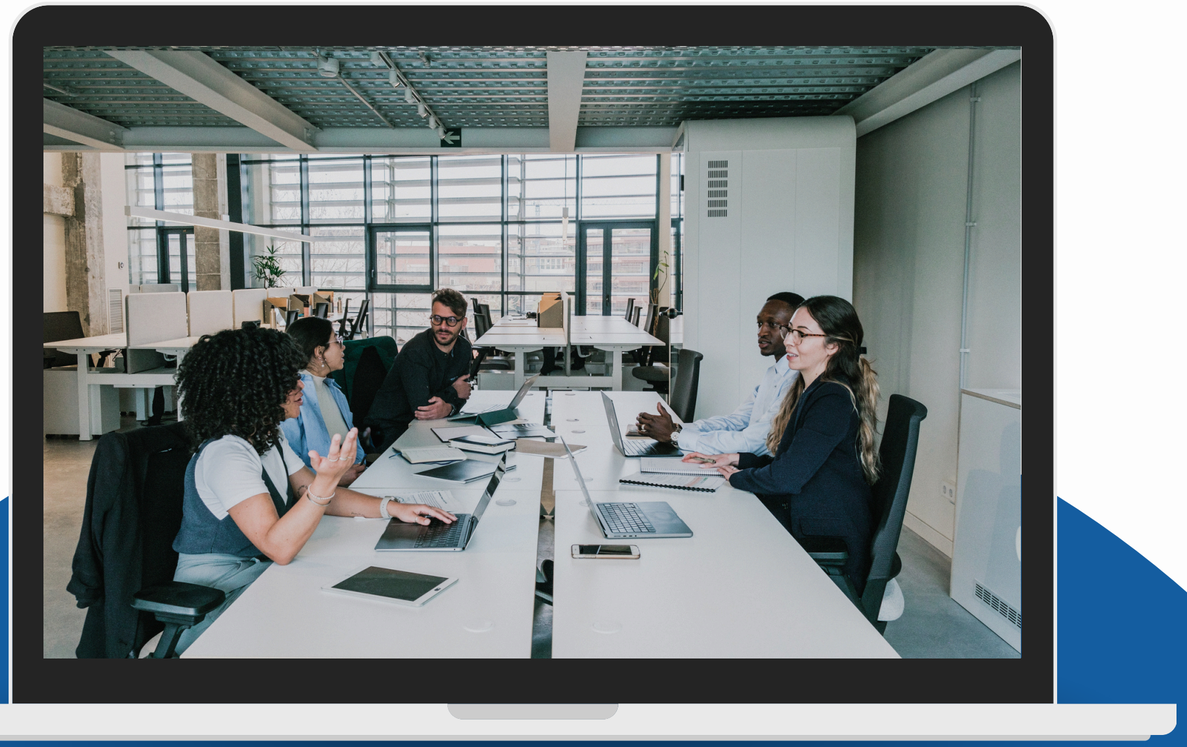
Travel Insurance

Predicting Insurance Claims for Strategic Planning

By: Ivan Robi Septian



Overview

Introduction

Travel insurance is a type of insurance that provides protection during both domestic and international trips. Several companies offer travel insurance services, including Chubb, Generali, and Sompo, among others

Benefit

- Medical protection and evacuation
- Compensation for baggage damage or loss
- Trip cancellation protection
- Coverage for accidents, death
- And many more

General Term

1. Policyholder
2. Insured
3. Premium
4. Claim
5. Coverage

Data Understanding

Attribute	Data Type, Length	Description
Agency	object	Name of agency
Agency Type	object	Type of travel insurance agencies
Distribution Channel	object	Channel of travel insurance agencies
Product Name	object	Name of the travel insurance products
Gender	object	Gender of insured
Duration	Int	Duration of travel
Destination	object	Destination of travel
Net Sales	Float	Amount of sales of travel insurance policies
Commission (in value)	Float	Commission received for travel insurance agency
Age	Int	Age of insured
Claim	Text	No – Claim status is rejected, Yes – Claim status is accepted

Objective

Develop a robust machine learning model that accurately predicts the likelihood of travel insurance insured filing claims in the future. Utilize explainable AI techniques to interpret and understand the factors influencing claim likelihood

