|  |  |  |
| --- | --- | --- |
| A picture containing text, sign, clipart  Description automatically generated | **BOSTON**  **UNIVERSITY** | **METROPOLITAN COLLEGE** |

**AD 699 DATA MINING FOR BUSINESS ANALYTICS**

**ASSIGNMENT 3**

**MARCH 31, 2023**

**Aravind Hanumantha Rao**

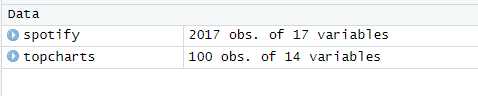
**BU ID - U55859882**

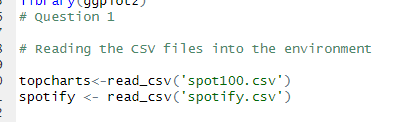
**Question 1-.**

Read the dataset spot100 into your environment. The dataset includes the 100 most

streamed songs on Spotify of all-time. In this dataset, each row represents one particular

song?





**Question 2**

**Extract the row that contains info for your song from spot100.csv. Verify**

**that your song is now stored as its own dataframe (if it’s not currently a dataframe,**

**convert it into one now).**

a.

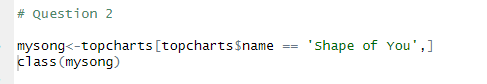
What song did you pick?

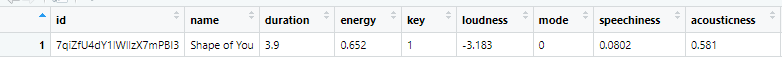
Ed Sheeran – Shape of you

b.

In a sentence or two, why did you pick this song?

My favourite artist is ed Sheeran and listen to his songs frequently and therefore I had choose his song for my assignment







c.

What values does your song have for the following categories:

danceability:0.825

energy:0.652

loudness:-3.183

speechiness:0.0802

acousticness:0.581

instrumentalness:0

liveness:0.0931

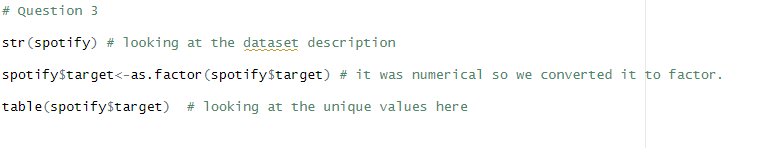
tempo:95.977

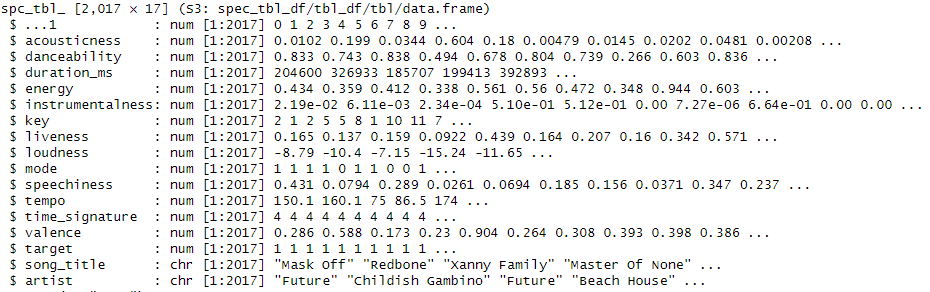
duration:3.9

valence:0.931

Question 3 (a)

Now, read spotify.csv dataset into your environment. Call the str() function on your dataset and show the results.



****

**What type of variable is target? If target is not currently a factor, convert it into a**

**factor. It will be our response variable in this model. Target tells us whether**

**George, the person who uploaded this dataset, liked the song. “1” means that**

**George liked it, and “0” means that he did not?**

****

The target variable is a binary variable that indicates whether George liked the song or not. Specifically, the variable takes on the values of either 0 or 1, where 0 means that George did not like the song, and 1 means that George did like the song. This type of variable is often referred to as a binary response variable or a binary outcome variable.

In R, the target variable is likely to be stored as an integer or numeric variable by default. In order to use it as a response variable in a statistical model, we need to convert it to a factor variable. A factor variable is a categorical variable that takes on a limited number of possible values or levels, and it is used to represent categorical data in R.

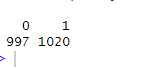
Question 3 (b)

**What unique values does the target variable have? For each of these outcome**

**values, ﬁnd out how many records in the dataset have that value, and state it**

**here?**

****

****

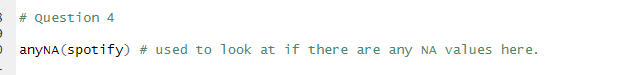
Based on the output you provided, the unique values in the target variable are 0 and 1. There are 997 records with a value of 0 and 1020 records with a value of 1 in the dataset.

Question 4

**Are there any NAs in this dataset? Show the code that you used to ﬁnd this out. If there**

**are any NA values in any particular column, replace them with the median value for**

**that column ?**

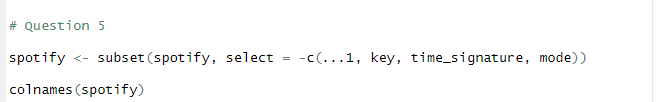
****

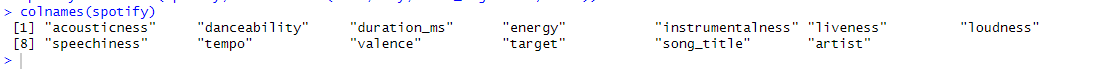
****

There aren’t any NA’s values in the dataset.

Question 5

**Remove these columns from the dataframe: X,key,mode, and timesignature. We won’t use them here ?**

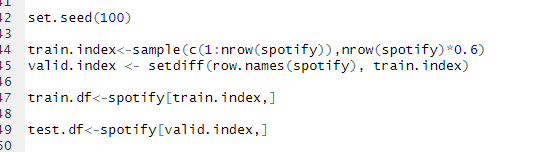


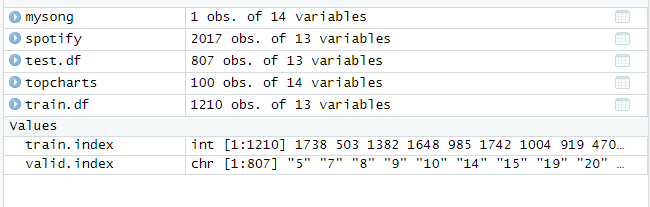


Question 6

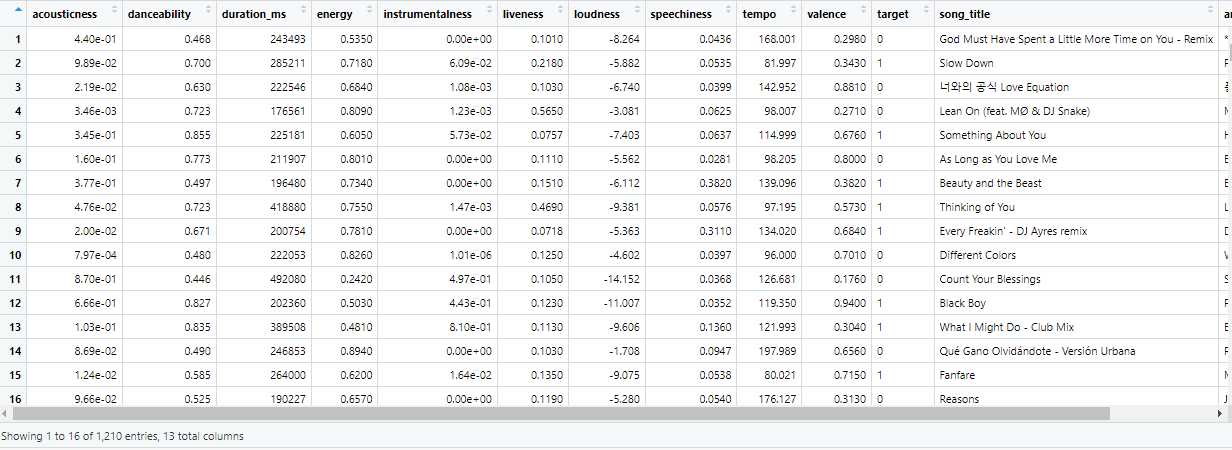
**Using your assigned seed value (from Assignment 2), partition the spotify dataset into**

