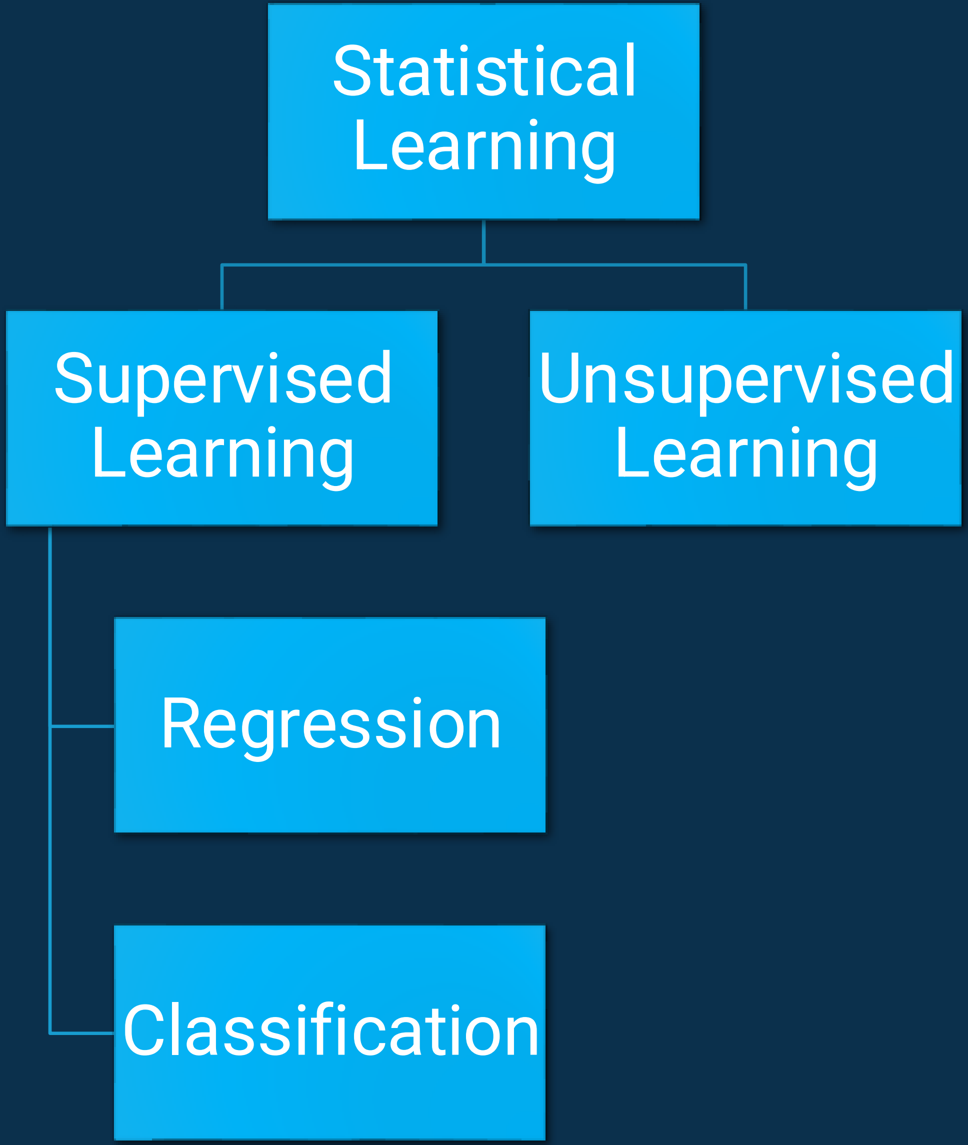
Session 1 (DAY 1)

Statistical and Machine Learning Models

Workshop on Quantitative Literacy and Statistics

UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES


Statistical Learning
Supervised Learning
Unsupervised Learning
Regression
Classification

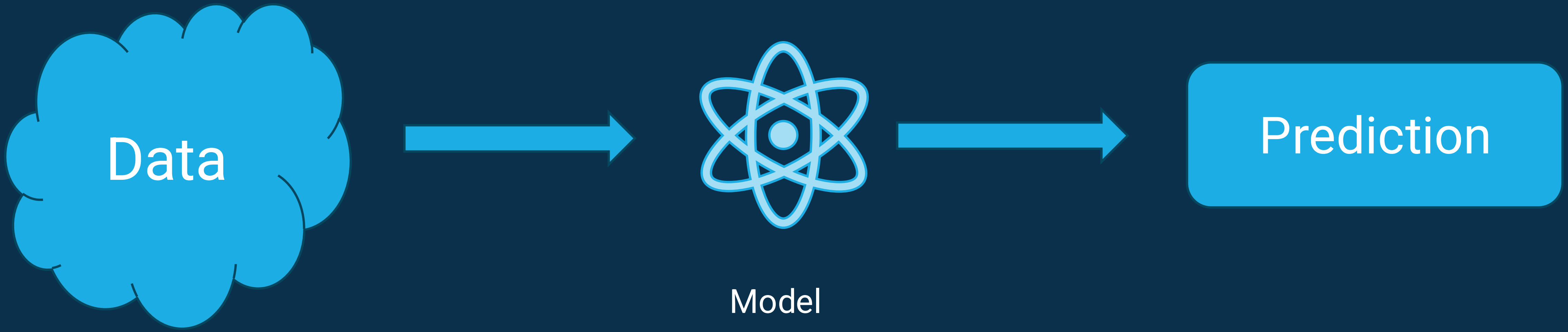
Session 2: Part 1

Supervised Learning

Workshop on Quantitative Literacy and Statistics

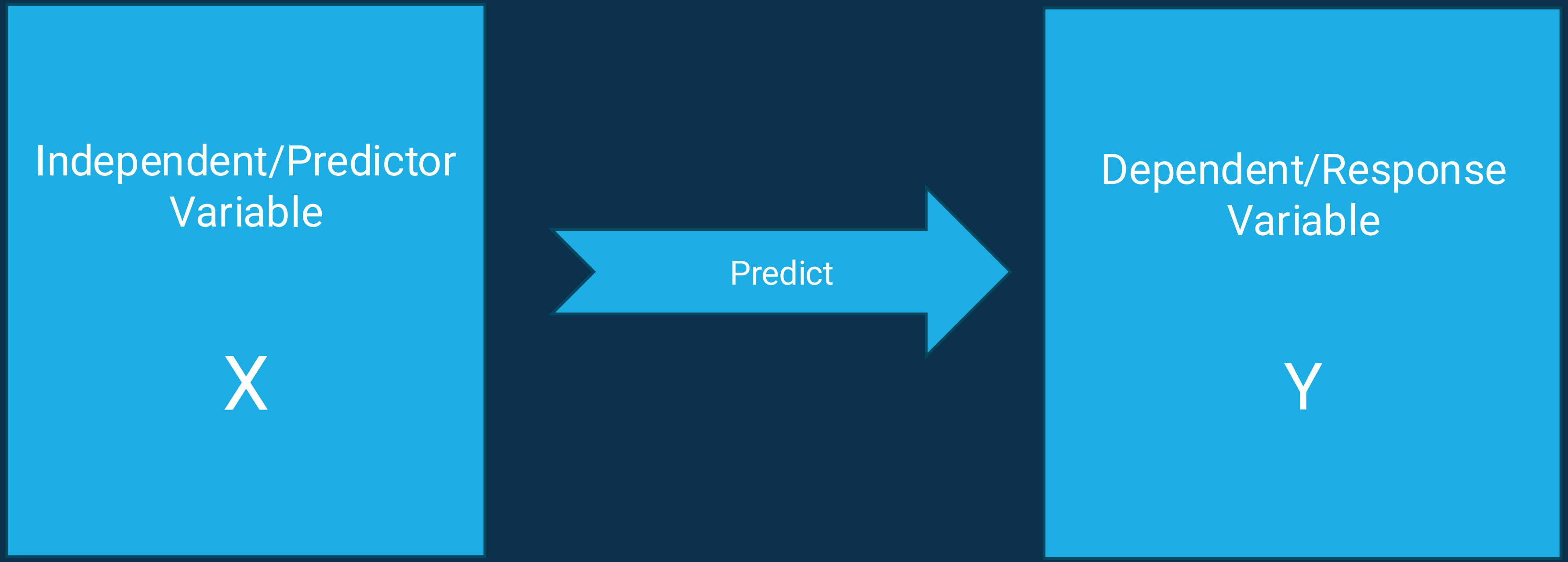
UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES

Supervised Learning



Supervised Learning

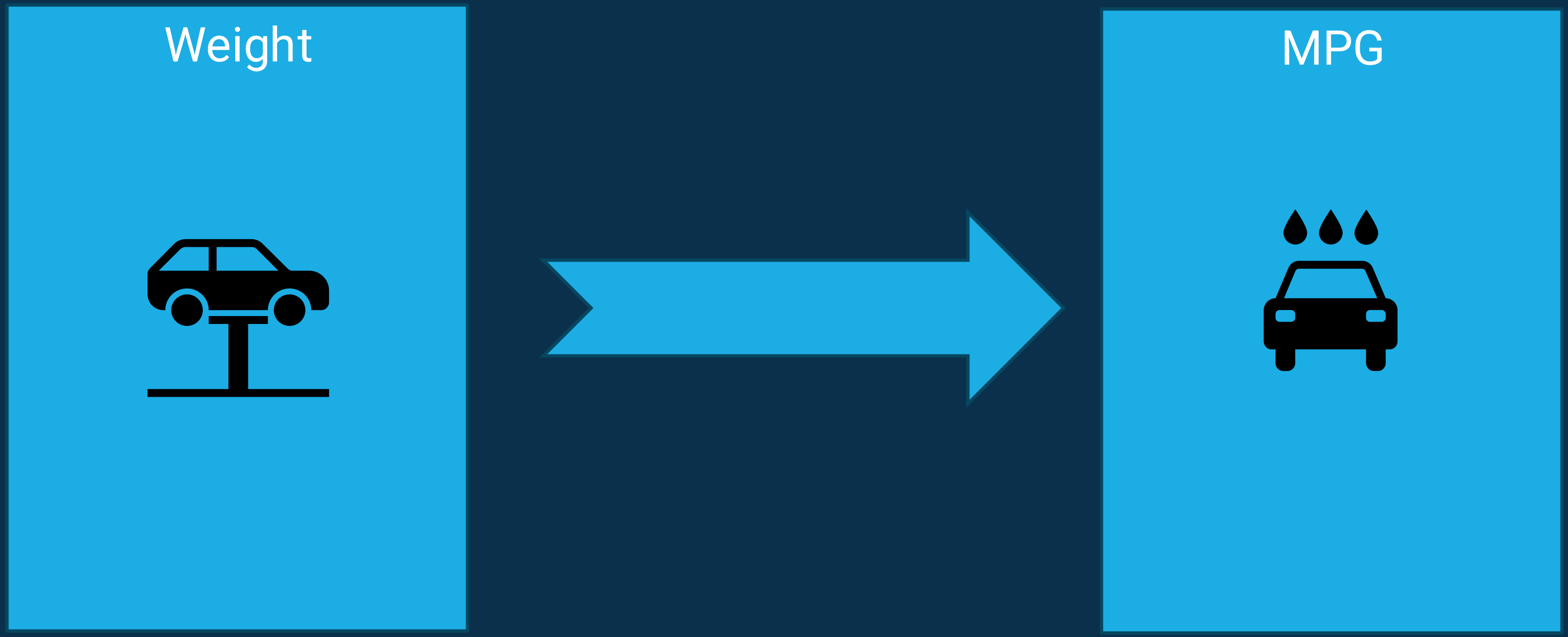
Learning techniques to find a function to **<u>predict a response</u>** variable (Y) from set of independent variables (X)



$$Y = f(x)$$

Supervised Learning: Regression

Predict **Continuous Response (Y)** from set of Independent Variables (X)



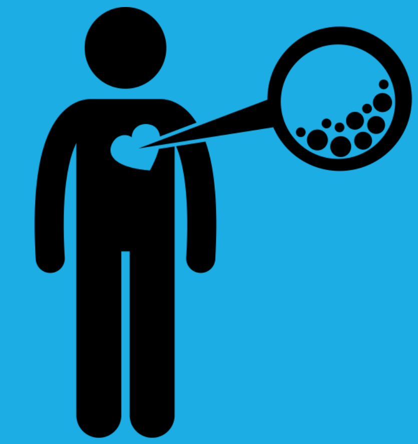
$$MPG = f(Weight)$$

Supervised Learning: Classification

Predict a **Categorical Response Variable** from a set of Independent Variables

Cholesterol Level

Get a Heart Disease or Not



$$\text{Heart Disease}_{(Yes\ or\ No)} = f(\text{Cholesterol Level})$$

Supervised Learning

1. Linear Models

Regression	: Linear Regression
Classification	: Logistic Regression

2. Bagging Methods

Regression	: Random Forest Regression
Classification	: Random Forest Classification

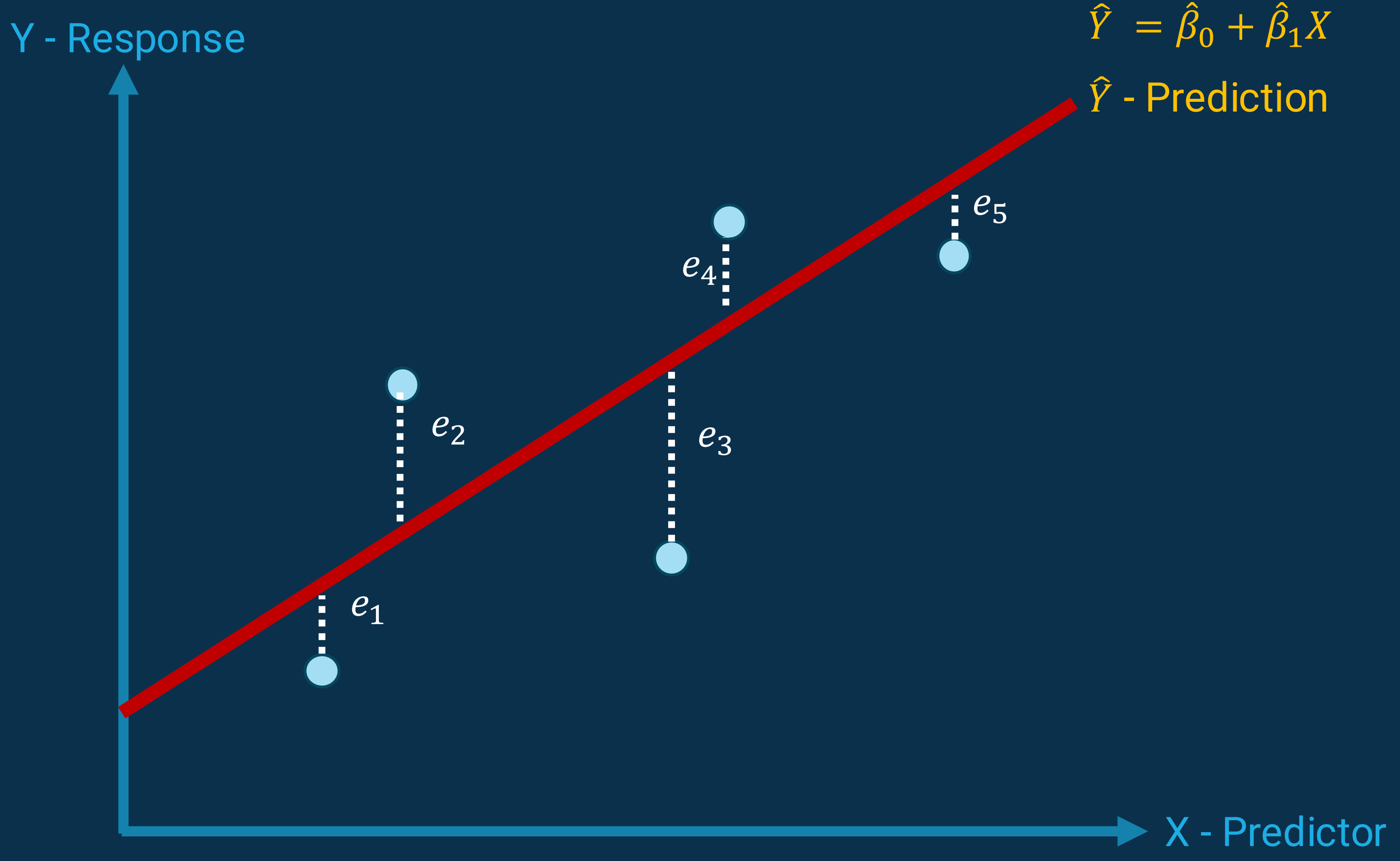
3. Boosting Methods

Regression	: Gradient Boost Regression
Classification	: Gradient Boost Classification

