STAT418 Homework 4

Siyuan Li, 904884144

6/7/2017

Introduction

The dataset is “Adult (https://archive.ics.uci.edu/ml/datasets/adult)” dataset from UCI Machine Learning Repository. This dataset contains features that are associated with predicting whether annual income will exceed 50k US Dollars.

After data loading, cleaning missing rows, converting the response variable, the data contains 45222 obervations, 13 variables, with 34014 negative cases, and 11208 positive cases.

Exploratory Analysis

Summary Statistics

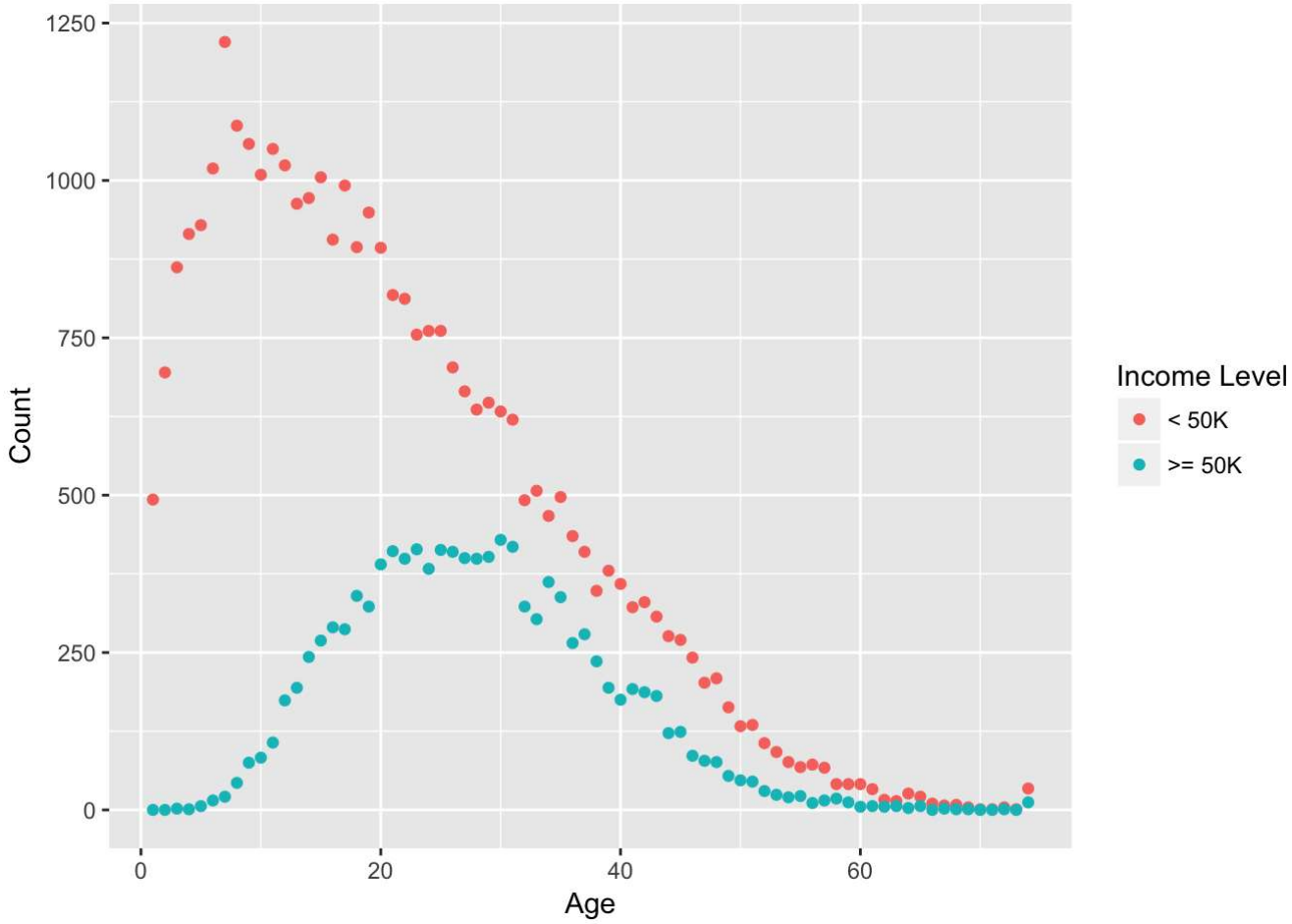
The variables that are taken into account are: Age, Work Class, Education Level, Marital Status, Occupation, Relationship in Marital Status, Race, Sex, Capital Gain(through investment), Capital Loss, Hours per Week, Ethnicity, and Income.

