CPSC-6300

Insurance Fraud Detection

Checkpoint 1

Dineshchandar Ravichandran
dravich@g.clemson.edu
Archana Lalji Saroj
asaroj@g.clemson.edu
Prashanth Reddy Kadire
pkadire@g.clemson.edu

Contents

Introduction & Problem statement

The insurance industry consists of over 7,000 companies that collect over $1 trillion in premiums yearly. The massive size of the industry contributes significantly to the cost of insurance fraud by providing more opportunities and bigger incentives for committing illegal activities. The total cost of insurance fraud (non-health insurance) is estimated to be more than $40 billion annually. That means Insurance Fraud costs the average U.S. family between $400 and $700 per year in the form of increased premiums.[1] Insurance fraud also steals at least $308.6 billion yearly from American consumers.[2]

For this project, we will focus on Automobile-insurance fraud where 25%-33% of insurance claims have an element of fraud.[3] Automobile claim fraud and buildup added $5.6 billion-$7.7 billion in excess payments to auto-injury claims paid in the U.S. in 2012. 21 % of bodily injury (B.I.) claims and 18 % of personal injury protection (PIP) claims closed with payment had the appearance of fraud and/or buildup. Buildup involves inflating otherwise legitimate claims.[4]

The government and other organizations are responding to this by investing in technology to detect fraudulent claims, 21% of insurance institutes plan to invest in A.I. (Artificial Intelligence) in the next two years.[5]

Motivation & Goals

We will construct a supervised machine learning model-based Fraud Detection system to predict the chances of a fraudulent insurance claim. This system aims to analyze the insurance claim data set containing 38 distinctive features and generate a classification logic to identify genuine and fraudulent claims. By constructing such a fraud detection system, we can alert the insurance institutes and allow them to take necessary action against fraudulent claims.

Data Exploration

Data Summary

We collected the insurance claims dataset from Kaggle "insurance_claims_data."
This dataset comprises 1000 total data points. The dataset is imbalanced since there are a total of 247 fraud claims and 753 genuine claims, according to the preliminary data exploration.

[1] Federal Bureau of Investigation insurance reports and publication <https://www.fbi.gov/stats-services/publications/insurance-fraud>
[2] Coalition Against Insurance Fraud is working to update this figure in 2022.
[3] http://www.insurancejunction.co.za/insurance-fraud/
[4] https://www.michigan.gov/difs/consumers/fraud/insurance-fraud-statistics
[5] https://insurancefraud.org/fraud-stats/

What is the Unit of Analysis?

For this project, the accuracy, Precision, and F1 Score will be calculated on each model. These measures are considered as our unit of analysis for selecting the best model.

- **F1 score**: this is the harmonic mean of precision and recall and gives a better measure of the incorrectly classified cases than the accuracy matrix.

$$\text{F1-score} = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

- **Precision:** It is implied as the measure of the correctly identified positive cases from all the predicted positive cases. Thus, it is useful when the costs of False Positives are high.

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})} :$$

- **Accuracy:** One of the more obvious metrics is the measure of all the correctly identified cases. It is most used when all the classes are equally important.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{(\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative})}$$

How many observations in total are in the dataset?

We have a data set containing 1000 insurance claims and 38 features pertaining to it, in addition we also have a column stating which of these insurance claims are fraudulent and which are genuine as illustrated below:

```
df_insurance.shape
✓ 0.5s
(1000, 39)
```

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	witnesses
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132	...	2
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176	...	0
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632	...	3
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117	...	2
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706	...	1

police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year	fraud_reported
YES	71610	6510	13020	52080	Saab	92x	2004	Y
?	5070	780	780	3510	Mercedes	E400	2007	Y
NO	34650	7700	3850	23100	Dodge	RAM	2007	N
NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014	Y
NO	6500	1300	650	4550	Accura	RSX	2009	N

How many unique observations are in the dataset?

In our dataset, every claim is unique, though we do not have a unique identifier like claim I.D. the policy number for all the claims are unique.

What time period is covered?

Our dataset contains claims with incidents occurring from 1st Jan 2015 to 1st March 2015.

Data cleaning:

Three columns had '?' Missing value:

1. Collision Type
2. Property Damage
3. Police Report Available

The missing values are imputed using KNN imputer. The categorical features have been transformed into numerical features using one hot encoding and label encoding.

