# Assignment 3

(Team 6)

**Goal**: Predicting whether a person makes over 50K a year.

**Sources for Data and its descriptions**: UCI Machine Learning Repository
1. https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data
2. https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names

**Key findings/Summary/Results:**

As the error summary table below shows, Ridge regression outperforms all the other methods, other than kNN. kNN has high flexibility so that a lower prediction error is expected; however, if we consider model simplicity, simple logistic regression is preferred. On top of that, since ridge regression is trading a little biasness to smaller variance, ridge gives us smaller error rate in total.

**Error Summary Table**

| Model | Training error | Test error |
|---|---|---|
| Logistic Regression | 15.27% | 15.81% |
| Lasso Regression | 15.33% | 16.22% |
| Ridge Regression | 15.26% | 15.16% |
| LDA | 16.48% | 16.09% |
| QDA | 19.69% | 19.77% |
| RDA | 16.23% | 16.64% |
| kNN | 13.67% | 13.48% |

**Methods and Analysis:**

**1. Data preparation**

After importing data from the url, several manipulations were taken action to make a readily dataset for building a predictive modeling on.

1) Treating unknown values
We detected “?”s in the data which accounts for less than 7 percent of the total observations (30,162) and removed them. Since we believe that deleting them would make a little difference

and the data description (source 2) implemented the same process, it is justifiable to remove them.

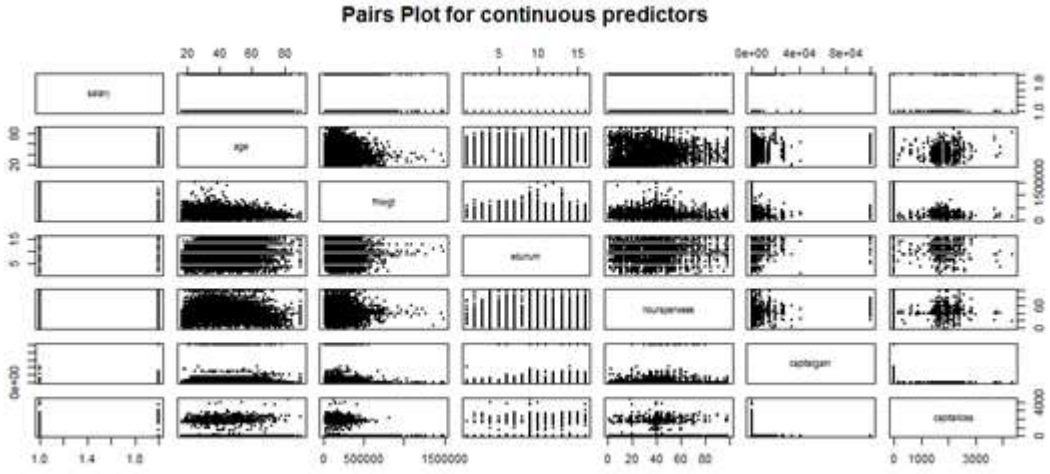
2) Checking outliers of numeric variables
Checking outlier is an important step prior to modeling since outlier could distort our result. In this regard, using “outliers” package in R, outlier checking procedure was implemented on numeric variables such as ”fnlwgt”, “age”, “education number of year”, “capital gain”, “capital loss” and “hours per week3”. There were no obvious outliers detected so no further action was required.

3) Constructing a sparse matrix for categorical predictor variables
We have categorical predictor variables in our dataset. Theses are usually first transformed into factors, then a dummy variable matrix of predictors is created and along with the continuous predictors, is passed to the model.

