Supermarket Predictions via Python Machine Learning Models

Andrew Perez-Ledo
University of Florida
Gainesville, United States
andrewperezledo@gmail.com

***Abstract*— The employment of machine learning to aid in various aspects of society has become more and more common. In this report, we will cover the implementation of the scikit-learn Python library to predict various values that have potential for commercial supermarket use based upon simulated supermarket data.**

Keywords—Pipeline, Cross validation, Regression, Model

I. The Data

We are provided with a comma separated value (CSV) dataset which contains simulated supermarket data. Some notable values consist of total, date, and unit price. Reference figure 1.

II. The Goal

The end goal we are trying to achieve is the correct implementation of Python code to train and assess three models for predicting purchases gross income for the supermarket, unit price of goods sold, and day of purchase within the week. This goal provides additional smaller milestones which must be met such as the visualization of given data to uncover useful information for our models, using this newfound information and carrying out hyperparameter tuning for model creation, and deploying final effective learning models.

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	548.9715	1/5/2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	80.2200	3/8/2019	10:29	Cash	76.40	4.761905	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	340.5255	3/3/2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	42.3675	1/29/2019	13:46	Ewallet	40.35	4.761905	2.0175	6.2
996	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	1022.4900	3/2/2019	17:16	Ewallet	973.80	4.761905	48.6900	4.4
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	33.4320	2/9/2019	13:22	Cash	31.84	4.761905	1.5920	7.7
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	69.1110	2/22/2019	15:33	Cash	65.82	4.761905	3.2910	4.1
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	649.2990	2/18/2019	13:28	Cash	618.38	4.761905	30.9190	6.6

Figure 1

III. Preparation for Model Creation

Before we construct our models, we must first review our data, identify any correlations and clean said data to ensure that it can be properly utilized. We first checked for any missing data which resulted in no outcomes. Then, we visualize our data to better understand the overall trends for the entire dataset (see Figure 2). Next, we identify any data that can be discarded due to their irrelevancy such as "Invoice ID" and "gross margin percentage." Now, numerical correlations can be made using built-in scikit-learn functions and visualized using the matplotlib library (see Figure 3). Note that the indexes for Figure 3 are as follows: 0-Unit Price, 1-Quantity, 2-Total, 3-Rating, 4-Gross Income, 5- Cost of Goods Sold (COGS).

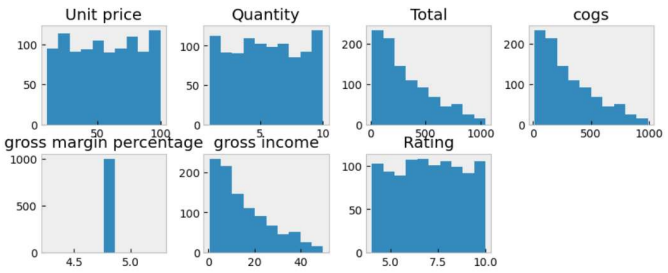

Figure 2

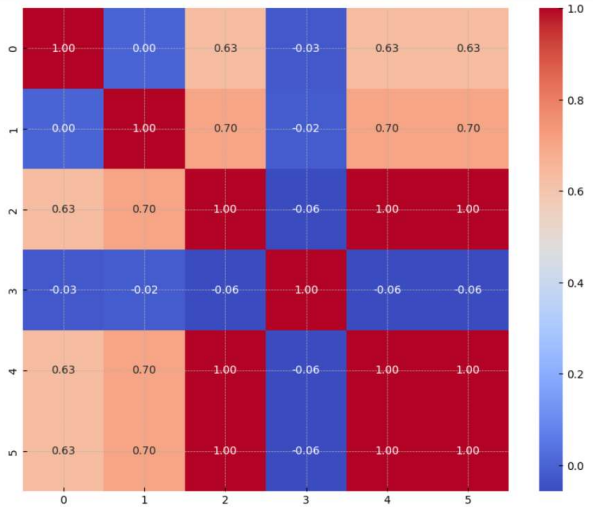

Figure 3

Once we have identified some numerical correlations through a correlation matrix, we can begin preliminary constructions for our models. Our plotted matrix shows an identical correlation between total, gross income, and COGS. This suggests that these provide the same information, therefore, total and COGS will be omitted when training models that will predict gross income. However, unit price and quantity have shown strong, but not identical, correlations to gross income making them the best candidates to include in our first model.

