Vehicle Insurance Fraud Detection

DS105 Project Presentation

Yeo Siew Ping

Content

- Dataset Introduction
- Problem Statement
- Summary of Approach
- Exploratory Data Analysis
- Key Steps for Data-Preprocessing
- Machine Learning
  - Overview of Approaches
  - Evaluation Metrics
  - Model Training and Performance
  - Selection of Best Model
- Conclusion and Way Ahead

1

Dataset Introduction

1. Dataset Introduction

- Vehicle insurance fraud involves conspiring to make false or exaggerated claims involving property damage or personal injuries following an accident
- Dataset consists of vehicle and insurance-related details
- Taken from Kaggle (https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection) and is originally a real-life fraud machine learning case study used by Oracle
- Columns : 33
- Rows: 15420

1. Dataset Introduction

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15420 entries, 0 to 15419
Data columns (total 33 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Month                 15420 non-null  object
 1   WeekOfMonth           15420 non-null  int64
 2   DayOfWeek             15420 non-null  object
 3   Make                  15420 non-null  object
 4   AccidentArea          15420 non-null  object
 5   DayOfWeekClaimed      15420 non-null  object
 6   MonthClaimed          15420 non-null  object
 7   WeekOfMonthClaimed    15420 non-null  int64
 8   Sex                   15420 non-null  object
 9   MaritalStatus         15420 non-null  object
 10  Age                   15420 non-null  int64
 11  Fault                 15420 non-null  object
 12  PolicyType            15420 non-null  object
 13  VehicleCategory       15420 non-null  object
 14  VehiclePrice          15420 non-null  object
 15  FraudFound_P          15420 non-null  int64
 16  PolicyNumber          15420 non-null  int64
 17  RepNumber             15420 non-null  int64
 18  Deductible            15420 non-null  int64
 19  DriverRating          15420 non-null  int64
 20  Days_Policy_Accident  15420 non-null  object
 21  Days_Policy_Claim     15420 non-null  object
 22  PastNumberOfClaims    15420 non-null  object
 23  AgeOfVehicle          15420 non-null  object
 24  AgeOfPolicyHolder     15420 non-null  object
 25  PoliceReportFiled     15420 non-null  object
 26  WitnessPresent        15420 non-null  object
 27  AgentType             15420 non-null  object
 28  NumberOfSuppliments   15420 non-null  object
 29  AddressChange_Claim   15420 non-null  object
 30  NumberOfCars          15420 non-null  object
 31  Year                  15420 non-null  int64
 32  BasePolicy            15420 non-null  object
dtypes: int64(9), object(24)
memory usage: 3.9+ MB
```

- There are 1 continuous features and 32 categorical features
- Label – FraudFound_P (0,1)

2

Problem Statement

2. Problem Statement

- In this project, we aim to help the insurance company to filter out potential fraud cases and minimise actual fraud cases
- **End Goal** : Create a machine learning model to predict if a specific vehicle insurance claim is a fraudulent one
- Supervised classification model – predict if case is fraudulent or not (Binary Classification)

2. Problem Statement

- **Sub-goals**:
  - How do the features vary for fraud cases ?
  - How do the **demographics** (e.g. Age, Gender, Marital Status) vary with the features for fraud cases?
    - There was a decreasing trend for fraud from 1994 to 1996 - why? Were there a difference in the demographics along the years?
    - As most of the vehicles involved in fraud were priced from $20000–$29000 and mostly Sedan, what were the demographics for this group of fraudsters?
    - For the common car models in fraud cases, what were the demographics like?

3

Summary of Approach

3. Summary of Approach

1. Import relevant libraries and dataset
2. Initial data exploration
3. Initial data cleaning and wrangling
4. Exploratory data analysis
   - *to address the sub-goals*
5. Data-preprocessing and feature engineering
6. Machine Learning
   - *to address the end-goal*
7. Conclusion and Way Ahead

3. Summary of Approach

1. Import relevant libraries and dataset
2. Initial data exploration
3. Initial data cleaning and wrangling
4. Exploratory data analysis
   - *to address the sub-goals*
5. Data-preprocessing and feature engineering
6. Machine Learning
   - *to address the end-goal*
7. Conclusion and Way Ahead

4

Exploratory Data Analysis

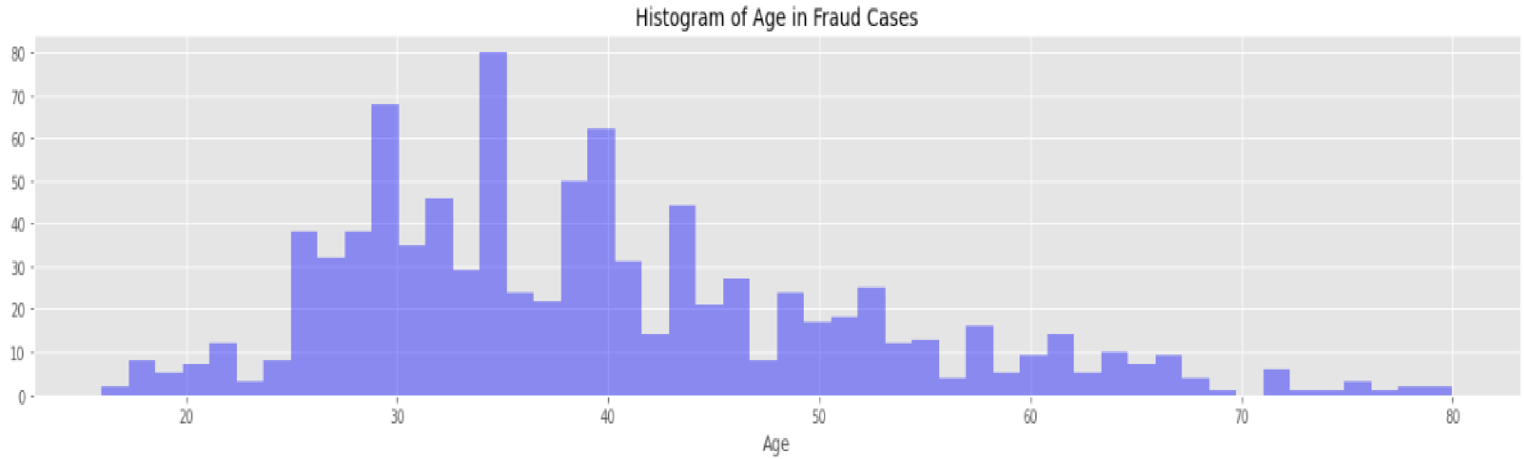
Addressing the Sub-Goals

