EXPERIMENT REPORT

Student Name	Shaqran Bin Saleh
Project Name	Classification Models
Date	26/4/2024
Deliverables	Notebook Name : 36106_AT2_25010238_experiment_4.ipynb model name : Random Forest Classifier

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	The goal of this project is to build a predictive model with accurate likeliness of repurchasing a car for customers. The model's predictions can be used in a myriad of ways. The model can be used to segment customers by repurchase likelihood, focusing marketing efforts on high-potential features of the dataset for efficient resource use. Insights from the data and trained model can also enhance customer retention strategies and optimize inventory management. Additionally, the results would provide feedback to refine sales tactics and customer interactions, continually improving approaches to meet customer needs effectively. Accurate predictions would mean enhance efficiency in targeting the right customers in reducing marketing efforts and higher sales. Through proper data exploration and trained model insights proper understanding of customer needs and help improve customer satisfaction. Inaccurate repurchase likelihood prediction by the model might result in diminished business opportunities, reduced ROI, wasted marketing budget, and unhappy clients. Customers may become disconnected from a brand due to improper targeting, which could be perceived as spam and harm the company's reputation.
1.b. Hypothesis	I wish to test the hypothesis that several characteristics: Gender, The model of vehicle, The type of vehicle, Age of their last vehicle, Number of scheduled services used under warranty, Number of non-scheduled services used under warranty, Amount paid for scheduled services, Amount paid for nonscheduled services, Amount paid in total for services, Total number of services, The number of months since the last service, Annualized vehicle mileage, Number of different dealers visited for servicing, Number of services had at the same dealer where the vehicle was purchased have significant effects on the customers' likeliness of repurchasing a vehicle. I like to get the answers of few questions like "What are the primary repurchasing variables for customers and how do they interact to influence their repurchasing likelihood?" By testing this hypothesis and answering the accompanying question, we can acquire a deeper understanding of the factors that influence customers' likelihood of repurchasing.

1.c. Experiment Objective	From this experiment we will be deploying a predictive model using Random Forest Classifier model that accurately handles the data and make accurate predictions. The goal is to identifying key variable effecting the customers' likelihood of repurchasing, developing an accurate model, apply it on the dataset and find the accuracy of the model basing upon the different hyperparameter tuning.

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

For proper data preparation, the following steps were taken:

1. The assigned .csv file was loaded into the notebook as a dataframe for the experiment.
2. The dimensions (rows, columns) of the data loaded from the .csv file was found. This step was performed so that we can properly understand the amount of data we are working with. It also gives us information about the number of features present in the data.
3. We have 17 features among which 1 is our target variable which is “Target”. We explored all the features of the dataset.
4. We check all the features if there are any null values and handle the situation.
5. We check the statistics (count, mean, standard deviation, minimum values, maximum value) of the features.
6. We drop the rows containing missing values from the dataset.
7. We drop the rows containing duplicate data.
8. Since we have 4 features containing categorical data. We introduced OneHotEncoder through which we transform the data and concatenate the new columns into our dataset. And finally remove the previous columns on which the transformation was done.
9. The feature “Target” is our target variable so we separate it as our ‘y’ variable and the rest were are considering our independent variable which
10. Then we split the data to three categories. Which are training data, testing data and validation data.
11. We distribute the data firstly into two segments Train + Validation and Test sets. The training + validation set would hold 80% of the data and rest 20% is testing data.
12. Then we again split the Train + Validation to separate Training and Validation sets. The validation would hold 30% of the Train + Validation data, meaning it hold 24% of the original data. The rest of the 56% of the original data is reserved for the training dataset.
13. And finally, we performed feature scaling

All the above steps were vital in this experiment is also vital for the next experiments.

2.b. Feature Engineering

While performing data preparation, we found out that the features : 'age_band', 'gender', 'car_model', 'car_segment' are categorical and are all important in the influence of the target variable. So, we had to encode these features. We achieved this encoding using the OneHotEncoder()

As we concatenated the new columns, we made sure to drop the columns on which the transformation was performed.

2.c. Modelling

For the experiment to predict customer repurchase likelihood, the Random Forest Classifier model was chosen.

Random Forest Classifier is straightforward to implement and understand.