The response variable in the data is “income” (being kept as in numeric for the moment).

```
##       age                    workclass            education
##  Min.   :17.00   Federal-gov     : 1406   HS-grad     :14783
##  1st Qu.:28.00   Local-gov       : 3100   Some-college: 9899
##  Median :37.00   Private         :33307   Bachelors   : 7570
##  Mean   :38.55   Self-emp-inc    : 1646   Masters     : 2514
##  3rd Qu.:47.00   Self-emp-not-inc: 3796   Assoc-voc   : 1959
##  Max.   :90.00   State-gov       : 1946   11th        : 1619
##                  Without-pay     :   21   (Other)     : 6878
##                   marital                occupation
##  Never-married        :14598   Craft-repair   : 6020
##  Married-civ-spouse   :21055   Prof-specialty : 6008
##  Married-spouse-absent:  552   Exec-managerial: 5984
##  Married-AF-spouse    :   32   Adm-clerical   : 5540
##  Separated            : 1411   Sales          : 5408
##  Divorced             : 6297   Other-service  : 4808
##  Widowed              : 1277   (Other)        :11454
##          relationship                    race           sex
##  Husband       :18666   Amer-Indian-Eskimo:  435   Female:14695
##  Not-in-family :11702   Asian-Pac-Islander: 1303   Male  :30527
##  Other-relative: 1349   Black             : 4228
##  Own-child     : 6626   Other             :  353
##  Unmarried     : 4788   White             :38903
##  Wife          : 2091
##
##     capgain         caploss             hpw                ethnicity
##  Min.   :    0   Min.   :   0.00   Min.   : 1.00   United-States:41292
##  1st Qu.:    0   1st Qu.:   0.00   1st Qu.:40.00   Mexico       :  903
##  Median :    0   Median :   0.00   Median :40.00   Philippines  :  283
##  Mean   : 1101   Mean   :  88.59   Mean   :40.94   Germany      :  193
##  3rd Qu.:    0   3rd Qu.:   0.00   3rd Qu.:45.00   Puerto-Rico  :  175
##  Max.   :99999   Max.   :4356.00   Max.   :99.00   Canada       :  163
##                                                    (Other)      : 2213
##      income
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2478
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

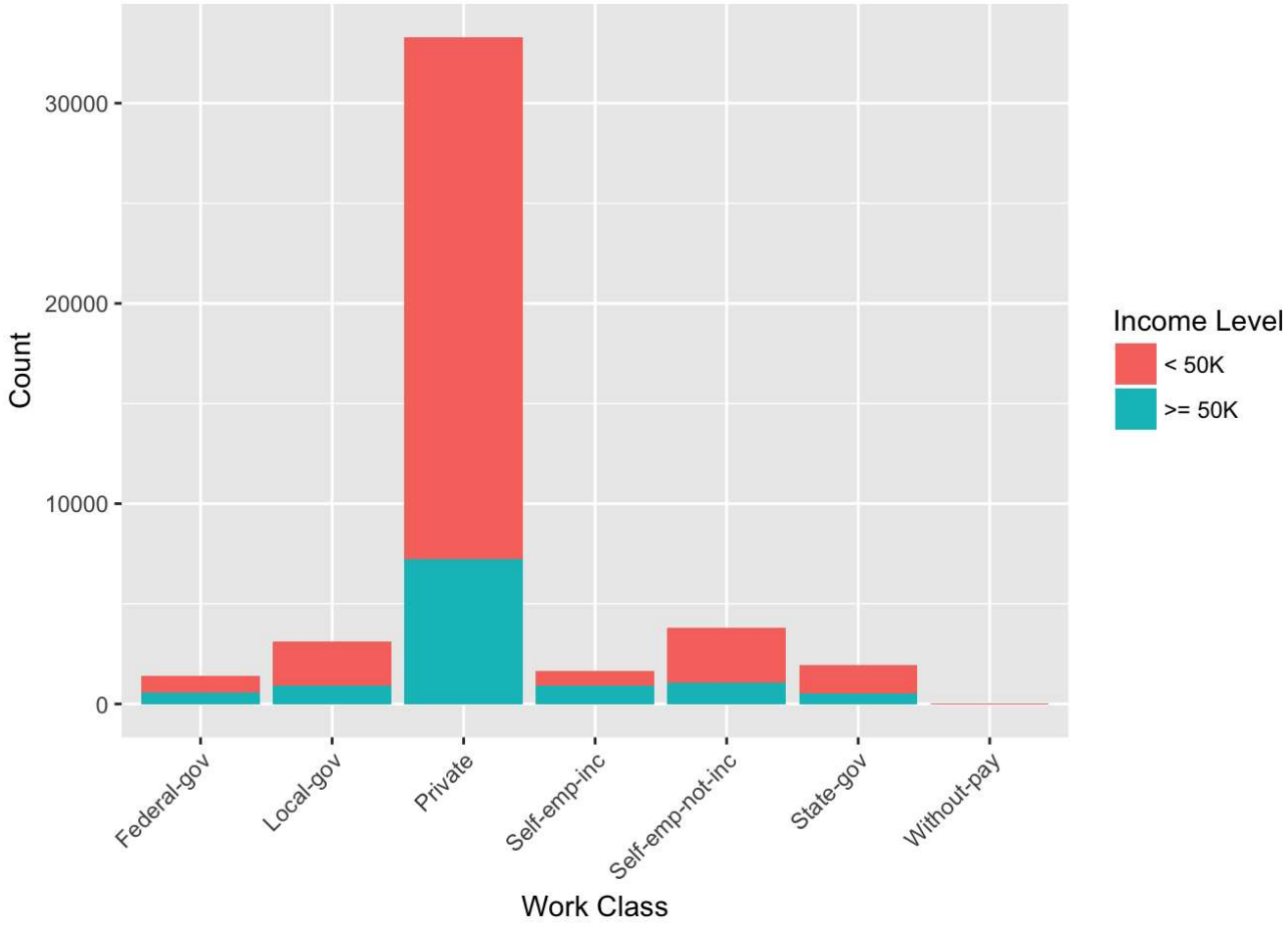
Age

Dot plot shows that there are more samples in the sub 50k income range, and while lower age group present more diverse income difference, higher age group does not show significant diversity. This is most likely the result of sample selection.



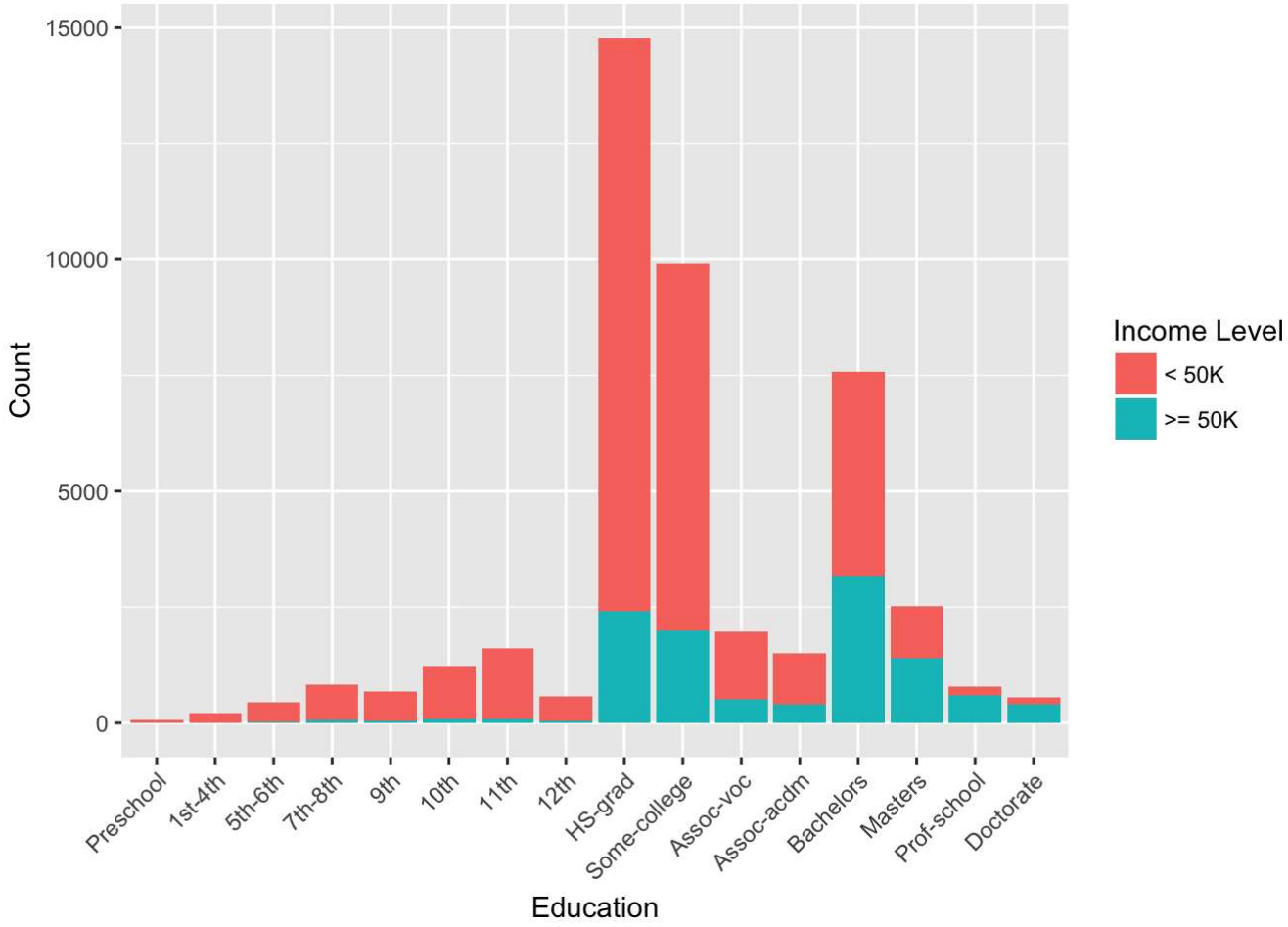
Work Class

Bar plot of Work Class shows that our sample mostly work in private sector. It also shows that while majority of the sample from private sector earns less than 50K annually, there must be other more significant factors that affects salary, since there is no cutoff anywhere in the bar plot where a specific work class dictates the income.



Education

Bar plot of education shows that most of our sample's level of education is between high school and bachelors degree. We can also see from those who graduates from professional schools or has a doctorate degree that samples from these groups tend to have higher level of income. Speculate that the higher the education level, the higher chance of getting a higher level pay.

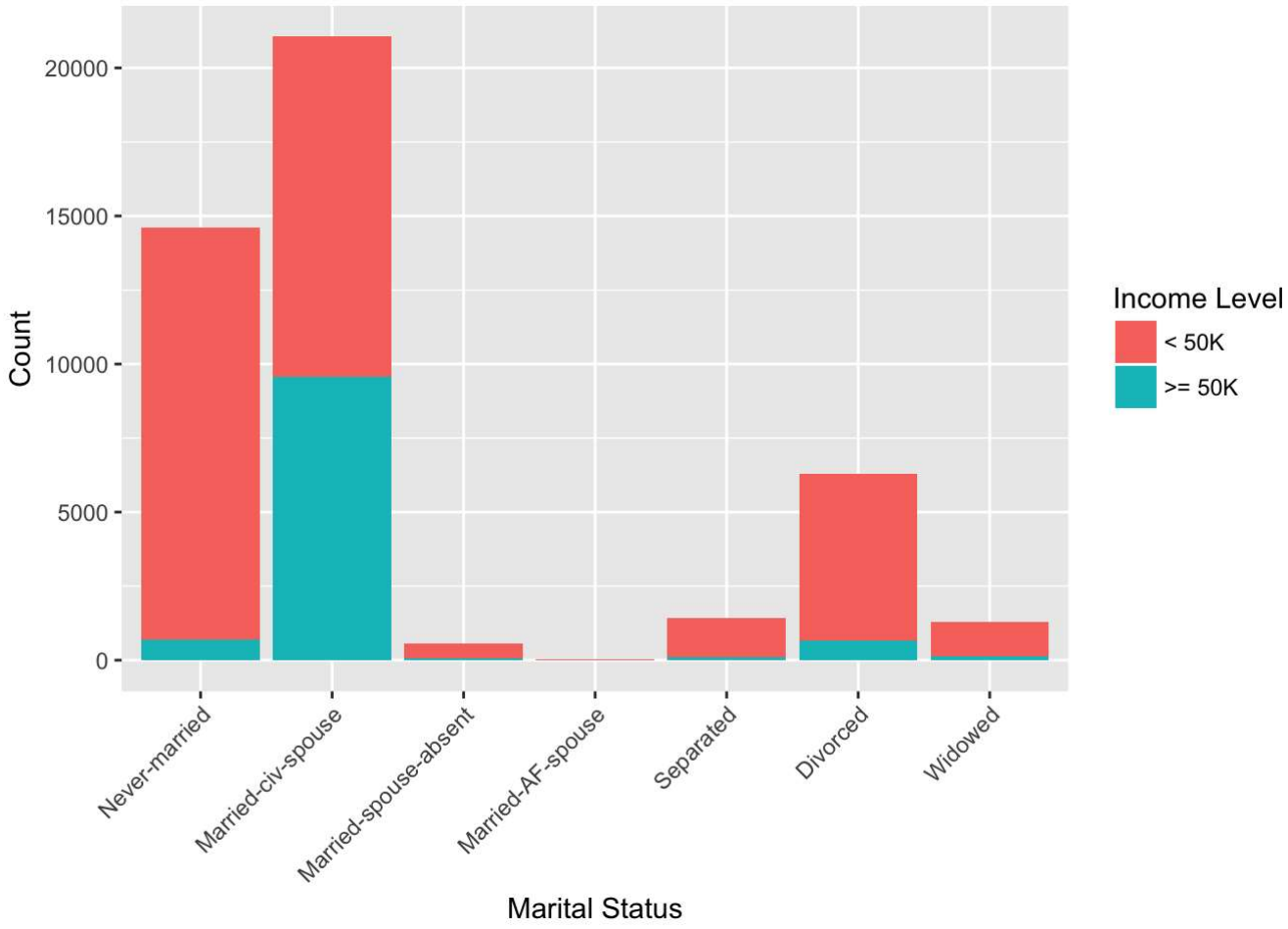


Calculating the percentage of sample that has higher income level based on education, confirms our assumption.

