

## Part 1

This is two part problem: selecting between supplier A and B for product one; selecting between supplier C and D for product two. Work was performed in excel.

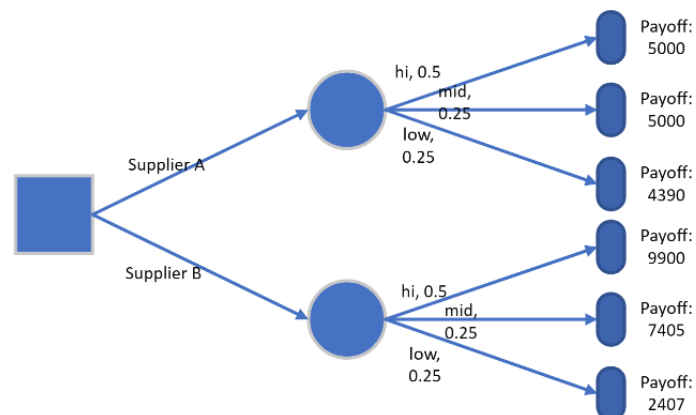
### Product 1

With product one, parameters re: states, probabilities, supplier constraints of both suppliers are entered. Simulation tables are set up for each supplier and each state of demand (hi, med, lo). RNG is set up for 1000 samples for uniform distribution in min/max range. The value is rounded to nearest integer to be demand. With order sizes constrained to multiples of 1000 and 1300, and range of demand from 900 to 1500, the first multiples are placed (1k, 1.3k). Total cost is  $\text{order\_size} * \text{unit\_cost} + \text{order\_fixed\_cost}$ . Inventory on hand is determined by  $\# \text{orders} * \text{order\_size}$ . Revenue is the lesser of demand or inventory on hand, times unit price. Profit is revenues minus total costs.

Supplier A									
supplier A constraints:				order breakeven quantity					
	order size	1000	units per order	$(\text{fixed cost} + \text{varCost} * \text{batchSize}) / \text{price}$					
	fixed cost	0		800					
	unit cost	20 \$/un		799	0				
				800	1				
				801	1				
				1799	0				
simulations (explanations in notes)									
scenario	A, hi		(demand actual)						
	round	diceroll (data a	rounded dice	total cost	inventory	revenue	profit		
	1	1300.250252	1300	20000	1000	25000	5000		

Figure 1: Excerpt from simulation table, order breakeven was another thing

The average profit of each state is determined. Expected Value (EV) is state avg profit (payoff) times state probability. Total expected value is the expected profit of selecting a supplier, and is determined by adding the supplier, state EV's. Alternatively, sum product of state avg profits and state probabilities, that is,  $0.5 * 5000 + 0.25 * 5000 + 0.25 * 4390$  for supplier A. This is repeated for supplier B.



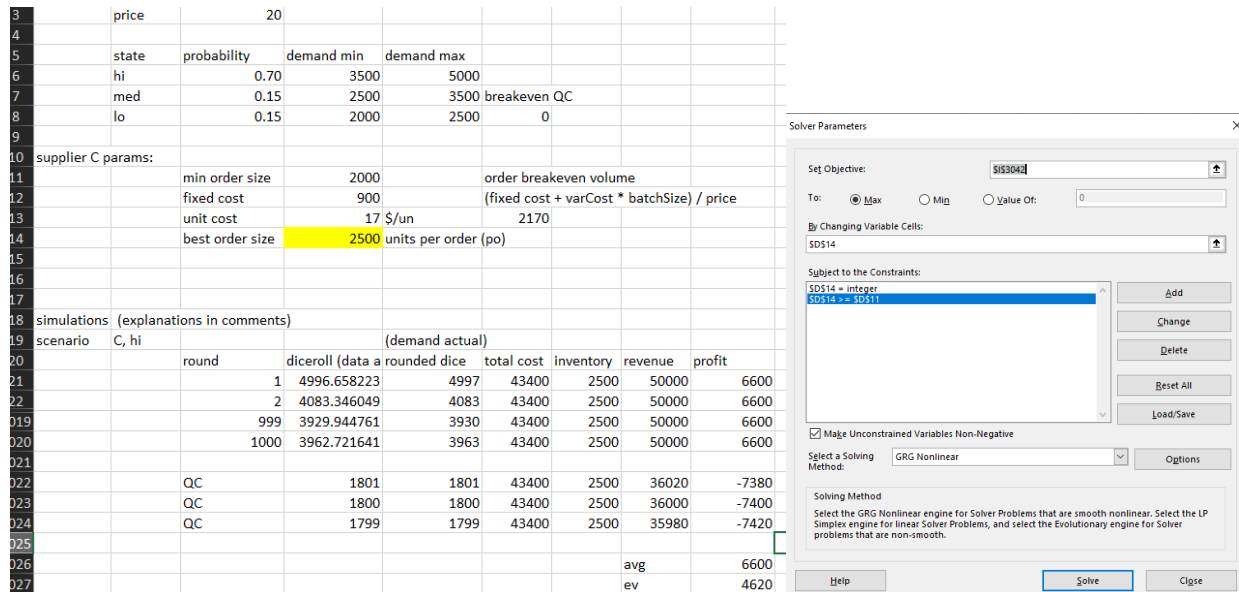
Decision tree diagram made in visio for earlier draft, payoffs about the same

