An investigation into the Mendeley Insurance Fraud database

R

November 9, 2025

Contents

1 The Shape of the Data

It is important to understand the shape and observed distribution of the data. Checking the dataframe, there are only 247 counts of fraud cases, out of the 1000 on record, which we will need to be mindful of during the test-train split. We can observe the distribution of fraud claims across differently sized insurance claims to identify whether there are any clear trends or internal variations in fraud across claims.

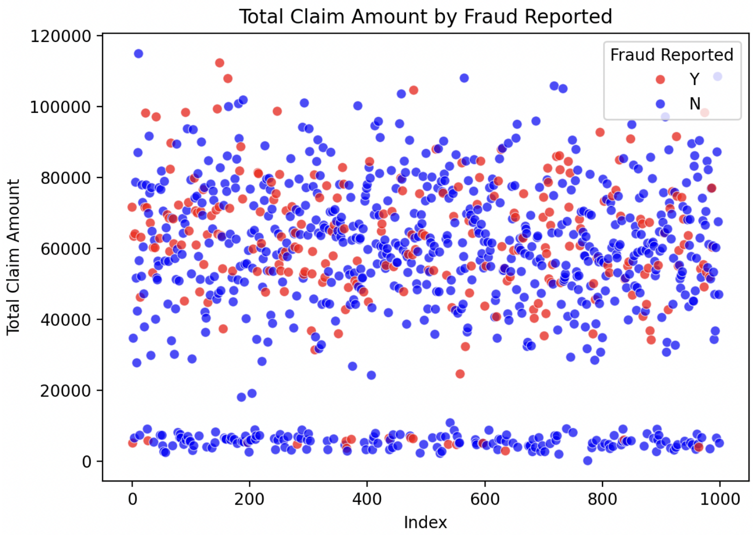

Figure 1: Fraud coded claim amounts

From Figure 1, we can see that fraud cases across claim amounts are largely random, making it unlikely that the data generating processes for fraud violate the zero-conditional-mean assumption of generalised linear models (GLM), at least across total claim amounts.

We know that the data has 1000 cases across 40 variables. We can also check the geographical location of the data. The three states listed in the dataframe are Illinois, Ohio, and Indiana, meaning that the data is largely collected from the American Midwest. We can also observe that all these incidents took place between 01/01/2015 and 01/03/2015. The mean age of claimants was 39, with the youngest being 19 and the oldest being 64.

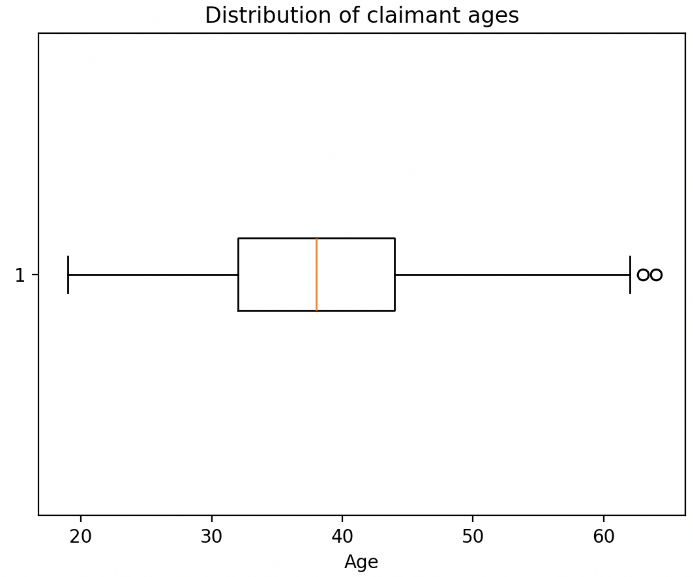

Figure 2: Boxplot of ages in dataset

We can see that generally, the majority of claimants are somewhere below 45, with a few outliers above (Figure 2). For further investigation, it could be useful to compare the distribution of claimants across the companies to the distribution of policyholders. We can see that of the 1000 claimants, 537 were female, assuming that the underlying insured population is split exactly down the middle, this result (or one more extreme) would only naturally occur in less than 1.05% of samples. This is suspect, but not evidence of anything without knowing the gender split of the insured population.

2 Initial Data Exploration

The first step was to identify all missing data and catalogue it efficiently. Of the columns in the dataset, only four had missing data, and most were closely related to police involvement in the claims. Unfortunately, this would likely be non-randomly related to fraudulent insurance claims, so these four columns were ignored for the rest of the investigation.