1. Linear Models

Regression

Linear Regression

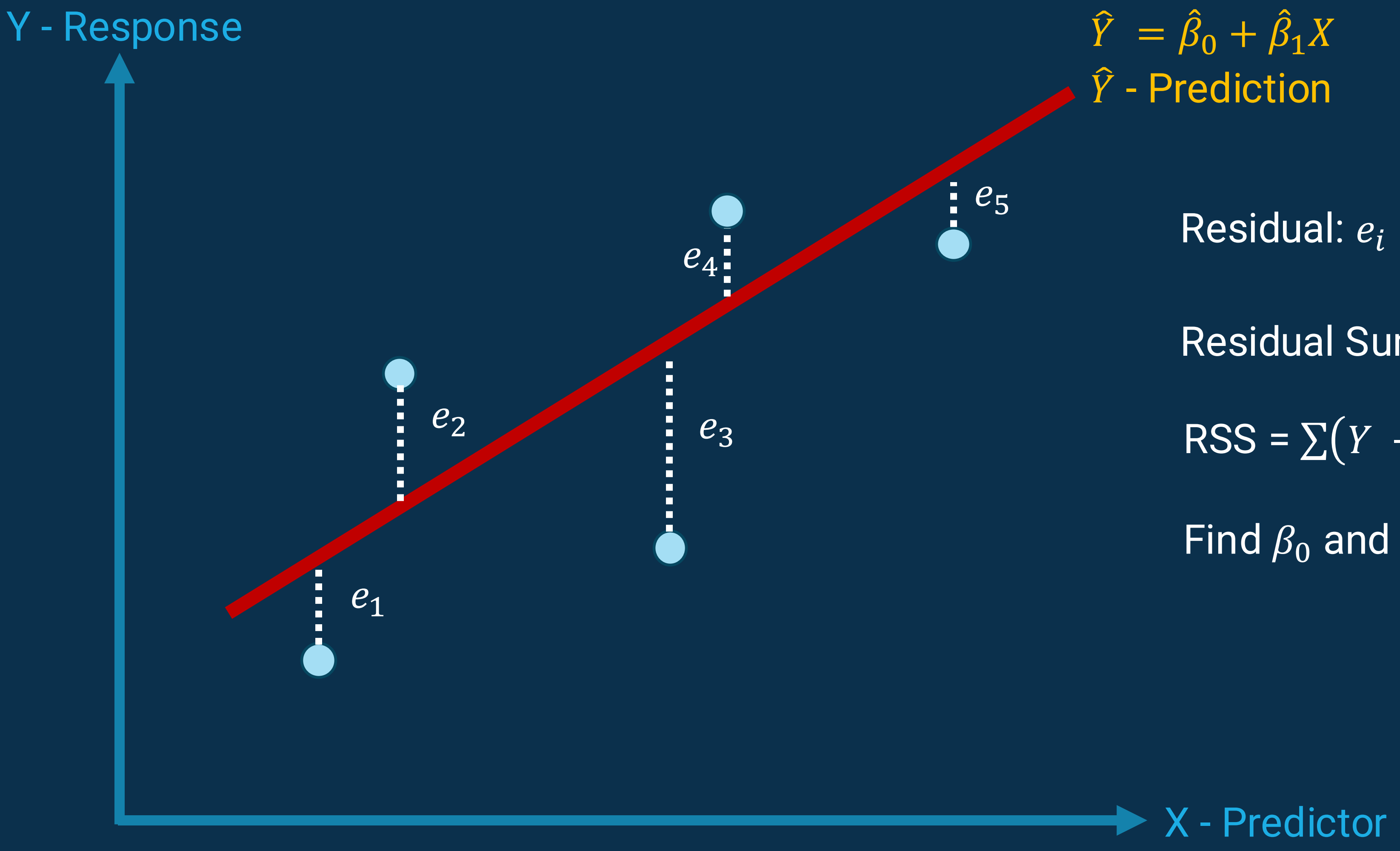


X	Y
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
x_5	y_5

Simple Linear Regression Model: $Y = \beta_0 + \beta_1 X + \text{Error}$

β_0 - Response when X = 0

β_1 - Increase of Response when X increase by 1-unit

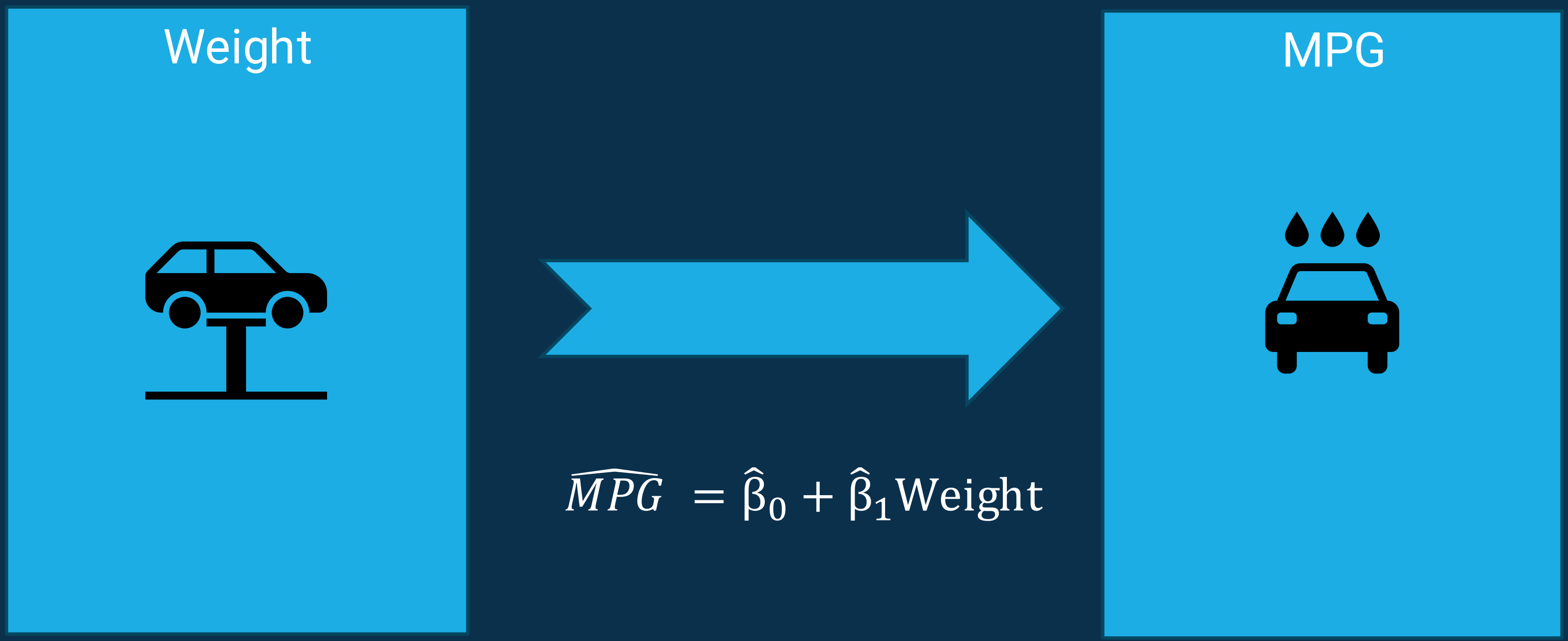


Residual: $e_i = y_i - \hat{y_i}$

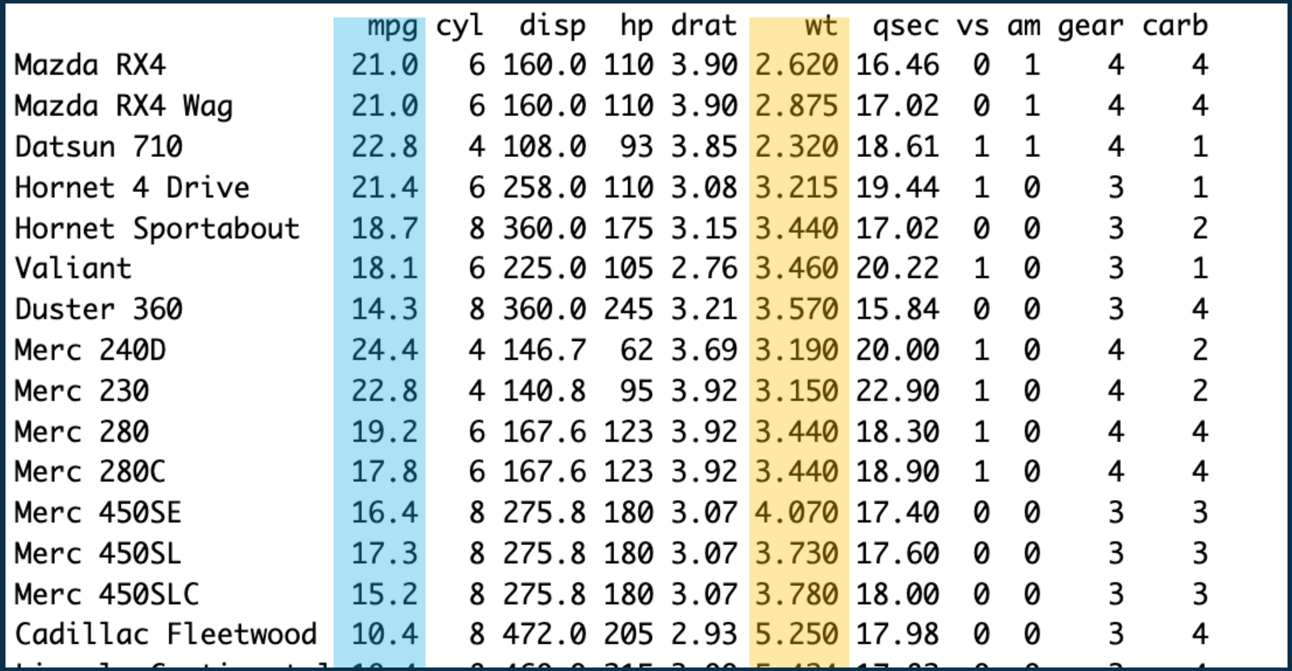
Residual Sum of Square: RSS = $e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$

RSS = $\sum(Y - \hat{Y})^2$

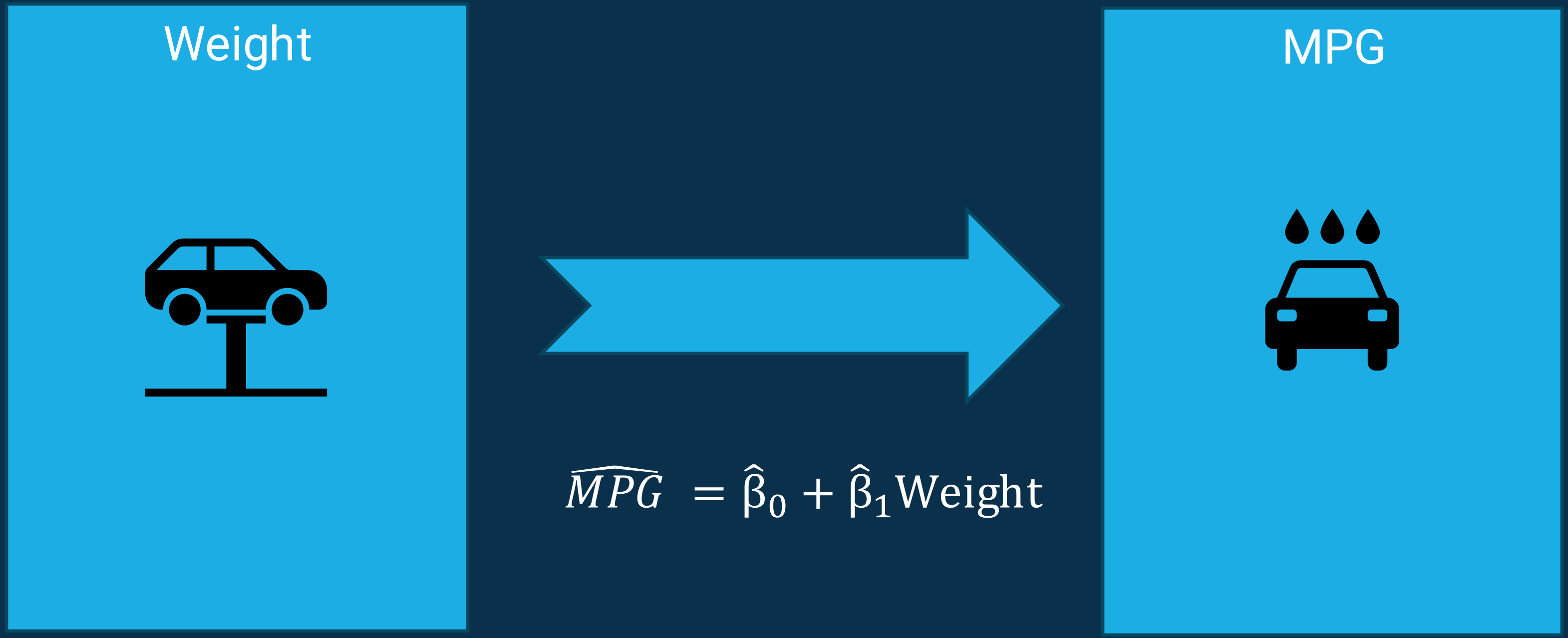
Find β_0 and β_1 which minimize the RSS



$$\widehat{MPG} = \hat{\beta}_0 + \hat{\beta}_1 \text{Weight}$$



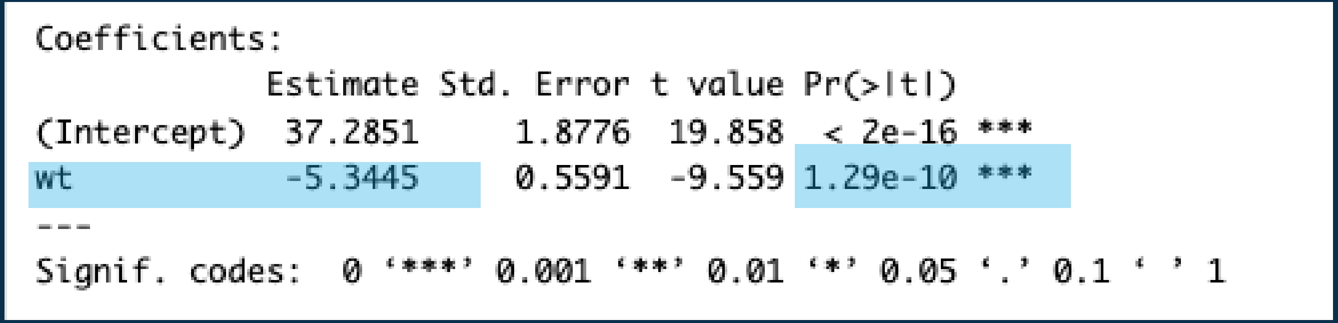
```
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
```



$$\widehat{MPG} = \hat{\beta}_0 + \hat{\beta}_1 \text{Weight}$$

```
lm(formula = mpg ~ wt ,data = mtcars)
```

$$\hat{\beta}_0 = 37.28$$
$$\hat{\beta}_1 = -5.34$$

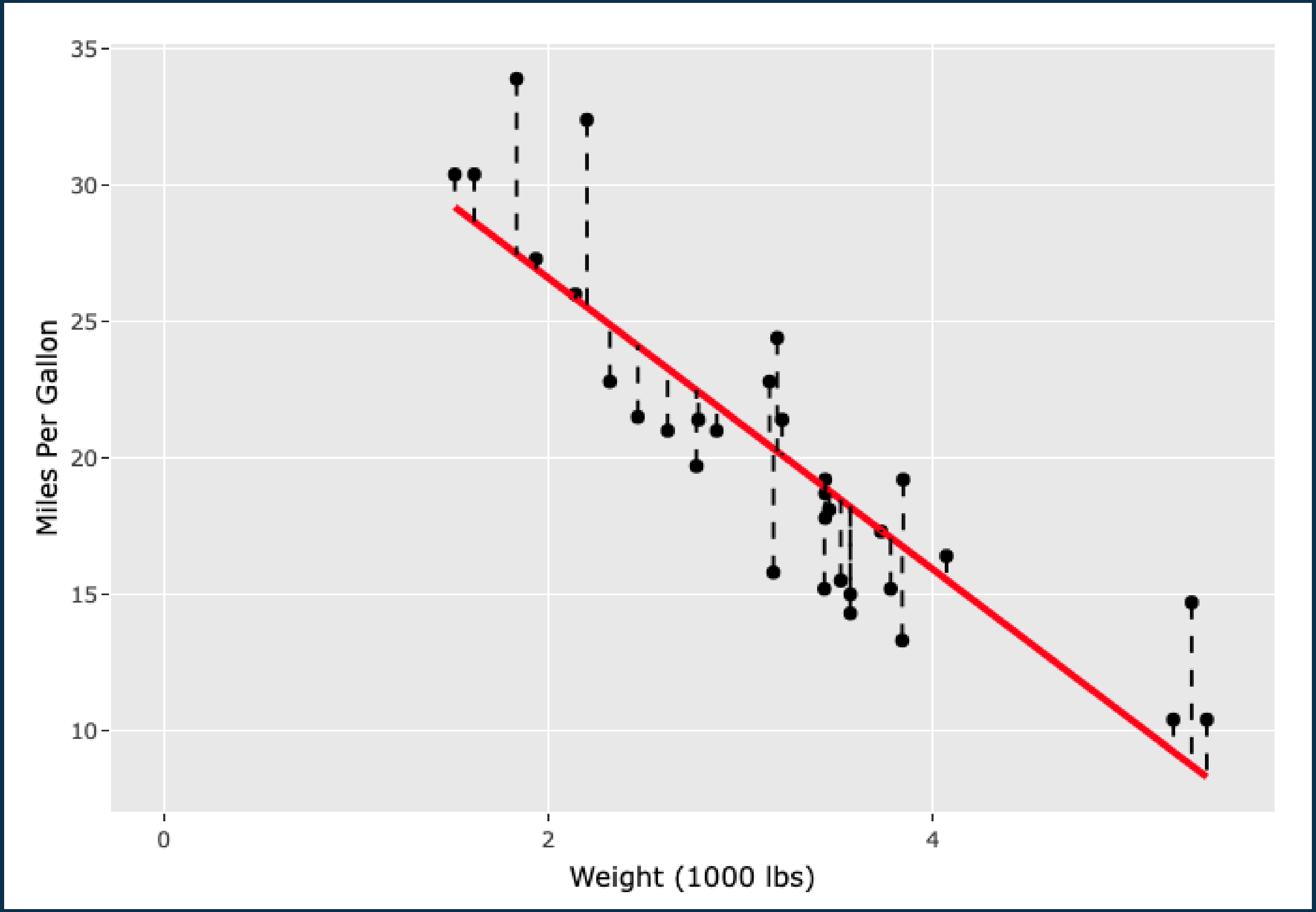


```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

$$\widehat{mpg} = 37.28 - 5.34\,Weight$$

p-value will provide statistical significance
β_1 – Weight is a significant variable in estimating MPG

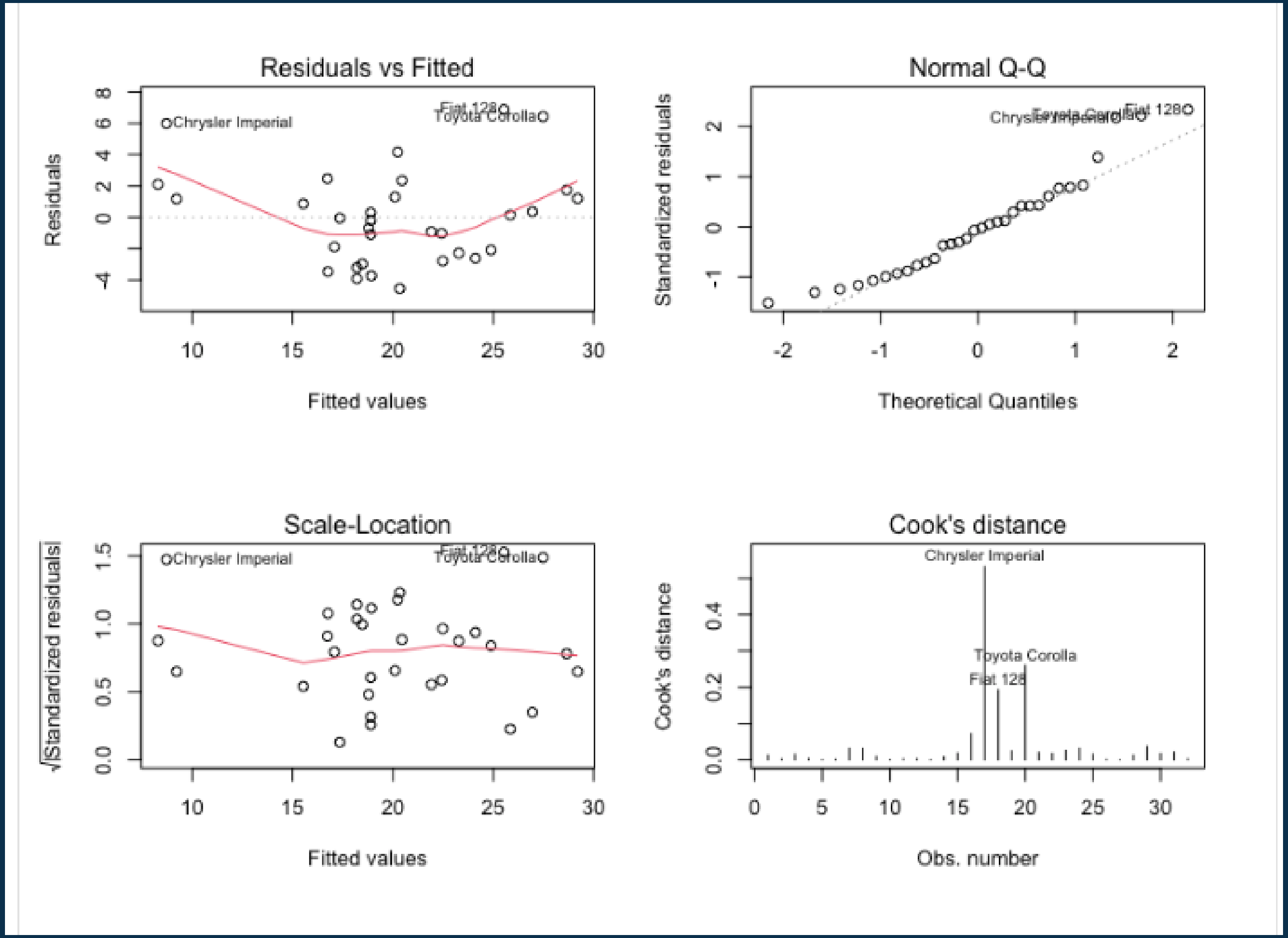
$$\widehat{mpg} = 37.28 - 5.34\,Weight$$



Linear Regression Assumptions

Assumptions

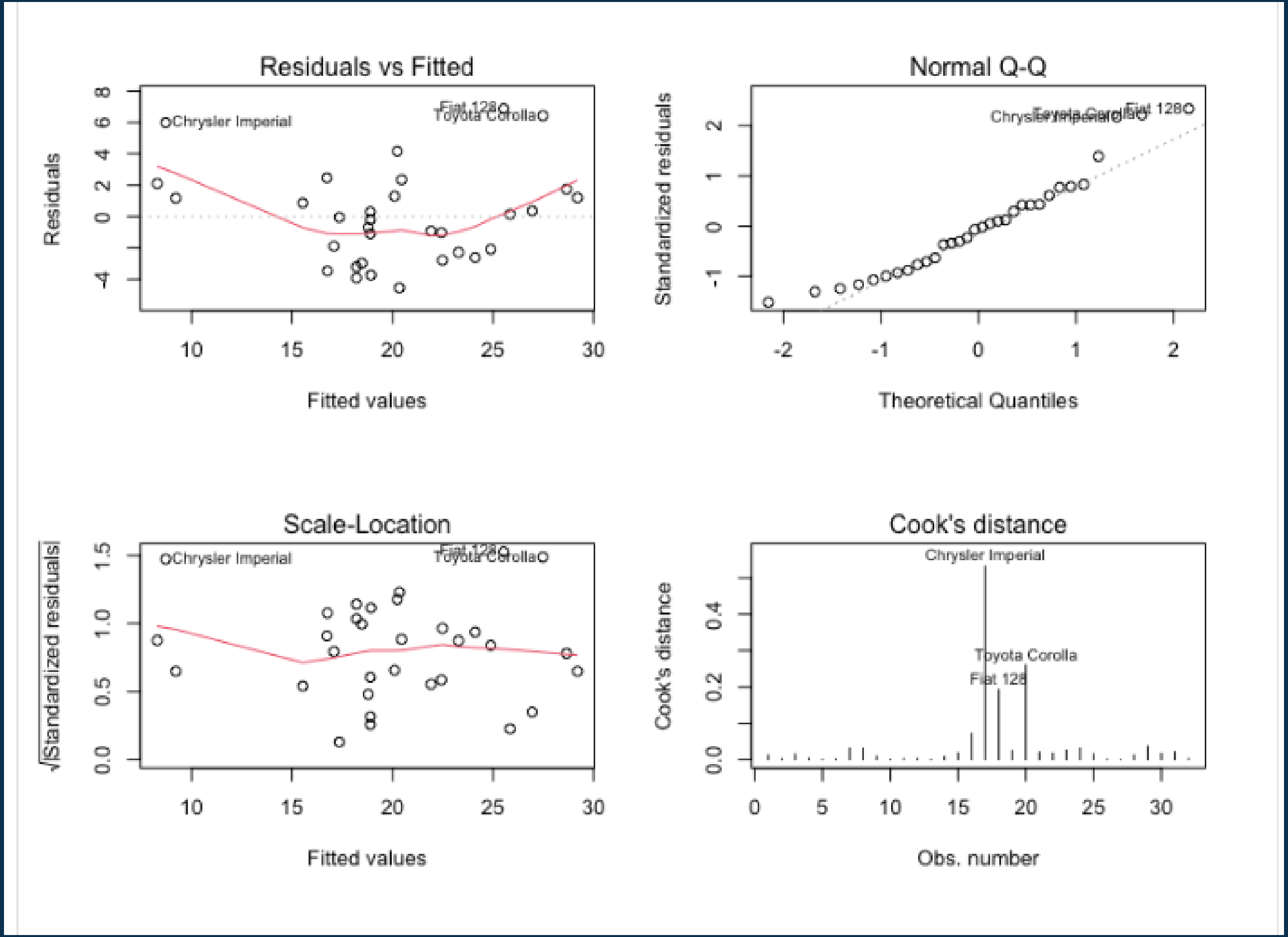
- Linearity of Data
- Independence of Errors
- Constant Variance of residuals
- Normality of Residuals
- No Multicollinearity



Linear Regression Assumptions

Assumptions

- Linearity of Data
  - Transform X variables
- Independence of Errors
  - Use Autoregressive Models
- Constant Variance of residuals
  - Transform dependent variable
- Normality of Residuals
  - Remove/impute outliers
  - Transform dependent variables
- No Multicollinearity
  - Remove or combine highly correlated X variables
  - Use Principal component analysis



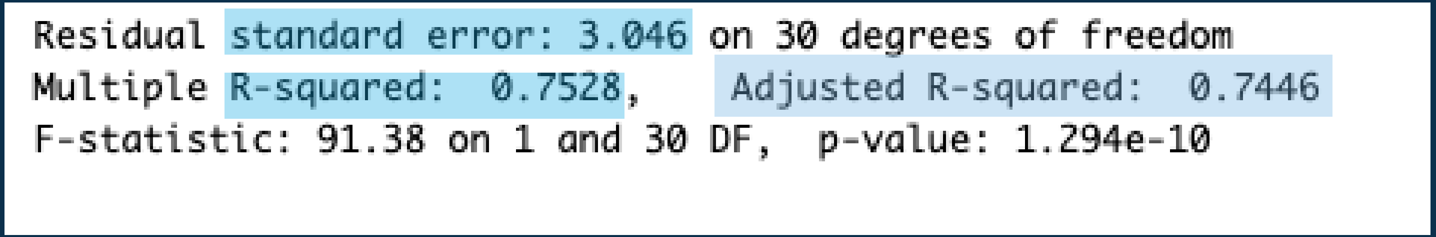
Linear Regression: Evaluation Model Performance

R^2 - Coefficient of Determination = 0.75

- How much variability of response is explained by the predictor variables
- $0 \leq R^2 \leq 1$
- Higher the better

Adjusted R^2 = 0.74

- Panelize for additional predictor variables
- $R^2_{adj} \leq 1$
- Higher the better



