DBA3803 Project 2: Group 24

Portfolio Optimization

Chloe Ng Ying Xuan (A0188827J)

Davin William (A0189019W)

Lim Fang Yi (A0183225J)

2) Introduction

The goal of this report is to build and evaluate an appropriate model to predict whether policyholders of the company are interested in purchasing vehicle insurance provided by the company. This is through weighing between the interpretability and the performance of the model in achieving our model's objectives.

This is a classification problem in a supervised-learning environment. The label, to classify is the response of the customer - whether potential customer (each observation / row) is interested in purchasing vehicle insurance , given the attributes (features) of the customer, such as gender, age, region, age of vehicle, whether the customer has damaged his or her vehicle before, his annual premium and number of days customer has been associated with the company.

Interpretability refers to the degree which a human can understand the cause of a decision while the performance in this case refers to how well the model is able to reliably predict the model's result.

The features (covariates_ include; id, Gender, Age, Region_Code, Vehicle Age, Vehicle_Damage. Annual Premium, and Vintage. The label is the Response of the customer.

2) Data Preprocessing

While we did not create any interaction terms or performed logarithmic or polynomial transformation of covariates, we did scale the numerical covariates.

Age (of customer) and Vintage (Number of days customer associated with the company) were scaled by the StandardScaler, which standardizes the feature (covariate) by subtracting each entry in the feature column by the mean of the feature column, then divide each entry by the standard deviation of the feature column. We transform them to approximate a standard normal distribution with mean = 0 and variance = 1.

Annual Premium was scaled by the MinMaxScaler, which subtracts each entry by the minimum value of the feature column, then divide by the range, which is the difference between the maximum and minimum values of the feature column. This preserves the original distribution of the annual premium covariates.

Meanwhile, one-hot-encoding approach (dummy encoding) was applied to categorical covariates (including Boolean, aka. Binary covariates), which were Gender, Region_Code, Vehicle_Age (which was binned), Vehicle_Damage). An alternative approach is to encode Vehicle_Age by OrdinalEncoder, which was only thought of at the time of writing after the model has been chosen and relative performance measured.

3) Area Under Curve (AUC)

The Area Under Curve (AUC) is the probability that the model ranks a random positive example more highly than a random negative example, in this case ranking an interested customer more highly than an uninterested customer. The receiver operating characteristic curve (ROC) is the representation of the performance of the binary classifier.

The plotting of the ROC is done by the calculation of True Positive Rate (TPR, ratio of positive observations correctly classified as positive by the classifier) and False Positive Rate (FPR, ratio of negative observations wrongly classified as positive), which are on the x-axis and y-axis respectively, with each point being a different threshold (a value in the range [0, 1]) and the resulting TPR and FPR in the given threshold value.

AUC is much more reliable and robust than using accuracy. Assume the test set a classifier fits on has labels dominated by 1 class, i.e. a skewed data set, it can have high accuracy by predicting all or absolutely majority of observations being the dominated class. But it will fail to accurately predict classes that are minority. This high accuracy fools the user that the classifier is good, but in actual fact very bad. AUC can also be a good performance measure as the plotting of ROC to get AUC considers all possible thresholds, which means that the quality of the model's prediction will not take into account what classification threshold is chosen. Furthermore, it measures how well these predictions are ranked instead of just the absolute value.

However, in this case as there is a wide disparity in the cost of the false negatives (i.e. company promoting to an uninterested customer) and false positives (i.e. company missing an interested customer), it may be more important to prioritize minimizing the false positives and hence AUC in this case would not be useful in this optimization. Thus, although the aforementioned issue does not affect the stability of the results, it should not be used in the comparison of models since the optimization of false positives and false negatives is prioritized.

The flaw of using AUC, happens when one ignores the overall ROC curve shape. Ideally, a good classifier should its ROC curve as far as possible from the straight line whose equation is: TPR (y) = FPR (x), which represents a purely random classifier. Yet, there might be an ROC curve of classifier that is not as far away as a random one but yet scoring high AUC, fooling the person interpreting the AUC value. In another example, a skewed data set with majority having 1 class may also score a high AUC score if the classifier purely predicts all observations to be of the dominant class.

