DID IT HAPPEN?

Insurance Fraud Detection

with Machine Learning

Neil Bhikha | Institute of Data | 2022

TABLE OF CONTENTS

Insurance Explained

WHAT IS INSURANCE?

insurance is a contract, represented by a policy, in which a policyholder receives financial protection or reimbursement against losses from an insurance company. The company pools clients' risks to make payments more affordable for the insured. [1]

Insurance policies are used to hedge against the risk of financial losses, both big and small, that may result from damage to the insured or their property, or from liability for damage or injury caused to a third party. [1]

HOW DOES INSURNACE WORK?

The insurance company takes the risk of the insured person and for such "risk management," cost is required for an insurance company.

The cost of risk is spread across a large group of people who share the similar risks.

When you buy an insurance policy, you transfer your risk to the insurance company. In exchange for this risk, you pay a premium to the company.

This, in turn, will create a big pool of funding, which will be used to cover the claim in case of occurrence of any unexpected event in anyone's life.

After the claim acceptance, the insurer will assist to settle down the claim after the concerned process and verification. [2]

AUTO INSURANCE

When you buy or lease a car, it’s important to protect that investment. Getting auto insurance can offer reassurance in case you’re involved in an accident or the vehicle is stolen, vandalized, or damaged by a natural disaster. Instead of paying out of pocket for auto accidents, people pay annual premiums to an auto insurance company; the company then pays all or most of the costs associated with an auto accident or other vehicle damage. [1]

Insurance Fraud

WHAT IS INSURANCE FRAUD?

Insurance fraud is when someone does something dishonestly (or doesn’t do something they should do under their policy) to try and illegally benefit in a way that they’re not entitled to. Any act or omission made with dishonest or illegal intent, to obtain a benefit or advantage, is considered fraudulent.

Most insurance fraud occurs at claim time, including:

- Exaggerated claims: Claiming for more than they require.

- Events/losses that didn’t happen: Making up situation that didn’t happen.

- Staged losses: for example vehicle theft, Crashing on purpose.

In addition, a large number of fraudulent claims occur through non-disclosure of information that insurers require, in order to match the correct premium to the risk. [3]

INSURANCE FRAUD IN NEW ZEALAND

In 2019 general insurance fraud in New Zealand was estimate to have cost policyholders over $688 million. That means that every

year, honest New Zealanders are paying to cover other people's fraudulent claims. [4]

The estimated cost of insurance fraud to all policyholders this year: 580,000,000 [still going up]. You can check out current total count here: https://ifb.org.nz/

HOW DOES IT AFFECT YOU?

The cost of insurance fraud isn't limited to claim values. Insurance fraud increases insurers' operating costs through additional time, staff and investigation. These costs also need to be passed on to consumers.

The larger the size of the dishonest claim, the more everyone ends up paying. [4]

BUSINESS CONTEXT

CURRENT STATE

Currently insurance claims are coming in at an increasing rate. Investigating each claim to see if it is legitimate is very time consuming. If the illegitimate claim slips to the vetting process that will lead to loss for the company and the customers. The more loss that occurs the more the company will have to charge its customers, which leads to a decrease in customer dissatisfaction.

PROPOSED SOLUTION

What if we can implement a model that can predict which claims are fraud?

IMPACT

This would increase efficiency for the claims process and would reduce the amount of additional time, staff and investigation. Also decrease the amount of capital lost. In turn this would benefit the

company’s customer retention and profits. Potentially help reduce premiums.

WHAT ARE WE INVESTIGATING?

1 Can we predict which claims will be reported as fraud?

2 What trends does the data show that leads to fraud? - Red Flags

3 Can we minimize the loss made from fraud?

INITIAL IMPLEMENTATION

This Machine Learning Model would come in when the claim is filed as long as we have all the required information we should be able to predict which claims are fraud.

What it could help with:

- Highlight claims that would need to be focused on
- Increase in claims processing efficiency
- Pick out trends that highlight fraud

THE DATA

WHERE WAS THE DATA GATHERED FROM?

Insurance data is hard to come by due to its sensitive nature.

The data was taken from Data Bricks: Insurance Claims - Fraud Detection

Website link to dataset: https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4954928053318020/1058911316420443/1677039324426 45/latest.html

ABOUT THE DATA SET

Name: Insurance Claims - Fraud Detection

Range Index: 1000 entries, 0 to 999

Data columns [total 40 columns]:

'months_as_customer', 'policy_csl', 'policy_deductable', 'policy_annual_premium', 'umbrella_limit', 'insured_sex', 'insured_education_level', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'capital-gains', 'capital-loss', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'incident_hour_of_the_day', 'number_of_vehicles_involved', 'property_damage', 'bodily_injuries', 'witnesses', 'police_report_available', 'total_claim_amount', 'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make', 'auto_model', 'auto_year', 'fraud_reported', 'loss_by_claims'

Table:

Index	Variable name	Data type	Type	Description
1	age	Integer	Ratio	Age In Years Of Insured Driver
2	authorities_contacted	Object	Nominal	Type Of Authorities Contacted Such As Fire Department Or Police
3	auto_make	Object	Nominal	Brand Of The Auto Vehicle
4	auto_year	Integer	Ratio	Year The Auto Vehicle Was Made
5	bodily_injuries	Integer	Ordinal	Were Bodily Injuries Present
6	capital-gains	Integer	Ratio	Data Not Defined By Original Owner And Non-Deducible From Analysis
7	capital-loss	Integer	Ratio	Data Not Defined By Original Owner And Non-Deducible From Analysis
8	collision_type	Object	Nominal	Type Of Collision
9	fraud_reported	Integer	Nominal	Weather The Claim Is A Fraud
10	iclaim_severity_int	Integer	Nominal	Interaction Term Of Injury Claims And Incident Severity
11	incident_city	Object	Nominal	City Of Which Incident Occurred
12	incident_hour_of_the_day	Integer	Ratio	Hour Of Which Incident Occurred
13	incident_month	Integer	Nominal	Month Of Which Incident Occurred
14	incident_severity	Integer	Ordinal	Severity Of Incident
15	incident_state	Object	Nominal	State Of Which Incident Occurred
16	incident_type	Object	Nominal	Type Of Incident
17	injury_claim	Integer	Ratio	Injury Claim Amount
18	insured_education_level	Object	Nominal	Level Of Education Of Insured Driver
19	insured_hobbies	Object	Nominal	Insured Driver's Hobbies
20	insured_occupation	Object	Nominal	Insured Driver's Occupation
21	insured_relationship	Object	Nominal	Insured Driver's Relationship Status
22	insured_sex	Integer	Nominal	Insured Driver's Sex
23	loss_by_claims	Float	Nominal	Difference Between Annual Premiums Paid And Total Claims
24	months_as_customer	Integer	Ratio	Months As A Customer To The Insurer
25	number_of_vehicles_involved	Integer	Ratio	Number Of Vehicles Involved In Incident
26	pclaim_severity_int	Integer	Nominal	Interaction Term Of Property Claims And Incident Severity
27	police_report_available	Object	Nominal	Whether A Police Report Is Available
28	policy_annual_premium	Float	Ratio	Annual Premium Of The Policy
29	policy_bind_year	Integer	Ratio	Binding Year Of The Policy
30	policy_csl	Object	Nominal	Policy Combined Single Limits Scheme
31	policy_deductable	Integer	Ordinal	Policy Deductable
32	policy_state	Object	Nominal	State Of Which Policy Was Purchased
33	prem_claim_int	Float	Nominal	Interaction Between Policy Annual Premiums And Total Claims
34	property_claim	Integer	Ratio	Property Claims Amount
35	property_damage	Object	Nominal	Whether There Were Property Damage
36	tclaim_severity_int	Integer	Nominal	Interaction Between Total Claims And Incident Sevevrity
37	total_claim_amount	Integer	Ratio	Total Claim Amount
38	umbrella_limit	Integer	Ratio	Type Of Insurance Add On Rider To Protecct From Excess Liabilities
39	umlimit_tclaim_int	Integer	Nominal	Intereaction Between Umbrella Limit And Total Claim
40	vclaim_severity_int	Integer	Nominal	Interaction Between Vehicle Claims And Incident Sevevrity
41	vehicle_claim	Integer	Ratio	Vehicle Claim Amount
42	witnesses	Integer	Ordinal	Number Of Witnesses

Snip off Data Frame: Refer to Capstone Insurance Fraud EDA

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.910000	0
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.220000	5000000
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.140000	5000000
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.740000	6000000
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.910000	6000000

DATA CLEANING

Key findings:

- No null values were present
- Three columns had values as '?'
- _c39 had all values missing

Steps taken:

1. Dropped _c39
2. Converted '?' to NaN
3. Checked Unique Values for Policy Number – no duplicates present
4. Converted Date time for Policy Bind Date

Missing values:

Collision type – 178 values missing – 17.8%

Property Damage – 360 values missing – 30.0%

Police report available – 343 values missing – 34.3%

Decided to impute them with the word "Not documented" instead of dropping them as they may be important.

Missing information from these variables may be a predictor to fraud.

Missing information on a collision type or on a police report seems suspicious.

Umbrella Limit

What is umbrella limit?

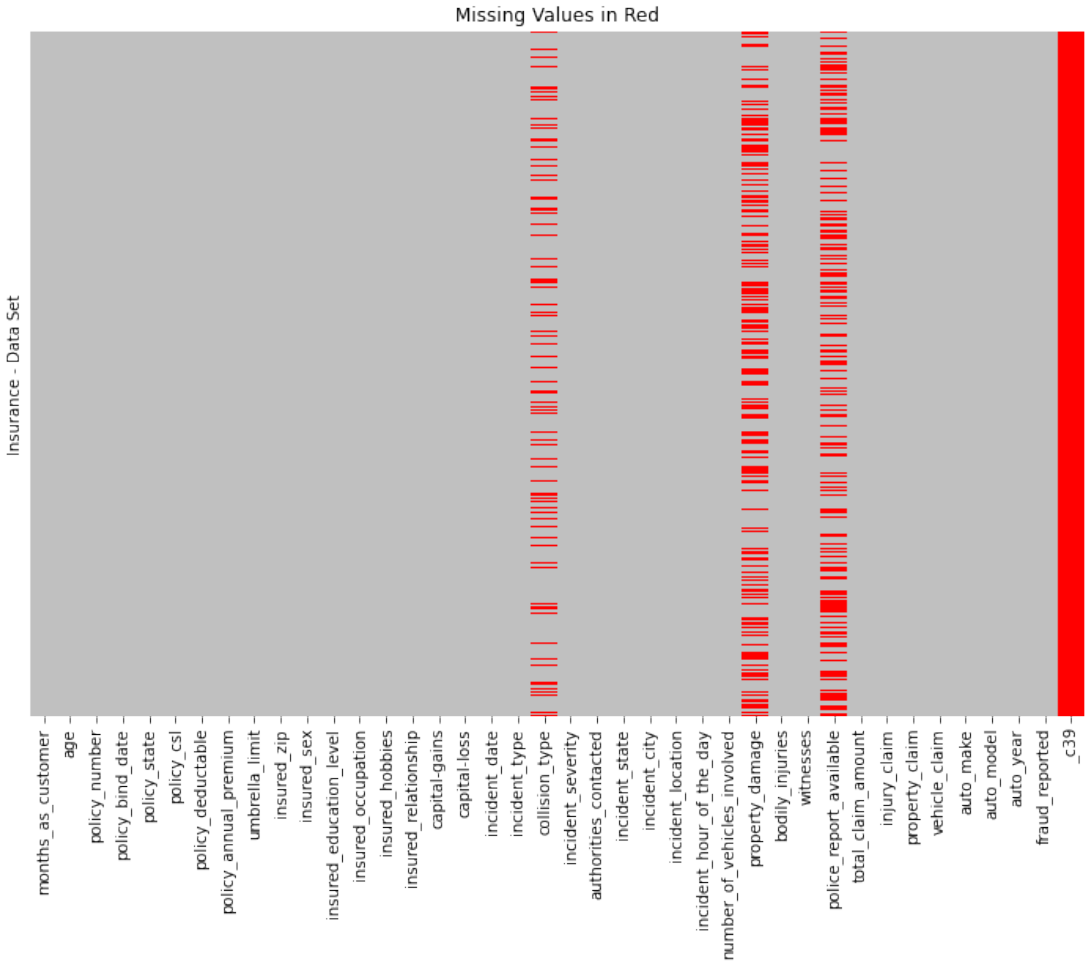
Umbrella insurance covers a wide range of problems and provides funds above and beyond the limits of your other policies, such as your car insurance or homeowners insurance. For example, if you cause a car accident and the medical bills for the other driver exceeds your car insurance policy limit, your umbrella insurance policy can kick in. [4]

Umbrella limit can’t be below zero

Row 290 at an umbrella limit of -1000000

Changed that one limit to 1000000 as this may have been a data entry error as there was only one row that had this issue.

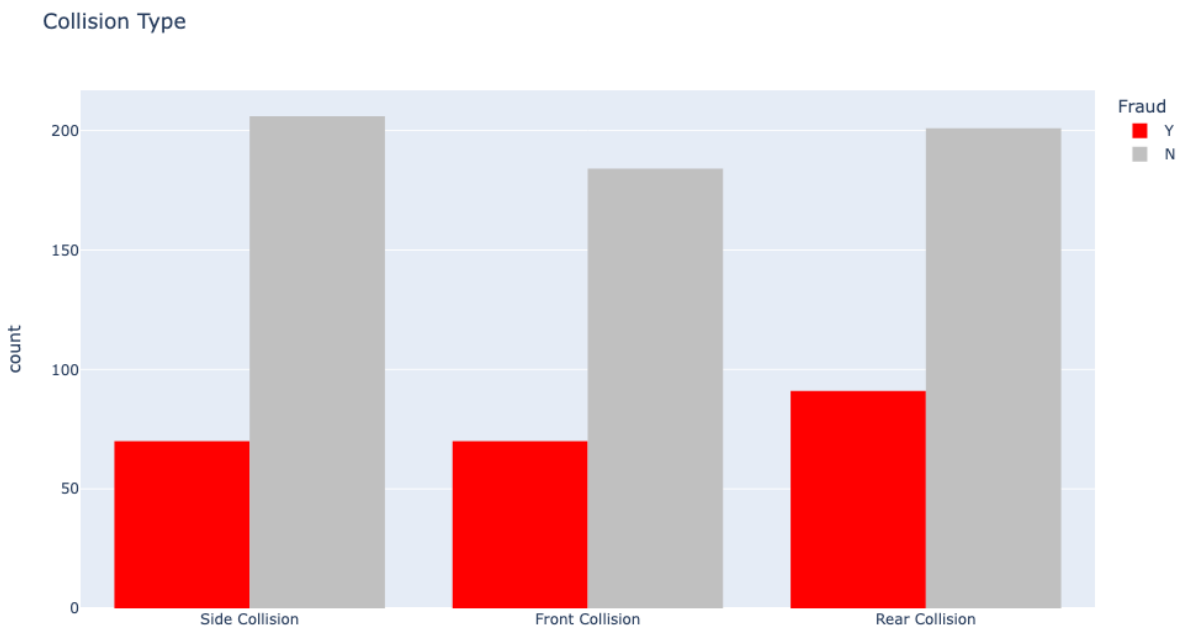
Before Cleaning



Closer look at '?'