- Random forests exhibit reduced susceptibility to overfitting compared to single decision trees. This is achieved by an ensemble approach, where the final prediction is derived by averaging the outputs from multiple, independently constructed decision trees. This strategy provides robustness and generalizability of the model by mitigating the influence of variance.
- This method demonstrates the capability to process high-dimensional datasets containing a large number of variables. It achieves this without the need for variable deletion, thereby preserving information and potentially leading to a superior level of accuracy.
- It provides a clear and direct assessment of the relative significance of each feature for the model's predictions. Understanding these feature importance is essential for interpreting the model's decision-making process, especially in exploratory contexts where the goal is to gain insights into the underlying relationships within the data..
- Random Forest constructs a multitude of decision trees, effectively mitigating overfitting and reducing variance. This approach yields a solid prediction model with enhanced accuracy..

The decision to use Random Forest Classifier should be aligned with the specific characteristics of the data and the business or research objectives.

While performing the experiment hyperparameters tuning was done quite some time. The hyperparameter which were tuned were: n_estimator, max_depth, min_samples_leaf.
The following values for n_estimator were tested: 200, 50.
The following value for max_depth was tested: 15
The following value for min_samples_leaf were tested: 2, 10.

The hyperparameters you've tuned for your Random Forest model are crucial for controlling the model's complexity and its ability to generalize to new data
The hyperparameter n_estimator specifies the number of decision trees included in the random forest model. While increasing the number of trees generally enhances the model's stability and accuracy, it comes at the expense of greater processing time.
The hyperparameter max_depth is basically the maximum depth of the tree. With limiting the depth of the tree we can prevent overfitting but again setting it too low will hamper the models ability to learn complex patterns.
The hyperparameter min_samples_leaf is the minimum number of samples a leaf node can have. There is a trade-off between the minimum number of samples required for a leaf node in a decision tree model and the model's generalization performance. A higher minimum number of samples reduces variance and can lead to improved ability to perform well on unseen data but may introduce bias. Conversely, a lower minimum allows the model to capture more detail from the training data but increases the risk of learning patterns specific to the training data that do not generalize well which is overfitting.
For future we plan on exploring Extra Trees.

14. 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

The performance metrics used in this experiment are :
1. Accuracy Score
2. Precision Score
2. F1 Score
3. Confusion Matrix

We used the Random Forest Classifier model and trained it multiple times as we performed multiple Hyperparameter tuning.

The results of the metrics according to different hyperparameters are as follow:

1. For default hyperparameters:
   - Accuracy Scores:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - F1 Scores:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - Precision Score:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

2. For Hyperparameter n_estimator =50:
   - Accuracy Scores:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - F1 Scores:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - Precision Score:
     a. On Training set: 1.0
     b. On Validation set: 1.0
     c. On Testing set: 1.0
   - Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

3. For Hyperparameters n_estimator =200:

- Accuracy Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- F1 Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Precision Score:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

4. For Hyperparameters n_estimator =50, max_depth=15:

- Accuracy Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Precision Score:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- F1 Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

5. For Hyperparameters n_estimator =50, max_depth=15, min_samples_leaf=2:

- Accuracy Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Precision Score:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- F1 Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

6. For Hyperparameters n_estimator =50, max_depth=15, min_samples_leaf=10:

- Accuracy Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Precision Score:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- F1 Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

7. For the best hyperparameters after grid search, n_estimator =30, max_depth=10, min_samples_leaf=2:

- Accuracy Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Precision Score:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- F1 Scores:
  a. On Training set: 1.0
  b. On Validation set: 1.0
  c. On Testing set: 1.0
- Confusion Matrix:

```
[[25608    0]
 [  0   660]]
```

Main Issues and Causes:

- The consistently perfect performance on all metrics (accuracy, precision, F1 score, confusion matrices) raises concerns about potential data leakage. This occurs when information from the test set inadvertently influences the model during training, leading to overly optimistic performance evaluations.
- The lack of performance variation with changes in hyperparameters like number of estimators, maximum tree depth, and minimum samples per leaf suggests a potentially simplistic dataset. Highly distinctive patterns within the features might be making classifications trivially easy for the model, regardless of its complexity
- Perfect scores on all sets suggest either very homogeneous data (lacking challenge) or a few exceptionally strong features dominating classification across all data splits.
- The lack of performance change with adjustments to maximum depth and minimum samples per leaf indicates potentially very simple decision boundaries within the data. This suggests that even the simplest decision trees (low complexity) can perfectly separate the classes without error, regardless of tree structure.