**training (60%) and validation (40%) sets**

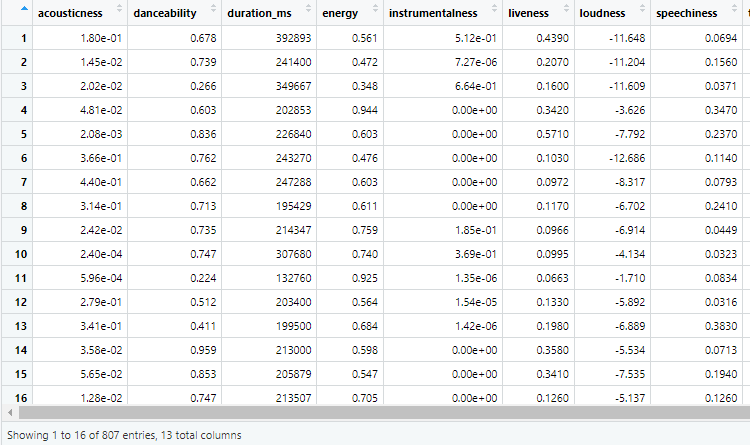




Train.df- sample pic (how it looks ) all columns not shown in pic



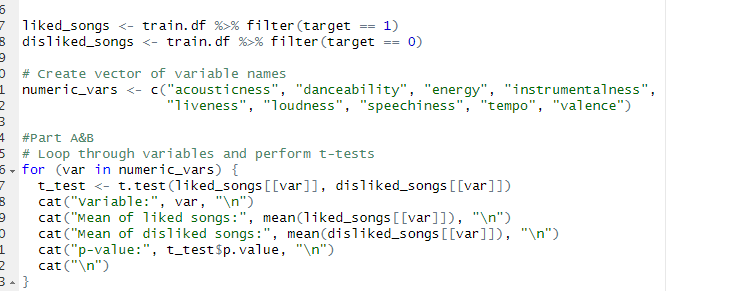
Test.df-sample pic (how it looks ) all columns not shown in pic



Question 7

**Using your training set, perform a series of two-sample t-tests to compare the**

**numeric values for songs that George liked vs ones that he did not like.**



Question 7 b

**Which variables show a signiﬁcant difference? For any variables for which there**

**is not a signiﬁcant difference between the ‘like’ and ‘dislike’ values, remove**

**them entirely from your data ?**

Variable: acousticness

Mean of liked songs: 0.1516721

Mean of disliked songs: 0.2193423

p-value: 3.788772e-06

Variable: danceability

Mean of liked songs: 0.6470575

Mean of disliked songs: 0.5895641

p-value: 2.370988e-10

Variable: energy

Mean of liked songs: 0.691821

Mean of disliked songs: 0.6756436

p-value: 0.1783516

Variable: instrumentalness

Mean of liked songs: 0.1698125

Mean of disliked songs: 0.08550324

p-value: 3.430511e-08

Variable: liveness

Mean of liked songs: 0.1982177

Mean of disliked songs: 0.1903945

p-value: 0.3928195

Variable: loudness

Mean of liked songs: -7.390412

Mean of disliked songs: -6.753955

p-value: 0.002709753

Variable: speechiness

Mean of liked songs: 0.1105642

Mean of disliked songs: 0.07784709

p-value: 2.976932e-10

Variable: tempo

Mean of liked songs: 122.2891

Mean of disliked songs: 122.2019

p-value: 0.9552488

Variable: valence

Mean of liked songs: 0.5208021

Mean of disliked songs: 0.4792301

p-value: 0.003344141

Based on the p-values obtained from the t-tests, the variables that show a significant difference between the 'like' and 'dislike' values are:

acousticness

danceability

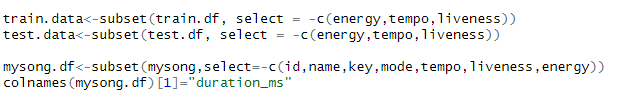
instrumentalness

loudness

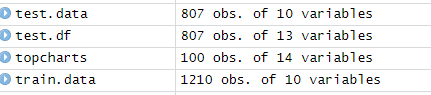
speechiness

valence

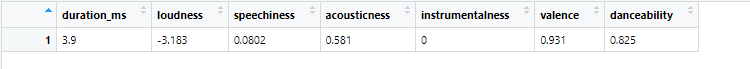
The variable 'energy' does not show a significant difference, as the p-value is greater than 0.05. The variable 'liveness' and 'tempo' also do not show a significant difference, as the p-value is much greater than 0.05. Therefore, we can remove the variables 'energy', 'liveness', 'tempo' from our data.



Seen in the changes on my code where energy, tempo and liveness is taken out . Even on my song I have made those changes. Duration also becomes duration\_ms



Test.data and train.data becomes 10 variables



Mysong.df becomes 7 variables with duration\_ms . helps forward in preprocessing .

Question 7 C

**In a sentence or two, why might it make sense to remove variables from a k-nn**

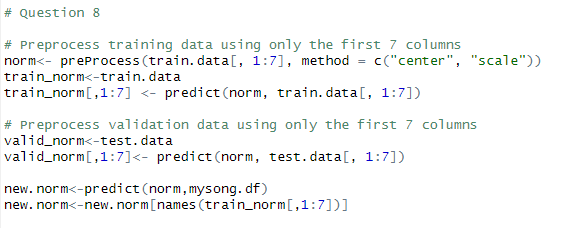
**model when those variables’ values are very similar for both outcome classes?**

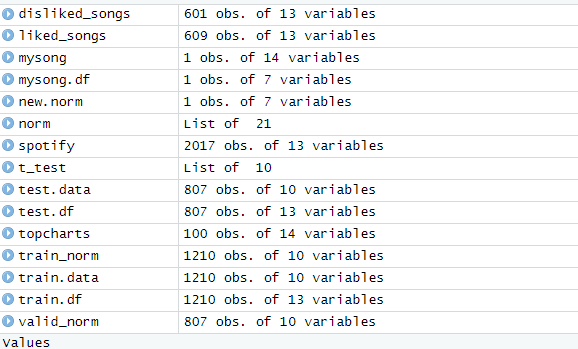
Removing variables that have similar values for both outcome classes in a k-nn model can help to improve the model's performance by reducing noise and increasing the signal-to-noise ratio. This is because variables with similar values for both classes do not contribute much to distinguishing between the two classes and can therefore be safely removed without losing much predictive power.

