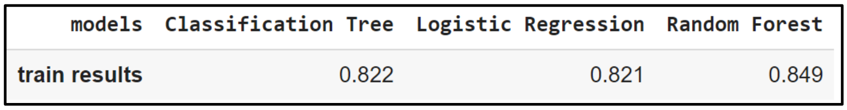
Table of Content

Technical Requirements

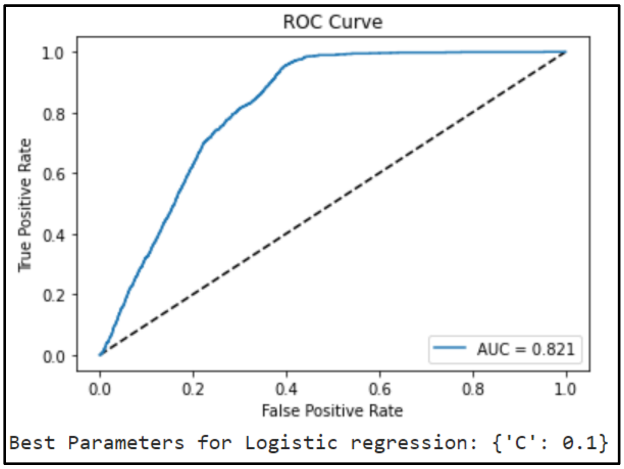
1. Logistic Regression

1.1 Model Performance



models	Classification Tree	Logistic Regression	Random Forest
train results	0.822	0.821	0.849

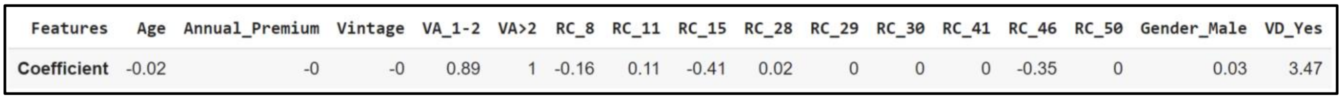
Based on the overall AUC performances by the three models, the logistics regression model seemed to have performed decently well. Despite being a relatively simple model, it can achieve a result of about 82% which is comparable to even the complex model such as random forest. Furthermore, even though a payoff between performance and interpretability is present within logistic regression, this model tends to be very interpretable with intuitive features as well as performs at high accuracy.



```
Best Parameters for Logistic regression: {'C': 0.1}
```

Moreover, the ROC curve shows that the model provides a high sensitivity towards the dataset. In other words, the used model is inclined to correctly classify labels with a higher probability, as the model tends to disproportionally have a higher True Positive Rate over False Positive Rate. Hence, the AUC score of the proposed Logistics Regression model is larger than 0.5 (0.821), concluding that it significantly outperforms a random guess (which is shown by the dotted line). Therefore, it can be said that it does not perform perfectly, but given its degree of generalization and sensitivity, it performs good enough to be considered a production model.

1.2 Model Coefficients



Features	Age	Annual_Premium	Vintage	VA_1-2	VA>2	RC_8	RC_11	RC_15	RC_28	RC_29	RC_30	RC_41	RC_46	RC_50	Gender_Male	VD_Yes
Coefficient	-0.02	-0	-0	0.89	1	-0.16	0.11	-0.41	0.02	0	0	0	-0.35	0	0.03	3.47

By regularizing the coefficients of the logistics regression by using the L1 loss and setting C as 0.1, we can benefit from feature selection. In other words, coefficients that tend to be less important in impacting the label are set to zero or close to it.

The sign of the coefficients determines how the feature is impacting the label. For example, Age has a negative sign, which can be interpreted as, if everything else is held constant, a higher age may not lead to an individual taking vehicle insurance. This is very intuitive, as one person who is old may not be able to drive a vehicle anymore and as such, not insure their vehicle. Whereas if one looks at the most significant factor, which is "VD_Yes" (Vehicle is damaged) and has a positive sign, the opposite interpretation is applicable. When all other features are held constant, owners will tend to insure their car if a vehicle has been damaged in the past. This is also very straightforward because if the owner had an experience of his vehicle being damaged in the past, he would have realised how important it is to insure their vehicle.

