Final Quiz

Modern Data Mining

April 30, 2019

Instruction: This is an open book, 30-minute quiz.

Note: We have shortened the quiz significantly at the end. This version should be used for the purpose of studying the final quiz. All the lectures will be covered. You can tell on the other hand we focus on materials not covered in our midterm. Neural network questions are newly added.

The quiz is divided into two parts: concepts and data analysis.

Statistics Concepts

1. Linear regression with one covariate is always better than linear regression with no covariates.

(A) TRUE

(B) FALSE

(C) It can be either way depending on the data.

(B). Depends on the criterion

2. For linear regression with more than 10 covariates (or independent variables) which of the following is TRUE?

(A) LASSO (with `lambda.1se`) and C_p will choose the same set of variables.

(B) LASSO (with `lambda.min`) and C_p will choose the same set of variables.

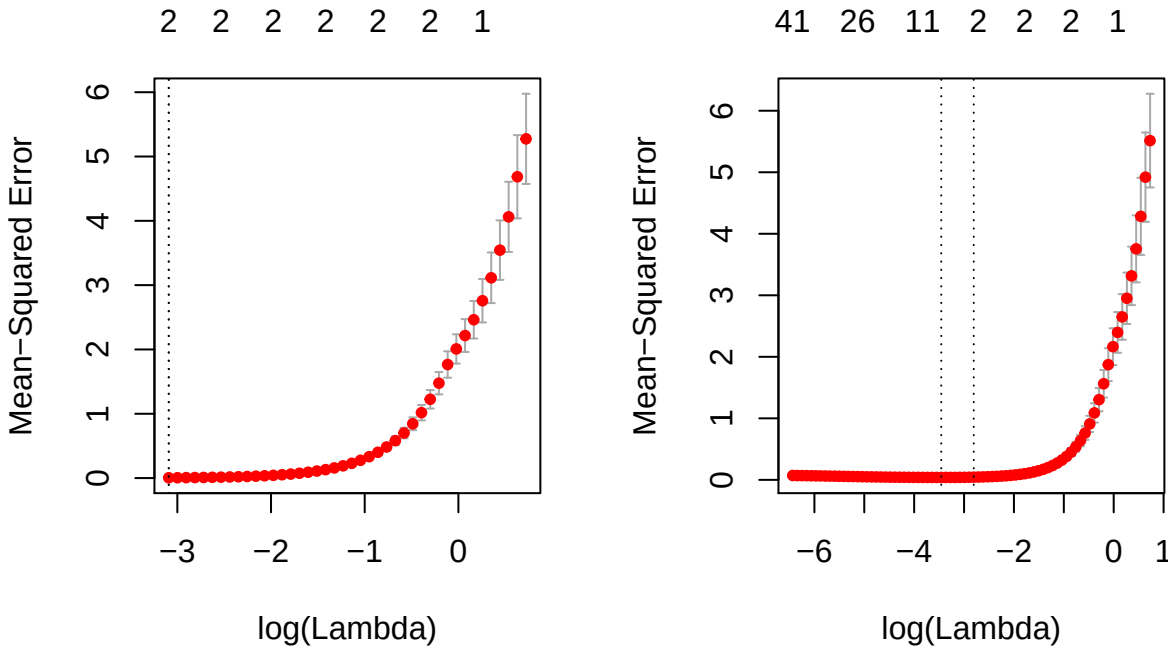
(C) C_p estimates the testing error but not prediction error.

(D) The model with the smallest C_p is expected to perform well on testing data.

(E) None of the above is TRUE.

(D)

```
set.seed(471)
x <- matrix(rnorm(100 * 50), nrow = 100, ncol = 50)
beta0 <- c(1, 2, rep(0, 48))
## data 1 with 0 error
y1 <- x %*% beta0 + rnorm(100, 0, 0)
fit.lasso1 <- cv.glmnet(x, y1)
## data 2 with 0.2 error variance
y2 <- x %*% beta0 + rnorm(100, 0, 0.2)
fit.lasso2 <- cv.glmnet(x, y2)
par(mfrow = c(1, 2))
plot(fit.lasso1)  # main = 'Lasso with No Error'
plot(fit.lasso2)  # main = 'Lasso with non-zero Error'
```



Problem **3.** to **7.** are based on the model generated from the above chunk.

3. The above code generates two response vectors (`y1` and `y2`) one with no error and one with non-zero error.

(A) TRUE

(B) FALSE

(C) More Information Needed.

(A). In fact $y_1 = x_1 + 2x_2$ and $y_2 = x_1 + 2x_2 + \epsilon$

4. For both the responses there are only two truly significant variables.

(A) TRUE

(B) FALSE

(C) Need to see summary of `lm()` to answer this.

(A).

```
coef.min1 <- coef(fit.lasso1, s = "lambda.1se")
coef.min1[which(coef.min1 != 0), ]
```

```
## (Intercept)          V1          V2
##    -0.00742     0.95133     1.95425
```

```
coef.min2 <- coef(fit.lasso2, s = "lambda.1se")
coef.min2[which(coef.min2 != 0), ]
```

```
## (Intercept)          V1          V2
##      0.0194      0.9814      1.9559
```

5. LASSO picked the right model in both cases.

(A) TRUE

(B) FALSE

(A).

6. If we fit `lm()` using the covariates chosen by `fit.lasso2` then the variable `V2` will turn out to be statistically insignificant at level 0.05.

(A) TRUE

(B) FALSE

(C) Cannot tell definitely.

C)

7. In the code above changing `set.seed(471)` to `set.seed(571)` will again result in the same set of variables chosen by `fit.lasso1` and `fit.lasso2`.

(A) TRUE

(B) FALSE

(B)

8. Which of the following is preferred for a good classification rule?

(A) sensitivity: **low**, specificity: **high**, MCE: **low**.

(B) sensitivity: **high**, specificity: **high**, MCE: **low**.

(C) sensitivity: **low**, specificity: **low**, MCE: **high**.

(D) sensitivity: **high**, specificty: **low**, MCE: **high**.

(E) The correct configuration is not listed above.

(B)

9. A classifier with smallest training MCE (misclassification error) is preferred to one with smaller testing MCE.

(A) TRUE

(B) FALSE

B)

10. OOB in random forests provides an estimate of training error.

(A) TRUE

(B) FALSE

(C) It can be used as an estimate of both training and testing errors.

(B)

11. Suppose we have a sample -0.651, 0.350, 0.497, 0.313, -0.602. Which of the following cannot possibly be a bootstrap sample?

(A) -1.478, -0.651, 0.497, 0.313, -0.602

(B) -0.651, -0.651, -0.651, -0.651, -0.651

(C) 0.313, -0.651, -0.602, 0.35, 0.497

(D) -0.651, -0.651, 0.35, 0.35, 0.35

(E) All the above cannot be bootstrap samples.

(A) A bootstrap sample can only take values from the original sample.**

12. The idea of principal components is to create new variables from old ones while most of the variance into a smaller set of new variables.

(A) TRUE

(B) FALSE

(A)

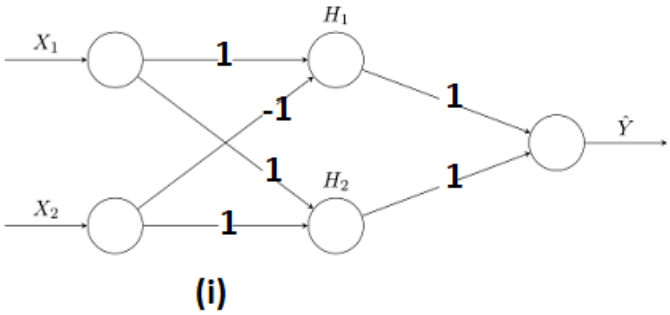
13. `PC1` has smaller variance than `PC2` for any dataset.

(A) TRUE

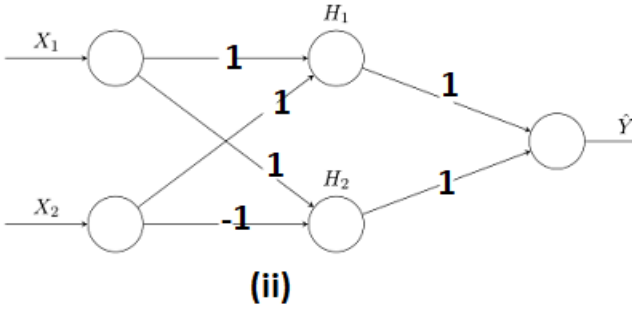
(B) FALSE

(B) `PC1` has the largest variance among all `PC's`.

14. Which of the following is the correct NN architecture for the function $\sigma(X_1 + X_2) + \sigma(X_1 - X_2)$ where σ is the sigmoid function?



(i)



(ii)

(A) Only (i) is correct

(B) Only (ii) is correct

(C) Both are correct

(D) None is correct

C).

15. C_p increases by adding further covariates in linear model.

(A) TRUE

(B) FALSE

(B)

Data Analysis: Wisconsin Breast Cancer Data

The data `train_wdbc` contains 30 predictors and the response `diagnosis` is a binary variable with `M=malignant` and `B=Benign`.

```
train_wdbc <- read.csv("breast-cancer.csv")[, -c(1, 33)]
## This data contains 30 independent variables and the response is
## train_wdbc$diagnosis which is either B or M (M for malignant).
train_wdbc$diagnosis <- ifelse(train_wdbc$diagnosis == "M", "1", "0")
dim(train_wdbc)
```

```
## [1] 569  31
```

```
sum(is.na(train_wdbc))
```

```
## [1] 0
```

16. From the code above, which of the following is FALSE?

(A) There are 569 observations on 31 variables.

(B) There are 31 observations on 569 variables.

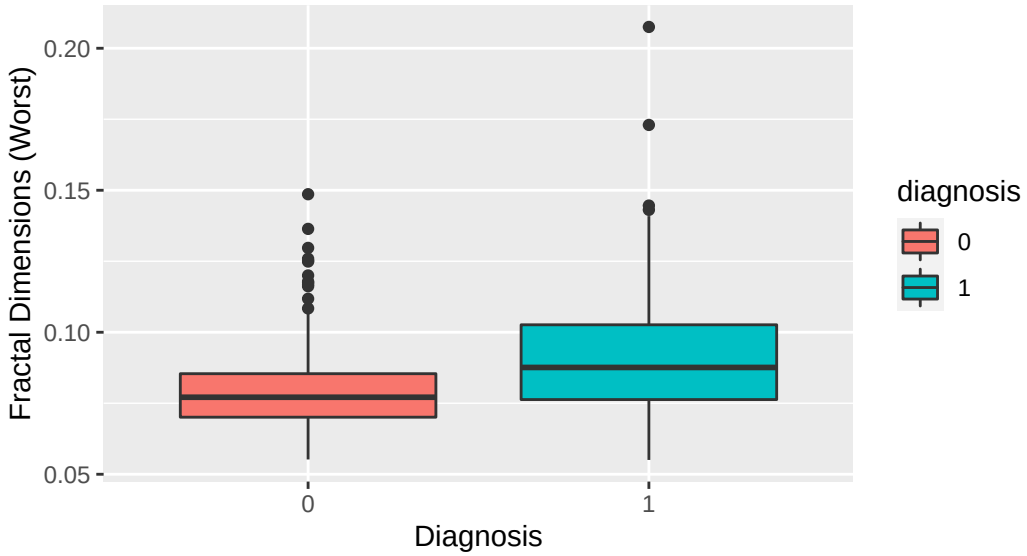
(C) The response in the data is categorical and `diagnosis=M` is labeled as a "0".

(D) There are no missing values in the data.

(A)

The following shows the boxplots of the `fractal_dimensions_worst` by `diagnosis`.

```
train_wdbc %>% ggplot(aes(x = diagnosis, y = fractal_dimension_worst, fill = diagnosis)) +
    geom_boxplot() + labs(x = "Diagnosis", y = "Fractal Dimensions (Worst)")
```



17. Based on the boxplot above, predicting malignant tumor if `fractal_dimension_worst` is larger than some threshold is a good rule (in comparison to predicting malignant if `fractal_dimension_worst` is smaller than some threshold).

(A) TRUE

(B) FALSE

(C) Not enough information to conclude.

(A)

```
# summary(train_wdbc)
X <- model.matrix(diagnosis ~ ., train_wdbc)[, -1]
Y <- train_wdbc$diagnosis
set.seed(10)
sample1 <- sample(569, replace = TRUE)
X1 <- X[sample1, ]
Y1 <- Y[sample1]
fit.cv1 <- cv.glmnet(X1, Y1, alpha = 1, family = "binomial", nfolds = 10, type.measure = "deviance")
sample2 <- sample(569, replace = TRUE)
X2 <- X[sample2, ]
Y2 <- Y[sample2]
fit.cv2 <- cv.glmnet(X2, Y2, alpha = 1, family = "binomial", nfolds = 10, type.measure = "deviance")
```

```
coef.1se1 <- coef(fit.cv1, s = 0.1)
coef.1se1 <- coef.1se1[which(coef.1se1 != 0), ]
coef.1se2 <- coef(fit.cv2, s = 0.1)
coef.1se2 <- coef.1se2[which(coef.1se2 != 0), ]
var.fit.cv1 <- names(coef.1se1)[-1]  # variables selected from sample 1
var.fit.cv1
```

```
## [1] "concave.points_mean"  "radius_worst"         "texture_worst"
## [4] "concave.points_worst"
```

```
var.fit.cv2 <- names(coef.1se2)[-1]  # variables selected from sample 2
var.fit.cv2
```

```
## [1] "radius_worst"         "concave.points_worst"
```

