hackathon

dharmik

10 November 2017

2.Define your data exploration, imputation and visualization approach.

```
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

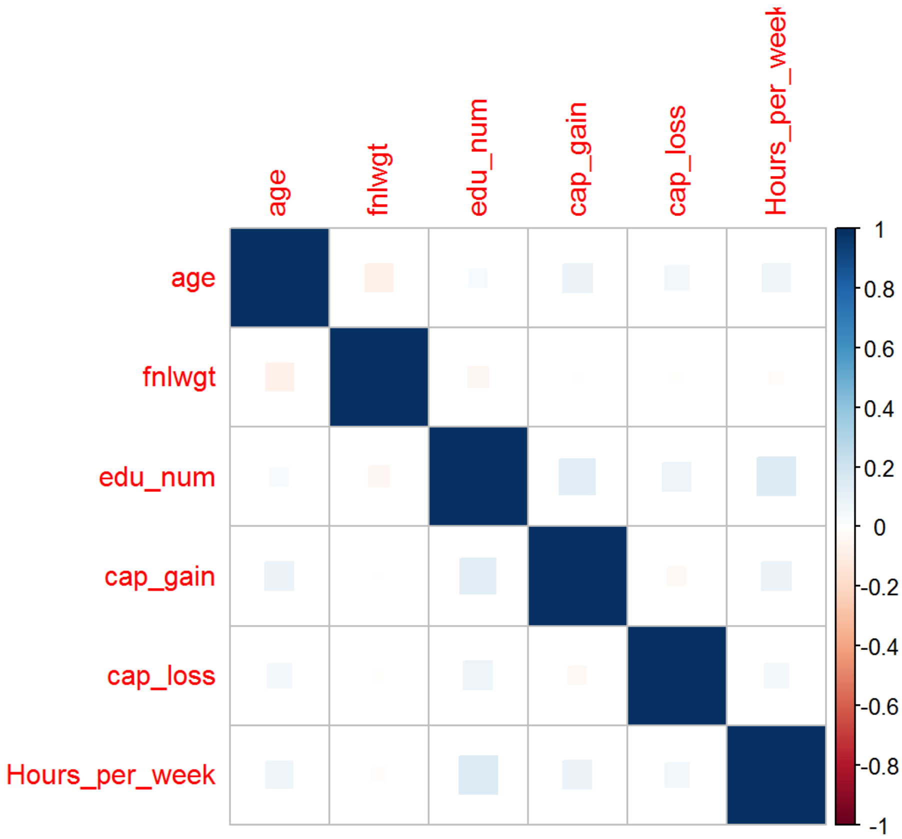
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(corrplot)
```

