EXPERIMENT REPORT

Name	Monali Patil
Project Name	Customer-Vehicle-Repurchase-Prediction
Deliverables	Model name: Random Forest Classifier Notebook name: MachineLearning_RandomForest_Exp4.ipynb Project Repo: https://github.com/monalippatil/MachineLearning-Customer-Vehicle-Repurchase-Prediction.git

---

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	Explain clearly what is the goal of this project for the business. How will the results be used? What will be the impact of accurate or incorrect results? Objective: The aim of this project is to develop a binary classifier model that can predict customers who are likely to purchase a new vehicle. To achieve this, a dataset of car repurchases will be used for this binary classification problem and through data exploration, cleaning, feature scaling, selection of appropriate algorithm, hyperparameters and employing various techniques such as cross-validation, hyperparameter tuning, feature importance etc., to improve the accuracy of the model. Application: The outcomes of this model can be used to identify the customers who are most likely to buy a new vehicle, and this information can be used to focus marketing efforts on those potential customers which can lead to an effective and efficient promotional campaign. Impact: The model's accurate results can help the company to target the right customers who are more likely to buy the vehicle through a marketing campaign which could save marketing costs and the campaign could generate higher sales with fewer marketing expenses. However, incorrect results of the model can result in wasted marketing efforts and resources and thus, the marketing campaign may fail to achieve the desired results and sales figures.

1.b. Hypothesis	Present the hypothesis you want to test, the question you want to answer or the insight you are seeking. Explain the reasons why you think it is worthwhile considering it, Hypothesis: The objective of this experiment is to effectively identify if present customers are likely to buy a new vehicle, and to carry out this initial experiment, the Random Forest Classifier algorithm has been employed for conducting and evaluating the performance to determine if this algorithm is suitable for this Binary Classification task. Rationale: Considering the business objective of determining potential customers leads for a marketing campaign, there is no good theory to map and select a suitable algorithm for this binary classification problem, therefore performing different experiments to discover which algorithm and algorithm configuration results in the best performance for this binary classification task.
1.c. Experiment Objective	Detail what will be the expected outcome of the experiment. If possible, estimate the goal you are expecting. List the possible scenarios resulting from this experiment. The expected outcome of this experiment would be the development of a binary classifier model utilizing Random Forest Classifier algorithm that can accurately predict which customers are more likely to buy a new vehicle. The model would be trained on a dataset of current customer information and their buying behavior, and it would use Random Forest Classifier algorithm to identify patterns and make predictions about future buying behaviour of the customers. The goal of this experiment would be to identify potential customers for a promotional marketing campaign aimed at increasing vehicle sales, employing Random Forest Classifier algorithm. By focusing on customers who are more likely to buy a new vehicle, the marketing campaign could be more targeted and effective. The possible scenarios resulting from this experiment include following: • The model accurately predicts which customers are likely to buy a new vehicle, allowing the marketing team to focus on these potential customers and increase sales. • The model has a high false positive rate, indicating that the model predicts some customers will purchase a new vehicle when they actually won't purchase. This could result in wasted marketing resources and a lower return on investment. • The model has a high false negative rate, indicating that the model fails to predict some customers will purchase a new vehicle when they actually will. This could result in missed opportunities for sales and revenue. • The model is not accurate enough to be useful for predicting customer behavior, and the experiment is unsuccessful. In this case, the marketing team may need to explore other methods for identifying potential customers for their promotional campaign.

2. EXPERIMENT DETAILS

Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.

2.a. Data Preparation

Describe the steps taken for preparing the data (if any). Explain the rationale why you had to perform these steps. List also the steps you decided to not execute and the reasoning behind it. Highlight any step that may potentially be important for future experiments.

To prepare for using the binary classification algorithm, the data for current customers was explored, cleaned up and prepared through the subsequent activities.

• Data Understanding

1] Loading Data: To use and create a binary classification model, imported data from a CSV file into the pandas dataframe.

2] Exploring Data: Explored and analysed current customer's data utilizing various following pandas functionalities to understand and identify patterns to further determine potential customers who are more likely to purchase a new vehicle.

