Building and Evaluating Machine Learning Algorithms

Diana Sanchez

Undergraduate Student

University of Florida

***Abstract*—This report describes the process of data analysis, visualization techniques, training data using machine learning algorithms for regression and classification, model selection, and performance evaluation for a supermarket sales dataset. The observations made during training and performance evaluation will reveal which model is most suitable for predicting gross income, unit price, and day of purchase.**

I. Introduction

Regression and classification are the two types of Supervised Learning. This means that the input data is mapped to the desired output values using the given labeled training data. To characterize the relationship between the input data and continuous desired output, regression could be used since it's a form of predictive modeling. Classification is also a form of predictive modeling, but it forms a relationship between the input data and a given set of categorical labels. The goal of this project was to implement regression and classification tasks to make predictions using different models and compare each model's performance for the given target value. Specifically, we will be using regression to predict the gross income and unit price using the supermarket dataset, and classification for predicting the day of purchase.

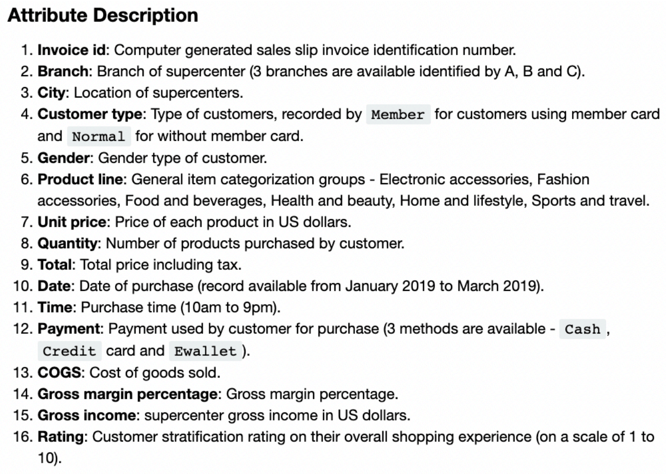
II. Predicting gross income

A. Framing the Problem

Since our goal for this part is to predict the gross income given a set of inputs using supervised learning, we are going to use a regression task to make the model. Classification could only be used if we separate the gross income into categories (low, medium, high), which is not our desired approach.

B. Get the Data

In this project, we used a supermarket sales dataset, which included 3 months' worth of data recorded from three different branches. The features in the dataset are described in Figure 1.



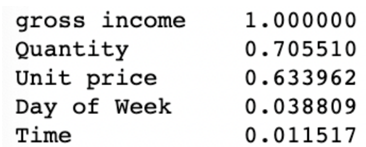
Attribute Description

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

Fig.1 Supermarket_sales.csv attribute description

C. Discover and Visualize Data

Before splitting the data into training and test samples, stratification is used to ensure the prior probabilities of each feature of gross income are kept in the training and test sets. To successfully perform this, it was necessary to find the most predictive attribute for gross income. This was done by evaluating the correlation between gross income and all other numerical values. It was found that the attribute 'Quantity' had the most correlation to gross income.

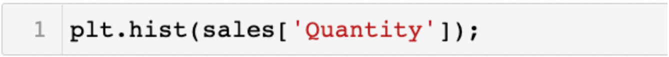


```
gross income    1.000000
Quantity        0.705510
Unit price      0.633962
Day of Week     0.038809
Time            0.011517
```

Fig.2 Correlation matrix between gross income and numerical attributes.

In order to see how to stratify the 'Quantity' attribute, we would look at the histogram to view where the data is mainly concentrated and apply them to bins so that it is evenly distributed.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE



```
plt.hist(sales['Quantity']);
```

