Cancer Level Prediction

emw232

12/16/2020

STSCI 4740 Fall 2020 Final Project

Emily Weed

Section I: Introduction

The goal of this project is to predict the cancer level using the features provided to us in the data set. The cancer level is encoded as ordinal data taking the values 'Low', 'Medium', and 'High'. There are 24 features in the data set such as 'Age', 'Alcohol Use', 'Obesity' and 1000 rows. This is a classification problem because the target variable, cancer level, is categorical. I will start by exploring and cleaning (if needed) the data then segway into feature selection and begin to explore some models.

Section II: Data Exploration

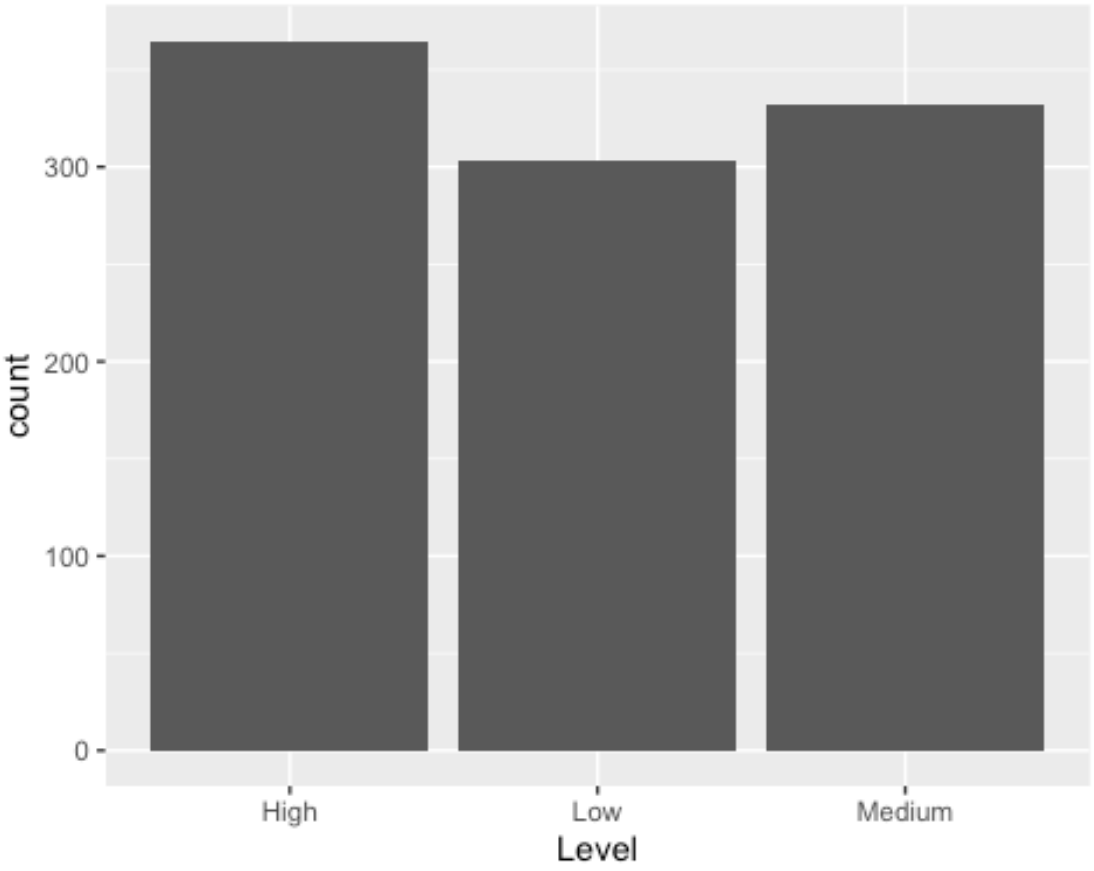
I will start by constructing some graphs, visualizations and summary statistics for the columns to get a better idea of the structure of the data set. In code Excerpt 1 in the appendix, I read in the data set and take a look at the first couple rows. In code Excerpt 2 I call summary on the data set. This gives me summary statistics for each column and tells me how many entries I have (1000 rows). This is a good first step in getting acquainted with my data.

From the summary of the data set we can see there are 24 features. Gender is categorical with gender 1 and gender 2 and all the rest are ints. We also have a Patient.Id variable which is just an unique identifier for each patient thus I suspect it will not be entirely useful.

Let's see if there are any missing values in the data set. In code Excerpt 3 I calculate the sum of NA values in each of the columns. As we can see, there are none so I will not have to deal with any imputation of missing values.