There are clearly identifiable clusters in the sizes of claims (Figure 3). To study this further, I made a histogram comparing claims by their incident types (Figure 4).

The graphic (Figure 4) showed that the claim clusters are actually closely related to the kinds of claims being made, with vehicle theft and parked car accidents generating smaller claims, and vehicle collisions generating a far higher distribution of claims prices. We therefore need to be mindful that the claims distributions visibly follow at least two clusters. Observing vehicle collisions further, they both appear to be quite similar and somewhat normally distributed. To test this, I initially charted their experimental cumulative distribution functions:

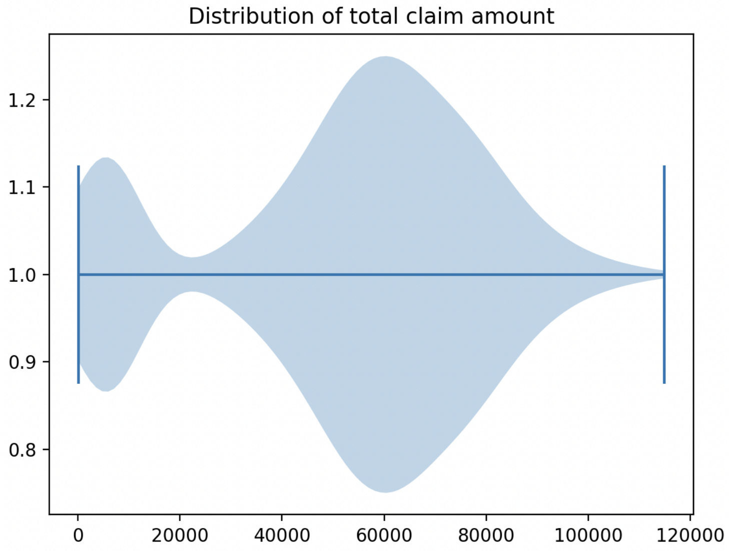

Figure 3: Violin Plot of total claim amounts ($)

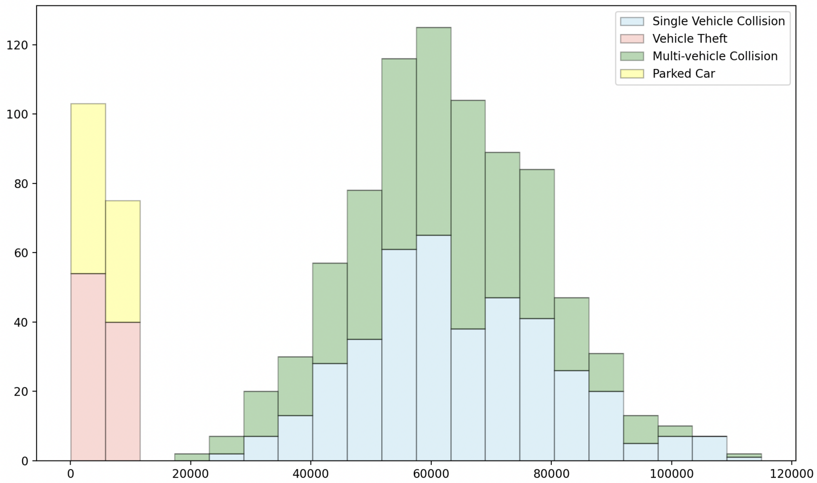

Figure 4: Total Claim Amounts for Incident Types

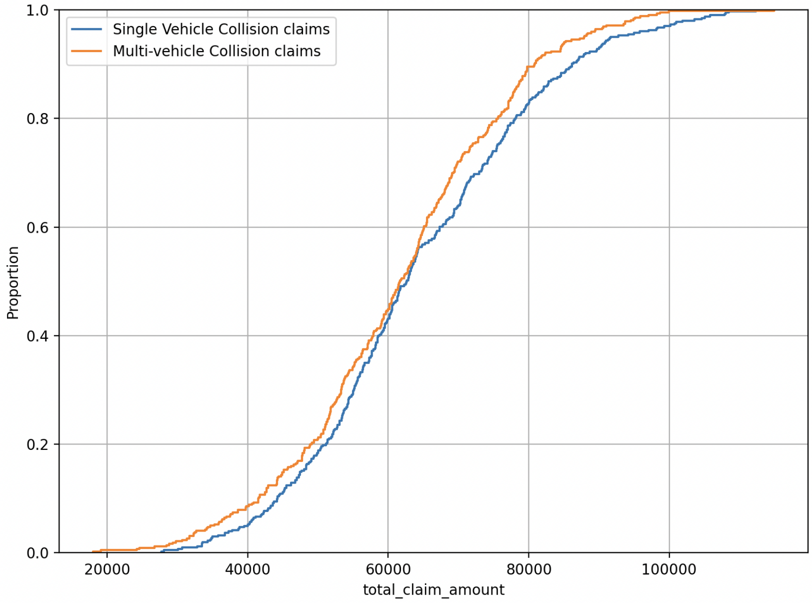

Figure 5: CDFs of Vehicle Collision Claims Sizes ($)

