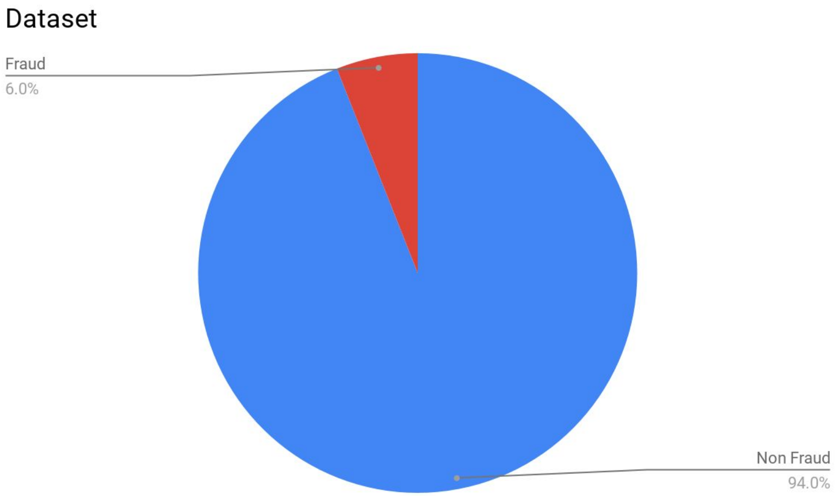
STAT 4011 Project 1

Fraud detection of insurance claims

Group 10
Member:
Chin Yan Yi
Law Lap Fei
Yuen Chun Wing
Li JianLiang

About the Dataset



Number of Clients : 15430

Number of Variables : 33

Number of Fraud Cases : 923(~6%)

Type of Variables:

Numeric, Ordinal, Categorical

*so many categorical variables

Background information

In 2013, the average auto liability claim for property damage was $3,231; the average auto liability claim for bodily injury was $15,443 (ISO, a Verisk Analytics company).

In 2013, the average collision claim was $3,144; the average comprehensive claim was $1,621 (ISO, a Verisk Analytics company).

Aims

1. Creata Classifier from old dataset
2. Detect fraud clients
3. Find out suspicious case
4. Reduce the cost of investigation
5. Reduce the loss from insurance fraud claims

Data Cleaning

Missing Data Problem:

- 1 Missing Data (The 1517th observation) Found
- Remove Directly

Data Mismatching Problem:

- Age/ Age of Policy Holder
- Policy Type/ Vechicle Category/ Base Policy
  Remove by the feature selection methods

Strong Association between the variables

Categorical Variables:

- Maybe misunderstanding if turning into numerical such as dummy variable

Variable Selection

1. Binary Logistic Regression
2. Random Forest-Recursive Feature Elimination algorithm

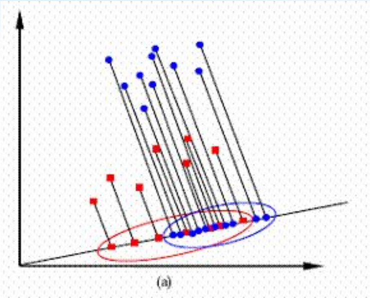
The top 5 variables (out of 8):
Fault, Date, WeekNumber, Date.claimed, Month

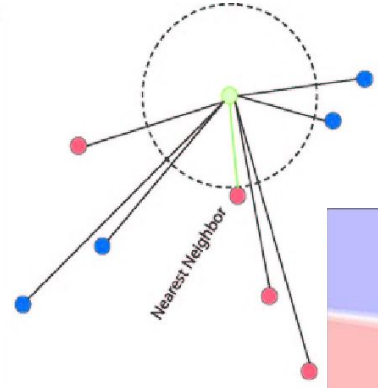
```
> # list the chosen features
> predictors(results)
[1] "Fault"        "Date"        "WeekNumber"     "Date.claimed"    "Month"
"PolicyType"
[7] "VehicleCategory" "BasePolicy"
> # plot the results
> plot(results, type=c("g", "o"))
```

