STAT 4011 – Statistical Project (Part I)
Fraud Detection of Insurance Claims

Report

Group 10

Chin Yan Yi 1155078130
Law Lap Fei 1155079694
Li JianLiang 1155072287
Yuen Chun Wing 1155077960

I. Project Aim

According to the survey conducted by ISO, a Verisk Analytics Company, the average auto liability claim for property damage was $3,231; the average auto liability claim for bodily injury was $15,443. In addition, the average collision claim and the average comprehensive claim were $3,144 and $1,621 respectively. And according to simple statistics from FBI, the total cost of insurance fraud (non-health insurance) is estimated to be more than $40 billion per year. That means Insurance Fraud costs the average U.S. family between $400 and $700 per year in the form of increased premiums. (https://www.fbi.gov/stats-services/publications/insurance-fraud) It can cause enormous financial lost if there is any fraudulent claims. We would like to set up a classifier model to (learn the usual pattern of fraudulent cases by statistical measure, so that the company are able to filter the case out at the early stage of claim.)

In addition, the fraud can cause serious burden to the insurance companies. The ways to fraud are ever-changing, the fraudulent claim can take a variety of tactics to pretend they are eligible for the claims. So, the model is expected to maximize the flags of the suspicious claims, even if there may be mis-classification to fraud class, so that the company can take further investigation to prevent all frauds from happening.

II. Dataset Summary

The total number of the claim observation is 15,430. The dependent variable is set to be 'FraudFound_P' which has binary outcome '1' & '0', among the observations, there are 923 fraud cases, about 6% of the whole data. There are variables containing continuous, ordinal and categorical data which are:

a) Customer demographic details (Discrete and categorical variable)
b) Purchased policy (Categorical and Ordinal variable)
c) Claim circumstances (Discrete and Ordinal variable)
d) Other customer data (Ordinal variable)
e) Fraud found (Dichotomous variable)

, and 33 variables in total.

III. Dataset Pre-processing

We discovered 3 major problems in the dataset. The first one is the missing data found in the 1517 th row. Then, there are data column with repeated details, such as 'PolicyType' are combination of 'PolicyType' & 'Vehicle Category'. Moreover, there are strong association between the variables, for example 'NumberOfCars' & 'AddressChange_Claim'. The final problem is that the dataset contains lots of categorical, ordinal and interval variables which may not be applicable to our classifiers.We coped with them with the corresponding solutions listed below.

For the first problem, the missing data is an example of non-fraud case. We think that it has a relatively less effect to the model training since our focus is fraud detection, there are sufficient data to train the model for differentiating non-fraud case. So, our decision is to remove it directly.

Secondly, the mixed types of the variables are re-coded by the following rules. The ordinal variables will be ranked by the ordering of the variables from '1' to the number of the categories in the variable. The categorical variables are ranked by the frequencies of each categories. Categories with higher frequency are assigned to higher ranking, and vice versa.

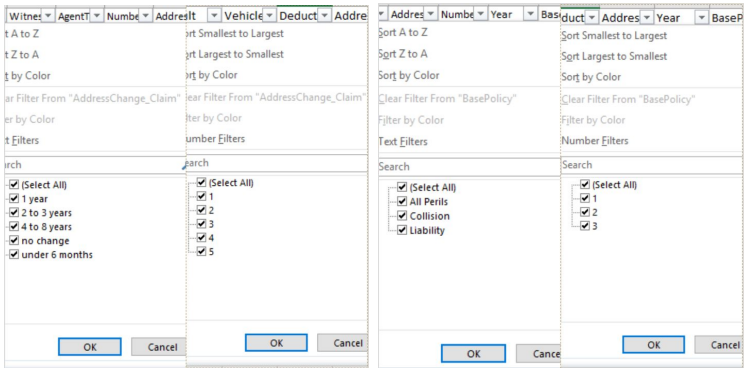

Figure (No. 1). The comparison of recode process of the variables

Thirdly, we have standardized the discrete and continuous variables to prevent the huge effect due to the scale and unit of the data.

Finally, we deal with the strong association by the feature selection step. We first applied binary logistic regression in R to check the significance of the variables in the model. The variable with Pr(>|z|)<0.05 (i.e. the variables with star(s)) are selected. We further check the variables with top importance by the random forest - recursive feature elimination algorithm. It recursively ran the random forest algorithm on each iteration, the accuracy graph and the suggestion of variables are then shown.

```
The top 5 variables (out of 8):
  Fault, Date, WeekNumber, Date.claimed, Month

> # list the chosen features
> predictors(results)
[1] "Fault"          "Date"           "WeekNumber"     "Date.claimed"   "Month"
"PolicyType"
[7] "VehicleCategory" "BasePolicy"
> # plot the results
> plot(results, type=c("g", "o"))
```

