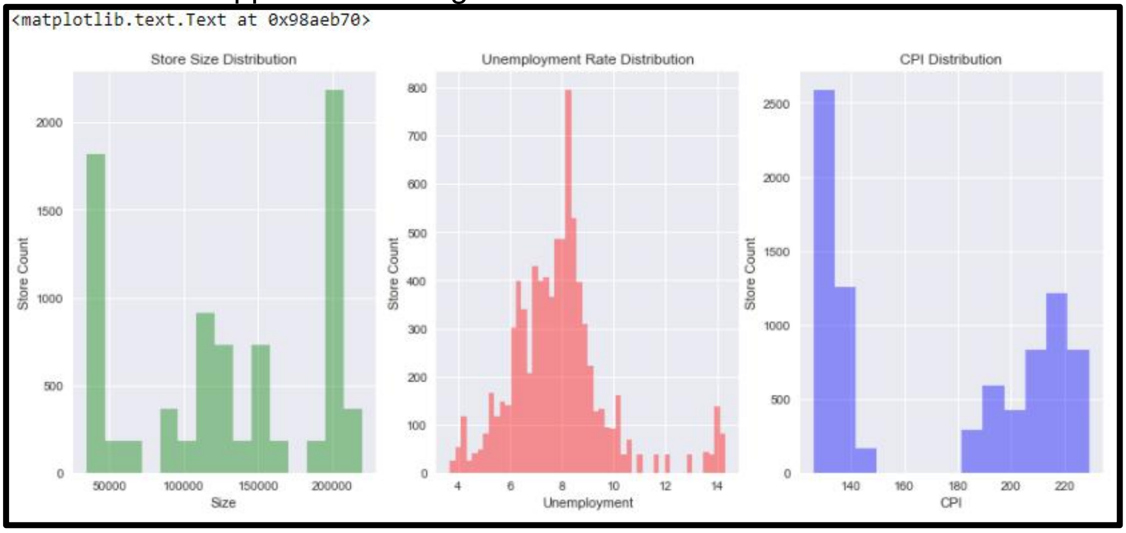
- Home-items
- Electronic-items
- Health-items
- Kids-items
- Office-items
- Auto-items
- Wears-items
- Food-items

7.0 Distribution Plots & Clustering

7.1.1 Distribution Plots (Size - Unemployment - CPI)

- Store Size - There are high concentration on large size and small size stores. This is a bimodal distrubtion of Stores for low sizes (less than 100000) and large sizes (more than 150000)
- Unemployment - The rate of unemployment is normally distributed across the 45 store locations.
- CPI - There is a huge spike in the number of stores located in regions of low consumer price index. This is another bimodal distribution of stores with lower and higher CPI (less than 150 & more than 180)
- The Size and CPI graphs both share a similar distribution.
- Small Size are located in places of lower CPI and vice versa. This impact is more pronounced in Store Type C. If you look back at the

correlation matrix for store type C, the high correlation between size and CPI does support this finding.



```
<matplotlib.text.Text at 0x98aeb70>
```

Questions:

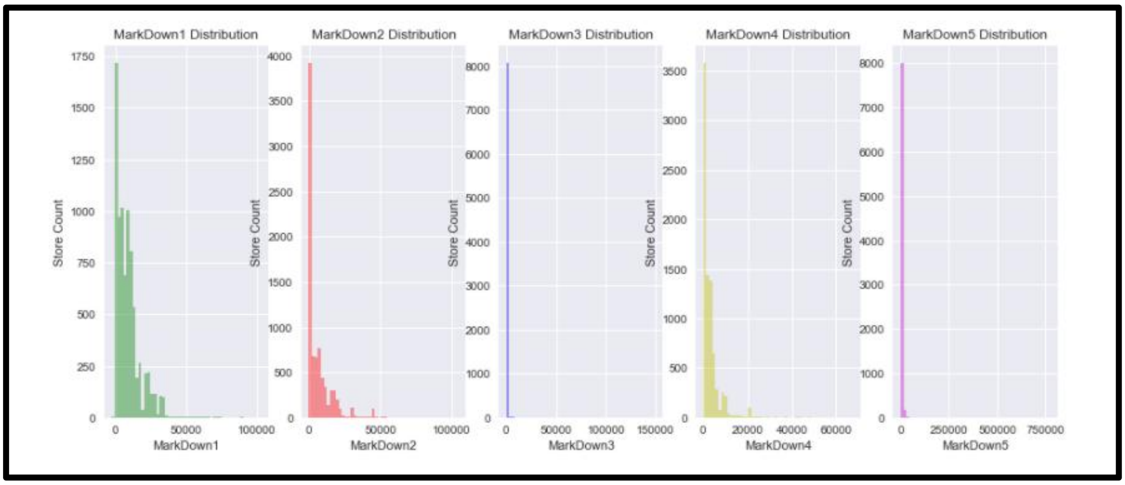
- Is there a reason for the high spike in concentration of stores in low CPI regions?
- Could the stores be grouped in a way with these features?
- Is there a correlation between store sizes and CPI?

7.1.2 Distribution Plots (MarkDowns)

- The distribution of the Markdowns follows negative exponential distribution. This suggests that more stores return the low end of MarkDown. This is more pronounced in MarkDown3 and MarkDown5.
- MarkDown1, MarkDown2 and MarkDown4 have similar distribution. The intercorrelation between MarkDown1 and MarkDown2, and MarkDown1 and MarkDown4 tends to support this finding.

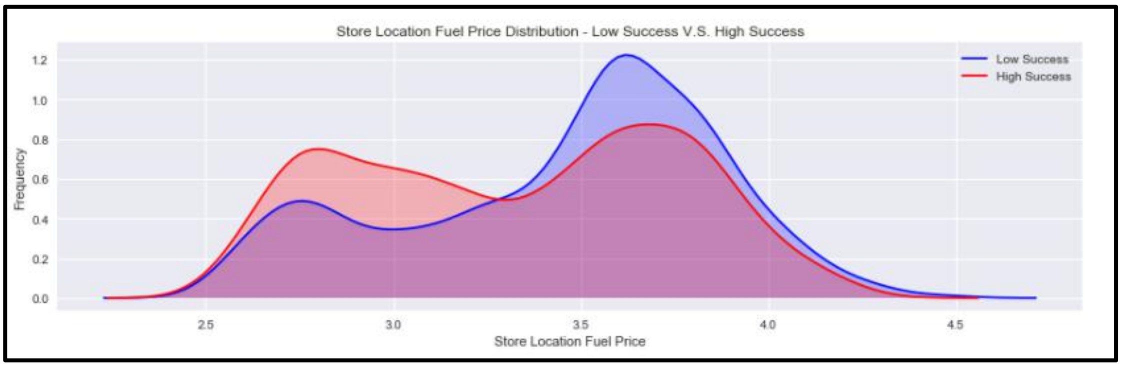
Questions

- Is there a relationship between MarkDown1, MarkDown2 and MarkDown4?
- Does the size of the stores have impact on the MarkDowns?
- How does the unemployment rate affect the Markdowns?
- Does the CPI have any impact on the Markdowns?
- Does the type of Store impact on the Markdowns?
- What is the impact of the holidays on the MarkDown Sales



7.1.3 Fuel Price V.S. MarkDown

- There is a biomodal distribution for both level of MarkDown successes.
- When the fuel price is low, promotional events tend to be more successful
- When fuel price is high, the reverse is the case, that is less sales are achieved through the markdowns

