Predictive Modeling II

Types of Regression Analysis:

1. Linear Regression:

Applies in case of 1 continuous target (response/dependent) attribute and 1 or more continuous (or categorical) predictors.

2. Polynomial Regression :

A linear regression which applies when there is a nonlinear relationship between the target and the predictor.

3. Logistic Regression :

The target (response/dependent) attribute is binary in nature. The predictors can be continuous or binary.

4. Quantile Regression:

Applies when the assumptions of linear regression are not met and for cases where interest is in the quantiles.

Types of Regression Analysis:

5. Elastic Net Regression:

Applies to handle very high correlated predictor.

6. Principal Components Regression (PCR):

Applies when there are too many predictor or multicollinearity exist in the data (dimension reduction).

7. Partial Least Squares (PLS) Regression:

It is an opposite method of principal component. It is also applicable when there are many independent variables.

8. Support Vector Regression (SVR):

This can provide a solution to linear and non-linear models. It makes use of non-linear kernel functions to find the optimal solution for non-linear models.

Types of Regression Analysis:

9. Ordinal Regression:

It is used to predict behavior of ordinal level target (response/dependent) attribute with a set of predictors.

10. Poisson Regression:

This is applicable when the target has count data. It assumes the variance equal to its mean.

11. Negative Binomial Regression:

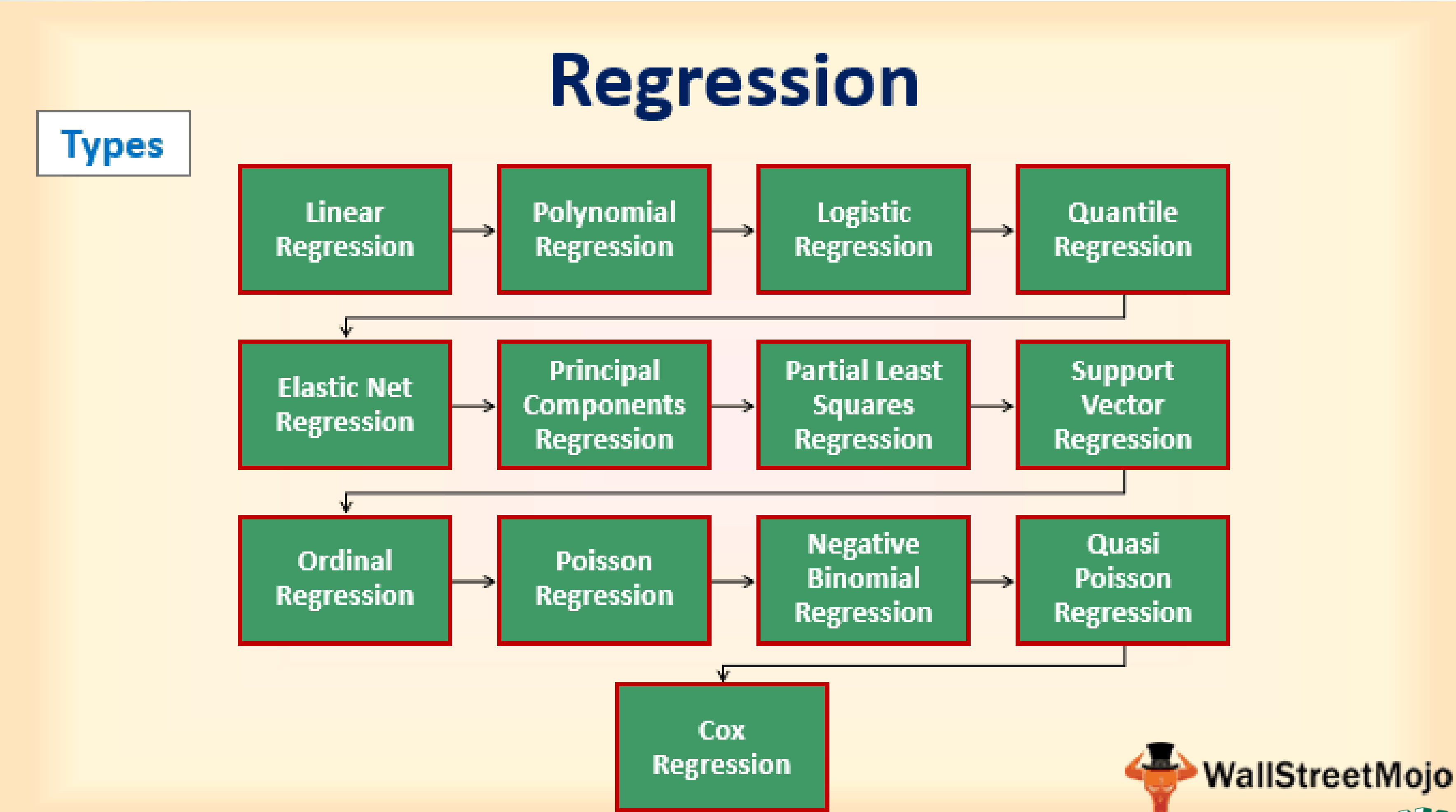
A special case of the Poisson regression, but doesn’t assume the variance equal to its mean.

