DSA4263 Sense Case Making Analysis: Business and Commerce (Fraud Analytics) Report

Lecturer: Professor Valerie Lim Hui Yi

TA: Mr. Deng Ruizhe

Name	Student Number
Kevin Christian	A0219722E
Loo Guan Yee	A0223941J
Sun Peizhi	A0219930A
Vivek Bagai	A0219992M

Table of Contents

Abstract

Insurance payouts are often the last safety net against unfortunate incidents, but they are also targeted by fraudsters. Fraudulent claims have bad impacts including increased company costs, reduced user satisfaction, and poorer company reputation. Consequently, approving manual claims while exercising sufficient due diligence against fraud takes more time. Unfortunately, some cases may require expeditious decision-making, especially when patients in critical conditions are involved. Hence, in this project, we aim to deliver a machine-learning solution for automated insurance claim processing, speeding up the approval time and preventing insurance fraud. Several challenges were encountered in this project, including significant dataset imbalance and model performance optimisation with limited data. Based on our experimentation, LightGBM with the Adaptive Synthetic Sampling Approach for imbalanced dataset handling showcases the greatest potential.

Introduction

There are about 1.475 billion cars in the world, or approximately one car for every 5.5 people (Hedges & Company, 2021). Looking deeper, in Singapore, there are 851,210 registered vehicles as of February 2024 (LTA, 2024). While making travel faster and easier, cars also lead to more road traffic accidents. It is estimated that road injuries will cost the world economy approximately USD1.8 trillion between 2015 and 2030 (Chen et al., 2019). This is where auto insurance comes in -- to ensure that the emotional and physical toll of motor accidents is not made worse by catastrophic financial loss.

Auto insurance coverage largely falls under three categories (Probasco, 2024). The first is Property, which is concerned with situations involving damage to or theft of your vehicle. The second category, Liability, covers the expenses that arise when you are at fault for causing physical harm to others or damaging another person’s vehicle. The third and final category is Medical, which addresses the expenses associated with medical treatment for injuries that you or your vehicle’s passengers sustain during an accident.

While your total coverage can differ depending on what you require for your vehicle, in Singapore, it is against the law to drive a vehicle without a valid insurance policy to cover third party liability (LTA, 2024). The above has resulted in auto insurance being a booming business. In fact, the global auto insurance industry is valued at USD810 billion, and it is estimated to grow to USD1765 billion by 2032 (The Brainy Insights, 2023).

Of course, auto insurance is no stranger to fraud. In an industry that relies on good faith communication between consumers and companies, fraud is a multi-billion-dollar challenge that puts a huge burden not only on insurance companies, but also consumers with increased premium rates. In Singapore, it is estimated that about 20 percent of all motor claims are fraudulent (GIAS, 2018). This includes staged accidents, exaggerated claims, or phantom claims (e.g. when a passenger not in the vehicle makes a personal injury claim).

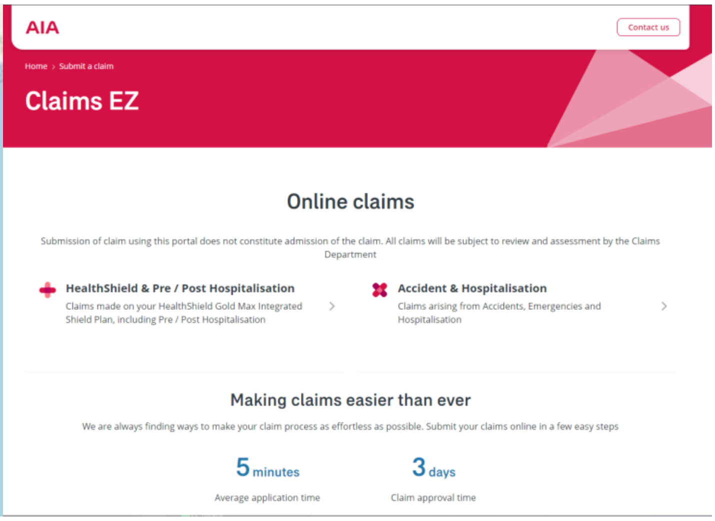
Thus, there is room for an analytical solution to this business problem. In this paper, we examine motor claims fraud and aim to answer the following problem statements. Firstly, it is to gain preliminary insights on fraudulent behaviours from EDA on our chosen dataset. Secondly, it is to provide a product solution in predicting claims fraud via an ML model. Lastly, it is to explain the model decision process on whether the claim is fraudulent or not.

Data

The dataset contains 1000 rows and 40 columns. The columns recorded consist of the clients' details and their respective claims. More details about each column belonging to policy and claims details can be found in the appendix.

The dataset is sampled from various insurance companies from three US states: South Carolina, Virginia and Albany. Each row represents a single anonymised claim record submitted by the insurance client to the insurance companies. The *fraud_reported* label is the classification made to decide if this claim is fraudulent or not. There are 200 fraudulent claims present in this dataset and this dataset is imbalanced.

Ottawa
Montreal
Toronto
Hamilton
Milwaukee
Detroit
Chicago
Boston
Philadelphia
Washington
Richmond
Raleigh
Charlotte
Memphis
Atlanta
Jacksonville
Baton Rouge
15
14
163
122
60
70
148
States:
● South Carolina
● Virginia
● Albany



AIA

Contact us

Home > Submit a claim

Claims EZ

Online claims

Submission of claim using this portal does not constitute admission of the claim. All claims will be subject to review and assessment by the Claims Department

HealthShield & Pre / Post Hospitalisation
Claims made on your HealthShield Gold Max Integrated Shield Plan, including Pre / Post Hospitalisation

Accident & Hospitalisation
Claims arising from Accidents, Emergencies and Hospitalisation

Making claims easier than ever

We are always finding ways to make your claim process as effortless as possible. Submit your claims online in a few easy steps

5 minutes
Average application time

3 days
Claim approval time

Although the dataset does not provide any details about this dataset acquisition, this report will, nonetheless, offer our group’s educated guess on this dataset’s creation based on the existing insurance industry practices in Singapore. Currently, the insurance professionals will perform manual checks on these claims. Before commencing the manual checks, the clients’ policy details and their claims details must be available. The policy details from the customers are retrieved from the latest customers’ database in the insurance companies and updated frequently to reflect their age or changes to their insurance policy portfolio.

From the client's side, they will need to submit their car claims, a PDF and fill in the details about their respective incident via the dropdown button or fill-in-box online. Then, the process of transferring data from this softcopy could be done either with OCR technology or manual data entry by administrative personnel.

Once this information is available to the insurance professionals, we postulate that manual checks are done to ensure impartiality and fairness. The first preliminary round of checks is done to check for any anomaly between the profile details and claims details. The second round is to moderate the insurance claim classification to confirm if the insurance claim is correct. The final output for the *fraud_label* column represents the final decision from the insurance experts.

Methods

Exploratory Data Analysis (EDA)

This report's EDA was split into four distinct approaches: claims summary statistics, criminology literature, fraudster's literature and anomaly detection to account for all the possible angles for fraudulent claims investigation.

Summary statistics analysis

Firstly, EDA was used to investigate the profile of the insurance clients via their insurance claims. For brevity's sake, the client's age was categorised into the age intervals and then segregated by Fraud status shown in the table below.

The analysis of summary statistics consists of qualitative and quantitative analysis. In the Qualitative analysis done below, we investigated the top 3 most common categories in the insured clients' education level, occupation level and auto car brand. There is no distinct difference in the education level based on the fraud status. Thus, it was removed from the qualitative analysis table.

Age Interval	Fraud Status	Top 3 Occupation Level	Top 3 Auto Car Brand
20 to 29	N	● Prof-specialty ● Machine-op-inspct ● tech-support	● Dodge ● Nissan ● Toyota
20 to 29	Y	● Exec-managerial ● Farming-fishing ● sales	● Audi ● Volkswagen ● Ford
30 to 39	N	● Machine-op-inspct ● Other-service ● protective-serv	● Nissan ● Dodge ● Suburu
30 to 39	Y	● Exec-managerial ● Tech-support ● craft-repair	● BMW ● Chevrolet ● Dodge
40 to 49	N	● Prof-specialty ● Machine-op-inspct ● transport-moving	● Chevrolet ● Ford ● Saab
40 to 49	Y	● Machine-op-inspct ● Prof-specialty ● sales	● Audi ● Mercedes ● Ford
Above 50	N	● priv-house-serv ● Other-service ● sales	● Suburu ● Saab ● Jeep
Above 50	Y	● Craft-repair ● Transport-moving ● farming-fishing	● Dodge ● Mercedes ● Chevrolet

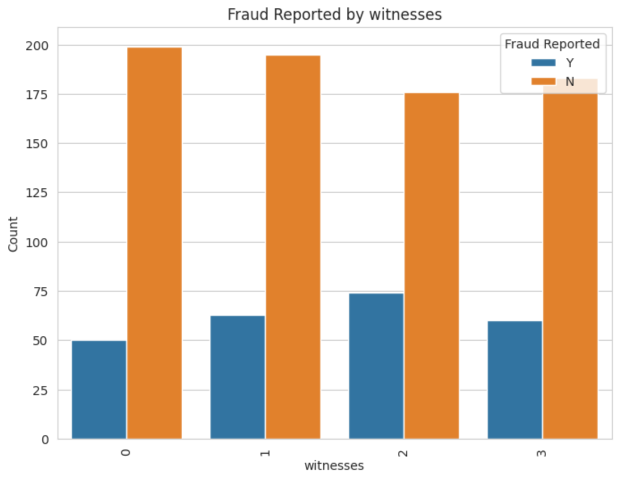
From the qualitative analysis table of the occupation, we noticed that insured clients with "Exec-managerial" as their occupation declaration are quite common in fraudulent claims. In addition, across all the age intervals, European-based vehicles such as Audi, Mercedes, and BMW were commonly flagged as fraudulent claims. In contrast, there was no distinct difference in American-based cars. This might mean there is an equal probability of fraudulent claims for American vehicles. The insights from the qualitative analysis provided us with the motivation to investigate further if the automobile car origin contributes to the fraudulent claims' outcome.

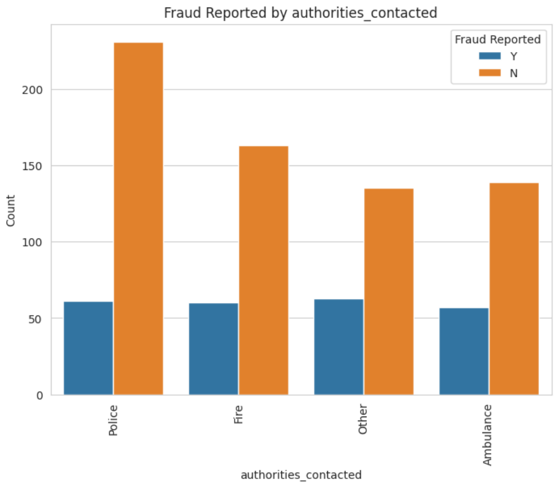
Age Interval	Fraud Status	Policy Annual Premium 95% CI	Policy Tenure 95% CI	Car age 95% CI
20 to 29	N	(1258.35, 1271.87)	(13.01, 13.38)	(9.77, 10.09)
20 to 29	Y	(1210.59, 1236.25)	(11.58, 12.34)	(8.47, 9.02)
30 to 39	N	(1237.9, 1248.61)	(12.93, 13.24)	(9.96, 10.22)
30 to 39	Y	(1254.47, 1278.33)	(13.03, 13.74)	(9.67, 10.28)
40 to 49	N	(1277.88, 1281.49)	(13.21, 13.33)	(9.72, 9.81)
40 to 49	Y	(1243.5, 1252.28)	(12.76, 13.0)	(10.42, 10.63)
Above 50	N	(1256.68, 1259.86)	(12.54, 12.64)	(9.78, 9.86)
Above 50	Y	(1208.82, 1219.35)	(13.4, 13.72)	(9.81, 10.08)

For quantitative analysis, the dataset represented a small subset of the large insurance claims record. Thus, a 95% confidence interval was performed on the numerical variables to quantify the uncertainty of the mean of the numerical variables investigated. From the quantitative analysis table above, the confidence interval for Annual policy premiums across different age intervals and fraud status does not overlap. Similarly, patterns could be observed for the Policy Tenure and the insured claimant's car age as well. This motivated us to create the car age column from the auto year column provided in the dataset to be trained by the machine learning algorithms.

Incident Exposure Analysis

Secondly, EDA was performed based on the existing literature about fraudsters' modus operandi on exposure theory, whereby the fraudsters seek to maximise their exposure to keep themselves clean under the radar. We thus analysed the exposure of the case, i.e., how "well-known" is the incident, to both the authorities and the general public. In our dataset, two columns, "witnesses" and "authorities_contacted", represent the exposure of the event in the above dimensions.

