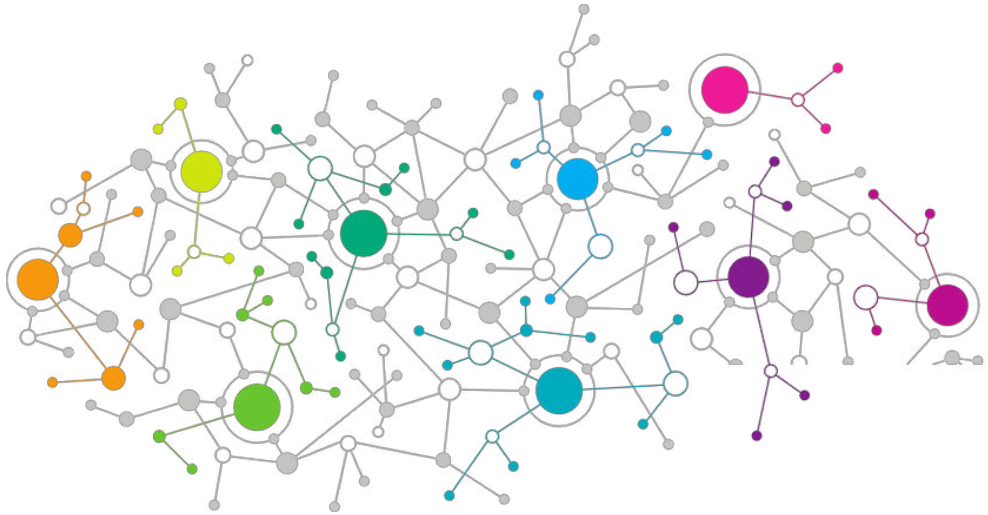
Data Science

Introduction to Logistic Regression



Agenda

01 Understanding Logistic Regression

02 Math behind Logistic Regression

03 Performance Metrics

04 Logistic Regression in R

Logistic Regression

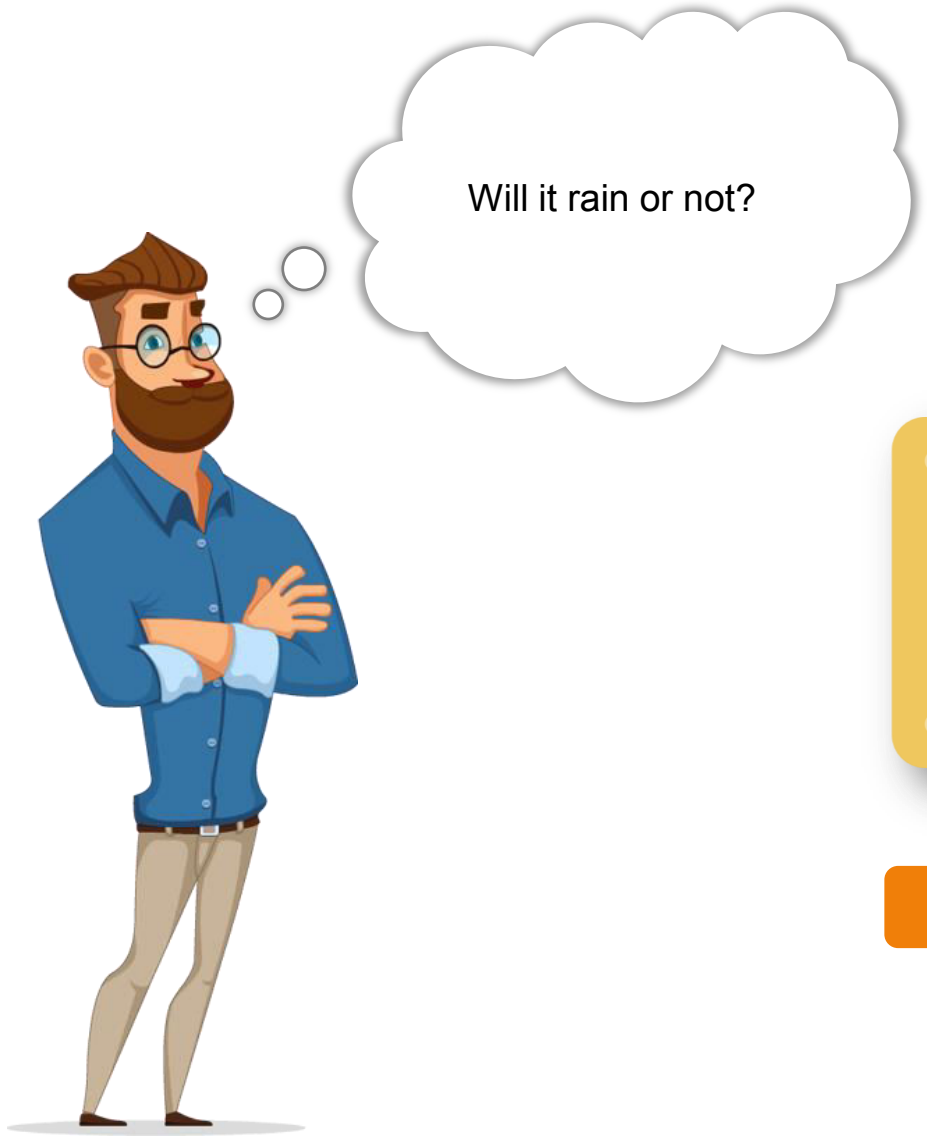
Logistic Regression

Logistic regression is a classification algorithm.

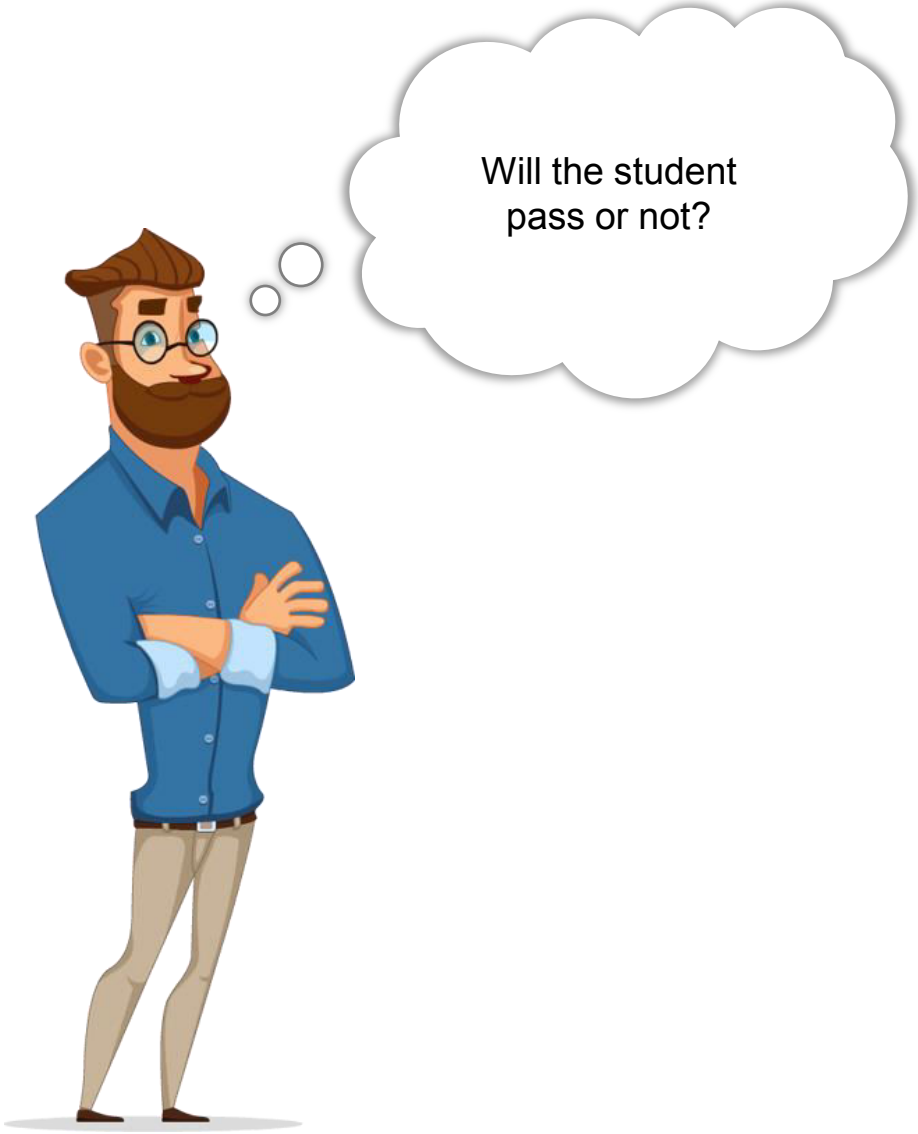
It is used to predict a binary outcome (1/0, Yes/No, True/False).