```
Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,	Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

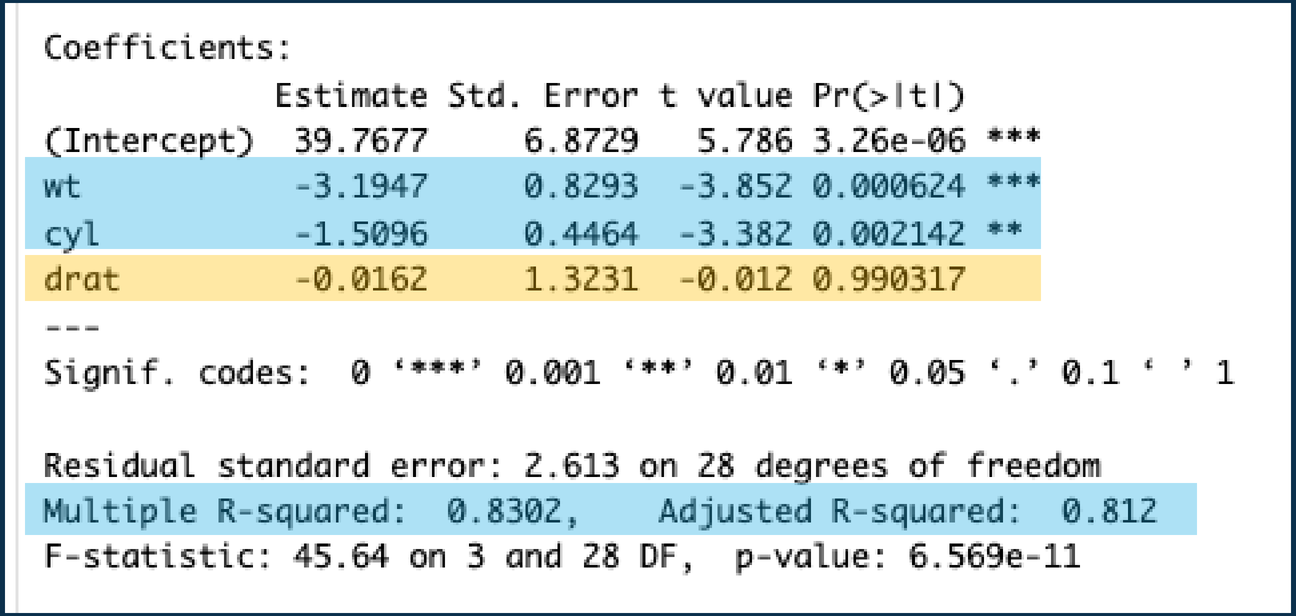
RMSE – Root mean squared error = 3.04

- Lower the Better

Multiple Linear Regression

$$MPG = \beta_0 + \beta_1\, Weight + \beta_2\, Cylinders + \beta_3\, Rear\_axle\_Ratio + Error$$

```
multiple_linear_regression = lm(formula = mpg ~ wt + cyl + drat ,data = mtcars)
summary(multiple_linear_regression)
```



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.7677     6.8729   5.786 3.26e-06 ***
wt           -3.1947     0.8293  -3.852 0.000624 ***
cyl          -1.5096     0.4464  -3.382 0.002142 **
drat         -0.0162     1.3231  -0.012 0.990317
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.613 on 28 degrees of freedom
Multiple R-squared:  0.8302,	Adjusted R-squared:  0.812
F-statistic: 45.64 on 3 and 28 DF,  p-value: 6.569e-11
```

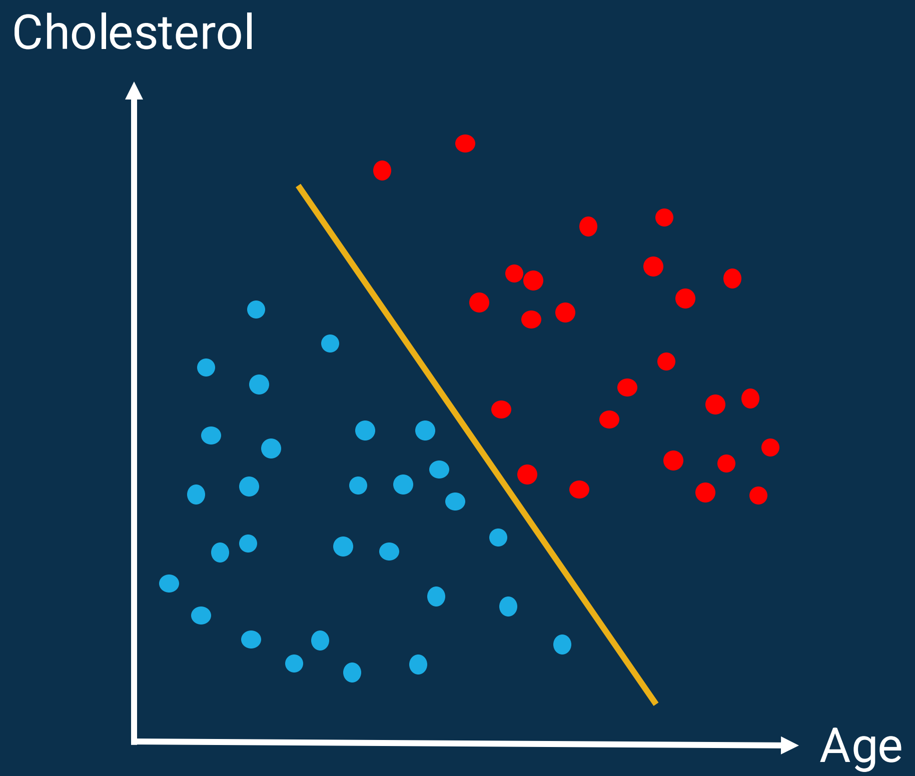
$$R^2 = 0.83$$

$$R^2_{adj} = 0.81$$

$$\widehat{MPG} = 39.76 - 3.19\, Weight - 1.51\, Cylinders - 0.016\; Rear\_axle\_Ratio$$

Classification

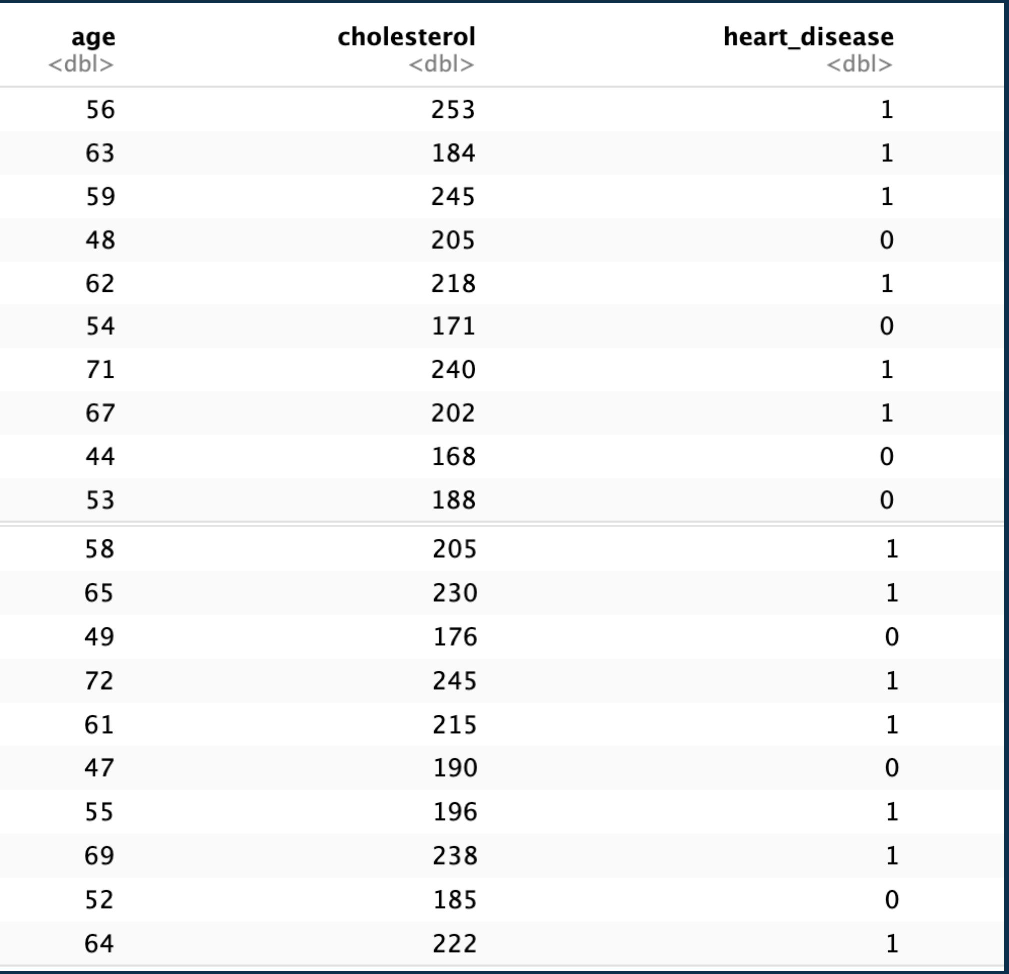
Logistic Regression



🟦 No Heart Disease

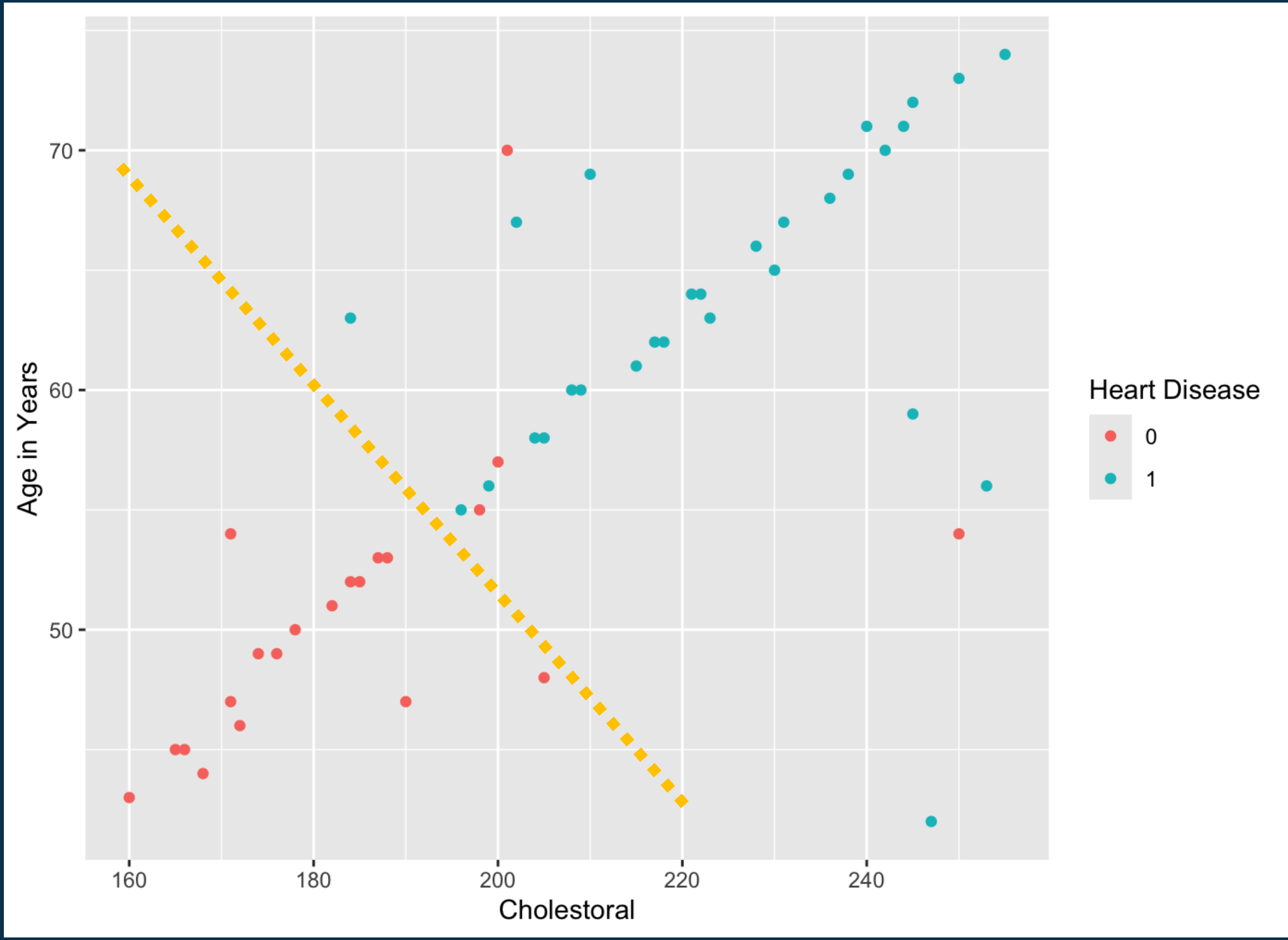
🟥 Hear Disease

50 Subjects



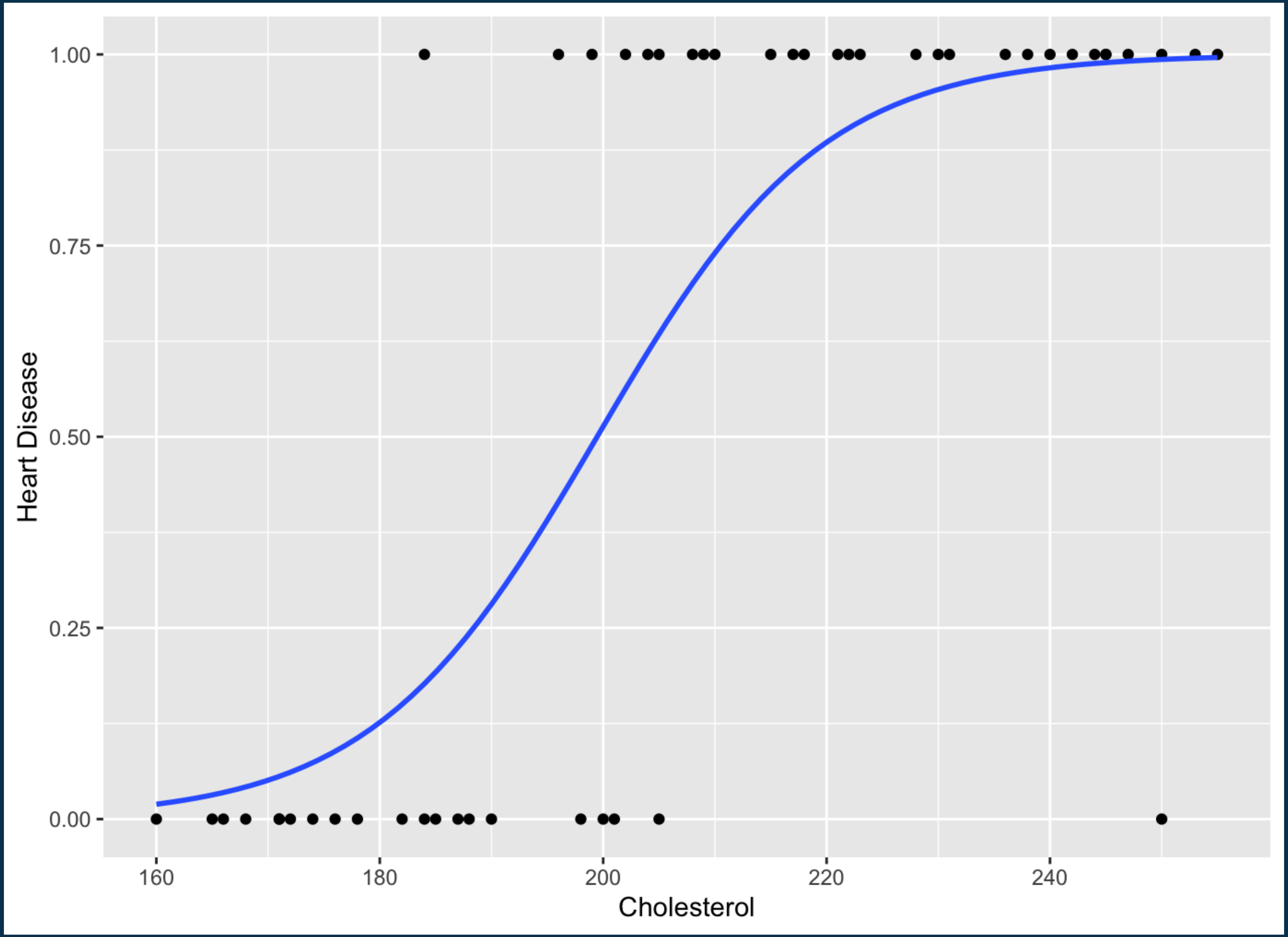
age <dbl>	cholesterol <dbl>	heart_disease <dbl>
56	253	1
63	184	1
59	245	1
48	205	0
62	218	1
54	171	0
71	240	1
67	202	1
44	168	0
53	188	0
58	205	1
65	230	1
49	176	0
72	245	1
61	215	1
47	190	0
55	196	1
69	238	1
52	185	0
64	222	1

And more ….



p – **Probability of Getting a Heart Disease**

$$Log\left[\frac{p}{1-p}\right] = \beta_0 + \beta_1 Cholesterol$$



$$p(x) = \frac{\exp(x)}{1 + \exp(x)}$$

p – Probability of Getting a Heart Disease

$$\text{Logit}\, p = Log\left[\frac{p}{1-p}\right] = \beta_0 + \beta_1 Cholesterol$$

Estimate β_0 and β_1 such that maximizing:

$$L = \sum\{y \ln p + (1-y)\ln(1-p)\}$$

p – **Probability of Getting a Heart Disease**

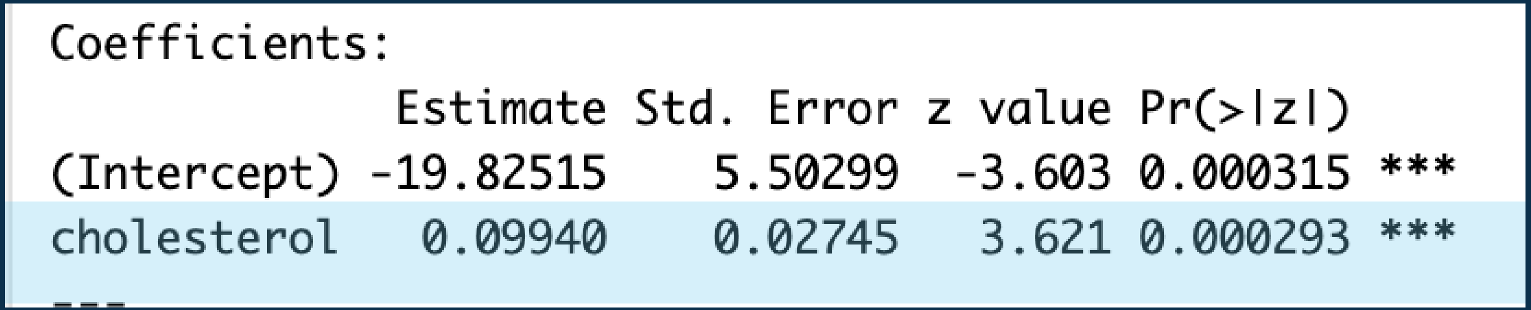
$$\mathbf{Logit}\,\{\mathbf{p}\} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 Cholesterol}$$

$$\mathbf{Logit}\,\boldsymbol{p} = \boldsymbol{Log}\left[\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right] = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 Cholesterol}$$

This is also equivalent to:

$$p = \frac{\exp(\boldsymbol{\beta_0} + \boldsymbol{\beta_1 Cholesterol})}{\mathbf{1} + \exp(\boldsymbol{\beta_0} + \boldsymbol{\beta_1 Cholesterol})}$$

```
logistic_regression = glm(formula = heart_disease ~ cholesterol,
                          data = loan_df,family = 'binomial’)

summary(logistic_regression)
```



```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -19.82515    5.50299  -3.603 0.000315 ***
cholesterol   0.09940    0.02745   3.621 0.000293 ***
---
```

$$logit\ (\hat{p}) = -19.82 + 0.099\ Cholesterol$$

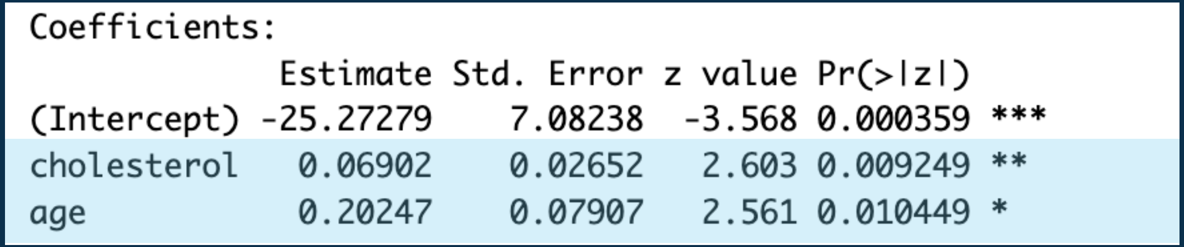
$$\hat{p} = \frac{\exp(-19.82 + 0.099\ Cholesterol)}{1 + \exp(-19.82 + 0.099\ Cholesterol)}$$

$\hat{p}$ - Probability of getting a heart disease given the Cholesterol Level

If $\mathbf{p > 0.5}$ $\rightarrow$ Heart Disease

We can change 0.5 to a different probability threshold.

