EXPERIMENT REPORT [C]

Student Name	*Nathan Collins*
Project Name	*MLAA Assignment 2*
Date	*Apr 28 by 23:59*
Deliverables	*<MLAA_Assignment_2_3>* *<Experiment_Report_3>*

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 specific 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 3. will consist of a **Support Vector Machine** model, examining 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.

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.

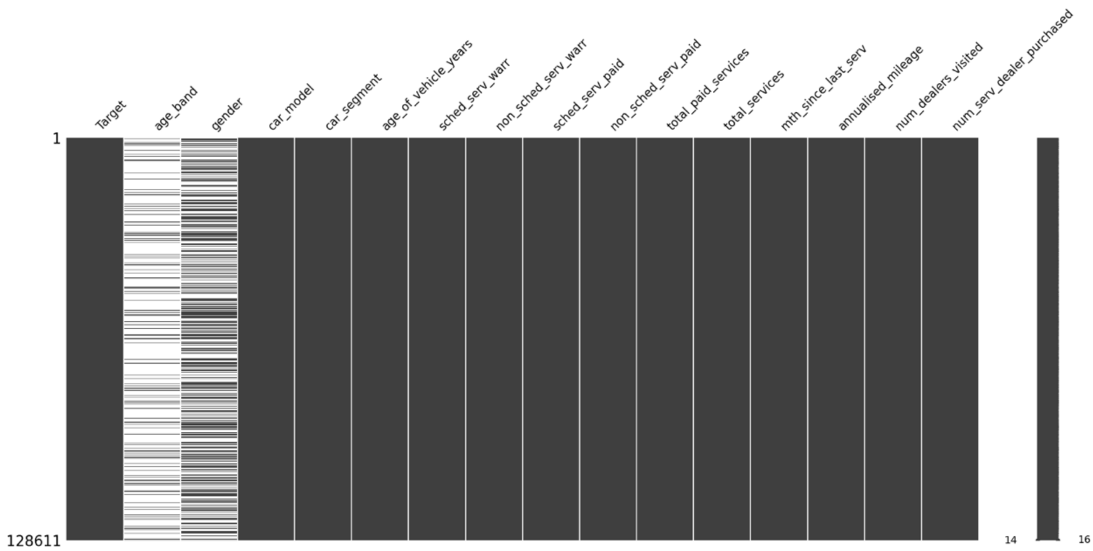


Figure 1 NaN values visualised with Missingno.

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.

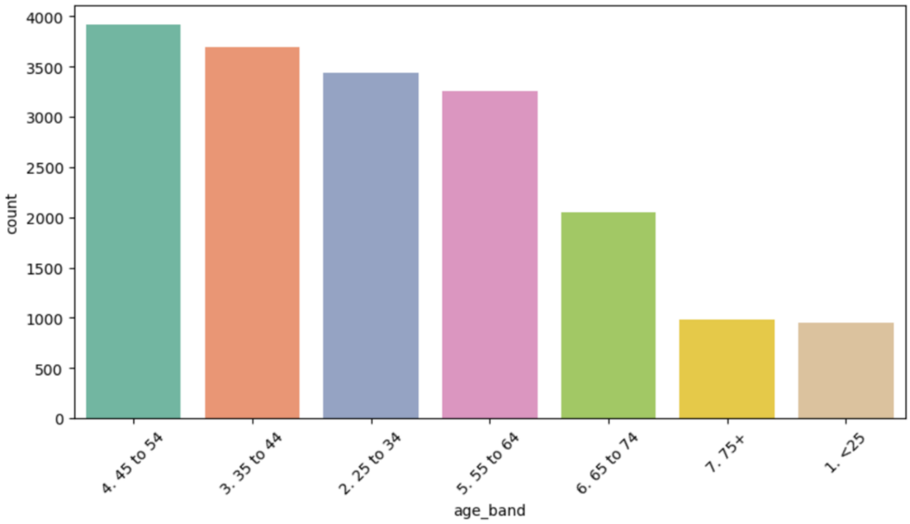


Figure 2 The age_band feature visualised.