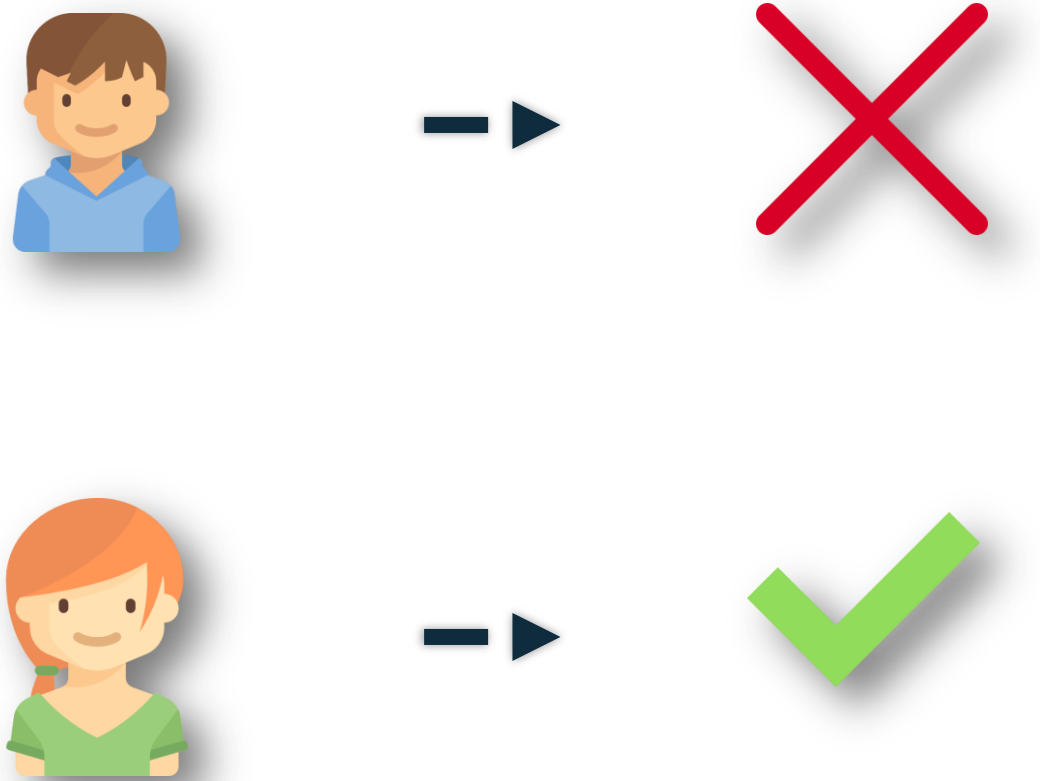
It predicts the probability of occurrence of an event.

Logistic Regression



Logistic Regression





Logistic Regression Applications

Logistic Regression Applications

Spam Detection

- Predicting if an email is a spam or not

Credit Card Fraud

- Predicting if a given credit card transaction is fraud or not

Health

- Predicting if a given mass of tissue is benign or malignant

Marketing

- Predicting if a given user will buy an insurance product or not

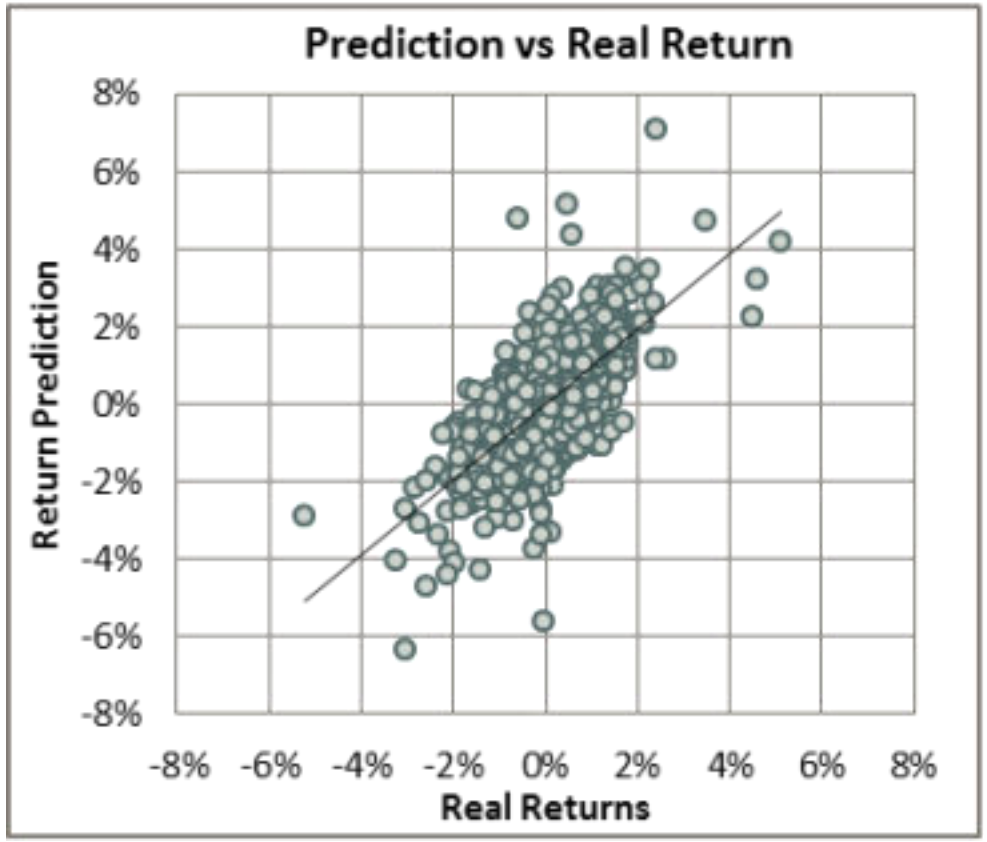
Banking

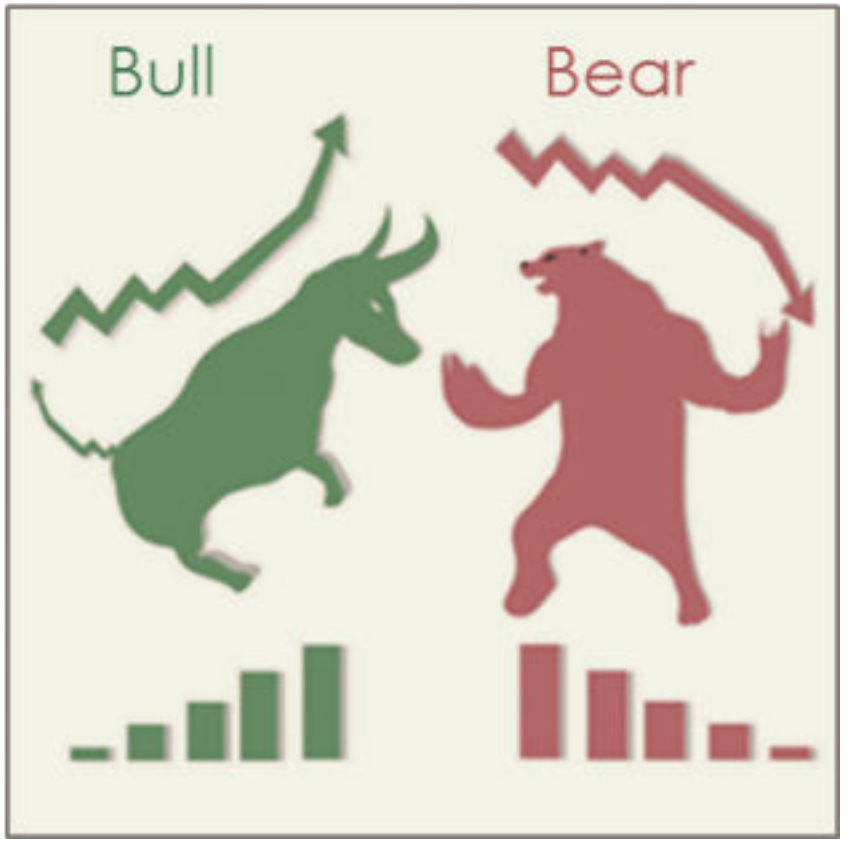
- Predicting if a customer will default on a loan or not

Linear Regression
vs.
Logistic Regression

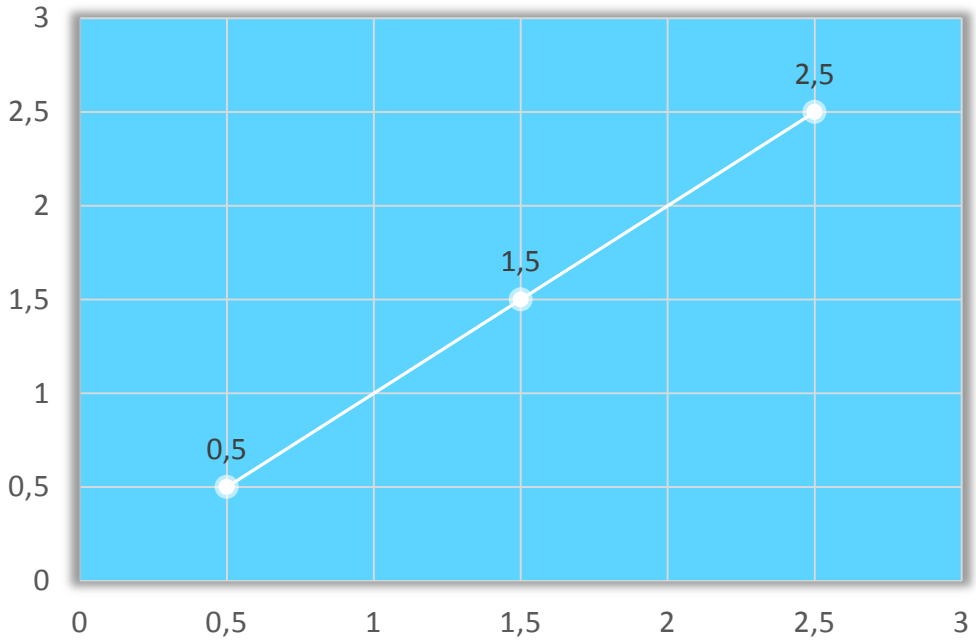
Linear Regression vs. Logistic Regression