```
logistic_regression = glm(formula = heart_disease ~ cholesterol + age ,
                          data = heart_df, family = 'binomial')

summary(logistic_regression)
```



```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -25.27279    7.08238  -3.568 0.000359 ***
cholesterol   0.06902    0.02652   2.603 0.009249 ** 
age           0.20247    0.07907   2.561 0.010449 *  
```

$$logit\ (\hat{p}) = -25.3 + 0.069\ Cholesterol + 0.2\ Age$$

$$\hat{p} = \frac{\exp(-25.3 + 0.069\ Cholesterol + 0.2\ Age)}{1 + \exp(-25.3 + 0.069\ Cholesterol + 0.2\ Age)}$$

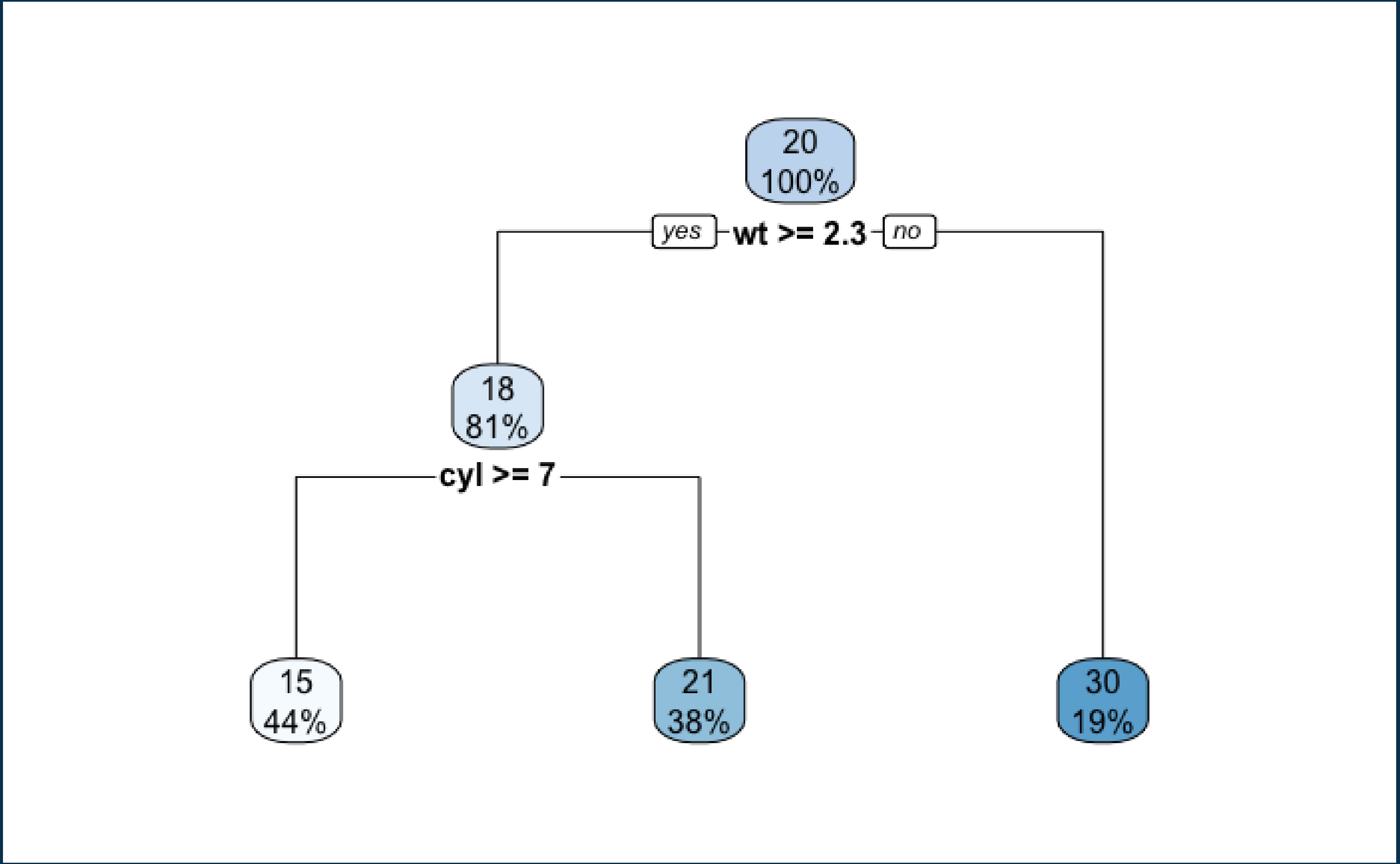
$\hat{p}$ - Probability of approving the loan when income and age is known

- The odds of getting a heart disease will <u>increase by 7.1%</u> (i.e., $\exp(0.069) = 1.071$) when the Cholesterol level increases by 1 unit.

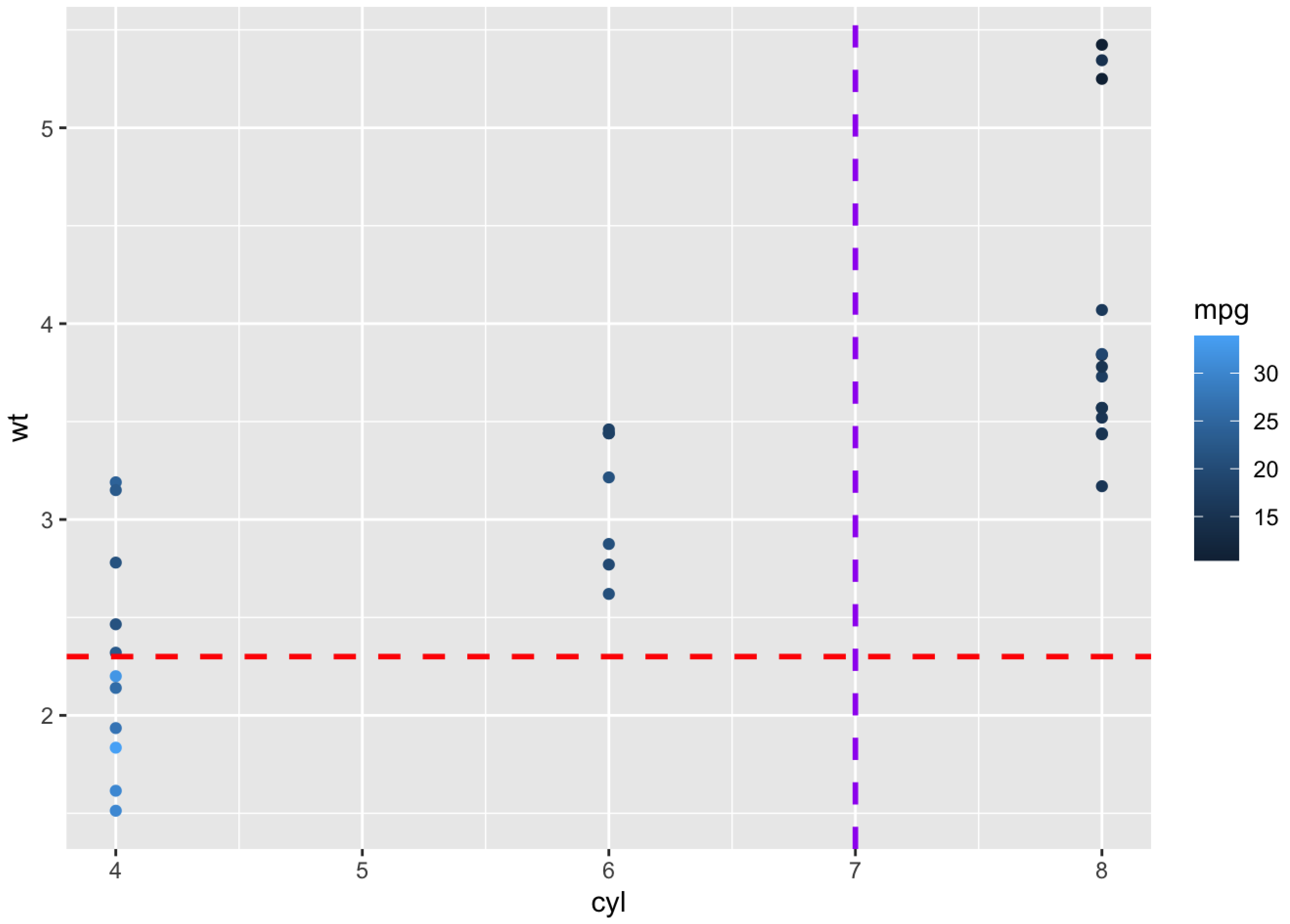
- The odds of getting heart disease will <u>increase by 22%</u> (i.e.,$\exp(0.2) = 1.22$) when the age increases by 1 year.

2. Bagging Methods

Regression Trees (Decision Trees)



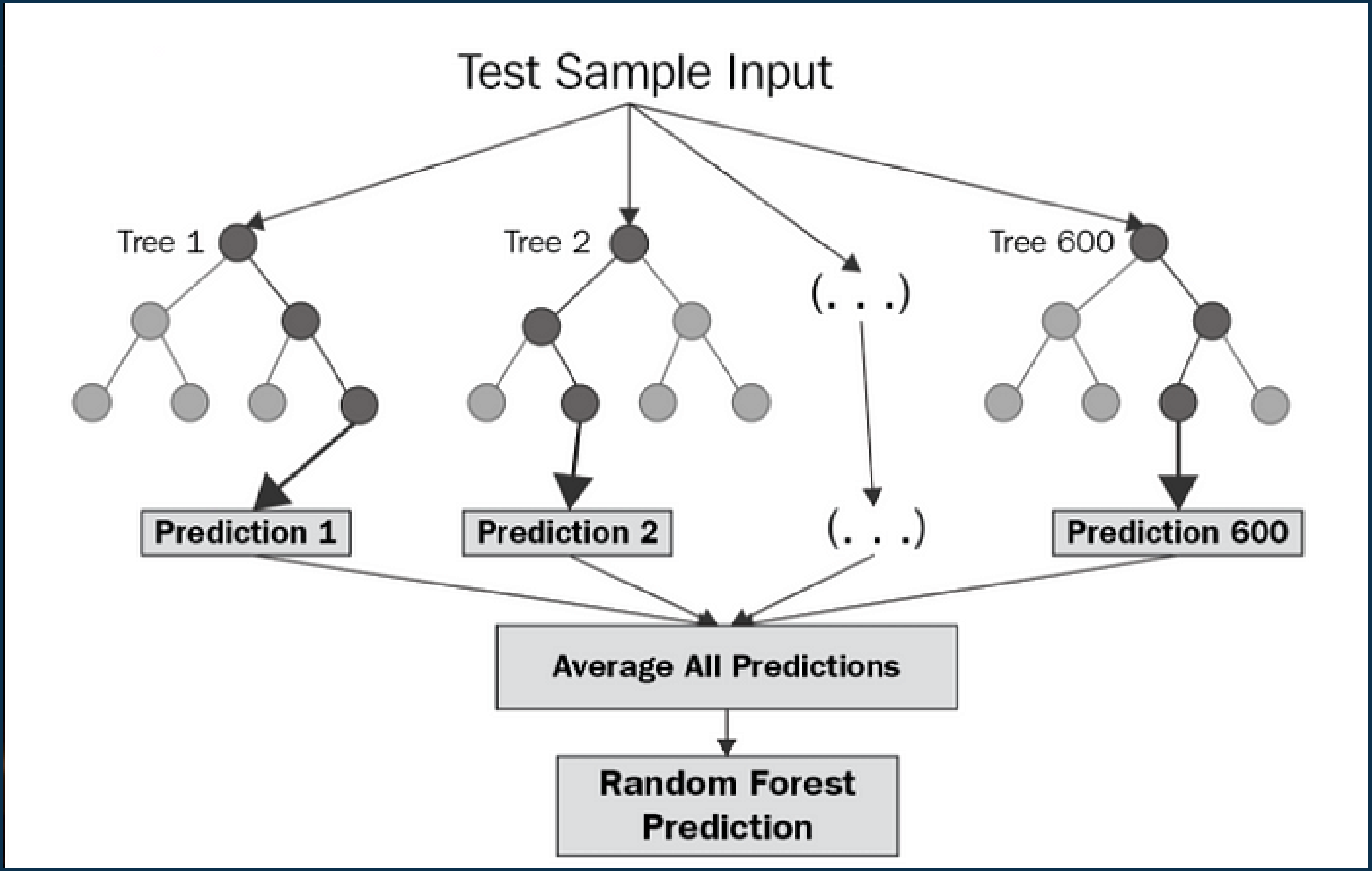
Partition the data set into decision boundaries.


wt
5
4
3
2
cyl
4
5
6
7
8
mpg
30
25
20
15

Regression

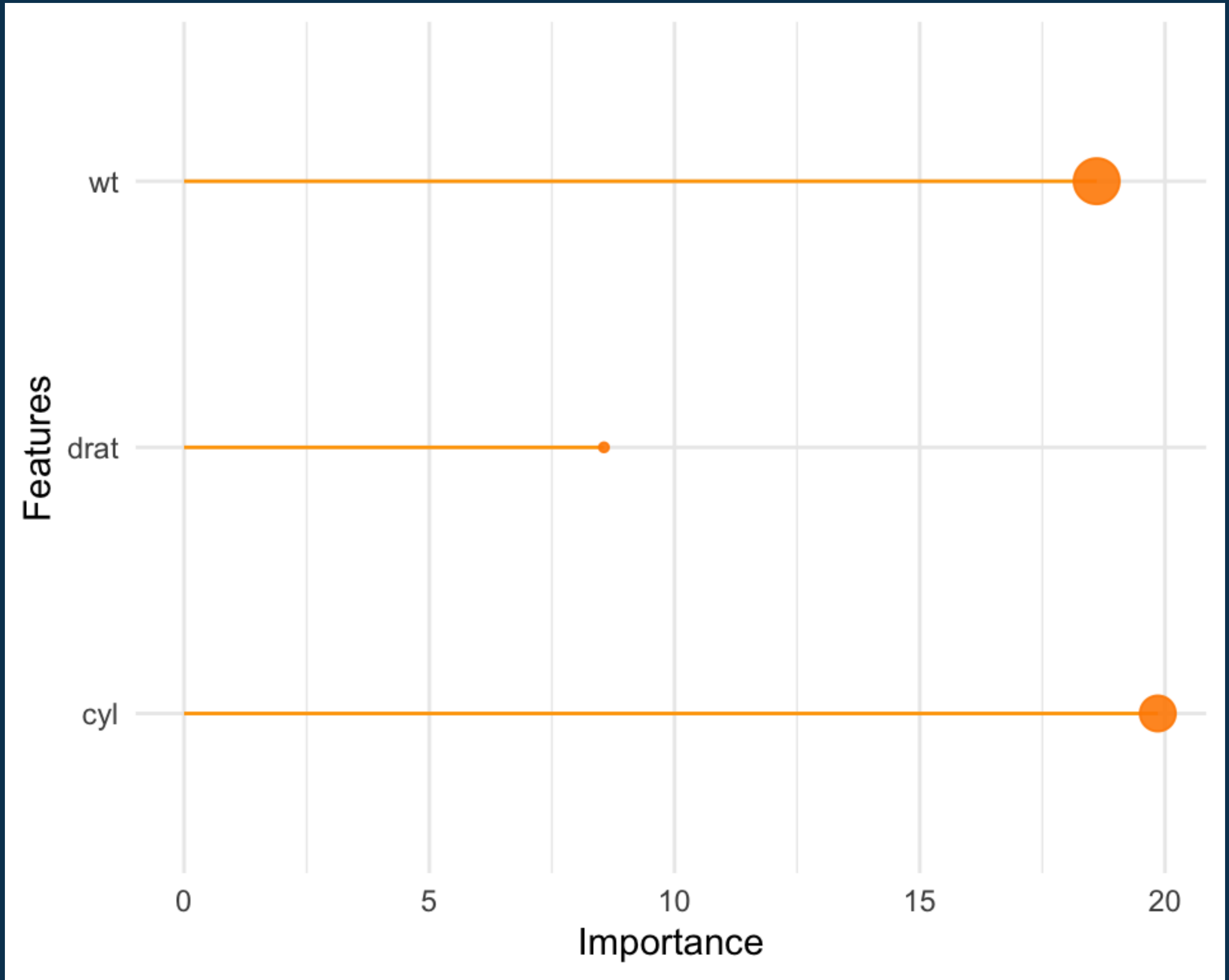
Random Forest Regression

Random Forest Regression use the “Bagging” Technique



Source: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

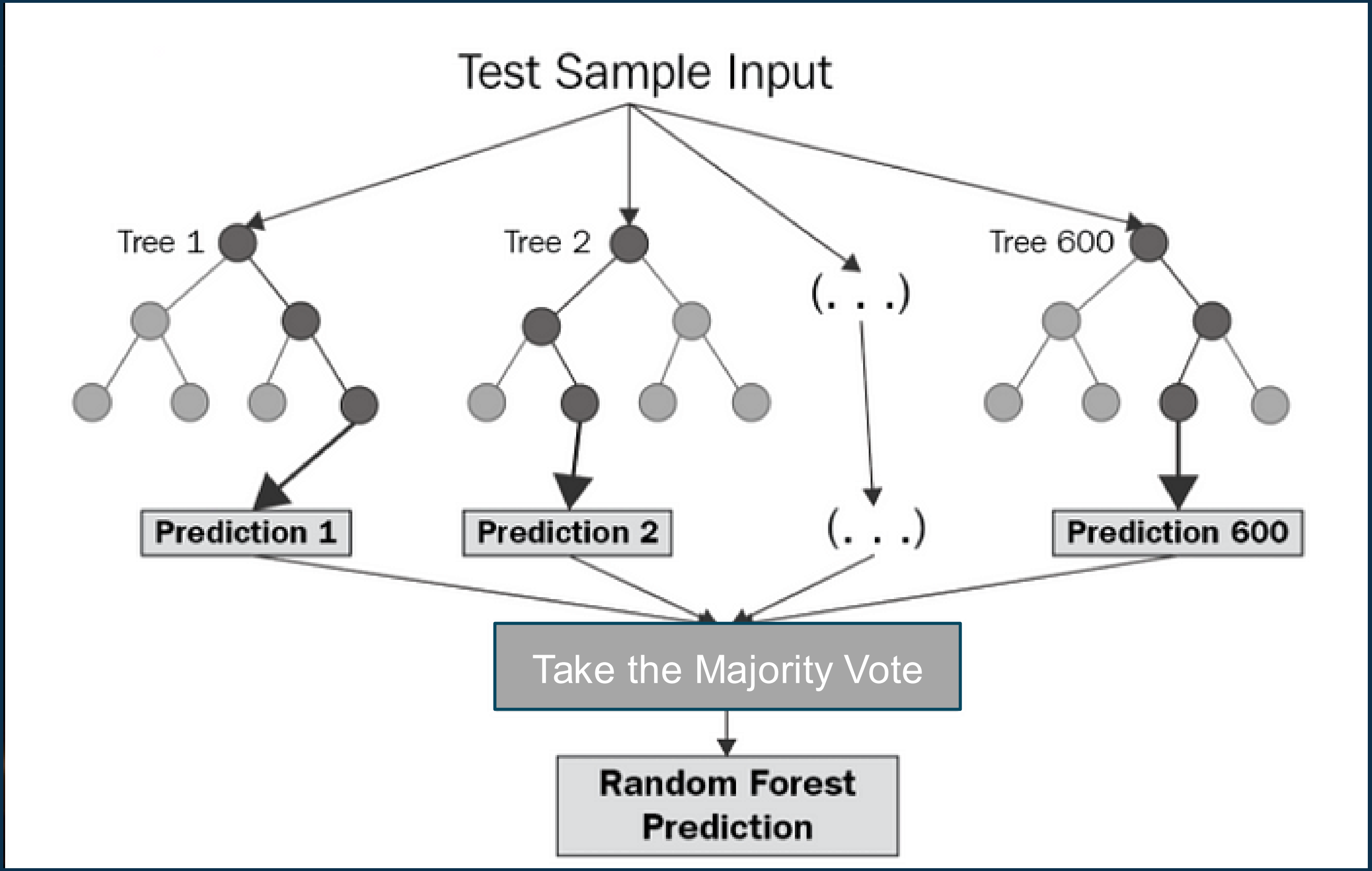
```
library(randomForest)
Random_Forest_model = randomForest(formula = mpg ~ wt + cyl + drat,
                                                            data = mtcars,
                                                            ntree=500, importance=TRUE)
```



Classification

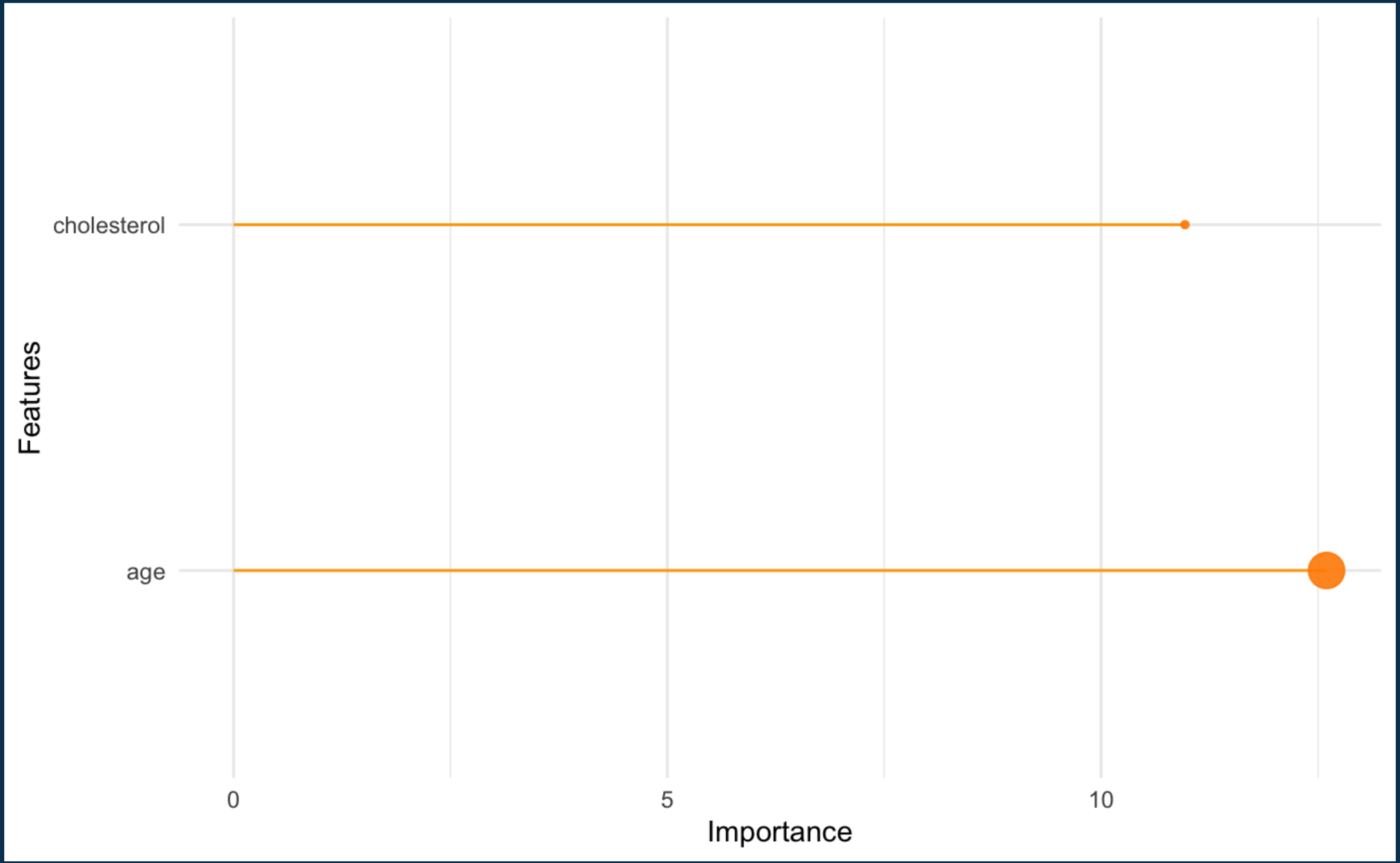
Random Forest Classification

Random Forest Classification



Source: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

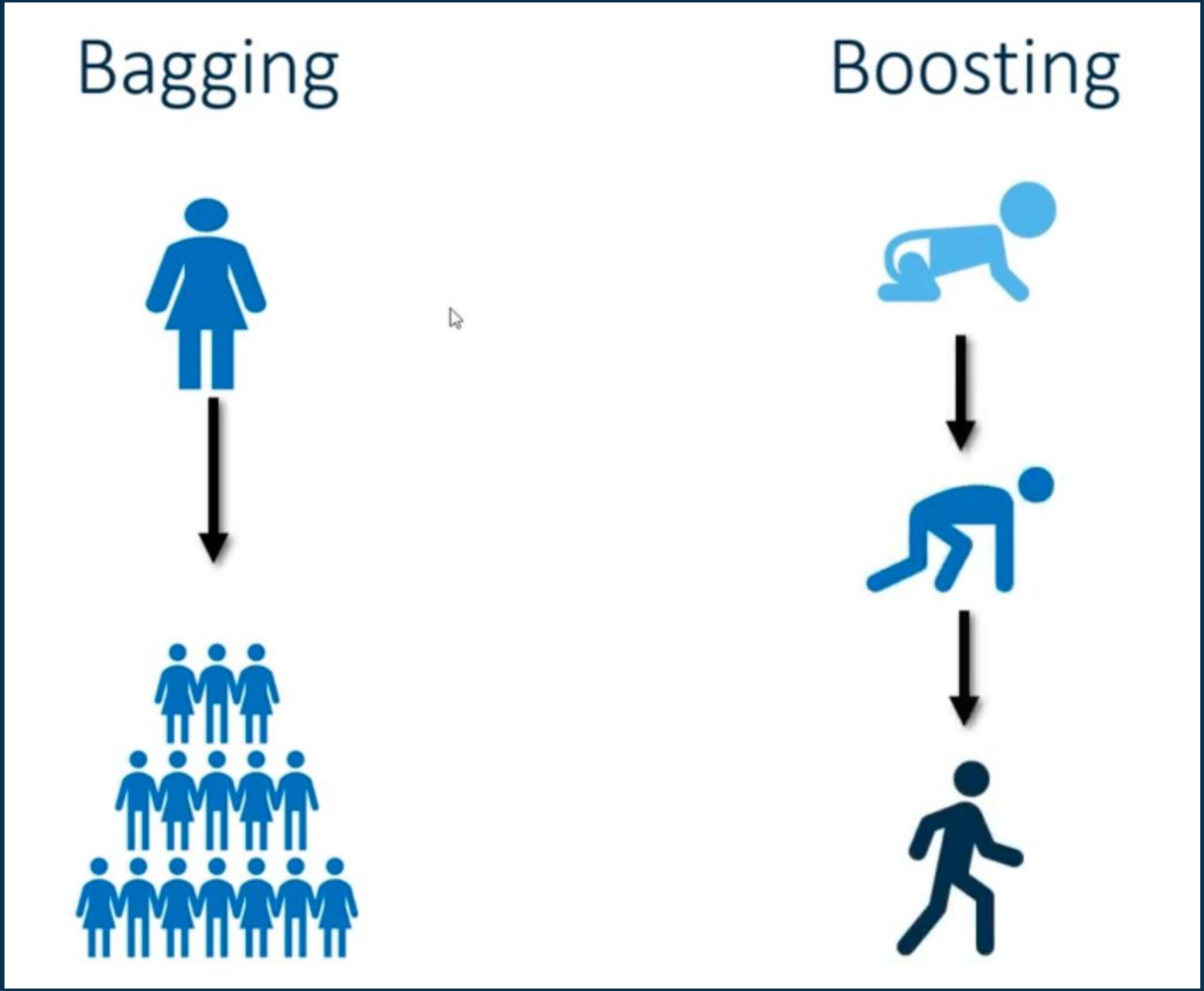
```r
library(randomForest)