Personally in the context of this case where our focus is to find out correctly who are the potential customers that are interested in vehicle insurance, the confusion matrix might be a more useful tool, by looking at number of observations correctly identified as positive, and find out the True Positive Rate (or sensitivity or recall).

4) Model Training

3 different classifiers were given: Logistic Regression, Decision Tree and Random Forest.
The approach to optimize each classifier was to perform Grid-Search of hyperparameters of each classifier and perform a 5-fold cross validation.

For all 3 models, we tuned class weights as the default settings gave both equal weightage to Response = 1 and Response = 0. Yet, the validation sets and test sets suggest that there is an imbalance of number of customers in the dataset between those interested and uninterested dataset. Grid Search with cross validation does suggest the use of ‘balanced’ to address the issue of imbalanced classes, replicating the Response = 1 class (which is rarer) until you have as many samples as in the larger class, Response = 0.

(i) **Logistic Regression**

This is the commonly used algorithm used for classification, since it can output a value corresponding to the probability of belonging to a given class (in this case, Response, i.e. whether the customer is interested in vehicle insurance or not).

It is an appropriate classifier in this case, since the Response label which we are trying to classify customers into, is a binary variable, by estimating each customer's probability of being interested in vehicle insurance.
Default threshold is 50%, so if the model predicts a probability higher than this threshold, it classifies the customer as interested (value 1 in 'Response' covariate. Otherwise, the customer is classified as uninterested.The equation that logistic regression produces is the log (odds of probability of customer being interested in vehicle insurance) as a weighted sum of input features (including Age, whether vehicle had damage in the past, gender etc).

The parameters we tuned are: (1) C - degree of regularization of the Logistic Regression Classifier, and (2) Class Weight.

Its confusion matrix managed to pick 341 True Positives and 15 False Negatives, giving it fairly good performance in detecting True Positives with a True Positive Rate = 341 / 341 + 15 = 96% on the validation set, indicating the Logistic Regression model's fairly good ability to pick up True Positives. Hence, we think the performance is good enough. We also compared the performance on the test set, which similarly was able to correctly classify the positives. With True Positive Rate = 1292 / 1292 + 72 = 94.7% on the test set, as well as their AUC scores only differing by less than 0.1, with SUC score on training set = 0.823, AUC score on the validation set = 0.817 and AUC score on test set = 0.819, there seems to be no clear evidence of overfitting on the training set due to the highly even AUC scores on all 3 sets.

(ii) **Classification Tree**

A Classification Tree is a subset of the Decision Tree algorithm which is capable of performing both classification and regression tasks, which also is an important component of the Random Forest algorithm which was what we used as well. Without hyperparameter tuning, Decision Tree makes very few assumptions of the training data. Without hyperparameter tuning to constrain the hyperparameters of Decision Tree, the Decision Tree will overfit the training data and poorly generalize on unseen data. Hence, the minimum leaf samples (that a node must have), the maximum number of features evaluated for splitting at each node and maximum depth of trees are restricted, to reduce the risk of overfitting by the Decision Tree by performing regularization. Other parameters that restrict the shape of the tree include: min_samples_split (the minimum number of samples a node must have before it can be split), min_weight_fraction_leaf (minimum number of samples a leaf node expressed as a fraction of the total number of weighted instances), max_leaf_nodes (the maximum number of nodes) We do not train all of them during the GridSearch cross validation process due to computing power constraints.

Unfortunately, the AUC scores under the Decision Tree model only returned 0.780 on the validation set and 0.795 on the test set which far underperformed the more complex Random Forest model and the Logistic Regression Model. This is a huge difference compared to the 0.82s of the Random Forest and Logistic Regression Models.