These visually look extremely similar (Figure 5), and, after running a Kolmogorov-Smirnov test, I found no statistically significant evidence to reject that they followed different sampling distributions. Running a Shapiro-Wilk test, I found evidence at the 95% CI that single vehicle accident claims were normally distributed, but the same did not hold true for multi-vehicle accident claims. This could be a result of minor sampling biases in multi-vehicle collision claims, and is a reason to question how the data was collected. Further, an F-test found no statistical evidence of the two having different variances, implying that both collision claim sampling distributions are highly similar.
This cluster of claims have more visually dissimilar ECDFs (Figure 6), but no statistically significant evidence was found that they follow different sampling distributions. However, the Shapiro-Wilk test also could not reject that both distributions were not normal, casting mild doubt on the methods used to collect claims data.

We can also assess how fraud cases varied across different policy durations.
It is again difficult to spot any sort of trend of clustering of fraud cases across policy days (Figure 7). This implies that Policy Days would be a reasonable variable in a regression, since fraud cases seem randomly distributed across it.

These even principles do not seem to apply in the case of incident severity however, where there is a clear overrepresentation of fraud cases in insurance claims for major damage to vehicles (Figure 8). It is unclear whether this is due to better investigation into claims with major damage, or a different reason like fraudulently claiming major damage seeming more lucrative to scammers.