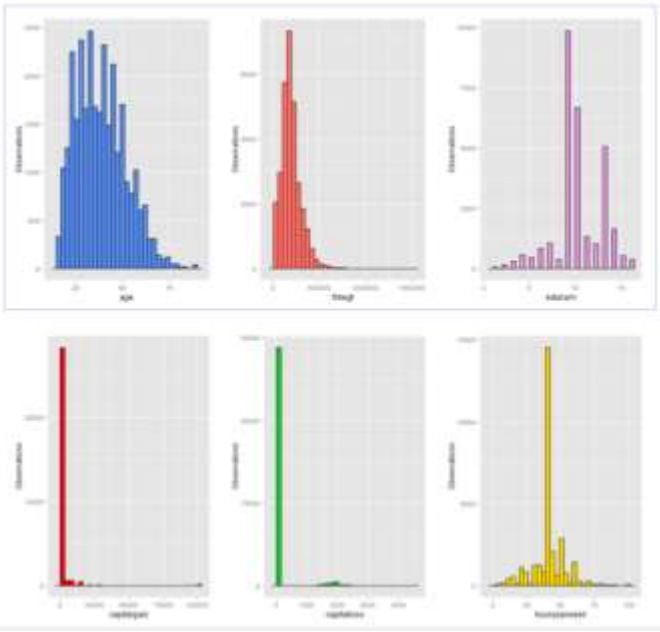
**2. Exploratory data analysis**

As requested, the first half of the data was used as a training set to explore data further. Since our goal is to predict, if a person has his/her salary over $50,000, we should first take a look at our response vector. Those whose salary is higher than $50,000 accounts for 24.89 percentage and it implies that “salary” vector is unbalanced. Thus, we fitted models that can deal with the unbalanced data with priors, such as Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) in addition to logistic regression model, widely used for a binary response.

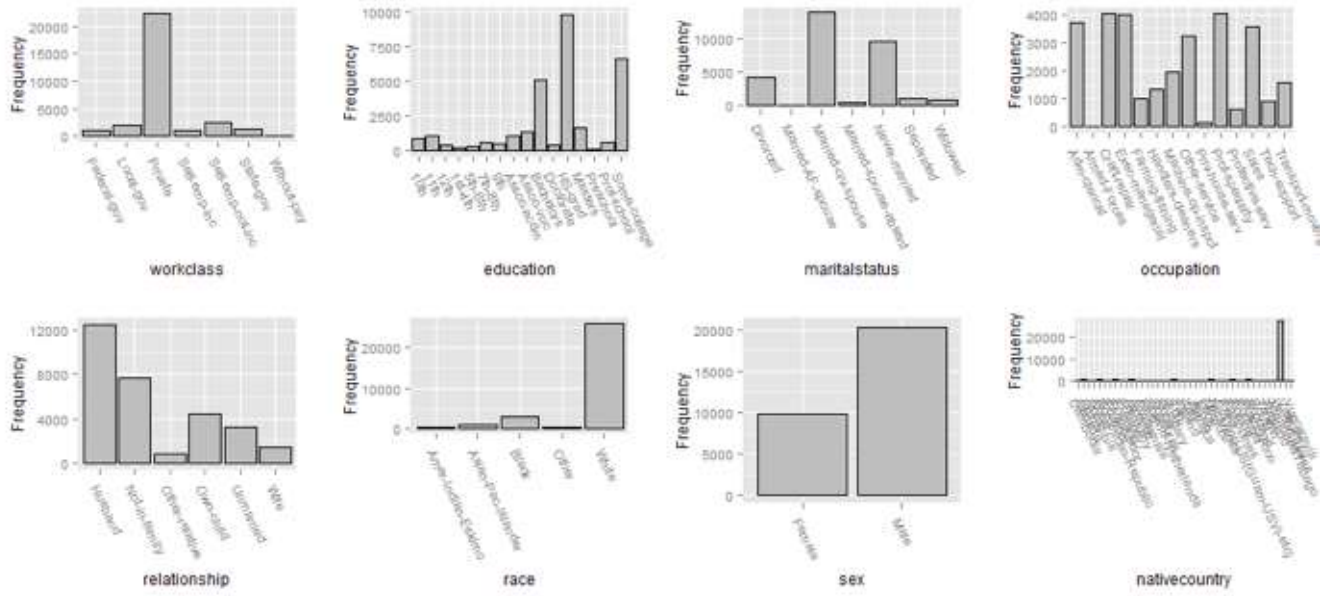
It is important to see if there exists some specific patterns between the response variable and predictors. As seen in the scatter matrix below, we found that Final weight (fnlwgt) and “age” are highly correlated and so are final weight and capital loss. It is perhaps because final weight is partially controlled by “age”. Other than that, no particular pattern was found.




