```
#Loaded the appropriate packages for analysis
library(tidyverse)
library(tidymodels)
library(dplyr)
library(purrr)
```

```
── Attaching packages ───────────────────────────────────────
tidyverse 1.3.1 ──

✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.7     ✔ dplyr   1.0.9
✔ tidyr   1.2.0     ✔ stringr 1.4.0
✔ readr   2.1.2     ✔ forcats 0.5.1

── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()

── Attaching packages ──────────────────────────────────────
tidymodels 1.0.0 ──

✔ broom        1.0.0     ✔ rsample      1.0.0
✔ dials        1.0.0     ✔ tune         1.0.0
✔ infer        1.0.2     ✔ workflows    1.0.0
✔ modeldata    1.0.0     ✔ workflowsets 1.0.0
✔ parsnip      1.0.0     ✔ yardstick    1.0.0
✔ recipes      1.0.1

── Conflicts ─────────────────────────────────────────
tidymodels_conflicts() ──
✖ scales::discard() masks purrr::discard()
✖ dplyr::filter()   masks stats::filter()
✖ recipes::fixed()  masks stringr::fixed()
✖ dplyr::lag()      masks stats::lag()
✖ yardstick::spec() masks readr::spec()
✖ recipes::step()   masks stats::step()
• Learn how to get started at https://www.tidymodels.org/start/
```

```
#Read the data, and added colummn names/headings
cleveland<-read.delim("processed.cleveland.data", header=FALSE,
sep=",")
cleveland<- rename(cleveland,
                   age= V1,
                   sex= V2,
                   cp= V3,
                   trestbp= V4,
                   chol= V5,
                   fbs= V6,
                   restecg= V7,
                thalach= V8,
                   exang = V9,
                   oldpeak = V10,
                   slope =V11,
                   ca = V12,
                   thal = V13,
                   num= V14)
#Converted the num column(which tells us the severity and if the
patient has heart disease) to a factor
cleveland|> mutate(num=as_factor(num))
head(cleveland)
```

```
    age sex cp trestbp chol fbs restecg thalach exang oldpeak slope ca
thal
1   63  1   1  145     233  1   2       150     0     2.3     3
0.0 6.0
2   67  1   4  160     286  0   2       108     1     1.5     2
3.0 3.0
3   67  1   4  120     229  0   2       129     1     2.6     2
2.0 7.0
4   37  1   3  130     250  0   0       187     0     3.5     3
0.0 3.0
5   41  0   2  130     204  0   2       172     0     1.4     1
0.0 3.0
6   56  1   2  120     236  0   0       178     0     0.8     1
0.0 3.0
7   62  0   4  140     268  0   2       160     0     3.6     3
2.0 3.0
8   57  0   4  120     354  0   0       163     1     0.6     1
0.0 3.0
9   63  1   4  130     254  0   2       147     0     1.4     2
1.0 7.0
10  53  1   4  140     203  1   2       155     1     3.1     3
0.0 7.0
11  57  1   4  140     192  0   0       148     0     0.4     2
0.0 6.0
12  56  0   2  140     294  0   2       153     0     1.3     2
0.0 3.0
13  56  1   3  130     256  1   2       142     1     0.6     2
1.0 6.0
14  44  1   2  120     263  0   0       173     0     0.0     1
0.0 7.0
15  52  1   3  172     199  1   0       162     0     0.5     1
0.0 7.0
16  57  1   3  150     168  0   0       174     0     1.6     1
0.0 3.0
17  48  1   2  110     229  0   0       168     0     1.0     3
0.0 7.0
18  54  1   4  140     239  0   0       160     0     1.2     1
```

```
0.0 3.0
19  48  0   3  130     275  0   0       139     0     0.2     1
0.0 3.0
20  49  1   2  130     266  0   0       171     0     0.6     1
0.0 3.0
21  64  1   1  110     211  0   2       144     1     1.8     2
0.0 3.0
22  58  0   1  150     283  1   2       162     0     1.0     1
0.0 3.0
23  58  1   2  120     284  0   2       160     0     1.8     2
0.0 3.0
24  58  1   3  132     224  0   2       173     0     3.2     1
2.0 7.0
25  60  1   4  130     206  0   2       132     1     2.4     2
2.0 7.0
26  50  0   3  120     219  0   0       158     0     1.6     2
0.0 3.0
27  58  0   3  120     340  0   0       172     0     0.0     1
0.0 3.0
28  66  0   1  150     226  0   0       114     0     2.6     3
0.0 3.0
29  43  1   4  150     247  0   0       171     0     1.5     1
0.0 3.0
30  40  1   4  110     167  0   2       114     1     2.0     2
0.0 7.0
⋮  ⋮   ⋮  ⋮ ⋮       ⋮   ⋮  ⋮       ⋮      ⋮     ⋮      ⋮    ⋮   ⋮
274 71  0   4  112     149  0   0       125     0     1.6     2
0.0 3.0
275 59  1   1  134     204  0   0       162     0     0.8     1
2.0 3.0
276 64  1   1  170     227  0   2       155     0     0.6     2
0.0 7.0
277 66  0   3  146     278  0   2       152     0     0.0     2
1.0 3.0
278 39  0   3  138     220  0   0       152     0     0.0     2
0.0 3.0
279 57  1   2  154     232  0   2       164     0     0.0     1
1.0 3.0
280 58  0   4  130     197  0   0       131     0     0.6     2
0.0 3.0
281 57  1   4  110     335  0   0       143     1     3.0     2
1.0 7.0
282 47  1   3  130     253  0   0       179     0     0.0     1
0.0 3.0
283 55  0   4  128     205  0   1       130     1     2.0     2
1.0 7.0
284 35  1   2  122     192  0   0       174     0     0.0     1
0.0 3.0
285 61  1   4  148     203  0   0       161     0     0.0     1
1.0 7.0
```

```
286 58  1   4  114     318  0   1       140     0     4.4     3
3.0 6.0
287 58  0   4  170     225  1   2       146     1     2.8     2
2.0 6.0
288 58  1   2  125     220  0   0       144     0     0.4     2     ?
7.0
289 56  1   2  130     221  0   2       163     0     0.0     1
0.0 7.0
290 56  1   2  120     240  0   0       169     0     0.0     3
0.0 3.0
291 67  1   3  152     212  0   2       150     0     0.8     2
0.0 7.0
292 55  0   2  132     342  0   0       166     0     1.2     1
0.0 3.0
293 44  1   4  120     169  0   0       144     1     2.8     3
0.0 6.0
294 63  1   4  140     187  0   2       144     1     4.0     1
2.0 7.0
295 63  0   4  124     197  0   0       136     1     0.0     2
0.0 3.0
296 41  1   2  120     157  0   0       182     0     0.0     1
0.0 3.0
297 59  1   4  164     176  1   2        90     0     1.0     2
2.0 6.0
298 57  0   4  140     241  0   0       123     1     0.2     2
0.0 7.0
299 45  1   1  110     264  0   0       132     0     1.2     2
0.0 7.0
300 68  1   4  144     193  1   0       141     0     3.4     2
2.0 7.0
301 57  1   4  130     131  0   0       115     1     1.2     2
1.0 7.0
302 57  0   2  130     236  0   2       174     0     0.0     2
1.0 3.0
303 38  1   3  138     175  0   0       173     0     0.0     1     ?
3.0
    num
1   0
2   2
3   1
4   0
5   0
6   0
7   3
8   0
9   2
10  1
11  0
12  0
13  2
```

```
14  0
15  0
16  0
17  1
18  0
19  0
20  0
21  0
22  0
23  1
24  3
25  4
26  0
27  0
28  0
29  0
30  3
⋮   ⋮
274 0
275 1
276 0
277 0
278 0
279 1
280 0
281 2
282 0
283 3
284 0
285 2
286 4
287 2
288 0
289 0
290 0
291 1
292 0
293 2
294 2
295 1
296 0
297 3
298 1
299 1
300 2
301 3
302 1
303 0
```

```
  age sex cp trestbp chol fbs restecg thalach exang oldpeak slope ca
thal num
1 63  1   1  145     233  1   2       150     0     2.3     3     0.0
6.0  0
2 67  1   4  160     286  0   2       108     1     1.5     2     3.0
3.0  2
3 67  1   4  120     229  0   2       129     1     2.6     2     2.0
7.0  1
4 37  1   3  130     250  0   0       187     0     3.5     3     0.0
3.0  0
5 41  0   2  130     204  0   2       172     0     1.4     1     0.0
3.0  0
6 56  1   2  120     236  0   0       178     0     0.8     1     0.0
3.0  0
```

Note that 0 means no heart disease, and 1-4 mean increasing levels of heart disease

```
#check for NA values
NA_<-sum(is.na(cleveland))
NA_
```

```
[1] 0
```

There are no NA values in our dataset, therefore we will not need to use any functions to account for NA values.

```
#selected only the columns that we will be using for our analysis

cleveland_0and4<-cleveland|> select(age, trestbp, chol, fbs, num)|
>mutate(num=as_factor(num))|> filter(num=="0"|num=="4")
cleveland_0and4
```

```
    age trestbp chol fbs num
1   63  145     233  1   0
2   37  130     250  0   0
3   41  130     204  0   0
4   56  120     236  0   0
5   57  120     354  0   0
6   57  140     192  0   0
7   56  140     294  0   0
8   44  120     263  0   0
9   52  172     199  1   0
10  57  150     168  0   0
11  54  140     239  0   0
12  48  130     275  0   0
13  49  130     266  0   0
14  64  110     211  0   0
15  58  150     283  1   0
16  60  130     206  0   4
17  50  120     219  0   0
18  58  120     340  0   0
```

```
19  66  150     226  0   0
20  43  150     247  0   0
21  69  140     239  0   0
22  59  135     234  0   0
23  44  130     233  0   0
24  42  140     226  0   0
25  61  150     243  1   0
26  65  150     225  0   4
27  40  140     199  0   0
28  71  160     302  0   0
29  59  150     212  1   0
30  58  112     230  0   4
⋮   ⋮   ⋮         ⋮    ⋮   ⋮
148 60  120     178  1   0
149 62  128     208  1   0
150 57  110     201  0   0
151 64  128     263  0   0
152 51  120     295  0   0
153 43  115     303  0   0
154 42  120     209  0   0
155 67  106     223  0   0
156 76  140     197  0   0
157 70  156     245  0   0
158 44  118     242  0   0
159 60  150     240  0   0
160 44  120     226  0   0
161 61  138     166  0   4
162 42  130     180  0   0
163 66  160     228  0   0
164 71  112     149  0   0
165 64  170     227  0   0
166 66  146     278  0   0
167 39  138     220  0   0
168 58  130     197  0   0
169 47  130     253  0   0
170 35  122     192  0   0
171 58  114     318  0   4
172 58  125     220  0   0
173 56  130     221  0   0
174 56  120     240  0   0
175 55  132     342  0   0
176 41  120     157  0   0
177 38  138     175  0   0
```

```
#Is the data balanced?
cleveland_balancecheck_0and4<- cleveland|> group_by(num)|>
summarize(count=n()) |>filter(num=="0"|num=="4")
cleveland_balancecheck_0and4
#The data is heavily imbalanced.
```

```
  num count
1 0   164
2 4    13
```

It might be better to use the numbers 0 and 3 or 0 and 2 or 0 and 1 for the presence and absence of heart disease. This is because severe heart disease might be very rare, and this classifier might be more helpful to the public if we use a numbers 2 or 3 which denote less severe heart disease, which is more common. This might also fix the issue of severe imbalance.

```
#0 and 3 test
cleveland_0and3<-cleveland|> select(age, trestbp, chol, fbs, num)|
>mutate(num=as_factor(num))|> filter(num=="0"|num=="3")
#balance check
cleveland_balancecheck_0and3<- cleveland_0and3|> group_by(num)|>
summarize(count=n())
cleveland_balancecheck_0and3

#0 and 2 test
cleveland_0and2<-cleveland|> select(age, trestbp, chol, fbs, num)|
>mutate(num=as_factor(num))|> filter(num=="0"|num=="2")
cleveland_0and2
#balance check
cleveland_balancecheck_0and2<- cleveland_0and2|> group_by(num)|>
summarize(count=n())
cleveland_balancecheck_0and2

#0 and 1 test
cleveland_0and1<-cleveland|> select(age, trestbp, chol, fbs, num)|
>mutate(num=as_factor(num))|> filter(num=="0"|num=="1")
cleveland_0and1
#balance check
cleveland_balancecheck_0and1<- cleveland_0and1|> group_by(num)|>
summarize(count=n())
cleveland_balancecheck_0and1
```

```
  num count
1 0   164
2 3    35
```

```
    age trestbp chol fbs num
1   63  145     233  1   0
2   67  160     286  0   2
3   37  130     250  0   0
4   41  130     204  0   0
5   56  120     236  0   0
6   57  120     354  0   0
7   63  130     254  0   2
8   57  140     192  0   0
9   56  140     294  0   0
```

```
10  56  130     256  1   2
11  44  120     263  0   0
12  52  172     199  1   0
13  57  150     168  0   0
14  54  140     239  0   0
15  48  130     275  0   0
16  49  130     266  0   0
17  64  110     211  0   0
18  58  150     283  1   0
19  50  120     219  0   0
20  58  120     340  0   0
21  66  150     226  0   0
22  43  150     247  0   0
23  69  140     239  0   0
24  60  117     230  1   2
25  59  135     234  0   0
26  44  130     233  0   0
27  42  140     226  0   0
28  61  150     243  1   0
29  40  140     199  0   0
30  71  160     302  0   0
⋮  ⋮   ⋮      ⋮   ⋮   ⋮
171 76  140     197  0   0
172 70  156     245  0   0
173 44  118     242  0   0
174 60  150     240  0   0
175 44  120     226  0   0
176 42  136     315  0   2
177 52  128     204  1   2
178 59  126     218  1   2
179 42  130     180  0   0
180 66  160     228  0   0
181 46  140     311  0   2
182 71  112     149  0   0
183 64  170     227  0   0
184 66  146     278  0   0
185 39  138     220  0   0
186 58  130     197  0   0
187 57  110     335  0   2
188 47  130     253  0   0
189 35  122     192  0   0
190 61  148     203  0   2
191 58  170     225  1   2
192 58  125     220  0   0
193 56  130     221  0   0
194 56  120     240  0   0
195 55  132     342  0   0
196 44  120     169  0   2
197 63  140     187  0   2
198 41  120     157  0   0
```

```
199 68  144     193  1   2
200 38  138     175  0   0
```

```
  num count
1 0   164
2 2    36
```

```
    age trestbp chol fbs num
1   63  145     233  1   0
2   67  120     229  0   1
3   37  130     250  0   0
4   41  130     204  0   0
5   56  120     236  0   0
6   57  120     354  0   0
7   53  140     203  1   1
8   57  140     192  0   0
9   56  140     294  0   0
10  44  120     263  0   0
11  52  172     199  1   0
12  57  150     168  0   0
13  48  110     229  0   1
14  54  140     239  0   0
15  48  130     275  0   0
16  49  130     266  0   0
17  64  110     211  0   0
18  58  150     283  1   0
19  58  120     284  0   1
20  50  120     219  0   0
21  58  120     340  0   0
22  66  150     226  0   0
23  43  150     247  0   0
24  69  140     239  0   0
25  64  140     335  0   1
26  59  135     234  0   0
27  44  130     233  0   0
28  42  140     226  0   0
29  57  150     276  0   1
30  61  150     243  1   0
⋮  ⋮   ⋮      ⋮   ⋮   ⋮
190 76  140     197  0   0
191 70  156     245  0   0
192 57  124     261  0   1
193 44  118     242  0   0
194 60  150     240  0   0
195 44  120     226  0   0
196 40  152     223  0   1
197 42  130     180  0   0
198 61  140     207  0   1
199 66  160     228  0   0
200 71  112     149  0   0
201 59  134     204  0   1
```

```
202 64  170     227  0   0
203 66  146     278  0   0
204 39  138     220  0   0
205 57  154     232  0   1
206 58  130     197  0   0
207 47  130     253  0   0
208 35  122     192  0   0
209 58  125     220  0   0
210 56  130     221  0   0
211 56  120     240  0   0
212 67  152     212  0   1
213 55  132     342  0   0
214 63  124     197  0   1
215 41  120     157  0   0
216 57  140     241  0   1
217 45  110     264  0   1
218 57  130     236  0   1
219 38  138     175  0   0
```

```
  num count
1 0   164
2 1    55
```

This data is still imbalanced, therefore we will try to combine the numbers 1-4 as yes heart disease and have the number 0 be no heart disease. We will do this through making a new column denoting named, "heart disease", we will then assign numbers 1-4 as yes and number 0 as no.

```
#combine 1-4 as disease and 0 as no disease
disease<- c(1:4)
no_disease<- c(0)
#check balance
cleveland_yes_no<-cleveland|> select(age, trestbp, chol, fbs, num)|>
mutate(heart_disease= if_else(num>0, "no disease", "disease"))|>
mutate(heart_disease=as_factor(heart_disease))
cleveland_yes_no
#balance check
cleveland_balancecheck_yes_no<- cleveland_yes_no|>
group_by(heart_disease)|> summarize(count=n())
cleveland_balancecheck_yes_no
```

```
    age trestbp chol fbs num heart_disease
1   63  145     233  1   0   disease
2   67  160     286  0   2   no disease
3   67  120     229  0   1   no disease
4   37  130     250  0   0   disease
5   41  130     204  0   0   disease
6   56  120     236  0   0   disease
7   62  140     268  0   3   no disease
8   57  120     354  0   0   disease
9   63  130     254  0   2   no disease
```

```
10  53  140     203  1   1   no disease
11  57  140     192  0   0   disease
12  56  140     294  0   0   disease
13  56  130     256  1   2   no disease
14  44  120     263  0   0   disease
15  52  172     199  1   0   disease
16  57  150     168  0   0   disease
17  48  110     229  0   1   no disease
18  54  140     239  0   0   disease
19  48  130     275  0   0   disease
20  49  130     266  0   0   disease
21  64  110     211  0   0   disease
22  58  150     283  1   0   disease
23  58  120     284  0   1   no disease
24  58  132     224  0   3   no disease
25  60  130     206  0   4   no disease
26  50  120     219  0   0   disease
27  58  120     340  0   0   disease
28  66  150     226  0   0   disease
29  43  150     247  0   0   disease
30  40  110     167  0   3   no disease
⋮   ⋮   ⋮      ⋮    ⋮  ⋮  ⋮
274 71  112     149  0   0   disease
275 59  134     204  0   1   no disease
276 64  170     227  0   0   disease
277 66  146     278  0   0   disease
278 39  138     220  0   0   disease
279 57  154     232  0   1   no disease
280 58  130     197  0   0   disease
281 57  110     335  0   2   no disease
282 47  130     253  0   0   disease
283 55  128     205  0   3   no disease
284 35  122     192  0   0   disease
285 61  148     203  0   2   no disease
286 58  114     318  0   4   no disease
287 58  170     225  1   2   no disease
288 58  125     220  0   0   disease
289 56  130     221  0   0   disease
290 56  120     240  0   0   disease
291 67  152     212  0   1   no disease
292 55  132     342  0   0   disease
293 44  120     169  0   2   no disease
294 63  140     187  0   2   no disease
295 63  124     197  0   1   no disease
296 41  120     157  0   0   disease
297 59  164     176  1   3   no disease
298 57  140     241  0   1   no disease
299 45  110     264  0   1   no disease
300 68  144     193  1   2   no disease
301 57  130     131  0   3   no disease
```

```
302 57  130     236  0   1   no disease
303 38  138     175  0   0   disease
```

```
  heart_disease count
1 disease       164
2 no disease    139
```

As is seen from above, this dataset is much more balanced not with only 25 observations different between one another. Therefore from now on we will use this grouping as our dataset to build our classifier.

```
cleveland<-cleveland_yes_no
cleveland
```

```
    age trestbp chol fbs num heart_disease
1   63  145     233  1   0   disease
2   67  160     286  0   2   no disease
3   67  120     229  0   1   no disease
4   37  130     250  0   0   disease
5   41  130     204  0   0   disease
6   56  120     236  0   0   disease
7   62  140     268  0   3   no disease
8   57  120     354  0   0   disease
9   63  130     254  0   2   no disease
10  53  140     203  1   1   no disease
11  57  140     192  0   0   disease
12  56  140     294  0   0   disease
13  56  130     256  1   2   no disease
14  44  120     263  0   0   disease
15  52  172     199  1   0   disease
16  57  150     168  0   0   disease
17  48  110     229  0   1   no disease
18  54  140     239  0   0   disease
19  48  130     275  0   0   disease
20  49  130     266  0   0   disease
21  64  110     211  0   0   disease
22  58  150     283  1   0   disease
23  58  120     284  0   1   no disease
24  58  132     224  0   3   no disease
25  60  130     206  0   4   no disease
26  50  120     219  0   0   disease
27  58  120     340  0   0   disease
28  66  150     226  0   0   disease
29  43  150     247  0   0   disease
30  40  110     167  0   3   no disease
⋮   ⋮   ⋮      ⋮    ⋮   ⋮   ⋮
274 71  112     149  0   0   disease
275 59  134     204  0   1   no disease
276 64  170     227  0   0   disease
277 66  146     278  0   0   disease
278 39  138     220  0   0   disease
```

```
279 57  154     232  0   1   no disease
280 58  130     197  0   0   disease
281 57  110     335  0   2   no disease
282 47  130     253  0   0   disease
283 55  128     205  0   3   no disease
284 35  122     192  0   0   disease
285 61  148     203  0   2   no disease
286 58  114     318  0   4   no disease
287 58  170     225  1   2   no disease
288 58  125     220  0   0   disease
289 56  130     221  0   0   disease
290 56  120     240  0   0   disease
291 67  152     212  0   1   no disease
292 55  132     342  0   0   disease
293 44  120     169  0   2   no disease
294 63  140     187  0   2   no disease
295 63  124     197  0   1   no disease
296 41  120     157  0   0   disease
297 59  164     176  1   3   no disease
298 57  140     241  0   1   no disease
299 45  110     264  0   1   no disease
300 68  144     193  1   2   no disease
301 57  130     131  0   3   no disease
302 57  130     236  0   1   no disease
303 38  138     175  0   0   disease
```

```r
#setting the seed
set.seed(1)
#created training(75%) and testing data
cleveland_split<- initial_split(cleveland, prop=0.75, strata=
heart_disease)
cleveland_train<- training(cleveland_split)
cleveland_test<- testing(cleveland_split)

cleveland_train
```

```
    age trestbp chol fbs num heart_disease
1   63  145     233  1   0   disease
4   37  130     250  0   0   disease
5   41  130     204  0   0   disease
11  57  140     192  0   0   disease
12  56  140     294  0   0   disease
16  57  150     168  0   0   disease
20  49  130     266  0   0   disease
21  64  110     211  0   0   disease
22  58  150     283  1   0   disease
27  58  120     340  0   0   disease
28  66  150     226  0   0   disease
29  43  150     247  0   0   disease
31  69  140     239  0   0   disease
34  59  135     234  0   0   disease
```

```
35  44  130     233  0   0   disease
36  42  140     226  0   0   disease
40  61  150     243  1   0   disease
42  40  140     199  0   0   disease
43  71  160     302  0   0   disease
44  59  150     212  1   0   disease
47  51  110     175  0   0   disease
49  65  140     417  1   0   disease
51  41  105     198  0   0   disease
52  65  120     177  0   0   disease
54  44  130     219  0   0   disease
59  54  125     273  0   0   disease
60  51  125     213  0   0   disease
64  54  135     304  1   0   disease
71  65  155     269  0   0   disease
76  65  160     360  0   0   disease
⋮  ⋮   ⋮      ⋮    ⋮  ⋮   ⋮
224 53  123     282  0   3   no disease
225 63  108     269  0   1   no disease
229 54  110     206  0   3   no disease
230 66  112     212  0   2   no disease
232 55  180     327  0   2   no disease
236 54  122     286  0   3   no disease
237 56  130     283  1   2   no disease
244 61  134     234  0   2   no disease
246 67  120     237  0   2   no disease
247 58  100     234  0   2   no disease
248 47  110     275  0   1   no disease
252 58  146     218  0   1   no disease
260 57  124     261  0   1   no disease
262 58  136     319  1   3   no disease
265 61  138     166  0   4   no disease
266 42  136     315  0   2   no disease
268 59  126     218  1   2   no disease
269 40  152     223  0   1   no disease
271 61  140     207  0   1   no disease
273 46  140     311  0   2   no disease
279 57  154     232  0   1   no disease
281 57  110     335  0   2   no disease
283 55  128     205  0   3   no disease
285 61  148     203  0   2   no disease
291 67  152     212  0   1   no disease
294 63  140     187  0   2   no disease
297 59  164     176  1   3   no disease
298 57  140     241  0   1   no disease
300 68  144     193  1   2   no disease
301 57  130     131  0   3   no disease
```

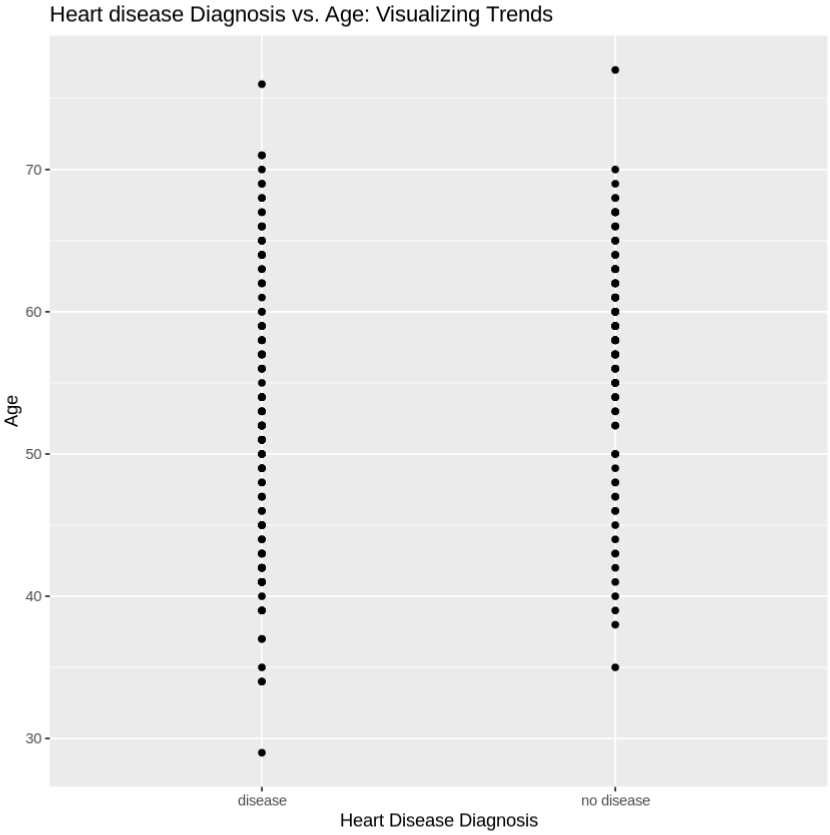
Below I will do some preliminary exploration and visualization of our variables in order to find answer our question, if well known risk factors are able to make an accurate classifier for heart disease.

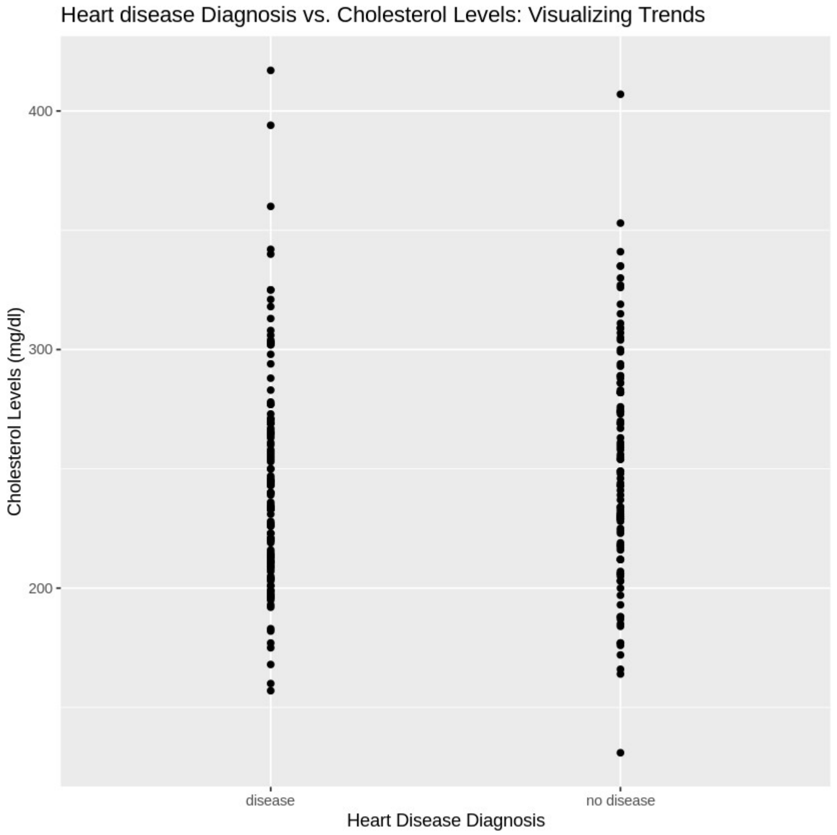
```
#table containing the average values of all predictors of each
severity of heart disease
average_predictors_diag<-cleveland_train|> group_by(heart_disease)|>
summarize(across(age:fbs, mean))
average_predictors_diag

max_predictors_diag<-cleveland_train|> group_by(heart_disease)|>
summarize(across(age:fbs, max))
max_predictors_diag

min_predictors_diag<-cleveland_train|> group_by(heart_disease)|>
summarize(across(age:fbs, min))
min_predictors_diag
```

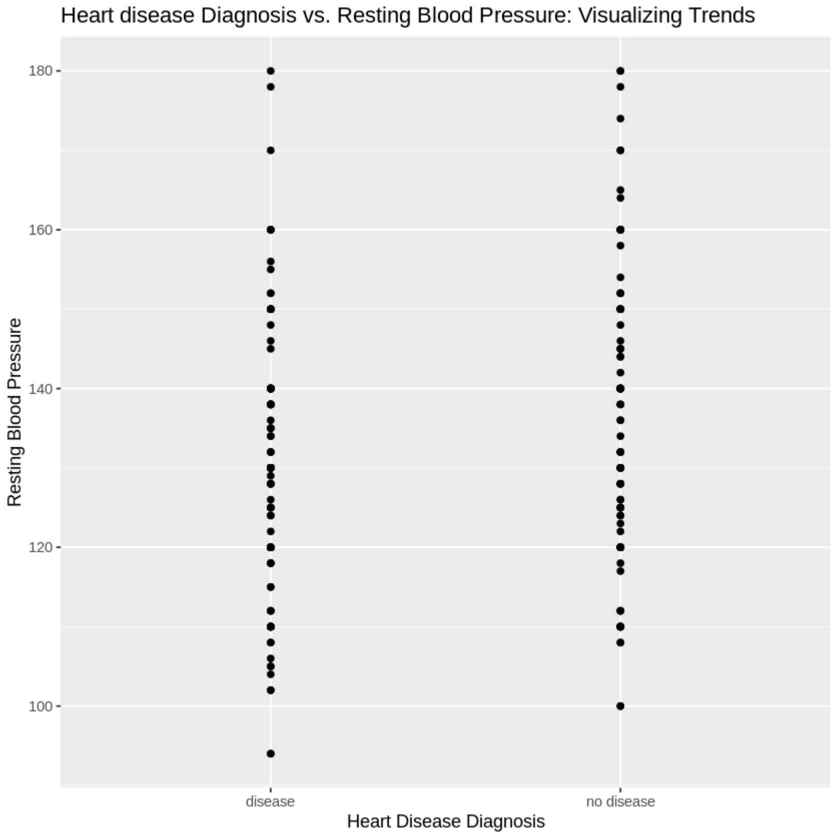
```
  heart_disease age      trestbp  chol     fbs
1 disease       52.96748 130.4553 243.9106 0.1382114
2 no disease    57.23077 135.5096 250.5673 0.1538462

  heart_disease age trestbp chol fbs
1 disease       76  180     417  1
2 no disease    77  180     407  1

  heart_disease age trestbp chol fbs
1 disease       29   94     157  0
2 no disease    35  100     131  0
```

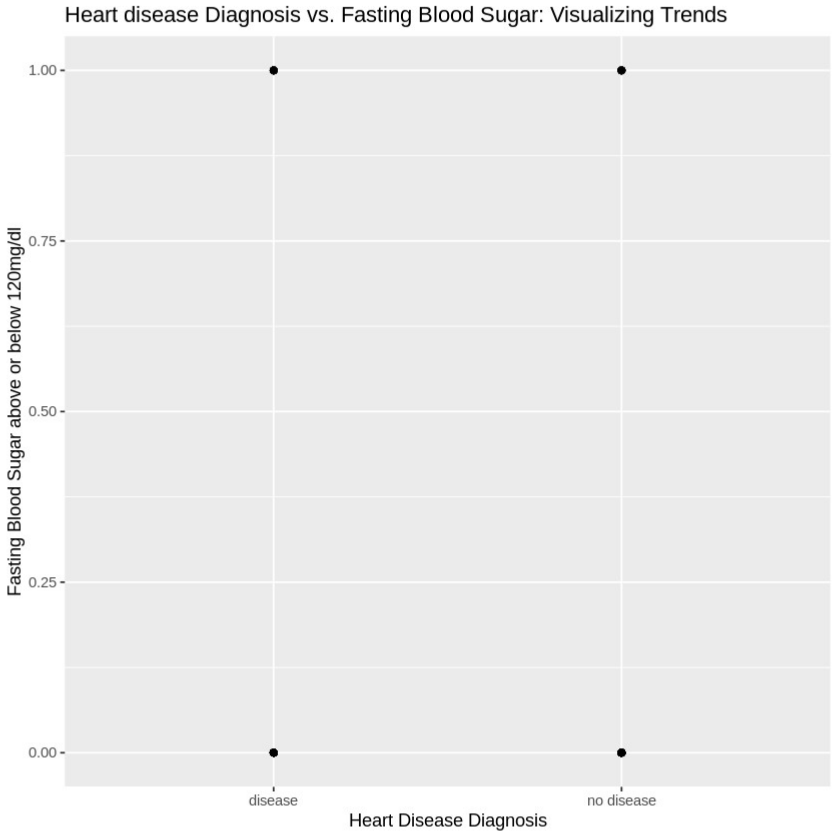
```
#graphs showing trends in predictor variables segregated by each class
of heart disease or not.
cleveland_agenum_trend <- cleveland_train|>
ggplot(aes(x=heart_disease, y=age))+geom_point()+labs(x=  "Heart
Disease Diagnosis", y="Age")+ ggtitle("Heart disease Diagnosis vs.
Age: Visualizing Trends")
cleveland_agenum_trend

cleveland_cholnum_trend <- cleveland_train|>
ggplot(aes(x=heart_disease, y=chol))+geom_point()+labs(x= "Heart
Disease Diagnosis", y="Cholesterol Levels (mg/dl)")+ ggtitle("Heart
disease Diagnosis vs. Cholesterol Levels: Visualizing Trends")
cleveland_cholnum_trend

cleveland_trestbpnum_trend <- cleveland_train|>
ggplot(aes(x=heart_disease, y=trestbp))+geom_point()+labs(x= "Heart
Disease Diagnosis", y="Resting Blood Pressure")+ ggtitle("Heart
disease Diagnosis vs. Resting Blood Pressure: Visualizing Trends")
cleveland_trestbpnum_trend

cleveland_fbsnum_trend <- cleveland_train|>
```

```
ggplot(aes(x=heart_disease, y=fbs))+geom_point()+labs(x= "Heart
Disease Diagnosis", y="Fasting Blood Sugar above or below 120mg/dl")+
ggtitle("Heart disease Diagnosis vs. Fasting Blood Sugar: Visualizing
Trends")
cleveland_fbsnum_trend
```




Heart disease Diagnosis vs. Cholesterol Levels: Visualizing Trends
400
300
200
Cholesterol Levels (mg/dl)
disease
no disease
Heart Disease Diagnosis


Heart disease Diagnosis vs. Resting Blood Pressure: Visualizing Trends
180
160
140
120
100
Resting Blood Pressure
disease
no disease
Heart Disease Diagnosis



Since the disease and no disease plots of each predictor varible is look extremely similar, we can hypothesize that these well known risk factors for heart disease will not be great predictor variables for our heart disease classifier. Additionally, it is because the max, min and mean of all predictor variables between heart disease and non-diseased are extrememly similar. This could be because we have combined groups 1-4 into one group, while keeping only group 0 as another group. We will visualize the data based on the num column now, to ensure that this is not the case.

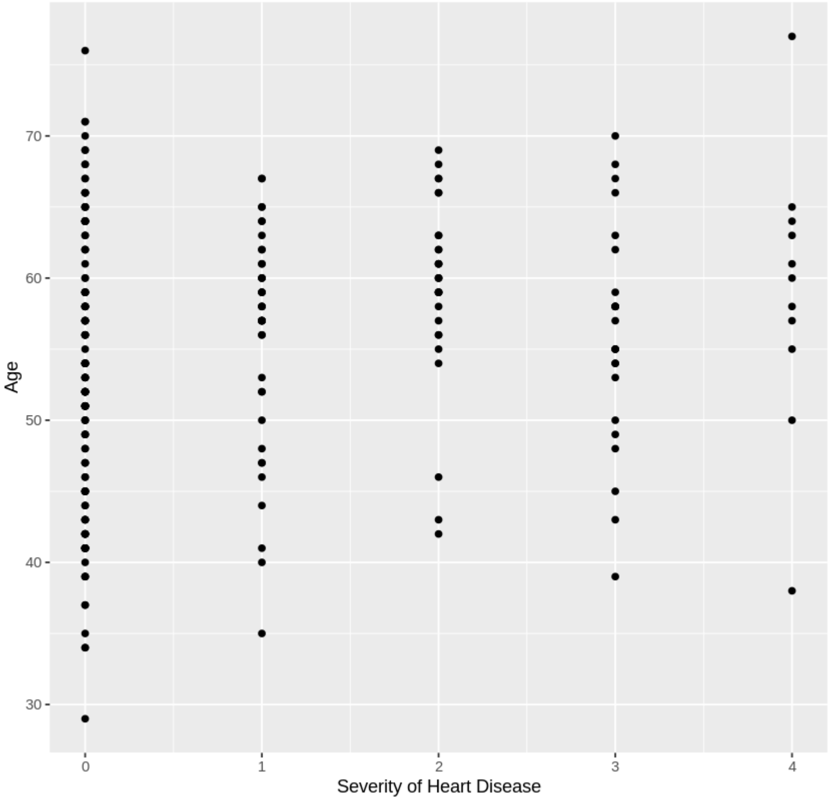
```
#table containing the average values of all predictors of each
severity of heart disease
average_predictors<-cleveland_train|> group_by(num)|>
summarize(across(age:fbs, mean))
average_predictors

max_predictors<-cleveland_train|> group_by(num)|>
summarize(across(age:fbs, max))
max_predictors

min_predictors<- cleveland_train|> group_by(num)|>
summarize(across(age:fbs, min))
min_predictors
```

```
  num age      trestbp  chol     fbs
1 0   52.96748 130.4553 243.9106 0.13821138
```

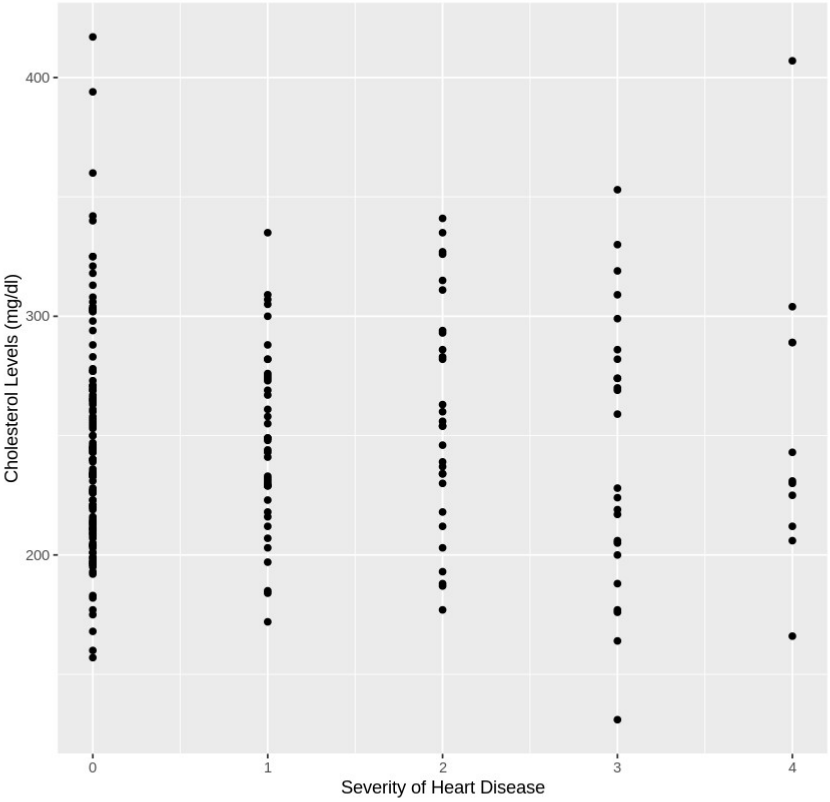
```
2 1   56.02500 134.3750 248.0000 0.07500000
3 2   59.27586 134.5517 257.8621 0.24137931
4 3   56.00000 136.2917 244.1250 0.20833333
5 4   58.90909 140.4545 254.7273 0.09090909

  num age trestbp chol fbs
1 0   76  180     417  1
2 1   67  174     335  1
3 2   69  180     341  1
4 3   70  180     353  1
5 4   77  165     407  1

  num age trestbp chol fbs
1 0   29   94     157  0
2 1   35  108     172  0
3 2   42  100     177  0
4 3   39  100     131  0
5 4   38  112     166  0
```

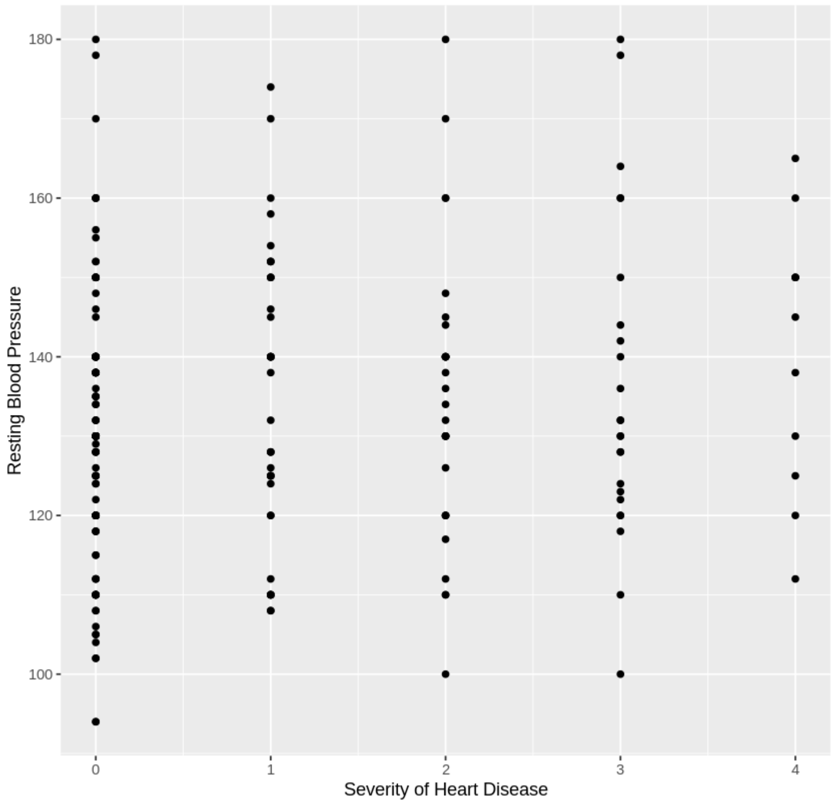
```
#graphs showing trends in predictor variables segregated by each class
of heart disease severity.
cleveland_agenum_trend <- cleveland_train|> ggplot(aes(x=num, y=age))
+geom_point()+labs(x= "Severity of Heart Disease", y="Age")+
ggtitle("Severity of Heart disease vs. Age: Visualizing Trends")
cleveland_agenum_trend

cleveland_cholnum_trend <- cleveland_train|> ggplot(aes(x=num,
y=chol))+geom_point()+labs(x= "Severity of Heart Disease",
y="Cholesterol Levels (mg/dl)")+ ggtitle("Severity of Heart disease
vs. Cholesterol Levels: Visualizing Trends")
cleveland_cholnum_trend

cleveland_trestbpnum_trend <- cleveland_train|> ggplot(aes(x=num,
y=trestbp))+geom_point()+labs(x= "Severity of Heart Disease",
y="Resting Blood Pressure")+ ggtitle("Severity of Heart disease vs.
Resting Blood Pressure: Visualizing Trends")
cleveland_trestbpnum_trend

cleveland_fbsnum_trend <- cleveland_train|> ggplot(aes(x=num, y=fbs))
+geom_point()+labs(x= "Severity of Heart Disease", y="Fasting Blood
Sugar above or below 120mg/dl")+ ggtitle("Severity of Heart disease
vs. Fasting Blood Sugar: Visualizing Trends")
cleveland_fbsnum_trend
```