Figure (No.2). The list of the final chosen variables

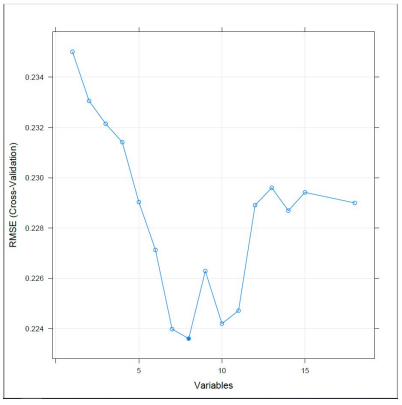

Figure (No. 3). The root-mean-square error of all variables

IV. Classifiers Methodology

i.Fisher Linear Discriminant Analysis & Quadratic Discriminant Analysis

Fisher Linear Discriminant Analysis (FLDA) characterizes or separates two or more classes of objects or events. In this project, FLDA is used to separates two classes from the dataset - fraud and not fraud. The main idea of FLDA is to find a projection to a line so that samples from different classes are separated. Therefore,we have to find a vector w, and project the n samples on the axis $y = w'x$. We have to choose the w which will maximize $J(w) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{S}_1^2 + \tilde{S}_2^2} = \frac{w' S_B w}{w' S_w w}$,where $\tilde{m}_i$ is the sample mean of the projected points in the ith class and $\tilde{S}_i$ is the variance of the projected points in the ith class. The criterion function $J(w)$ will be maximized when $w = S_W^{-1}(m_1 - m_2)$. FLDA will be equivalent to LDA for k=2, which is the case of our project. In R, we are using the package `MASS` and the function `lda()` to perform FLDA for our dataset and predict fraudulent cases.

Quadratic Discriminant Analysis (QDA) is quite similar to FLDA, rather separating sample by a projection of a line, QDA separates samples by a quadric surface. Therefore, The quadratic discriminant function is a quadratic function and contain second order terms

$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k$. For QDA, the decision boundary is determined by the quadratic function in *x*. Also, in QDA there is no assumption that the covariance of each of the classes is identical. For classification, we will need to find the class k which will maximize quadratic discriminant function. In R, we are using the `qda()` function inside the package `MASS` to perform QDA for our dataset and predict fraudulent cases.

For this project, we are training these two models with the training dataset, to find out the criteria on separating two classes, in order to predict which classes data in testing dataset belongs to.

ii.Logistics Regression

Logistic function is a monotonic, continuous function between 0 and 1 but never touch 0 and 1. The logistic function is also called the sigmoid function and it is often used to predict binary outcome. Mathematically, a binary logistic model has a dependent variable with a possible value within 0 to 1. Making predictions by logistic regression is a simple process like plugging the value of x and calculating to probability y from the logistic function.

$$f(x) = \frac{1}{1+e^{-x}}$$

Figure(No. 4) The Logistic Function

In the logistic model, the log-odds (x) is a weighted linear combination of the predictors. Therefore, the most essential issual is finding out the weight of each independent variable. By using the training dataset, the weighting of independent variables could be gotten by maximum likelihood estimation. Finally, input the value of the independent variables (x) into the logistics function, it will be converted to corresponding probability (y). When y is larger than a critical value, the result would be defined as positive and vice versa. However, setting up different critical value will cause different result , sensitivity, precision, specificity and accuracy. In order to get a higher sensitivity, a lower critical value could be set but it may result in lower precision rate.

In our case, we used logistics regression to predicted the fraud cases. Where the variables of each case have been put into the linear combination and given a weighting. Then a p-value would be generated for each cases through logistic regression. Finally, for the cases that the p-value is higher than the critical value would be defined as a supusious case.

iii. Nearest Neighbor Classification (k-NN)

Nearest Neighbor Classification, hereafter referred to as "k-NN " is a instance-based learning based on the distance between the observations. Nearer observations are voted to the same group which is called ‘cluster’. For continuous data, Euclidean distance is often used for measurement of the distance between the observations. After training of the model, the distance between the new test data and the observations are obtained. The new test point would be voted by the nearest neighbors, then assigned to the cluster of the highest number of voting.

$$nnc(\mathbf{x}, 1) = y_p, p = \arg\min_i ||\mathbf{x} - \mathbf{x}_i||^2.$$

Figure (No. 5). The formula to calculate the minimum distance between the nearest observation

The ‘k’ in the ‘k-NN’ is a factor that affects the decision boundary, discrete number ranging from 1 to positive infinity. The boundary visualised would be smoother with the value of k

increases. To choose the suitable value of 'k', we have tested different values of k with 10-fold cross validation, from 1 to 30 to understand the error of the k-NN model.