1. How do the features vary for fraud cases ?

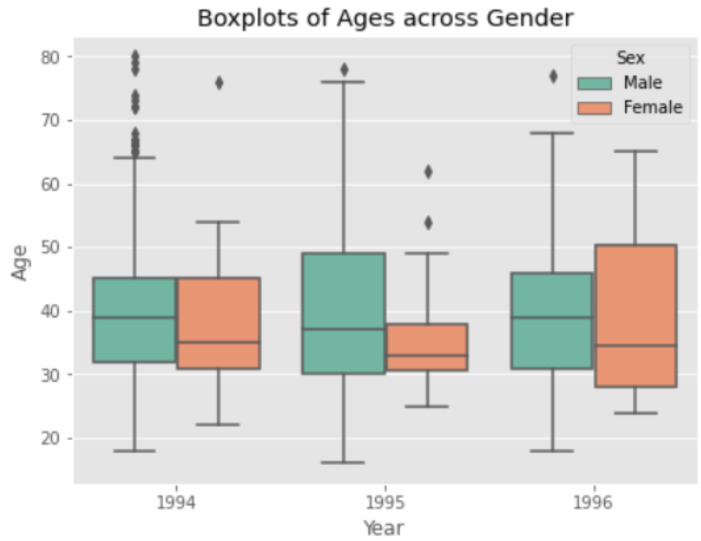
Time Features	Binary Features	Multi-class Features
January, March and May	Urban areas	Pontiac, Honda, Toyota
Week 2 , 3	Male	Policy Holder are usually Married or Single
Mondays, Fridays	Policy holder at fault	Sedan with vehicle price from 20000-29000
Fraud claims are usually made on a weekday instead of weekend	Police report not made	Policy deductible is usually at $400
	Witness not present	Claim made more than 30 days after policy purchase
	Policy under external agents	Vehicle age usually more than 6 years
		Age of policyholder usually from 31-50
		Number of suppliments mostly zero
		No address change
		Decreasing trend seen from 1994 to 1996

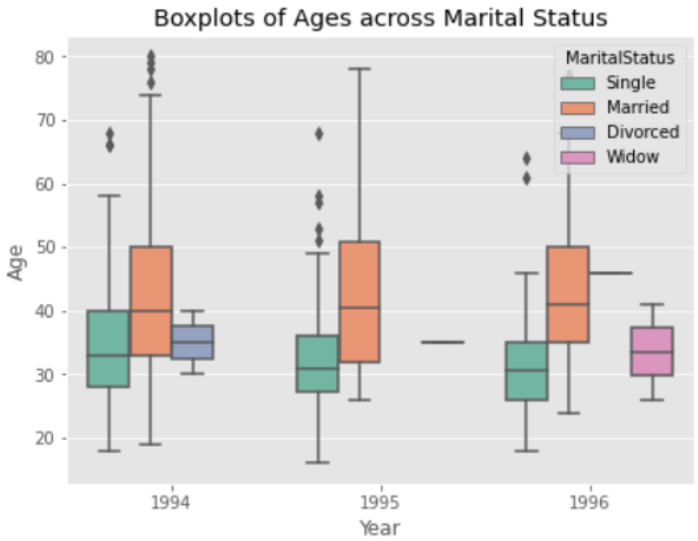
1. How do the features vary for fraud cases ?

- Age of person involved in fraud accident
- Slightly right-skewed → Log transformation



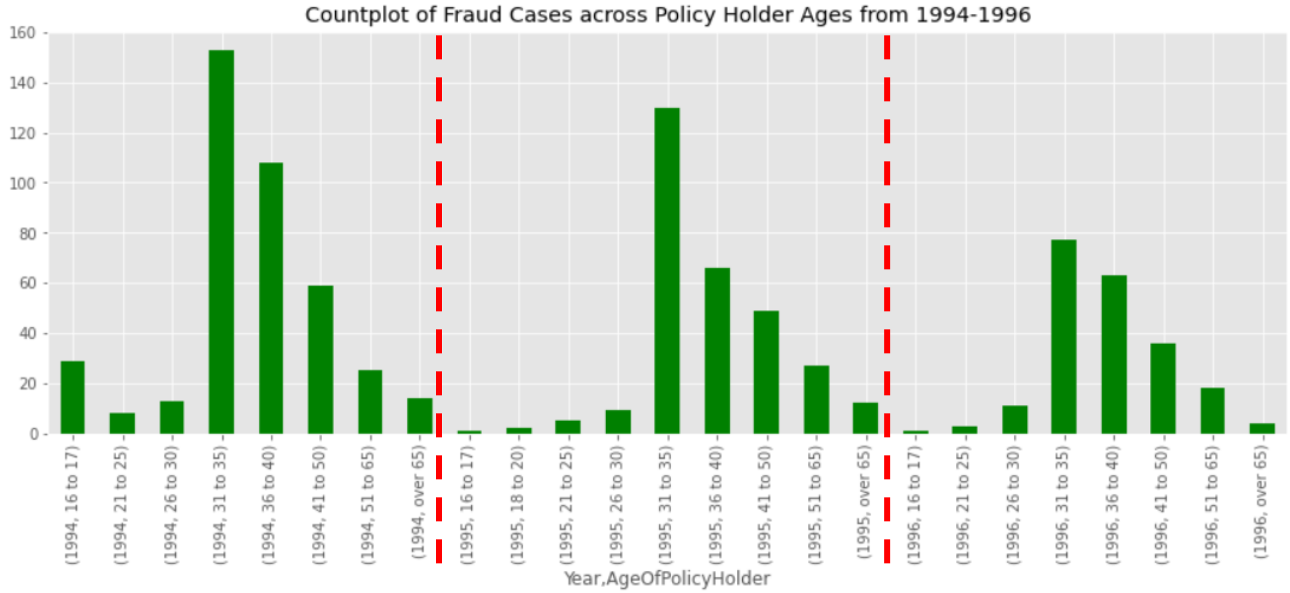
2. There was a decreasing trend for fraud from 1994 to 1996 - why? Were there differences in the demographics across the years?





