Supermarket Sales Data Exploration

Hosted on Jupyter Notebooks

1st Collin Sieffert
Electrical and Computer Engineering Department
University of Florida
Gainesville, United States
csieffert@ufl.edu

***Abstract*—This document contains the dataset breakdown, exploration, and workup of models designed to draw insightful conclusions from hidden patterns. The models in use will predict and classify a number of hypotheses about potential interactions within the dataset and how those results can be used to improve the sales at the supermarket branches.**

***Index Terms*—Logistic Regression, Gradient Boosting, Linear Regression, Lasso Regression**

I. Introduction

This document will cover data exploration, hyper parameter tuning and model evaluation.

II. Data Exploration

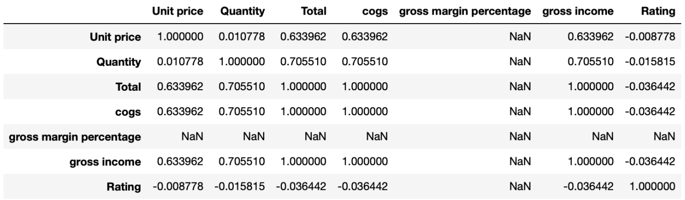
A. Attribute Descriptions

Here we'll begin with a quick breakdown of each feature collected for the initial dataset

- Invoice id: Computer generated sales slip invoice identification number.
- Branch: Branch of supercenter (3 branches are available identified by A, B and C).
- City: Location of supercenters.
- Customer type: Type of customers, recorded by Member for customers using member card and Normal for without member card.
- Gender: Gender type of customer.
- Product line: General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel.
- Unit price: Price of each product in US dollars.
- Quantity: Number of products purchased by customer.
- Total: Total price including tax.
- Date: Date of purchase (record available from January 2019 to March 2019).
- Time: Purchase time (10am to 9pm).
- Payment: Payment used by customer for purchase (3 methods are available - Cash, Credit card and Ewallet).
- COGS: Cost of goods sold.
- Gross margin percentage: Gross margin percentage.
- Gross income: supercenter gross income in US dollars.
- Rating: Customer stratification rating on their overall shopping experience (on a scale of 1 to 10).

Funded by Midnight Oil Co.

B. Preprocessing Selections



	Unit price	Quantity	Total	cogs	gross margin percentage	gross income	Rating
Unit price	1.000000	0.010778	0.633962	0.633962	NaN	0.633962	-0.008778
Quantity	0.010778	1.000000	0.705510	0.705510	NaN	0.705510	-0.015815
Total	0.633962	0.705510	1.000000	1.000000	NaN	1.000000	-0.036442
cogs	0.633962	0.705510	1.000000	1.000000	NaN	1.000000	-0.036442
gross margin percentage	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gross income	0.633962	0.705510	1.000000	1.000000	NaN	1.000000	-0.036442
Rating	-0.008778	-0.015815	-0.036442	-0.036442	NaN	-0.036442	1.000000

Fig. 1. Correlation matrix for the numerical features.

After looking at both the numerical and categorical features in the data set, there are a number of drops we can make to simplify our training. First the branch and city features are redundant as there's only 1 branch in each city. Thus the city feature can be dropped. The Invoice ID is a unique generated ID for each transaction, this carries no information with it and can be dropped without second thought. The gross margin percentage is a constant value for every entry and just like the Invoice ID, it can be dropped without hesitation. In order to make informed decisions about the sales at theses stores, the Total and Cost of Goods Sold (cogs) features should also be dropped, otherwise the models would learn that subtracting the two values results in the gross income. While this is accurate, it doesn't give any useful insight into the shopper, just that a computer can do simple math. On top of all of that, the Rating column has little to no correlation with any of the other numerical values. For this reason I dropped this feature, however it may be useful in other experiments should we want to pursue shopping experience questions.

Now that all the useless features have been removed, we can go about transforming the data set we have to better leverage machine learning techniques. Right off the bat we'll simplify the date feature to just be the day of the week in which the transaction occurred. Following that we can simplify our time feature down to four categories (Morning, Afternoon, Evening, Night) based on the hour of the day in which the transaction occurred. Now that every feature is either a numerical feature or a simple categorical feature we can go about encoding the features into simple numerical values. All multi-categorical features (3 or more categories) will be encoded with a one hot encoder. The binary categories will be transformed with an integer encoder. Finally the numerical features will all be scaled with the StandardScaler. The two features we want to predict, gross income and unit cost, will each be left alone in their respective data sets. As they could be used to predict each other we encode one or the other before training.