Question 8

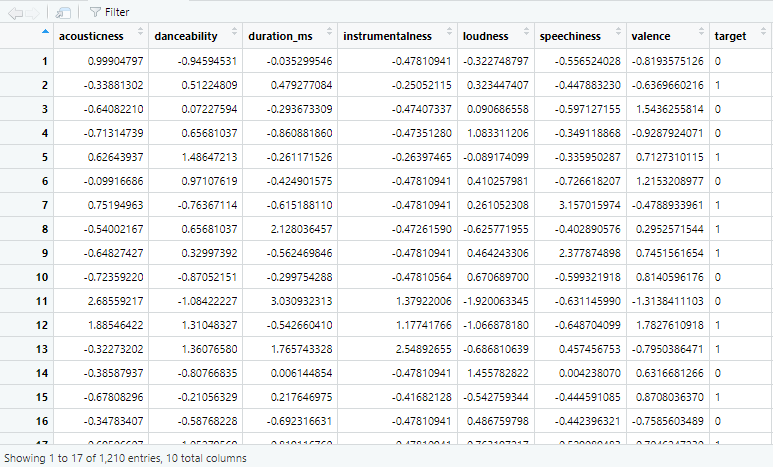
**Normalize your data using the preProcess() function from the caret package. Use**

**Table 7.2 from the book as a guide for this?**

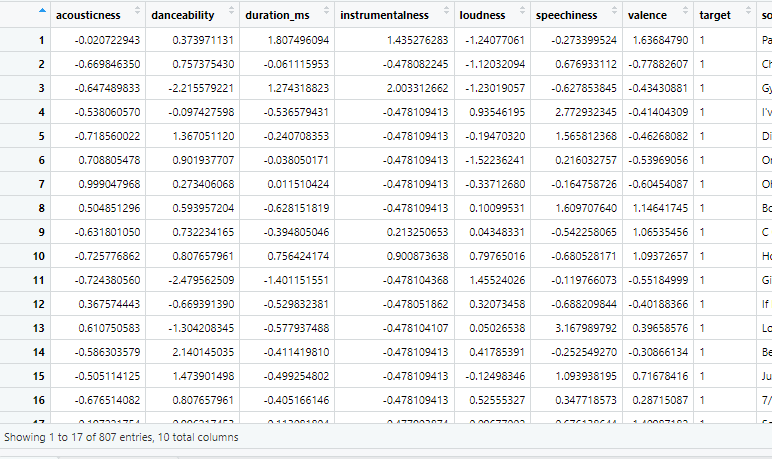
****

****

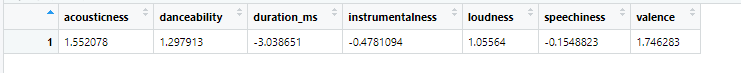
**Train\_norm**

****

**Valid\_norm**

****

**New.norm**

****

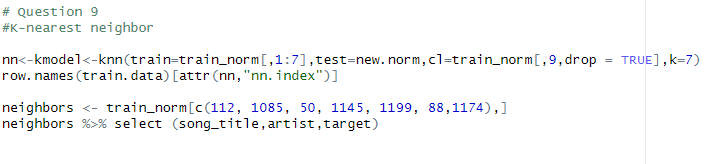
Question 9-

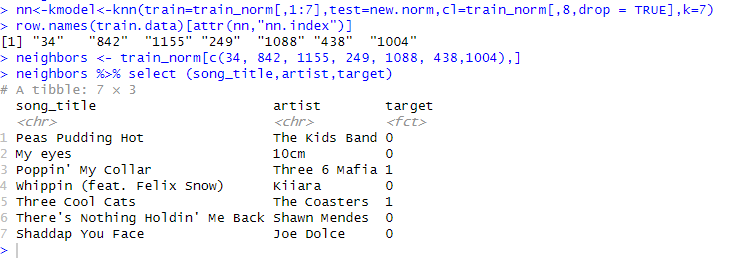
**Using the knn() function from the FNN package, and using a k-value of 7, generate a**

**predicted classiﬁcation for your song -- Will George like it or not? What outcome did**

**the model predict? Also, what were your song’s 7 nearest neighbors? List their titles,**

**artists, and outcome classes?**







Based on George's liking of the song and the comparison with my song, which is by Ed Sheeran, a prediction can be made about whether or not he likes the song.

This code is performing k-nearest neighbors (KNN) classification using a k-value of 7 to predict the target variable (like or dislike) of a new song represented by new.norm.

First, 7 nearest neighbors of new.norm are identified from the training data, as shown by the row numbers returned by attr(nn,"nn.index"). Then, the subset of the training data corresponding to these neighbors is extracted and displayed using neighbors %>% select (song\_title,artist,target).

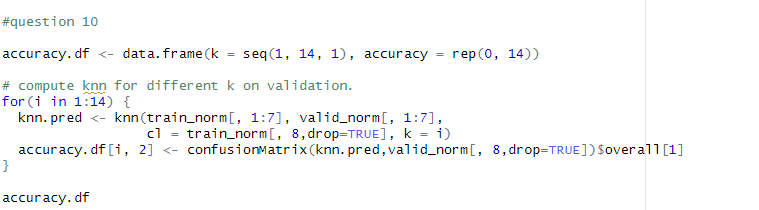
The displayed output is a tibble containing the song title, artist, and target variable (like or dislike) for each of the 7 nearest neighbors. This information could be useful in understanding the predicted target variable for the new song, as it allows for comparison to similar songs in the training data.

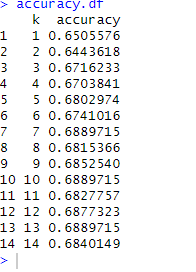
This is a table of the song title, artist and target for the 7 nearest neighbors to the new song in question. The target column shows whether each neighbor liked the song (1) or not (0). Based on this table, we can see that 2 out of the 7 nearest neighbors liked the song while the other 5 did not.

Question 10-

**Use your validation set to help you determine an optimal k-value. Use Table 7.3 from**

**the textbook as a guide here?**

****



This is the resulting dataframe accuracy.df after running a loop that computes the accuracy of k-nearest neighbors classification models for different values of k on a validation set. The dataframe contains two columns: "k" which corresponds to the value of k used for each model, and "accuracy" which corresponds to the accuracy of the model for that k value. The values of k range from 1 to 14, and the corresponding accuracies are shown in the dataframe.

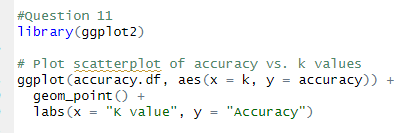
That being said ,K =10 has the most accuracy .

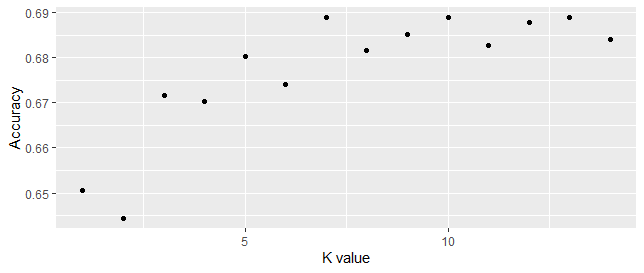
Question 11-

**Using either the base graphics package or ggplot, make a scatterplot with the various k**

**values that you used in the previous step on your x-axis, and the accuracy metrics on**

**the y-axis?**

****

****

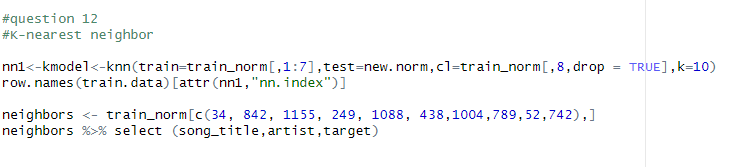
Question 12-

**Re-run your knn() function with the optimal k-value that you found previously. What**

**result did you obtain? Was it different from the result you saw when you ﬁrst ran the**

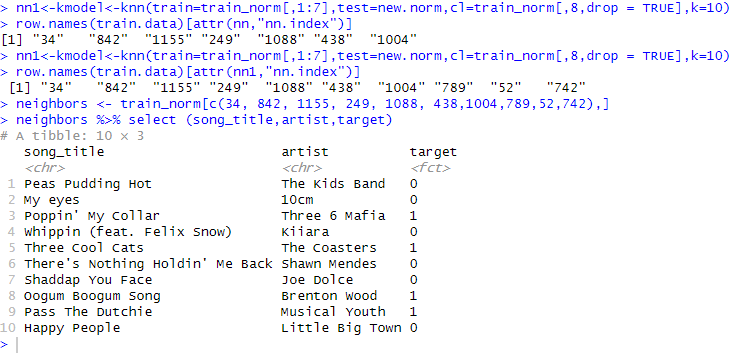
**k-nn function? Also, what were the outcome classes for each of your song’s k-nearest**

