WOW Portfolio Report

Group 14: Hsueh-Yao Lu, Jingyi Leng

Executive Summary

Business Problem

Using combined information of customers and products to identify whether product shipments are likely to be delivered on time or not for an international e-commerce company that sells electronic products.

Metrics

AUC is defined as our primary aggregate metric for this problem because we are mainly concerned about how well the classifier separates the on-time and not-on-time categories.

Accuracy is also considered to assess the overall percentage of error.

Confusion matrix is presented as a direct observation of classification performance.

Performance

The Random Forest Classifier outperforms all other basic and ensemble learners and reaches the AUC of 0.714 and the Accuracy of 0.694 after hyperparameter tuning. The Auto-ML model built by Auto-Sklearn reaches an AUC of 0.718 and Accuracy of 0.693, giving a slight improvement on AUC compared to our manually built Random Forest Classifier.

The overall prediction performance on the test data is not that satisfying, mainly due to the large number of false negative misclassifications as can be told by the confusion matrix.

Classifier	AUC	Accuracy	Confusion Matrix
Auto-Sklearn	0.718	0.693	$\begin{bmatrix} 452 & 80 \\ 325 & 462 \end{bmatrix}$
Random Forest	0.714	0.694	$\begin{bmatrix} 432 & 100 \\ 303 & 484 \end{bmatrix}$
TPOT	0.691	0.682	$\begin{bmatrix} 394 & 138 \\ 282 & 505 \end{bmatrix}$

Suggested Improvements

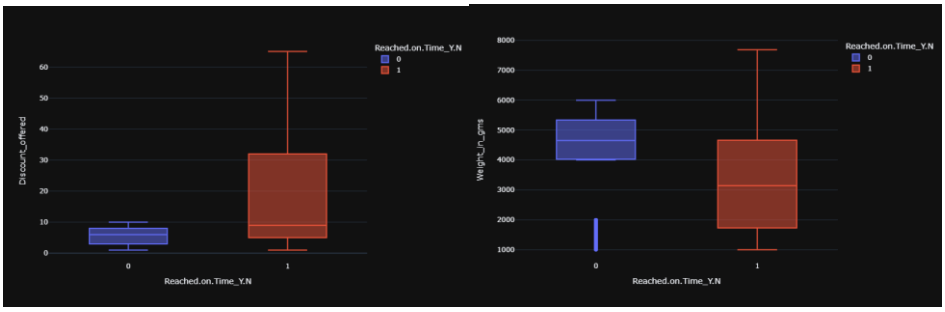
The unexpected weak performance on the dataset may be due to the lack of factors to fully reflect the cause for on-time or not-on-time delivery. More dimensions of data like weather condition, express company, purchase time (to reflect whether it's purchased near festivals like Black Friday, Christmas) can be collected to provide more explanatory power to the target and improve model performance.

The insufficient feature engineering of current predictors may also give rise to the weak performance. The relationship among predictors can be further discovered based on current efforts.

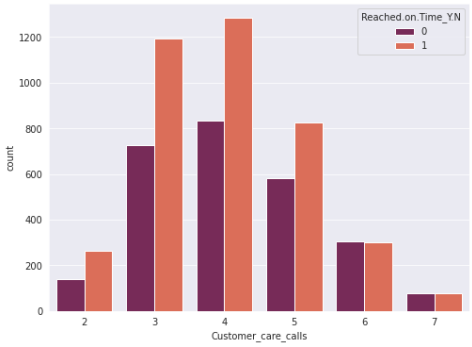
Exploratory Data Analysis

We applied EDA to first grab some knowledge about the data:

- The dataset is of normal size with 6598 instances and 10 predictors.
- It's quite clean with neither missing value nor duplicate.
- Predictors 'Prior_purchases' and 'Discount_offered' are slightly skewed and in need of skewness correction.
- Predictors 'Discount_offered' and 'Weight_in_gms' have relatively high correlation with the target 'Reached.on.Time_Y.N' of 0.38 and -0.27 compared with other predictors.
- The boxplot of 'Discount_offered' indicates that if the offered discount for a product is greater than 10, it's supposed to be delivered not on time;
- The boxplot of 'Weight_in_gms' indicates that if the weight of a product is lower than 4000 grams, it's highly likely to be delivered not on time.



- The countplot of ‘Customer_care_calls’ indicates the higher probability of on-time delivery if customers have more than 5 times enquiries about shipment.