Collision type:	Normalized		Values
Rear Collision	0.355231	|	292
Side Collision	0.335766	|	276
Front Collision	0.309002	|	254
NaN		|	178

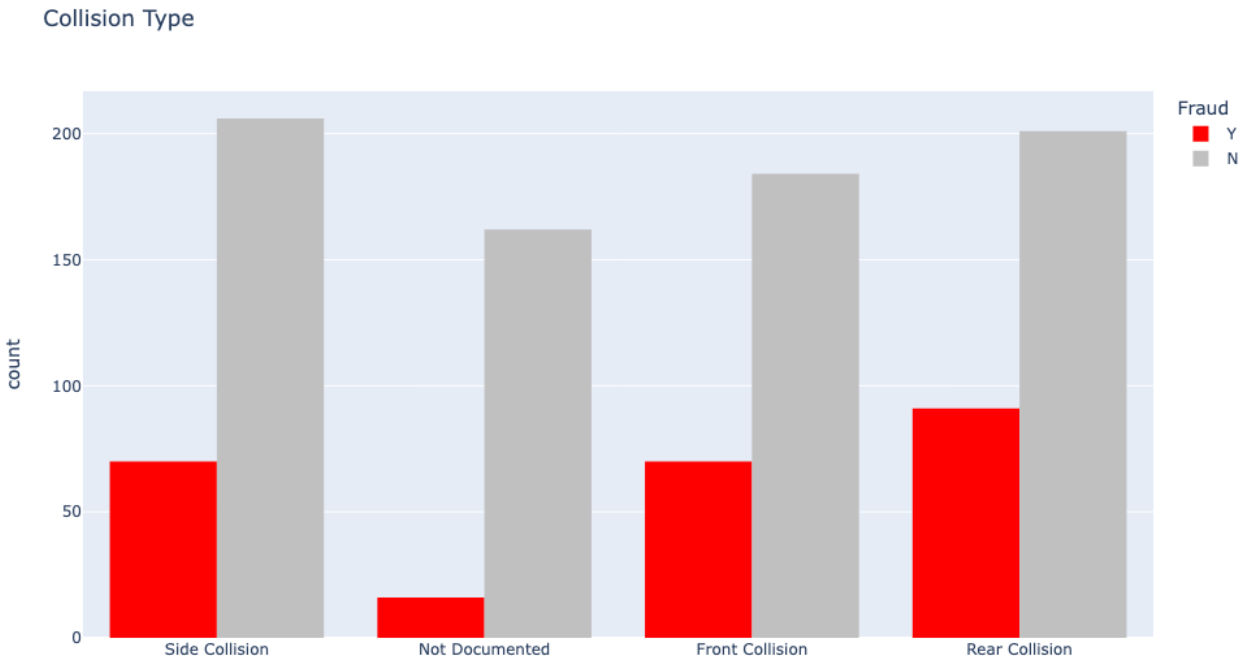
Didn't want to skew the data as it could change what indicates fraud

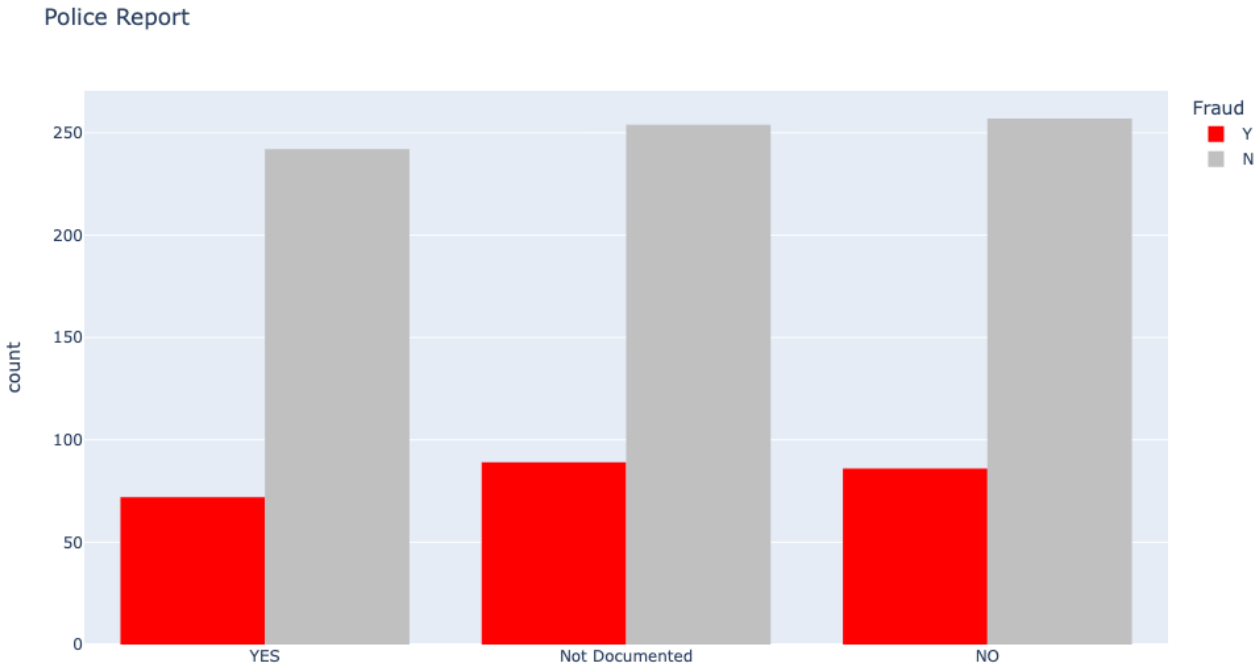


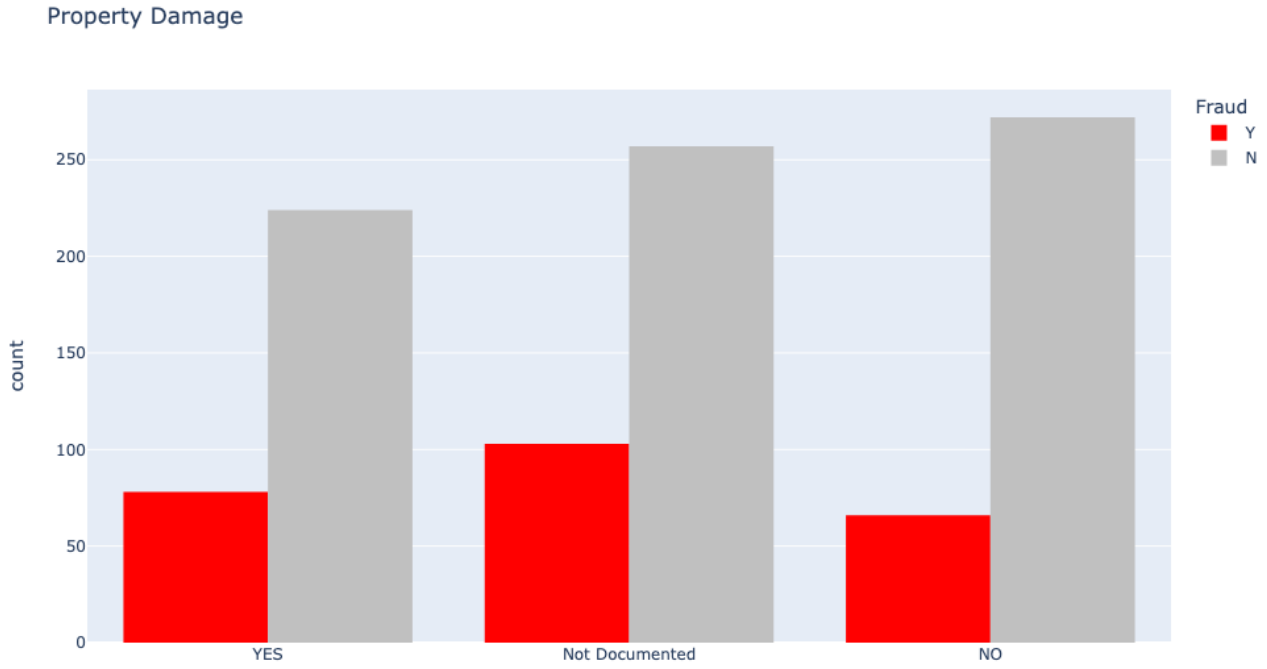
After Cleaning

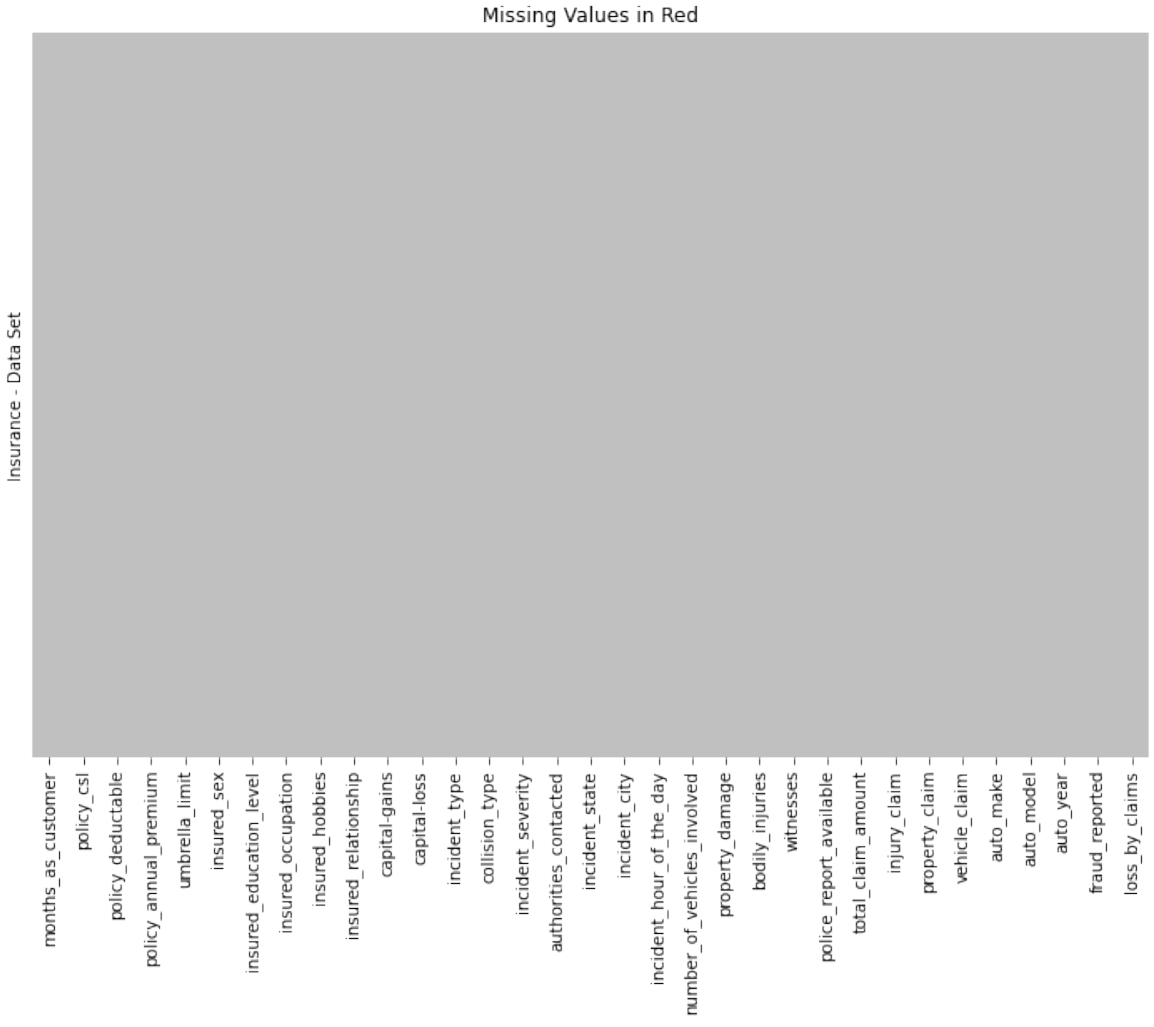
Imputing 'Not Documented' as NaN

Collision Type:	Normalized |	Values
Rear Collision	0.292	292
Side Collision	0.276	276
Front Collision	0.254	254
Not Documented	0.178	178


Collision Type
Fraud
Y
N
count
200
150
100
50
0
Side Collision
Not Documented
Front Collision
Rear Collision


Police Report
Fraud
Y
N
count
250
200
150
100
50
0
YES
Not Documented
NO


Property Damage
Fraud
Y
N
count
250
200
150
100
50
0
YES
Not Documented
NO

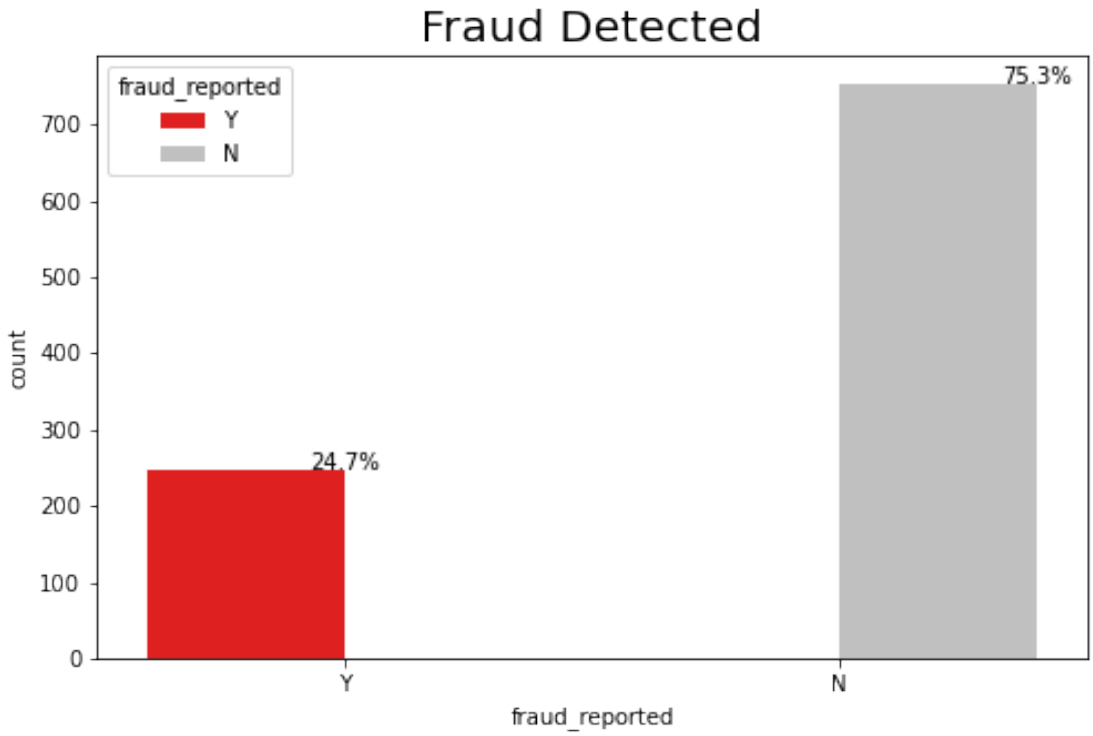

Missing Values in Red
Insurance - Data Set
months_as_customer
policy_csl
policy_deductable
policy_annual_premium
umbrella_limit
insured_sex
insured_education_level
insured_occupation
insured_hobbies
insured_relationship
capital-gains
capital-loss
incident_type
collision_type
incident_severity
authorities_contacted
incident_state
incident_city
incident_hour_of_the_day
number_of_vehicles_involved
property_damage
bodily_injuries
witnesses
police_report_available
total_claim_amount
injury_claim
property_claim
vehicle_claim
auto_make
auto_model
auto_year
fraud_reported
loss_by_claims

EXPLORATORY DATA ANALYSIS

Target Variables: Fraud Reported

As we are trying to predict fraud this would be our target variable

Out of the 1000 rows there were 247 frauds and 753 non-frauds. 24.7% of the data were frauds while 75.3% were non-fraudulent claims.



SEX

Male and Female: 53.7% Females and 46.3% Males in the data set

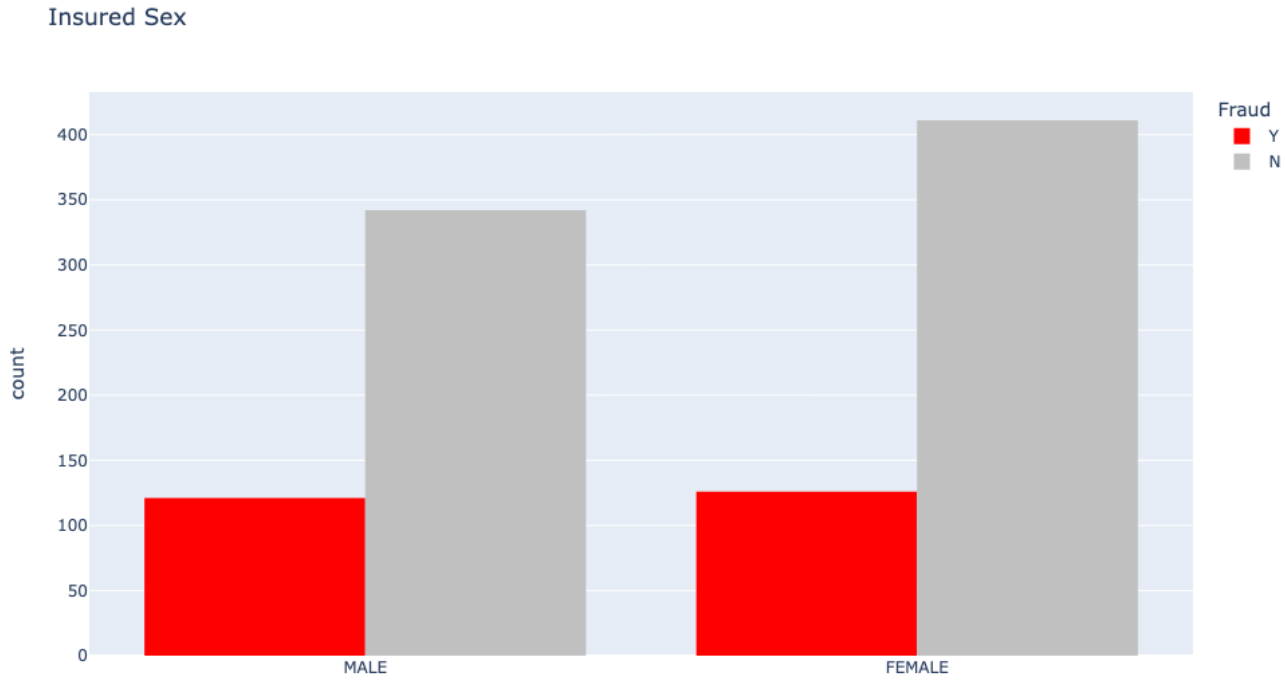
Fraud comparison:

Female: Not Fraud: 76.5%

Fraud: 23.5%

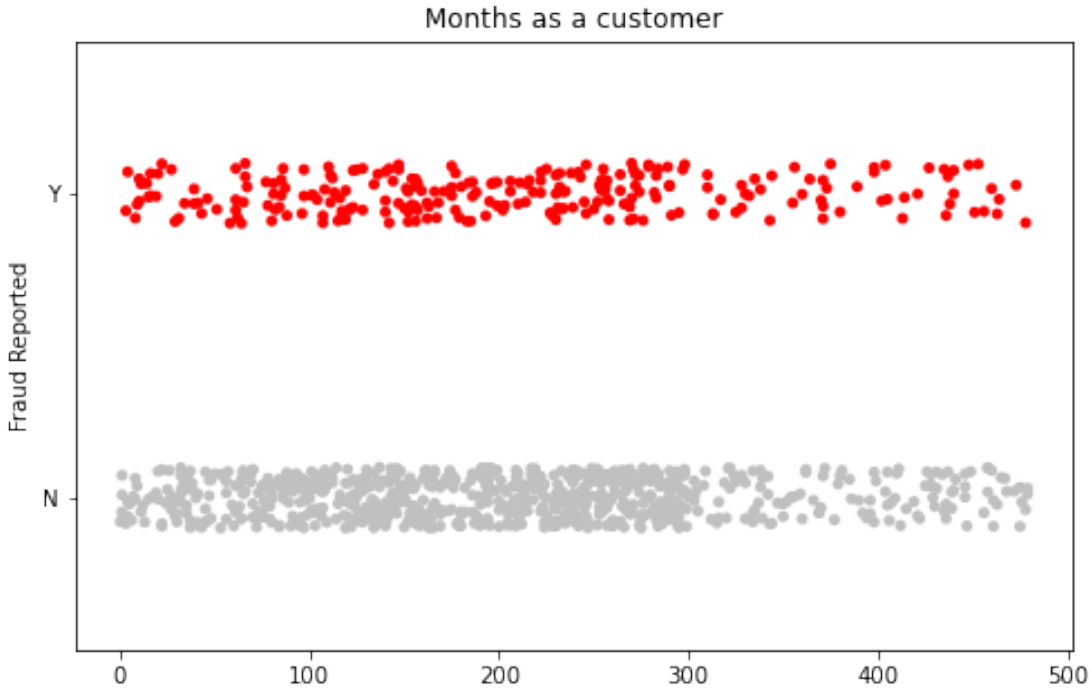
Male: Not Fraud: 73.9%

Fraud: 26.1%

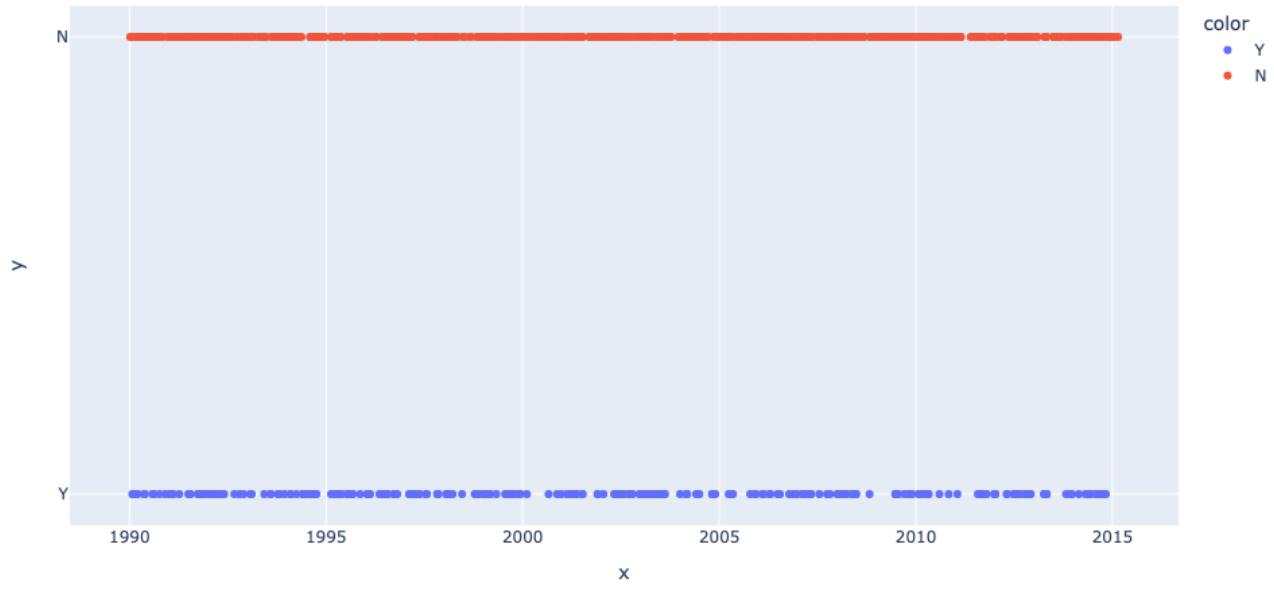


Slightly less amount of males in the data set but they are slightly more likely to commit fraud.

TIME

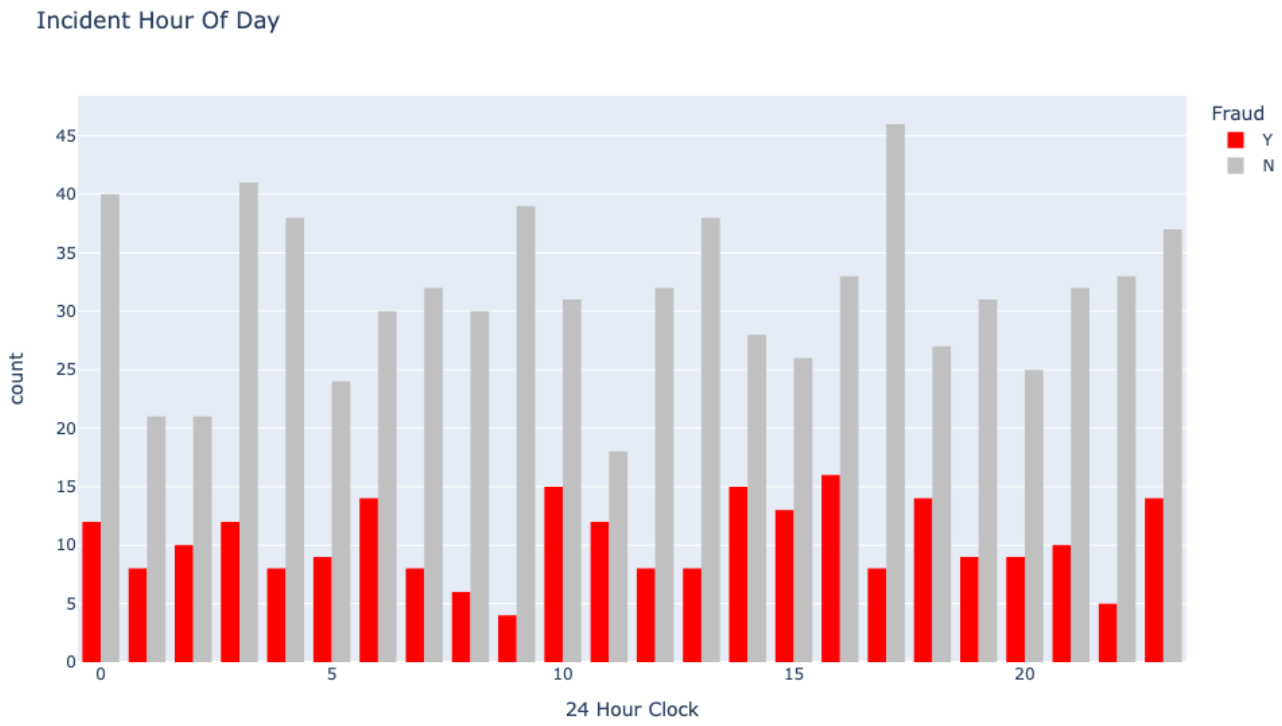


Becomes more dense between 50 months to 300 months



The data set ranges from 1990 – 2015

Fraud occurs throughout the 25 year period – there are some noticeable gaps in-between. Ranges from 1month to 3 months



24 hour clock shows certain times spiking

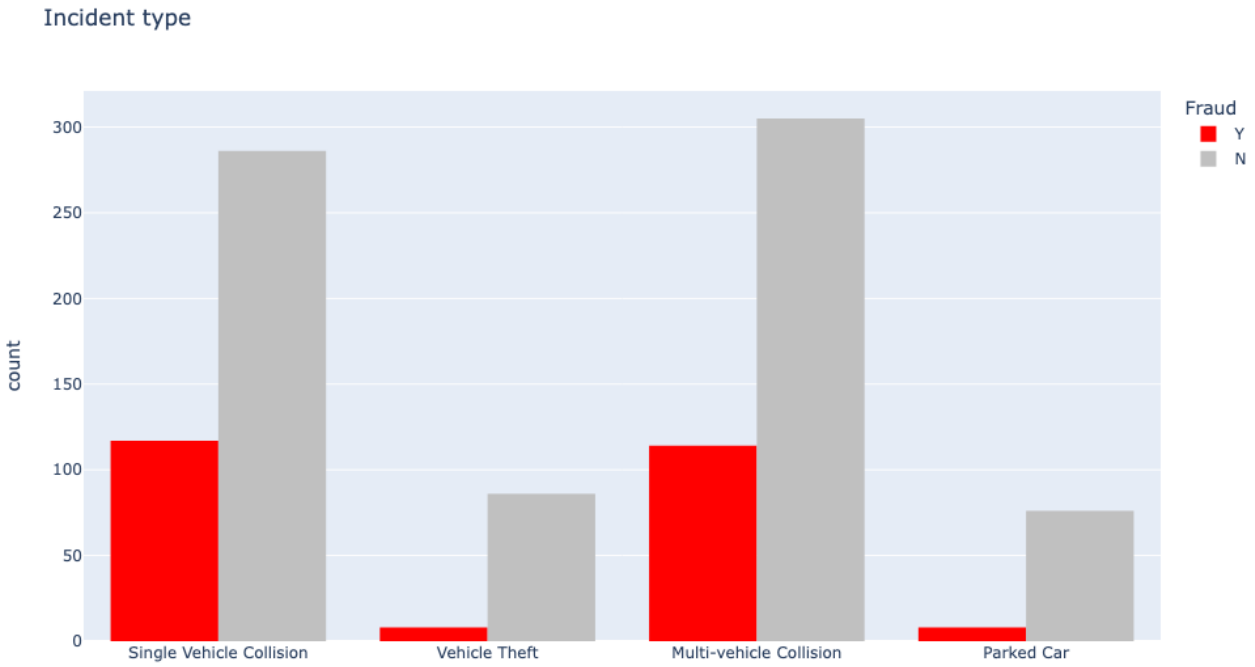
Times that show a higher chance of fraud committed:

6am – 10am – 2pm – 3pm – 4pm – 6pm

INCIDENT TYPE

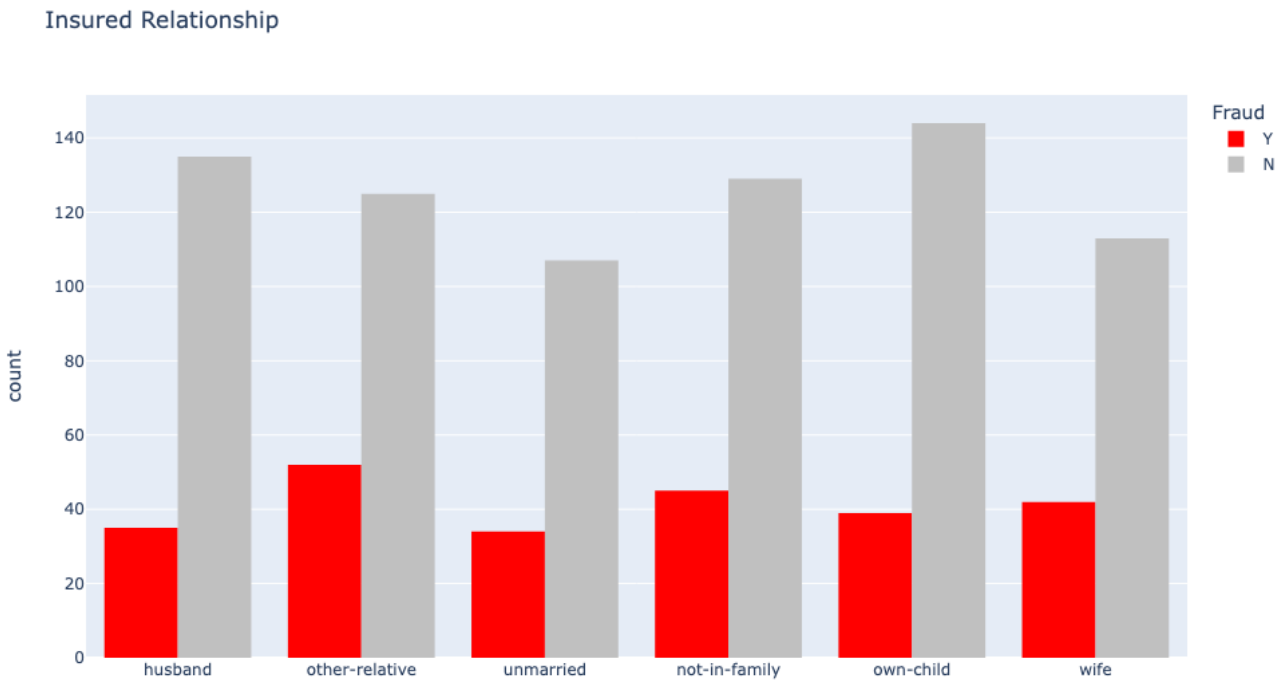
Multi-vehicle Collision	41.9%
Single Vehicle Collision	40.3%
Vehicle Theft	09.4%
Parked Car	08.4%

Incident type	Fraud Reported	Percentage
Multi-vehicle Collision	N	72.79%
	Y	27.20%
Parked Car	N	90.48%
	Y	9.52%
Single Vehicle Collision	N	70.97%
	Y	29.03%
Vehicle Theft	N	91.49%
	Y	8.51%