While regularization of the Decision Tree and hyperparameter tuning tried to reduce overfit, the confusion matrix on the validation and test sets seem to suggest overfit, with the poor AUC score consistency across Training, Validation and Test Set. AUC on training set = 0.745, AUC on validation set = 0.780. AUC on test set = 0.797. While AUC scores are still relatively consistent, they get outperformed consistently by Random Forest and Logistic Regression. Plus, the confusion matrix where no positives were correctly labelled in the validation and test sets seem to suggest Decision Tree's poor ability to generalize on unseen data, suggesting overfitting.

(iii) Random Forest

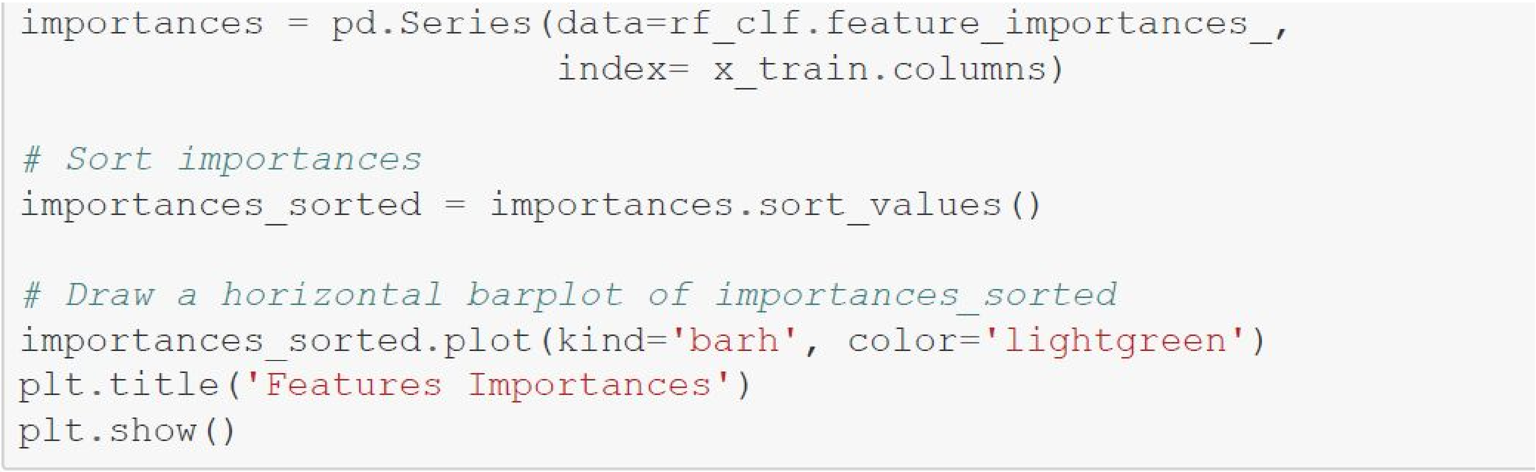
A Random Forest is an ensemble of Decision Trees, trained via the bagging method, by aggregating the predictions of multiple Decision Trees (classifier). This is done by training a group of Decision Tree Classifiers each on a different random subset of the training data, and make predictions by aggregating the predictions of each tree to predict the class (i.e. customer is interested in the automotive insurance) to predict the class with the most votes. Hence, the Random Forest is chosen as "a group of Classification Trees and aggregating their predictions" is often more reliable than having 1 classification tree. due to the property Similarly to Decision Tree, it shares the hyperparameters with Decision Tree and we also tune to optimize these hyperparameters i.e. minimum leaf samples (that a node must have), the maximum number of features evaluated for splitting at each node and maximum depth of trees are restricted.

The AUC scores for training set, validation set and test are: 0.864, 0.815 and 0.818 respectively, so the performance of random forest on validation and test sets generally match that of logistic regression.

Features Importance:

A useful quality of Random Forest is the ease of measuring the relative importance of each feature. By measuring the importance of a feature through calculating the weighted average of how much the tree nodes that use that particular feature, reduces impurity on average across all the trees in the forest. The following plot suggests that past history of Vehicle Damage, Age of customer, and vehicle age less than 1 year, vehicle age between 1 - 2 years are the 4 most important features, in the order of their importance.