Random_Forest_Classifier = randomForest(as.factor(heart_disease) ~ cholesterol + age,
                                        data=heart_df, importance = TRUE, ntree=500)
```



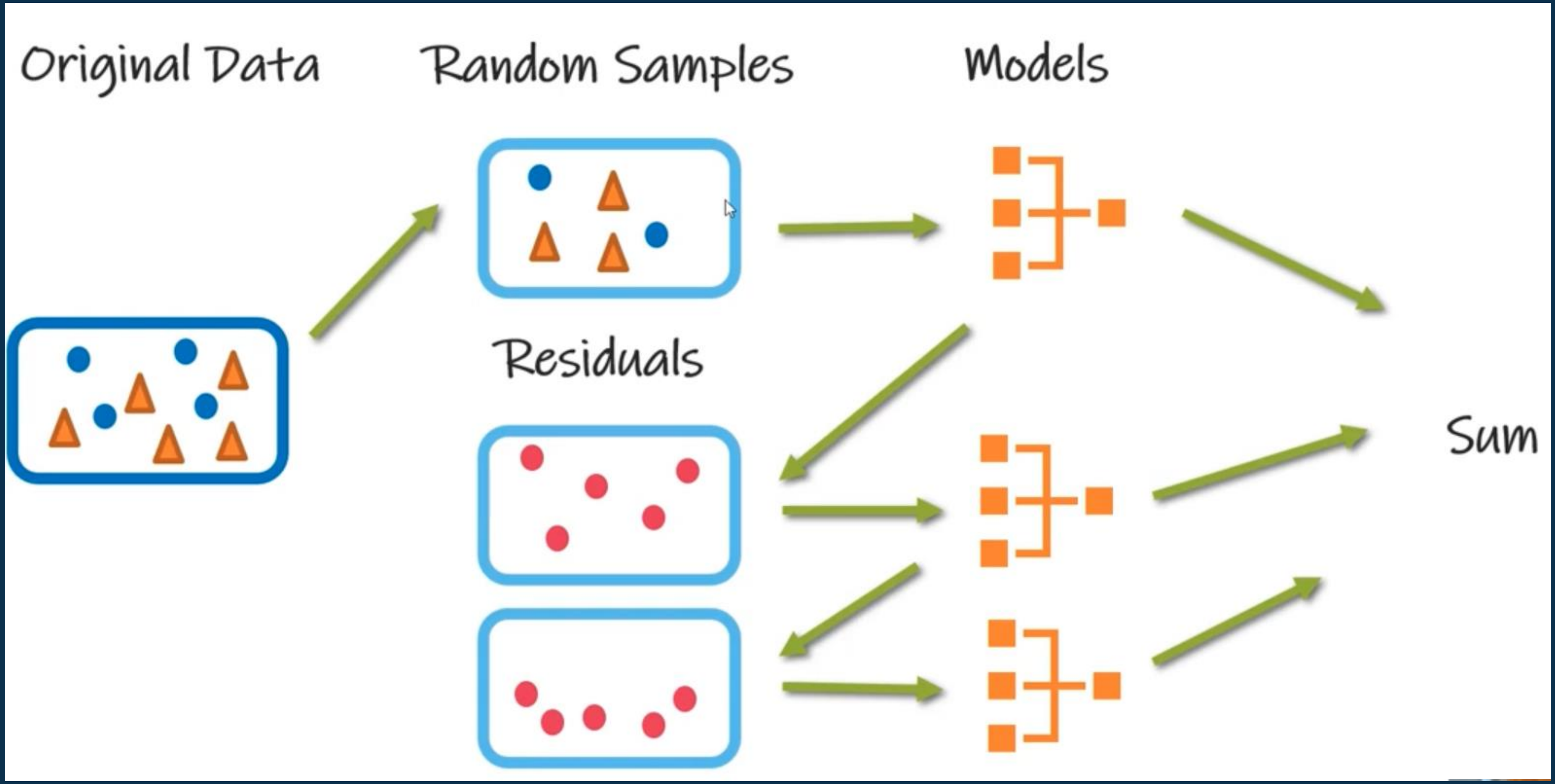
3. Boosting Methods

Trees are built in parallel

Trees are built in sequentially



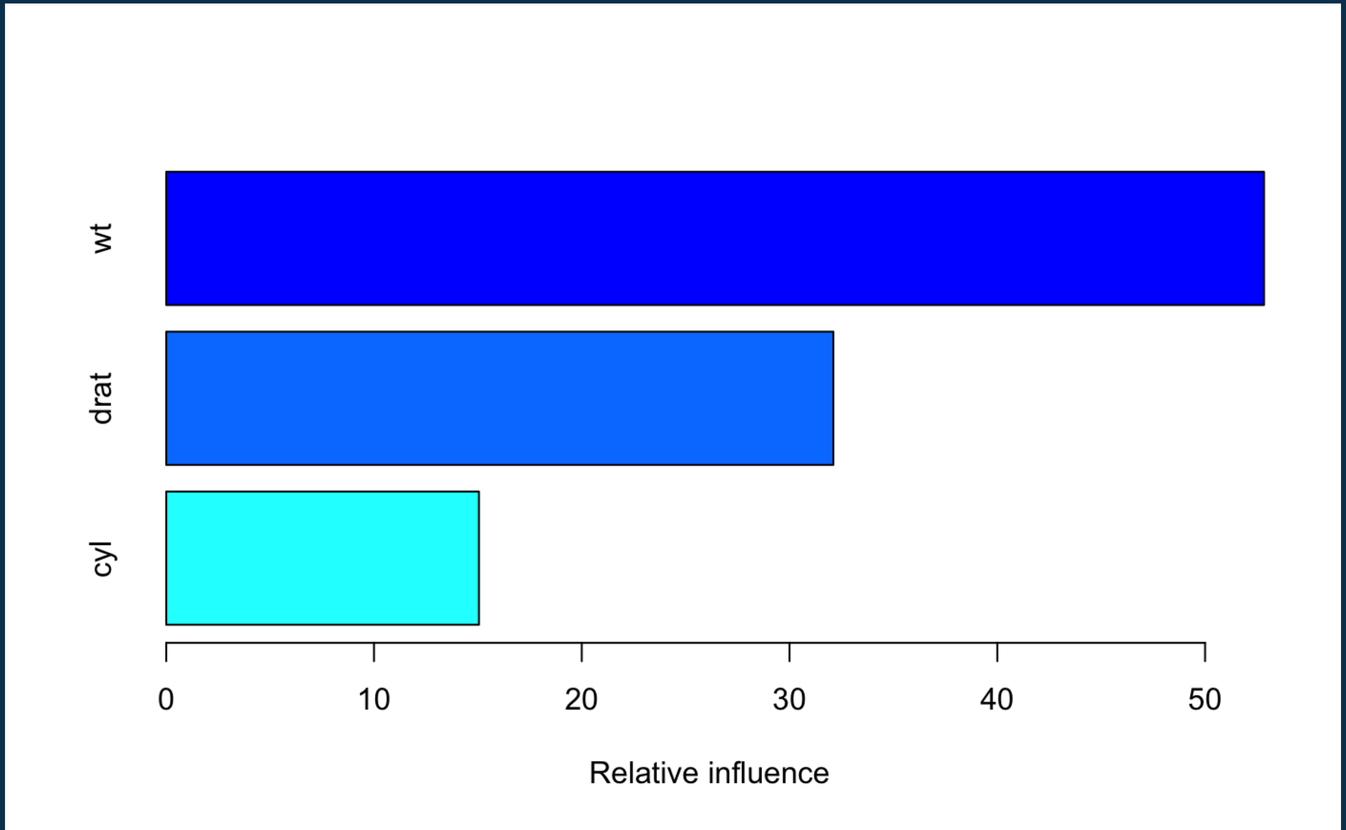
Source: SAS.com



Source: SAS.com

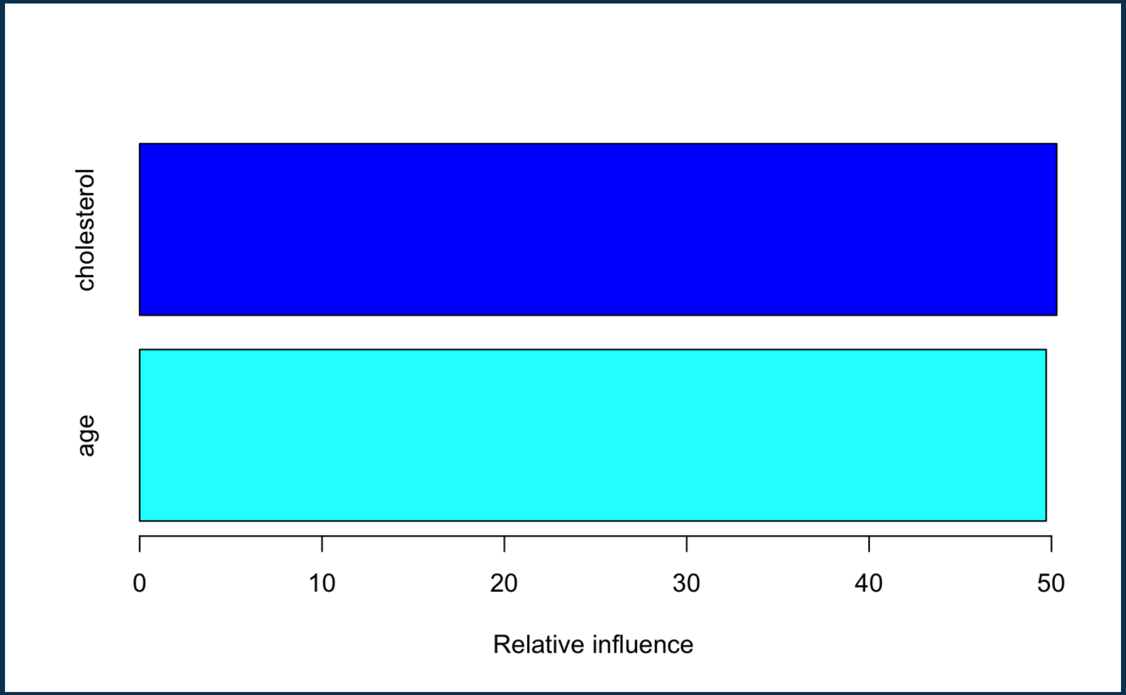
Gradient Boost Regression

```
library(gbm)

Gradient_Boost_Regression = gbm(formula = mpg ~ wt + cyl + drat,
                                data = mtcars, n.trees = 2000,
                                n.minobsinnode = 5
                               )
```



Gradient Boost Classification

```
library(gbm)

Gradient_Boost_Classifier = gbm(heart_disease ~ cholesterol + age,
                                data=heart_df,
                                n.trees = 2000)
```



Performance Metrices in Supervised Learning

Evaluate Model Performance in **Regression**

Coefficient of Determination: $0 \leq \mathrm{R}^2 \leq 1$

$$-\infty < \mathrm{R}^2_{\mathrm{adjusted}} \leq 1$$

$$Mean\ Squared\ Error = MSE = \frac{1}{\mathrm{N}} \sum \left(Y_i - \hat{Y}_i\right)^2$$

$$Root\ Mean\ Squared\ Error = RMSE = \sqrt{\frac{1}{\mathrm{N}} \sum \left(Y_i - \hat{Y}_i\right)^2}$$

Y – Observed Value
$\hat{Y}$ – Predicted Value
N – Total Number of Observations

Evaluate Model Performance in **Classification**

Confusion Matrix

Actual Classes

Predicted Classes

	0	**1**
0	True Negative	False Negative
1	False Positive	True Positive

$$\text{Accuracy} = \frac{TP + TN}{N}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

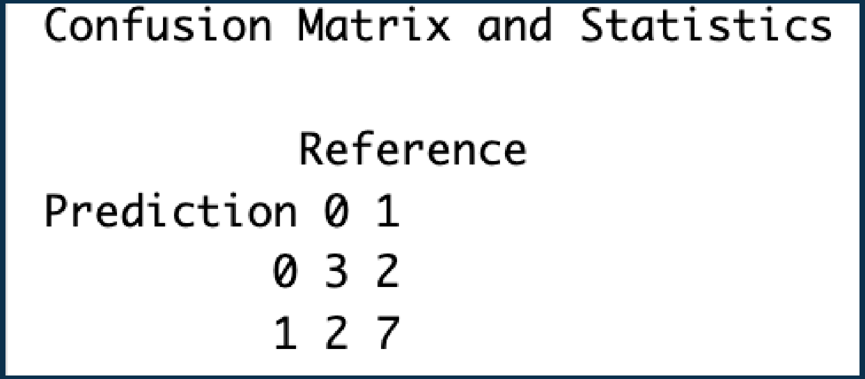
$$\text{Sensitivity} = \text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = 1 - \text{False Positive Rate} = \frac{TN}{TN + FP}$$

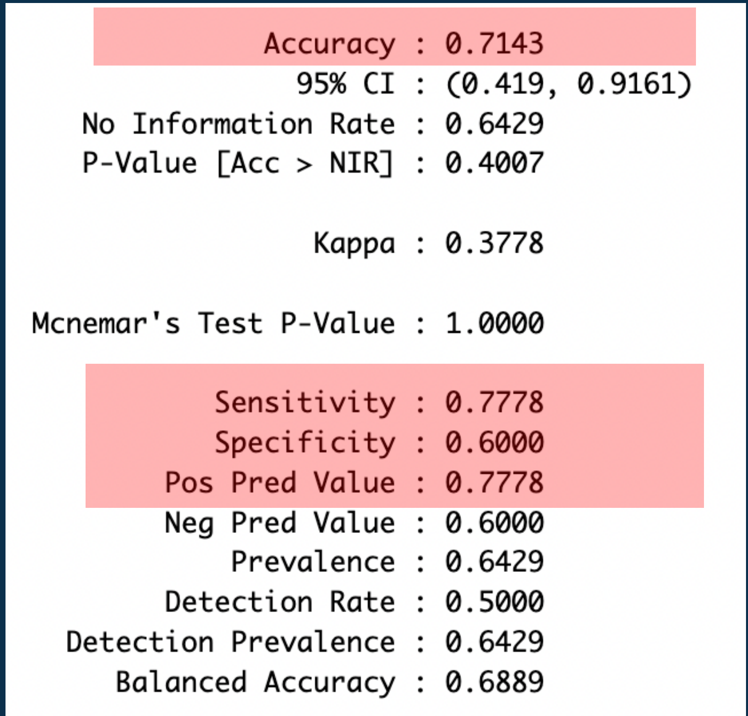
N – Total Number of Observations

Evaluate Model Performance in **Classification**

```
library(caret)

random_forest_prediction = predict(Random_Forest_Classifier)

observed_data = as.factor(heart_df$heart_disease)

confusionMatrix( random_forest_prediction, observed_data, positive = '1')
```



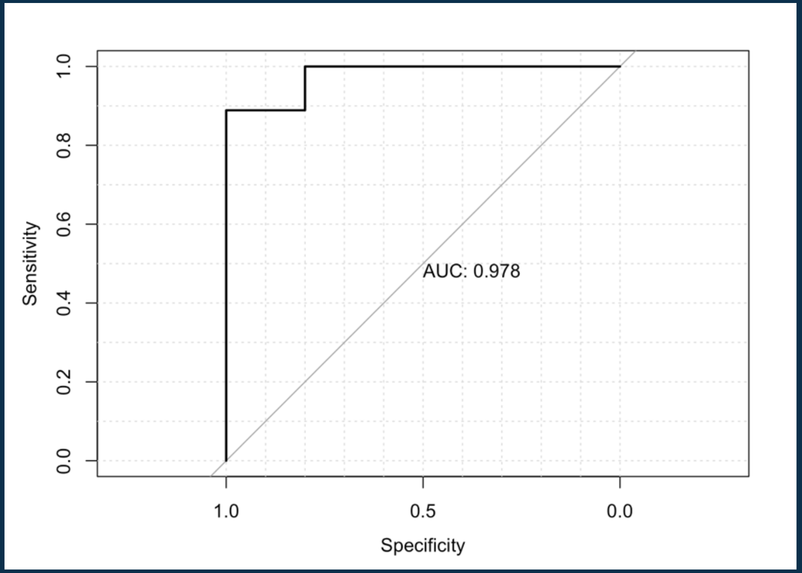
```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 3 2
         1 2 7
```



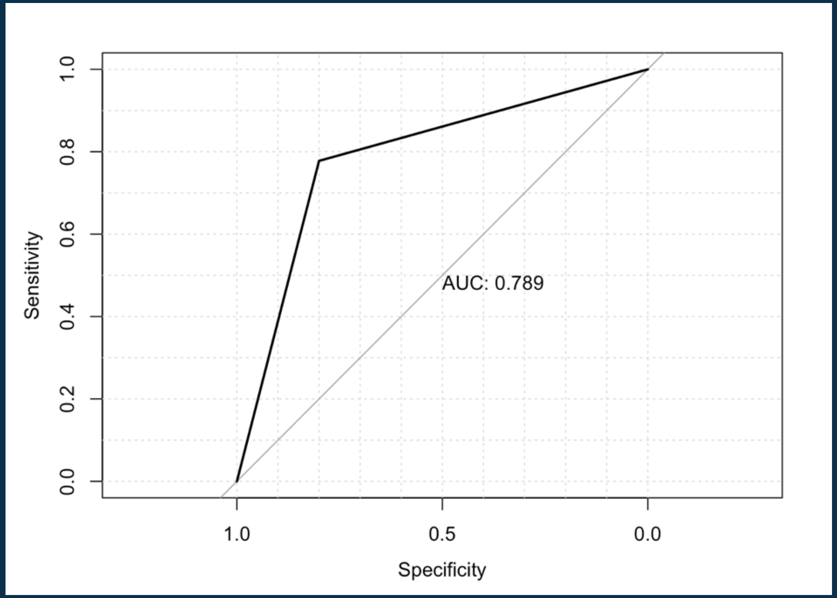
```
               Accuracy : 0.7143
                 95% CI : (0.419, 0.9161)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : 0.4007