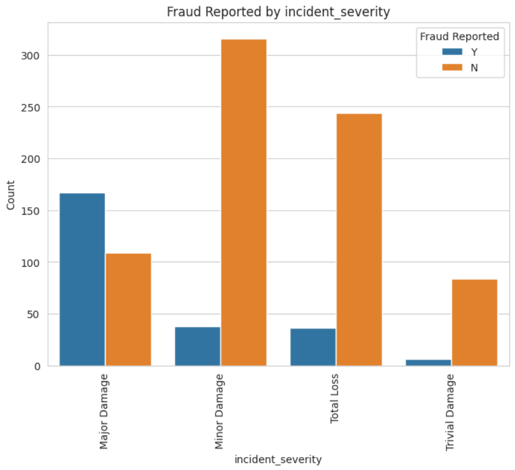


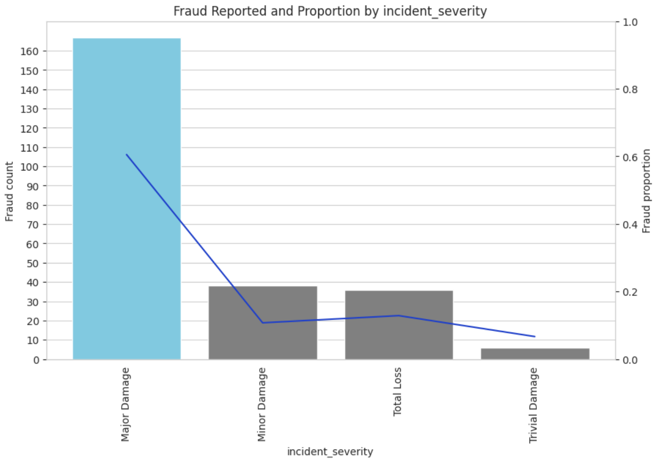


Based on the visualisations provided above, surprisingly, both types of cases exhibit similar distributions regarding the number of witnesses and the authorities_contacted column. This suggests a potential collusion between fraudsters and members of the public to procure witnesses, and possibly deceitful behaviour towards authorities to manipulate their involvement in the incident. It is plausible that such actions stemmed from past fraud detection methods that heavily relied on exposure, prompting fraudsters to pay special attention to these aspects and take extra precautions to conceal their actions. Consequently, we hypothesise that the public exposure of an incident might not reliably predict insurance fraud.

Fraudster Incentive Analysis

Thirdly, the report delved into the mindset of the fraudsters' psychology through fraudster incentives. In the context of fraudulent claims, it is important to ask cui bono. If the fraudster does not benefit from the act, they would not have the incentive to commit insurance fraud in the first place. Therefore, the severity of the incident was scrutinised closely for this third EDA.



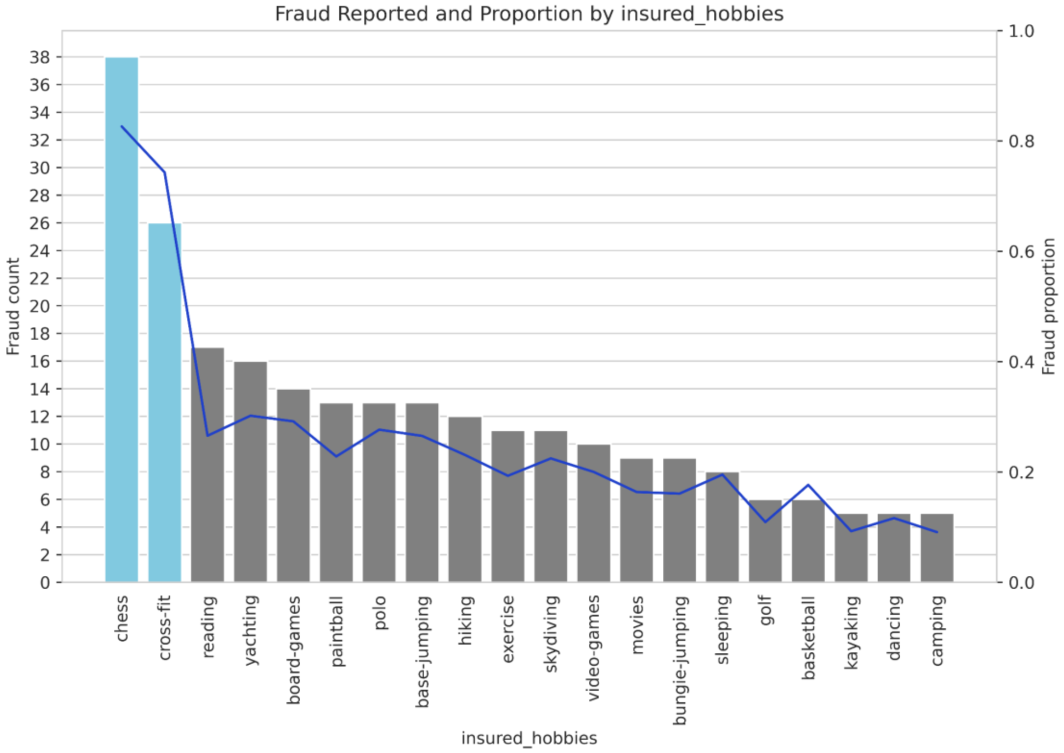


As observed from the visualisations above, the majority of Major Damage reported is fraudulent. This pattern is highlighted in particular by the left figure, in terms of the absolute Major Damage count, and the proportion plot on the right.

As such, we hypothesise that incident severity could be a significant predictor in fraud detection, where cases reported as Major Damage are likelier to be fraudulent. This presumption is grounded in economic motivations, as fraudsters are more inclined to label their cases as major for larger insurance payouts. Given the substantial monetary investment and logistical efforts involved in orchestrating fraudulent schemes, fraudsters seek commensurate economic gains to offset their expenses effectively.

Anomaly Analysis

Lastly, EDA was used to discover anomalies in the dataset. A noticeably large number of insurance claims, in which the clients who indicated “chess' or “cross-fit' under the insured_hobbies column, were fraudulent, as shown in the visualisation below.



As correlation does not imply causation, it is preposterous to conclude that insurance clients who declared “chess” and “cross-fit” are more likely to commit fraudulent claims. The high frequencies in the “chess” and “cross-fit” are attributed towards cultural factors and location-driven. Since the dataset used is based in the USA where the culture is arguably dominantly Anglo-Saxon, the popularity sentiments towards cross-fit in the USA and chess as part of common pastimes will be reflected in this dataset unsurprisingly.

As such, the report made the following three hypotheses. Firstly, the car brand and the insured claimants' policy affects the fraudulent claims. Secondly, either exposure theory or fraudster incentive influences fraudulent claims. Lastly, the hobbies “chess” and “cross-fit” do not contribute to the fraudulent claims outcome.

ML Preprocessing

Our group proposed a preprocessing pipeline to streamline the preprocessing of the insurance dataset so that the preprocessed trained and test dataset could be obtained with a function call from our script.

Training and testing data approach

The first step for the preprocessing pipeline is to split the insurance dataset into training and test datasets. This is to prevent data leakage whereby normalisation is performed on the entire dataset instead of separately on the training and test dataset. Since the dataset is imbalanced, a stratified split is to be performed on the dataset to split the training and testing dataset in the 80% to 20% ratio, while maintaining sufficient representation of the majority and minority classes, to ensure that the machine learning models could learn well from the model.

Feature Engineering

Upon splitting the dataset into train and test datasets, the preprocessing pipeline will kickstart three important steps: additional feature creation, augmentation and normalisation.

The EDA's summary statistics provided insights into the policy tenure of the insurance claimants where there is a distinct difference in the 95% CI. Therefore, the *policy_age_during_incident_in_days* feature was created by subtracting the incident date with the *policy_bind_date* feature which represents the date when the policy was initiated.

In addition, the *policy_csl* feature which is the coverage limit per accident has '/' in it. Thus, we decided to split this feature into *bodily_injured_maximum_coverage_per_accident* and *complete_maximum_coverage_per_accident* since the ML model could not take in string inputs.

The *auto_make*, *auto_year* and *auto_model* features record the insurance claimants' car details. Two features were created after performing summary statistics EDA: *auto_region* from the *auto_make* to capture the car's country of origin, and *car age* to capture the car age from the *auto_year*.

The *auto_model* feature is of utmost interest to the group. The multiple *auto_model* type under the same *auto_make* company contributes to high cardinality should one-hot encoding be used. Therefore, the *auto_type* feature is created by summarising the auto model into car types such as "Hatchback", "Coupe", "Truck", "SUV", and "Sedan".

One-hot encoding was performed on all of our categorical features, owing to them having relatively low cardinality and being not ordinal.

Scaling

Lastly, the *preprocessing_pipeline* function provided the option for the numerical variables to be scaled. It accounts for the distance/scale-sensitive ML models such as Logistic Regression, MLP and SVM and the scale-agnostic tree-based models.

The *preprocessing_pipeline* function used three different scaling methods, StandardScaler, MinMaxScaler, and RobustScaler, to account for different numerical distributions from the numerical features. StandardScaler was used for numerical features that displayed a normal distribution and had negative entries as well. MinMaxScaler scales and translates the numerical values between the specified ranges. It relaxes the assumption of the normal distribution for the numerical feature. However, the MinMaxScaler's sensitivity towards outliers means that it should not be used since the outliers will skew the scaling significantly. Therefore, RobustScaler was used to account for outliers or skewness in the numerical features. The table below shows the numerical columns and their respective normalisation types.

Scaling Type	Numerical Features
StandardScaler()	- months_as_customer - age - policy_annual_premium
MinMaxScaler()	- policy_deductable - umbrella_limit - policy_age_during_ incident_in_days (Feature engineered) - bodily_injured_maximum _coverage_per_accident (Feature engineered) - complete_maximum_coverage_per_accident (Feature engineered) - number_of_vehicles_involved - bodily_insuries - incident_hour_of_the_day - witnesses - car_age (Feature engineered)
RobustScaler()	- capital-gains - capital-loss - total_claim_amount - injury_claim - property_claim - vehicle_claim

After performing scaling on the numerical features, eight features from the dataset were dropped (see Appendix); the final dataset has 125 columns.

Resampling methods

This report explored these three oversampling methods: Synthetic Minority Over-sampling Technique for Nominal and Continuous (SMOTENC), Adaptive Synthetic Sampling (ADASYN) and Random Oversampler.

SMOTE generates synthetic data using the average of several existing data points for the minority class. The authors' experiment showcases its improved performance over other imbalance handling methods, including shrinkage and undersampling. This is presumably due to the algorithm's introduction of additional noise into the training data when generating synthetic samples through estimation, leading to improved generalisability (Tibshirani et al., 2021) of the trained model. SMOTENC is a variant of the original (SMOTE) that can handle categorical data types (Imbalanced-Learn, 2024). Since it is also the method utilised by the researchers of this dataset, ML models with SMOTENC and no tuning were used as the baseline models (Aqqad, 2023).

ADASYN was also used in our project. The main idea is to use weighted distribution for different minority class examples according to their level of difficulty in learning. Then, synthetic samples are generated based on hard to learn observations. This hopes to reduce the likelihood of misclassification due to the presence of synthetic examples in regions with low sample density, thereby improving the model's performance on the hard observations (He et al., 2008).

Lastly, the inclusion of the Random Oversampling method is motivated by the business problem and the oversampling literature. Firstly, it anticipated that some organisations might have a more stringent data governance or be conservative towards the dataset's usage. Hence, the stakeholders might not be open to the idea of using SMOTENC and ADASYN to generate synthetic examples for oversampling, and random oversampling is a good alternative for oversampling. Secondly, random oversampling is a simple and powerful method to address imbalanced datasets. From the existing literature, there are two arguably effective methods (Batista et al., 2004) to address imbalanced datasets. The first method is to perform random oversampling only to match the majority class. The alternate method will require oversampling of SMOTE and undersampling via TOMEK to achieve similar results. The alternate method will contribute to additional implementation costs to our current and future codebase. Therefore, the random Oversampling method is shortlisted.

In our project, no resampling was performed on the test data to prevent data leakage, and special precaution was also taken to ensure that the validation set was not resampled during the validation process for hyperparameter tuning.

ML Performance enhancement

Hyperparameter Tuning

We used Optuna as the hyperparameter tuning framework for its ease of use (Akiba et al., 2019). Unlike Random Search, which explores parameter combinations randomly, Optuna's Bayesian Optimisation leverages prior trials to guide its search towards promising regions of the hyperparameter space (Snoek et al., 2012), reducing the number of trials needed. Additionally, Optuna mitigates the risk of overfitting compared to Grid Search, by constraining the search space (Tibshirani et al., 2021), ensuring a more focused exploration.

K-Fold Stratified Cross Validation

The K-Fold stratified Cross Validation (CV) was used in Optuna's hyperparameter tuning to determine the ability of the tuned model to generalise to unseen data and to reduce the likelihood of overfitting the dataset through repeatedly testing the test dataset. The stratified CV ensures that there are sufficient fraudulent claims present in the tuned model and the validation dataset to account for the imbalance dataset. Since K-fold stratified CV is a time-consuming process and our ML's solution needs to be scalable, k = 5 is used to minimise the hyperparameter tuning time to obtain a tuned model.

Machine Learning Approaches

This report explored the variation of ML's performance based on different ML's architecture: Linear Models, Tree-Based models, and Gradient-Boosting Trees.

Logistic Regression

Logistic Regression is the first linear model we considered due to its interpretability and efficiency. Logistic regression is our group's baseline model. Logistic regression starts from a linear equation consisting of log-odds which is passed through a sigmoid function to squeeze the output to a probability between 0 and 1. The outcome is then predicted based on a specified threshold.

The training and testing data were normalised to improve the convergence speed, numerical stability and ensure consistency for the regularisation. Then, the Hyperparameter tuning is conducted to maximise f1-score on the following parameters: C (regularisation strength) and solver (e.g. Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS), Newton-CG) while preventing overfitting.

SVM

We shortlisted Support Vector Machine (SVM) as the second linear model. SVM aims to find the hyperplane that best separates the classes in the feature space. This hyperplane is chosen such that it maximises the margin between the closest data points of different classes, also known as support vectors. Since SVM performs well even in high-dimensional spaces, has regularisation parameters that help prevent overfitting, and can efficiently handle nonlinear decision boundaries, it is a highly suitable model to train our dataset on.

