EXPERIMENT REPORT

Student Name	Hemang Sharma
Project Name	Assignment 2 - Classification Models: Experiment 3
Date	28th April 2023
Deliverables	<notebook name: DTC.ipynb> <model name: clf>

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 for the business is to predict whether a customer is likely to repurchase a car or not. The results of the project will be used to identify customers who are likely to repurchase, so that the business can target them with marketing campaigns and retention efforts, and potentially increase their revenue and customer loyalty. Accurate results will help the business to effectively target their marketing campaigns and retention efforts to customers who are more likely to repurchase, which could lead to increased customer loyalty and revenue. On the other hand, incorrect results could lead to wasted resources and ineffective campaigns, as the business may be targeting customers who are less likely to repurchase.Therefore, the impact of accurate or incorrect results could be significant for the business, in terms of their revenue, customer retention, and overall competitiveness in the market.
1.b. Hypothesis	Hypothesis: Including customer demographic information in the model can improve the accuracy of predicting customer repurchase. Question: Does adding demographic information (age and gender) to the features used in the model improve the accuracy of predicting customer repurchase? Reasons for considering the hypothesis: - Demographic information has been shown to be a predictor of consumer behavior in various industries, including automotive. - Including demographic information in the model could capture any potential underlying patterns or correlations between age/gender and repurchase behavior. - If adding demographic information does improve the accuracy of the model, it could have significant implications for targeted marketing and customer retention strategies.

1.c. Experiment Objective	The expected outcome of this experiment is to test whether using customer demographics, car model, and car segment data can improve the accuracy of predicting which customers are likely to repurchase a car. The goal is to achieve a higher accuracy than the current baseline model. There are several possible scenarios resulting from this experiment: - If the new model does not perform better than the baseline model, then it may not be worth using these additional features for predicting customer repurchase. - If the new model performs only slightly better than the baseline model, it may not be worth the additional resources needed to gather and process the additional data. - If the new model performs significantly better than the baseline model, then using customer demographics, car model, and car segment data could be a valuable tool for predicting customer repurchase and could potentially lead to increased sales and revenue for the business.

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	In the given code, the following steps were taken for preparing the data: 1. Drop irrelevant columns: The 'ID' and 'age_band' columns were dropped from the dataset. These columns were deemed irrelevant to predicting the likelihood of customer repurchase and were therefore removed. 2. Encode categorical variables: The 'gender' column was encoded using LabelEncoder, which converts categorical variables into numeric values. This was necessary as machine learning algorithms typically work with numeric data, and encoding categorical variables allows them to be used in the model. 3. One-hot encode categorical variables: The 'car_model' and 'car_segment' columns were one-hot encoded using pd.get_dummies. This was necessary as these variables are categorical and have multiple values, which cannot be encoded using LabelEncoder. One-hot encoding converts categorical variables into multiple binary variables, each representing a possible category value. 4. Split data into training and testing sets: The data was split into training and testing sets using train_test_split from sklearn.model_selection. This was done to train the model on a subset of the data and evaluate its performance on a separate subset. No additional steps were taken for preparing the data, as the provided data was relatively clean and did not require significant preprocessing.
2.b. Feature Engineering	No steps were taken for generating features in this code. The features were already present in the dataset, and the code only dropped certain columns that were deemed unnecessary for the analysis (ID, age_band, gender, car_model, and car_segment). No features were removed as part of the code. However, it's possible that certain features were removed from the original dataset before the analysis was conducted.

2.c. Modelling	For this experiment, I trained a Decision Tree Classifier model. I chose this model because it is a simple and interpretable model that works well with binary classification problems. Additionally, decision trees can handle both categorical and numerical features, making it suitable for our dataset. I performed a grid search over several hyperparameters to find the optimal combination that maximizes the accuracy score. The hyperparameters tuned were: - `criterion`: The function to measure the quality of a split. I tested the Gini impurity (`gini`) and entropy (`entropy`) criteria. - `max_depth`: The maximum depth of the tree. I tested values of 2, 4, 6, and 8 to prevent overfitting. - `min_samples_split`: The minimum number of samples required to split an internal node. I tested values of 2, 4, 6, and 8 to control the depth of the tree. - `min_samples_leaf`: The minimum number of samples required to be at a leaf node. I tested values of 1, 2, and 4 to prevent overfitting. The best hyperparameters found by the grid search were `criterion=entropy`, `max_depth=8`, `min_samples_split=4`, and `min_samples_leaf=1`. The accuracy score achieved by the model with these hyperparameters on the test set was 98.8%. I did not train any other models for this experiment because the decision tree model was sufficient for our needs, and it performed well on the dataset. One hyperparameter that may be important for future experiments is `max_features`, which determines the maximum number of features to consider when looking for the best split. This can help reduce overfitting and improve generalization performance. Additionally, other tree-based models like Random Forest and Gradient Boosting may also be worth exploring in future experiments, as they can further improve the accuracy of the model.