2. Classification Tree

2.1 Hyperparameters of Tree

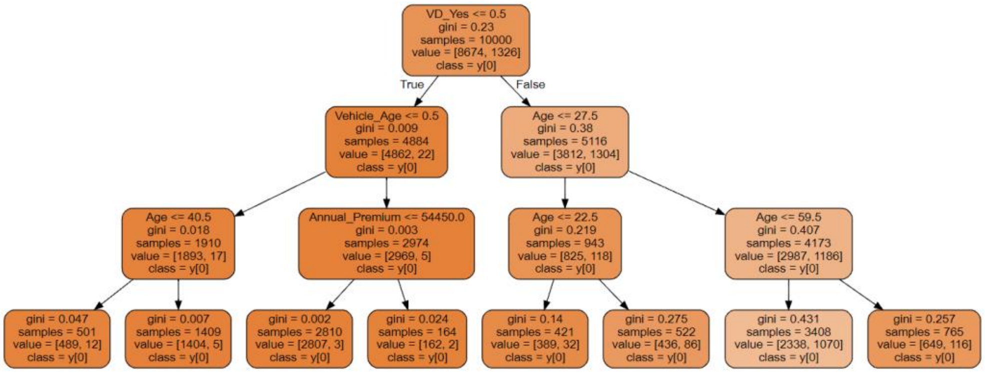
The hyperparameters of the classification tree are maximum depth and the minimum number of samples per leaf.

2.2 Explanation of the classification tree result

The first parameter that splits the data is the “VD_Yes” (Vehicle is damaged) variable which suggests that this feature brings about the most significant reduction in variance and thus maximises loss reduction. This aligns with the model coefficients found in our logistic regression where Vehicle_Damage was the most significant factor. This is followed by Vehicle_Age and Age, which are also among the most important features in determining whether the customer would purchase the insurance.

The identified features make intuitive sense because out of the 10,000 samples, 8674 of them would not have purchased insurance if their vehicles were not damaged in the past, as they would not have realised its importance. Moving to the second layer, if a vehicle has not been damaged in the past and the owner is younger than 27.5 years old, a majority of them are still unlikely to purchase insurance. This interpretation will move on all the way until it reaches the leaf node, with other important features being considered before a prediction is made for a particular group of customers with certain characteristics.

2.3 Classification Tree Plot



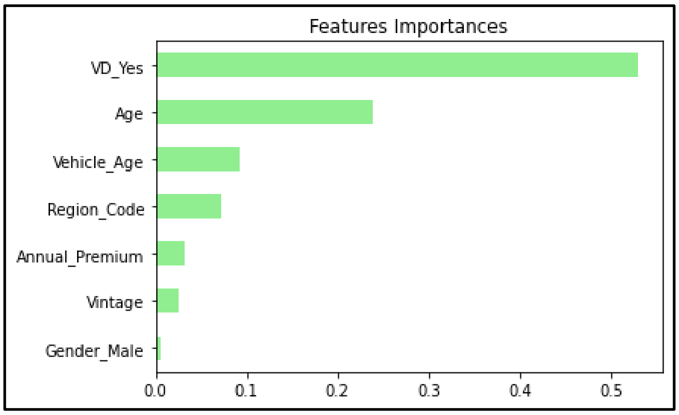
Best Parameters for Classification Tree: {'max_depth': 3, 'min_samples_leaf': 20}

3. Random Forest

3.1 Hyperparameters of Random Forest

The hyperparameters are the **maximum depth**, the **minimum number of samples per leaf**, and the **number of estimators** used in the random forest. The maximum depth and the minimum number of samples plays into the idea of a bias-variance trade-off. The lesser depth and more samples per leaf reduce the variance since there are fewer split points during classification. However, since there are fewer splits, the prediction's accuracy might be less, increasing bias. The number of estimators plays more into the trade-off between accuracy and time taken by the model. More estimators help to improve the accuracy of the model but increase the time complexity.

