

EXPERIMENT REPORT

| | |
|---------------------|---|
| Name | Monali Patil |
| Project Name | Customer-Vehicle-Repurchase-Prediction |
| Deliverables | Model name: Random Forest Classifier Notebook name: MachineLearning_RandomForest_Exp5.ipynb Project Repo: https://github.com/monalipatil/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: By leveraging the Feature Importance information obtained, building the Random Forest Classifier model alongside fine-tuned hyperparameters derived in the 4th experiment with the aim to determine if the model's performance can be improved by utilizing only the features that contribute the most to the model's predictions.

| | |
|----------------------------------|--|
| | <p>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.</p> |
| 1.c. Experiment Objective | <p>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.</p> <p>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.</p> <p>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.</p> <p>The possible scenarios resulting from this experiment include following:</p> <ul style="list-style-type: none"> • 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

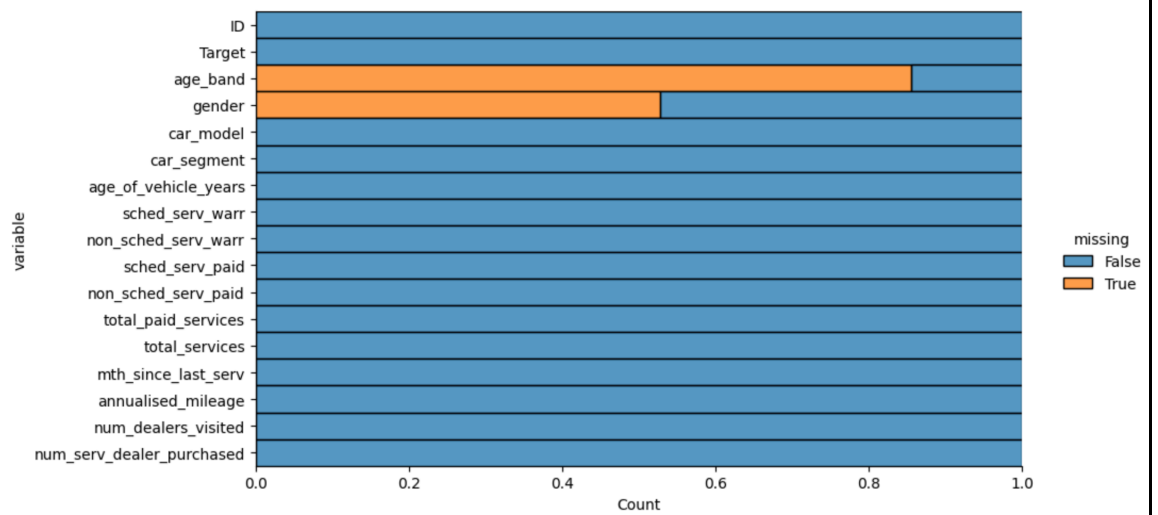


Figure 1: Missing values present in all attributes of the dataset.

* 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

The categorical features namely 'car_model' and 'car_segment' had the smallest impact on the model's prediction, so they were excluded, and no data transformation is required as the dataset contains all the numerical features.

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.

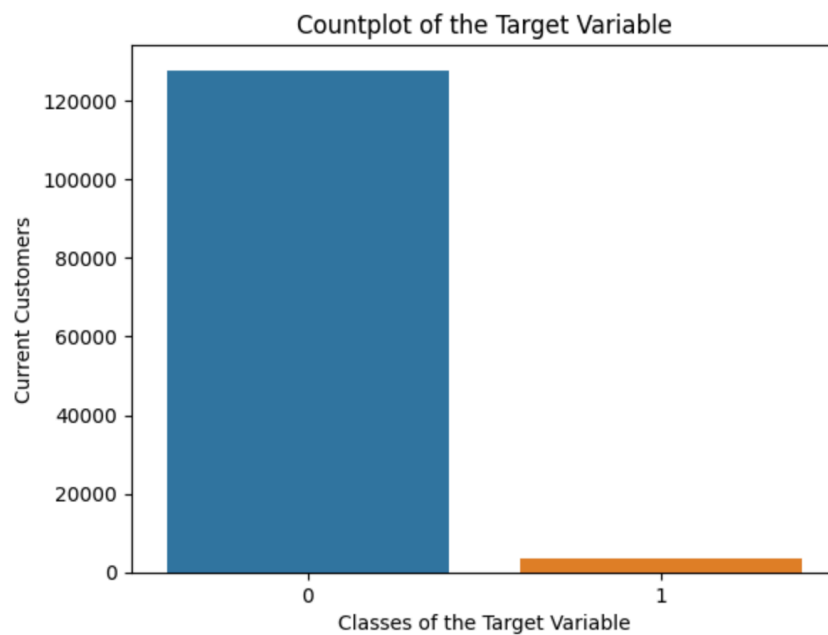


Figure 2: Examining the quantity of each class of the target variable from the actual dataset.

* 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.

* 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

Utilizing the Feature Importance information derived from 4th experiment, developing the Random Forest Classifier model (alongside fine-tuned hyperparameters obtained in 4th experiment) and evaluating the performance to investigate if the performance could be enhanced by applying only features that are contributing the most to the model's predictions. The model's performance is assessed and compared 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 |

* Utilizing above listed fine-tuned hyperparameters obtained from the 4th experiment for the Random Forest classifier model and evaluating the model's performance by considering only the most significant features that were identified as contributing to the prediction from the 4th experiment.

* 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.

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.

Among all the hyperparameters available for the Random Forest Classifier algorithm, could not analyse which are most critical hyperparameters which would enhance the model's performance due to time constraints.