12. Quasi Poisson Regression:

A special case of the Poisson regression where the variance is a linear function of the mean.

13. Cox Regression:

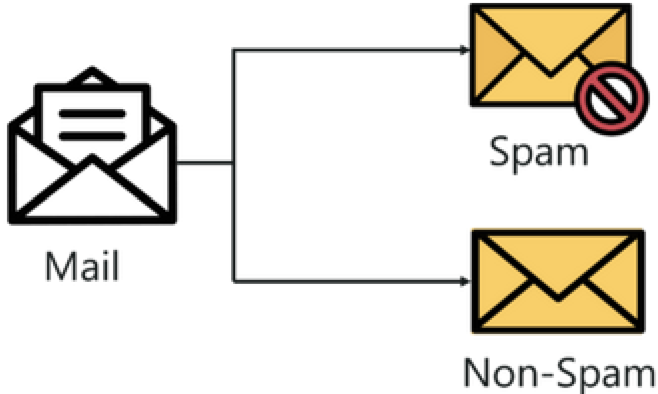
It comes more into use for analyzing time-to-event data.


Regression
Types
Linear Regression
Polynomial Regression
Logistic Regression
Quantile Regression
Elastic Net Regression
Principal Components Regression
Partial Least Squares Regression
Support Vector Regression
Ordinal Regression
Poisson Regression
Negative Binomial Regression
Quasi Poisson Regression
Cox Regression
WallStreetMojo

Logistic Regression:

Logistic regression is a special type of regression where binary target (response/dependent) attribute is related to a set of predictors, which can be discrete, continuous, and/or categorical.

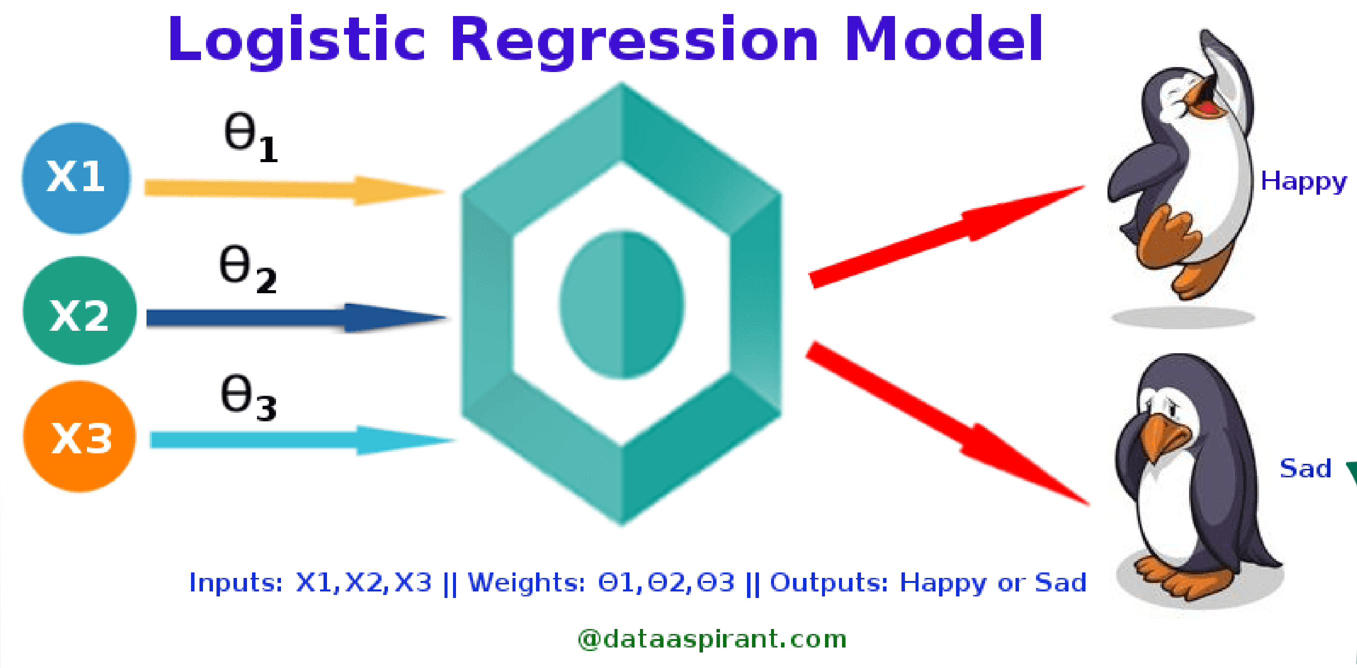
- It is a nonlinear regression.
- Simple logistic regression when there is one predictor.
- Multiple logistic regression when there are more than one predictor.
- It is predictive classification model.