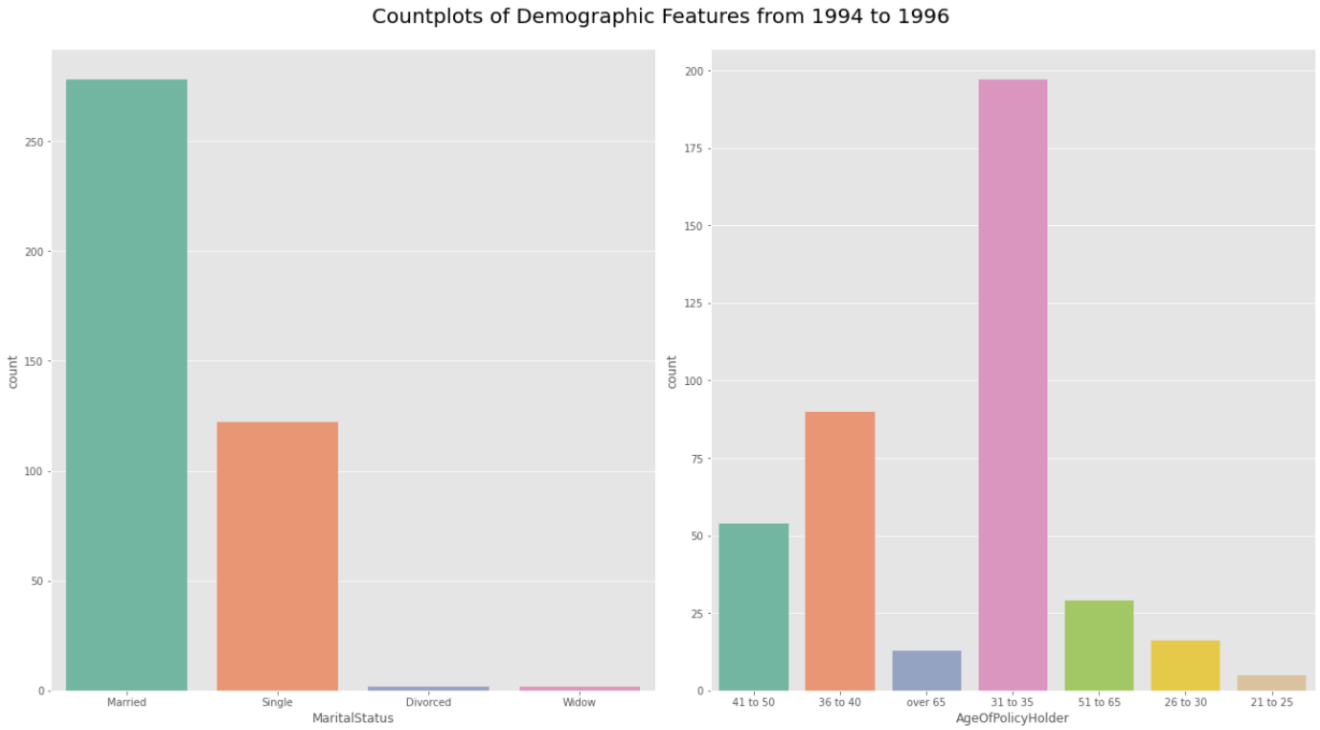
- Average age of both gender did not fluctuate much over the years
- Age for singles decreased over the years

2. There was a decreasing trend for fraud from 1994 to 1996 - why? Were there differences in the demographics along the years?



