SMU®

CLASSIFICATION ANALYSIS OF WORK VISAS IN THE UNITED STATES

Project 3 in EMIS 7331 - Data Mining.

Nikhila Byreddy & Samarth Suresh Kumar
CSE/EMIS 7331, Project 2

Table of Contents

Executive Summary

In this report, using data analytics, we use the publicly available immigration data for the available work visas in the United states to analyze trends in the visa applications over time.

For a foreign national to work in the United States, an employer must offer the job and file a petition for an H1B visa with the US immigration department. The primary focus of this project is to analyze about the attributes in the dataset that contribute to the final visa status.

For each dataset, before clustering the data, cleaning is performed. All the duplicate data is removed, appropriate methods are implemented in dealing with the missing data and outliers. Some attributes are selected, and the analysis is performed individually.

We created various models using CART, Artificial Neural Network, KNN, Conditional interface Tree and Random Forest to predict the credit ratings and compared the results obtained.

Models were later used for primarily predicting Case Status, Job Title and Prevailing wages. These models had maximum accuracies ranging above 90%.

List of Figures:

List of Tables:

Data Preparation

The given dataset has 3,002,458 observations, after removing the duplicate entries and cleaning the data, we have 2,068,217 observations.

Considering that the data set is large, performing the cross-validation would be difficult. So, sample the dataset into 100,000 observations and perform the classification on these observations.

All the features available from the dataset are not used for classification. We remove the row count column. Classification is performed on the dataset which has the following features: PREVAILING_WAGE, SOC_NAME_SHORT, FULL_TIME_POSITION, STATE, CERTIFIED. Also, SOC_NAME for upper case and lower case are considered separately. We converted all the SOC_NAMEs to uppercase to make it concise and then performed classification. SOC_NAME_ SHORT has the all the names with first four characters. SOC_NAME is converted into factor before performing the classification. Prevailing wage is converted into numeric before we begin classification. LOW_WAGE has all the wages which are less than 68,000. We removed the commas present in the salaries and also convert these wages which are strings into factors. Classification is performed to explain the variable LOW_WAGE with the help of all the other variables in the dataset which we constructed.

Table 1 : Class Variables

Variable	Description	Scale
LOW_WAGE	Wages less than 68000	Ordinal
CERTIFIED	Certified Case Status or Not	Ordinal
SOC_NAME_SHORT	Shortened Job Title.	Nominal

Features used for prediction of class variables are shown below:

Table 2: Features

Feature	Description	Type
PREVAILING_WAGE	Salary of the applicant	Ratio
SOC_NAME_SHORT	First four characters of SOC_NAME s of each applicant.	Nominal
FULL_TIME_POSITION	True or False	Nominal
STATE	Location of the Employer	Nominal
Certified	Status of the applicants	Nominal
COUNTRY_OF_CITIZENSHIP	Country of Citizenship	Nominal
Class of Admission	Visa on Arrival	Nominal
Employer Name	Name of the Employer	Nominal

We reduce the no. of factor levels in nominal variables by finding the top 10 categories by frequency and performing classification based on results using aggregation and data selection.

Table 3 : Final dataset

Feature	Description	Type
PREVAILING_WAGE	Salary of the applicant	Ratio
SOC_NAME_SHORT	First four characters of SOC_NAME s of each applicant.	Nominal (10 Levels)

FULL_TIME_POSITION	True or False	Nominal (10 Levels)
STATE	Location of the Employer	Nominal (10 Levels)
CERTIFIED	Status of the applicants	Nominal (2 Levels)
LOW_WAGE	Employers with wages less than 68,000	Ratio
COUNTRY_OF_CITIZENSHIP	Country of citizenship	Nominal (10 Levels)
CLASS_OF_ADMISSION	Class of admission	Nominal (10 Levels)
EMPLOYER_NAME	Name of the employer	Nominal (10 Levels)

Modelling:

Model 1: Decision Tree

As the dataset is massive, we sampled the data to 100,000 records for the better classification. The features in the dataset which we are using for the classification is SOC_NAME, FULL_TIME_POSITION, STATE to predict PREVAILING WAGE. We made the SOC_NAME short, i.e., we reduced the number of characters for SOC_NAME to 4. The condition for the decision tree at the root node is the PREVAILING_WAGE less than 68,000. This is because both the mean and median are approximately equal to 68,000.

The plot for decision tree is shown below:

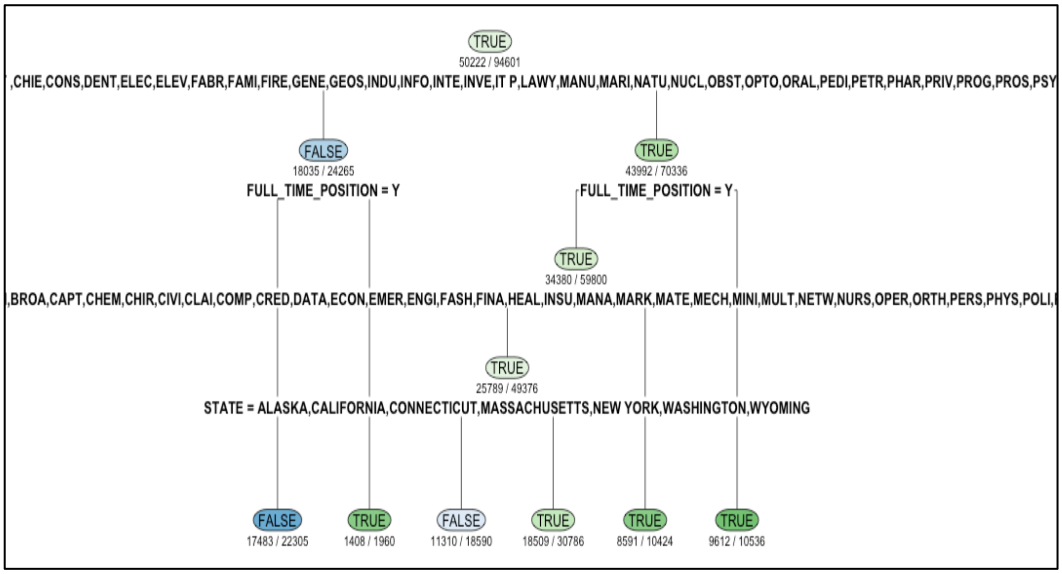
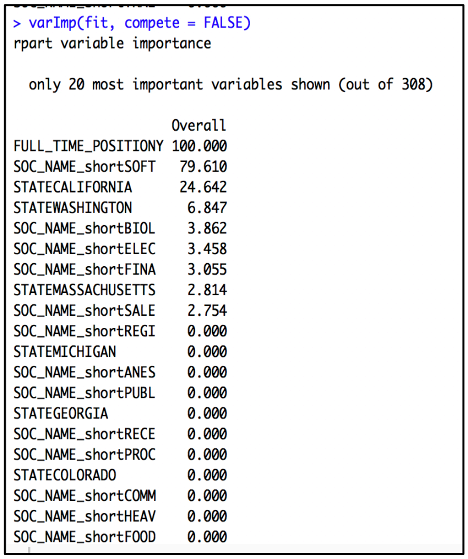

TRUE
50222 / 94601
,CHIE,CONS,DENT,ELEC,ELEV,FABR,FAMI,FIRE,GENE,GEOS,INDU,INFO,INTE,INVE,IT P,LAWY,MANU,MARI,NATU,NUCL,OBST,OPTO,ORAL,PEDI,PETR,PHAR,PRIV,PROG,PROS,PSY
FALSE
18035 / 24265
FULL_TIME_POSITION = Y
TRUE
43992 / 70336
FULL_TIME_POSITION = Y
TRUE
34380 / 59800
,BROA,CAPT,CHEM,CHIR,CIVI,CLAI,COMP,CRED,DATA,ECON,EMER,ENGI,FASH,FINA,HEAL,INSU,MANA,MARK,MATE,MECH,MINI,MULT,NETW,NURS,OPER,ORTH,PERS,PHYS,POLI,
TRUE
25789 / 49376
STATE = ALASKA,CALIFORNIA,CONNECTICUT,MASSACHUSETTS,NEW YORK,WASHINGTON,WYOMING
FALSE
17483 / 22305
TRUE
1408 / 1960
FALSE
11310 / 18590
TRUE
18509 / 30786
TRUE
8591 / 10424
TRUE
9612 / 10536

Figure 1: Decision tree to predict low wages

From the above figure, we found that 50222 out of 94601 wages are not less than 68,000. Then the classification continued based on the SOC_NAME_SHORT.SOC_NAMES are divided into two groups and for each group the classification is performed based on FULL_TIME_POSITION.

