**Name: Indraja Nutalapati**

**Final Examination**

**PART I: MULTIPLE CHOICE** (10 points)

INSTRUCTIONS: Please write the letter of the single *best* answer to each question in the space provided. Each question is worth **2** points. This exam is meant to be challenging, so please check your answers carefully. This is an open-everything exam – the only resource you may not use is another person.

**1.1. A student builds a supervised classification model using a training data set. The model is able to classify 92% of the training data set items accurately. However, when the model is applied to a test data set, it predicts only 53% of the test data set items accurately. This model has high \_\_\_\_\_\_\_\_\_. Which choice should be used to fill in the blank?**

a) Bias

b) Statistical Significance

c) Interpretability

d) Variance

**Answer: d**

**1.2. A technical lead has noticed that several developers on her team are using different custom SQL statements to perform the same sales tax calculation. She would like to centralize the logic for the calculation so that it is used consistently in all parts of the application and can be edited in one single place when needed. Which of the following would be least helpful in centralizing the sales tax calculation logic?**

a) PL/SQL Function

b) PL/SQL Stored Procedure

c) PL/SQL Anonymous Block

d) View

**Answer: c**

**1.3. Multiple applications are updating a numeric column in a table. You would like to make sure that no single update statement will cause any of the values in the column to increase by more than 10%. You decide to write a trigger that enforces this. If an update statement would increase a value in the column by more than 10%, your trigger will ensure that it increases by exactly 10%. What kind of timing should your trigger have?**

a) Before Statement

b) Before Row

c) After Row

d) After Statement

**Answer: a**

**1.4. What is the relationship between precision and recall?**

a) Precision and recall typically move together – as one rises, so does the other.

b) Precision and recall are typically opposed – as one rises, the other falls.

c) Precision and recall do not influence one another.

d) Precision and recall are two names for the same concept.

**Answer: b**

**1.5. A more complex machine learning model may increase \_\_\_\_\_ at the expense of \_\_\_\_\_.**

a) backpropagation / interations

b) misclassification rate / p-values

c) predictive accuracy / intelligibility

d) bias / gini impurity

**Answer: a**

**PART II: CODE AND EXPLANATION** (90 points)

INSTRUCTIONS: The following questions on machine learning are somewhat open-ended. There isn’t necessarily a single correct answer for each question. Instead, a variety of answers could be acceptable, as long as you demonstrate command of the relevant concepts and techniques. You will be graded both on the quality of your code with respect to the problem at hand and your explanations (when requested) of why you took the approach you did.

I hope I don’t need to reiterate this, but given that the questions are so open form, it will probably be \*\*very obvious\*\* to me if you collaborate on the answers. Please work independently. However, you may use any other resources (notes, texts, Internet) you choose.

**\*\*\*You must use a different kind of machine learning model for each one of these problems. Which one you use for a given problem is up to you. You just can’t use any given model type (e..g, decision tree, random forest, boosted trees, logistic regression, naïve Bayes, neural networks, SVMs) for more than one of these questions.\*\*\***

**2.1 Predicting Alcohol Consumption** (30 points)

You will find a data file in Blackboard alongside this exam. It is called “student\_alcohol.csv” and has the following structure:

# Attributes for student-alcohol.csv data set – data from hundreds of Portuguese students  
1 school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)   
2 sex - student's sex (binary: 'F' - female or 'M' - male)   
3 age - student's age (numeric: from 15 to 22)   
4 address - student's home address type (binary: 'U' - urban or 'R' - rural)   
5 famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)   
6 Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)   
7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education

or 4 - higher education)   
8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or