**neighbors? Be sure to show their outcome classes in your write-up?**

****

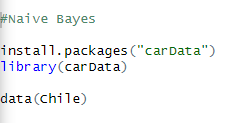


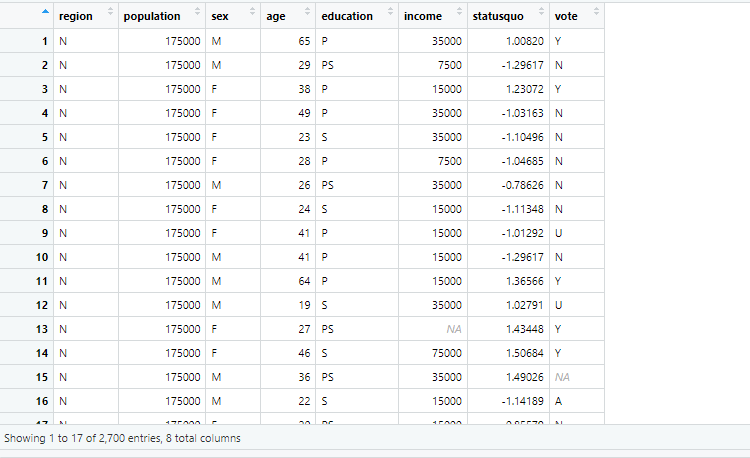
The result wasn’t different from the last time, its just more k values than before due to the rise in accuracy from before.



**Naïve bayes**

**Bring the Chile dataset from the CarData package in your local environment?**

****

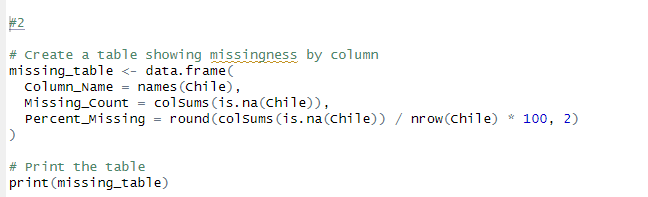
****

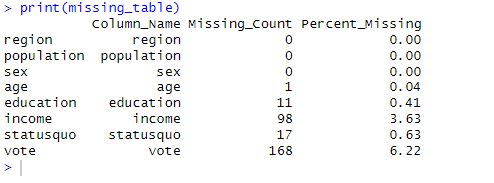
2.

**Exploring the dataset and preparing the variables**

**a. Generate a table showing missingness by column for the entire dataset (this is**

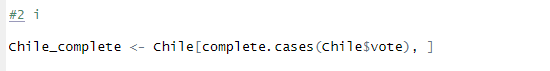
**very similar to something you did on Assignment 1).**

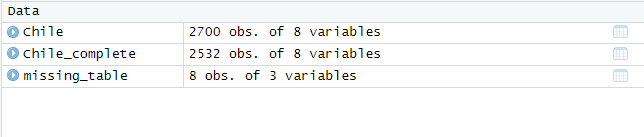




**i. The outcome variable in our model will be vote. If there are any rows**

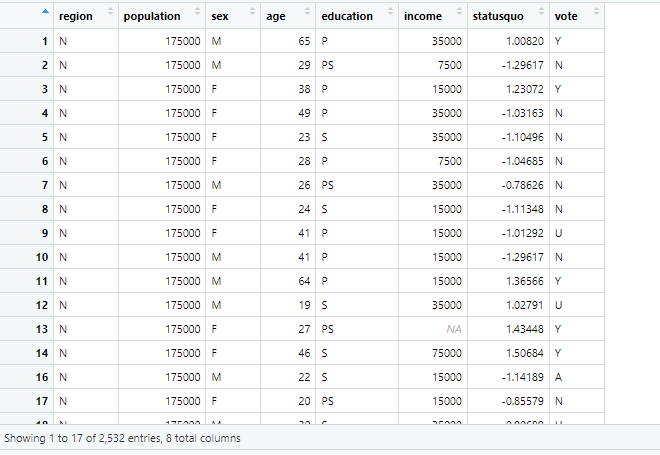
**with NA for vote, just remove those rows entirely.**





Rows deleted seen from the observations

Chile\_completed



**1. Why could it be problematic to impute values for the response**

**variable when preparing data for the modeling process?**

imputing values for the response variable can be problematic because it essentially involves predicting or guessing what the missing values should be, which can introduce bias into the modeling process.

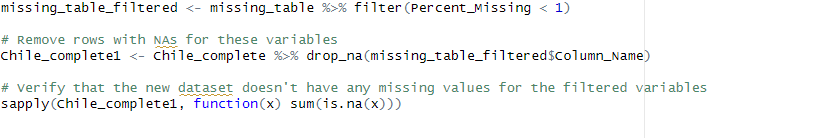
For example, if we have missing values for the response variable and we impute them with the mean or median of the observed values, this can lead to an underestimation or overestimation of the true variability of the response variable.

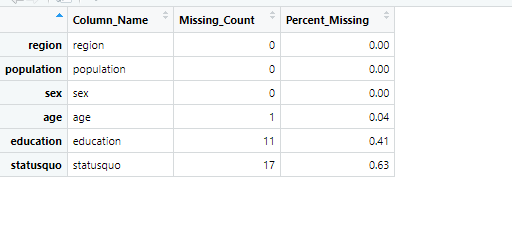
Additionally, imputing values for the response variable assumes that the missing values are missing completely at random (MCAR), which may not always be the case. In other words, the missingness of the response variable may depend on other variables in the dataset, and imputing values without taking this into account can lead to biased estimates and incorrect conclusions.

Therefore, it is generally recommended to try and obtain as much complete data as possible for the response variable, rather than imputing missing values, to avoid introducing bias and uncertainty into the modeling process.