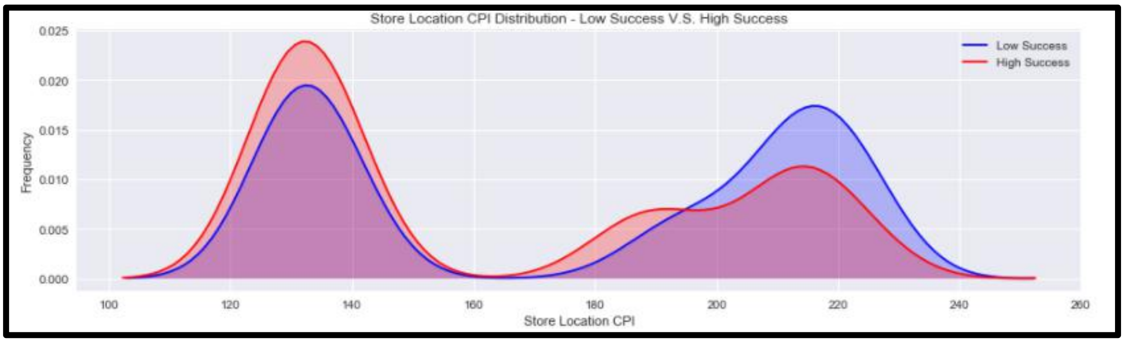


7.1.4 Temperature V.S. MarkDown

- There is a normal distribution for both level of MarkDown successes.
- The sweet spot temperature that sales are made is between 60F and 70
- The difference in level of success achieved does not seem to be affected by temperature

7.1.5 CPI vs MarkDowns

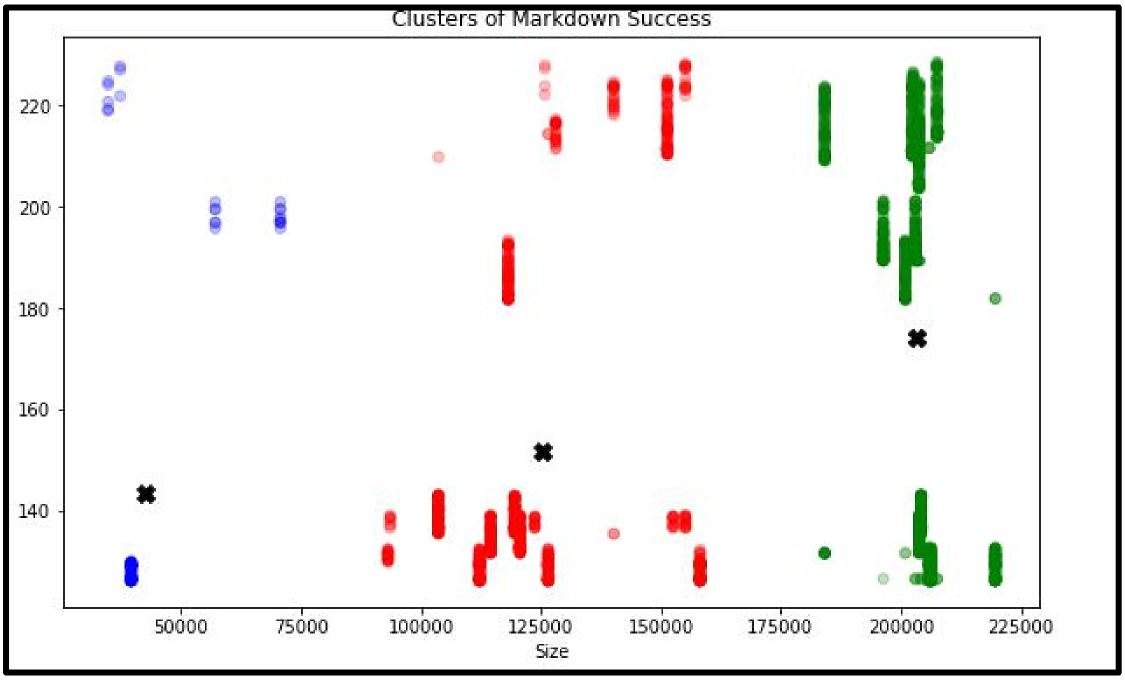
- There is a bimodal distribution for both level of MarkDown successes with respect to consumer price index.
- Stores that are located in high CPI areas tend to achieve high success in promotional sales while in areas where the CPI is high, promotional sales are relatively less successful.