Example:

1. How does the chance of getting lung cancer (yes vs. no) change for every additional pound a person is overweight and for every pack of cigarettes smoked per day?

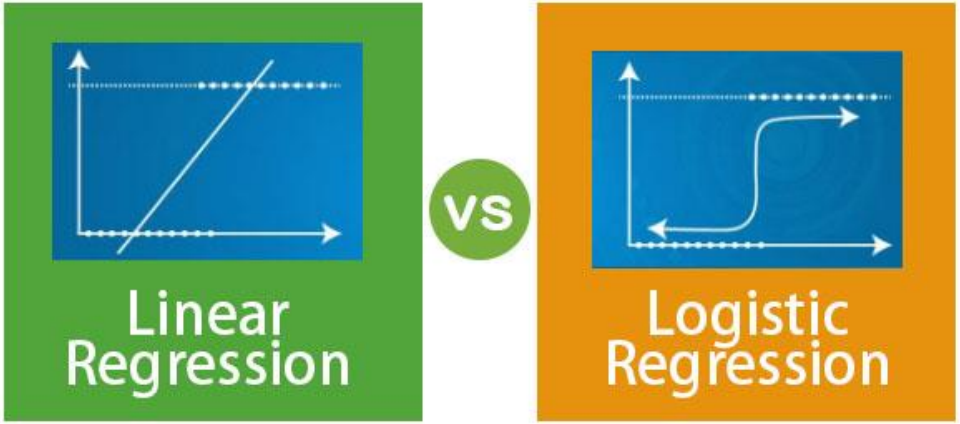
2. Do body weight, calorie intake, fat intake, and age have an influence on the probability of having a heart attack (yes vs. no)?



Linear Regression vs. Logistic Regression:

Given data on time spent studying and exam scores. Linear Regression and logistic regression can predict different things:

- Linear Regression could help us predict the student's test score on a scale of 0 – 100 (continuous).
- Logistic Regression could help use predict whether the student passed or failed (binary). View probability scores underlying the model's classifications.



Linear Regression vs. Logistic Regression:

A logistic regression is a nonlinear regression because it is nonlinear in coefficients (β_1, β_2, ..., β_k).