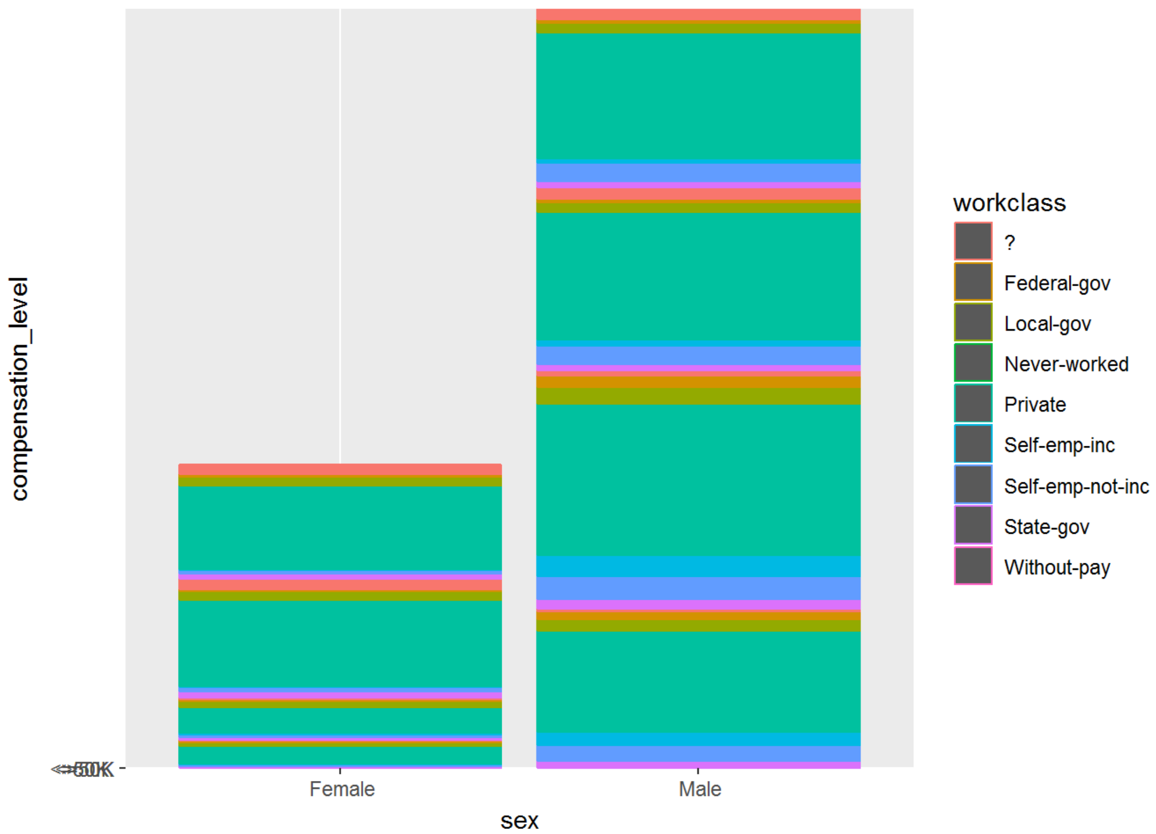
```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

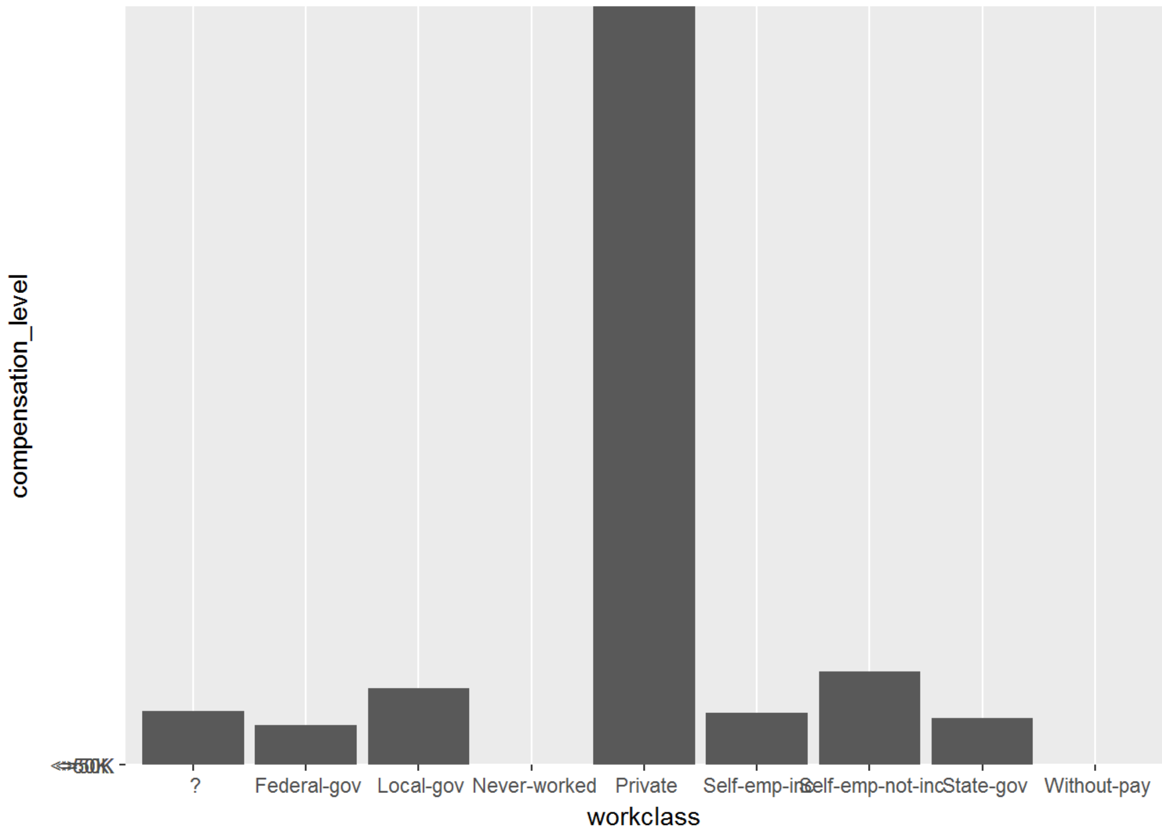
```
setwd("D:/data science term sylabus/2nd term/ML/hackathon")

model_data <- read.csv("Model_Data.csv")


#correlation on integer variables
cor_values <- model_data %>% select(age,fnlwgt,edu_num,cap_gain,cap_loss,Hours_per_week)

corr <- cor(cor_values)
corrplot(corr,method = "square")
```




```
# most of attribute are moderately correlated with each other

ggplot(model_data,aes(x =sex ,y = compensation_level,color=workclass))+geom_bar(stat = "identity"
)
```

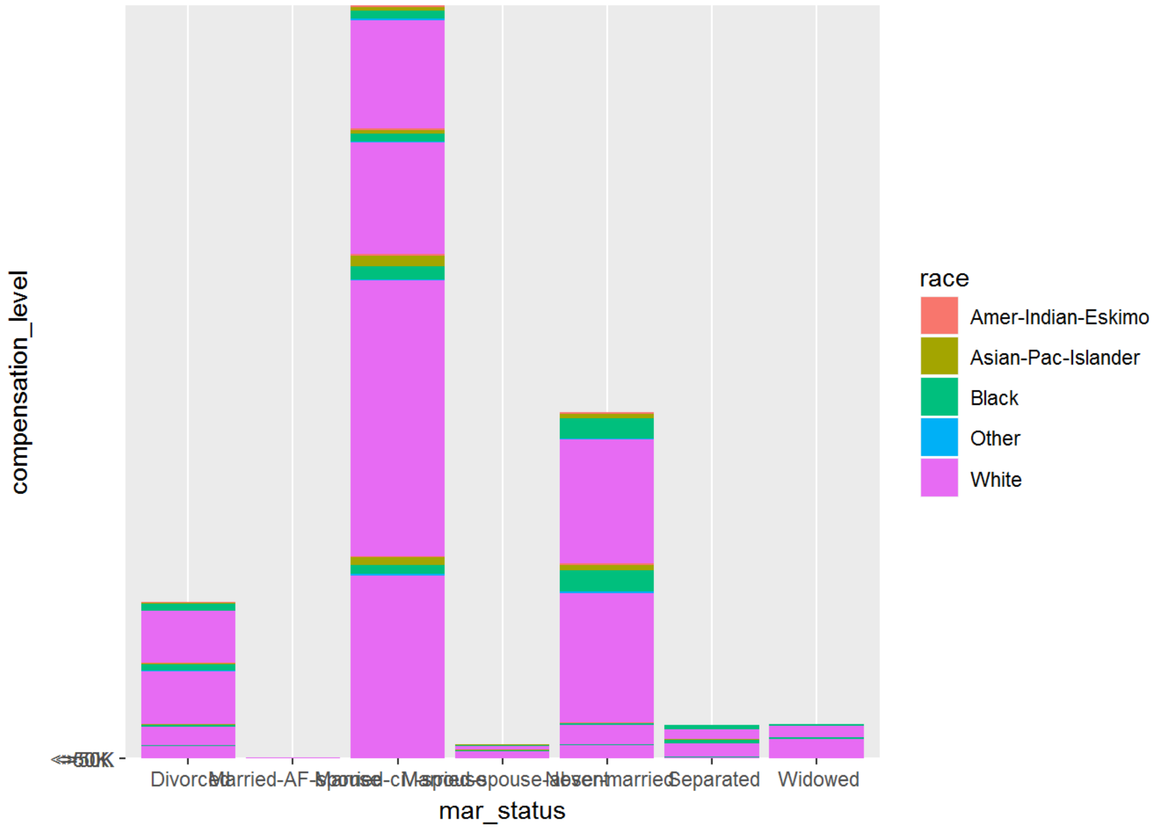


```r
      # from plot we get the insight that less number of female get compensation more than 50k and highest count of male get more than           50k compensation
      # gender working in private sector company gets compensation greater than 50k compare to genders working in local-gov and                  self-emp-not-inc


ggplot(model_data,aes(x =workclass ,y = compensation_level))+geom_bar(stat = "identity")
```