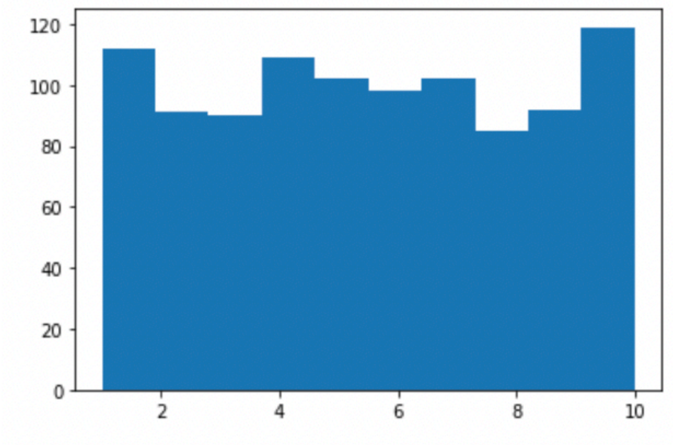


Fig.3 Quantity attribute histogram before placing in bins

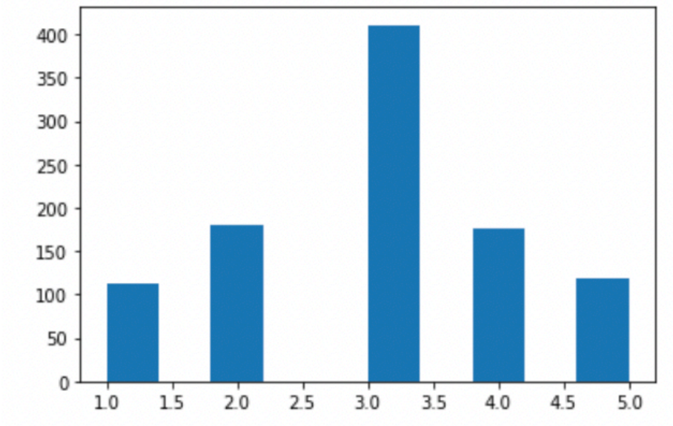


Fig.4 Quantity attribute histogram after placing in bins

Now we can partition it into training and test samples before cleaning the data.

D. *Prepare Data for Machine Learning Algorithms*

Since this prediction only needs the attributes: 'Product line', 'Date'. 'Time', 'Quantity', and 'Unit price', we can drop all of the other attributes. Because we are dealing with three categorical attributes ('Date', 'Time', and 'Product line), we had to convert the categories into numerical representations using One-Hot encoding. One-Hot encoding encodes each category into a sparse vector with as many entries as there are categories, where 1 represents it belongs in the category and a 0 means it doesn't. We also had to implement feature scaling for the numerical attributes so that the features have a similar range, thus optimizing the models. Since the numerical attributes don't have any significant outliers, using Min-Max scaling would be the best approach to scaling the data. Creating a column transformer that applies one hot encoding to the categorical attributes and Min-Max Scaler to the numerical attributes would be the best method since it pipes the two commands together into a single preprocessed pipeline ready to use.

E. *Multiple linear regression with/without Lasso regularization*

We trained a multiple linear regression model with lasso regularization using the preprocessed pipeline and grid search to find the most optimal hyperparameters, and a linear regression model without lasso.

F. *Findings for Model 1*

One of the main questions we are looking answer for is how the gross income is affected by the input attributes. As we saw earlier in Figure 2, we are able to see the correlations between the input attributes to the gross income attribute. In the table, Quantity has the highest correlation, meaning that the number of products a customer purchases highly affects the gross income. Unit price was also an attribute with high correlation to the gross income, while Time was less significant.

When we trained and tested the multiple linear regression with lasso regularization, the only features that were included were Product line, Time, Unit Price, Quantity and Day of week. This is because the other features provided had no correlation to the gross income. We found that the best hyperparameter for alpha, λ, was 0.076 using Grid Search Cross-Validation.

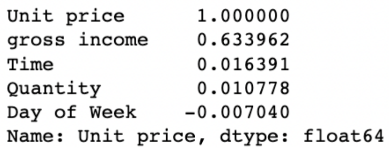
When comparing the model that used Lasso regularization to the one that didn't, we found that the model that didn't use Lasso regularization had a higher coefficient of determination and it produced a 95% confidence interval with smaller possible errors than the model that used Lasso regularization

III. Predicting Unit Price

Since the goal of this problem is to predict the unit price given a set of inputs using supervised learning, regression would also be used since the unit price is not split into categories. We will be using the same dataset we used for gross income.

A. *Discover and Visualize Data*

The first step before splitting out data into training and test samples was to find the most predictibe attribute for Unit price by looking at the correlation matrix (shown below).



```
Unit price       1.000000
gross income     0.633962
Time             0.016391
Quantity         0.010778
Day of Week     -0.007040
Name: Unit price, dtype: float64
```

Fig.5 Correlation matrix for Unit Price

Since gross income has the highest value, it is the most predictive attribute for Unit price. In order to place the gross income data into bins, we have to look at the historgram for gross income to see how it's partitioned.

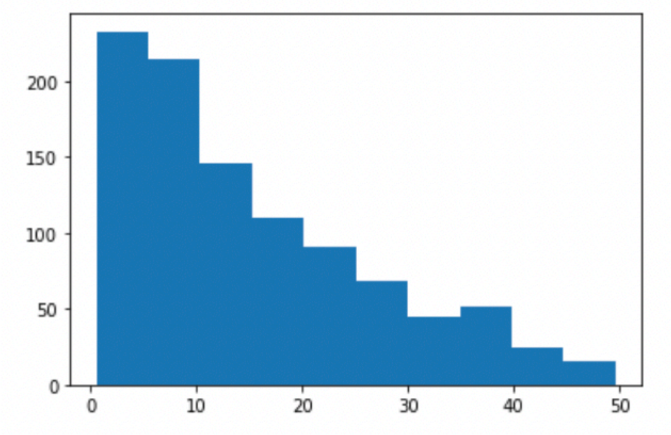


Fig.5 gross income attribute histogram before placing in bins.