4 - higher education)   
9 Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other')   
10 Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other')   
11 reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')   
12 guardian - student's guardian (nominal: 'mother', 'father' or 'other')   
13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)   
14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)   
15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)   
16 schoolsup - extra educational support (binary: yes or no)   
17 famsup - family educational support (binary: yes or no)   
18 paid - extra paid classes within the course subject  
19 activities - extra-curricular activities (binary: yes or no)   
20 nursery - attended nursery school (binary: yes or no)   
21 higher - wants to take higher education (binary: yes or no)   
22 internet - Internet access at home (binary: yes or no)   
23 romantic - with a romantic relationship (binary: yes or no)   
24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)   
25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)   
26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)   
27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)   
28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)   
29 health - current health status (numeric: from 1 - very bad to 5 - very good)   
30 absences - number of school absences (numeric: from 0 to 93)

31 studentid - unique identification number for each student

The file was adapted (edited by me) from a publicly available data source at: <http://archive.ics.uci.edu/ml/datasets/STUDENT+ALCOHOL+CONSUMPTION>

Your mission is to build a model that predicts whether students will drink heavily on weekends. For our purposes, “drink heavily” means a Walc value of 4 or 5. You will use R to build your model.

Set a seed using your Villanova id, and then break your data into training and test sets using random sampling. Use 80% for your training set and the remaining 20% for your test set. Build a model that allows you to predict heavy weekend drinking, and then determine the percentage that you were able to classify correctly.

You will submit, **by inserting into this document:**

* A table showing how your predictions compared to the actual values in the test set
* Your calculation of the % of students in the test set you classified correctly
* All of the R code you used to arrive at your predictions and outcomes (no screen shots – I want code that I can copy and run on my own – I should be able to run the code you include and arrive at the same outcome you did, after adjusting for filesystem paths)
* Any supporting output from R commands – such as summary() or inspect() – that helped you to make decisions.
* An assessment of whether your model gave more accurate results than always predicting the most commonly occurring class in the data.
* A written description of why you took the approach you took (between a few paragraphs and one page)

When I am grading, I will look for evidence of:

* Code that is appropriate to the problem
* Output from the learning method(s) interpreted correctly
* Reasonable, defensible choices in your approach

Note that this isn’t a simple problem with an obvious solution. Your results are unlikely to be perfectly convincing. Just build the best model you can, describe your results, and defend how you arrived at these. The goal here is to show that you can independently apply the concepts we have been practicing.

**Answer:**

**#install.packages("tree")**

**library(tree)**

**student <- read.csv(file = 'D:/Data Mining & Data Base Systems/R/student\_alcohol.csv')**

**#install.packages("dplyr")**

**library(dplyr)**

**student = mutate(student, Walc = ifelse(Walc %in% 4:5, "heavily", "not heavy"))**

**sample\_size = .8**

**set.seed(02137024)**

**train = sample(1:nrow(student), sample\_size \* nrow(student))**

**student\_train = student[train,]**

**student\_test = student[-train,]**

**prop.table(table(student\_test$Walc))**

**student\_train\_tree = tree(as.factor(Walc) ~ ., student\_train)**

**summary(student\_train\_tree)**

**plot(student\_train\_tree)**

**text(student\_train\_tree, pretty = 0)**

**student\_pred = predict(student\_train\_tree, student\_test, type = "class")**

**library(caret)**

**confusionMatrix(as.factor(student\_pred), as.factor(student\_test$Walc), mode = "prec\_recall" )**

Confusion Matrix and Statistics

Table showing predictions compared to the actual values in the test set.

**Reference**

**Prediction heavily not heavy --------🡪**

**heavily 15 7**

**not heavy 11 42**

**Accuracy : 0.76**

95% CI : (0.6475, 0.8511)

Accuracy by predicting the most commonly occurring class in the data i.e 65.33%

No Information Rate : 0.6533 -------🡪

P-Value [Acc > NIR] : 0.03172

Kappa : 0.4503

Mcnemar's Test P-Value : 0.47950

Precision : 0.6818

Recall : 0.5769

F1 : 0.6250

Prevalence : 0.3467

Detection Rate : 0.2000

Detection Prevalence : 0.2933

Balanced Accuracy : 0.7170

'Positive' Class : heavily

prop.table(table(student\_test$Walc))

heavily not heavy

0.3466667 0.6533333

**Percentage of students in the test set, the model Classified correctly is 76%**

**Yes, the model gave more accurate results than always predicting the most commonly occurring class in the data. The accuracy of the model is 76% which is greater than the accuracy by predicting the most commonly occurring class in the data (65.33%).**