18. Which of the following is FALSE regarding the code above?

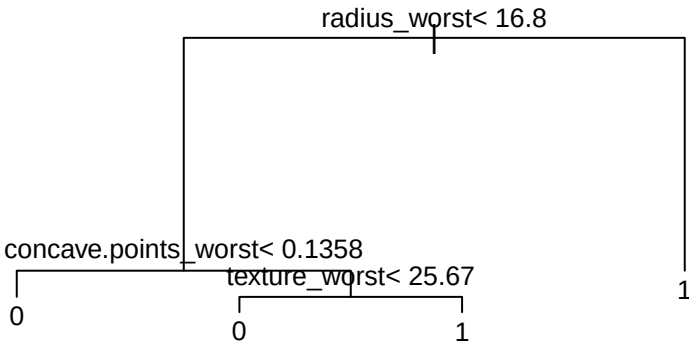
(A) The code provides a bootstrap replication of logistic lasso.

(B) The variable `concave.points_mean` will turn out insignificant when we refit `glm()`.

(C) With more bootstrap samples, the more a variable gets selected the more important it is (for prediction).

(C)

```
fit.single.rp <- rpart(diagnosis ~ ., train_wdbc, minsplit = 20, cp = 0.009)
plot(fit.single.rp, margin = 0.2)
text(fit.single.rp, pretty = TRUE, cex = 0.8)
```



19. Variables other than `radius_worst`, `concave.points_worst` and `texture_worst` are irrelevant for prediction based on `fit.single.rp`.

(A) TRUE

(B) FALSE

(C) Depends on what other variable values are.

(A)

19.2 Based on `fit.single.rp`, we would like to predict the diagnosis result for John who has the following readings. `radius_worst = 15`, `concave.points_worst = .14`, `texture_worst = 27` and `concave.points_mean = .05`. Which statement is correct?

(A) We can not predict John's diagnosis.

(B) The predicted label for John is 1

(C) The predicted label for John is 0

(A)

```
fit.glm <- glm(as.numeric(diagnosis) ~ radius_worst + concave.points_worst +
    texture_worst, train_wdbc, family = binomial(logit))
summary(fit.glm)
```

```
##
## Call:
## glm(formula = as.numeric(diagnosis) ~ radius_worst + concave.points_worst +
##     texture_worst, family = binomial(logit), data = train_wdbc)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.975  -0.067  -0.008   0.005   3.847
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -32.8621     4.3740   -7.51  5.8e-14 ***
## radius_worst           1.1436     0.1833    6.24  4.4e-10 ***
## concave.points_worst  51.3369     9.0742    5.66  1.5e-08 ***
## texture_worst          0.2782     0.0528    5.27  1.4e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 101.69  on 565  degrees of freedom
## AIC: 109.7
##
## Number of Fisher Scoring iterations: 9
```

20. The summary of `fit.glm` shows that the three variables chosen by decision tree are statistically significant irrespective of what other variables are included.

(A) TRUE

(B) FALSE

(B)

```
fit.rf <- randomForest(as.factor(diagnosis) ~ ., train_wdbc, mtry = 10, ntree = 100)
```

21. Which statement(s) is correct?

(A) fit.rf is obtained by taking average of 10 trees.

(B) fit.rf is obtained by exaggerating 100 bootstrap trees where each tree is a regular single tree with all 30 variables.

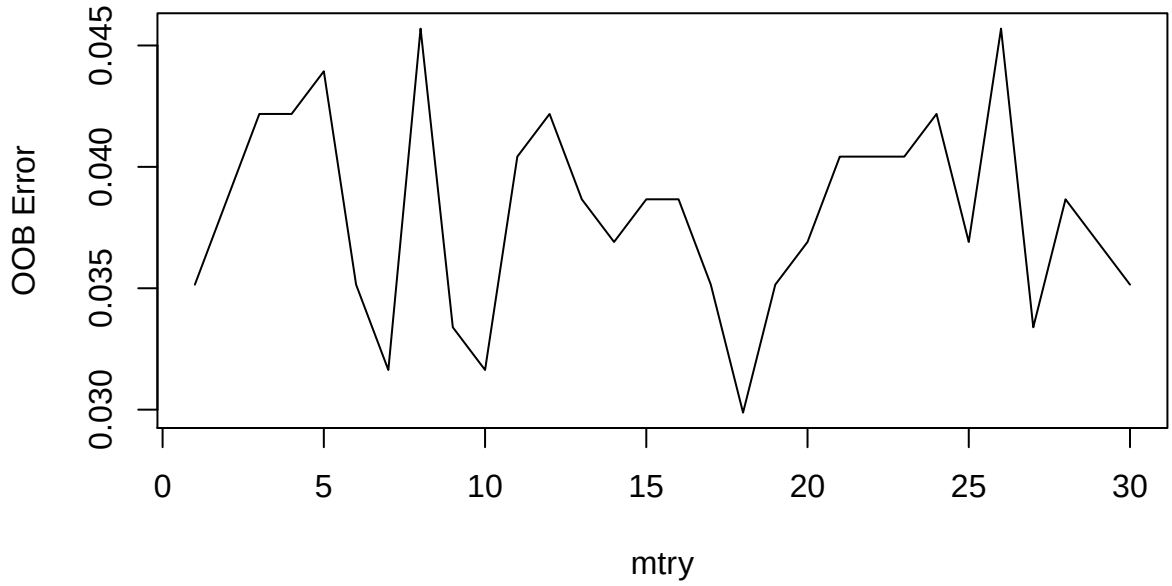
(C) fit.rf is obtained by exaggerating 100 bootstrap trees where each tree is a random tree where the node is chosen among randomly chosen 10 variables to be split.

(C).

Read the following chunk first.

```
set.seed(2)
rf.error.p <- 1:30 # set up a vector of length 30
for (p in 1:30) # repeat the following code inside { } 30 times
{
```

```
fit.rf <- randomForest(as.factor(diagnosis)~., train_wdbc, mtry=p, ntree=100)
rf.error.p[p] <- fit.rf$err.rate[100,1] # collecting oob mse based on 100 trees
}
plot(1:30, rf.error.p, xlab = "mtry", ylab = "OOB Error", type = 'l')
```




```
fit.rf.final <- randomForest(as.factor(diagnosis)~., train_wdbc, mtry=which.min(rf.error.p), ntree=100)
```

22. What does `mtry` in the above plot stand for?

(A) `mtry` is the number of variables on which a tree on bootstrapped sample is built.

(B) `mtry` is the number of variables used to each split in each bootstrapped tree.

(C) None of the above.

(B)