* Additionally, to ensure the quality of the data to be utilized by the model analyzed.
- Missing/null values.
- Duplicate records.
- Outliers for numerical features.
- Data distribution for various categorical and numerical features.
- Distinct values of the categorical features.

a) df.head(): Examining initial observations of the dataset.
b) df.shape(): Examining the dimension of the dataset.
c) df.columns: Inspecting features name.
d) df.info(): Inspecting the summary information of the attributes of the dataset.
e) df.describe(): Examining the statistical information of integer variables of the dataset.
f) df.describe(include='all'): Examining statistical summary information for all variables of the dataset across different data types.
g) df.isnull().sum(): Examining whether there are any null values in the dataset.
h) df.duplicated().sum(): Examining whether there are any missing values in the dataset.

• Data Preparation

3] Treating Missing Values

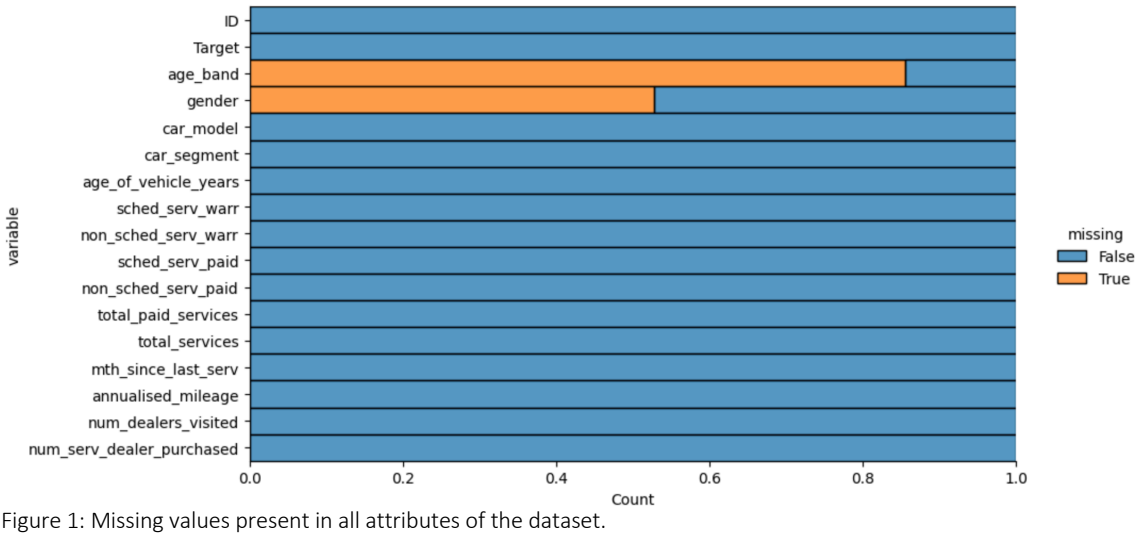

Figure 1: Missing values present in all attributes of the dataset.

* The chart from Figure 1, indicates that only two variables, 'age_band' with around 85% and 'gender' with over 50% contain missing values. However, there are no missing values for the rest of the features.

* As machine learning algorithms are incapable of processing features with missing values, we have two options to address these missing values.

i) Replace missing values with the relevant value: Using the most frequent occurring (mode) value from the feature to replace missing values in categorical variables may not accurately represent the distribution of data in those variables 'age_band' and 'gender', especially when large portion of the data is missing. This approach could also create an imbalance in the data by introducing a high number of imputed mode values, potentially leading to biased results.

ii) Eliminate the columns containing missing values entirely: It would be suitable to remove these two 'age_band' and 'gender' features since replacing any value would significantly distort the data due to the high percentage of missing values in these variables.

* Therefore, removed 'age_band' and 'gender' features from the vehicle repurchase dataset using pandas drop() function and specifying these two feature names.

4] Transforming Categorical Data into Numerical

Attribute Name	Distinct/Unique Values	Count of Distinct Values
car_model	model_1, model_2, model_3, model_5, model_6, model_4, model_7, model_8, model_9, model_10, model_11, model_13, model_12, model_14, model_15, model_16, model_17, model_18, model_19	19
car_segment	LCV, Small/Medium, Large/SUV, Other	4

Table 1: Distinct values present in ‘car_model’ and ‘car_segment’ attributes of the dataset.

