Insurance fraud detection

1. Introduction

Fraud has been plaguing the insurance industry since the beginning of its existence. False claims incur substantial loss for the insurance providers, and they drive up the costs of insurance products for everyone. On the one hand, fraud detection itself incurs costs. If the company develops an accurate fraud detection system but the implementation is time-consuming, expansive or involving the breach of confidentiality of data, it is not of practical use. On the other hand, failure to recognize any frauds is clearly unadvisable. Therefore, investigating an effective model and fraud detection techniques is required to improve the quality of the service and minimize the unnecessary costs. This problem falls into the category of classification in the realm of machine learning.

then you need to briefly mention the methods

2. Expanded literature review

Due to lack of experience and domain knowledge, we realize a comprehensive literature review on fraud detection projects is necessary. In order to get inspired on some similar ideas we read though a dozen of academic papers and we discover the followings are most related. Yi Peng etc. [1] introduced three predictive models: Naïve Bayes, decision tree and Multiple Criteria Linear Programming to be trained, they gave out the test results to compare the accuracy and also proposed some suggestions for future projects on frauds. Capelleveen etc [2] provided the outlier method of data mining technology for the health insurance fraud detection. This is also used for detecting the suspicious behavior of medical service providers. Zhenxing Hou [3] proposed a fraud risk analysis according to cluster analysis for isolation by distance clustering method. Clifton Phua etc [4] conducted a research survey which explored almost all published fraud detection studies and gave a comprehensive overview of different types of fraud, the methods and techniques people used and their limitations. They indicated unsupervised, semi-supervised and text mining from law enforcement approaches for different types of data.

Jing Li [7] conducted a detailed survey on statistical methods for fraud detection in health care data area. This survey classified the behaviors of fraud cases, identified the sources on which fraud detection has been conducted, provided crucial steps in data preprocessing, compared statistical methods that currently in use, and provided the advice on future directions. Thorton et [8] indicated a multidimensional data model and analyzed important approaches to help predict fraudulent cases. In addition, Weizong Zhang [9] has conducted the single-factor and two-factor analysis by SAS software

according to the application of Logist regression model to classify the considerable factors in fraudulent claims. These techniques that have been mentioned above are quite effective and are good references for our own project.

Additionally, Rafael et [10] evaluated the behavior and influence of feature selection methods, performed undersampling strategy to improve the performance and used real data to check the results. The model achieved high efficiency by reducing the number of features. Qi Liu [5] introduced fraudulent behaviors in health care system and analyzed characteristics of dataset, compared and reviewed some existing fraud detection approaches. He also proposed a clustering model that contains information of geo-location to identify dubious claims. Furthermore, Ayhan Demiriz [6] evaluated the value of geographical information for deriving business rules to detect and prevent financial frauds and scams in his paper. These papers inspire us that location information could become one of the significant features.

In general, the papers suggest and prove that the machine learning techniques are effective and beneficial in detecting fraud activities. They are informative guide and inspiration for our following research.

3. Material and methods

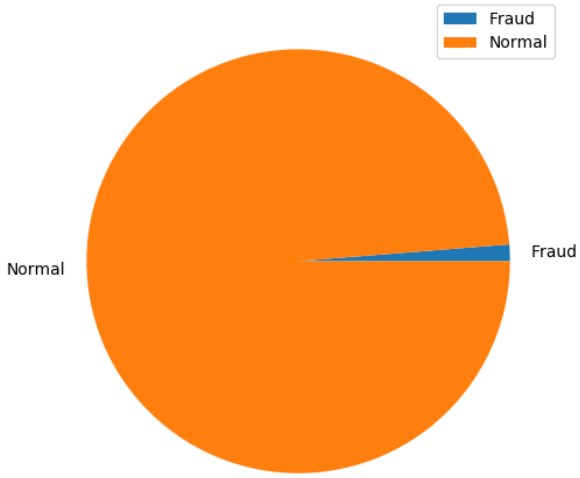
3.1 Data Description

Yuumi Insurance provided the log data that records the interaction between the customers and the company. The dataset contains more than 3 million records which enhances the level of difficulty. Each piece of information is structured, timestamped and describes a specific activity of a customer, including quotes, claims and payment status. When a customer enters one of the platforms (mobile app, website or phone calls), a quote will be generated based on the client's information.

If the client chooses to accept the insurance policy and make a payment, this information is recorded as well, without the specific amount of money the client paid. When a claim is made, the company will look through the client's claim history and examine all the information related to the client and decided if they will accept or deny the claim.

3.2 Data Cleaning

The log data provided by the insurance company has 2 components, the message and the time stamp. Each piece of data has an activity type, including but not limited to: "Quote Started", "Payment Completed", "Claim Started", "Claim Accepted" and "Claim Denied". In reality a customer's claim could be denied for many reasons, but in this dataset "Claim Denied" are precisely the cases that are identified as frauds. The labels in this data is really imbalanced.



The data is in the log form and therefore should go through data cleaning, data transformation and data curation processes to be in tabular form, on which we build a classifer of fraudulant activities. Challenges encountered with this data include:

- Not all data are relevant. We split the original data according to the activity type. "Quote Completed", "Claim Started", "Claim Accepted" and "Claim Denied" are the most important ones and are examined individually.
- There are several different delimer seperating fields in each piece of data. For example, the case ID, platform and activity type a
  These two sub-problems describe two ends of a specturm. In reality, the situaion would fall somewhere in between, depending on how quicky the companies' policies expire. The real-time fraud-detector would have to deal with both cases, the new customers as well as current ones. re separated by "-", whereas the time stamp and the rest of the information is separated by ",".
- Due to confidentiality concerns, the insurance company's clients' information are transfromed and substituted by pseudo names. No meaningful conclusion can be drawn from these names and addresses directly, but the distribution of these data are kept intact.
- One customer ID could correspond to different payload, in which they submit different information about their households and homes. One customer ID could also submit multiple claims and there is no given link between a "Claim Started" and "Claim Denied" or "Claim Accepted" cases. We will process the data on the assumption that each "Claim Started" case corresponds to the most recent submitted payload from "Quote Completed". The customers with more than one payload and those who submitted more than one claims only constitute a small proportion of all data.