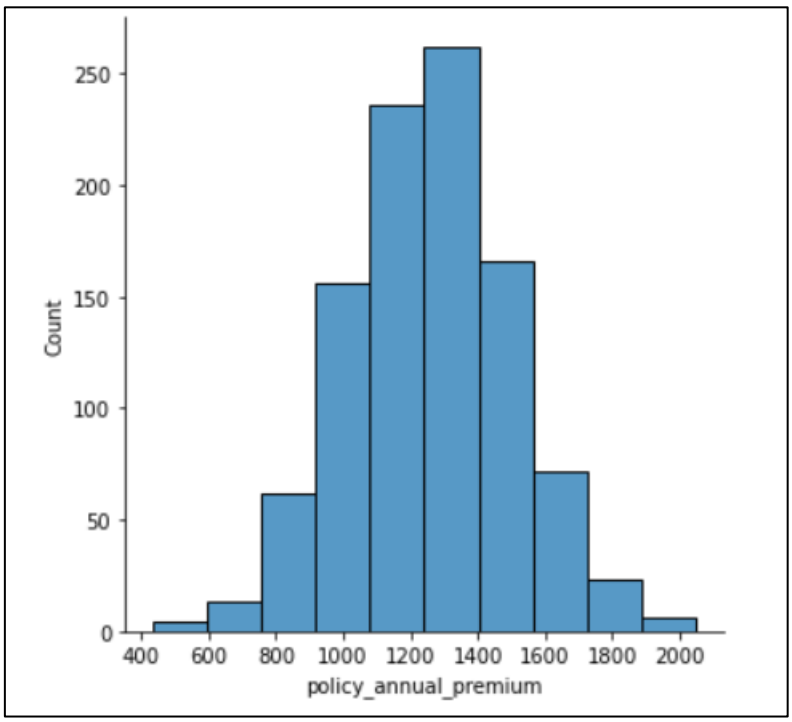
Description of outcome with an appropriate visualization technique

Description of the dataset:

Column Name	Data Type	Description
months_as_customer	int64	No.of months customer has been a customer to the insurance organization.
age	int64	Age of the customer.
policy_number	int64	Policy no.of the customer's insurance.
policy_bind_date	object	The moment when the insurance coverage goes into force, it's date and time specific.
policy_state	object	State in which the policy was procured.
policy_csl	object	Combined single limits are a provision of an insurance policy limiting coverage for all components of a claim to a single dollar amount. A combined single limit policy has a maximum dollar amount that covers any combination of injuries or property damage in an incident.
policy_deductable	int64	The amount customers have to pay for covered services before their insurance plan starts to pay.
policy_annual_premium	float64	Annual payment for the policy.
umbrella_limit	int64	Extra insurance that provides protection beyond existing limits and coverages of other policies. Umbrella insurance can cover injuries, property damage, certain lawsuits, and personal liability situations.
insured_zip	int64	
insured_sex	object	Male/ Female.
insured_education_level	object	Education level of the customer.
insured_occupation	object	Occupation of the customer.
insured_hobbies	object	Customers' hobbies.
insured_relationship	object	Relationship of the person involved in the incedent to the actual insurance holder.
capital-gains	int64	Increase in a capital asset's value.
capital-loss	int64	Loss in a capital asset's value.
incident_date	object	Date of incident.
incident_type	object	Type of incident (Single Vehicle Collision, Vehicle Theft, Multi-vehicle Collision, and Parked Car).
collision_type	object	Type of collision (Side Collision, Rear Collision, and Front Collision).
incident_severity	object	The severity of the incident classified as per damage (Major Damage, Minor Damage, Total Loss, and Trivial Damage).
authorities_contacted	object	Type of authorities contacted after the incident.
incident_state	object	U.S. state where the incident occurred.
incident_city	object	City in which the incident occurred.
incident_location	object	Address of the incident.
incident_hour_of_the_day	int64	Time of incendent in 24hrs.
number_of_vehicles_involved	int64	No.of vehicles involved with the incident.
property_damage	object	If 'YES,' then the insurance carrier helps pay to repair the damage customer caused to other involved parties.
bodily_injuries	int64	No.of bodily injuries.