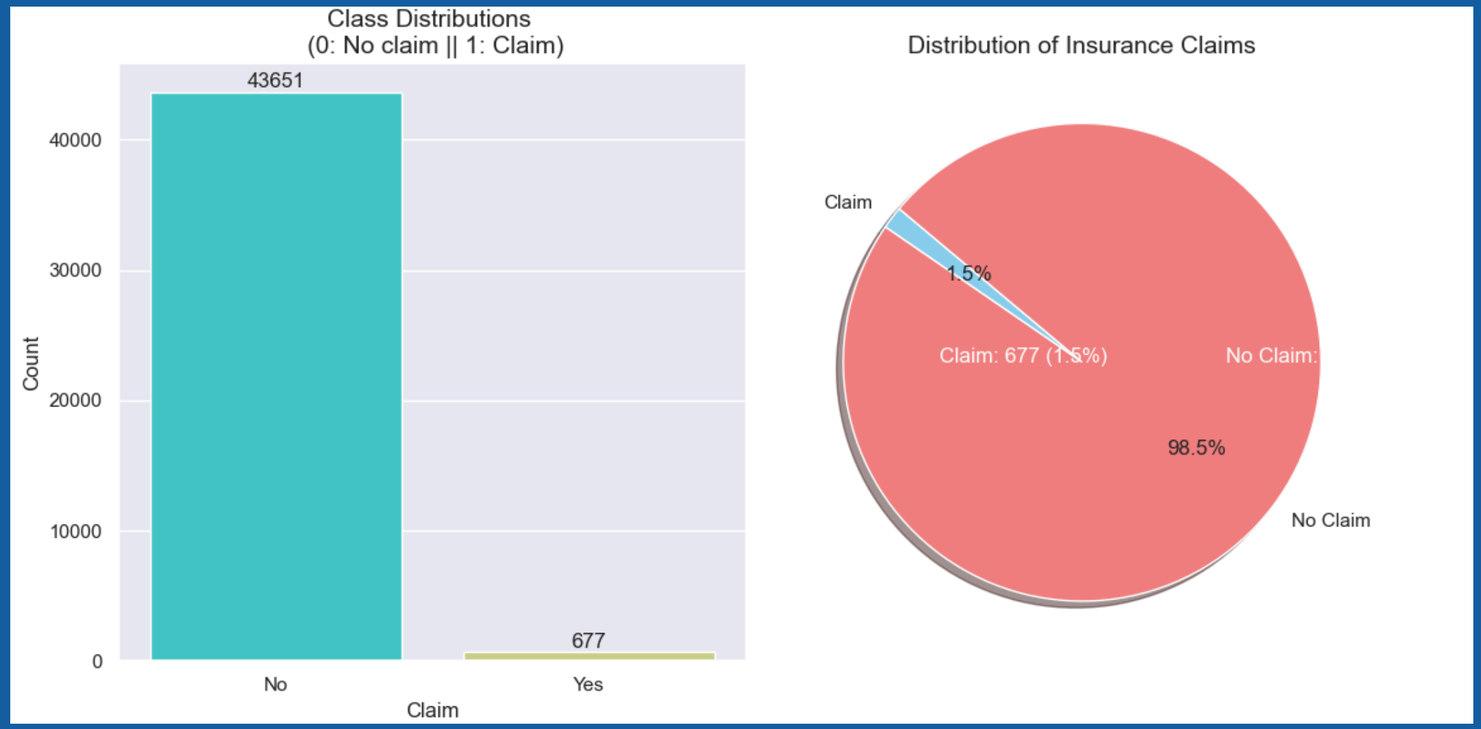
GOAL

01 Forecast the insured who will make a future claim.

02 Savings for the company

03 come up with actionable insight

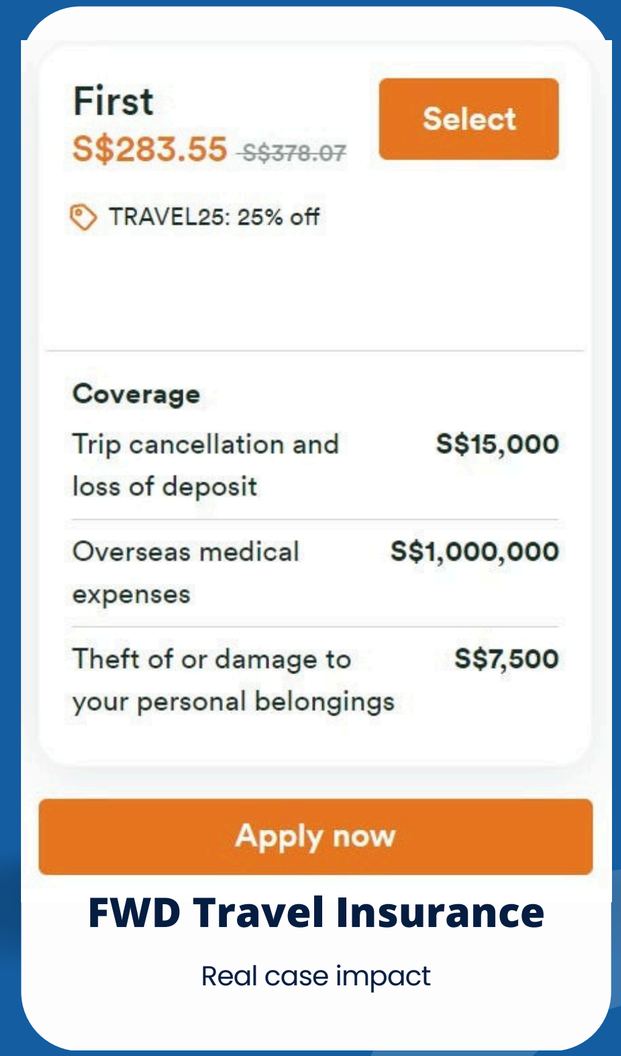
The Impact of Claim In Travel Insurance



Extremely imbalance dataset !