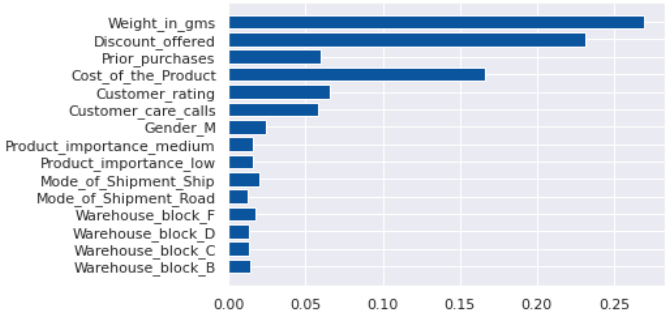


- Outliers need to be addressed for predictors ‘Discount_offered’ and ‘Prior_purchases’.

Data Preprocessing

Data preprocessing is applied to the raw data to get prepared for building classifiers:

- Skewness correction for predictors ‘Prior_purchases’ and ‘Discount_offered’.
- Outliers are winsorized using the Tukey Rule for predictors ‘Discount_offered’ and ‘Prior_purchases’.
- Dummy encoding is applied for categorical predictors: 'Warehouse_block', 'Mode_of_Shipment', 'Product_importance', 'Gender'.
- RFE and Feature Importance Analysis are both applied for feature selection. We use the top 6 predictors (‘Weight_in_gms’, ‘Discount_offered’, ‘Cost_of_the_Product’, ‘Customer_ratings’, ‘Prior_purchases’, ‘Customer_care_calls’)that are defined as important features by both of the methods to further build our classifiers. PCA is also applied while not much information is gained because of the similar explanatory power of components.



Classification Model Selection

In this part, we fit a set of classifiers including Perceptron, Logistic Regression, Support Vector Machine (SVM) (RBF kernel), Decision Tree, Naive Bayes, k Nearest Neighbors, Multilayer Perceptron (MLP), Random Forest, XG Boost, and Light GBM. We used K-fold cross validation to train our data and we set k=5. We also standardized and performed Synthetic Minority Oversampling Technique (SMOTE) on our training data. We calculated the mean accuracy and AUC for each classifier and got the results below:

Classifier	AUC	Accuracy
Random Forest	0.733	0.689
XGBoost	0.729	0.688
Decisiom Tree	0.727	0.685
Support Vector Machine	0.722	0.673
MLP	0.721	0.678
Naive Bayes	0.703	0.647
LightGBM	0.694	0.675
Logistic Regression	0.677	0.649
kNN	0.656	0.649
Perceptron	0.607	0.595

Since we are using AUC as our main metric, we will choose Random Forest as our classifier and proceed from here. This result is consistent since most ensemble classifiers have higher AUC and accuracies.

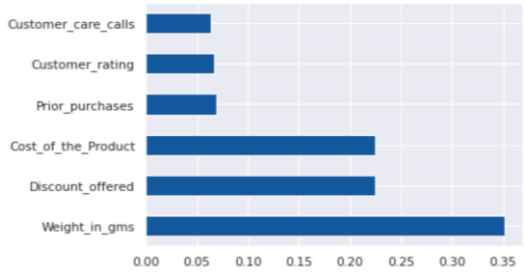
We then used grid search to optimize the hyperparameters for our Random Forest model. After trying a few different combinations of the parameters, we got the following results for the optimal hyperparameters and the accuracy:

```
[[432 100]
 [303 484]]
{'class_weight': {0: 0.5, 1: 0.5}, 'criterion': 'gini', 'max_depth': 25, 'n_estimators': 150} 0.7879354497354497
AUC =  0.7135118609739087
Accuracy =  0.6944655041698257
```

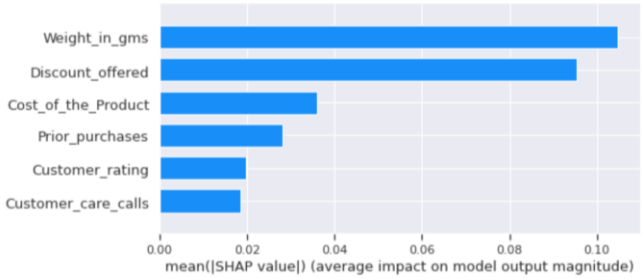
We can see from the confusion matrix that we are getting a lot of false negatives when we are predicting the test data. Also, the accuracy is about 0.694, which is a bit higher than what we had from the average accuracy of the training data.