Severity of Heart disease vs. Age: Visualizing Trends

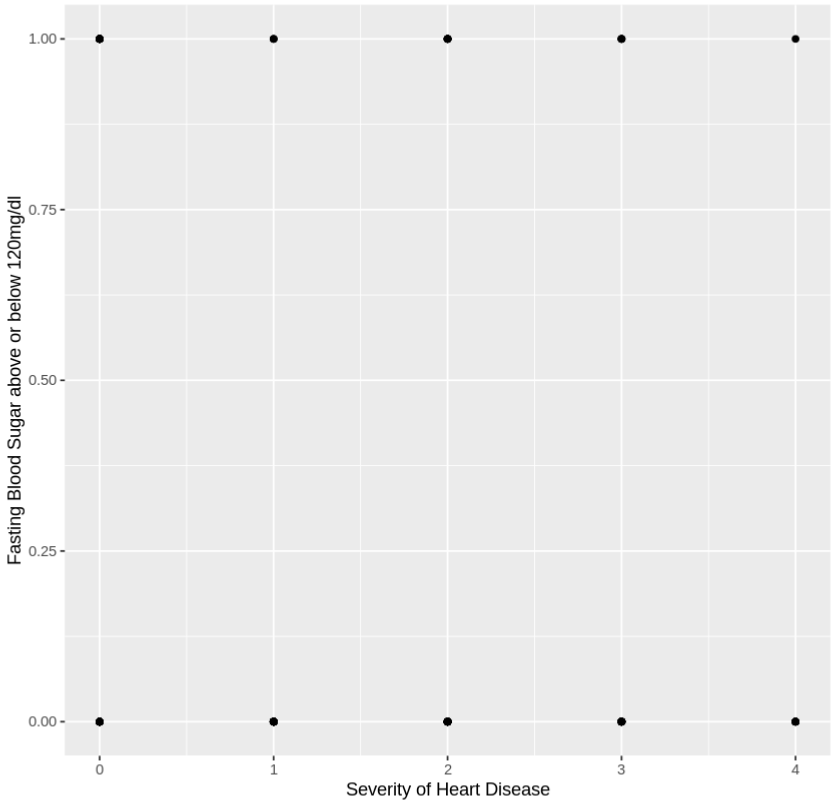


Severity of Heart disease vs. Cholesterol Levels: Visualizing Trends





Severity of Heart disease vs. Resting Blood Pressure: Visualizing Trends



Severity of Heart disease vs. Fasting Blood Sugar: Visualizing Trends

Since the 0:4 plots of each predictor varible is look extremely similar, we can hypothesize that these well known risk factors for heart disease will not be great predictor variables for our heart disease classifier. Additionally, it is because the max, min and mean of all predictor variables between heart disease and non-diseased are extrememly similar.

From this, we are likely to determine that these common risk factors are not good predictor variables of heart disease. We will proceed and build our classifier, to determine just how accurate these predictor variables could be. We will also proceed with the data that has groups 1-4 combined and group 0 as its own group as it has better balance.

```
#Splitting the data in order to perform a 5 fold cross-validation
cleveland_vfold<- vfold_cv(cleveland_train, v=5, strata=heart_disease)

#creating the recipe to do conduct the cross validation
cleveland_recipe<-
recipe(heart_disease~fbs+trestbp+chol+age,data=cleveland_train)|>
step_scale(all_predictors())|> step_center(all_predictors())

#creating the model
cleveland_spec<- nearest_neighbor(weight_func="rectangular",
neighbors=tune())|> set_engine("kknn")|> set_mode("classification")

#number of neighbours/k values to try
k_values<- tibble(neighbors=seq(from=5, to=20, by=1))

#adding them to a workflow
cleveland_workflow<- workflow()|> add_recipe(cleveland_recipe)|>
add_model(cleveland_spec)|> tune_grid(resample=cleveland_vfold,
grid=k_values)|> collect_metrics()

cleveland_metrics<- cleveland_workflow|> filter(.metric=="accuracy")|>
arrange(desc(mean))|> select(neighbors, mean)
cleveland_metrics
```

```
   neighbors mean
1  17        0.5637813
2  18        0.5637813
3  19        0.5594335
4  20        0.5594335
5  15        0.5551910
6  16        0.5551910
7   9        0.5461968
8  10        0.5461968
9  11        0.5417479
10 12        0.5417479
11 13        0.5154633
12 14        0.5154633
13  5        0.5066711
14  6        0.5066711
```

```
15  7        0.5065744
16  8        0.5065744
```

From the above cross validation, we can see that we should use K=17 as it has the highest accuracy. Below we will retain the model.

```
#create a new model and workflow
cleveland_specvalidated<- nearest_neighbor(weight_func="rectangular",
neighbors=17)|> set_engine("kknn")|> set_mode("classification")

cleveland_workflowvalidated<- workflow()|>
add_recipe(cleveland_recipe)|> add_model(cleveland_specvalidated)|>
fit(data=cleveland_train)

#test the model on the testing set

cleveland_predict<- predict(cleveland_workflowvalidated,
cleveland_test)|> bind_cols(cleveland_test)

#collect prediction metrics
prediction_metrics<- cleveland_predict|> metrics(truth=heart_disease,
estimate=.pred_class)|> filter(.metric=="accuracy")|>
select(.estimate)
prediction_metrics

#make a confusion matrix for a visual representation of the accuracy
of the model
cleveland_confmatrix<- cleveland_predict|>
conf_mat(truth=heart_disease, estimate=.pred_class)
print(cleveland_confmatrix)
```

```
  .estimate
1 0.6578947
```

```
            Truth
Prediction   disease no disease
  disease         34         19
  no disease       7         16
```

Our classifier is 66% accurate, when using our test set to compute the accuracy of the retrained model

From looking at the conufsion matrix show above, we can say that our classifier is not very accurate, as it only identified 26 observations wrong out of a total of 76 predictions attempted. This is an accuracy of 0.6578947, which was computed in the varibles, prediction_metrics. Below we will now create some new observations and predict whether these patients have heart disease.

```
#creating 7 random observations in a tibble format for the classifier
to predict
newage<- c(44, 66,50,80,20,16,80)
```

```
newfbs<- c(0,1,1,0,1,0,1)
newtrestbp<- c(120, 50,60,70, 30,66,99)
newchol<- c(100,200, 150,250,300,239,167)
new_obs<- tibble(age=newage, fbs=newfbs, trestbp=newtrestbp,
chol=newchol)

randomobs_prediction<- predict(cleveland_workflowvalidated, new_obs)
randomobs_prediction
```

```
  .pred_class
1 disease
2 no disease
3 disease
4 no disease
5 disease
6 disease
7 disease
```

According to our classifier the majority of the 7 new observatiions will have heart disease, while only two will not be diseased.