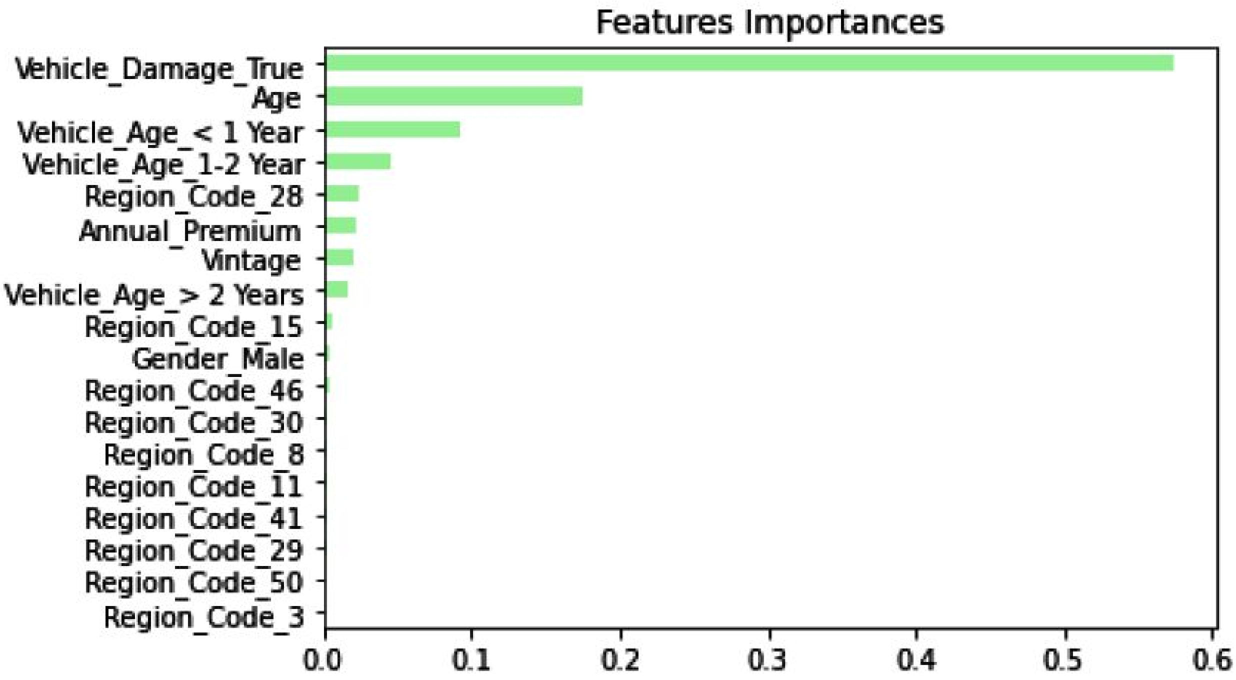
Intuitively they make a lot of sense as well. For instance, past history of vehicle damage by an owner will definitely make the owner want to buy vehicle insurance, as the cost of repairing vehicle damage or even replacing an entire vehicle is usually more costly than paying for the premium itself. So owners want to buy insurance to offset losses to themselves. Age is correlated to prudence which makes them likely to buy insurance, or alternatively explained by the relatively young age of a owner of a car maybe correlated to his inexperience in driving and higher likelihood of damaging his vehicle. Lastly, vehicle age makes sense as well, as a newer car which is in good condition and still holds good residual value is worth insuring by owners, who are keen to avoid damage to their car. And if they do, they want their losses to be minimal, be it loss in value of car or repairing cost. However, an older car in poor condition is associated with lower residual value, so owners are not as concerned about damages and repair costs as the cost of repairing might be beyond economical repair, or they are simply content with the damage. In this case, they are less likely to repair their car and less interested in vehicle insurance.



```
importances = pd.Series(data=rf_clf.feature_importances_,
                        index= x_train.columns)

# Sort importances
importances_sorted = importances.sort_values()

# Draw a horizontal barplot of importances_sorted
importances_sorted.plot(kind='barh', color='lightgreen')
plt.title('Features Importances')
plt.show()
```