7.1.6 Clustering

- MarkDown Success Used as Target Variable
- Used k-means clustering

Three Clusters distinguishable based on Store Size



8.0 Classification- DecisionTreeClassifier

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Once we completed modeling the Decision Tree classifier, we will use the trained model to predict whether the more sales are carried out during holidays or during ordinary days.

8.0.1 Data Slicing

Data slicing is a step to split data into train and test set. Training data set can be used specifically for our model building. We separate the Test dataset from the Training data set.

8.0.2 Measuring the Model's Performance

The above snippet divides data into feature set & target set. The “X ” set consists of predictor variables. It consists of data from 4th column to 95th column. The “Y” set consists of the outcome variable. It consists of data in the 1st column. We are using “.values” of numpy converting our dataframes into numpy arrays.

Let’s split our data into training and test set. We will use sklearn’s train_test_split() method to split the data into:

- a training set
- a test set

We did fit/train the classifier on the training set, then we will make predictions on the test set. And then we will compare the prediction with the known labels by computing the accuracy of the model.

The above snippet will split data into training and test set. X_train, y_train are training data & X_test, y_test belongs to the test dataset.

The parameter test_size is given value 0.3; it means test sets will be 30% of whole dataset & training dataset’s size will be 70% of the entire dataset. random_state variable is a pseudo-random number generator state used for random sampling.

8.0.3 Decision Tree Training

Now we fit Decision tree algorithm on training data, predicting labels for validation dataset and printing the accuracy of the model using various parameters.

DecisionTreeClassifier():

This is the classifier function for DecisionTree. It is the main function for implementing the algorithms. Some important parameters are:

criterion:

It defines the function to measure the quality of a split. Sklearn supports “gini” criteria for Gini Index & “entropy” for Information Gain. By default, it takes “gini” value.

splitter:

It defines the strategy to choose the split at each node. Supports “best” value to choose the best split & “random” to choose the best random split. By default, it takes “best” value.

max_features:

It defines the no. of features to consider when looking for the best split. We can input integer, float, string & None value.

- If an integer is inputted then it considers that value as max features at each split.
- If float value is taken then it shows the percentage of features at each split.
- If “auto” or “sqrt” is taken then max_features=sqrt(n_features).
- If “log2” is taken then max_features= log2(n_features).
- If None, then max_features=n_features. By default, it takes “None” value.

max_depth:
The max_depth parameter denotes maximum depth of the tree. It can take any integer value or None. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. By default, it takes “None” value.

min_samples_split:
This tells above the minimum no. of samples reqd. to split an internal node. If an integer value is taken then consider min_samples_split as the minimum no. If float, then it shows percentage. By default, it takes “2” value.

