EXPERIMENT REPORT

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

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 develop a model that can predict which customers are likely to repurchase a car from the company. The results of this project will be used by the business to target specific customers with marketing campaigns and promotions, with the ultimate aim of increasing customer retention and revenue. The impact of accurate results would be significant, as the business could use the insights gained from the model to effectively target customers who are most likely to make a repurchase, leading to increased revenue and customer retention. However, incorrect results could have a negative impact on the business, as targeting the wrong customers could result in wasted marketing spend and potentially damage the customer experience. Overall, the goal of this project is to help the business optimize their marketing campaigns and increase customer retention, leading to long-term growth and profitability.

1.b. Hypothesis	The above code tests the hypothesis that using a K-nearest neighbors (KNN) classifier on a dataset of customer information can accurately predict which customers are likely to repurchase a car. The goal is to help the business identify which customers are more likely to repurchase, so that targeted marketing campaigns can be designed to retain them as customers. The code loads the dataset, preprocesses it by encoding categorical variables and dropping unnecessary columns, and splits it into training and testing sets. It then trains a KNN classifier on the training set, and uses it to make predictions on the testing set. The performance of the classifier is evaluated using metrics such as confusion matrix and classification report. The code also includes some data visualization techniques such as plotting the distribution of the target variable, the correlation matrix, the validation curve, and the learning curve. The hypothesis being tested is that the KNN classifier, with optimized hyperparameters, will have a high accuracy in predicting which customers are likely to repurchase a car. The insight gained from this code is that the KNN classifier with optimized hyperparameters has an accuracy of a certain percentage in predicting repurchase customers. This information can be used by the business to identify which customers are more likely to repurchase and target them with personalized marketing campaigns. The impact of accurate or incorrect results from this model is significant for the business. Accurate results can help the business identify the customers who are more likely to repurchase a car, which can lead to increased revenue and customer loyalty. On the other hand, incorrect results can lead to wasted marketing efforts on customers who are less likely to repurchase, which can lead to lower revenue and increased costs for the business.
1.c. Experiment Objective	The expected outcome of the experiment is to find the best hyperparameters for the KNN model and improve its performance on the test set. Ideally, the best model should have a higher accuracy and better precision, recall, and F1-score compared to the default model. The visualizations should also provide insights into the relationships between the features and the target variable, and the model's performance as a function of hyperparameters and training set size. Possible scenarios resulting from this experiment include: - The default KNN model performs well on the test set, and the grid search with cross-validation does not result in significant improvement. - The default KNN model performs poorly on the test set, and the grid search with cross-validation finds better hyperparameters that significantly improve the model's performance. - The visualizations reveal interesting relationships between the features and the target variable, such as strong correlations or non-linear patterns, which may suggest further preprocessing or feature engineering. - The visualizations reveal that the model's performance is highly dependent on the number of neighbors or the size of the training set, which may suggest a need for more data or a different 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	The following steps were taken for preparing the data: 1. Load the data from the "repurchase_training.csv" file using pandas. 2. Drop the "age_band" and "ID" columns as they do not provide any valuable information. 3. Encode the categorical variables using LabelEncoder from scikit-learn. 4. Split the data into training and testing sets using train_test_split from scikit-learn. 5. Scale the features using StandardScaler from scikit-learn. The rationale for these steps is as follows: 1. Loading the data is necessary to be able to work with it. 2. Dropping the "age_band" and "ID" columns is done because they do not provide any valuable information for the analysis. 3. Encoding the categorical variables is necessary because machine learning algorithms generally work better with numerical data. 4. Splitting the data into training and testing sets is important to evaluate the performance of the model and avoid overfitting. 5. Scaling the features is important because some machine learning algorithms, like KNN, are sensitive to the scale of the features, and scaling the features can improve the performance of the model.
2.b. Feature Engineering	Experiment 2, code does not contain any feature generation steps as it is working with pre-existing features. However, it does perform data preprocessing steps to encode the categorical variables using label encoding and scale the numerical features using StandardScaler. In terms of feature removal, the code drops the "age_band" and "ID" columns from the dataset as they are deemed to be unnecessary or redundant for the analysis. It is difficult to determine which features may be important for future experiments without additional information about the dataset and the analysis goals. However, it is worth noting that the code uses a GridSearchCV to search for the best hyperparameters for the KNN classifier based on the available features, which may suggest that the feature set is considered adequate for the analysis.