Similarly, normalisation is to be performed on the training and test dataset before training the dataset on SVM. During hyperparameter tuning, SVM was tuned on the following parameters: C (regularisation parameter), kernel (type of non linear transformation), degree, and coef0 (required parameters for certain types of kernels).

Decision Tree

In a complex task that has multiple factors that can be used for consideration, the model attempts to split the task into a series of "decision-making" steps such that the task becomes more predictable along the way (Chauhan, 2022). In this model, several hyperparameters are being finetuned, such as *max_depth*, *min_samples_split* and *min_samples_leaf*.

Random Forest

From the explanation above, the decision tree model will create its structure based on the data. However, if the data is limited, it may cause the decision tree to not work well with unseen data. Additionally, it tends to have overfitting issues as it is sensitive to noisy data. Hence, instead of using a single decision tree, this model will use multiple trees where each tree is assigned with different sub-samples / sub-features for objectivity. In the end, the model will obtain the majority vote across all of the trees. This majority vote system will avoid overfitting issues by eliminating the noise that each tree has. In sum, the model tries to make decisions from multiple perspectives. In this model, several hyperparameters are being finetuned such as *n_estimators*, *max_depth*, *min_samples_split*, *min_samples_leaf*, *max_features*, and *bootstrap*.

XGBoost

XGBoost was shortlisted for its high performance and scalability in this project. It applies Sparsity-aware Split Finding in the model, enabling the model to handle sparse data more effectively (Chen & Guestrin, 2016). This would be particularly useful with the large number of categorical variables that were present in this dataset, and the one-hot encoding that was applied in the pre-processing process.

In this project, a tuned model for XGBoost was produced, by tuning variables related to individual tree depth, the number of trees constructed, the share of features that will be used in the construction of individual trees, and L1 and L2 regularisation of the gradient descent process.

LightGBM

Multiple one-hot encoded columns created in the dataset contributed to a very sparse dimensional space. The Exclusive Feature Bundling (EFB) algorithm combines multiple features into a single bundle, increasing the training speed (Ke et al., 2017). Furthermore, there are some features with small feature importance after LightGBM training. The Gradient One-Sided Sampling (GOSS) algorithm makes a selection on the top subset of the gradients and performs random sampling with the remaining b unselected small gradients. Then, the small gradients are multiplied by the ratio $\frac{1-a}{b}$ for the data instances with small gradients to provide a sufficient representation of small gradients during the information gain computation, making LGBM a highly accurate model (Ke et al., 2017).

Similarly, the hyperparameter tuning was performed on these eight LightGBM parameters to improve performance while controlling overfitting: *n_estimators*, *num_leaves*, *learning_rate*, *subsample*, *colsample_bytree*, *min_child_samples*, *reg_alpha* and *reg_lambda*. The *num_leaves* is the most significant parameter that contributes to the hyperparameter performance (Microsoft, 2024).

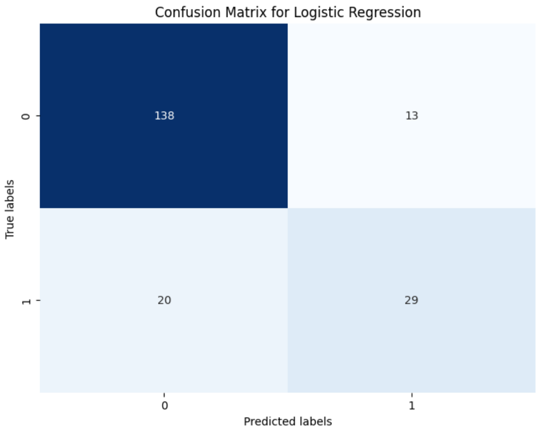
Results

We will investigate the ML's performance via confusion matrices and area-under-curve (AUC) plots on different oversampling methods, analyse the parameters changed after hyperparameter tuning, and lastly perform an explainable ML technique with SHAP plot (Lundberg & Lee, 2017). Since this dataset is imbalanced, F1-Score will be the primary metric to determine which model should be shortlisted for the best model.

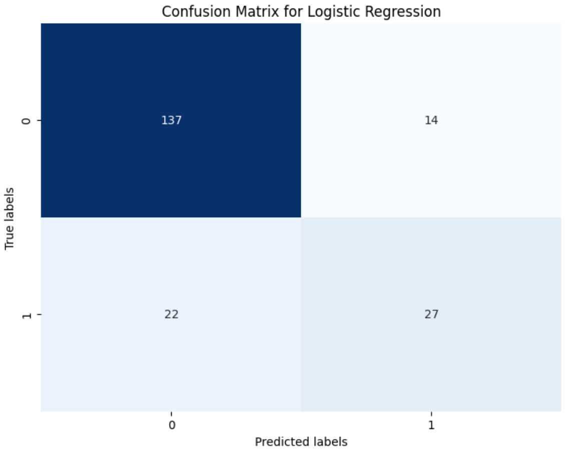
Logistic Regression

Finetuned	Imbalanced Dataset Handling	Accuracy	Precision	Recall	F1-Score	AUC Score
False	SMOTENC	0.835	0.690	0.592	0.637	0.834
True	SMOTENC	0.820	0.659	0.551	0.600	0.816
True	Random Oversampling	0.845	0.650	0.796	0.716	0.846
True	ADASYN	0.825	0.675	0.551	0.607	0.819

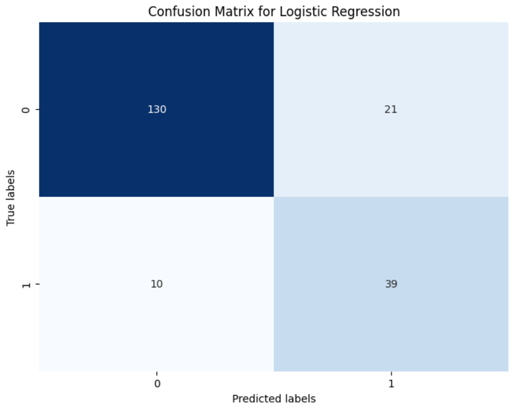
From the machine learning model table, the logistic regression model with Random Oversampling performed the best with an F1-Score of 0.716 and AUC score of 0.816.



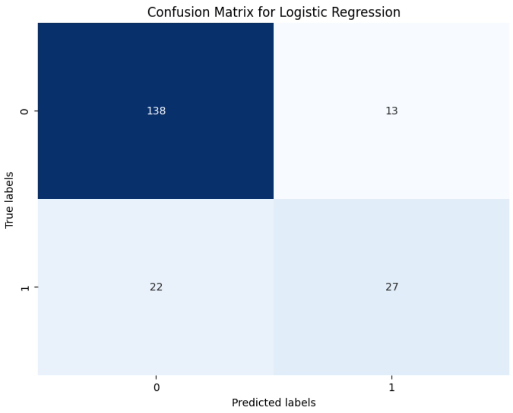
SMOTENC without finetuning



SMOTENC

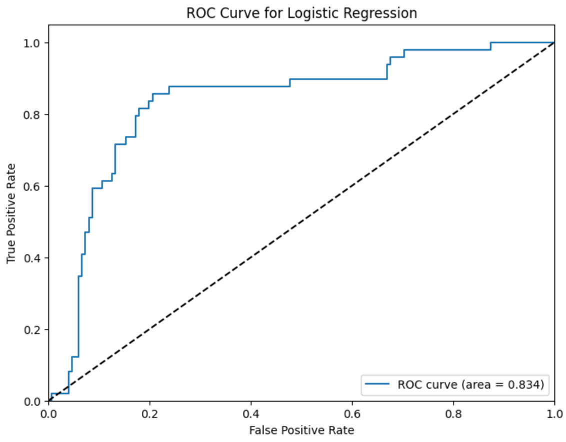


Random Oversampling

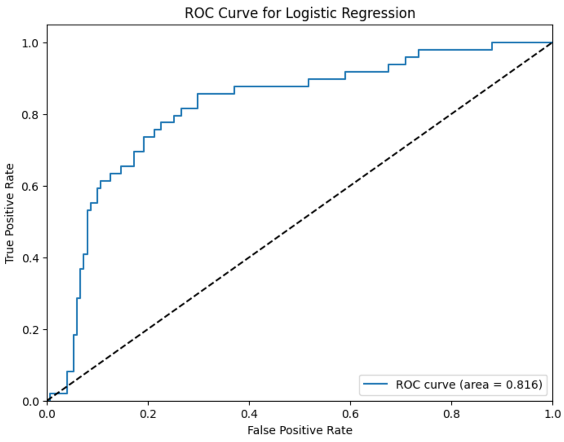


ADASYN

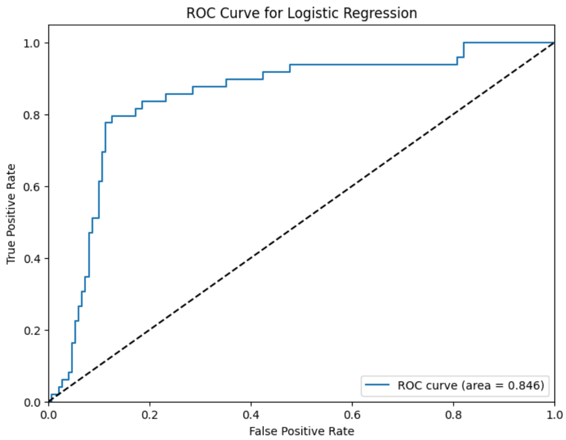
From the confusion matrix above, we can see that the random forest models with SMOTENC and ADASYN have higher False-Negative values and lower False-Positive values. In addition, there is a reduction in the False Negatives and a slight increase in the False Positive after applying random Oversampling and hyperparameter tuning.



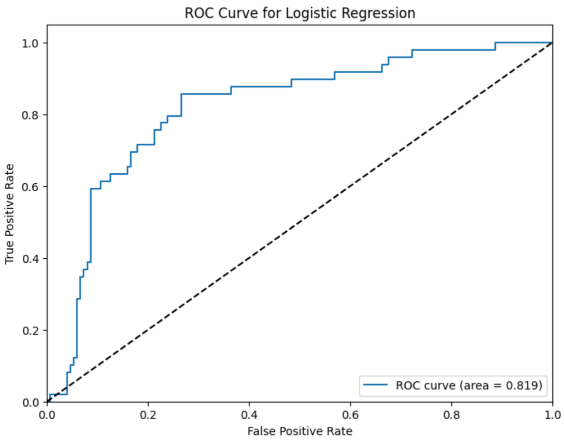
SMOTENC without finetuning



SMOTENC



Random Oversampling



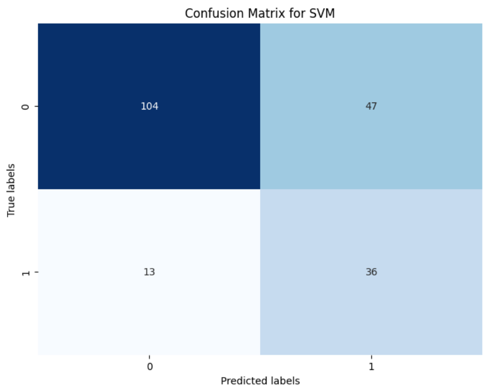
ADASYN

From the AUC ROC curves above, the Random Oversampling curve has the steepest increase in the ROC curve compared to other oversampling methods, contributing to a large area for the AUC of the LR with random oversampling.

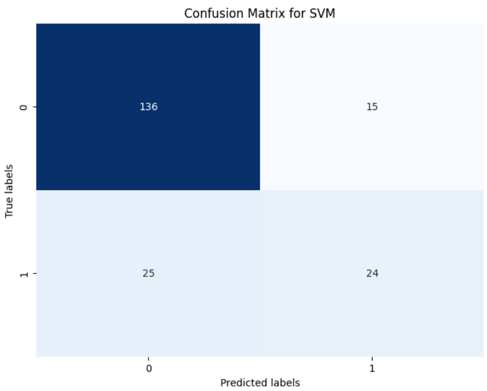
SVM

Finetuned	Imbalanced Dataset Handling	Accuracy	Precision	Recall	F1-Score	AUC Score
False	SMOTENC	0.700	0.434	0.735	0.545	0.500
True	SMOTENC	0.800	0.615	0.490	0.545	0.816
True	Random Oversampling	0.785	0.548	0.694	0.613	0.811
True	ADASYN	0.810	0.634	0.531	0.578	0.818

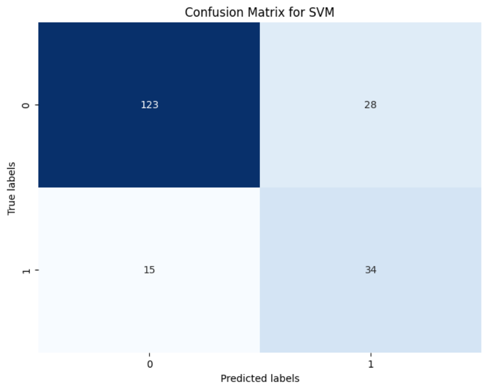
The SVM with random Oversampling has the best F1-score of 0.613 and 0.818 AUC score.