For some SOC_NAME s such as CHEF, CHEM etc. we found there are 25420 out of 59800 observations have their FULL_TIME_POSITION Y.

These observations are classified based on states as shown above.



```
> varImp(fit, compete = FALSE)
rpart variable importance

  only 20 most important variables shown (out of 308)

                   Overall
FULL_TIME_POSITIONY 100.000
SOC_NAME_shortSOFT   79.610
STATECALIFORNIA      24.642
STATEWASHINGTON       6.847
SOC_NAME_shortBIOL    3.862
SOC_NAME_shortELEC    3.458
SOC_NAME_shortFINA    3.055
STATEMASSACHUSETTS    2.814
SOC_NAME_shortSALE    2.754
SOC_NAME_shortREGI    0.000
STATEMICHIGAN         0.000
SOC_NAME_shortANES    0.000
SOC_NAME_shortPUBL    0.000
STATEGEORGIA          0.000
SOC_NAME_shortRECE    0.000
SOC_NAME_shortPROC    0.000
STATECOLORADO         0.000
SOC_NAME_shortCOMM    0.000
SOC_NAME_shortHEAV    0.000
SOC_NAME_shortFOOD    0.000
```

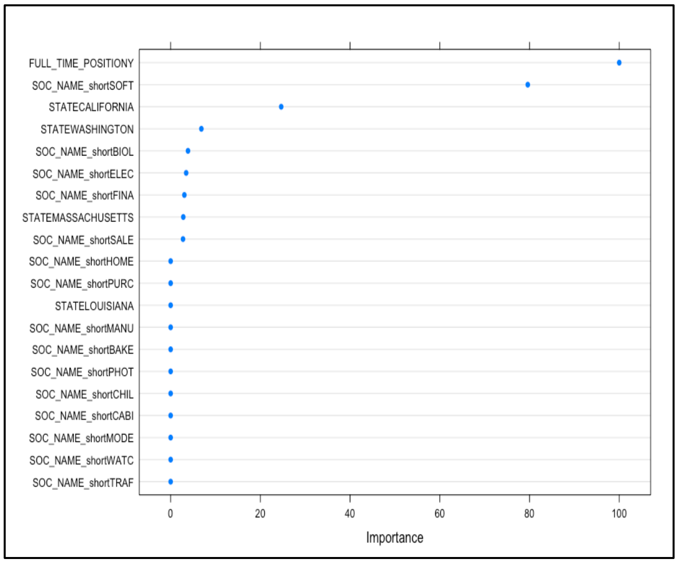


Figure 2:Important variables for decision tree

Figure 3: Important Variables

Above figure shows us top 20 important variables. After evaluation of a decision tree, we found out the best classifier. From the best decision tree, we can find that the nodes which are used for splitting and classifying the data are considered as important variables.

Model 2: Random forest

This classifier uses the same dataset as used for model 1. In this model, we will use all the features in the dataset to predict low wages.

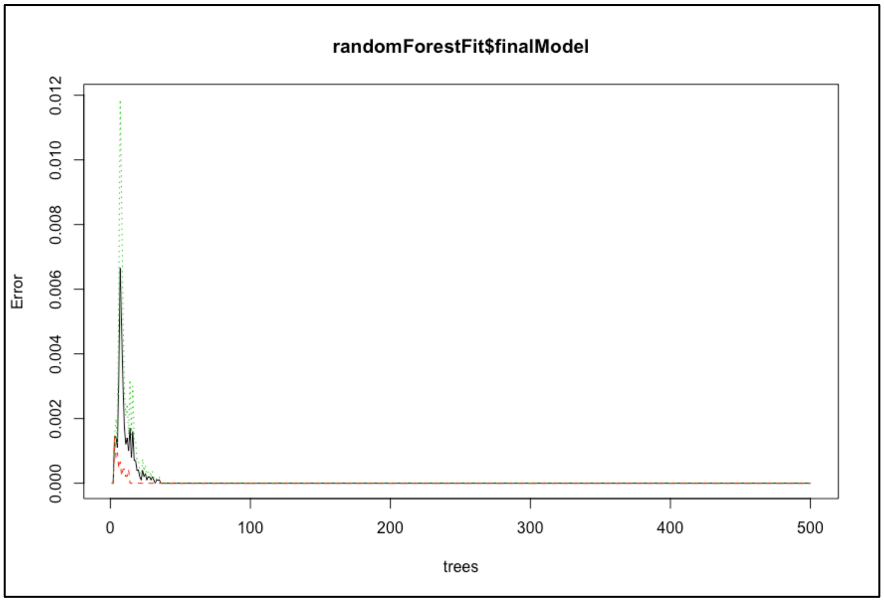
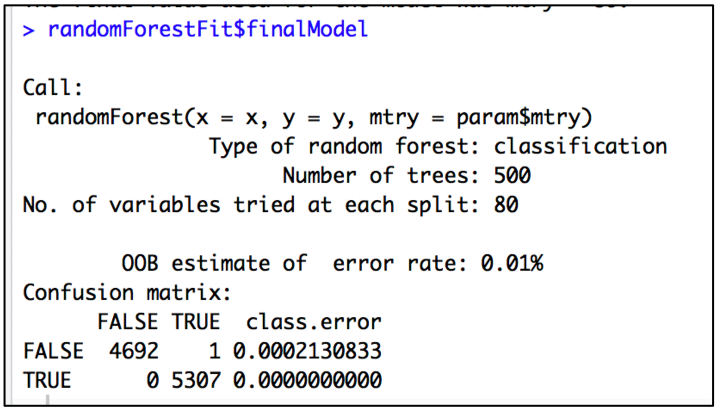


Figure 4: Error vs Number of Trees

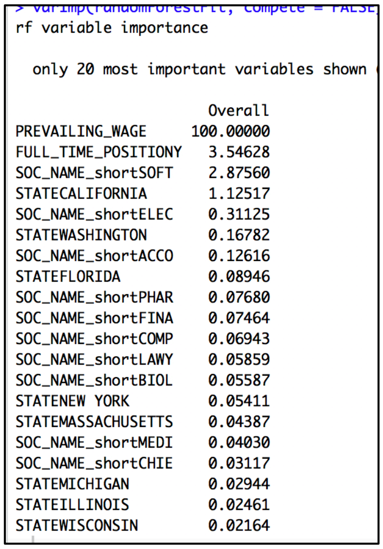
The above plot shows the relation between error and number of trees. We can observe that as we increase the number of trees, error rate reduces. Out of bag error estimation is used in Random forest model. Here, 2/3 of training data is used for training and remaining 1/3 is used to validate the trees.