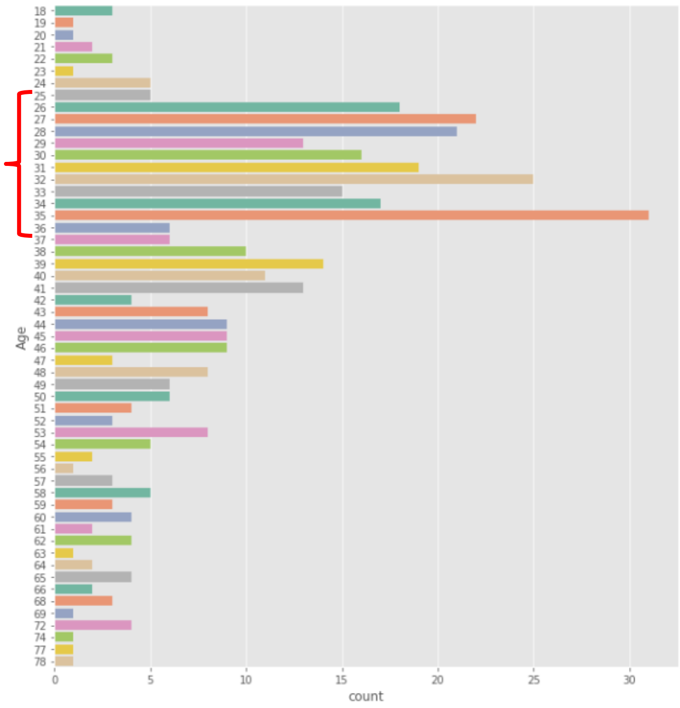
- Age of policy holder follows the decreasing trend over the years

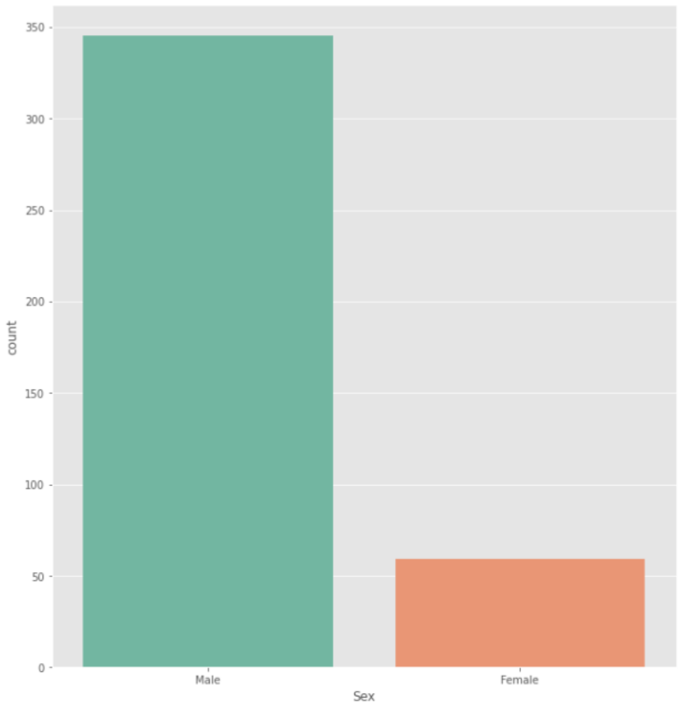
3. As most of the vehicles involved in fraud were priced from $20000 - $29000 and mostly Sedan, what were the demographics for this group of policyholders?



- Most of the fraudsters are Married or single
- Age group of policy holder – 31 to 35

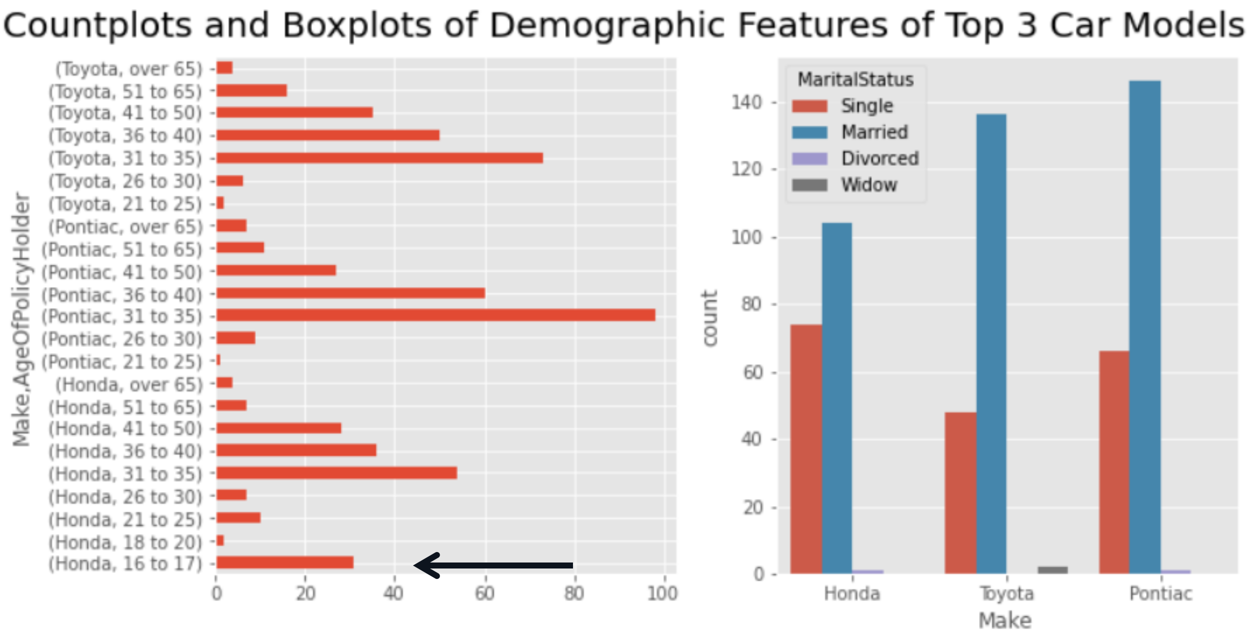
3. As most of the vehicles involved in fraud were priced from $20000 - $29000 and mostly Sedan, what were the demographics for this group of policyholders?





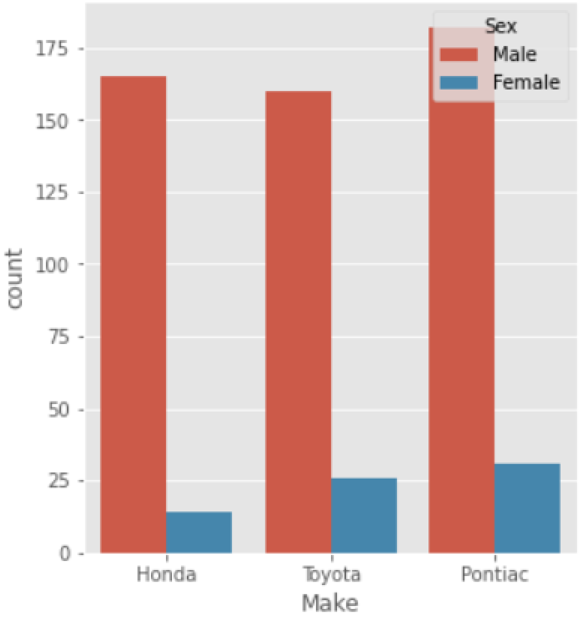
- Most of the fraudsters are male
- Age of person involved in fraud accident – 26 to 35 years old