**The given problem asks to build a model that predicts whether students will drink heavily on weekends. For this we need to classify data based on the variable Walc(if Walc = 4 or Walc = 5, students drink heavily on weekends). For such data, predictions are good if we use decision trees. Hence, I used a classification model, decision tree to predict the values.**

**2.2 Determining the Author** (30 points)

You will find a data file called “tweets.csv” in Blackboard alongside this exam. This file contains tweets from three different senders. There are two columns – the sender of the tweet, and the text of the tweet. Your job is to build a model that will allow you to predict who sent a tweet based on the text it contains.

Set a seed using your Villanova id, and then break your data into training and test sets using random sampling. Use 80% for your training set and the remaining 20% for your test set. Build a model that allows you to predict the authors of the tweets in your test set, and then determine the percentage that you were able to classify correctly.

You will submit, **by inserting into this document:**

* A table showing how your predictions compared to the actual values in the test set
* Your calculation of the % of tweets in the test set you classified correctly
* All of the R code you used to arrive at your predictions and outcomes (no screen shots – I want code that I can copy and run on my own – I should be able to run the code you include and arrive at the same outcome you did, after adjusting for filesystem paths)
* Any supporting output from R commands – such as summary() or inspect() – that helped you to make decisions.
* A written description of why you took the approach you took (between a few paragraphs and one page)

When I am grading, I will look for evidence of:

* Code that is appropriate to the problem
* Output from the learning method(s) interpreted correctly
* Reasonable, defensible choices in your approach

**Answer:**

**tweets <- read.csv("D:/Data Mining & Data Base Systems/R/tweets.csv")**

**library(dplyr)**

**tweets$name <- factor(tweets$name)**

**str(tweets$name)**

**#install.packages("tm")**

**library(tm)**

**tweets\_corpus <- VCorpus(VectorSource(tweets$text))**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus, content\_transformer(tolower))**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus\_clean, removeNumbers)**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus\_clean, removeWords, stopwords())**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus\_clean, removePunctuation)**

**#install.packages("SnowballC")**

**library(SnowballC)**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus\_clean, stemDocument)**

**tweets\_corpus\_clean <- tm\_map(tweets\_corpus\_clean, stripWhitespace)**

**tweets\_dtm <- DocumentTermMatrix(tweets\_corpus\_clean)**

**train\_pct <- .8**

**set.seed(100)**

**train = sample(1:nrow(tweets),train\_pct \* nrow(tweets))**

**tweets\_dtm\_train <- tweets\_dtm[train, ]**

**tweets\_dtm\_test <- tweets\_dtm[-train, ]**

**tweets\_train\_labels <- tweets[train, ]$name**

**tweets\_test\_labels <- tweets[-train, ]$name**

**barry <- subset(tweets, name == "Barry")**

**don <- subset(tweets, name == "Don")**

**hillary <- subset(tweets, name == "Hillary")**

**tweets\_freq\_words <- findFreqTerms(tweets\_dtm\_train, 5)**

**tweets\_dtm\_freq\_train <- tweets\_dtm\_train[ , tweets\_freq\_words]**

**tweets\_dtm\_freq\_test <- tweets\_dtm\_test[ , tweets\_freq\_words]**

**convert\_counts <- function(x) {**

**return(ifelse(x > 0, "Yes", "No"))**

**}**

**tweets\_train <- apply(tweets\_dtm\_freq\_train, MARGIN = 2, convert\_counts)**

**tweets\_test <- apply(tweets\_dtm\_freq\_test, MARGIN = 2, convert\_counts)**

**#install.packages("e1071")**

**library(e1071)**

**tweets\_classifier <- naiveBayes(tweets\_train, tweets\_train\_labels)**

**tweets\_test\_pred <- predict(tweets\_classifier, tweets\_test)**

**summary(tweets\_test\_pred)**

**#install.packages("caret")**

**library(caret)**

**confusionMatrix(as.factor(tweets\_test\_pred), tweets\_test\_labels,**

**mode = "prec\_recall")**

Confusion Matrix and Statistics

**Reference**

**Prediction Barry Don Hillary**

**Barry 549 91 69**

**Don 9 444 32**

**Hillary 44 52 510**

Overall Statistics

**Accuracy : 0.835**

95% CI : (0.817, 0.8519)