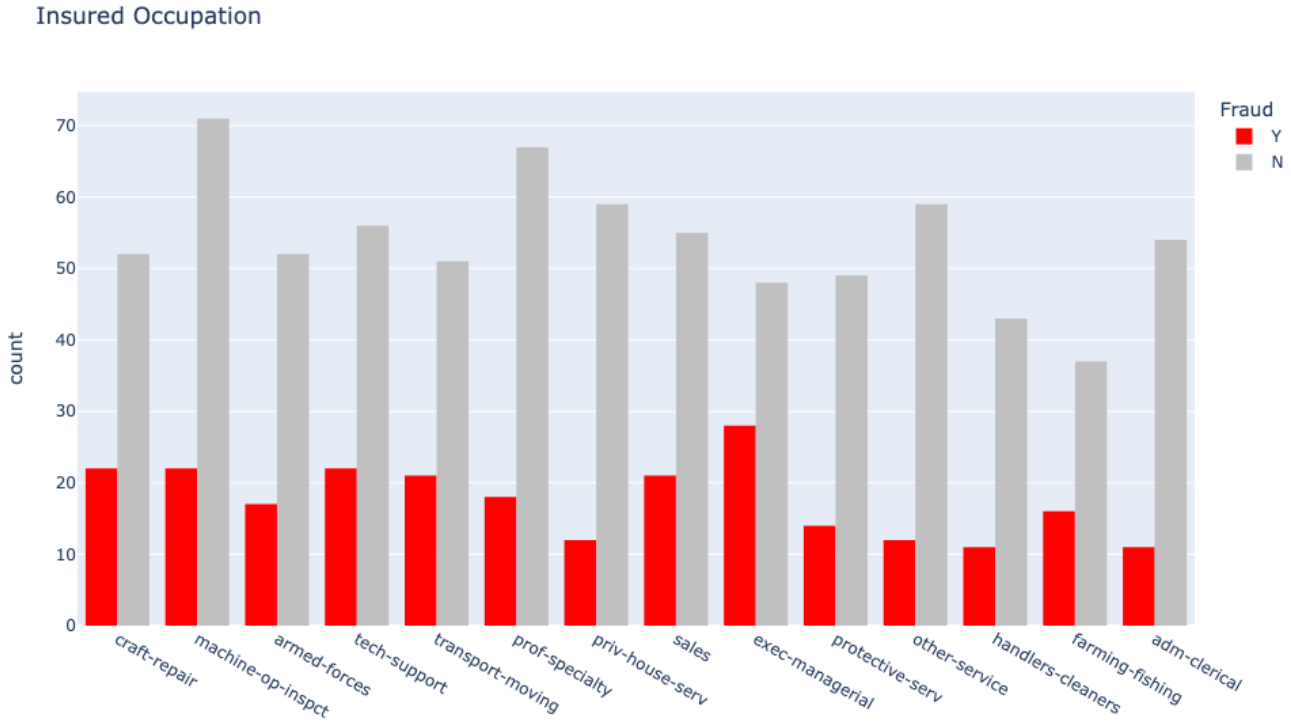
INSURED RELATIONSHIP

Not much that stands out here – slight difference with other relative



OCCUPATION

Exec managerial seems to stand out of the lot

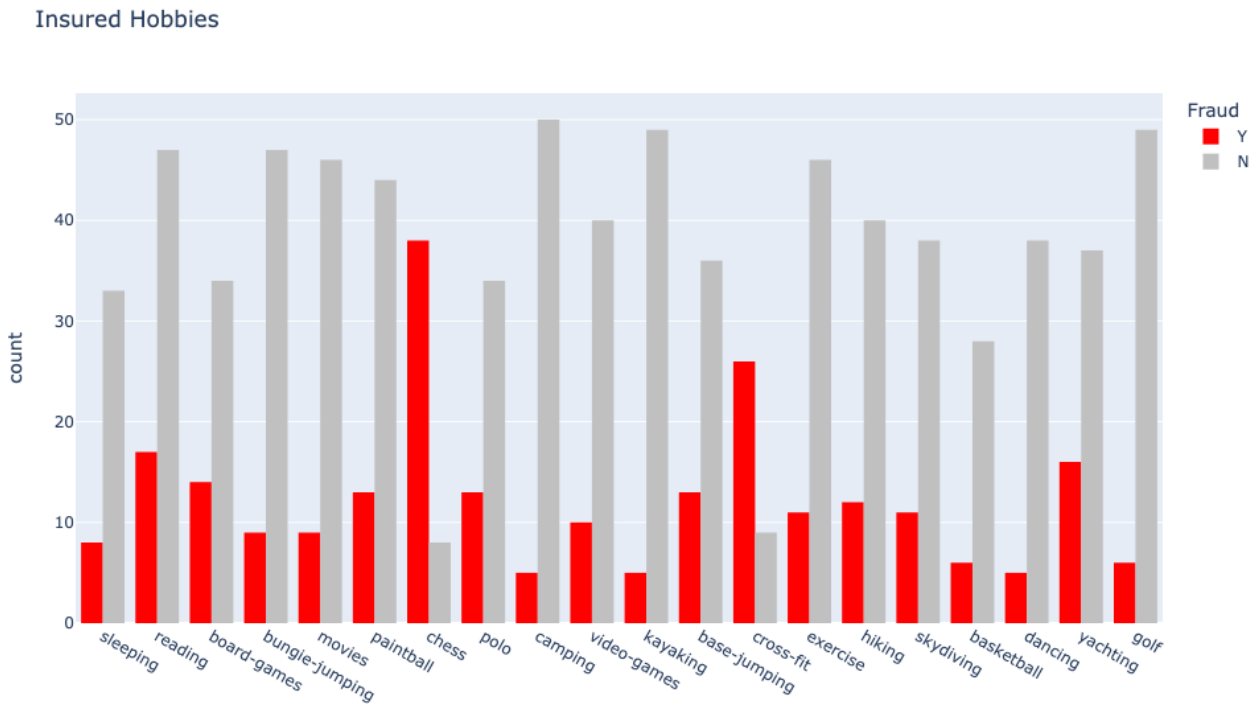


INSURED HOBBIES

Chess and CrossFit show major red flags

Yachting and Board games do stand out

While camping and golf seem very low risk

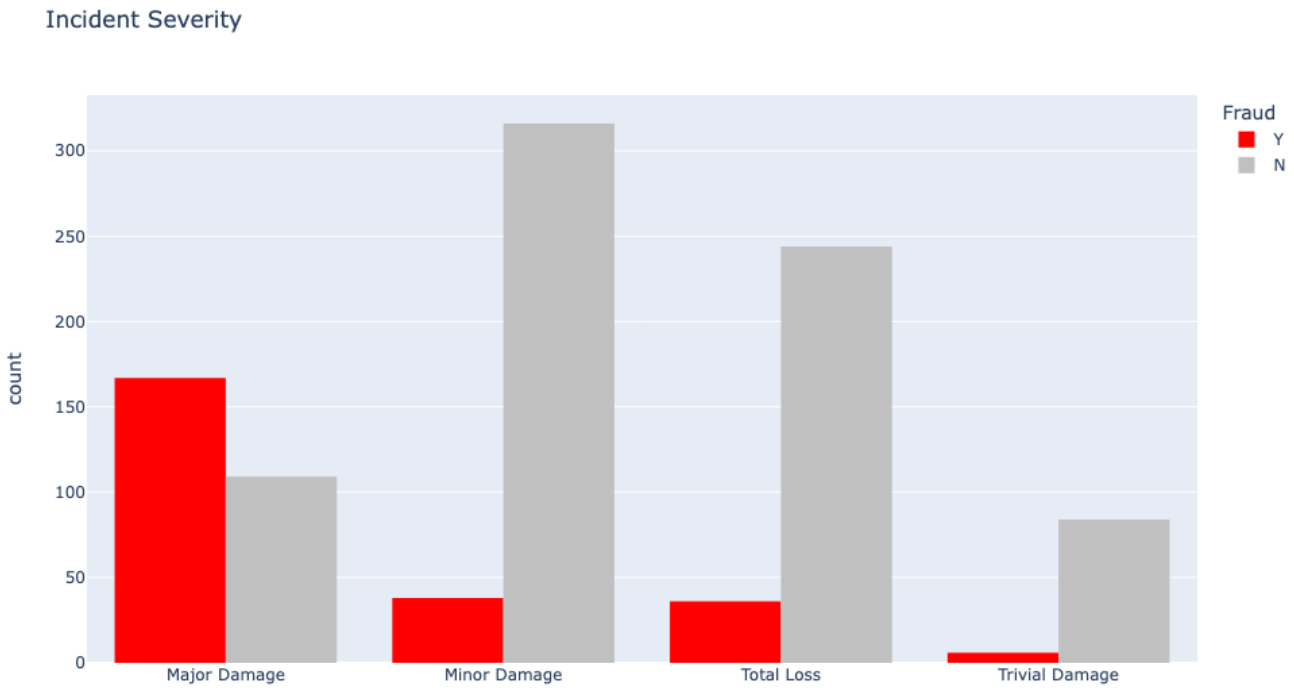


INCIDENT SEVERITY

Major damage seems to sick out a lot

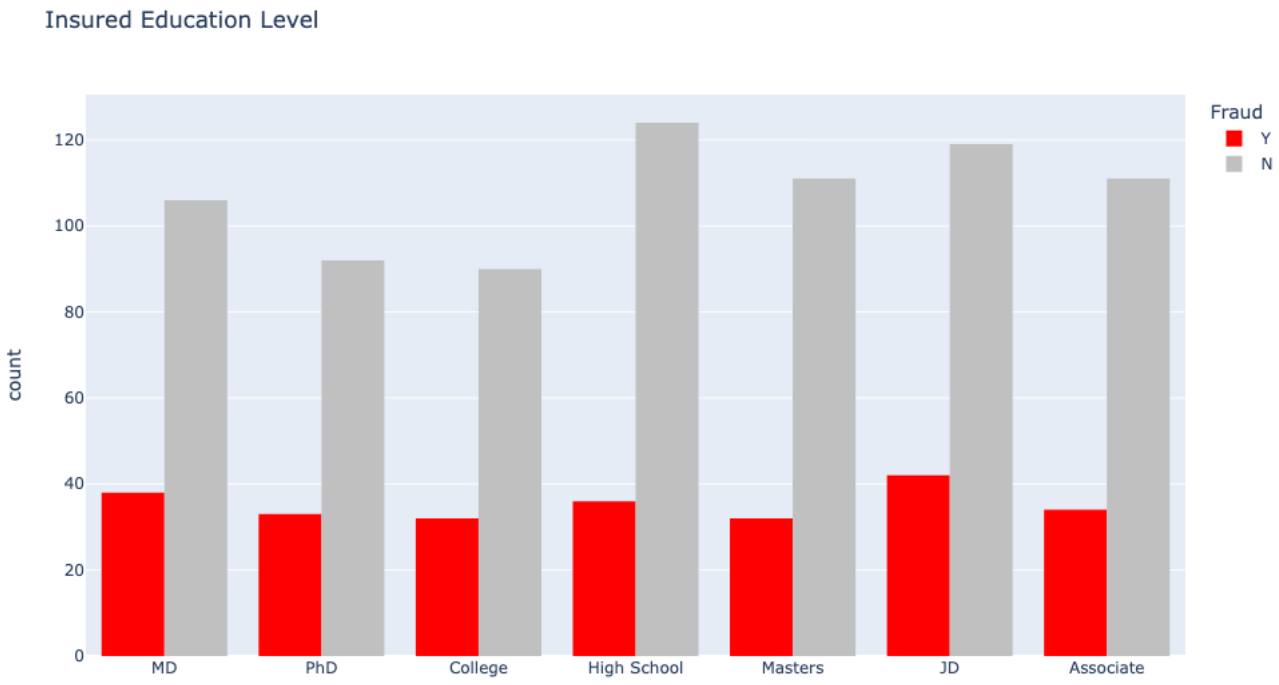
Incident Severity	Data %	Fraud Reported	Fraud %
Major Damage	35.4%	Y	60.50%
		N	39.49%
Minor Damage	28.0%	N	89.27%
		Y	10.73%
Total Loss	27.6%	N	87.14%
		Y	12.86%

Trivial Damage	9.0%	N	93.33%
		Y	6.67%



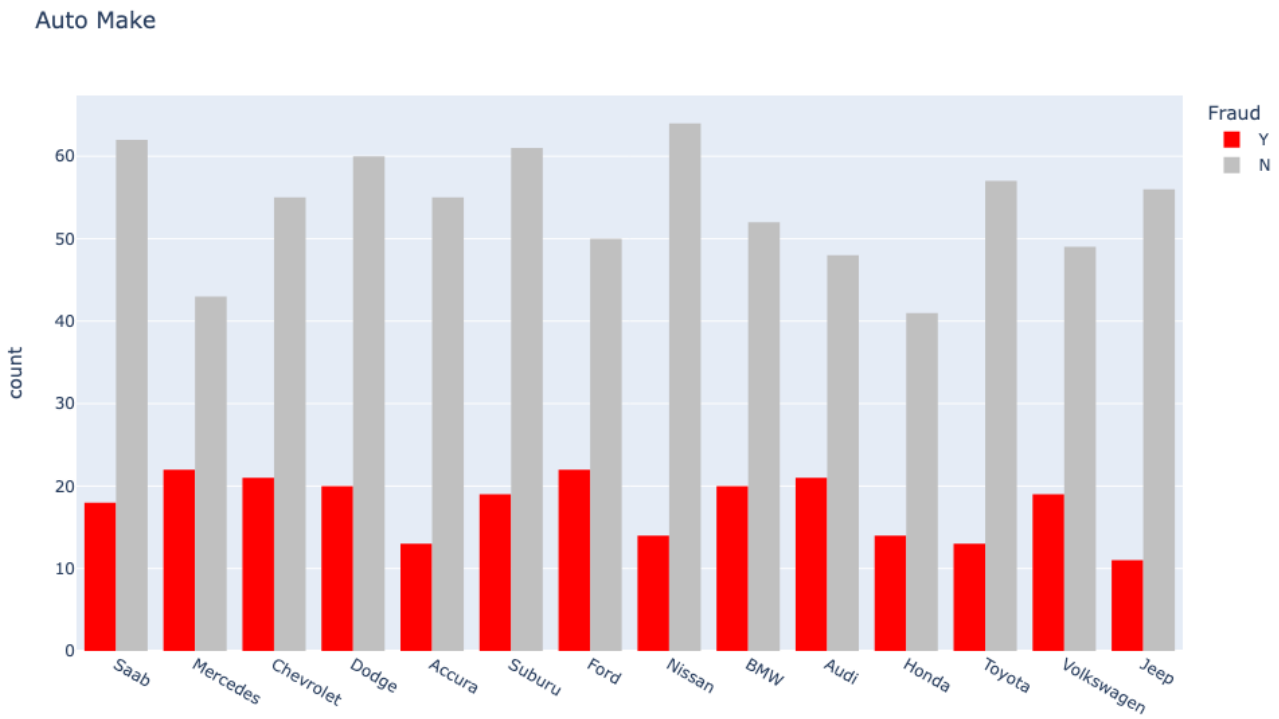
EDUCATION LEVEL

No certain Education level seem to stick out



CAR MAKE

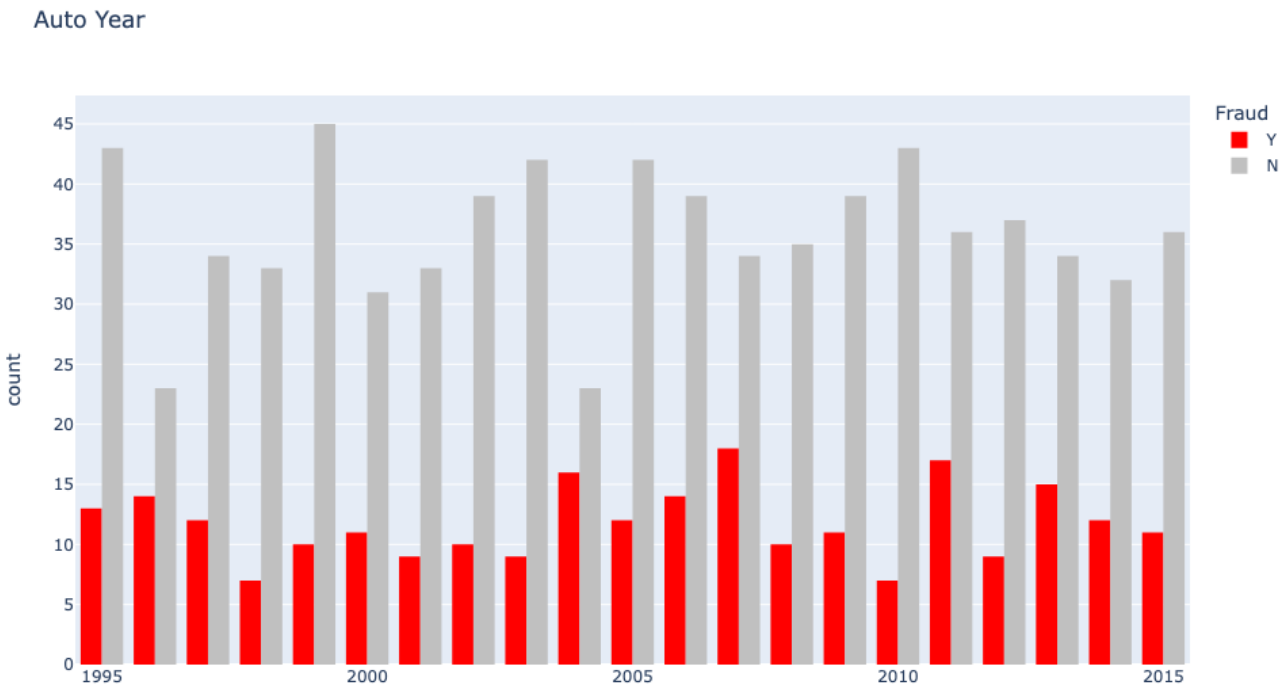
Mercedes, Ford and Volkswagen seem to sick out for card that are more likely for fraud.



CAR YEAR

What year the card was made.

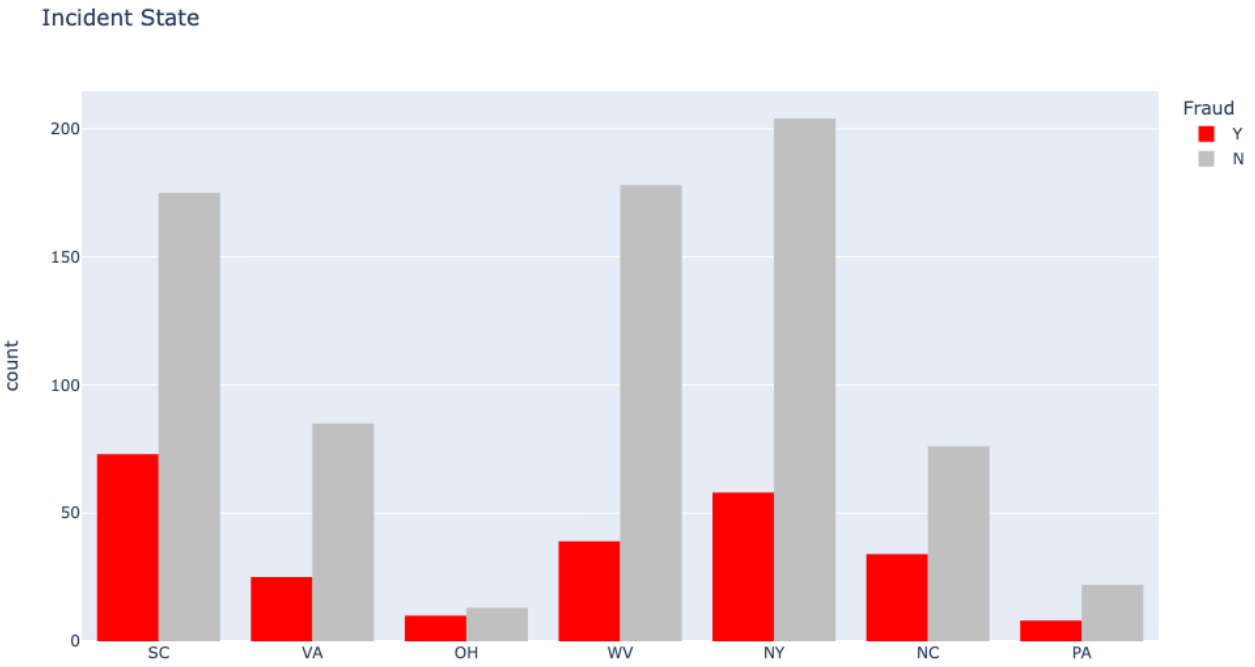
Years that stand out for fraud: 1996, 2004, 2007 & 2011 and 2013



LOCATION

Incident State:

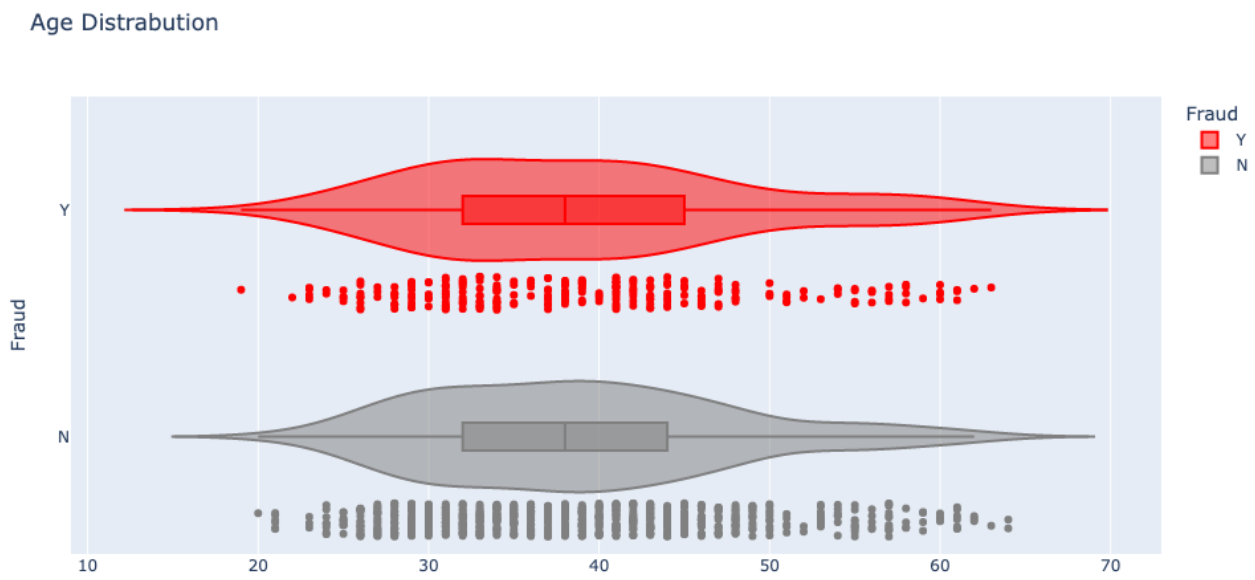
OH, SC and NC stand out



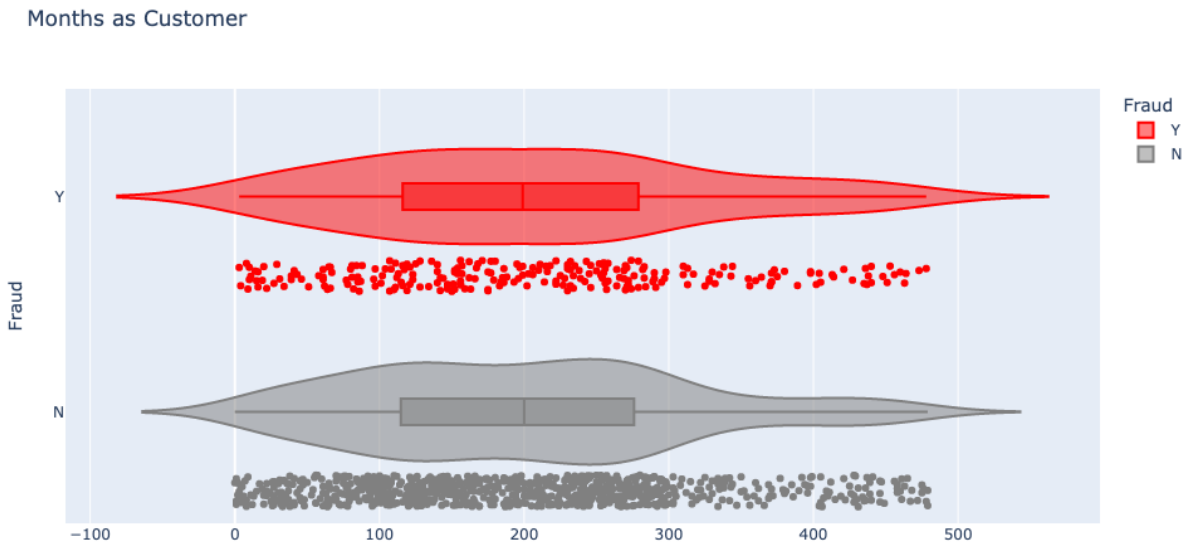
DISTRABUTION

AGE

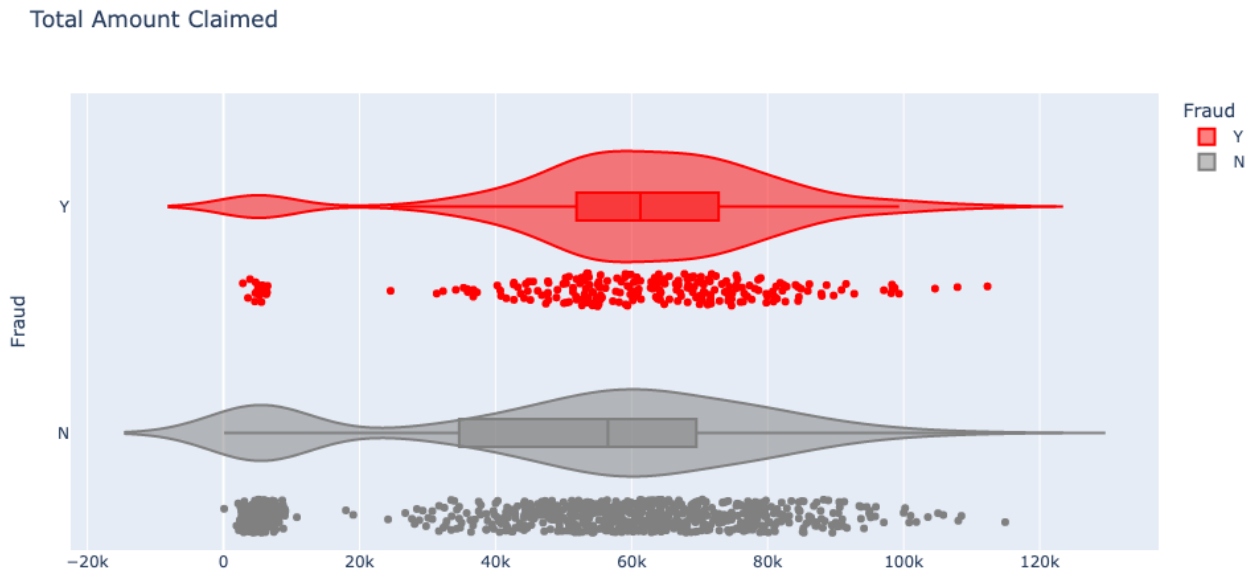
Both fraud and not fraud seem to have similar distribution



MONTHS AS CUSTOMER

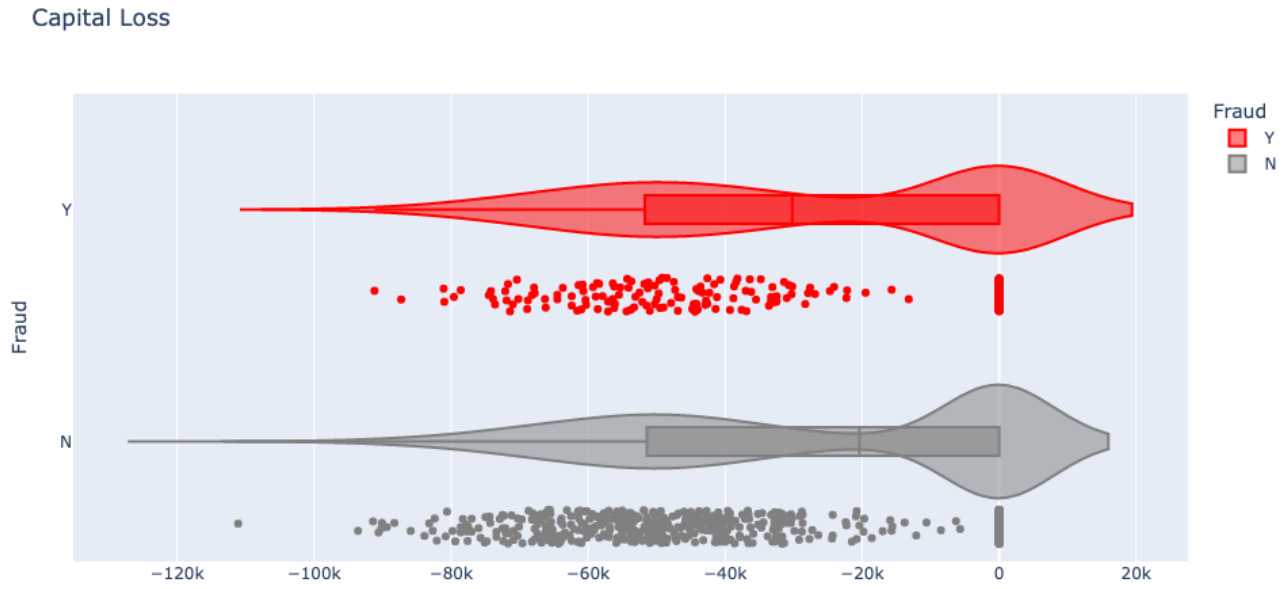


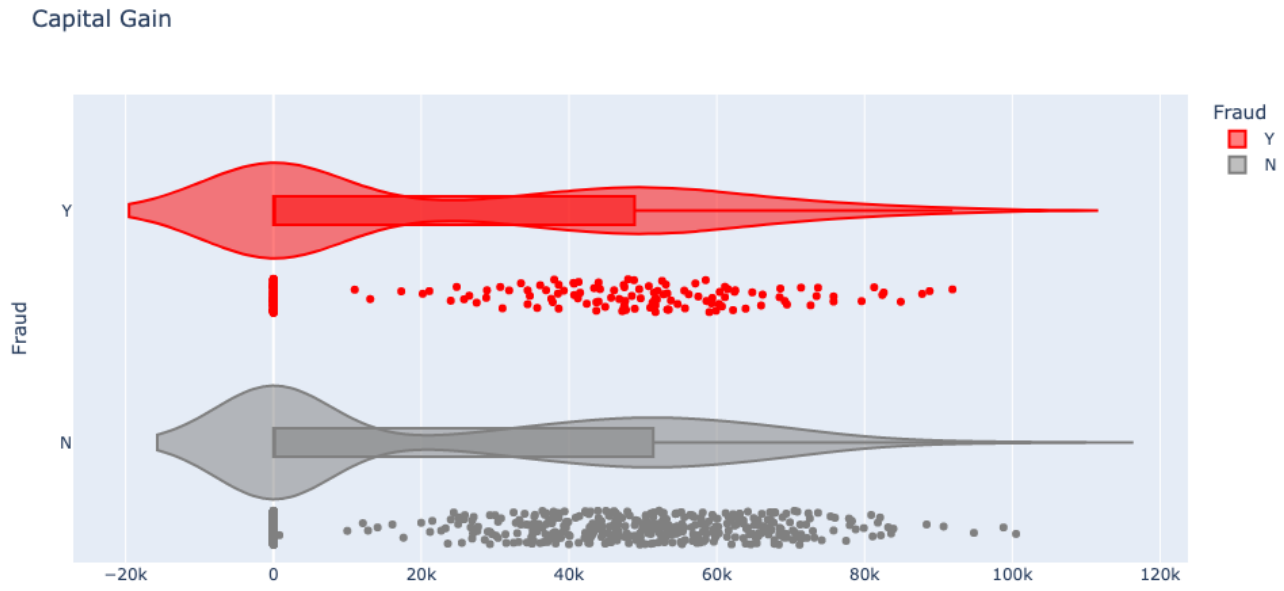
TOTAL AMOUNT CLAIMED



Motetary

Gain Vs Loss Capital





Sum of Capital Gain: $25,126,100

Sum of Capital Loss: $-26,793,700

Total Capital: $-1,667,600

Fraud: Focus on capital that was gain/loss on cases that were fraud

Fraud capital gained: 5975800

Fraud capital loss: -6798100

Total Fraud capital: -822300

Capital without fraud loss vs with fraud loss

Total Capital Without Fraud Loss: 5130500

Total Capital With Fraud Loss: -1667600

Company has a loss of: -32.50%

Claims: How much was claimed?

Total amount claimed: $52,761,940

Total amount claimed that was fraud: $14,894,620

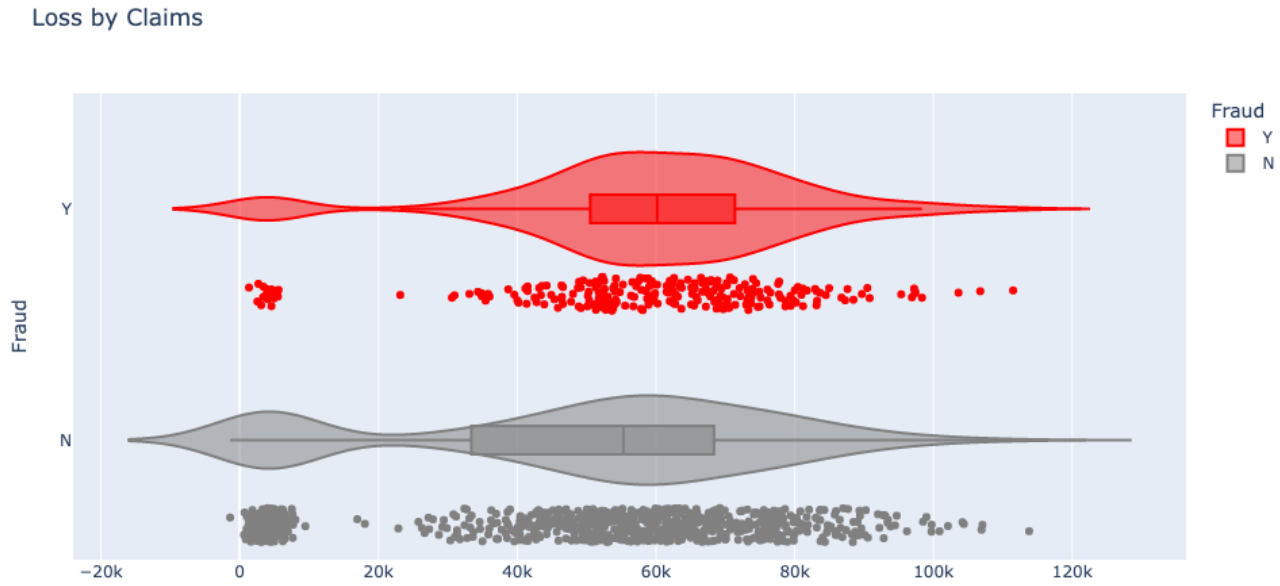
Percentage of loss from fraud: 28.23%

LOSS BY CLAIM

Feature that was engineered utilizing: Total claim amount – Policy annual premium.

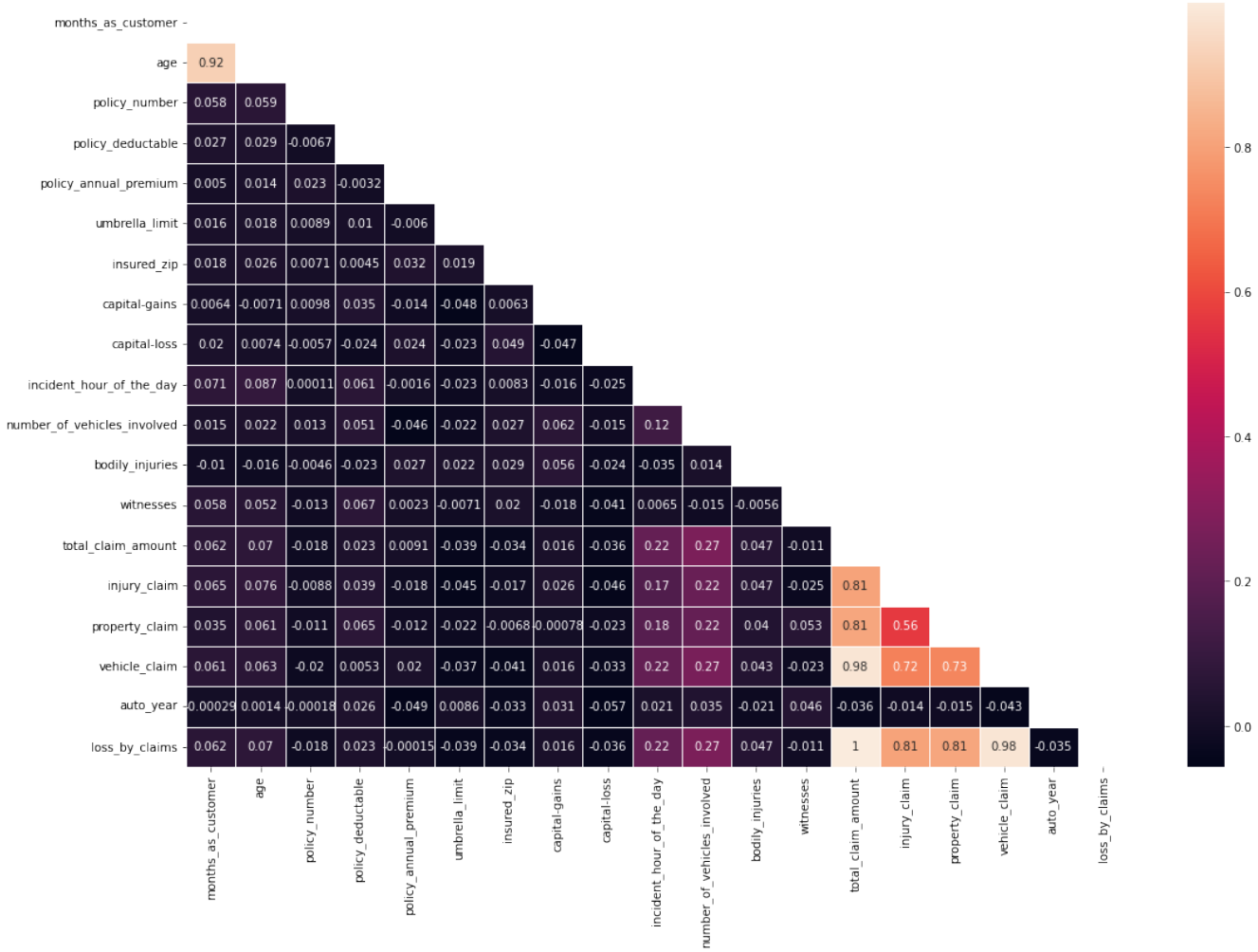
Total Sum	Not Fraud	Fraud
$51,505,533	$26,919,722	$14,585,811
Mean:	$49,030	$59,051

Mean of loss by fraud is $10K higher than not fraud



Correlation

Month as customer and age had a correlation of 0.92. Probably because drivers buy auto insurance when they own a car and this time measure only increases with age.



Pre-Processing

Steps Taken:

1. Encode Data: Make Categorical values into Numerical values. Utilized dummy encode.
2. Scaled Data: Scaled numerical data using Standard Scaler.
3. Bought numerical data with categorical to make final Data Frame

Machine Learning

Applying Models

Will be looking to Apply 7 different models + baseline

1. Logistic regression
2. KNN
3. Decision Tree
4. Random Forest
5. SVC
6. XGBoost
7. Stacking

They will all be supported with Grid Search to find best parameters

Best Scores will be compared.

SMOTE

To make the imbalance in minority class "fraud" equal to "not fraud" Making the training sample 50/50 Fraud

Will be applying 3 Models:

1. Logistic Regression
2. XGBoost
3. Decision Tree

GRID SEARCH

What is grid search?

GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. [6]

BASELINE

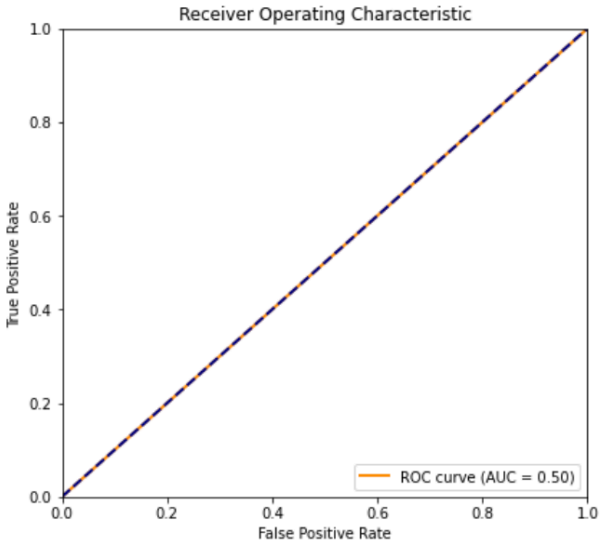
Model Used: Dummy Classifier: Strategy='Most frequent

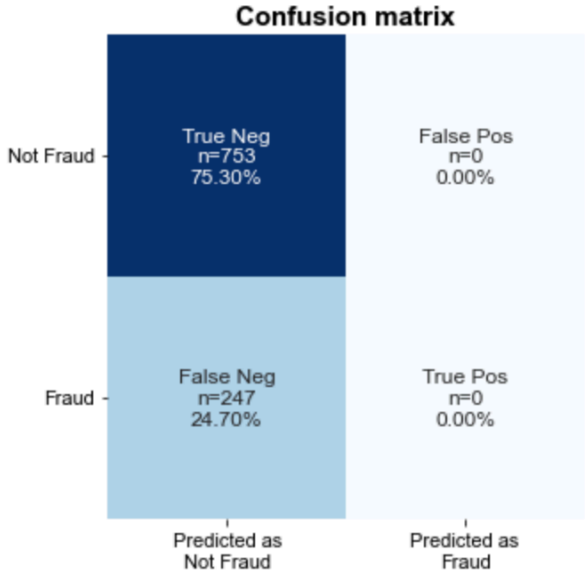
What is this?

DummyClassifier makes predictions that ignore the input features.

This classifier serves as a simple baseline to compare against other more complex classifiers.[5]

```
Accuracy is:  0.753
              precision    recall  f1-score   support

           0       0.75      1.00      0.86       753
           1       0.00      0.00      0.00       247

    accuracy                           0.75      1000
   macro avg       0.38      0.50      0.43      1000
weighted avg       0.57      0.75      0.65      1000

Accuracy : 0.7530 [TP / N]          Best: 1, Worst: 0
           Proportion of predicted labels that match the true lab
els.
Precision: 0.0000 [TP / (TP + FP)] Best: 1, Worst: 0
           Not to label a negative sample as positive.
Recall   : 0.0000 [TP / (TP + FN)] Best: 1, Worst: 0
           Find all the positive samples.
ROC AUC  : 0.5000                  Best: 1, Worst: < 0.5
```





As expected since there was 75.3% of non-fraud data sets it is predicting this value.

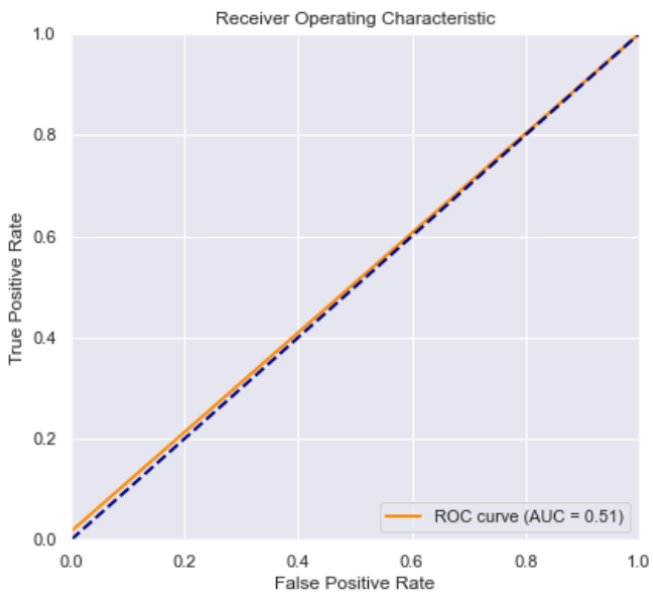
LOGISTIC REGRESSION

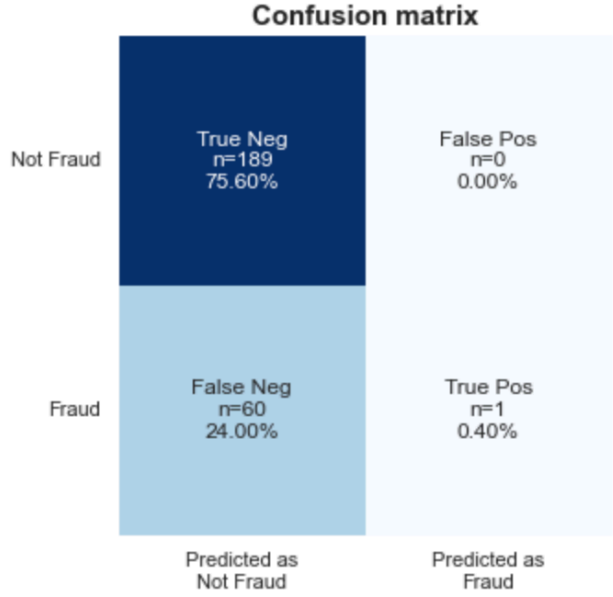
What is logistic Regression?

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. [7]

Model Applied: LogisticRegression()

```
Training Model Score:  0.748
Accuracy is:  0.76
              precision    recall  f1-score   support

           0       0.76      1.00      0.86       189
           1       1.00      0.02      0.03        61

    accuracy                           0.76       250
   macro avg       0.88      0.51      0.45       250
weighted avg       0.82      0.76      0.66       250
```

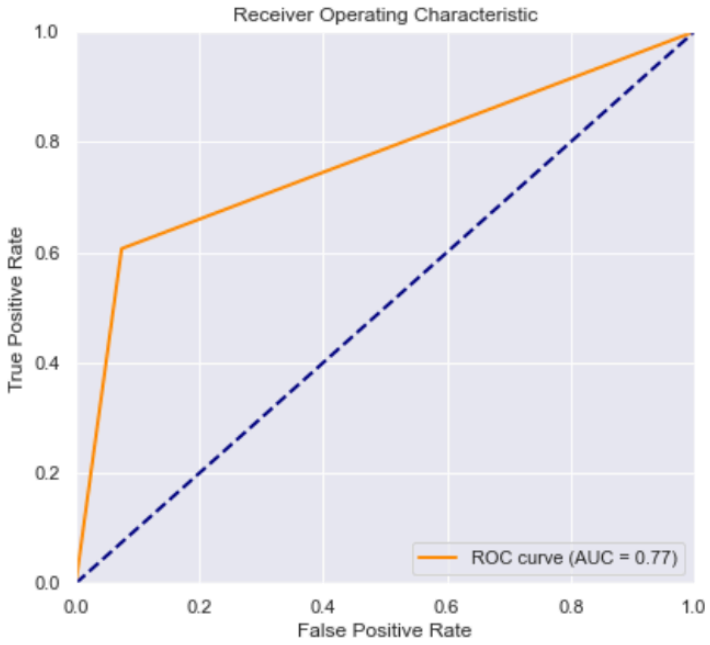


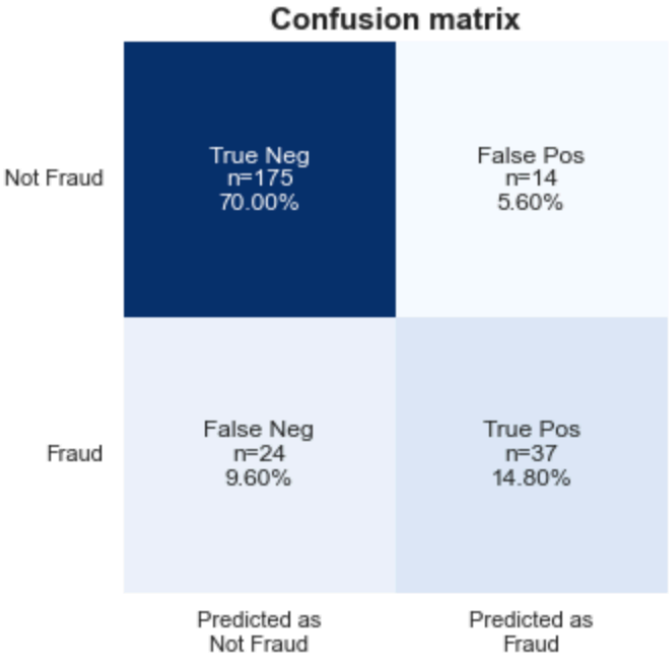


Logistic Regression with Grid Search

Best parameters: C=1, penalty='l1', solver='liblinear'

```
Training Model Score:  0.8706666666666667
Accuracy is:  0.848
              precision    recall  f1-score   support

           0       0.88      0.93      0.90       189
           1       0.73      0.61      0.66        61

    accuracy                           0.85       250
   macro avg       0.80      0.77      0.78       250
weighted avg       0.84      0.85      0.84       250
```





KNN

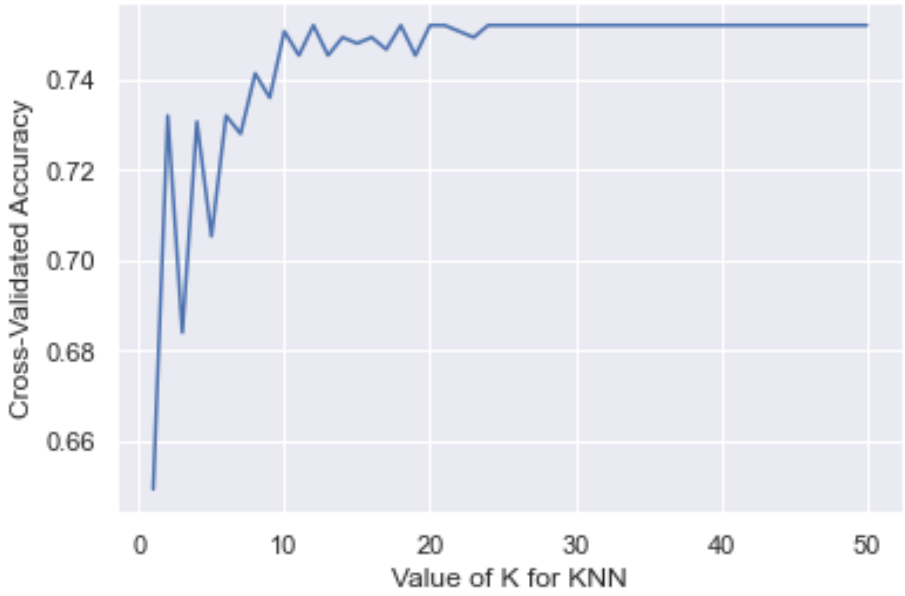
What is KNN?

A **supervised machine learning** algorithm is one that relies on labelled input data to learn a function that produces an appropriate output when given new unlabelled data. [8]

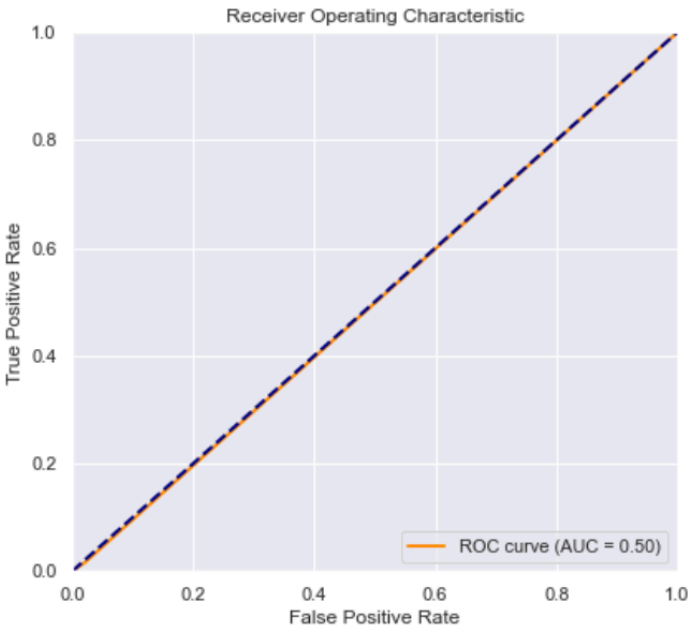
Model = KNeighborsClassifier()

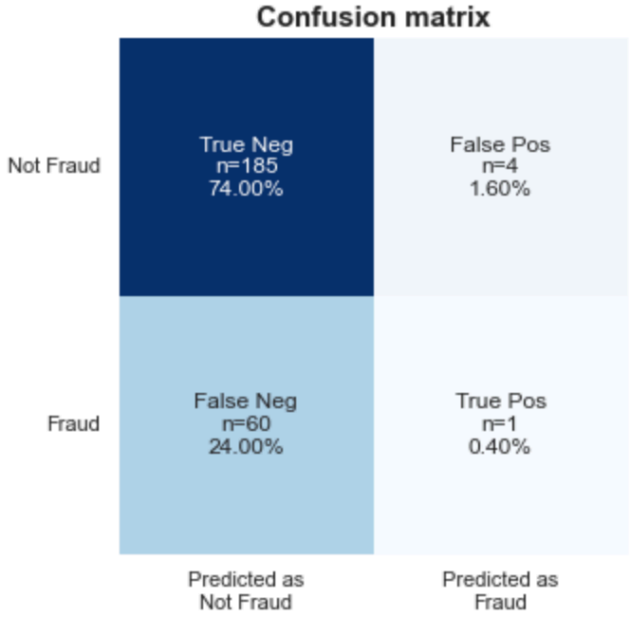
```
Training Model Score:  0.7853333333333333
Accuracy is:  0.712
              precision    recall  f1-score   support

           0       0.76      0.91      0.83       189
           1       0.26      0.10      0.14        61

    accuracy                           0.71       250
   macro avg       0.51      0.50      0.48       250
weighted avg       0.64      0.71      0.66       250
```

Grid search best: n_neighbors=18



Model: KNeighborsClassifier(n_neighbors=18)





DECISION TREE

What is Decision Tree?

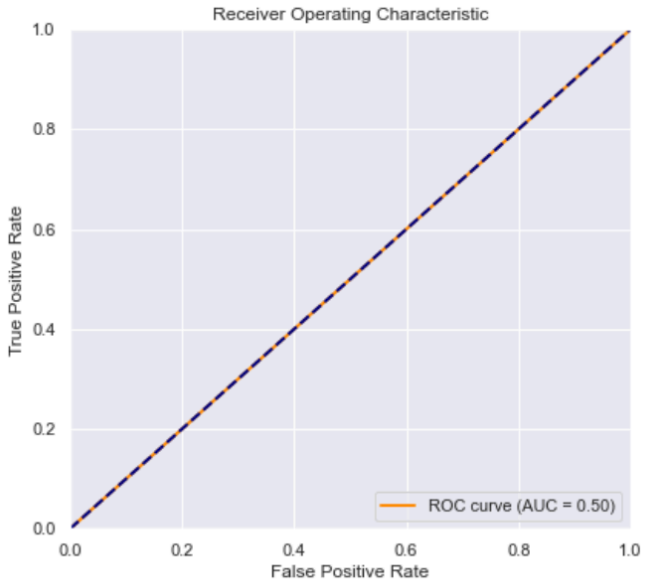
Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. [9]

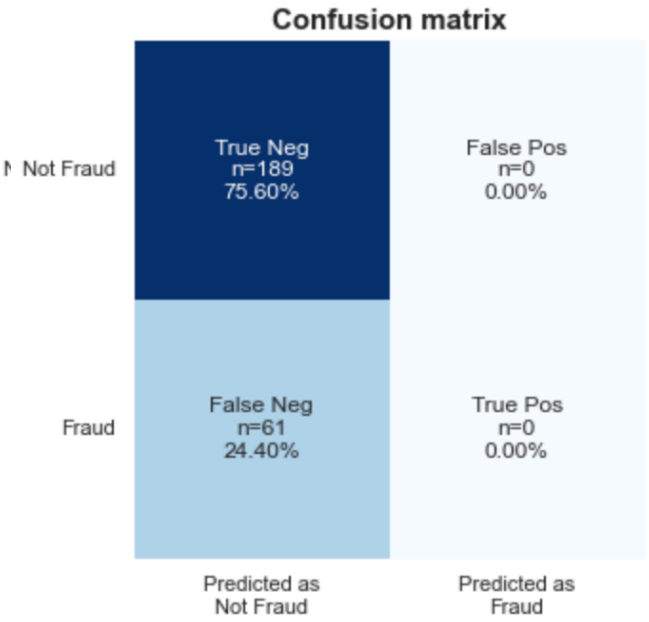
Model: DecisionTreeClassifier()

```
Training Model Score:  1.0
Accuracy is:  0.796
              precision    recall  f1-score   support

           0       0.84      0.90      0.87       189
           1       0.60      0.48      0.53        61

    accuracy                           0.80       250
   macro avg       0.72      0.69      0.70       250
weighted avg       0.78      0.80      0.79       250
```

Grid search best parameters:
DecisionTreeClassifier(ccp_alpha=0.01, criterion='entropy', max_depth=16, max_features='auto')

```
Training Model Score:  0.752
Accuracy is:  0.756
              precision    recall  f1-score   support

           0       0.76      1.00      0.86       189
           1       0.00      0.00      0.00        61

    accuracy                           0.76       250
   macro avg       0.38      0.50      0.43       250
weighted avg       0.57      0.76      0.65       250
```





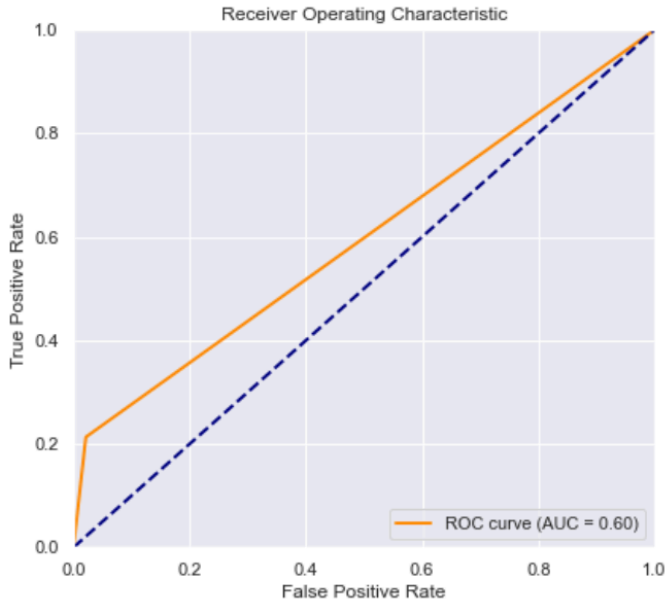
RANDOM FOREST

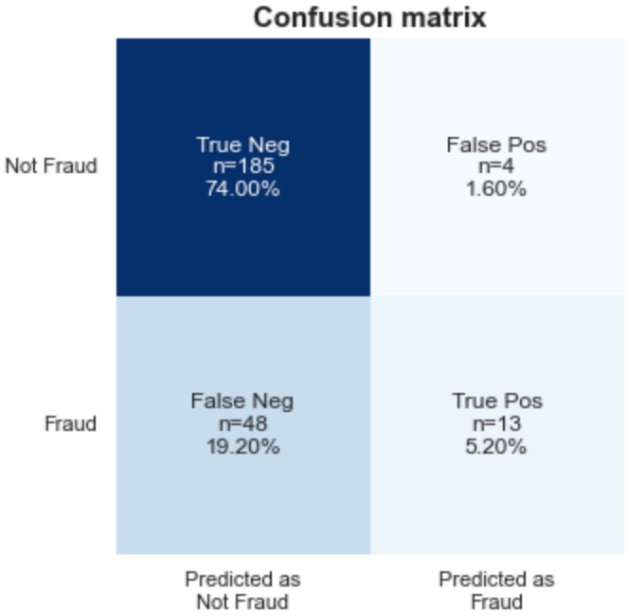
What is Random Forest?

Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems. [10]

Model: RandomForestClassifier()

```
Training Model Score:  1.0
Accuracy is:  0.792
              precision    recall  f1-score   support

           0       0.79      0.98      0.88       189
           1       0.76      0.21      0.33        61

    accuracy                           0.79       250
   macro avg       0.78      0.60      0.61       250
weighted avg       0.79      0.79      0.74       250
```





Grid Search best parameters:
RandomForestClassifier(max_features='sqrt', n_estimators= 500, max_depth=8, criterion='gini')

```
Training Model Score:  0.852
Accuracy is:  0.764
              precision    recall  f1-score   support

           0       0.76      1.00      0.86       189
           1       1.00      0.03      0.06        61

    accuracy                           0.76       250
   macro avg       0.88      0.52      0.46       250
weighted avg       0.82      0.76      0.67       250
```

SVC

What is SVC?

SVC, or Support Vector Classifier, is **a supervised machine learning algorithm typically used for classification tasks**. SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes. [11]

Model: SVC()

```
Training Model Score:  0.752
Accuracy is:  0.756
              precision    recall  f1-score   support

           0       0.76      1.00      0.86       189
           1       0.00      0.00      0.00        61

    accuracy                           0.76       250
   macro avg       0.38      0.50      0.43       250
weighted avg       0.57      0.76      0.65       250
```

XGBOOST

What is XGBoost?

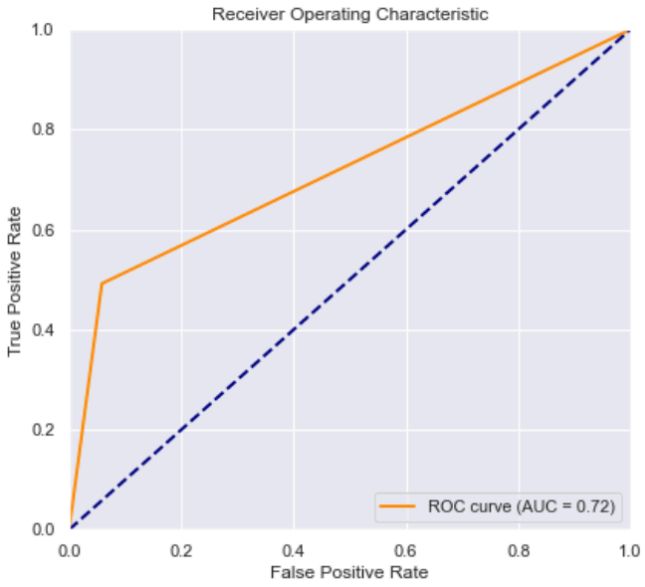
XGBoost, which stands for Extreme Gradient Boosting, is **a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library**. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. [12]

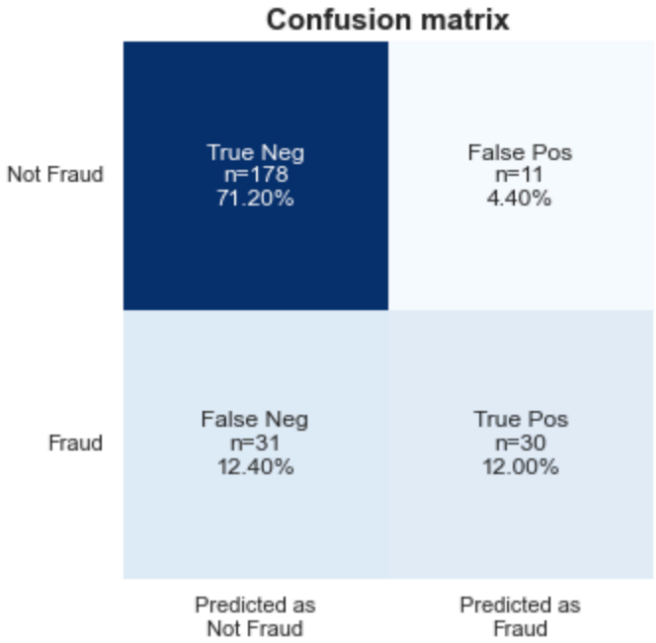
Model: XGBClassifier()

```
Training Model Score:  1.0
Accuracy is:  0.832
              precision    recall  f1-score   support

           0       0.85      0.94      0.89       189
           1       0.73      0.49      0.59        61

    accuracy                           0.83       250
   macro avg       0.79      0.72      0.74       250
weighted avg       0.82      0.83      0.82       250
```

Gride Search best parameters: XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None, reg_lambda= 0.05, reg_alpha= 0.5, n_estimators= 350, max_depth= 5, gamma= 0, eta= 0.05)

```
Training Model Score:  1.0
Accuracy is:  0.836
              precision    recall  f1-score   support

           0       0.86      0.94      0.90       189
           1       0.73      0.52      0.61        61

    accuracy                           0.84       250
   macro avg       0.79      0.73      0.75       250
weighted avg       0.83      0.84      0.83       250
```





STACKING

What Is stacking?

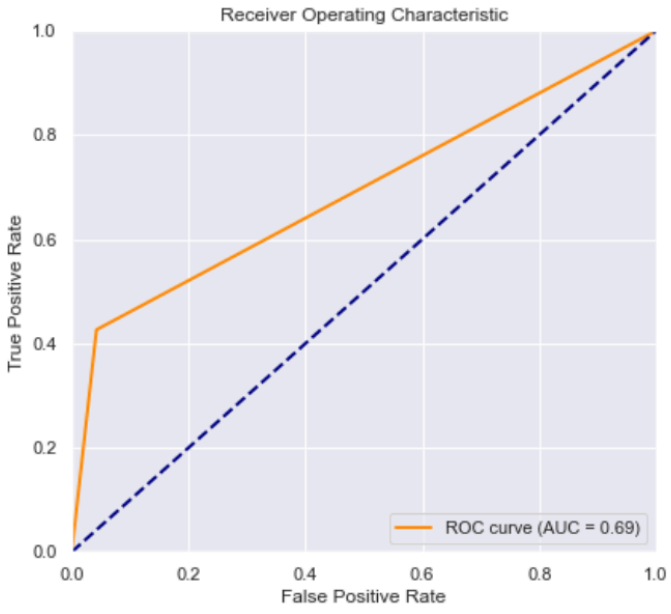
Stacking is **one of the most popular ensemble machine learning techniques used to predict multiple nodes to build a new model and improve model performance**. Stacking enables us to train multiple models to solve similar problems, and based on their combined output, it builds a new model with improved performance. [13]

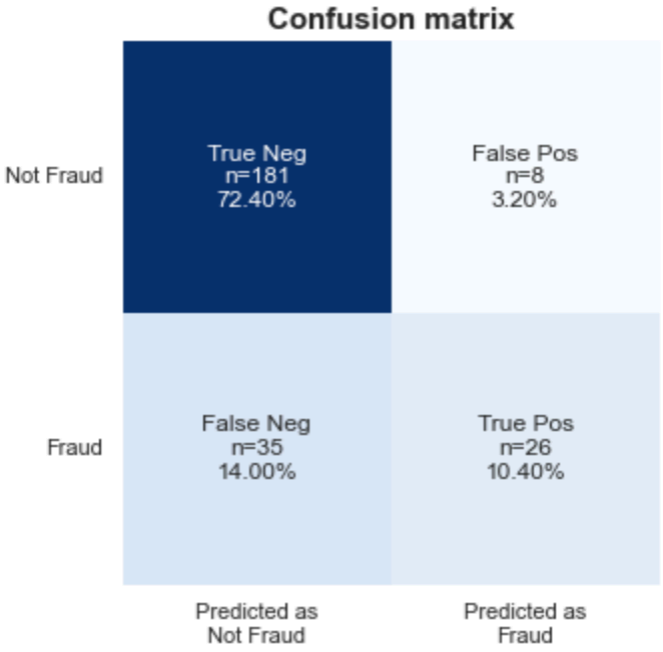
Models that were stacked: Logistic Regression, Random forest, xgboost.

Meta Classifier: Logistic Regression

Model: StackingClassifier(classifiers = [lin_modelf, model_rf,xgb_model], meta_classifier = lin_model)

```
Training Model Score:  1.0
Accuracy is:  0.828
              precision    recall  f1-score   support

           0       0.84      0.96      0.89       189
           1       0.76      0.43      0.55        61

    accuracy                           0.83       250
   macro avg       0.80      0.69      0.72       250
weighted avg       0.82      0.83      0.81       250
```

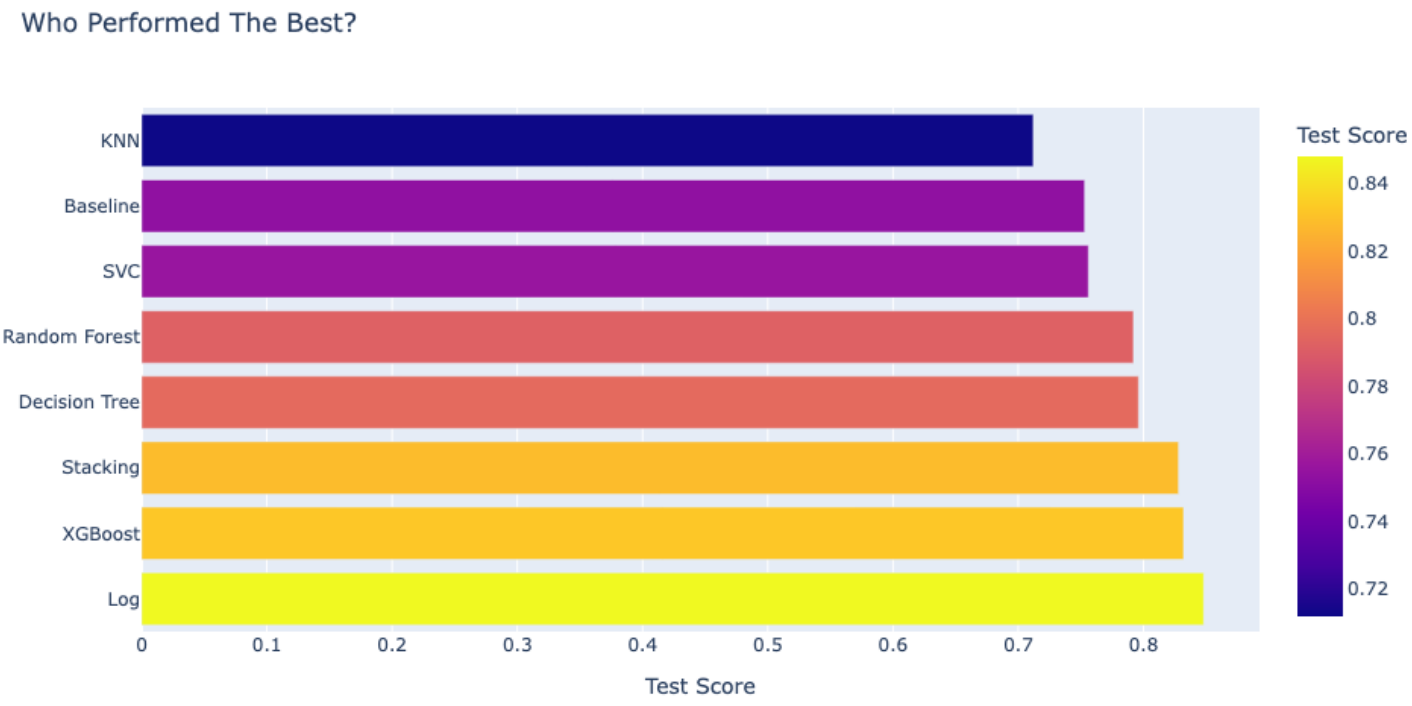




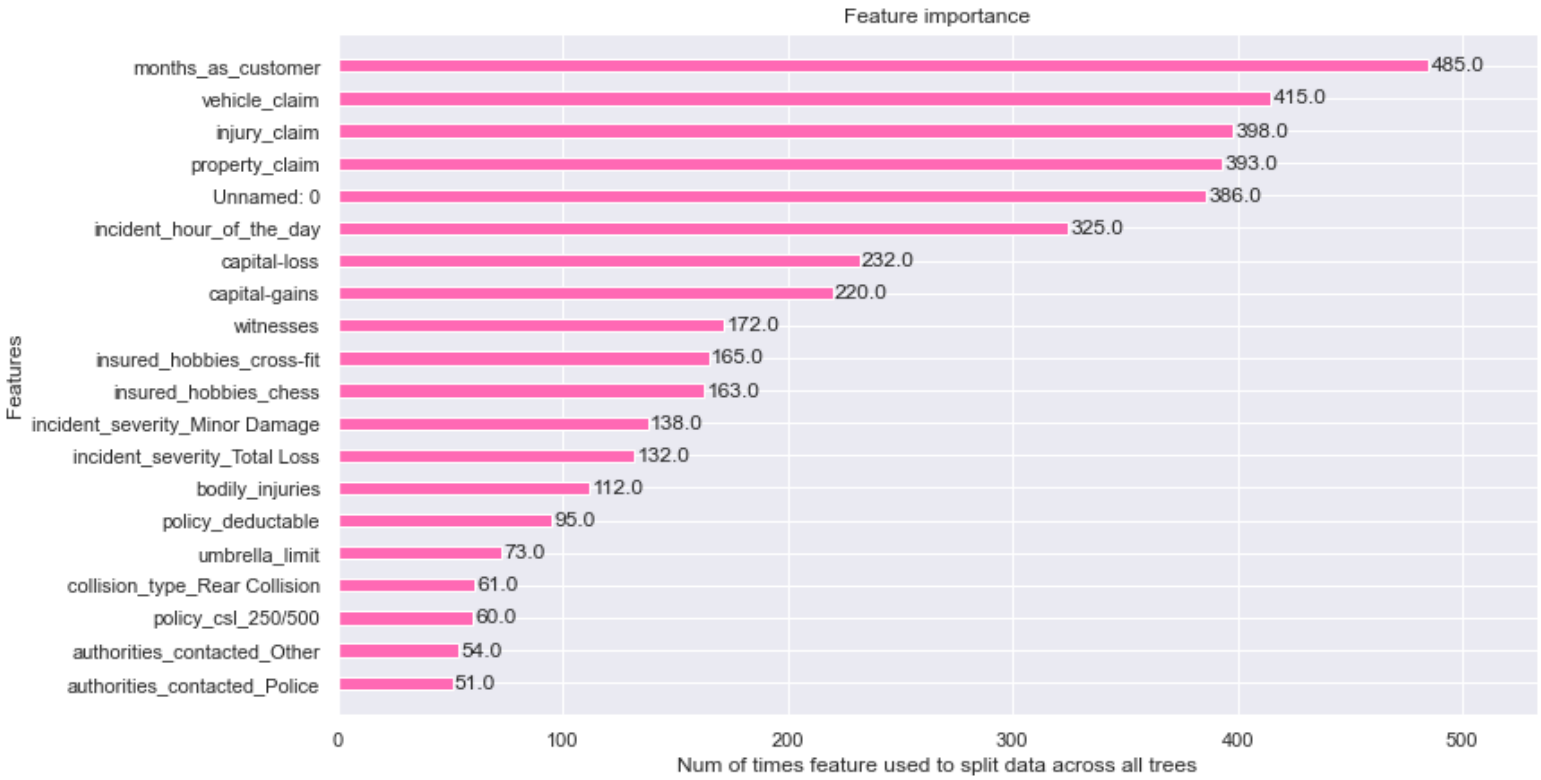
COMPARE

	Model	Score
0	Log	0.848
1	XGBoost	0.832
2	Stacking	0.828
3	Decision Tree	0.796
4	Random Forest	0.792
5	SVC	0.756
6	Baseline	0.753
7	KNN	0.712

Logistic Regression had performed the best in terms of test score.