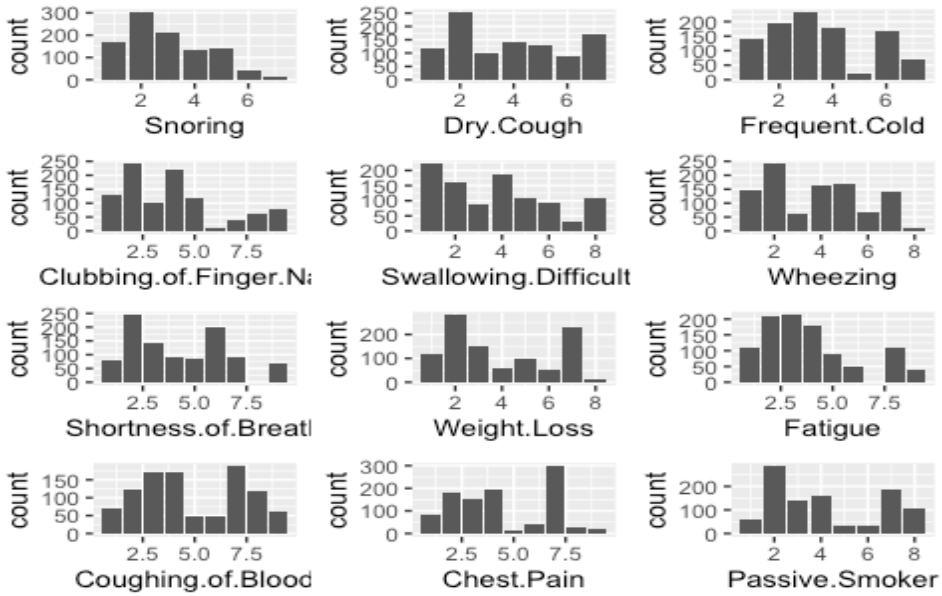
First let's look at the target variable to see if the proportion of each level is roughly equal.

```
library(ggplot2)
ggplot(df,aes(Level)) + geom_bar()
```

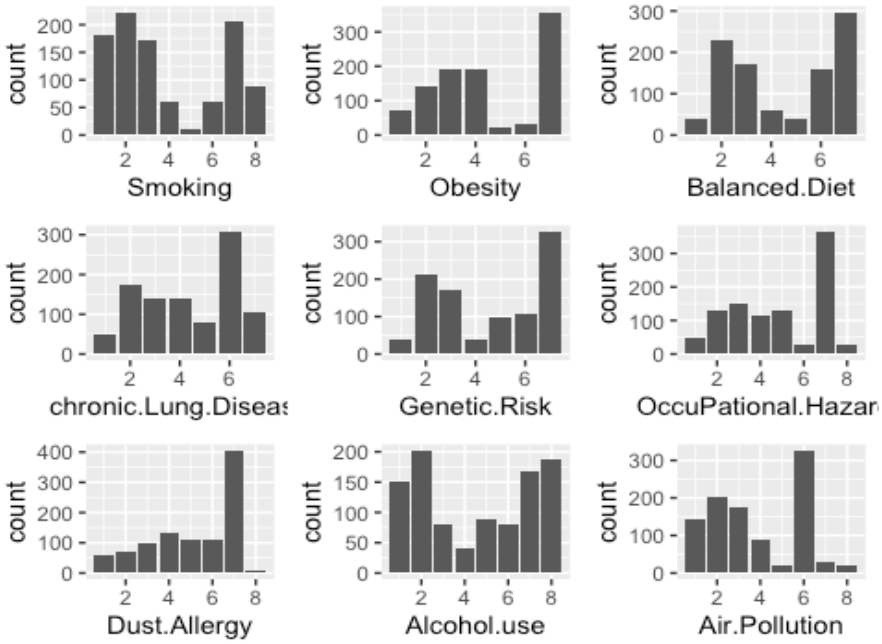


The bars are of roughly equal height so we will not have to deal with accounting for unproportional categories in our data set.

Let's make some visualizations to get a sense for the distribution of each variable. The code to create this visual is in code Excerpt 4 below.

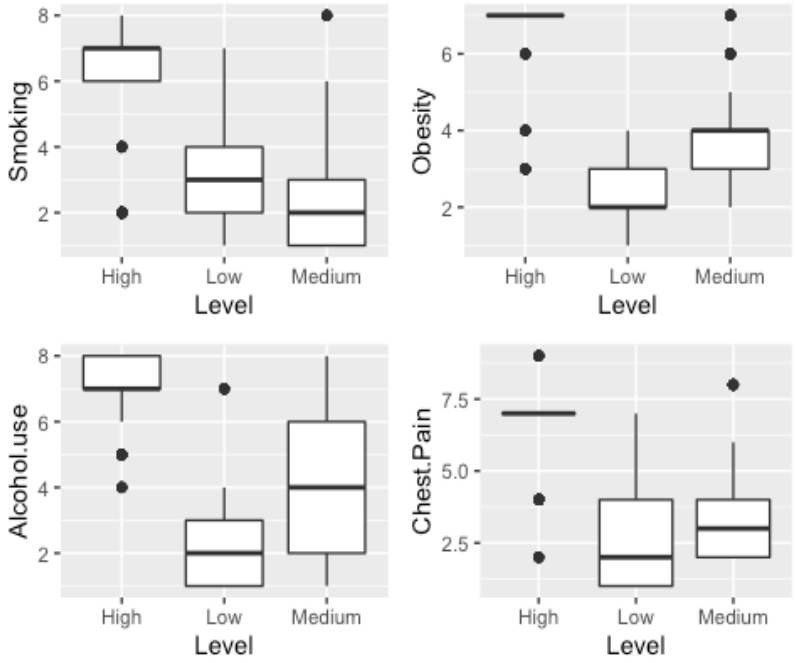


The code to create this visual is in code Excerpt 5 below.