```
##                  0    1
## Preschool       71    1 0.0139
## 1st-4th        214    8 0.0360
## 5th-6th        427   22 0.0490
## 7th-8th        768   55 0.0668
## 9th            638   38 0.0562
## 10th          1141   82 0.0670
## 11th          1530   89 0.0550
## 12th           534   43 0.0745
## HS-grad      12367 2416 0.1634
## Some-college  7909 1990 0.2010
## Assoc-voc     1455  504 0.2573
## Assoc-acdm    1109  398 0.2641
## Bachelors     4392 3178 0.4198
## Masters       1121 1393 0.5541
## Prof-school    193  592 0.7541
## Doctorate      145  399 0.7335
```

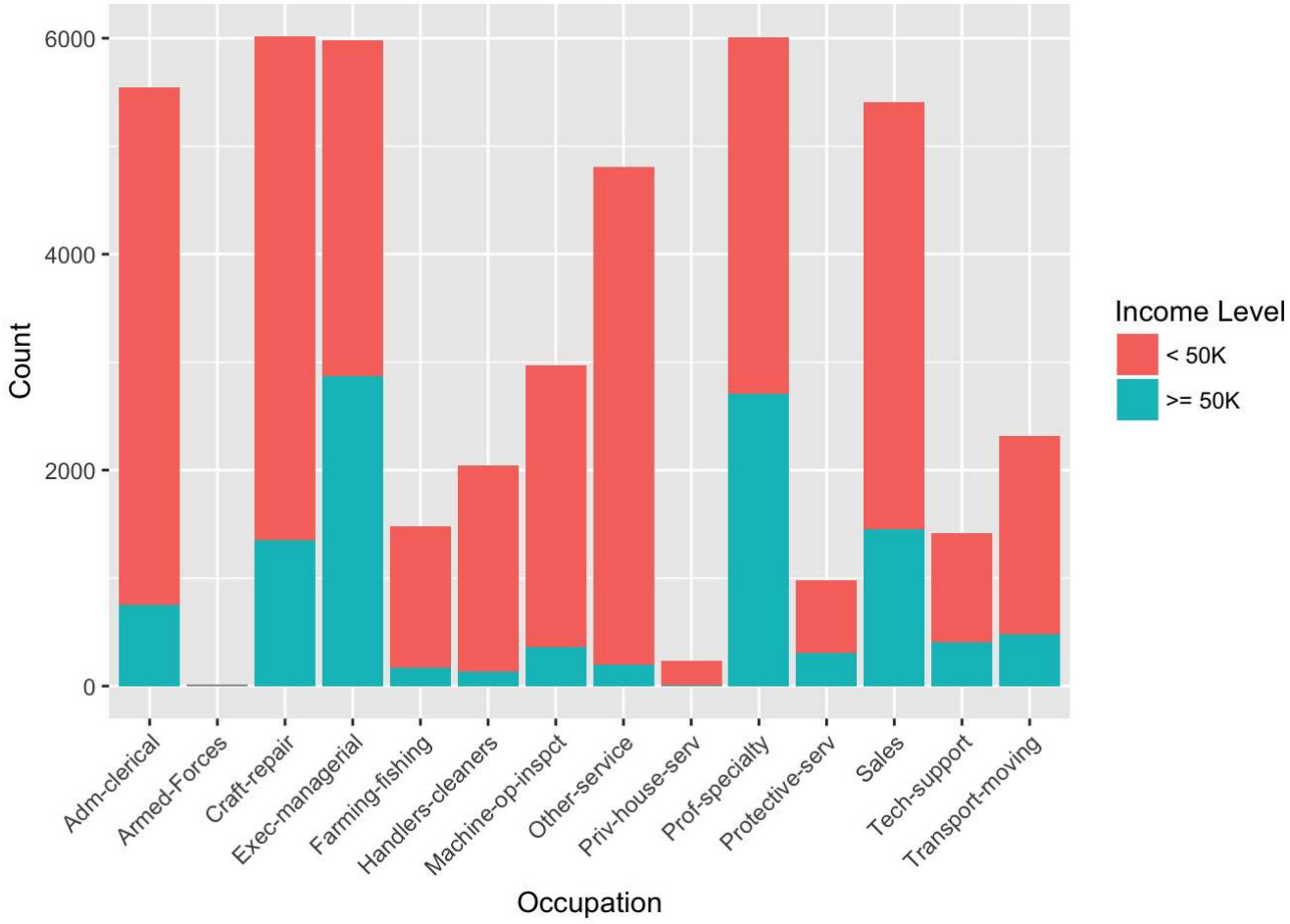
Marital Status

Bar plot of Marital Status shows that samples that are single has a higher tendency to have lower level of income, while married samples tend to have a balanced spread.



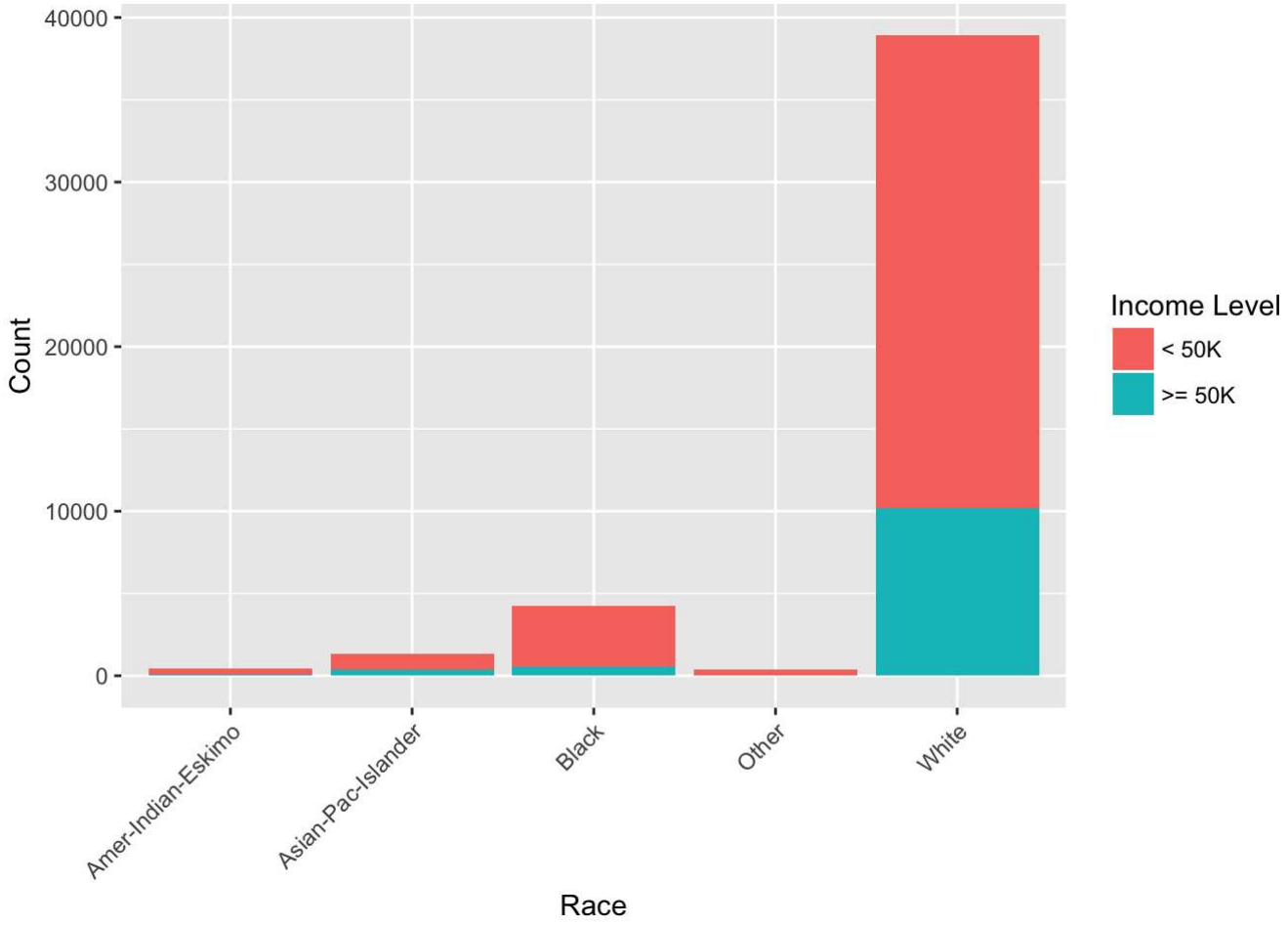
Occupation

Bar plot of occupation reveals that titles like executives, managerial, or specialist positions tend to have higer level of income comparing to others. This is quite obvious when comparing crafting and repairing titles with executives and managerial titles. While they have similar sample count, executives and managerial positions has significantly higher percentage of higher level of pay.



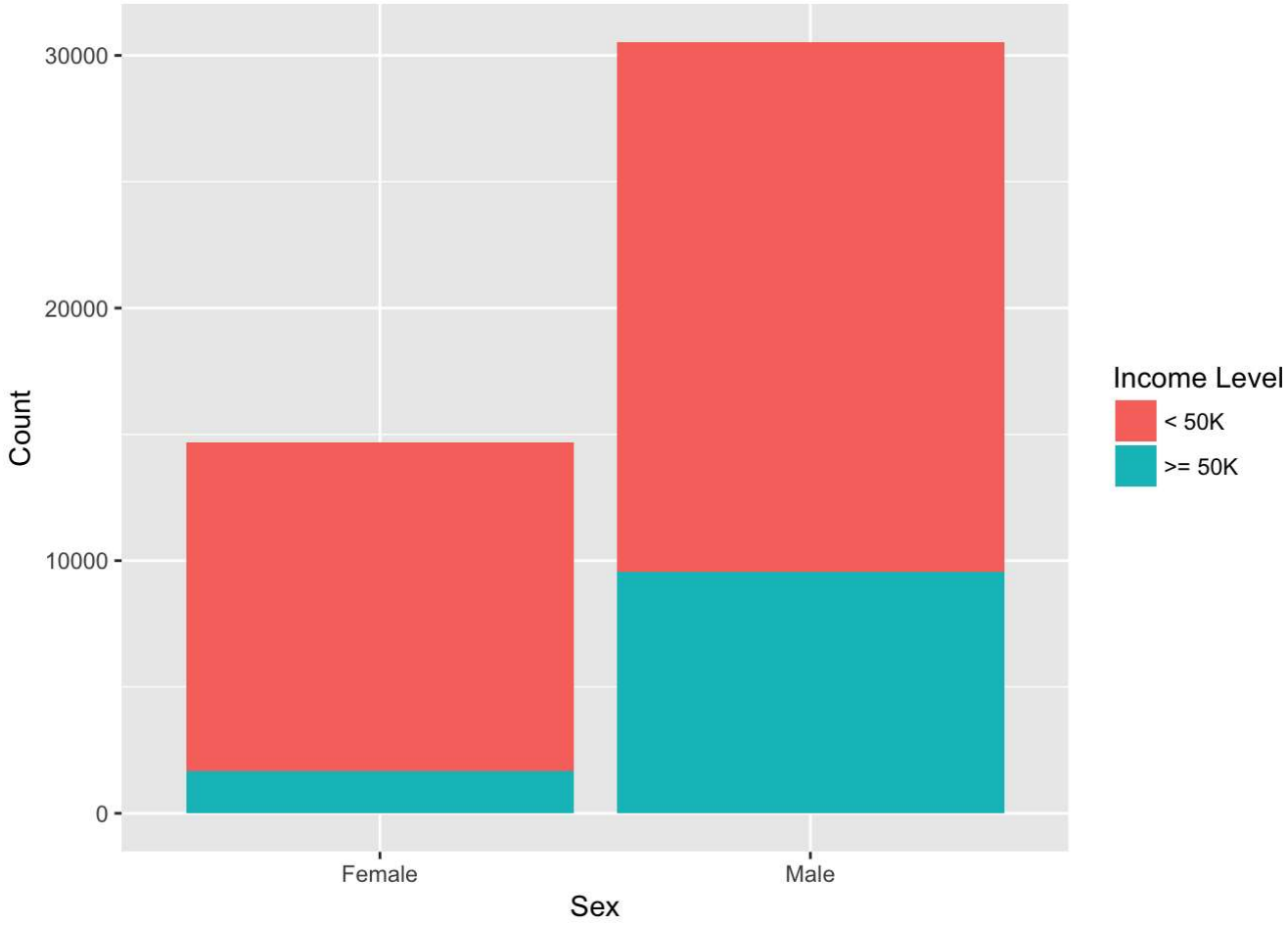
Race

Bar plot of race shows that majority of our sample are white.



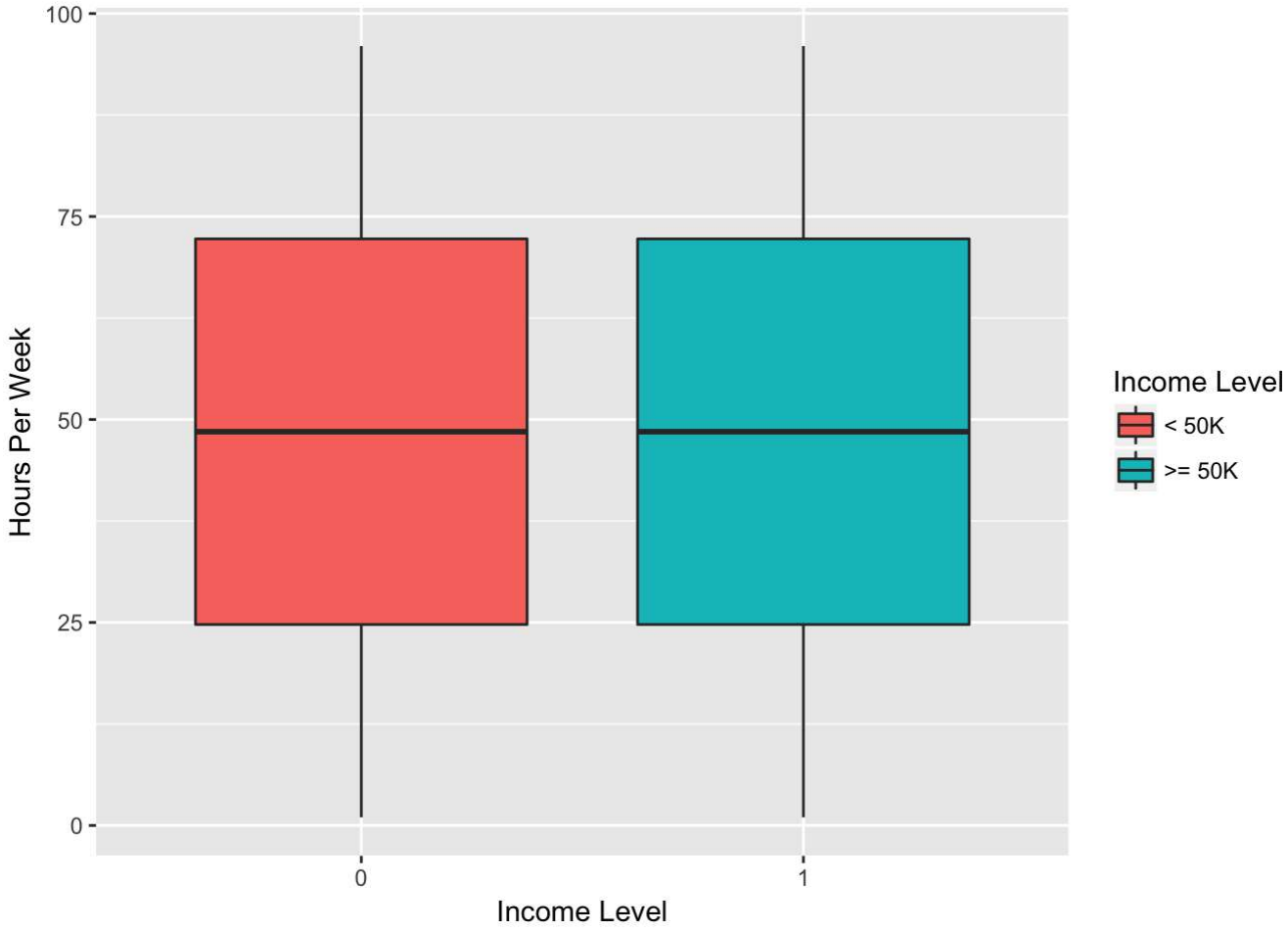
Sex