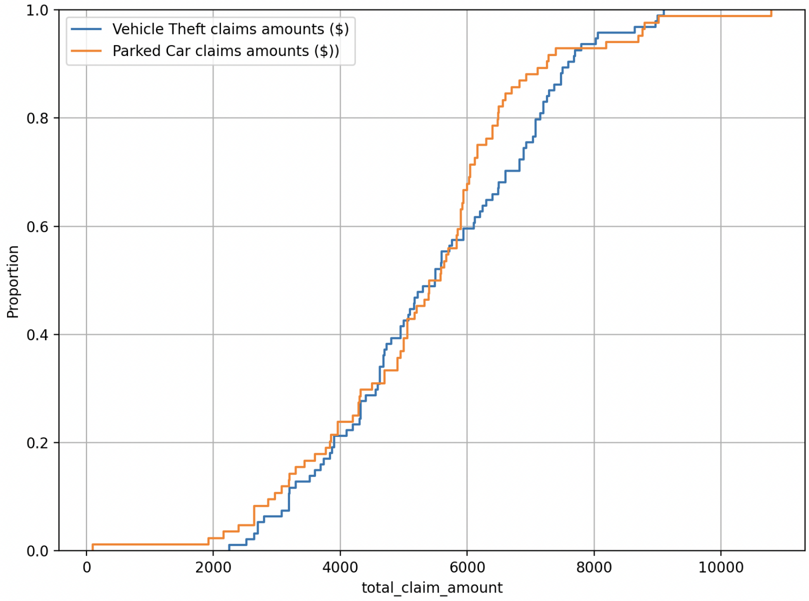

Figure 6: Experimental CDFs for Vehicle Theft claims and Parked Car claims

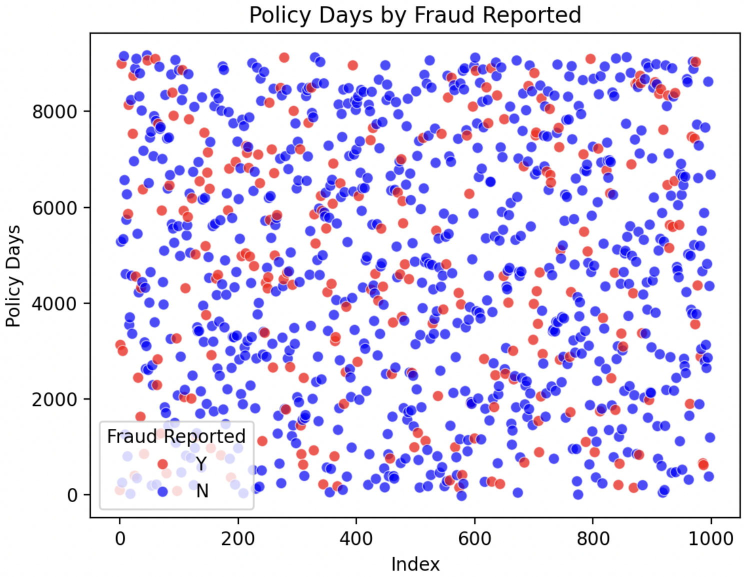

Figure 7: Duration of policy coded by fraud cases

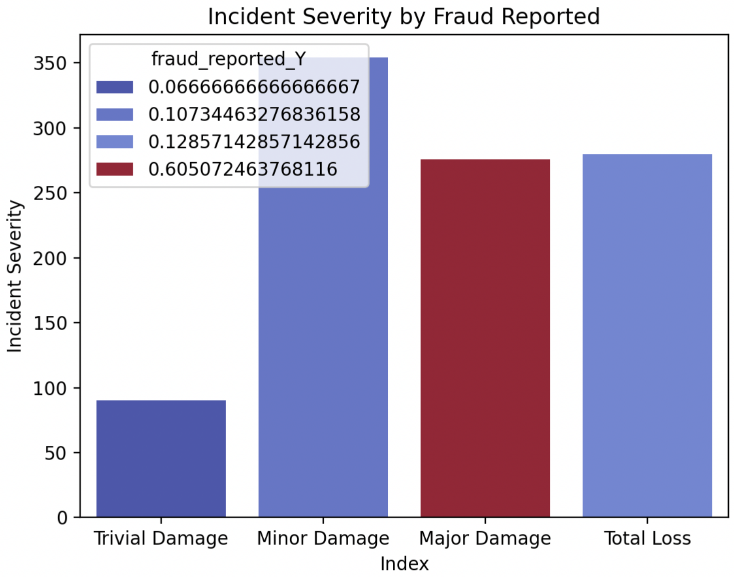

Figure 8: Incident severity of claims coded by fraud cases

It is probable that incident severity will be a significant variable in any regression of fraud cases, but care must be taken in data collection to ensure that this is not a sampling bias.

We can observe a similar situation in the incident types.
Much like before, while parked car and vehicle thefts seem to follow their own, lower, distributions, collisions seem to have a higher rate of fraud cases (Figure 9). It is important to ensure that the method used to collect the data and inspect cases for fraud does not encourage overrepresentation of fraud cases in the collisions dataset.

3 GLM regression (Logit)

Pandas provides dummy generating functionality, allowing us to collect the set of dummy variables we need. For risk-prediction models using GLM, it can be more effective to collect data using the weight of evidence method, which states:

$$\text{WoE} = \ln \frac{\text{Proportion of Good Outcomes}}{\text{Proportion of Bad Outcomes}}$$

Where Goods are an outcome that we want, and bads are ones that we do not. Running a python command to collect these and their associated information values, we can assess that the best regressors by WoE. This yields:

- incident date,
- incident type,

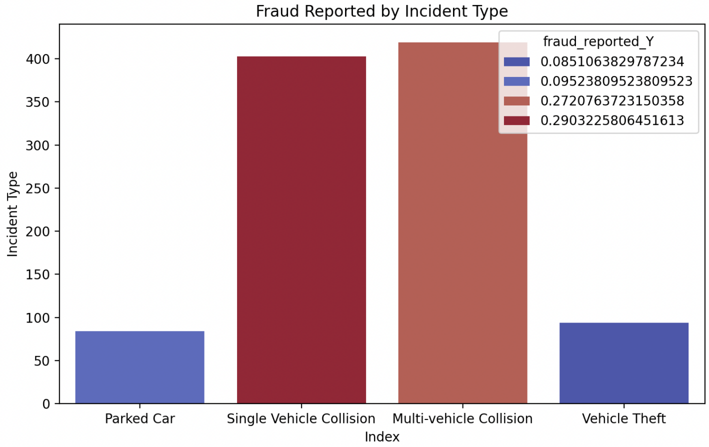

Figure 9: Incident Type coded by fraud

- collision type,
- whether authorities were contacted,
- and the total claim amount,

Due to the limited timeframe of the data and the nature of a linear regression model, directly regressing the incident date would likely inappropriately make a linear time trend of increasing fraudulence when the trend could very easily be cyclical. Consequently, policy days will be used instead. Alongside this, authority contact and the collision type both have high instances of missing data, and would definitely not be independently distributed across fraudulent and real insurance claims. Consequently, the list of variables used in the logit model were:

- the incident type,
- the incident severity,
- the total claim amount,
- the days the policy was held over.