For this model, the number of trees that are randomly generated are 500 and number of variables that are randomly chosen for split in each tree are 80.



```
> randomForestFit$finalModel

Call:
 randomForest(x = x, y = y, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 80

        OOB estimate of  error rate: 0.01%
Confusion matrix:
      FALSE TRUE  class.error
FALSE  4692    1 0.0002130833
TRUE      0 5307 0.0000000000
```

Figure 5 : Random forest fit model to predict low wages



```
> varImp(randomForestFit, compete = FALSE
rf variable importance

  only 20 most important variables shown

                      Overall
PREVAILING_WAGE     100.00000
FULL_TIME_POSITIONY   3.54628
SOC_NAME_shortSOFT    2.87560
STATECALIFORNIA       1.12517
SOC_NAME_shortELEC    0.31125
STATEWASHINGTON       0.16782
SOC_NAME_shortACCO    0.12616
STATEFLORIDA          0.08946
SOC_NAME_shortPHAR    0.07680
SOC_NAME_shortFINA    0.07464
SOC_NAME_shortCOMP    0.06943
SOC_NAME_shortLAWY    0.05859
SOC_NAME_shortBIOL    0.05587
STATENEW YORK         0.05411
STATEMASSACHUSETTS    0.04387
SOC_NAME_shortMEDI    0.04030
SOC_NAME_shortCHIE    0.03117
STATEMICHIGAN         0.02944
STATEILLINOIS         0.02461
STATEWISCONSIN        0.02164
```

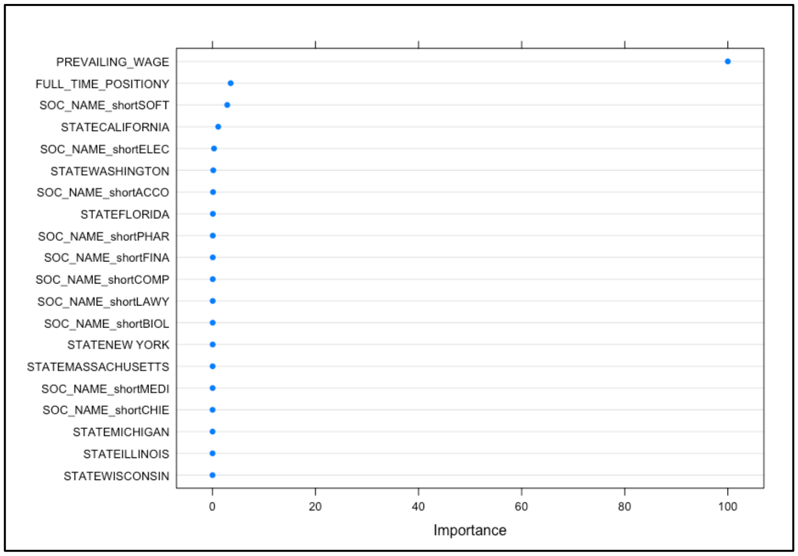


Figure 6 : Plot between the variables and their importance for random forest fit

Figure 7 : Important variables for random forest

In this case, we have 500 random trees and each split has 80 random variables. The important variables are chosen from the best classifier.

There are many common variables for both the models. All the important variables in decision tree are present in random forest. But there are certain variables in random forest fit which are not present in decision tree.

So, we may assume that random forest fit requires more variables for classification.

Model 3: Conditional Interface tree

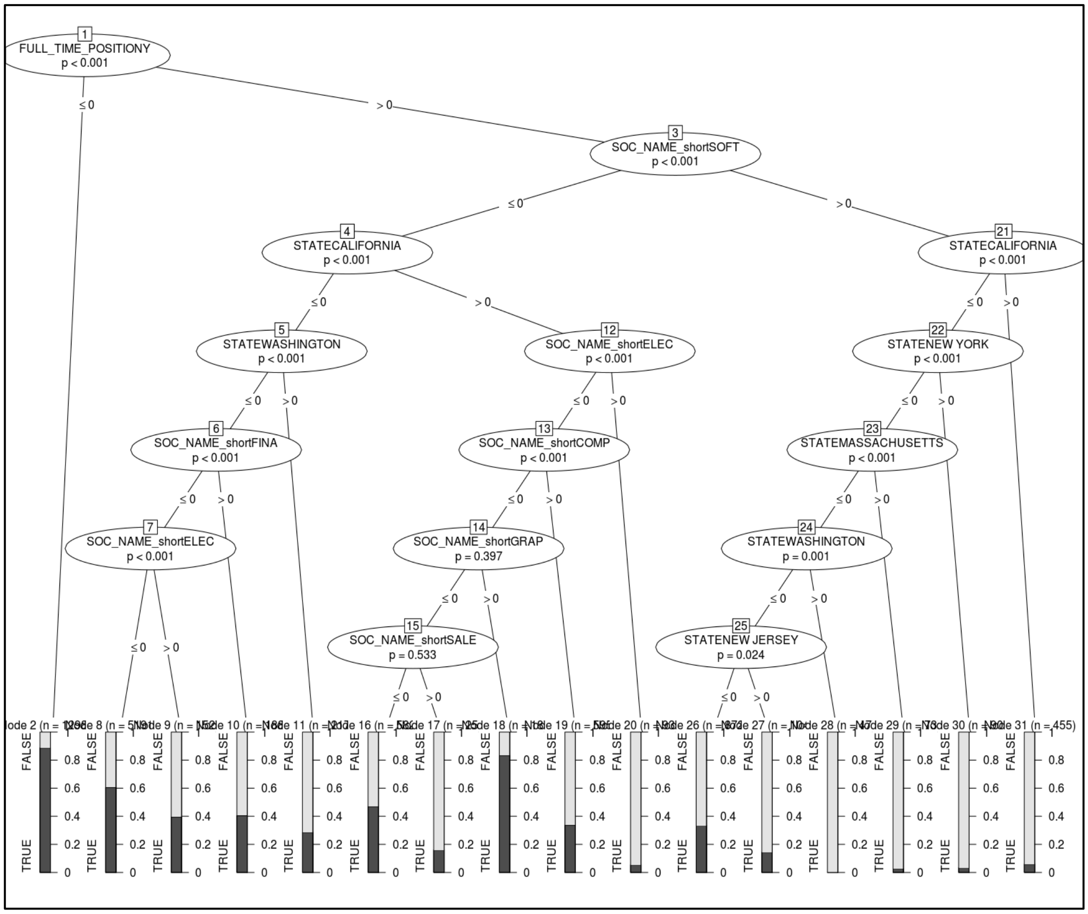


Figure 8 : Ctree to predict low wages

The above figure shows the plot of Ctree fit classifier. The terminal nodes show the distribution of class variables. In this case, it gives us the distribution of low wages. The terminal nodes tell us the if the given wage is less than 68,000 or not.

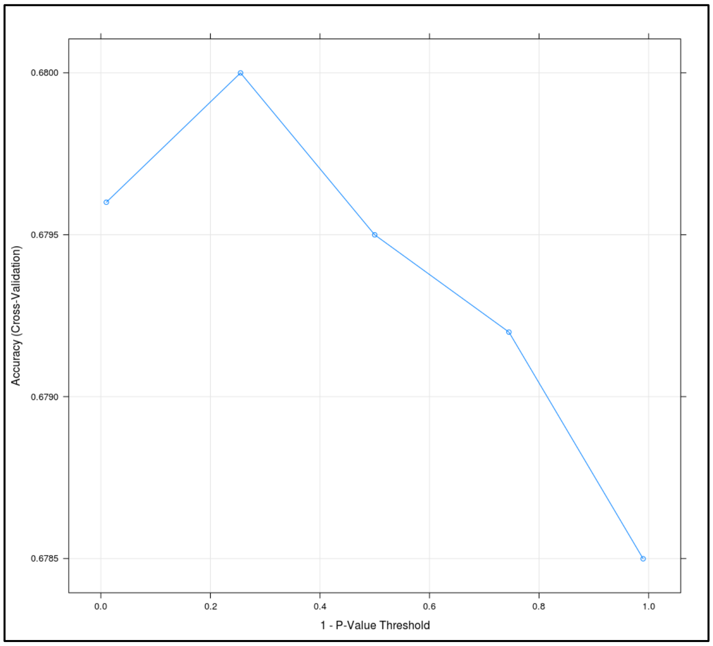
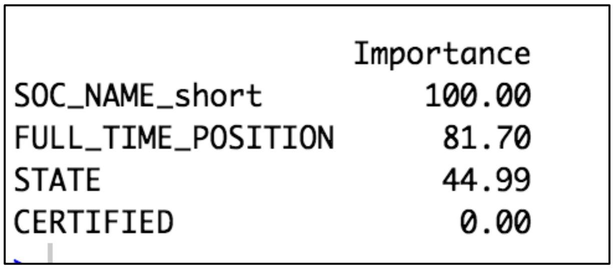


Figure 9 : Plot between p-value and accuracy for Ctree

From the above plot, the p-value is used to indicate the association between the variables. The split is usually made on the variable with the lowest p-value. The above figure shows us the plot of p-value and accuracy for Ctree. For the p-values in between 0.2 and 0.4, the accuracy is high. As the optimal model for Ctree is selected based on accuracy, we may assume that the variables in the optimal model have p-values in the range of 0.2 and 0.4.