3.2 Feature Importance



Based on the feature importance plot, we can see that the most important features are **(1) whether the vehicle has been damaged in the past** and **(2) the owner's age**. This makes intuitive sense since if the vehicle has been damaged before, the individual is more likely to be aware of the possibility of accidents and be more willing to buy vehicle insurance. In addition, the customer's age is also an intuitive factor since those who are much older likely drive less and feel less of a need for vehicle insurance. In fact, these features that were being highlighted as important are consistent with the ones that have been mentioned by the simpler model earlier on.

3.3 Potential pitfalls on categorical variables

Region Code has 10 categorical values. With so many categories, there are many ways to split the data, which may result in overfitting. Random Forest mitigates this since it takes a subset of covariates during the tree construction. Hence, there is less concern as Random Forest is less likely to overfit despite many categories.

Another issue with categorical values arises from encoding. Python cannot handle categorical values and uses dummy variables instead. This limits the importance of each individual category. Hence, we sum all the dummy variables up to estimate the total feature importance of the Region Code.

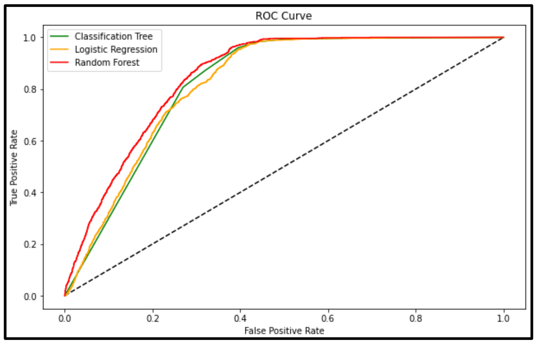
3.4 Reason for this hard-to-explain model

Despite being less interpretable, Random Forest models are less likely to overfit than Logistic Regressions and Simple Classification Trees. This is because a random forest creates a lot of randomness during the creation of a model. For example, it makes multiple trees from bootstrapped data and performs random selections of certain features each time it creates a tree. As a result, Random Forest tends to perform better in the test set than the other two, and by extension, is likely better for actual use. Moreover, based on *CV AUC Results above*, we can see that random forest has the best performance compared to the other two models.

3.5 Hyperparameter Tuning on Random Forest

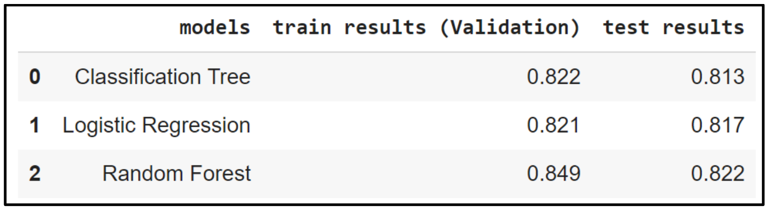
We found that the **optimal maximum depth is 5**, the **best minimum samples per leaf is 5,** and the **best number of estimators in the random forest is 300**.

4. Chosen model: Random Forest



After training the three models with GridSearchCV, we can obtain the validation result for the respective models. Based on the validation result, our team has decided to choose **Random Forest**, which has the highest AUC validation score at 0.849. Although this model is more complex than the other two models, we can still understand the important features through the features importance plot.

4.1 Performance on Test set and validation set



	models	train results (Validation)	test results
0	Classification Tree	0.822	0.813
1	Logistic Regression	0.821	0.817
2	Random Forest	0.849	0.822

To derive further insights on whether the model has any overfitting issue, we decided to test on all three models instead of just one (chosen model) which we would normally do. With that, we test all three models on the test set (Unseen Data). From the result above, we can see that all three models have obtained a test score that is relatively similar to their validation set. This could mean that there is not much overfitting in the three models. Additionally, our chosen model (Random Forest) has performed the best out of the three models with a score of 0.822.