In order to maximize the accuracy, k-NN model is trained by the 3 datasets which are the simply standardized raw data, dataset filtered after variable selection of binary logistic regression and dataset filtered after variable selection of binary logistic regression and random forest - recursive feature elimination algorithm. The model trained by the raw dataset outperformed the other two, it is the final choice of the k-NN algorithm.

iv. Classification and Regression Tree (CART)

Since our project aims at predicting the fraudulent cases which is a qualitative variable "FraudFound_P" in the data set, we choose to use classification tree instead of regression tree which is used to predict quantitative response. Classification tree involves segmenting predictor space into a number of simple regions. We predict that each observation belongs to the most commonly occurring class of training observations in the region (particular terminal node region) to which it belongs. The set of splitting rules used to segment the predictor space can be summarized in a binary tree. (James, G., Witten, D., Hastie, T., Tibshirani, R., 2017) The representation for the CART model is a binary tree, i.e. each internal node splits into two branches. The leaf nodes in the bottom of the tree contain the output observations, i.e. Fraud in our project.

However, when the tree model built tightly fits the training data, it causes inaccuracy in predicting the outcome of testing data, which is called overfitting. To solve this problem, we can prune the tree to reduce the size of decision tree by removing sections of trees that provide little power to classify instances. Pruning reduce the complexity of the classifier, improving predictive accuracy by reduction of overfitting. We use `printcp()` to examine the cross-validation error results and select the complexity parameter (which is used to control the size of decision tree and select the optimal tree size) associated with minimum cross-validation error, then place into `prune()` function to do more precise prediction on fraudulent cases.

V. Comparison of the Classifiers Performance

Refer to the project aim and background,we are going to detect the suspicious claims to prevent the enormous financial lost. So, our major criterion for choosing the best model is the sensitivity which measures the proportion of true positive prediction to the sum of the true positive and false negative prediction.

	Predicted Class	
Actual Class	0	1
0	TN (True Negative)	FP (False Positive)
1	FN (False Negative)	TP (True Positive)

Accuracy= (TN+TP)/(TN+FP+FN+TP)
Precision=TP/(FP+TP)
Sensitivity= TP/(TP+FN)
Specificity= TN/(TN+FP)

Cross-validation is used for each model in order to test the model's ability to predict new data that was not used in estimating it, in order to search out problems like overfitting or selection bias. Also, it shows how the model will generalize to an independent dataset. This helps us to estimate how accurately each particular model will perform in practice. For this project, 10-fold cross-validation is used for each model.

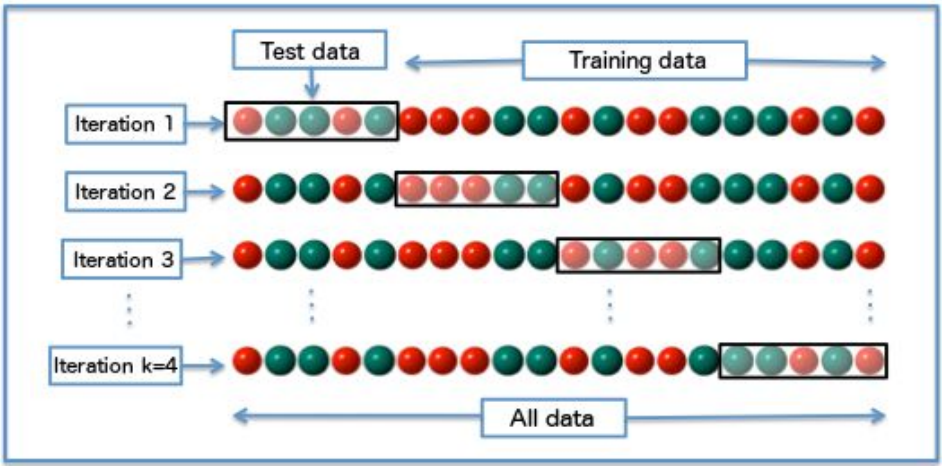

Figure (No. 6). Diagram of k-fold cross-validation with k=4. (https://en.wikipedia.org/wiki/Cross-validation_(statistics)#/media/File:K-fold_cross_validation_EN.jpg)

i. Fisher Linear Discriminant Analysis

	Predicted Class	
Actual Class	0	1
0	9416	1
1	583	0

Accuracy: 0.9416	Precision: 0
Sensitivity: 0	Specificity: 0.9999

```
"TN"
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
       "932"        "938"        "942"      "941.6"      "946.5"        "949"

      "SD=" "6.02218122167265"
"FP"
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
         "0"          "0"          "0"        "0.1"          "0"          "1"

       "SD=" "0.316227766016838"
"FN"
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
        "51"       "53.25"         "58"       "58.3"         "62"        "68"

      "SD=" "6.09280085200741"
"TP"
        Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
         "0"     "0"     "0"     "0"     "0"     "0"   "SD="     "0"
```

Figure (No. 7). Summary statistics of the 10 confusion matrices from FLDA

ii. Quadratic Discriminant Analysis

	Predicted Class	
Actual Class	0	1
0	9187	230

1	556	27

Accuracy: 0.9214	Precision: 0.1051
Sensitivity: 0.0463	Specificity: 0.9756

```
"TN"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
     "910"         "915.75"           "919"           "918.7"        "921.75"           "928"

     "SD=" "5.1218486246016"
"FP"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
      "18"          "19.25"          "23.5"            "23"           "25"            "31"

     "SD=" "4.2687494916219"
"FN"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
      "48"            "52"           "55.5"           "55.6"          "58"            "66"

     "SD=" "5.42012709978076"
"TP"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
       "1"             "2"             "2"             "2.7"           "3"             "6"

     "SD=" "1.63639169448448"
```