We wanted to check if our continuous variables are distributed as Normal for balance check. For this purpose, it is believed best to draw histogram.



From the plots below, we could see that among our continuous predictors, both “age” and “weight” skew to the left. “Education number” trend skews to right. The rest three predictors appear not to follow Gaussian distributions. As a result, we suspected that our data does not follow Gaussian distribution and this could mean LDA and QDA models may not be a good fit for our data.




With regard to eight qualitative predictors, our initial approach was to use methods that handle disjoint variables rather than to use simple linear regression model. These models supposedly work well as our data displays disjointed pattern on qualitative predictors as shown above.

**3. Predictive modeling**

We explored models including Logistic Regression, Lasso, Ridge, Linear Discriminant Analysis, Quadratic Discriminant Analysis, and K Nearest Neighbors, Principal Component Analysis Regression. Our members cross-checked each other's work using github and decided on logistic model which provides the lowest test error for the test set. It is also prefered because of the simplicity (and interpretability). Our chosen model method is mainly discussed in the report, although other attempted methods are also mentioned briefly.

1. Logistic Regression result

We first used the function glmnet in the package "glmnet" (we denote the response as salary more than $50,000), and produced a logistic regression model with overall training misclassification rate of 15.81% and overall test misclassification rate of 15.84%. Since the goal of our model is to predict whether a person makes over $50,000 a year, we need to examine specifically the proportion that people were misclassified to the group over $50,000. We get the training misclassification rate is 46.41% and test is 46.11% within the group, which is not so good.

Thus we take advantage of function cv.glmnet to find the best lambda for the best subset. From the fit we get the minimal lambda=0.0001957. Plug this value in the previous model we can get the overall training misclassification rate is 15.27% and test misclassification rate is 15.81% . Again, we check the error in high salary group and we obtain training misclassification rate of 40.18% and test of 39.68%, which are much better than before. This result comes from the unbalanced proportion of salary group.

Training error of logistic regression

| | Training | |
|---|---|---|
| Prediction | 0 | 1 |
| 0 | 10699 | 1730 |
| 1 | 654 | 1998 |

Test error of logistic regression

| | Test | |
|---|---|---|
| Prediction | 0 | 1 |
| 0 | 10655 | 1743 |
| 1 | 646 | 2037 |

Training error of logistic regression with penalty (lambda)

| | Training | |
|---|---|---|
| prediction | 0 | 1 |
| 0 | 10548 | 1498 |
| 1 | 805 | 2230 |

Test error of logistic regression with penalty (lambda)

| | Test | |
|---|---|---|
| prediction | 0 | 1 |
| 0 | 10512 | 1500 |
| 1 | 789 | 2208 |

2. Lasso Regression
Using 'glmnet' with setting 'alpha' = 1, we can build a Lasso model. Then, cross-validation is used to select the optimal λ. We are hoping to shrink the coefficients, even as far as reduce number of predictors, by introducing the lasso penalty term.

3. Ridge Regression
Similarly, we use 'glmnet' with setting 'alpha' = 0 to build a Ridge Regression. Then, And we use cross-validation to select the optimal λ. We are hoping to shrink the coefficients by introducing penalty term.

4. Linear Discriminant Analysis
As for prediction on social science, predict error around 15% is already a really good prediction. However, since our goal is to predict "whether a person makes over 50K a year". And when we looked in detail as the type 1 error on our test data set, we found type 1 error is 43.26%. 1635 out of 3780 people with salary over 50K have been misclassified. Therefore, comparing to logistic model, LDA performs worse.

5. Quadratic Discriminant Analysis
Just like LDA: 2982 of test samples were misclassified thus the test error rate is 0.1977322. And 2970 of train samples were misclassified thus the train error is 0.1969365. It's even worse than LDA. This maybe because a lot of coefficients need to be estimated inside QDA model,especially for high dimension. As usual, the tradeoff between LDA and QDA is one of bias and variance: LDA makes stronger assumptions and obtains estimates with lower variance Besides, if take detailed look at QDA prediction, we could also find the type 1 error is 65%. 2459 out of 3780 people with salary over 50K have been misclassified. It's really bad prediction so we will not use QDA for our prediction.

