Vehicle Insurance Fraud

By Tham Xiang Cong

5 September 2022

Problem Statement

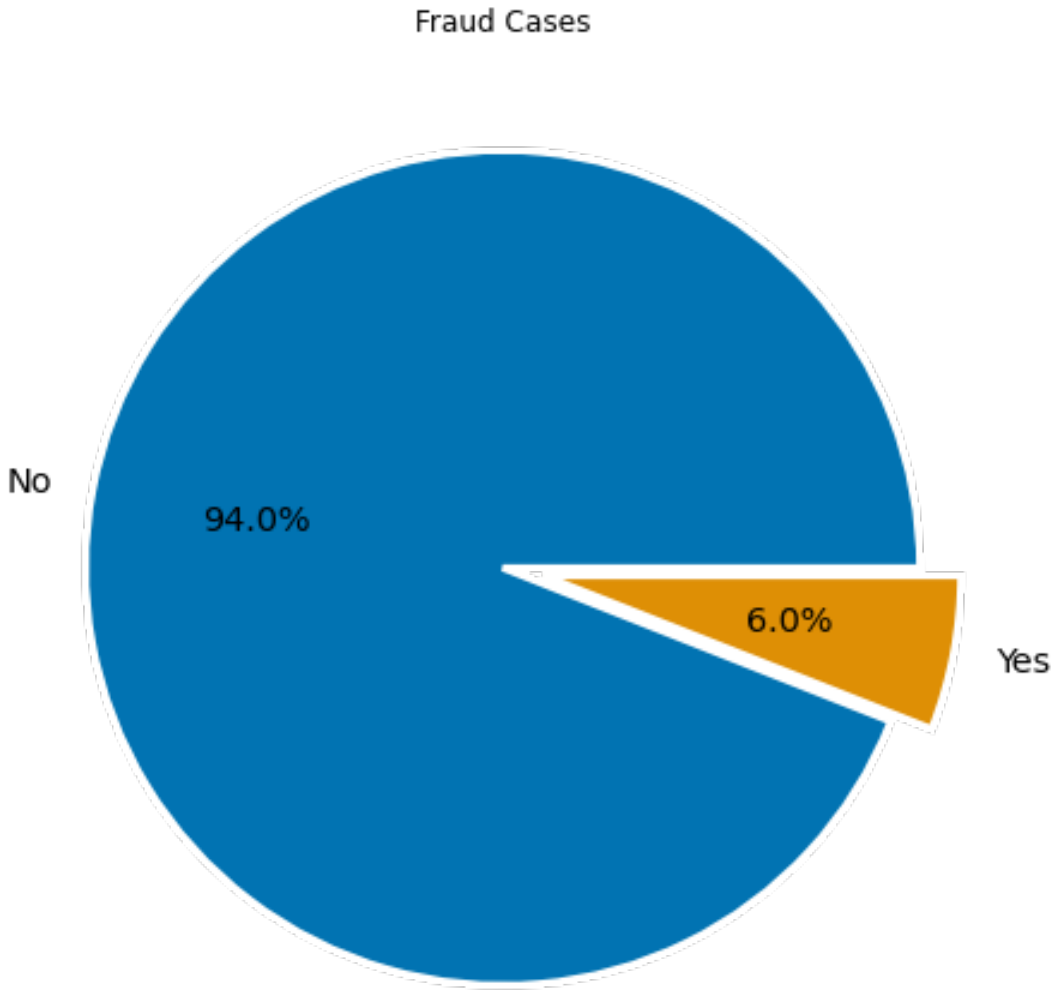
- An insurance company notices that it is losing money due to fraudulent claims.
- This insurance company that I am working with, has assigned me a task to create a predictive model to detect fraud cases among vehicle insurance claims.

Dataset given to me

- 15420 entries in total.
- 33 variables.

```
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  PolicyNumber          15420 non-null  int64
 16  RepNumber             15420 non-null  int64
 17  Deductible            15420 non-null  int64
 18  DriverRating          15420 non-null  int64
 19  Days:Policy-Accident  15420 non-null  object
 20  Days:Policy-Claim     15420 non-null  object
 21  PastNumberOfClaims    15420 non-null  object
 22  AgeOfVehicle          15420 non-null  object
 23  AgeOfPolicyHolder     15420 non-null  object
 24  PoliceReportFiled     15420 non-null  object
 25  WitnessPresent        15420 non-null  object
 26  AgentType             15420 non-null  object
 27  NumberOfSuppliments   15420 non-null  object
 28  AddressChange-Claim   15420 non-null  object
 29  NumberOfCars          15420 non-null  object
 30  Year                  15420 non-null  int64
 31  BasePolicy            15420 non-null  object
 32  FraudFound            15420 non-null  object
```

Proportion of fraud cases



- Only 6% of all claims were fraud cases.
- There is an imbalance in the data (there are very much more no-fraud cases compared to fraud cases).