Other categorical data provided such as product line, day, and time slots were shown to have little effect on model training and failed to provide effective predictions of gross income or unit price.

Two of the columns, Date and Time, are in formats which need reformatting or encoding. To do this, custom transformers have been built to replace their values with easily digestible information categories. Both have been encoded into integers to represent the classes of days of the week and time of day respectively.

Now that we know what values to scale and provide our model, we also encode some categorical data values such as branch, gender, and product line using a one-hot encoding method to aid the model's predictions.

IV. Model Creation, Training, and Evaluation

Scikit-learn provides extremely simplistic methods of creating pipelines which can be used to assemble our preprocessing and models into one object. This is where we can also add some regularization to our models via a Lasso regularization. Additionally, the implementations of training our models are condensed into single lines of code.

After this has been used, matplotlib functions are utilized to plot our model's predictions compared to true values. Furthermore, scikit-learn also provides some functions for evaluating scoring methos such as r2 score and accuracy which can also be used. With this, we can then calculate the 95% confidence intervals for these scores from our model's predictions.

V. Model 1

Our first model predicts the gross income for the business generated by each order through a linear regression learning model that incorporates a lasso regularization and grid search cross validation. The tuning of needed hyperparameters is handled within our built pipeline which reported a best λ or alpha value of 0.15. Additionally, our r2 score had a mean of 88% and a 95% confidence interval of (89.342%, 89.343%). This model did not consider the following features due to low correlation: 'Invoice ID', 'City', 'gross margin percentage', 'cogs', 'Payment', 'Total'. Our main correlating data came from the unit price and quantity data points. Other values such those that were categorical, did not have a significant effect on the prediction of gross income, which would suggest a small or no relation to the value. See Figure 4.

```
Best found hyperparameters for Grid Search: {'alpha': 0.15}
 Highest R^2 score for Grid Search: 0.8870963838646044
95% CI for R^2 of Grid Search): (0.8934294652454788, 0.8934345695036503)
```

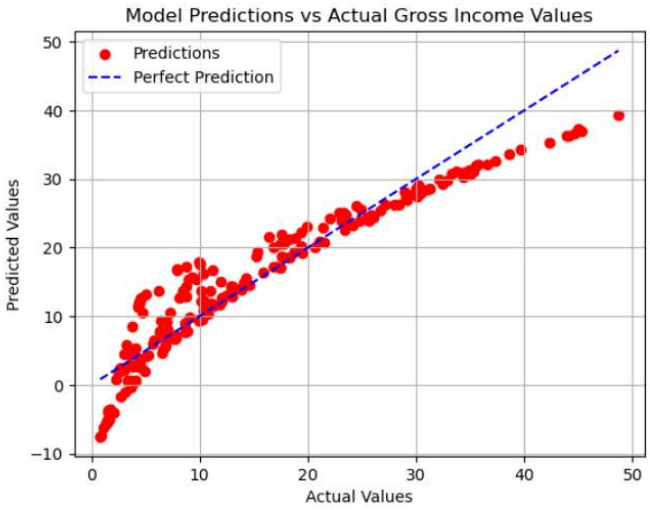

Figure 4

VI. Model 2

Our second model predicts the unit price of the goods purchased at the supermarket using a lasso regularization with a random search cross validation. When viewing the previous correlation matrix, we can identify that the features of total, COGS, and gross income are highly correlated to unit price. However, they portray the same information once again, and only one of these will be implemented into our model. One interesting phenomenon was that even though the quantity was very poorly correlated to unit price on the correlation matrix, adding this feature into the model's calculations unexpectedly provided an ample increase in model prediction accuracy. Therefore, it was implemented into our final model. Additionally, the encoding of several categorical data features has also been implemented into this model but has shown very little effect of the prediction outcomes when removed or altered. The best λ or alpha value found for this model was 0.3449. The r2 score had a mean of 0.77 and its 95% confidence interval was (75.424%, 80.052%). See Figure 5.