EV total for supplier A: **4847.43** ; supplier B: **7402.60**

**Supplier B is selected as their total EV is higher than of supplier A.**

## Product 2:

Setup is like Product 1, the big difference is that order size is selectable, though with constraints (minimum order size). By using Excel solver, optimal order size is calculated to maximize supplier total EV. This process is used in sheets "Q1 P2 SC" and "Q1 P2 SD" for suppliers C and D.



Figures 2: Single table and selectable batch size set up for D, hi. (left)

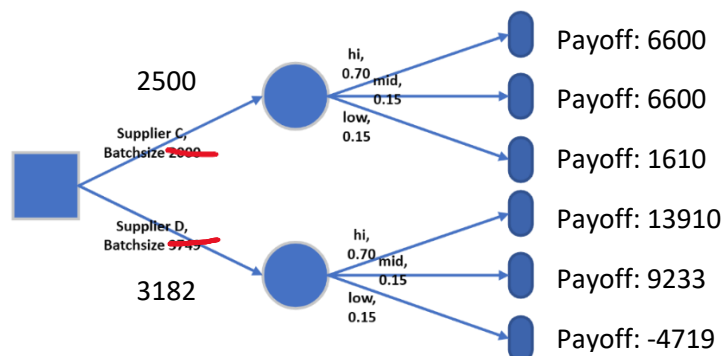
Figure 3: solver optimizing for supplier D's sum total ev at cell K4032 (right)

By using solver to maximize profits by varying order size per supplier,

Supplier C at order size 2000 has net expected profit: 7755.04

Supplier D at order size 3749 has net expected profit: 11,957.47

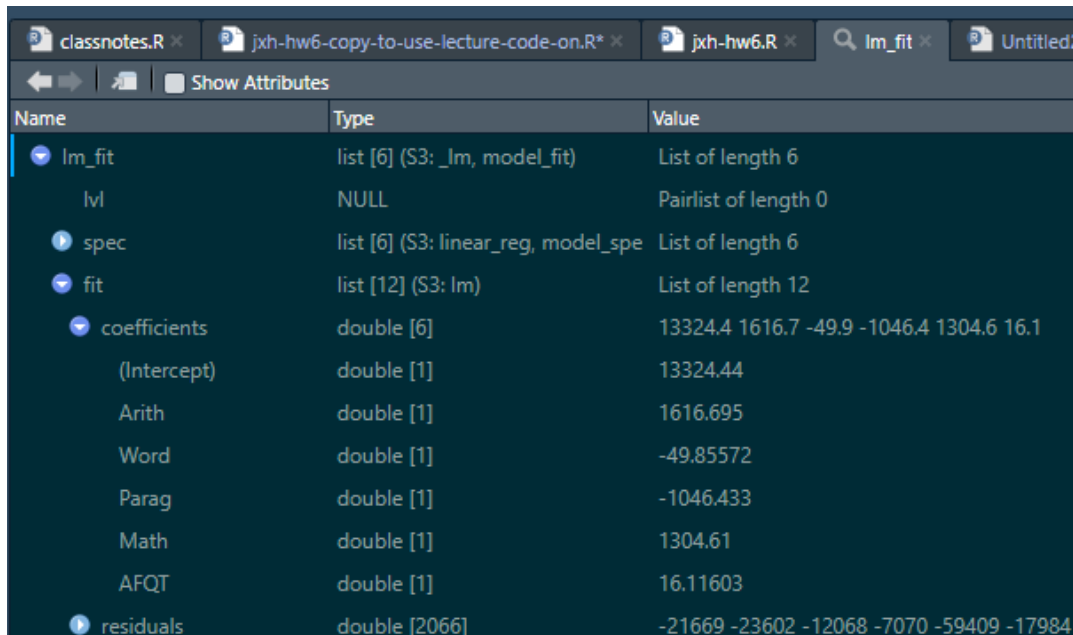
**Then, for product 2, select Supplier D at order size 3749 is more profitable.**