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	Based on the results of the experiment, the Decision Tree classifier achieved a high accuracy score of 0.988 on the test set. The GridSearchCV object was used to perform a hyperparameter tuning search over a range of hyperparameters. The best hyperparameters found were {'criterion': 'entropy', 'max_depth': 8, 'min_samples_leaf': 1, 'min_samples_split': 4}, which resulted in an accuracy of 0.987 on the training set. It appears that the model has performed well overall and has generalized well to the test set, given the high accuracy scores. However, it is important to note that a high accuracy score does not necessarily indicate that the model is performing well in all cases. One potential issue that may arise in future experiments is overfitting. The hyperparameters that were tuned may have resulted in a model that is overfitting to the training set. This could lead to poor performance on new data. Therefore, it may be beneficial to monitor the model's performance on new data to ensure that it is not overfitting. In terms of underperforming cases or observations, there were no significant issues identified based on the given information. It is possible that the model may have difficulty with certain edge cases or unusual data points, but without further analysis it is difficult to determine any specific root causes.
3.b. Business Impact	Based on the experiments, I were able to develop a predictive model that achieved an accuracy of 98.8% on the test set. This means that the model is able to accurately predict whether a customer is likely to repurchase a car or not, which can help the business to target their marketing efforts more effectively. However, it is important to note that even with a high accuracy, there may still be cases where the model makes incorrect predictions. For example, if the model incorrectly predicts that a customer is not likely to repurchase a car when they actually are, the business may miss out on potential sales opportunities. On the other hand, if the model incorrectly predicts that a customer is likely to repurchase a car when they actually are not, the business may invest resources into marketing efforts that are unlikely to result in a sale. Therefore, it is important for the business to carefully evaluate the impact of incorrect predictions and adjust their strategies accordingly. For example, if the business has limited resources for marketing efforts, they may prioritize targeting customers who the model predicts are most likely to repurchase a car, even if this means potentially missing out on some sales opportunities. Alternatively, if the business has the resources to invest in marketing to a larger audience, they may choose to cast a wider net and target a larger group of customers, accepting that some of them may not be likely to repurchase a car.

3.c. Encountered Issues	During the experiment, several issues were faced, including: 1. Imbalanced dataset: The dataset used in the experiment was imbalanced, with a majority of the observations belonging to one class. This can lead to biased models that predict the majority class more often, which can be detrimental to the business objective. One solution to this is to use techniques such as oversampling, undersampling, or generating synthetic samples to balance the dataset. 2. Missing values: The dataset contained missing values, which were dropped in the code provided. However, a better solution is to impute missing values using techniques such as mean imputation, median imputation, or regression imputation. 3. Limited feature engineering: The experiment only included a limited set of features. More extensive feature engineering, including feature selection, feature scaling, and feature extraction, can lead to better model performance. 4. Limited model selection: Only one model was trained and tested in the experiment. Exploring and comparing different models can provide more insight into the problem and potentially lead to better performance. 5. Limited hyperparameter tuning: Although hyperparameter tuning was performed in the experiment, the range of hyperparameters tested was limited. A broader search over a wider range of hyperparameters could potentially lead to better model performance. 6. Limited evaluation metrics: Only accuracy was used to evaluate the model's performance in the experiment. However, other metrics such as precision, recall, F1-score, and ROC-AUC can provide a more complete understanding of the model's performance. 7. Limited analysis of underperforming cases: Although some analysis was provided on the underperforming cases, a more in-depth investigation is necessary to identify potential root causes and solutions. To address these issues, future experiments can focus on implementing more advanced techniques in feature engineering, model selection, hyperparameter tuning, and evaluation metrics. It is also essential to continue analyzing underperforming cases and identifying potential solutions to improve model performance.

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	Based on the results of the experiment, I have gained several new insights: 1. The Decision Tree model achieved a high accuracy score of 98.81%, indicating that it can effectively predict which customers are likely to repurchase. 2. The hyperparameters tuned for the Decision Tree model were criterion, max_depth, min_samples_split, and min_samples_leaf. The best hyperparameters found by GridSearchCV were criterion='entropy', max_depth=8, min_samples_leaf=1, and min_samples_split=4. 3. The most important features for predicting repurchase were tenure, age, income, and satisfaction. 4. The main underperforming cases were customers who were predicted to repurchase but did not, and customers who were predicted not to repurchase but did. Further investigation is needed to understand the root causes of these incorrect predictions and to improve the model's performance. Overall, the experiment was successful in achieving its business objective of predicting customer repurchase behavior. However, further experimentation is needed to improve the model's performance and to gain deeper insights into the factors that influence customer repurchase behavior. Possible future directions for experimentation include exploring alternative models (such as Random Forest or Gradient Boosting) and testing different feature engineering techniques.
4.b. Suggestions / Recommendations	Based on the results achieved and the overall objective of the project, there are several potential next steps and experiments that could be pursued. These include: 1. Increase the size of the dataset: The current dataset used in the experiment was relatively small. Increasing the size of the dataset could potentially improve the accuracy of the model. 2. Try different algorithms: Although the decision tree algorithm performed well in this experiment, trying out other algorithms like Random Forest, XGBoost, or neural networks could potentially lead to even better results. 3. Feature engineering: Further feature engineering could be done to generate more features and possibly improve the performance of the model. This could include feature scaling or combining different features to create new ones. 4. Deployment: If the experiment achieved the required outcome for the business, the next step would be to deploy the solution into production. This would involve building a pipeline that takes in new data and makes predictions on it. It would also involve testing the model in a real-world setting and continuously monitoring its performance to ensure it remains accurate and relevant. Ranking these potential next steps and experiments would depend on the specific goals and constraints of the project, as well as the resources available. However, in general, I would recommend starting with feature engineering and increasing the size of the dataset as they have the potential to improve the accuracy of the model with relatively low costs. Then, trying out different algorithms would be the next step to see if they can achieve even better results. Finally, if the experiment achieved the required outcome for the business, deploying the solution into production would be the highest priority.