* The table 1 illustrates that the features 'car_model' and 'car_segment' contain categorical data, with 19 distinct car models and 4 different vehicle segments respectively. However, this data

cannot be used directly in binary classification algorithms for machine learning because these algorithms operate on mathematical equations that require numerical inputs.

* Additionally, these features 'car_model' and 'car_segment' hold significant information regarding the model and category of vehicles that customers have purchased. This information can be used to predict whether the customer is inclined to purchase a new vehicle.

One-Hot Encoding

* Therefore, it is necessary to convert the categorical data of these features into numerical data. To achieve this, employing one-hot encoding method, which creates binary columns for every distinct category in the feature.

* This method is preferred as there is no inherent order among the category values of the feature, and do not want to create any random relationships between them and prevent model from assuming natural ordering among these distinct categories that may suffer from model bias.

5] Selecting Target Variable & Features

* The 'TARGET' feature is the target variable representing customers who have bought more than one vehicle with class 1 and those who have only purchased one vehicle with class 0. This, along with the predictor features, are separated into 'X' and 'y' variables to build a binary classification model.

6] Splitting Data into Different Sets: Training, Validation and Testing

* Creating a validation set provides adaptability to carry out multiple experiments and allows the comparison of the model's performance on the training set and the validation set to determine if the model is overfitting or underfitting.

* Furthermore, using a validation set, we can adjust model's hyperparameters and retrain the model until it performs well on the validation set. After we have optimized the model's performance on the validation set, we can then apply the model on the test set to assess its ability to generalize to new and unseen data.

* Therefore, utilizing train_test_split() function from sklearn.model_selection module of sklearn library, the car repurchase dataset is split into training (60%), validation (20%), and testing (20%) sets to leverage the flexibility of multiple experimentation.

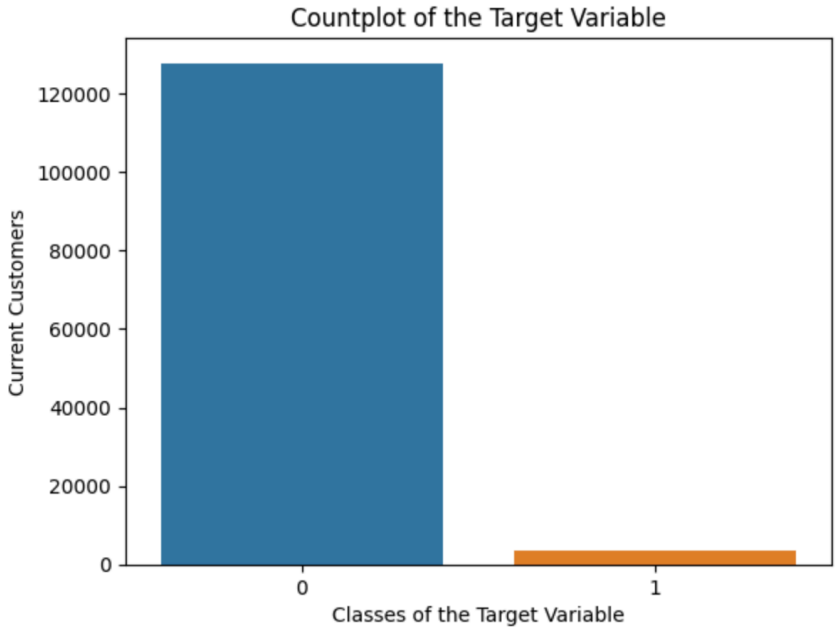

Figure 2: Examining the quantity of each class of the target variable from the actual dataset.

* Additionally, for any (binary) classification problem, it is essential to examine the frequencies/rates of every class within the target variable and ensure that the splitting of data into different sets matches as similar to the actual data spread.

* As indicated in the Figure 2, the car repurchase dataset is highly imbalanced, it is necessary to obtain as similar distribution of distinct classes of the target variable for all of the different sets as original dataset. After splitting process, following are the frequencies of the target variable classes.

Dataset/Frequency Rates	Class 1	Class 0
Actual Data:	0.97	0.026
Training Set:	0.97	0.026
Validation Set:	0.97	0.026
Testing Set:	0.97	0.028

7] Features Scaling

* Machine learning algorithms often use some form of distance calculation to determine the relationship between different features in the dataset. If the features are on different scales, the algorithm may prioritize some high value range features over others which may have more significant information, leading to biased results.