Same note about diagram here

## Part 2 – Linear Regression in R

Income data from 'income.csv' was loaded into R. This data is split 70/30 into training and testing sets. A multiple linear regression was performed using tidymodel's `linear_reg()` with engine 'lm' on training data. Resulting coefficients were:

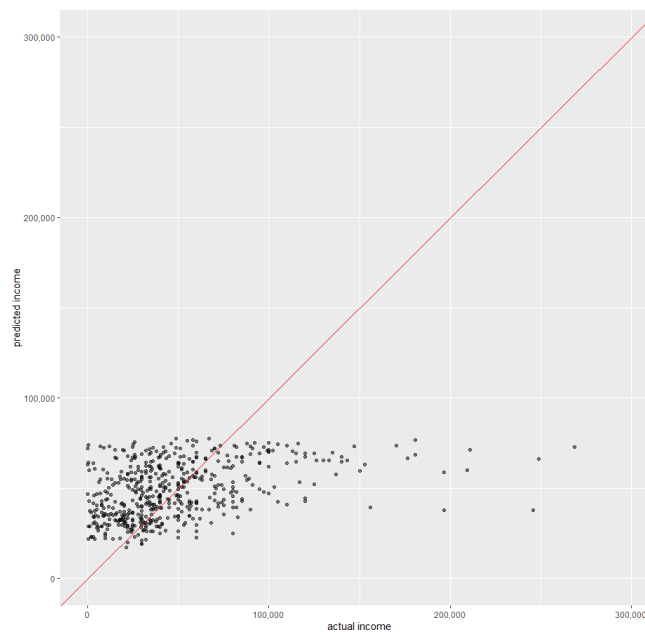


Name	Type	Value
lm_fit	list [6] (S3: _lm, model_fit)	List of length 6
lvi	NULL	Pairlist of length 0
spec	list [6] (S3: linear_reg, model_spe	List of length 6
fit	list [12] (S3: lm)	List of length 12
coefficients	double [6]	13324.4 1616.7 -49.9 -1046.4 1304.6 16.1
(Intercept)	double [1]	13324.44
Arith	double [1]	1616.695
Word	double [1]	-49.85572
Parag	double [1]	-1046.433
Math	double [1]	1304.61
AFQT	double [1]	16.11603
residuals	double [2066]	-21669 -23602 -12068 -7070 -59409 -17984

Fitted model was tested by `predict()`-ing using predictor variables in test data. These predictions are checked against Income2005 actuals of test subset. Predictions vs actual income are plotted. For readability, axes are forced to same scale and same x & y limits. A line was drawn uh, diagonally.

```
##### 1.4 - viz #####
#pdf(file="testplot.pdf")
ggplot(income_test_results,
       aes(x = Income2005, y=.pred)) +
  geom_point(alpha = 0.5) +
  geom_abline(color='red', linetype=1) +
  labs(x="actual income", y="predicted income") +
  scale_y_continuous(labels=comma, limits=c(0,300e3)) +
  scale_x_continuous(labels=comma, limits=c(0,300e3))
#dev.off()
```

Resulting plot is below.



### Part 2.1

R-squared value between predictions and actual 2005 income is **0.0721, via rsq()**. Little correlation between variation of actuals and predictions based on the fitted model.

RMSe (rms error) between predicted and actual **(rmse()) is 39574**. This would be average(rms) error between predictions and actual income.

The model does not seem to predict incomes beyond some 80,000, while test dataset has a few out there. The model tends to predict higher income than actual, with the mass above the line (higher predicted than actual), this may be due to high-income outliers in dataset.