Regression vs Classification

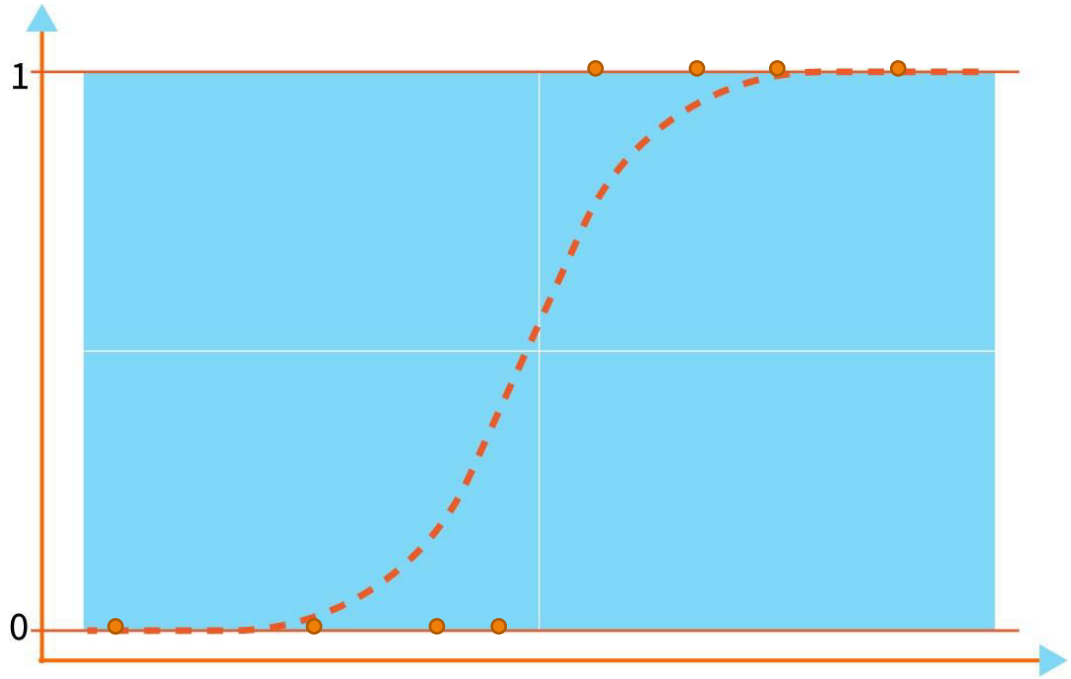




Linear Regression vs. Logistic Regression



Predicted ***Y*** can exceed 0 to 1 range



Predicted ***Y*** lies within 0 to 1 range

Assumptions in Logistic Regression

Assumptions in Logistic Regression

1. Logistic regression does not assume a linear relationship between the dependent variable and independent variables.
2. Logistic regression assumes linearity of independent variables and log odds.
3. Binary logistic regression requires the dependent variable to be binary.
4. Independent variables need not be normally distributed.
5. Logistic regression requires observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data.

Math behind Logistic Regression

Math behind Logistic Regression

The idea in logistic regression is to cast the problem in the form of a generalized linear regression model.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

This can be compactly expressed in the vector form:

$$w^T = [\beta_0, \beta_1, \ldots, \beta_n]$$

$$x^T = [1, x_1, \ldots, x_n]$$

Then

$$\hat{y} = w^T x$$

Math behind Logistic Regression

Logistic regression is part of a larger class of algorithms known as the **Generalized Linear Model (glm).**

The equation of a generalized linear model is:

$$g(E(y)) = \alpha + \beta x1 + \gamma x2$$

Here,

g() is the link function

E(y) is the expectation of target variable

α + βx1 + γx2 is the linear predictor (α, β, γ to be predicted)

The role of the link function is to '**link' the expectation of *y* to linear predictor.**

Odds

Odds are determined from probabilities and range between 0 and infinity.

The odds of success

odds (success) = p/(1-p) or **p/q**

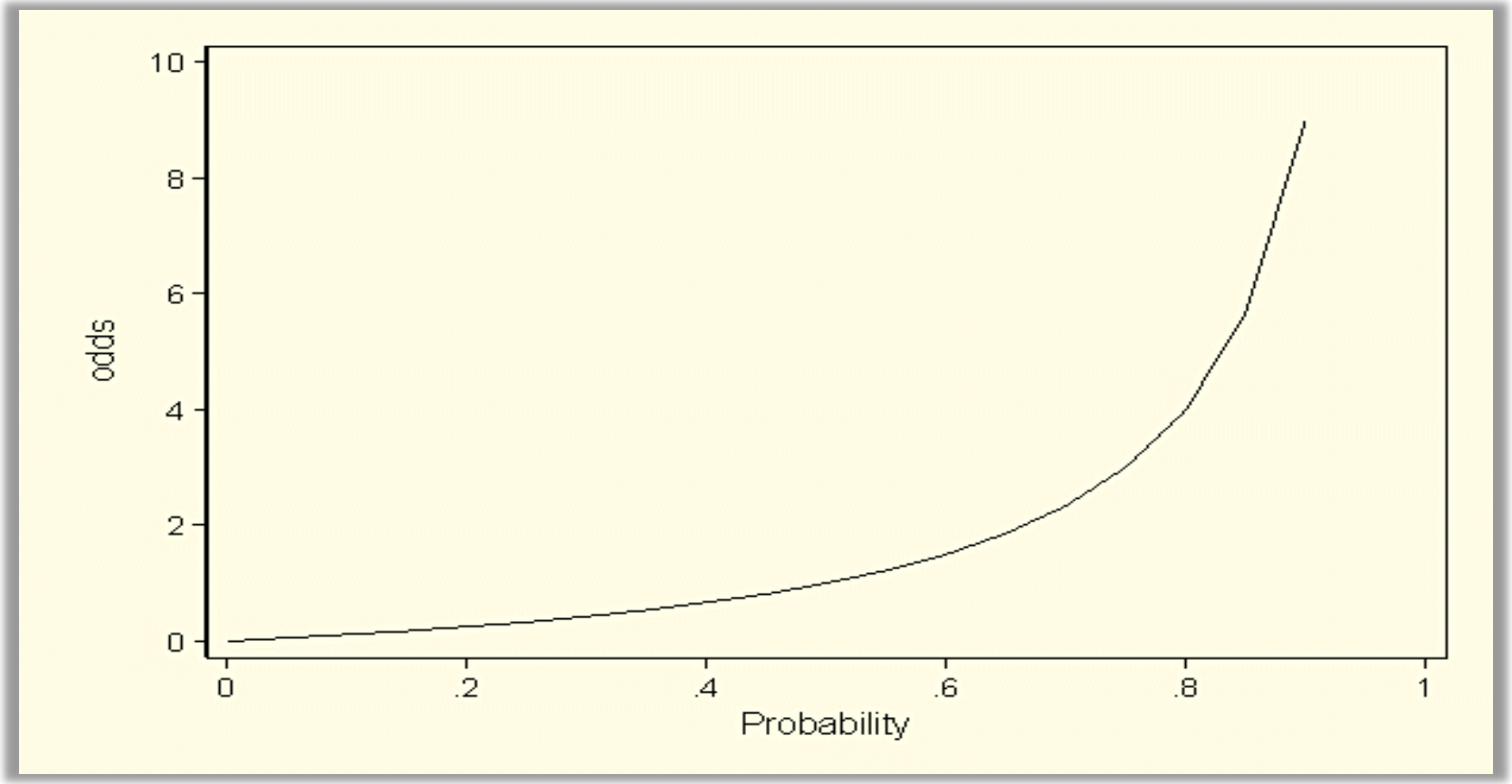
The odds of failure

odds (failure) = q/p

Odds are defined as the ratio of the probability of success and the probability of failure.

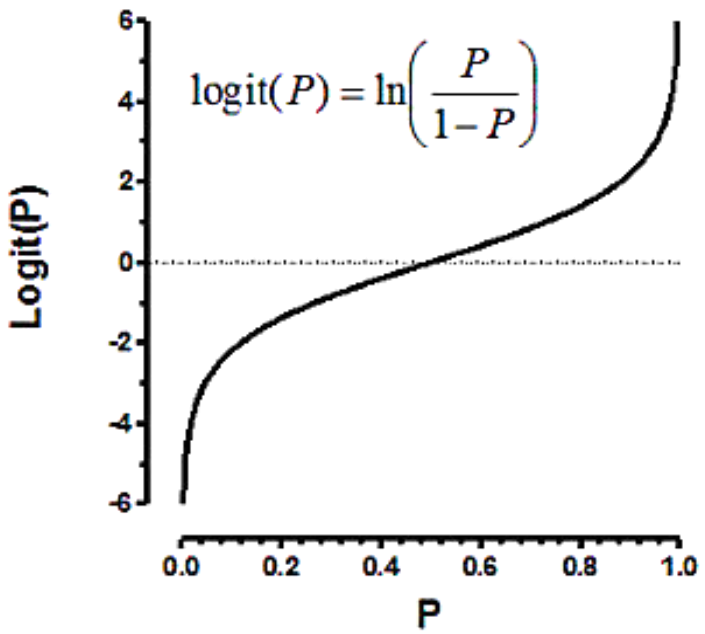
Odds

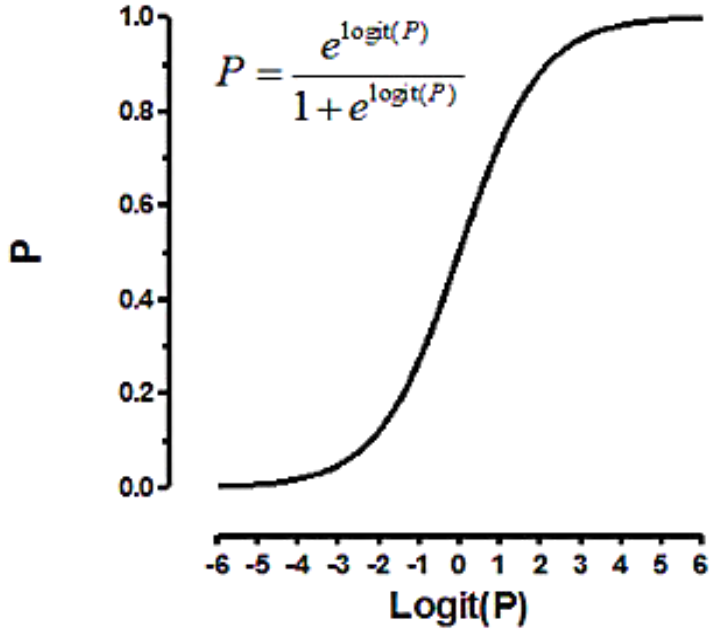
Odds are determined from probabilities and range between 0 and infinity.