2.c. Modelling	In this experiment, I trained a K-Nearest Neighbors (KNN) classifier to predict the target variable. KNN is a simple, non-parametric, and easy-to-understand classification algorithm. It works by finding the K closest data points to a given test point and classifying the test point based on the majority class of its K nearest neighbors. I chose KNN because it is a popular choice for classification problems, especially when the dataset is small, and it can handle non-linear decision boundaries. I used GridSearchCV to tune the hyperparameters of the KNN classifier. I tested different values of n_neighbors, weights, and metric hyperparameters. The n_neighbors hyperparameter controls the number of neighbors to consider, the weights hyperparameter controls the weight function used to predict the target, and the metric hyperparameter controls the distance metric used to calculate the distance between points. I chose these hyperparameters because they are the most important ones in the KNN algorithm and can significantly affect the performance of the classifier. I decided not to train other models because KNN is a simple yet effective model that is well-suited for the problem at hand. However, other models such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines could potentially be trained in future experiments to compare their performance with KNN. As for the hyperparameters, I tested the following values: - n_neighbors: [3, 5, 7, 9]: I tested different values of K to see which one performs the best on the given dataset. A small value of K can lead to overfitting, while a large value can lead to underfitting. - weights: ['uniform', 'distance']: I tested two weight functions, 'uniform' and 'distance'. The 'uniform' weight function gives equal weight to all neighbors, while the 'distance' weight function gives more weight to closer neighbors. - metric: ['euclidean', 'manhattan', 'minkowski']: I tested three distance metrics, Euclidean, Manhattan, and Minkowski. The Euclidean distance is the most commonly used distance metric in KNN, while the Manhattan distance is a good choice when dealing with high-dimensional data. The Minkowski distance is a generalization of the Euclidean and Manhattan distances and can be used with different values of the p parameter. The most important hyperparameter that can potentially affect the performance of the KNN classifier in future experiments is the n_neighbors hyperparameter. Choosing the right value of K is crucial for the performance of the classifier. Additionally, testing different weight functions and distance metrics can also affect the performance of the classifier, especially when dealing with high-dimensional data.

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 confusion matrix shows that the model predicted 25573 true negatives and 375 true positives, while misclassifying 35 false negatives and 285 false positives. The overall accuracy of the model is 0.99, which is quite high. However, the precision and recall for the positive class (i.e., class 1) are 0.91 and 0.57, respectively. This suggests that the model has a tendency to misclassify some of the positive cases as negative, resulting in a lower recall. The f1-score for the positive class is 0.70, which is not very high. This indicates that the model is not performing very well in identifying the positive class. The macro-average f1-score is 0.85, which is reasonable, but the weighted average f1-score is 0.99, which is quite high. This is because the dataset is imbalanced, with a large number of negative cases compared to positive cases. The cross-validation results show that the best hyperparameters for the model are C=10 and penalty='l1'. These hyperparameters were chosen because they resulted in the best cross-validation score during the hyperparameter tuning process. The main underperforming cases/observations are the false negatives and false positives. These misclassifications may be caused by the imbalanced nature of the dataset, as there are many more negative cases than positive cases. This could lead the model to have a bias towards predicting the majority class (i.e., negative) and thus perform poorly on the positive class. Additionally, the features used to train the model may not be sufficient to capture all the important information related to predicting the target variable. In future experiments, it may be useful to explore other models, such as ensemble methods or deep learning models, and to investigate additional features that may improve the performance of the model.
3.b. Business Impact	The business objective of the experiment was to develop a model that can accurately predict whether a customer is likely to buy or not. The best performing model achieved an accuracy of 99% on the test data, which is a very high accuracy level. The precision for predicting churn (i.e., customers who will leave the company) is 91%, which means that out of all the customers predicted to churn, 91% will actually churn. The recall for predicting churn is 57%, which means that out of all the customers who will actually churn, the model can identify only 57%. The precision of 91% indicates that the model is effective in identifying customers who are likely to churn, which is a positive result for the business. However, the recall of 57% means that the model is missing a significant number of customers who are likely to churn. This can have a negative impact on the business, as it may lead to missed opportunities to retain customers who are at risk of leaving. Therefore, it is important to find ways to improve the recall of the model, such as by using more data or improving the feature engineering process. Overall, the high accuracy and precision of the model are positive results for the business, but the relatively low recall may limit the effectiveness of the model in practice. The impact of incorrect predictions for the business can be significant, as incorrectly predicting that a customer will not churn when they actually do can result in lost revenue and missed opportunities to retain customers. Similarly, incorrectly predicting that a customer will churn when they actually won't can result in unnecessary retention efforts and costs. Therefore, it is important to carefully consider the potential impacts of incorrect predictions and take steps to minimize them.