### Part 2.2

For this study of incomes, a linear regression based on dimensions provided (test scores, afqt percentile rank) are not a very solid predictor of 2005 income. On further reading, the AFQT score mentioned may be military aptitude test<sup>1</sup>, which is based on the arithmetic, math, word, and paragraph sub-tests. Then, independence between predictor(?) (independent?) variables is in question. Shape of tests is in question, they might not normal distribution underneath. There are assumptions for multiple regression. F-test and QQ-plot might be used to check for these.

### Part 2.3

If it is military, other variables may help for better model: length of service, rank, enlisted/officer, disciplinary history, etc. If it is general population, common demographic variables (age, sex, gender, education level, geographic area, race/ethnicity) may improve model. More samples in the higher income range may help, as training data is sparse there and resulting fitted model does not predict high incomes.

---

<sup>1</sup> <https://www.indeed.com/career-advice/career-development/how-the-asvab-afqt-score-is-computed>

## Part 3 – Supply Chain Risk

### Task 3.1 – Historical Data

Task is to create and compare logistic, decision-tree, and random forest models to predict fraud for orders. Historical data `scmdatanew.csv` is imported, which has been already cleaned to have only non-nominal numerical predictor variables. Fraud column is identified as factor type (result we are trying to predict) and converted. The data are split 70/30 into training and testing data sets. A model of each is trained against training data set and attempts to predict the test data set. Of note when predicting, a warning message appeared:

```
> # predicting probability
> prob_pred = predict(logistic_fit,
+                      new_data = scmdata_test,
+                      type="prob")
warning message:
In predict.lm(object, newdata, se.fit, scale = 1, type = if (type ==
  prediction from a rank-deficient fit may be misleading
>
```

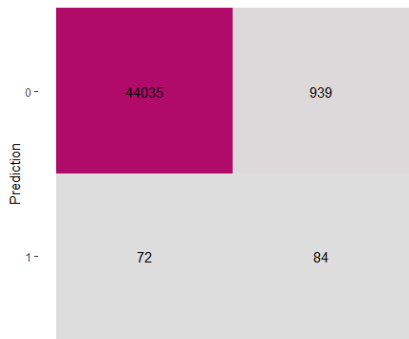
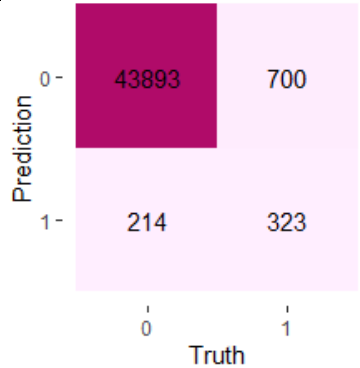
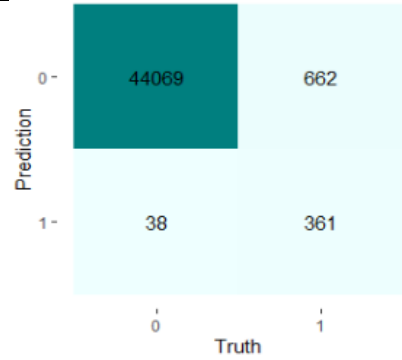
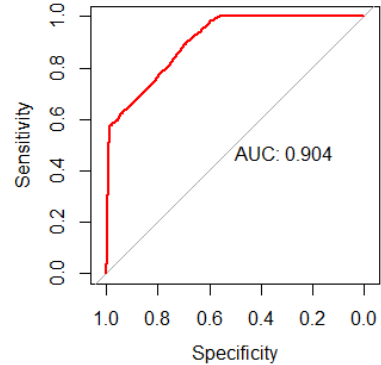
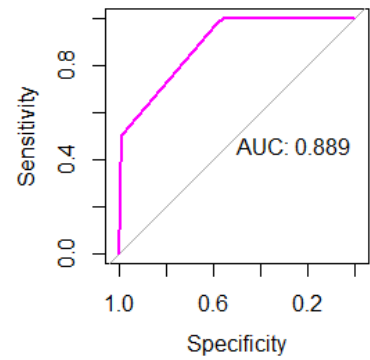
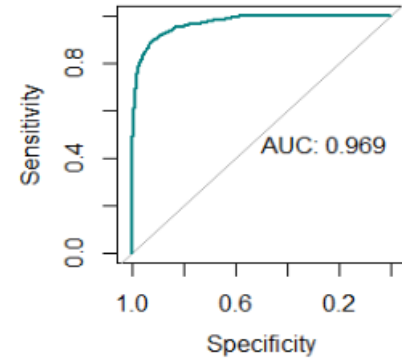
On internet search, this may be due to small number of rows, or columns of training data having correlated values<sup>2</sup>. From `glimpse()` of the data, two columns have values close to one another. An assumptions of some regression models are that the predictor variables are not correlated (so they ought move independently).

```
> glimpse(scmdata)
Rows: 180,519
Columns: 26
$ Days.for.shipping..real.      <int> 3, 5, 4, 3, 2, 6, 2,
$ Days.for.shipment..scheduled. <int> 4, 4, 4, 4, 4, 4, 1,
$ Sales.per.customer           <dbl> 314.64, 311.36, 309.
$ Category.Id                  <int> 73, 73, 73, 73, 73,
$ Customer.Id                   <int> 20755, 19492, 19491,
$ Customer.Zipcode              <int> 725, 725, 95125, 900
$ Department.Id                 <int> 2, 2, 2, 2, 2, 2,
$ Order.Id                      <int> 77202, 75939, 75938,
$ Order.Item.Discount           <dbl> 13.11, 16.39, 18.03,
$ Order.Item.Discount.Rate      <dbl> 0.04, 0.05, 0.06, 0.
$ Order.Item.Id                 <int> 180517, 179254, 1792
$ Order.Item.Product.Price      <dbl> 327.75, 327.75, 327.
$ Order.Item.Profit.Ratio       <dbl> 0.23, 0.88, 0.88,
$ Order.Item.Quantity           <int> 1, 1, 1, 1, 1, 1,
$ Sales                         <dbl> 327.75, 327.75, 327.
$ Order.Item.Total              <dbl> 314.64, 311.36, 309.
$ Order.Profit.Per.Order        <dbl> 91.25, -249.09, -247
$ Product.Card.Id               <int> 1360, 1360, 1360, 13
$ Product.Category.Id           <int> 73, 73, 73, 73, 73,
$ Product.Price                 <dbl> 327.75, 327.75, 327.
$ order_yr                      <int> 2018, 2018, 2018, 20
$ order_month                   <int> 1, 1, 1, 1, 1, 1,
$ order_day                     <int> 2, 5, 5, 5, 5, 5,
$ order_hour                    <int> 22, 12, 12, 11, 11,
$ late_delivery                 <int> 0, 1, 0, 0, 0, 0, 1,
$ fraud                         <int> 0, 0, 0, 0, 0, 0, 0,
```