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

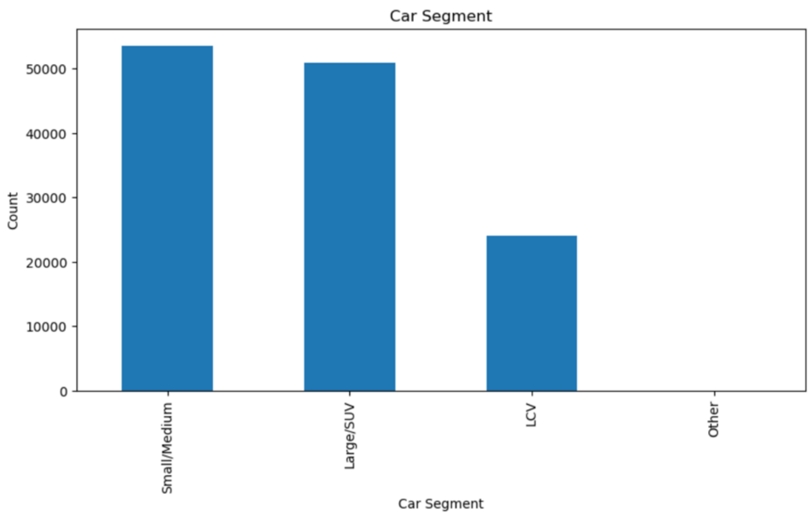


Figure 3 The car_segment feature of the cars_NAG data frame, visualised.

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.

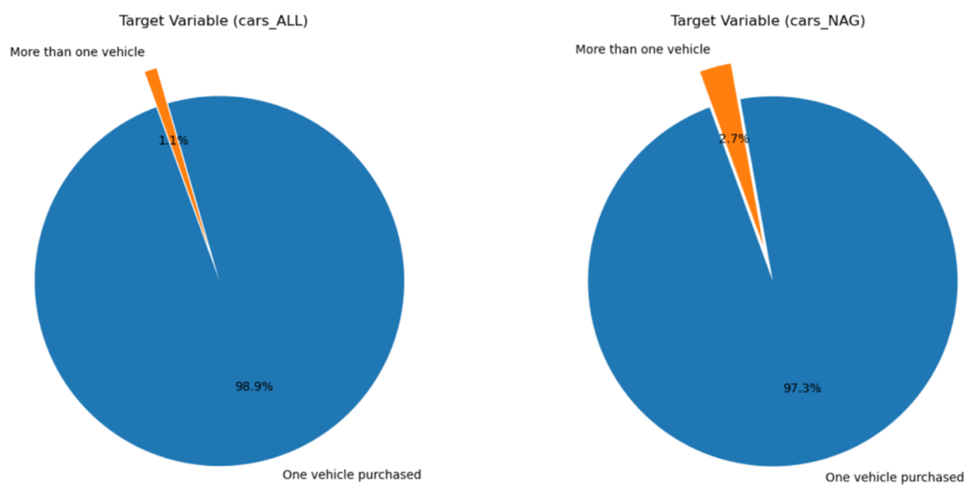


Figure 5 Pie graph visualisations of the Target variable and the class of interest.

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.