6. RDA (Regularised discriminant analysis)
We also tried RDA model on our data. However, since RDA is basically a combination of LDA and QDA. And neither of these models works well. Therefore the result didn't show up well also. We got misclassification rate around 16.5% and still bad prediction on our goal.
Machine Learning Algorithms for Classification
Much like regression, there are problems where linear methods don't work well for classification.

7. K Nearest Neighbors
First step, we try k=1 first. There are 14.5% of the observations in the test set are incorrectly predicted. Of course, it may be that K = 1 results in an overly flexible fit to the data and increase the variance. As we know, as K increases, the MSE of test data is a U-shape. So later, we repeat the analysis with different K to find the optimal one. We then try K=5, 10,15,20,25 and 30 respectively and narrow the range between 25 and 30. In order to find the optimal K that minimize misclassified rate of the test data, we try every integer from 26 to 29 as K. When K=27, we get the min test error of 13.67%. And the train error is 13.48%.

8. PCA (Principal Component Analysis)

We apply PCA to extract features of the original high dimensional data. Then we get the scores with dimensions 15081*33 instead of the original 15081*105 data. Later, we tried linear regression and get the misclassified rate for test data 17.21%. It seems not as good as other models, but better than the linear regression using original data.

**4. Essential R scripts**

[insert R scripts that is important]

1. Data munging

```
# Importing data from URL#
theURL<-getURL("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data")
data=read.table(text=theURL, header=FALSE, sep=",", stringsAsFactors=FALSE)
names(data)=c("age","workclass","fnlwgt","education","edunum","maritalstatus","occupation","
relationship","race","sex","capitalgain","capitalloss","hoursperweek","nativecountry","salary")

# Removing “?” and checking outliers #
check<-(data==" ?")
answer<-which(check,arr.ind=TRUE)
delete<-unique(answer[,1])
data=data[-delete,]
chisq.out.test(data$fnlwgt, variance=var(data$fnlwgt), opposite = FALSE)

# Create dummy variables #
require("caret")
```

Same R scripts for creating dummy variables of ‘age’, ‘capital gain’, ‘capital loss’, and ‘hoursper week’

```
data$age=as.character(data$age)
age=data$age
agdummy=dummyVars(~age,data)
agmatrix=predict(agdummy, newdata=data)
```

Same R scripts for creating dummy variables of ‘work class’, ‘education’, ‘education number’, ‘marital status’, ‘occupation’, ‘relationship’, ‘sex’, ‘native country’, and ‘salary’

```
sparse=cbind(agmatrix,wcmatrix1,edumatrix,enmatrix,msmatrix,ocmatrix1,rsmatrix,rcmatrix,sex
matrix,cgmatrix,clmatrix,hwmatrix,ncmatrix1)

# 0 for <=50K, 1 for >50K
y=samatrix[,2]
y=as.factor(y)
trainy=y[1:15081]
testy=y[15082:30162]
train.sparse=sparse[1:15081,]
test.sparse=sparse[15082:30162,]
train=data[1:15081,]
```

```
test=data[15082:30162,]

# 1. Ridge

fit1=glmnet(train.sparse,trainy,family="binomial", alpha=0)
print(fit1)

plot(fit1, xvar = "dev", label = TRUE)
plot(fit1, xvar = "lambda", label = TRUE)
plot(fit1, xvar = "norm", label = TRUE)

# using CV for choosing best lambda
cvfit1 <- cv.glmnet(train.sparse,trainy,family = "binomial", type.measure = "class", alpha=0)
plot(cvfit1)
cvfit1$lambda.min
cvfit1$lambda.1se
coef(cvfit1, s = "lambda.min")

pred1=predict(cvfit1,newx=train.sparse, s = "lambda.min", type ="class", alpha=0)
table(pred1,trainy)
pred2=predict(cvfit1,newx=test.sparse, s = "lambda.min", type ="class", alpha=0)
table(pred2,testy)
```