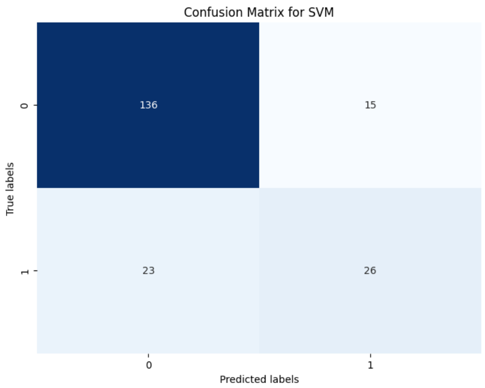
SMOTENC without finetuning



SMOTENC

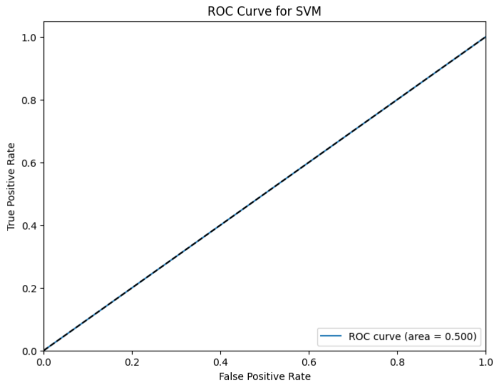


Random Oversampling

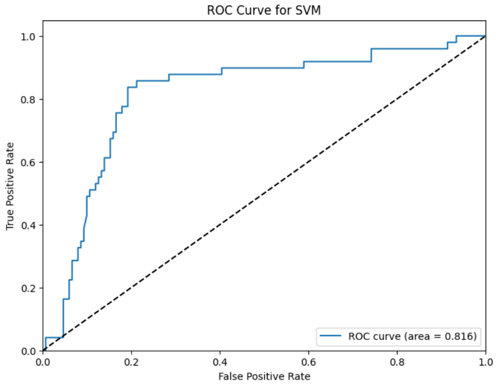


ADASYN

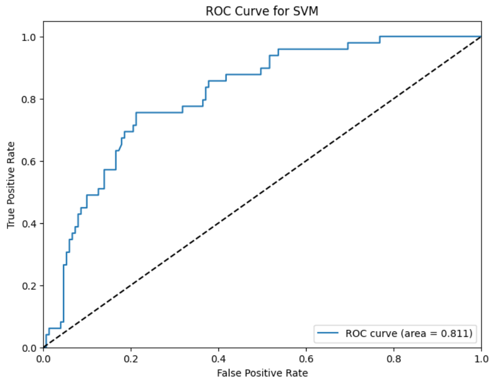
From the confusion matrix, we could see that the SVM model performed quite badly with SMOTE oversampling with high false-positive before tuning and high false-negative after tuning. The False-Negatives decrease post random oversampling while the False-Negatives increase post ADASYN oversampling.



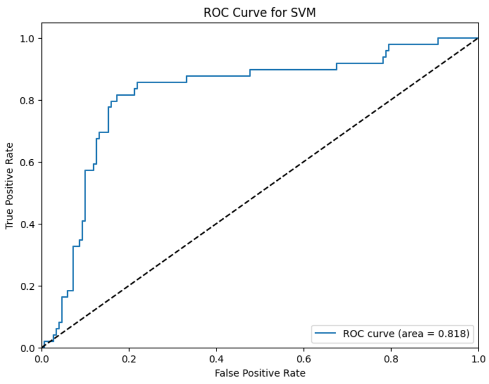
SMOTENC without finetuning



SMOTENC



Random Oversampling



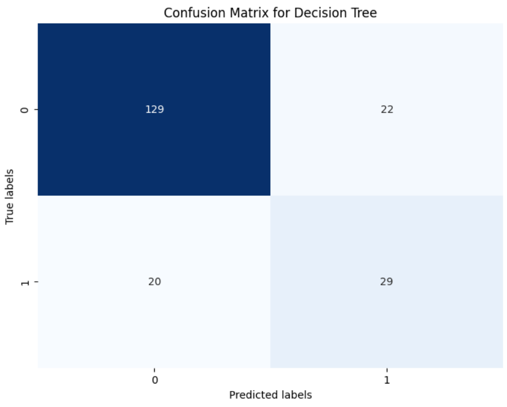
ADASYN

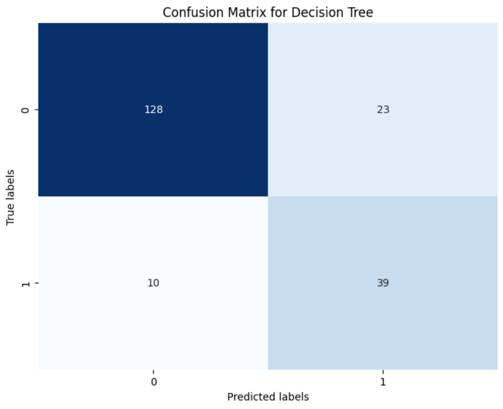
From the ROC curve, we could see that the Baseline SVM performs the worst with the smallest ROC area. The ROC curve of SMOTENC with hyperparameter tuning and ADASYN plateaued at a False Positive Ratio of 0.20

Decision Tree

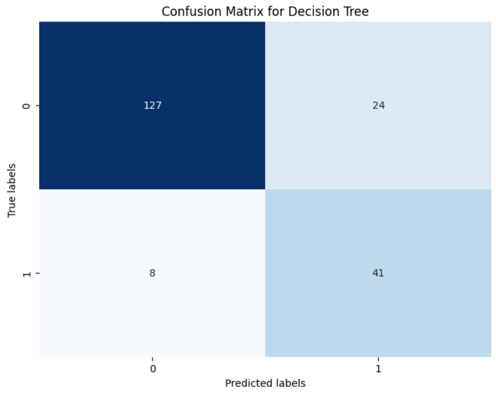
Finetuned?	Imbalanced Dataset Handling	Accuracy	Precision	Recall	F1-Score	AUC Area
False	SMOTENC	0.790	0.569	0.592	0.580	0.723
True	SMOTENC	0.835	0.629	0.796	0.703	0.791
True	Random Oversampling	0.830	0.619	0.796	0.696	0.796
True	ADASYN	0.840	0.631	0.837	**0.719**	**0.833**

The table above shows that the decision tree with ADASYN oversampling strategy produces the best decision tree model based on the F1-score and AUC Area. But, other than the oversampling strategy, it is also important to observe the other hyperparameters as well.

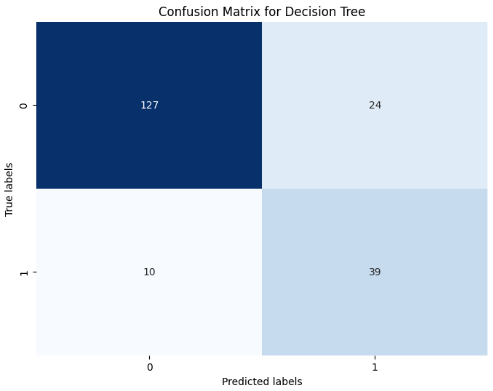




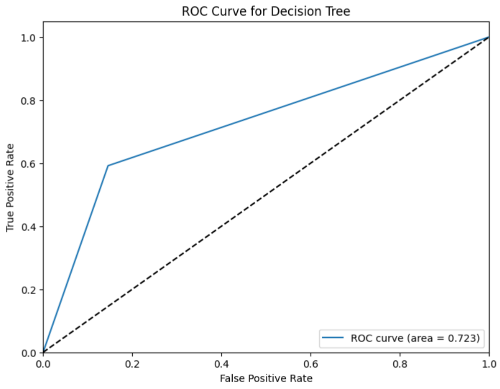
SMOTENC without Fine Tuning **SMOTENC**



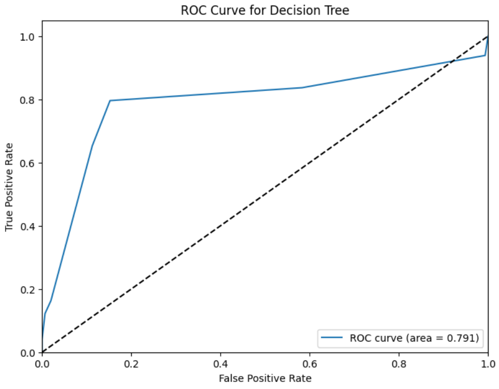
Random Oversampling



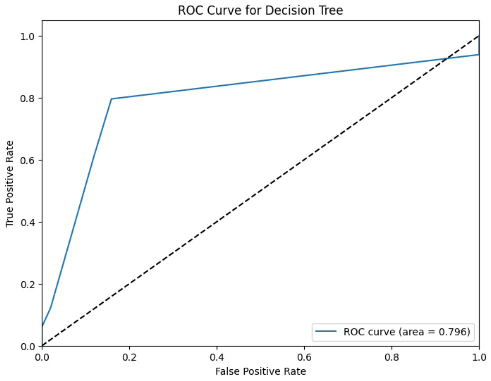
ADASYN



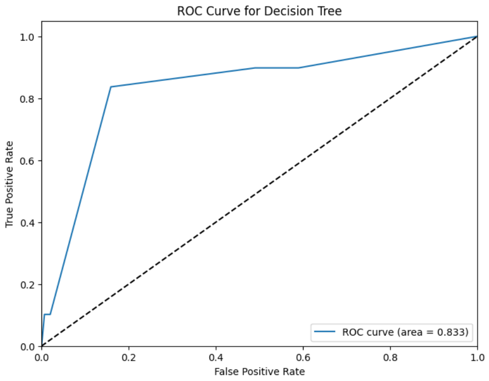
SMOTENC without Fine Tuning



SMOTENC



Random Oversampling



ADASYN

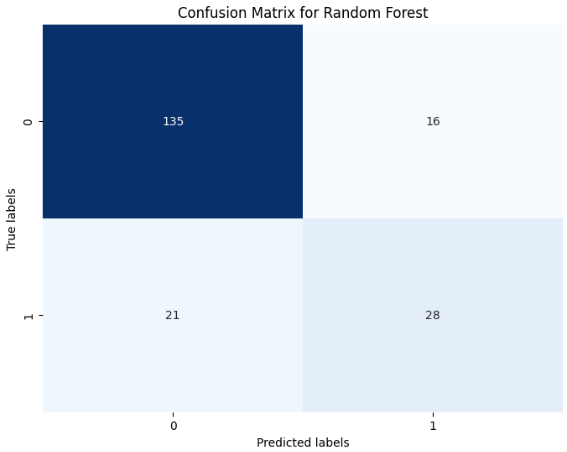
Based on the confusion matrix, our hyperparameter tuning solution has significantly reduced the False-Negative quantities across all the models while the False Positive has slightly increased across these four models.

From the ROC-AUC curves above, the boundary of the ROC AUC curve is quite smooth compared to other ML ROC-AUC curves. In addition, it is observed that the ADASYN oversampling strategy is the best decision tree model as the ROC-AUC curve is the largest.

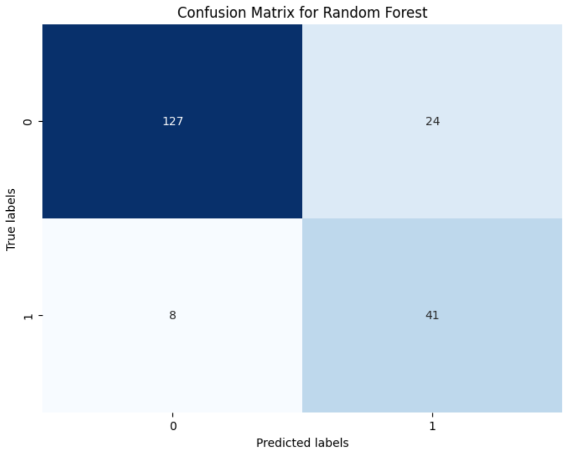
Random Forest

Finetuned	Imbalanced Dataset Handling	Accuracy	Precision	Recall	F1-Score	AUC Area
False	SMOTENC	0.815	0.636	0.571	0.602	0.830
True	SMOTENC	0.84	0.631	0.837	0.719	0.837
True	Random Oversampling	0.845	0.636	0.857	**0.730**	**0.859**
True	ADASYN	0.845	0.636	0.857	0.730	0.824

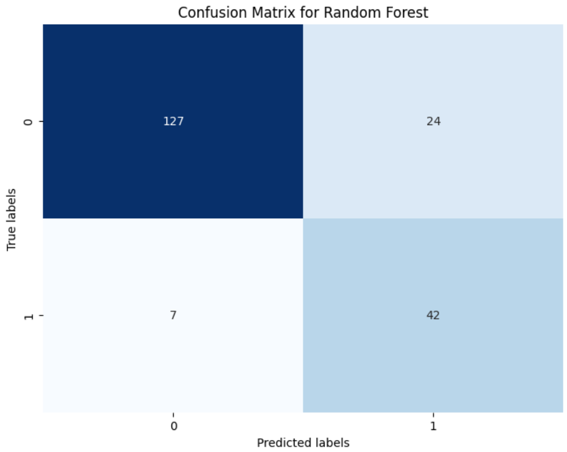
From the table above, the Random Oversampling strategy produces the best model for the F1-Score and AUC area.