Checking of relationships between variables

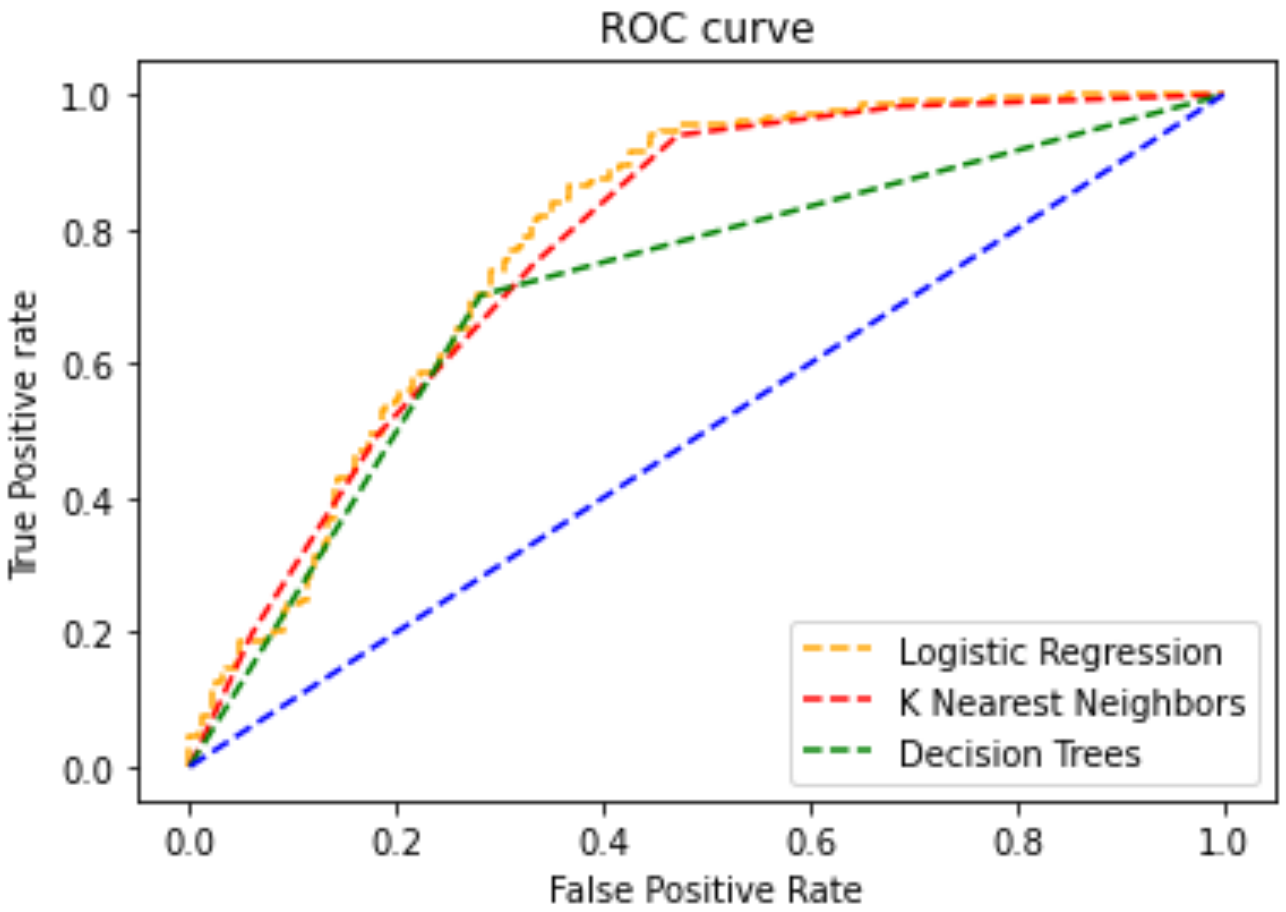
AgeOfVehicle	AgeOfPolicyHolder	0.743915
AgeOfPolicyHolder	AgeOfVehicle	0.743915
BasePolicy	PolicyType	0.785248
PolicyType	BasePolicy	0.785248
Month	MonthClaimed	0.833637
MonthClaimed	Month	0.833637
AgeOfPolicyHolder	Age	0.848918
Age	AgeOfPolicyHolder	0.848918

- The above variables have strong relationships with each other (r value > 0.7).
- There are no variables that have strong inverse relationship (r value less than -0.7).