Bar plot of Sex shows, while the amount of male samples in the data is twice of female, it is still significant that male samples tend to have higher level of income.



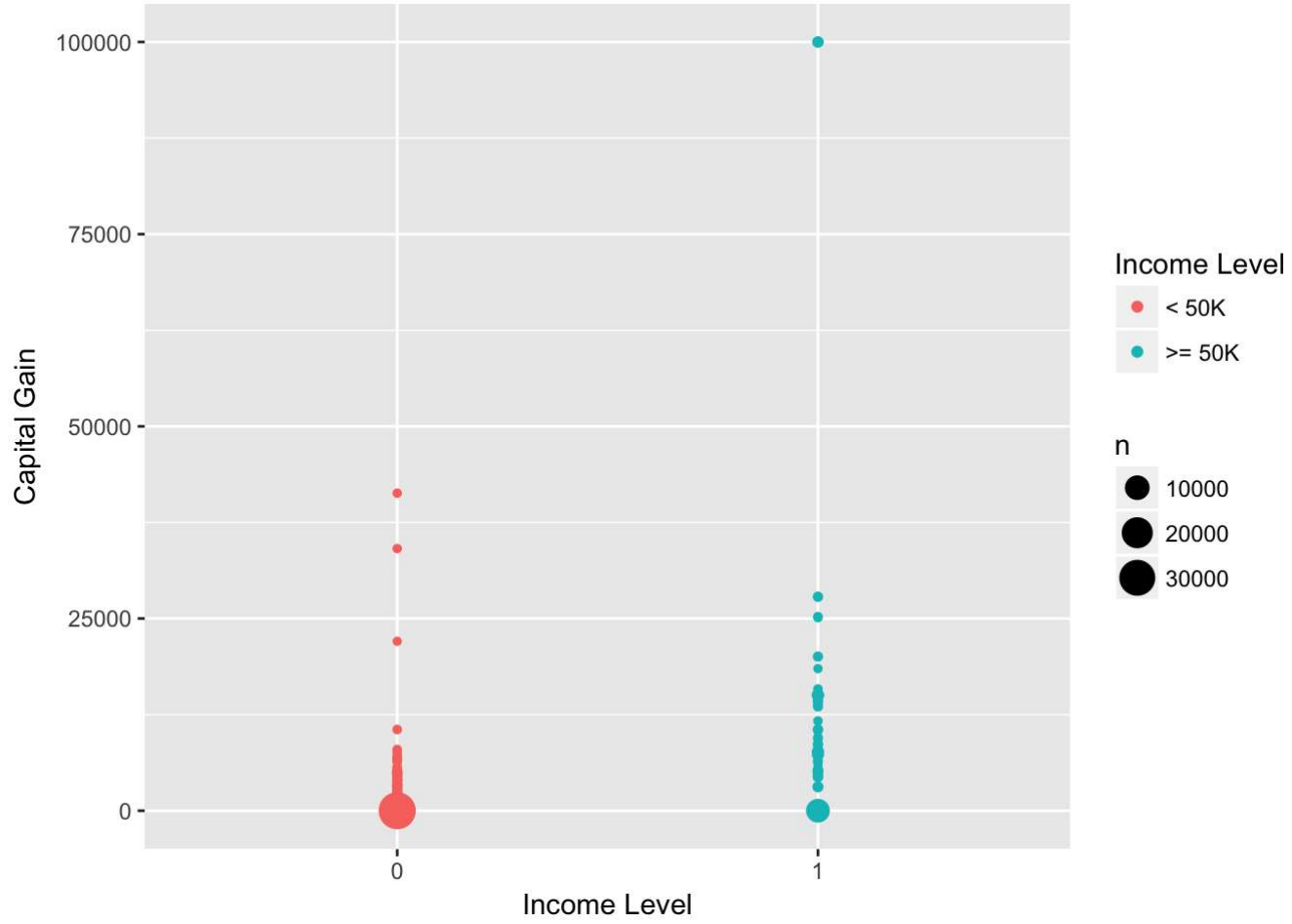
Hours per Week

Boxplot of Hours per Week show that hours per week is not a significant predictor with association to level of income.



Capital Gain

Dot plot of capital gain through investment shows that sample with lower level of income tend to have low capital gain, while sample with higher level of income tend to have more investment returns.



Statistical Analysis

H2O is an awesome framework to base the analysis on, because it fully utilizes the computing power of my machines and provide various options for validation, tuning and optimization.

Logistic Regression

The first algorithm is logistic regression. The initial run uses lambda equals 0.

Total run time:

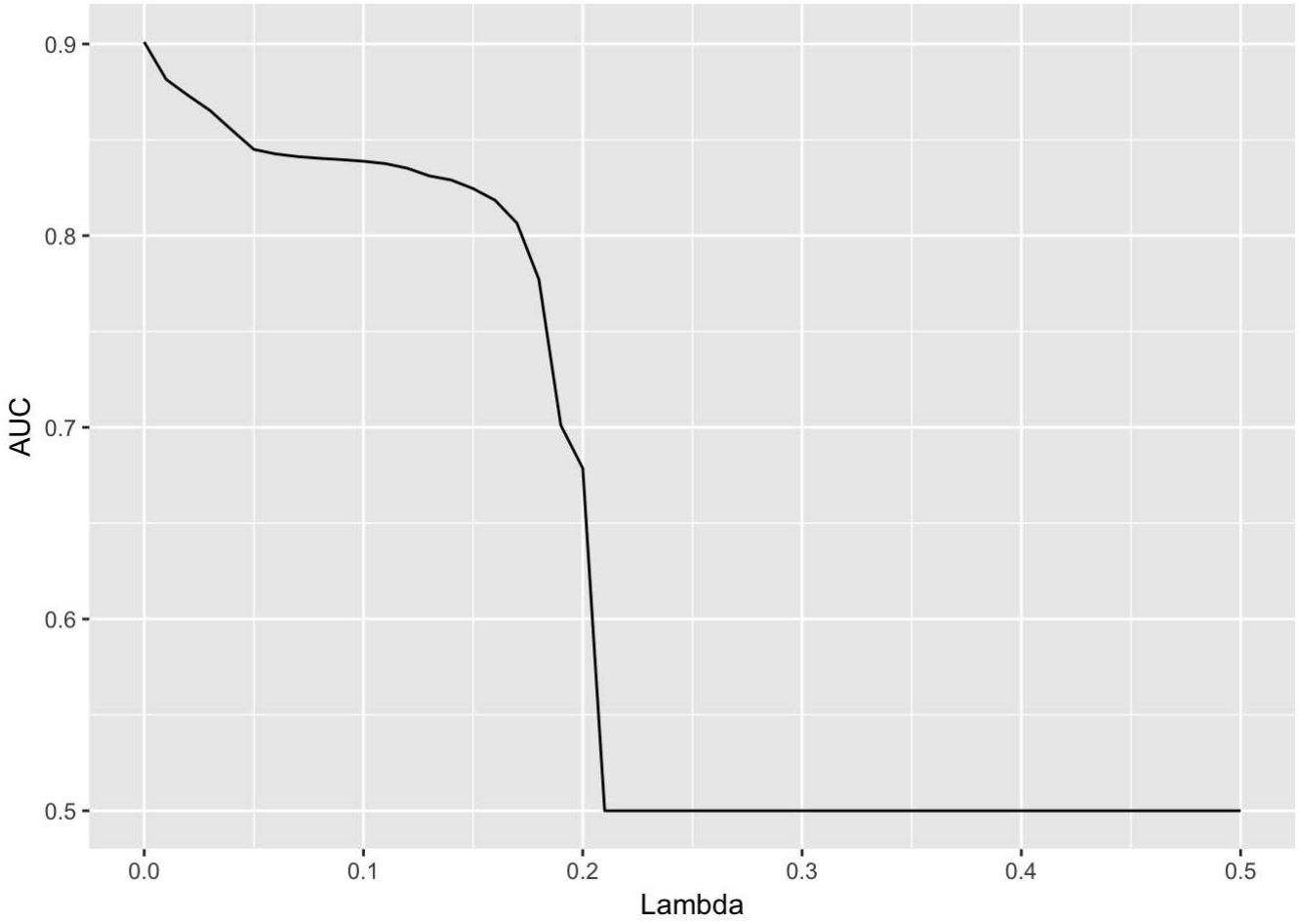
```
##    user  system elapsed
##   0.164   0.005   1.423
```

The ROC curve shows that the logistic model is a really good fit for starters. Model Selection based on ROC curve is used to determine the trade-off between the sensitivity (True Positive) of the model versus the specificity (False Positive Rate), through which aims to reduce the false positive rate while imrpoving upon true positive rate, increasing the area under the curve (AUC). Which in other words, the higher the AUC, the better the model fits.

The resulting test set AUC is:

```
## [1] 0.901045
```

Bootstrap lambda and validate if lambda = 0 is indeed the best. Let the training algorithm run through 51 lambdas from 0 to 0.5 and return a list of all the AUC with corresponding lambda.



Bootstrapping confirms that the best lambda is 0, or at least very close to 0.

Random Forest

The second algorithm is random forest. Initial model defaults to 100 trees.

Total run time:

```
##    user  system elapsed
##   0.146   0.008   9.299
```

The ROC curve:

The initial test set AUC is:

```
## [1] 0.9124391
```

Next, do hyperparamter tuning for random forest model. The mothod employed is random grid search since it is efficient in determinig a somewhat close estimate to the best model without running through all the possible parameters.

Hyperparameters are set as:

- trees: 100, 200, 300, 400, 500;
- maximum tree depth: 10, 20, 30;
- maximum number of variables considered for tree split: 2, 3, 4.

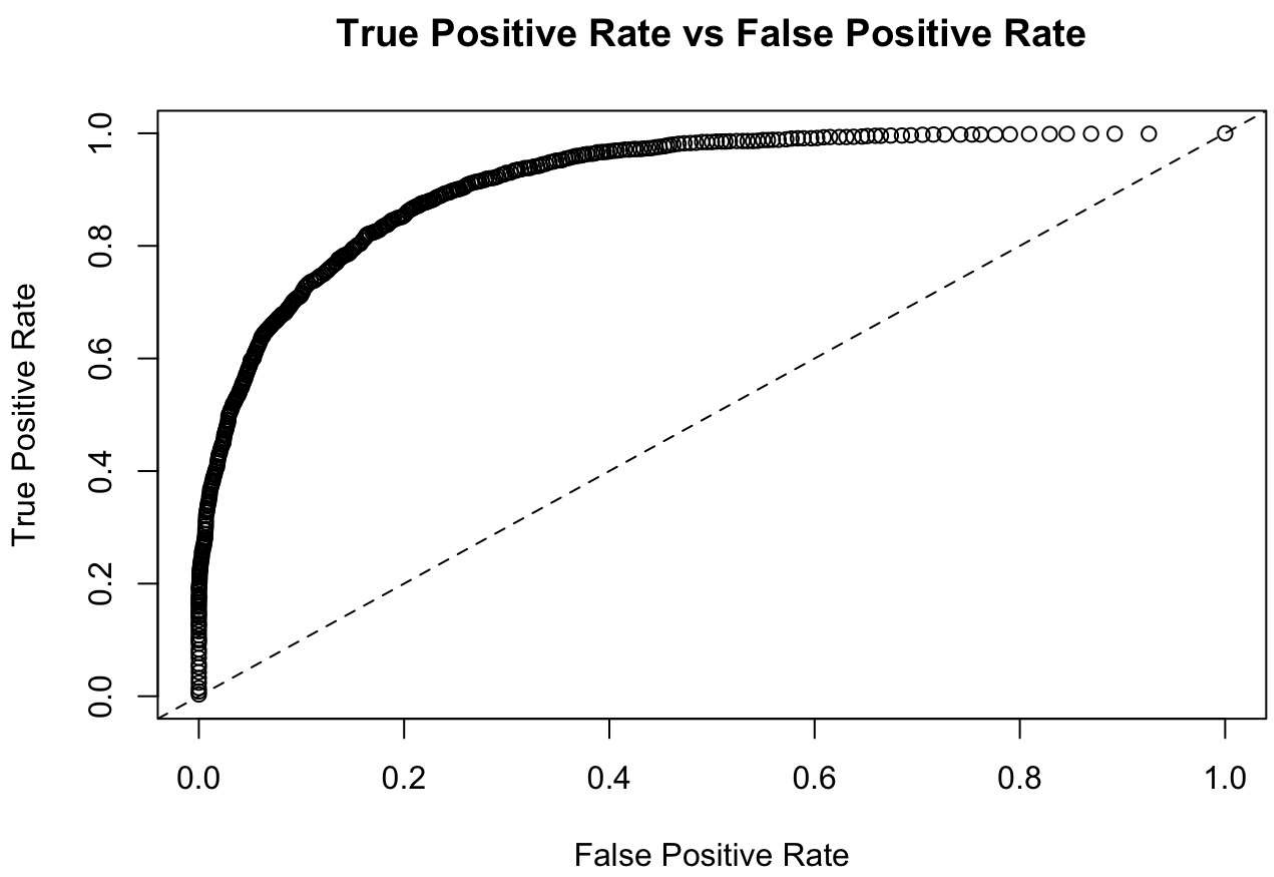
Limit the maximum training run time to 5 minutes and maximum number of trained models to 20. Include early stopping mechanism with AUC as stopping metric and tolerance of 0.

Total run time:

```
##    user  system elapsed
##   1.638   0.148 302.351
```

The model with the most AUC has 400 trees, maximum tree depth of 20 and maximum number of variables considered for tree split of 2.

The ROC curve of the best model:



The resulting test set AUC is:

```
## [1] 0.9150001
```

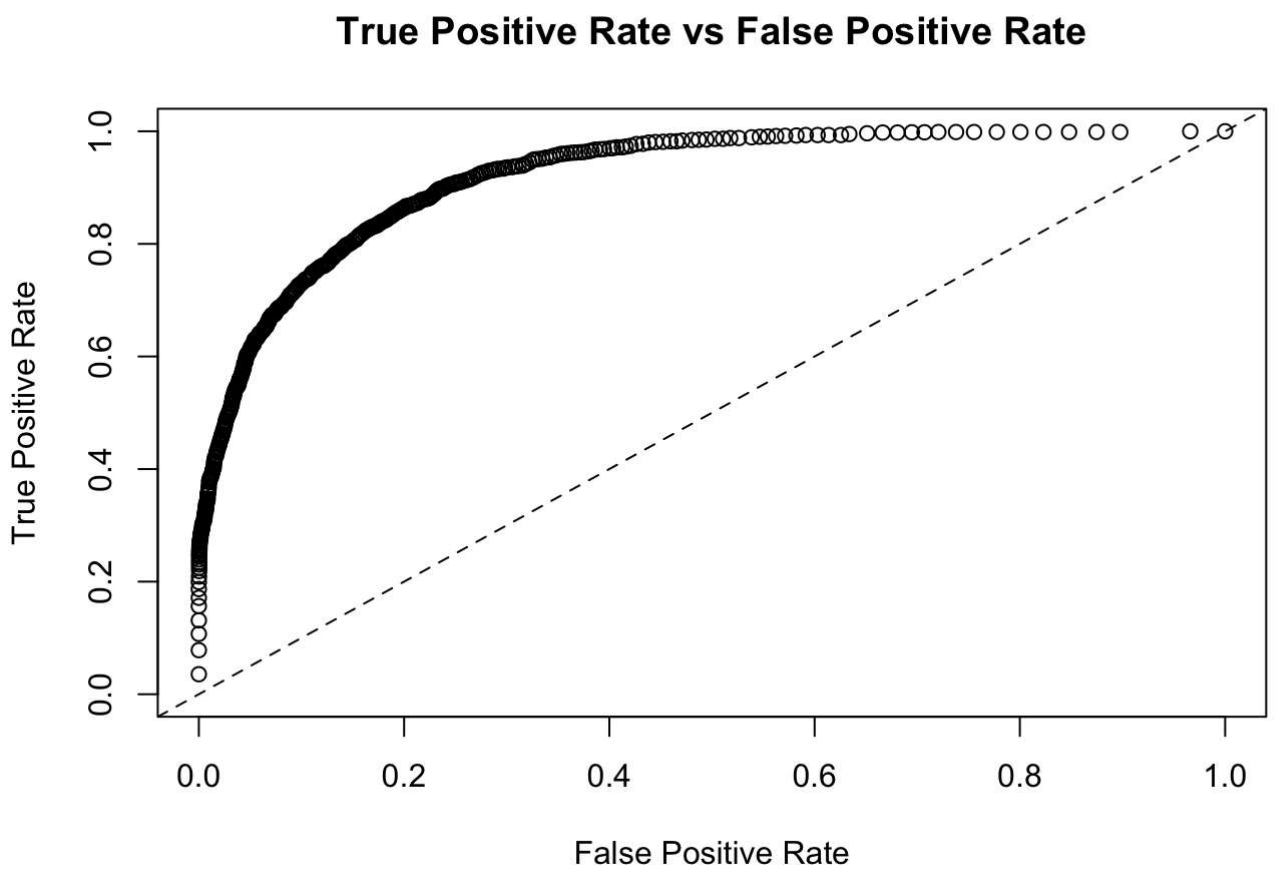
Gradient Boosting Machine

The third algorithm is gradient boosting machine. Initial model defaults to 100 tress, and maximum tree depth of 10.

Total run time:

```
##    user  system elapsed
##   0.142   0.007   7.267
```

The ROC curve:



The initial test set AUC is:

```
## [1] 0.9194251
```

Also do a hyperparameter tuning for gradient boosting machine model.

Hyperparameters are set as:

- number of trees: 300, 400, 500;
- maximum tree depth: 10, 20, 30;
- minimum observations per leaf: 1, 5, 10, 20, 50;
- learning rate: 0.01, 0.03, 0.05, 0.07, 0.1;
- learning rate scaler: 0.99, 0.995, 1.

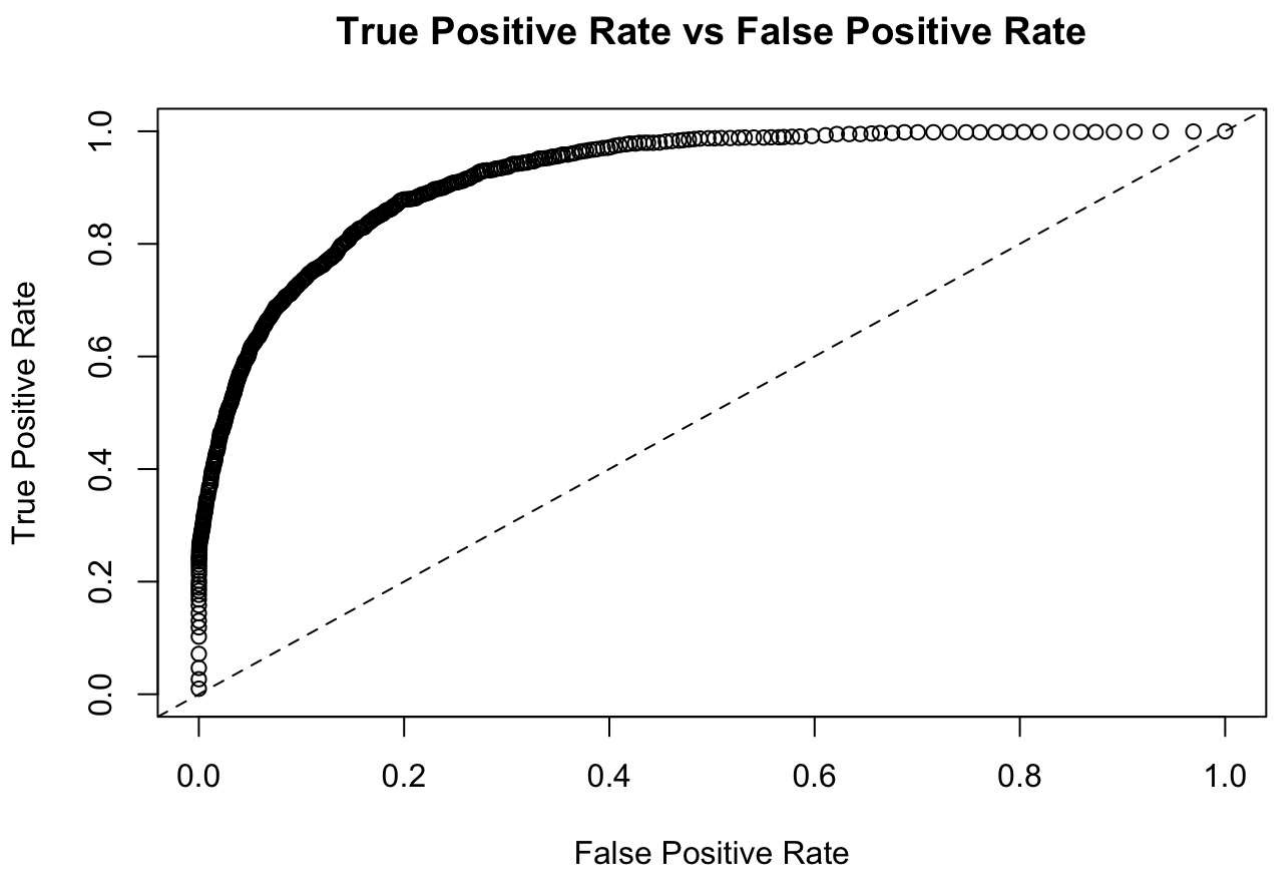
Limit the maximum training run time to 10 minutes and maximum number of trained models to 50. Considering the amount of calculation involved in gradient boosting machine, guessing we will reach the maximum run time before train through 50 models. Include early stopping mechanism with AUC as stopping metric and tolerance of 0.

Total run time:

```
##    user  system elapsed
##   3.294   0.292 605.035
```

The model with the most AUC has 300 trees, maximum tree depth of 10, minimum observations per leaf of 50, learning rate of 0.05 and learning rate scaler of 1.0.

The ROC curve of the best model:



The resulting test set AUC is:

```
## [1] 0.9211464
```

Neural Network

The fourth algorithm is neural networks. *Initial* model defaults to 100 data run-throughs, early stopping metric using AUC and tolerance of 0.

