Joint Project ML & SQL:
H_Retail Written Document

Team 3

Hult International Business School

Professor Luis Escamilla

March 11th, 2022

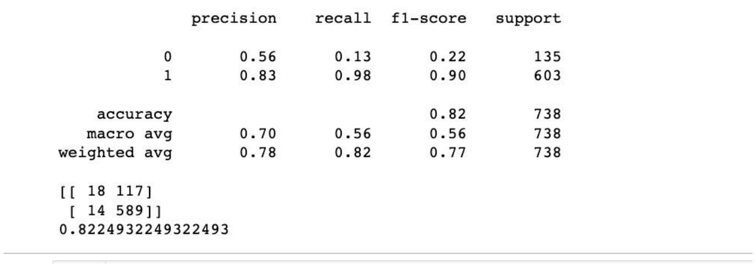
Summary

With a high training rate of 82% and a high testing rate of 80%, the algorithm was neither underfitted nor overfitted. The model and accuracy will not change even if our random seed was changed from 219. Based on the total dataset there were more categorical variables than quantitative which may be a slight limitation in leaving the model a bit biased. The target of the algorithm was to test for whether the clients were wholesalers or retailers, and these were coded as 1 or 0.

The Model

A total of 983 records formed the dataset to be trained and tested. 75% of the data(737) was trained while 25% of the data (246) was tested. The model was created with a random seed of 219. The trained model was 82% accurate while the tested model was 80%. Thus, a high training accuracy and a high testing accuracy, the model was neither under-fitted nor over-fitted.

Trained model:



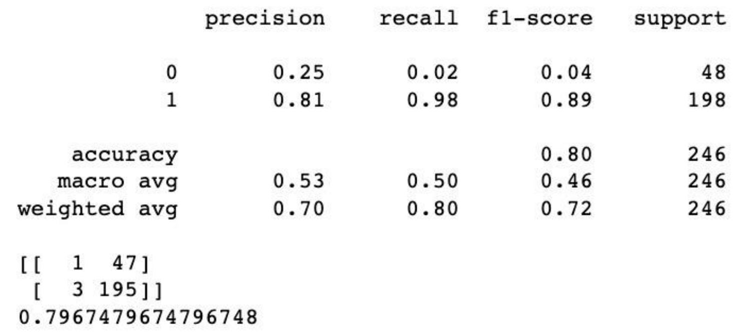
```
              precision    recall  f1-score   support

           0       0.56      0.13      0.22       135
           1       0.83      0.98      0.90       603

    accuracy                           0.82       738
   macro avg       0.70      0.56      0.56       738
weighted avg       0.78      0.82      0.77       738

[[ 18 117]
 [ 14 589]]
0.8224932249322493
```

Tested model:



```
              precision    recall  f1-score   support

           0       0.25      0.02      0.04        48
           1       0.81      0.98      0.89       198

    accuracy                           0.80       246
   macro avg       0.53      0.50      0.46       246
weighted avg       0.70      0.80      0.72       246

[[  1  47]
 [  3 195]]
0.7967479674796748
```

Assumptions

Given a different random seed of 123, the accuracy rate decreases by an insignificant number, and with a random seed of 400, the accuracy rate increases insignificantly. Thus, there is no fundamental change in the model with a change in the random seed.

Limitations

One limitation of our algorithm is that based on the shape of the dataset, extracted from only three months, October, November, and December of 2011, it may have an accurate training and testing rate but only represents about 20% of the whole 4098 customers.

Another limitation is the number of variables that are unbiased and quantitative. There were only two quantitative variables amongst categorical variables, representing just the demographics of the client.

Advantages based on the ERD

It helped to understand how the tables are connected and made the writing of the query rather seamless and understandable. This allowed for our team to better visualize where information was being pulled from. In a bigger database with much more data, we could construct queries to only include fields we needed to faster serve our analytical needs.

Replicable Scenarios

A typical usage of the classification algorithm can be for a sports retail company optimizing on who to run campaigns for based on the type of products they buy(i.e., size, color, quantity). With this can confirm whether their customers are organizations or individuals, male or female, have children or not then they can channel the right campaigns to address them.

Again, a chef can use a classification algorithm to determine the type of clientele and type of menu that is preferred based on the time the customers enter the restaurant or when they order and the amount they spend per visit.

Key Learnings

The most difficult part of this assignment was figuring out which data to pull in from SQL to use for Python. The most enjoyable part of this assignment was the collaborative environment which enhanced learning. We struggled to figure out which exact columns were needed for Python. Some of the opportunity areas that are team got to focus on in were inner join, aggregate functions, creating dummy variable, and logistic regressions. These were areas that we got to practice and become efficient at.

The team worked efficiently and communicated well together. We all collectively worked on the SQL, Python, and written document together. We would screen share and work on the code together. This helped avoid errors and with debugging codes. In the future, we could work on spending more time brainstorming ideas. We jumped right into SQL but in the future, we can spend more time brainstorming ideas.