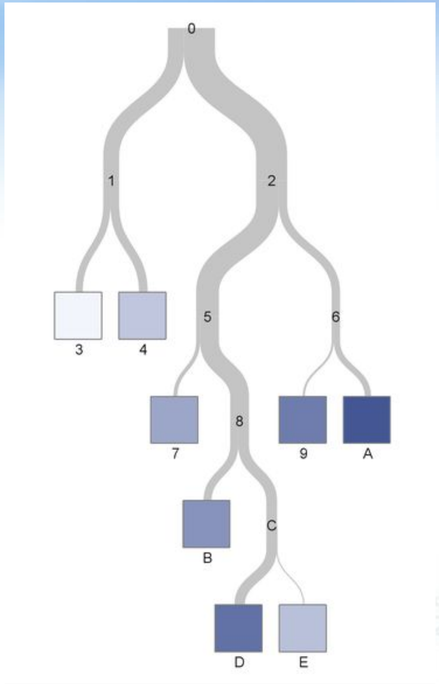
	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	1.435e+01	4.909e+00	2.924	0.003461	**
Date	-2.665e+01	8.698e+00	-3.064	0.002186	**
WeekNumber	1.622e+01	4.249e+00	3.819	0.000134	***
Date.claimed	3.615e-01	1.150e-01	3.142	0.001676	**
Weekofyearclaimed	2.702e-01	1.022e+00	0.264	0.791580	
Age	-1.858e-01	1.521e-01	-1.222	0.221890	
Deductible	8.191e-02	3.118e-02	2.627	0.008622	**
Year	2.441e+01	8.120e+00	3.007	0.002641	**
Month	-2.143e+00	5.034e-01	-4.256	2.08e-05	***
WeekOfMonth	-5.167e-01	1.199e-01	-4.311	1.62e-05	***
DayOfWeek	7.570e-02	3.283e-02	2.306	0.021110	*
Make	-9.098e-03	1.604e-02	-0.567	0.570673	
AccidentArea	2.492e-01	1.027e-01	2.428	0.015202	*
DayOfWeekClaimed	-1.312e-01	6.152e-01	-0.213	0.831096	
MonthClaimed	-5.956e-02	1.437e-01	-0.414	0.678531	
WeekOfMonthClaimed	-4.524e-03	2.954e-02	-0.153	0.878276	
Sex	2.533e-01	1.119e-01	2.263	0.023631	*
MaritalStatus	1.175e-01	8.446e-02	1.391	0.164307	
Fault	-2.640e+00	1.703e-01	-15.502	< 2e-16	***
PolicyType	-8.728e-01	5.903e-02	-14.786	< 2e-16	***
	1.922e+00	1.127e-01	17.065	< 2e-16	***
	7.020e-02	2.680e-02	2.619	0.008817	**
	3.453e-05	2.267e-05	1.523	0.127669	
	-8.600e-03	7.672e-03	-1.121	0.262293	
	2.219e-02	3.163e-02	0.701	0.483008	
t	-2.532e-01	1.024e-01	-2.473	0.013395	*
	-6.946e-02	2.438e-01	-0.285	0.775708	
	-2.815e-02	3.752e-02	-0.750	0.453102	
	-6.806e-03	3.863e-02	-0.176	0.860145	
	3.939e-02	1.178e-01	0.334	0.738196	
	-5.327e-01	2.683e-01	-1.986	0.047081	*
	-2.127e-01	6.154e-01	-0.346	0.729621	
	-9.466e-01	5.169e-01	-1.831	0.067081	.
	-5.136e-02	3.031e-02	-1.695	0.090148	.
	1.195e-01	4.114e-02	2.906	0.003660	**
NumberOfCars	-7.761e-02	1.081e-01	-0.718	0.472775	
BasePolicy	5.586e-01	6.590e-02	8.477	< 2e-16	***

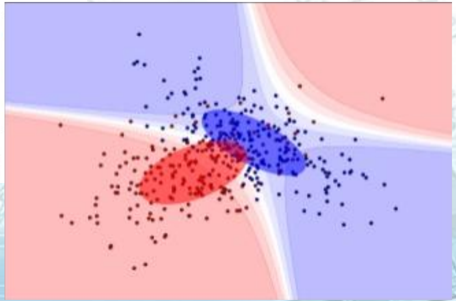
Classification Methods

1. Fisher Linear Discriminant Analysis
2. Quadratic Discriminant Analysis
3. Logistics Regression
4. Nearest Neighbor Classification (kNN)
5. Classification Tree









Methodology

1. Standardize the continuous data, e.g. Age.
2. Create train and test datasets by the 10-fold cross validation
3. Model the train data by the classifiers