Logit Function

Logistic regression is an estimation of logit function.





A **logit function** is simply a log of odds in favor of the event.

Logit Function

Logistic regression is an estimation of Logit function.

$$logit(p) = \hat{y} = w^T x$$

The logit function acts as a link between logistic regression and linear regression, and thus it is called a **link function.**

Logit Function

We know that the exponential of any value is always a positive number.

And, any number divided by the number + 1 will always be lower than 1.

$$P(Y = 1|X) = \frac{e^{(\beta_o + \beta_1 x)}}{e^{(\beta_o + \beta_1 x)} + 1}$$

Math behind Logit Function

$$p(X) = \frac{e^{(\beta_o+\beta_1 x)}}{e^{(\beta_o+\beta_1 x)} + 1}$$

$$p(e^{(\beta_o+\beta_1 x)} + 1) = e^{(\beta_o+\beta_1 x)}$$

$$p.e^{(\beta_o+\beta_1 x)} + p = e^{(\beta_o+\beta_1 x)}$$

$$p = e^{(\beta_o+\beta_1 x)} - p.e^{(\beta_o+\beta_1 x)}$$

$$p = e^{(\beta_o+\beta_1 x)}(1 - p)$$

$$\frac{p}{1-p} = e^{(\beta_o+\beta_1 x)}$$

$$\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x$$

Math behind Logit Function

$$\log\left(\frac{p}{1-p}\right) = y$$

Any number divided by number + 1 will always be lower than 1.

Logarithmic transformation on the outcome variable allows us to model a nonlinear association in a linear way.

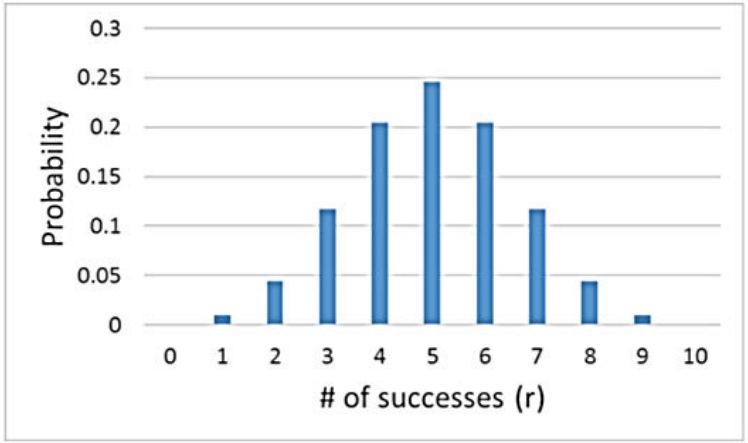
Maximum Likelihood Function

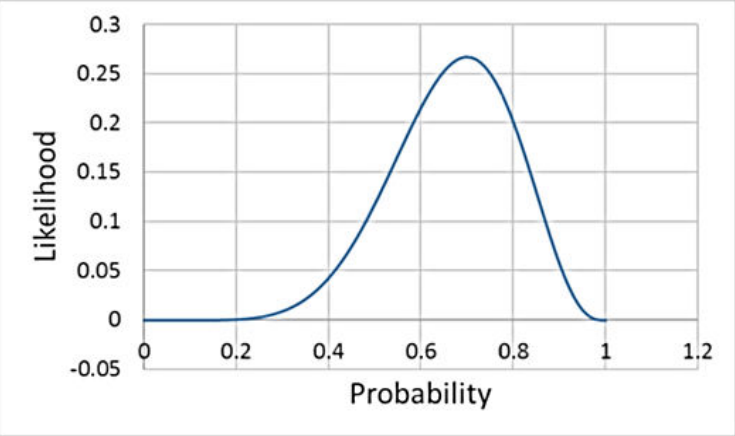
Now that you know what we are trying to estimate, next is the definition of the function we are trying to optimize to get the estimates of the coefficient.

This function is analogous to the square of error in linear regression and is known as the likelihood function.

Likelihood is also known as reverse probability.

In **probability**, we **predict data** based on **known parameters**. In **likelihood**, we **predict parameters** based on **known data.**





Maximum Likelihood Function

- The goal is to maximize the likelihood.
- We have seen the **logistic regression** model:

$$\log \frac{p(x)}{1-p(x)} = \beta_0 + x \cdot \beta$$

$$p(x; b, w) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}}$$

Maximum Likelihood Function

The **likelihood** function is:

$$L(\beta_0, \beta) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i)^{1-y_i}$$

Why log-likelihood?

The log-likelihood turns products into sums.

Log *x* is a monotonically increasing function; thus, log-likelihoods have the same relations of order as the likelihoods.

Maximum Likelihood Function

The likelihood is:

$$
\begin{aligned}
\ell(\beta_0, \beta) &= \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log 1 - p(x_i) \\
&= \sum_{i=1}^{n} \log 1 - p(x_i) + \sum_{i=1}^{n} y_i \log \frac{p(x_i)}{1 - p(x_i)} \\
&= \sum_{i=1}^{n} \log 1 - p(x_i) + \sum_{i=1}^{n} y_i (\beta_0 + x_i \cdot \beta) \\
&= \sum_{i=1}^{n} -\log 1 + e^{\beta_0 + x_i \cdot \beta} + \sum_{i=1}^{n} y_i (\beta_0 + x_i \cdot \beta)
\end{aligned}
$$

To find the maximum likelihood estimates, we would differentiate the log-likelihood with respect to the parameters, set the derivatives equal to zero and solve.

Simple Logistic Regression in R

Problem Statement

Building a simple logistic regression model on top of the customer_churn dataset

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
7590-VHVEG	Female	0	Yes	No	1	No	No phone service
5575-GNVDE	Male	0	No	No	34	Yes	No
3668-QPYBK	Male	0	No	No	2	Yes	No
7795-CFOCW	Male	0	No	No	45	No	No phone service
9237-HQITU	Female	0	No	No	2	Yes	No
9305-CDSKC	Female	0	No	No	8	Yes	Yes
1452-KIOVK	Male	0	No	Yes	22	Yes	Yes
6713-OKOMC	Female	0	No	No	10	No	No phone service
7892-POOKP	Female	0	Yes	No	28	Yes	Yes
6388-TABGU	Male	0	No	Yes	62	Yes	No

Tasks to be Performed

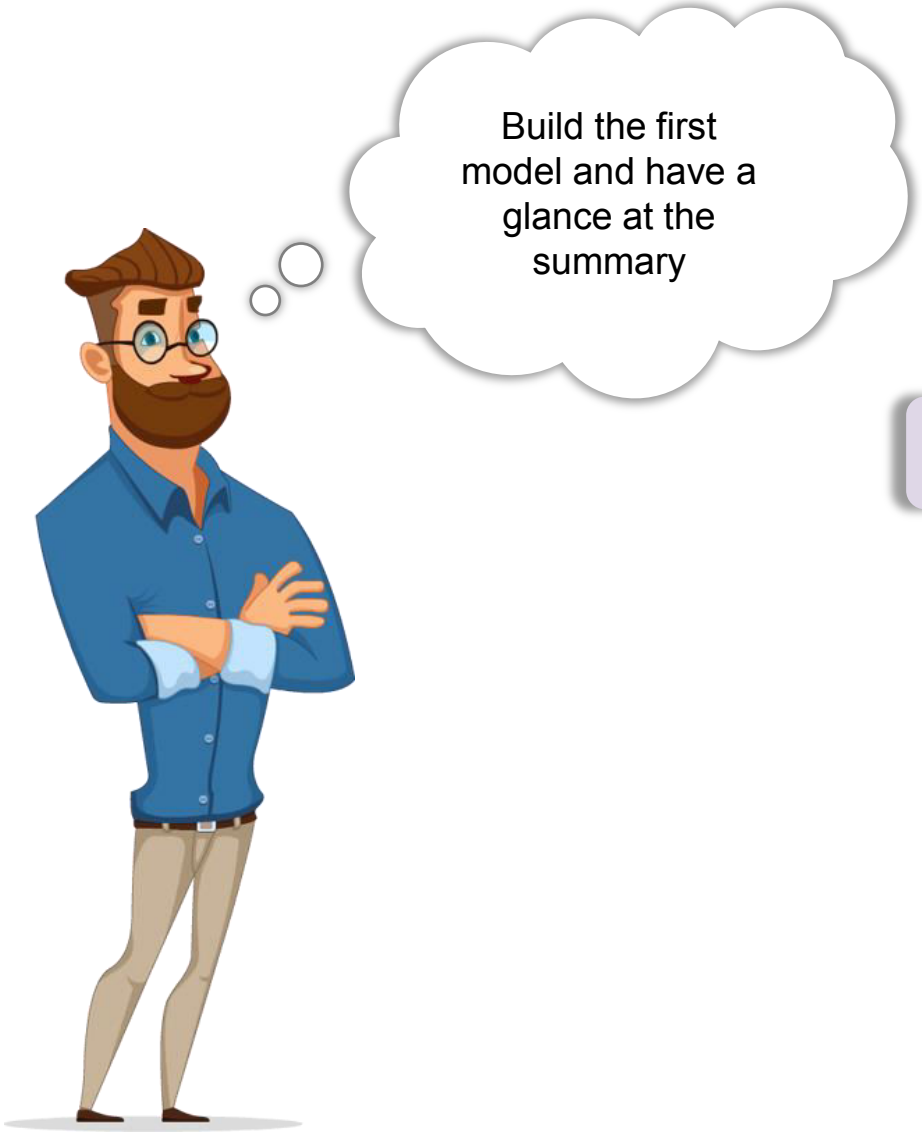
1. Build a simple logistic regression model where the dependent variable is “Churn” and the independent variable is “MonthlyCharges”

