Student Name	*Nathan Collins*
Project Name	*MLAA Assignment 2*
Date	*Apr 28 by 23:59*
Deliverables	*<MLAA_Assignment_2_4>* *<Experiment_Report_4>*

EXPERIMENT REPORT **[D]**

EXPERIMENT BACKGROUND

a. Business Objective

The overarching aim of the project is to utilise analytical and statistical methods to determine the likelihood that an existing customer of an automotive manufacturer will purchase a new vehicle. To achieve this, several experiments will be conducted using a provided historical dataset. The success of the model will empower business stakeholders to implement a cost-effective re-purchase campaign targeting existing customers who are prospective in purchasing a new or secondary vehicle.

The accuracy of the results will determine the value of the leads attained following the marketing campaign. Precise results may yield a positive return on investment by identifying these fulfilling customer leads, while incorrect results could lead to revenue loss through an unsuccessful marketing campaign, overstocking incorrect car varieties (models and segments) in anticipation of certain buyers, or granting unnecessary warranty and servicing inclusions with purchases, enacting a toll on the business, and damaging its longevity.

b. Hypothesis

Alternative hypothesis:

There is a relationship between some features in the car sales dataset and the likelihood that existing customers of an automotive manufacturer will purchase a new vehicle.

Null hypothesis:

There is no relationship between the features in the car sales dataset and the likelihood that existing customers of an automotive manufacturer will purchase a new vehicle.

The null hypothesis assumes no relationship is present between the dataset features, while the alternative hypothesis suggests that there is a relationship. By comparing the model's performance against the null hypothesis, it determines whether the model is significantly better than random chance at predicting customer purchase behaviour.

Investigating these questions will assist with accomplishing the business goal by determining which existing customers who have purchased a second vehicle share certain features.

c. Experiment Objective

Experiment 4. will consist of a **Random Forest Classification** model. This model was selected as it offers immense versatility in creating decision trees and flexibility when tuning the associated hyperparameters. As previous Experiment 3 has yielded a strong outcome regarding predictability, it is presumed more parameter flexibility may help increase the recall performance metric and meet the business objective. The model will examine the **Target** variable (which determines if a customer has purchased more than 1 vehicle) against all other variables independently, each found in two separate data frames:

Table 1 Data frames applied for analysis and modelling.

Data Frame	Feature
cars_ALL	"*all features*" 18289 entries, 42 features
cars_NAG	"*no age-gender*" 128611 entries, 34 features

Working with two data frames facilitates a broader scope of outcome possibilities while retaining as many variables as possible without resorting to oversampling. As features such as **age_band** and **gender** offer decisive analytical data and insights into the classification of certain population groups, they will be retained where possible.

For this experiment to be deemed a success, it is anticipated that precision and recall values will result in high coefficients, close to a value of 1. If a low or no depicted coefficient value is produced, it will pave the way for further experimentation with more complex algorithms.

These features will undergo further classification through evaluating model output coefficients to determine **feature importance**, where key traits about existing customers can be utilised to determine if they have purchased a second vehicle, achieving the business objective.

EXPERIMENT DETAILS

a. Data Preparation

< Data preparation method unchanged from Experiment 1. >

Data Understanding

The data frame was explored to derive foundational insights, indicating the initial seventeen features, two of which contain NaN values (**age_band** and **gender**) and four consisting of object variables (**age_band**, **gender**, **car_model** and **car_segment**) which will need to be transformed. All remaining features are full and are integer values.

Data Cleaning

In order to prepare the data for analysis and modelling, the ID column was removed to reveal 2726 duplicate entries. As duplicate entries can weigh certain outcomes and impede data accuracy and consistency, these were dropped.

Next, the NaN values were totalled: 109668 in **age_band** (85.2% of total rows) and 67455 in **gender** (52.4% of total rows) and visualised with **missingno**. As age and gender offer alternative means of classification over the dataset, aside from car-related data, it was decided to retain these features in a separate yet more condensed dataset.

