EXPERIMENT REPORT

|  |  |
| --- | --- |
| Student Name | Balakumaran Sivanesan |
| Project Name | Assignment 2 - Classification Models: Experiment 3 |
| Date | 28th April 2024 |
| Deliverables | <notebook name: DTC.ipynb>  <model name: dt\_model> |

|  |  |
| --- | --- |
| 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 | This study aims to forecast a customer's likelihood of making another car purchase for the company. The project's outcomes will be utilized to determine which consumers are most likely to make additional purchases, allowing the company to focus its marketing campaigns and customer retention initiatives on them in an effort to boost sales and foster client loyalty.  Precise outcomes will enable the company to focus its marketing campaigns and customer retention initiatives on those who are most likely to make another purchase, thus increasing client loyalty and income. On the other side, since the company might be focusing on clients who are less likely to make repeat purchases, inaccurate data could result in resource waste and unsuccessful campaigns. As a result, the company may have substantial effects from accurate or inaccurate outcomes in terms of revenue, client retention, and general market competitiveness. |
| 1.b. Hypothesis | The accuracy of forecasting a client's likelihood to make another purchase may be increased by adding consumer demographic data to the model.  Question: Does the model's ability to forecast customer repurchase accuracy increase with the addition of demographic data, such as age and gender, to its feature set?  Reasons for considering the hypothesis:   * It has been demonstrated that demographic data can predict customer behavior in a number of industries, including the automobile * By incorporating demographic data into the model, it may be possible to identify any underlying trends or associations between gender/age and repurchase behavior. * Adding demographic data may have a substantial impact on client retention and targeted marketing techniques if it increases the model's accuracy. |
|  |  |
| 1.c. Experiment  Objective | The goal of this project is to determine whether forecasting which consumers are likely to repurchase a car can be made more accurate by utilizing data on car models, auto segments, and customer demographics. The objective is to surpass the accuracy of the existing baseline model.  There are several possible scenarios resulting from this experiment:   * Using these extra variables to forecast customer repurchase may not be worthwhile if the new model does not outperform the baseline model. * The extra resources required to collect and evaluate the extra data might not be worthwhile if the new model only marginally outperforms the baseline model. * Utilizing customer demographics, vehicle model, and auto segment data could be a useful tool for forecasting consumer repurchase and could potentially result in higher sales and revenue for the company if the new model outperforms the baseline model. |

|  |  |
| --- | --- |
| 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. Remove unnecessary columns from the dataset: The 'ID' and 'age band' columns were removed. These columns were eliminated since it was determined they had no bearing on forecasting the possibility of a consumer making another purchase. 2. Encode category variables: Label Encoder, which transforms categorical data into numeric values, was used to encode the 'gender' column. This was required since categorical variables can only be included in the model by encoding them, whereas machine learning algorithms usually operate on numerical data. 3. One-hot encoding of categorical variables: pd.get\_dummies was used to one-hot encode the 'car\_model' and 'car\_segment' columns. This was required since Label Encoder could not encode these variables because they are category and have numerous values. Categorical data are transformed into several binary variables, each of which represents a potential category value, using one-hot encoding. 4. Divide the data into training and testing sets: Train\_test\_split from sklearn.model\_selection was used to divide the data into training and testing sets. In order to train the model on a subset of the data and assess its performance on another subset, this was done.   Since the supplied data was comparatively clean and didn't require a lot of preprocessing, no extra steps were performed to prepare it |
| 2.b. Feature  Engineering | In this code, no efforts were made to generate features. The code simply removed the columns ID, age\_band, gender, car\_model, and car\_segment that weren't thought necessary for the study because the attributes were included in the dataset.   The code did not eliminate any functionalities. It is plausible, therefore, that certain traits were eliminated prior to the analysis being carried out from the initial dataset. |
| 2.c. Modelling | I trained a Decision Tree Classifier model just for this experiment. This model is easy to understand and applies well to binary classification problems, which is why I chose it. Furthermore, decision trees are appropriate for our dataset since they can handle both numerical and categorical information.   To determine which combination of hyperparameters would optimize the accuracy score, I ran a grid search across a number of them. The adjusted hyperparameters 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 5 to prevent overfitting. * `min\_samples\_split`: The minimum number of samples required to split an internal node. I tested values of 2 and 5 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 to prevent overfitting.   The grid search yielded the following optimal hyperparameters: {criterion=entropy}, {max\_depth=5}, {min\_samples\_split=2}, and {min\_samples\_leaf=1}. With these hyperparameters, the model's accuracy score on the test set was 99%.  The decision tree model worked well on the dataset and was adequate for our purposes, therefore I didn't train any more models for this experiment.    The hyperparameter {max\_features}, which establishes the maximum number of features to take into account when determining the optimal split, is one that might be significant for further research. By doing so, generalization performance can be enhanced and overfitting can be decreased. Furthermore, as they can further increase the model's accuracy, other tree-based models like Random Forest and Gradient Boosting would be worthwhile to investigate in subsequent studies. |