We split the dataset by activity types and label each incoming case. The incoming cases are sorted according to their time stamps. Each incoming case has features attached to it, which are used to predict the label. Other than "platform" (the platform the claimer used), the majority of the features are extracted from the "Quote Completed" data, where the customers submitted Json payload describing their demographic and household information.

We extracted an important feature not in the original data, which is "matched_time_stamp". From our observations, many claims were started right after the customer made the payment, mostly the fraudulent ones. The time stamps are exactly the same. "matched_time_stamp" simply checks if there is a "Payment Completed" activity associated with the same customer ID, at the exact same time a "Claim Started" activity happened. Our analysis validates that all of the frauds have this feature equalling "True", but not the other way around. There are still some cases with matched time stamps but are not labelled as "Fraud".

3.2 Definition of Problems

Since the time-span of this dataset is not very long, most claims are made by existing customers(those already in the company's database). In the short run, these claims are prevelant as the customers' insurance policies hasn't expired yet ,with new customers coming in slowly. However, in the long run, customers will come and go, and claims made by new incoming customers deserve special attention, as their claim history is unknown. Therefore, this reprot aims to solve 2 subproblems in the context of insurance fraud deteciton:

1. Given all the predictors including the claim history of each customer, predict as accurately as possible whether or not an incoming claim is fradulent.
2. Predict whether or not a new customer is fradulent. (A customer is said to be fradulent if at some point he or she makes false claims.)
3. Calculate the estimated amount of coverage saved with these classifiers.

The first two sub-problems describe two ends of a specturm. In reality, the situaion would fall somewhere in between, depending on how quicky the companies' policies expire. The real-time fraud-detector would have to deal with both cases, the new customers as well as current ones.

3.2 Assumptions

The following assumptions are made on this dataset:

1. One customer ID can only represent one customer in real life. In the dataset, one customer ID could correspond to a lot of activities. One customer ID could even have different Json payloads (with claimer's name, age, gender and other basic information). In reality, most companies would

not reuse its customer IDs as this creates unnecessary anomolies and inconsistencies. Therefore, in our analysis, each customer ID is treated as an individual customer.
2. For the majority of the customers, their identities do not change over time. We assume that most normal customers would never start any fraudulent activities, at least not under the same customer ID. We also assume that most frauds would not turn from their wrongdoings and start making legitimate claims, at least not under the same ID. This assumption can be validated through the following analysis:

Based on assumption 2, if an existing customer makes a new claim, the claim history tells us a lot about whether the incoming claim is fraud or not. We define an additional feature for each incoming claim case, "number of denials". For each claim case along with a time stamp, this feature shows how many times a customer's claims got denied by that time stamp. For example, if a person got denied 5 times (although this is unlikely in the dataset), the "number of denials" for the 5 incoming cases should be 0,1,2,3,4, indicating how many times the person were denied already.

3.3 Methods

A classification model is appropriate for this dataset since the incoming cases have 2 labels, fraud or normal. Based on the original features and 2 extracted ones, classifiers are built to predicts the label of incoming claims. An approach widely adpoted is to split the data into training, testing and validation sets. The training set is used to build the classifier and the validation set will evaluate the performance of the model. Looking at performance metrics of the classifier on the validation set can help identify the most relevant features and combination of parameters of the classifier. After obtaining the most important features and the optimal combination of parameters, a final model is built on training and validation sets, and evaluated on the test set.

Often the testing set is set aside in advance, and cross validation technique is applied to the rest of the data. The remainder of data is split into K even shares. Each share is used as the validation set in each iteration.

However, cross validation cannot be applied when solving subproblem 1 (claim history considered and data sorted by time). The reason is two-fold:

1. Every case is time-stamped. Cross validation means at some point a classifier using data in the future is built to predict labels in the past. There could be some information leak from the future to the past that makes the classifier perform better than it should be. For example, if some feature is dependent on past labels, then the value of such features in the future already include the information about labels in the past, which is precisely the case with a extracted feature in our model, "number of denials".
2. Even if there is no risk of information leak when doing cross validation, we cannot make the assumption that time does not play a part in the pattern in which fraudulent activities occur. No

prior knowledge or evidence is available to make such assumption.

If parameter-tuning is needed for modelling in subproblem 1, the predefined performance metrics can be calculated on different split ratios.

When considering subproblem 2, the claim history of each customer is not considered. Based on assumption 3, it does not matter when does a customer come into the database. As time is taken out of the equation, cross validation can be used for parameter-tuning of models, evaluation of models and selection of features. The abovementioned procedures to train a classification model are done in the solution of subproblem 2.

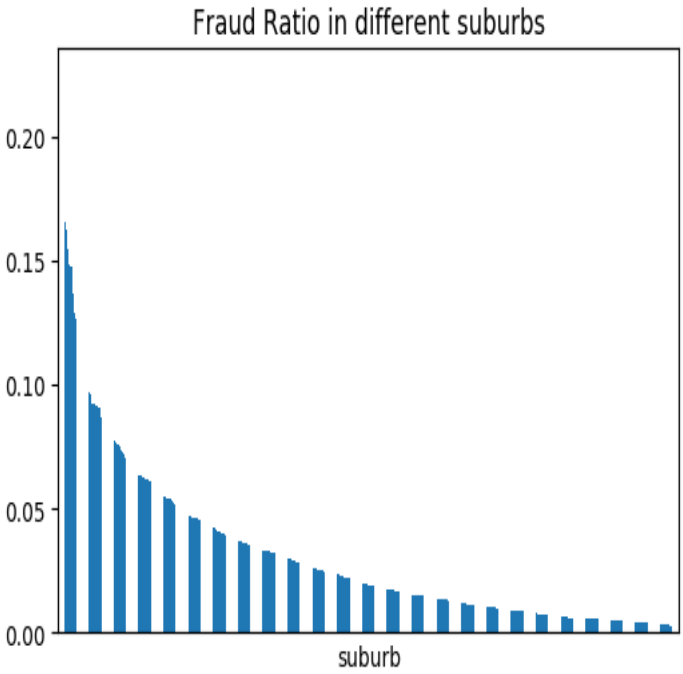
As for the classification algorithm, Decision tree and Random-Forest are adpoted because it's convenient to conduct feature selection with them. Random-Forest is basically a collection of randomly generated decision trees. Decision Tree are non-parametric supervised learning algorithm. It doesn't require much preprocessing of the data and deals well with missing values. A decision tree is a set of rules that split the dataset into different subsets(called leaves). Each time the tree splits the data in current node by one of the attributes. The tree does not stop splitting the dataset unless prior restriction is given and met. Conventionally, decision trees are grown and then pruned to a smaller depth to avoid overfitting. Despite the "max depth", "max number features" or "min leaf node" constraints set to overcoming overfitting in decision trees, one single tree tends to bring in too much bias into the model. As Random-forest takes the average "opinion" of many randomly generate trees, its predictions have less bias, lower variance and more consistent.