For Ctree, the important variables are shown below:



```
                   Importance
SOC_NAME_short         100.00
FULL_TIME_POSITION      81.70
STATE                   44.99
CERTIFIED                0.00
```

For Conditional Interface tree, when I tried to find the important variables using VarImp function, it gave me the above values as shown above. We may assume that Ctree will not provide us all the important variables that are used in the classification

Model 4: Neural Networks:

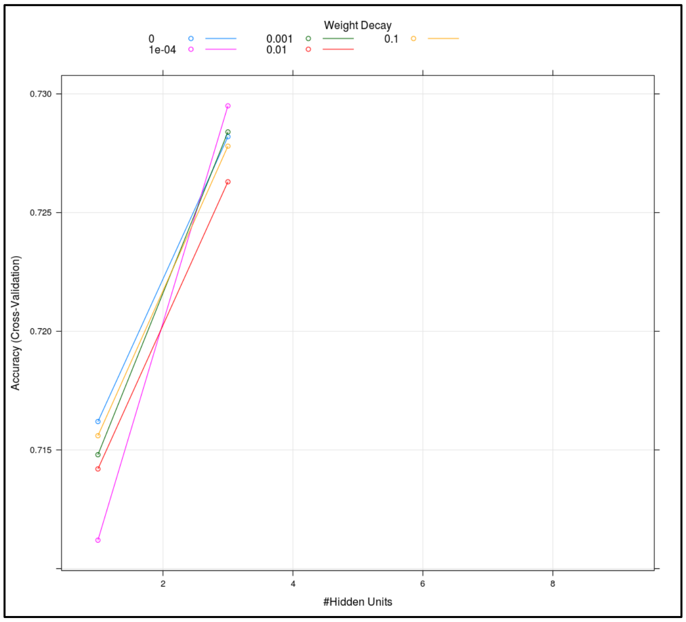
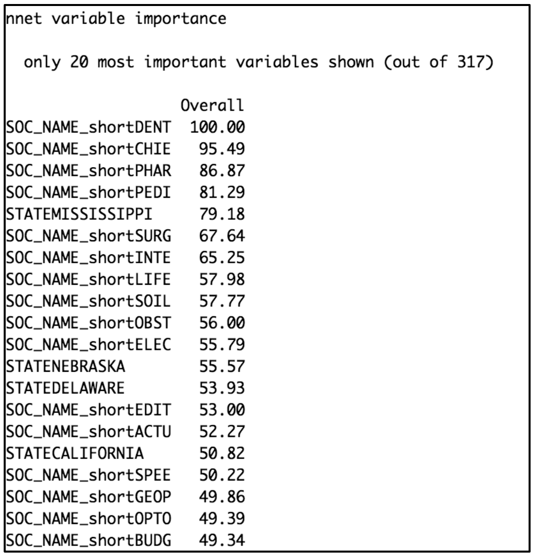


Figure 10 : Accuracy Vs Hidden Units ANN

Hidden units in Neural networks are the units which are used for processing the output.

We have chosen 0, 0.001,0.1,0.01, 0.0001 as hidden units. Weight decay of 0.0001 gives good accuracy.

Important variables:



```
nnet variable importance

  only 20 most important variables shown (out of 317)

                   Overall
SOC_NAME_shortDENT  100.00
SOC_NAME_shortCHIE   95.49
SOC_NAME_shortPHAR   86.87
SOC_NAME_shortPEDI   81.29
STATEMISSISSIPPI     79.18
SOC_NAME_shortSURG   67.64
SOC_NAME_shortINTE   65.25
SOC_NAME_shortLIFE   57.98
SOC_NAME_shortSOIL   57.77
SOC_NAME_shortOBST   56.00
SOC_NAME_shortELEC   55.79
STATENEBRASKA        55.57
STATEDELAWARE        53.93
SOC_NAME_shortEDIT   53.00
SOC_NAME_shortACTU   52.27
STATECALIFORNIA      50.82
SOC_NAME_shortSPEE   50.22
SOC_NAME_shortGEOP   49.86
SOC_NAME_shortOPTO   49.39
SOC_NAME_shortBUDG   49.34
```

Figure 11 : Important Variables

Fig. Important variables for Neural networks

The above figure shows the important variables used in the Neural networks classifier.

Model 5: SOC_NAME SHORT Prediction.

We train our model to predict the Shortened Job Titles using "FULL_TIME_POSITION", "PREVAILING_WAGE","STATE", "CERTIFIED" features.

In order to boost our training performance, we reduce our feature's factor lengths to 10 levels. We do this by finding the top 10 categories by frequency and take a subset of it from our data. This vastly improved in terms of time taken to train.

We perform KNN, Random Forest, Ctree, Linear SVM and CART:

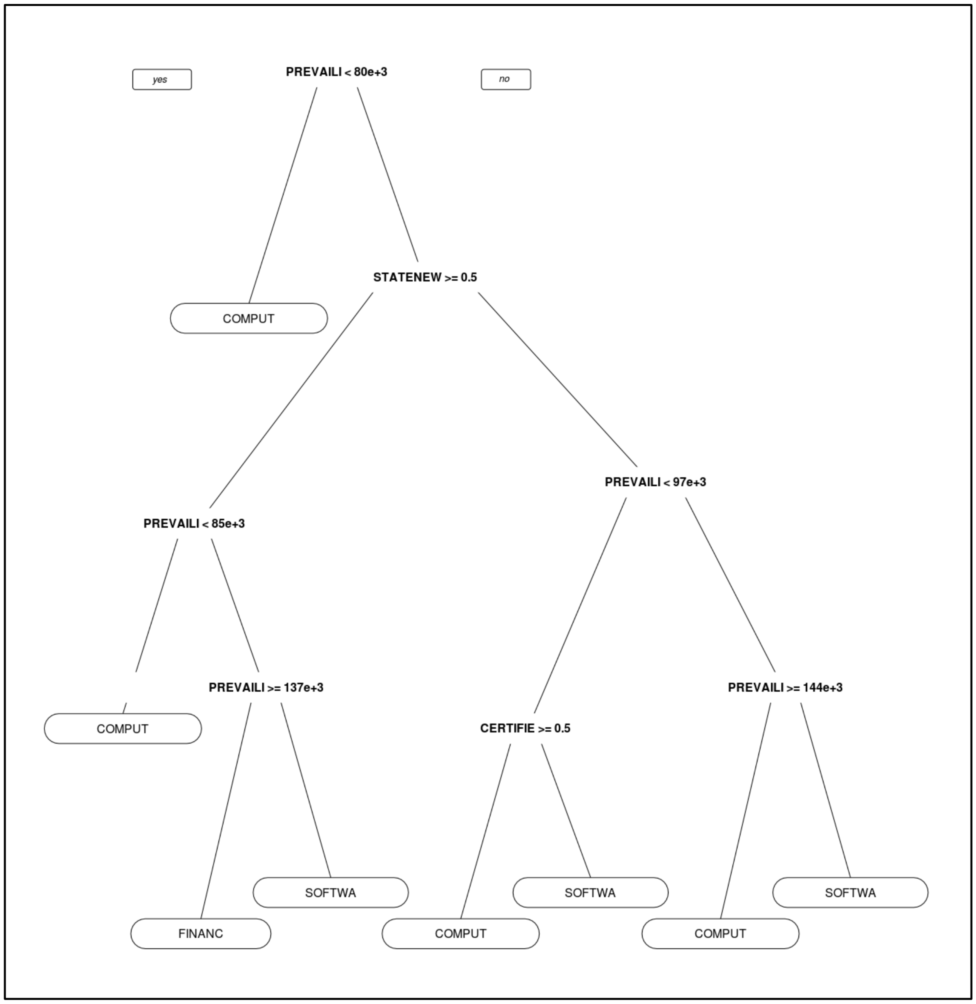
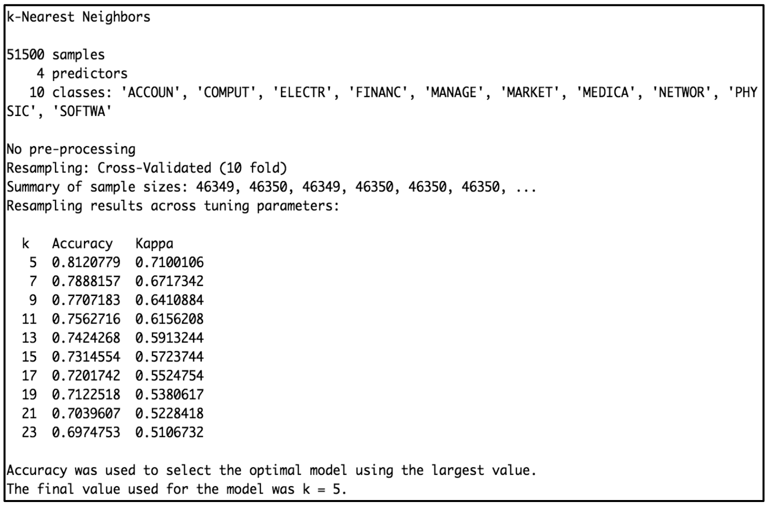