|  |  |
| --- | --- |
| 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 | The Decision Tree classifier obtained a high accuracy score of 0.988 on the test set, based on the experiment findings. A hyperparameter tuning search was conducted over a variety of hyperparameters using the GridSearchCV object. With an accuracy of 0.987 on the training set, the optimal hyperparameters were {'criterion': 'entropy','max\_depth': 8,'min\_samples\_leaf': 1,'min\_samples\_split': 4}.  Given the excellent accuracy ratings, it looks that the model has performed well overall and has well-generalized to the test set. It is crucial to remember that a high accuracy score does not always mean that the model is operating effectively.  Overfitting is one such problem that could surface in subsequent research. It's possible that the configured hyperparameters produced a model that overfits the training set. This can cause bad results when using fresh data. Consequently, to make sure the model is not overfitting, it could be helpful to keep an eye on how well it performs with fresh data.  Based on the provided information, no noteworthy problems related to underperforming cases or observations were found. It's probable that the model won't work well in some edge circumstances or with odd data sets, but it's hard to pinpoint any particular core causes without more investigation. |
| 3.b. Business Impact | I was able to create a prediction model based on the experiments, and it was 98.8% accurate on the test set. This indicates that the model can estimate a customer's likelihood of repurchasing an automobile with accuracy, which can assist the company in more successfully focusing their marketing efforts.  It's crucial to remember that despite a high accuracy, the model could occasionally still produce inaccurate predictions. For example, the company can lose out on possible sales possibilities if the model forecasts wrongly that a customer is not likely to repurchase an automobile when in fact they are. However, if the model forecasts erroneously that a consumer is likely to repurchase an automobile when they are not, the company can waste money on marketing campaigns that are unlikely to generate a sale.  As a result, it is critical that the company thoroughly assess the consequences of inaccurate projections and modify their plans as necessary. For instance, if a company's marketing budget is tight, it might decide to focus more on attracting clients who the model indicates are more likely to make another automobile purchase, even if doing so means passing up some possible sales chances. Alternatively, the company may decide to cast a wider net and target a greater set of customers, acknowledging that some of them might not be likely to buy an automobile, if they have the resources to engage in marketing to a larger audience. |
| 3.c. Encountered  Issues | During the experiment, several issues were faced, including:   1. Imbalanced dataset: The experiment's dataset was unbalanced, with most observations falling into a single class. This may result in skewed models that more frequently anticipate the majority class, which could be harmful to the corporate goal. Using methods to balance the dataset, such as oversampling, undersampling, or creating synthetic samples, is one way to address this. 2. Missing values: The provided code eliminated the missing values included in the dataset. Imputation of missing values via methods like regression, mean, or median imputation, on the other hand, is a preferable approach. 3. Restricted feature engineering: There were just a few features in the experiment. Improved model performance may result from more thorough feature engineering, which includes feature extraction, feature scaling, and feature selection. 4. Limited model selection: In the experiment, a single model was trained and evaluated. Examining and contrasting several models can help shed light on the issue and possibly improve performance. 5. Restricted hyperparameter tuning: While the experiment included hyperparameter tuning, the range of hyperparameters studied was constrained. Improved performance of the model might result from a more comprehensive search across a larger range of hyperparameters. 6. Restricted evaluation metrics: In the experiment, the model's performance was assessed solely based on accuracy. Nevertheless, additional measures like ROC-AUC, F1score, precision, recall, and recall might offer a more thorough picture of the model's performance. 7. Limited research of underperforming cases: While some analysis of the underperforming cases was given, a more thorough study is required to find potential reasons and remedies.   Future research can concentrate on applying more sophisticated methods in feature engineering, model selection, hyperparameter tuning, and assessment metrics in order to address these problems. To enhance model performance, it is also critical to keep examining underperforming scenarios and coming up with viable fixes. |

|  |
| --- |
| 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. With a high accuracy score of 98.81%, the Decision Tree model demonstrated its ability to accurately forecast which customers are most likely to make repeat purchases 2. The Decision Tree model's hyperparameters, criterion, max\_depth, min\_samples\_split, and min\_samples\_leaf, were adjusted. Max\_depth=8, min\_samples\_leaf=1, min\_samples\_split=4, and criterion='entropy' were the optimal hyperparameters discovered by GridSearchCV 3. The key variables for repurchase prediction were satisfaction, tenure, age, and income. 4. Predicted customers who did not repurchase and predicted non-repurchase customers who did were the primary underperforming situations. To enhance the model's performance and comprehend the underlying reasons of these inaccurate forecasts, more research is required.   All things considered, the trial was effective in reaching its business goal of forecasting customer repurchase behavior. To enhance the model's functionality and have a better understanding of the elements influencing customers' repurchase behavior, more research is necessary. Future research could look into investigating alternate models (such Random Forest or Gradient Boosting) and experimenting with various feature engineering methods. |
| 4.b. Suggestions / Recommendations | Several different experiments and following steps could be pursued based on the project's ultimate goal and the data obtained. Among them are:   1. Expand the dataset: The experimental dataset utilized in this study was comparatively small. The accuracy of the model may be enhanced by expanding the dataset's size 2. Experiment with various algorithms: Random Forest, XGBoost, or neural networks are some examples of other algorithms that might produce even better results than the decision tree algorithm, which did well in this experiment. 3. Feature engineering: To produce more features and perhaps enhance the model's performance, more feature engineering could be done. This could involve integrating many features to create new ones or scaling features. 4. Deployment: The solution would be put into production if the experiment produced the desired results for the company. This would include developing a pipeline that receives fresh data and forecasts it. In order to make sure the model stays accurate and relevant, it would also need to be tested in a real-world environment and its performance would need to be regularly monitored.   The particular objectives and limitations of the project, together with the available resources, would determine how these possible tests and next steps should be ranked. All things considered, I would generally advise beginning with feature engineering and expanding the dataset since they have the ability to raise the model's accuracy at a comparatively cheap cost. The next stage would be to experiment with various algorithms to see if they could produce even better outcomes. Ultimately, the first goal would be to put the solution into production if the experiment produced the desired results for the company. |