VII. Model 3

Our third model was a classifier which was to classify purchases to which day of the week they were made. This model used a logistic regression learning model. As the days of the week were encoded from the date column, their definitions are 0-6: Mon-Sun in ascending and chronological order. This model included the same numerical features as Model 2 for consistency. The scoring metric used for this model was pure accuracy. This model achieved an accuracy score of only 16% and a 95% confidence interval of (10.919%, 21.080%). From this incredibly low accuracy, it can be deduced that either our given data does not provide good estimations for classifying which day of the week orders are made, or that there was a fundamental issue in the implementation of the model. See Figure 6.

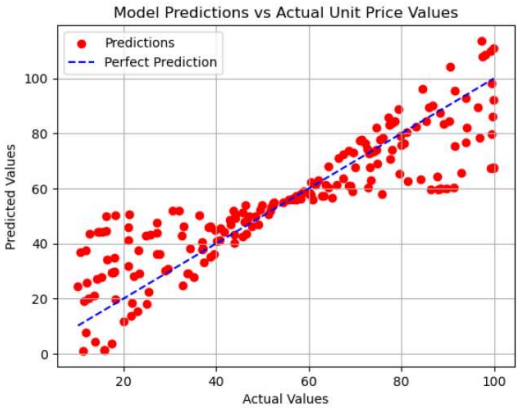

Figure 5

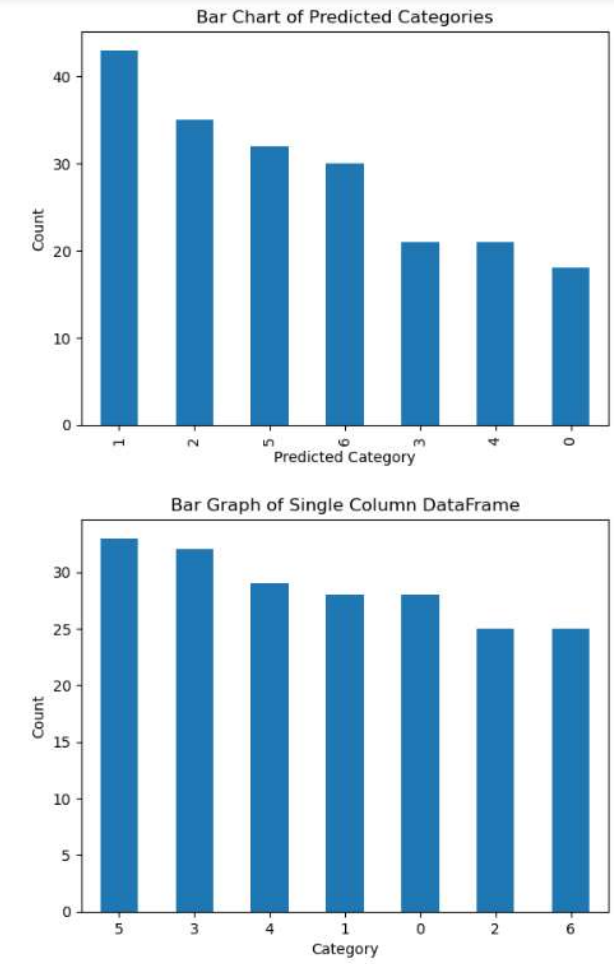

Figure 6

A. Authors and Affiliations

Andrew Perez-Ledo (University of Florida)