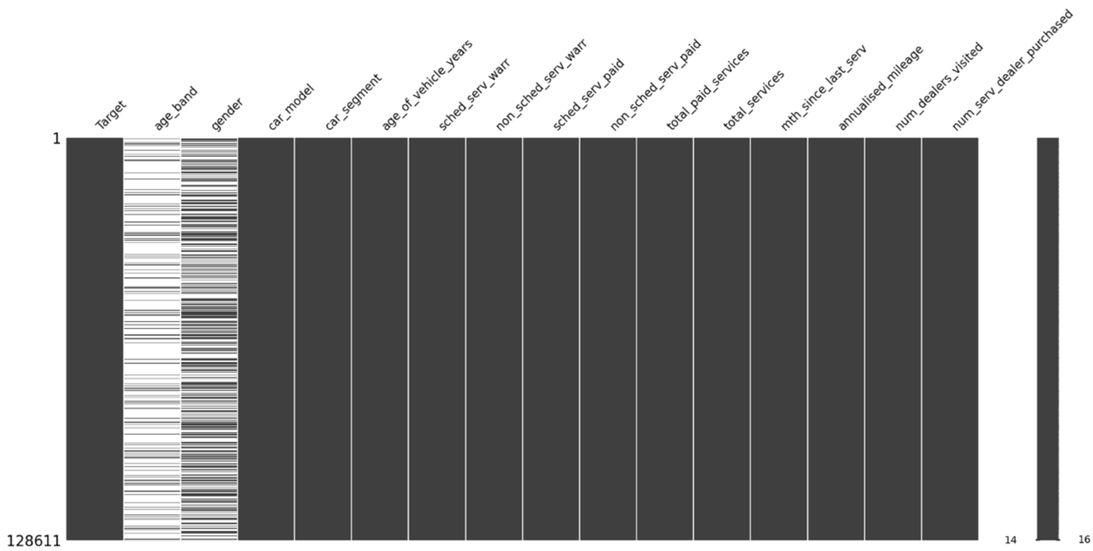


Figure 1 NaN values visualised with Missingno.

Producing Two Data Frames with All Values Converted to Integers:

cars_ALL and **cars_NAG** (see Table 1.)

For cars_ALL:

The **age_band** distribution was visualised prior to conducting **one-hot-encoding**, converting the feature into six rows with integer values of 1 and 0.

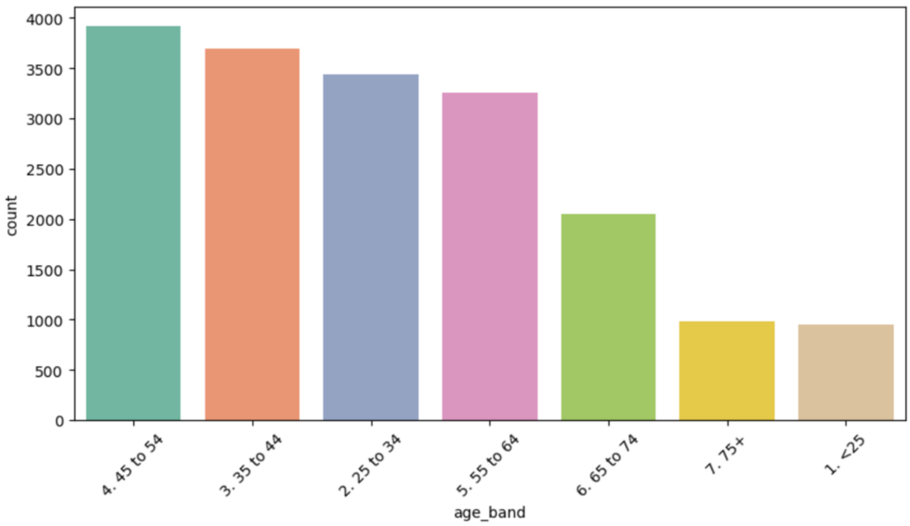


Figure 2 The age_band feature visualised.

One-hot-encoding was subsequently performed on the **gender** feature, converting the variable into two rows with integer values of 1 and 0, male and female.

For cars_ALL and cars_NAG:

The **car_segment** distribution was also visualised across both datasets prior to **one-hot-encoding**, converting the feature into four separate features with integer values of 1 and 0, Small/Medium, Large/SUV, LCV and Other

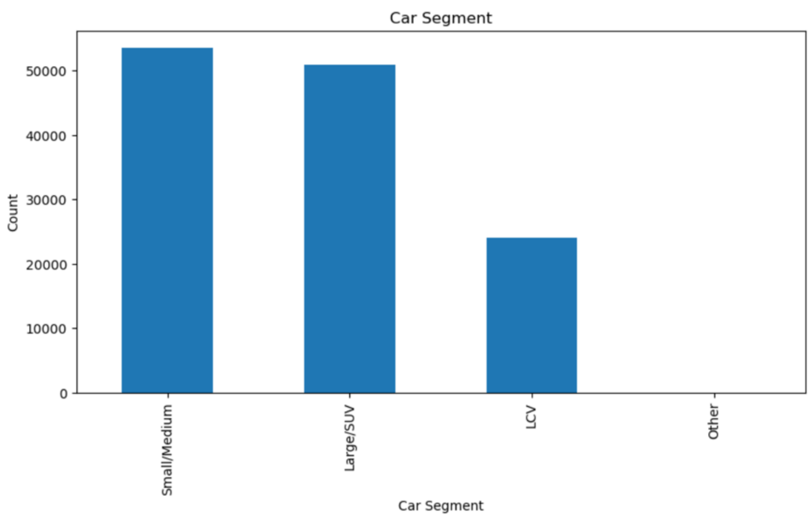


Figure 3 The car_segment feature of the cars_NAG data frame, visualised.

The **car_model** feature was the last categorical variable converted to an integer feature through **one-hot-encoding**, yielding nineteen features representing different car models with integer values of 1 and 0.