$y = \beta^2 x + \varepsilon$ - non linear

$y = \frac{1}{\beta} x + \varepsilon$ - non linear

$y = e^{\beta x} + \varepsilon$ - non linear

$y = \frac{1}{1+\beta x} + \varepsilon$ - non linear

$y = \beta x^2 + \varepsilon$ - linear

$y = \beta \frac{1}{x} + \varepsilon$ - linear

$y = \beta \ln x + \varepsilon$ - linear

Types of Logistic Regression:

1. Binary Logistic Regression:

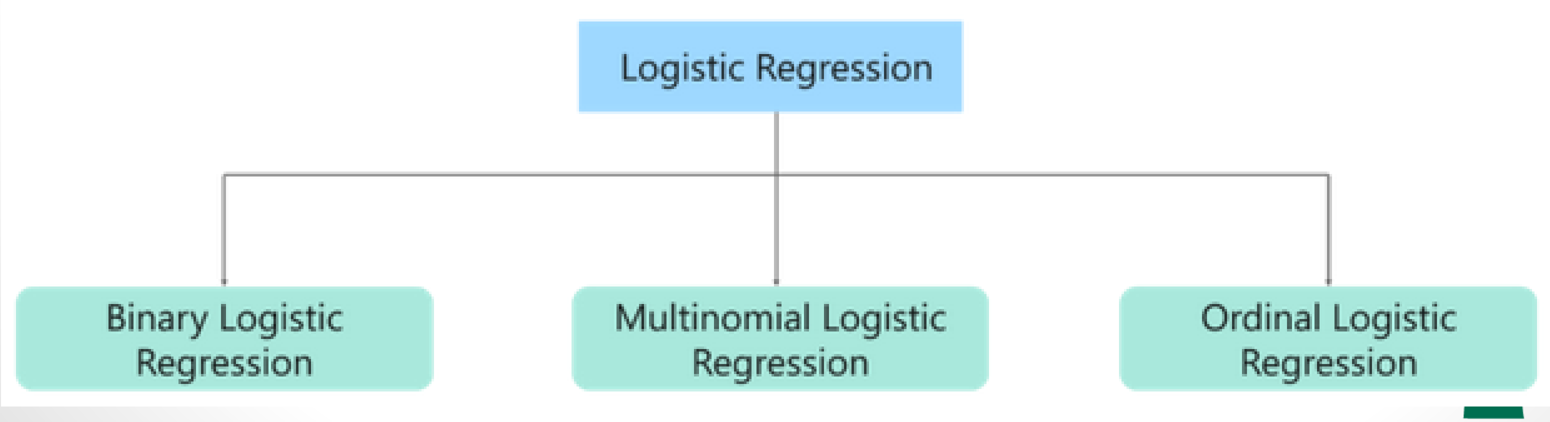
This is used when the target (response) attribute has only 2 classes (categories).

2. Multinomial Logistic Regression:

This is used when the target (response) attribute has more than 2 classes (categories).

3. Ordinal Logistic Regression:

This is used when the target (response) attribute has used when the response variable is ordinal in nature.



Logistic Regression:

The logistics regression model can be written by one of the following two formulas:

$$\pi_i = Pr(Y_i = 1 | X_i = x_i) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

$$\begin{aligned} \text{logit}(\pi_i) &= \log\left(\frac{\pi_i}{1 - \pi_i}\right) \\ &= \beta_0 + \beta_1 x_i \\ &= \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} \end{aligned}$$

where Y is a binary target (response) attribute, X = (X_1, X_2, ..., X_k) be a set of predictors which can be discrete, continuous, or a combination, and β = (β_1, β_2, ..., β_k) are the regression coefficients.

Example 1: (Heart)

Heart.csv dataset (from Kaggle and available on canvas) contains medical history of patients of Hungarian and Switzerland origin. Attribute Information:

1. age : age in year
2. sex : (1 = male; 0 = female)
3. cp: the chest pain experienced (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)
4. trestbps: resting blood pressure (in mm hg on admission to the hospital)
5. chol: serum cholesterol in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. restecg: resting electrocardiographic measurement (0 = normal, 1 = having st-t wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by estes' criteria)
8. thalach: maximum heart rate achieved
9. exang: exercise induced angina (1 = yes; 0 = no)
10. oldpeak: the slope of the peak exercise st segment (1: upsloping, 2: flat, 3: downsloping)
11. slope: the slope of the peak exercise st segment (1: upsloping, 2: flat, 3: downsloping)
12. ca: number of major vessels (0–3) colored by flourosopy
13. thal: a blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)
14. target: heart disease (0 = no, 1 = yes)