Figure 12 : CART Tree to Predict Job Title

The above diagram is the classification tree of the final model. We are able to distinctly classify Financial Jobs, Computer and Software Jobs based on Certified Status Prevailing wage. We see that New York was a distinguishing factor here to decide between Financial and Software Jobs. This diagram is just to have a visual representation of how classification was performed using RPart.

We perform KNNearest neighbor classification only to get a better maximum accuracy of > 80% which is good for predictions. Nearest neighbor classifiers compute the Euclidean distance to other records to identify nearest neighbors. It uses the nearest neighbors to determine class label of unknown record.

Using 5 neighbors, our model was able to hit its highest accuracy only reducing as no. of neighbors increased.



```
k-Nearest Neighbors

51500 samples
    4 predictors
   10 classes: 'ACCOUN', 'COMPUT', 'ELECTR', 'FINANC', 'MANAGE', 'MARKET', 'MEDICA', 'NETWOR', 'PHY
SIC', 'SOFTWA'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 46349, 46350, 46349, 46350, 46350, 46350, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.8120779  0.7100106
   7  0.7888157  0.6717342
   9  0.7707183  0.6410884
  11  0.7562716  0.6156208
  13  0.7424268  0.5913244
  15  0.7314554  0.5723744
  17  0.7201742  0.5524754
  19  0.7122518  0.5380617
  21  0.7039607  0.5228418
  23  0.6974753  0.5106732

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

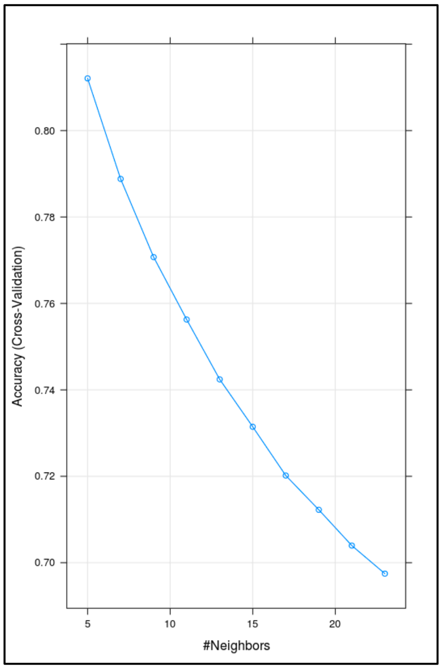
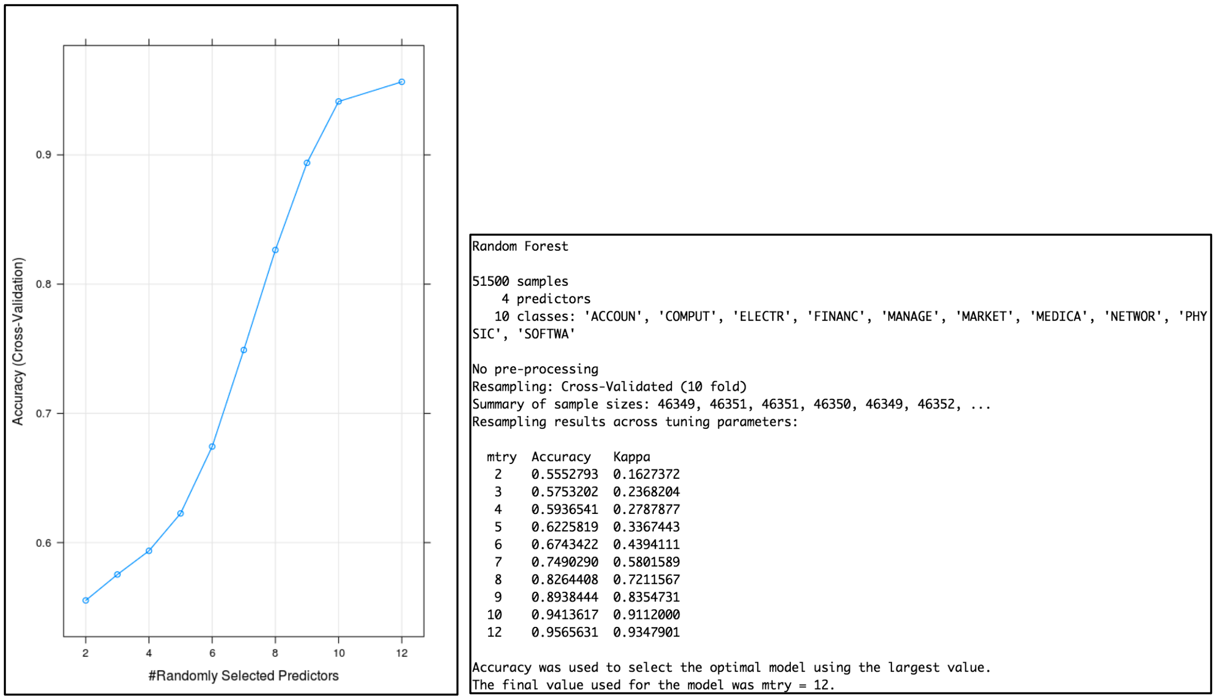


Figure 13 : KNN Statistics

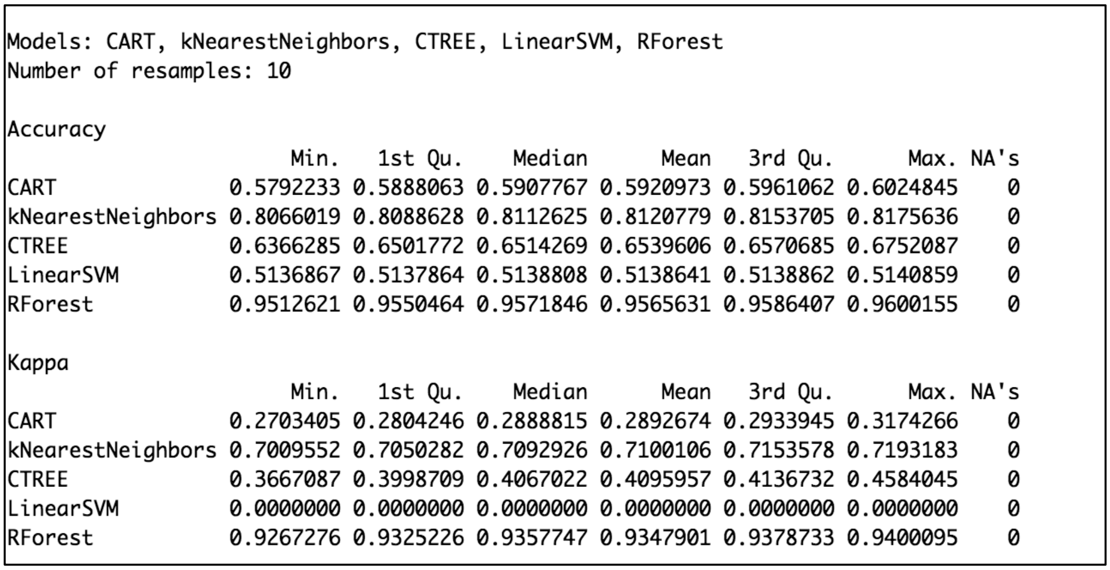
Random Forest : We found most accurate model to be Random Forest hitting accuracy greater than 90% for 12 randomly selected predictors :