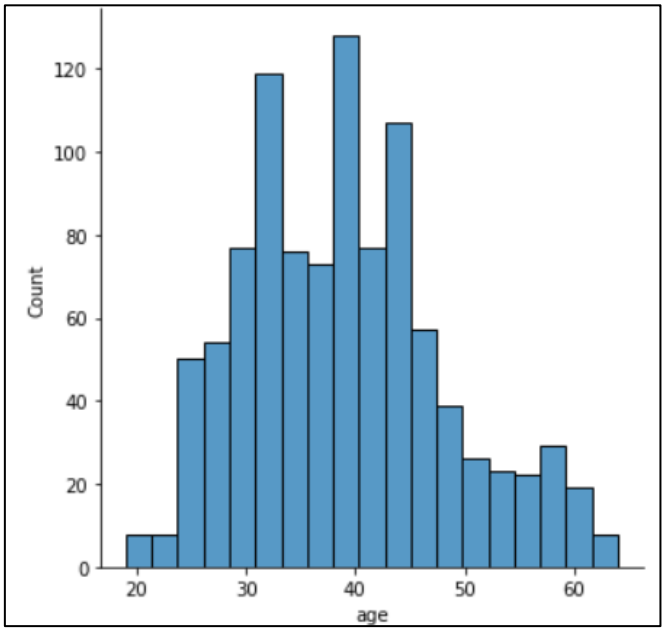
witnesses	int64	No.of witnesses to the incident.
police_report_available	object	If a police report exists of the incident.
total_claim_amount	int64	The total amount claimed by the customer fot the incedent.
injury_claim	int64	The portion of the total claim requested for the injury claim.
property_claim	int64	The portion of the total claim requested to pay for property damages.
vehicle_claim	int64	The portion of the total claim requested for vehicle damage.
auto_make	object	Manufacturer of the customer's vehicle that was involved in the incident.
auto_model	object	The specific automobile model of the customer's that was involved in the incident.
auto_year	int64	Model year
fraud_reported	object	Determining whether a claim is fraudulent or not.

Annual policy premium distribution



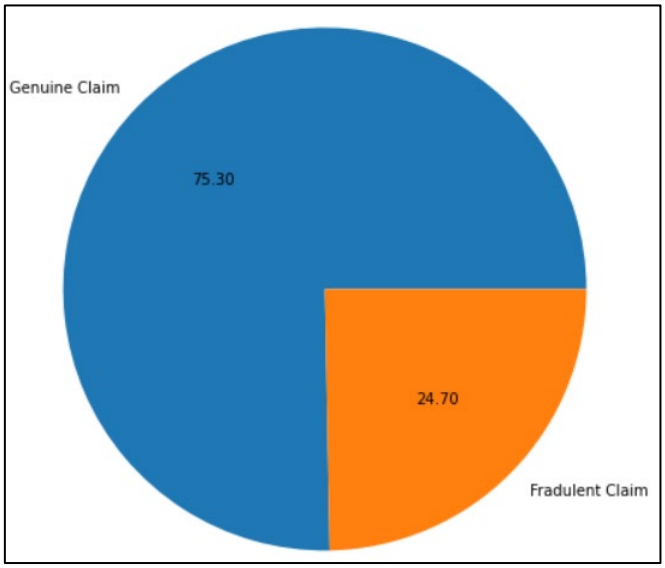
The above graph gives us the distribution of the annual premium paid by the customers for their insurance, where the bulk of the population pays between 1000-1500.

Age distribution of policyholders



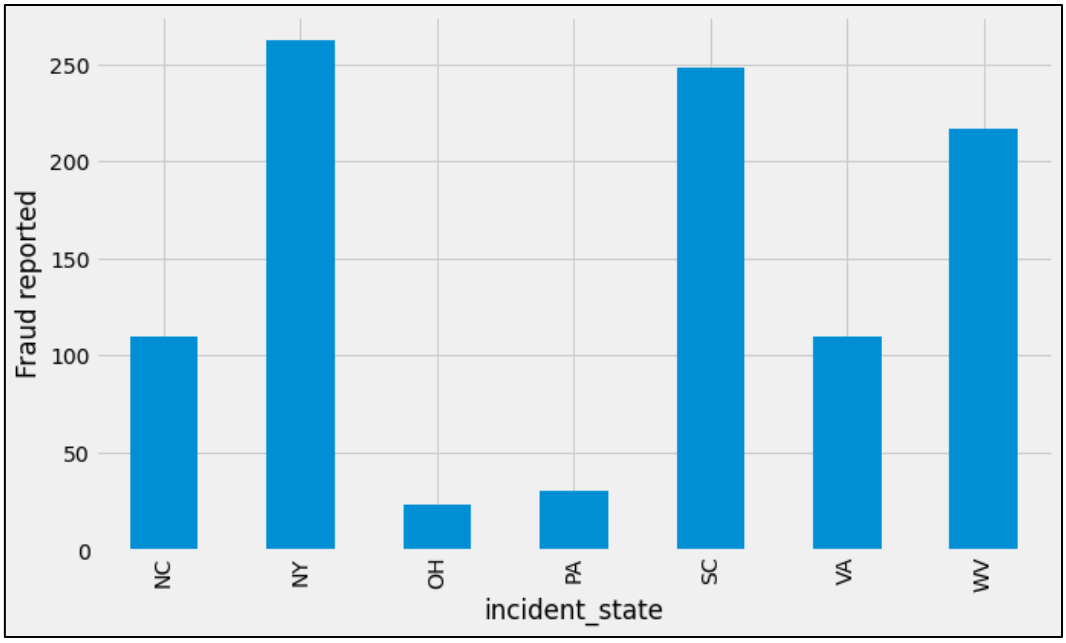
The above graph gives us the distribution of the age of the customers.