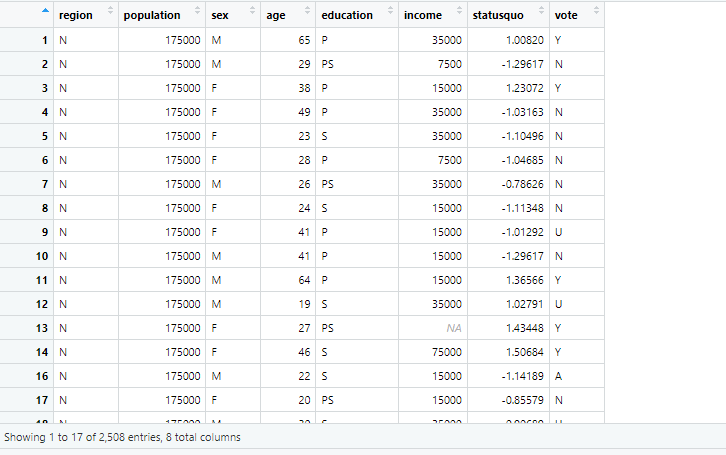
**ii. Which variables have less than 1% missingness? For any such variables,**

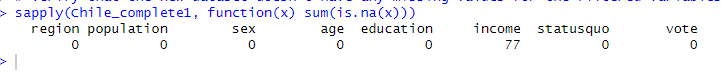
**remove the rows that have NAs for these.**





Chile\_complete1



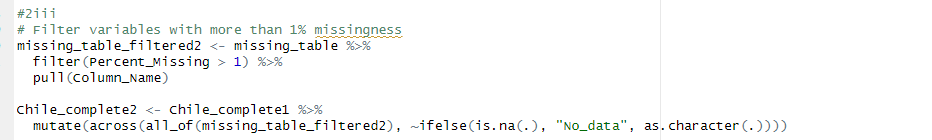


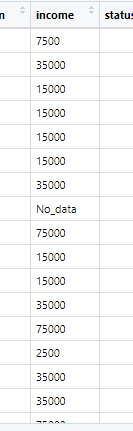
Proof – less than one percent data has been removed.

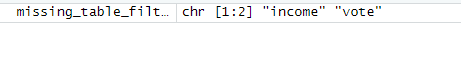
**iii. Which variable(s) have more than 1% missingness? For these, replace the**

**NAs with a separate text value - we will treat the NAs here as their own**

**category.**

****

****

****

Greater than one percent na values are vote and income but as we removed the na values in vote only income has the text no data.

**1. Why might an “NA” actually be an interesting value for some**

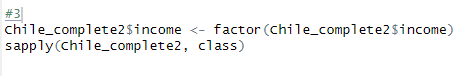
**variables, rather than simply being seen as a complete**

**‘information void’?**

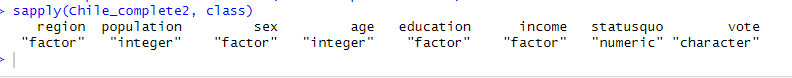
In some cases, an "NA" value may represent a valid and meaningful response for a variable. For example, if a survey question asks about the number of children someone has and the respondent answers "0", this could be coded as an "NA" if the question specifically asks for the number of biological children. In this case, the "NA" indicates that the respondent does not have any biological children, which is a meaningful response. Another example is when dealing with medical data where a missing value for a lab test could mean that the test was not administered to the patient due to some condition, which in itself can be an important piece of information. Therefore, treating "NA" as a complete information void may lead to misleading results and could even cause bias in the analysis.

Question 3-

**Are there any variables in the dataset seen as characters ?if so , convert them to factors now ?**

****

Income was a variable that was a character due to the text value. Income changed into a factor



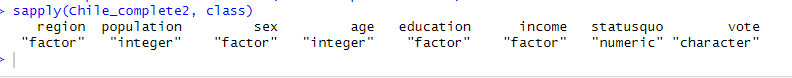
Question 4-

**For any numeric variables in your data, bin them into factors. Bin them using equal**

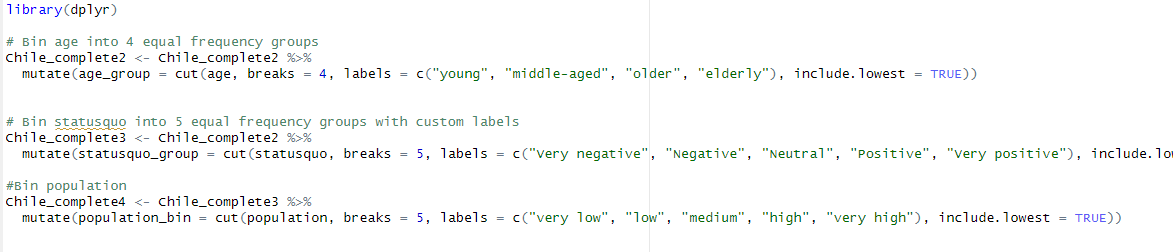
**frequency binning. Be sure to give a label to each bin. Select a bin number of your**

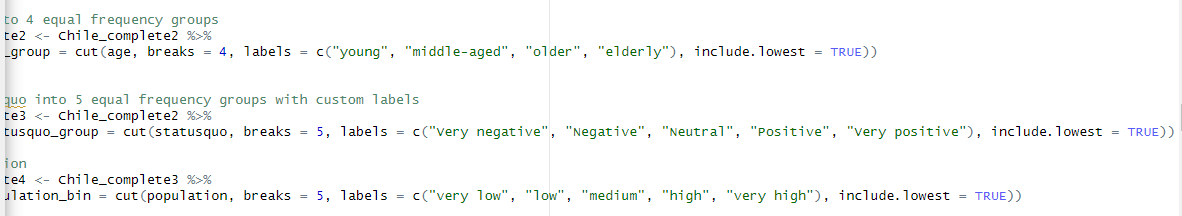
**choice. If you cannot place the numerical variables into bins with perfectly equal**

**frequency, just do the best you can with them?**

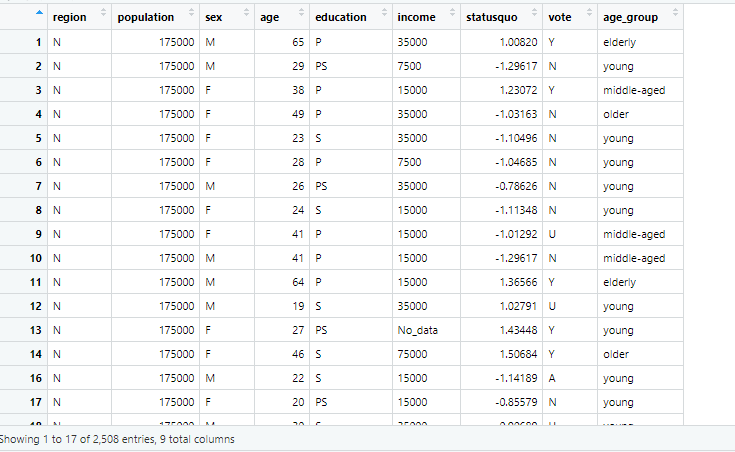


The numerical variables-population , integer and numeric ( statusquo , age and population)

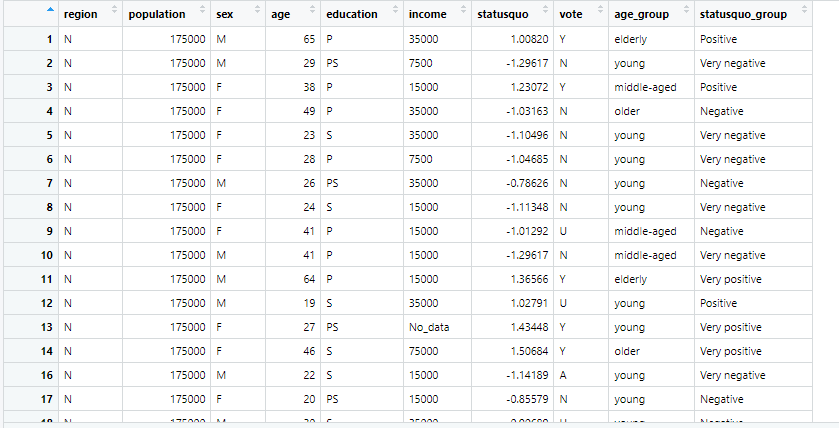
****



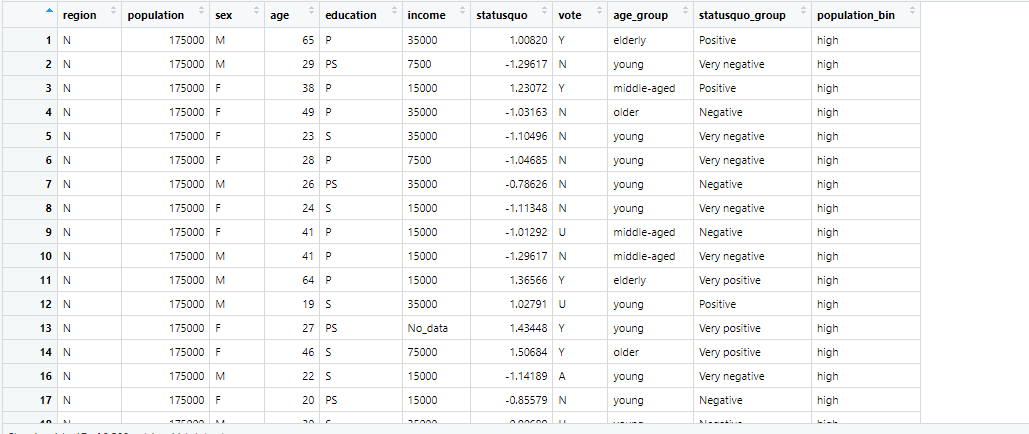
Age\_group



Age\_group and statusquo\_group

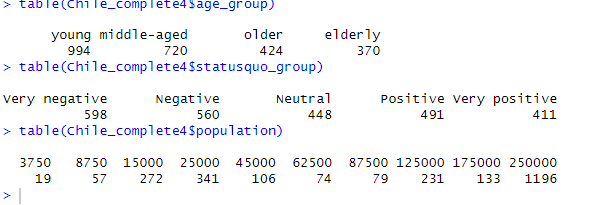


Age\_group and statusquo\_group and population



**Question 4 (a)**

**Show the results of this process, using the table() function.**

****

**Question 4 (b)**

**What is the difference between equal width binning and equal frequency**

**binning? Why might equal frequency binning be preferable in some scenarios?**

Equal width binning and equal frequency binning are two common methods for binning continuous data into discrete categories or intervals. The main difference between them is in how they determine the size of the bins.

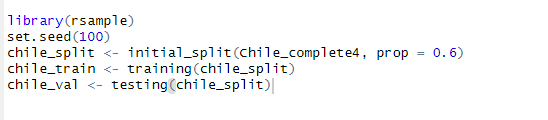
Equal width binning involves dividing the range of the data into equally sized intervals, regardless of how many data points fall into each interval. This method can be easy to implement, but it can be problematic if the distribution of the data is skewed or contains outliers, as it may result in some bins having very few or no data points, while others may contain a large number of data points.

Equal frequency binning, on the other hand, involves dividing the data into intervals that contain roughly the same number of data points. This method can be more appropriate in situations where the data is not evenly distributed, as it can result in more balanced bin sizes and fewer empty or overcrowded bins. However, it can also be more computationally intensive, as it may require sorting the data and calculating quantiles.

In general, equal frequency binning may be preferable in scenarios where the goal is to ensure that each bin contains a similar amount of information, or to minimize the impact of outliers on the resulting bins. However, the choice of binning method ultimately depends on the specific characteristics of the data and the goals of the analysis.

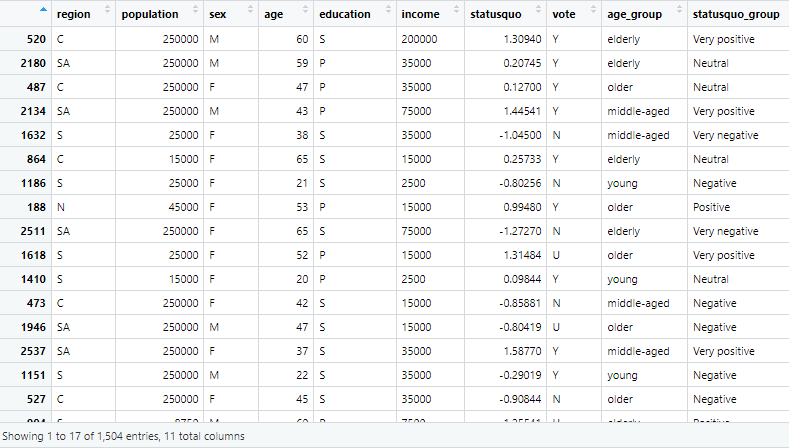
**Question 5**

**Using your seed value (the same one from Assignment #2)Partition your data into training (60%) and validation (40%) sets?**

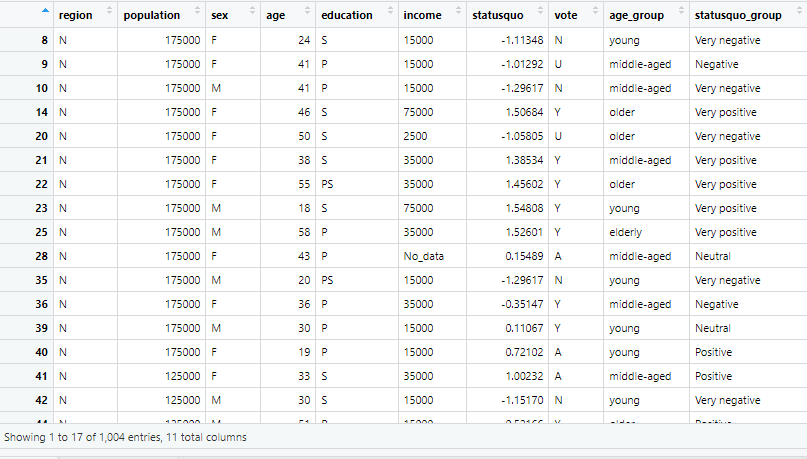




Chile\_train



Chile\_val



**Question 6-**

**Let’s take a look at the variables from the dataset, and explore the way that they might**

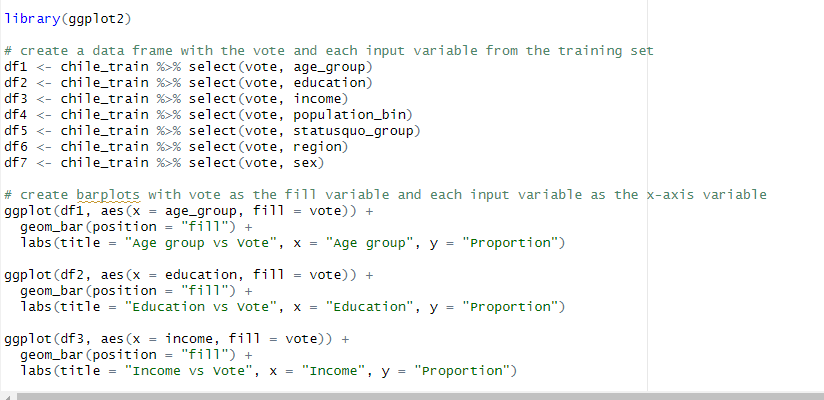
**impact vote. Using your training set data only, make a proportional barplot for each**