Example 1: (Heart)

1. Import and view the data.

Note: The response is target which is a binary attribute.

```
> Heart = read.csv(file = "C:\\Users\\ajornaz\\Desktop\\Data Mining\\Data\\heart.csv", sep = ",")
> View(Heart)
```

Lecture 9.R | Heart

Filter

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
2	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
3	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
4	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
5	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Example 1: (Heart)

2. Present the summary statistics.

Note: The information in summary function gives more sense about the continuous attributes.

```
> summary(Heart)
      age             sex               cp           trestbps          chol            fbs        
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000  
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000  
 Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :240.0   Median :0.0000  
 Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :246.3   Mean   :0.1485  
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000  
 Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000  
    restecg          thalach          exang           oldpeak         slope             ca        
 Min.   :0.0000   Min.   : 71.0   Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
 Median :1.0000   Median :153.0   Median :0.0000   Median :0.80   Median :1.000   Median :0.0000  
 Mean   :0.5281   Mean   :149.6   Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294  
 3rd Qu.:1.0000   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
 Max.   :2.0000   Max.   :202.0   Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000  
      thal           target      
 Min.   :0.000   Min.   :0.0000  
 1st Qu.:2.000   1st Qu.:0.0000  
 Median :2.000   Median :1.0000  
 Mean   :2.314   Mean   :0.5446  
 3rd Qu.:3.000   3rd Qu.:1.0000  
 Max.   :3.000   Max.   :1.0000  
```

Example 1: (Heart)

3. Use count() function to present the number of male and female in the data.

Note: count() function counts the number of values that satisfy the specified conditions.

4. Present the gender distribution in the target.

```
> count(Heart, vars = sex)
# A tibble: 2 x 2
   vars     n
  <int> <int>
1     0    96
2     1   207
```

```
> count(Heart, vars = sex, target)
# A tibble: 4 x 3
   vars target     n
  <int>  <int> <int>
1     0      0    24
2     0      1    72
3     1      0   114
4     1      1    93
```

Example 1: (Heart)

4. Split the data into two groups training data (80%) and test data (20%). Fit the logistic regression model using glm() function.

```
> smp_size = floor(0.80 * nrow(Heart))
> index = sample(seq_len(nrow(Heart)), size = smp_size)
> train = Heart[index, ]
> test = Heart[-index, ]
```

```
> model_full = glm( target ~ ., data = train, family = binomial)
> summary(model_full)

Call:
glm(formula = target ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5298  -0.3790   0.1034   0.5914   2.4790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.291143   2.959469   0.774 0.438828
age         -0.008927   0.026077  -0.342 0.732106
sex         -1.715267   0.514379  -3.335 0.000854 ***
cp           0.929236   0.211595   4.392 1.13e-05 ***
trestbps    -0.007213   0.011544  -0.625 0.532076
chol        -0.002561   0.004215  -0.608 0.543424
fbs         -0.185478   0.603781  -0.307 0.758696
restecg      0.429932   0.386760   1.112 0.266300
thalach      0.018923   0.011681   1.620 0.105240
exang       -1.015920   0.447630  -2.270 0.023235 *
oldpeak     -0.840706   0.247003  -3.404 0.000665 ***
slope        0.398122   0.389482   1.022 0.306694
ca          -0.727355   0.207190  -3.511 0.000447 ***
thal        -0.778784   0.318031  -2.449 0.014335 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 335.42  on 241  degrees of freedom
Residual deviance: 169.31  on 228  degrees of freedom
AIC: 197.31

Number of Fisher Scoring iterations: 6
```

Example 1: (Heart)

5. Select the best model. Use stepwise regression.

```
> step(model_full, direction = "both")
Start:  AIC=197.31
target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + 
    exang + oldpeak + slope + ca + thal

           Df Deviance    AIC
- fbs       1   169.40 195.40
- age       1   169.42 195.42
- chol      1   169.67 195.67
- trestbps  1   169.70 195.70
- slope     1   170.34 196.34
- restecg   1   170.55 196.55
<none>          169.31 197.31
- thalach   1   172.05 198.05
- exang     1   174.47 200.47
- thal      1   175.41 201.41
- sex       1   182.10 208.10
- ca        1   182.43 208.43
- oldpeak   1   182.75 208.75
- cp        1   191.70 217.70
```