But

Each Claim similar to 45 Insured



FWD Travel Insurance

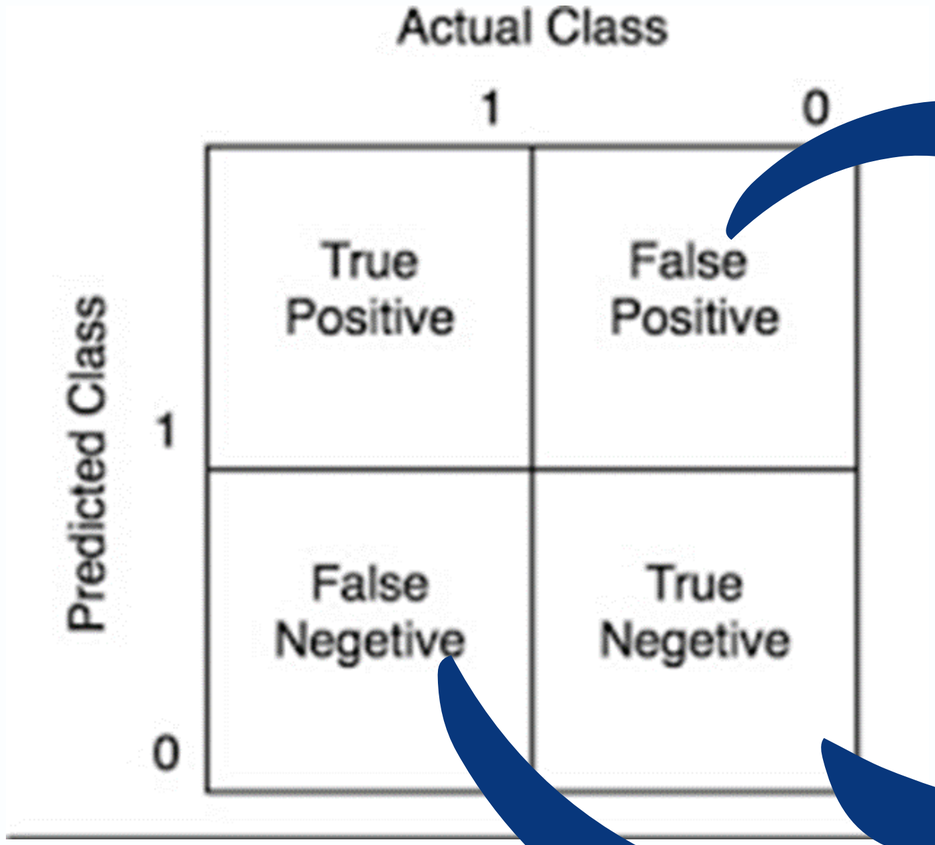
Real case impact

ANALYTICAL APPROACH

- Premium: S$283.55
- Coverage: S$15,000
- Campaign: S$15

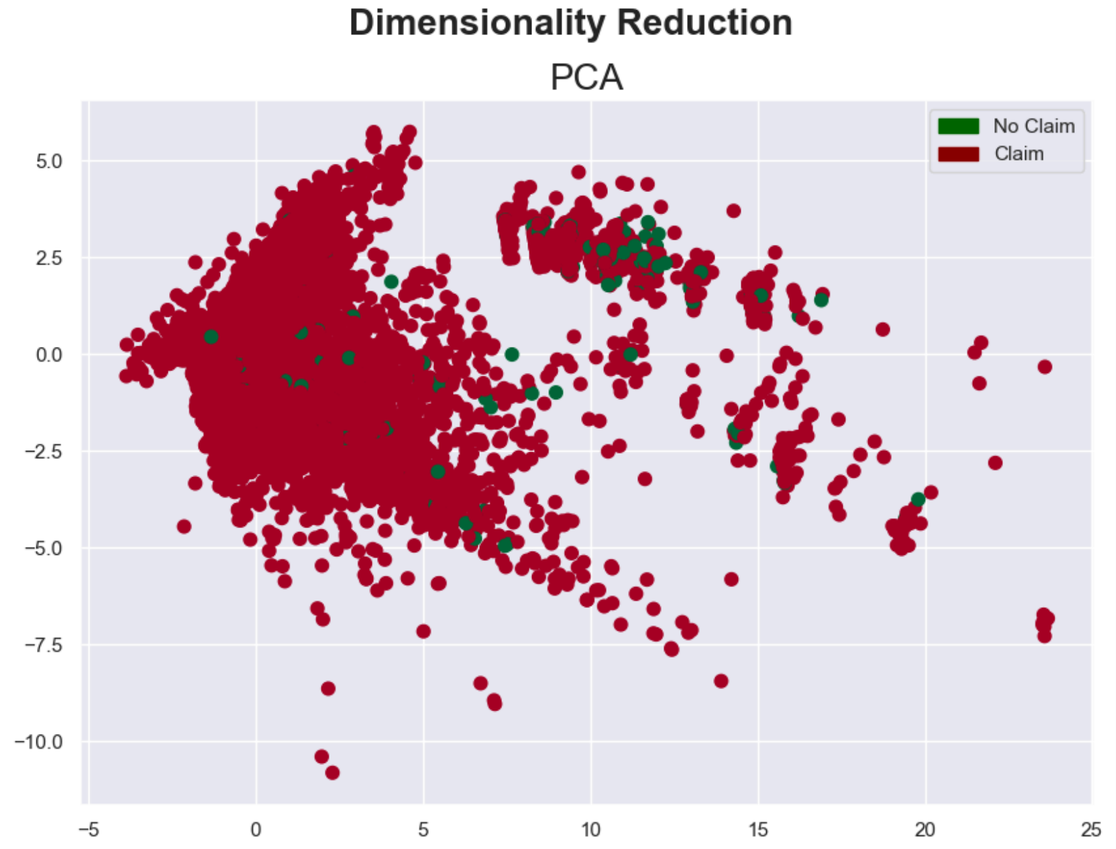
Coverage = 45x Premium

So at the end we are gonna focus on recall to manage False Negative



We put fix cost to manage campaign

The cost is too big



DIMENSIONALITY REDUCTION

PCA projects data onto these principal components, capturing the most important information in fewer dimensions.

Data Condition

Potential Problem	Condition	Treatment
Null Value	Col Gender (71%)	Impute with 'Not Specified'
Duplicates	11%	Drop
Invalid Data	Duration > 4000 Duration < 0 Age > 100	Drop
Data Anomaly	All feature similar but not target (80 data)	Drop Majority Class
Outlier	10%	Parsial Drop & Robust Scaler
Imbalance	98.5:1.5	Resampling, giving weight

FEATURE ENGINEERING

1. **Scaling**
   Robust Scaler

2. **Encoding**
   OneHotEncoding
   Binary Encoding

3. **Imputer**
   Simple Imputer

4. **Add new feature**
   - Continent
   - Product Name Category

5. **Feature Selection**
   SelectKBest

MODELLING

Need Scaler

- KNeighborsClassifier (KNN) - K-Nearest Neighbors Classifier
- SVC (SVM) - Support Vector Classifier
- LogisticRegression (logreg) - Logistic Regression