Figure (No.8). Summary statistics of the 10 confusion matrices from QDA

iii. Logistics Regression

The critical p-value cut-off is set at 0.05 as the highest precision could be gotten (> glm.pred=ifelse(glm.probs>0.05,1,0)

	Predicted Class	
Actual Class	0	1
0	3711	5706
1	76	507

Accuracy: 0.4218	Precision: 0.0816
Sensitivity: 0.8696*	Specificity: 0.3941

```
"TN"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
     "340"           "358"          "367.5"           "371.1"        "386.75"           "412"

     "SD=" "21.9314588864692"
"FP"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
     "529"         "561.25"          "573.5"           "570.6"        "578.25"           "597"

     "SD=" "19.5004273457447"
"FN"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
       "6"             "7"             "7"             "7.6"          "8.75"            "10"

     "SD=" "1.34989711542111"
"TP"
      Min.          1st Qu.          Median            Mean         3rd Qu.            Max.
      "42"            "44.5"          "51.5"           "50.7"          "55"            "62"

     "SD=" "7.18099343173817"
```

Figure (No.9). Summary statistics of the 10 confusion matrices from logistic regression

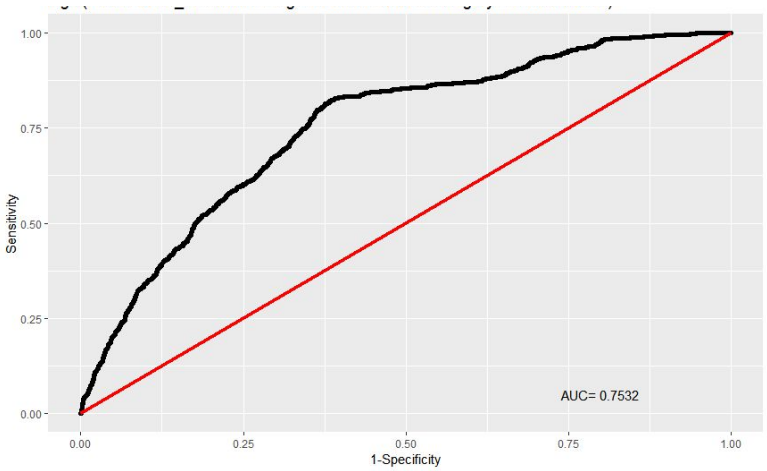

Figure (No.10). The ROC Curve for Logistic Regression. (AUC: 0.7532)

iv. Nearest Neighbor Classification (k-NN)

This model is tested by 3 datasets with 10-Fold cross validation, only the best result is shown below.
Dataset: Data with standardized continuous variables

k=1	Predicted Class	
Actual Class	0	1
0	9296	111
1	537	46

Accuracy: 0.9009	Precision: 0.1324
Sensitivity: 0.1836	Specificity: 0.9379

```
>   print(c(summary(tp),sd(tp)))
       Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
 5.000000000  9.250000000 10.500000000 10.400000000 12.000000000 15.000000000  2.913569784
>   print(c(summary(tn),sd(tn)))
        Min.       1st Qu.        Median          Mean       3rd Qu.          Max.
891.000000000 895.000000000 900.000000000 900.600000000 904.250000000 918.000000000   8.030497425
>   print(c(summary(fp),sd(fp)))
       Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
28.000000000 37.000000000 40.000000000 40.100000000 44.500000000 49.000000000  6.244108334
>   print(c(summary(fn),sd(fn)))
      Min.     1st Qu.      Median        Mean     3rd Qu.        Max.
39.00000000 46.25000000 47.00000000 47.90000000 50.50000000 59.00000000  5.48634669
```

Figure (No. 11). Summary statistics of the 10 confusion matrices from KNN

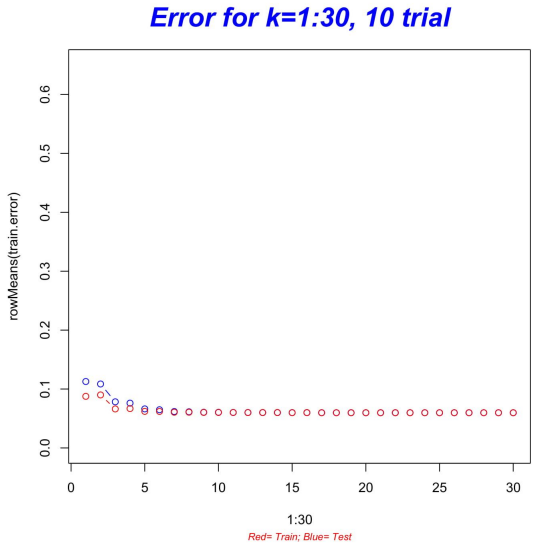

Figure (No.12). The train and test error of the k-NN model with k =1 to 30

v. Classification and Regression Tree (CART)