The ranges of each dataset were visualised as a final measure to ensure no outliers or abstract inputs were present.

Exploratory Data Analysis

Despite the problem residing in binary classification, linear correlations of all features against all other features of the dataset were first visualised using a heatmap, where the top 10 correlating features for each dataset were collated in a table.

	cars_ALL	correlation_ALL	cars_NAG	correlation_NAG
0	Target	1.000000	Target	1.000000
1	Male	0.033980	Large/SUV	0.015211
2	Large/SUV	0.027986	LCV	0.010342
3	4. 45 to 54	0.016244	car_model	0.000575
4	5. 55 to 64	0.013153	Other	-0.001319
5	car_model	0.008100	Small/Medium	-0.023228
6	LCV	0.004723	non_sched_serv_paid	-0.033297
7	7. 75+	-0.001011	num_dealers_visited	-0.053589
8	Other	-0.001083	num_serv_dealer_purchased	-0.058963
9	3. 35 to 44	-0.004205	annualised_mileage	-0.080251
10	2. 25 to 34	-0.007470	non_sched_serv_warr	-0.088442

Figure 4 The cars_ALL dataset and the cars_NAG dataset highest correlating features.

Next, the percentage of the **Target** variable class of interest (integer values of 1) was visualised with a pie graph. Indicating that only 2.7% of the 128611 entries were the population group of interest. This interpretation establishes the dataset as unbalanced.

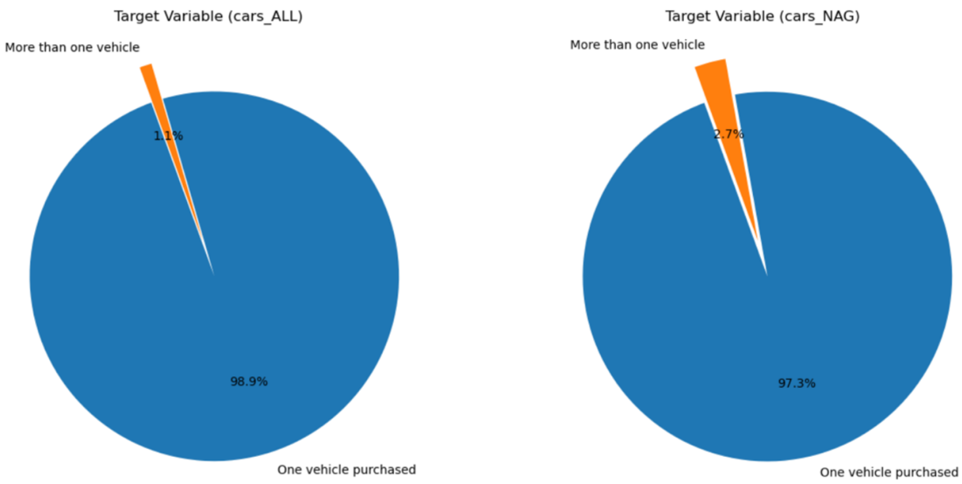


Figure 5 Pie graph visualisations of the Target variable and the class of interest.

Further visuals were subsequently constructed to gauge the perceived influence of the **gender** and **age_band** variables against other key classification features, such as the total, which share the **Target** variable and the relative **car_segment** and **car_model** choices made with their second car purchases. Overall it was interpreted that males tended to purchase a second car more than females, with the largest age brackets resting between 45 to 54. Out of these groups, the highest interest resided with Large/SUVs, models 5 and 3, among males. It is important to note that these features represent marginal percentages of the total and are not a full representation of the target cohort.

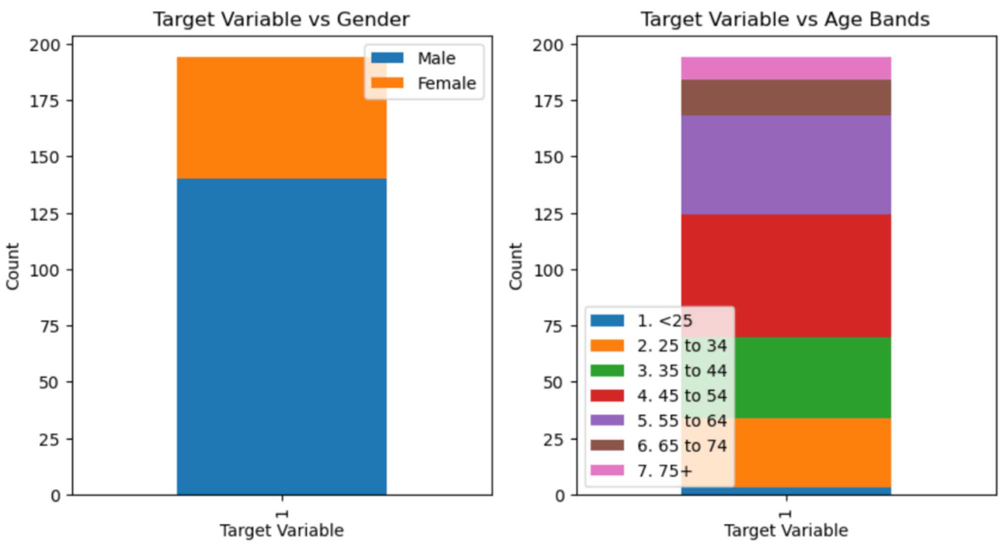


Figure 6 A segmented bar graph illustrating the male and female cohorts, beside the age band cohorts, with a Target value of 1.