4. Analyses and Results

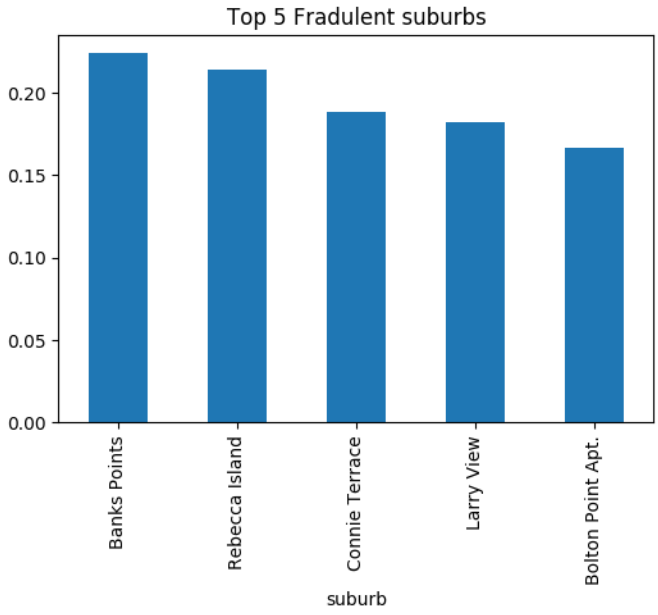
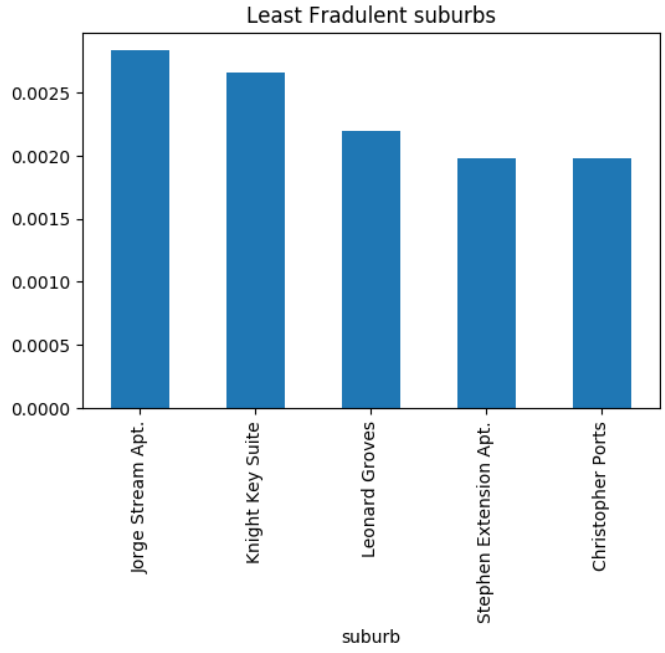
4.1 Feature Engineering

Explorative data analyis is conducted on the features. For categorical variables, the data is by the variable and check if the proportions of fraudulent claims are similar. "Fraud ratio" illustates what percentage of claims in one category are frauds. If across different categories the "fraud ratio" varies dramatically, then it is probable that such variable is relevant to fraud detection.

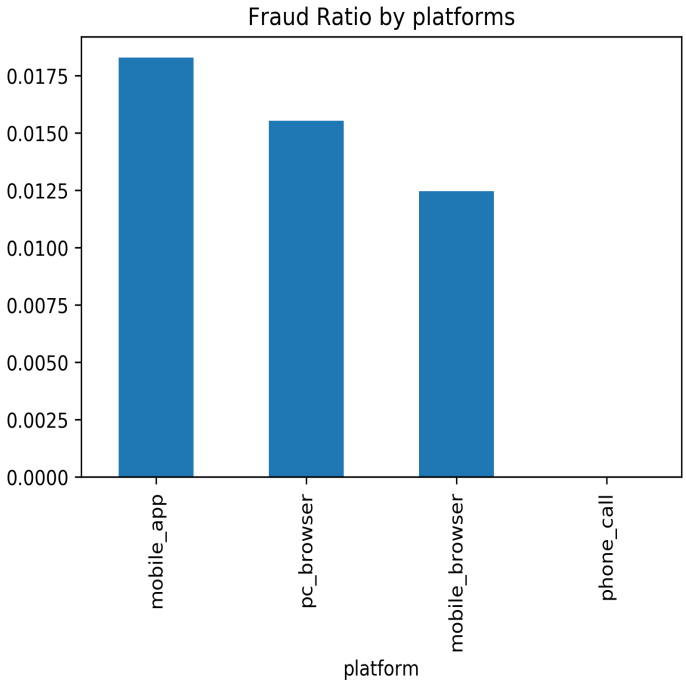
The first variable examined is suburb, as related literature has reported that this is generally relevant to fraud detection.



The above plot shows that fraud ratio vary a lot across different suburbs. In some suburbs, as many as 20% of the claims are fraudulent, whereas in the other suburbs almost all claims are legitimate. The distinction is furthere depicted in the following tww graphs. The left list the suburbs with the highest fraud percentages and the right list the suburbs with the lowest fraud ratios.

1	2
	

The variable "platform" refers to the platform the customer used to make a claim to the insurance company. Platforms include "Mobile APP", "PC Browser", "Mobile Browser" and "Phone Call".



As shown above, the fraud ratios across different platforms are not alike. A noteble point is that there is no fraudulent claims via phone calls. All of the claims made via phone calls are legitimate. It would be beneficial to include "platform" in the model.