   In Logistic regression,

4. Enter the test data to the model
5. Calculate the z-value by the formula, for example $z = w_0 + w_1 x_1 + w_2 x_2$
6. Map the z-value to probability by Sigmoid Function $s(z) = \frac{1}{1 + e^{-z}}$
7. Return the prediction according to the thresold value/ decision boundary set
8. Generate confusion matrix to compare the actual and predicted values
9. Repeat the above procedures 9 times

Performance

Logistic Regression

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

predict actual	0	1
0	3711	5706
1	76	507

Accuracy 0.4218	Precision 0.0816
Sensitivity 0.8696*	Specificity 0.3941

Performance

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

Classification tree

With **prune tree**

Cut off the least important splits, based on complexity parameter (cp).

predict actual	0	1
0	14451	66
1	822	101

Accuracy 0.9425	Precision 0.6048
Sensitivity 0.1094	Specificity 0.9954

Performance

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

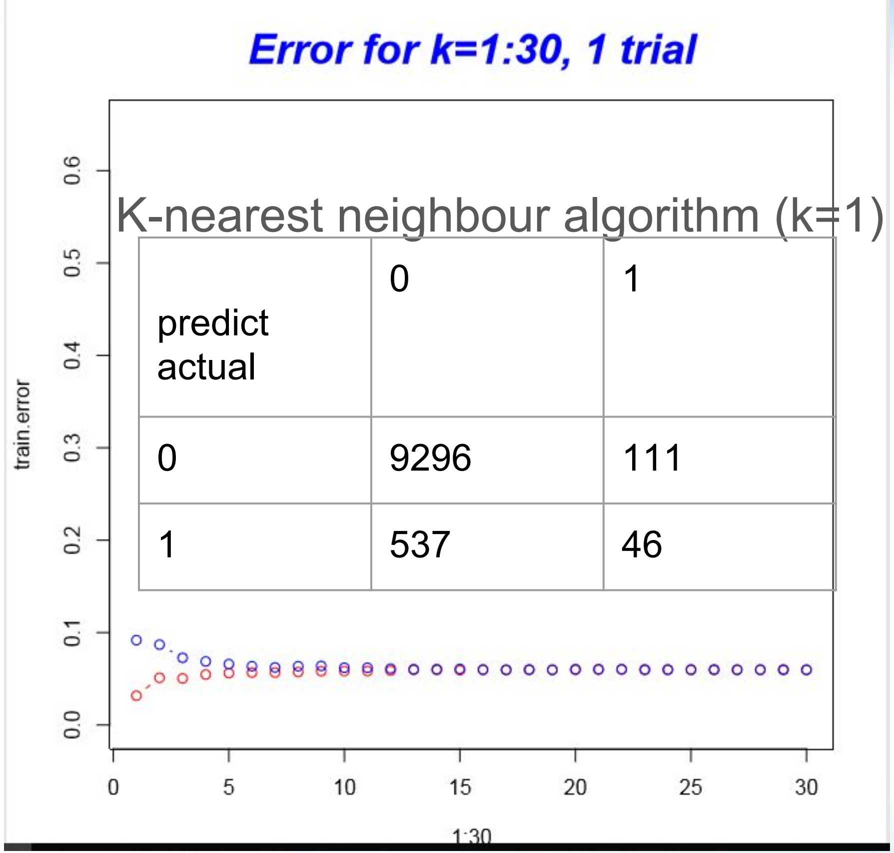
Classification tree

Without prune tree

predict actual	0	1
0	14035	482
1	577	346

Accuracy 0.9314	Precision 0.4179
Sensitivity 0.3749	Specificity 0.9668

Performance



K-nearest neighbour algorithm (k=1)

