DATA SCIENCE

Clarissa Beatrice Kosasih - 2702209350 | Jeremy Djohar Riyadi - 2702219572 | Kelvin Jonathan Yusach - 2702209533 | Sherly Vaneza - 2702222163

OPTIMIZING "TRAVEL INSURANCE CLAIMS"

With Machine Learning

Problem Description

The travel insurance claim process often poses challenges for insurers due to high claim volumes and diverse influencing factors, leading to slower decision-making and higher error risks. However, customers expect a fast process. To address this, we designed a classification model leveraging historical data to predict claim outcomes based on factors such as trip duration, destination, insurance type, and more. This model aims to streamline claim processing, reduce human errors, and enhance customer service efficiency.

Dataset Description

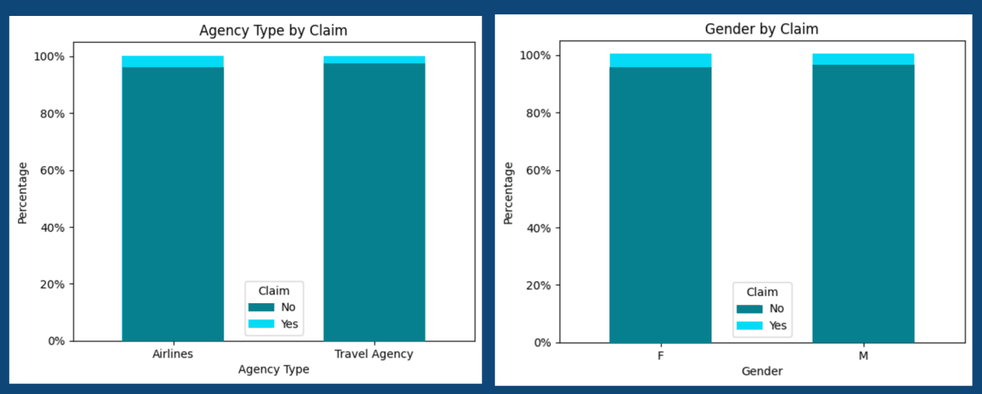
1. Insurance Agency Information: Agency, Agency Type, dan Distribution Channel
2. Insurance Product Information: Product Name
3. Insurance Holder Information: Gender dan Age
4. Travel Information: Duration dan Destination
5. Financial Information: Net Sales dan Commision
6. Claim Information: Claim

Feature Engineering and Justification

Grouping Countries by Region

The countries in the Destination column are too numerous, which may cause prediction errors in the model. Therefore, the countries will be grouped according to their regions.

Column Removal



After analyzing the distribution, it was concluded that Agency Type and Gender do not contribute significantly to the Claim column. Therefore, these columns were removed from the dataset.

Encoding

- Label Encoding for Binary Columns

Claim and Distribution Channel columns were label-encoded because they contain only two unique values. This approach was chosen over one-hot encoding to avoid generating additional columns, making the dataset more compact and easier to interpret.

- One-Hot Encoding for Categorical Columns

One-hot encoding was applied to the Agency and Region columns to prevent any unintended ordinal relationships or rankings in these categorical variables.

- Label Encoding for Ordered Data

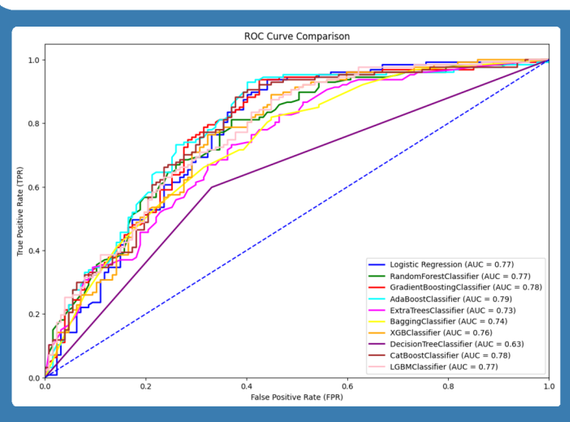
The Product Name column was label-encoded as it represents data with an inherent order. The plans were ranked based on their value or quality of service, starting with Value Plan (rank 1) as the most basic, followed by Basic Plan, and so on, up to Annual Silver Plan (rank 5) as the highest tier.