* In addition, the values for the numerical features are divided into ten deciles, ranging from 1 to 10, while categorical data that has been converted into numerical features, consist of either 0 or 1.

* Therefore, performing scaling for all the features so that all the features values are at same level and that the algorithm can use all the information from the dataset features to learn generalised patterns, identify buying behaviour and make accurate predictions.

StandardScaler

* Employing StandardScaler class which is imported from sklearn.preprocessing module to scale and bring all features values at same level. Choosing this method because it does not change the shape of the data distribution and preserves outliers as it scales the data based on the mean of 0 and standard deviation of 1, of the entire dataset, rather than individual datapoints.

The following are some crucial steps that could be important for any of the classification experiments in future.

- It is important to thoroughly examine and handle missing values in order to avoid introducing biases in the model.
- Outliers and duplicate values should be carefully analysed and reviewed from business perspectives and treated accordingly.
- The decision to use LabelEncoder, OrdinalEncoder or OneHotEncoder for converting categorical data into numerical form should depend on various factors, including the nature of the categorical values and whether or not there is any inherent order among them. Careful consideration is necessary to determine the appropriate method.
- Similarly, the selection of feature scaling method - whether to use MinMaxScaler, MaxAbsScaler, or StandardScaler - should be made based on whether or not there is an inherent order in the data, as well as other factors such as the need to maintain the integrity of outliers.
- For classification problems and especially for imbalanced data, is essential to ensure the frequency of each class in the target variable in each set is representative of the actual data. The "stratify=y" parameter could be used during the splitting process if the class distribution is not similar to the actual data. This will help to maintain the integrity of the data and ensure accurate results.
- To prevent the model from overfitting on specific data points and to enable it to learn generalized patterns, it is important to eliminate any identifier attributes.

2.b. Feature Engineering

Describe the steps taken for generating features (if any). Explain the rationale why you had to perform these steps. List also the feature you decided to remove and the reasoning behind it. Highlight any feature that may potentially be important for future experiments

	ID
count	131337.000000
mean	77097.384180
std	44501.636704
min	1.000000
25%	38563.000000
50%	77132.000000
75%	115668.000000
max	154139.000000

Figure 3: Statistical summary of 'ID' attribute.

	* The 'ID' attribute seems to be an identifier and its values ranges from 1 to 154139 as indicated in Figure 3. It does not have any predictive value as it contains a unique value for each observation, which means that its values do not contribute to any patterns or trends in the data. * And including them in the analysis can lead to overfitting straight a way, where model would fit these specific values from the feature rather than the underlying generic patterns in the data. * Therefore, removed 'ID' attribute from the vehicle repurchase dataset using pandas drop() function and specifying this feature name. Note: Removal of 'age_band' and 'gender' features is stated in earlier part 3] Treating Missing Values of 2.a Data Preparation section.
2.c. Modelling	Describe the model(s) trained for this experiment and why you choose them. List the hyperparameter tuned and the values tested and also the rationale why you choose them. List also the models you decided to not train and the reasoning behind it. Highlight any model or hyperparameter that may potentially be important for future experiments Being one of the algorithms applicable to problems involving Binary Classification, employing Random Forest Classifier in this binary classification experiment to accurately predict whether current customers are likely to purchase a new vehicle, and comparing the performance of this classifier with previous models to determine the most suitable one for this binary classification task which can then be used to target potential customers for a marketing campaign. The target variable consists of either 0 or 1, where 0 represents the purchase of a single vehicle by the customer, while 1 represents the purchase of multiple vehicles. As we aim to predict only these two possible classes outcome, this Binary Classification model is the right choice along with learning objective purpose. Hyperparameters Selected. Hyperparameters Name: Values Tested n_estimators: 91 max_depth: 9 min_samples_leaf: 13 max_features: log2 class_weight: balanced criterion: entropy random_state: 19 * Initially, utilizing the default hyperparameters of the Random Forest Classifier algorithm and evaluated the performance. * The n_estimators hyperparameter determines the number of Decision Trees in the Random Forest. While min_samples_leaf hyperparameter determines the minimum number of samples required to be present at a leaf node in the Decision Tree.

* The hyperparameter max_features='log2' is used to control the maximum number of features that are considered for splitting each node in the Decision Tree of the Random Forest.

* The max_depth hyperparameter is used to specify the maximum depth of each Decision Tree in the Random Forest, which is the maximum number of levels allowed in the tree.

* Utilising class_weight='balanced' hyperparameter to adjust the weights of the classes based on their proportion in the training set, which can help to address the issue of high imbalance classes and improve model performance.

* The criterion='entropy' hyperparameter is used to specify the splitting criterion used by each Decision Tree in the Random Forest.

* The random_state parameter is used to ensure that the model can be reproduced exactly in the future which is useful when comparing different models or tuning hyperparameters.

Hyperparameter Tunning

* With the learnings from the previous 3rd experiment, it is possible to gauge a range of values for hyperparameters that are common to both algorithm's Decision Tree and Random Forest because they both utilize similar logic for identifying and classifying datapoint classes.

* Therefore, employing Hyperparameter Tuning to determine optimal values for the hyperparameters of the Random Forest Classifier algorithm.

Random Search (RandomizedSearchCV)

* Utilizing Random Search method for hyperparameter tuning, as it introduces randomness by randomly selecting values across the search space which may provide combination of hyperparameter values that result in the lowest possible error.

* This may not always be possible with Grid Search, as evenly spaced grid points search may not capture the optimal hyperparameter values.

Cross-Validation: StratifiedKFold

* Performing cross-validation to train and test the model over multiple folds of the dataset to promote model generalisation and to understand the model performance over the whole dataset rather than relying on a single train-validation split.

* Utilising StratifiedKFold technique for cross validation to address the issue of imbalanced dataset as this technique takes into account the distribution and frequency of each class in the target variable and defines the split ensuring that the split between training and validation sets maintains similar distribution to the original data.

* With k-fold method, random split is performed, which may result in biased outcomes due to uneven class representation in the splits.

Because of time limitations, could not fully analyse and experiment with hyperparameters such as criterion='gini' and max_features='sqrt', min_samples_split, min_weight_fraction_leaf, and max_leaf_nodes of the Random Forest Classifier algorithm, which could potentially influence the

	model's performance. However, I intend to investigate these hyperparameters in future experiments.

3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance

Score of the relevant performance metric(s). Provide analysis on the main underperforming cases/observations and potential root causes.

Performance Metrics

To evaluate the model's performance utilizing the below performance metrics.

- Precision
- Recall
- Weighted F1 Score
- Binary F1 Score
- Confusion Matrix

Baseline Performance (Null Accuracy)

weighted F1 Score: 0.9608

Exercise 1:

Algorithm: Random Forest Classifier model

Hyperparameters: Default Hyperparameters

Performance Metrics:

Dataset	Precision	Recall	weighted F1 Score	binary F1 Score
Training:	1.0	0.9986	0.9999	0.9993
Validation:	0.9427	0.7304	0.9910	0.8231

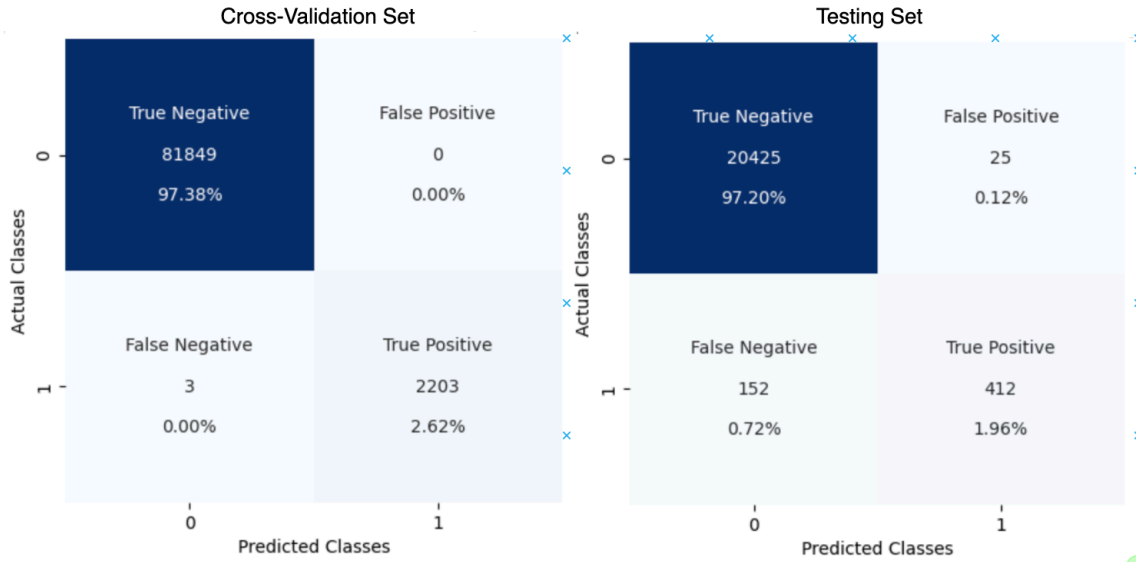

Figure 4: Confusion Matrix of Random Forest Classifier model with default hyperparameters.

* The binary F1 score achieved by the model with 0.8231 on the validation set is relatively lower compared to the 0.9993 score achieved on the training set, indicating that the model is overfitting.

* Moreover, the Precision and Recall scores of validation are considerably lower compared to the training scores, suggesting the similar overfitting nature of the model.

* Additionally, from the confusion matrix, the false negative errors observed in the training are 3 and validation is 152, which is reasonable for the validation data informing that the model is not capturing important patterns and not generalised enough to identify potential customers from unseen data.

* The binary F1 Score of the classification results that belongs to positive class with 0.9997 for training and 0.7726 for validation, indicates that the model is significantly overfitting. The Precision and Recall scores of both the training and validation sets demonstrate similar findings.

* Therefore, employing Hyperparameter Tuning to determine optimal values for the hyperparameters of the Random Forest Classifier algorithm to address the overfitting nature of the model.

Below are the Hyperparameters tuned values of the Random Forest classifier:

{'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 9, 'max_features': 'log2', 'min_samples_leaf': 13, 'n_estimators': 91}

Exercise 2:

Algorithm: Random Forest Classifier model

Hyperparameters: class_weight='balanced', criterion='entropy', max_depth=9, max_features='log2', min_samples_leaf=13, n_estimators=91

Performance Metrics:

Dataset	weighted F1 Score	binary F1 Score
Cross-validation:	0.9439	0.3971
Testing:	0.9427	0.4098

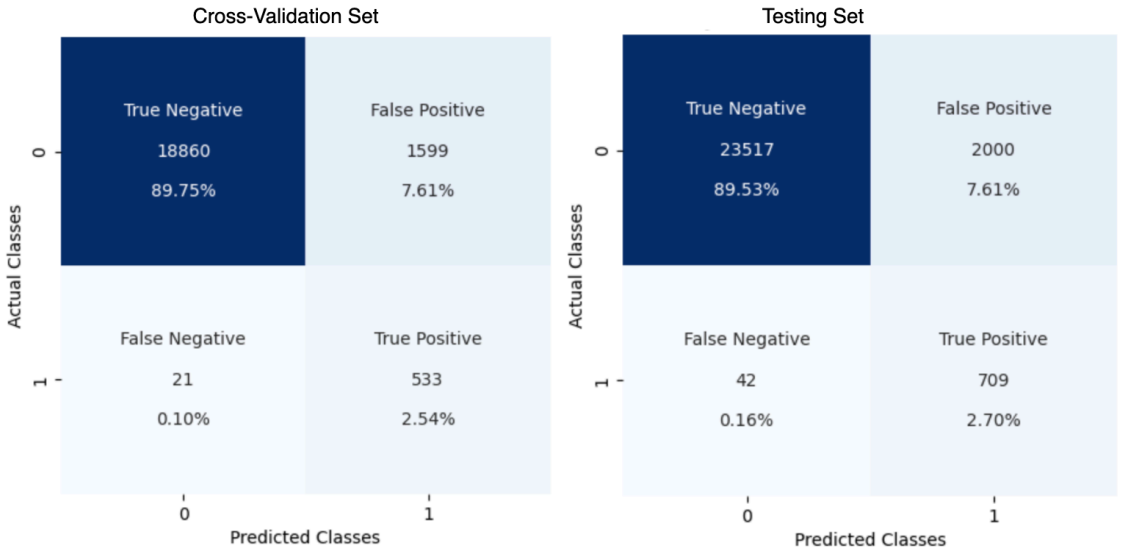

Figure 5: Confusion Matrix of Random Forest Classifier model with hyperparameter tuned values.

* The difference between the weighted F1 score with 0.9439 for training-validation using cross-

validation and testing with 0.9427 is minute indicating that the model with tuned hyperparameters is performing consistently overall across the three sets.

* Similarly, the binary F1 Score for training-validation using cross-validation is 0.3971 and the testing set is 0.4098, illustrating that the model is behaving almost uniformly on all the sets with minute variation.

* Moreover, the False Negative errors for training-validation and testing are 21 and 42 respectively which are low which is important to ensure that the marketing campaign is targeting the right customers, which can lead to more efficient and effective use of company resources.

Feature Importance

* Extracting feature importance measure that could provide valuable insights into the data and help in understanding which features are contributing the most to the model's predictions.

* Additionally, this information is readily available, as the Random Forest Classifier algorithm calculates the change of purity level(group of datapoints with same target class) for each feature while splitting and by aggregating these values for each feature, and thus, determining the contribution of each feature to predictions.

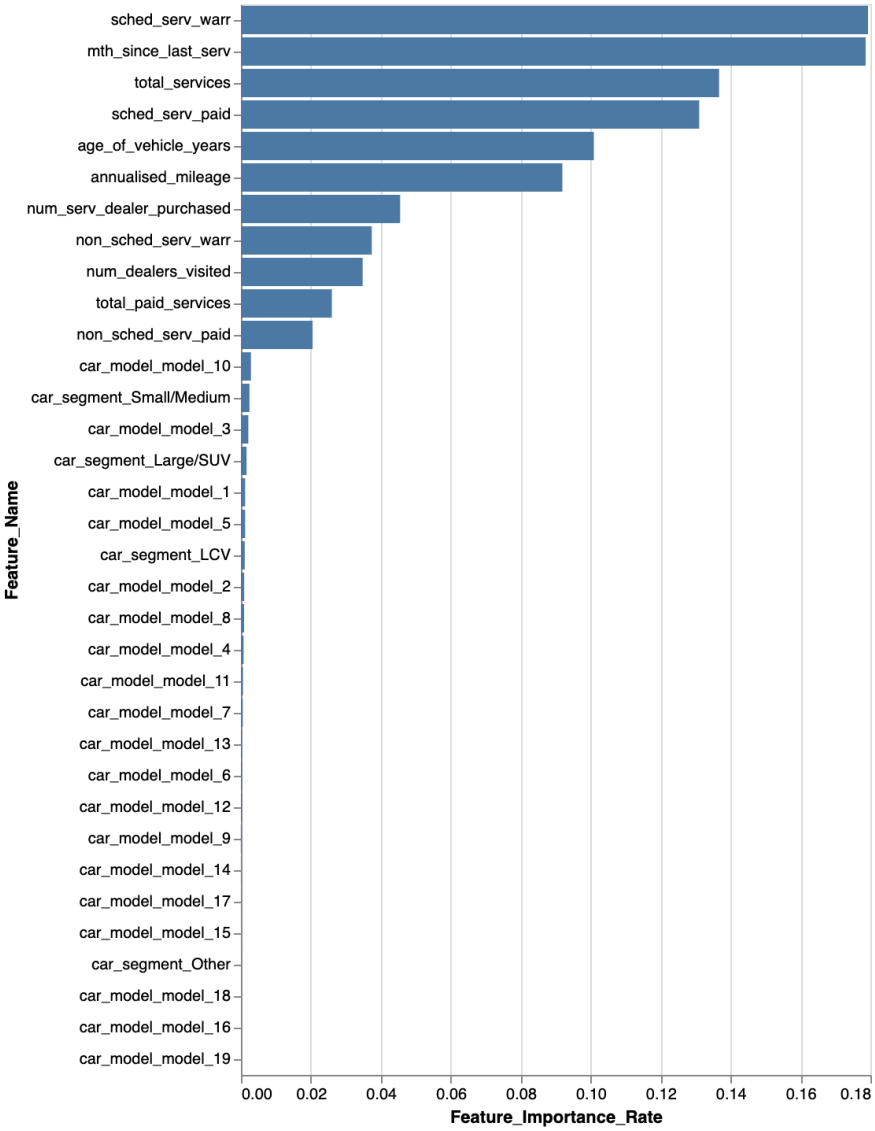

Figure 6: Chart informing the Features that are important and contributing to the prediction.

	* The bar graph from Figure 4, illustrates that the least contributing features are related to car models and segments represented by separate features for every distinct categorical value transformed to numerical. * Additionally, the rest of the features (except those related to car models and segments) are influencing most for prediction of target class 1 representing the customers that are likely to purchase a vehicle or class 0 of not interested and their feature's importance rate ranges between 0.02% to 0.19%. * In the next experiment, utilizing the Feature Importance information with this improved Random Forest Classifier model to verify if the model's generalized performance could be enhanced.
3.b. Business Impact	Interpret the results of the experiments related to the business objective set earlier. Estimate the impacts of the incorrect results for the business (some results may have more impact compared to others) The Random Forest Classifier model from 2nd exercise, utilizing hyperparameters tuned values resulted in 0.3971 for cross-validation and 0.4098 for the testing set, illustrating that the model has overcome the problem of overfitting and is performing consistently across all sets and capturing underlying patterns and buying behaviours of the customers. The False Negative error rate is 0.10% for cross-validation sets and 0.16% for testing, which is considerably low informing that model is making fewer mistakes for customers that are willing to purchase new vehicles. However, there is a relative rise in False Positive errors, which indicate customers being wrongly classified as potential buyers of a new vehicle, but this would not have a significant impact on the company's expenses.
3.c. Encountered Issues	List all the issues you faced during the experiments (solved and unsolved). Present solutions or workarounds for overcoming them. Highlight also the issues that may have to be dealt with in future experiments. While trying to understand the significance of various hyperparameters of a Random Forest algorithm, could not understand the mechanism for hyperparameter max_features with its values 'sqrt, 'log2' even after going through various online material. Therefore, plan to further analyse and study this hyperparameter that could increase the diversity of Decision Trees in the Random Forest and help reduce overfitting.

---

4. FUTURE EXPERIMENT
Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.

4.a. Key Learning	Reflect on the outcome of the experiment and list the new insights you gained from it. Provide rationale for pursuing more experimentation with the current approach or call out if you think it is a dead end. Unlike the Decision Tree model from the 3rd experiment, the Random Forest Classifier model, which employed fine-tuned hyperparameters, was effective in mitigating the problem of overfitting. It appears that the use of cross-validation has assisted to capture the underlying patterns more effectively, promoted generalization of the model, and helped to reduce the issue of overfitting. As indicated in section 2.c Modeling, plan to evaluate the performance of the Random Forest Classifier model by testing additional hyperparameters such as criterion='gini', max_features='sqrt', min_samples_split, min_weight_fraction_leaf, and max_leaf_nodes to determine if these could enhance the performance of the model. Additionally, would try to employ and test the model with cross-validation which could promote generalization of the model in the future machine learning experiments.
4.b. Suggestions / Recommendations	Given the results achieved and the overall objective of the project, list the potential next steps and experiments. For each of them assess the expected uplift or gains and rank them accordingly. If the experiment achieved the required outcome for the business, recommend the steps to deploy this solution into production. Would perform below two task. * Build the Random Forest Classifier model by utilizing further hyperparameters, such as criterion='gini', max_features='sqrt', min_samples_split, min_weight_fraction_leaf, and max_leaf_nodes, and assess whether the model's performance improves. * Understand the Random Forest Classifier's hyperparameter max_features with its values 'sqrt, 'log2' in depth. To successfully deploy a binary classification algorithm in a production environment, it is advisable to carry out below steps. - Scale and transform the model to handle large datasets. - Select an appropriate deployment environment whether cloud-based or on-premises. - Modify the model to suit production settings while complying with various security, ethical, and privacy guidelines. - Conduct testing and monitoring of the deployed model. - Periodically updated the model with new data. - Provided documentation for usage. - Review the performance of the model and retrain the model as needed.