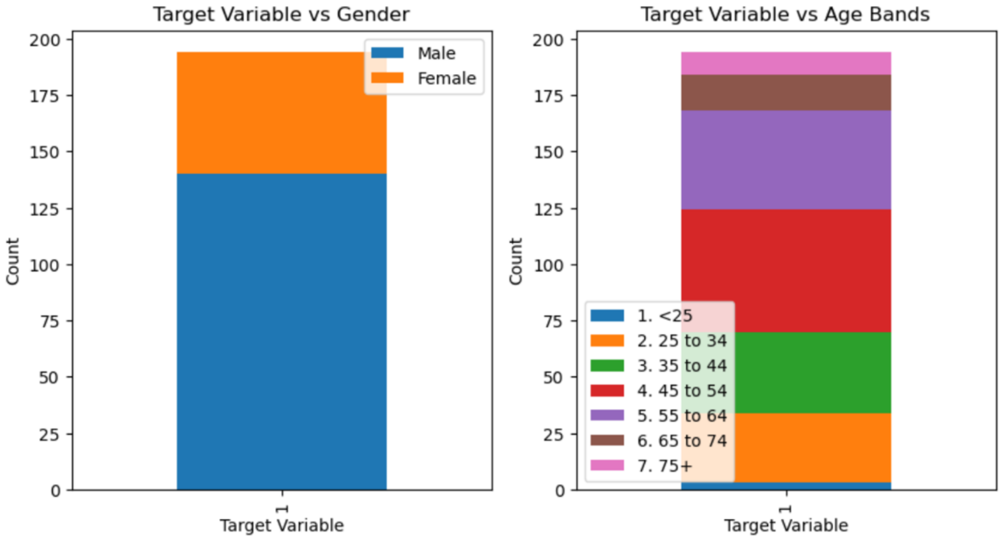


Figure 6 A segmented bar graph illustrating the male and female cohorts, beside the age band cohorts, with a Target value of 1.

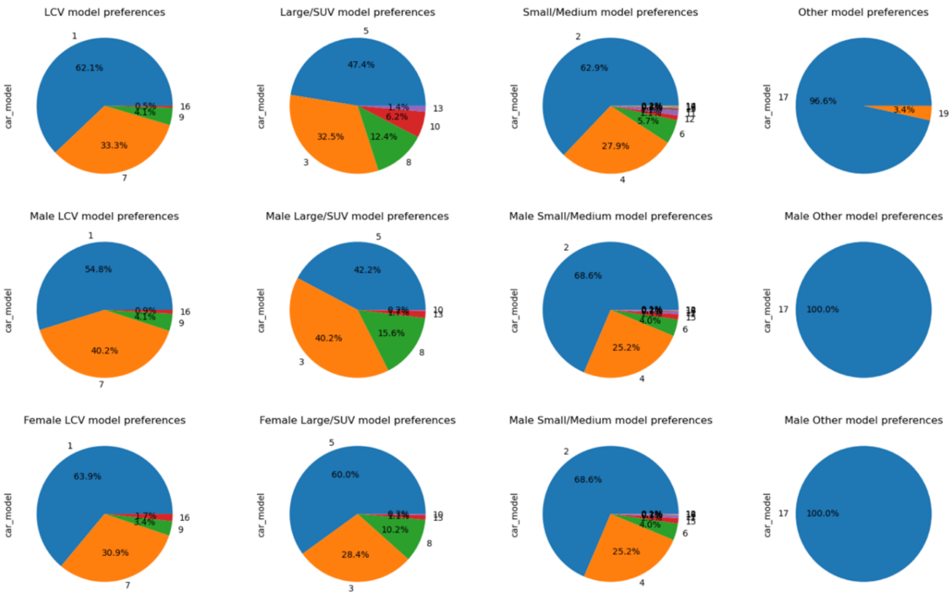


Figure 7 A series of pie charts illustrating the popularity of car models against the variety of car segments. The first row covers the overall majority, while the second and third rows cover the male and female preferences.

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.