No need Scaler

- GradientBoostingClassifier (gbc) - Gradient Boosting Classifier
- XGBClassifier (xgbc) - Extreme Gradient Boosting Classifier (XGBoost)
- DecisionTreeClassifier (tree) - Decision Tree Classifier
- RandomForestClassifier (rf) - Random Forest Classifier
- AdaBoostClassifier (ada) - AdaBoost Classifier

1. **Recall**
   F2
2. Resampling
   Penalized Models
3. Cross Validation

Model Benchmark

Best Model with Penalized

model	cv_recall_score	cv_score_f2	cv_score_std	test_recall_score	test_score_f2	diff
XGBClassifier	0.385858	0.192736	0.051038	0.481203	0.220842	0.028106
SVC	0.140416	0.135259	0.025602	0.180451	0.172414	0.037155
AdaBoostClassifier	0.213490	0.155017	0.026076	0.210526	0.142566	0.012450
DecisionTreeClassifier	0.213490	0.154653	0.026076	0.210526	0.141988	0.012665
LogisticRegression	0.172333	0.161384	0.016854	0.150376	0.139860	0.021524

XGBClassifier

Best Model with Resampling

model	train_score_f2	train_recall_score	train_sampling_method	test_score_f2	test_recall_score	test_sampling_method	diff
XGBClassifier	0.070449	0.848298	Nearmiss	0.254731	0.526316	SMOTETomek	0.184281
XGBClassifier	0.163544	0.234103	SMOTE	0.254731	0.526316	SMOTETomek	0.091186
XGBClassifier	0.169040	0.245336	SMOTETomek	0.254731	0.526316	SMOTETomek	0.085691
GradientBoostingClassifier	0.253988	0.554311	SMOTETomek	0.240730	0.654135	SMOTETomek	0.013257
GradientBoostingClassifier	0.250850	0.546852	SMOTE	0.240730	0.654135	SMOTETomek	0.010119
GradientBoostingClassifier	0.067860	0.805255	Nearmiss	0.240730	0.654135	SMOTETomek	0.172870

Hyperparameter Tuning

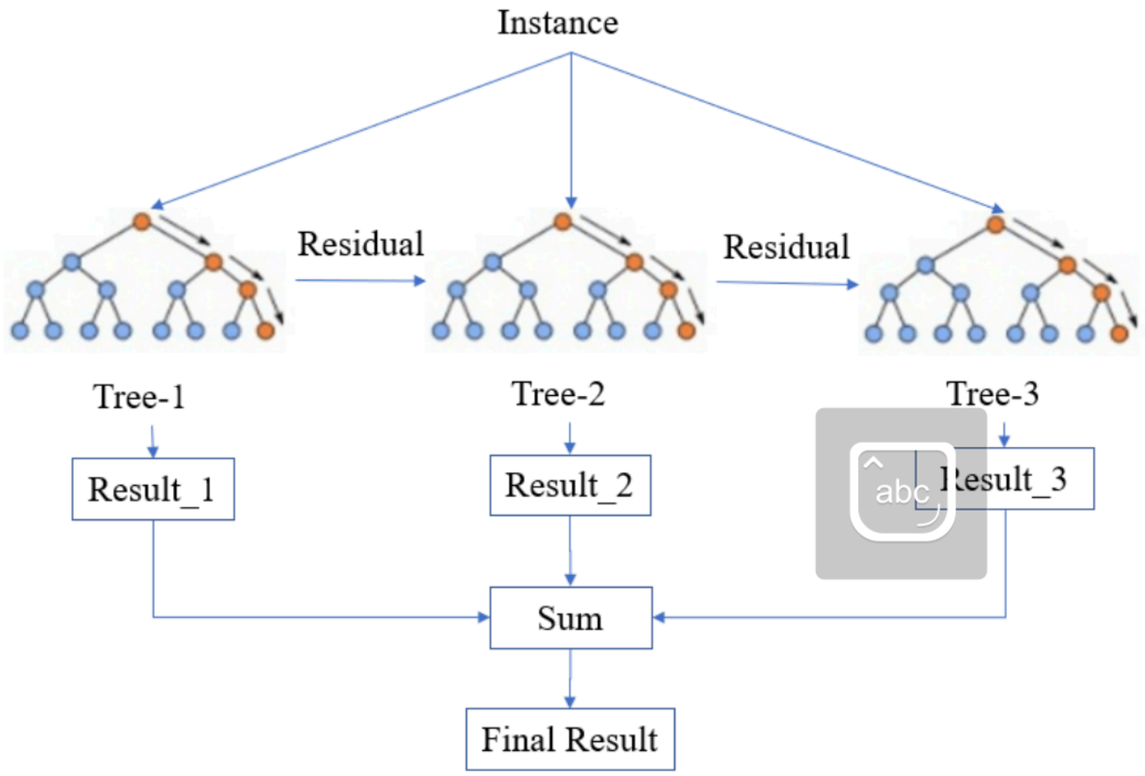
depth: [3]	specifies the maximum depth of the trees in the boosting model. It controls the complexity of the individual trees
subsample: [0.6]	This parameter specifies the fraction of samples to be used for fitting the individual trees.
learning_rate: [0.007]	This parameter controls the step size at each iteration while moving toward a minimum of the loss function.
model iterations: [250]	this controls how much each boosting iteration contributes to the final model.
scale_pos_weight: [class_weights*3]	balance the weight of positive and negative samples in the case of imbalanced classes