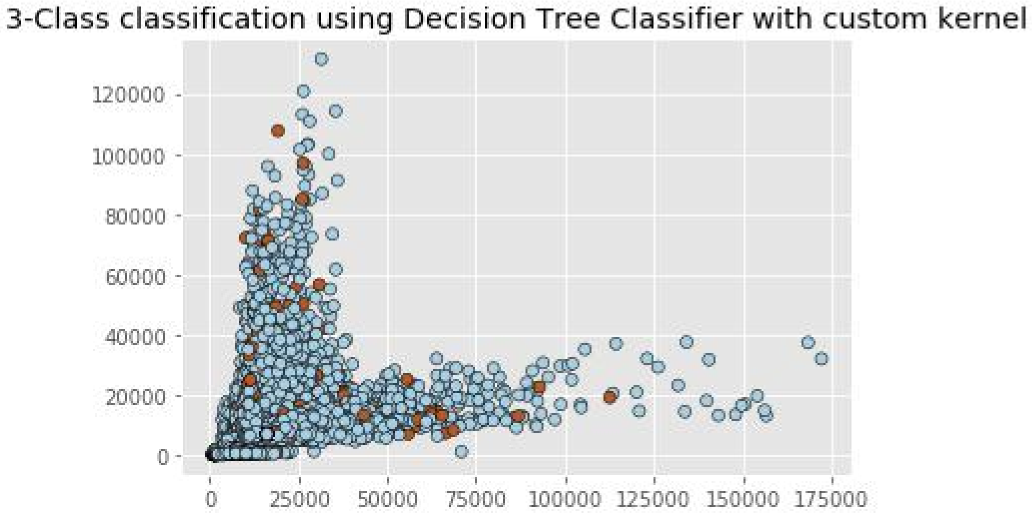
min_samples_leaf:
The minimum number of samples required to be at a leaf node. If an integer value is taken then consider min_samples_leaf as the minimum no. If float, then it shows percentage. By default, it takes “1” value.

max_leaf_nodes:
It defines the maximum number of possible leaf nodes. If None then it takes an unlimited number of leaf nodes. By default, it takes “None” value.

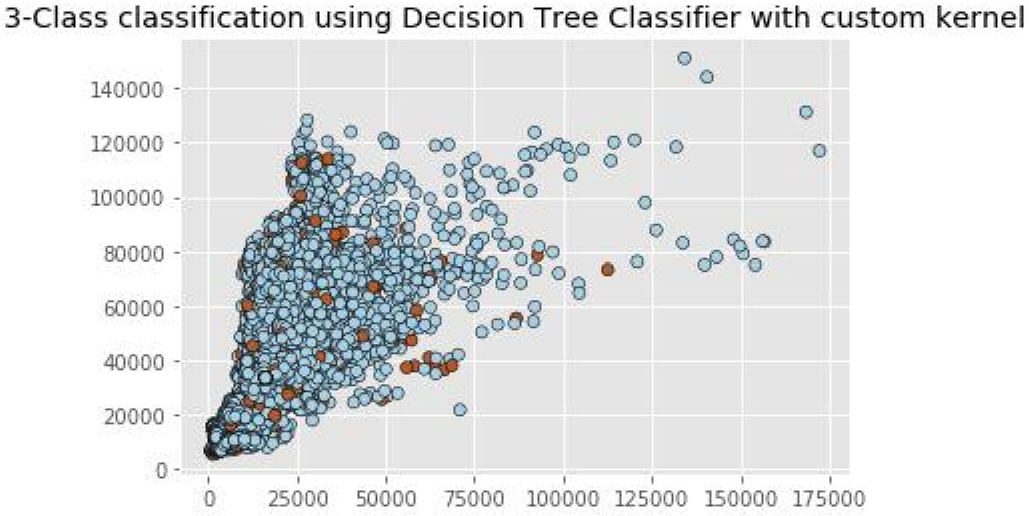
min_impurity_split:
It defines the threshold for early stopping tree growth. A node will split if its impurity is above the threshold otherwise it is a leaf.

We will now build classifiers using criterion as gini index and information gain. We need to fit our classifier using fit(). We will plot our decision tree classifier’s visualization too.