2. Build a simple logistic regression model where the dependent variable is “Churn” and the independent variable is “tenure”

Simple Logistic Regression in R


```
customer_churn<-read.csv("C:/Users/INTELLIPAAT/Desktop/customer_churn.csv")
```

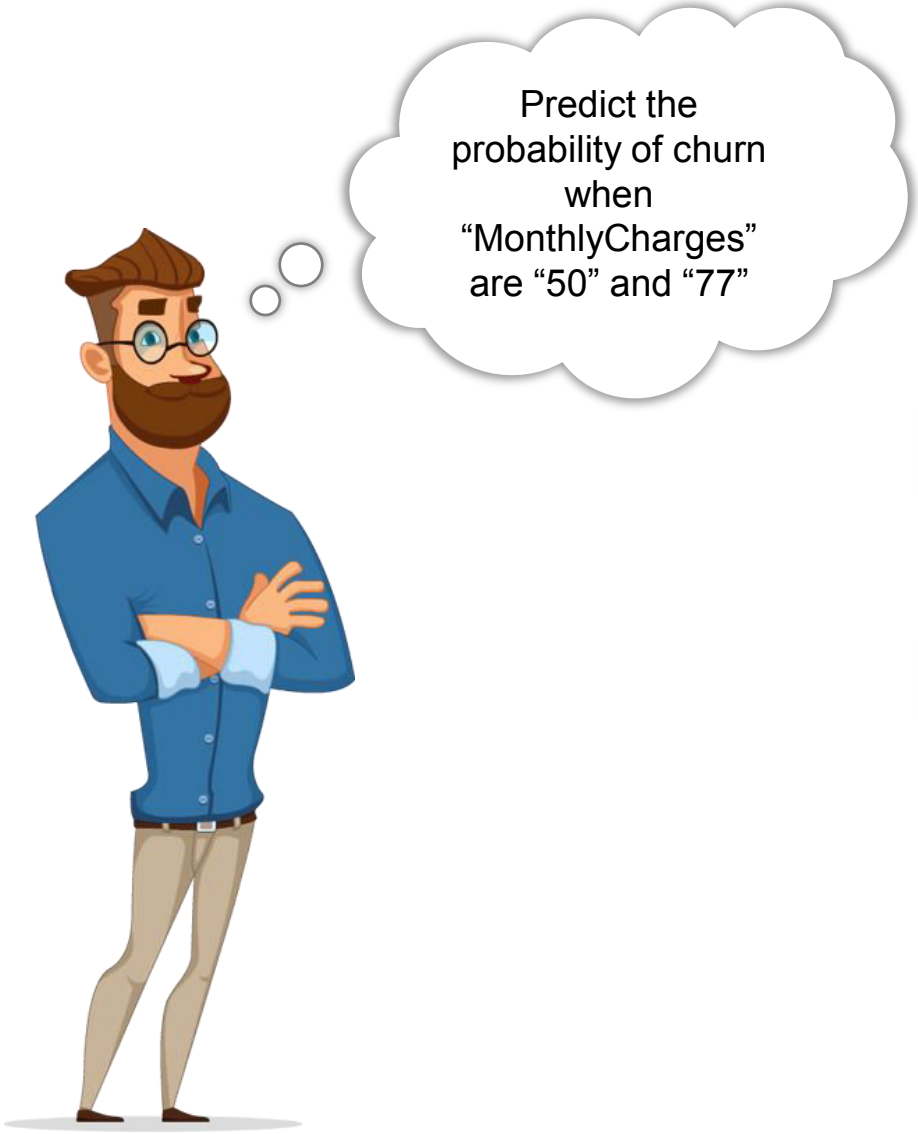
Simple Logistic Regression in R




```
glm(Churn~MonthlyCharges, data= customer_churn, family="binomial") ->log_mod1
```

```
summary(log_mod1)
```

Simple Logistic Regression in R




```
predict(log_mod1,data.frame(MonthlyCharges=50),type="response")
```

```
predict(log_mod1,data.frame(MonthlyCharges=77),type="response")
```

Simple Logistic Regression in R


```
predict(log_mod1,data.frame(MonthlyCharges=20:100),type="response")
```

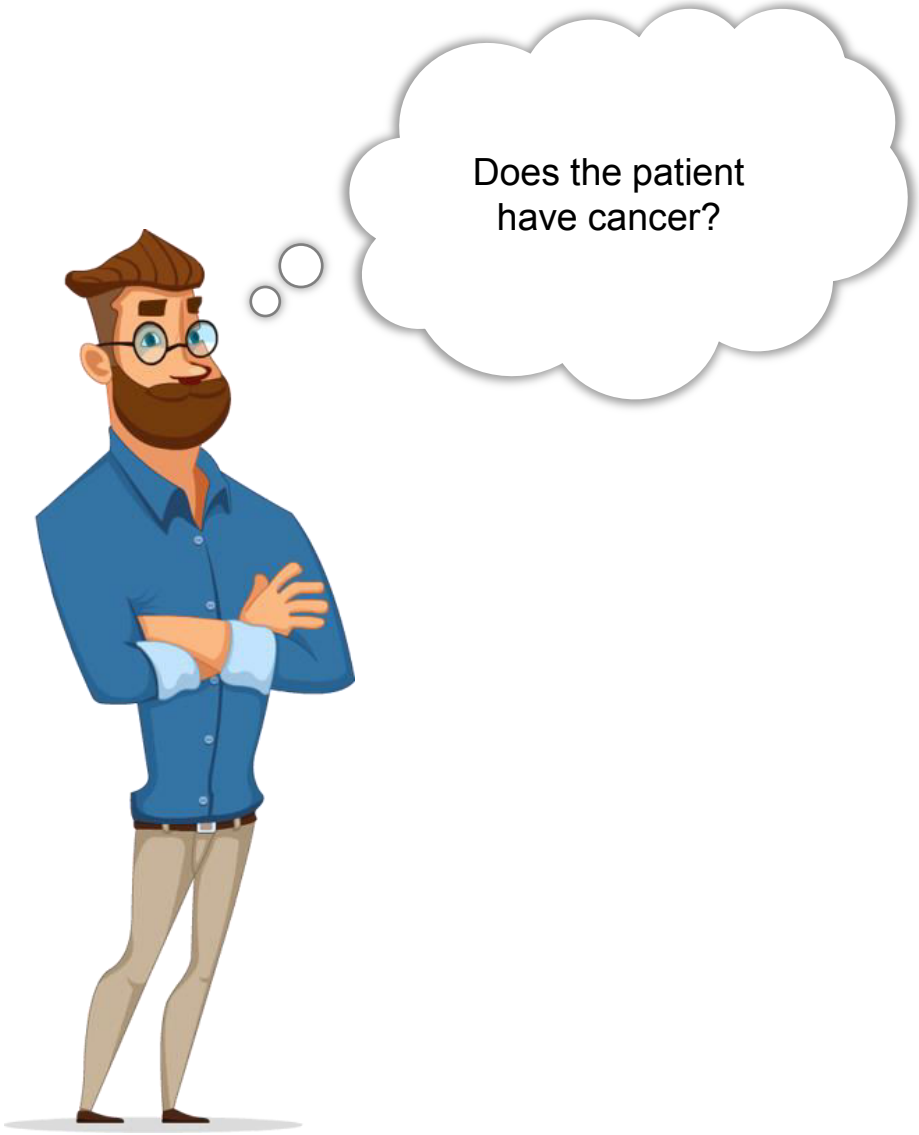
Confusion Matrix

Confusion Matrix

A **confusion matrix** helps you to describe the **performance** of a **classification model**!

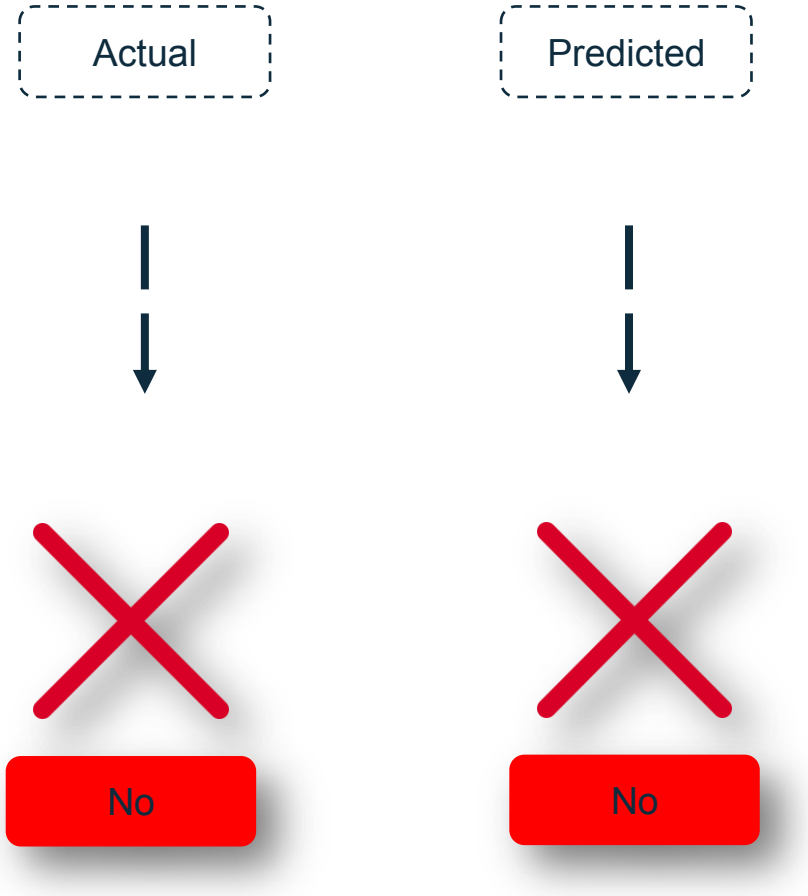
		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