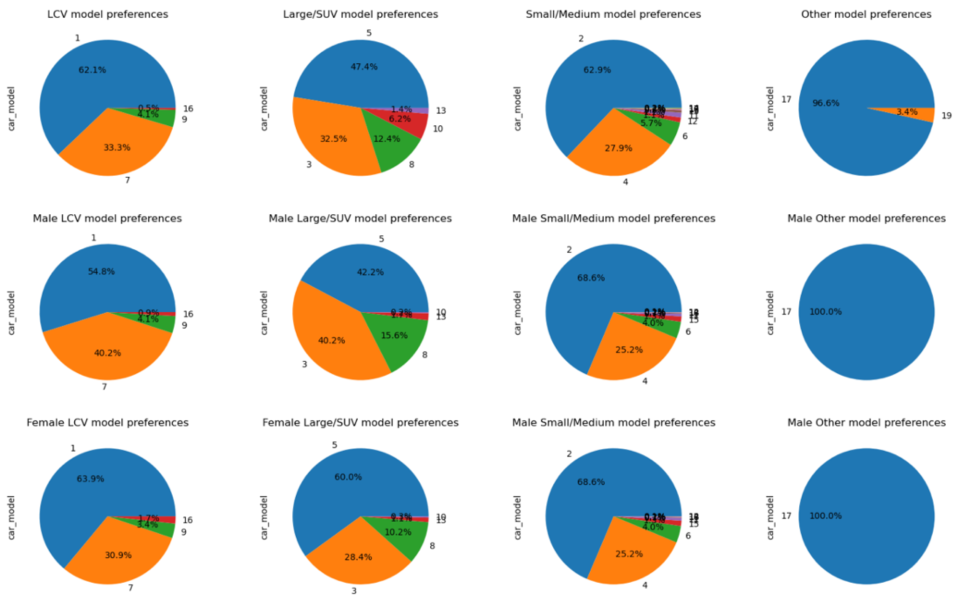


Figure 7 A series of pie charts illustrating the popularity of car models against the variety of car segments. The first row covers the overall majority, while the second and third rows cover the male and female preferences.

The final key exploratory analysis involved charting the frequencies of the **Target** variable with a value of 1 against **total_services**, **annualised_mileage**, **age_of_vehicle_years**, and **non_sched_serv_war.** The value was to visualise whether specific features of a customer's existing had an influence on their decision to purchase a new one. A key insight derived from this is that recurring customers typically engage with 2 to 3 total services (deciles), with slightly more non-scheduled services.

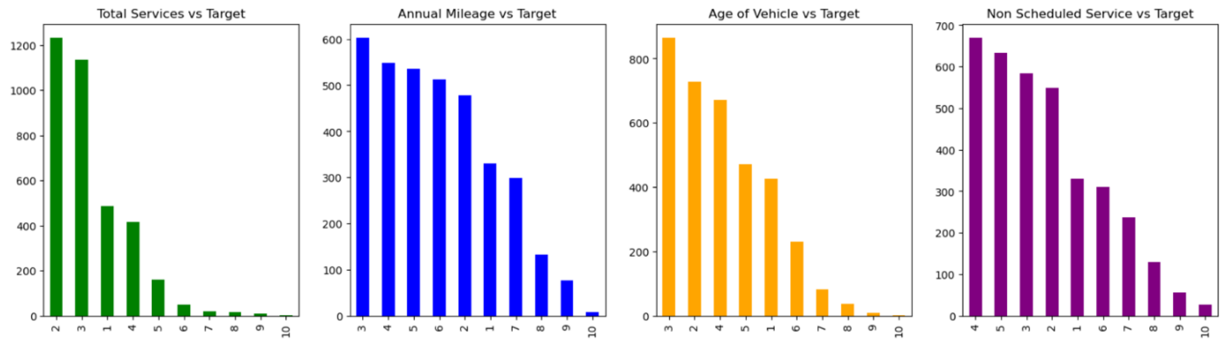


Figure 8 Four bar graphs illustrating total_services, annualised_mileage, age_of_vehicle_years, and non_sched_serv_war in deciles, against the target variable with a value of 1.

b. Feature Engineering

*< Aside from applying one-hot-encoding on **age_band**, **gender**, **car_model** and **car_segment**, no further feature engineering was conducted for Experiment 3. >*

c. Modelling

Selecting a performance metric

As the business objective still resides with a binary classification problem, **Precision** remains the key performance metric to determine the model's success. Precision examines the proportion of true positives among the total positive predictions. By applying precision, a reduction in predicting the wrong customer is achieved. As precision metrics may share output similarity across models, an additional metric, **Recall**, will also be evaluated. Recall examines the proportion of true positives among all actual positive outcomes, meaning higher recall results in more lost opportunities for the marketing campaign.