Full model: the model with all predictors.

```
Step:  AIC=189.7
target ~ sex + cp + restecg + thalach + exang + oldpeak + ca + 
    thal

           Df Deviance    AIC
<none>          171.70 189.70
- restecg   1   173.75 189.75
+ slope     1   170.68 190.68
+ trestbps  1   171.13 191.13
+ chol      1   171.14 191.14
+ age       1   171.25 191.25
+ fbs       1   171.40 191.40
- thalach   1   176.82 192.82
- thal      1   177.57 193.57
- exang     1   177.71 193.71
- sex       1   183.41 199.41
- ca        1   185.38 201.38
- cp        1   194.05 210.05
- oldpeak   1   197.84 213.84

Call:  glm(formula = target ~ sex + cp + restecg + thalach + exang + 
    oldpeak + ca + thal, family = binomial, data = train)

Coefficients:
(Intercept)          sex           cp      restecg      thalach        exang      oldpeak           ca  
    0.19001     -1.50016      0.89660      0.53335      0.02157     -1.06241     -0.97547     -0.70331  
       thal  
   -0.73253  

Degrees of Freedom: 241 Total (i.e. Null);  233 Residual
Null Deviance:      335.4 
Residual Deviance: 171.7        AIC: 189.7
```

Reduced model: the model with only significant predictors.

Example 1: (Heart)

6. check the significance of predictors in the reduced model.

```
> Reduced_model = glm(target ~ sex + cp + thalach + exang +
+                     oldpeak + ca + thal, family = binomial, data = train)
> summary(Reduced_model)

Call:
glm(formula = target ~ sex + cp + thalach + exang + oldpeak +
    ca + thal, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3445  -0.4467   0.1425   0.5781   2.4330

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.437800   1.668335   0.262 0.793000
sex         -1.512121   0.461515  -3.276 0.001051 **
cp           0.884758   0.199982   4.424 9.68e-06 ***
thalach      0.021744   0.009767   2.226 0.025997 *
exang       -1.051763   0.432273  -2.433 0.014970 *
oldpeak     -0.964380   0.213833  -4.510 6.48e-06 ***
ca          -0.697873   0.194200  -3.594 0.000326 ***
thal        -0.724244   0.299861  -2.415 0.015724 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 335.42  on 241  degrees of freedom
Residual deviance: 173.75  on 234  degrees of freedom
AIC: 189.75

Number of Fisher Scoring iterations: 6
```

All predictors in the reduced model are significant at 5% significance level.

Example 1: (Heart)

7. Compare AIC and BIC for the full and reduced model.

```
> list(model_full = broom::glance(model_full),
+      Reduced_model = broom::glance(Reduced_model))
$`model_full`
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance df.residual
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int>
1          335.     241  -84.7  197.  246.     169.         228

$Reduced_model
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance df.residual
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int>
1          335.     241  -86.9  190.  218.     174.         234
```

The reduced model has less AIC and BIC.