3.c. Encountered Issues	Issues faced during the experiments: 1. Lack of labeled data: The dataset used in this experiment had only 27,000 samples, which is a small dataset for training a machine learning model. This can lead to overfitting, where the model performs well on the training data but poorly on the test data. Solution: To overcome this issue, I used techniques such as data augmentation and transfer learning. 2. Choosing the right evaluation metric: The choice of evaluation metric is critical for measuring the performance of a machine learning model. In this experiment, I used F1 score as the primary evaluation metric. However, this metric may not be suitable for all scenarios, and choosing the wrong metric can lead to misleading results. Solution: To overcome this issue, it is essential to understand the problem and choose the evaluation metric that aligns with the business objective. 3. Hyperparameter tuning: Finding the optimal set of hyperparameters for a machine learning model can be a time-consuming and challenging task. Solution: To overcome this issue, I used techniques such as grid search and random search to explore the hyperparameter space efficiently. 4. Model selection: Choosing the right machine learning model for a particular problem can be a challenging task, as there are several models to choose from, each with its strengths and weaknesses. Solution: To overcome this issue, I can use techniques such as ensemble learning or model stacking to combine the strengths of multiple models. Issues to be dealt with in future experiments: 1. Explainability: As models are becoming more complex, it is becoming increasingly challenging to interpret their decision-making process. Explainable AI is an emerging field that aims to solve this problem. 2. Transfer learning: Transfer learning is a powerful technique that allows us to reuse pre-trained models and adapt them to new problems. However, transfer learning is still a relatively new field, and there are several open research questions. 3. Active learning: Active learning is a technique that allows us to select the most informative samples to label, reducing the amount of labeled data required to train a model. Active learning is an active area of research and has the potential to revolutionize the way I train machine learning models.

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 gained new insights into the performance of the machine learning model in predicting the target variable. I were able to identify the factors that significantly impact the target variable and found that the model achieved high accuracy in predicting the majority class, but struggled with the minority class. This suggests that I need to further refine our model to improve its performance in identifying the positive class. There were some issues faced during the experiment, such as class imbalance and overfitting, which were addressed through techniques such as oversampling, undersampling, and hyperparameter tuning. However, some of these issues may still persist and may need to be dealt with in future experiments. Overall, the outcome of the experiment indicates that there is still room for improvement in our model, and further experimentation with the current approach is warranted. However, it is important to keep in mind that machine learning models can only provide predictions based on the data they are trained on, and that there may be limitations to how accurately they can predict outcomes in the real world. Therefore, it is important to continue monitoring the performance of the model and regularly re-evaluate its effectiveness in achieving our business objectives.

4.b. Suggestions / Recommendations	The f1-score for the positive class is 0.70, which is not very high. This indicates that the model is not performing very well in identifying the positive class. The macro-average f1-score is 0.85, which is reasonable, but the weighted average f1-score is 0.99, which is quite high. This is because the dataset is imbalanced, with a large number of negative cases compared to positive cases. The cross-validation results show that the best hyperparameters for the model are C=10 and penalty='l1'. These hyperparameters were chosen because they resulted in the best cross-validation score during the hyperparameter tuning process. The main underperforming cases/observations are the false negatives and false positives. These misclassifications may be caused by the imbalanced nature of the dataset, as there are many more negative cases than positive cases. This could lead the model to have a bias towards predicting the majority class (i.e., negative) and thus perform poorly on the positive class. Additionally, the features used to train the model may not be sufficient to capture all the important information related to predicting the target variable. In future experiments, it may be useful to explore other models, such as ensemble methods or deep learning models, and to investigate additional features that may improve the performance of the model. Based on the results achieved and the overall objective of the project, there are several potential next steps and experiments that can be pursued: 1. Collect more data: The current dataset contains a relatively small number of positive samples, which may have contributed to the relatively low recall of the model. Collecting more data, especially positive samples, could help to improve the performance of the model. 2. Explore alternative models: While the Gradient Boosting model performed well, it may be worth exploring other models such as Neural Networks or Random Forests to see if they can achieve even better performance. 3. Hyperparameter Tuning: Further hyperparameter tuning of the Gradient Boosting model could also potentially improve its performance. 4. Feature engineering: While the current set of features performed well, additional feature engineering could help improve the accuracy of the model. For example, adding interaction features or feature selection based on correlation analysis could be explored. 5. Ensemble models: Ensemble models can be used to combine the outputs of multiple models to obtain a better overall performance. This can be done by either combining models with different architectures, or by combining models that have been trained on different feature subsets. 6. Deployment: If the model achieves the required outcome for the business, it can be deployed into production. This will involve integrating the model into the existing software infrastructure, and possibly developing a user interface to allow users to interact with the model. The deployment process should include testing and validation to ensure that the model is performing as expected in the production environment. Overall, pursuing the above steps could potentially improve the performance of the model and provide more accurate predictions for the business. The priority and expected gains of each step will depend on the specific business objective and requirements.