From the table, we can also see that the test set performance is worse than that of the validation set. This is not surprising because, based on intuition, ***Validation Performance = Test Performance + Randomness***. When tuning the model's hyperparameters, we pick the best model based on the validation set performance. This is reminiscent of the skill of the prediction. The test set should be similar to the validation set which should result in similar performances. However, due to randomness, there are often slight differences between the two sets, attributed to the luck aspect. Hence, the model fitted to the validation set often does not produce the exact same degree of accuracy on the test set.

Discussion Questions

5. Standardization of data for logistic regression

For logistics regression, it is not necessary to standardize the input values, as this model is not sensitive to the magnitude of the data. Unlike a linear regression, the sigmoid function of a logistic regression does not consider the range of a specific feature. Instead, the range of value is used to determine the probability in the logistic regression model. For example, in linear regression, the model takes on the form of ***[y = B0 + B1*x + …]*** and the coefficient of the model would directly impact the predicted value. However, in logistic regression, the model takes on the form of ***[y = e^(B0 + B1*x + ...) / (1 + e^(B0 + B1*x + ...))]*** where the output value being modeled is a binary value (0 or 1) rather than a numeric value. As a result, the large magnitude of certain features would not have a huge impact on the performance and as such standardization of the data is not required.

6. Is AUC a good performance measure?

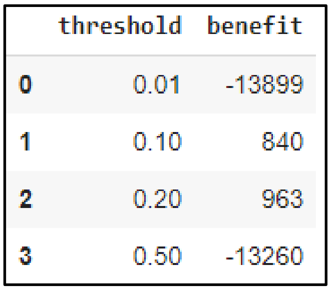
No, to a certain extent. To determine whether AUC is a good performance measure for the business questions, we would have to determine which metric has a more severe consequence.

Given that AUC refers to the area under the ROC curve with the y axis as the **True Positive Rate (Recall)**, TP / (TP + FN), and the x axis as the **False Positive Rate**, 1 - (TN / TN + FP), this would mean that this performance measure is placing more emphasis in two particular areas. Firstly, it is focused on getting the correct prediction out of all the actual positive outcomes (*True Positive Rate*). Secondly, this performance metric emphasises on reducing the number of times that the model predicts positive when the actual is negative *(False Positive Rate)*. As such, AUC will only be a good performance measure if the business question is focusing on improving True Positive Rate and reducing False Positive Rate.

However, in this project, based on the benefit structure, we can see that **False Negative** has a larger implication than the **False Positive** due to the larger penalty that was given. As such, having a low **False Positive Rate** is not as important as compared to **False Negative**. Therefore, using AUC to evaluate the performance in this case may not be the most ideal choice.

On the other hand, AUC does have its advantages. For example, AUC is able to **summarise the result or performance of different models into one single number**. This would make it easier to compare the different models' performances. Moreover, AUC also provides a more intuitive understanding on measurement of likelihood. This is because it tells us the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance.

7. Probability Thresholds & Benefit Value based on Random Forest



	threshold	benefit
0	0.01	-13899
1	0.10	840
2	0.20	963
3	0.50	-13260

Old Benefit Structure

Promote to an interested customer + 10 (TP)

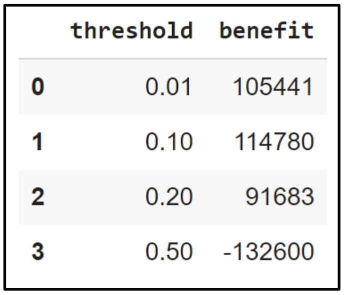
Miss an interested customer - 10 (FN)

Promote to an uninterested customer - 2 (FP)

Each promotion - 1

The table above shows the result of the benefit value with the various threshold amounts based on our chosen model, random forest. From this table, we can see that the threshold that provides the most payoffs would be at 20% which is a relatively low threshold. This is because the penalty for False Positive (-2) is relatively low. As such, it would be wiser to classify more customers as interested so as to avoid the heavier penalty of False Negative (-10).