After the data was partitioned, it was passed into the training and test split function to be used for data cleaning and pipeline models.

B. Prepare Data for Machine Learning Algorithms

When predicting Unit Price, the following attributes were used as input variables: 'gross income, 'quantity', 'day', 'timeslot', and 'product line'; All other attributes were dropped from the dataset.
One-Hot Encoding was applied to the categorical attributes ('Product line', 'Day of Week', 'Time') and Min-Max Scaler was applied to the numerical attributes ('gross income' and 'Quantity') in order to perform feature scaling. These functions were added to a column transformer so that it is ready to use as a preprocessed pipeline.

C. Multiple linear regression with/without Lasso regularization

Using the preprocessed pipeline and the Lasso() library, we created a parameter grid to apply a variety of alpha, λ, values to the model using Grid Search Cross Validation in order to find the most optimal value.

D. Findings for Model 2

For this problem we assessed how the Unit Price is affected by gross income, quantity, day, time, and product line by using the correlation matrix to measure the relationship of Unit price with each of the input attributes. In doing so, we were able to see that the gross income was a highly predictive attribute for unit price, meaning that the price for each product is highly affected by the gross income. One of the least correlative features was 'Day of Week', which makes sense because the unit price for a product doesn't change based on the day of the week.

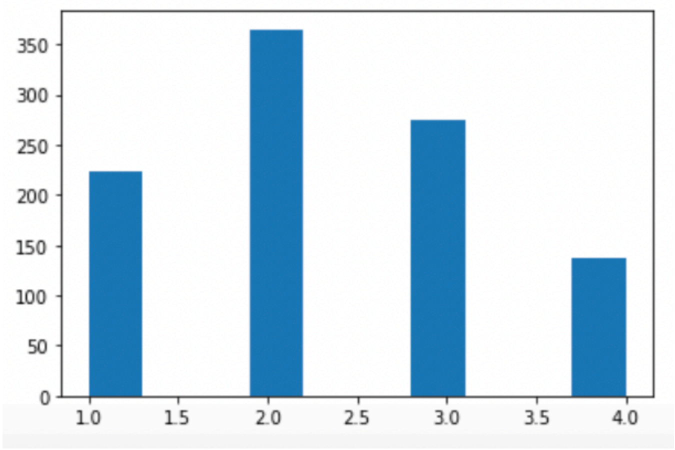


Fig.5 gross income attribute histogram after placing in bins.

When applying Lasso regularization for this model in a Grid Search CV, I found the hyperparameter, λ, that worked best for this dataset was 0.1. The features such as invoice id, branch, city, customer type, gender and payment were disregarded for this model because they didn't offer any relevance to the target value 'Unit Price'.

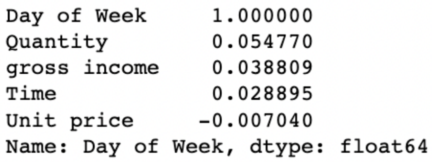
Finally, when comparing the model that used lasso regularization to the one that didn't, it was found that the model that used Lasso regularization had a higher coefficient of determination and produced a 95% confidence interval with smaller possible errors that the second model that didn't use lasso regularization.

IV. Predicting Day of Purchase

When it comes to predicting an attribute that is categorical, the best approach would be to use classification rather than regression. Since the day of purchase, also known as the 'Day of Week' in my dataset, is split into 7 categories (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and Saturday), we applied two classifiers to predict the day of purchase: Decision Tree Classifier and Random Forest Classier.

A. Discover and Visualize Data

Even through this is a classification task, the preprocessing techniques are the same as regression: stratify the most predictive attribute, partition the data into training and test sets, apply one-hot encoding to the categorical attributes and Min-Max scaling to the numerical, pass it into a preprocessed pipeline for later use.



```
Day of Week      1.000000
Quantity         0.054770
gross income     0.038809
Time             0.028895
Unit price      -0.007040
Name: Day of Week, dtype: float64
```

Fig.6 Correlation matrix for Day of Week

The day of the week had the same predictive attribute as the first model, so the partitioned data for Quantity could be used in this case.

A. Findings for Model 3

After finding the most optimal parameters for the classifier models, it was found that both classifiers have a similar accuracy score of around 0.16, but the random forest classifier model produced a 95% confidence interval with smaller possible errors, so it was the most optimal classifier for this case.

References

[1] Sklearn.compose.ColumnTransformer, *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html.