Genuine claims VS Fraudulent claims comparisons



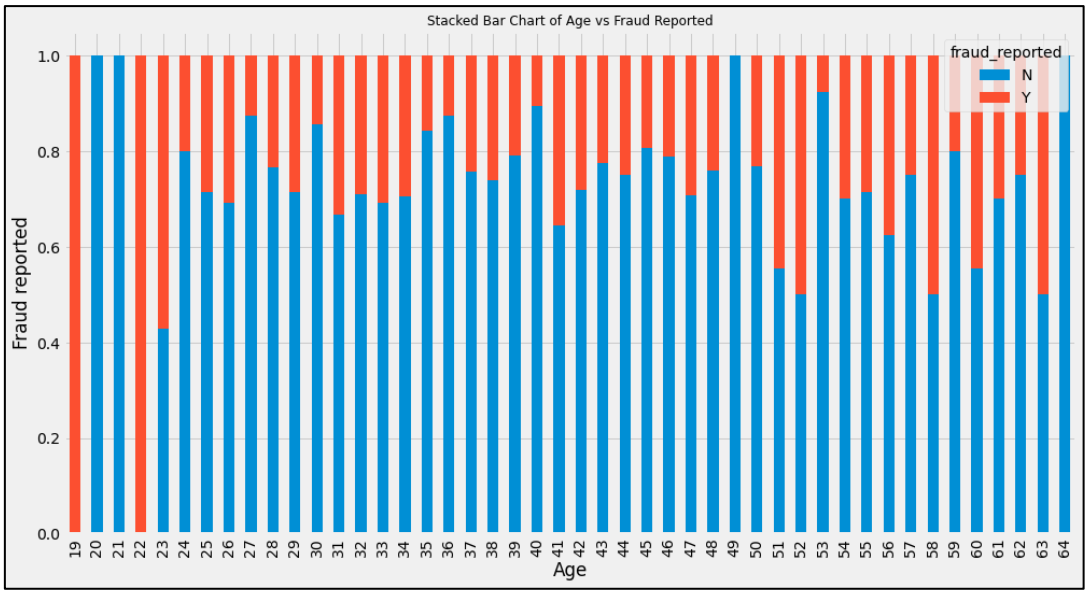
As illustrated in the above graph, only 24.70% of the total claims are fraudulent.

Incident state-wise fraudulent claims



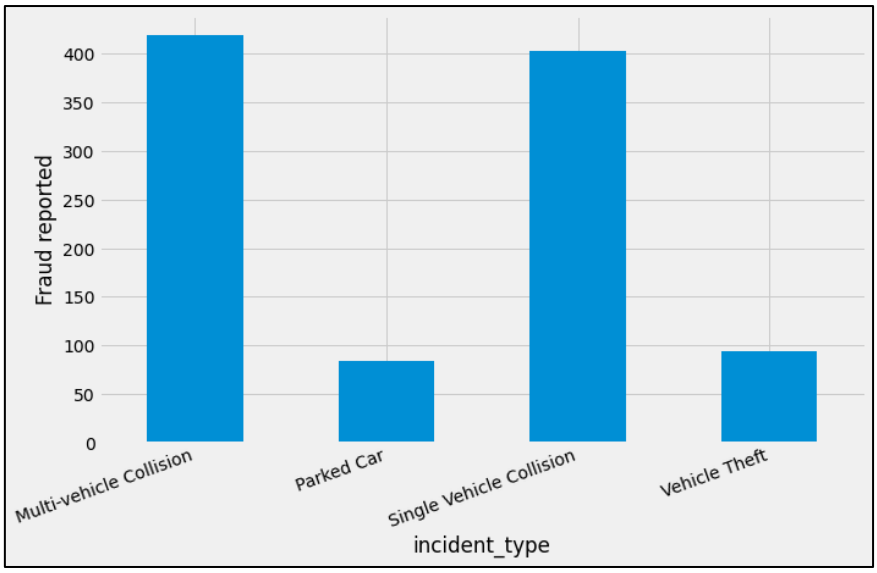
As illustrated above, most fraud-reported incidents occurred in N.Y. and S.C.