<sup>2</sup> <https://www.statology.org/prediction-from-rank-deficient-fit-may-be-misleading/>



Results are below, as confusion matrix, performance statistics, and ROC/AUC..

	LOGREGRESSION	TREE	FOREST
PERFORMANCE STATS	<pre>&gt; all_metrics(scmdata_results, +             truth=fraud, +             estimate=.pred_class) # A tibble: 4 × 3   .metric .estimator .estimate   &lt;chr&gt;   &lt;chr&gt;     &lt;dbl&gt; 1 accuracy binary    0.978 2 sensitivity binary    0.998 3 specificity binary    0.0821 4 recall binary    0.998</pre>	<pre>&gt; all_metrics(scmdata_results_tree, +             truth=fraud, +             estimate=.pred_class) # A tibble: 4 × 3   .metric .estimator .estimate   &lt;chr&gt;   &lt;chr&gt;     &lt;dbl&gt; 1 accuracy binary    0.980 2 sensitivity binary    0.995 3 specificity binary    0.316 4 recall binary    0.995</pre>	<pre>&gt; all_metrics(scmdata_results_forest, +             truth=fraud, +             estimate=.pred_class) # A tibble: 4 × 3   .metric .estimator .estimate   &lt;chr&gt;   &lt;chr&gt;     &lt;dbl&gt; 1 accuracy binary    0.984 2 sensitivity binary    0.999 3 specificity binary    0.353 4 recall binary    0.999</pre>
CONFUSION MATRIX			
ROC + AUC			