```
Random Forest

51500 samples
    4 predictors
   10 classes: 'ACCOUN', 'COMPUT', 'ELECTR', 'FINANC', 'MANAGE', 'MARKET', 'MEDICA', 'NETWOR', 'PHY
SIC', 'SOFTWA'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 46349, 46351, 46351, 46350, 46349, 46352, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.5552793  0.1627372
   3    0.5753202  0.2368204
   4    0.5936541  0.2787877
   5    0.6225819  0.3367443
   6    0.6743422  0.4394111
   7    0.7490290  0.5801589
   8    0.8264408  0.7211567
   9    0.8938444  0.8354731
  10    0.9413617  0.9112000
  12    0.9565631  0.9347901

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 12.
```

Figure 14: Random Forest Statistics

After training five different models to predict Job Titles , we directly compare their performances with each other.



```
Models: CART, kNearestNeighbors, CTREE, LinearSVM, RForest
Number of resamples: 10

Accuracy
                       Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART              0.5792233 0.5888063 0.5907767 0.5920973 0.5961062 0.6024845    0
kNearestNeighbors 0.8066019 0.8088628 0.8112625 0.8120779 0.8153705 0.8175636    0
CTREE             0.6366285 0.6501772 0.6514269 0.6539606 0.6570685 0.6752087    0
LinearSVM         0.5136867 0.5137864 0.5138808 0.5138641 0.5138862 0.5140859    0
RForest           0.9512621 0.9550464 0.9571846 0.9565631 0.9586407 0.9600155    0

Kappa
                       Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART              0.2703405 0.2804246 0.2888815 0.2892674 0.2933945 0.3174266    0
kNearestNeighbors 0.7009552 0.7050282 0.7092926 0.7100106 0.7153578 0.7193183    0
CTREE             0.3667087 0.3998709 0.4067022 0.4095957 0.4136732 0.4584045    0
LinearSVM         0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000    0
RForest           0.9267276 0.9325226 0.9357747 0.9347901 0.9378733 0.9400095    0
```

Figure 15 : Comparing Models

Note LinearSVM has 0 Kappa values throughout indicating it did not work well for our Multi Classification problem.

Advantages between Different Classifications :

Decision Tree:

A decision tree helps in feature selection. For the range of important features in the dataset, decision tree makes it easy. When we fit a decision tree to a training dataset, top few nodes on which tree is split are mostly the most important variables within the dataset.

When we tried to perform the evaluation, it gives us the tree which is the best classifier among all the ten different sets of samples. From the best classifier, we can conclude that the importance of variable for the top node is more. One can verify this from the varImp function.

Decision tree saves the time for data preparation. In our case for model 1, we are not eliminating the missing values, as they do not prevent splitting the data for building trees. A decision tree is easy to understand and easy to explain.

Artificial Neural Network:

ANN s has the ability to learn and model non-linear and complex relationships which is really important. Also, in this case relationships between inputs and outputs are non-linear and complex.

For our given dataset, after learning from initial inputs and their relationships, ANN can infer unseen relationships on unseen data, making the model generalize and predict on unseen data.

ANN will not impose any restrictions on the input variables.

Random forests:

They help in reduction in overfitting, by averaging several trees which significantly lowers the risk of overfitting.

Also, Random forests have less variance. By using multiple trees, one can reduce the chance of stumbling across a classifier that doesn’t perform well because of relationship between train and test data.

Conditional Interface tree:

Solves the problems of the decision tree. This tree stops early, and it uses permutation steps for splitting instead of Gini index. As the tree stops early, it is smaller in most of the cases compared to a decision tree.

SVM:

The main advantage of SVM is that once the boundary is established, most of the training data is redundant. It just needs the core set of points which can help identify and set the boundary. This means that small changes in the data will not greatly affect the SVM.

Also, Support vector machines are resistant to overfitting. SVM has kernel trick embedded which helps to separate data in hyperplanes. SVMs are effective in high dimensional spaces.

For the given dataset, to perform SVM classification, it uses the subset of training points in the decision function, so it is memory efficient.

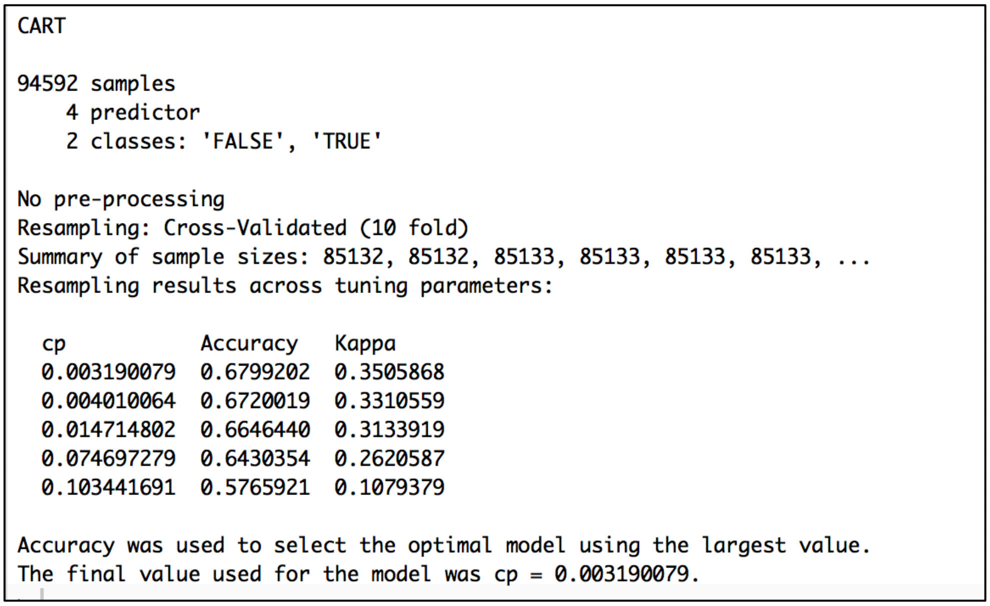
Cross Validation:

Decision tree:

For the evaluation of this decision tree, let us do cross validation as shown below,

Here the data is internally split into training and testing data. Here for this model train tries to tune the cp parameter using accuracy to choose the best model.

For the decision tree, we considered 100,000 observations from the entire dataset. So, cross validation is also performed on 100,000 observations.

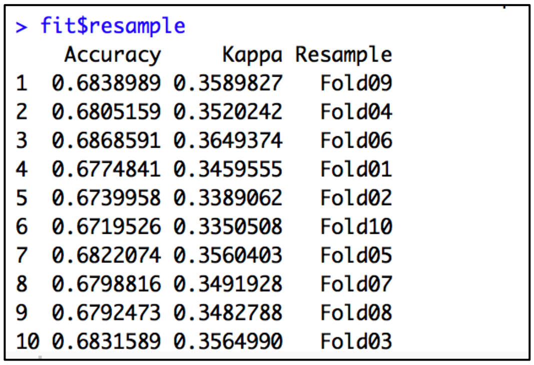