Moving forward, a **precision-recall curve** will be constructed with each model to act as a visual depiction of the model's performance. In a successful model, the concavity of the curve determines the overall accuracy of results. In addition, an F1 and accuracy score will be shown alongside a confusion matrix. While excessive with verifying the accuracy, these steps feel important, as publishing incorrect data may enact a toll on the business. The variation in reporting performance metrics facilitates a deeper understanding of the model's predictive capability and how it compares with future experiments

Establishing the model:

All models across all experiments, will be constructed through a Python function. This helps establish consistency across all metrics within the dataset and throughout the transformation of the data prior to and during machine learning. The subsequent model selected for this stage of the experiment is intended to expand upon Experiment 1's insights and tangent away from a simple model.

Random Forest Classification

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}}$$

Complexity: Intermediate

< The completed function marginally differs from Experiment 2 & 3.>

Prior to constructing the function, both data frames (cars_ALL and cars_NAG) were first split at a **random_state** of 100 into a training and testing set with embedded stratification (operating under the conventional **80% training / 20% testing** approach).

A function for the support vector machine was then defined as **ML_results**, where subsequent iterations of the function could interchange the **dataset**, **feature**, **target,** and **model** variables. The model variable, in this case, is **RandomForestClassifier()**. The function would first normalise the data first through **StandardScaler()**, which subtracts the mean from each data point and divides it by the standard deviation. The following enables data points with different units of measurement to be compared against one another and is essential prior to running a multivariate model. The output variables from this process will produce **X_train_s** and **X_test_s**.

The function then instantiates and fits the model to the training data (**.fit**) and creates predictions on the training and test data (**.predict**).

To gauge the model's performance, the function will print a confusion matrix, followed by the accuracy score of both training and testing sets, followed by an F1 score of both training and testing sets (**confusion_matrix, accuracy_score, f1_score).**

The last aspect of the function is establishing a Boolean with .**predict_proba** to plot a precision-recall curve (**.precision_recall_curve**). The precision results extracted to construct the curve will also be embedded in a Pandas data frame displayed below, presenting all precision scores of **0.75** or **greater**.

No baseline metric (for assessing null accuracy) was applied to compare performance against naïve predictions and determine whether the model is adding value. This was selected, as the majority of the data in the **Target** cohort is negative, making the null accuracy equal to the proportion of negative samples.

Hyperparameter tuning:

Hyperparameter tuning comprises determining the most fitting set of hyperparameters for a model. These parameters are not learned from the data and are required to be set by the user prior to training a model. Like previous experiments, a grid approach (**.GridSearchCV**) will be applied to a hyperparameter dictionary (**hyperparmeters_dict**). The dictionary will be embedded into a different function with the same functionality as **ML_results**, although renamed to **ML_results_cv**. Should hyperparameters need to be tuned, the function **ML_results_cv** will be applied in conjunction with separate hyperparameter dictionaries. The corresponding dictionaries used are present in the results section of this report.

These finalised functions were first applied to the **cars_ALL** and **cars_NAG** data frames where appropriate, defining the **dataset, feature**, **Target** and **model** with each iteration.

Models to consider for future experiments:

XGBoosting Classification

- Compounds simpler models to build a larger analysis.
- Capable of accounting for missing data.
- Requires additional data preparation and parameter tuning than previous algorithms.

EXPERIMENT RESULTS

a. Technical Performance

Evaluation of precision

In general, a high precision score indicates the model has a low rate of false positives, while a high recall score indicates that the model has a low rate of false negatives. To meet the business objective, successful model outcomes should aim to achieve both precision and recall scores close to 1.

Technical evaluation

The first Random Forest classification model was performed on the cars_ALL dataset without prior hyperparameter tuning, resulting in a very strong precision but only a moderate recall. The **precision** score of 0.95 means that out of all the instances predicted as positive, 95% were positive, while the remaining 5% were false positives. While the precision is strong, there is room for improvement.

However with a **recall** score of 0.54, describing that out of all the instances that were actually positive, only 54% were correctly identified by the model, with the remaining 46% being false negatives. While the model is a strong predictive tool, with a mediocre recall score like this, the model is sadly not an improvement on the previous experiment and ultimately is not ideal for classifying the cars_ALL dataset.

The precision-recall curve shape for this dataset still lacks concavity and uniformity, indicating “noise” is apparent within current features, a lack of data to represent the entire population, and the possible need to tune hyperparameters.