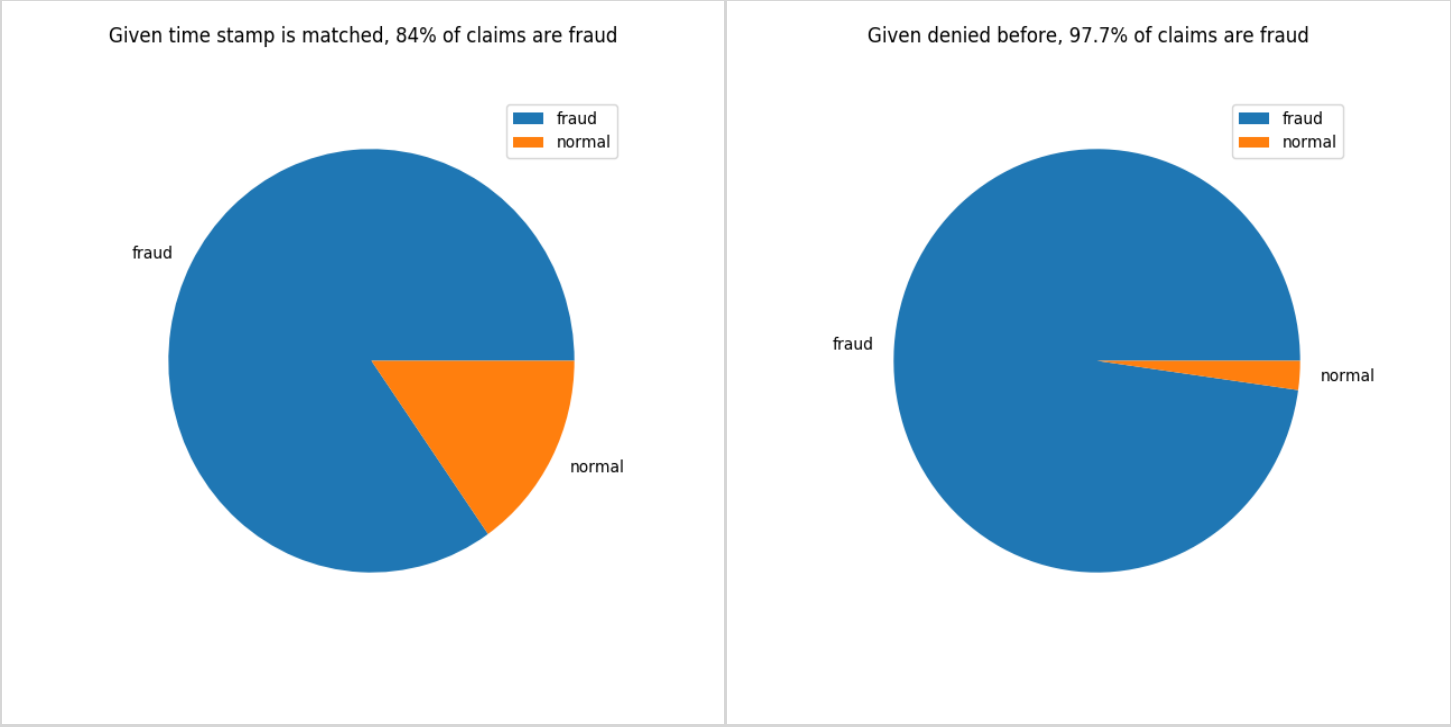
4.2 Modelling and Results

4.2.1 Subproblem 1

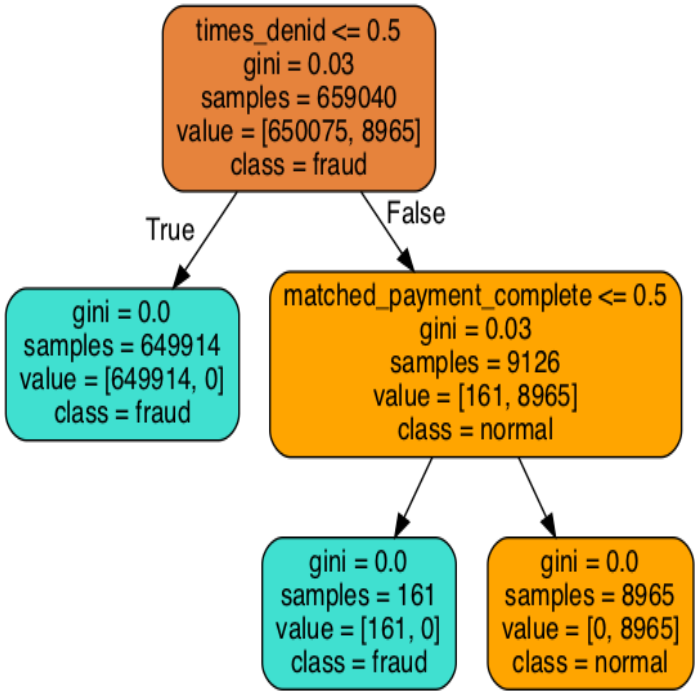
In this part we fit a decision tree to classify the incoming claims. The following pie charts demonstrate the conditional probabilities that are relevant to our model.

$$
\begin{aligned}
P(fraud|matched\ time\ stamp) &= \frac{P(fraud\ \&matched\ time\ stamp)}{P(matched\ time\ stamp)} \\
&= \frac{|fraud\ \&matched\ time\ stamp|/|all\ cases|}{|matched\ time\ stamp|/|all\ cases|} \\
&= \frac{|fraud\ \&matched\ time\ stamp|}{|matched\ time\ stamp|} \\
&= 84\%
\end{aligned}
$$

$$
\begin{aligned}
P(fraud|\ denied\ before) &= \frac{P(fraud\ \&\ denied\ before)}{P(denied\ before)} \\
&= \frac{|fraud\ \&\ denied\ before|/|all\ cases|}{|denied\ before|/|all\ cases|} \\
&= \frac{|fraud\ \&\ denied\ before|}{|matched\ time\ stamp|} \\
&= 97.7\%
\end{aligned}
$$



With these two predictors, we build the following decision tree. As shown in the tree below, all the leaf nodes have pure labels (either all frauds or all normal). It means that this tree is able to classify all the incoming cases correctly. This is not surpirsing and it vadlidates our assumption that a customer's identity does not change over time. The classifier can be built on as little as the first 5% of the dataset and still produce this result, because it would have enough claim history of customers to know if a claim associated with a customer ID is fraud or not. In reality, if the fraudulent activities become more complex, the classifier needs more features to make correct predictions. However, we believe this model mimics how companies process frauds in the short run: when the majority of claims are made by existing customers, looking at the claim history is the rational choice.