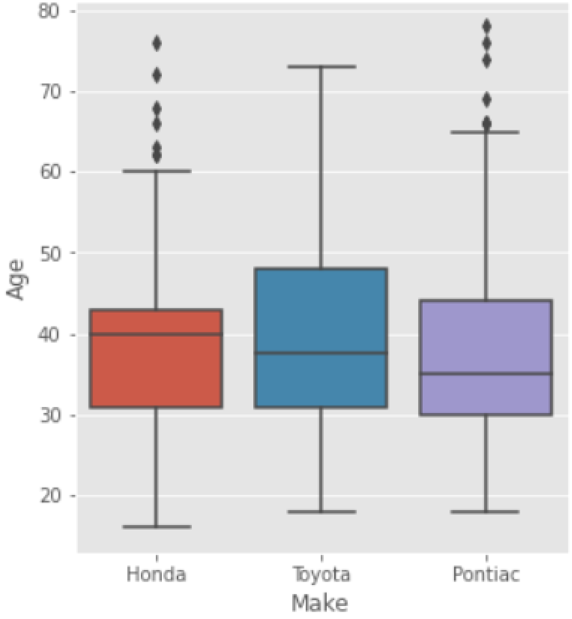
4. For the common car models in fraud cases, what were the demographics like?



- Top 3 car models involved in fraud cases - Pontiac, Honda and Toyota
- Honda had a significant higher number of policy holders in age group 16-17 years old

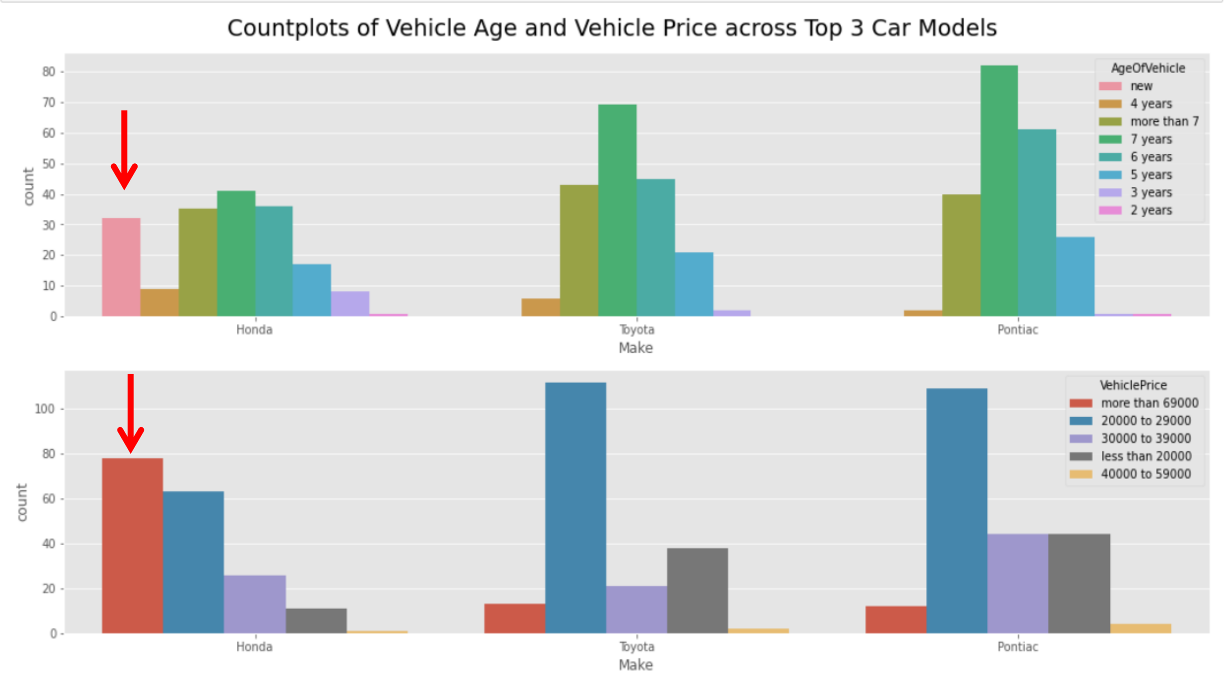
4. For the common car models in fraud cases, what were the demographics like?





- Although Honda had a significant group of young policyholders, the average age of person involved in accident was the highest
- Possible misuse of policy by fraudsters due to mismatch in ages
- Possible higher payout for young policy holders; Lax regulation by Honda

4. For the common car models in fraud cases, what were the demographics like?



- Honda vehicles used in fraud were much younger and more expensive (> $69000)
- All brand new cars priced above $69000 in fraud cases were from Honda
- Brand new or expensive Honda cars can get higher payout?