### Explanations and Suggestions:

Performance statistics (accuracy, sensitivity, specificity, recall) are based values on the confusion matrix (plot of model predictions vs actual results). Confusion matrix's four quadrants are based on model prediction and whether it matches actuality of test data: true negative (0 0), false negative (0 1), false positive (1 0), true positive (1 1). Performance statistics associated are:

- Accuracy: correct predictions over entire space, and is proportion of how often model gets it right.
- Recall/sensitivity is how often the model would predict (catch) actual positive cases.  $(tp / (tp+fn))$
- Specificity is the inverse of above, how often model correctly predicts negative cases  $(tn / (tn+fp))$
- Precision is how often model's positive predictions are true  $(tp / (tp + fp))$

Depending on application the cost of false positives and false negatives differ. Cost of punishing/hassling innocent people to jail vs. what-if's of significant crimes. Cost of invasive medical intervention vs missing diagnoses.

Further, ROC and AUC are related, with roc being a plot, as threshold for positive predictions are dialed up. Picture a motion sensor light on a porch outfitted with sensitivity dial. At low threshold, motion sensor would correctly alert on activity, but nuisance light-ups (false positives) are high. At high threshold, nuisance light-ups are low, however often the porch light would leave one fumbling in the dark with their keys (false negatives). ROC plots sensitivity and specificity of the system, and area under the curve is associated with the goodness of the model. (closer to 1 the better, at AUC = 0.5, it's no better than coinflip).

**In all measures, the random forest model is superior.**

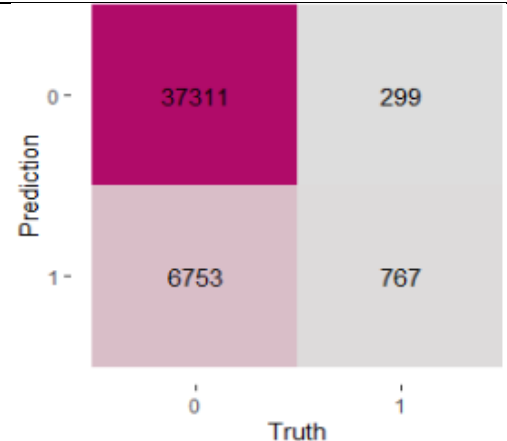
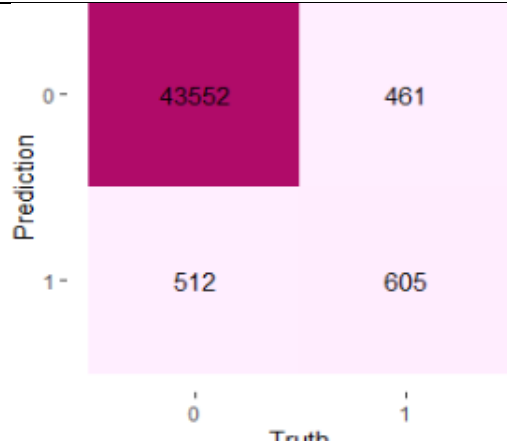
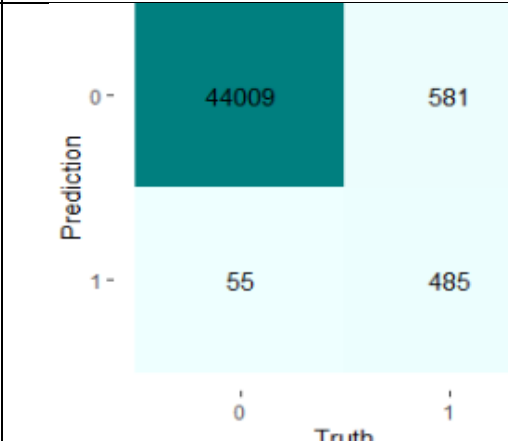
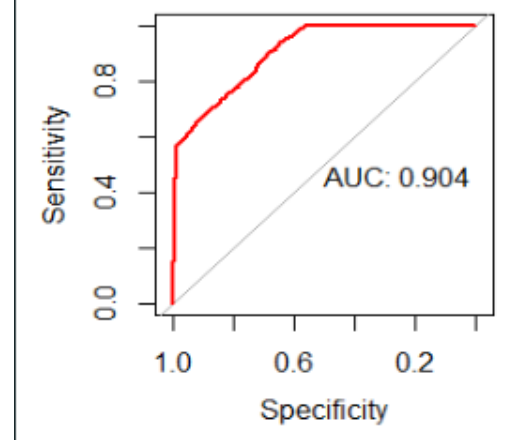
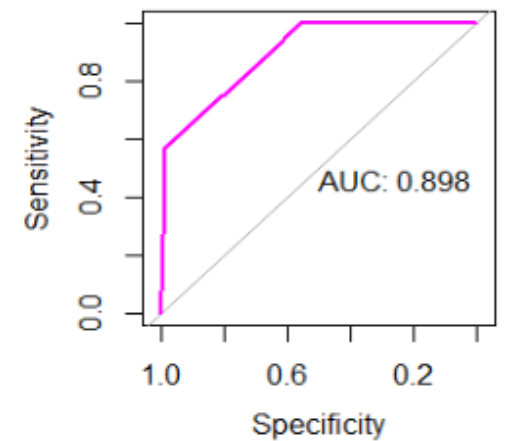
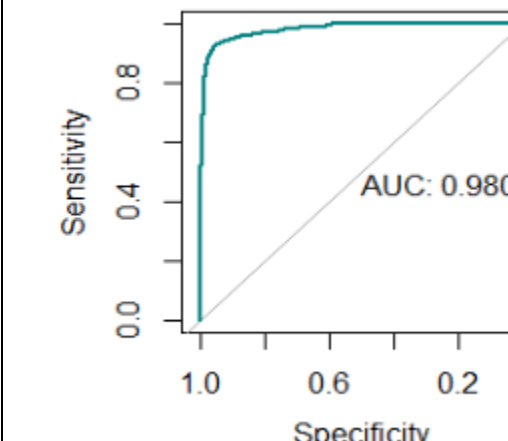
### Task 3.1 – Historical and Simulated Data