23. What is the best `mtry` based on the plot?

(A) 7

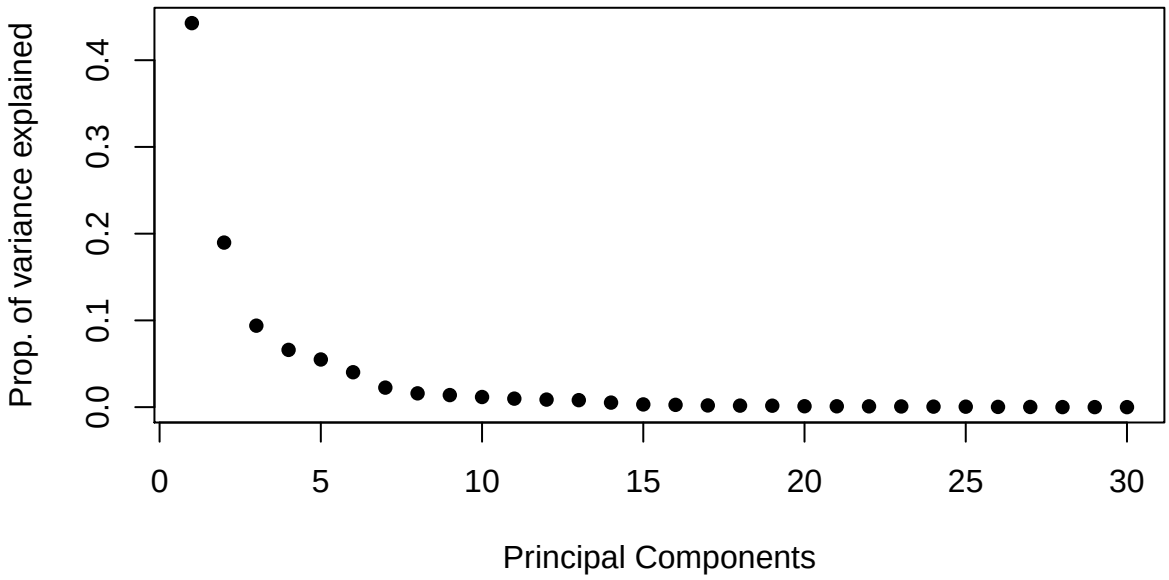
(B) 10

(C) 18

(D) 30

(C)

```
PC <- prcomp(as.matrix(train_wdbc[, -1]), scale = TRUE)
pve.4 <- (PC$sdev)^2/sum((PC$sdev)^2)
plot(pve.4, pch = 16, xlab = "Principal Components", ylab = "Prop. of variance explained")
```



24. Based on the above chunk and plot, which of the following is true?

(A) The first two principal components capture at least 60% of variance in the covariates of `train_wdbc`.

(B) The proportion of variances explained by PC1 and PC2 implies they would be statistically significant in a logistic fit.

(A).

```
## Recall X denotes the covariate matrix (before Q18.)
XPC <- X %*% PC$rotation
set.seed(1)
# XPC denotes covariates transformed in the direction of PCs.
fit.rf.PC1 <- randomForest(as.factor(train_wdbc$diagnosis) ~ XPC[, 1], mtry = 1,
    ntree = 100)
fit.rf.PC2 <- randomForest(as.factor(train_wdbc$diagnosis) ~ XPC[, 1] + XPC[,
    2], mtry = 1, ntree = 100)
fit.rf.PC3 <- randomForest(as.factor(train_wdbc$diagnosis) ~ XPC[, 1] + XPC[,
    2] + XPC[, 3], mtry = 1, ntree = 100)
fit.rf.PC4 <- randomForest(as.factor(train_wdbc$diagnosis) ~ XPC[, 1] + XPC[,
    2] + XPC[, 3] + XPC[, 4], mtry = 1, ntree = 100)
fit.rf.PC5 <- randomForest(as.factor(train_wdbc$diagnosis) ~ XPC[, 1] + XPC[,
    2] + XPC[, 3] + XPC[, 4] + XPC[, 5], mtry = 1, ntree = 100)
rf.error <- c(fit.rf.final$err.rate[100, 1], fit.rf.PC1$err.rate[100, 1], fit.rf.PC2$err.rate[100,
    1], fit.rf.PC3$err.rate[100, 1], fit.rf.PC4$err.rate[100, 1], fit.rf.PC5$err.rate[100,
    1])
names(rf.error) <- c("RF_Final", "RF_PC1", "RF_PC2", "RF_PC3", "RF_PC4", "RF_PC5")
rf.error
```

```
## RF_Final   RF_PC1   RF_PC2   RF_PC3   RF_PC4   RF_PC5
##   0.0351   0.1547   0.1213   0.1072   0.0808   0.0703
```

25. The above code performed random forest using the principal components instead of original variables. Which of the following is TRUE about `RF_PC5`?

(A) It only uses five of the original covariates.

(B) It depends on all the original covariates.

(C) It does not capture all the variance in the original covariates.

(D) It is guaranteed to be better than `RF_PC4` always.

BB) and (C)

26 Yelp data with Neural Network

The data structure is as follows: word frequencies of 1032 words for each review is extracted as the predictors, the response variable y is a binary of review rating, 1 indicates good and 0 bad.

The following R-chunk specifies a neural network to model P(y=1|a review) and P(y=0|a review)

```
model <- keras_model_sequential() %>% layer_dense(units = 32, activation = "relu",
    input_shape = c(1032)) %>% layer_dense(units = 16, activation = "relu") %>%
    layer_dense(units = 2, activation = "softmax")  # output
```

Which statement(s) are correct?

(A) The model is same as logistic regression model.

(B) The model is built on one layer with 48 neurons.

(C) There are two layers with 32 and 16 neurons in each layer respectively. Each neuron is created by taking the linear combination of all the neurons from the previous layer then apply relu function.

(C)