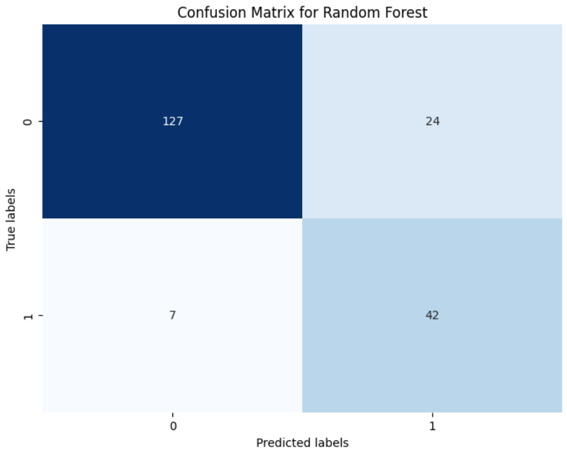
SMOTENC without finetuning



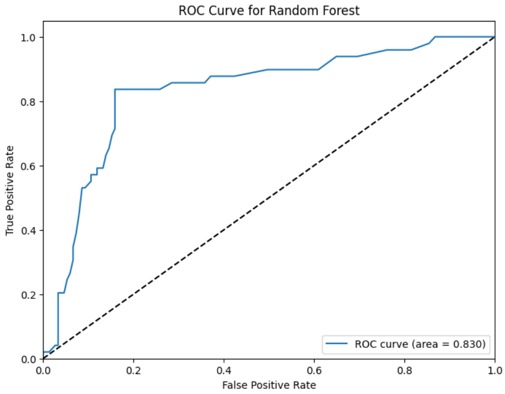
SMOTENC



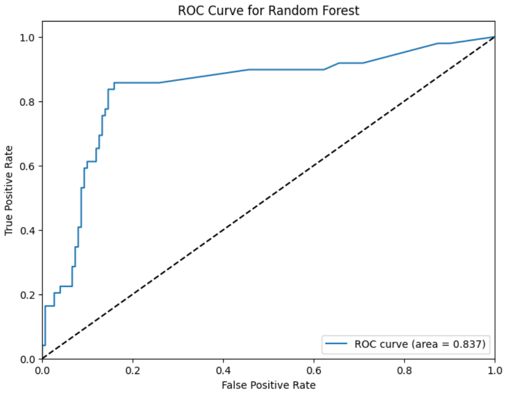
Random Oversampling



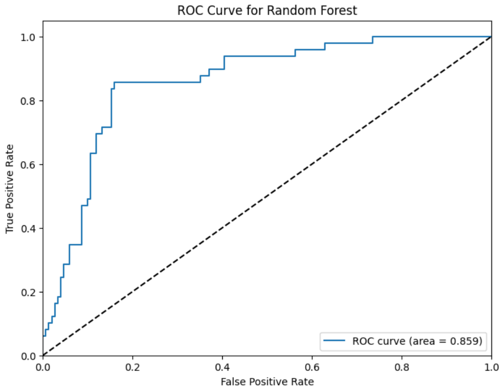
ADASYN



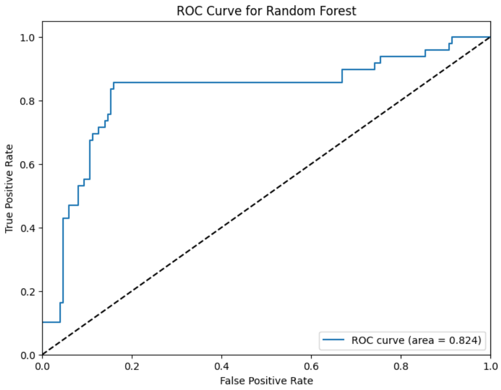
SMOTENC without finetuning



SMOTENC



Random Oversampling



ADASYN

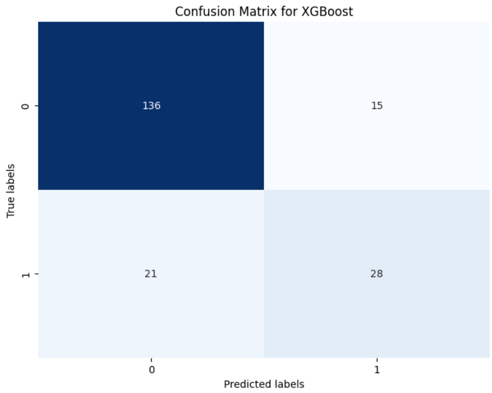
The random forest models are also more skilful in identifying True Negatives instead of True Positives due to the imbalanced dataset. Hence, the models have high accuracy but relatively lower precision/recall.

From the ROC-AUC curves above, the ROC AUC curve's boundary is more "boxy" compared to DT ROC-AUC's boundary. Furthermore, the random forest with random oversampling is the best since its ROC-AUC curve is the largest compared to the other oversampling strategies.

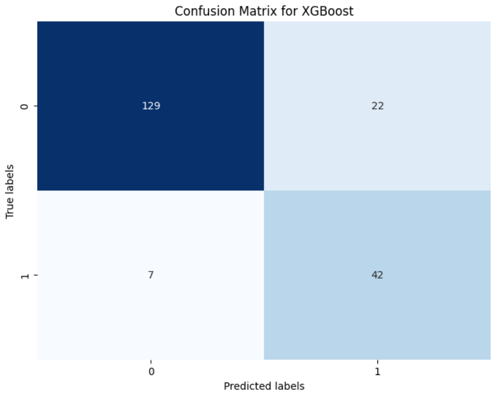
XGBoost

Finetuned	Imbalanced Dataset Handling	Accuracy	Precision	Recall	F1-Score	AUC Score
False	SMOTENC	0.820	0.651	0.571	0.609	0,865
True	SMOTENC	0.855	0.656	0.743	0.70	0.844
True	Random Oversampling	0.845	0.641	0.837	0.726	0.845
True	ADASYN	0.835	0.629	0.796	0.703	0.833

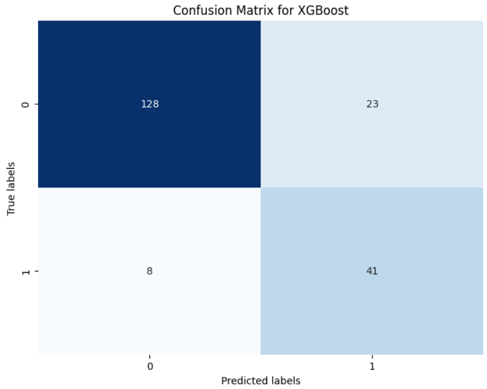
The table shows that the XGBoost with Random Oversampling produced the best F1-Score and AUC score.



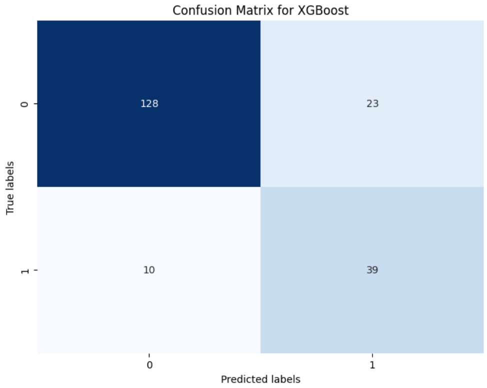
SMOTENC without finetuning



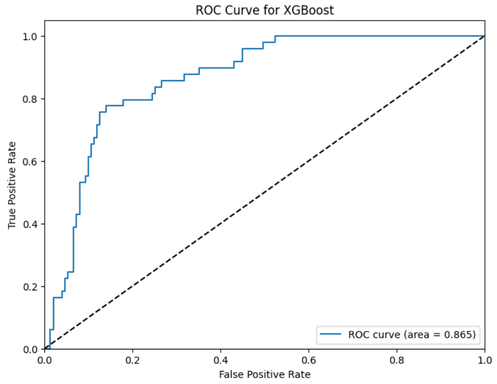
SMOTENC



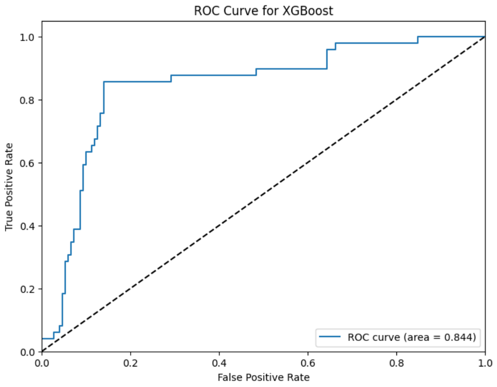
Random Oversampling



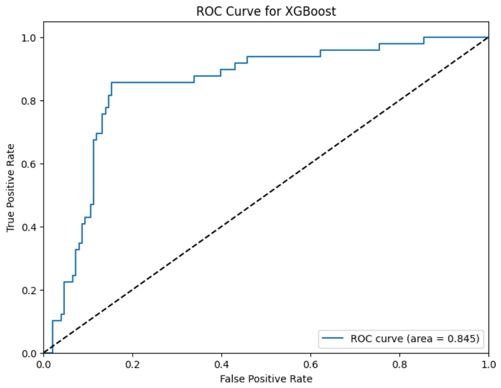
ADASYN



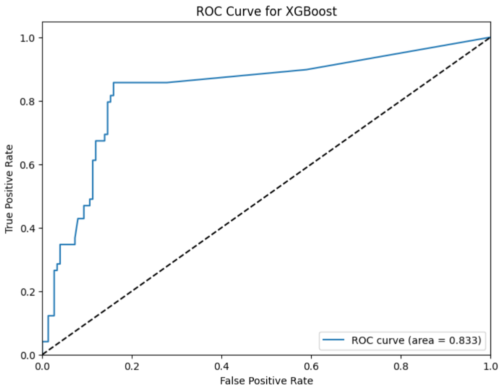
SMOTENC without finetuning



SMOTENC



Random Oversampling



ADASYN

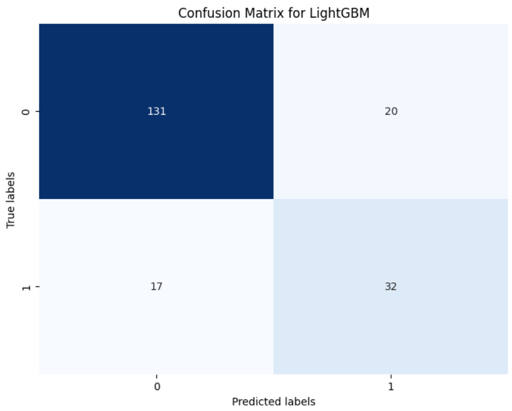
The results above illustrate that finetuning would lead to considerable improvement in model performance, especially its capacity to identify fraud cases, seen from the rise in Recall.

LightGBM

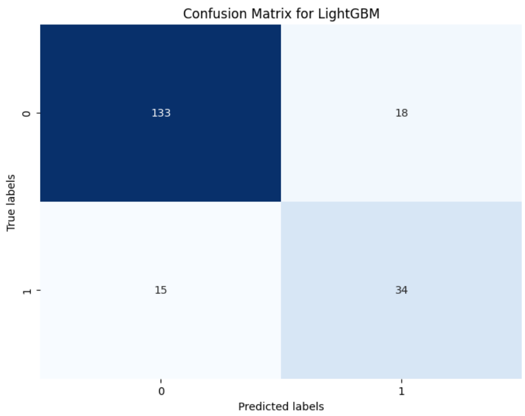
The LightGBM results are as follows:

Fine Tuned	**Imbalanced Dataset Handling**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC Score**
False	SMOTENC	0.815	0.615	0.653	0.634	0.841
True	SMOTENC	0.835	0.654	0.694	0.673	0.862
True	Random Oversampling	0.830	0.619	0.796	0.696	0.854
True	ADASYN	0.850	0.646	0.857	0.737	0.820

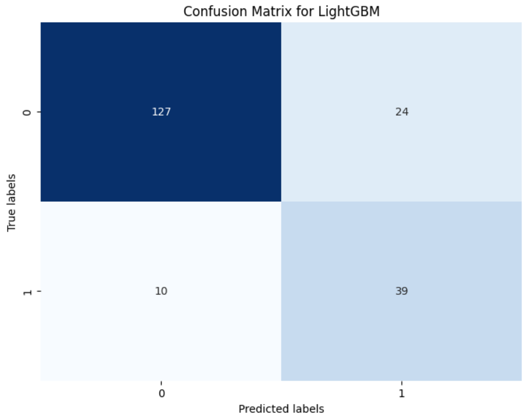
The LightGBM model with the ADASYN has the best F1-score across all models of 0.737.



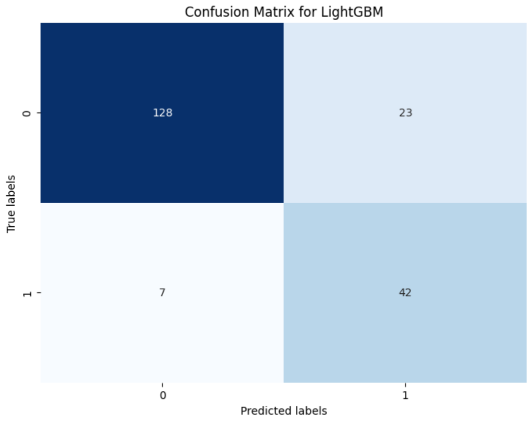
SMOTENC without finetuning



SMOTENC

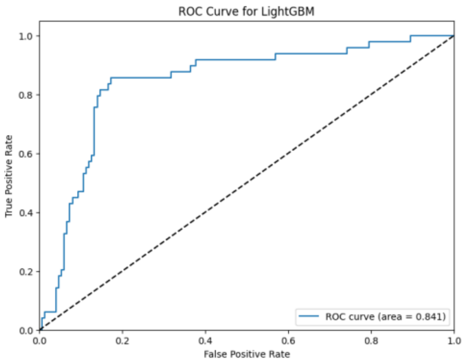


Random Oversampling

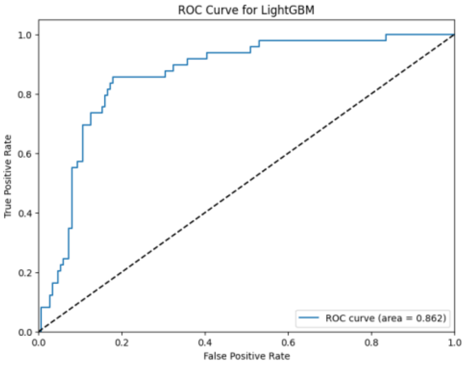


ADASYN

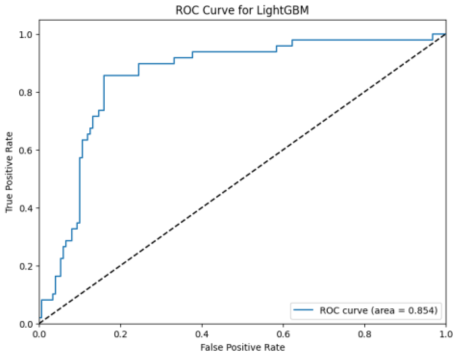
From the confusion matrix above, there is a reduction in the false-negative across three oversampling methods after hyperparameter tuning. For the ADASYN method, there is a great reduction in the false-negatives while having a slight increase in the False-positives compared to the SMOTENC baseline.



