EXPERIMENT REPORT

Student Name	Shaqran Bin Saleh
Project Name	Classification Models
Date	26/4/2024
Deliverables	Notebook Name : 36106_AT2_25010238_experiment_1.ipynb model name : KNN 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 KNN (k-nearest neighbors) 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 K-Nearest Neighbors (KNN) classifier was chosen.

KNN is straightforward to implement and understand, making it an accessible choice for initial modeling efforts. It is well-known for its effectiveness in classification scenarios, especially when the relationship between feature variables and the class is not linear.
As KNN does not assume any underlying distribution for the data, it is suitable for real-world datasets that often do not adhere to theoretical distributions.

While performing the experiment hyperparameters Tuning was done quite some time. The following hyperparameters were tuned:

- Number of Neighbors (n_neighbors):
  Values Tested: 3, 10, 20, 40
  The number of neighbors significantly impacts the prediction accuracy and generalization. Fewer neighbors can make the model sensitive to noise in the data (overfitting), whereas too many can smooth over important distinctions between classes (underfitting).
- Distance Metric (metric):
  Values Tested: Euclidean, Manhattan
  Different metrics can perform better based on the dataset geometry. Euclidean distance is common and effective in many cases, but Manhattan distance can be more suitable for high-dimensional data as it measures the distance linearly.

For future we plan on exploring Decision Tree, Random Forest, Extra Trees and also Support Vector Classifier.

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 KNN 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: 0.9924878048780488
     b. On Validation set: 0.9906690942193901
     c. On Testing set: 0.991807755324959
   - F1 Scores:
     a. On Training set: 0.4460431654676259
     b. On Validation set: 0.2545454545454546
     c. On Testing set: 0.46428571428571425
   - Precision Score:
     a. On Training set: 0.9117647058823529
     b. On Validation set: 0.7777777777777778
     c. On Testing set: 1.0
   - Confusion Matrix:
```
[[3619    0]
 [  30   13]]
```

2. For Hyperparameters 3 neighbors and Euclidean Distance:
   - Accuracy Scores:
     a. On Training set: 0.995219512195122
     b. On Validation set: 0.9915794264906691
     c. On Testing set: 0.9923539049699618
   - Precision Score:
     a. On Training set: 1.0
     b. On Validation set: 0.8
     c. On Testing set: 0.9411764705882353
   - F1 Scores:
     a. On Training set: 0.6956521739130436
     b. On Validation set: 0.39344262295081966
     c. On Testing set: 0.5333333333333333
   - Confusion Matrix:
```
[[3618    1]
 [  27   16]]
```

3. For Hyperparameters 10 neighbors and Euclidean Distance:
   - Accuracy Scores:

     a. On Training set: 0.9911219512195122

     b. On Validation set: 0.9906690942193901

     c. On Testing set: 0.9893500819224468

   - 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:

     ```
     [[3619    0]
      [  39    4]]
     ```

4. For Hyperparameters 20 neighbors and Euclidean Distance:
   - Accuracy Scores:

     a. On Training set: 0.9901463414634146

     b. On Validation set: 0.9897587619481111

     c. On Testing set: 0.9890770070999454
   - 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: 0.07339449541284404

     b. On Validation set: 0.042553191489361694

     c. On Testing set: 0.13043478260869565
   - Confusion Matrix:

     ```
     [[3619    0]
      [  40    3]]
     ```

5. For Hyperparameters 20 neighbors and Manhattan Distance:
   - Accuracy Scores:

     a. On Training set: 0.9900487804878049

     b. On Validation set: 0.9897587619481111

     c. On Testing set: 0.9893500819224468

   - 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: 0.05555555555555556

     b. On Validation set: 0.042553191489361694

     c. On Testing set: 0.1702127659574468
   - Confusion Matrix:

     ```
     [[3619    0]
      [  39    4]]
     ```

6. For Hyperparameters 40 neighbors and Manhattan Distance:

- Accuracy Scores:
  a. On Training set: 0.9897560975609756
  b. On Validation set: 0.9895311788802913
  c. On Testing set: 0.9882577826324412

- 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: 0.0
  b. On Validation set: 0.0
  c. On Testing set: 0.0
- Confusion Matrix:

```
[[3619    0]
 [  43    0]]
```

Main Issues and Causes

- Low F1 Scores with high accuracy indicates that the model is biased towards the majority class (0), failing to adequately identify the minority class (1).
- Worsening F1 Score with increased neighbors, as the number of neighbors grows, the model's sensitivity to the minority class (1) diminishes, likely due to the dilution of the minority class influence in larger neighborhoods.
- The confusion matrices consistently show very few false positives across all configurations but vary significantly in the number of false negatives, which impacts the F1 score.

