Project 1: Supermarket Sales

EEL5934: Applied Machine Learning Project 1

Nathan Oliver Noronha
Electrical and Computer Engineering
University of Florida
Gainesville, USA
n.noronha@ufl.edu

***Abstract*—This report is made for the purpose of Project 1 of the course EEL5934: Applied Machine learning systems. The data set used for the project is supermarket sales of various branches in cities. According to the mentioned requirements, the appropriate are regressors and classifiers are used in the project and the corresponding decisions and explanations are given.**

I. Data Preprocessing

There are 16 features present in the dataset. The dataset does not have any missing samples, hence there is no need to drop any columns or fill NaN values. The date and time column are considered to be of object type, hence both these columns are converted into date time format, for ease of encoding.

A. Encoding Date Feature

After the conversion of Date feature to the date time format, we are converting the format into the day of the week and performing One Hot encoding using the scikit learn pipeline.

B. Encoding Time Feature

After the conversion of Time feature to the date time format, we are converting the format according to the part of the day based on the hour. It is further encoded using One Hot encoding using the above pipeline. The range is converted as follows:

- 1 = Morning (10:00 - 11:59),
- 2 = Afternoon (12:00 - 17:00)
- 3 = Evening (17:01 - 19:00)
- 4 = Night (19:01 - 21:00)

II. Predicting Gross Income using Linear Regression

Two models were trained, one with Lasso and the other without Lasso. The models were run on validation sets to get the scores on 10 folds. The folds were made using Stratified K fold validation so as avoid overfitting. The model without lasso regularizer seems to overfit the data as the weights of the coefficients of the 'best line' are not penalized. This is resolved when lasso regularizer is used which gives a very good score. The solution space is narrowed down using Random Search as it is faster. Furthermore, Grid Search is used to get a good tuned hyperparameter.

Method	Accuracy	Confidence Interval
Without Lasso	100%	Overfitting
With Lasso	99.7%	[0.999999968, 0.999999977]

A 95 percent confidence interval between [0.9999999692, 0.9999999767] is achieved for predicting Gross income. The narrow the Confidence interval, the better accuracy we have.

A. Relationship of gross income with other features

Gross income has perfect correlation with the features Total and cogs. Next, it has a linear relationship with the feature Unit price. The most informative features that have high coefficients are:

Attribute	Coefficients
Total	1.16498333e+01
Quantity	2.79006237e-03
Unit Price	2.28689184e-03

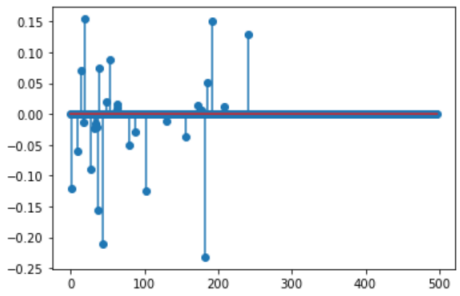

Fig. 1. Relationship of gross income with other features

B. Optimal Hyperparameter and considered features

Random Search and Grid search was used for hyperparameter tuning. The hyperparameter alpha was given the range between 0 and 3 (Narrowed down from Random Search) to get an optimal value for alpha. The optimal value for the lasso regularizer came out to be 0.001437.

Invoice ID and and gross margin percentage were not considered as Invoice ID is a unique value and gross margin percentage has a constant value. City was also removed as the Branch feature corresponds to City. Hence anyone can be

kept. Similarly, cogs can also be removed since the cogs is the same as gross income.

III. Predicting Unit Price using Linear Regression

Similar to predicting gross income, Here two models are trained, one with Lasso and the other without Lasso. The data preprocessing pipeline encodes features 'Branch', 'Customer type', 'Gender' as Ordinal as they have few categories. Features 'Date', 'Time','Product line','Payment' are encoded using One hot encoding as they have many categorical samples which could cause data biasing for the model if ordinal encoded. The models were run on validation sets to get the scores on 10 folds. The folds were made using Stratified K fold validation so as avoid overfitting. We use a Lasso Regularizer to penalize the less important features. The solution space is narrowed down using Random Search as it is faster. Furthermore, Grid Search is used to get a good tuned hyperparameter.