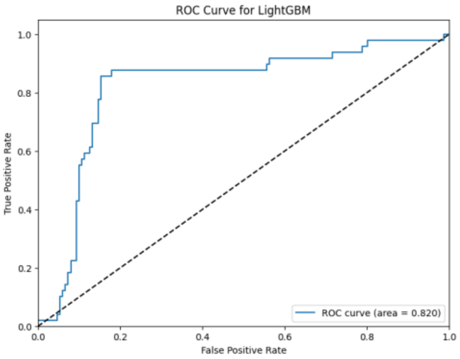
SMOTENC without finetuning



SMOTENC



Random Oversampling

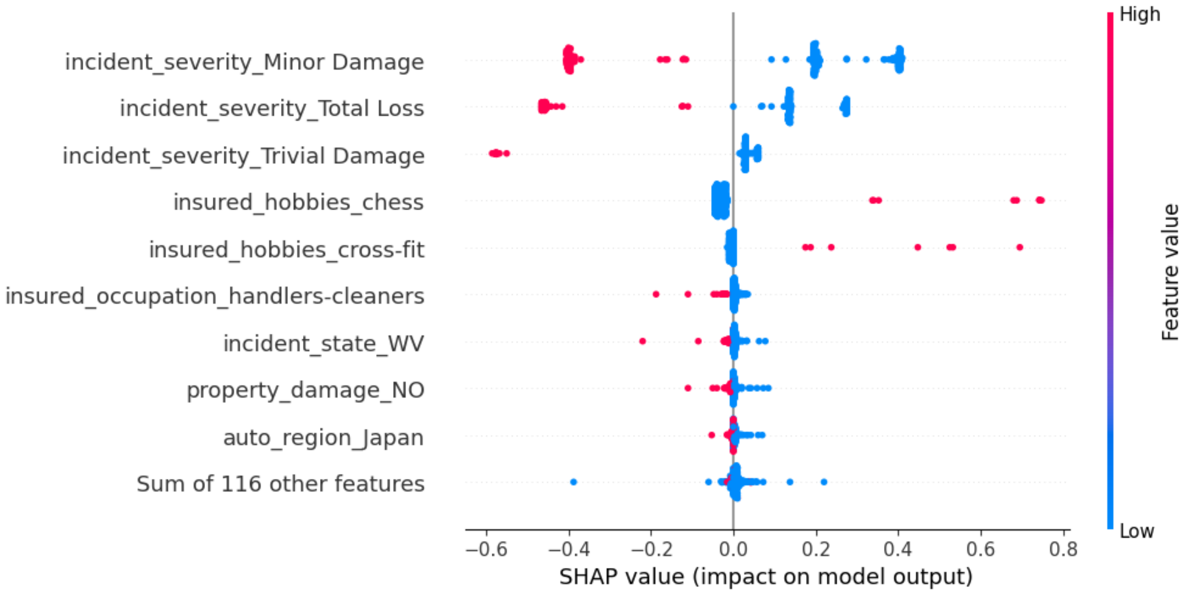


ADASYN

From the ROC curve, we noticed that the AUC is the largest for SMOTENC with hyperparameter tuning. However, the LightGBM with ADASYN is still shortlisted for its high recall despite having a lower AUC score compared to other plots.

Model Explanation

Since LightGBM is the best-performing model in terms of F1-Score, we will plot the SHAP plot for this best model.



In the LightGBM with ADASYN SHAP plot, we could see that the insurance claims that have *incident_severity_Minor Damage*, *incident_severity_Total Loss*, *incident_severity_Trivial Damage*, *insured_occupation_handlers-cleaners, incident_state_WV*, *property_damage_NO*, and *auto_region_Japan* are of low fraud risk. In contrast, insurance clients who declared chess and cross fit as hobbies have the highest fraud risk.

Conclusion

Findings

The model results reveal some surprising outcomes when compared to our hypothesis. To answer the first hypothesis, the car brand and claimant's policy were insignificant predictors. While the feature importance results did not provide us an answer on the significance of exposure theory, fraudster incentive had a definite positive effect on the probability that a claim was fraudulent. Surprisingly, chess and cross fit did in fact have a strong positive effect on fraudulent claims.

To understand our insights better, we had an interview with the head of motor claims 'J' for a major insurance company in Malaysia. Hobby data is not collected when writing a coverage plan, and is not a ground predictor for traditional fraud like fake vehicle theft. However, it played a part when it came to claims exaggeration in bodily injury cases. For example, the injured person without amputation might deliberately buy the most expensive prosthetic (e.g. bionic) when it is not needed. Thus, the insurance company would have to investigate and get a second opinion by obtaining the individuals' hobbies.

Similarly, the major damage factor is not a significant fraud factor. Nonetheless, it is useful for checking claims exaggeration, where the damage is claimed as major.

Significance for the industry

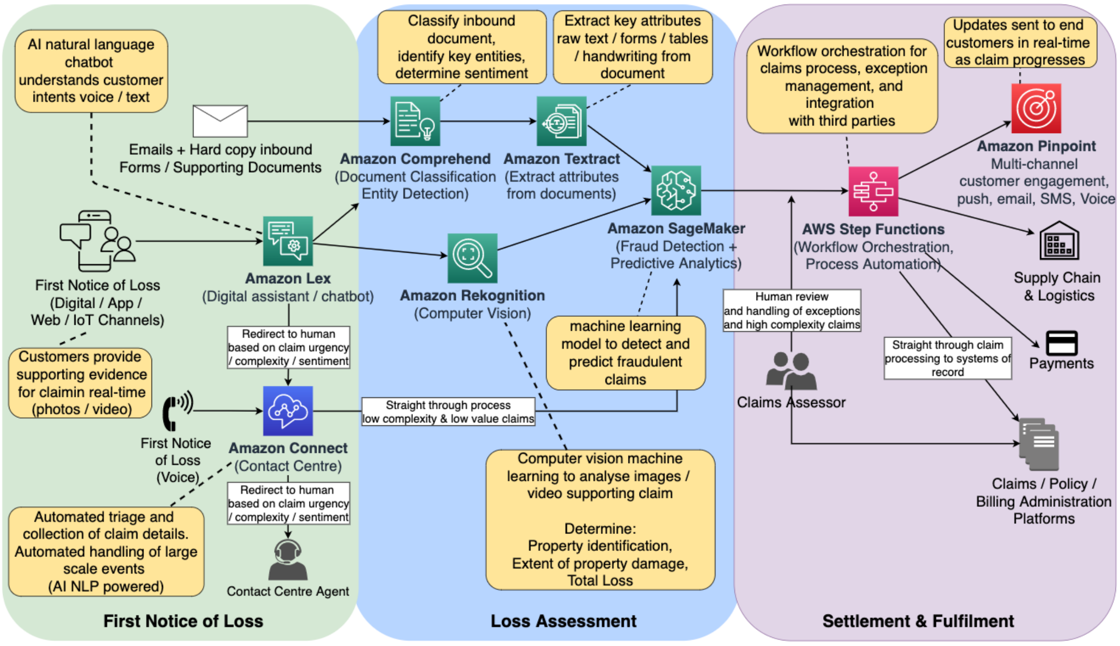
Generally, the insurance industry in the Asia-Pacific (APAC) region has yet to leverage data analytics at its full strength to combat the issue of fraud. The existing AI fraud solutions in the region tend to lack accuracy in detecting fraud, generating high volumes of false alerts (Pathe, 2023).

In Malaysia, major insurance firms utilise a central system known as the Fraud Intelligence System (FIS). Upon the registration of claims, typically by the mechanic shop, users can access a wealth of data about a specific vehicle through FIS, ranging from its previous insurers to its claim history. The system also assigns a score to each claim and flags those falling below a certain threshold as suspicious. This mechanism aids investigators in focusing their attention, and expedites claim approval times. Following the claim registration and data lookup in FIS, an on-the-ground investigation is conducted. If fraud is suspected, external adjusters are brought in to ascertain the company's true liability. This process forms the typical timeline of a motor claim in Malaysia.

In fraud detection, the false negatives need to be minimised, and the LightGBM had a strong recall of 0.86, despite being hindered by the small dataset. Thus, our model fits into step 2 of the claims process, where it provides even stronger accuracy in detecting fraudulent behaviour and can further optimise claims processing times.

Proposed Data Architecture

This report proposed a hypothetical implementation of our ML product on Amazon Web Services (AWS) based on AWS's recommendations on the insurance industry (Cuneo & Singh, 2022). In this hypothetical AWS architecture below, it consists of data collection, claims assessment, and Settlement and Fulfilment. The data collection will start with the claim's registration, from which all relevant variables can be extracted from a central data lake using the licence number of the vehicle as the unique identifier. Consumers who are unable to provide all the needed details at the time of registration can also make use of the Amazon Lex chatbot that will assess what information is still missing and ask for details in order to fill in these values.


AI natural language chatbot understands customer intents voice / text
Emails + Hard copy inbound Forms / Supporting Documents
First Notice of Loss (Digital / App / Web / IoT Channels)
Customers provide supporting evidence for claimin real-time (photos / video)
First Notice of Loss (Voice)
Automated triage and collection of claim details. Automated handling of large scale events (AI NLP powered)
Amazon Lex (Digital assistant / chatbot)
Redirect to human based on claim urgency / complexity / sentiment
Amazon Connect (Contact Centre)
Redirect to human based on claim urgency / complexity / sentiment
Contact Centre Agent
First Notice of Loss
Classify inbound document, identify key entities, determine sentiment
Extract key attributes raw text / forms / tables / handwriting from document
Amazon Comprehend (Document Classification Entity Detection)
Amazon Textract (Extract attributes from documents)
Amazon SageMaker (Fraud Detection + Predictive Analytics)
Amazon Rekognition (Computer Vision)
machine learning model to detect and predict fraudulent claims
Straight through process low complexity & low value claims
Computer vision machine learning to analyse images / video supporting claim
Determine: Property identification, Extent of property damage, Total Loss
Loss Assessment
Workflow orchestration for claims process, exception management, and integration with third parties
Updates sent to end customers in real-time as claim progresses
Amazon Pinpoint Multi-channel customer engagement, push, email, SMS, Voice
AWS Step Functions (Workflow Orchestration, Process Automation)
Supply Chain & Logistics
Human review and handling of exceptions and high complexity claims
Straight through claim processing to systems of record
Payments
Claims Assessor
Claims / Policy / Billing Administration Platforms
Settlement & Fulfilment

After preprocessing, claims officers can view the dataset. Our model will be situated in Amazon SageMaker. The model will ingest the preprocessed dataset and make preliminary investigations. After the data is passed through the model and a classification is made, explainability visual aids will be provided to evaluate the model, and on-the-ground investigations based on the evidence provided could be launched. If the claim passes the relevant checks, it will progress to the settlement and fulfilment stage. Otherwise, a flag will also be raised to delay the claims payment process.

Limitation and Future Work

The main limitation of this project is the dataset's size. With 1000 observations, we cannot investigate yearly trends and monthly seasonality, which may help in understanding collusion and syndicates in fraud. Another limitation that we faced is the absence of data that claims adjusters currently use to make judgements on claims. This includes visual data such as photos of the accident scene, and when the accident report was filed. In addition, late reporting is a big red-flag according to ‘J’, especially if it is a multi-car collision and the third party reports are late as well.

Future work should focus on testing the model on much larger datasets to assess generalisability, along with visual data analysis. One possible application of ML’s fraud detection is to use Optical Character Recognition (OCR) to assess surface damage on vehicles. Ultimately, the preliminary results from our study have been encouraging, and we see opportunities for further expansionary work towards combating motor insurance fraud with ML in the future.

(4984 words excluding sections, headings, and table annotation)

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Aqqad, A. (2023a). insurance_claims. *Mendeley Data*, *2*. https://doi.org/10.17632/992mh7dk9y.2

Aqqad, A. (2023b, September 8). *Leveraging machine learning techniques for enhanced detection of insurance fraud claims: An empirical study*. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4552815

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20–29. https://dl.acm.org/doi/10.1145/1007730.1007735

The Brainy Insights. (2023, November 16). *Auto insurance market size worth $1764.9 billion by 2032 - increased auto sales and laws mandating insurance to explode demand*. Yahoo! Finance. https://finance.yahoo.com/news/auto-insurance-market-size-worth-210000858.html

Chauhan, N. S. (2020, January). *Decision tree algorithm, explained*. KDnuggets. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (1AD). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357. https://dl.acm.org/doi/10.5555/1622407.1622416

Chen, S., Kuhn, M., Prettner, K., & Bloom, D. E. (2019). The global macroeconomic burden of road injuries: Estimates and projections for 166 countries. *The Lancet Planetary Health*, *3*(9). https://doi.org/10.1016/s2542-5196(19)30170-6

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Cuneo, N., & Singh, G. (2022, June 16). *Zero touch claims – how P&C insurers can optimize claims processing using AWS AI/ML Services* | *Amazon Web Services*. AWS for Industries. https://aws.amazon.com/blogs/industries/zero-touch-claims-how-pc-insurers-can-optimize-claims-processing-using-aws-ai-ml-services/

GIAS. (2018). Motor Insurance Fraud - Protect yourselves from Insurance Fraud. Singapore; General Insurance Association of Singapore.