No Information Rate : 0.3394

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7523

Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

Class: Barry Class: Don Class: Hillary

Precision 0.7743 0.9155 0.8416

Recall 0.9120 0.7564 0.8347

F1 0.8375 0.8284 0.8381

Prevalence 0.3344 0.3261 0.3394

Detection Rate 0.3050 0.2467 0.2833

Detection Prevalence 0.3939 0.2694 0.3367

Balanced Accuracy 0.8892 0.8613 0.8770

**The percentage of tweets in the test set, the model Classified correctly is 83.5%**

**The problem asks to build a model that will allow you to predict who sent a tweet based on the text it contains. Hence, I have used a text classification model with naïve bayes function to build the model by cleaning all the unnecessary that is not needed in the prediction.**

**2.3 Predicting Wine Quality** (30 points)

You will find one last data file in Blackboard alongside this exam. The file is called “wine-quality.csv” and has the following structure:

1 - fixed acidity

2 - volatile acidity

3 - citric acid

4 - residual sugar

5 - chlorides

6 - free sulfur dioxide

7 - total sulfur dioxide

8 - density

9 - pH

10 - sulphates

11 - alcohol

12 - quality (score between 0 and 10)

All of these values are numeric. For our purposes, you don’t really need to understand what they mean, and so don’t worry if you don’t know how these things relate to the quality of wine. (I don’t.)

These data files were adapted (edited by me) from a publicly available data source at: <http://archive.ics.uci.edu/ml/datasets/Wine+Quality>

Your mission is to randomly divide the data set into training and test data sets. Use 75% of the data for your training set and the remaining 25% for your test set. **Set the seed 54321** before you divide the data so that I can try to reproduce your results. Use any machine learning techniques that you like to build a model that predicts quality (the last variable in the training data set). Then, apply your model to the test data set. Your goal is to get the highest possible correlation between the values predicted for the test data set and the actual values for the test data set. So, for whatever machine learning method you ultimately use, you need to determine and report this correlation between the predicted values and the actual values.

You will submit, **by inserting into this document:**

* All R code you used to create your training/test data sets, build your models, and test your models (no screen shots – I want code that I can copy and run on my own).
* Any supporting output from R commands – such as summary() or inspect() – that helped you to make decisions.
* Evidence of the correlations you obtained, such as the output from the cor() function.
* A clear statement of which model and results you are putting forth as your best effort to solve this problem.
* A written description of why you took the approach you took (between a few paragraphs and one page).

When I am grading, I will look for evidence of:

* Code that is appropriate to the problem
* Output that is interpreted correctly
* Reasonable, defensible choices in your approach

I will assign **extra credit points** for this problem. You will get an **extra half point on your exam grade** for every person who gets lower predictive success on the test data set than you do, up to a **maximum of five points**. You will not get extra credit for people who tie with you or people score higher (build a stronger predictive model) than you do. I will round all correlations to the second decimal place before making this determination. You will not lose any earned points through this comparison – you can only gain points by building a stronger predictive model than your classmates do.