Method	**Accuracy**	**Confidence Interval**
Without Lasso	76.8%	[0.750, 0.785]
With Lasso	77.4%	[0.758, 0.791]

The narrow the Confidence interval, the better accuracy we have. Hence, the model without Lasso has a better accuracy than with Lasso.

A. Relationship of Unit Price with other features

Gross income has perfect correlation with the features Total and cogs. Next, it has a linear relationship with the feature Unit price. The most informative features that have high coefficients are:

Attribute	**Coefficients**
Quantity	-2.21219971e+01
Total	3.21668485e+01
Rating	4.09380303e-01
Credit card	-2.39234912e-01

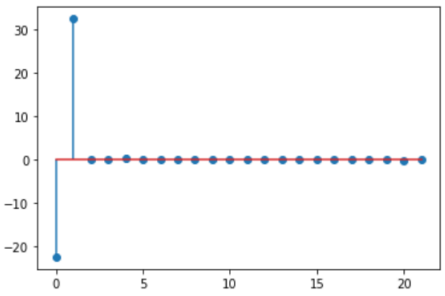

Fig. 2. Relationship of Unit Price with other features

B. Optimal Hyperparameter and considered features

Random Search and Grid search was used for hyperparameter tuning. The hyperparameter alpha was given the range between 0.2 and 0.5 (Narrowed down from Random Search) to get an optimal value for alpha. The optimal value for the lasso regularizer came out to be 0.3605.

Invoice ID and and gross margin percentage were not considered as Invoice ID is a unique value and gross margin percentage has a constant value. City was also removed as the Branch feature corresponds to City. Hence anyone can be kept. Similarly, cogs can also be removed since the cogs is the same as gross income.

IV. Classifying Gender using Logistic Regression

To train a classifier for gender prediction, I used ordinal encoder for the gender training set and one hot encoder for other categorical attributes. The sales dataset was split using stratification on the target gender feature. We are training the model using Logistic regression with a Lasso regularizer. The hyperparameters are tuned using Random Search and Grid Search CV. The solution space is narrowed down using Random Search as it is faster. Furthermore, Grid Search is used to get a good tuned hyperparameter.

For Branch C attributes, we study the relationships between gender, product line, payment and gross income. We use the polynomial features with interaction enabled to produce features with degree equal to or less than 2. The result we get for the coefficients is show below.

Attribute	**Coefficients**
Intercept	0.0180
Product line	0.0806
Payment	-0.1344
gross income	0.0677
Product line + Payment	-0.1120
Product line + gross income	0.0278
Payment + gross income	0.0099

The higher the coefficients, the better the direct relation between gender and the feature. For negative coefficients, it would imply a inverse relation.

A. Parameters values for all attributes for male customers

Since we used One hot encoding, the matrix was sparse and there were attributes which were not essential for predicting the gender. Lasso helps to increase model interpretation. by penalizing the less important features. Hence we used a Lasso regularizer to penalize the unnecessary interactions.

Below is the plot of parameter values for the Male gender:

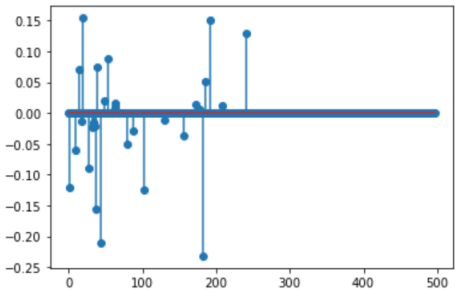

Fig. 3. Parameters values for all attributes for Gender = Male

B. Most Informative features

The most informative features for all the branches would be the features with the highest coefficients. They are as follows:

Attribute	Coefficients
EWallet	0.1541
Rating + Saturday	0.1498
Branch B + Afternoon	0.12876
Unit price + Evening	0.08759

Other coefficients are also presented in the training file which have a lower value of coefficient.