```
CART

94592 samples
    4 predictor
    2 classes: 'FALSE', 'TRUE'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 85132, 85132, 85133, 85133, 85133, 85133, ...
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.003190079  0.6799202  0.3505868
  0.004010064  0.6720019  0.3310559
  0.014714802  0.6646440  0.3133919
  0.074697279  0.6430354  0.2620587
  0.103441691  0.5765921  0.1079379

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.003190079.
```

Figure 16: Statistics for decision tree

There are 10 samples and each sample has approximately same size as shown above. Considering both the accuracy and Kappa, we may conclude that the model with complexity 0.003190079 is the best classifier.

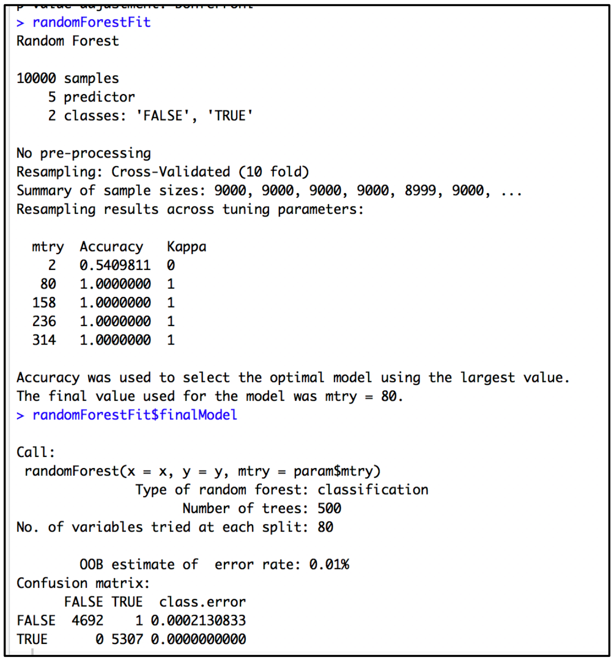
We use fit$resample gives us accuracy, Kappa for 10 different folds as shown below



```
> fit$resample
    Accuracy     Kappa Resample
1  0.6838989 0.3589827   Fold09
2  0.6805159 0.3520242   Fold04
3  0.6868591 0.3649374   Fold06
4  0.6774841 0.3459555   Fold01
5  0.6739958 0.3389062   Fold02
6  0.6719526 0.3350508   Fold10
7  0.6822074 0.3560403   Fold05
8  0.6798816 0.3491928   Fold07
9  0.6792473 0.3482788   Fold08
10 0.6831589 0.3564990   Fold03
```

Figure 17 : Accuracy Per Sample

Random Forest:



```
> randomForestFit
Random Forest 

10000 samples
    5 predictor
    2 classes: 'FALSE', 'TRUE' 

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 9000, 9000, 9000, 9000, 8999, 9000, ... 
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.5409811  0    
   80   1.0000000  1    
  158   1.0000000  1    
  236   1.0000000  1    
  314   1.0000000  1    

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 80.
> randomForestFit$finalModel

Call:
 randomForest(x = x, y = y, mtry = param$mtry) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 80

        OOB estimate of  error rate: 0.01%
Confusion matrix:
      FALSE TRUE  class.error
FALSE  4692    1 0.0002130833
TRUE      0 5307 0.0000000000
```

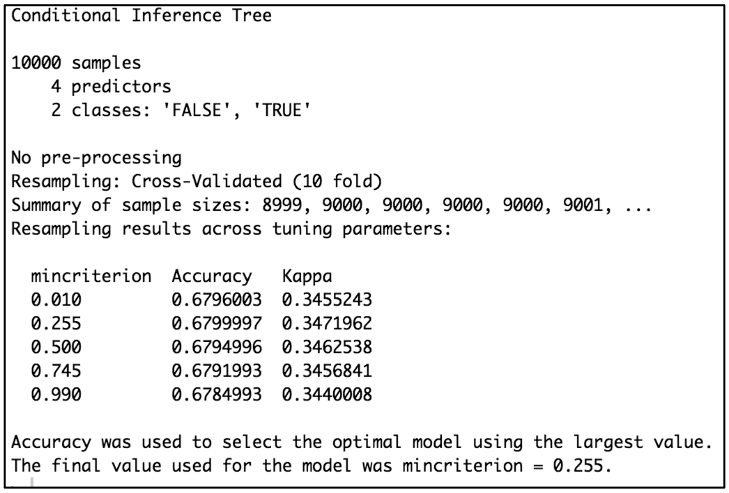
Figure 18 : Statistics for Random forest fit

From the above figure, based on accuracy which is 1.00, final tree is selected after selecting 80 as the number of variables at each split and number of trees as 500 as shown above, mtry 80 is the final value used for the model.

We are predicting the low wages from the dataset. Also from the above confusion matrix: Number of false negatives the model detected are 4692 and the number of true positives it detected are 5307.

The error rate for the model is 0.01 % and the accuracy and kappa values for most of the mtry s are 100%, from which we may assume that this model can act as a good predictor.

Ctree fit:

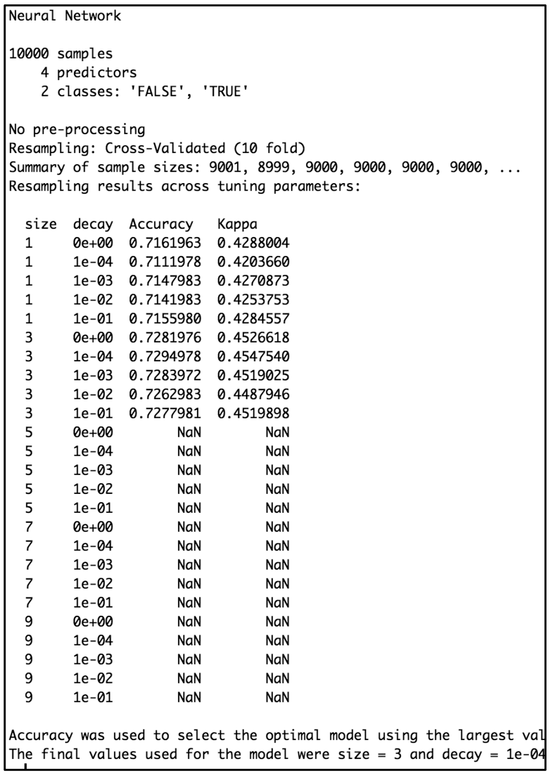


```
Conditional Inference Tree 

10000 samples
    4 predictors
    2 classes: 'FALSE', 'TRUE' 

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 8999, 9000, 9000, 9000, 9000, 9001, ... 
Resampling results across tuning parameters:

  mincriterion  Accuracy   Kappa    
  0.010         0.6796003  0.3455243
  0.255         0.6799997  0.3471962
  0.500         0.6794996  0.3462538
  0.745         0.6791993  0.3456841
  0.990         0.6784993  0.3440008

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mincriterion = 0.255.
```

Figure 19 : Statistics for Conditional Interface tree

From the above figure, we may assume that based on the accuracy, the model with minicriterion 0.255 is selected as it has the highest accuracy. We may also note that the kappa statistic for this model is highest for the minicriterion 0.255.

Neural Network:



```
Neural Network

10000 samples
    4 predictors
    2 classes: 'FALSE', 'TRUE'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 9001, 8999, 9000, 9000, 9000, 9000, ...
Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa
  1     0e+00  0.7161963  0.4288004
  1     1e-04  0.7111978  0.4203660
  1     1e-03  0.7147983  0.4270873
  1     1e-02  0.7141983  0.4253753
  1     1e-01  0.7155980  0.4284557
  3     0e+00  0.7281976  0.4526618
  3     1e-04  0.7294978  0.4547540
  3     1e-03  0.7283972  0.4519025
  3     1e-02  0.7262983  0.4487946
  3     1e-01  0.7277981  0.4519898
  5     0e+00        NaN        NaN
  5     1e-04        NaN        NaN
  5     1e-03        NaN        NaN
  5     1e-02        NaN        NaN
  5     1e-01        NaN        NaN
  7     0e+00        NaN        NaN
  7     1e-04        NaN        NaN
  7     1e-03        NaN        NaN
  7     1e-02        NaN        NaN
  7     1e-01        NaN        NaN
  9     0e+00        NaN        NaN
  9     1e-04        NaN        NaN
  9     1e-03        NaN        NaN
  9     1e-02        NaN        NaN
  9     1e-01        NaN        NaN