**Answer:**

**wine <- read.csv("D:/Data Mining & Data Base Systems/R/wine-quality.csv")**

**str(wine)**

**normalize <- function(x) {**

**return((x - min(x)) / (max(x) - min(x)))**

**}**

**wine\_norm <- as.data.frame(lapply(wine, normalize))**

**summary(wine$quality)**

**summary(wine\_norm$quality)**

**train\_pct <- 0.75**

**set.seed(54321)**

**train <- sample(1:nrow(wine\_norm),train\_pct \* nrow(wine\_norm))**

**wine\_train <- wine\_norm[train, ]**

**wine\_test <- wine\_norm[-train, ]**

**#install.packages("neuralnet")**

**library(neuralnet)**

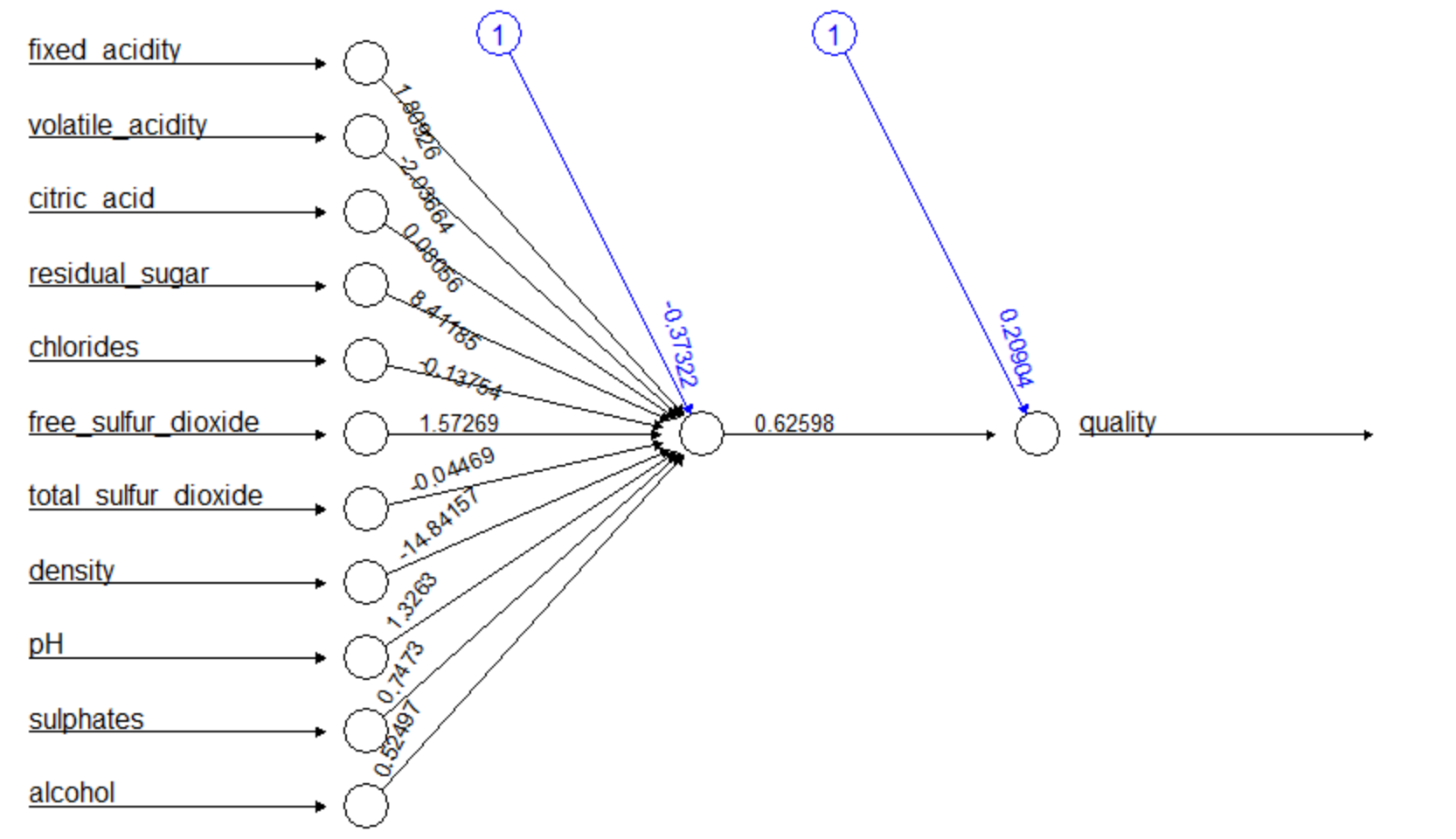
**net <- neuralnet(quality ~ ., wine\_train, hidden = 1)**

