Pstat131 FINAL PROJECT

Sung Hee(Sunny),Hong and Jason Edwards

04 June, 2020

1.) Introduction

Our research question : What socioeconomic qualities lead to a higher income in US adults? What qualities influence economic outcome the most?

What kind of qualities of a grown adult will be predicted to earn a higher income? What kind of qualities makes a huge impact to earning a higher income? How would different case of observation be predicted with the given data of adults? This project uses data from the 1994 United States Census data base in order to classify whether a given adult's income is greater than $50,000/yr. There are 14 predictors in the data set, so we aim to find out which ones are the most significant in classifying a given adult and minimizes test error. We apply a basic classifier method using KNN and an ensemble method using a random forest in our data analysis.

2.) Data Overview

The data was extracted by **Barry Becker** from the 1994 Census database. The records follow these following conditions : *((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))*. With this dataset obtained from UCI datasets source, we will model to predict the income whether a person makes over 50K or not. There are *32,561* observations and *15* columns in total. Our response variable is **"Income"**. Income is a binary variable with values "<=50K" and ">50K", indicating whether a person makes an annual income less than or equal to 50K or makes an annual income of more than 50K. The rest of the 14 variables will be determined to be used as an explanatory value or not for each methods. The following are the 14 variables : "age", "workclass", "fnlwgt", "education", "education_num", "marital_status", "occupation", "relationship", "race","sex", "capital_gain", "capital_loss", "hours_per_week" and "native_country". There are **6 continuous/numerical predictor** variables and **8 categorical variables.** According to the table of values, there are missing data on the observation's workclass, occupation, and native class. To summarize, missing data occurs when no data value is stored in an observation either with intent or not. These missing data can either have a big or small significant effect on the conclusion that can be drawn from the data. We will be omitting all the missing explanatory variables.

Data Overview continue. . .

Data Levels:

- **age**: continuous.
- **workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- **fnlwgt (final weight, which is the number of units in the target population that the responding unit represents)**: continuous.
- **education**: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. education-num: continuous.
- **marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- **occupation**: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex**: Female, Male.
- **capital-gain**: continuous.
- **capital-loss**: continuous.
- **hours-per-week**: continuous.
- **native-country**: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Data Overview continue. . .

Continuous Variables Data Overview

Here are some Boxplot graphs that can help us gauge some obvious analysis. The x-axis of the boxplot is set to be all one in order to look at the spread of the data. By looking at the spread of the explanatory variables using the histogram we see that capital gain, finalweight, and capital loss(Fig 1,2,3) don't have distinguishing features between adults who earn <=50K and >50K.

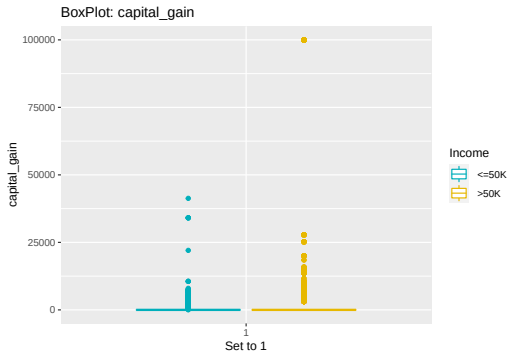

Figure 1: Box Plot Capital Gain

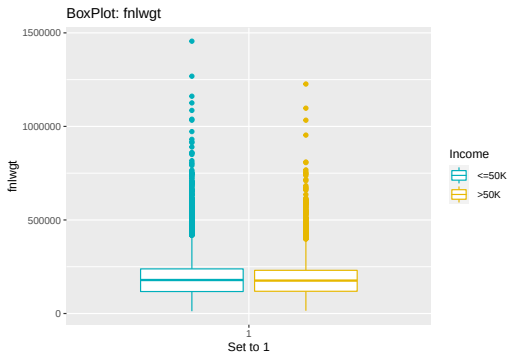

Figure 2: Box Plot Final Weight

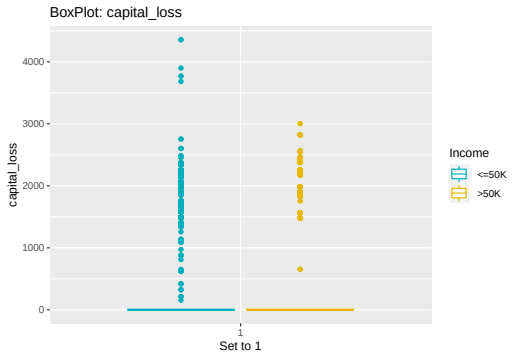

Figure 3: Box Plot : Capital Loss

Here are some graphs that can help us point out the obvious analysis. By looking at the spread of the explanatory variables using the boxplot we see that education number(Fig 5), and Age (Fig 4) have distinctive features that groups observations of Income earnings. We can see that observations with higher education number and higher age bumber seem to earn incomes >50K. Whereas, observations with a there are more adults with lower education number and age earning incomes <=50K.

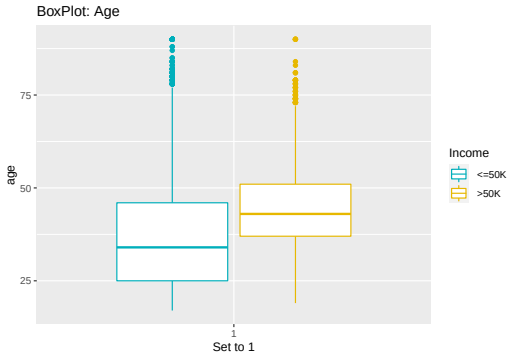

Figure 4: Box Plot : Age

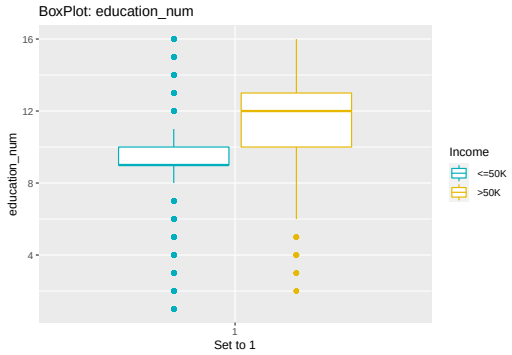

Figure 5: Box Plot : Education Number

Data Overview continue. . .

Categorical Variables Data Overview

It is obvious that the dataset has more observations under the income category "<=50K". The majority of both observations that earn <=50K and >50K are under the working class (Fig 6) "Private". The majority of observations earning ">50k" have education (Fig 7) some college, Prof-school, Masters, Hs-grad, Doctorate, and Bachelors. Majority of "<=50K" observations earned education from some-college, HS-grad, and Bachelors's. There are both a higher count of males earning both "<=50K" and ">50K" (Fig 8). People earning "<=50K" are mostly married-civ-spouse or never married (Fig 9). Most income ">50k" workers are married-civ-spouse. Most ">50k" workers have the occupation of Exec-management or Prof-specialty. Most "<=50K" workers have the occupationOFig 10) of adm-clerical, craft repair, or other services. ">50K" has to the relationship of husband. "<=50k" mostly has the relationship of husband and not in the family. Lastly, "<=50K" adults have a better spread for the count of native_country than ">50K" adults.

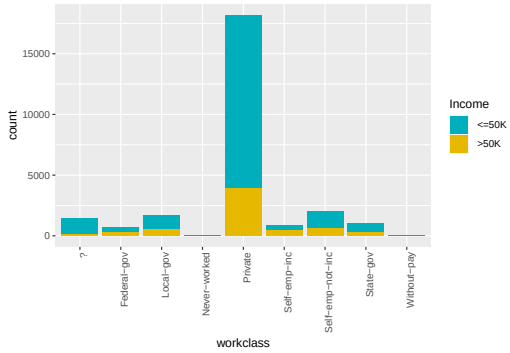

Figure 6: Bar Graph: WorkClass

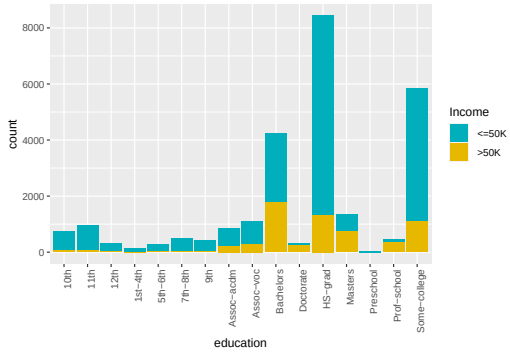

Figure 7: Bar Graph: Education

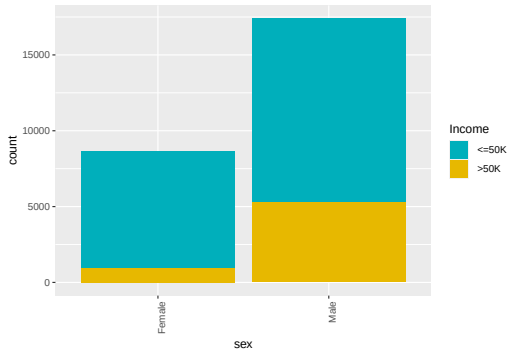

Figure 8: Bar Graph: Sex

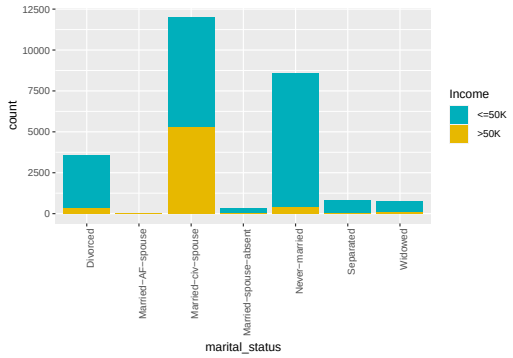

Figure 9: Bar Graph: Marital Status

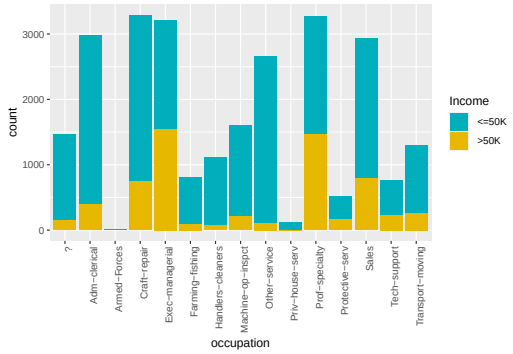

Figure 10: Bar Graph: Occupation

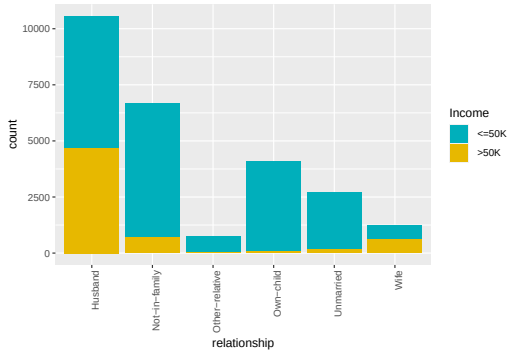

Figure 11: Bar Graph: Relationship

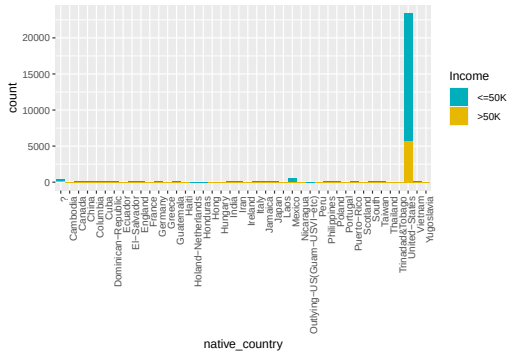

Figure 12: Bar Graph: Native Country

3.)Methods & Model Building

Method 1: Random Forest

For this method we will be using all categorical and numerical values as explanatory values.

- Type of random forest : classification
- number of predictors sampled for spliting at each nod = 3
- number of trees grown = 300

Methodology

We will use the importance of each variable to gauge whether it is efficient or useful to continue including it in our model. Since this is a classification random forest, we will use the mean decrease in Gini index as our measurement of importance. We chose random forest because it is an accurate and robust approach given our task of income classification. The main considerations we have with a random forest approach is concerning model complexity and the bias-variance tradeoff.

Training Error

	True		
Predicted	<=50K	>50K	Class.Error
<=50K	18397	1358	0.06874209
>50K	2222	4071	0.35309074

Testing Error

	True	
Predicted	<=50K	>50K
<=50K	4663	532
>50K	302	1016

Data Preparation & Modeling

Since we are using a random forest approach, we do not have to worry about scaling our data or pruning an individual decision tree because it averages the results of different decision trees. This helps with the issues of overfitting and high variance usually seen in single decision trees.

Model Building

In the context of classification trees, we can add up the total amount that the Gini index is decreased by splits over a given predictor, averaged over all trees (300).

We can see capital_gain is overwhelmingly the most important predictor in our model, given that it results in the largest decrease in Gini index. Our model tells us that capital gains are a telltale sign of indicating whether a given person's income is greater than 50,000 dollars. A given subject's age, family status, and fnlwgt, which is the number of people the census believes the entry represents, are fairly equal in their relative importance. We can easily see how a person's occupation and whether they have a multiple income household affects their income. Software engineers are more likely to have an income greater than 50,000 dollars than a food services worker and the same principle follows with a married person vs. a single person.

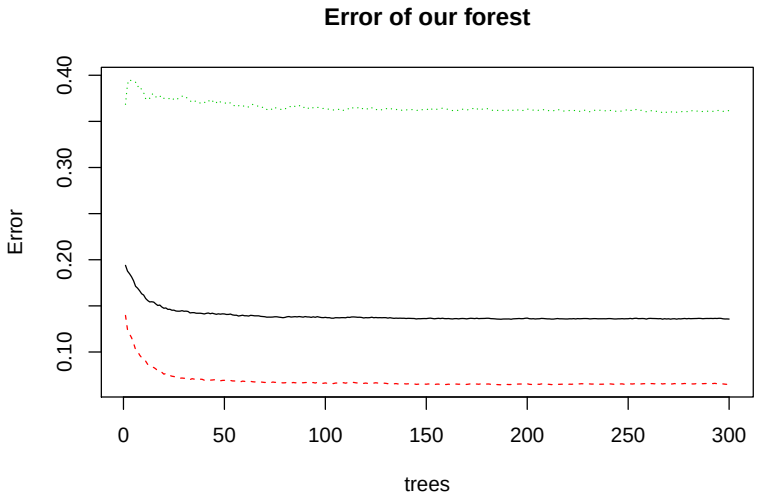

Figure 13: Error of our forest

Method 2: Knn (k-Nearest Neighbours)

Methodology

Knn is supervised learning that stands for K-neighest neighbor. It is a supervised learning method for classification. The K-nn algorithm stores all available cases and classifies based the count of training/modeled classified observations. The input k stands for k-closest training examples in the feature space. This method can work for our data because we are building a model to classify whether an observation earns an income of ">50K" or "<=50K." Because all categorical values are nominal we will be dropping categorical explanatory values and only using numerical explanatory values: age, fnlwgt, education_num,capital_gain ,capital_loss,hours_per_week. We will carry out model selection by choosing different K-values with the K-fold method. We will tune each model by comparing training errors and test errors.

Euclidean use to find the closest neighbor: distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Data Preparation & Modeling

Before performing kNN on the data, we first sample 80% of the observations as a training set and the other 20% as a test set. We have to train the kNN classifier on the training set and predict Income on the same training set, then we can construct the 2 by 2 confusion matrix to get the training error rate. Based on this idea, we should have train=XTrain, test=XTrain, and cl=YTrain in knn(). We worked with the first model when K=2.

Training Error (k=2)

	Observed	
Predicted	<=50K	>50K
<=50K	18253	1504
>50K	1572	4719

Train Accuracy Rate : 0.8819103

Train Error Rate : 0.1180897

Test Accuracy Rate : 0.7535698

Test Error Rate : 0.2464302

Model Building

To arrive to our final model we proceed K-fold for selecting the best number of neighbors. We will choose kth fold and divide the observation into kth folds. For each K the the first fold will be treated as a validation set and fit the remaining k-1 folds. Each time a different group of observation will be treated as a validation set and the rest k-1 fold as training set. The average of the test errors : $MSE_1, ..., MSE_k$ is the error rate.

We decided to do 3 fold on K values from 1 to 35.

Fold Resuls

```
##   fold  train.error val.error neighbors
## 1    1 0.0011517420 0.2415064         1
## 2    2 0.0008061730 0.2351993         1
## 3    3 0.0005182839 0.2360935         1
## 4    1 0.1167290527 0.2375907         2
## 5    2 0.1249568122 0.2369270         2
## 6    3 0.1162107688 0.2424277         2
```

	True	
Predicted	<=50K	>50K
<=50K	4894	1587
>50K	1	31

Best number of neighbor : 27

Test Accuracy Rate : 0.7561799

Test Error Rate : 0.2438201

Figure 15: Red=Train Error Blue= Test Error

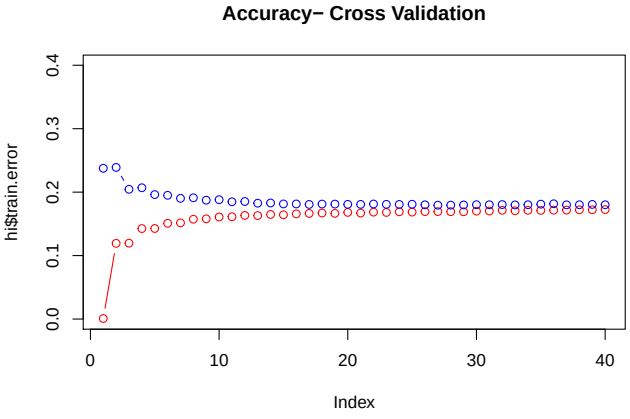

Figure 14: Accuracy Plot

4.) Conclusions

Method 1: Random Forest

This shows the importance of our predictors with higher values indicating greater importance with respect to model accuracy. It was interesting that the explanatory value age and educaiton mattered more than sex, native country, and race. The system does not seem to be bias to a certain quality of a adult other than the extra-ordinary amount of capital gaining.

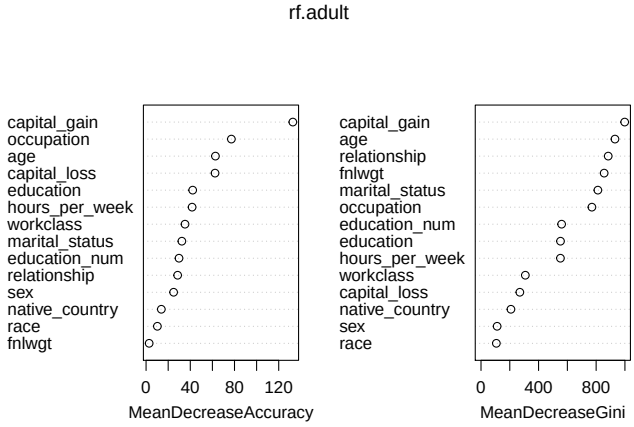

Figure 15: Variance Importance Plot

Method 2: Knn (k-Nearest Neighbours)

The test error is 24.64% when K=2. The test error when using the best fold is 24.36% when K=27. We can see than Random Forest will produce a better test accuracy due to the fact that it is able to use most and all of the categorical and numerical values as predictor values.

Compared to random forest, Knn can be used to make analysis of each new cases of observations. With the model we built we test for the average values of all explanatory values for this model with an outstanding capital gain value of 9999. Knn predicted that the 27 observations distantly close to this new observation had more observations earning >50K, therefore predicting the new observation to be also earning >50K. We hope to use this tool to quickly predict how new observations will follow our testing bias or not to make more analysis.

Average adult with outstanding capital gain.

- age = 38.62949
- final weight = 189465.4
- education number = 10.07908
- capital gain = 9999
- capital loss = 87.01728
- hours per week = 40.43669

Reference

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Appendix

```
#reading the data in
#setwd("~/Desktop/Pstat131/Project")
adult_data<-read.delim("adult.data.txt",sep=",",header = FALSE)
adult_data <-adult_data %>% dplyr::rename(age="V1",workclass="V2",fnlwgt="V3",education="V4",education_
smp_siz = floor(0.80*nrow(adult_data))
set.seed(123)
train_ind = sample(seq_len(nrow(adult_data)),size = smp_siz)
train =adult_data[train_ind,]
test=adult_data[-train_ind,]


ggplot(train, aes(x = factor(1), y = age )) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(aes(color = Income , shape = Income),
              width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL)   + ggtitle("Age")

ggplot(train, aes(x = factor(1), y = fnlwgt )) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(aes(color = Income , shape = Income),
              width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL) + ggtitle("fnlwgt")

ggplot(train, aes(x = factor(1), y = capital_loss )) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(aes(color = Income , shape = Income),
              width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL) + ggtitle("capital loss")



ggplot(train, aes(x = factor(1), y = education_num )) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(aes(color = Income , shape = Income),
              width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL)  + ggtitle("education_num")

ggplot(train, aes(x = factor(1), y = capital_gain )) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(aes(color = Income , shape = Income),
              width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL)  + ggtitle("capital_gain")



ggplot(train, aes(workclass, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = element_te
```

```
ggplot(train, aes(education, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = element_te
ggplot( train, aes(sex, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = element_text(an
ggplot(train, aes(marital_status, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = eleme
ggplot(train, aes(occupation, ..count..)) + geom_bar(aes(fill = Income),)+ theme(axis.text.x = element_
ggplot(train, aes(relationship, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = element
ggplot(train, aes(native_country, ..count..)) + geom_bar(aes(fill = Income))+ theme(axis.text.x = eleme
```

```
set.seed(123)
#growing a beautiful forest
rf.adult = randomForest(Income~.,data=train,mtry=3,ntree=300,importance=TRUE)
rf.adult

#plot the error of our forest

#predict using RF
yhat.rf = predict(rf.adult,newdata = test)

#confusion matrix
rf.err = table(pred = yhat.rf, truth = test$Income)

test.rf.err = 1 - sum(diag(rf.err))/sum(rf.err)

test.rf.err

#importance
importance(rf.adult)
```

```
#reading the data in
#setwd("~/Desktop/Pstat131/Project")
adult_data<-read.delim("adult.data.txt",sep=",",header = FALSE)
adult_data <-adult_data %>% dplyr::rename(age="V1",workclass="V2",fnlwgt="V3",education="V4",education_
```

```
smp_siz = floor(0.80*nrow(adult_data))
set.seed(123)
train_ind = sample(seq_len(nrow(adult_data)),size = smp_siz)
train =adult_data[train_ind,]
test=adult_data[-train_ind,]

adult_data$age<-as.numeric(adult_data$age)
adult_data$fnlwgt<-as.numeric(adult_data$fnlwgt)
adult_data$education_num<-as.numeric(adult_data$education_num)
adult_data$capital_gain<-as.numeric(adult_data$capital_gain)
adult_data$capital_loss<-as.numeric(adult_data$capital_loss)
```

```
train =adult_data[train_ind,]
test=adult_data[-train_ind,]

YTrain = train$Income
XTrain = train %>% select( age, fnlwgt, education_num,capital_gain ,capital_loss,hours_per_week )



XTrain <- scale(XTrain,center = TRUE, scale = TRUE)
str(XTrain)

meanvec <- attr(XTrain,'scaled:center')
sdvec <- attr(XTrain,'scaled:scale')


YTest = test$Income
XTest = test %>%  select( age, fnlwgt, education_num,capital_gain ,capital_loss,hours_per_week )


pred.YTtrain = knn(train=XTrain, test=XTrain, cl=YTrain, k=2)
# Calculate confusion matrix
conf.train = table(predicted = pred.YTtrain, observed=YTrain)
conf.train



# Train accuracy rate
sum(diag(conf.train)/sum(conf.train))


# Train error rate
1 - sum(diag(conf.train)/sum(conf.train))

######
######
# Test Error


# knn - train the classifier on TRAINING set and make predictions on TEST set!
pred.YTest = knn(train=XTrain, test=XTest, cl=YTrain, k=2)
# Get confusion matrix
conf.test = table(predicted=pred.YTest, observed=YTest)
conf.test

# Test accuracy
sum(diag(conf.test)/sum(conf.test))


# Test error rate
1 - sum(diag(conf.test)/sum(conf.test))
```

```r
do.chunk <- function(chunkid, folddef, Xdat, Ydat, ...){ # Function arguments
  train = (folddef!=chunkid) # Get training index
  Xtr = Xdat[train,] # Get training set by the above index
  Ytr = Ydat[train] # Get true labels in training set
  Xvl = Xdat[!train,] # Get validation set
  Yvl = Ydat[!train] # Get true labels in validation set
  predYtr = knn(train=Xtr, test=Xtr, cl=Ytr, ...) # Predict training labels
  predYvl = knn(train=Xtr, test=Xvl, cl=Ytr, ...) # Predict validation labels
  data.frame(fold = chunkid, # k folds
             train.error = mean(predYtr != Ytr), # Training error for each fold
             val.error = mean(predYvl != Yvl)) # Validation error for each fold
}


nfold = 3
set.seed(123)
folds = cut(1:nrow(train), breaks=nfold, labels=FALSE) %>% sample()
folds

error.folds = NULL
# Give possible number of nearest neighbours to be considered
allK = 1:40
# Set seed since do.chunk() contains a random component induced by knn()
set.seed(123)
# Loop through different number of neighbors
for (j in allK)
{
  tmp = ldply(1:nfold, do.chunk, # Apply do.chunk() function to each fold
              folddef=folds, Xdat=XTrain, Ydat=YTrain, k=j)
  # Necessary arguments to be passed into do.chunk
  tmp$neighbors = j # Keep track of each value of neighors
  error.folds = rbind(error.folds, tmp) # combine results
}


dim(error.folds)

head(error.folds)



# Transform the format of error.folds for further convenience
errors = melt(error.folds, id.vars=c('fold', 'neighbors'), value.name='error')
# Choose the number of neighbors which minimizes validation error
val.error.means = errors %>%
  # Select all rows of validation errors
  filter(variable=='val.error') %>%
  # Group the selected data frame by neighbors
  group_by(neighbors, variable) %>%
  # Calculate CV error rate for each k
  summarise_each(funs(mean), error) %>%
  # Remove existing group
  ungroup() %>%
```

```
  filter(error==min(error))


# Best number of neighbors
# if there is a tie, pick larger number of neighbors for simpler model
numneighbor = max(val.error.means$neighbors)
numneighbor


set.seed(123)
pred.YTest = knn(train=XTrain, test=XTest, cl=YTrain, k=numneighbor)
# Confusion matrix
conf.matrix = table(predicted=pred.YTest, true=YTest)
# Test accuracy rate
sum(diag(conf.matrix)/sum(conf.matrix))

1 - sum(diag(conf.matrix)/sum(conf.matrix))




plot(rf.adult)
plot(error.folds$neighbors ,error.folds$val.error,type='o',ylim=c(0,.5),xlab="k",ylab="Test Error",col=
lines(error.folds$neighbors ,error.folds$val.error,type='o',col="red")

###############
## KNN NEW TEST (Conclusion Analysis )
##############

testt<-c(38.62949,189465.4,10.07908,9999,87.01728,40.43669)

pred.YTest = knn(train=XTrain, test=testt, cl=YTrain, k=numneighbor)


testt<-c(38.62949,189465.4,10.07908,1086.914,87.01728,40.43669)

pred.YTest = knn(train=XTrain, test=testt, cl=YTrain, k=numneighbor)
```