As historical data has low proportion of fraud transactions, the models generated are affected. Then, process of oversampling from ROSE package is used to generate simulated fraud transactions, based on code provided by assignment instructions, and an augmented training data set is used to train the logistic, decision-tree, and random forest models.

These are tested against original test data set, and performance evaluated below: conf. matrix, performance statistics (accuracy, recall, specificity, precision), ROC/AUC.



Results are below, as confusion matrix, performance statistics, and ROC/AUC..

	LOGREGRESSION	TREE	FOREST
PERFORMANCE	1 accuracy binary 0.844 2 sensitivity binary 0.847 3 specificity binary 0.720 4 recall binary 0.847	1 accuracy binary 0.978 2 sensitivity binary 0.988 3 specificity binary 0.568 4 recall binary 0.988	1 accuracy binary 0.986 2 sensitivity binary 0.999 3 specificity binary 0.455 4 recall binary 0.999
CONFUSION MATRIX			
ROC + AUC			

Again, depending on costs of false and true positives, the specificity and recall rates may be relevant.

Random forest model performs better in regards in all but specificity, this may be due to paucity still of fraud.

### Task 3.3 – Selected Questions

#### 1. Significant change?

. Specificity is lower for models trained on over-sampled data. Result is that models on over-sampled data would be less likely to correctly predict a true negative, as proportion all negatives.

#### 2. Model to recommend?

. The random forest model trained off the over-sampled data is improved over original training data in all respects. Differences are still very slight. It would depend on the cost of false-positives or false-negatives. Pursuing fraud can be expensive in general, and if a false-positive, relationship with customer can be impacted. With false-negatives, fraud is not discovered, and the loss is unabated. The decision would need to weigh the costs/benefits and probabilities between missing fraud and wrongly accusing fraud.

#### 3. Disadvantages of proposed model?

. Random forest of over-sampled training data has lower specificity than other over-sampled models. If fraud is occurring, random forest's sensitivity would catch only  $TP/(FN+TP) = 485/(485+581)$ , about half-and-half. However, its less likely to predict fraud when there isn't (lower false-positive rate).

#### 4. ROC and AUC useful measure? Why?

. See Explanation a couple sections above with the motion sensor light. In short, ROC curve is plot of sensitivity vs specificity (effectively true positive rate vs false positive rate per towardsdatascience article)<sup>3</sup> as threshold for "positive" prediction increases. Area under the curve is proportion of correct answers overall. A diagonal line represents a system doing no better than random guessing, and a filled curve (auc=1) represents perfect system able to correctly predict positives, and with no false-positives.

---

<sup>3</sup> <https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb>