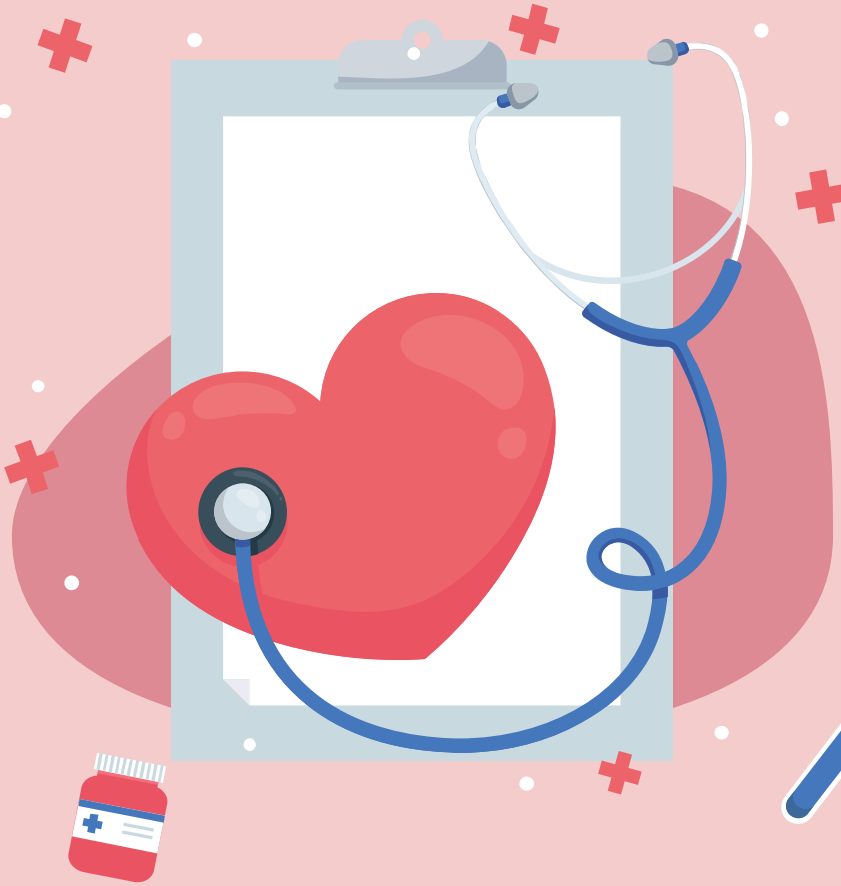
CARDIOVASCULAR DISEASE PREDICTION

Team Member: Ankit Jain, Darshana Daga ,Emily Han, Mira Daya



PROJECT OVERVIEW

Project Scope

- Identifying important factors in determining presence of Cardiovascular Disease (CVD)
- Creating models to classify presence of Cardiovascular disease

*"Coronary **heart disease** is the most common type of **heart disease**, killing **365,914** people in 2017. About 18.2 million adults age 20 and older **have** CAD (about 6.7%). About 2 in 10 deaths from CAD happen in adults less than 65 years old."* - CDC

Project Value

- Provide recommendations to avoid CVD
- Identifying at-risk patients to take early precautionary measures
- Even just 1% correctly classified CVD has potential to save over 30,000 lives.

DATA

CHOLESTEROL

AGE

HEIGHT

BLOOD PRESSURE (DIASTOLIC AND SYSTOLIC)

SMOKING

GLUCOSE

ALCOHOL

ACTIVE

GENDER

WEIGHT

DATA

id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	50	0	168	136.6864	110	80	1	1	0	0	1	0

SOURCE: Kaggle
DATA SIZE: 70000 rows x 13 columns

1. **Age**: Quantitative Variable: (in days)
2. **Height**: Quantitative Variable: (in cm)
3. **Weight**: Quantitative Variable (in kg)
4. **Gender**: Categorical Variable (1 = Female, 2 = Male)
5. **Systolic Blood Pressure**: Quantitative Variable (in mmHG)
6. **Diastolic Blood Pressure**: Quantitative Variable (in mmHG)
7. **Cholesterol**: Categorical Variable (1: Normal 2: Above Normal, 3: Well Above Normal)
8. **Glucose**: Categorical Variable (1: Normal 2: Above Normal, 3: Well Above Normal)
9. **Smoking**: Binary Variable (0 = Doesn't Smoke, 1 = Smokes)
10. **Alcohol**: Binary Variable (0 = Doesn't Drink, 1 = Drinks)
11. **Active**: Binary Variable (0 = Not Active, 1 = Active)
12. **Cardio** (Target): Binary Variable (0 = doesn't have CVD, 1 = have CVD)

DATA CLEANING

General:

- Age, days to years
- Weight, Kg to Lb
- Gender, 1,2 to 0,1
- Factorize Cholesterol, Glucose, Cardio, Smoke, Alcohol, Active

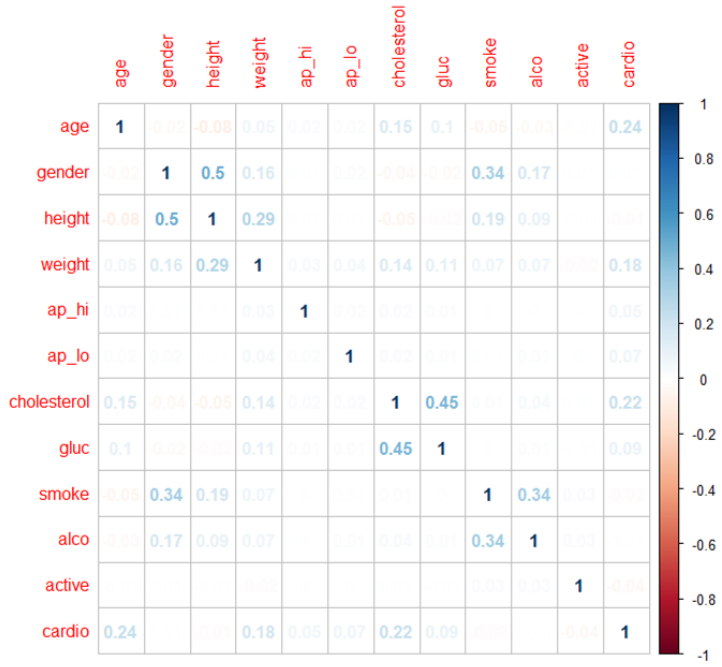
Outliers:

- Diastolic & Systolic
- Height

EDA

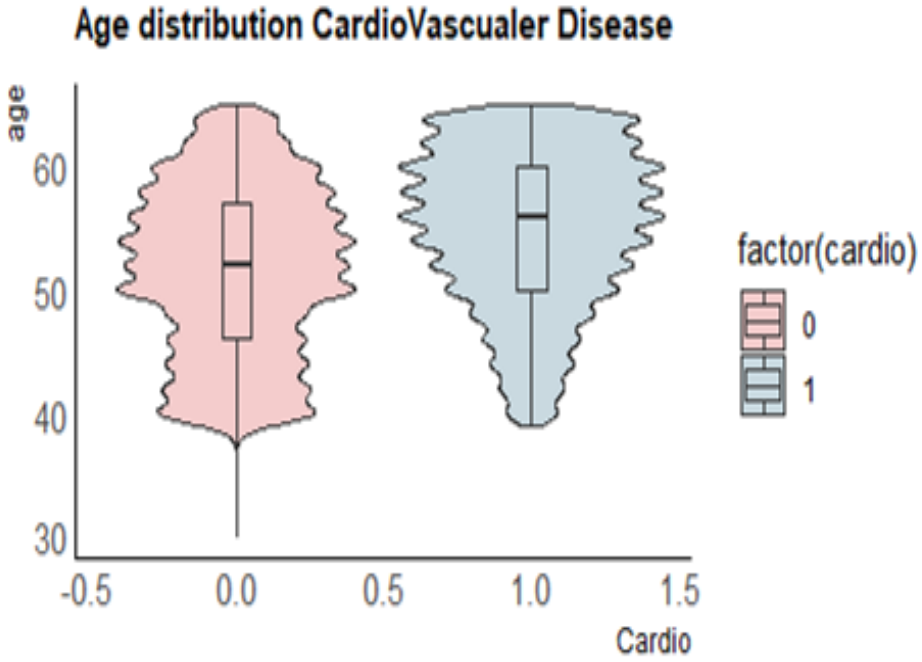
Correlation Plot

Given the importance of the outcome variable, we kept a stricter threshold of 0.15 for our analysis.

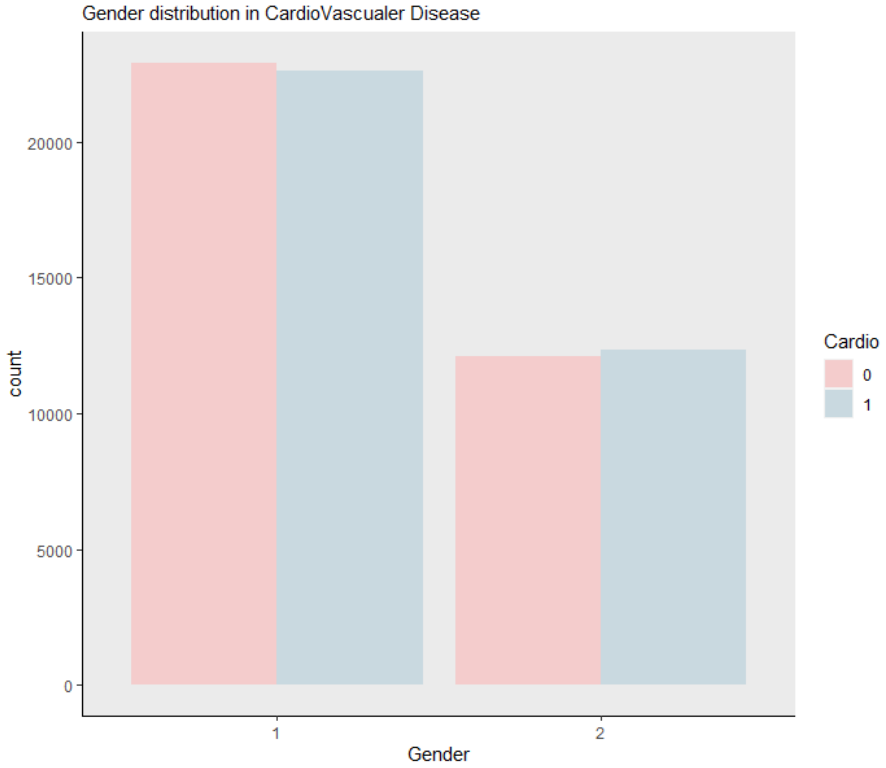


EDA

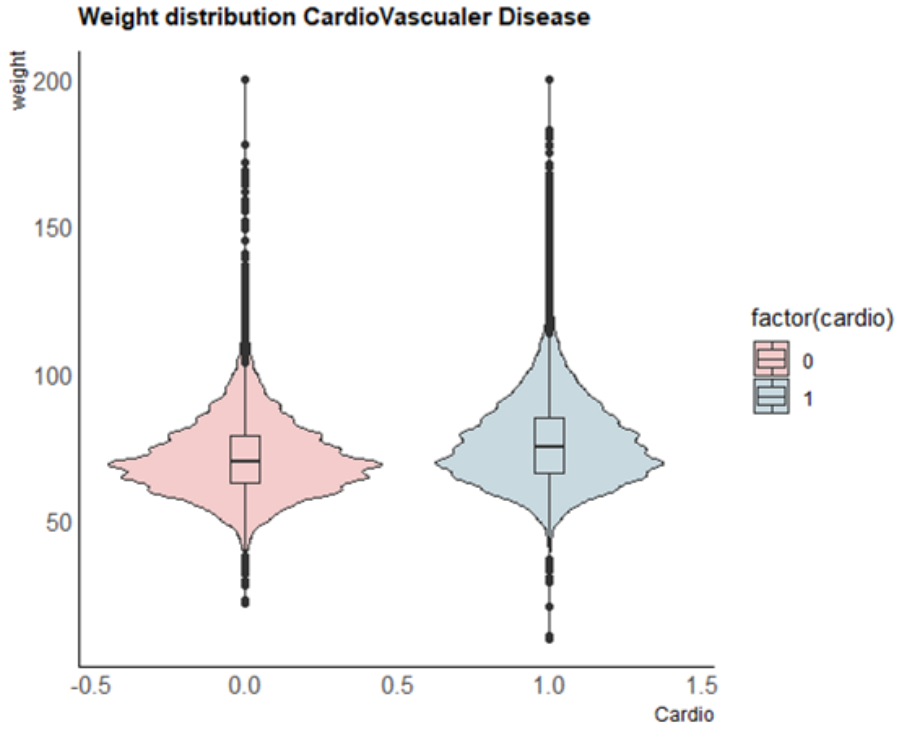
Age Plot


Age distribution CardioVascualer Disease
age
60
50
40
30
-0.5
0.0
0.5
1.0
1.5
Cardio
factor(cardio)
0
1

EDA
Gender Plot

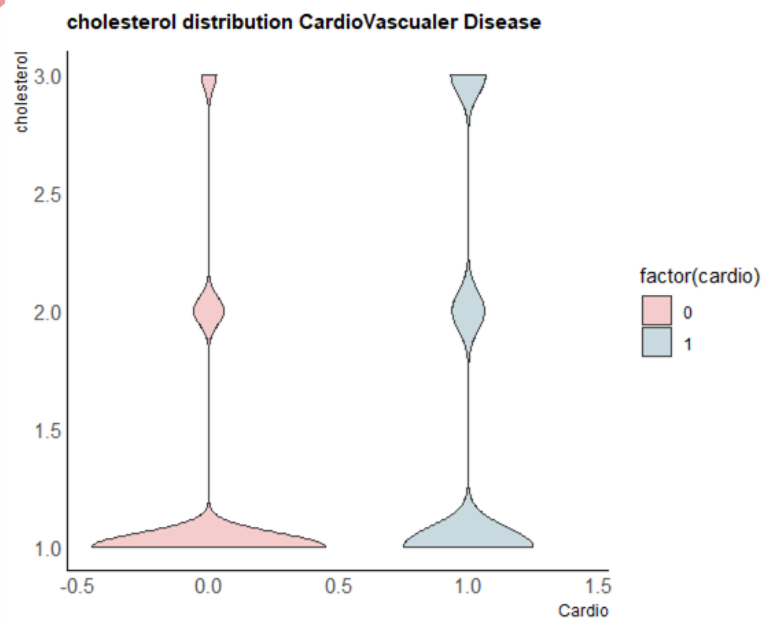


EDA Weight Plot



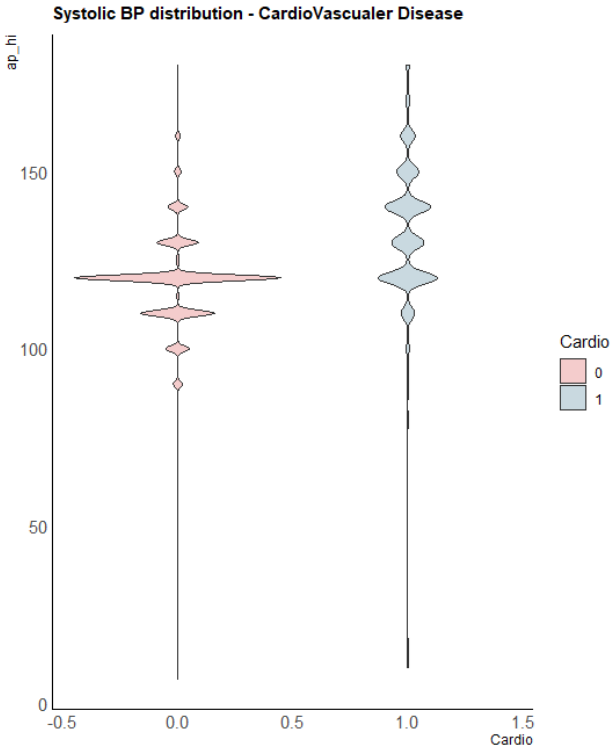
EDA

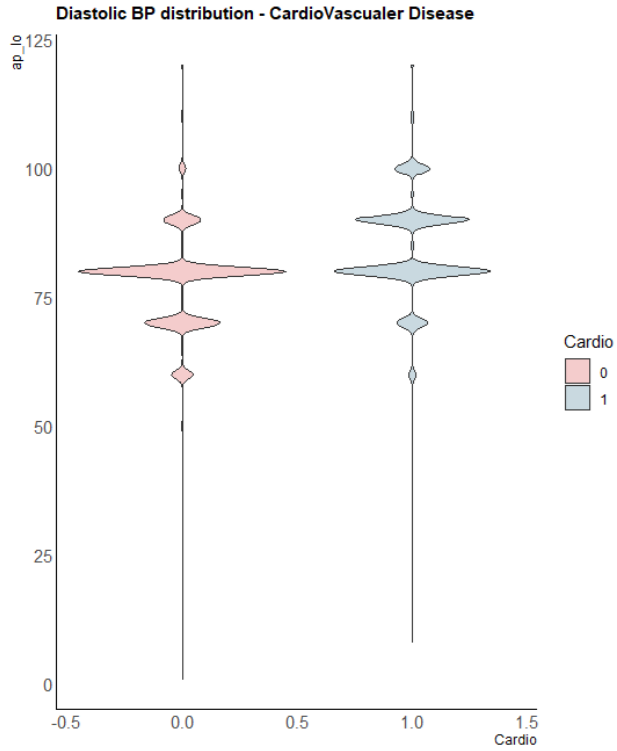
Cholesterol Plot



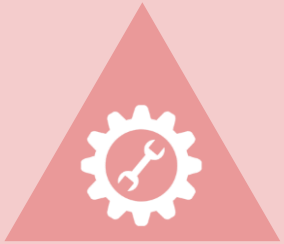
EDA

Blood Pressure





APPROACH



BUILD A MODEL TO DETECT CVD

CHOOSE VARIABLES FOR INSIGHTS

UNDERSTAND IMPLICATIONS

Logistic Regression

Determine significant variables (p-values < 0.05)
- E.g. **age**, **smoking status**, **ap_hi**, **ap_lo**, etc.

Add interaction variables
- E.g. **ap_hi*age, ap_lo*age**, etc.

Test different classification thresholds
- **0.4** threshold generates a reasonable **precision** while **maintaining a relatively high overall accuracy**

Accuracy: 0.7039

Cutoff Threshold	**Overall Accuracy**	**Precision**
0.35	0.6784	0.8657
0.4	**0.7039**	**0.8198**
0.5	0.7270	0.7054

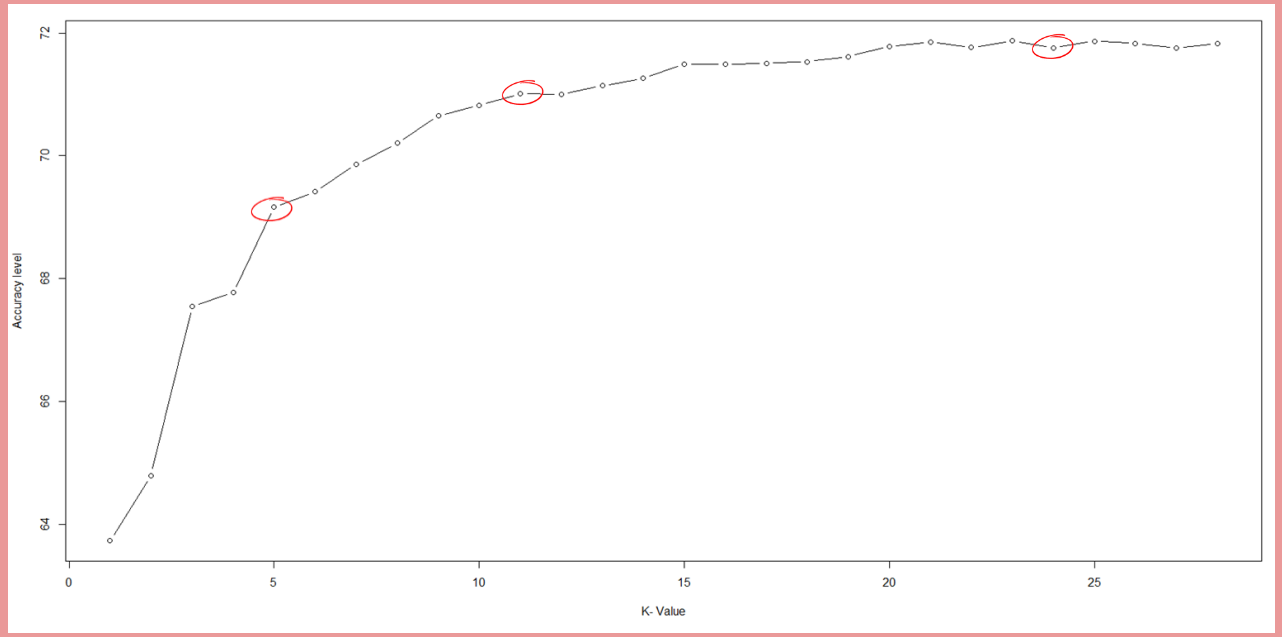
cardio = (gluc*cholesterol) + (age*cholesterol) + (age*gluc) + (ap_hi*cholesterol) + (ap_lo*cholesterol) + (ap_hi*age) + (ap_lo*age) + (ap_hi*gluc) + (ap_lo*gluc)

KNN

Use the **elbow method** to find what k value generates the highest accuracy

The best k = 11

- Small K: noise have higher influence
- Large K: expensive to compute
- The change in precision is higher when increasing k from 5 to 11 than the change in precision when increasing k from 11 to 23



K Value	**Overall Accuracy**	**Precision**
5	0.6908	0.6613
11	**0.7102**	**0.6720**
23	0.7188	0.6737

DECISION TREES

R-packages:

- **Rpart, Tree**
  - Similar uses, rpart has more flexibility in parameters
  - employ **information criteria** for selecting the current covariate
- **Party**
  - **ctree() < conditional inference trees**
  - uses **significance tests** in order to:
    - **select variables** instead of selecting the variable that maximizes an information measure (e.g. Gini coefficient)
    - predictors are **only included if** the predictor is **significant**
    - Better to determine **true effect of predictor**

DECISION TREES

	MODEL	ACCURACY	PRECISION
1	***Rpart depth = 10***	***0.7253858***	***0.6690127***
2	Dtrees depth = 1	0.7092748	0.6076246
3	Ctree depth = 10	0.7224346	0.6770283
4	***Ctree depth = 5***	***0.7244182***	***0.6246334***
5	***Ctree depth = 4***	***0.7226765***	***0.7136852***
6	Ctree depth = 2	0.7092748	0.6076246

DECISION TREES

		Actual	
		Doesn't Have CVD	Has CVD
Predicted	Doesn't Have CVD	8149	2290
	Has CVD	3386	6844

Rpart w/ depth = 10

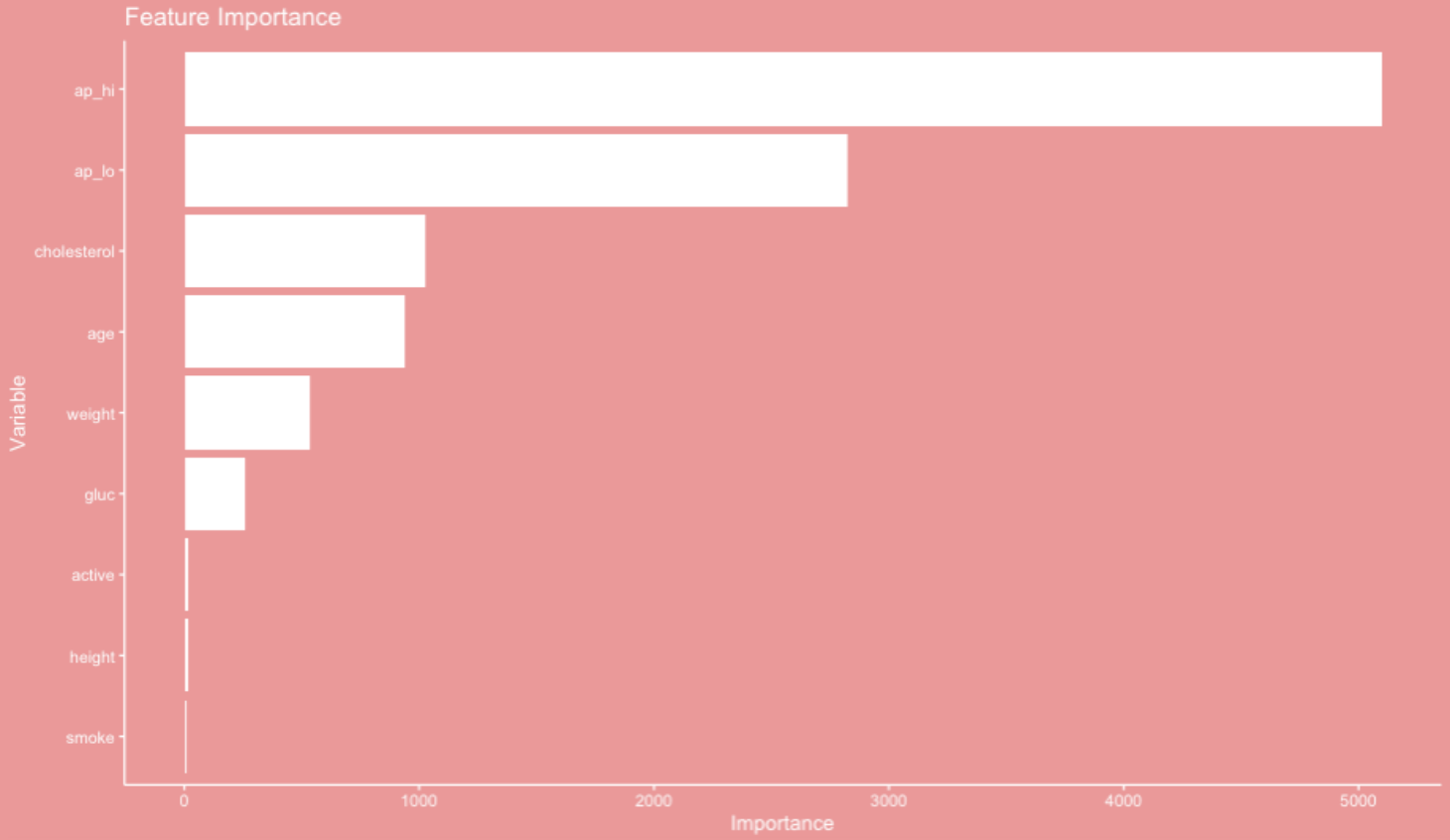
		Actual	
		Doesn't Have CVD	Has CVD
Predicted	Doesn't Have CVD	8583	1856
	Has CVD	3840	6390

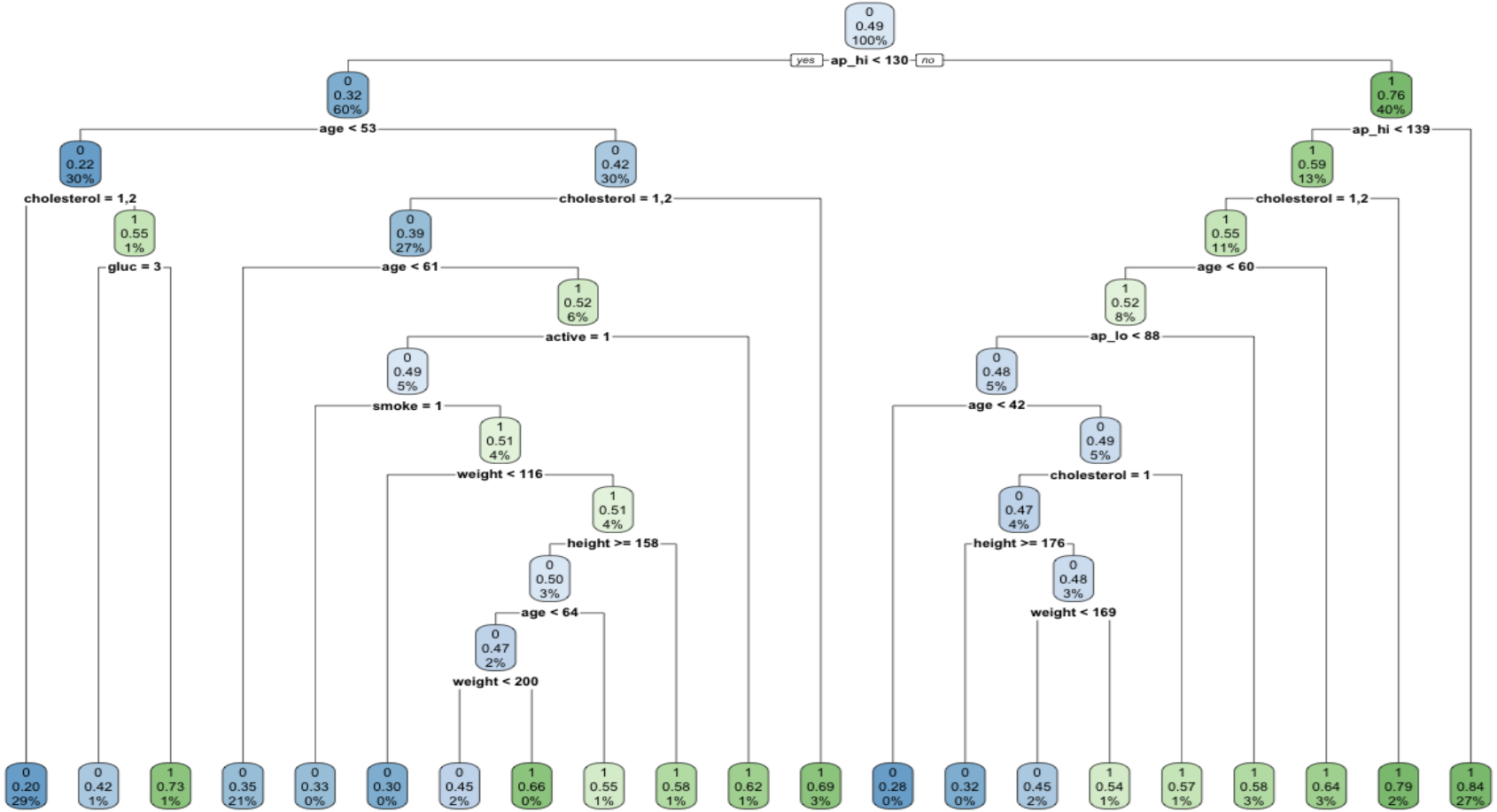
cTree w/ depth = 5

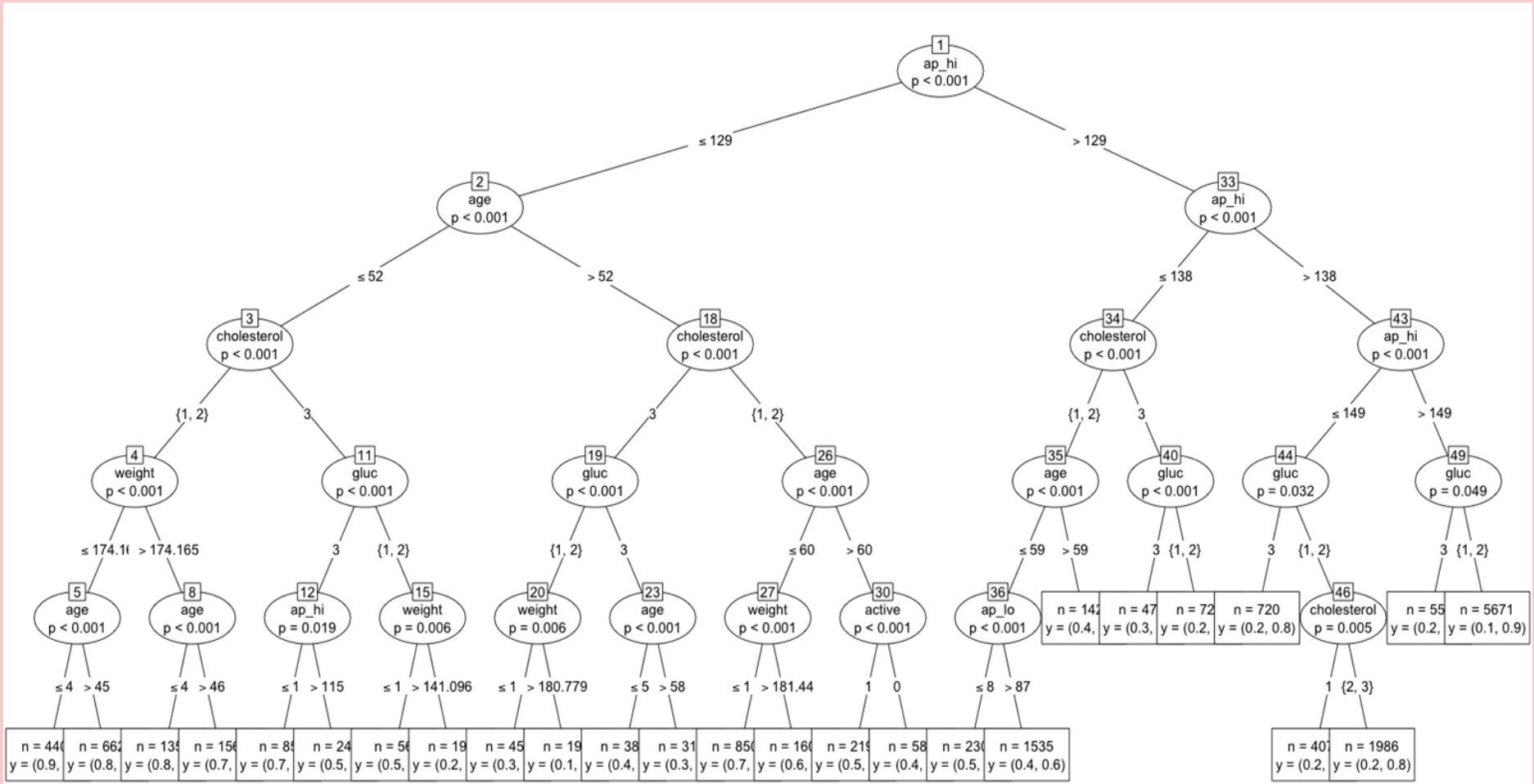
		Actual	
		Doesn't Have CVD	Has CVD
Predicted	Doesn't Have CVD	7636	2803
	Has CVD	2929	7301

cTree w/ depth = 4

DECISION TREES: FEATURE IMPORTANCE




0
0.49
100%
yes
ap_hi < 130
no
0
0.32
60%
1
0.76
40%
age < 53
ap_hi < 139
0
0.22
30%
0
0.42
30%
1
0.59
13%
cholesterol = 1,2
cholesterol = 1,2
cholesterol = 1,2
1
0.55
1%
0
0.39
27%
1
0.55
11%
gluc = 3
age < 61
age < 60
1
0.52
6%
1
0.52
8%
active = 1
ap_lo < 88
0
0.49
5%
0
0.48
5%
smoke = 1
age < 42
1
0.51
4%
0
0.49
5%
weight < 116
cholesterol = 1
1
0.51
4%
0
0.47
4%
height >= 158
height >= 176
0
0.50
3%
0
0.48
3%
age < 64
weight < 169
0
0.47
2%
weight < 200
0
0.20
29%
0
0.42
1%
1
0.73
1%
0
0.35
21%
0
0.33
0%
0
0.30
0%
0
0.45
2%
1
0.66
0%
1
0.55
1%
1
0.58
1%
1
0.62
1%
1
0.69
3%
0
0.28
0%
0
0.32
0%
0
0.45
2%
1
0.54
1%
1
0.57
1%
1
0.58
3%
1
0.64
3%
1
0.79
2%
1
0.84
27%


1
ap_hi
p < 0.001
≤ 129
> 129
2
age
p < 0.001
33
ap_hi
p < 0.001
≤ 52
> 52
≤ 138
> 138
3
cholesterol
p < 0.001
18
cholesterol
p < 0.001
34
cholesterol
p < 0.001
43
ap_hi
p < 0.001
{1, 2}
3
{1, 2}
3
{1, 2}
3
≤ 149
> 149
4
weight
p < 0.001
11
gluc
p < 0.001
19
gluc
p < 0.001
26
age
p < 0.001
35
age
p < 0.001
40
gluc
p < 0.001
44
gluc
p = 0.032
49
gluc
p = 0.049
≤ 174.1
> 174.165
3
{1, 2}
{1, 2}
3
≤ 60
> 60
≤ 59
> 59
3
{1, 2}
3
{1, 2}
3
{1, 2}
5
age
p < 0.001
8
age
p < 0.001
12
ap_hi
p = 0.019
15
weight
p = 0.006
20
weight
p = 0.006
23
age
p < 0.001
27
weight
p < 0.001
30
active
p < 0.001
36
ap_lo
p < 0.001
n = 142
y = (0.4,
n = 47
y = (0.3,
n = 72
y = (0.2,
n = 720
y = (0.2, 0.8)
46
cholesterol
p = 0.005
n = 55
y = (0.2,
n = 5671
y = (0.1, 0.9)
≤ 4
> 45
≤ 4
> 46
≤ 1
> 115
≤ 1
> 141.096
≤ 1
> 180.779
≤ 5
> 58
≤ 1
> 181.44
1
0
≤ 8
> 87
1
{2, 3}
n = 44
y = (0.9,
n = 662
y = (0.8,
n = 135
y = (0.8,
n = 156
y = (0.7,
n = 85
y = (0.7,
n = 24
y = (0.5,
n = 56
y = (0.5,
n = 19
y = (0.2,
n = 45
y = (0.3,
n = 19
y = (0.1,
n = 38
y = (0.4,
n = 31
y = (0.3,
n = 850
y = (0.7,
n = 160
y = (0.6,
n = 219
y = (0.5,
n = 58
y = (0.4,
n = 230
y = (0.5,
n = 1535
y = (0.4, 0.6)
n = 401
y = (0.2,
n = 1986
y = (0.2, 0.8)

SUPPORT VECTOR MACHINE MODEL

- Variations with different parameter values (11 Cases)
- Cross Validation on data sample for SVM Parameters (Kernel, Cost, Gamma)

Best Model:

- Kernel: Radial
- Cost: 1
- Gamma: 0.1
- Training Accuracy: **74.13%**
- Testing Accuracy: **73.28%**
- Precision Accuracy: **67.97%**

```
svm(formula = cardio ~ ., data = train, kernel = "radial", cost = 1, gamma = 0.1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  28779

 ( 14562 14217 )


Number of Classes:  2
```

SVM Model		Actual	
		Doesn't Have CVD	**Has CVD**
Predicted	**Doesn't Have CVD**	8194	2245
	Has CVD	3277	6953

TAKEAWAYS AND LEARNINGS

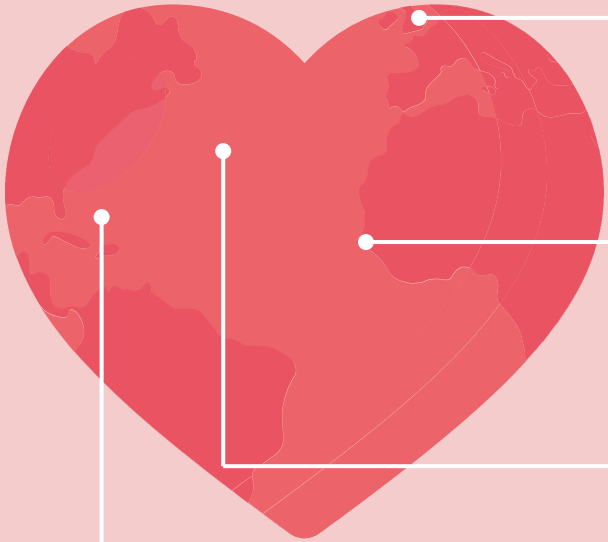
At-Risk Patients

- Age > 50
- Weight > 165 lbs
- Cholestrol > Normal
- Blood Pressure > 130/90 mmHg
- Smoking = Yes
- Alcohol = Yes

TAKEAWAYS AND LEARNINGS

Model/Accuracy	Logistic Regression	K-Nearest Neighbours	Decision Tree	Support Vector Machine Model
Testing Accuracy	70.39%	71.02%	72.50%	73.28%
Precision Accuracy	81.98%	67.20%	66.90%	67.97%

TAKEAWAYS AND LEARNINGS



1. Cholesterol and Glucose Level are signs to look out for in CVD Detection (Logistic Regression)

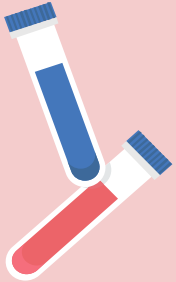
2. Reducing Alcohol and Smoking can have benefits of avoiding CVD.

3. Best Model for classifying CVD is the Logistic Regression Model based on the current dataset because of the best precision

4. People at age 35-50, having borderline cholesterol, glucose and blood pressure should receive CVD awareness.

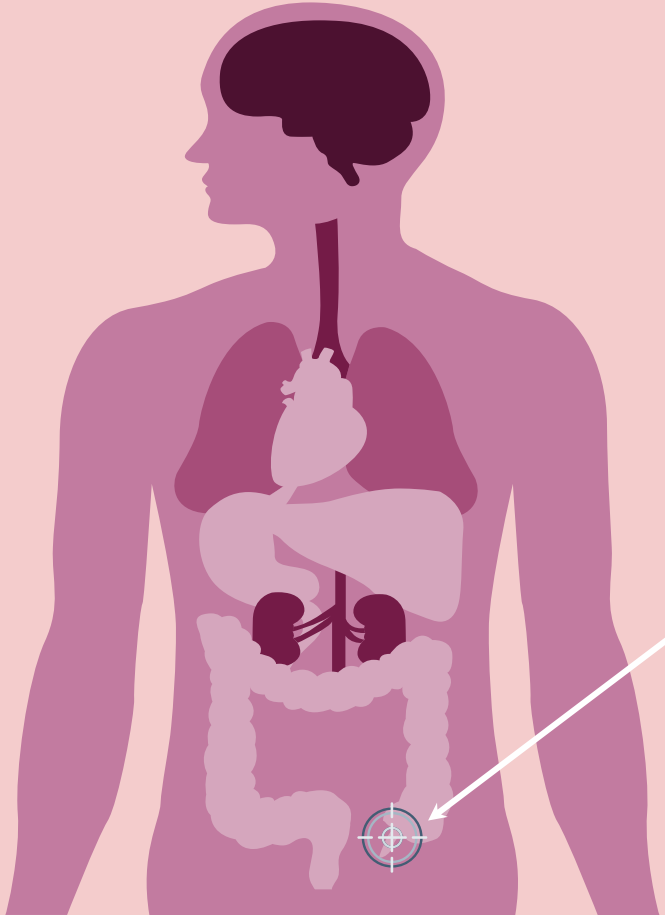
CHALLENGES

1. More patient information is required
2. The bias in the data needs to addressed.
3. The overall accuracy is not good enough for model selection



THANK YOU

Q & A



APPENDIX

For the appendix, we have added more details about certain parts of our presentation we could not go into detail, and also included some parts we didn't go over earlier.

The rest of our steps are present in our code.

Data

Data Dictionary

```
> str(dat)
'data.frame':   70000 obs. of  13 variables:
 $ id          : int  0 1 2 3 4 8 9 12 13 14 ...
 $ age         : int  18393 20228 18857 17623 17474 21914 22113 22584 17668 19834 ...
 $ gender      : int  2 1 1 2 1 1 1 2 1 1 ...
 $ height      : int  168 156 165 169 156 151 157 178 158 164 ...
 $ weight      : num  62 85 64 82 56 67 93 95 71 68 ...
 $ ap_hi       : int  110 140 130 150 100 120 130 130 110 110 ...
 $ ap_lo       : int  80 90 70 100 60 80 80 90 70 60 ...
 $ cholesterol: int  1 3 3 1 1 2 3 3 1 1 ...
 $ gluc        : int  1 1 1 1 1 2 1 3 1 1 ...
 $ smoke       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ alco        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ active      : int  1 1 0 1 0 0 1 1 1 0 ...
 $ cardio      : int  0 1 1 1 0 0 0 1 0 0 ...
```

Cleaned

```
> str(dat2)
'data.frame':   68899 obs. of  13 variables:
 $ id          : int  0 1 2 3 4 8 9 12 13 14 ...
 $ age         : num  50 55 52 48 48 60 61 62 48 54 ...
 $ gender      : num  0 1 1 0 1 1 1 0 1 1 ...
 $ height      : int  168 156 165 169 156 151 157 178 158 164 ...
 $ weight      : num  137 187 141 181 123 ...
 $ ap_hi       : int  110 140 130 150 100 120 130 130 110 110 ...
 $ ap_lo       : int  80 90 70 100 60 80 80 90 70 60 ...
 $ cholesterol: Factor w/ 3 levels "1","2","3": 1 3 3 1 1 2 3 3 1 1 ...
 $ gluc        : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 3 1 1 ...
 $ smoke       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ alco        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ active      : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
 $ cardio      : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 1 1 ...
```

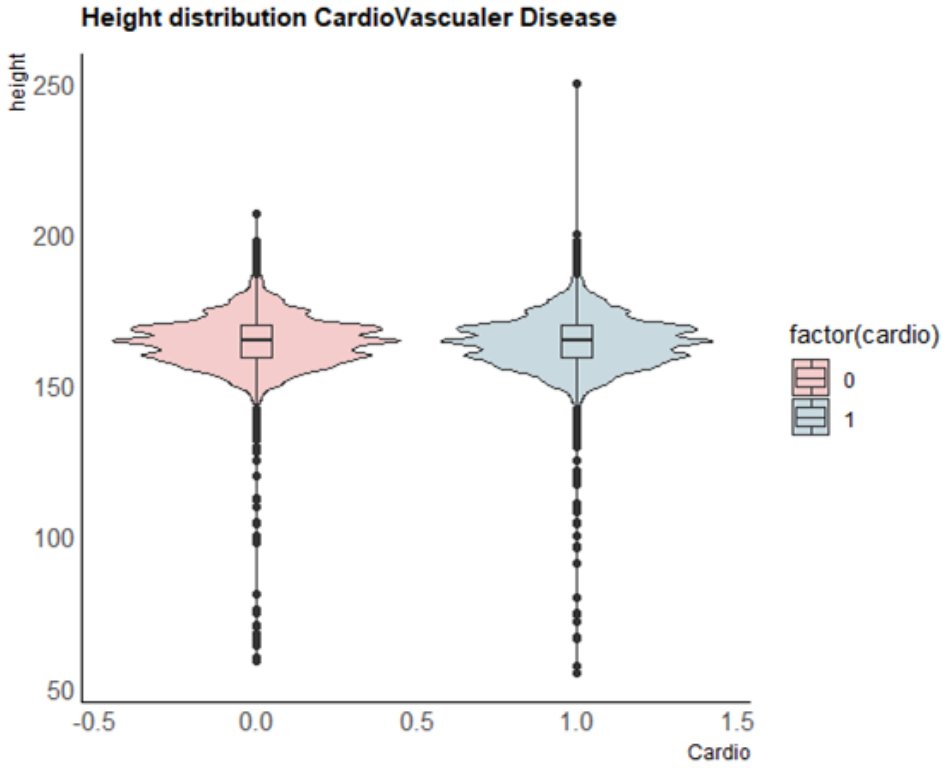
Data Dictionary

1. **Age**: Quantitative Variable: (in days)
2. **Height**: Quantitative Variable: (in cm)
3. **Weight**: Quantitative Variable (in kg)
4. **Gender**: Categorical Variable (1 = Female, 2 = Male)
5. **Ap_hi = Systolic Blood Pressure**: Quantitative Variable (in mmHG)
6. **Ap_lo = Diastolic Blood Pressure**: Quantitative Variable (in mmHG)
7. **Cholesterol**: Categorical Variable (1: Normal 2: Above Normal, 3: Well Above Normal)
8. **Gluc = Glucose**: Categorical Variable (1: Normal 2: Above Normal, 3: Well Above Normal)
9. **Smoking**: Binary Variable (0 = Doesn't Smoke, 1 = Smokes)
10. **Alco = Alcohol**: Binary Variable (0 = Doesn't Drink, 1 = Drinks)
11. **Active**: Binary Variable (0 = Not Active, 1 = Active) - whether a patient is a
12. **Cardio** (Target): Binary Variable (0 = doesn't have CVD, 1 = have CVD)

EDA

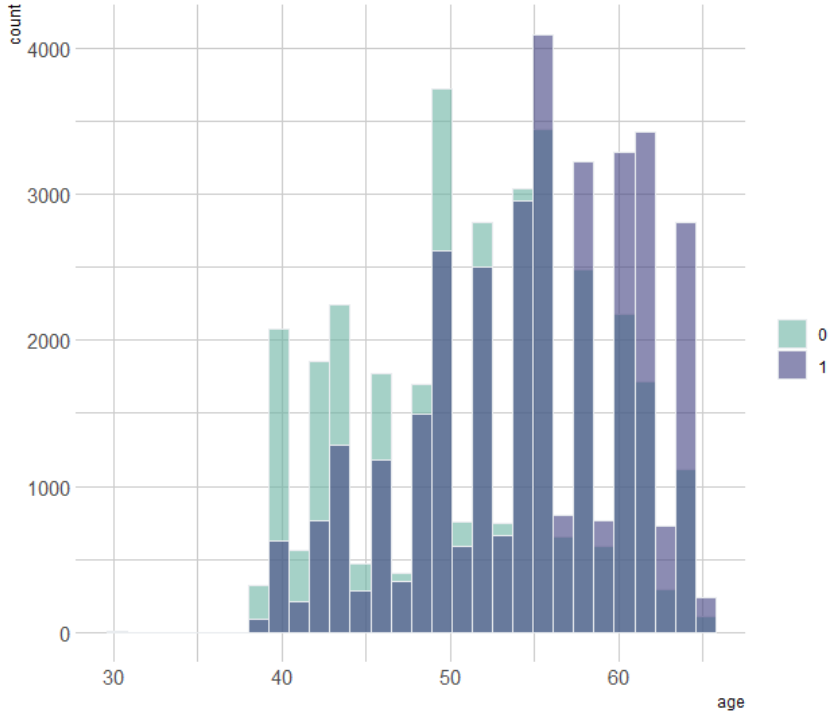
EDA

Height Plot



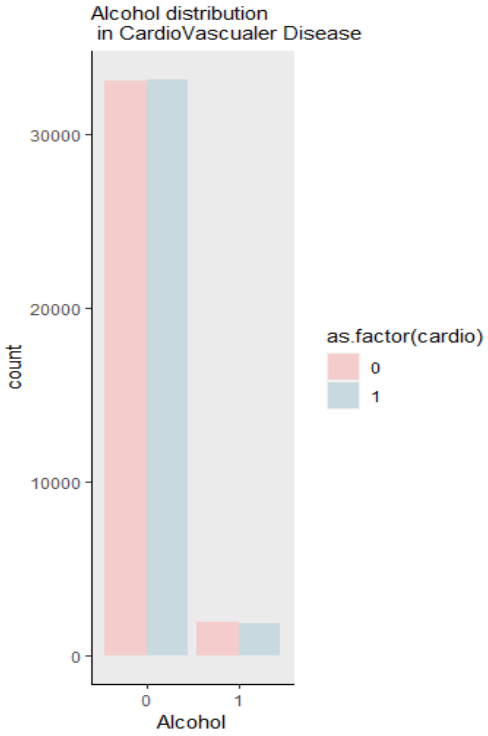
EDA

Age Distribution Plot



EDA

Smoking & Alcohol Plot



DATA CLEANING

General:
The first step was to transform variables and factorize the categorical variables

- Age, days to years
- Weight, Kg to Lb
- Gender, 1,2 to 0,1
- Factorize Cholesterol, Glucose, Cardio, Smoke, Alcohol, Active

Outliers:
Then based on domain acumen and visualization insights remove the outlier values.

- Diastolic & Systolic
- Height

Even though weight had outlier we didn't remove it as its important to understand is obesity causes CVD.

Logistic Regression

Logistic Regression

We ran a initial general linear regression to understand the behaviour and significance of the variables.

glm(cardio ~ . - id, data = dat.train, family = "binomial")

Find out variable “age” has the p-value less than 0.05

Add interaction variables because there are some connections between independent variables (e.g. the blood pressure goes up as people get older)

```
> sum.lr.all$coefficients[sum.lr.all$coefficients[,4]<0.05,]
                 Estimate   Std. Error    z value      Pr(>|z|)
(Intercept) -10.397106105 0.3091365514 -33.632730 5.577122e-248
age           0.052042796 0.0016066570  32.391976 3.560385e-230
height       -0.005168496 0.0016327890  -3.165440  1.548485e-03
weight        0.012005429 0.0008218277  14.608207  2.489861e-48
ap_hi         0.046258282 0.0009862834  46.901613  0.000000e+00
ap_lo         0.022201272 0.0015497737  14.325493  1.516477e-46
cholesterol2  0.397175650 0.0323336103  12.283678  1.108424e-34
cholesterol3  1.100198961 0.0425153827  25.877668 1.188268e-147
gluc3        -0.379404780 0.0469465934  -8.081625  6.390918e-16
smoke1       -0.120644227 0.0411939999  -2.928684  3.403998e-03
alco1        -0.211954932 0.0494946696  -4.282379  1.849057e-05
active1      -0.246382301 0.0259421515  -9.497373  2.152509e-21
```

```
Call:
glm(formula = cardio ~ . - id, family = "binomial", data = dat.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9424  -0.9252  -0.3275   0.9380   3.5343

Coefficients:
                Estimate  Std. Error z value             Pr(>|z|)
(Intercept)  -10.5132592   0.3101984 -33.892 < 0.0000000000000002 ***
age            0.0538499   0.0016182  33.277 < 0.0000000000000002 ***
gender        -0.0086982   0.0264962  -0.328             0.742701
height        -0.0058800   0.0016324  -3.602             0.000316 ***
weight         0.0124022   0.0008282  14.975 < 0.0000000000000002 ***
ap_hi          0.0467203   0.0009858  47.393 < 0.0000000000000002 ***
ap_lo          0.0229173   0.0015430  14.852 < 0.0000000000000002 ***
cholesterol2   0.3868906   0.0324303  11.930 < 0.0000000000000002 ***
cholesterol3   1.1476008   0.0429126  26.743 < 0.0000000000000002 ***
gluc2          0.0629580   0.0432690   1.455             0.145659
gluc3         -0.3849717   0.0473512  -8.130 0.000000000000000429 ***
smoke1        -0.1234577   0.0414586  -2.978             0.002903 **
alco1         -0.2168735   0.0504987  -4.295 0.000017497832174652 ***
active1       -0.2394363   0.0260776  -9.182 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66856  on 48229  degrees of freedom
Residual deviance: 54235  on 48216  degrees of freedom
AIC: 54263

Number of Fisher Scoring iterations: 4
```

Logistic Regression

glm(cardio ~ . - id - gender + (gluc*cholesterol)+ (age*cholesterol) + (age*gluc) + (ap_hi*cholesterol)+ (ap_lo*cholesterol) + (ap_hi*age) + (ap_lo*age)+ (ap_hi*gluc) + (ap_lo*gluc) , data = dat.train, family = "binomial")

Age: each additional year of the age increases the odds of having the CVD by 1.4%

Based on our understanding of CVD and research from different source, we found that age, cholesterol, glucose levels and blood pressure have big impact on having CVD. We also understand that a combination of any of these health issues can have much more effect on the possibility of having CVD. Like a person having cholesterol level 3 and have higher glucose level have increased chances of cardiac problems.

```
Call:
glm(formula = cardio ~ . - id - gender + (gluc * cholesterol) +
    (age * cholesterol) + (age * gluc) + (ap_hi * cholesterol) +
    (ap_lo * cholesterol) + (ap_hi * age) + (ap_lo * age) + (ap_hi *
    gluc) + (ap_lo * gluc), family = "binomial", data = dat.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6144  -0.9249  -0.1975   0.8953   4.4169

Coefficients:
                     Estimate  Std. Error z value             Pr(>|z|)
(Intercept)       -27.1266487   0.9993819 -27.143 < 0.0000000000000002 ***
age                 0.3477807   0.0178827  19.448 < 0.0000000000000002 ***
height             -0.0060980   0.0014573  -4.184   0.0000286031884607 ***
weight              0.0117858   0.0008357  14.103 < 0.0000000000000002 ***
ap_hi               0.1610968   0.0085608  18.818 < 0.0000000000000002 ***
ap_lo               0.0484976   0.0133122   3.643             0.000269 ***
cholesterol2        2.7502346   0.3969595   6.928   0.0000000000042608 ***
cholesterol3        7.4821598   0.5588687  13.388 < 0.0000000000000002 ***
gluc2               2.9009192   0.4893719   5.928   0.0000000030694175 ***
gluc3              -0.5843902   0.6253790  -0.934             0.350068
smoke1             -0.1309331   0.0404980  -3.233             0.001225 **
alco1              -0.2154128   0.0506147  -4.256   0.0000208179695740 ***
active1            -0.2425715   0.0263575  -9.203 < 0.0000000000000002 ***
cholesterol2:gluc2 -0.3235554   0.0877198  -3.689             0.000226 ***
cholesterol3:gluc2 -0.2523802   0.1667261  -1.514             0.130092
cholesterol2:gluc3 -0.2527995   0.1638896  -1.542             0.122952
cholesterol3:gluc3 -0.7058400   0.1006642  -7.012   0.0000000000023523 ***
age:cholesterol2   -0.0254508   0.0050159  -5.074   0.0000003894701088 ***
age:cholesterol3   -0.0175519   0.0071321  -2.461             0.013856 *
age:gluc2          -0.0179071   0.0064648  -2.770             0.005607 **
age:gluc3           0.0017655   0.0079012   0.223             0.823192
ap_hi:cholesterol2 -0.0049744   0.0027985  -1.778             0.075483 .
ap_hi:cholesterol3 -0.0265305   0.0034449  -7.701   0.0000000000000135 ***
ap_lo:cholesterol2 -0.0039465   0.0045957  -0.859             0.390484
ap_lo:cholesterol3 -0.0212456   0.0057832  -3.674             0.000239 ***
age:ap_hi          -0.0020025   0.0001553 -12.893 < 0.0000000000000002 ***
age:ap_lo          -0.0004647   0.0002439  -1.905             0.056718 .
ap_hi:gluc2        -0.0096800   0.0035101  -2.758             0.005821 **
ap_hi:gluc3        -0.0039694   0.0038127  -1.041             0.297825
ap_lo:gluc2        -0.0057116   0.0058150  -0.982             0.325989
ap_lo:gluc3         0.0113218   0.0063136   1.793             0.072933 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66856  on 48229  degrees of freedom
Residual deviance: 53474  on 48199  degrees of freedom
AIC: 53536

Number of Fisher Scoring iterations: 5
```

KNN

KNN

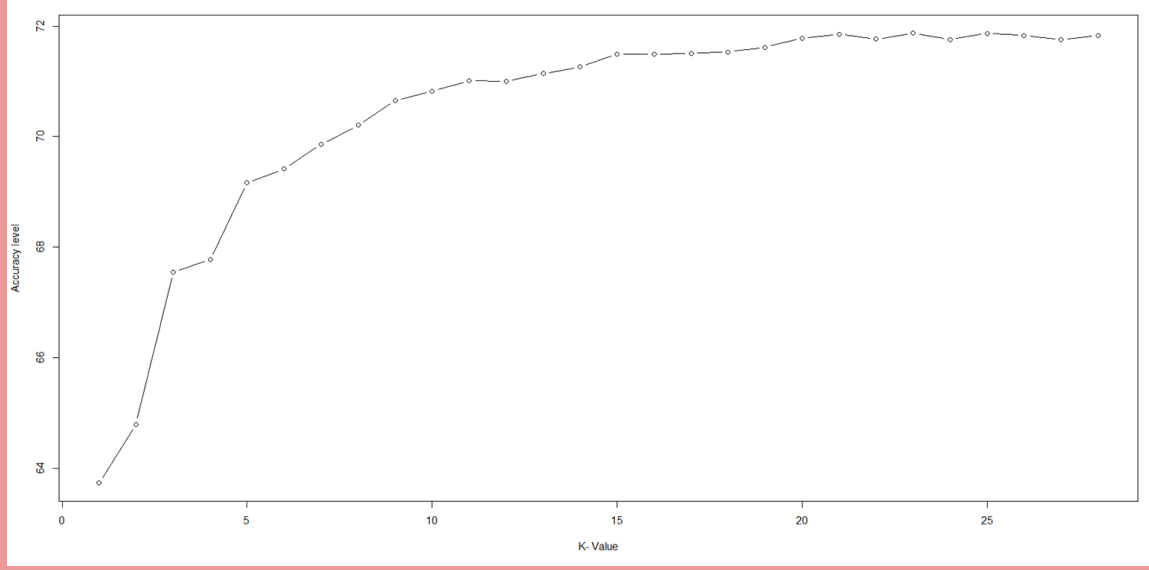
Generate the elbow plot to find which k value generates the highest overall accuracy

Try k = 5, 11, and 23 to check how the precision changes

- 5: benchmark to compare how the accuracy changes
- 11: mid-point between 5 and 23, generate a relatively high accuracy before k becomes greater than 20
- 23: generate the highest accuracy based on the elbow plot

A small k value means that noise will have a higher influence on the result

A large k value make it computationally expensive




```
out5 <- knn(dat.train.x, dat.test.x, dat.train.y, k=5)

tab.knn5 <- table(dat.test.y, out5,
                  dnn = c("Actual", "Predicted"))
tab.knn5

out11 <- knn(dat.train.x, dat.test.x, dat.train.y, k=11)

tab.knn11 <- table(dat.test.y, out11,
                   dnn = c("Actual", "Predicted"))
tab.knn11

out23 <- knn(dat.train.x, dat.test.x, dat.train.y, k=23)

tab.knn23 <- table(dat.test.y, out23,
                  dnn = c("Actual", "Predicted"))
tab.knn23
```

Decision Trees

Overall Models

```
> tab_rpart
   y_hat_rpart
      0    1
  0 8149 2290
  1 3386 6844
> tab_tree1
   y.hat.tree1
      0    1
  0 8444 1995
  1 4014 6216
> tab_ctree1  #DEPTH 5
   y.hat.ctree1
      0    1
  0 8583 1856
  1 3840 6390
> tab_ctree2 #depth 10
   y.hat.ctree2
      0    1
  0 8006 2433
  1 3304 6926
> tab_ctree3 #depth 4
   y.hat.ctree3
      0    1
  0 7636 2803
  1 2929 7301
> tab_ctree4 # depth 2
   y.hat.ctree4
      0    1
  0 8444 1995
  1 4014 6216
>
```

The process for Decision Trees was to use different packages to find the best fitting tree based on Accuracy, Precision and FN Rate.

MODEL	ACCURACY	PRECISION
Rpart depth = 10	***0.7253858***	***0.6690127***
Dtrees depth = 1	0.7092748	0.6076246
Ctree depth = 10	0.7224346	0.6770283
Ctree depth = 5	***0.7244182***	***0.6246334***
Ctree depth = 4	***0.7226765***	***0.7136852***
Ctree depth = 2	0.7092748	0.6076246

The best model based on Accuracy, PRecision and, FNRate was determined to be cTree depth = 5. Though cTree depth 4 has higher precision, this model is mainly accounting for Type I error, in our case we believe Type II error is more critical in saving someone life, that is when they are predicted to not have CVD but do in fact have CVD.

RPart

```
> summary(fit1)
Call:
rpart(formula = cardio ~ ., data = train, method = "class", cp = 0.001)
  n= 48230

           CP nsplit rel error    xerror        xstd
1  0.420803485      0 1.0000000 1.0000000 0.004599767
2  0.010933769      1 0.5791965 0.5792803 0.004160469
3  0.005152696      3 0.5573290 0.5583344 0.004114134
4  0.002555402      4 0.5521763 0.5522601 0.004100184
5  0.002304051      5 0.5496209 0.5506263 0.004096392
6  0.001164593      6 0.5473168 0.5498722 0.004094636
7  0.001162498     11 0.5414939 0.5487412 0.004091995
8  0.001131080     15 0.5368439 0.5487412 0.004091995
9  0.001047296     19 0.5322358 0.5466466 0.004087083
10 0.001000000     20 0.5311885 0.5458087 0.004085110

Variable importance
      ap_hi       ap_lo cholesterol         age      weight        gluc
         48          26          10           9           5           2
```

Summary for all models

Tree

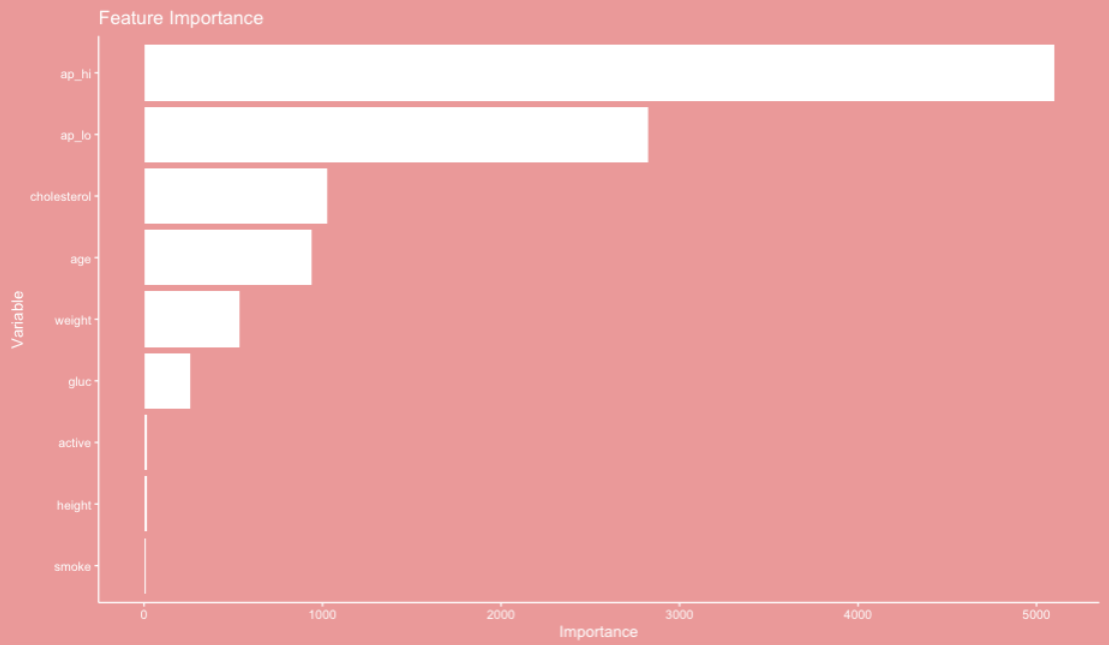
```
> summary(tree1)

Classification tree:
tree(formula = cardio ~ ., data = train)
Variables actually used in tree construction:
[1] "ap_hi" "age"
Number of terminal nodes:  4
Residual mean deviance:  1.135 = 54740 / 48230
Misclassification error rate: 0.2867 = 13826 / 48230
```

cTree

```
> summary(ctree1)
    Length      Class       Mode
         1 BinaryTree         S4
> summary(ctree1)
    Length      Class       Mode
         1 BinaryTree         S4
> summary(ctree2)
    Length      Class       Mode
         1 BinaryTree         S4
> summary(ctree3)
    Length      Class       Mode
         1 BinaryTree         S4
> summary(ctree4)
    Length      Class       Mode
         1 BinaryTree         S4
```

```
> mat.accuracies
                Accuracy - Validation
CTree1* D = 5               0.7244182
CTree2 D = 10               0.7224346
CTree3 D = 4                0.7226765
CTree3 D = 2                0.7092748
```



This feature importance graph was included with RPart package. You can see this also in the output for the summary in the previous slide. Note that the trees for both selected models in the presentation have similar splits to start with which include ap_hi, age, and cholesterol which are all considered important variables through this graph

Support Vector Machine

Case 1: Kernel = Linear, Cost = 1

```r
# Case 1: Linear Kernel
svmfit1 <- svm(cardio ~ .,
               data = train, kernel = "linear",
               cost = 1)

traintable1 <- table(truth = train$cardio, predict = svmfit1$fitted)
traintable1
testtable1 <- table(truth = test$cardio, predict = predict(svmfit1, test))
testtable1

confusionMatrix(traintable1)
confusionMatrix(testtable1)

# Train Accuracy: 0.7268
# Test Accuracy: 0.7239
```

```
> confusionMatrix(traintable1)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19970  4389
    1  8786 15085

               Accuracy : 0.7268
                 95% CI : (0.7228, 0.7308)
    No Information Rate : 0.5962
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4526

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

            Sensitivity : 0.6945
            Specificity : 0.7746
         Pos Pred Value : 0.8198
         Neg Pred Value : 0.6319
             Prevalence : 0.5962
         Detection Rate : 0.4141
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7345

       'Positive' Class : 0
```

```
> confusionMatrix(testtable1)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8508 1931
    1 3776 6454

               Accuracy : 0.7239
                 95% CI : (0.7177, 0.73)
    No Information Rate : 0.5943
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4467

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

            Sensitivity : 0.6926
            Specificity : 0.7697
         Pos Pred Value : 0.8150
         Neg Pred Value : 0.6309
             Prevalence : 0.5943
         Detection Rate : 0.4116
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7312

       'Positive' Class : 0
```

Case 2: Kernel = Radial, Cost = 1

```r
# Case 2: Radial Kernel
svmfit2 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 1)

traintable2 <- table(truth = train$cardio, predict = svmfit2$fitted)
traintable2
testtable2 <- table(truth = test$cardio, predict = predict(svmfit2, test))
testtable2

confusionMatrix(traintable2)
confusionMatrix(testtable2)

# Train Accuracy: 0.7391
# Test Accuracy: 0.7324
```

```
> confusionMatrix(traintable2)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19468  4891
    1  7693 16178

               Accuracy : 0.7391
                 95% CI : (0.7351, 0.743)
    No Information Rate : 0.5632
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4775

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

            Sensitivity : 0.7168
            Specificity : 0.7679
         Pos Pred Value : 0.7992
         Neg Pred Value : 0.6777
             Prevalence : 0.5632
         Detection Rate : 0.4036
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7423

       'Positive' Class : 0
```

```
> confusionMatrix(testtable2)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8212 2227
    1 3304 6926

               Accuracy : 0.7324
                 95% CI : (0.7263, 0.7384)
    No Information Rate : 0.5572
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4642

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

            Sensitivity : 0.7131
            Specificity : 0.7567
         Pos Pred Value : 0.7867
         Neg Pred Value : 0.6770
             Prevalence : 0.5572
         Detection Rate : 0.3973
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7349

       'Positive' Class : 0
```

Case 3: Kernel = Radial, Cost = 10

```r
# Case 3: Increasing Cost from 1 to 10
svmfit3 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 10)

traintable3 <- table(truth = train$cardio, predict = svmfit3$fitted)
traintable3
testtable3 <- table(truth = test$cardio, predict = predict(svmfit3, test))
testtable3

confusionMatrix(traintable3)
confusionMatrix(testtable3)

# Train Accuracy: 0.7487
# Test Accuracy: 0.7298
```

```
> confusionMatrix(traintable3)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19698  4661
    1  7458 16413

               Accuracy : 0.7487
                 95% CI : (0.7448, 0.7526)
    No Information Rate : 0.5631
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4968

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

            Sensitivity : 0.7254
            Specificity : 0.7788
         Pos Pred Value : 0.8087
         Neg Pred Value : 0.6876
             Prevalence : 0.5631
         Detection Rate : 0.4084
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7521

       'Positive' Class : 0
```

```
> confusionMatrix(testtable3)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8196 2243
    1 3342 6888

               Accuracy : 0.7298
                 95% CI : (0.7237, 0.7358)
    No Information Rate : 0.5582
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4589

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

            Sensitivity : 0.7103
            Specificity : 0.7544
         Pos Pred Value : 0.7851
         Neg Pred Value : 0.6733
             Prevalence : 0.5582
         Detection Rate : 0.3965
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7324

       'Positive' Class : 0
```

Case 4: Kernel = Radial, Cost = 0.1

```r
# Case 4: Decreasing Cost from 1 to 0.1
svmfit4 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 0.1)

traintable4 <- table(truth = train$cardio, predict = svmfit4$fitted)
traintable4
testtable4 <- table(truth = test$cardio, predict = predict(svmfit4, test))
testtable4

confusionMatrix(traintable4)
confusionMatrix(testtable4)

# Train Accuracy: 0.7331
# Test Accuracy: 0.7325
```

```
> confusionMatrix(traintable4)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19291  5068
    1  7803 16068

               Accuracy : 0.7331
                 95% CI : (0.7292, 0.7371)
    No Information Rate : 0.5618
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4656

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

            Sensitivity : 0.7120
            Specificity : 0.7602
         Pos Pred Value : 0.7919
         Neg Pred Value : 0.6731
             Prevalence : 0.5618
         Detection Rate : 0.4000
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7361

       'Positive' Class : 0
```

```
> confusionMatrix(testtable4)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8211 2228
    1 3301 6929

               Accuracy : 0.7325
                 95% CI : (0.7264, 0.7385)
    No Information Rate : 0.557
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4644

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

            Sensitivity : 0.7133
            Specificity : 0.7567
         Pos Pred Value : 0.7866
         Neg Pred Value : 0.6773
             Prevalence : 0.5570
         Detection Rate : 0.3973
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7350

       'Positive' Class : 0
```

Case 5: Kernel = Radial, Cost = 1, Gamma = 1

```r
# Case 5: Adding Gamma = 1
svmfit5 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 1, gamma = 1)

traintable5 <- table(truth = train$cardio, predict = svmfit5$fitted)
traintable5
testtable5 <- table(truth = test$cardio, predict = predict(svmfit5, test))
testtable5

confusionMatrix(traintable5)
confusionMatrix(testtable5)

# Train Accuracy: 0.7928
# Test Accuracy: 0.7203
```

```
> confusionMatrix(traintable5)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 20317  4042
    1  5953 17918

               Accuracy : 0.7928
                 95% CI : (0.7891, 0.7964)
    No Information Rate : 0.5447
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5852

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

            Sensitivity : 0.7734
            Specificity : 0.8159
         Pos Pred Value : 0.8341
         Neg Pred Value : 0.7506
             Prevalence : 0.5447
         Detection Rate : 0.4213
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7947

       'Positive' Class : 0
```

```
> confusionMatrix(testtable5)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 7720 2719
    1 3063 7167

               Accuracy : 0.7203
                 95% CI : (0.7141, 0.7264)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4403

 Mcnemar's Test P-Value : 6.458e-06

            Sensitivity : 0.7159
            Specificity : 0.7250
         Pos Pred Value : 0.7395
         Neg Pred Value : 0.7006
             Prevalence : 0.5217
         Detection Rate : 0.3735
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7205

       'Positive' Class : 0
```

Case 6: Kernel = Radial, Cost = 0.01

```r
# Case 6: Decreasing Cost from 1 to 0.01
svmfit6 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 0.01)

traintable6 <- table(truth = train$cardio, predict = svmfit6$fitted)
traintable6
testtable6 <- table(truth = test$cardio, predict = predict(svmfit6, test))
testtable6

confusionMatrix(traintable6)
confusionMatrix(testtable6)

# Train Accuracy: 0.7246
# Test Accuracy: 0.7254
```

```
> confusionMatrix(traintable6)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 18578  5781
    1  7501 16370

               Accuracy : 0.7246
                 95% CI : (0.7206, 0.7286)
    No Information Rate : 0.5407
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4488

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

            Sensitivity : 0.7124
            Specificity : 0.7390
         Pos Pred Value : 0.7627
         Neg Pred Value : 0.6858
             Prevalence : 0.5407
         Detection Rate : 0.3852
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7257

       'Positive' Class : 0
```

```
> confusionMatrix(testtable6)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 7938 2501
    1 3174 7056

               Accuracy : 0.7254
                 95% CI : (0.7193, 0.7315)
    No Information Rate : 0.5376
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4505

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

            Sensitivity : 0.7144
            Specificity : 0.7383
         Pos Pred Value : 0.7604
         Neg Pred Value : 0.6897
             Prevalence : 0.5376
         Detection Rate : 0.3841
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7263

       'Positive' Class : 0
```

Case 7: Kernel = Radial, Cost = 1, Gamma = 0.1

```r
# Case 7: Adding Gamma = 0.1
svmfit7 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 1, gamma = 0.1)

traintable7 <- table(truth = train$cardio, predict = svmfit7$fitted)
traintable7
testtable7 <- table(truth = test$cardio, predict = predict(svmfit7, test))
testtable7

confusionMatrix(traintable7)
confusionMatrix(testtable7)

# Train Accuracy: 0.7413
# Test Accuracy: 0.7328
```

```
> confusionMatrix(traintable7)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19478  4881
    1  7595 16276

               Accuracy : 0.7413
                 95% CI : (0.7374, 0.7452)
    No Information Rate : 0.5613
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.482

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

            Sensitivity : 0.7195
            Specificity : 0.7693
         Pos Pred Value : 0.7996
         Neg Pred Value : 0.6818
             Prevalence : 0.5613
         Detection Rate : 0.4039
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7444

       'Positive' Class : 0
```

```
> confusionMatrix(testtable7)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8194 2245
    1 3277 6953

               Accuracy : 0.7328
                 95% CI : (0.7267, 0.7389)
    No Information Rate : 0.555
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4651

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

            Sensitivity : 0.7143
            Specificity : 0.7559
         Pos Pred Value : 0.7849
         Neg Pred Value : 0.6797
             Prevalence : 0.5550
         Detection Rate : 0.3964
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7351

       'Positive' Class : 0
```

Case 8: Kernel = Radial, Cost = 1, Gamma = 5

```r
# Case 8: Adding Gamma = 5
svmfit8 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 1, gamma = 5)

traintable8 <- table(truth = train$cardio, predict = svmfit8$fitted)
traintable8
testtable8 <- table(truth = test$cardio, predict = predict(svmfit8, test))
testtable8

confusionMatrix(traintable8)
confusionMatrix(testtable8)

# Train Accuracy: 0.8729
# Test Accuracy: 0.6841
```

```
> confusionMatrix(traintable8)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 22288  2071
    1  4061 19810

               Accuracy : 0.8729
                 95% CI : (0.8699, 0.8758)
    No Information Rate : 0.5463
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7455

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

            Sensitivity : 0.8459
            Specificity : 0.9054
         Pos Pred Value : 0.9150
         Neg Pred Value : 0.8299
             Prevalence : 0.5463
         Detection Rate : 0.4621
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.8756

       'Positive' Class : 0
```

```
> confusionMatrix(testtable8)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 6608 3831
    1 2698 7532

               Accuracy : 0.6841
                 95% CI : (0.6777, 0.6905)
    No Information Rate : 0.5498
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3689

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

            Sensitivity : 0.7101
            Specificity : 0.6629
         Pos Pred Value : 0.6330
         Neg Pred Value : 0.7363
             Prevalence : 0.4502
         Detection Rate : 0.3197
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.6865

       'Positive' Class : 0
```

Case 9: Kernel = Radial, Cost = 5

```r
# Case 9: Changing cost from 1 to 5
svmfit9 <- svm(cardio ~ .,
               data = train, kernel = "radial",
               cost = 5)

traintable9 <- table(truth = train$cardio, predict = svmfit9$fitted)
traintable9
testtable9 <- table(truth = test$cardio, predict = predict(svmfit9, test))
testtable9

confusionMatrix(traintable9)
confusionMatrix(testtable9)

# Train Accuracy: 0.7458
# Test Accuracy: 0.7306
```

```
> confusionMatrix(traintable9)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19616  4743
    1  7515 16356

               Accuracy : 0.7458
                 95% CI : (0.7419, 0.7497)
    No Information Rate : 0.5625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.491

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

            Sensitivity : 0.7230
            Specificity : 0.7752
         Pos Pred Value : 0.8053
         Neg Pred Value : 0.6852
             Prevalence : 0.5625
         Detection Rate : 0.4067
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7491

       'Positive' Class : 0
```

```
> confusionMatrix(testtable9)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8193 2246
    1 3323 6907

               Accuracy : 0.7306
                 95% CI : (0.7245, 0.7366)
    No Information Rate : 0.5572
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4605

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

            Sensitivity : 0.7114
            Specificity : 0.7546
         Pos Pred Value : 0.7848
         Neg Pred Value : 0.6752
             Prevalence : 0.5572
         Detection Rate : 0.3964
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7330

       'Positive' Class : 0
```

Case 10: Kernel = Radial, Cost = 0.1, Gamma = 0.1

```r
# Case 10: Cost = 0.1, Gamma = 0.1
svmfit10 <- svm(cardio ~ .,
                data = train, kernel = "radial",
                cost = 0.1, gamma = 0.1)

traintable10 <- table(truth = train$cardio, predict = svmfit10$fitted)
traintable10
testtable10 <- table(truth = test$cardio, predict = predict(svmfit10, test))
testtable10

confusionMatrix(traintable10)
confusionMatrix(testtable10)

# Train Accuracy: 0.7333
# Test Accuracy: 0.73
```

```
> confusionMatrix(traintable10)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19180  5179
    1  7684 16187

               Accuracy : 0.7333
                 95% CI : (0.7293, 0.7372)
    No Information Rate : 0.557
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.466

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

            Sensitivity : 0.7140
            Specificity : 0.7576
         Pos Pred Value : 0.7874
         Neg Pred Value : 0.6781
             Prevalence : 0.5570
         Detection Rate : 0.3977
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7358

       'Positive' Class : 0
```

```
> confusionMatrix(testtable10)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8126 2313
    1 3267 6963

               Accuracy : 0.73
                 95% CI : (0.7239, 0.7361)
    No Information Rate : 0.5512
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4595

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

            Sensitivity : 0.7132
            Specificity : 0.7506
         Pos Pred Value : 0.7784
         Neg Pred Value : 0.6806
             Prevalence : 0.5512
         Detection Rate : 0.3931
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7319

       'Positive' Class : 0
```

Case 11: Kernel = Radial, Cost = 5

```r
# Case 11: Cost = 10, Gamma = 0.1
svmfit11 <- svm(cardio ~ .,
                data = train, kernel = "radial",
                cost = 10, gamma = 0.1)

traintable11 <- table(truth = train$cardio, predict = svmfit11$fitted)
traintable11
testtable11 <- table(truth = test$cardio, predict = predict(svmfit11, test))
testtable11

confusionMatrix(traintable11)
confusionMatrix(testtable11)

# Train Accuracy: 0.7534
# Test Accuracy: 0.7286
```

```
> confusionMatrix(traintable11)
Confusion Matrix and Statistics

     predict
truth     0     1
    0 19781  4578
    1  7315 16556

               Accuracy : 0.7534
                 95% CI : (0.7495, 0.7573)
    No Information Rate : 0.5618
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5062

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

            Sensitivity : 0.7300
            Specificity : 0.7834
         Pos Pred Value : 0.8121
         Neg Pred Value : 0.6936
             Prevalence : 0.5618
         Detection Rate : 0.4101
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7567

       'Positive' Class : 0
```

```
> confusionMatrix(testtable11)
Confusion Matrix and Statistics

     predict
truth    0    1
    0 8182 2257
    1 3352 6878

               Accuracy : 0.7286
                 95% CI : (0.7225, 0.7347)
    No Information Rate : 0.558
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4566

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

            Sensitivity : 0.7094
            Specificity : 0.7529
         Pos Pred Value : 0.7838
         Neg Pred Value : 0.6723
             Prevalence : 0.5580
         Detection Rate : 0.3959
   Detection Prevalence : 0.5051
      Balanced Accuracy : 0.7312

       'Positive' Class : 0
```

Best Model

```r
# BEST MODEL: Case 7

svmBest <- svm(cardio ~ ., data = train, kernel = "radial",
               cost = 1, gamma = 0.1)

traintablebest <- table(truth = train$cardio, predict = svmBest$fitted)
traintablebest
testtablebest  <- table(truth = test$cardio, predict = predict(svmBest, test))
testtablebest

confusionMatrix(traintablebest)
confusionMatrix(testtablebest)

# Train Accuracy: 0.7413
# Test Accuracy: 0.7328

summary(svmBest)
```

```
svm(formula = cardio ~ ., data = train, kernel = "radial", cost = 1, gamma = 0.1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  28779

 ( 14562 14217 )


Number of Classes:  2
```

Insights

Case	Train Set			Test Set		
	Accuracy	Positive Precision	Negative Precision	Accuracy	Positive Precision	Negative Precision
1	0.7268	0.8198	0.6319	0.7239	0.8150	0.6309
2	0.7391	0.7992	0.6777	0.7324	0.7867	0.6770
3	0.7487	0.8087	0.6876	0.7298	0.7851	0.6733
4	0.7331	0.7919	0.6731	0.7325	0.7866	0.6773
5	0.7928	0.8341	0.7906	0.7203	0.7395	0.7006
6	0.7246	0.7627	0.6858	0.7254	0.7604	0.6897
7	**0.7413**	**0.7996**	**0.6818**	**0.7328**	**0.7849**	**0.6797**
8	0.8729	0.9150	0.8299	0.6841	0.6330	0.7363
9	0.7458	0.8053	0.6852	0.7306	0.7848	0.6752
10	0.7333	0.7874	0.6781	0.7300	0.7784	0.6806
11	0.7534	0.8121	0.6936	0.7286	0.7838	0.6723

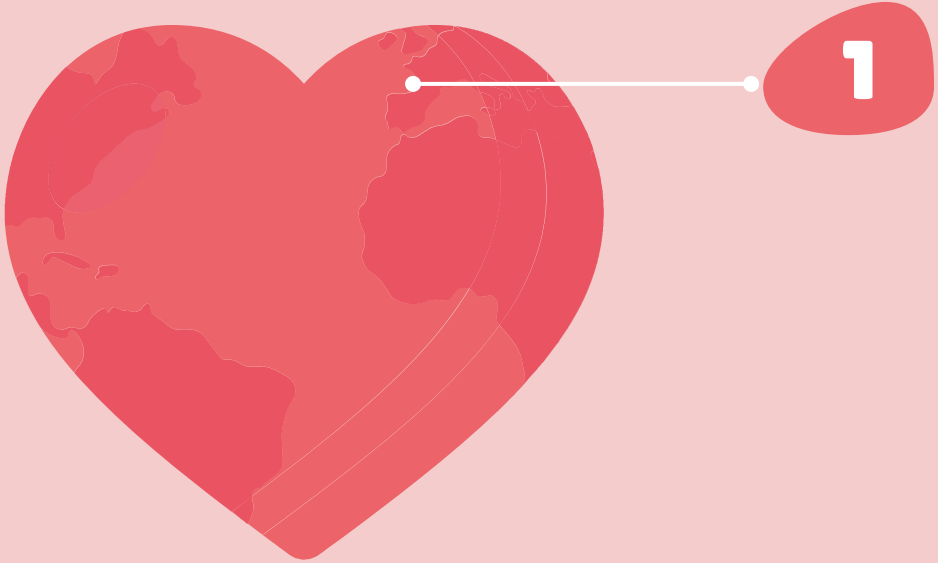
Cross Validation

```
cv_sample = dat3[sample(1:nrow(dat3), 6870),]
cv_sample$cardio <- as.factor(cv_sample$cardio)
tune.out <- tune(svm, cardio ~ ., data = cv_sample, ranges = list(cost = c(0.1, 1, 10), gamma = c(0.1, 1, 10),
                                                                  kernel = c('linear', 'radial')))
tune.out$best.parameters
```

```
> tune.out$best.parameters
   cost gamma kernel
11    1   0.1 radial
```

- We manually ran 11 different models trying different kernel, gamma & cost inputs. As shown previous matrix showing the outcomes and based on overall accuracy and precision Model 7 is performing best among this set.
- Cross Validation Sample is 10% of the dataset
- From both cross validation SVM and Model 7 train & test SVM model, we have the best overall accuracy while ensuring we have high precision and minimize both type 1 & type 2 errors.

TAKEAWAYS AND LEARNINGS

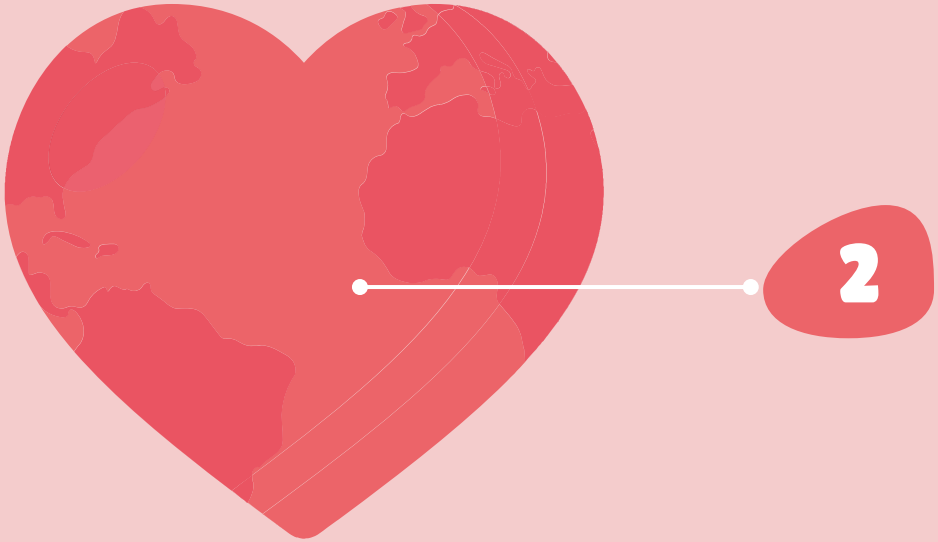


1

Cholesterol and Glucose Level are signs to look out for in CVD Detection (Logistic Regression)

Cholesterol & glucose both are significant and positive variables in our logistic regression model, indicating increase in CVD when the levels are high. Also the interaction between these variables also is significant.

TAKEAWAYS AND LEARNINGS

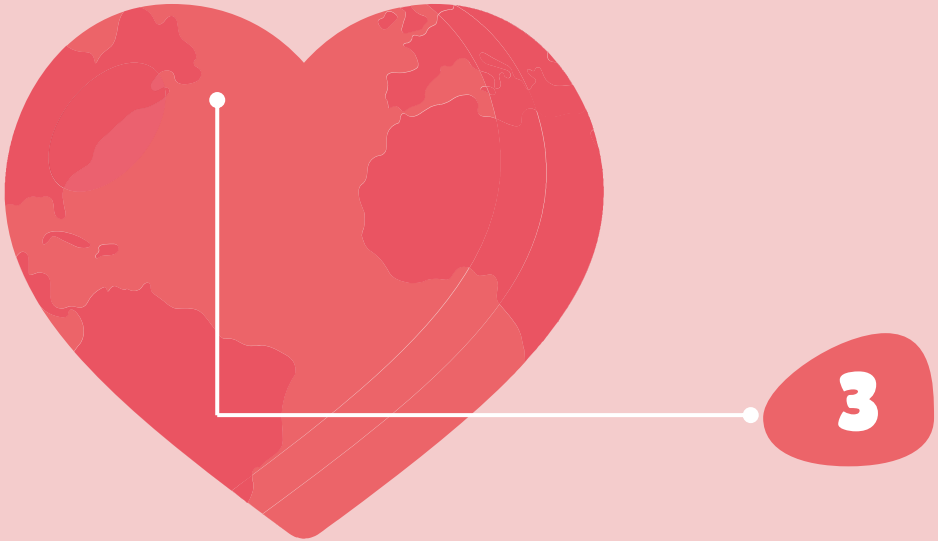


2

Reducing Alcohol and Smoking can have benefits of avoiding CVD.

In our data there is less number of people indicating they smoke and consume alcohol. It is expected people won't be forthcoming about habits and its important to try and get more information on this variable as its significant for predicting CVD.

TAKEAWAYS AND LEARNINGS



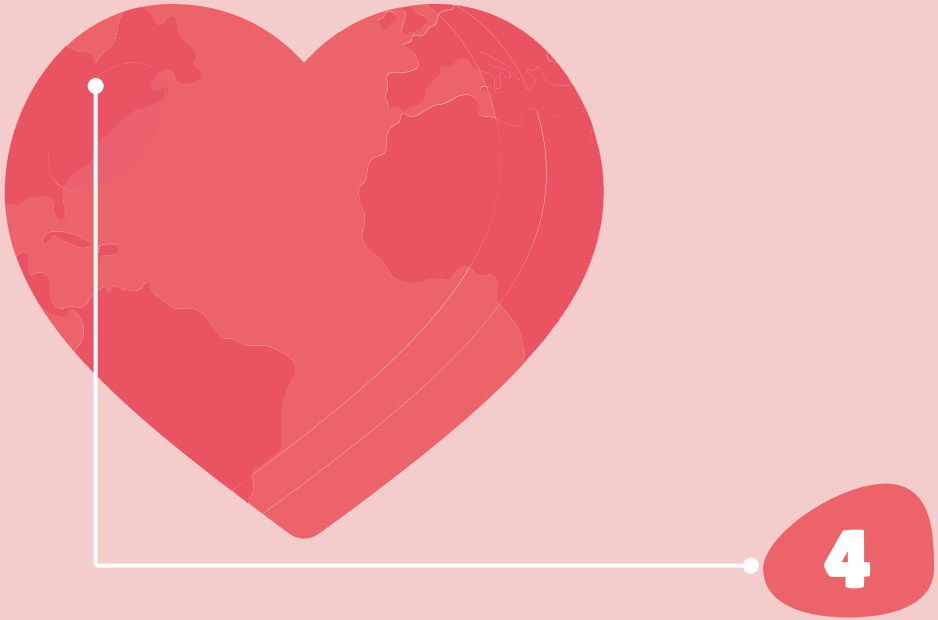
Best Model for classifying CVD is the SVM Model based on the current dataset

With the below accuracies;

- Training Accuracy: 74.13%
- Testing Accuracy: 73.28%
- Precision Accuracy: 67.97%

We fee this is the best model as it has highest overall accuracy as well as high precision, which is very important in classifying if a person has CVD or not

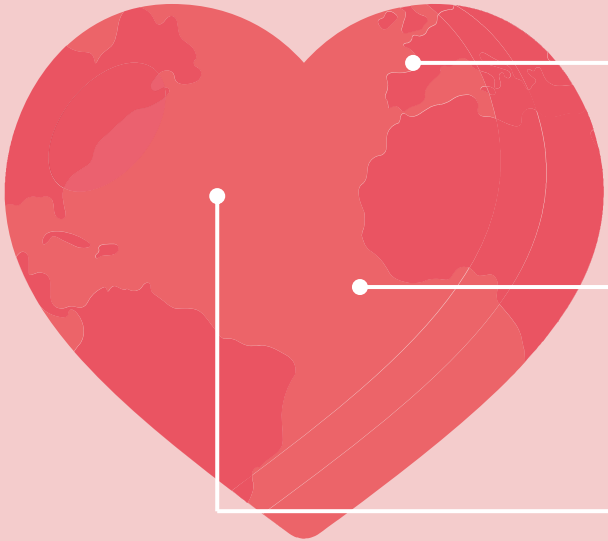
TAKEAWAYS AND LEARNINGS



People at age 35-50, having borderline cholesterol, glucose and blood pressure should receive CVD awareness.

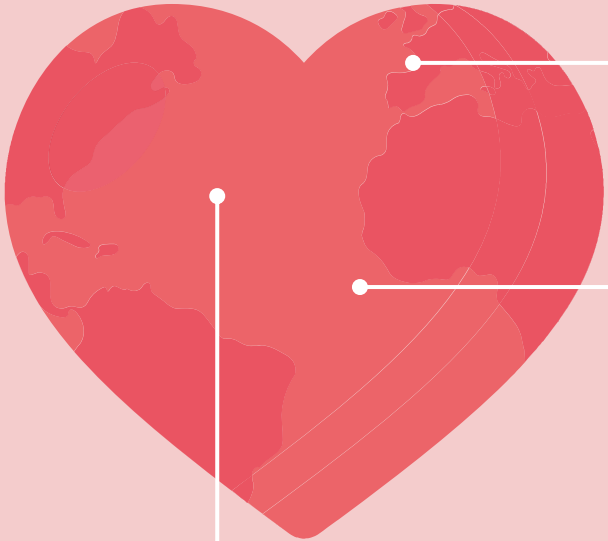
It's a ideal to avoid CVD, focusing on the onset of the disease or related symptoms will likely help reducing it affecting a lot of people. Business wise the cost of providing healthier lifestyle is lower compared to medical expenses associated with CVD.

CHALLENGES



1. More patient information is required
2. The gender bias is data needs to addressed.
3. The overall accuracy is not good enough for model selection

CHALLENGES



1. More patient information is required such as family medical history, allergies, lifestyle information to make more informative predictions

2. The biases is data needs to addressed as there is certain biases in terms of certain variables which needs to be addressed and change around for the future

3. The overall accuracy is not good enough for model selection as there is certain degree of bias in the data which causes very close values in the accuracies