3.b. Business Impact

According to our objective which we set earlier which was to deploy a predictive Random Forest Classifier model that performs well. We can come the following conclusions about the results of this experiment:

- For most configurations, the model achieved perfect scores (1.0) for accuracy, precision, and F1 score across the training, validation, and test sets. This suggests that the model is highly effective at classifying the given data under these configurations.
- Despite changes in hyperparameters, the confusion matrix remains consistent with no false positives or negatives, which initially suggests exceptional model performance.
- While the model achieves impressive perfect scores, this consistency across datasets and hyperparameters raises concerns. This level of flawless performance is uncommon for real-world data, suggesting potential issues like overfitting or data leakage.
- Two possible explanations exist for the model's indifference to hyperparameter adjustments. Either the model might be inherently robust, performing well across various configurations. Or, the dataset itself might be so simple that even basic models can achieve high accuracy, making hyperparameter tuning less impactful.

And the impacts the business might face due to imperfect results would be:

- If the model is deployed in this critical decision-making business, consistent inaccuracies or a failure to perform as expected could lead to loss of trust among stakeholders. Because even though perfect scores across all datasets might be initially impressive but are suspicious and could suggest overfitting or data leakage.
- Missed opportunities due to incorrect model predictions can be significant. For instance, in a marketing campaign of this business, failing to correctly identify potential repurchasing customers (false negatives) could mean missed revenue opportunities.

3.c. Encountered Issues

As we performed this experiment, we were encountered with the following issues:

- Our experiment identified a potential case of overfitting, despite achieving perfect accuracy, the model's performance might be misleading. It could be memorizing the training data instead of learning to handle unseen examples.
- The lack of performance variation with hyperparameter changes suggests the data itself might dictate model behavior, rendering these parameters less influential.
- Random Forests are slow to train, particularly with many trees or complex structures.

1. 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

Reflecting on the outcomes and performance of the Random Forest Classifier model experiments for predicting customer repurchase likelihood, several key insights emerge.

- While the Random Forest model achieved perfect scores across all hyperparameter settings, this outcome requires further investigation. It's crucial to determine if the results reflect the model's true ability or potential issues like data leakage or overly simplistic data.
- The model's performance remains unchanged despite adjustments to key hyperparameters. This suggests either a simple dataset or that these parameters have minimal impact in this scenario.
- The model's consistent production of perfect results suggests a potential over-generalization issue. To gain a clearer picture of the model's limitations and real-world applicability, evaluating it on more complex or noisy datasets is crucial. This will reveal how well the model generalizes to unseen data beyond the current, potentially simplistic, scenario.
- The findings suggest a need to explore other machine learning models. This could help in finding a model that might be more sensitive to underlying patterns.

Given the insights and the issues encountered, here are the reasons to continue experimenting with the Random Forest Classifier:

- To assess the true robustness of our Random Forest and its susceptibility to overfitting, we can introduce controlled noise and variability into the data. This will simulate real-world scenarios and help us develop a model that performs well not only under ideal conditions, but also when faced with unexpected variations.
- While the current hyperparameter settings achieve perfect results, further experimentation is crucial to identify truly impactful settings and understand which parameters influence performance most significantly across various data scenarios.

4.b. Suggestions / Recommendations

The experimentation with the Random Forest Classifier for predicting customer repurchase likelihood identified several pathways for further improvement. These refinements can be categorized into two main phases: model optimization and deployment preparation.

1 Model Optimization:

- Given a potential class imbalance (significantly fewer repurchasing customers), consider these approaches to improve model performance for the minority class:
  - i) Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be employed for balancing to create synthetic data points for the under-represented class (repurchasing customers).
  - ii) Assigning higher weights to the minority class during training can encourage the model to prioritize its accurate classification.

2 Deployment Preparation:

Following the optimization phase, a series of steps will ensure a successful deployment of the chosen model:

1. Once all testing and validation efforts are complete, the model configuration demonstrating the best performance will be selected for deployment.

Given the current model's seemingly perfect performance, exploring a simpler or pruned Random Forest variant holds promise.

a) Simpler models often require fewer resources and compute power, leading to faster execution. This is crucial when considering deployment and scaling the model for real-world applications.
b) While the current model achieves perfect scores, a simpler version might still deliver comparable performance, especially if the data is indeed very simple or dominated by a few strong features.

2. A comprehensive plan outlining the integration of the chosen model into the existing IT infrastructure should be formulated.

3. A monitoring system should be established to track the model's performance in a production environment using real-world data. This will be accompanied by a maintenance schedule for periodically updating the model to maintain its effectiveness.

4. Training materials and comprehensive documentation should be developed to educate end-users on the model's functionality, its intended use within business processes, and its limitations.

5. The deployment process should adhere to all relevant compliance regulations and data privacy rules, ensuring the integrity and security of customer data.

By prioritizing model optimization followed by a well-defined deployment plan, we aim to maximize the predictive power of the customer repurchase prediction model while guaranteeing a smooth and effective transition to real-world applications.