Confusion Matrix

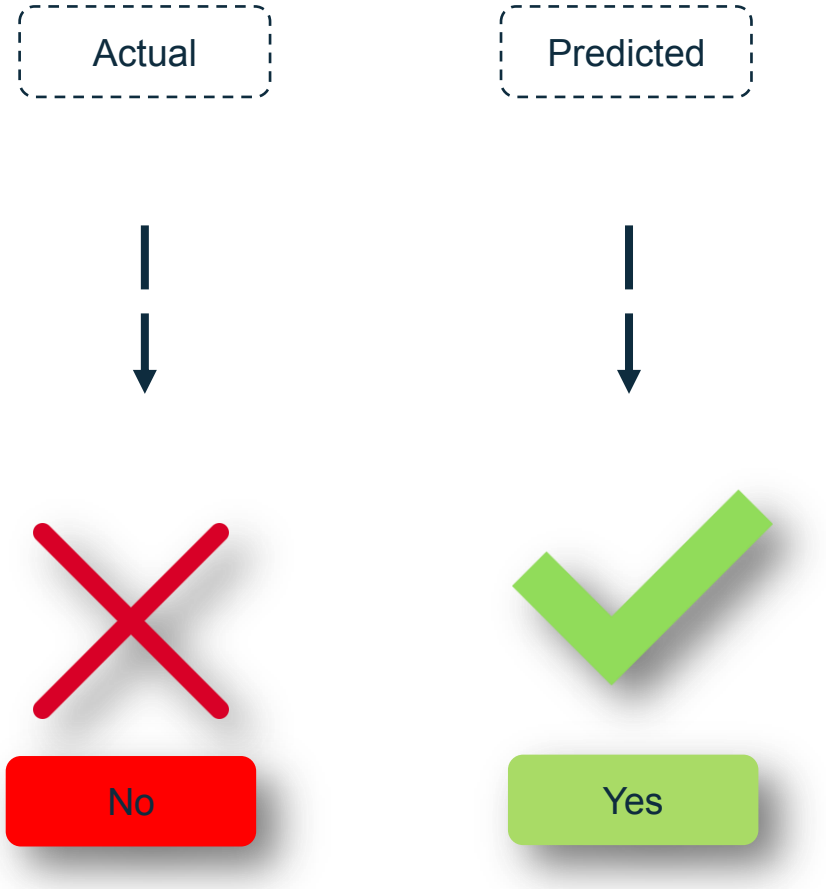


True Positives

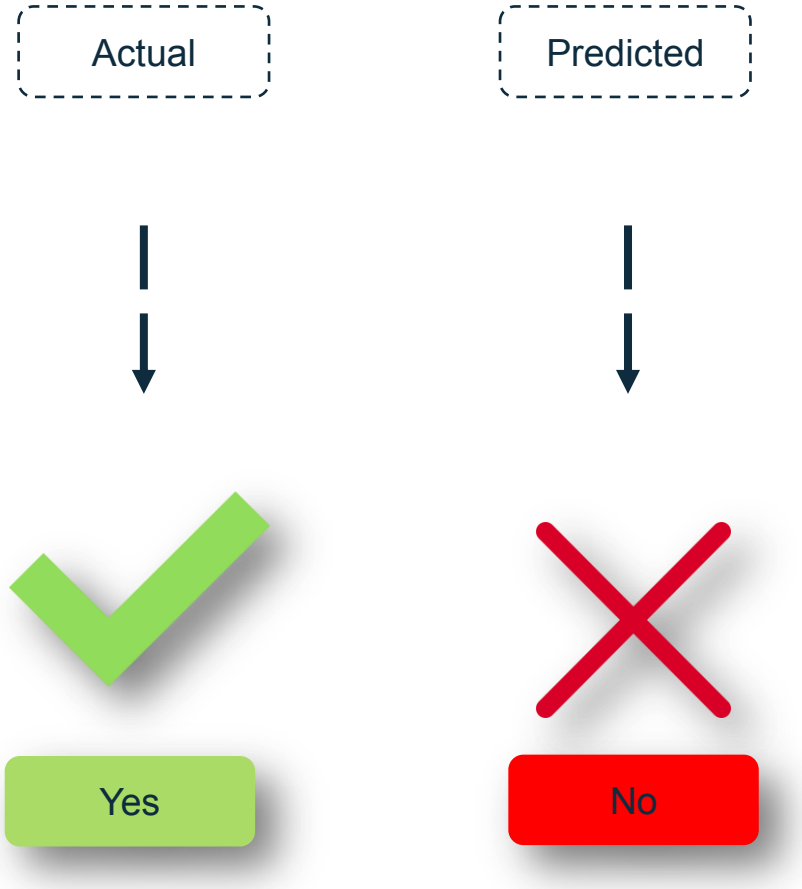
True Negatives



False Positives



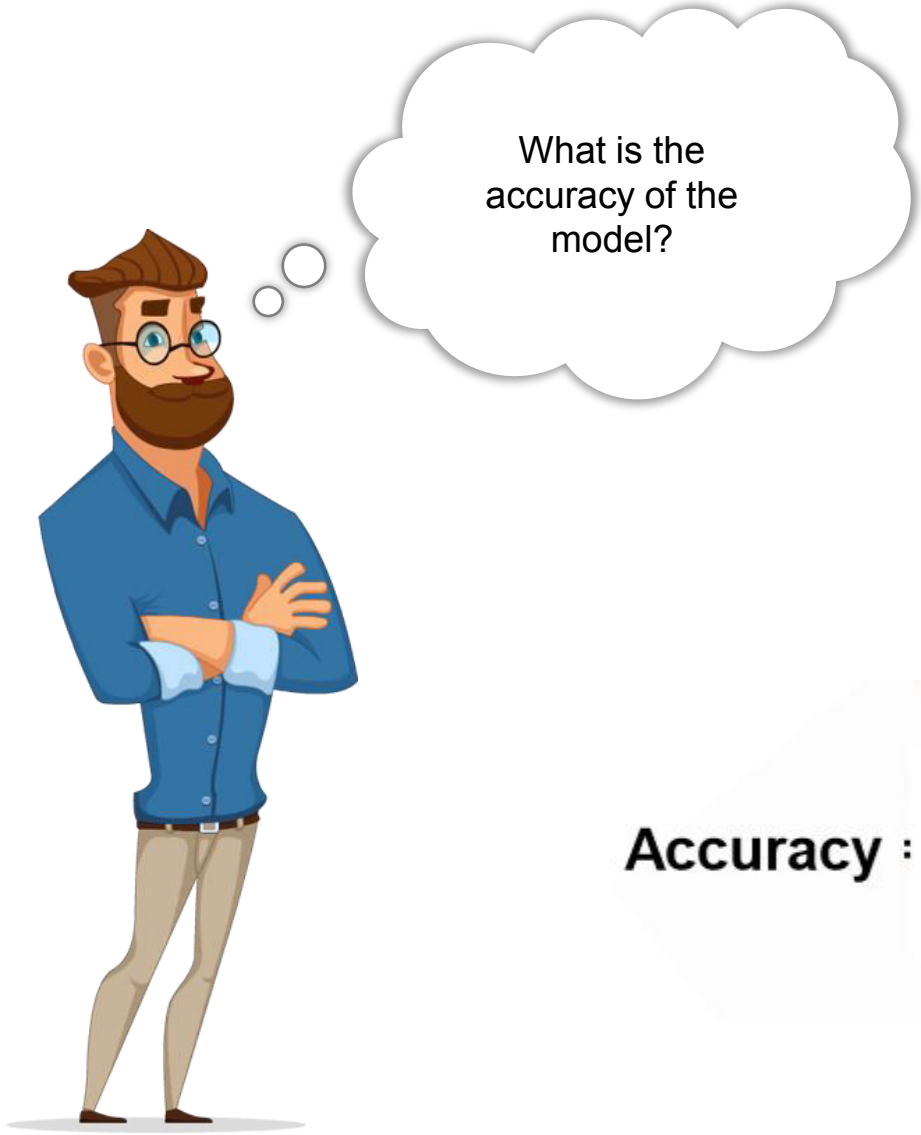
False Negatives



IntelliPaat

Performance Metrics

Accuracy



		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

$$\text{Accuracy} : \frac{\text{True Positive + True Negatives}}{\text{True Positive + True Negatives + False Positives + False Negatives}}$$