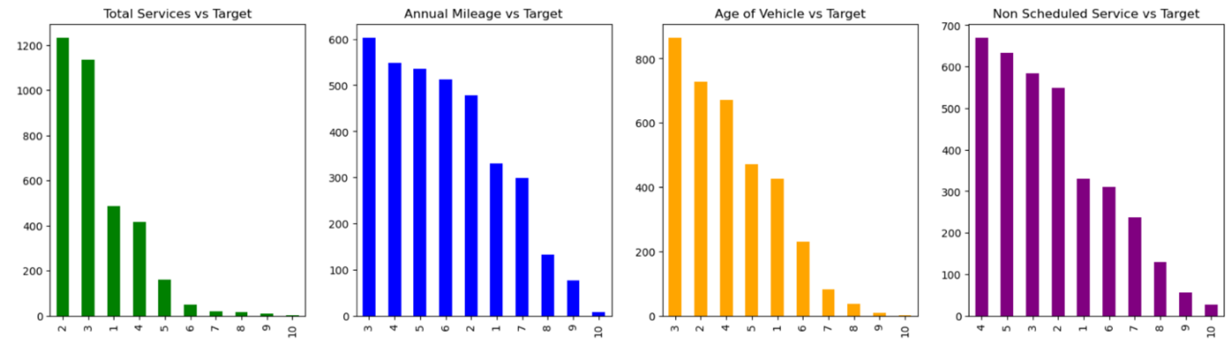


Figure 8 Four bar graphs illustrating total_services, annualised_mileage, age_of_vehicle_years, and non_sched_serv_war in deciles, against the target variable with a value of 1.

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

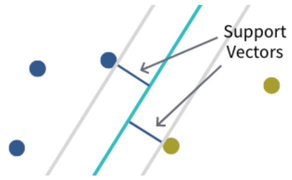
Selecting a performance metric

As the business objective resides with a binary classification problem, **Precision** was selected as 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. Should precision metrics 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. 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.



Support Vector Machine

Complexity: Intermediate to Complex

< The completed function marginally differs from Experiment 2.>

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 **SVC()**. 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 produces **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. For this experiment, 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:

Random Forest Classification

- Reduced overfitting functionality through decision trees.
- Accounts for missing values, outliers, and nonlinearity.
- Wide range of hyperparameter tweaking possibilities.

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 Support Vector Machine classification model was performed on the cars_ALL dataset without prior hyperparameter tuning, resulting in a strong precision but moderate to poor recall. The **precision** score of 0.88 means that out of all the instances predicted as positive, 88% were positive, while the remaining 12% were false positives. While the precision is strong, there is room for improvement.

The downside is the **recall** score of 0.39, describing that out of all the instances that were actually positive, only 39% were correctly identified by the model, with the remaining 61% being false negatives. With this low recall rate, the model is presently not ideal for classifying the cars_ALL dataset.

The precision-recall curve shape additionally lacks concavity and uniformity, indicating "noise" is apparent within current features and the possible need to tune hyperparameters.

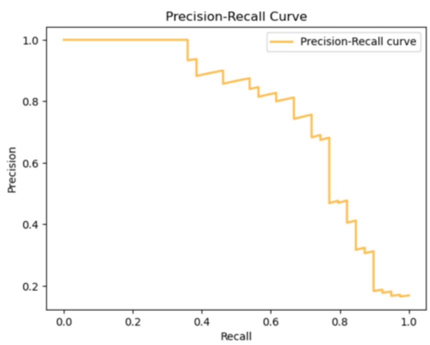
From a business perspective, the model in its current form is not adequate enough to accurately meet the business objective in forecasting the features that relate to the target class of prospective buyers. This verdict, however, has improved from the last experiment.

Performance Metrics for Support Vector Machine Classification 1.
Dataset: cars_ALL

Confusion Matrix Training Set	
14473	3
78	77
Confusion Matrix Testing Set	
3617	2
24	15

Training Set	
Precision	0.962
Recall	0.497
Testing Set	
Precision	0.882
Recall	0.385

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



The second Support Vector Machine classification model was performed on the cars_NAG dataset without hyperparameter tuning, resulting in, likewise, a high to strong precision but lower recall. The **precision** score of 0.83 represents that out of all the instances predicted as positive, 83% were actually positive, while the remaining 17% were false positives.