XAI

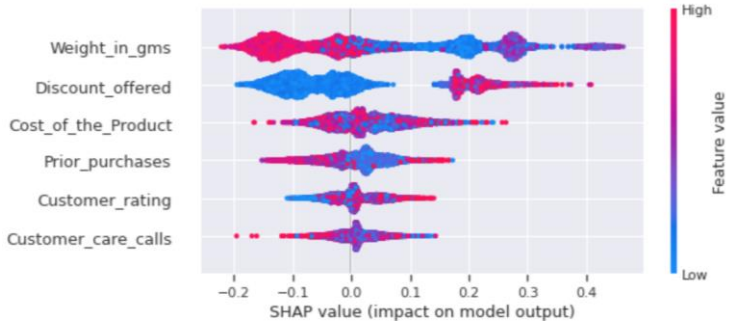
In order to understand how much each feature contributes to the model's prediction, we will make a few XAI plots. First, we will plot the feature importance plot with the random forest model, which is shown below:



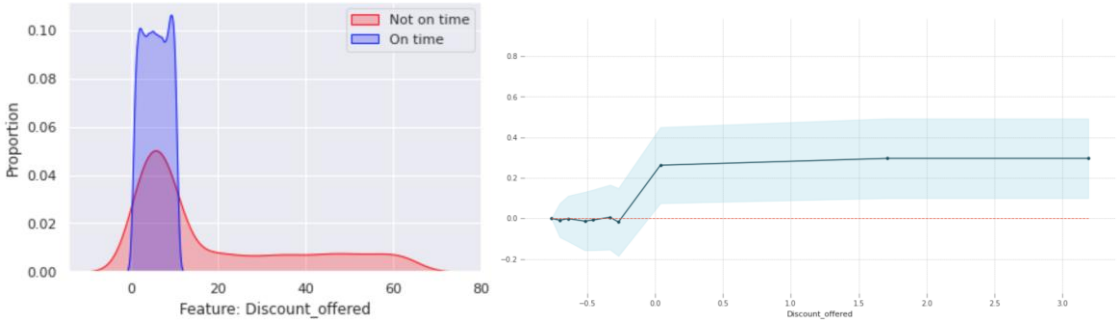
We can see that the product weight, discount offered on the product and cost of the product are the three most important features among the six features that we have selected. Then, we looked at the SHAP values to find out the impact of each feature on the model output:



We can see that weight and discount has more impact on the output than the other four features, and even though cost was an "important" feature, it did not affect the final output as much as the other two. After that, we also looked at the SHAP values for every feature on every sample:



We can see that all of the packages that arrive on time have a low discount value, and there are no significant findings on other attributes. The KDE plot and PDP plot below gives a clear indication of this finding:



Auto-ML

As the AUC and Accuracy reached by our selected classifier is not that satisfying, we also tried Auto-ML pipelines Auto-Sklearn and TPOT to look at their performances on the data.

The Auto-Sklearn reaches the AUC of 0.718 and the Accuracy of 0.693 using an ensemble model:

model_id	rank	ensemble_weight	type	cost	duration
26	1	0.10	random_forest	0.199313	7.003133
2	2	0.08	random_forest	0.199804	6.807065
8	3	0.18	extra_trees	0.203572	5.990951
24	4	0.08	gradient_boosting	0.213974	6.085298
30	5	0.06	gradient_boosting	0.216182	4.000762
28	6	0.02	extra_trees	0.305734	6.546726
42	7	0.32	lda	0.310905	1.654449
35	8	0.16	bernoulli_nb	0.485879	1.850235

The TPOT reaches the AUC of 0.691 and the Accuracy of 0.682 with the automatically generated model:

```
Pipeline(steps=[('randomforestclassifier',
                 RandomForestClassifier(bootstrap=False, criterion='entropy',
                                        max_features=0.8, min_samples_split=15,
                                        random_state=1))])
```

Both of the auto generated models have similar performance with our manually built classifier. The ensemble model generated by Auto-Sklearn outperforms a little among the three using AUC benchmark.

Conclusion

Our goal was to identify the best machine learning model and the most important features in predicting whether product shipment would be delivered on time. We chose AUC and accuracy to be our metric because we are more concerned about distinguishing on-time packages and not-on-time packages. We also preprocessed our data by correcting skewness and outliers, dummy encoding categorical variables and selecting important features from RFE.

After comparing different machine learning models with K-fold cross validation, we found that random forest fits the best with AUC of 0.733 and accuracy of 68.9%. Then, we optimized the hyperparameters by grid search and got AUC of 0.713 and accuracy of 69.4% on the test data set. Furthermore, through XAI analysis, we found that cost of the product, discounts offered on the product and the weight of the product are the three most important features in our prediction model. Discounts offered on the product and the weight of the product have significantly higher SHAP values than other variables, indicating that these two variables affect the outcome the most. We also found that all packages that arrive on time have a low discount, which can be elaborated more in future studies. To improve the accuracy, we suggest that we get a larger data set or more features for each data point.