predict actual	0	1
0	9296	111
1	537	46

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

Accuracy 0.9009	Precision 0.1324
Sensitivity 0.1836	Specificity 0.9379

Performance

Quadratic Discriminant Analysis

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

predict actual	0	1
0	9187	230
1	556	27

Accuracy 0.9214	Precision 0.1051
Sensitivity 0.0463	Specificity 0.9756

Performance

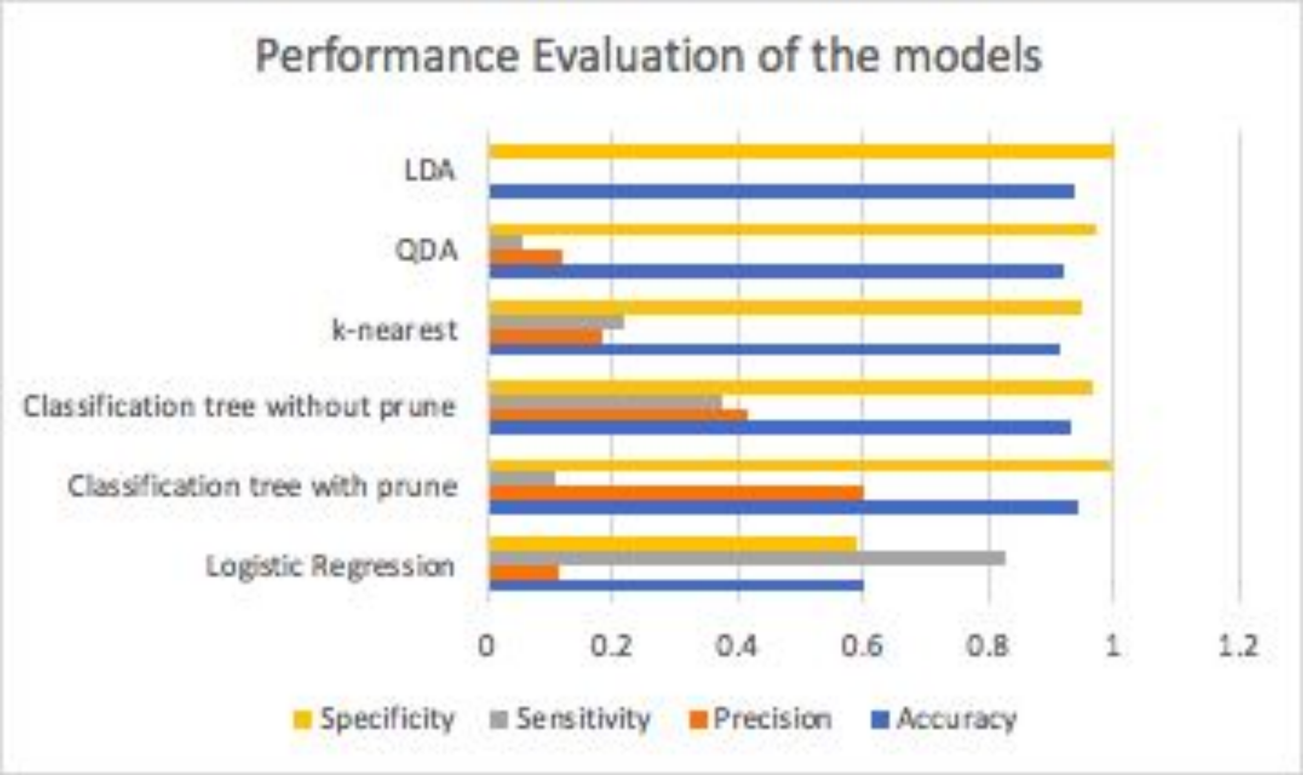
Fisher Linear Discriminant Analysis

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

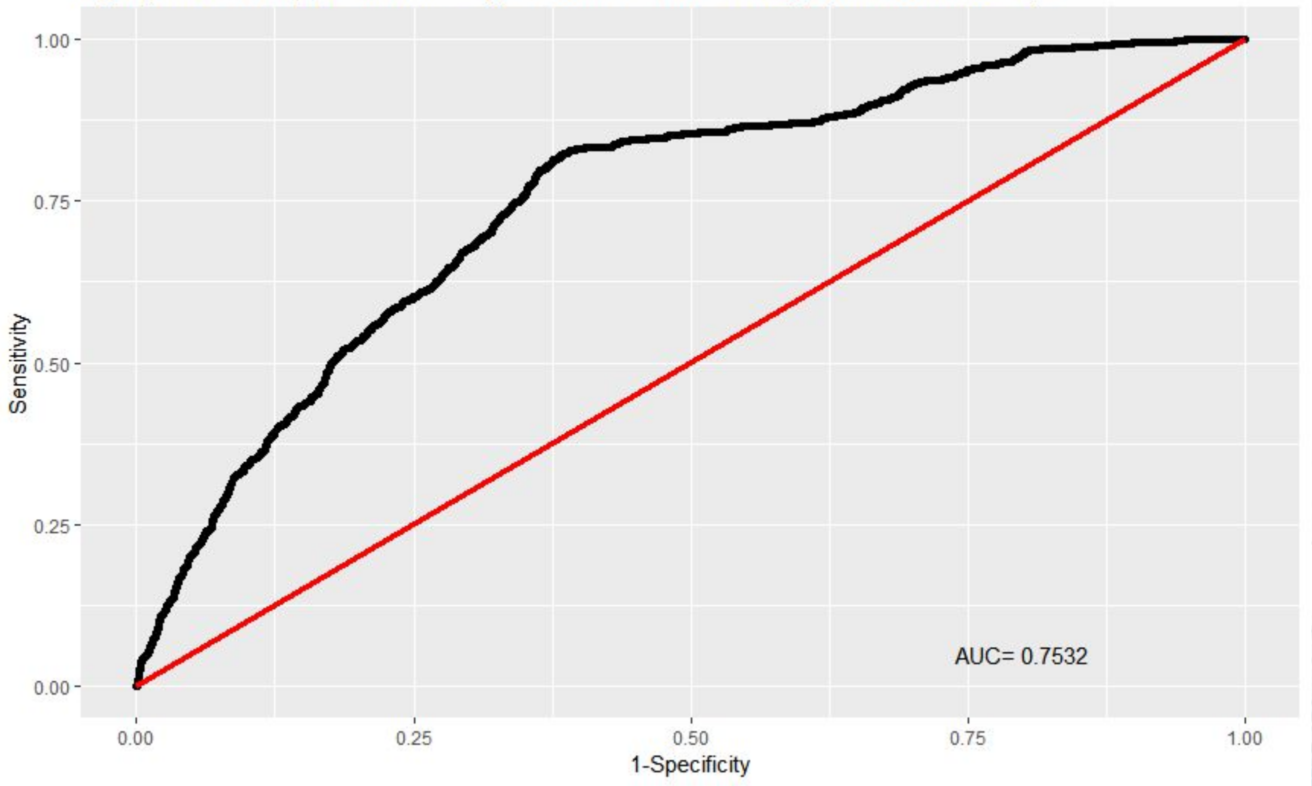
predict actual	0	1
0	9416	1
1	583	0

Accuracy 0.9416	Precision 0
Sensitivity 0	Specificity 0.9999

Logictics Regression



ROC Curve - Performance Evaluation



Sensitivity(TPR):

TP/(TP+FN)

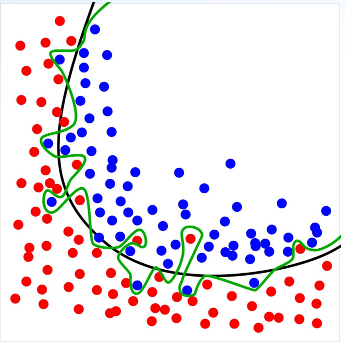
Specificity(TNR):

TN/(FP+TN)

Area under curve:

0.7532>0.5

Comparison



Classification tree:

- Overfitting: robust to outliers

training data <-difference is big-> testing data → sampling errors

- Cross-validation impurity increases for same split

  Solution: stop growing the tree

  because divergence in error (impurity) → overfitting more serious

Comparison

Knn is specially bad for high-dimensional data due to the curse of dimensionality.

Computationally expensive

Not work well for categorical data

Not work well for skewed data

LDA is a parametric method, that it assumes unimodal Gaussian likelihoods)

The LDA projections may not preserve complex structure in the data needed for classification

Conclusion

Our final choice: Logistic Regression which wins the sensitivity

- Advantage:

It returns discrete prediction (1 or 0).

Fast to train, returns probability scores
It effectively catches more TP cases.

- Disadvantage:

Change the decision boundary for classifying an observation to non-fraud sacrifices the FP rate to reduce the FN rate.

Improvement suggestion

- Data Collection:

Increase the sample size

Use recent data(e.g. past 10 years) to suit the recent behavioral and social changes

- Data Analysis:

Oversampling/ Downsampling to solve the imbalance data problem

Regulate the data cleaning and classifiers to accomodate the categorical data.

Thank You

Have a nice day!