8. Probability Thresholds & Benefit Value with new Benefit Structure



	threshold	benefit
0	0.01	105441
1	0.10	114780
2	0.20	91683
3	0.50	-132600

New Benefit Structure

Promote to an interested customer + 100 (TP)

Miss an interested customer - 100 (FN)

Promote to an uninterested customer - 2 (FP)

Each promotion - 1

With the new benefit structure where the TP and FN value is increased by 10 times, this would mean that it would be **extremely beneficial to have correct prediction and extremely costly to have any False Negative**. As such, it is not surprising to see that the threshold has fallen from 20% to 10% which implies a more lenient approach in classifying customers as interested. This is because it is now more advantageous to classify more customers as interested since the penalty for missing out on interested customers is now heavier (10 times more as compared to the old benefit structure at -100) as compared to promoting to an uninterested customer which only has a penalty of -2.

9. Combination of logistic regression and the hard-to-explain model

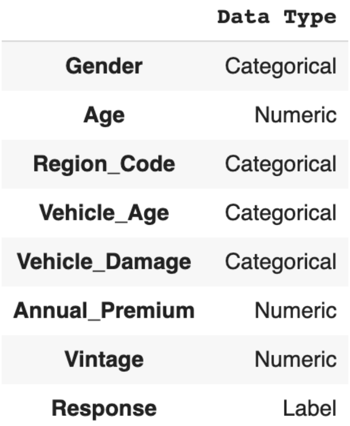
Combining logistic regression and the hard-to-explain model, which in this case refers to random forest, would not be necessary due to two reasons.

Firstly, the AUC result of these two different models is pretty decent where logistic regression has attained an AUC result of 0.821 while random forest has attained an AUC result of 0.848. In addition, based on the feature importance plot and the feature coefficient derived from logistic regression, both models have highlighted similar features as important such as the Vehicle_Damage and Vehicle_Age.

Secondly, combining the two models together will increase the complexity of the models and it would be more computationally expensive to yield the result. Given that this project **requires a balance of between interpretability and performance**, the increased complexity will make it harder for the managers to interpret and thus, a lower chance of passing the review.

Therefore, given that both logistic regression and random forest have already proven its ability to provide a certain level of performance of about 0.8, we would not suggest combining these two models that would risk the interpretability of the model.

10. Dummy Variable Trap



	Data Type
Gender	Categorical
Age	Numeric
Region_Code	Categorical
Vehicle_Age	Categorical
Vehicle_Damage	Categorical
Annual_Premium	Numeric
Vintage	Numeric
Response	Label

Using the python pycaret package, we are able to infer the data into categorical and continuous variables. The categorical variables can be further subset into nominal (without order) and ordinal (with order). Depending on which categorical subgroup the feature belongs to as well as which model is used, data preprocessing has to be adjusted accordingly. For example, for logistic regression, it does not matter whether the ordinal categorical variables (Vehicle_Age) has a natural order implied with it, whereas for the tree structure it does. Therefore, we had to do integer encoding (allow natural order) for tree structures and dummy encoding (Split into N feature number of columns) for logistic regression.

However, there is something essential one has to take care of to prevent multicollinearity or correlated comovement between features when creating dummy variables. This is called the dummy variable trap. If one creates dummy variables for each N number of features one creates a relationship between the features which naturally does not exist ($\sum X_i = 1$ where i = feature category). As a result, the model loses predictive power. To prevent introducing an unnatural correlation between the features, we dropped one column for each dummy encoded feature to break a possible artificial relationship between them.