XGBOOST FEATURE IMPORTANCE

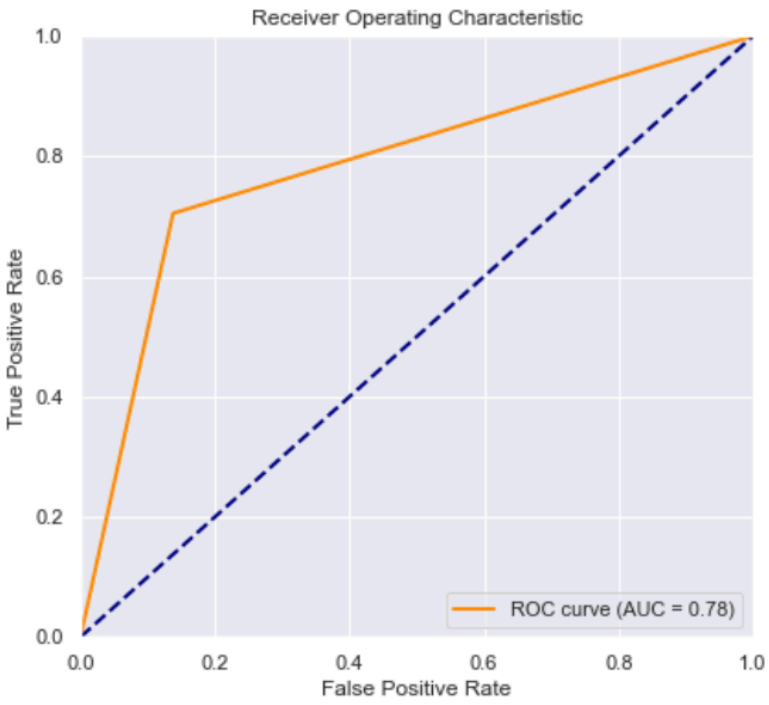


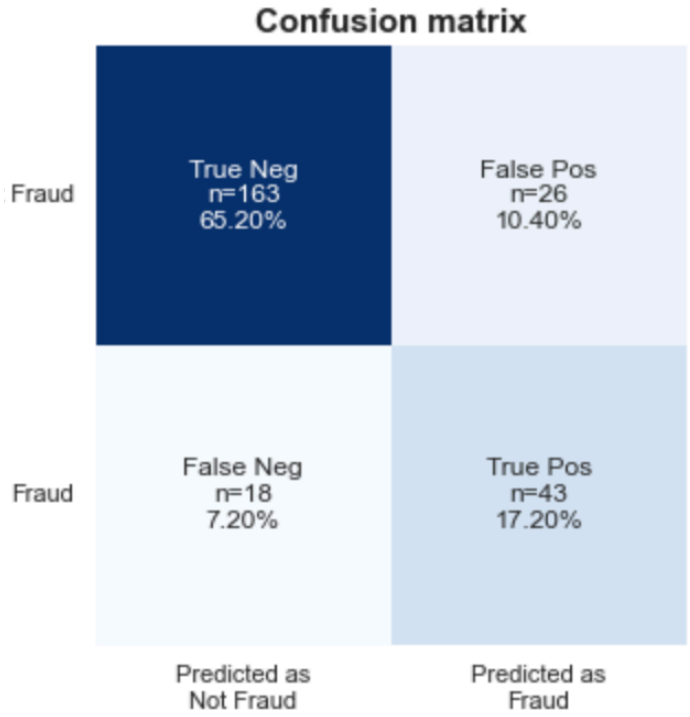
SMOTE

LOGISTIC REGRESSION SMOTE

Model Grid search: LogisticRegression(C=0.09,penalty='l1', solver='liblinear')

```
Training Model Score:  0.8643617021276596
Accuracy is:  0.824
              precision    recall  f1-score   support

           0       0.90      0.86      0.88       189
           1       0.62      0.70      0.66        61

    accuracy                           0.82       250
   macro avg       0.76      0.78      0.77       250
weighted avg       0.83      0.82      0.83       250
```

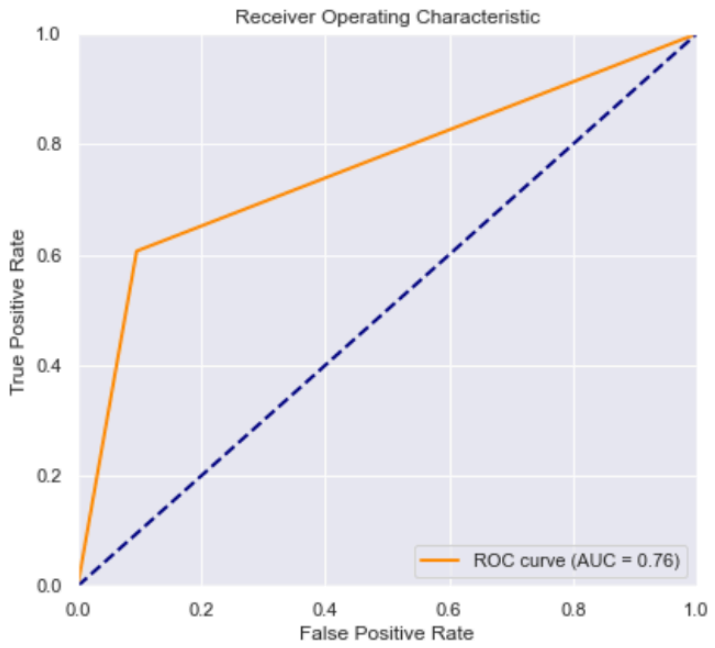


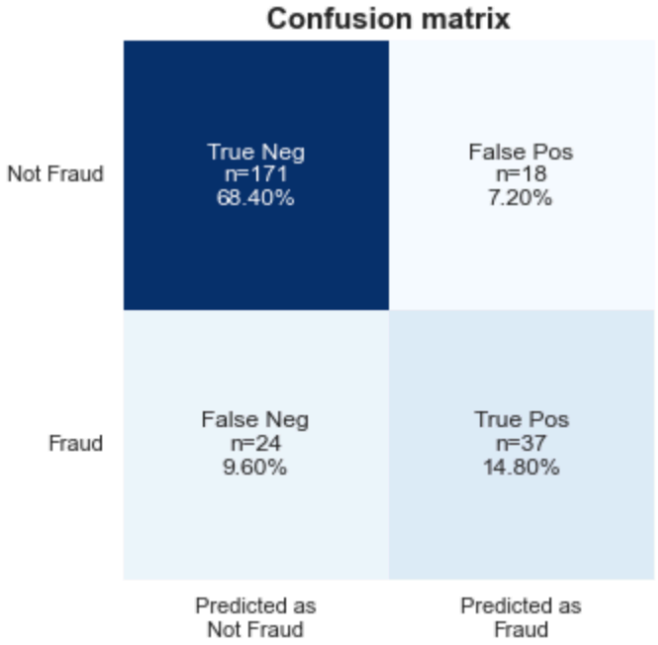


SMOTE XGBOOST

Model: XGBClassifier()

```
Training Model Score:  1.0
Accuracy is:  0.832
              precision    recall  f1-score   support

           0       0.88      0.90      0.89       189
           1       0.67      0.61      0.64        61

    accuracy                           0.83       250
   macro avg       0.77      0.76      0.76       250
weighted avg       0.83      0.83      0.83       250
```



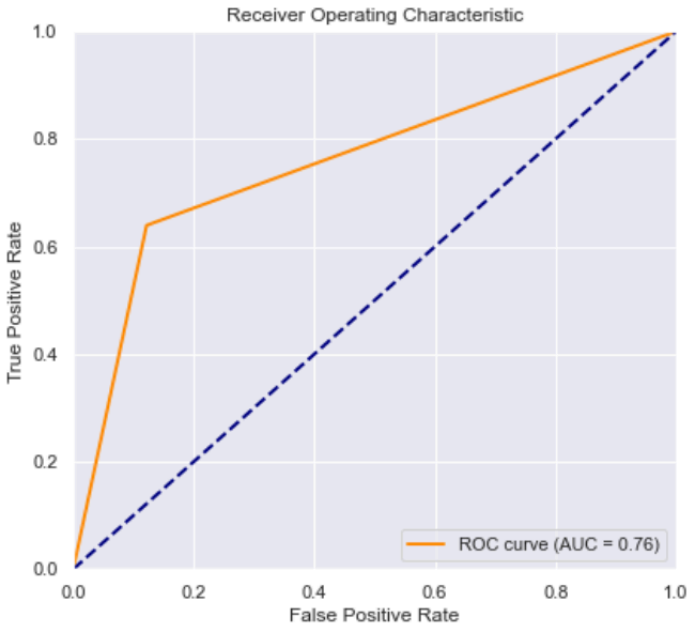


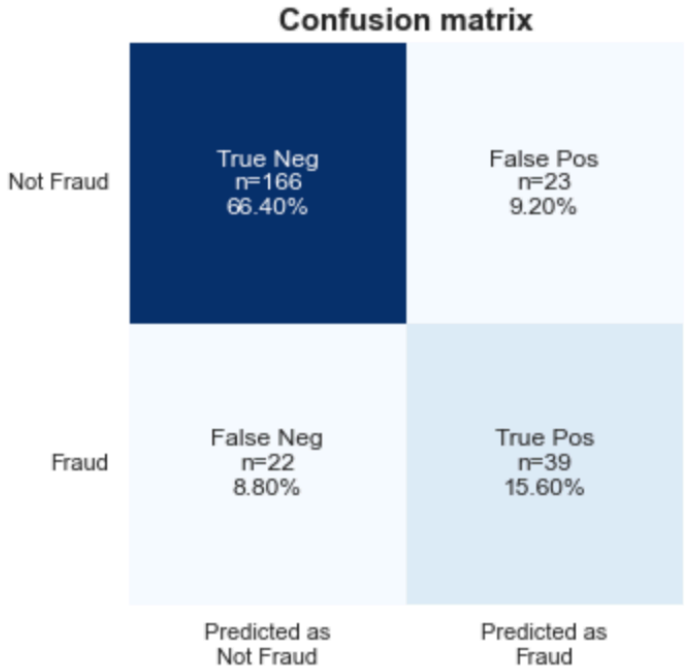
FEATURE IMPORTANCE

SMOTE DECISION TREE

Model: DecisionTreeClassifier()

```
Training Model Score:  1.0
Accuracy is:  0.848
              precision    recall  f1-score   support

           0       0.90      0.90      0.90       189
           1       0.69      0.67      0.68        61

    accuracy                           0.85       250
   macro avg       0.80      0.79      0.79       250
weighted avg       0.85      0.85      0.85       250
```

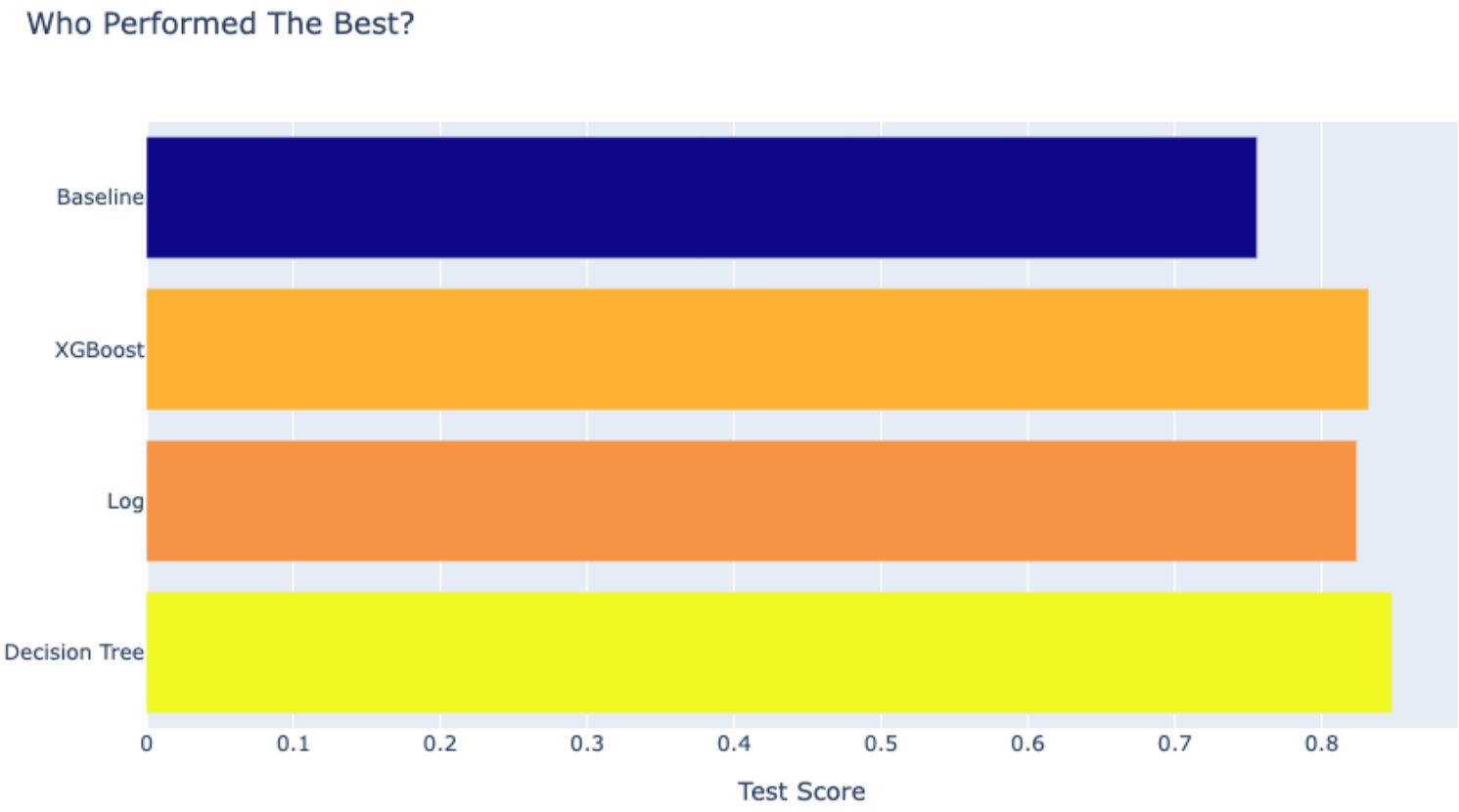




COMPARE MODEL

	Model	Score
0	Decision Tree	0.848

	Model	Score
2	XGBoost	0.832
1	Log	0.824
3	Baseline	0.756



Decision tree had performed the best out of the SMOTE Results

BEST MODEL

The best model is the Decision Tree with SMOTE over fitting of 50/50.

It had a test score of 84.8% accuracy [0.848]

AUC: 0.76

Would apply this model for the company.

Business Application

Can we predict which claims will be reported as fraud?

Yes we are able to predict which claims would be reported as fraud. It isn't the best and needs further tuning to be able to get the most out of the data. The model I would apply is SMOTE Over fitting [50/50] with decision tree to predict the companies up and coming claims.

What trends does the data show that leads to fraud?

- **Red Flags**

From the EDA we have seen a few features that do have weight behind them. If we have a better look at these features. The XGBoost feature importance did also highlight the categories that we had covered too.

We can highlight these aspects during the claims process or even the underwriting process. This could prevent fraud from even occurring.

Can we minimize the loss made from fraud?

We should be able to minimize fraud loss if we apply this model during the claims process. If we can flag up potential fraud cases have a further look into them this would stop the claims from being paid out.

BUSINESS IMPLEMENTATION

If a company applied this model during the initial stages when a claim was submitted, They could make the triage process more streamlined and efficient.

They could make a ‘Quick Claim’ category where claims that aren't flagged/low risk are passed off for approval. Reducing the amount of time spent on low risk claims.

Flag up risky events like Major damage, missing police report for senior assessors to have a look into.

This would help with distributing the claims to the appropriate people for example the high risk claims will be sent to highly experienced claims officers.

During the underwriting process we could have a better look at hobbies like chess and CrossFit - potential deny additional customers that participate in this.

Conclusion

We can utilise Machine Learning in multiple ways. As we can see we can detect suspicious patterns and trends that other fraudsters have already tried.

Be able to increase efficiency and utilise the extra time on things that need to be focused on.

This data set had some limitation as it had small sample size, statistical models become better as the sample size gets larger.

For future studies I would like a larger sample size. I would also utilise less models and focus on the imbalance of the data set. Potentially try ADASTN and Bootstrap method.

Reference

1. https://www.investopedia.com/terms/i/insurance.asp
2. https://www.softwaretestinghelp.com/basics-of-insurance-domain-for-testers/#5_Auto_Insurance
3. https://www.icnz.org.nz/industry-leadership/fraud
4. https://www.forbes.com/advisor/car-insurance/umbrella-insurance/
5. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html
7. https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/
8. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
9. *https://www.geeksforgeeks.org/decision-tree/*
10. https://www.ibm.com/cloud/learn/random-forest

11. *https://vitalflux.com/svm-classifier-scikit-learn-code-examples/#:~:text=SVC%2C%20or%20Support%20Vector%20Classifier,the%20data%20into%20two%20classes.*

12. https://www.nvidia.com/en-us/glossary/data-science/xgboost/

13.] *https://www.javatpoint.com/stacking-in-machine-learning#:~:text=Stacking%20is%20one%20of%20the,new%20model%20with%20improved%20performance*