III. Predicting Gross Income

In order to predict the gross income as a function of the encoded features we test a multiple linear regression model both with and without the Lasso regularization. These two models placed vastly different values on the predictiveness/impact of the encoded features. In the non-regularized linear regression, the two most impactful features for predicting the gross income were the Product Line and the Branch. Surprisingly, the unit price and quantity features had weight measurements of 0 associated with them. Time of day had a negative weight of approximately the same magnitude as the positive impact payment type has. The day of the week has the smallest non-zero weight of any feature, with a minute positive impact on the prediction. In short, gross income is most affected by Branch and Product line, somewhat impacted by Time of Day and Payment Type, barely impacted by Day of Week, and not impacted by any of customer type, quantity, gender, or unit price. It would appear that when you need to buy something you go and buy it regardless of who you are, or what time of whatever day it is. As such, the gross income is predicted by what product line you get and where you buy it from. However, in the linear regression with the lasso regularization, the two most untactful features for predicting the gross income were the Quantity and the Unit Price. It would appear that including the Lasso penalty greatly changes the conclusions drawn from this dataset. Every other feature had a weight of 0 with lasso regularization. In this case it would appear that the gross income is predicted by how much you bought and how much each of those items cost. This is a much more intuitive result than the linear regression one before. For the second model, the optimal alpha hyper parameter value was 0.1.

IV. Predicting Unit Cost

Unsurprisingly, this linear regression model predicts in the exact same way as the linear regression model for the gross income. Hence the equivalent analysis.

The two most impactful features for predicting the unit cost were the Product Line and the Branch. Again, the unit price and quantity features had weight measurements of 0 associated with them. Time of day had a negative weight of approximately the same magnitude as the positive impact payment type has. The day of the week has the smallest non-zero weight of any feature, with a minute positive impact on the prediction. In short, unit cost is most affected by Branch and Product line, somewhat impacted by Time of Day and Payment Type, barely impacted by Day of Week, and not impacted by any of customer type, quantity, gender, or unit price. It would appear that when you need to buy something you go and buy it regardless of who you are, or what time of whatever day it is. As such, the unit cost is predicted by what product line you get and where you buy it from. And so, for the lasso regression, the two most impactful features for predicting the unit cost were the Quantity and the gross income. It would appear that including the Lasso penalty grealty changes the conclusions drawn from this dataset. Every other feature had a weight of 0 with lasso regularization. In this case it would appear that the unit cost is predicted by how much you bought and how much each of those items cost. This is a much more intuitive result than the linear regression one before. For this lasso regression, once again the optimal alpha hyper parameter value was 0.1.

V. Exploring Polynomial Features

A. Classifying Gender

We will now train a logistic regression classifier to classify gender. The dataset will be limited to only rows from Branch C, and only the following features:

- gender
- product line
- payment
- gross income

Looking into the Attribute and Coefficient table shown in the accompanying test notebook, we see that the strongest negative and positive relations to predicting gender lay in the combination of Product Line and another feature, whether it be payment method or gross income. Plotting the parameter values for all attributes and the 2nd order interactions, for just Gender=male customers, yields roughly the same information as plotting the original attribute table above used to study the relationship between all the attributes. We still find that no single attribute is useful to much degree, but the combination of a Product line with a Payment method, or in 1 case gross income, yields useful information about the gender of the shopper.

B. Classifying Customer Type

We will now train a logistic regression classifier to classify customer type. The dataset will be limited to only rows from Branch C, and only the following features:

- customer type
- gender
- day
- timeslot

Looking into the Attribute and Coefficient table in the accompanying test notebook, we see that the strongest negative and positive relations to predicting gender lay in specifically which Day of Week as well as the combination of Day of Week and Time of Day. Monday Evening is very strongly negative, and Monday Night is strongly positive. Add in the Afternoon, Sunday, and Friday to cover the strongest positive and negative relations. Nothing comes close to Monday evening in terms of coefficient magnitude for predicting customer type. We find the Day of Week combined with the Time of Day to be the strongest interaction.Plotting the parameter values for all attributes and the 2nd order interactions, for just Customer type=normal customers, yields roughly the same information as plotting the original attribute table above used to study the relationship between all the attributes. We still find the Day of Week combined with the Time of Day to be the strongest interaction.

VI. Predicting the day of purchase

In experimenting with predicting the day of purchase, we test both a simple Logistic Regression classifier as well as a Gradient Boosting classifier. For the Logistic Regression Classifier, the optimal hyper parameters turned out to be the default parameters with a penalty of None. However given the Accuracy of 100% on both train and test, I fear that it may have overfit the data. That or the predictions are just so simple that the model can easily get it right every time. For the Gradient Boosting Classifier, the optimal hyper parameters turned out to be the default parameters with a learning rate of 0.001 and a subsample rate of 0.5. With once again an accuracy of 100% there is a potential that the data set has been overfit.

VII. Conclusion

All in all this has been an enjoyable eye opening experiment into a multi-model data exploration and experimentation process. At the end of the day I would suggest boosting sales efforts for Monday evenings as that appeared to be the strongest day for moving product through Branch C. In order to draw more detailed conclusions I would want to study the ins and outs of that particular store and the surrounding area more. Please view the accompanying notebooks for quantitative model evaluation metrics.