3 classification models used

	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.94	0.00	0.00	0.00
K Nearest Neighbors	0.94	0.14	0.02	0.04
Decision Trees	0.89	0.17	0.27	0.21

- The accuracy of the three models are very close to 0.9, which indicates perfect models.
- Logistic regression has a value of zero for precision, recall and F1 score.
- For other two models, the values are very close to zero.
- It is very likely due to the imbalance of data as mentioned earlier.



- Area under curve score is:

- Logistic Regression: 0.77
- K Nearest Neighbors: 0.67
- Decision Trees: 0.60

- From the ROC curve, Logistic Regression is the best model.

Due to imbalance of data, I resampled the data

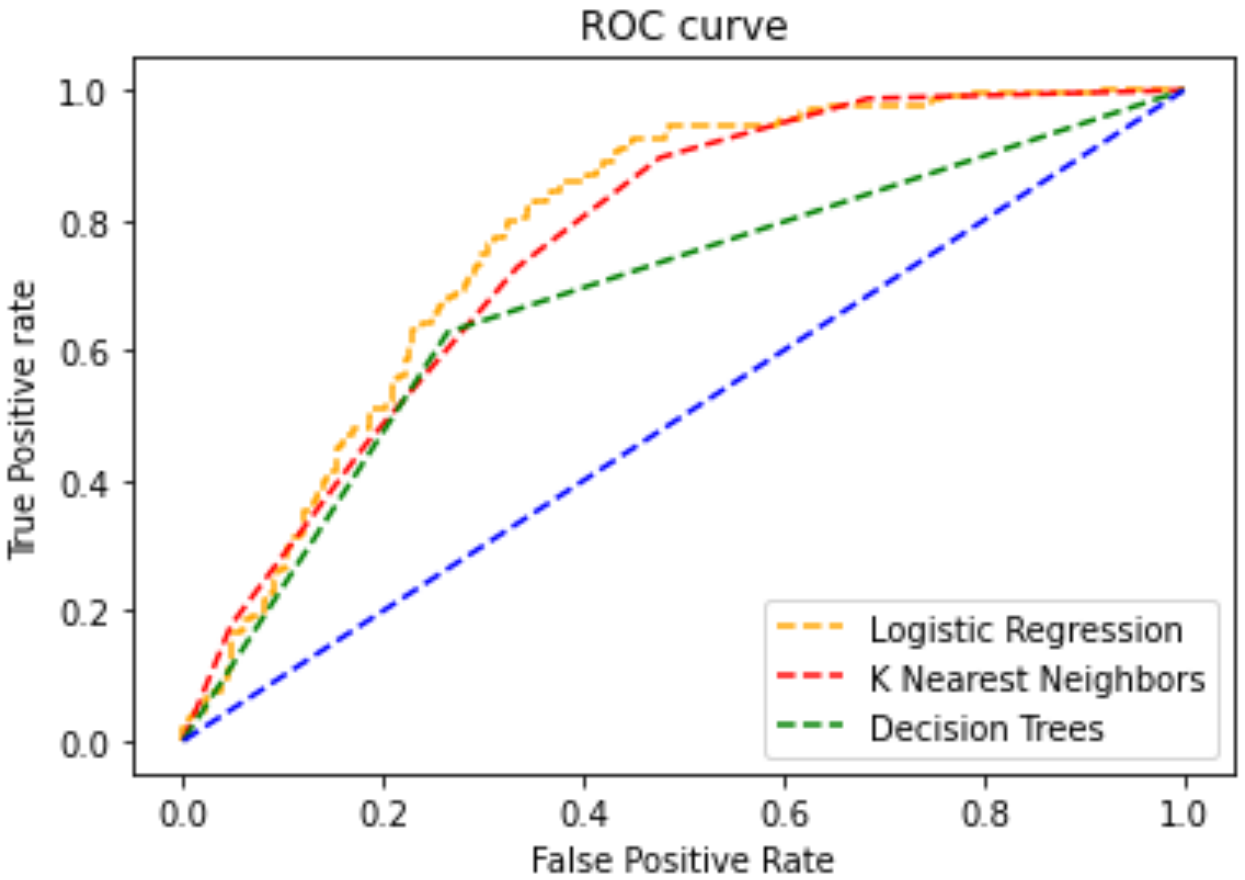
```
original dataset shape: Counter({0: 14496, 1: 923})
Resample dataset shape Counter({0: 923, 1: 923})

0       0
1       0
2       0
3       0
4       0
       ..
1841    1
1842    1
1843    1
1844    1
1845    1
Name: FraudFound, Length: 1846, dtype: int64
```

- After resampling (undersampling), there are equal number of fraud cases and no-fraud cases.
- 0 → No fraud cases (n=923)
- 1 → Fraud cases (n=923)

3 classification models refitted

	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.71	0.66	0.80	0.72
K Nearest Neighbors	0.70	0.68	0.73	0.70
Decision Trees	0.73	0.73	0.71	0.72

- The accuracy of the three models is very close to 0.7.
- Decision Trees model has the highest precision score.
- Logistic Regression model has the highest recall score.
- The F1 scores of the three models are very close to 0.7.
- False negatives in this case, have high cost to the company. Hence, we will look at ‘recall’.