Accuracy was used to select the optimal model using the largest val
The final values used for the model were size = 3 and decay = 1e-04
```

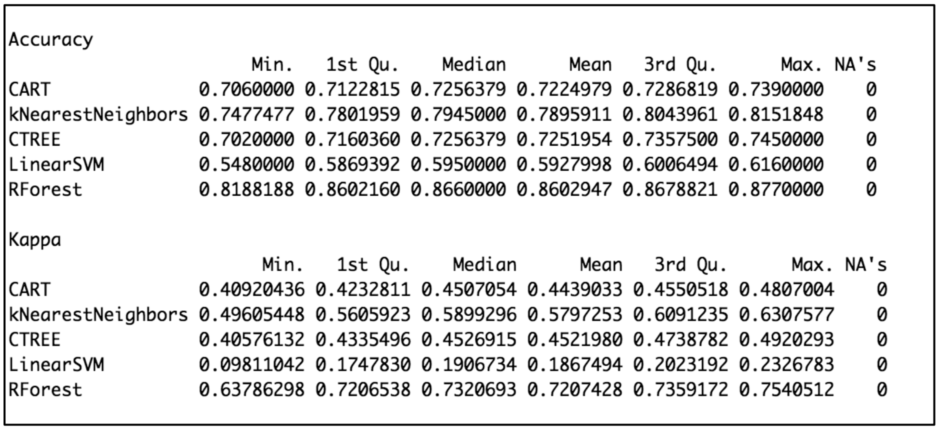
Figure 20 : Statistics for ANN

From the above statistics, we may assume that accuracy is statistic that is used to find the optimal model. The highest accuracy is for size 3 which is selected as the optimal model. Also, we may note that the kappa value is highest for this optimal model.

Exceptional Work:

We go beyond our requirement of finding 5 different classification models by exploring the permanent visa dataset to predict Case Status using Employer name, Class of Admission, Shortened SOC name, Prevailing Wages and Country of Citizenship.

We generate five more different models using RPART, KNN, CTree, LinearSVM and RForest. Upon training, we find get the below results:

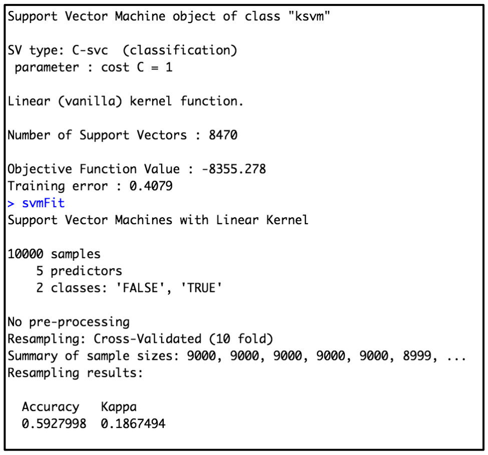


```
Accuracy
                       Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART              0.7060000 0.7122815 0.7256379 0.7224979 0.7286819 0.7390000    0
kNearestNeighbors 0.7477477 0.7801959 0.7945000 0.7895911 0.8043961 0.8151848    0
CTREE             0.7020000 0.7160360 0.7256379 0.7251954 0.7357500 0.7450000    0
LinearSVM         0.5480000 0.5869392 0.5950000 0.5927998 0.6006494 0.6160000    0
RForest           0.8188188 0.8602160 0.8660000 0.8602947 0.8678821 0.8770000    0

Kappa
                        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART              0.40920436 0.4232811 0.4507054 0.4439033 0.4550518 0.4807004    0
kNearestNeighbors 0.49605448 0.5605923 0.5899296 0.5797253 0.6091235 0.6307577    0
CTREE             0.40576132 0.4335496 0.4526915 0.4521980 0.4738782 0.4920293    0
LinearSVM         0.09811042 0.1747830 0.1906734 0.1867494 0.2023192 0.2326783    0
RForest           0.63786298 0.7206538 0.7320693 0.7207428 0.7359172 0.7540512    0
```

Figure 21 : Model Comparison for Case Status Prediction

LinearSVM Evaluation:

Since predicting Certified was a binary classification we were able to get and measure performance of SVM:



```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 8470

Objective Function Value : -8355.278
Training error : 0.4079
> svmFit
Support Vector Machines with Linear Kernel

10000 samples
    5 predictors
    2 classes: 'FALSE', 'TRUE'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 9000, 9000, 9000, 9000, 9000, 8999, ...
Resampling results:

  Accuracy   Kappa
  0.5927998  0.1867494
```

Figure 22 : SVM Statistics

Given Low accuracy performance of SVM, we would not recommend using this method for our given problem. We see that Random forest again triumphs in terms of Classification performance.

Evaluation and Deployment:

The classification model can be used to predict variety useful categories including Wages, State, Certified status, SOC. The stakeholders in our report are assumed to be potential Visa Applicants and Visa Officers who are accepting new applicants.

Visa applicants can use this report to check which category they belong to and predict accordingly.

Given parameters such as Wages, Job titles, State, one can predict if they would be certified or not. We have also created a model to predict if a Certified Applicant's wages belong to a lower category using Shortened Job title, State, Case Status and Full-Time position

They can also check Jobs they can get based on the factors such as wages, state and certified status.

As a visa application officer, one can use these models to pre-screen candidates and use data mining and machine learning to predict if the candidate can be certified or now.

We wouldn't recommend using these models in real life as it could have adverse penalties to predict false negatives and or False positives. However, upon further refining outliers, remove noise, dealing with imbalance after reducing factors and condensing specific elements can result in better prediction performance.

Conclusion

The data of the H1B had useful variables and clusters which influence the final decisions.

The following conclusion is made from mining the data in this project:

- The results support the findings of previous project.
- For every data set, we compared the classification techniques and found out the best clustering algorithm for our dataset was Random Forest.
- We found accuracies as well evaluated generalized performance.
- Handling the imbalance is an important step to improve the models.
- We found the important variables for different classifiers using the same dataset for the same classifier. Although, these set of variables are different for different models, we found that these classifiers have common variables.
- Some factors such as LOW_WAGE, COUNTRY_OF_CITIZENSHIP, CLASS_OF_ADMISSION is added to the model for the better classification.
- We concluded that the prediction can be improved if we increase the size of the data.
- We predict the following classes with accuracies not limited to:
    - Job Title: 94%
    - Case Status: 87%
    - Low Prevailing wage: 100%

References

1. https://www.foreignlaborcert.doleta.gov/docs/Performance_Data/Disclosure/FY16Q2/PERM_FY16_Record_Layout.pdf
2. https://www.uscis.gov/working-united-states/permanent-workers
3. http://www.cookbook-r.com/Graphs/Bar_and_line_graphs_(ggplot2)/
4. https://www.kaggle.com/nsharan/h-1b-visa
5. https://stackoverflow.com/questions/36568070/extract-year-from-date
6. https://www.kaggle.com/jboysen/us-perm-visas
7. https://rpubs.com/njvijay/16444
8. https://machinelearningmastery.com/tuning-machine-learning-models-using-the-caret-r-package/
9. https://rawgit.com/mhahsler/Introduction_to_Data_Mining_R_Examples/master/chap5.html#random-forest
10. https://rawgit.com/mhahsler/Introduction_to_Data_Mining_R_Examples/master/chap4.html#make-predictions-for-new-data
11. https://machinelearningmastery.com/machine-learning-evaluation-metrics-in-r/