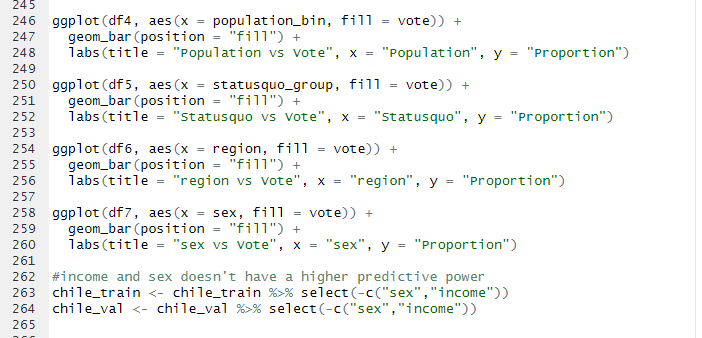
**one of your prospective input variables. Each barplot should show one of your input**

**variables as a category on the x-axis, with vote as the ﬁll variable. You should build**

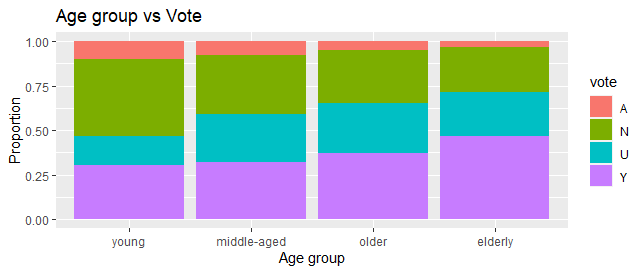
**proportional barplots (you can achieve this by adding position=”ﬁll” inside your geom**

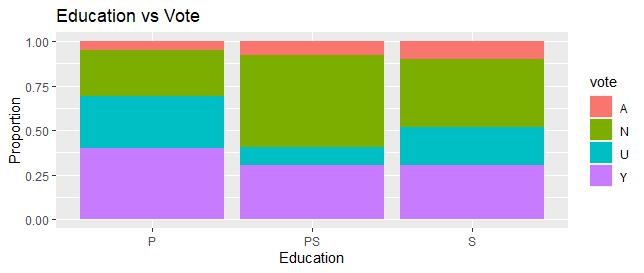
**layer)?**

****

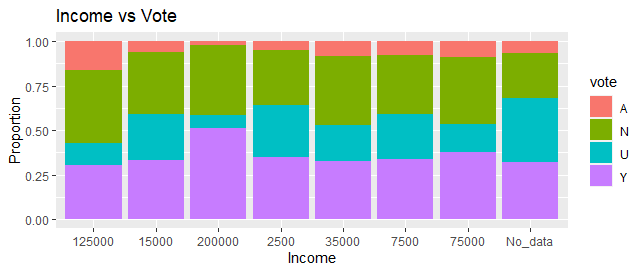
****

**1)**

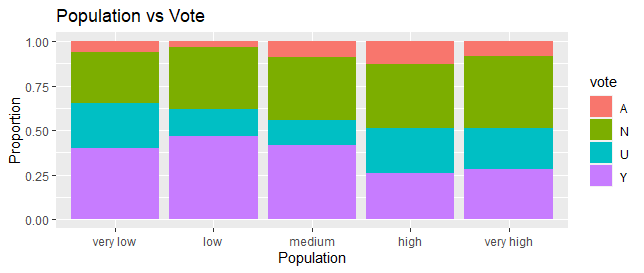
****

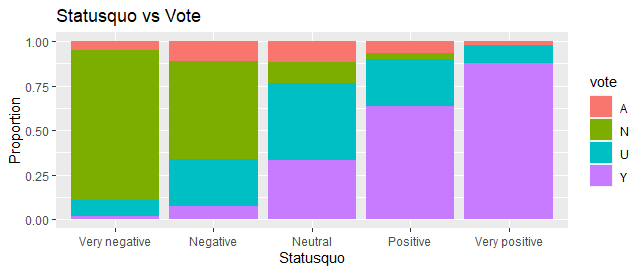
**2)**

**3)**

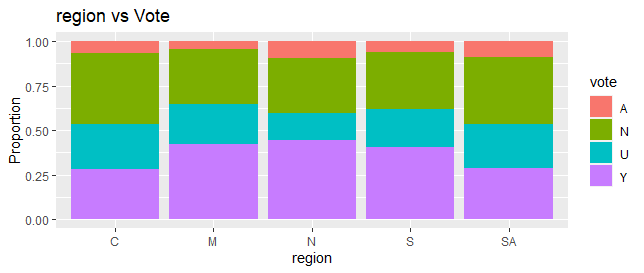
****

**4)**

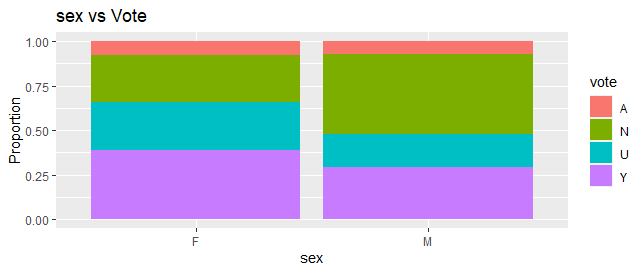
****

**5)**

**6)**

****

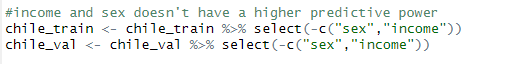
**7)**

****

**a. Based on the barplots that you see here, select any variable(s) that seem like they**

**will not have a strong amount of predictive power in a naive Bayes model. Drop**

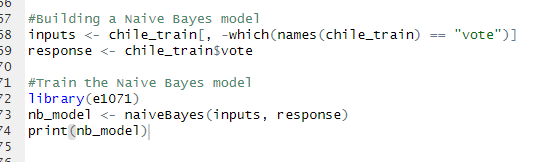
**any such variable.**

****

The decision to remove sex and income from the voting model was based on their lack of significant predictive power in determining whether a person votes or not, as evidenced by the uneven differences observed in the bar plot. Therefore, these variables were deemed unnecessary and were excluded to improve the accuracy of the model.

**Question 7**

**Build a naive bayes model, with the response variable vote. Use all of the other remaining variables in your training set as inputs. Show your model results?**



Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = inputs, y = response)

A-priori probabilities:

response

A N U Y

0.07579787 0.35505319 0.22739362 0.34175532

Conditional probabilities:

region

response C M N S SA

A 0.20175439 0.01754386 0.14912281 0.22807018 0.40350877

N 0.24344569 0.02621723 0.10674157 0.25655431 0.36704120

U 0.23976608 0.02923977 0.07894737 0.26900585 0.38304094

Y 0.18093385 0.03696498 0.15758755 0.33657588 0.28793774

population

response [,1] [,2]

A 163728.1 98489.14

N 165159.2 98806.07

U 147244.2 106052.75

Y 131208.7 102323.97

age

response [,1] [,2]

A 33.30702 12.01704

N 35.51685 14.11965

U 40.84503 14.17556

Y 40.45136 15.33825

education

response P PS S

A 0.26315789 0.17543860 0.56140351

N 0.29775281 0.25468165 0.44756554

U 0.52339181 0.07894737 0.39766082

Y 0.47665370 0.15369650 0.36964981

statusquo

response [,1] [,2]