He, H., Bai, Y., Garcia, E. A., & Shutao Li. (2008). Adasyn: Adaptive Synthetic Sampling Approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. https://doi.org/10.1109/ijcnn.2008.4633969

Hedges & Company. (2024, February 2). *How many cars are there in the world? statistics by country.* Automotive Market Research. https://hedgescompany.com/blog/2021/06/how-many-cars-are-there-in-the-world/

imblearn. (2024). *SMOTENC#*. SMOTENC - Version 0.12.2. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTENC.html

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., & Ma, W. (4AD). LightGBM: a highly efficient gradient boosting decision tree. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 3149–3157. https://doi.org/https://dl.acm.org/doi/10.5555/3294996.3295074

LTA. (2024a, February 16). Insurance. https://onemotoring.lta.gov.sg/content/onemotoring/home/owning/ongoing-car-costs/insurance.html

LTA. (2024b, March 12). *MOTOR VEHICLE POPULATION BY VEHICLE TYPE*. Statistics. https://www.lta.gov.sg/content/dam/ltagov/who_we_are/statistics_and_publications/statistics/pdf/MVP01-1_MVP_by_type.pdf

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *NIPS 2017*. https://doi.org/10.48550/arXiv.1705.07874

Microsoft. (2024). *Parameters tuning*. Parameters Tuning - LightGBM 4.3.0.99 documentation. https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

Pathe, T. (2023, February 16). *Motor insurance scams lose speed as MSIG Singapore adopts AI* | *the Fintech Times*. Insurtech. https://thefintechtimes.com/motor-insurance-scams-lose-speed-as-msig-singapore-adopts-ai/

Probasco, J. (2024, March 12). *Everything you need to know about choosing the right auto insurance*. Fortune Recommends. https://fortune.com/recommends/insurance/what-is-auto-insurance/

Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical Bayesian Optimization of Machine Learning Algorithms*. https://doi.org/10.48550/arXiv.1206.2944

Tibshirani, R., Hastie, T., Witten, D., & James, G. (2021). *An introduction to statistical learning: With applications in R*. Springer.

Appendix

Data Dictionary

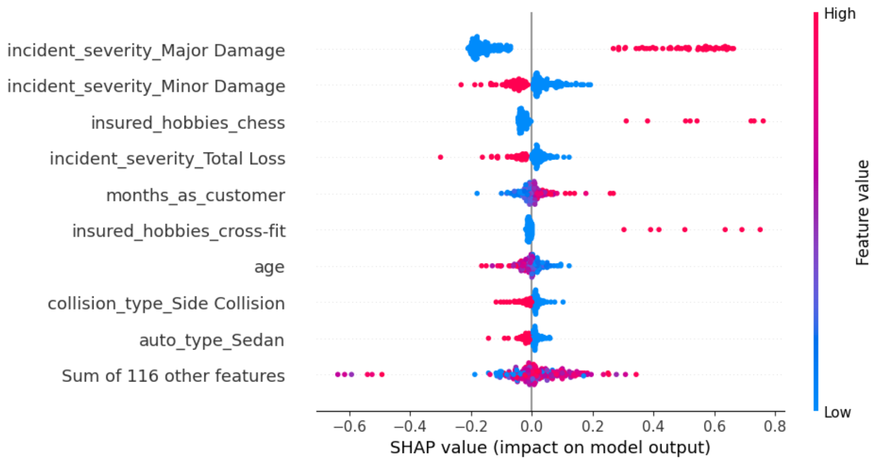
Column Name	Description	Column Type	Feature Engineering Action
months_as_customer	Duration the individual has been a customer, in months.	numerical	
age	Age of the insured person.	numerical	
policy_number	Unique number assigned to the insurance policy.	numerical	drop
policy_bind_date	Date when the policy was initiated.	date	Create new feature policy_age_during_incident_in_days, by substracting it with the incident_date
policy_state	State where the policy was issued.	categorical	
policy_csl	Combined Single Limit – Coverage limit per accident.	numerical	str split from the EDA notebook, convert the values in the splitted columns into integers, then normalize it
policy_deductable	Amount the insured pays out of pocket before the insurer pays the claim.	numerical	
policy_annual_premium	Yearly premium amount for the insurance policy.	numerical	
umbrella_limit	Additional liability coverage limit beyond the standard policy.	numerical	
insured_zip	ZIP code of the insured individual.	categorical	drop
insured_sex	Gender of the insured individual.	categorical	
insured_education_level	Education level of the insured (e.g., MD, PhD, Associate).	categorical	
insured_occupation	Occupation of the insured person.	categorical	
insured_hobbies	Hobbies of the insured individual.	categorical	
insured_relationship	Relationship status of the insured (e.g., husband, unmarried).	categorical	
capital-gains	Profit from the sale of assets like stocks or property.	numerical	
capital-loss	Loss from the sale of assets.	numerical	
incident_date	Date when the incident occurred.	date	retrieve the year column since it is needed for the auto_year
incident_type	Type of incident (e.g., Vehicle Theft, Single Vehicle Collision).	categorical	
collision_type	Specific type of collision (e.g., Rear Collision, Front Collision).	categorical	
incident_severity	Severity of the incident (e.g., Minor Damage, Major Damage).	categorical	
authorities_contacted	Authorities that were contacted post-incident (e.g., Police, Fire).	categorical	
incident_state	State where the incident took place.	categorical	

incident_city	City where the incident occurred.	categorical	drop
incident_location	Exact location/address of the incident.	categorical	drop
incident_hour_of_the_day	Hour of the day when the incident occurred.	categorical	12 hours interval augmentation
number_of_vehicles_involved	Total number of vehicles involved in the incident.	categorical	
property_damage	Whether there was property damage (YES, NO, or ? if uncertain).	categorical	
bodily_injuries	Number of bodily injuries sustained during the incident.	ordinal	
witnesses	Number of witnesses present at the time of the incident.	ordinal	
police_report_available	Indicates whether a police report was available (YES, NO, or ? if uncertain).	categorical	
total_claim_amount	Total claim amount for the incident.	numerical	
injury_claim	Claim amount for injuries.	numerical	
property_claim	Claim amount for property damages.	numerical	
vehicle_claim	Claim amount for vehicle damages.	numerical	
auto_make	Make/brand of the vehicle involved.	categorical	Country of origin Japan, Germany, USA, etc
auto_model	Model of the vehicle involved.	categorical	Categorise it as SUV, HatchBack, Sedan
auto_year	Year of manufacture of the vehicle.	date	use my year column created from year_column to count the car age
fraud_reported	Indicates whether the claim was fraudulent (Y for Yes, N for No).	categorical	

SHAP Plots for Other Models

This section contains the Model Explanation discussions for all models except for the best-performing one.

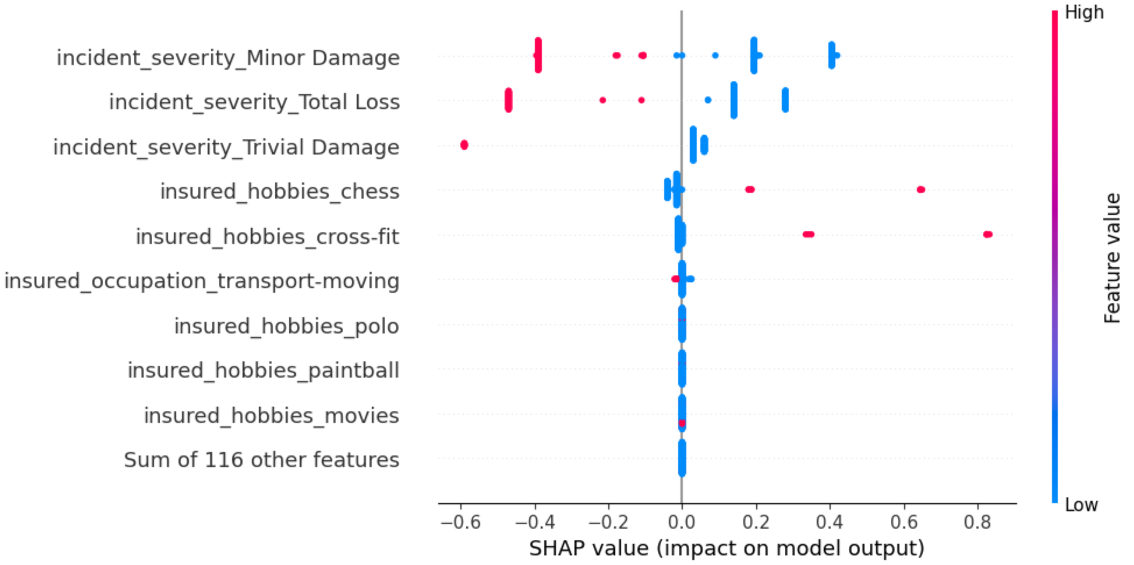
Logistic Regression



From the LR with Random Oversampling's SHAP plot, the *incident_severity_Major Damage*, *months_as_customer*, *insured_hobbies_chess*, and *insured_hobbies_cross-fit* as predictors of fraudulent insurance claims. In contrast, the *incident_severity_Minor Damage*, insurance claims with *incident_severity_Total Loss*, lower age, *collision_type_Side Collision* and *auto_type_Sedan* were at low risk of being a fraud.

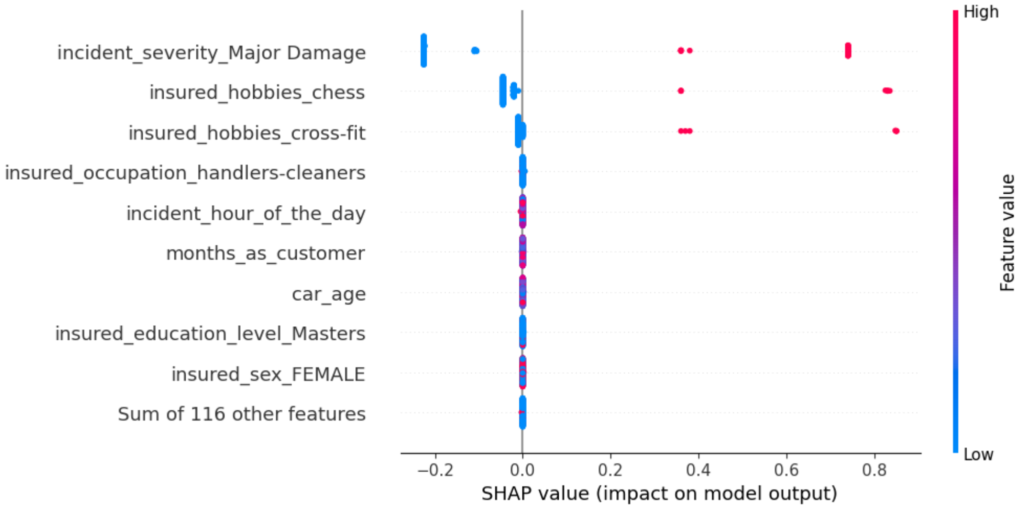
Decision Tree

We will investigate the SHAP plot for the decision tree with ADASYN.



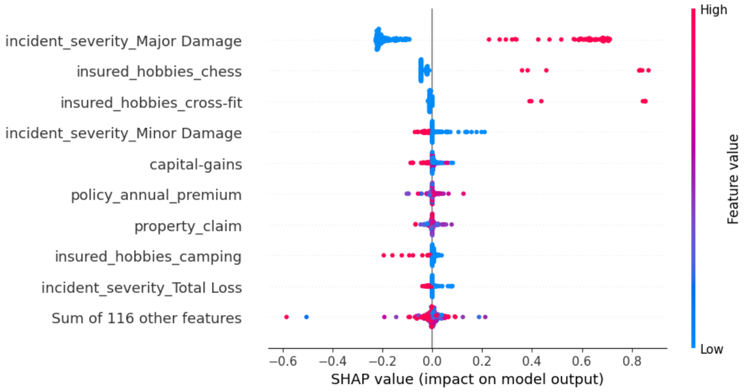
From the SHAP plot, the *incident_severity_Minor Damage*, *incident_severity_Total Loss* and *incident_severity_Trivial Damage* contributed to a lower probability of fraudulent insurance claims. Interestingly, most fraud cases were associated with the insurance clients that have chess and crossfit as hobbies.

Random Forest



Based on the SHAP plot for random forest with oversampling, incident severity and insurance clients with chess or crossfit hobbies have a notable impact on determining whether a claim is fraudulent. Additionally, apart from these variables -- *incident_severity with Major Damage*, *insured_hobbies_chess*, and *insured_hobbies_crossfit* -- no other factors significantly influenced the classification of fraudulent claims, which was unexpected.

XGBoost



As XGBoost attains the best performance in terms of F1 score using a Random Oversampler, this setup will be used to generate the Model Explanation results. From the Shapley value results, making a claim for Major Damage, and having chess and cross-fit as hobbies indicates higher chances of fraud. This corroborates the findings from our EDA, where hobbies and damage type are expected to be strong predictors of fraud.