5

Key Steps for Data Pre-Processing

5. Data-Preprocessing

- Binning for the following features:
  - Make
  - Marital Status
  - Days_Policy_Accident
  - Days_Policy_Claim
  - AddressChange_Claim
  - NumberOfCars

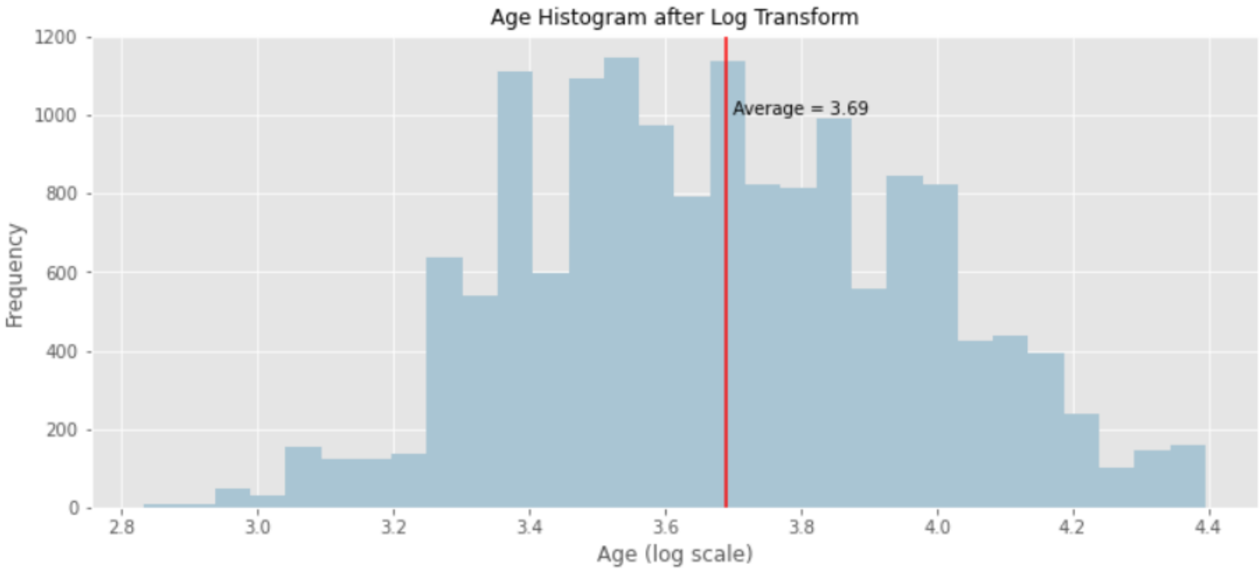
5. Data-Preprocessing

As a guide for encoding in this project, only features with max 3 categories will be considered for dummy encoding.

No.	Get_dummies	Ordinal Encoding	Frequency Encoding	Label
1	AccidentArea	VehiclePrice	Month	FraudFound_P
2	Sex	Deductible	WeekOfMonth	
3	Fault	DriverRating	DayOfWeek	
4	PoliceReportFiled	PastNumberOfClaims	DayOfWeekClaimed	
5	WitnessPresent	AgeOfVehicle	MonthClaimed	
6	AgentType	AgeOfPolicyHolder	WeekOfMonthClaimed	
7	MaritalStatus	NumberOfSuppliments	Make	
8	Days_Policy_Accident		RepNumber	
9	Days_Policy_Claim			
10	AddressChange_Claim			
11	NumberOfCars			
12	PolicyType			
13	VehicleCategory			
14	Year			
15	BasePolicy			

5. Data-Preprocessing

- Log Transformation for Age Column



6

Machine Learning

Addressing End-Goal

Overview of Approaches

- As this dataset is highly imbalanced (6% fraud cases) , techniques are applied to address this problem
- Oversampling methods are only applied to the train set and not the entire dataset → prevent bias and data leakage
- Multiple approaches :
    1. Using oversampling method ADASYN and standard scaler
    2. Using oversampling method SMOTE and standard scaler
        - Stacking multiple models (Ensemble Technique)
    3. Using oversampling method SMOTE, min-max scaler and standard scaler
    4. Adjusting class weights in model training and standard scaler

Decide on the best model and conduct hyperparameter tuning

Overview of Approaches

- Classification algorithms to be explored:
  - Logistic Regression
  - K-Nearest Neighbour
  - Naives Bayes
  - Random Forest
  - XGBoost
  - Gradient Boosting
  - Support Vector Machine

Evaluation Metrics

- **Recall** – Out of total fraud cases, how many fraud cases have been caught by the model?
- **Precision** – How many fraud cases are classified correctly?
- **ROC-AUC Score –** How well does the model perform under different probability thresholds
- **Precision-Recall curve –** Trade-off between precision and recall for different probability threshold

- **Best Model**

  1. High Recall Score – catch as many fraud cases
  2. Reasonable Precision – reduce false positives and investigation cost
  3. Good AUC Score
  3. Reasonable Precision – Recall Curve

Model Training and Performance

1. Using oversampling method ADASYN and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.62	0.737
KNN	0.12	0.50	0.567
Naives Bayes	0.12	0.81	0.770
Random Forest Classifier	0.32	0.04	0.777
XGBoost	0.38	0.07	0.729
SVM	0.15	0.42	0.495
Gradient Boosting	0.00	0.00	0.777

Model Training and Performance

1. Using oversampling method ADASYN and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.62	0.737
KNN	0.12	0.50	0.567
Naives Bayes	0.12	0.81	0.770
Random Forest Classifier	0.32	0.04	0.777
XGBoost	0.38	0.07	0.729
SVM	0.15	0.42	0.495
Gradient Boosting	0.00	0.00	0.777

Hyperparameter Tuning

Precision	Recall	AUC Score
0.13	0.69	0.784

0.09	0.87	0.726

- Recall and AUC Score improved but precision for Naives Bayes model is bad

Model Training and Performance

