GMMA 869
Machine Learning and AI

Dr. Stephen W. Thomas

Individual Assignment 1
July 5, 2020

William Chak Lim Chan

YUM, ORANGE JUICE!

PREAMBLE

Download the file *OJ.csv*. The target feature is *Purchase*. The rest of the features are self-explanatory, hopefully.

SCENARIO

One cup of fresh orange juice has 124 mg of vitamin C, which is 200% of the recommended daily intake of vitamin C for an adult. With this as (completely unrelated) motivation, your task is to build a model to predict whether a grocery store customer will Purchase Citrus Hill (CH) or Minute Maid (MM) orange juice.

TASKS

1. [Text] Choose an appropriate metric to analyze a model's performance. Justify.

We will use Precision to analyze the model's performance as we want to know how accurate the model is with its "Yes" predictions.

2. [Code] Build a prediction model as follows:
    a. Preprocess the data however you see fit. In code comments, describe what you did and why.
    b. Split the data into training and testing sets. In code comments, describe what you did and why.
    c. Build three different models, using three different machine learning algorithms. (Any three will do.) Tune each model. Print out the best hyperparameter values for each model. Print out performance of each fine-tuned model.
3. [Text] Using business language (not technical language), describe and compare the performance of each fine-tuned model.

Logistic Regression Model:

- Accuracy: 83.5%
- AUC: 88.7%
- Recall: 71.1%
- Precision: 84%
- F1: 77%

This is the best model to predict whether a customer is going to buy CH or MM orange juice. It was able to predict the correct customer selection 84% of the time in the test data, up from 81% in the training data. The Logistic Regression model had low recall but high precision meaning it didn't get all of the predictions right, but when it did, it had a high level of accuracy. The F1 score is a measure of that balance. The model saw that the LoyalCH and PriceDiff were the two most important factors in predicting CH or MM.

Linear Discriminant Analysis Model:

- Accuracy: 66%
- AUC: 74.6%
- Recall: 15%
- Precision: 84%
- F1: 26%

The Linear Discriminant Analysis Model was the worst model from a performance standpoint. The model's accuracy was 66% meaning that it wasn't able to predict whether a customer would buy CH or MM very well. The model also wasn't very good predicting overall, but the ones that were predicted correctly were accurate. I would not deploy this model for production. The model saw that the LoyalCH and ListPriceDiff were the two most important factors in predicting CH or MM.

Extreme Gradient Boosting Model:

- Accuracy: 76.8%
- AUC: 84.5%
- Recall: 66.4%
- Precision: 71.8%
- F1: 69%

The Extreme Gradient Boosting Model was the 2nd best model from a performance standpoint. It was able to accurately predict what the customer would by 77% of the time. It was very average at getting predictions right including False Positives and False Negatives. It performed worst on False Negatives in fact. The model saw that the LoyalCH and PriceDiff were the two most important factors in predicting CH or MM.
Best: Logistic Regression
Middle: XGBoost
Worst: Linear Discriminant

4. [Text] Overall, which model is best suited to this business problem? Justify.

The Logistic Regression model is the best model according to the five performance measures both on test and train data, in fact, performance increased on the test data set suggesting that the model is not overfitted and seems to generalize well with both datasets. We gave more weight to the Precision measure in our evaluation since we want to be more correct in predicting "Yes". We simply need a "yes" or a "no" to a customer purchasing MM or CH. The false positives/negatives wouldn't really impact the business case that much here since I'm assuming that the results will be used to serve up advertisements or to refine a marketing campaign, but it's still important to get them right.

5. [Text] Is this model good enough to deploy today? Justify. If you had more time, how could you make the performance of your model better?

Our model is good enough to deploy since it's within the 75%-90% range across all performance measures. A 77% score in F1 is enough to deploy and start making predictions. Given that the Logistic Regression model outperformed on all measures when compared to the other three models, we wouldn't consider ensembling them together. We would straight deploy it to an AWS or Azure server and start making predictions.