This model is tested by 3 datasets with 10-Fold cross validation, only the best result is shown below.
Dataset: Data with standardized continuous variables

With pruning, the performance of the classification tree model is shown in the following:

```
> conmat
       predict 10
actual    0    1
     0 9393   24
     1  527   56
```

Accuracy= 0.9449
Precision= 0.6839
Sensitivity= 0.0961
Specificity= 0.9975

Without pruning, the performance of the classification tree model is shown in the following:

```
> conmat
       predict 10
actual    0    1
     0 9103  314
     1  369  214
```

Accuracy=0.9317
Precision=0.4053
Sensitivity=0.3671
Specificity=0.9667

After pruning to ease the overfitting problem, the precision is increased, showing that when the prediction is “fraud” (1), the probability to get the correct prediction increased. On the other hand, the sensitivity is decreased after pruning, showing that when the observation is fraudulent ,the probability of getting correct prediction decreased.

With pruning, the classification tree is

Without pruning, the classification tree is

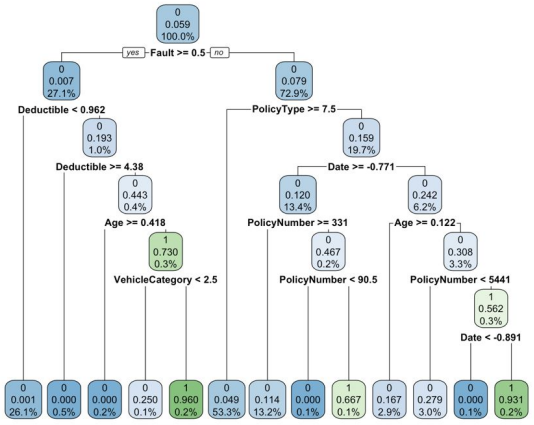

Figure (No.13). The pruned tree

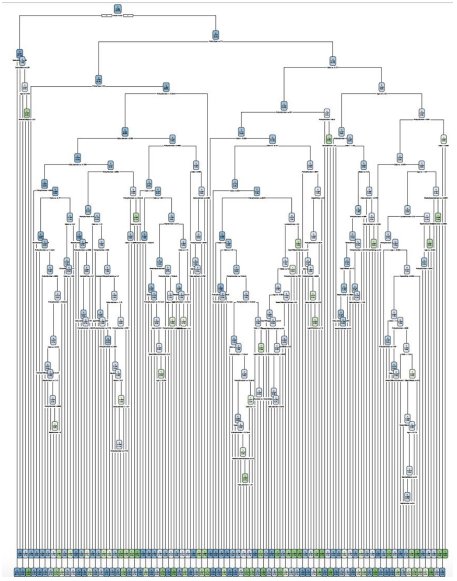

Figure (No.14). The original tree

The decision tree after pruning contains much fewer branches, being less complicated and easier to generate pattern of predicting fraud cases. According to the pruned decision tree, there are several patterns of predicted fraud:

1. If “Deductible”<4.38, “Age”<0.418, and “VehicleCategory”>=2.5, then there are 0.2% data is predicted as fraud.
2. If “PolicyType”<7.5, “Date”>=-0.771, and “PolicyNumber” is between 90.5 and 331, then there are 0.1% data is predicted as fraud.
3. If “PolicyType”<7.5, “Date”<-0.771, “Age”<0.122, “PolicyNumber”>=5441, and “Date”>=-0.891, then 0.2% data is predicted as fraud.

With pruning, the graph of cross validated error (xerror) against complexity parameter (cp) is

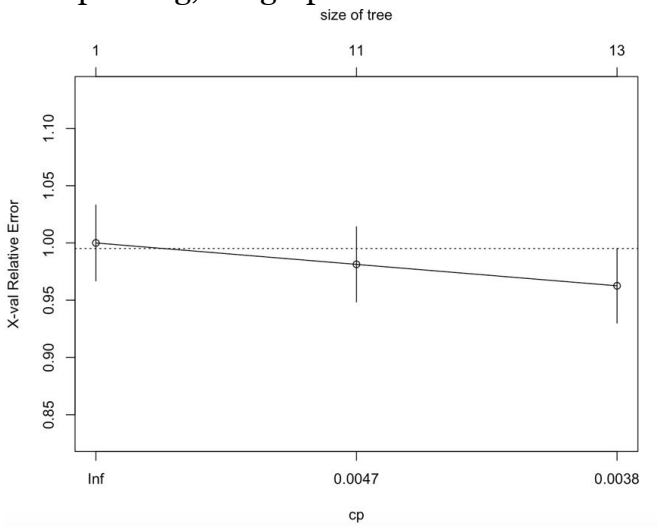

Figure (No.15). The error tested by 10-fold cross validation with pruning

Without pruning, the graph of cross validated error (xerror) against complexity parameter (cp) is