2. Using oversampling method SMOTE and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.62	0.732
KNN	0.13	0.53	0.572
Naives Bayes	0.12	0.82	0.767
Random Forest Classifier	0.50	0.06	0.768
XGBoost	0.44	0.06	0.766
SVM	0.15	0.42	0.491
Gradient Boosting	0.00	0.00	0.776

Model Training and Performance

2. Using oversampling method SMOTE and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.62	0.732
KNN	0.13	0.53	0.572
Naives Bayes	0.12	0.82	0.767
Random Forest Classifier	0.50	0.06	0.768
XGBoost	0.44	0.06	0.766
SVM	0.15	0.42	0.491
Gradient Boosting	0.00	0.00	0.776

Model Training and Performance

2. Using oversampling method SMOTE and standard scaler

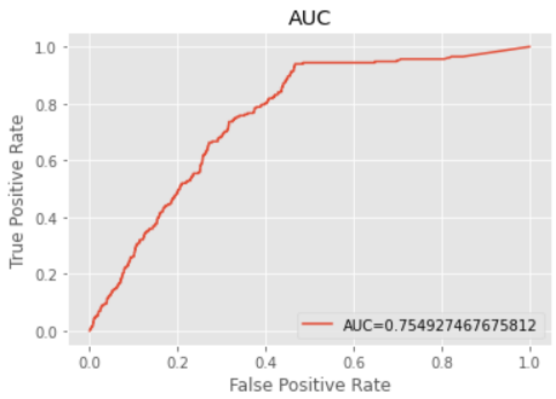
- Stacking Models (Ensemble Technique) with class weights adjusted

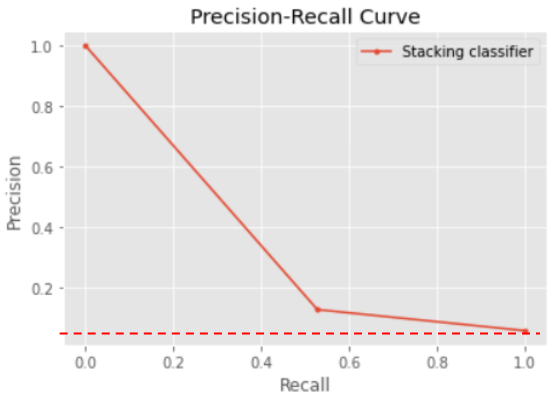
Precision	Recall	AUC Score
0.13	0.53	0.755

Model Training and Performance

2. Using oversampling method SMOTE and standard scaler

- Stacking Models (Ensemble Technique) with class weights adjusted





Model Training and Performance

3. Using oversampling method SMOTE, min-max scaler and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.64	0.516
KNN	0.12	0.53	0.519
Naives Bayes	0.12	0.82	0.495
Random Forest Classifier	0.36	0.06	0.529
XGBoost	0.44	0.06	0.472
SVM	0.14	0.53	0.506
Gradient Boosting	0.00	0.00	0.637

Model Training and Performance

3. Using oversampling method SMOTE, min-max scaler and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.64	0.516
KNN	0.12	0.53	0.519
Naives Bayes	0.12	0.82	0.495
Random Forest Classifier	0.36	0.06	0.529
XGBoost	0.44	0.06	0.472
SVM	0.14	0.53	0.506
Gradient Boosting	0.00	0.00	0.637

- Precision and Recall did not change much
- AUC Score fares even poorly

Model Training and Performance

4. Adjusting class weights in model training and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.91	0.796
KNN	0.27	0.03	0.531
Random Forest Classifier	1.00	0.01	0.757
XGBoost	0.24	0.29	0.686
SVM	0.15	0.64	0.502

Model Training and Performance

4. Adjusting class weights in model training and standard scaler

Model	Precision	Recall	AUC Score
Logistic Regression	0.13	0.91	0.796
KNN	0.27	0.03	0.531
Random Forest Classifier	1.00	0.01	0.757
XGBoost	0.24	0.29	0.686
SVM	0.15	0.64	0.502

Hyperparameter Tuning

Precision	Recall	AUC Score
0.13	0.92	0.804

- Recall and AUC score improved slightly after tuning
- Recall and AUC score is highest
- Best model out of all

Selection of Best Model

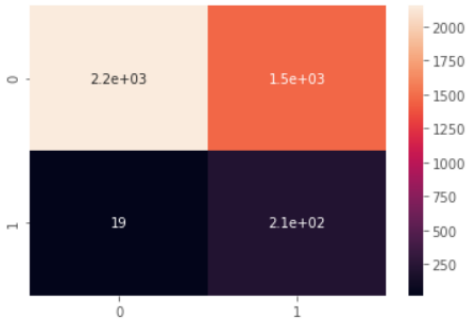
Best Model : Logistic Regression Model with class weights adjusted and standard scaler used

Classification Report

```
              precision    recall  f1-score   support

           0       0.99      0.59      0.74      3624
           1       0.13      0.92      0.22       231

    accuracy                           0.61      3855
   macro avg       0.56      0.76      0.48      3855
weighted avg       0.94      0.61      0.71      3855
```

Confusion Matrix

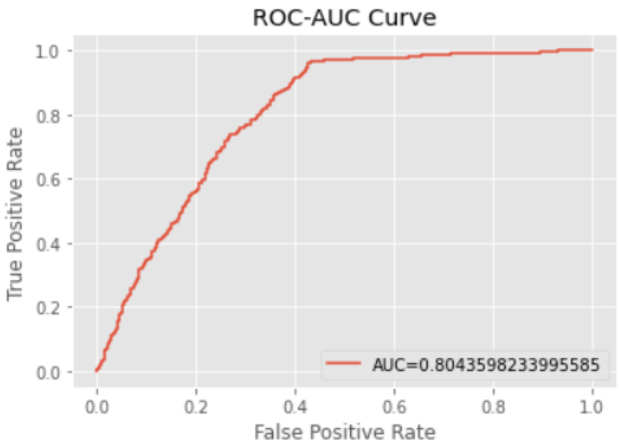
```
[[2151 1473]
 [  19  212]]

<AxesSubplot:>
```

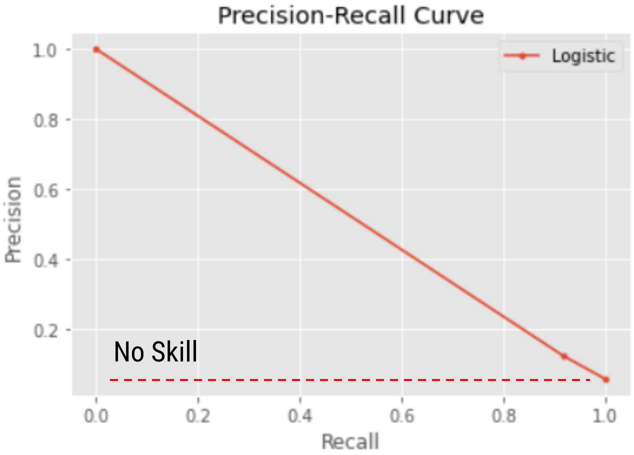


- 212 out of 231 (92%) fraud cases are caught by model
- 19 (8%) fraud cases undetected
- 1473 (38%) flagged transactions but not fraud cases (false positive)

