# EXPERIMENT REPORT

| | |
|---|---|
| **Student Name** | Shaqran Bin Saleh |
| **Project Name** | Classification Models |
| **Date** | 26/4/2024 |
| **Deliverables** | Notebook Name : 36106_AT2_25010238_experiment_5.ipynb<br>model name : Extra Trees 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.<br><br>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.<br><br>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.<br><br>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:<br>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<br>have significant effects on the customers' likeliness of repurchasing a vehicle.<br><br>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 Extra Trees Classifier model that accurately handles the data and make accurate predictions.<br><br>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()<br><br>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 Extra Trees Classifier model was chosen.<br><br>Extra Trees Classifier is straightforward to implement and understand.<br>• Extra Trees builds a "forest" of numerous decision trees. Crucially, these trees are constructed with randomness, meaning each tree uses different features and random split points. This diversity prevents overreliance on specific patterns in the training data and allows the model to adapt better to unseen data<br>• Traditional decision trees can overfit the training data, memorizing noise or irrelevant details. Extra Trees combats this by injecting randomness into the tree-building process<br>• Extra Trees leverages the "random forest" concept to provide valuable insights into feature importance. By analyzing how each feature contributes to the final prediction across the entire ensemble of trees, it identifies which features are most influential in predicting the target variable.<br><br>The decision to use Extra Trees Classifier should be aligned with the specific characteristics of the data and the business or research objectives.<br><br>While performing the experiment hyperparameters tuning was done. We took the help of Random Search CV. Through its estimator we found out the best values for the hyperparameters.<br>The hyperparameter which were tuned were: n_estimator, max_depth, min_samples_leaf, max_features. The best values for the hyperparamters was:<br>The n_estimator: 155<br>The max_depth: 6<br>The min_samples_leaf: 2<br>The max_features: 2<br><br>The hyperparameters we tuned for your Extra Trees model are crucial for controlling the model's complexity and its ability to generalize to new data<br>• 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.<br>• 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.<br>• 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.<br>• The max_features determines the number of features to consider when looking for the best split during the construction of each tree in the forest. |

## 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 Extra 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: 0.9897560975609756
     b. On Validation set: 0.9895311788802913
     c. On Testing set: 0.9882577826324412
   - F1 Scores:
     a. On Training set: 0.0
     b. On Validation set: 0.0
     c. On Testing set: 0.0
   - Precision Score:
     a. On Training set: 0.0
     b. On Validation set: 0.0
     c. On Testing set: 0.0
   - Confusion Matrix:
     [[3619 0]
     [ 43 0]]

2. For the best hyperparameters found from Grid Search ('max_depth': 6, 'max_features': 2, 'min_samples_leaf': 2, 'n_estimators': 155 ):
   - Accuracy Scores:
     a. On Training set: 0.9897560975609756
     b. On Validation set: 0.9895311788802913
     c. On Testing set: 0.9882577826324412
   - F1 Scores:
     a. On Training set: 0.0
     b. On Validation set: 0.0
     c. On Testing set: 0.0
   - Precision Score:
     a. On Training set: 0.0
     b. On Validation set: 0.0
     c. On Testing set: 0.0
   - Confusion Matrix:
     [[3619 0]
     [ 43 0]]