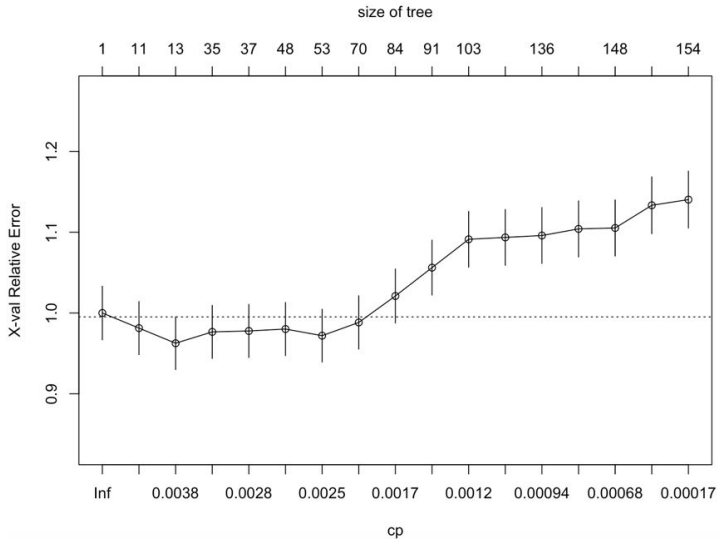

Figure (No.16). the error tested by 10-fold cross validation without pruning

For the tree model without pruning, when the cp value decreases as the increased size of tree, the cross validation error increases, implying the sign of overfitting. For the tree model with pruning, when the cp value decreases as the increased size of tree, the cross validation error decreases, implying the reduction of overfitting. Figures 15 and 16 show that the optimal size of tree is having 13 leaves which can obtain the minimum cross-validation error.

With pruning,

```
[1] "TN"
     Min.   1st Qu.    Median      Mean
    "943"  "3057.75"  "5160.5"  "5168.1"
  3rd Qu.      Max.
 "7284.25"   "9393"     "SD="  "2843.55456153815"
[1] "FP"
     Min.   1st Qu.    Median      Mean
      "3"    "7.5"      "12"      "12.9"
  3rd Qu.      Max.
     "16"    "24"      "SD="  "7.14065045278712"
[1] "FN"
     Min.   1st Qu.    Median      Mean
     "43"  "160.75"   "293.5"   "285.6"
  3rd Qu.      Max.
    "409.5"   "527"     "SD="  "164.325692858218"
[1] "TP"
     Min.   1st Qu.    Median      Mean
     "11"    "24"      "34"      "33.4"
  3rd Qu.      Max.
    "40.25"   "56"     "SD="  "13.251415018782"
```

Figure (No. 17). Summary statistics of the 10 confusion matrices from classification tree with pruning

Without pruning,

```
[1] "TN"
     Min.   1st Qu.    Median      Mean
    "906"  "2952.5"  "4997.5"  "5001.8"
  3rd Qu.      Max.
 "7052.25"   "9103"     "SD="  "2758.61397565275"
[1] "FP"
     Min.   1st Qu.    Median      Mean
     "40"  "112.75"    "175"     "179.2"
  3rd Qu.      Max.
    "248"    "314"     "SD="  "91.8656023160404"
[1] "FN"
     Min.   1st Qu.    Median      Mean
     "31"  "118.25"   "207.5"   "202.7"
  3rd Qu.      Max.
    "290.25"   "369"     "SD="  "114.594987286142"
[1] "TP"
     Min.   1st Qu.    Median      Mean
     "23"   "66.5"    "120"     "116.3"
  3rd Qu.      Max.
    "159.5"   "214"     "SD="  "62.8278954322397"
```

Figure (No. 18). Summary statistics of the 10 confusion matrices from classification tree without pruning

VI. Discussion of the Classifiers Performance

Although Knn is simple and intuitive and it can be applied to the data from any distribution, it is specially bad for high-dimensional data due to the curse of dimensionality. (Tomar, A., & Nagpal, A., 2016) The curse of dimensionality is a phenomenon that the additional dimensions dilute the 'relative contrast' of the data points. The cluster algorithm is unable to differentiate the distance or significance on an x,y plane.

There is also possibility that the aggregation of the large amount of data may unintentionally create a collection of irrelevant but correlated data, affect the subsequent analysis. (Patel, N. P., 2018) Also, it is computationally expensive, a large storage and computational cost is required for the distances between each data point to update the cluster belongs to. In addition, it does not work well for categorical data due to the Euclidean distance cannot reflect the differences between the categories. Finally, it need a large amount of samples to learn the pattern of clusters, so it does not work well for skewed data which has less amount of samples.