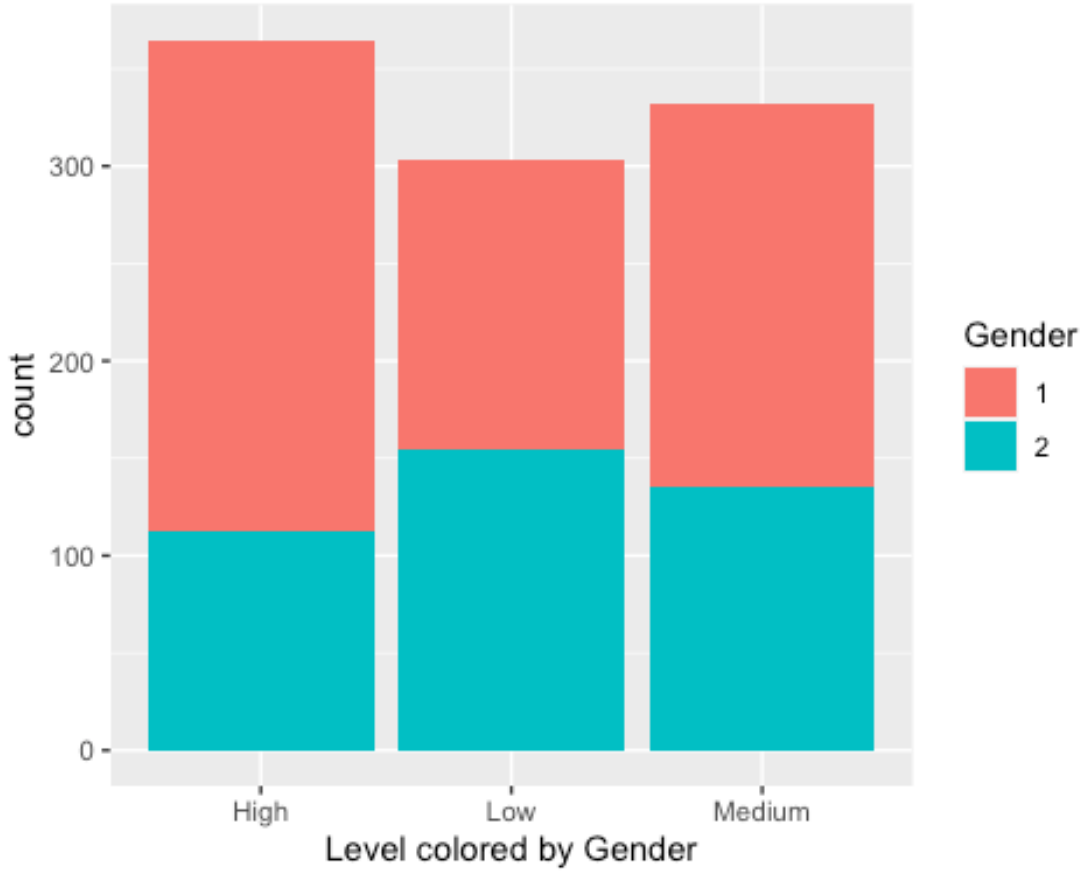
All the plots look relatively reasonable. Some are skewed left or right but none cause major concern for the reliability of the data. Smoking appears to be bimodal, peaking at both 2 and 7 which is interesting to note. Dust.Allergy, Occupational.Hazards, and chronic.Lung.Disease all have a large number of values at the higher levels (6/7) - skewed left.

Now let's visualize some of the variables' relationships with Level, our target variable. The code to make these boxplots is in code Excerpt 6 below.



We can see from these box plots that there are some outlier values within each Level for each variable. For example in Chest.Pain for Level "High" the majority of the patients have Chest.Pain value 7 while there are 3 outliers with values around 8,4,and 2. In Alcohol.Use, for Level "Medium" the patients seems have very normally distributed values of Alcohol.Use. The box plots in general have small interquartile ranges. Overall, these box plots seemed to be pretty condensed, with each level mostly occupying a small, certain range of values in each variable. I suspect I would see similar results if I plotted all of the variables against Level individually.

Now let's look at the gendered breakdown within levels. The code to make this visual is in code Excerpt 7 below. Looking at this plot we can see that more than 50% of the people with level "High" are gender 1, similarly for level "Medium". For level "Low" it looks to be about even.