**plot(net)**

**net\_results <- compute(net, wine\_test[,1:11])**

**predicted\_quality <- net\_results$net.result**

**cor(predicted\_quality, wine\_test$quality)**



**[,1]**

**[1,] 0.5006418 (correlation value with 1 hidden neuron)**

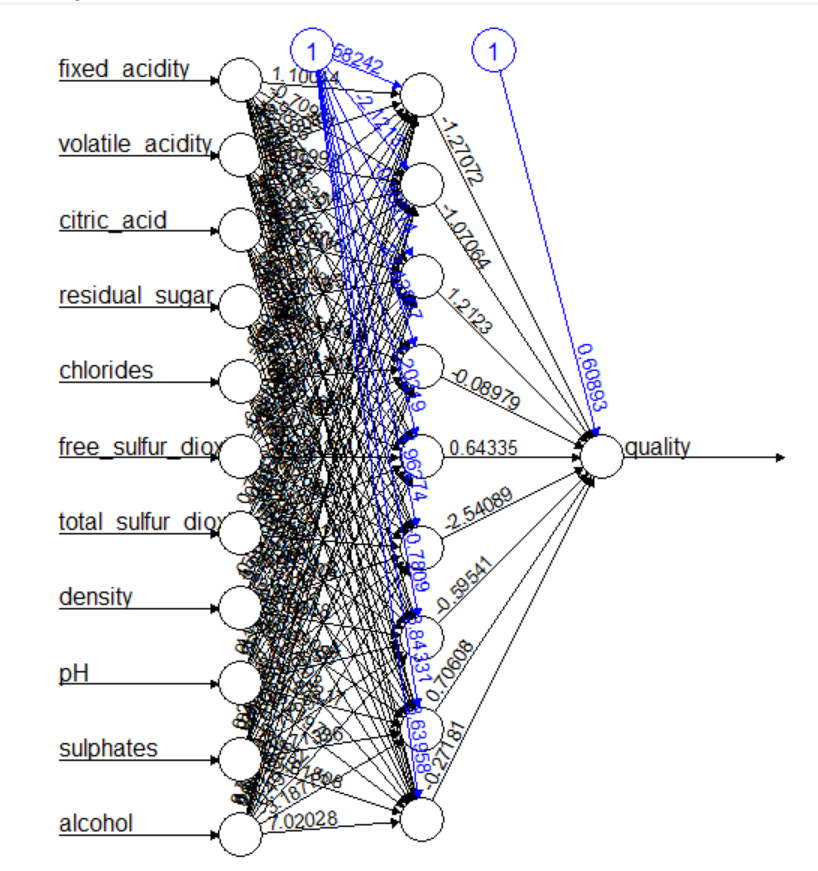
**net2 <- neuralnet(quality ~ .,wine\_train,hidden = 9)**

**plot(net2)**

**net\_results2 <- compute(net2, wine\_test[,1:11])**

**predicted\_strength2 <- net\_results2$net.result**

**cor(predicted\_strength2, wine\_test$quality)**



**[,1]**

**[1,] 0.5995015(correlation value with a model of 9 hidden neurons)**

**The given data contains all numeric values. Neural networks model works well with numeric data when compared to other models. A neural network model is stronger and depends on how we learn the neurons. The model is stronger by tarining the neurons with hidden = 1 in the first step and hidden = 9 in the second step as shown above. The model gave the highest correlation value 0f 0.5995015 with 9 hidden neurons. Hence, I have considered to take the neural network model with 9 hidden neurons.**

**So, net2 <- neuralnet(quality ~ .,wine\_train,hidden = 9) is the strongest model with 9 hidden neurons.**

**END OF EXAM**