Why XGBoost So Good?

Best Recall after Hyperparameter Tuning = 0.947368

Core Components

- Base Learners (Decision Trees): XGBoost builds decision trees iteratively.
- Loss Function: Measures how well the model's predictions match the actual outcomes.
- Objective Function: Combines the loss function with a regularization term to prevent overfitting.



Cost Analysis

- Premium: S$283.55
- Coverage: S$15,000
- Campaign: S$15

Coverage = 45x Premium

Without Machine Learning

Premium x All Insured

283.55 x 7804 = S$ 2.212.824,2

Claim cost

Claim Insured x coverage

133 x 15000 = S$ 1.995.000

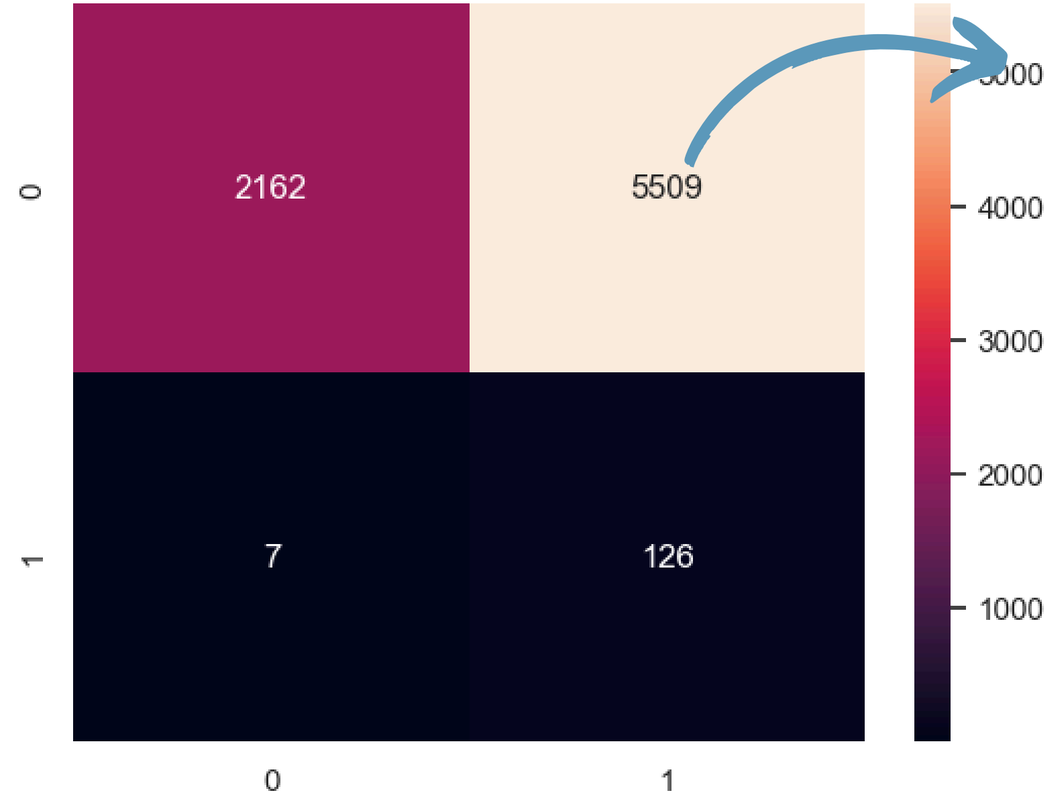
Promo cost

All insured x campaign

7804 x 15 = S$ 117.060

Profit

S$ 2.212.824,2 - S$ 1.995.000 - S$ 117.060 = S$100.764,2



The cost consequence of a wasted campaign

With Machine Learning

Premium x (TN + FN)