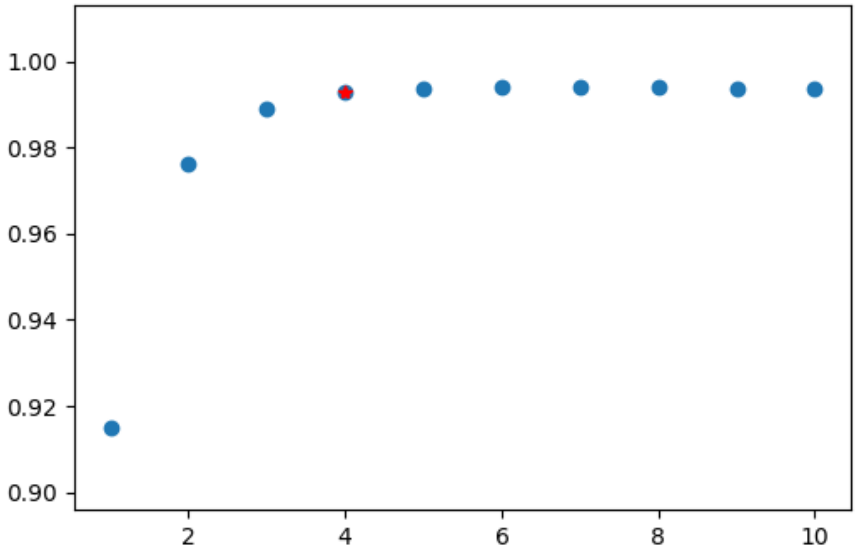


4.2.2 Subproblem 2

In this section, the claim history of each customer is not considered. We focus on giving customers correct labels instead of incoming cases. If a customer in this dataset has never made a false claim, the label "normal" is assigned. Otherwise,

In this part, a Random Forest classifier is built to predict fraudulent customers. Random Forests are less biased and can produce more consistent results. The results are equally easy to interpret. Since it is a bit more complicated than a single decision tree, the optimal parameters of a Random Forest need to be decided. First we need to find out how many decision trees we need to grow in order to effectively reduce bias. In theory the more trees we have, the more consistent the results will be. However, a model too complicated would take too long to fit and need more data.

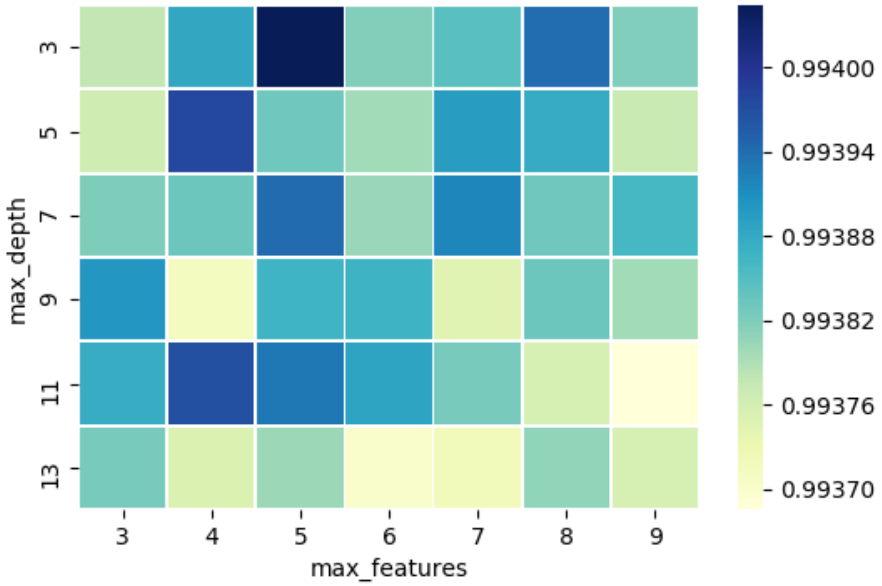
Follwing is a result from a one-dimensional grid-search. For each number of trees, we calculate the performance metric, area under ROC curve (later referred to as ROC score) from cross validation.



As shown in the graph, when we increase the number of trees to 4, the performance significantly increses. It doesn't change too much if we keep increasing the size of the forest. This suggests that having 4 trees in a random forest is enough to offset the biases.
Then we need to decide at most how many features the model need and how deep each tree can be. We need to find the best combination of parameters "max_features" and "max_depth" for this dataset. If too many features are used, some irrelevant features would be involved. If the trees on the training set are too deep, they may not fit the data in the testing set, which causes the problem of "overfitting". The following is the pseudocode that explains the grid-search for the two parameters, with area under ROC curve as the performance metric.

for $p_1, p_2 \in P_1 \times P_2$:
    split the training set into K pieces
    for each piece in the K-fold:
        Train the model on the remainder of data
        Use current piece as validation set
        Calculate ROC score
    End for
    Calculate the mean aread under ROC curve for p_1, p_2
End for

The above algorithm produces a ROC score for all possible parameter combinations in the parameter space, using K-fold cross validation (K =5). The following graph demonstates the results from grid-search.