Selection of Best Model

Best Model : Logistic Regression Model with class weights adjusted and standard scaler used





Changing Probability Threshold

- **AIM :** Balance the costs incurred by the Type I and Type II errors

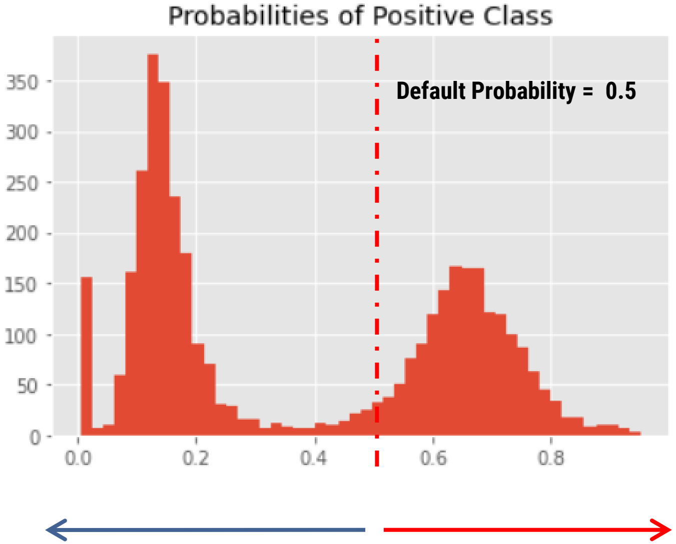
Type I Error	**False Positives** (*Flagged out as a fraud but not a fraud case)*	Precision	Incur cost of human labour for investigating the flagged transactions	Low precision score
Type II Error	**False Negatives** (*Fraud cases undetected by model)*	Recall	Incur cost of fraud	Low recall score

- To lower cost of Type I error → increase probability threshold
- To lower cost of Type II error → decrease probability threshold

Changing Probability Threshold

Best Model : Logistic Regression Model with class weights adjusted and standard scaler used

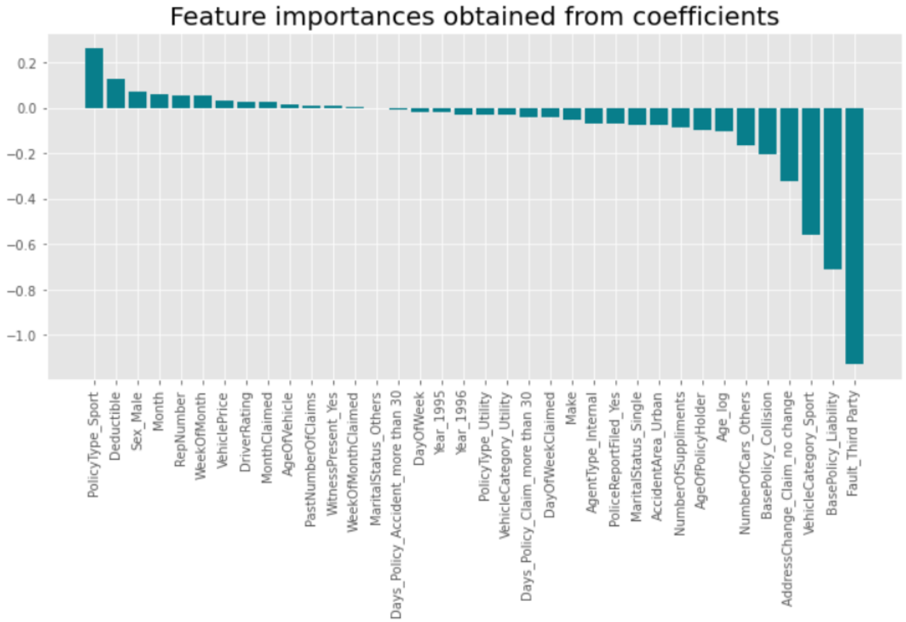
- 8% of the fraud cases has probability < 0.5
- If probability threshold is shifted to the left, false positive cases will increase, false negative cases will decrease
- Feasible if cost of fraud > cost of investigation and monitoring



- 92% of the fraud cases has probability > 0.5
- If probability threshold is shifted towards the right , false positive cases will decrease, false negative cases will increase
- Feasible if cost of investigation and monitoring > cost of fraud

Feature Importance

Best Model : Logistic Regression Model with class weights adjusted and standard scaler used



7

Conclusion and Way Ahead

Conclusion

- It is essential that model catches most of the fraud cases (Recall), while keeping the cost to monitor and investigate fraud cases flagged out by the model under control(Precision)

- Need to incorporate actual business costs incurred by stakeholders before deciding if the model is suitable to be deployed

- As such, we will have to look at the cost incurred from Type I and Type II errors in order to determine the optimal probability threshold to balance precision and recall of the model

Way Ahead

To incorporate business impact into the model :

1. Adjusting class weights during model training using the true cost ratio

   - Find out the labour cost needed for investigation (e.g. cost of monitoring per fraud case) and the cost of uncaught fraud (e.g. the average cost of an undetected fraud case)

   - Calculate the cost ratio and adjust the class weights accordingly (customised class weights) in the model training process.

2. Adjusting the probability threshold to balance the cost of type I errors and type II errors accordingly.

3. Optimising for F1-score , which balances precision and recall of the model.

Way Ahead

How can this model be improved further :

1. Include more relevant features in the machine learning process

   - From the feature importance plot, the weights of the features that lead to the prediction of positive class are not as high.

2. Increase the size of data to get a bigger portion of insurance claims that are fraudulent

   - Oversampling techniques and adjusting class weights might still not be the ideal solution in some imbalanced dataset. Although it is hard to get a larger proportion of fraud cases, it can help significantly in model training if there are enough positive labels.

The End