                  Kappa : 0.3778

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.7778
            Specificity : 0.6000
         Pos Pred Value : 0.7778
         Neg Pred Value : 0.6000
             Prevalence : 0.6429
         Detection Rate : 0.5000
   Detection Prevalence : 0.6429
      Balanced Accuracy : 0.6889
```

Evaluate Model Performance in **Classification**

Receiver Operating Characteristic (ROC) Curve



Logistic Regression



Random Forest Classifier

- ROC curve is a plot between the Sensitivity and Specificity of a classification model
- If the area under the curve is 0.5, we are making random guesses
- If the area under the curve is close to 1, we are making accurate predictions

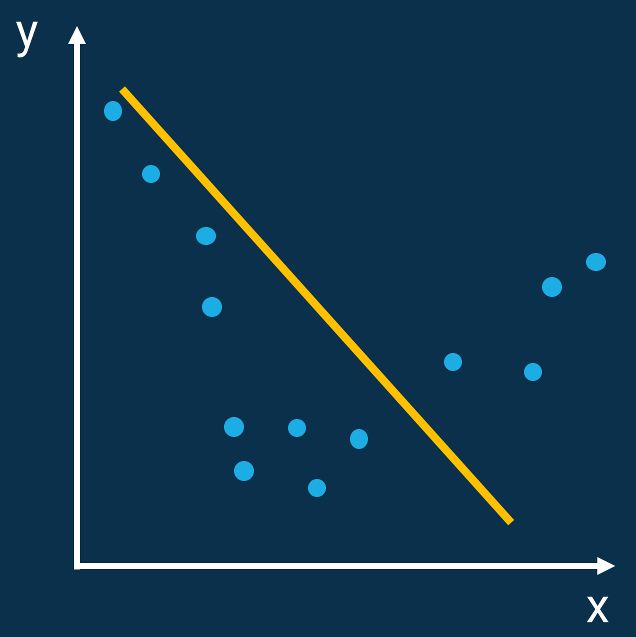
$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

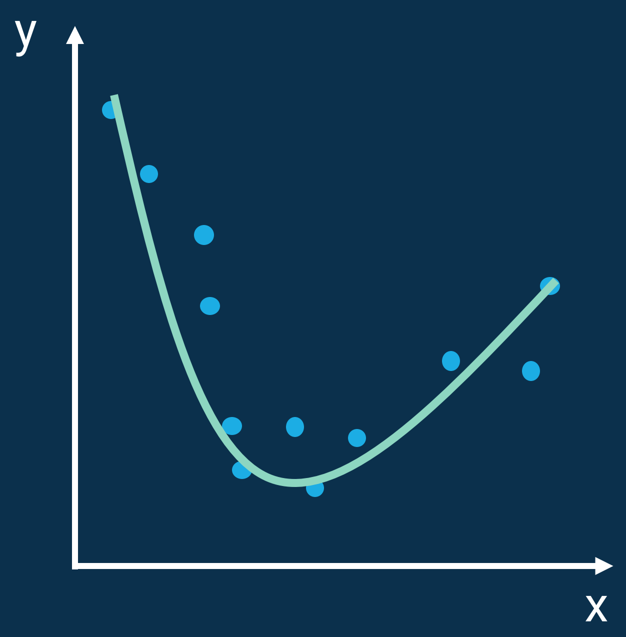
Method Type	Examples	Advantages	Disadvantages
Linear Models	Linear Regression Logistic Regression	- Easy to interpret - Works well for simple, linear patterns	- Struggles with nonlinear relationships - Sensitive to outliers - Requires assumptions (linearity, independence, multicollinearity)
Bagging Methods	Random Forest	- Handles nonlinear patterns - Robust to noise/outliers	- Less interpretable - Can be slow for large data
Boosting Methods	Gradient Boosting XGBoost GBM	- Handles complex nonlinearities - Good for small-to-medium datasets	- Can overfit if not tuned - Less interpretable - Slower training

Overfitting & Cross-Validation

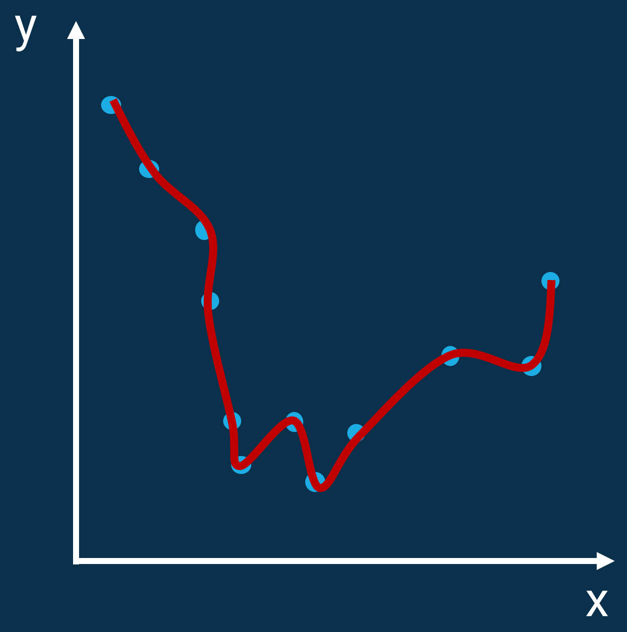
Overfitting in Regression



Underfit
– less complex model

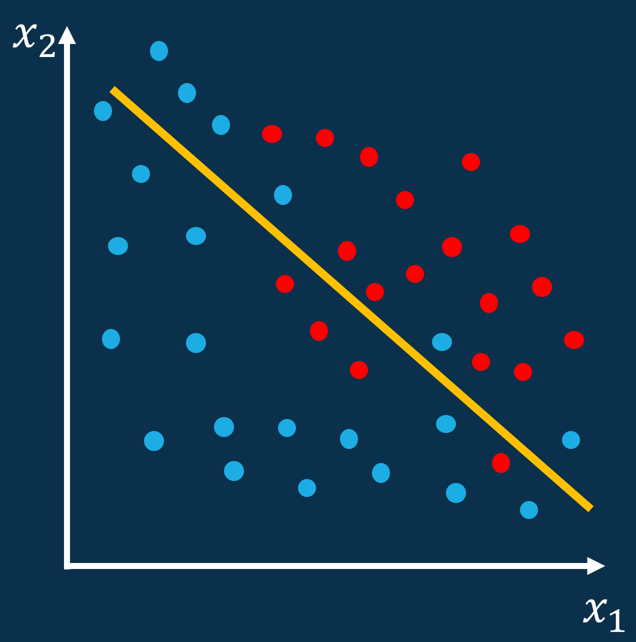


Better Fit
- moderately complex model

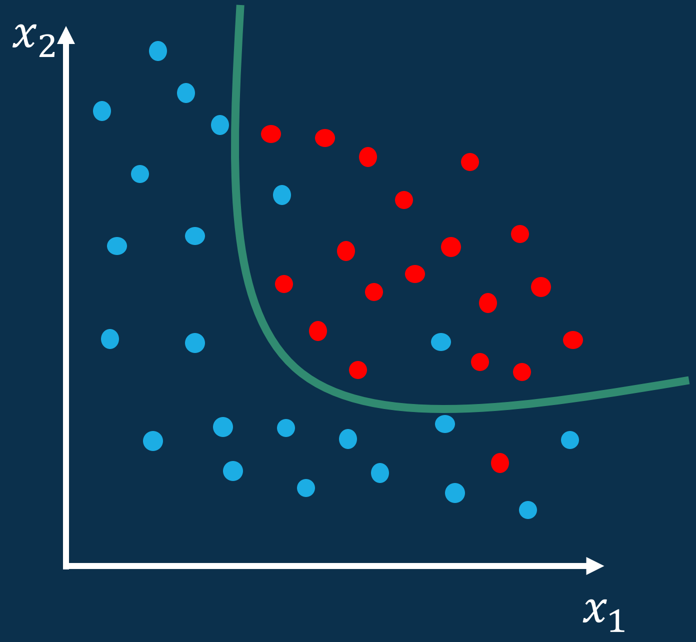


Overfit
– high complex model

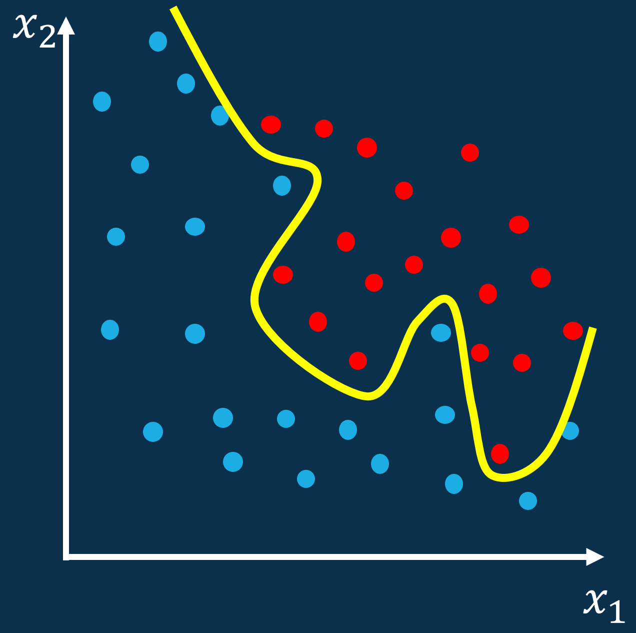
Overfitting in Classification



Underfit
– less complex model



Better Fit
- moderately complex model



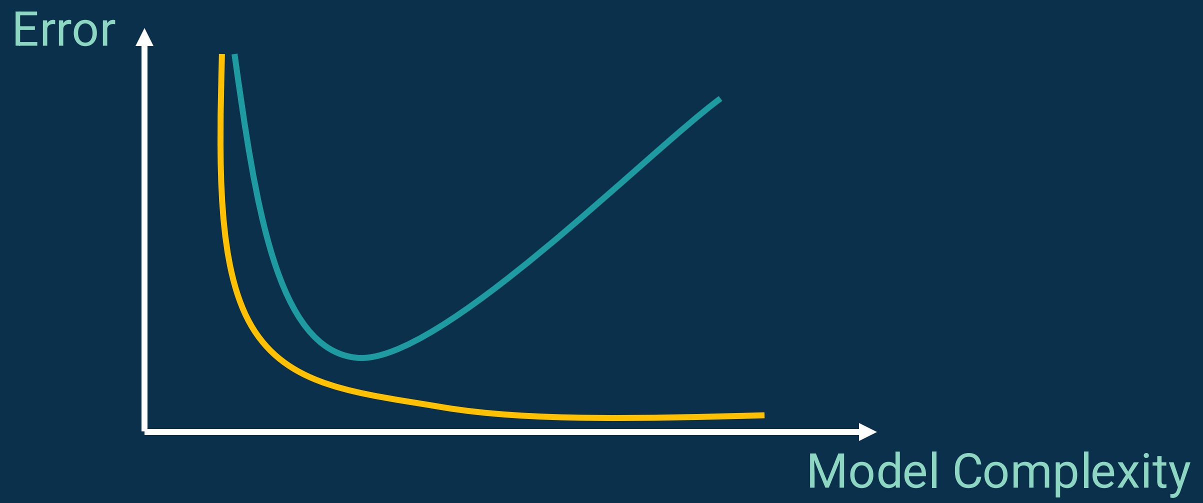
Overfit
– high complex model

How to Overcome Overfitting

- Train/Test Data Split
- Regularization
- Cross Validation

Train/Test Data Split

Training Data	Test Data



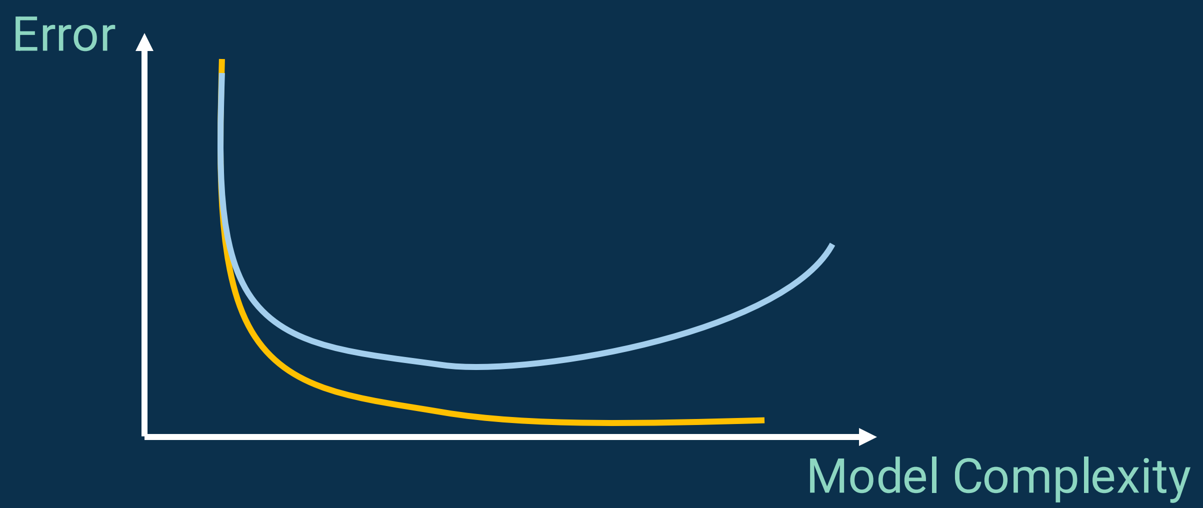
🟨 Training Error
🟦 Test Error

Error = Actual - Predicted

Regularization

Total Error = Training Error + Regularization

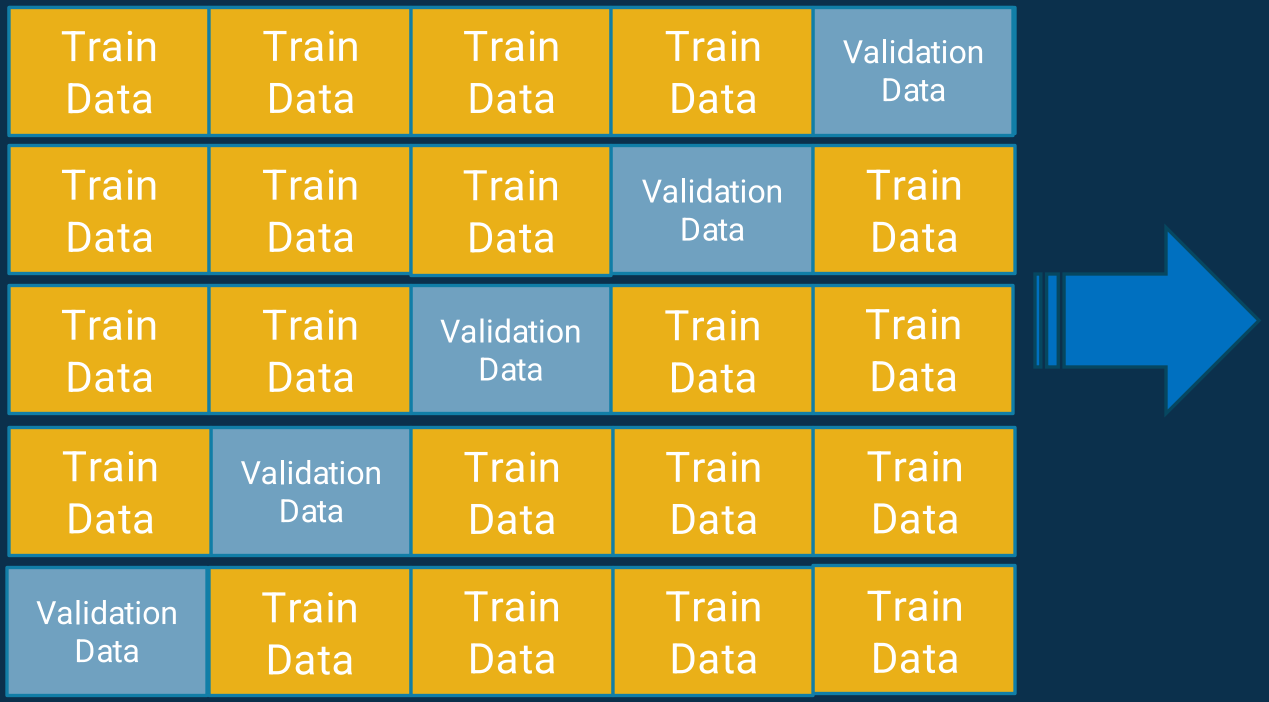
Regularization is proportional to the model complexity and prevents overfitting
Regularization prevents the training error from going to zero



🟨 Training Error
🟦 Training Error with Regularization

Cross-validation

K-Fold Cross-Validation

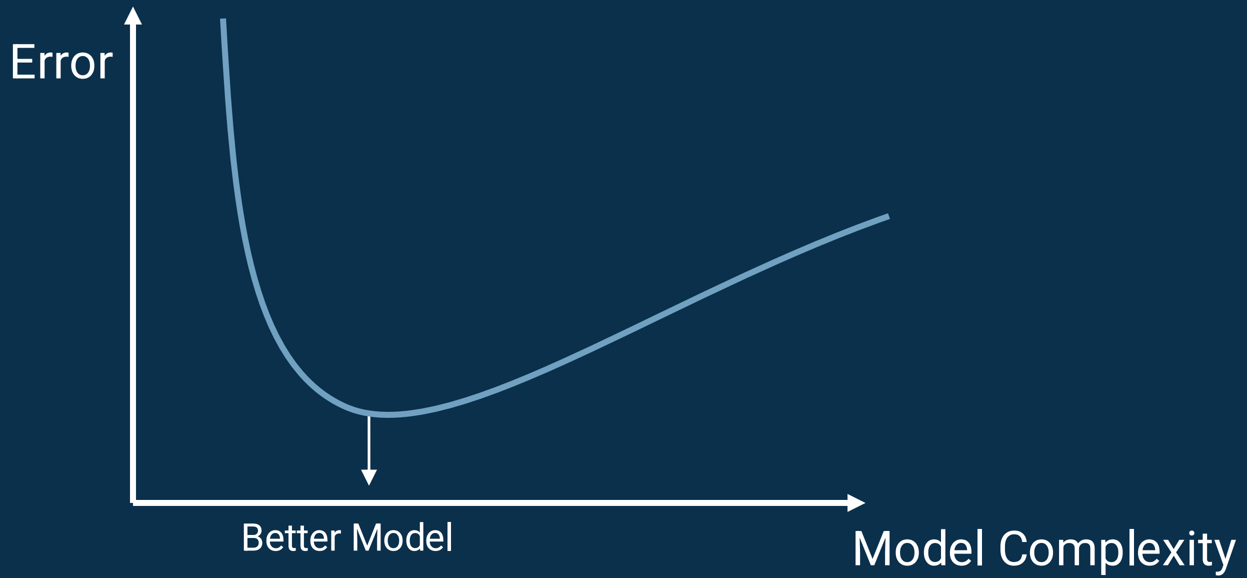


K-Fold Cross Validation Error

Take average of all Validation Errors

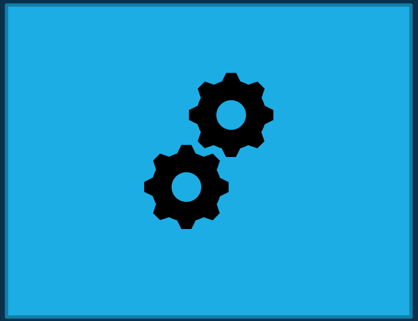
Cross-validation

K-Fold Cross-validation



K-Fold Cross Validation Error

Cross-validation: Hyperparameter Tuning



Equipment 1 Settings

Equipment 2 Settings

Model 1 Settings

Model 2 Settings

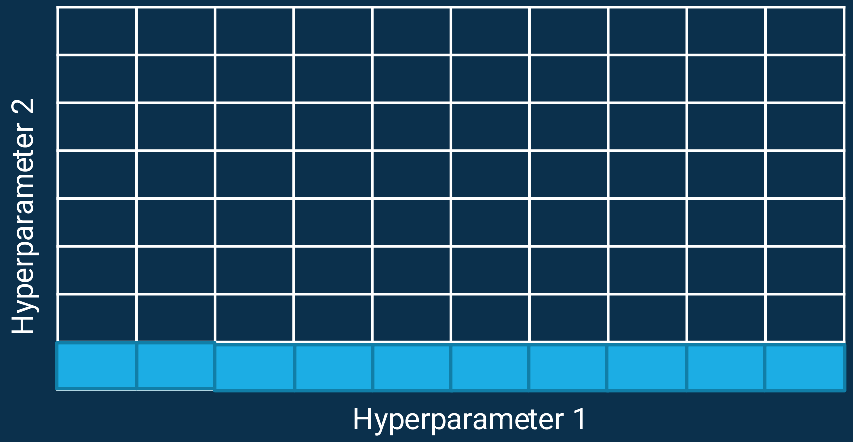
Cross-validation: Hyperparameter Tuning

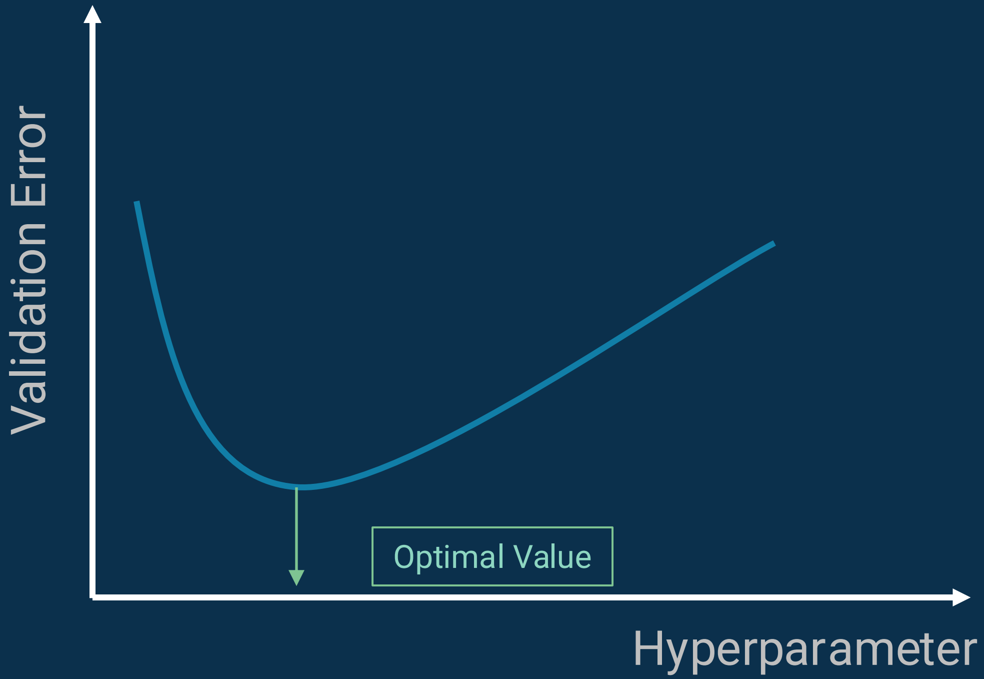
Hyperparameters

- Model specific parameters which can not be estimated with data
- Example: ntrees and mtry in Random Forest

Grid Search

- Require to find optimal values with a search





Training Data	Use to Train Model
Validation Data	• Use to control overfit • Use to tune model parameters
Test Data	Test model with unseen data

Session 1: Part 2

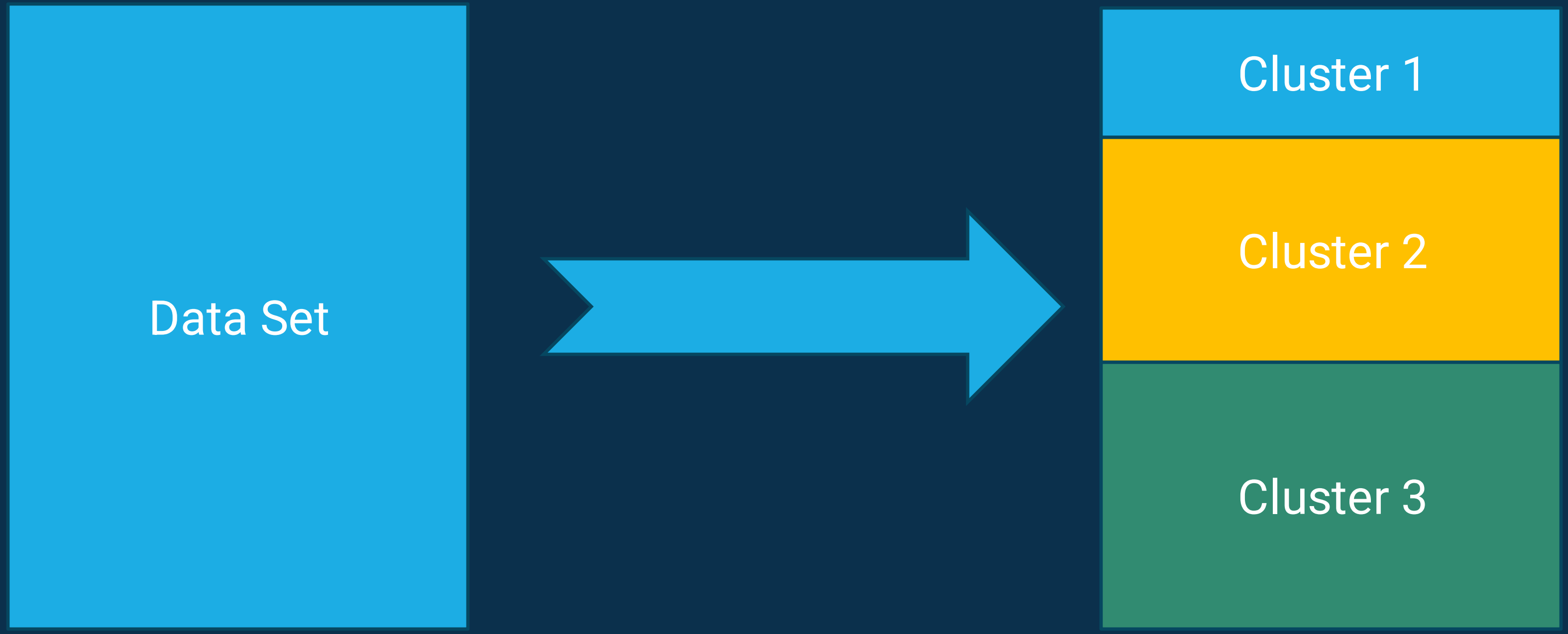
Unsupervised Learning

Workshop on Quantitative Literacy and Statistics

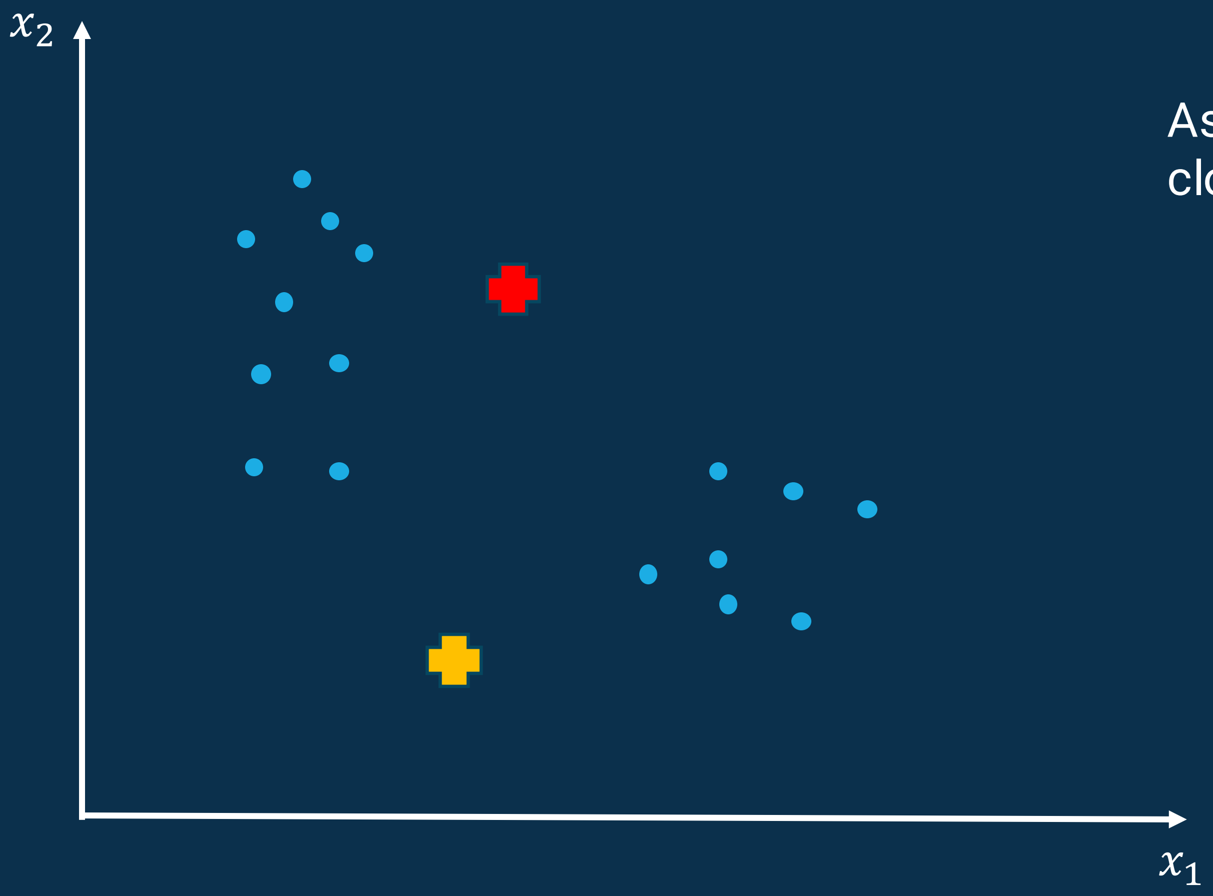
UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES

Unsupervised Learning

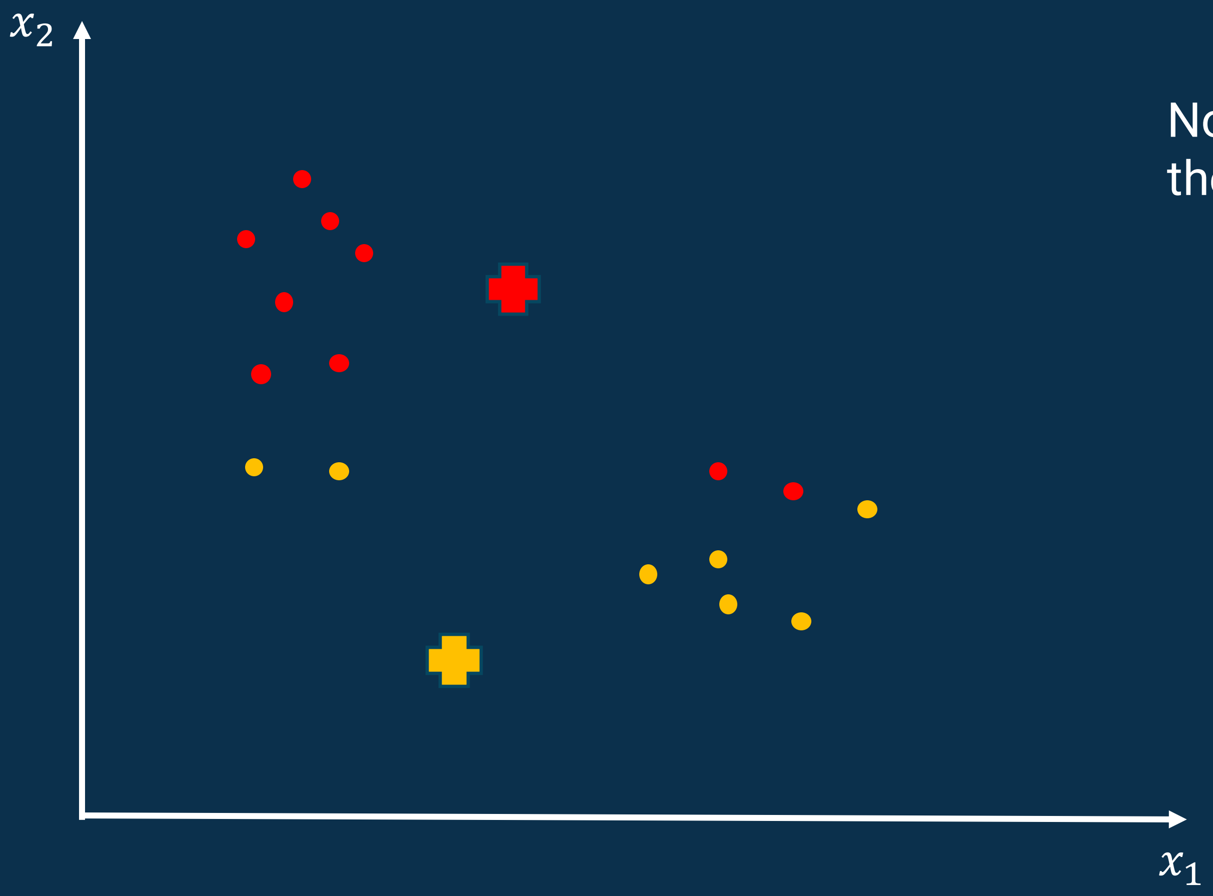
Learning techniques to find a patterns and cluster unlabeled data



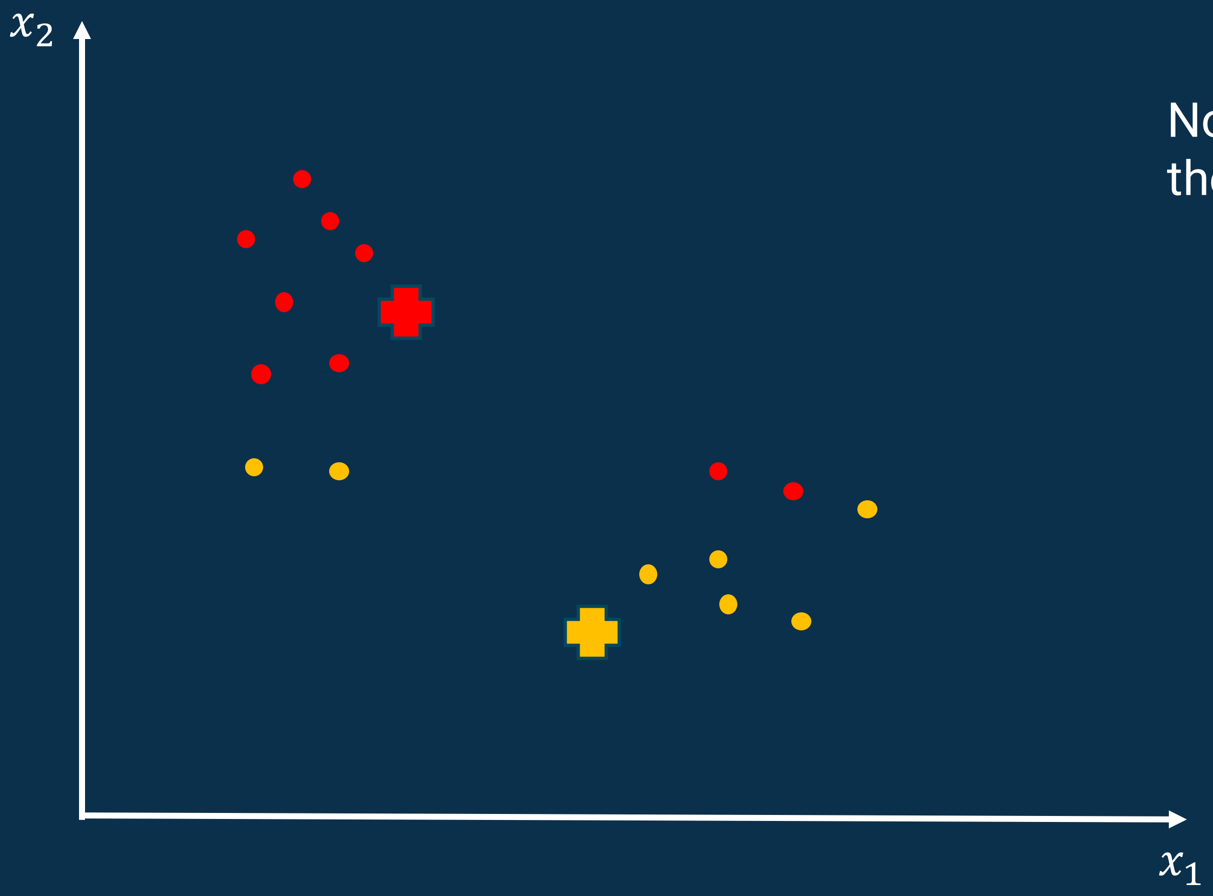
K-Means Clustering



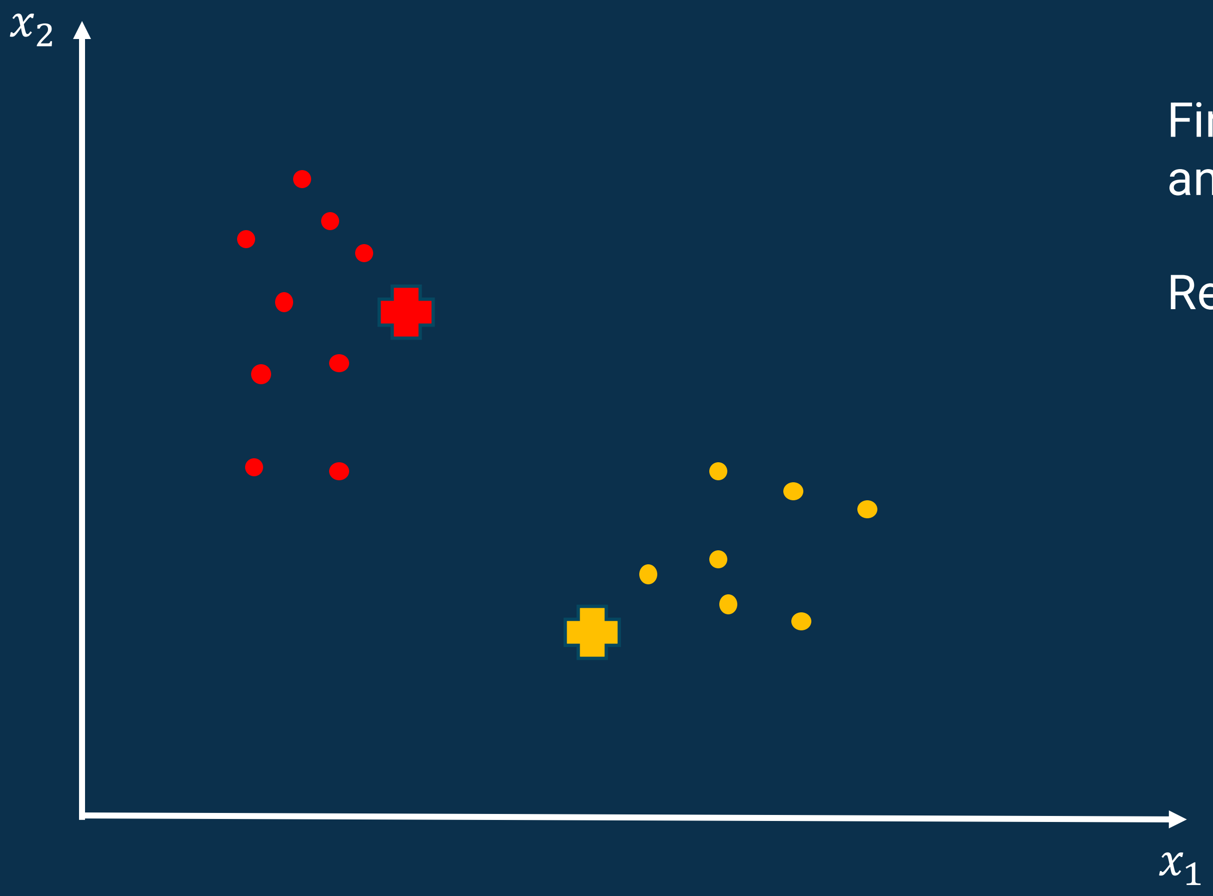
Assign data points for the closest centroid



Now update the center of the centroids

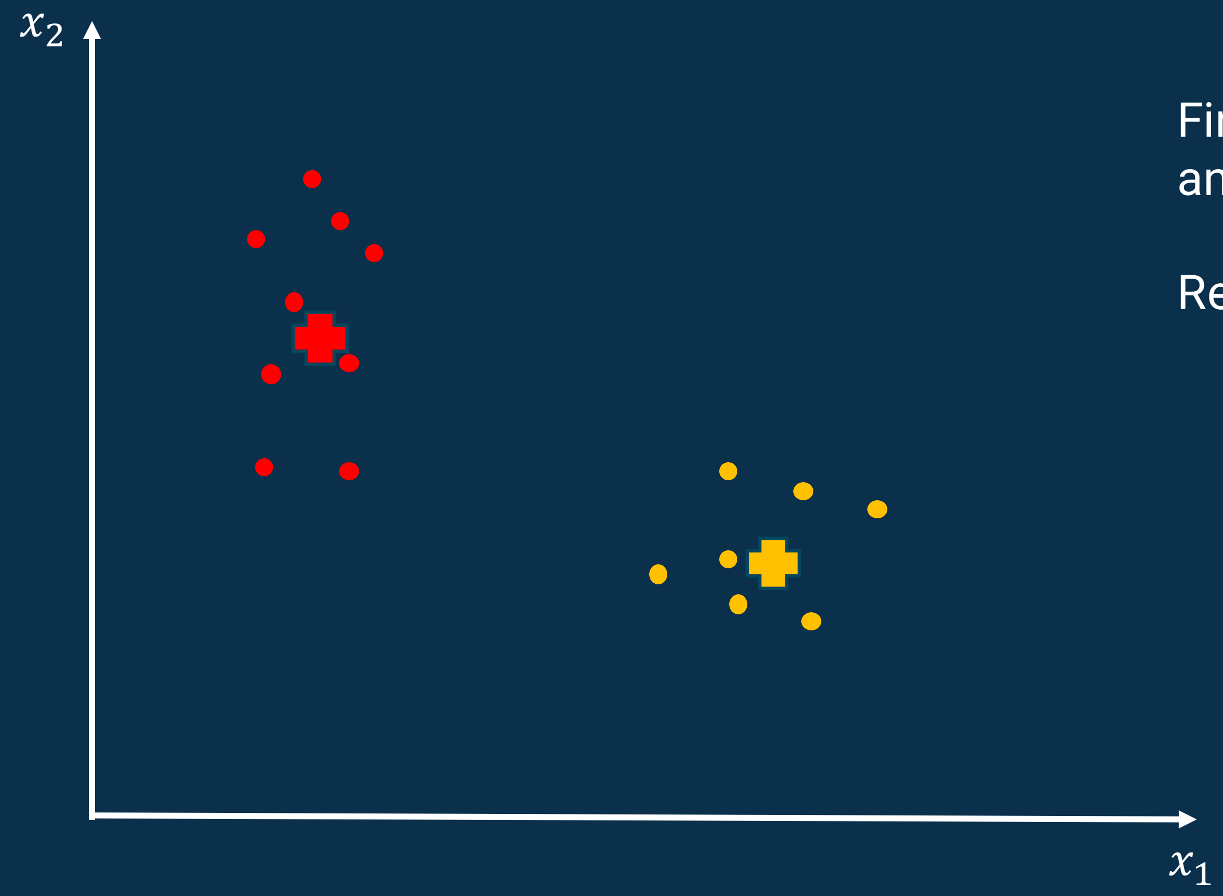


Now update the center of the centroids



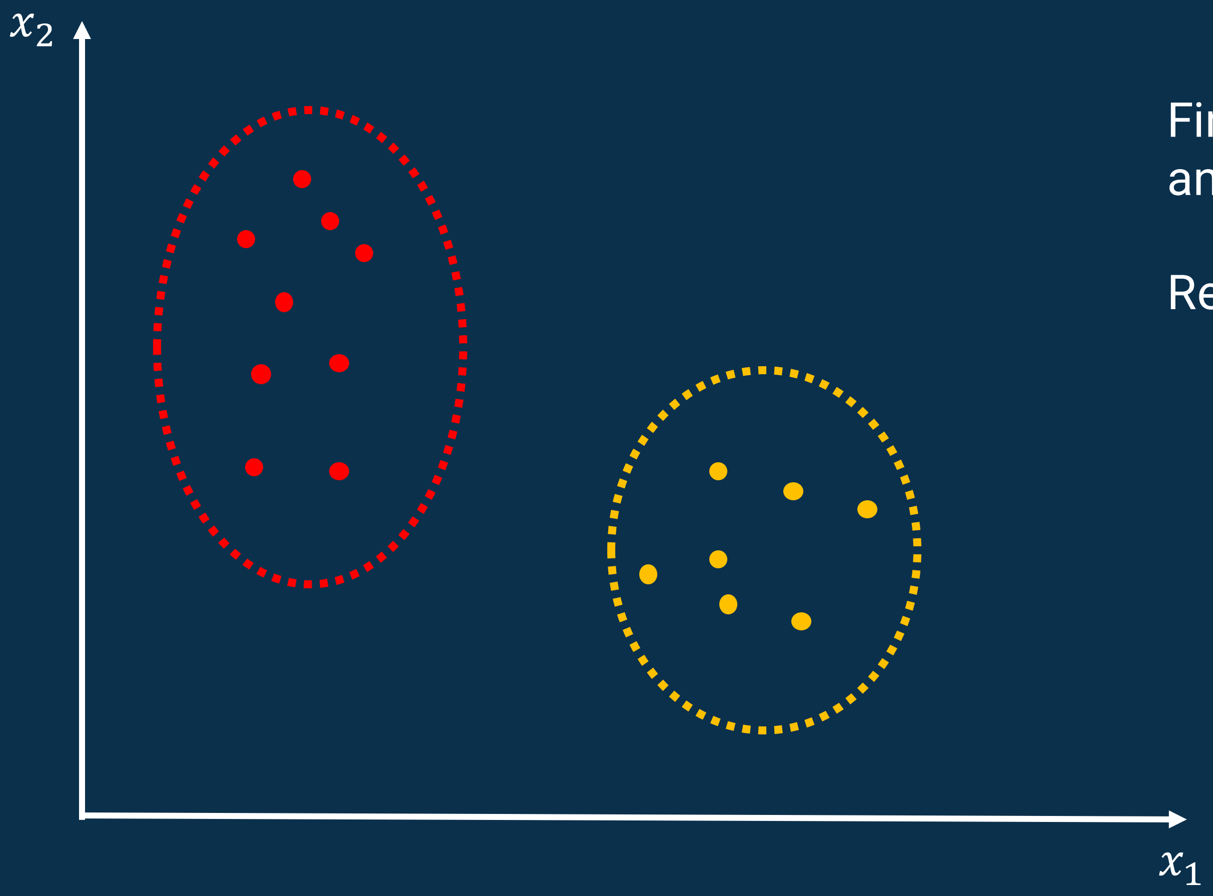
Find the the closest points and and update the centroid

Repeat the process



Find the the closest points and and update the centroid

Repeat the process



Find the the closest points and and update the centroid

Repeat the process

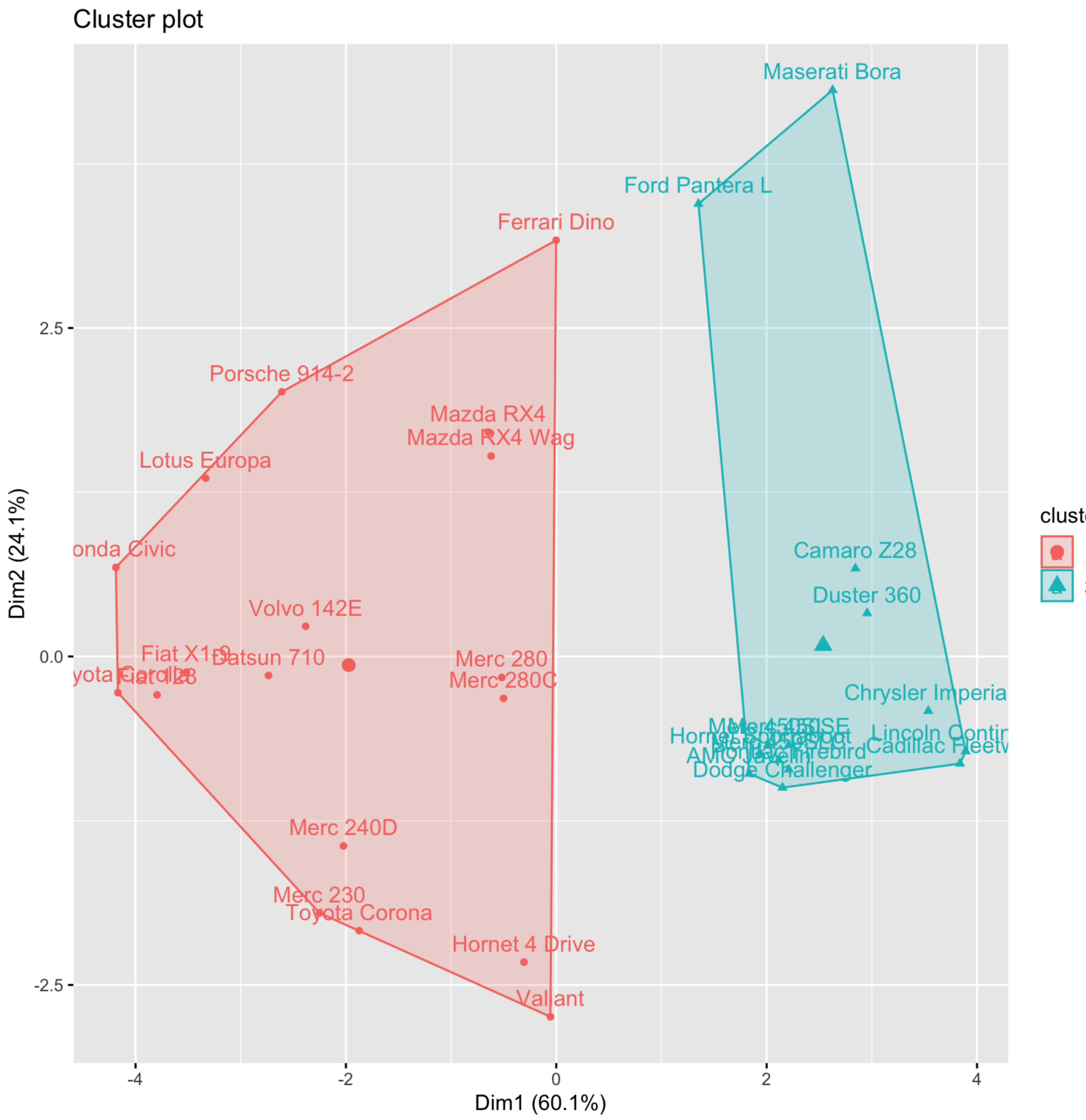
K-Means Clustering