The **recall** score of 0.2, however, means that out of all the instances that were actually positive, only 20% were correctly identified by the model, while the remaining 80% were false negatives. With such a low recall rate, the model is remains presently not ideal for classifying the cars_NAG dataset.

The precision-recall curve is considerably more uniform than the previous curve and indicates the necessary concavity seen in models with strong prediction capacities. The curve is currently asymmetric, indicating noise with some features and possibly alluding to some features in the dataset which lack relationships. The irregularity here may possibly be remedied through tuning hyperparameters.

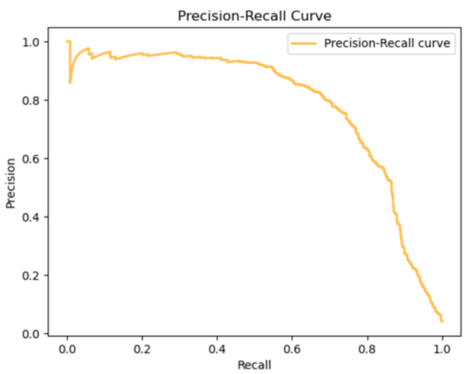
From a business perspective, the model in its current form is still not adequate enough to accurately meet the business objective in forecasting the features that may relate to the target class of prospective buyers.

Performance Metrics for Support Vector Machine Classification 2.
Dataset: cars_NAG

Confusion Matrix Training Set	
99951	120
2193	624
Confusion Matrix Testing Set	
24991	28
563	141

Training Set	
Precision	0.839
Recall	0.222
Testing Set	
Precision	0.834
Recall	0.200

Accuracy Training Set	0.98
Accuracy Testing Set	0.98
F1 Score Training Set	0.97
F1 Score Testing Set	0.97



The final Support Vector Machine classification model was also performed on the cars_NAG dataset with hyperparameter tuning. This dataset was selected as the cars_ALL dataset's precision-recall curve lacked uniformity and concavity. Hyperparameters were tweaked across several iterations, with subsequent tweaks in the combinations of **C, kernal** and **gamma**. The most successful of these hyperparameter combinations are listed below.

The **precision** score of 0.9 represents that out of all the instances predicted as positive, 90% were actually positive, while the remaining 10% were false positives. While a 9/10 precision is not perfect, it is a strong outcome, and the model has a high degree of confidence in its positive predictions.

The **recallability** was also moderately high at 0.73. This meant that out of all the instances that were actually positive, 73% were correctly identified by the model, while the remaining 27% were false negatives. While not as strong as the precision score, the results look promising, where the model is able to find a significant portion of positive instances but may still miss some.

The precision-recall curve demonstrates a clear concavity and uniformity with a minor irregularity in the top left. a possible lack of relationship between some features with the SVC model applied.

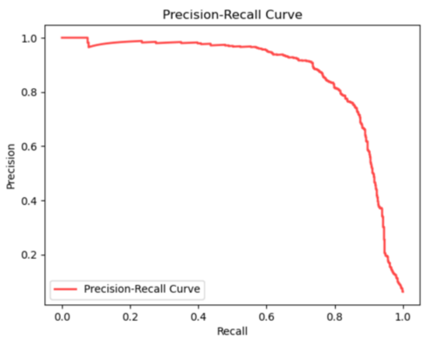
From the perspective of the business objective, the model would be capable of identifying possible future customers, but not all. There is still room for improvement in elevating the recallability score through future models.

Performance Metrics for Support Vector Machine Classification 3.
Dataset: cars_NAG
Hyperparameters tuned: 'C' (200)
'kernal' (rbf)
'gamma' (auto)

Confusion Matrix Training Set	
99951	120
619	2198
Confusion Matrix Testing Set	
24967	52
188	516

Training Set	
Precision	0.948
Recall	0.780
Testing Set	
Precision	0.908
Recall	0.733

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



Feature Significance

By ranking the coefficient values generated from the model, the significance of each feature could be charted. As SVC doesn't offer feature significance tools, insights, while not satisfactory, were derived from SVCLinear model. The goal was to derive some insights, regardless of the dataset requiring a classification model over a linear model. Ultimately, **these insights are not indicators of priorities for the business**, however when examining the cars_NAG dataset features with SVCLinear, car_segment and some car_model features appear to play stronger, more decisive roles when determining if a customer is likely to make another purchase.