Section III: Data Cleaning/Manipulation

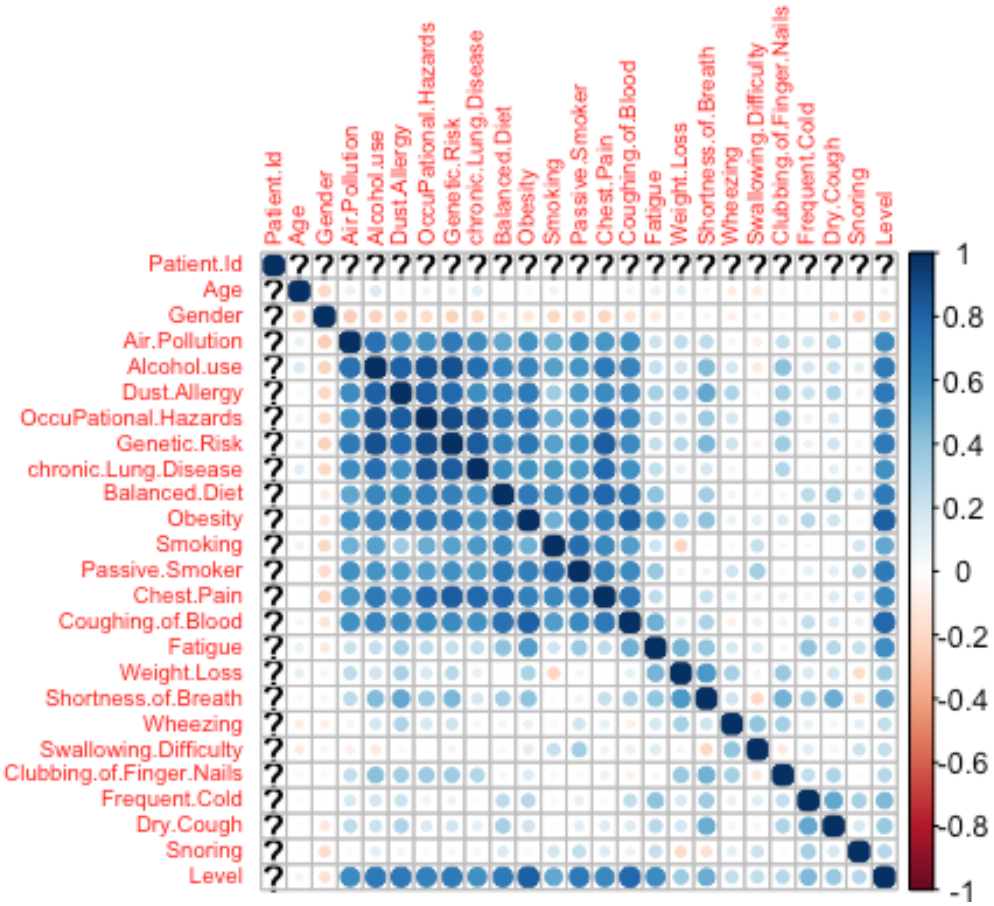
I will start by converting the Level to an ordinal variable with levels "Low"<"Medium"<"High" (ordered factor variable). The code for this is in code Excerpt 8 below.

Looking at the other variables, they seem to also be ordinal variables with the exception of Age and Gender. We have not really gone over in class how to deal with a scenario with all ordinal factor predictor variables so I will keep them all as continuous int variables (as they are right now). This was also confirmed as a possible approach on Piazza by Professor Ning. I will convert Gender to a categorical variable. Age is already correctly encoded an int (this is also included in code Excerpt 8 below).

Also included in code Excerpt 8 below, I look at the data types of the whole data set to confirm everything is as it should be.

Section IV: Feature Selection

I will start by looking at the correlation matrix of the data set. I want to select features that have a high correlation with Level but do not want to select features that are highly correlated with eachother as they will be redundant. The code to create this plot is in code Excerpt 9 below.



Based on this plot we can see that some variables have very little if any correlation with Level. For example Age, Gender, Wheezing, Swallowing.Difficulty, Clubbing.Of.Finger.Nails, Dry.cough, and Snoring have a very small correlation coefficient with Level so most likely they will not be very helpful in predicting Level.

I am going to drop these variables as well as PatientID (since it is just a unique identifier for each patient and thus is different for every row). I will then assess the importance of my remaining variables using Random Forest to select a smaller, more helpful set of features. The code dropping these variables is in code Excerpt 10 below.

Using variable importance through Random Forest is in code Excerpt 11 below.

We want to select variables for which the MeanDecreaseGini is highest. Looking at these values (from the table in code excerpt 11 below) we can see that these features have high importance: Coughing.of.Blood (`108.49103`), Obesity (`81.39806`), Passive.Smoker (`75.63726`), and Fatigue (`65.29201`). After this the importance value drops off into the 40s and below, thus I will now proceed with these features to use for model construction. In code Excerpt 12 below I drop the other features, to end with just this subset.

Section V: Model Construction

First I will split my data set into train and test sets. I will use an 80/20 split for the train test split (code Excerpt 13 below). In this section I will train the three models I plan to investigate. I will use 5 fold cross validation on my training set. This will allow me to assess which model will be the most likely to preform the best on the test set in my next section, Model Selection and Evaluation.

I will use a couple different models to attempt to best predict Level.

Section V.I: Linear Discrimminant Analysis

First I will use LDA with 5-fold cross validation. In LDA, it models the distribution of each of the parameters in each of the classes separately. Using Bayes' Theorem, it flips it around and obtains a probability that a certain example is in a class given the feature values, assigning whichever class has the highest probability. It is usually assumed that the distribution of the features in each of the classes follows a multivariate normal distribution with a convariance matrix in common across the classes. This assumption may be a little too strong for our data set here but I will construct this model acknowledging this. Code Excerpt 14 contains my code for this model.

The average misclassification rate for LDA is `0.14` which is not too bad. Let's see if our other models can preform better

Section V.II: Quadratic Discrimminant Analysis

I will next do QDA, with 5-fold cross validation again. QDA is similar to LDA in that they both assume the predictors for each class are drawn from multivariate normal distributions, and use Bayes' Theorem to make predictions. Where they differ is with the covariance matrix. QDAs assume different covariance matrices in each class, giving the model more flexibility. The QDA decision boundary does not have to be linear. My code running this model is in code Excerpt 15 below.

The average misclassification rate for QDA is `0.12125`. This is lower than the one for LDA indicating that the flexibility of QDA served to improve our preformance here

Section V.III: K Nearest Neighbours

Now I will do KNN with 5-fold cross validation. The KNN classifier uses the k closest points to a test point and determines the conditional probability of a certain class, x, as the proportion of those points k points whose class label is k. Using Bayes' rule, it classifies the test point to the class with the highest probability. This is a very different approach than LDA and QDA. There are no assumptions being made. With this model we do, however, have to define the value of k we want to use. To attack choosing this extra hyperparamter, I will run 4 different KNN models within each fold, each with a different number of neighbours. This will help me determine the best value for the number of neighbours to use for this method. My code running this model is in code Excerpt 16 below.