| | |
|---|---|
| | Main Issues and Causes:<br>• Our analysis revealed a concerning discrepancy between the model's overall accuracy and its performance on the minority class. While the model achieved high accuracy across all data splits (training, validation, testing), the F1 score, precision, and recall scores for the minority class were all zero. This suggests an "accuracy paradox": the model excels at predicting the majority class but fails entirely at identifying instances of the minority class.<br>• Further investigation through the confusion matrix confirmed this issue. Both the default and best hyperparameter settings resulted in a matrix with the following structure: [3619, 0; 43, 0]. This indicates a complete absence of true positives (correct predictions) for the minority class(actual repurchasing customers). All 43 minority class instances were misclassified as the majority class.<br>• A significant class imbalance is evident within the data. This disparity can lead to a model biased towards the majority class. Such a model might simply predict the majority class for all inputs, neglecting the presence of the minority class altogether.<br>• Despite the high accuracy reported, the model's complete inability to identify true positives raises concerns about overfitting. This suggests the model might not be capturing the underlying patterns crucial for class distinction. Overfitting on the majority class data is a potential culprit for this behavior. |
| **3.b. Business Impact** | According to our objective which we set earlier which was to deploy a predictive Extra Trees Classifier model that performs well. We can come the following conclusions about the results of this experiment:<br><br>• The model exhibits a concerning bias towards the majority class. This is evident from the high overall accuracy coexisting with zero values for precision, recall, and F1-score on the minority class. While excelling at predicting frequent outcomes, the model fails to identify the less frequent, potentially more critical ones.<br>• The employed Grid Search hyperparameter tuning did not address the class imbalance issue. Despite applying both default and tuned hyperparameters, the confusion matrices and metric scores for the minority class remained unchanged. This suggests that the tuning process did not tackle the underlying problem.<br>• The consistent failure to detect the minority class compels to explore alternative training strategies. These might include:<br>Assigning higher weights to the minority class instances during training can incentivize the model to focus on learning from these rarer examples.<br>Techniques like oversampling or undersampling the data can help balance the class distribution and improve the model's performance on the minority class.<br><br>And the impacts the business might face due to imperfect results would be:<br>• 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.<br>• 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:<br><br>• The data exhibits a significant class imbalance, with a considerably larger majority class compared to the minority class. This disparity led the model to prioritize predicting the majority class, resulting in zero precision, recall, and F1-score for the minority class.<br>• The hyperparameter tuning strategy failed to address the issue of class imbalance. Both the default and best tuned hyperparameter settings provided identical confusion matrices and metric scores for the minority class. This implies that the tuning process was unable to rectify the model's core limitations in handling imbalanced data.<br>• The primary reliance on accuracy as the evaluation metric misled the assessment of the model's performance. While the model achieved high accuracy by overwhelmingly predicting the majority class, it completely failed to identify the minority class correctly. |

---

| **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.<br><br>• The experiment emphasized the significant influence of class imbalance on model behavior. When not addressed, even robust models like Extra Trees can become biased towards the majority class, neglecting the crucial minority class entirely.<br>• While hyperparameter tuning is a valuable tool, it may have limitations in overcoming fundamental issues like data imbalance or inherent model biases.<br>• This work underlined the potential pitfalls of relying solely on accuracy, especially for imbalanced datasets. The experiment demonstrated the need for more informative metrics like precision, recall, F1-score, and the confusion matrix to provide a comprehensive and realistic assessment of model performance across all classes.<br><br>Given the insights and the issues encountered, here are the reasons to continue experimenting with the Extra Trees Classifier:<br><br>• Compared to other tree-based models, Extra Trees exhibits a natural resistance to overfitting.<br>• The model offers a multitude of hyperparameters (e.g., max_depth, min_samples_leaf, n_estimators) that remain open to further experimentation.<br>• Extra Trees boasts computational efficiency, particularly when dealing with large datasets. This stems from its method of constructing trees using random splits, making it an attractive choice in such scenarios compared to many other ensemble methods. |

| | |
|---|---|
| **4.b. Suggestions / Recommendations** | The experimentation with the Extra Trees 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.<br><br>1 Model Optimization:<br><br>i) Moving beyond Grid Search, more sophisticated techniques like randomized search or Bayesian optimization can be employed to identify optimal settings for hyperparameters like max_depth, min_samples_leaf, max_features, and n_estimators. These advanced algorithms explore the hyperparameter space more efficiently, potentially leading to superior model performance<br>ii) Given the class imbalance issue, various strategies to balance the class distribution and improve minority class prediction should be investigated. These techniques include:<br>(1) Synthetic Minority Over-Sampling Technique (SMOTE) artificially generates synthetic data points for the minority class, effectively increasing its representation within the training data.<br>(2) Directly assigning higher weights to minority class instances during training incentivizes the model to prioritize learning from these rarer examples.<br>(3) Reducing the number of majority class instances can help counteract its dominance and allow the model to focus on the minority class.<br><br>2. Deployment Preparation:<br><br>Following the optimization phase, a series of steps will ensure a successful deployment of the chosen model:<br><br>1. Once all testing and validation efforts are complete, the model configuration demonstrating the best performance will be selected for deployment.<br><br>Given the current model's seemingly perfect performance, exploring a simpler or pruned Random Forest variant holds promise.<br><br>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.<br>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.<br><br>2. A comprehensive plan outlining the integration of the chosen model into the existing IT infrastructure should be formulated.<br><br>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.<br><br>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.<br><br>5. The deployment process should adhere to all relevant compliance regulations and data privacy rules, ensuring the integrity and security of customer data.<br><br>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. |