As shown, when the "max_feature" is 5 and "max_depth" is 3, the performance metric is optimised in the restricted parameter space. Therefore, these parameters along with "n_estimator" (number of trees grown in the random forest) being 4 will be used to train the final model. The final model is built on the whole training set and performance metrics will be calculated on the test set. The actual decision trees in the random forests are attached in Appendix 1.

$$TNR = \frac{TrueNegative}{TrueNegative + FalsePositive} = 98.83\%$$

$$TPR = \frac{TruePositive}{TruePositives + FalseNegative} = 100\%$$

The True Positive Rate (TPR) and True Negative Rate (TNR) show that the final classifier does a good job identifying the fraudulent customer, as well as not misclassifying normal people.

features	importance
matched_payment_complete	0.994826
platform	0.004539
square_footage	0.000379
suburb	0.000143

The top 5 features are displayed above. Obviously, "matched_payment_complete" is the most important feature. We have shown that all the fradulent claims have this feature. The fraudulent customers' behaviour resemble the behaviour of bots, which explains why these customers can start

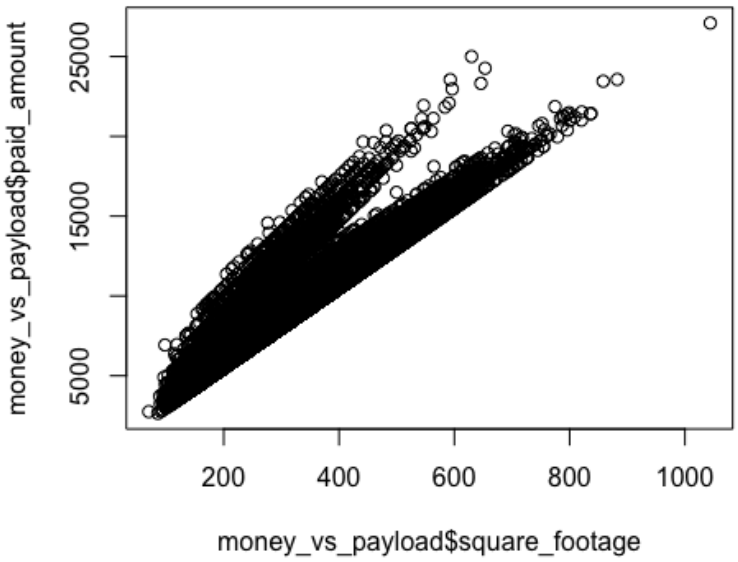
a claim at the exact same time when they completed the payment for the insurance policy. "Platform" is somewhat important because none of the fraudulent claims are made over the phone. All of the fraud claims are made on the internet.

4.2.3 Subproblem 3

We need to examine the economic implications of abovementioned classifiers. From a business point of view, the loss from a fraud detection system comes from two sources, the false positives and the false negatives. The false negatives are the frauds who got away with their wrongdoings and are paid coverage by the insurance company. The loss associated with them is the amount of money paid as coverage. The loss associated with the false positives is harder to explicitly measure. The false positives are people making normal claims, desersing to be compensated but got denied by this model. It does not incur loss to the company directly but it stains the company's reputation and credibility. In reality, this automated fraud detection model should merely serve as a initial filter. Based on the customers' intercations with the database, it produces a list of "identified frauds", which will be examined by the company's professional investigators thoroughly. This will, to the greatest extent, elimilate the loss incurred by false positives.

We attempt to estimate the potential loss generated by the false negatives. For each claim case denied, the coverage that could have been paid to the customer should the case wasn't denied is not available in this dataset. Fortunately, for each claim accepted, the coverage paid to the customer is accessible. A regression model can be built to fit the coverage on the home information ("home_type","number of bedrooms","square_footage") provided. Then we can estimate the amount of money that could have been paid to a fradulent customer, if he or she is misclassified as a normal customer.

Some explorative analysis suggests that square footage of a home is the only signiifcant predictor of coverage. Below is a scatterplot of coverage vs square footage of the house, alongwith he R output of the regression model. A linear relationship is clearly displayed.




```
Call:
lm(formula = paid_amount ~ square_footage, data = money_vs_payload)

Residuals:
   Min     1Q Median     3Q    Max
-793.2 -276.4 -270.8  228.3 7968.9

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.262e+03  6.701e+00   188.4   <2e-16 ***
square_footage 2.504e+01  2.147e-02  1165.9   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 684.6 on 152648 degrees of freedom
Multiple R-squared:  0.899,     Adjusted R-squared:  0.899
F-statistic: 1.359e+06 on 1 and 152648 DF,  p-value: < 2.2e-16
```

The Adjusted R-squared of this regression model is almost 90%. This is an indication that this model is a good fit. Since the true positive rate of both classifers are 100%, all of the fraudulent claims are identified. We sum up all the estimated coverage of each denied claim. The total estimated money saved on coverage is:

$$\begin{aligned}\sum \hat{y}_i &= \sum (1262.36 + 25.04 x_i) \\ &= 90092789.63701384\end{aligned}$$

An alernative way of estimating the money saved on coverage is to take advantage of the coverage distribution.

$$\begin{aligned}\sum \hat{y}_i &= N \times \hat{\bar{y}} \\ &= 10459 \times 8710.88 \\ &= 91107138.29375029\end{aligned}$$