The average misclassification rate was lowest for k =5 here at `0.0075. This` misclassification rate was lower than LDA's and QDA's

Section VI: Model Selection and Evaluation

In order to decide which model preformed the best on my data, I can compared the average misclassification rates I obtained in the previous part

LDA: `0.14`

QDA: `0.12125`

KNN: `0.0075` (for k = 5)

We can see that they are very similar for LDA and QDA however it improved for KNN. I will use KNN with k = 5 as my final model. I will train it with the whole training set (no cross-validation) and then run it on the test set and assess how well it preforms on the test set, which I have not touched at all yet. I set this test set aside for this purpose; to be able to accurately assess the preformance of my chosen model on a set of data the model has never seen before. The code running this is in code Excerpt 17 below.

Looking at the misclassification rate and the confusion matrix it seems that this model has performed very well on the test set. The misclassification rate is `0.035` and there are only 7 points incorrectly represented in the confusion matrix.

Section VII: Conclusion

In this project I aimed to predict cancer Level using the features Coughing.of.Blood, Obesity, Passive.Smoker, and Fatigue. These were a subset that I chose from the original 24 features that were given to us. This data was already very clean when provided which facilitated the process of feature selection. I chose this subset by looking at a correlation plot of the

features with my target, Level. I was able remove some features that were not correlated at all or only very slightly with Level. I then used RandomForest and analyzed the importance value of each feature in this method. This gave me my final subset of features. I then split the data into my training and testing subsets to transistion into model construction and selection. The three models I chose to investigate were LDA, QDA, and KNN. These were the three classification models we spent the most time discussing in class. To choose between these three models I used 5-fold cross validation on the training set to get a better idea of which model would preform the best on my set aside test set. After running and assessing each model, KNN with k = 5 preformed the best, with the lowest average misclassification rate of `0.0075`. I then proceeded to run this model on my test data and got a misclassification rate of `0.035`. I saw no evidence that this model overfit the data since these preformance measures are reported for a test set that was held out and it was fairly similar to our training misclassification rate. Additionally our data set here was rather small, with only 1000 exmaples. To correctly assess the reliability of a model for this objective there would need to be a lot more data. Overall, this project shows that in order to construct a model with very good preformance, a lot of preprocessing and model selection needs to be done prior to using one's test set.

Appendix

Excerpt 1:

```
df = read.csv('cancer_data.csv')
head(df)
```

```
##   Patient.Id Age Gender Air.Pollution Alcohol.use Dust.Allergy
## 1         P1  33      1             2           4            5
## 2        P10  17      1             3           1            5
## 3       P100  35      1             4           5            6
## 4      P1000  37      1             7           7            7
## 5       P101  46      1             6           8            7
## 6       P102  35      1             4           5            6
##   OccuPational.Hazards Genetic.Risk chronic.Lung.Disease Balanced.Diet Obe
sity
## 1                    4            3                    2             2
4
## 2                    3            4                    2             2
2
## 3                    5            5                    4             6
7
## 4                    7            6                    7             7
7
## 5                    7            7                    6             7
7
## 6                    5            5                    4             6
7
##   Smoking Passive.Smoker Chest.Pain Coughing.of.Blood Fatigue Weight.Loss
## 1       3              2          2                 4       3           4
## 2       2              4          2                 3       1           3
## 3       2              3          4                 8       8           7
## 4       7              7          7                 8       4           2
## 5       8              7          7                 9       3           2
## 6       2              3          4                 8       8           7
##   Shortness.of.Breath Wheezing Swallowing.Difficulty Clubbing.of.Finger.Na
ils
## 1                   2        2                     3
1
## 2                   7        8                     6
2
## 3                   9        2                     1
4
## 4                   3        1                     4
5
## 5                   4        1                     4
2
## 6                   9        2                     1
```

```
4
##   Frequent.Cold Dry.Cough Snoring  Level
## 1             2         3       4    Low
## 2             1         7       2 Medium
## 3             6         7       2   High
## 4             6         7       5   High
## 5             4         2       3   High
## 6             6         7       2   High
```