- Area under curve score is:

- Logistic Regression: 0.78
- K Nearest Neighbors: 0.76
- Decision Trees: 0.68

- From the ROC curve, Logistic Regression is still the best model.

Hyperparameters Tuning of Logistic Regression

- Parameters used: parameters = {'penalty':['l1', 'l2'], 'C' : np.logspace(-3,3,7), 'solver':['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga’]}
- Grid Search Technique is used for hypertuning
- Best parameters = {'C': 1.0, 'penalty': 'l1', 'solver': 'liblinear’}
- Best estimators = {penalty='l1', solver='liblinear'

Comparison of hypertuned and pre-hypertuned models

	Accuracy	Recall	AUC
Hypertuned	0.73	0.86	0.80
Pre-hypertuned	0.71	0.80	0.78

- Overall, not much difference between the two models. Pre-hypertuned model can still be used as a predictive model.

Implementation of the model

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
0	0.1533	0.3327	0.4609	0.6449	-0.4988	0.8055
1	-0.2160	0.1765	-1.2241	0.2209	-0.5620	0.1299
2	0.2169	0.1659	1.3073	0.1911	-0.1083	0.5421
3	0.0139	0.4772	0.0292	0.9767	-0.9213	0.9492
4	0.0462	0.1656	0.2791	0.7801	-0.2783	0.3708
5	0.2329	0.2315	1.0062	0.3143	-0.2208	0.6866
6	-0.0160	0.3377	-0.0473	0.9622	-0.6778	0.6458
7	0.0762	0.1828	0.4167	0.6769	-0.2821	0.4344
8	-0.4205	0.1635	-2.5709	0.0101	-0.7410	-0.0999
9	-0.3363	0.3926	-0.8566	0.3917	-1.1057	0.4332
10	-3.0194	0.2096	-14.4070	0.0000	-3.4301	-2.6086
11	9.3987	0.9747	9.6424	0.0000	7.4883	11.3092
12	-4.3588	0.2996	-14.5491	0.0000	-4.9460	-3.7716
13	0.4692	0.2346	2.0001	0.0455	0.0094	0.9289
14	0.6032	0.4842	1.2457	0.2129	-0.3459	1.5522
15	0.0189	0.1515	0.1247	0.9007	-0.2780	0.3158
16	-2.4042	1.0723	-2.2420	0.0250	-4.5059	-0.3024
17	2.8724	1.1016	2.6076	0.0091	0.7134	5.0314
18	-0.1530	0.1778	-0.8606	0.3894	-0.5014	0.1954
19	-0.1969	0.4298	-0.4580	0.6470	-1.0393	0.6456
20	-0.3206	0.4845	-0.6618	0.5081	-1.2701	0.6289
21	-0.3247	0.4011	-0.8096	0.4182	-1.1108	0.4614
22	-0.0217	1.0182	-0.0213	0.9830	-2.0173	1.9740
23	-0.6514	0.6334	-1.0284	0.3038	-1.8927	0.5900
24	-0.0608	0.1424	-0.4268	0.6695	-0.3398	0.2183
25	1.1338	0.2727	4.1572	0.0000	0.5993	1.6684
26	-0.8403	0.5805	-1.4476	0.1477	-1.9781	0.2974
27	-0.1669	0.1421	-1.1744	0.2402	-0.4454	0.1116
28	-1.5922	0.2597	-6.1319	0.0000	-2.1011	-1.0833

- 'Sex', 'Fault', 'PolicyType', 'VehicleCategory', 'VehiclePrice', 'Days:Policy-Accident', 'Days:Policy-Claim', 'AddressChange-Claim', 'BasePolicy' are predictive variables in the model. (*P-value* < 0.05)

Limitations

- The logistic regression model is built on all X-variables.
- However, from the logit table, it is noted that the majority of the variables are not predictive variables based on *P-value > 0.05.*
- Unnecessary variables in the model can increase complexity, resulting in suboptimal performances.
- Suggested solution: Use backward elimination method-removal of variables which are *P-value > 0.05* from the model.

Conclusion

- Logistic Regression Model is the best predictive model.
- It has the highest recall score and highest AUC score.
- Hyperparameters tuning makes no significant difference to the model.
- 'Sex', 'Fault', 'PolicyType', 'VehicleCategory', 'VehiclePrice', 'Days:Policy-Accident', 'Days:Policy-Claim', 'AddressChange-Claim', 'BasePolicy' are predictive variables in the model.