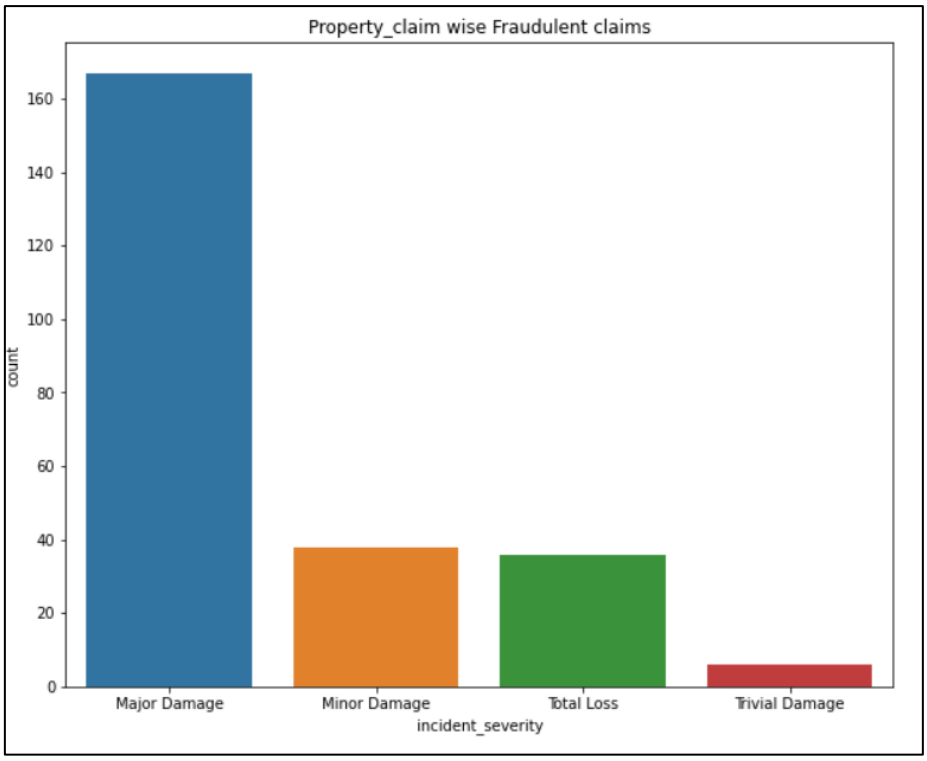
Age-wise fraudulent claims



As illustrated above, most fraud claims are committed by customers aged between 19-23, and hence is an essential feature of our detection model.

Incident type and damage relation to fraudulent claims





Based on the above graphs, we can determine that most of the fraudulent claims tend to be collisions with reported major damage.

Project Approach:

Based on the above analysis, we will solve this binary classification problem using different Machine Learning algorithms such as Regression, SVM, Random Forest, Adaboost classifier, and XGboost classifier. According to the performance metrics, we will choose the best classifier.

To start with we will be selecting the following models:

Logistic Regression from sklearn.linear_model:

Logistic Regression is a supervised learning classification algorithm used to predict the probability of a target variable. The target or dependent variable's nature is binary, meaning there would be only two possible classes: 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

RandomForestClassifier from sklearn.ensemble:

Random forest algorithm builds decision trees on data samples, obtains predictions from each one, and then uses voting to determine the best option. Because it averages the outcomes, the ensemble method is superior to a single decision tree in that it lessens over-fitting.

XGBClassifier from XGBoost:

Extreme Gradient Boosting is abbreviated as XGBoost. The "eXtreme" part of XGBoost's name refers to speed-improving features such as parallel processing and cache awareness that make it around ten times faster than conventional gradient boosting. A special split-finding method and integrated regularization that lessens over-fitting are also included in XGBoost. XGBoost is a quicker, more accurate version of gradient boosting.