```r
set.seed(123)

k_means_clustering <- kmeans(mtcars, centers = 2)

print(k_means_clustering$cluster)
```

```
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive   Hornet Sportabout             Valiant          Duster 360
                  1                   1                   1                   1                   2                   1                   2
          Merc 240D            Merc 230            Merc 280           Merc 280C          Merc 450SE          Merc 450SL         Merc 450SLC
                  1                   1                   1                   1                   2                   2                   2
 Cadillac Fleetwood Lincoln Continental   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla       Toyota Corona
                  2                   2                   2                   1                   1                   1                   1
   Dodge Challenger         AMC Javelin          Camaro Z28    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
                  2                   2                   2                   2                   1                   1                   1
     Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
                  2                   1                   2                   1
```

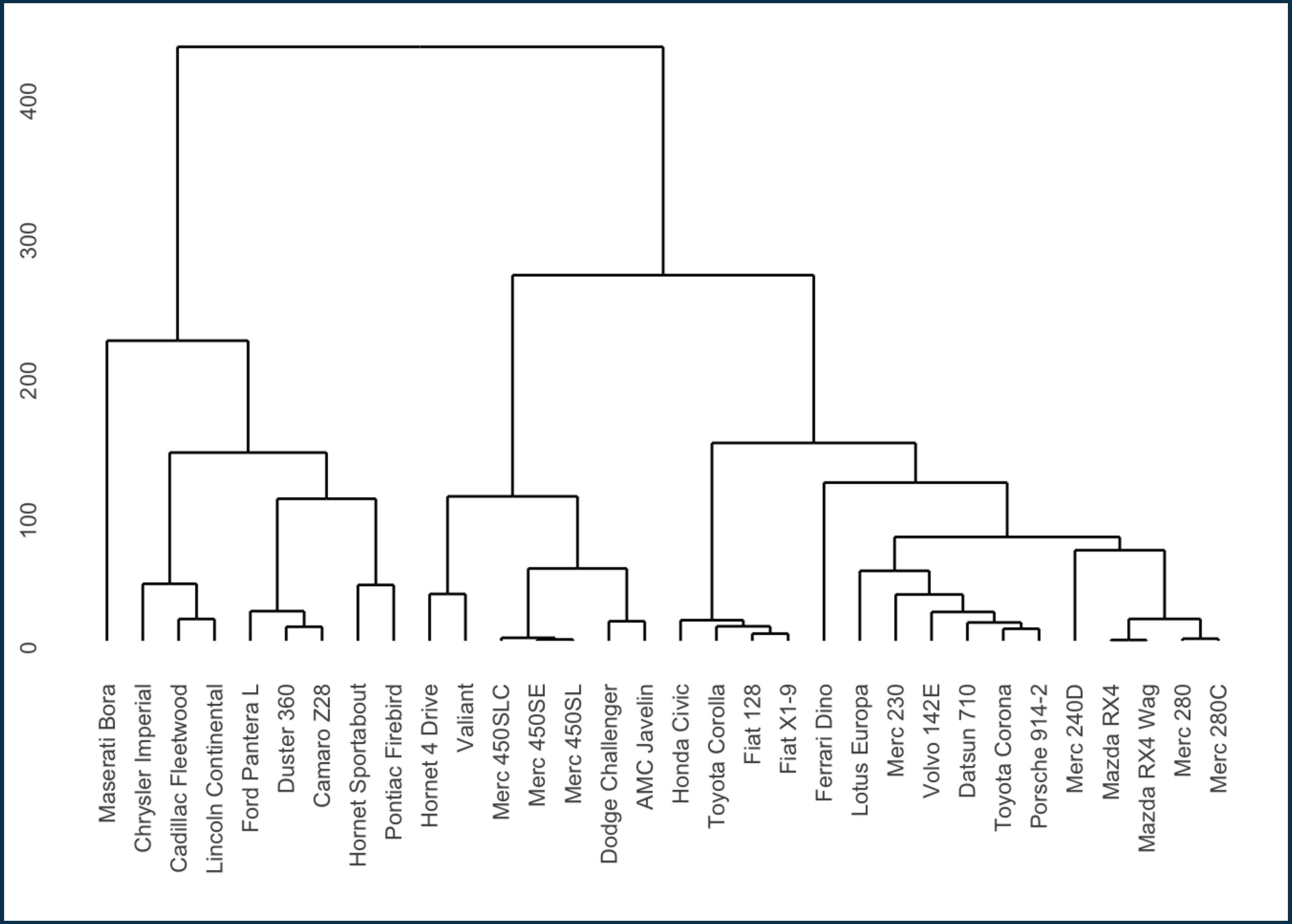
Plot K-Means Clustering

```
library(factoextra)

fviz_cluster(k_means_clustering,
data = mtcars)
```



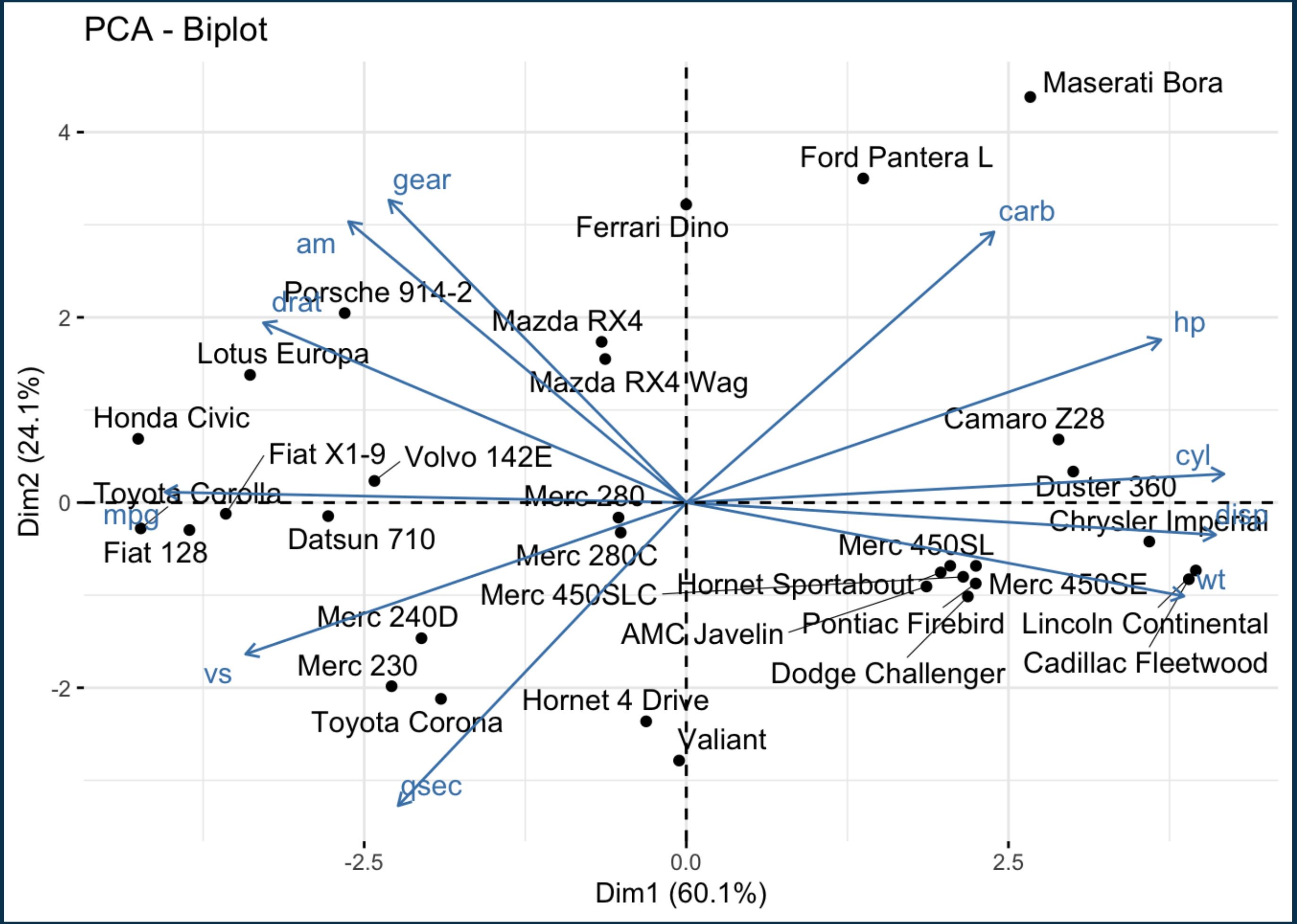
Hierarchical Clustering

```r
library(ggdendro)
set.seed(123)
mtcars_distance = dist(mtcars, method = 'euclidean')
h_clustering = hclust(mtcars_distance)
ggdendrogram(h_clustering)
```

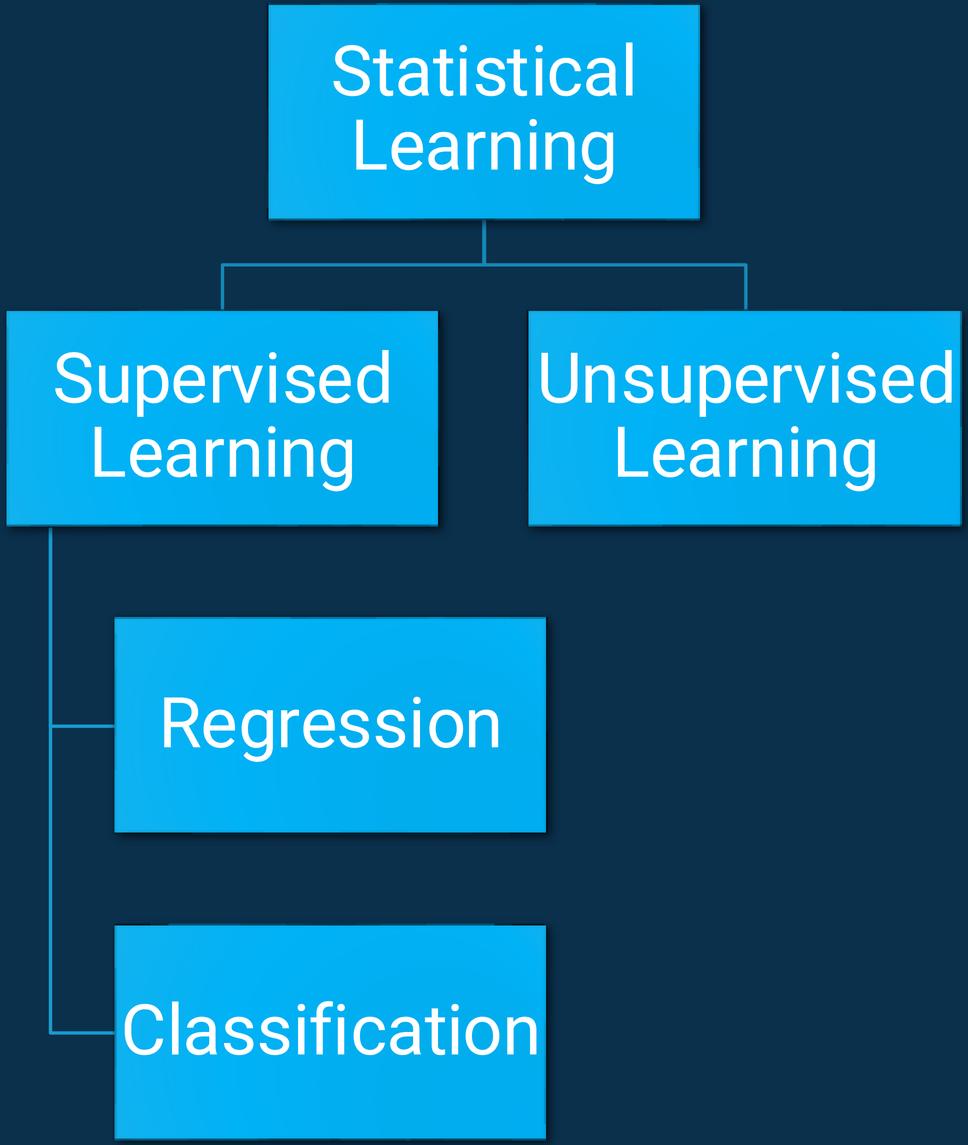


Principal Component Analysis Biplot

```
library(FactoMineR)

pca_mtcars <- PCA(mtcars,  graph = FALSE)

fviz_pca_biplot(pca_mtcars, repel = TRUE)
```



PCA Biplot can be used to understand the correlation structure of the variables


Statistical Learning
Supervised Learning
Unsupervised Learning
Regression
Classification

Thank you!