Excerpt 2:

```
summary(df)
```

```
##   Patient.Id             Age            Gender      Air.Pollution
##  Length:1000        Min.   :14.00   Min.   :1.000   Min.   :1.00
##  Class :character   1st Qu.:27.75   1st Qu.:1.000   1st Qu.:2.00
##  Mode  :character   Median :36.00   Median :1.000   Median :3.00
##                     Mean   :37.17   Mean   :1.402   Mean   :3.84
##                     3rd Qu.:45.00   3rd Qu.:2.000   3rd Qu.:6.00
##                     Max.   :73.00   Max.   :2.000   Max.   :8.00
##   Alcohol.use     Dust.Allergy   OccuPational.Hazards  Genetic.Risk
##  Min.   :1.000   Min.   :1.000   Min.   :1.00         Min.   :1.00
##  1st Qu.:2.000   1st Qu.:4.000   1st Qu.:3.00         1st Qu.:2.00
##  Median :5.000   Median :6.000   Median :5.00         Median :5.00
##  Mean   :4.563   Mean   :5.165   Mean   :4.84         Mean   :4.58
##  3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.00         3rd Qu.:7.00
##  Max.   :8.000   Max.   :8.000   Max.   :8.00         Max.   :7.00
##  chronic.Lung.Disease Balanced.Diet      Obesity         Smoking
##  Min.   :1.00         Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.00         1st Qu.:2.000   1st Qu.:3.000   1st Qu.:2.000
##  Median :4.00         Median :4.000   Median :4.000   Median :3.000
##  Mean   :4.38         Mean   :4.491   Mean   :4.465   Mean   :3.948
##  3rd Qu.:6.00         3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000
##  Max.   :7.00         Max.   :7.000   Max.   :7.000   Max.   :8.000
##  Passive.Smoker    Chest.Pain    Coughing.of.Blood    Fatigue
##  Min.   :1.000   Min.   :1.000   Min.   :1.000     Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:3.000     1st Qu.:2.000
##  Median :4.000   Median :4.000   Median :4.000     Median :3.000
##  Mean   :4.195   Mean   :4.438   Mean   :4.859     Mean   :3.856
##  3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000     3rd Qu.:5.000
##  Max.   :8.000   Max.   :9.000   Max.   :9.000     Max.   :9.000
##   Weight.Loss    Shortness.of.Breath    Wheezing     Swallowing.Difficulty
##  Min.   :1.000   Min.   :1.00        Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.00        1st Qu.:2.000   1st Qu.:2.000
##  Median :3.000   Median :4.00        Median :4.000   Median :4.000
##  Mean   :3.855   Mean   :4.24        Mean   :3.777   Mean   :3.746
##  3rd Qu.:6.000   3rd Qu.:6.00        3rd Qu.:5.000   3rd Qu.:5.000
##  Max.   :8.000   Max.   :9.00        Max.   :8.000   Max.   :8.000
##  Clubbing.of.Finger.Nails Frequent.Cold     Dry.Cough        Snoring
```

```
##  Min.   :1.000           Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000           1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :4.000           Median :3.000   Median :4.000   Median :3.000
##  Mean   :3.923           Mean   :3.536   Mean   :3.853   Mean   :2.926
##  3rd Qu.:5.000           3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:4.000
##  Max.   :9.000           Max.   :7.000   Max.   :7.000   Max.   :7.000
##     Level
##  Length:1000
##  Class :character
##  Mode  :character
##
##
##
```

Excerpt 3:

```
sapply(df, function(x) {sum(is.na(x))})
```

```
##               Patient.Id                      Age                   Gender
##                        0                        0                        0
##            Air.Pollution              Alcohol.use             Dust.Allergy
##                        0                        0                        0
##     OccuPational.Hazards             Genetic.Risk     chronic.Lung.Disease
##                        0                        0                        0
##            Balanced.Diet                  Obesity                  Smoking
##                        0                        0                        0
##           Passive.Smoker               Chest.Pain        Coughing.of.Blood
##                        0                        0                        0
##                  Fatigue              Weight.Loss      Shortness.of.Breath
##                        0                        0                        0
##                 Wheezing    Swallowing.Difficulty Clubbing.of.Finger.Nails
##                        0                        0                        0
##            Frequent.Cold                Dry.Cough                  Snoring
##                        0                        0                        0
##                    Level
##                        0
```

Excerpt 4:

```
library(ggplot2)
library(gridExtra)
p1 = ggplot(df,aes(Snoring)) + geom_bar()
p2 = ggplot(df,aes(Dry.Cough)) + geom_bar()
p3 = ggplot(df,aes(Frequent.Cold)) + geom_bar()
p4 = ggplot(df,aes(Clubbing.of.Finger.Nails)) + geom_bar()
p5 = ggplot(df,aes(Swallowing.Difficulty)) + geom_bar()
p6 = ggplot(df,aes(Wheezing)) + geom_bar()
p7 = ggplot(df,aes(Shortness.of.Breath)) + geom_bar()
p8 = ggplot(df,aes(Weight.Loss)) + geom_bar()
p9 = ggplot(df,aes(Fatigue)) + geom_bar()
p10 = ggplot(df,aes(Coughing.of.Blood)) + geom_bar()
```

```
p11 = ggplot(df,aes(Chest.Pain)) + geom_bar()
p12 = ggplot(df,aes(Passive.Smoker)) + geom_bar()
grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12)
```

Excerpt 5:

```
p13 = ggplot(df,aes(Smoking)) + geom_bar()
p14 = ggplot(df,aes(Obesity)) + geom_bar()
p15 = ggplot(df,aes(Balanced.Diet)) + geom_bar()
p16 = ggplot(df,aes(chronic.Lung.Disease)) + geom_bar()
p17 = ggplot(df,aes(Genetic.Risk)) + geom_bar()
p18 = ggplot(df,aes(OccuPational.Hazards)) + geom_bar()
p19 = ggplot(df,aes(Dust.Allergy)) + geom_bar()
p20 = ggplot(df,aes(Alcohol.use)) + geom_bar()
p21 = ggplot(df,aes(Air.Pollution)) + geom_bar()

grid.arrange(p13,p14,p15,p16,p17,p18,p19,p20,p21)
```

Excerpt 6:

```
b1 = ggplot(df,aes(x=Level,y=Smoking)) + geom_boxplot()
b2 = ggplot(df,aes(x=Level,y=Obesity)) + geom_boxplot()
b3 = ggplot(df,aes(x=Level,y=Alcohol.use)) + geom_boxplot()
b4 = ggplot(df,aes(x=Level,y=Chest.Pain)) + geom_boxplot()
grid.arrange(b1,b2,b3,b4)
```

Excerpt 7:

```
df$Gender = factor(df$Gender)
ggplot(df, aes(x = Level, fill = Gender)) +
  geom_bar(stat='count', position='stack') +
  labs(x = 'Level colored by Gender')
```