283.55 x (2162 + 7) = S$ 615.019,95

Claim cost

Claim Insured x coverage

7 x 15000 = S$ 105.000

Promo cost

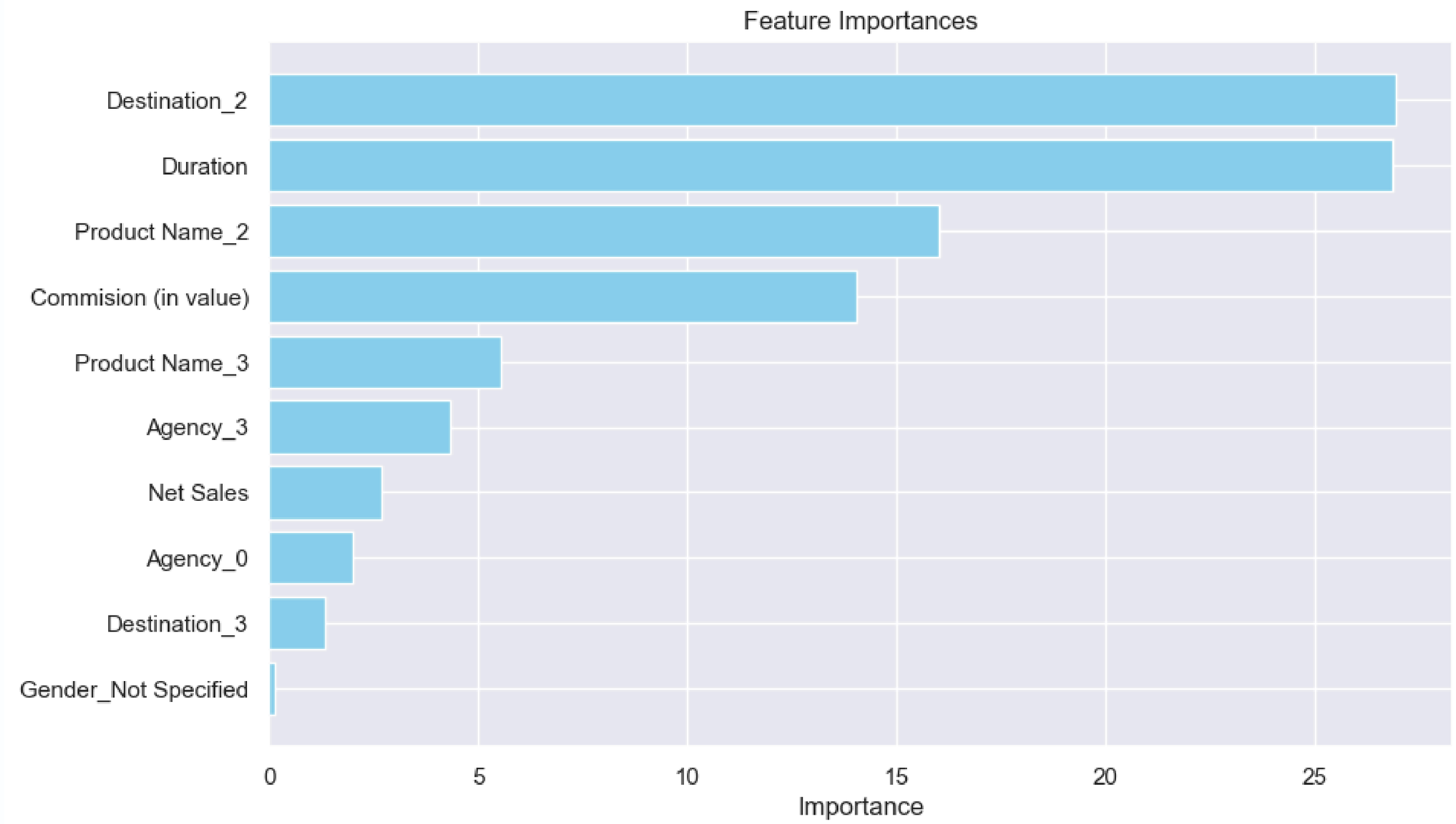
All insured x campaign

7804 x 15 = S$ 117.060

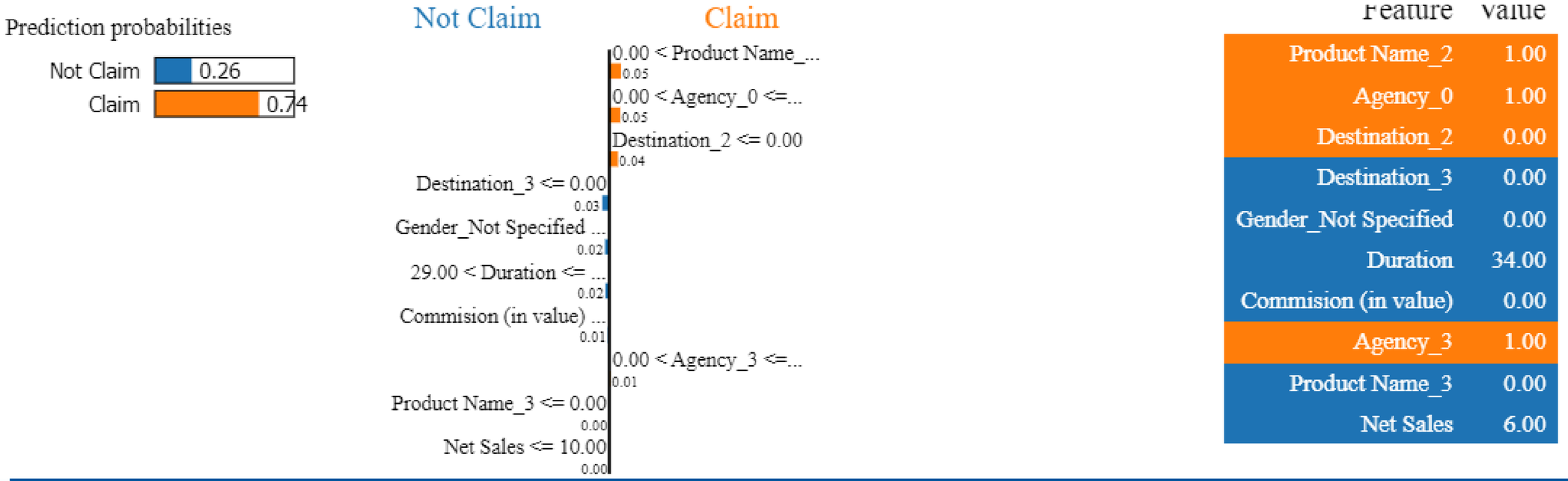
Profit

S$ 615.019,95 - S$ 105.000 - S$ 117.060 = S$ 392.959,95

Feature Importances



Explainable AI



Conclusion

- XGBoost Classification, after tuning, achieved the best performance with a recall of 0.9473, making it very effective in minimizing false negatives which is claims.
- With Explainable AI, we can identify the most influential features affecting the final prediction outcome, allowing for in-depth analysis of these features. Additionally, it enables us to explain how each prospective insured's data will result in a claim or not.
- By using machine learning, we can achieve greater profits by minimizing the number of undetected claims, up to S$ 392,959.95. In this case, the profit can reach almost four times as much.

Recommendation

- Given its high recall score of 0.9473, this is highly effective in minimizing false negatives, thereby ensuring that fewer legitimate claims go undetected. However, we need more data, especially for the minority class, so the model can better predict both classes.
- Maximizing the capabilities of machine learning to explain which prospective insured individuals are likely to make a claim. However, we need to add features such as medical history and the number of dependents so that the data can be more clearly interpreted.
- Implementing a machine learning model in business to minimize claims is crucial, but it should be noted that reducing false negatives often comes at the cost of increasing false positives. Therefore, continuous improvement of the model is necessary to achieve the best possible results.

MODEL DEPLOYMENT ON CLOUD