A -0.17905386 0.7838863

N -0.93092032 0.4540481

U 0.04972436 0.7860015

Y 0.92049276 0.6382959

age\_group

response young middle-aged older elderly

A 0.51754386 0.31578947 0.10526316 0.06140351

N 0.48314607 0.27902622 0.13108614 0.10674157

U 0.28947368 0.35380117 0.19590643 0.16081871

Y 0.34824903 0.27821012 0.16926070 0.20428016

statusquo\_group

response Very negative Negative Neutral Positive Very positive

A 0.166666667 0.333333333 0.280701754 0.175438596 0.043859649

N 0.563670412 0.355805243 0.058052434 0.020599251 0.001872659

U 0.096491228 0.266081871 0.339181287 0.230994152 0.067251462

Y 0.011673152 0.048638132 0.173151751 0.369649805 0.396887160

population\_bin

response very low low medium high very high

A 0.27192982 0.02631579 0.10526316 0.08771930 0.50877193

N 0.26404494 0.05617978 0.09176030 0.05243446 0.53558052

U 0.37134503 0.03801170 0.05555556 0.05847953 0.47660819

Y 0.38715953 0.07782101 0.11089494 0.03891051 0.38521401

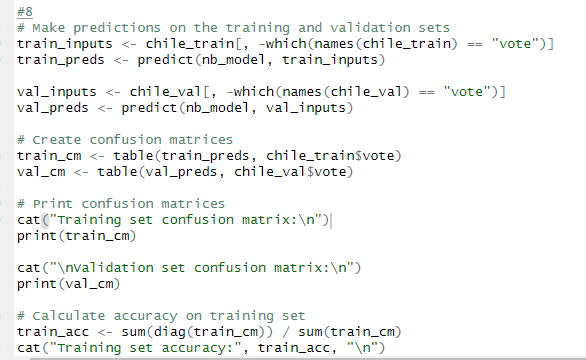
**Question 8**

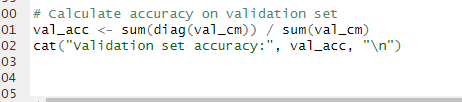
**Show a confusion matrix that compares the performance of your model against the**

**training data, and another that shows its performance against the validation data (just**

**use the accuracy metric for this analysis). How did your training set’s performance**

**compare with your validation set’s performance?**

****

****

|  |
| --- |
| > # Make predictions on the training and validation sets  > train\_inputs <- chile\_train[, -which(names(chile\_train) == "vote")]  > train\_preds <- predict(nb\_model, train\_inputs)  >  > val\_inputs <- chile\_val[, -which(names(chile\_val) == "vote")]  > val\_preds <- predict(nb\_model, val\_inputs)  >  > # Create confusion matrices  > train\_cm <- table(train\_preds, chile\_train$vote)  > val\_cm <- table(val\_preds, chile\_val$vote)  >  > # Print confusion matrices  > cat("Training set confusion matrix:\n")  Training set confusion matrix:  > print(train\_cm)    train\_preds A N U Y  A 12 10 14 14  N 53 479 104 25  U 21 27 110 59  Y 28 18 114 416  >  > cat("\nValidation set confusion matrix:\n")  Validation set confusion matrix:  > print(val\_cm)    val\_preds A N U Y  A 5 10 17 11  N 34 300 79 19  U 13 26 63 32  Y 19 16 75 285  >  > # Calculate accuracy on training set  > train\_acc <- sum(diag(train\_cm)) / sum(train\_cm)  > cat("Training set accuracy:", train\_acc, "\n")  Training set accuracy: 0.6761968  >  > # Calculate accuracy on validation set  > val\_acc <- sum(diag(val\_cm)) / sum(val\_cm)  > cat("Validation set accuracy:", val\_acc, "\n")  Validation set accuracy: 0.6503984 |
|  |
| |  | | --- | |  | |

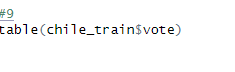
**Question 9**

**In classiﬁcation, what is the naive rule? If you had used the naive rule as an approach**

**to classiﬁcation, how would you have classiﬁed all the records in your training set?**

**(Note: Although their names are very similar, the naive rule for classiﬁcation is very**

**different from a naive Bayes approach to classiﬁcation).**

****

|  |
| --- |
| > table(chile\_train$vote)  A N U Y  114 534 342 514 |
|  |
| |  | | --- | | > | |

**534/1504**

0.35 percent

**No’s were the most featured and if classified in percentage , its 35 percentage.**

**a. How did your model’s accuracy compare with the naive rule accuracy, in**

**percentage terms? (Answer in percentage di erence, not in percentage points**

**difference).**

**0.65-0.35/0/65 =0.46**

A 0.46 percentage change in values

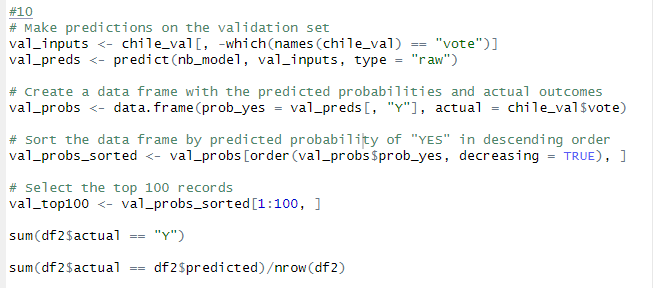
**Next, take a subset of the 100 records in your validation set that your model predicted**

**to be most likely to vote “YES.” (Table 8.6 in the textbook will be a very good thing to**

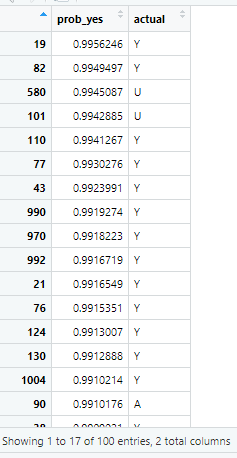
**look at in order to build this)**

**a. Among those 100 records, how many of the people actually voted yes? How**

**does the accuracy for these predictions compare to the overall model?**



**How does the accuracy for these predictions compare to the overall model?**



**b. How could a political party use this information? In a few sentences, indicate**

**what it would mean for a political party to be able to identify this particular**

**subset of records, and the way they could act on this information**.

If a political party were to identify a subset of records that have a high likelihood of not voting, they could use this information to strategically target their campaign efforts towards individuals who are more likely to vote for their party. This could involve focusing resources on encouraging and mobilizing these individuals to vote, or even tailoring their campaign messages and platforms to appeal specifically to this demographic.

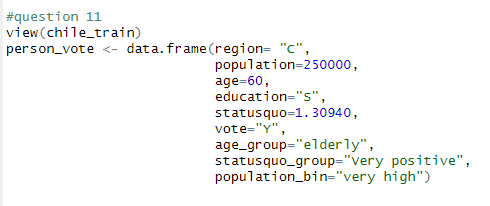
Additionally, if a political party were able to identify the factors that are most strongly correlated with voting behavior, such as age, education level, and political affiliation, they could use this information to better understand their voter base and develop more effective campaign strategies. By leveraging data analytics and machine learning techniques to analyze large volumes of voter data, political parties can gain valuable insights into the preferences and behaviors of their constituents, and use this information to improve their chances of electoral success.

11.

Pick any ONE record from your training set. It can be any row in the training set – it

doesn’t matter.

1. How did this person vote in the referendum?



1. Use the predict() function to see what your model predicted for this person.



What did it predict? (Table 8.6 in the textbook might be helpful here)

person\_vote\_pred <- predict(nb\_model,person\_vote)

> person\_vote\_pred

[1] Y

Levels: A N U Y

c. Now, use the predict() function again but with a slight modiﬁcation, in order to

have it generate the probability that your record would vote “Yes.” What was the

probability?

|  |
| --- |
| > person\_vote\_prob <- predict(nb\_model, person\_vote, type = "raw")  > person\_vote\_prob\_Y <- person\_vote\_prob[,"Y"]  > person\_vote\_prob\_Y  Y  0.9524009 |
|  |
| |  | | --- | | > | |