FLDA is useful when n is small, or the classes are well separated, and Gaussian assumptions are reasonable. Also, when $K > 2$. As FLDA with $K=2$ equivalent to LDA, it has the assumptions of LDA such as normally distributed classes or equal class covariances. This may result in FLDA projections may not preserve complex structure in the data needed for classification as there are too many categorical variables existing in the dataset. FLDA will also fail if discriminatory information is not in the mean but in the variance of the data.

QDA, because it allows for more flexibility for the covariance matrix, tends to fit the data better than FLDA in this project, but then it has more parameters to estimate.Therefore, the number of parameters increases significantly with QDA especially with a dataset with so many parameters in our project. Also, with QDA, we will have a separate covariance matrix for every class. If we have many classes and not so many sample data, this could be a problem when we are making prediction.

For logistics regression method, which is often used in predicting binary result. it has a pretty sensitive prediction when the critical value is set with a lower level. However, the higher sensitivity, the lower specificity it gets with a decreasing critical value. Therefore, there maybe a disadvantage that there are too much false positive result. And finally, the precision of the prediction result is much lower when compared with the other motheds.

For CART, since the variables in the dataset are mostly categorical, we decide to use classification tree instead of regression tree. In the classification tree, it can easily handle qualitative predictors without creating dummy variables. Also, trees can be displayed graphically, being easily interpreted by non-experts. However, there is an overfitting problem, i.e. the tree model tightly fits the training data, resulting in inaccuracy in predicting the outcome of testing data. This problem can be resolved by pruning tree. Removing some sections of trees can reduce the complexity of classifiers and improve the predictive accuracy, hence the overfitting problem is resolved. Besides, highly complicated decision tree tends to have a low bias, but the model may be difficult to to fit the new data. Moreover, small change in data can cause large change in structure of optimal decision tree.

We also attempted for different approaches to perfect our classification. Discussion briefed due to limited contexts available. For data-preprocessing, we tried to use Principal Components Analysis to reduce the dimensions but later Prof. Yam taught us that PCA is not efficient to treat categorical data in lectures of RMSC 4002. We also tried to recode the categorical variables to dummy variables . However, this approach produced 75 variables. They are all significant in logistic regression variable selection, redundant variables cannot be identified in the feature selection process. So, we finally recoded the variable according to their frequency. Our final decision is only conduct two steps in data-preprocessing.

In addition, we gave trials on another classifier which is artificial neural network (ANN). For artificial neural network, it takes advantage of the functions and mechanism of our brain, it process the record

one by one, classify them and check the classification result with the actual classification. The error is used to modify the network, mimic the the neuron get the input and output in the layers. (Yam, S.C., 2018) However, the trained network in our work returns all '0' prediction.

VII. Final Choice of our model

Our final model is set to be logistic regression model. It is expected to provide reliable information about fraud claims. So, the performance parameters (i.e. Precision, accuracy, sensitivity and specificity) are our important indices to show the efficiency of fraud detection. Although we chose the logistic regression model which generates a relatively large portion of negative positive cases, the indices are not our sole concern.

We also pay attention to the balance between the cost of fraud investigation and financial lost due to fraud claims. From the information provided by Indeed.com which is an American worldwide employment-related search engine for job listings, with 452 records provided by their users and fraud investigators in America, the distribution is right-skewed, and the average annual salary is USD$43,526. Compared to the total annual cost of non-health insurance fraud estimated by FBI which is USD$40,000,000,000, it is about 634,316 times of the huge lost due to fraud, which also means that this financial lost can be converted about 600,000 job openings of fraud investigators. The financial cost of fraud significantly exceeds the cost of hiring fraud investigators.

To apply this to our project, the classifier is never perfectly fit for the future data since the construction of the model is fully based on the historical data, there will be new tactics for the frauds. The threshold value of assigning the observation to the cluster of fraud is set higher to handle the future necessity. At the same time, it sacrifices the rate of detecting true negative case. Again, considering the cost of fraud claims and fraud investigation, in our opinion, it is tolerable to accept less true negative case based on the statistics. It can effectively reduce the financial cost of frauds.

VIII. Improvement Suggestion

We would like to divide the our discussion of model improvement in several aspect.

The first aspect is data collection. As we mentioned before, the strategy of frauds is ever-changing, the model has to be always updated. So, sufficient information of recent claim cases is needed to provide the model with a good knowledge about the recent fraud and non-fraud normal cases such as the changes of customer behaviors and society. We suggest the past 10 years information should be applied. Also, the dataset format also need reform, eliminate redundant data columns such as the 'BasePolicy', refine the data collection step to collect more accurate data such as 'NumberOfCars' should record actual number of cars instead of interval. Such policies can refine the data analysis result.