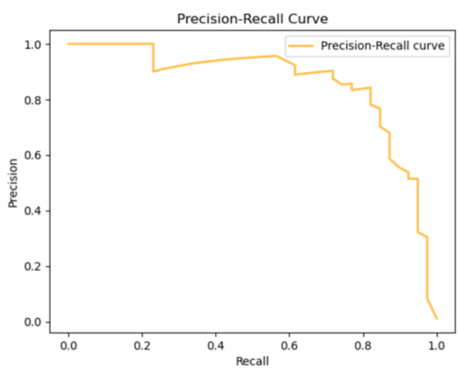
From a business perspective, the model is not considered adequate to accurately meet the business objective in forecasting the features that relate to the target class of prospective buyers. This verdict, however, has improved from the last experiment.

Performance Metrics for Random Forest Classification 1.
Dataset: cars_ALL

Confusion Matrix Training Set	
14476	0
0	155
Confusion Matrix Testing Set	
3618	1
18	21

Training Set	
Precision	1.0
Recall	1.0
Testing Set	
Precision	0.955
Recall	0.538

Accuracy Training Set	1.0
Accuracy Testing Set	0.99
F1 Score Training Set	1.0
F1 Score Testing Set	0.99



The second Random Forest classification model was performed on the cars_NAG dataset without hyperparameter tuning, resulting in a very strong precision and a high recall, making it the best-performing model so far. The **precision** score of 0.95 represents that out of all the instances predicted as positive, 95% were actually positive, while the remaining 5% were false positives.

The high **recall** score of 0.71 means that out of all the instances that were actually positive, only 71% were correctly identified by the model, while the remaining 29% were false negatives. With such an improvement in the recall score, the model is a strong predictive tool for classifying the cars_ALL dataset.

The precision-recall curve is clearly more uniform than all previous curves and indicates a strong concavity in models with strong prediction capacities. Where minor bumps are apparent, this may indicate noise with some features and possibly allude to the need to tune hyperparameters.

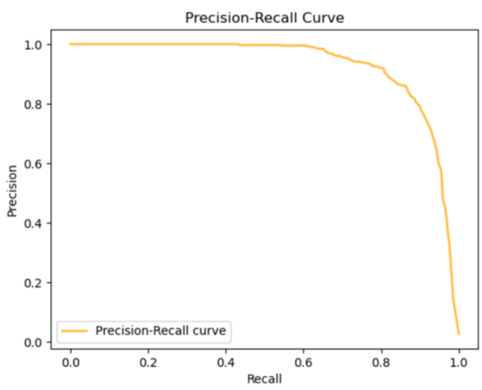
From a business perspective, the model in its current form is a strong tool that may provide insights about future customers and accurately meet the business objective in forecasting the features that may relate to the target class of prospective buyers.

Performance Metrics for Random Forest Classification 2.
Dataset: cars_NAG

Confusion Matrix Training Set	
100071	0
1	2816
Confusion Matrix Testing Set	
24994	25
201	503

Training Set	
Precision	1.0
Recall	1.0
Testing Set	
Precision	0.953
Recall	0.714

Accuracy Training Set	1.0
Accuracy Testing Set	0.99
F1 Score Training Set	1.0
F1 Score Testing Set	0.99



The final Random Forest classification model was also performed on the cars_NAG dataset, following several iterations with subsequent hyperparameter tuning. The decision to discontinue the use of the cars_ALL dataset was selected as its precision-recall curve continued to lack uniformity and concavity, meaning the dataset is likely inadequate. Hyperparameters were tweaked, altering the combinations primarily of **max_depth, n_estimators** and **criterion**. The most successful of these hyperparameter combinations are listed below. The impact of tuning hyperparameters resulted in a fractionally stronger precision and a higher recall, making it better than the previous iteration.

The **precision** score of 0.97 represents that out of all the instances predicted as positive, 97% were actually positive, while the remaining 3% were false positives. The high **recall** score of 0.73 means that out of all the instances that were actually positive, only 73% were correctly identified by the model, while the remaining 27% were false negatives. With such a high recall score, the model is still a strong predictive tool when classifying the cars_ALL dataset.

The precision-recall curve is very uniform than all previous curves and indicates a strong concavity in models with strong prediction capacities. Only one minor indent is apparent, accounting for the marginal loss in precision. As this data takes considerable time to generate, the use of this model has been postponed until the next experiment has explored applications with another model.

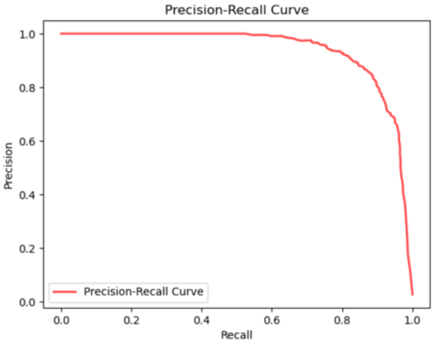
From a business perspective, the model is a strong tool that may provide insights about future customers and accurately meet the business objective in forecasting the features that may relate to the target class of prospective buyers.