Excerpt 8:

```
df$Level = factor(df$Level,ordered = TRUE, levels = c("Low","Medium","High"))

df$Gender = factor(df$Gender)

str(df)
```

```
## 'data.frame':    1000 obs. of  25 variables:
##  $ Patient.Id              : chr  "P1" "P10" "P100" "P1000" ...
##  $ Age                     : int  33 17 35 37 46 35 52 28 35 46 ...
##  $ Gender                  : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 2 2
1 ...
##  $ Air.Pollution           : int  2 3 4 7 6 4 2 3 4 2 ...
##  $ Alcohol.use             : int  4 1 5 7 8 5 4 1 5 3 ...
##  $ Dust.Allergy            : int  5 5 6 7 7 6 5 4 6 4 ...
```

```
##  $ OccuPational.Hazards    : int  4 3 5 7 7 5 4 3 5 2 ...
##  $ Genetic.Risk            : int  3 4 5 6 7 5 3 2 6 4 ...
##  $ chronic.Lung.Disease    : int  2 2 4 7 6 4 2 3 5 3 ...
##  $ Balanced.Diet           : int  2 2 6 7 7 6 2 4 5 3 ...
##  $ Obesity                 : int  4 2 7 7 7 7 4 3 5 3 ...
##  $ Smoking                 : int  3 2 2 7 8 2 3 1 6 2 ...
##  $ Passive.Smoker          : int  2 4 3 7 7 3 2 4 6 3 ...
##  $ Chest.Pain              : int  2 2 4 7 7 4 2 3 6 4 ...
##  $ Coughing.of.Blood       : int  4 3 8 8 9 8 4 1 5 4 ...
##  $ Fatigue                 : int  3 1 8 4 3 8 3 3 1 1 ...
##  $ Weight.Loss             : int  4 3 7 2 2 7 4 2 4 2 ...
##  $ Shortness.of.Breath     : int  2 7 9 3 4 9 2 2 3 4 ...
##  $ Wheezing                : int  2 8 2 1 1 2 2 4 2 6 ...
##  $ Swallowing.Difficulty   : int  3 6 1 4 4 1 3 2 4 5 ...
##  $ Clubbing.of.Finger.Nails: int  1 2 4 5 2 4 1 2 6 4 ...
##  $ Frequent.Cold           : int  2 1 6 6 4 6 2 3 2 2 ...
##  $ Dry.Cough               : int  3 7 7 7 2 7 3 4 4 1 ...
##  $ Snoring                 : int  4 2 2 5 3 2 4 3 1 5 ...
##  $ Level                   : Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 2
3 3 3 3 1 1 2 2 ...
```

Excerpt 9:

```
library(corrplot)

## corrplot 0.84 loaded

df2 = sapply(df,as.numeric)

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

c = cor(df2)
corrplot(c,tl.cex = 0.6)
```

Excerpt 10:

```
df = subset(df, select=-c(Patient.Id, Age, Gender, Wheezing, Swallowing.Diffi
culty, Clubbing.of.Finger.Nails, Dry.Cough, Snoring))
head(df)

##   Air.Pollution Alcohol.use Dust.Allergy OccuPational.Hazards Genetic.Risk
## 1             2           4            5                    4            3
## 2             3           1            5                    3            4
## 3             4           5            6                    5            5
## 4             7           7            7                    7            6
## 5             6           8            7                    7            7
## 6             4           5            6                    5            5
##   chronic.Lung.Disease Balanced.Diet Obesity Smoking Passive.Smoker Chest.
```

```
Pain
## 1                   2             2       4       3              2
2
## 2                   2             2       2       2              4
2
## 3                   4             6       7       2              3
4
## 4                   7             7       7       7              7
7
## 5                   6             7       7       8              7
7
## 6                   4             6       7       2              3
4
##   Coughing.of.Blood Fatigue Weight.Loss Shortness.of.Breath Frequent.Cold
## 1                 4       3           4                   2             2
## 2                 3       1           3                   7             1
## 3                 8       8           7                   9             6
## 4                 8       4           2                   3             6
## 5                 9       3           2                   4             4
## 6                 8       8           7                   9             6
##    Level
## 1    Low
## 2 Medium
## 3   High
## 4   High
## 5   High
## 6   High
```

Excerpt 11:

```
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

rf.fit = randomForest(Level~.,data=df)
importance(rf.fit)
```

```
##                       MeanDecreaseGini
## Air.Pollution                 21.98630
## Alcohol.use                   35.46243
## Dust.Allergy                  36.95352
## OccuPational.Hazards          23.41694
## Genetic.Risk                  26.25602
## chronic.Lung.Disease          12.46971
## Balanced.Diet                 40.36115
## Obesity                       81.39806
## Smoking                       20.21746
## Passive.Smoker                75.63726
## Chest.Pain                    21.74227
## Coughing.of.Blood            108.49103
## Fatigue                       65.29201
## Weight.Loss                   24.58980
## Shortness.of.Breath           42.87402
## Frequent.Cold                 26.91081
```

Excerpt 12:

```
df = subset(df, select=c(Coughing.of.Blood,Obesity,Passive.Smoker,Fatigue,Lev
el))
head(df)
```

```
##   Coughing.of.Blood Obesity Passive.Smoker Fatigue  Level
## 1                 4       4              2       3    Low
## 2                 3       2              4       1 Medium
## 3                 8       7              3       8   High
## 4                 8       7              7       4   High
## 5                 9       7              7       3   High
## 6                 8       7              3       8   High
```