3.b. Business Impact

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

- The consistently high accuracy across different hyperparameters indicates that the model generally predicts well based on the majority class (0). This might seem favorable at a glance, as it suggests a high overall rate of correct predictions.
- Despite high accuracy, the low F1 scores in many configurations, particularly on validation and test sets, suggest that the model struggles to predict the minority class effectively, which in this case could be the customers who are likely to repurchase. The model’s tendency to miss these potentially repurchasing customers due to class imbalance can be detrimental.
- The high precision in certain configurations (e.g., achieving a precision score of 1.0) indicates that when the model predicts a customer will repurchase, it is very likely correct. However, the low recall (evidenced by the low F1 score) suggests that the model fails to identify a significant number of customers who will repurchase.

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

- Missed Revenue Opportunities due to inability to identify customers (for low recall for the minority class) likely to repurchase may lead to missed chances for effective targeted marketing and, reducing potential revenue gains.
- Inefficient Resource Allocation as model misclassifications can cause misallocation of marketing resources, wasting efforts on less likely customers rather than those predisposed to repurchase.
- Customer Relationship Management as inaccurate predictions, which may happen if Bias towards the majority class happens as it fails to capture the diverse behaviors and needs of all customers, can degrade customer experience by misaligning marketing efforts with actual customer behavior and preferences.

3.c. Encountered Issues

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

- Class Imbalance was found significantly and it can lead to models that are biased towards the majority class, as indicated by high accuracy but low recall and F1 scores.
- Hyperparameter Optimization was difficult as it was difficult in choosing the optimal number of neighbors and the right distance metric, which affects model performance.
- Overfitting to Training Data which was seen as we found high performance on training data but poorer on validation or test sets.
- Feature Engineering was important as ineffective feature engineering can limit the model's ability to learn complex patterns.

2. 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 KNN classifier experiments for predicting customer repurchase likelihood, several key insights emerge.

- High accuracy but low F1 scores occurs. While the model consistently achieves high accuracy, the low F1 scores across various configurations indicate a struggle to balance precision and recall, particularly for the minority class (likely repurchasers).
  The number of neighbors and the distance metric significantly impact model performance, with lower numbers of neighbors generally providing better handling of the minority class.
- Class imbalance challenge. The class imbalance significantly affects the model's ability to predict less frequent outcomes accurately. Techniques to handle class imbalance need to be integral to the modeling process to improve minority class predictions.

Given the insights and the issues encountered, here are reasons to continue experimenting with the KNN approach:

- Adjusting hyperparameters showed potential for incremental improvements in F1 scores and handling of the minority class.
- Exploring more advanced hyperparameter optimization techniques could uncover better configurations.
- Further development and optimization of features could improve the model's predictive power, especially with domain-specific insights.

Depending on continued lack of improvement in handling the minority class, exploring other models like decision trees, ensemble methods, or even neural networks might be warranted. These models might offer better feature handling and intrinsic class balance mechanisms.

4.b. Suggestions / Recommendations

The experimentation with the KNN 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:

- Techniques for balancing the class distribution in the training data will be explored. This can potentially improve the model's ability to handle under-represented classes.

- We will investigate more advanced methods such as randomized search, grid search, or Bayesian optimization to fine-tune the KNN model's hyperparameters. These parameters include the number of neighbors considered for prediction, the distance metric used to measure similarity, and the weighting scheme applied to neighbors' influence.

- If the KNN model's performance reaches a plateau, we will consider employing alternative algorithms like Random Forest, Extra Trees. These models may be better suited to capture non-linear relationships and feature interactions within the data, potentially leading to enhanced prediction accuracy.

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.

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.