Performance Metrics for Random Forest Classification 3.
Dataset: cars_NAG
Hyperparameters tuned: 'max_depth' (40)
'n_estimators' (200)
'criterion' (entropy)

Confusion Matrix Training Set	
100071	0
1	2816
Confusion Matrix Testing Set	
25001	18
190	514

Training Set	
Precision	1.0
Recall	1.0
Testing Set	
Precision	0.966
Recall	0.73

Accuracy Training Set	1.0
Accuracy Testing Set	0.99
F1 Score Training Set	1.0
F1 Score Testing Set	0.99



Feature Significance

By utilising the **feature_importances_** tool, the significance of each feature could be charted and sorted by frequency. When examining the cars_NAG dataset features, the emphasised features are notably dissimilar to experiment 3 (SVCLinear). This is because the SVC classification model doesn't facilitate feature extraction.
The feature significances following a Random Forest classification appear to weigh higher value over the **mth_since_last_serv**, **annulised_mileage, num_serv_dealer_purchased** and **total_services** features, as key indicators if an existing customer is likely to make a repurchase. The highest of these is **mth_since_last_serv,** roughly meaning that in combination with other factors, the more months between a customer's last service has passed, the more likely they will buy a new car.

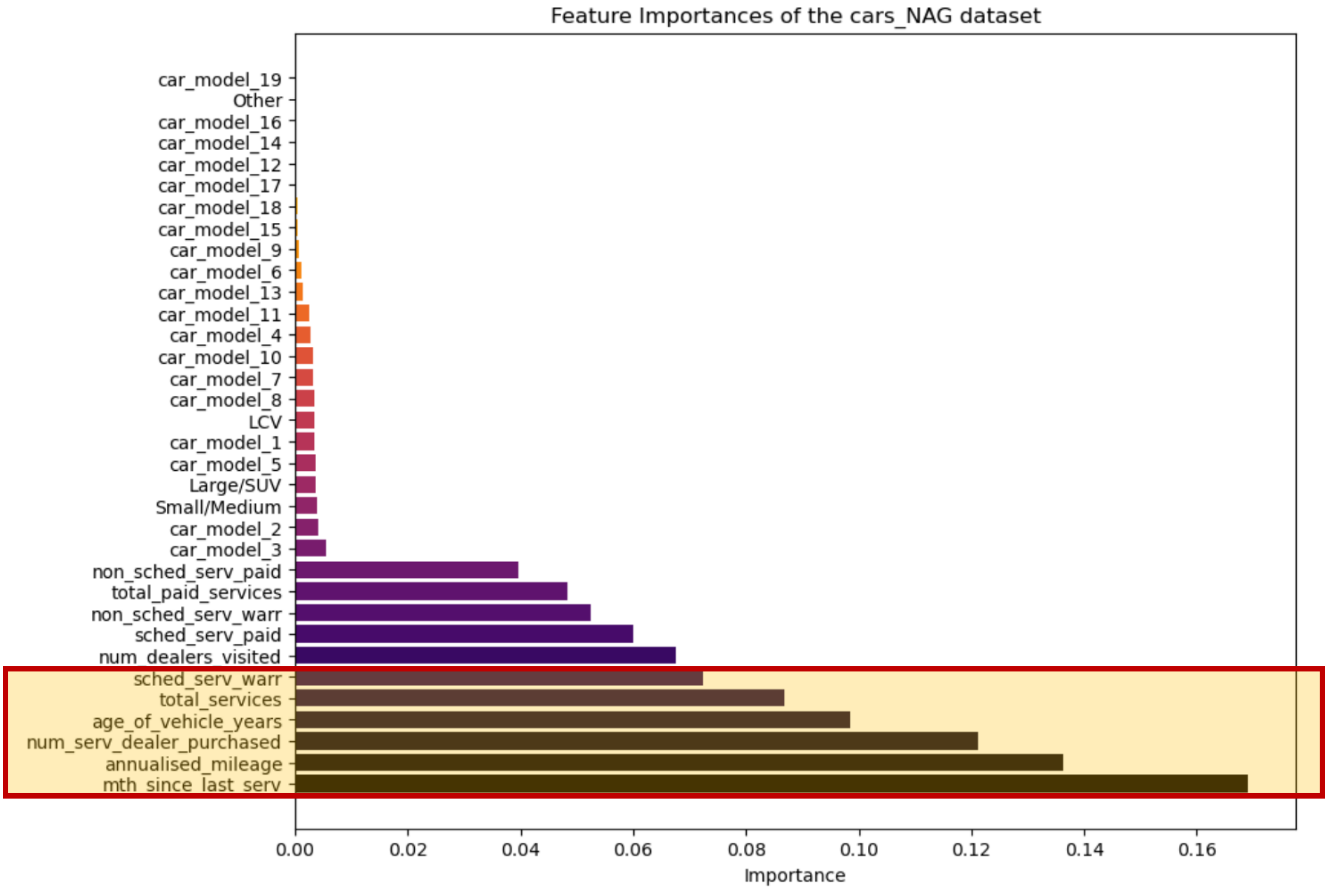


Figure 9 Feature Importance of the cars_NAG dataset following Random Forest modelling.

b. Business Impact

As the final **Random Forest** classification model (following hyperparameter tweaking) bears a stronger predictability capacity than the SCV model when searching for relationships between the **Target** variable and features of the cars_NAG dataset, it is deemed a successful experiment with room for improvement. The model, as it stands, can aid in evaluating existing customers to procure leads for marketing; however, it won't have the capacity to discern all due to a 75% success rate in recallability.