Which should be closely related to the missing variables.

Running the regression using scikit, I found:
With the omitted dummies of Incident Type (IT) Multiple-Vehicle Collision and Incident Severity(IS) Major Damage.
From this table, we can see that Multiple-Vehicle and Single-Vehicle Collisions both massively increase the log-odds of a claim being fraudulent, with multiple-vehicle collisions being almost 20% more likely to be fraudulent than vehicle theft cases. Major damage cases also seem to have a far higher probability of fraudulence, with any random major damage case being over 90% more likely to be fraudulent than a random minor damage case. Longer policy duration also seemed to predict a higher

Table 1: sklearn logit regression

Regressor	Coefficient	Log Odds Ratio
IT Single Vehicle Collision	0.147885	1.159379
IT Vehicle Theft	-0.233381	0.791852
IS Minor Damage	-2.304886	0.099770
IS Total Loss	-2.174822	0.113628
IS Trivial Damage	-2.245606	0.105863
Total Claim Amount	0.000003	1.000003
policyDays	-0.000012	1.894143

probability of a claim being fraudulent, while the total claim amount seemed to have very little impact on log odds of a claim being fraudulent.

If we want to see the p-values of the logit regression, we can use statsmodels logit package.

Table 2: statsmodel logit regression

Regressor	Coefficient	p-value
IT Single Vehicle Collision	0.1558	0.388
IT Vehicle Theft	0.0671	0.888
IS Minor Damage	-2.5292	0.000***
IS Total Loss	-2.3444	0.000***
IS Trivial Damage	-2.9672	0.000***
Total Claim Amount	0.000001	0.835
policyDays	-0.000012	0.709

*** significant at 99.9% CI
From this, it's clear that the incident severity was the only statistically significant set of variables, with major damage being the most extreme predictor of fraudulent claims.

We can compare the results from these logit regressions to a gradient boosting model. Probability trees can often identify non-linear relations that escape models like the logit or other GLM methods. However, their ability to identify relationships non-deterministically means that using a model selected by the weight of evidence information criteria is preloaded to find relationships, and this can encourage spurrious inferences. Consequently, I instead used a set of variables I reasoned should predict fraud well, prior to collecting the WoE statistics. For our regressors, we use:

- Incident severity,
- total claim amount,
- policy days,
- highest education level,
- claimant sex.

Collecting this, I ran a gradient boosting model, with MSE 0.168 and $R^2 = 0.053$. These are not very good, and imply the model is very poor at predicting fraud. XGBoost attempts to be a more interpretable machine learning interface, and so we can see the weight and gain of the model:

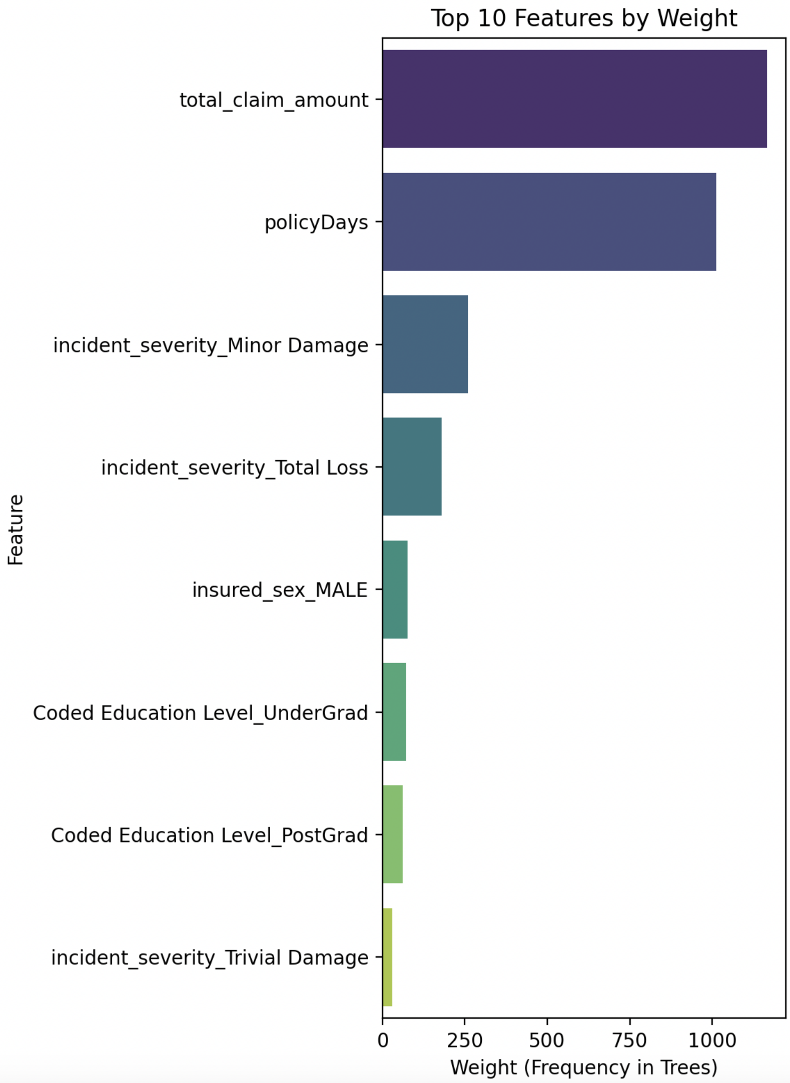

Figure 10: Variable weights in tree model

The ’weight’ of a variable in XG is how often it appears in the probability trees, one can interpret a higher weight as the model calling a variable more often when attempting to predict the regressand. Interestingly, incident severity did not have the highest weight, with the total claim amount and policy days having a far higher weight (Figure 10).
Gain is a measure of how much accuracy the model obtains when it includes a variable. The higher the gain, the more useful a variable is to the model’s calculations.

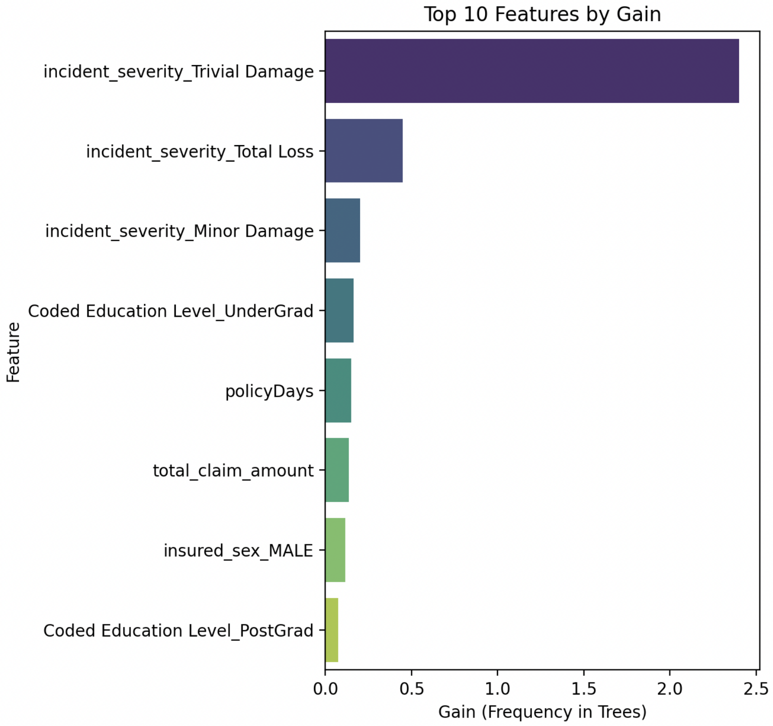

Figure 11: Variables by gain in the tree model

Unsurprisingly, incident severity had the highest gain (Figure 10). Interestingly, it had by far the most gain, with the other incident severity dummies having far smaller gain. This could signify that the lower the damage, the less likely a claim is to be fraudulent.

The logit regressions had a low pseudo R^2 of around 0.22, and while the ordinary least squares concept of R^2 does not carry over exactly into gradient boosting or logit regressions, the fact that the XGBoost model had an R^2 of 0.05 strongly implies that the logit regression was much better at predicting fraudulent claims. Overall, the regressions all agreed that incident severity was the highest predictor of fraudulent claims, with any random major damage case being far more likely to be fraudulent than any minor, trivial, or complete loss case.

4 Dataset

The dataset was obtained from Mendeley (Mukherjee, Anklan, et al., 2022).

5 References

Mukherjee, Anklan, et al. (2022). *Car Insurance Claim Severity Dataset*, Mendeley Data.