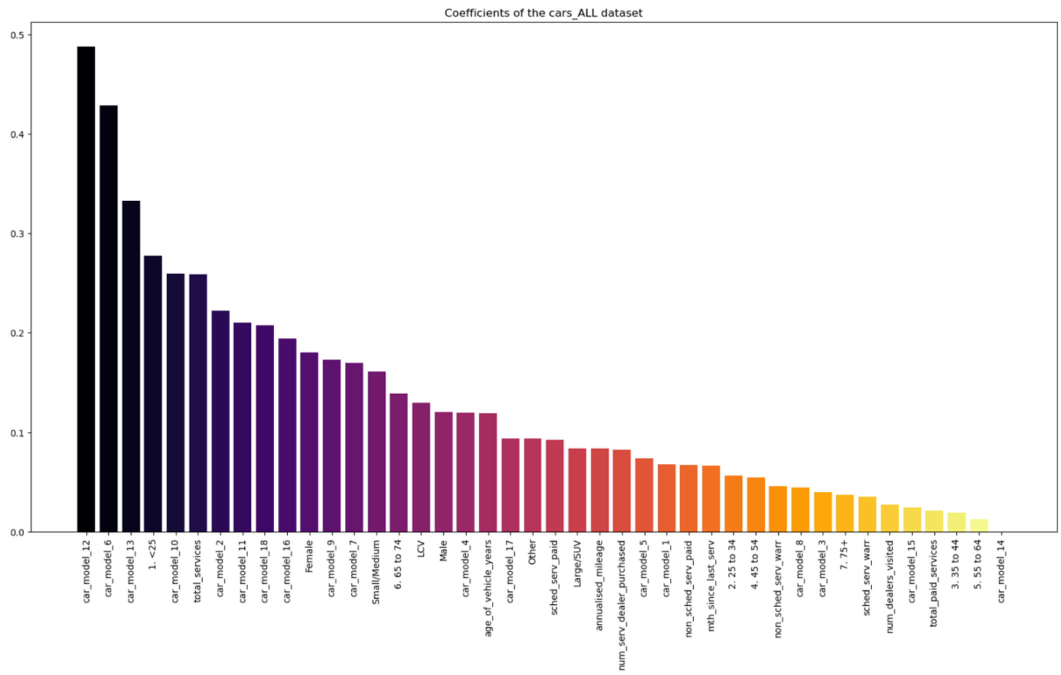


Figure 9 Coefficient values of the cars_ALL dataset following SVC modelling.