5) Chosen model: Logistic Regression

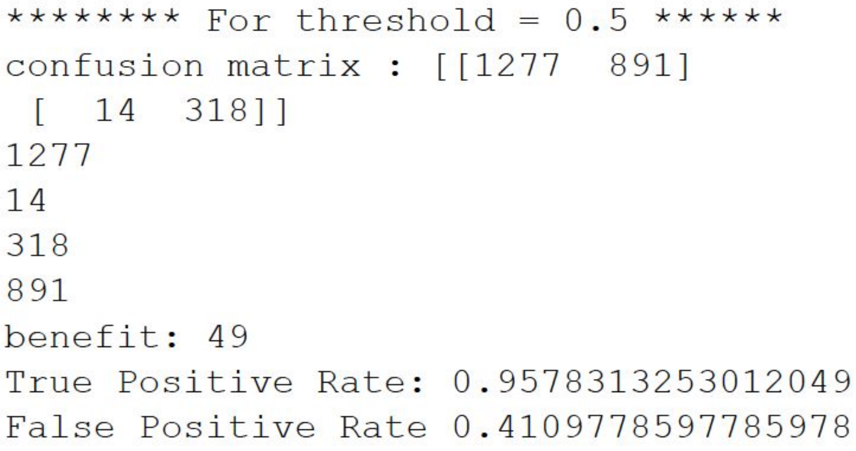
Firstly, the Decision Tree Classifier model is eliminated amongst the 3 for its worst AUC score on the training, validation and test sets amongst the 3 different classifiers. (Although the focus is to look at performance on validation set). This is further substantiated by looking at its Recall (i.e. True Positive Rate), which was in the 0.8s across training, validation and test sets, which were lower than the 0.9s of the Random Forests and the Logistic Regression.

Then there remains the comparison between Random Forest and Logistic Regression, which we eventually picked Logistic Regression. Firstly the AUC scores on validation sets of Random Forest and Logistic Regression were basically evenly matched at 0.815 and 0.817 respectively, and AUC scores on test sets on respective models were also evenly matched at 0.818 for both. Hence, we decided to look at the True Positive Rate since it is more importantly to pick the correct interested customers. The True Positive Rate for Logistic Regression slightly, in both validation and test sets, at 0.95 and 0.96 respectively. This consistently outperformed the True Positive Rate for Random Forest slightly across validation and test sets, which were at 0.93 and 0.91 respectively. Apart from its better performance, the interpretability of Logistic Regression is also better compared to Random Forest which is a sophisticated ensemble learner. As the number of observations is large as well, Logistic Regression is preferred as there are no overfitting issues.

6) Benefit structure

Benefit structure 1:

Promoted to an interested customer	+10
Miss an interested customer	-10
Promote to an uninterested customer	-2
Each promotion	-1



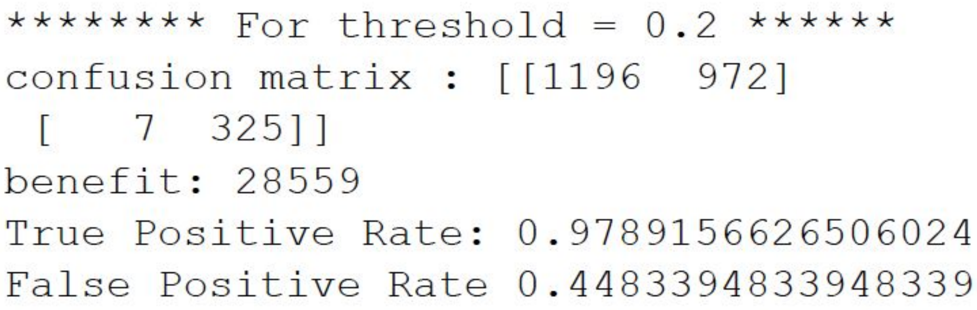
```
******** For threshold = 0.5 ******
confusion matrix : [[1277  891]
 [  14  318]]
1277
14
318
891
benefit: 49
True Positive Rate: 0.9578313253012049
False Positive Rate 0.4109778597785978
```