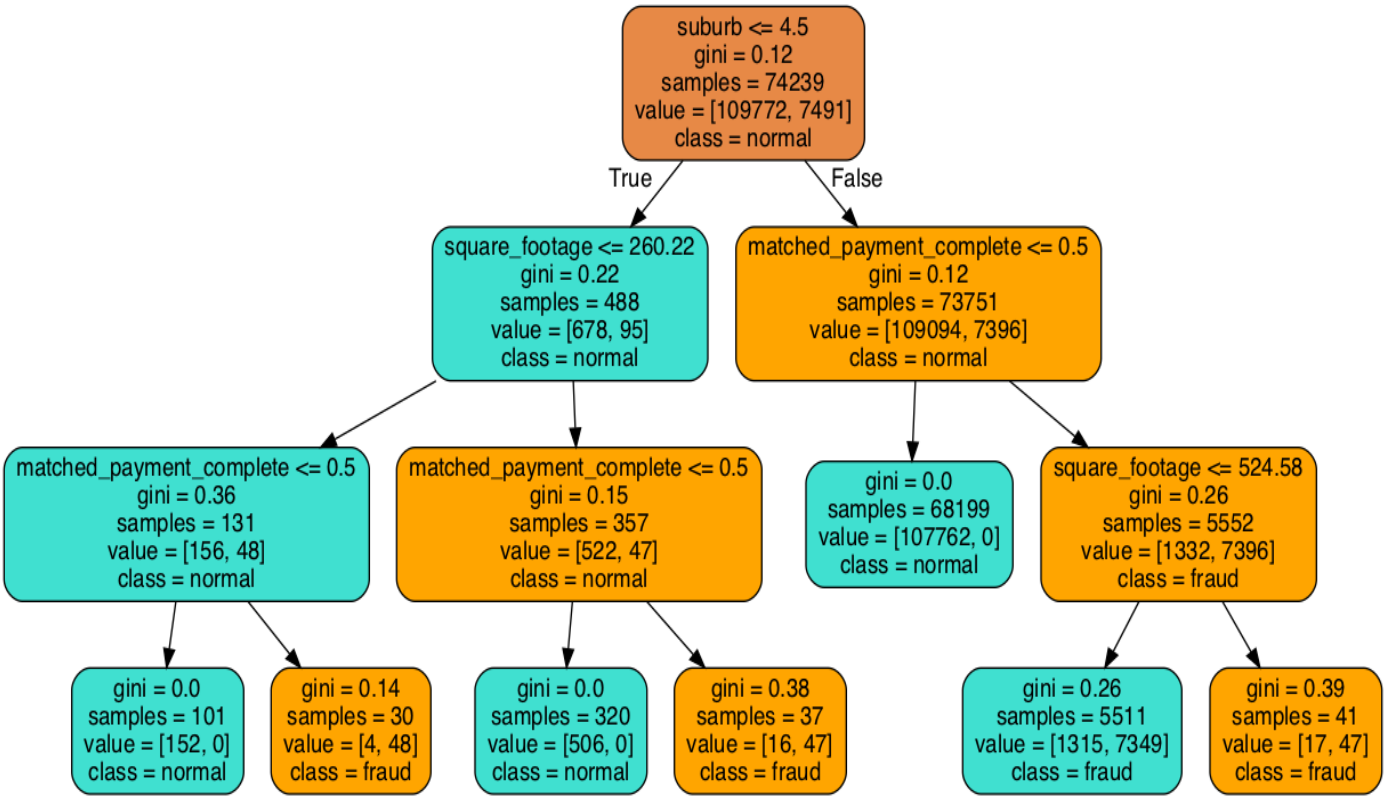
This results of these two independent approaches agree, which means that about 90 million dollars could be saved.

useful references
https://www.sciencedirect.com/science/article/pii/S0020025511006773

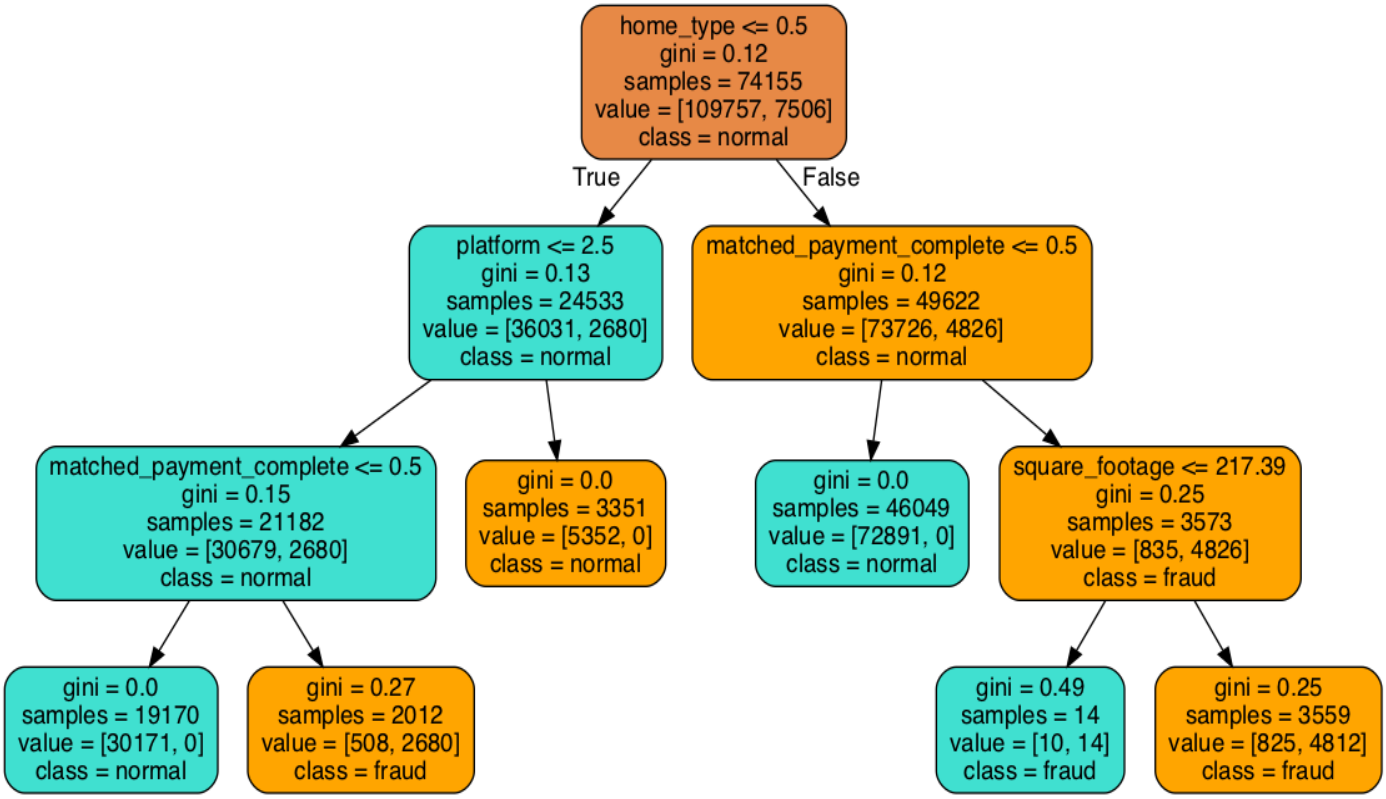
Appendix 1: Graphs

Decision trees in the final Random Forest:

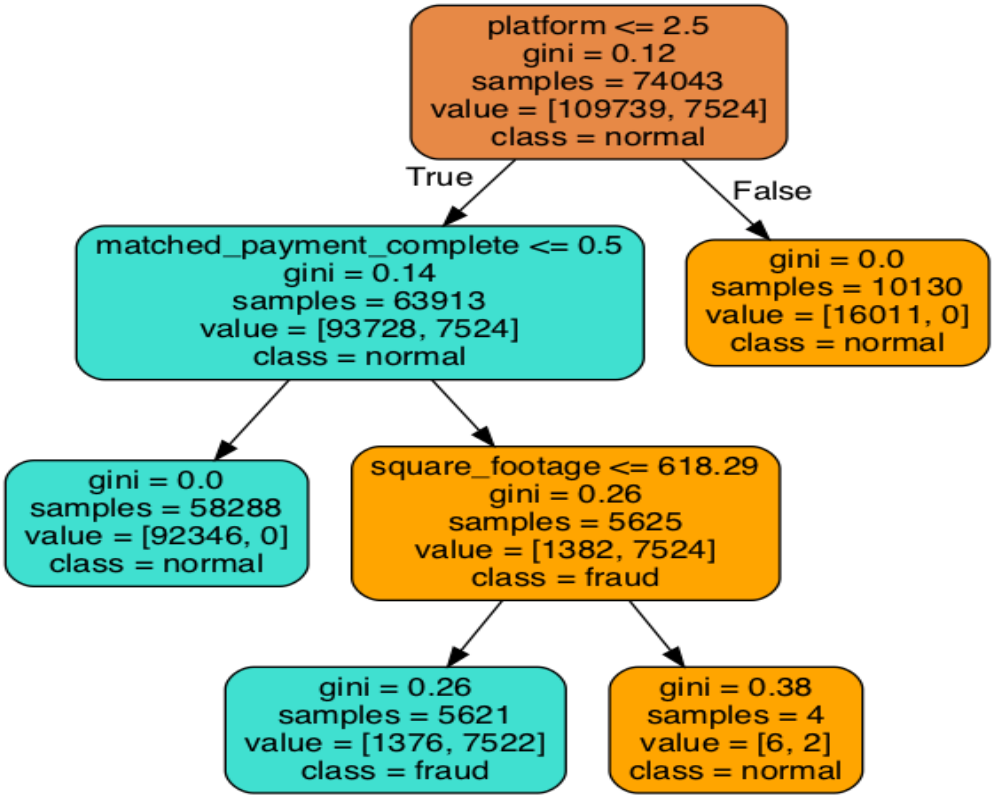
1. First decision tree



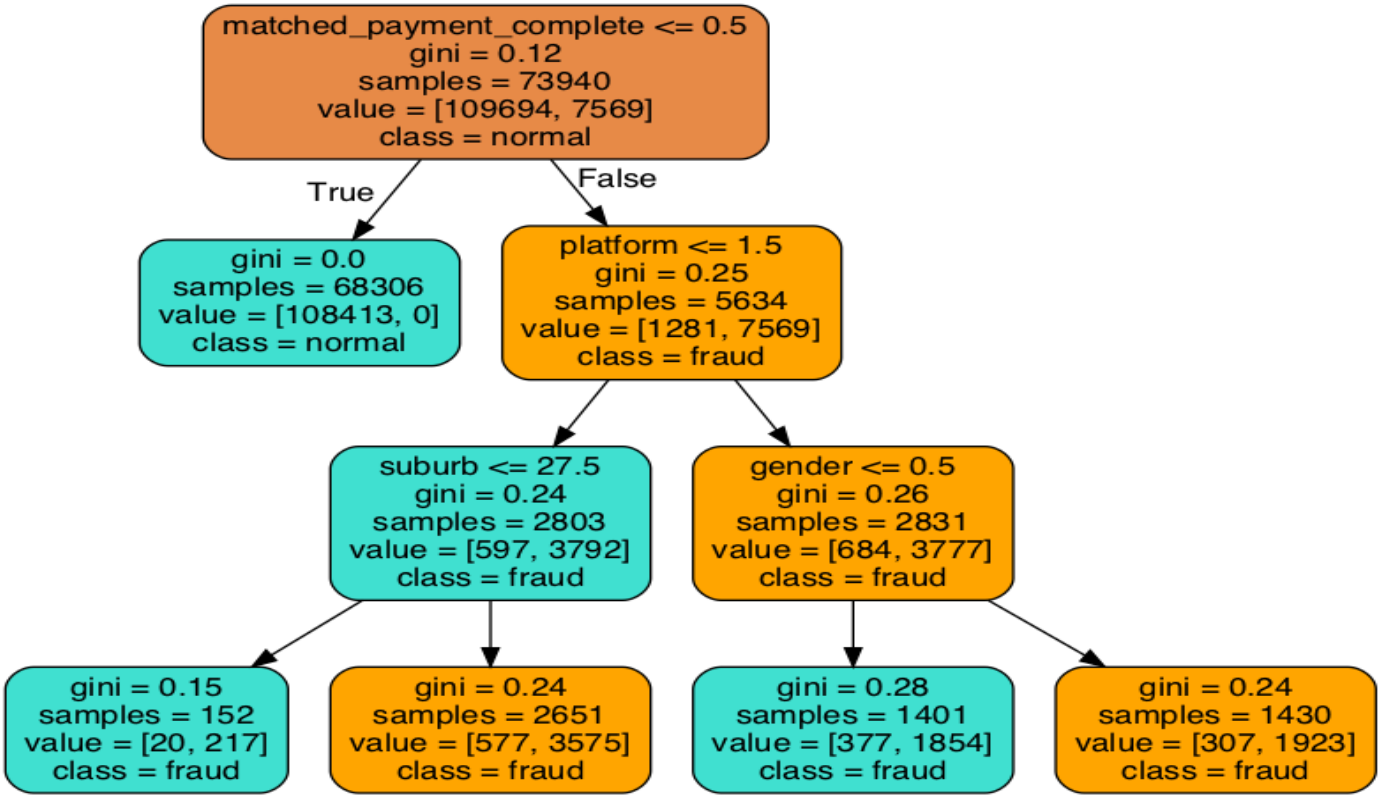
2. Second decision tree



3. Third decision tree



4. Fourth decision tree



Appendix 2: Code