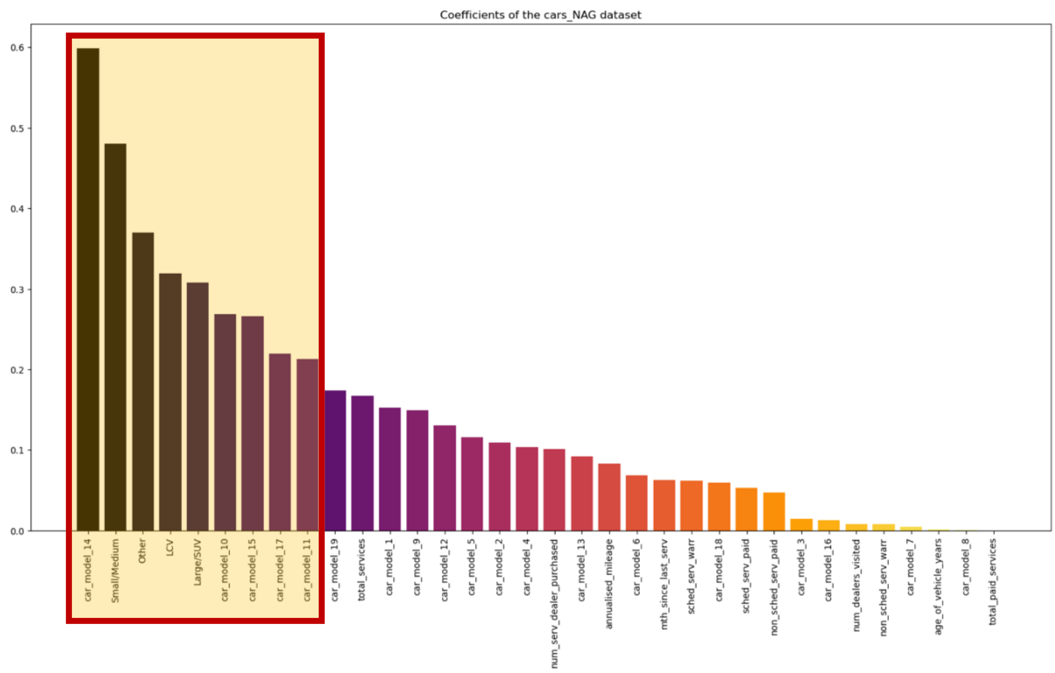


Figure 10 Coefficient values of the cars_NAG dataset following SVC modelling.

b. Business Impact

As the results of the final **support vector machine** classification model (following hyperparameter tweaking) bear some predictability capability of relationships between the **Target** variable and variables in the cars_NAG dataset, it is deemed a successful experiment with room for improvements; and, furthermore, as a step towards meeting the business objective. The model as it stands, could aid in evaluating existing customers to procure leads for marketing; however, it won't have the capacity to discern all.

As the results are verified to be accurate through the **confusion matrix** (90% 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 with 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 a relatively tedious and iterative process of trial-and-error, following hyperparameter tweaking. This was resolved by integrating the **search.best_estimator_** feature, enabling predictions to be made on the best hyperparameter choices. Often the wait times between each iteration were solved by filling the time with report writing.

Discerning feature significance from an SVC model

As Support Vector Machine classification models don't provide a means to extract features significance, discerning the importance of variables is difficult. It is hoped that exploring further models may shed insights into these features, to derive business insights.

FUTURE EXPERIMENT

a. Key Learning

Exploratory Data Analysis

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

Support Vector Machine Classification

- Overall, scikit-learn's **SCV** framework proved to be elegant and simple to integrate. The prediction metrics extracted from the model have demonstrated its utility within the cars_NAG dataset.
- A significant insight drawn from the SCV framework is that it now appears likely that the dataset has existing relationships to derive the capacity to predict future customers.
- While the performance metrics aren't perfect, they're high enough to gauge the model's prediction capacity is there, though there is room for improvement.
- The implementation and application of hyperparameter tuning is a time-intensive process that resulted in marginally shifting the outcomes to better suit the business objective needs.
- The significant features evaluated from the SVCLinear model, while not to be used, are mostly car_model and car_segment varieties (Figure 10).

b. Suggestions / Recommendations

Future experiments should explore further model frameworks and hyperparameter interactions, as the threshold of surpassing the output of the SVC framework is relatively high. Subsequent experiments 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.

Other models to consider next include:

Random Forest Classification

This classification algorithm may also provide value in future experiments by predicting classes of a new observation based on a highly adjustable set of input features.

Pros:

- Offers functionality to handle outliers (not applicable) and missing data.
- Lower susceptibility to overfitting through bagging.
- Provides modelling functionality to a non-linear set of data.

Cons:

- Requires a large set of trees to achieve optimal performance, which results in computationally intensive processing requirements.
- As the existing dataset is imbalanced, there is a poorer performance probability.

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

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