Undersampling

```
---------------------
Claim
No     16326
Yes      633
Name: count, dtype: int64
---------------------
```

We chose undersampling because the target class in the Claim column is imbalanced. Undersampling reduces the risk of overfitting to the minority class, which can occur when using oversampling techniques like SMOTE. Undersampling works by removing a portion of the majority class data, ensuring the model does not become overly focused on specific patterns while maintaining a balanced dataset for better generalization.

Model



Based on the ROC curve analysis, the two top-performing models identified are AdaBoostClassifier and CatBoostClassifier.

Evaluation Method

The evaluation of the classification model leverages a classification report and a confusion matrix to assess its performance. The classification report provides key metrics, including precision (the accuracy of positive predictions), recall (the ability to correctly identify actual positives), and F1-score (the harmonic balance of precision and recall). Additionally, the confusion matrix offers detailed insights into the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), allowing us to understand model errors and strengths in distinguishing between classes effectively.

Results

AdaBoostClassifier

Before Tuning

```
AdaBoostClassifier
Accuracy: 0.7283464566929134
Confusion Matrix:
 [[ 84  43]
 [ 26 101]]
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.66      0.71       127
           1       0.70      0.80      0.75       127

    accuracy                           0.73       254
   macro avg       0.73      0.73      0.73       254
weighted avg       0.73      0.73      0.73       254
```

After Tuning

```
Best AdaBoostClassifier
Confusion Matrix:
[[ 84  43]
 [ 26 101]]
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.66      0.71       127
           1       0.70      0.80      0.75       127

    accuracy                           0.73       254
   macro avg       0.73      0.73      0.73       254
weighted avg       0.73      0.73      0.73       254
```

CatBoostClassifier

Before Tuning

```
CatBoostClassifier
Accuracy: 0.7125984251968503
Confusion Matrix:
 [[82 45]
 [28 99]]
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.65      0.69       127
           1       0.69      0.78      0.73       127

    accuracy                           0.71       254
   macro avg       0.72      0.71      0.71       254
weighted avg       0.72      0.71      0.71       254
```

After Tuning

```
Best CatBoostClassifier
Confusion Matrix:
[[ 84  43]
 [ 23 104]]
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.66      0.72       127
           1       0.71      0.82      0.76       127

    accuracy                           0.74       254
   macro avg       0.75      0.74      0.74       254
weighted avg       0.75      0.74      0.74       254
```

After fine-tuning, the CatBoost model demonstrated superior performance compared to the AdaBoostClassifier. In the confusion matrix, CatBoost successfully improved both True Positive and True Negative values, reflecting its enhanced ability to accurately identify both classes. In contrast, AdaBoost showed no changes in its confusion matrix results before and after fine-tuning.

From the classification report, CatBoost exhibited improvements in precision, recall, and F1-score across both classes. The increase in precision and recall highlights the model's enhanced reliability in correctly detecting positive claims. Additionally, the improvement in F1-score indicates a better balance between precision and recall. Overall, CatBoost achieved higher accuracy, while AdaBoost remained at an accuracy of 0.73, with no changes to its precision, recall, or F1-score after fine-tuning.

CatBoost's advantages can be attributed to its robust mechanisms for handling overfitting, such as optimized settings for depth, the number of iterations, and regularization. Fine-tuning parameters like depth and l2_leaf_reg enabled CatBoost to effectively learn more complex data patterns. Furthermore, the model demonstrated superior generalization capabilities in managing imbalanced datasets, making it more effective at identifying positive claims with lower error rates compared to AdaBoost.

Conclusion

The CatBoostClassifier proved to be the best model for predicting travel insurance claim outcomes, surpassing AdaBoostClassifier, especially after fine-tuning. Advanced feature engineering, such as region grouping, tailored encodings, and undersampling, optimized the dataset. CatBoost achieved superior accuracy, precision, recall, and F1-scores, thanks to fine-tuned parameters like depth and regularization. Its ability to capture complex patterns and manage imbalanced data positions it as an effective solution for streamlining claims processing, reducing errors, and enhancing decision-making efficiency, ultimately improving customer service and operational performance for insurers.