V. Classifying Customer Type using Logistic Regression

To train a classifier for Customer Type prediction, I used ordinal encoder for the Customer Type training set and one hot encoder for other categorical attributes. The sales dataset was split using stratification on the target gender feature. We are training the model using Logistic regression with a Lasso regularizer. The hyperparameters are tuned using Random Search and Grid Search CV. The solution space is narrowed down using Random Search as it is faster. Furthermore, Grid Search is used to get a good tuned hyperparameter.

For Branch C attributes, we study the relationships between customer type, gender, day and timeslot. We use the polynomial features with interaction enabled to produce features with degree equal to or less than 2. The result we get for the coefficients is show below.

Attribute	Coefficients
Intercept	-0.00455
Gender	0.0051
Date	-0.00044
Time	-0.0053
Gender + Date	0.01375
Gender + Time	0.00063
Date + Time	-0.0140

The higher the coefficients, the better the direct relation between gender and the feature. For negative coefficients, it would imply a inverse relation.

A. Parameters values for all attributes for normal customers

Since we used One hot encoding, the matrix was sparse and there were attributes which were not essential for predicting the gender. Lasso helps to increase model interpretation. by penalizing the less important features. Hence we used a Lasso regularizer to penalize the unnecessary interactions.

Below is the plot of parameter values for the Normal Customer Type:

B. Most Informative features

The most informative features for all the branches would be the features with the highest coefficients. They are as follows:

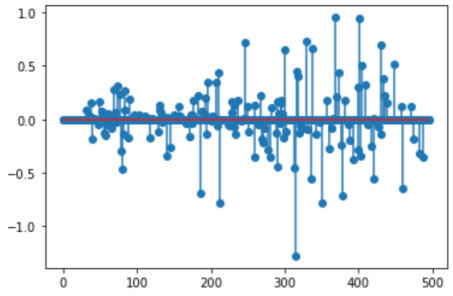

Fig. 4. Parameters values for Normal Customer Type

Attribute	Coefficients
Health and beauty + Afternoon	0.9554
Sports and travel + Saturday	0.9424
Fashion accessories + Cash	0.7343
Branch C + Electronic accessories	0.7205
Credit card + Night	0.7004

Other coefficients are also presented in the training file which have a lower value of coefficient.

VI. Classifier to predict day of purchase

Classifying the day of the purchase is a multi class classification problem. I will be using Random Forest Classifier and Logistic Regression Classifier, train them and then compare the two. We will be using Random Search CV to narrow down on the solution space and then use Grid Search CV to tune the Hyperparameters. An issue with the dataset is its data imbalance which has caused the classifiers to get a low accuracy score.

A. Selection of Classifiers

Below are the training results for the Random forest classifier:

Training Set:

	precision	recall	f1-score	support
0	0.08	0.04	0.05	25
1	0.21	0.34	0.26	32
2	0.22	0.21	0.21	29
3	0.12	0.07	0.09	27
4	0.21	0.11	0.14	28
5	0.11	0.24	0.16	33
6	0.12	0.04	0.06	26
accuracy			0.16	200
macro avg	0.16	0.15	0.14	200
weighted avg	0.16	0.16	0.15	200

Fig. 5. Random Forest to classify Day of the Week

Below are the training results for the Logistic Regression classifier:

	precision	recall	f1-score	support
0	0.19	0.17	0.18	100
1	0.19	0.24	0.21	126
2	0.26	0.25	0.26	114
3	0.23	0.20	0.21	111
4	0.23	0.12	0.15	111
5	0.20	0.24	0.22	131
6	0.16	0.20	0.18	107
accuracy			0.20	800
macro avg	0.21	0.20	0.20	800
weighted avg	0.21	0.20	0.20	800

Fig. 6. Logistic Regression to classify Day of the Week

ACKNOWLEDGMENT

I would like to extend our sincere appreciation and gratitude to Dr Catia Silva and the Teaching Assistant's for their valuable support and feedback. I would also like to thank the department of Electrical and Computer Engineering at the University of Florida.

REFERENCES

[1] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition (Aurélien Géron, 2019)

[2] Python Machine Learning - Third Edition (Sebastian Raschka , Vahid Mirjalili, 2019)