Model Tuning Analysis

Logistic Regression

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
Solver	LBFGS	LBFGS	Newton-CG	LBFGS
C (Regularisation)	1.0	5.84	0.14	2.97

Among all the finalised parameters, the common solver used for SMOTENC is LBFGS. Larger values of C were used for SMOTENC and ADASYN, which means that regularisation is reduced to let the Logistic regression to fit more with the training data. In contrast, the best model used Newton-CG as solver and a smaller C, indicating stronger regularisation.

SVM

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
C (Regularisation)	1.0	1.96	0.54	0.58
Kernel	RBF	Linear	Linear	Linear
Degree	-	-	-	-
Coef0	-	-	-	-

The “Linear” parameter is the common parameter shortlisted by the parameter. Although the other parameters Since the SVM results were lacklustre, the SHAP plot would not be plotted for SVM.

Decision Tree

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
max_depth	2	4	3	3
min_samples_split	2	2	2	6
min_samples_leaf	1	1	2	2

From the hyperparameters tuning table, we postulated that the decision tree might work well with ADASYN due to its relatively higher *min_samples_split* value compared to the other oversampling strategies (other than None). The higher *min_samples_split* value reduces the likelihood of unnecessary splitting, causing the model to overfit less.

Random Forest

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
n_estimators	100	200	500	400
min_samples_split	2	3	5	5
min_samples_leaf	gini	gini	gini	gini

min_samples_split	2	2	7	9
min_samples_leaf	1	9	8	2
max_features	sqrt	None	None	None
bootstrap	True	True	True	True

From the final hyperparameter table above, most of the better RF models used a high number of trees as it can reduce overfitting. Additionally, they also use higher maximum depth values which allows the models to understand the data in more detail, preventing underfitting from happening.

XGBoost

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
n_estimators	100	33	449	40
max_depth	6	15	8	35
learning_rate	0.300	0.060	0.200	0.010
Reg_alpha	0	3.300	4	0.400
Reg_lambda	0	1.700	3.00	3.800
subsample	1	0.850	0.950	0.800
colsample_bytree	1	0.950	0.850	1

Under the tuned parameters, the model undergoes increased regularisation, as evidenced by higher values assigned to *Reg_alpha* and *Reg_lambda*, representing first and second-order regularizations applied to the loss function, respectively. There is also a reduction in the number of estimators utilized, except in cases where a Random Oversampler is employed, resulting in fewer trees being utilized in the model. Notably, the reduction in the number of estimations also implies increased regularisation.

LightGBM

	SMOTENC (Baseline)	SMOTENC (Tuned)	Random Oversampler	ADASYN
n_estimators	100	100	500	250
num_leaves	31	148	82	5
learning_rate	0.100	0.149	0.0388	0.0164

subsample	1	0.850	0.200	0.600
colsample_bytree	1	0.750	0.950	0.900
min_child_samples	20	11	16	15
reg_alpha	0	1.507	6.257	0.700
reg_lambda	0	2.651	0.056	2.052

Generally, the *reg_alpha* and *reg_lambda* have increased across different oversampling methods after performing hyperparameter tuning. Among the oversampling methods, the LightGBM with ADASYN has the smallest *num_leaves*, and this reduces the likelihood of overfitting.

Meeting Agendas and Contributions List

Week 8 Meeting [APPENDIX A]

Date: Sunday, 17th March 2024
Time: 11:00AM SGT
Attendance:

-

Objective:

- Choosing Dataset
- Project Timeline Discussion
- Code Structure Discussion

Choosing Dataset
Choices: https://docs.google.com/spreadsheets/d/104iB0qnb48vX8ms-7851ULfPCTF3mxYflCDRB4B_PVA/edit#gid=0
Finalized: Insurance Dataset (https://data.mendeley.com/datasets/992mh7dk9y/2)

Project Timeline Discussion
Link: https://docs.google.com/document/d/1GEbTo6bdDMZR3HsYV3B0fnX6-X9qeg5G9ic4LnjwbK4/edit

Any feedback?

- Guan Yee:
    - From previous project, 1 person handle EDA
    - For this project, since EDA is important, we can delegate EDA based on features (not exclusively) meaning can still work on other features as well
- Peizhi:
    - Better use any platform for report don’t have to be latex, as long as everyone can contribute

        - Better start EDA + Preprocessing soon because more time needs to be allocated to report + PPT

Code Structure Discussion

Link:
https://docs.google.com/document/d/161tYBvyk8qYn4rCKUoHIJf5eXV_N7nxW32SVKUMkoio/edit

Any feedback?

- Regarding the EDA, better work on a single notebook for the whole team, so that it's much easier to create the main.ipynb
- For model finetuning, it's fine for each of us to work on 1 notebook; as the finalized parameters and model classes will be written in the file model.py in the end

What needs to finish by Week 9 Sunday?

- EDA + Preprocessing + Feature Engineering Delegation
    - Kevin
        - Months_as_customer
        - Age
        - Policy_number
        - Policy_bind_date
        - Policy_state
        - Policy_csl
        - Policy_deductable
        - Policy_annual_premium
        - Umbrella_limit
        - insured_zip

    - Guan Yee
        - Insured_sex
        - Insured_education_level
        - Insured_occupation
        - Insured_hobbies
        - Insured_relationship
        - Capital-gains
        - Capital-loss
        - Incident_date
        - Incident_type
        - collision_type

    - Peizhi
        - Incident_severity
        - Authorities_contacted
        - Incident_state
        - Incident_city
        - Incident_location

        - Incident_hour_of_the_day
        - Number_of_vehicles_involved
        - Property_damage
        - Bodily_injuries
        - witnesses
    - Vivek
        - Police_report_available
        - Total_claim_amount
        - Injury_claim
        - Property_claim
        - Vehicle_claim
        - Auto_make
        - Auto_model
        - Auto_year
        - Fraud_reported

Week 9 Meeting [APPENDIX B]

Date: Sunday, 24TH March 2024
Time: 11AM-12:30PM SGT
Attendance:

- Kevin
- Guan Yee
- Peizhi
- Vivek

EDA Updates

- Guan Yee:
    - Vivek, See if we can see the difference in price of vehicles depending on the brand, the model, as well as the year
    - Vivek, Try to find the country of the brand as well as the rough price of them
    - Kevin, try to see the distribution of the annual policy premium grouped by education
    - Peizhi, try to see if it's possible to reduce the dimension of the data by aggregating the hours into 2 hour-interval
- Peizhi:
    - Let's see how is the distribution of each feature group by the target variable

Precision vs Recall?
F1 Score + ROC-AUC

Model Delegation
https://scikit-learn.org/stable/supervised_learning.html
seed = 42
train_test_split = 0.8 / 0.2
stratify = Y (True)
All models

- Random Forest (Kevin)
- Decision Tree (Kevin)
- Logistic Regression (Vivek)
- AdaBoost / Catboost (Guan Yee)
- SVM Classifier (Vivek)
- XGBoost2 (Peizhi)
- MLP / LightGBM (Guan Yee)

Hyperparameter Tuning: Optuna (MIT License)

What to do next week

- Preprocessing should be done by next week
- Start to build the modelling class, instead of just waiting for the preprocessed dataset

Week 10 Meeting [APPENDIX C]

Date: Sunday, 31st March 2024
Time: 3PM-4:30PM SGT
Attendance:

- Kevin
- Guan Yee
- Peizhi
- Vivek

Objective

1. Finalized Preprocessing
2. Modelling code structure
3. Essay delegation

Essay Writing (Format)

1. Abstract
2. Introduction
3. Data
4. Methods
5. Results
6. Conclusion

Essay Delegation

1. Abstract + github README.md -> Kevin
2. Introduction + Problem statement (don't forget to create data pipeline architecture) -> Vivek
3. Literature Review / Related Work (not too long) -> **[OPTIONAL]**
4. Data (include EDA Insights) [Pick the most important, no need to include all features]

    a. Data Description -> Guan Yee
    b. Data insights -> Peizhi + Guan Yee
    c. Data preprocessing ->Guan Yee

5. Methods **(All of us)**
    a. Model writeup
        i. Kevin
            1. Decision Tree
            2. Random Forest
        ii. GY
            1. LightGBM
            2. MLP
        iii. Peizhi
            1. XGBoost
        iv. Vivek
            1. Log regression (vivek)
            2. SVM
    b. Choice of hyperparameter tuning with Optuna
6. Results **(All of us)**
    a. Model F1, Recall, Precision table for baseline vs oversampled vs tuned.
    b. Confusion Matrix
    c. Explain the model part + reasons why you think this feature is valid predictor for fraudulent insurance claims detection
7. Conclusion + Future work + limitation

Rough Space for Brainstorming

Week 11 (Preprocessing + Modelling), week 12 (1.5 days) + 1 day week 13

Preprocessing + Modelling + Essay + EDA + PPT

Preprocessing -> new features (guan yee) [tonight] -> guan yee will let me know

Next fridays -> modelling parameters done!

Peizhi -> LightGBM + Parameters tuning (0.5-1 day)

Week 10

- Sunday -> Guan Yee will finish the preprocessing

Week 11

- Class (models) -> methods train, predict, evaluate, explain (by friday)

- Set individual parameters, put somewhere first
- Friday -> Modelling parameters should be completed
- Saturday + Sunday -> Essay
- Sunday -> meeting -> potential of creating python scripts for modelling

Week 12

-

Week 11

Week 11 Meeting [APPENDIX D]

Date: Sunday, 7th April 2024
Time: 10:30AM SGT
Attendance:

- Kevin
- Guan Yee
- Peizhi
- Vivek

Objective:

- Delegation of Essay
- Delegation of PPT
- Business Problem + Product Solutioning
- ML Results
- Next steP

Delegation of Essay

- Abstract + GitHub README.md -> Kevin
- Introduction + Problem statement (don't forget to create data pipeline architecture) -> Vivek
- Literature Review / Related Work (not too long) -> **[OPTIONAL]**
- Data (include EDA Insights) [Pick the most important, no need to include all features]
    - Data Description -> Guan Yee
    - Data insights -> Peizhi + Guan Yee
    - Data preprocessing ->Guan Yee
- Methods **(All of us)**
    - Model writeup
        - Kevin
            - Decision Tree
            - Random Forest
        - GY
            - LightGBM

            - MLP
        - Peizhi
            - XGBoost
        - Vivek
            - Log regression (Vivek)
            - SVM
    - Choice of hyperparameter tuning with Optuna
- Results **(All of us)**
    - Model F1, Recall, Precision table for baseline vs oversampled vs tuned.
    - Confusion Matrix
    - Explain the model part + reasons why you think this feature is a valid predictor for fraudulent insurance claims detection **[Vivek]**
- Conclusion + Future work + limitation **[DON'T KNOW]**

Delegation of PPT (10 mins)

- Agenda + Problem Statement -> 2 min [**Kevin**]
- Insights (EDA) -> 3 mins [Guan Yee]
- Methods (Preprocessing + Models (high-level)) -> 2 mins [Peizhi]
- Results (Precision/Recall + Feature Importance) + Limitation -> 3 mins **[Vivek]**

Business Problem + Product Solutioning

https://docs.google.com/document/d/1GRqNECKZFAcdd_td1DJUrCFg25BKJ5K1TzKJeqh213c/edit

ML Results

https://github.com/kevinchs0808/DSA4263-Project/tree/dev

Next Step

Kevin will create README.md on Dev and will make a pull request to staging which needs everyone's approval

Brainstorm

- Preprocessing ->
    - Discuss about the oversampling
    - What are the Models (no need to really show how it works)
        - Random Forest
        - Decision Tree
    - How finetuning works
    - Precision / Accuracy
    - Feature Importance
- Modelling Side:
    - Should we do one-hot encoding?
        - If True
            - For example, the features will be ranked based on specific values (like specific hobbies)
        - If False
            - Show the high level importance

- Which model should be shown in the presentation
- Some hypothesis:
    - Incident_Severity may be the most important feature [**Random Forest**]
    - Incident Hobbies is overlooked because splitted across multiple classes [**Concern**]
- Severity + Hobbies are important to show
- Show both high level + individual (both sides)
    - Use XGBoost for high level illustration
    - Use another model to show the detail
- Concern:
    - If we use individual features some high level features may not be shown
    - But, stakeholders may want to know the exact scenario of when the fraud is likely to happen

Writing Section	Author	Editor
Abstract	Kevin Christian	Sun Peizhi
Introduction	Vivek Bagai	Sun Peizhi
Data	Loo Guan Yee	Sun Peizhi
EDA (Methods)	Sun Peizhi (Hypothesis 1\2) Loo Guan Yee (Summary statistics and Hypothesis 3)	Loo Guan Yee Sun Peizhi
ML Preprocessing (Methods)	Loo Guan Yee (Train-test split and feature engineering) Sun Peizhi (Resampling Method)	Loo Guan Yee Sun Peizhi
ML performance Enhancement (Methods)	Sun Peizhi (Hyperparameter tuning) Guan Yee (K-Fold Stratified cross validation)	Sun Peizhi Loo Guan Yee
Machine Learning Approaches (Methods)	Kevin Christian (RF and DT) Vivek Bagai (LR and SVM) Sun Peizhi (XGBoost) Loo Guan Yee (LightGBM)	Sun Peizhi Loo Guan Yee
Results	Kevin Christian (RF and DT) Sun Peizhi (XGBoost) Loo Guan Yee (LR, SVM and LightGBM)	Sun Peizhi Loo Guan Yee
Conclusion	Vivek Bagai	Loo Guan Yee