Decision Boundary Plot¶



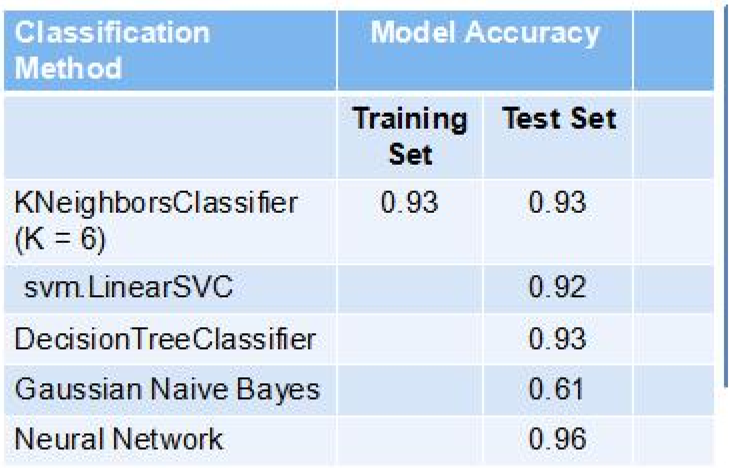
We modeled the decision tree for the retail store data set. In the process, we split the data into train and test dataset and we modeled decision tree classifier using the information gain, and gini index split criteria. In the end, we calculate the accuracy of these two decision tree models.



8.0.3 Model Accuracy

- Accuracy for Decision Tree classifier with criterion as gini index¶is **92.6332926333**
- Accuracy for Decision Tree classifier with criterion as information gain is **92.673992674**

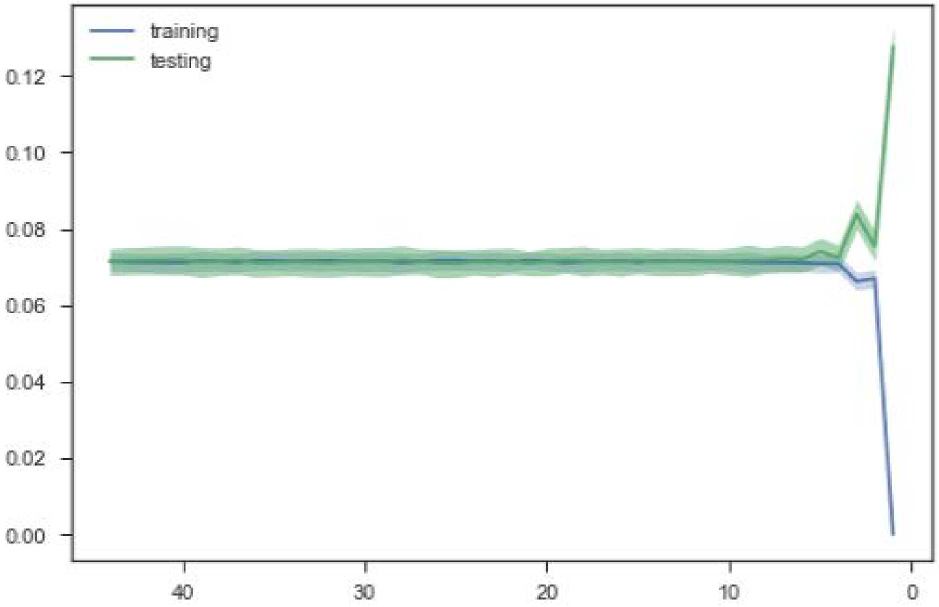
8.0.5 Classification



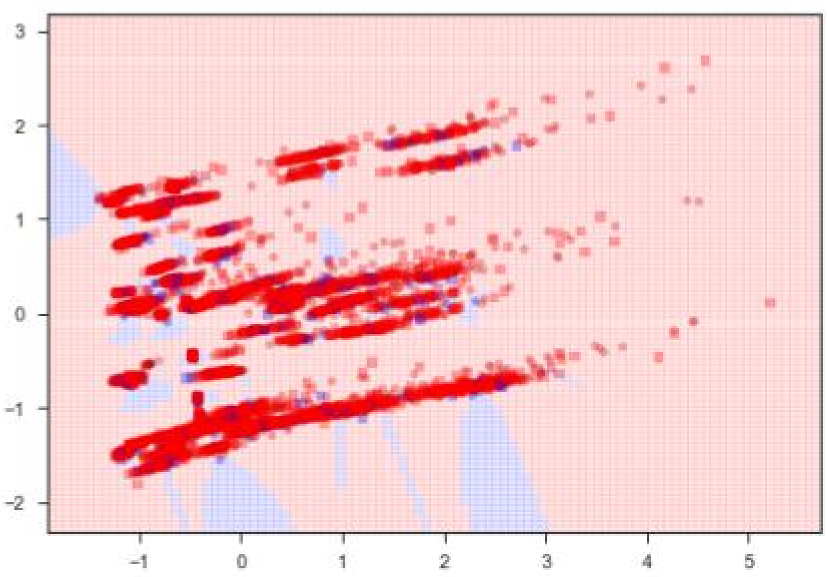
Classification Method	Model Accuracy		
	Training Set	Test Set	
KNeighborsClassifier (K = 6)	0.93	0.93	
svm.LinearSVC		0.92	
DecisionTreeClassifier		0.93	
Gaussian Naive Bayes		0.61	
Neural Network		0.96	