Accuracy

100	25
30	47

Accuracy :

$$\frac{100 + 47}{100 + 47 + 30 + 25} \rightarrow 0.72$$

Precision

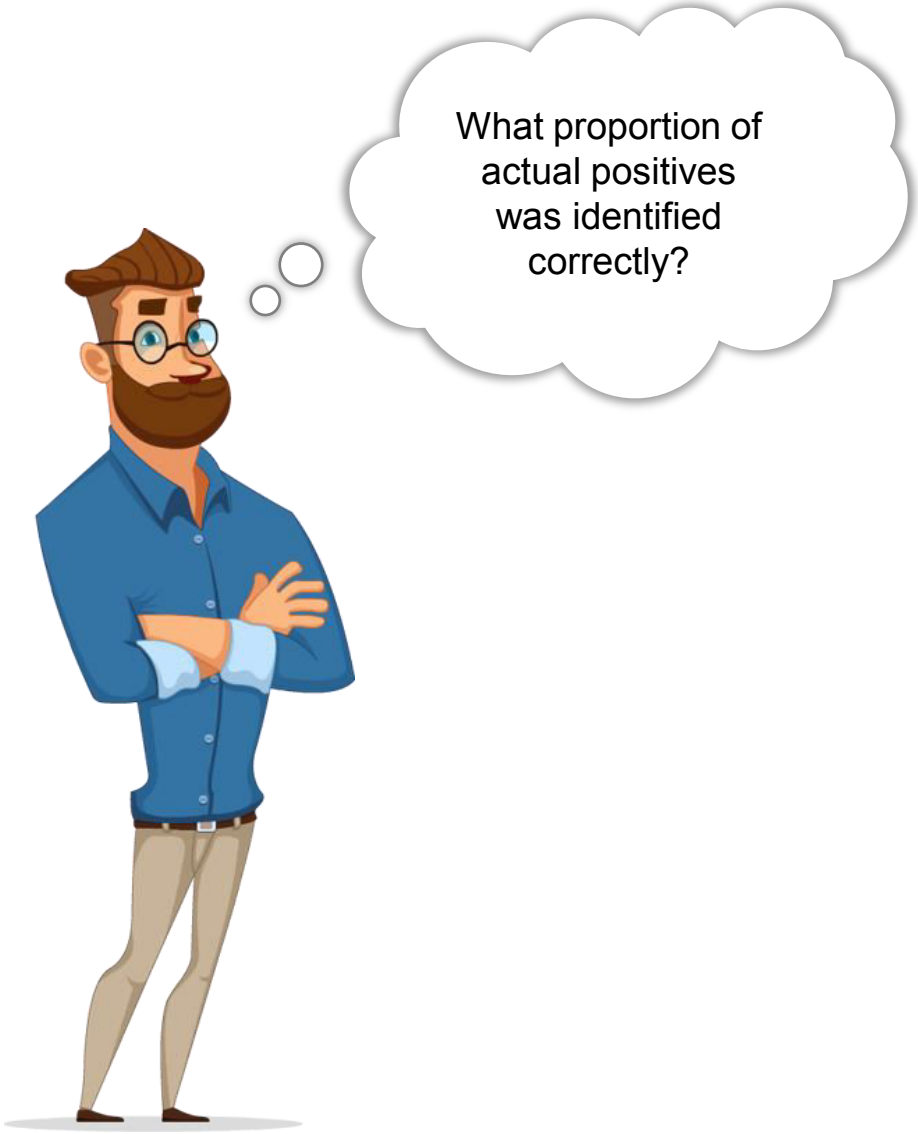
		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

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

Precision

$$\text{Precision} = \frac{100}{100 + 30} \rightarrow 0.76$$

Recall



		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

$$\text{Recall} = TP / TP+FN$$

Recall

$$\text{Recall} = \frac{100}{100 + 25} \rightarrow 0.8$$

Thresholding

Thresholding

Logistic regression returns a **probability,** and we would like to convert the returned probability into a **binary value.**

Threshold

Thresholding

Probability	Threshold	Result
0.73	0.65	Yes
0.59	0.65	No

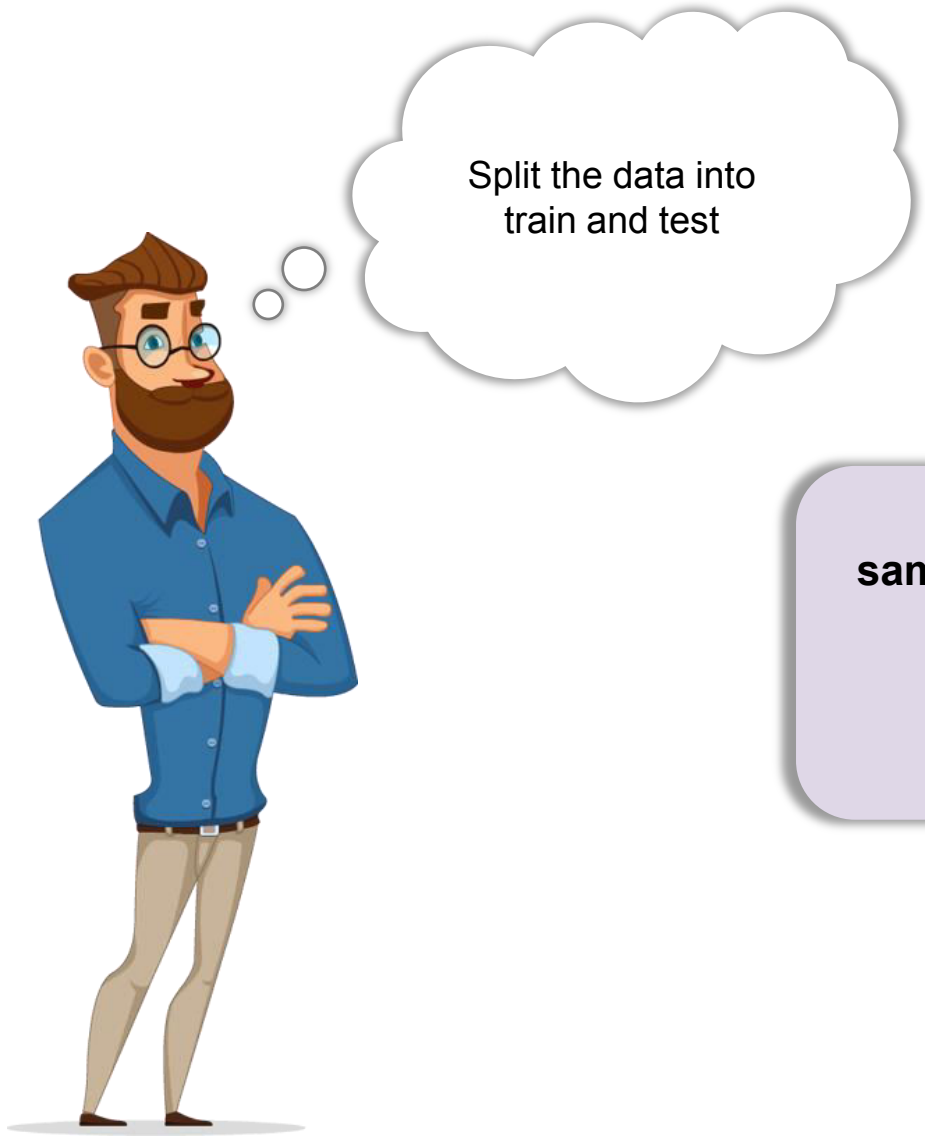
Confusion Matrix in R

Tasks to be Performed

1. Build a simple logistic regression model where the dependent variable is “Churn” and the independent variable is “MonthlyCharges”

2. Predict values and build the confusion matrix to calculate accuracy

Confusion Matrix in R




```
sample.split(customer_churn$Churn,SplitRatio = 0.65)-> split_tag
subset(customer_churn, split_tag==T)->train
subset(customer_churn, split_tag==F)->test
```

Confusion Matrix in R


```
glm(Churn~MonthlyCharges, data=train, family = "binomial")-> mod_log
```

```
predict(mod_log,newdata=test,type="response")->result_log
```

Confusion Matrix in R


```
table(test$Churn, result_log>0.3)
```

```
##       FALSE TRUE
##   No   1169  642
##   Yes   293  361
```