If p is the probability of promoting to an interested customer, a threshold should be determined such that we are able to conclude that policyholders with a probability higher than this threshold will purchase vehicle insurance. Our chosen probability threshold which gives us the maximum benefit, given the above benefit structure, is 50%. Since the profit on a true positive prediction (i.e. promoting to an interested company) and the loss of a false positive prediction (i.e. the company has missed an interested company) is the same, the loss of missing a customer would ultimately even out the profit of promoting to an interested customer. This means that there is a need to reduce the number of false positives so as to reduce the cancellation effect.

In benefit structure 1, both the false negatives (i.e. company promoting to an uninterested customer) and false positives (i.e. company missing an interested customer) are focused on since the cost of each promotion will not vary as the threshold changes. At the threshold of 50%, the number of FN and FP is the lowest which means that the cost is the lowest.

Benefit structure 2:

Promoted to an interested customer	+100
Miss an interested customer	-100
Promote to an uninterested customer	-2
Each promotion	-1



```
******** For threshold = 0.2 ******
confusion matrix : [[1196  972]
 [   7  325]]
benefit: 28559
True Positive Rate: 0.9789156626506024
False Positive Rate 0.4483394833948339
```

With the new benefit structure, we have chosen the probability threshold to be lower than from benefit structure 1, which is 20%. In this case, if the amount of profit from promoting to an interest customer and losses in missing an interested customer increases to 100, the number of false positives and false negatives again will need to be reduced so that the cancellation effect would be minimised. However, since now the disparity between the losses for FN and FP is larger, where the costs of FP is greater than FN (-100 versus -2), the reduction in FP would need to be prioritized. In addition, the results have shown that if the benefit is large enough, the proportion of cost for FN will now be less significant (compared to -10 versus -2) and thus the reduction in FPs will be prioritized more to achieve higher benefits. This means that a higher sensitivity model will be chosen which results in a lower threshold.

7) Combining Logistic Regression and Random Forest

We believe that the reason behind this question occurs when Logistic Regression and Random Forest gives different results. In this case, we can consider combining using an ensemble learning. A Voting Classifier is an ensemble that trains numerous models and predicts an output (class) based on their highest probability of chosen class as the output. By using the Voting Classifier, we can potentially have an improved performance in terms of AUC or F1 score and also less overfitting.

While combining logistic regression and random forest can improve performance, there is another problem where it can further harm the interpretability of the model since we are combining a linear based model and a tree based model. Thus, in a business context, even if there is a performance improvement, the lack of interpretability will likely not pass the review. Hence, combining both models is not a good idea.

However, if the combination of both models is referring to a hybrid algorithm, which is tuning the logistic regression model based on random forest variable importance, it would be a good idea for our case. This is since random forest measures the predictive value of attributes, and we are also able to obtain the ranking of importance from it. From this, we can extract information about any interaction effects in the data and this could be added to our logistic regression model to build an enhanced predictive model. Compared to Random Forest, Logistic Regression offers better interpretability even though it may have a lower performance. However, the lower performance from Logistic Regression can be improved with this approach, and it may ultimately outperform the ensemble method as aforementioned. This is since the Logistic Regression can now be further tuned by adding interaction terms optimally to improve its performance. Furthermore, as the model is not at risk of overfitting due to a large sample size, adding new features or any interaction terms will not be a problem here.

8) Conclusion

Overall, we chose Logistic Regression as a better model because it consistently outperformed in True Positive Rate compared to other models. On top of that, the interpretability of Logistic Regression and lack of overfitting (due to large sample size) are also the reasons for our choosing. Initially, we already eliminated the Decision Tree Classifier due to poor performance. While considering between Random Forest and Logistic Regression, we believe there is a potential improvement if we do a combination of these models. However, due to the lack of interpretability, we believe that using our chosen model, Logistic Regression, is best for this project.