```r
# employees working in private sector falls in category of "compensation more than 50k" compare to local-gov and self-emp-not-inc

ggplot(model_data,aes(x=mar_status,y=compensation_level,fill=race))+geom_bar(stat = "identity")
```




```
# married-civ-spouse gets higher compensation level and making comparison in people race we get to
know that white people get more compensation compare to other race people
```

DATA IMPUTATION

model_new_data <- read.csv(“Model_Data.csv”)

model_new_data$workclass < -gsub(" ? " , NA, model_new_data$workclass, fixed = TRUE) model_new_data $occupation < -gsub(" ? " , NA, model_new_data$occupation, fixed = TRUE)

model_new_data$occupation[is.na(model_new_data$occupation)]=Mode(model_new_data$occupation)model_new_data$ workclass[is.na(model_new_data$workclass)]=Mode(model_new_data$workclass) Mode <- function(v) { uniqv <- unique(v) uniqv[which.max(tabulate(match(v, uniqv)))] }

model_new_data$comp_new_level < -gsub(" . " , "" , model_new_data$compensation_level) model_new_data $comp_new_level < -as.factor(model_new_data$comp_new_level) model_new_data <- model_new_data[,-c(14,15)]
##

`DECISION TREE`

6.Build 3 Models, each using one of different type of algorithm. Send me the model building command.

```` `MODEL 1 = DECISION TREE`

set.seed(28) train_data <- sample.int(n = nrow(model_new_data), size =floor(0.8 *nrow(model_new_data)),replace = F)

train <- model_new_data[train_data,] test <- model_new_data[-train_data,]

train_model <- tree(comp_new_level ~.,train)

plot(train_model) text(train_model)

check_model <- predict(train_model,test) check_model

maxidx <- function(arr) { return(which(arr==max(arr))) } idx <- apply(check_model,c(1),maxidx) predict_model <- c(‘No’,‘Yes’)[idx]

confmat <- table(predict_model,test$comp_new_level) confmat confusion matrix

accuracy <- sum(diag(confmat))/sum(confmat) accuracy

ACCURACY = 0.8388021
````

` `MODEL 2 =Naive Bayes`

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.4.2
```

```
model_new_data <- read.csv("Model_Data.csv")
model_new_data$comp_new_level <- gsub("\\.", "",model_new_data$compensation_level)
model_new_data$comp_new_level <- as.factor(model_new_data$comp_new_level)
model_new_data <- model_new_data[,-15]
set.seed(28)


sample <- sample.int(n=nrow(model_new_data),size = floor(0.8*nrow(model_new_data)),replace = F)

train_data <- model_new_data[sample,]
test_data <- model_new_data[-sample,]

model <- naiveBayes(comp_new_level ~ age+workclass+fnlwgt+edu+edu_num+mar_status+occupation+race+
Hours_per_week+country,data = train_data)
#model

pred <- predict(model,test_data)
#pred
#chechking and creating conf matrix with pred values and labelled variable values
confmat <- table(pred,test_data$comp_new_level)
confmat
```

```
##
## pred      <=50K  >50K
##    <=50K   5082   687
##    >50K     724  1187
```

```
#checking accuracy
accuracy <- sum(diag(confmat))/sum(confmat)
accuracy
```

```
## [1] 0.816276
```

````````````MODEL 3=kNN`````````````

```
library(class)
set.seed(28)
sample <- sample.int(nrow(model_new_data),size = floor(0.80*(nrow(model_new_data))),replace = FAL
SE)

train <- model_new_data[sample,c(1,3,5,11,12,13)]
test <- model_new_data[-sample,c(1,3,5,11,12,13)]
train_label <-model_new_data[sample,14]
test_label <-model_new_data[-sample,14]

k=5
pred_label <- knn(train = train,test = test,cl =train_label,k )

confmat=table(test_label,pred_label)

accuracy <- sum(diag(confmat))/sum(confmat)
accuracy
```

```
## [1] 0.8933594
```
```

Country column didnt had any major impact on labelled variable so have ignored column to build the model

7.Predict your model performance on each of the 3 models and submit (1 mark each = total 3 marks) model1_accuracy= 0.8388021 model2_accuracy =0.816276 modell3_accuracy=0.8933594

8.Generalization:-

[illegible]0 % so my model are not underfit and overfit