For the data analysis part, there may exist imbalance data problem. Imbalance data means there is a big difference between the size of observed class from the dataset. (Seoung & Young, 2014) (Lemaître, 2017) In our case, the proportion of fraud case only occupies 6% in our sample. The algorithm may not obtain enough data to identify the pattern of classical fraud cases. We suggest the two sampling techniques to improve the efficiency which are oversampling and downsampling. Oversampling takes all observation values in the bigger class which is non-fraud in our project, and then increase the size of the smaller class by duplication of the data in smaller class. (Seoung & Young, 2014) (Lemaître, 2017) (Rahman, 2013) While undersampling uses all the sample from the smaller class, and the same size of the observed value in the bigger class. Both of them lead to different problems. For oversampling, it leads to overfitting. (Chawla) And undersampling may remove important data which also loses the reliability of the result. But still, they are effective to counter the imbalance data. (Chawla)

Furthermore, it is obvious that there are lots of categorical data. To greatly improve the model performance, other than data cleaning and sampling method, it is suggested that appropriate models should be used. Random forest is a good choice, it is based on the classification tree. Random forest

grows a lot of classification tree and each classification tree will vote for a new object to be classified. The forest will choose the classification with the most votes over all the tree in the forest. (Breiman, L., 2001) This bagging-like skills further improve the accuracy of classification tree which is already a good algorithm to the dataset.

IX. Conclusion

Through a series of practices on machine learning, all group members learn various skills.

First, our skills at using the data analysis software is greatly improved. Before our project, part of our group even forgot how to write a for loop to run cross-validation of his model. In this project, not only do we write our own code, but we also viewed the presentation of the other groups. Their performance wondered us, there are million ways to interpret and analyze the data. It is impressing that there are groups do not only follow the instruction of our great lecturer, they tried many new classifiers such as random forest and Naive Bayes classification. They also find many innovative parameters to evaluate their performance such as F-measure and ROC curve. We learnt how to convert our idea into programming language to satisfying various requirement of data analysis and visualisation of data.

Secondly, we understand that appropriate methods should be applied to cope with the problems. We attempted to construct the model with the suggested classifiers but we never considered the attributes of the data. The dataset contains lots of categorical, ordinal and interval data, the suggested classifier in fact has very weak effect on these data. Due to the time limit, we do not have enough time to choose a new classifier and investigate how do it suit our dataset. We still tried lots of methods to modify our data to fit in the classifiers, that is the reason of a relatively long passage discussing the data-preprocessing. Fortunately, the result is acceptable. We are happy to have this experience, so we have a deeper understanding about the mathematics algorithm behind the classifiers to choose an appropriate method to the data we will handle in the future.

Finally, we have a wonderful experience of work simulation. Our project aim is to promote our product which is a insurance fraud detector to our boss and our lecturer. We learn the unknown theories by ourselves, fight with the deadlines and try to visualise our best result. More importantly, we learn what is goal-oriented. All our models are not able to detect 100% true positive cases. We need to come through a solution. We tried to mix the models for a better result but failed. But we still need an reasonable answer for satisfying our customer. We recognized the real situation of the insurance fraud and fraud investigation and further compared them numerically. We finally concurred to choose logistic regression which is a economic choice to assign more fraud investigators to investigate the suspicious cases to prevent fraud from happening. We tried to make a decision with consideration of maximizing the customers' benefit. It feels like we are employees to persuade the boss to accept our proposal. It would be the best experience for us before entering the workplace.

XI. References

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Chawla, N. (n.d.). DATA MINING FOR IMBALANCED DATASETS: AN OVERVIEW. Retrieved October 21, 2018, from https://www3.nd.edu/~dial/publications/chawla2005data.pdf

Elliot, M., Fairweather, I., Olsen, W., & Pampaka, M. (2016). Oversampling. A Dictionary of Social Research Methods, A Dictionary of Social Research Methods.

Fraud Investigator Salaries in the United States. (2018, October 17). Retrieved October 21, 2018, from https://www.indeed.com/salaries/Fraud-Investigator-Salaries

James, G., Witten, D., Hastie, T., Tibshirani, R. (2017). An Introduction to Statistical Learning with Applications in R: Springer.

Lemaître, G. K., Nogueira, F., & Aridas, C. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18, 1-5.

Patel, N. P., Sarraf, E. H., & Tsai, M. (2018). The Curse of Dimensionality. Anesthesiology, 129(3), 614-615.

Rahman, M., & Davis, D. N. (2013). Addressing the Class Imbalance Problem in Medical Datasets. International Journal of Machine Learning and Computing, 3(2), 224-228. doi:10.18411/a-2017-023

Seoung-Hun Park, & Young-Guk Ha. (2014). Large Imbalance Data Classification Based on MapReduce for Traffic Accident Prediction. Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on, 45-49.

Tomar, A., & Nagpal, A. (2016). Comparing Accuracy of K-Nearest-Neighbor and Support-Vector-Machines for Age Estimation. International Journal of Engineering Trends and Technology, 38(6), 326-329. doi:10.14445/22315381/ijett-v38p260

Yam, S. C. (2018). RMSC4002_Combined_Lecture_Notes[Class handout]. Hong Kong: the Chinese University of Hong Kong, RMSC 4002.