Excerpt 13:

```
sample_size = floor(0.8* nrow(df))

train_index = sample.int(n = nrow(df), size = sample_size)

train = df[train_index,]
test = df[-train_index,]
```

Excerpt 14:

```
library(MASS)
folds <- rep_len(1:5, nrow(train)) #Creating the folds
lst = c()
```

```r
for(k in 1:5) {
    fold =  which(folds == k)
    train_k <- train[-fold,] #getting train set for fold k
    validation_k <- train[fold,] #getting validation set for fold k

    lda.fit = lda(Level~., data=train_k) #fitting the LDA model using all the
predictors we selected earlier

    predictions_k = predict(lda.fit, validation_k) #constructing predictions
on the validation set
    validation_set_error_k = mean(as.character(validation_k$Level) != as.char
acter(predictions_k$class)) # Must do this because Level     is encoded as an
ordinal factor and it cannot be compared directy

    print(validation_set_error_k)
    lst = c(lst,validation_set_error_k)
}
```

```
## [1] 0.125
## [1] 0.11875
## [1] 0.13125
## [1] 0.1375
## [1] 0.1875
```

```r
print(mean(lst)) #Finding the mean of the misclassification errors
```

```
## [1] 0.14
```

Excerpt 15:

```r
folds <- rep_len(1:5, nrow(train)) #creating the folds
lst = c()
for(k in 1:5) {

    fold =  which(folds == k)
    train_k <- train[-fold,] #getting train set for fold k
    validation_k <- train[fold,] #getting validation set for fold k

    qda.fit = qda(Level~., data=train_k) #fitting the qda using all the featu
res we previously defined

    predictions_k = predict(qda.fit, validation_k)
    validation_set_error_k = mean(as.character(validation_k$Level) != as.char
acter(predictions_k$class)) # Must do this because Level is encoded as an ord
inal factor and it cannot be compare directy

    print(validation_set_error_k)
```

```
    lst = c(lst,validation_set_error_k)
}
```

```
## [1] 0.10625
## [1] 0.1125
## [1] 0.125
## [1] 0.08125
## [1] 0.18125
```

```
print(mean(lst)) #getting average misclassification error
```

```
## [1] 0.12125
```

Excerpt 16:

```
# Separating out the features and the target variable
train_x = train[c("Coughing.of.Blood","Obesity","Passive.Smoker","Fatigue")]

train_y = train[c("Level")]

library(class)
folds <- rep_len(1:5, nrow(train)) #creating the folds

lst5 = c()
lst10 = c()
lst50 = c()
lst100 = c()
for(k in 1:5) {

    fold =  which(folds == k)
    train_x_k = train_x[-fold,] #getting our train and validate sets for fold
k
    validation_x_k = train_x[fold,]
    train_y_k = train_y[-fold,]
    validation_y_k = train_y[fold,]

    predict.knn_5 = knn(train_x_k, validation_x_k,train_y_k, k = 5) #construc
ting our models with the different ks
    predict.knn_10 = knn(train_x_k, validation_x_k,train_y_k, k = 10)
    predict.knn_50 = knn(train_x_k, validation_x_k,train_y_k, k = 50)
    predict.knn_100 = knn(train_x_k, validation_x_k,train_y_k, k = 100)

    validation_set_error_5 =  mean(as.character(validation_y_k) != as.charact
er(predict.knn_5)) #getting the misclassification rate
    validation_set_error_10 =  mean(as.character(validation_y_k) != as.charac
ter(predict.knn_10))
    validation_set_error_50 =  mean(as.character(validation_y_k) != as.charac
ter(predict.knn_50))
    validation_set_error_100 =  mean(as.character(validation_y_k) != as.chara
```

```
cter(predict.knn_100))

    lst5 = c(lst5,validation_set_error_5)
    lst10 = c(lst10,validation_set_error_10)
    lst50 = c(lst50,validation_set_error_50)
    lst100 = c(lst100,validation_set_error_100)}

cat("\nAverage misclassification rate for k = 5: ", mean(lst5)) #getting the
average misclassification rate for each value of k

##
## Average misclassification rate for k = 5:  0.0075

cat("\nAverage misclassification rate for k = 10: ", mean(lst10))

##
## Average misclassification rate for k = 10:  0.06375

cat("\nAverage misclassification rate for k = 50: ", mean(lst50))

##
## Average misclassification rate for k = 50:  0.19625

cat("\nAverage misclassification rate for k = 100: ", mean(lst100))

##
## Average misclassification rate for k = 100:  0.2075
```

Excerpt 17:

```
library(class)
#Grabbing features and target from my test set I previously had defined and s
et aside

test_y = test[c("Level")]
test_x = test[c("Coughing.of.Blood","Obesity","Passive.Smoker","Fatigue")]

cl = train_y[,1] #Extracting out the classes
predict.knn_final = knn(train = train_x, test = test_x, cl = cl, k = 5) #fitt
ing knn with k = 5

cl_test = test_y[,1]
mean(as.character(cl_test) != as.character(predict.knn_final)) #misclassifica
tion rate for my final model

## [1] 0.035

table(cl_test,predict.knn_final) #confusion matrix for my final predictions a
nd test set
```

```
##          predict.knn_final
## cl_test  Low Medium High
##   Low     51      7    0
##   Medium   0     70    0
##   High     0      0   72
```