9.0 Dimensionality

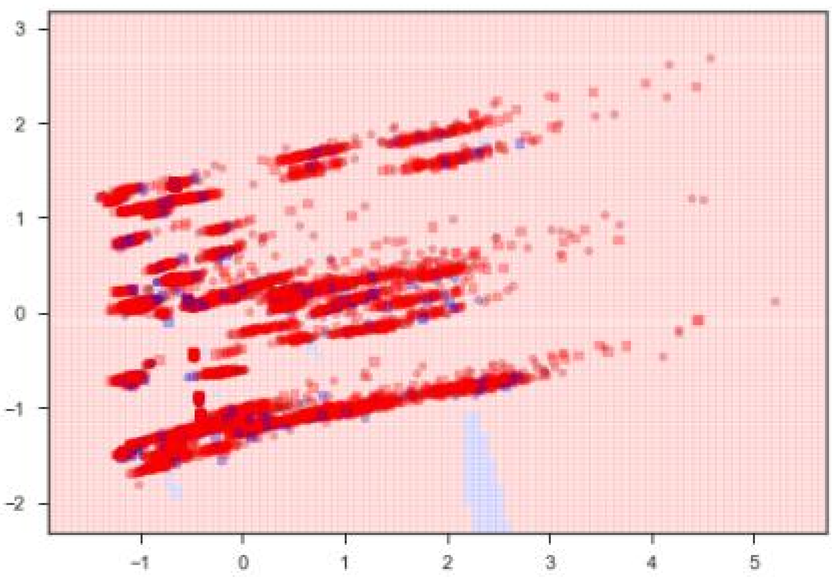
Here we used KNeighborsClassifier. The result shows that a group of three features can explain much of the variations in the dataset.



For k = 1
Accuracy on training data: 1.00
Accuracy on test data: 0.88



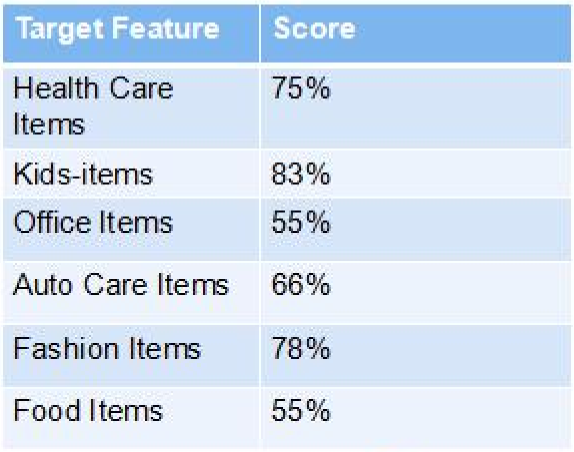
For k = 3
Accuracy on training data: 0.93
Accuracy on test data: 0.93



10.0 Regression

Here we studied effect of MarkDown on sales of:

- Health-items
- Kids-Items
- Office-Items
- Transport (Auto)-Items
- Fashion (Wears)-Items and
- Food-Items
- These Target Features generated using the VIF results