As the results are verified to be accurate through the **confusion matrix** (95% accuracy to discern a True Positive), **F1_score** (0.99) and **accuracy_score** (0.99); providing business advice on these results should entail a predominately accurate predictive capacity. Should the results not be accurate, a loss of capital would be at stake through funding an unsuccessful marketing campaign, in addition to overstocking specific models in anticipation of leads that commit to a purchase.

c. Encountered Issues

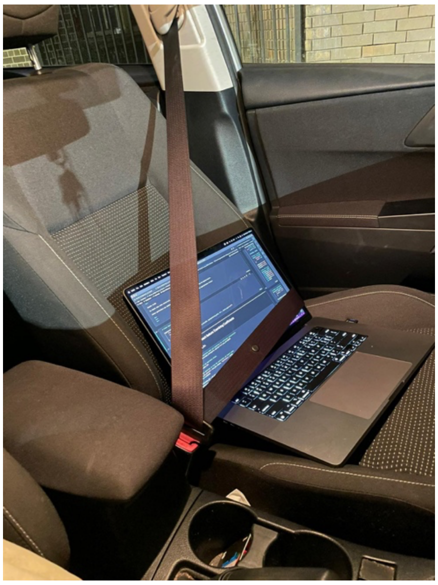
Existing issues from previous experiments were likewise encountered within this experiment.

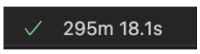
The application of both data frames

As with the previous experiment, the use of the cars_ALL data frame was ultimately discontinued during this experiment prior to hyperparameter tuning. This was because the resulting precision-recall curve was jagged and lacked uniformity.

Resolving time limitations

Processing times for these models were relatively tedious, with one particular hyperparameter dictionary requiring up to 300 minutes to process (due to the variety of selected parameters). The time requirement and computation intensity were mitigated by preventing computer sleep times and report writing while processing took place. The use of the **search.best_estimator_** tool also enabled predictions to be made on the best hyperparameter choices.





(Bonus Image): Hyperparameter tuning, while travelling to work.

FUTURE EXPERIMENT

a. Key Learning

Exploratory Data Analysis

< Data Insights from EDA remain unchanged from Experiment 1. >

Random Forest Classification

- Scikit-learn's **RandomForestClassifier** toolset proved to be powerful and flexible to integrate. The prediction metrics extracted from the model have demonstrated strong applications within the cars_NAG dataset.
- While the performance metrics aren't optimal (75% recall on testing data), they're high enough to gauge the model's prediction capacity, though there is always room for improvement.
- The implementation and application of hyperparameter tuning is a time and computational-intensive process, made more apparent through the use of RandomForestClassifier.
- The significant features evaluated from this model are **mth_since_last_serv**, **annulised_mileage, num_serv_dealer_purchased** and **total_services** (Figure 9).

b. Suggestions / Recommendations

Subsequent experiments, namely XGBoosting, may also benefit from further **feature engineering**, offering a means to transform data into more interpretable relationship modelling. As feature engineering can also reduce the dimensionality of the data and remove redundant aspects, it may lead to more promising precision and recall performance metric outcomes.

The business owner would also benefit from collecting gender and age_band with future datasets, as these features couldn't be integrated with reliability (due to entry shortages) within the current model.

The final model to consider next includes: **XGBoost Classification**

The XGB classification algorithm is a popular algorithm that combines multiple weak prediction models, like decision trees, iteratively adding trees and correcting errors of previous trees to generate an accurate prediction.

Pros:
- Efficient processing times.
- Capable of handling **imbalanced** and missing data.
- Scalable for larger datasets.

Cons:
- Typically requires more data preparation and parameter tuning than other algorithms.
- Typically doesn't perform well with too many categorical features.
- Prone to overfitting.

- XGBoost is an optimised gradient-boosting algorithm that builds an ensemble of weak models to make accurate predictions.
- It can handle missing data and has excellent predictive power for high-dimensional data.
- However, XGBoost may require more data preparation and parameter tuning than other algorithms and can be more prone to overfitting.

Deployment

While the model has the capacity to make relatively accurate predictions, deployment is not advised until an investigation into other modelling frameworks has been conducted. The rationale for this is to maximise the business' return on investment, whereas other models may provide a different method of exploring and interpreting the dataset's features.