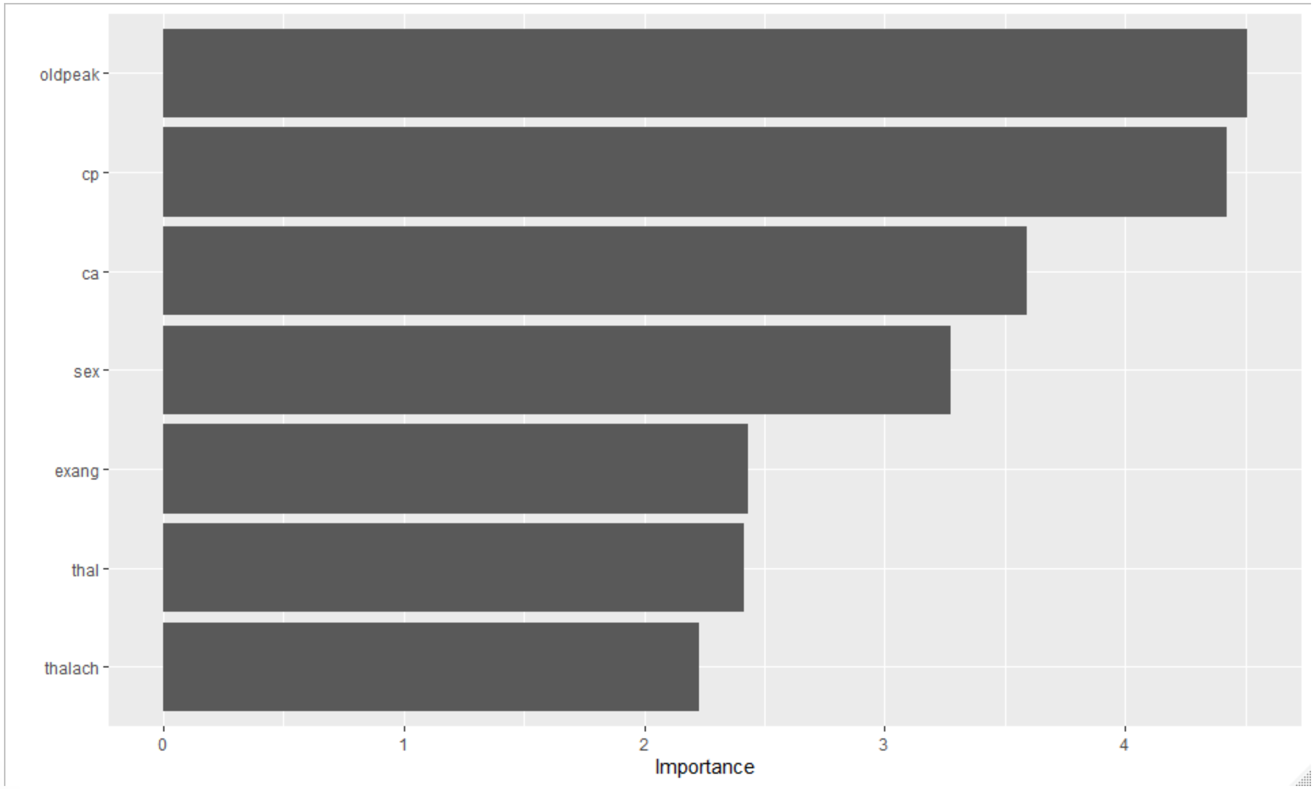
Example 1: (Heart)

```
install.packages("vip")
library("vip")
```

8. Determine the most important predictors.

Note: install "vip" package and use vip() function.

```
> vip(Reduced_model, num_features = 8)
```



Oldpeak and cp are the most important attributes; Moreover, both of them have the lowest p-values.

Example 1: (Heart)

9. Make predictions on training set using the reduced model. Present the first 8 predictions.

10. Assign the labels with decision rule that if the prediction is greater than 0.5, assign it 1 else 0.

```
> predictions <- predict(Reduced_model, test, type="response")
> head(predictions,8)
        4         7        10        14        20        27        28        31 
0.8116718 0.8751449 0.8156557 0.6165619 0.8575791 0.7535313 0.7929194 0.9442345 
```

```
> predictions1 <- ifelse(predictions > 0.5, 1, 0)
> head(predictions1, 8)
 4  7 10 14 20 27 28 31 
 1  1  1  1  1  1  1  1 
```

- First value in the test data is the 4th value in the original data, then 7th and so on.
- The first 8 values predicted to have heart disease.

Example 1: (Heart)

11. Evaluate the model using a cross-tabulation "contingency table". Use the function mean() to check how much of the values are correctly predicted.

Find the probability contingency table.

```
> accuracy = table(predictions1, test[,14])
> accuracy

predictions1  0  1
           0 14  3
           1  5 39
```

```
> prop.table = prop.table(table(predictions1, test[,14]))
> prop.table

predictions1          0          1
           0 0.22950820 0.04918033
           1 0.08196721 0.63934426
```

```
> sum(diag(accuracy))/sum(accuracy)
[1] 0.8688525
> mean(predictions1 == test$target)
[1] 0.8688525
```

- The model predicted 87% of the test data correctly.
- 14 out of 19 who don't have heart disease and 39 out of 42 who have heart disease in the test data were predicted (classified) correctly.