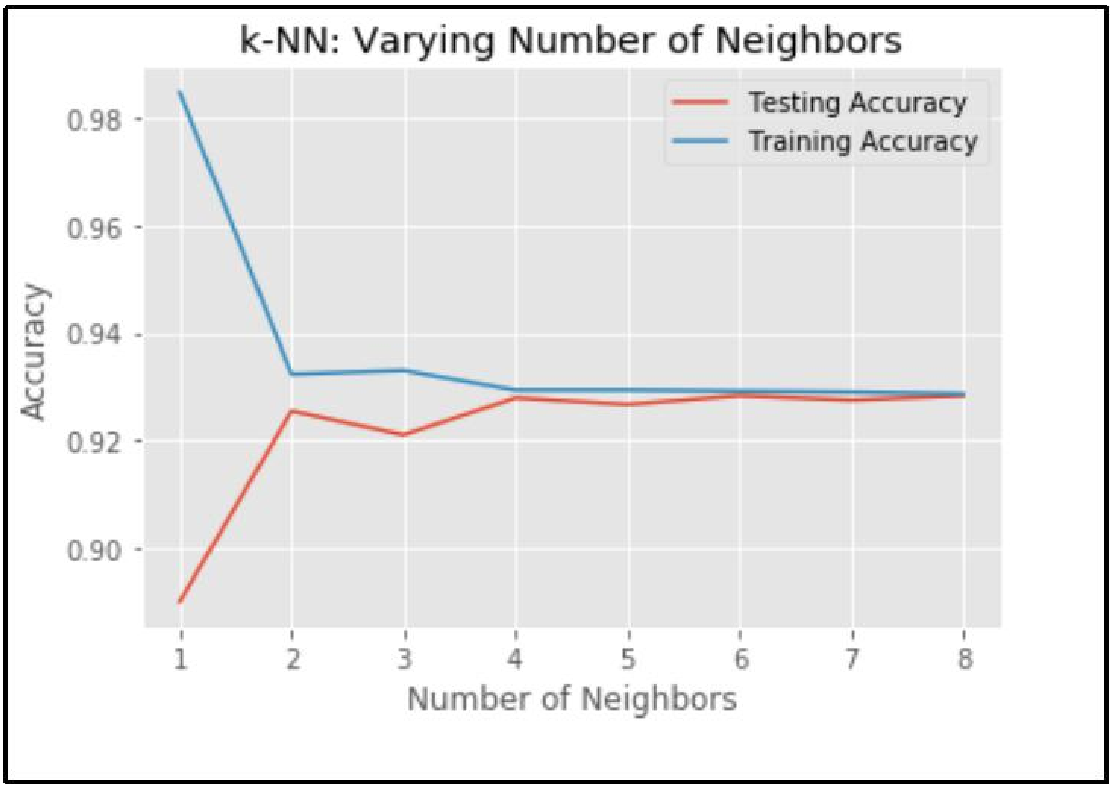


Target Feature	Score
Health Care Items	75%
Kids-items	83%
Office Items	55%
Auto Care Items	66%
Fashion Items	78%
Food Items	55%

As Expected, 83% and 78% of the variation in the sales for Kids' and Fashion items are accounted for by the markdowns

10.1.1 Overfitting/ Underfitting

Optimal K is achieved @ K = 6;



11.0 Conclusion/ Recommendation

- The performance of the promos is dependent on the size of the stores - The bigger the store, the more successful the promo (Markdown).
- Success of the sales are more pronounced during routine holidays and weekends more than ordinary week days.
- While promo sales tends to behave independently from each other, sales during MarkDown 1 and MarkDown 4 have strong positive correlation

- The MarkDowns (promos) have more effect on sales of kids items and fashion items (for teens and adults) than other items.
- It is recommended that retailers pay more attention on these items (kids and fashion) during promos