```
##    user  system elapsed
##   0.336   0.030  47.904
```

The ROC curve of the model:

The resulting test set AUC is:

```
## [1] 0.9073219
```

2nd run uses rectifier activation function and four hidden layer each with 50 neurons, with all other parameter kept at default.

```
##    user  system elapsed
##   0.303   0.013  21.569
```

The ROC curve of the model:

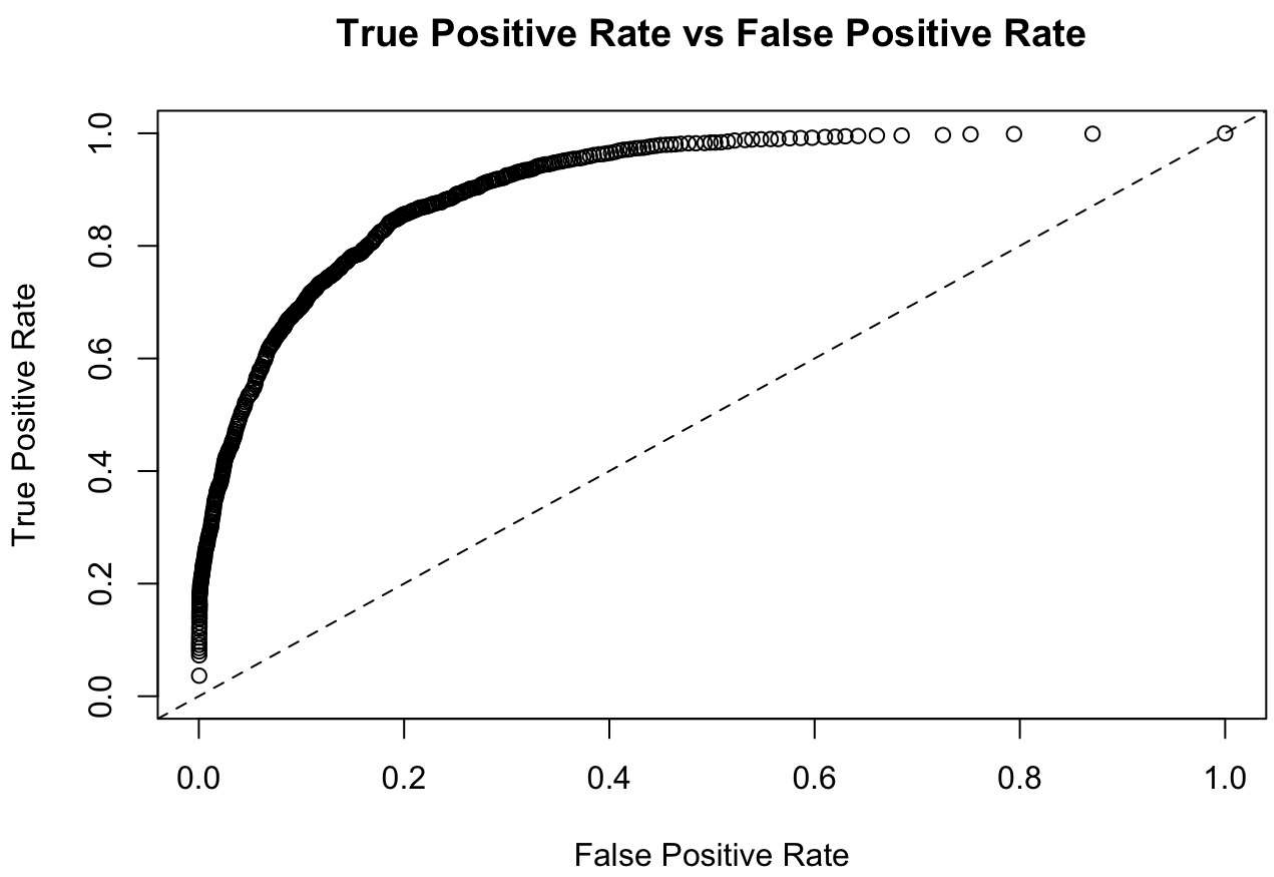
The resulting test set AUC is:

```
## [1] 0.9085076
```

3rd run uses rectifier activation function, four hidden layer each with 50 neurons, include an input drop out ratio of 20%, with all other parameter kept at default.

```
##    user  system elapsed
##   0.241   0.015  21.536
```

The ROC curve of the model:



The resulting test set AUC is:

```
## [1] 0.9085188
```

4th run uses rectifier activation function, two hidden layer with 50 neurons each, with all other parameter kept at default.

```
##    user  system elapsed
##   0.196   0.011  10.352
```

The ROC curve of the model:

The resulting test set AUC is:

```
## [1] 0.9066135
```

5th run uses rectifier activation function, one hidden layer with 50 neurons, with all other parameter kept at default.

```
##    user  system elapsed
##   0.165   0.006   6.277
```

The ROC curve of the model:

The resulting test set AUC is:

```
## [1] 0.908111
```

6th run uses rectifier activation function, four hidden layer with 100 neurons, with all other parameter kept at default.

```
##    user  system elapsed
##   0.367   0.030  56.014
```

The ROC curve of the model:

The resulting test set AUC is:

```
## [1] 0.9102031
```

7th run uses rectifier activation function, four hidden layer with 100 neurons, include hidden layer drop out ratios of 20%, 10%, 10% and 0%, with all other parameter kept at default.

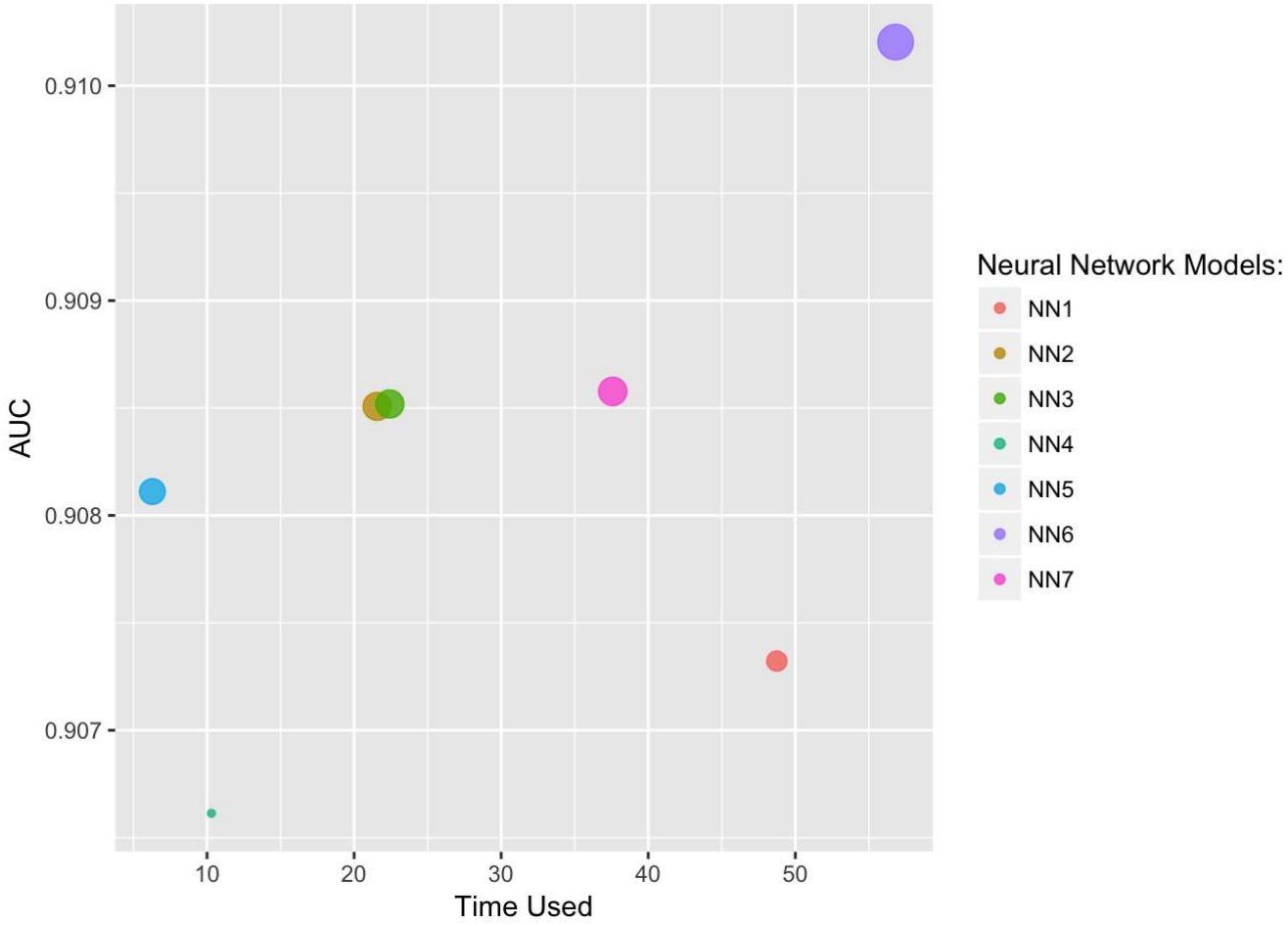
```
##    user  system elapsed
##   0.286   0.019  37.680
```

The ROC curve of the model:

The resulting test set AUC is:

```
## [1] 0.9085772
```

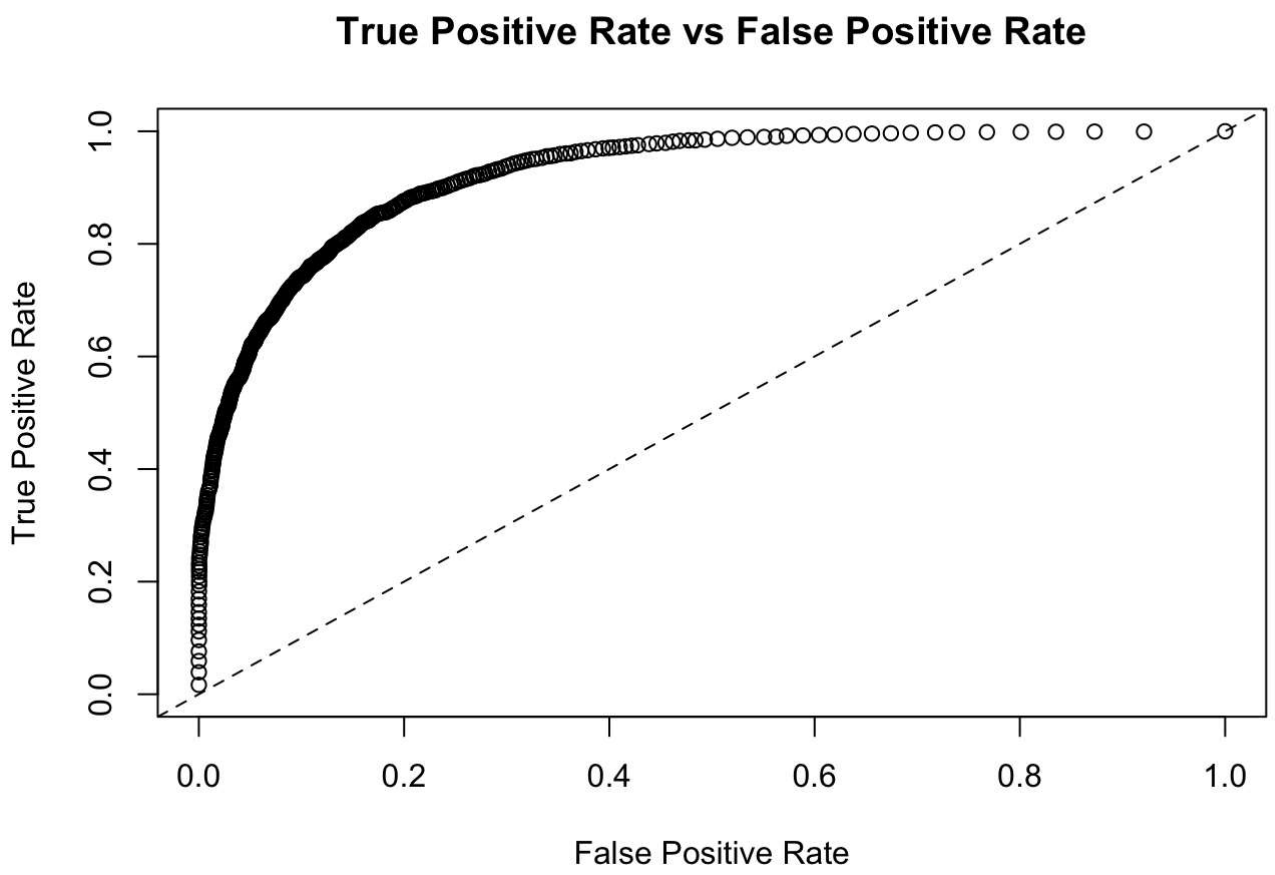
Now compare all the manually tuned neural network models. They all produced similar AUC around 0.906 to 0.910. However, looking at the time it takes to achieve high level of AUC (cost effectiveness) is an important metric here in determining the best model. Plotting the time each model took to form and their relative AUC indicate that, similar to how we evaluate a good ROC curve, the model that sits the most top left is the best, since it uses the least time to achieve a relatively high AUC. In this case, model number 5, which uses one hidden layer of 50 neurons is the best here.



Ensemble

Finally, try ensemble all the best model found from above section. Ensembling requires n-fold cross validation done to the models, which we did use but used validation frame do in the above sections. This cross validation process also do not require validation frame specified. So resample the data with 70% as training and 30% as testing, and re-do the best models with 5 fold cross validation and early stopping using AUC and tolerance 0.

The ROC curve of the ensemble model:



The resulting test set AUC is:

```
## [1] 0.9221101
```

Also take a look at how the ensemble model is made up of:

```
## Coefficients: glm coefficients
##                                     names coefficients
## 1                               Intercept    -3.299423
## 2          GLM_model_R_1496826971952_3423     0.492679
## 3          DRF_model_R_1496826971952_3441     2.199570
## 4          GBM_model_R_1496826971952_3752     3.823806
## 5 DeepLearning_model_R_1496826971952_4093     0.130472
##   standardized_coefficients
## 1                 -1.660284
## 2                  0.142858
## 3                  0.635591
## 4                  1.209076
## 5                  0.038151
```

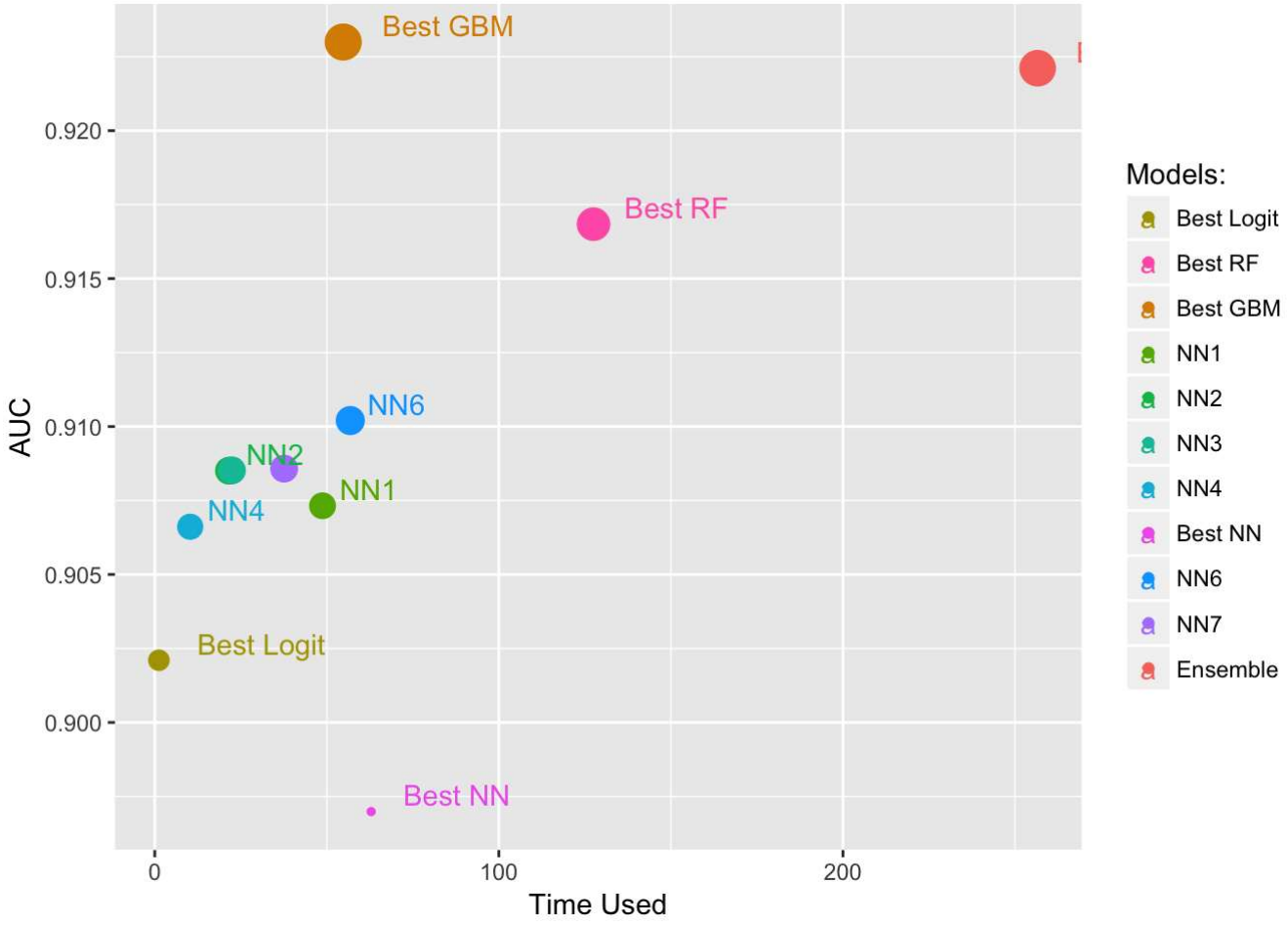
Notice that the models that have lowest AUC, namely Logistic Regression and Neural Network, has the lowest coefficients in the ensemble model.

Comparing All Models

Perform a cost effectiveness analysis on all the models we produced thus far. When factoring in the total time it took for ensemble to generate a model, keep in mind that for ensemble to work, you need the other models first. Thus, the total time ensemble model need is all other best models combined, resulting in very high "cost."

Based on the graph, the most efficient model is Gradient Boosting Machine, with the highest AUC with relatively fast speed.

Also notice on the graph, that our best Neural Network model did not work very well against cross validation, which means this neural network model has low generalization, which also means our manual tuning is not optimized at all. Consider H2O's deepwater for hyperparameter optimization in future updates.



Conclusion:

All of the models present really high standard accuracy in predicting the test set data. Thus, there are not much to discuss about the ROC curve between models. However, cost effectiveness analysis is fruitful in indicating a most efficient model among all.

Things to improve upon the project may include better hyperparameter optimization regarding neural network. Using tools such as Deepwater on H2O framework.