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.

- 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: class_weight='balanced', criterion='entropy', max_depth=9, max_features='log2', min_samples_leaf=13, n_estimators=91, random_state=19

Performance Metrics:

| Dataset | weighted F1 Score | binary F1 Score |
|-------------------|-------------------|-----------------|
| Cross-validation: | 0.9563 | 0.4672 |
| Testing: | 0.9548 | 0.4783 |

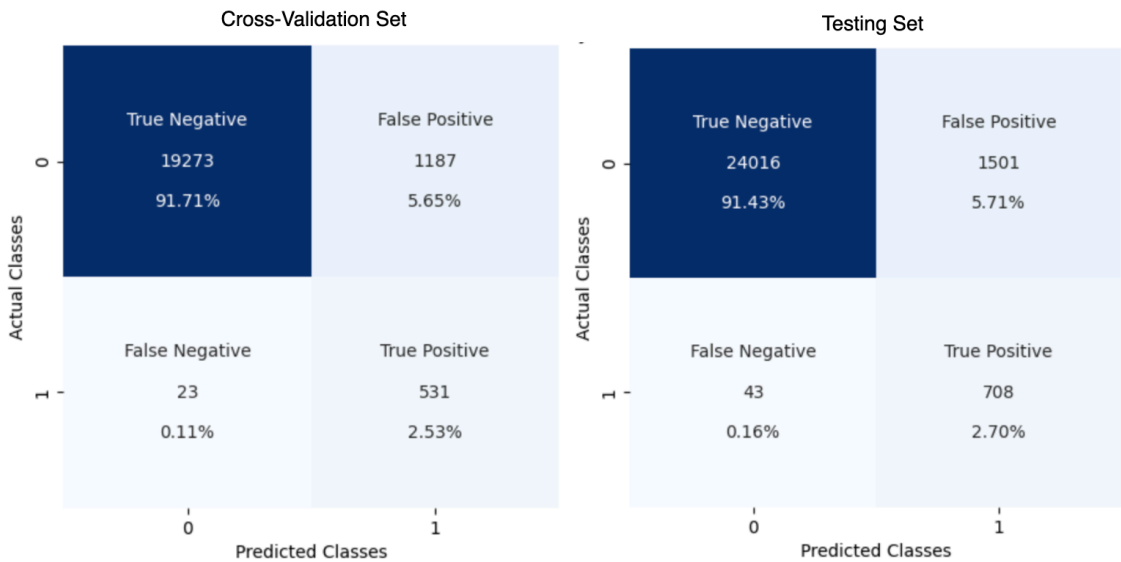


Figure 4: Confusion Matrix of Random Forest Classifier model with hyperparameter tuned values.

| | |
|--------------------------------|---|
| | <p>* The Random Forest classifier model employing only significant features shows an improvement to some extent in the binary F1 Score compared to the Random Forest classifier model from the 4th experiment. Additionally, the weighted F1 Score also has a small boost when compared to the previous model.</p> <p>* With the selection of the Important Features, the binary F1 Score representing the classification that belongs to positive class scores 0.4672 and 0.4783 for the Training-Validation and Testing respectively, indicating that the model is performing almost consistently across all sets.</p> <p>* Similarly, the weighted F1 Score for the Training-Validation and Testing of the model is 0.9563 and 0.9548, respectively illustrating that overall, the model is performing constantly on all the sets.</p> <p>* Additionally, the False Negative errors from the confusion matrix, with 23 for Training-Validation and 43 for Testing which are lower and almost identical to the False Negative errors from the 4th experiment (21 and 42 for training-validation and testing set respectively).</p> |
| 3.b. Business Impact | <p>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)</p> <p>By utilizing only significant features that are contributing most to the prediction of target classes, the model's performance has enhanced to some extent.</p> <p>The binary F1 Score of 0.4672 for training-validation and 0.4783 for testing has improved indicating that the model is correctly identifying potential customers for buying a new vehicle while misclassifying some negative instances as positive.</p> <p>The False Negative errors are low, with 23 for Training-Validation and 43 for Testing which indicates that the model is misclassifying less amount of customers as not interested in purchasing a new vehicle while in actual those who are interested.</p> |
| 3.c. Encountered Issues | <p>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.</p> <p>In this particular experiment, missed to specify the random_state hyperparameter and as a result, different outcomes were obtained during the process. Therefore, it is crucial to define this hyperparameter when conducting machine learning experiments.</p> |

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.

| | |
|--|---|
| <p>4.a. Key Learning</p> | <p>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.</p> <p>By leveraging the Feature Importance information and utilizing only the features that contribute the most to the model's predictions, the Random Forest Classifier model employing fine-tuned hyperparameters illustrates the improved performance.</p> <p>Even though the features car_model and car_segment seemed to contain important information about the type and category of vehicles that customers had bought, they had the lowest significance and contributed the least to predicting the target classes.</p> <p>Had to experiment and evaluate the model's performance by sampling the data, but due to time limitations, this could not be executed. So would like to test sampling and analyse which are most critical hyperparameters to verify if the model's performance is enhanced.</p> |
| <p>4.b. Suggestions / Recommendations</p> | <p>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.</p> <p>Intension is to conduct below two task.</p> <ul style="list-style-type: none"> * Experiment with sampling. * Analyse which hyperparameters of the Random Forest classifier are the most critical, to determine if the model's performance can be improved. <p>To successfully deploy a binary classification algorithm in a production environment, it is advisable to carry out below steps.</p> <ul style="list-style-type: none"> - 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. |