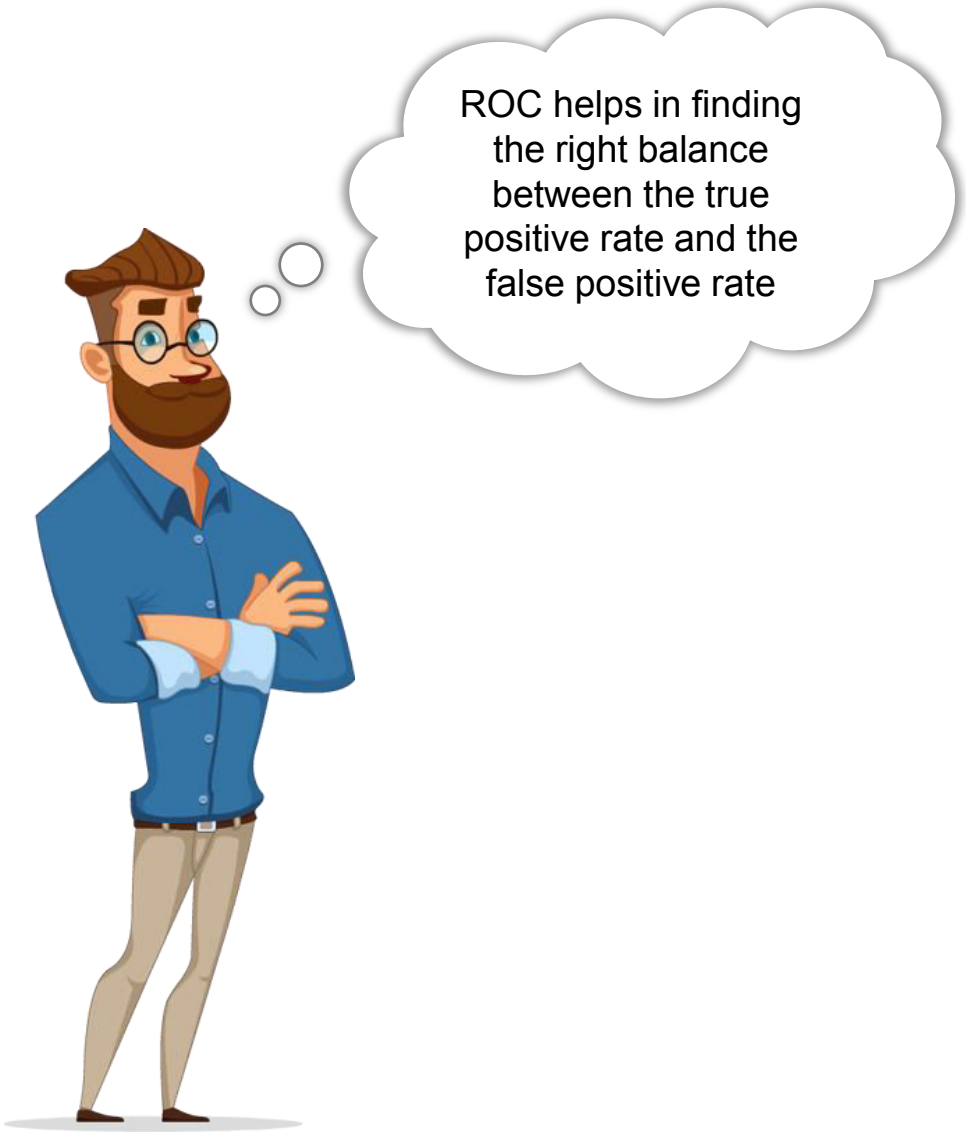
Confusion Matrix in R

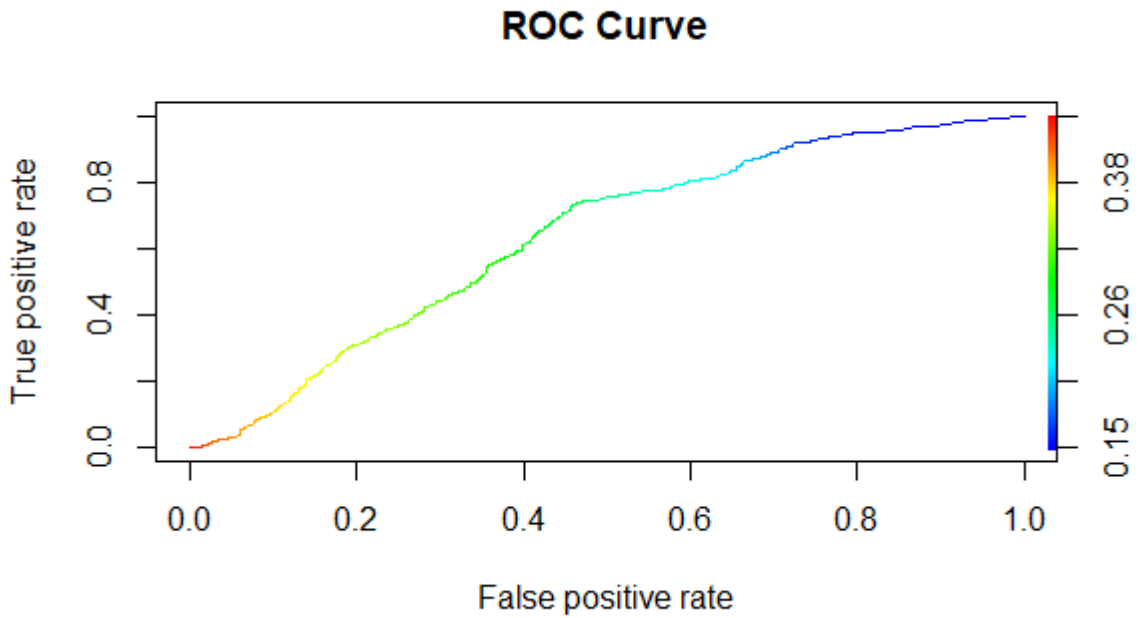

```
table(test$Churn, result_log>0.35)
```

```
##       FALSE TRUE
##   No   1475  336
##   Yes   475  179
```

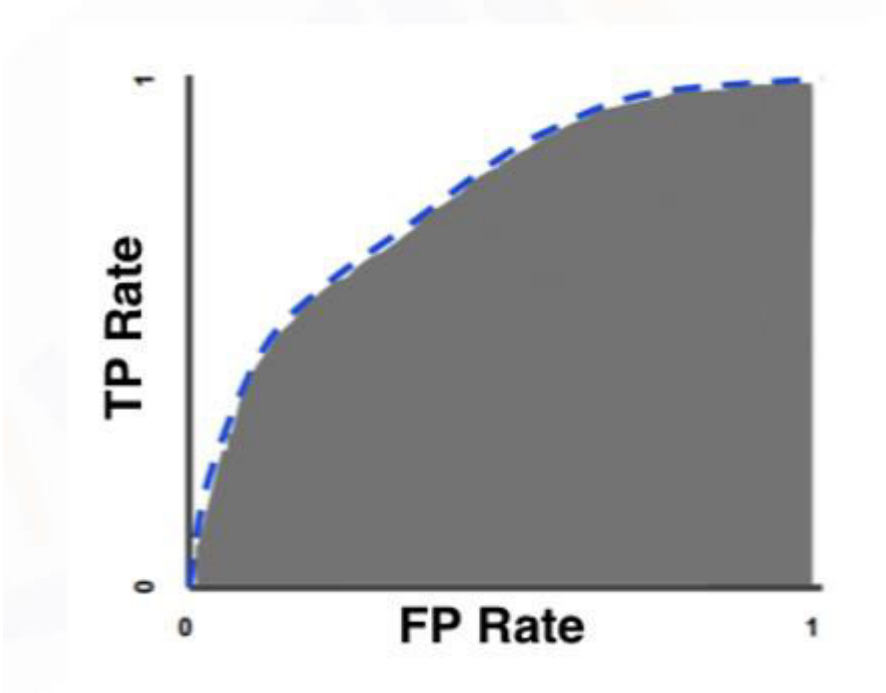
ROCR Package

ROC Curve





AUC



Working with ROCR Package

Tasks to be Performed

1. Build a logistic regression model where the dependent variable is “Churn” and the independent variable is “MonthlyCharges”

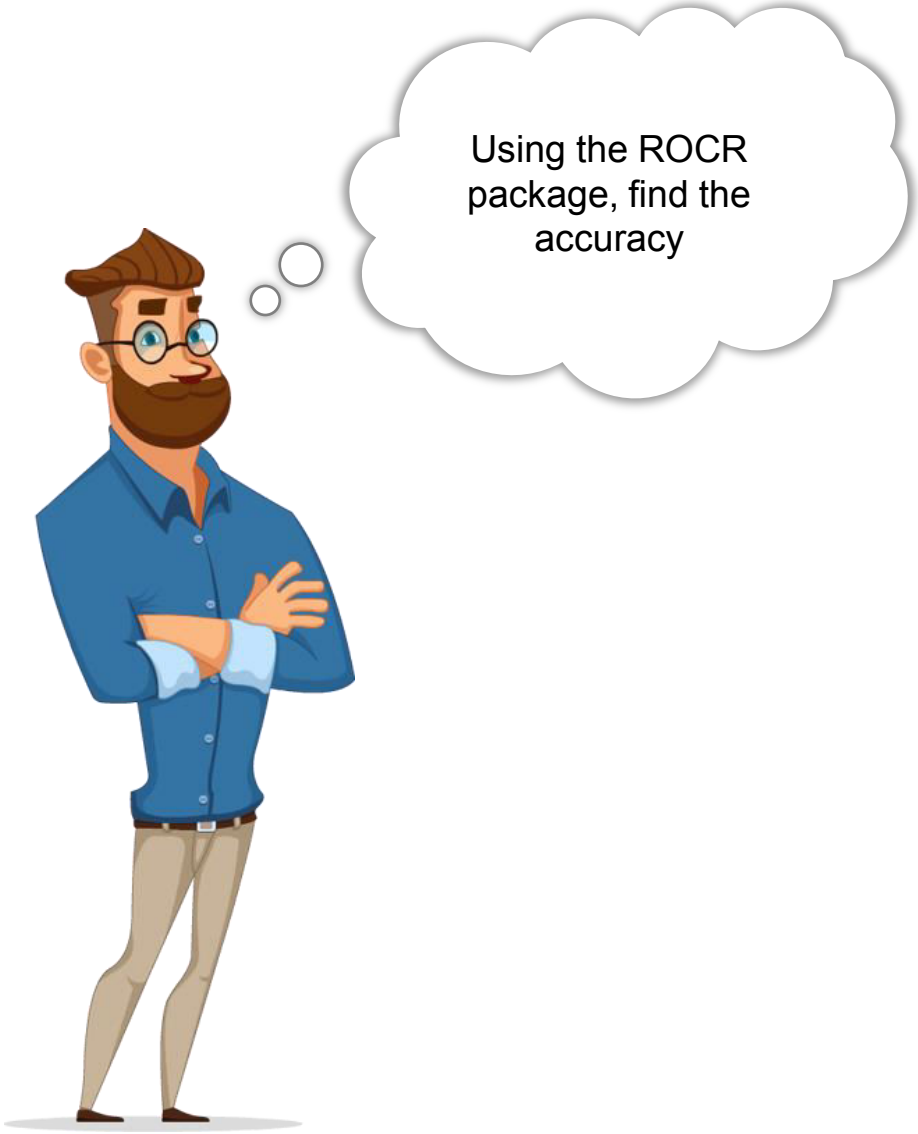
2. Using the ROCR package, find out the accuracy, ROC curve and AUC

ROCR Package


```
glm(Churn~MonthlyCharges, data=train, family = "binomial")-> mod_log
```

```
predict(mod_log,newdata=test,type="response")->result_log
```

ROCR Package



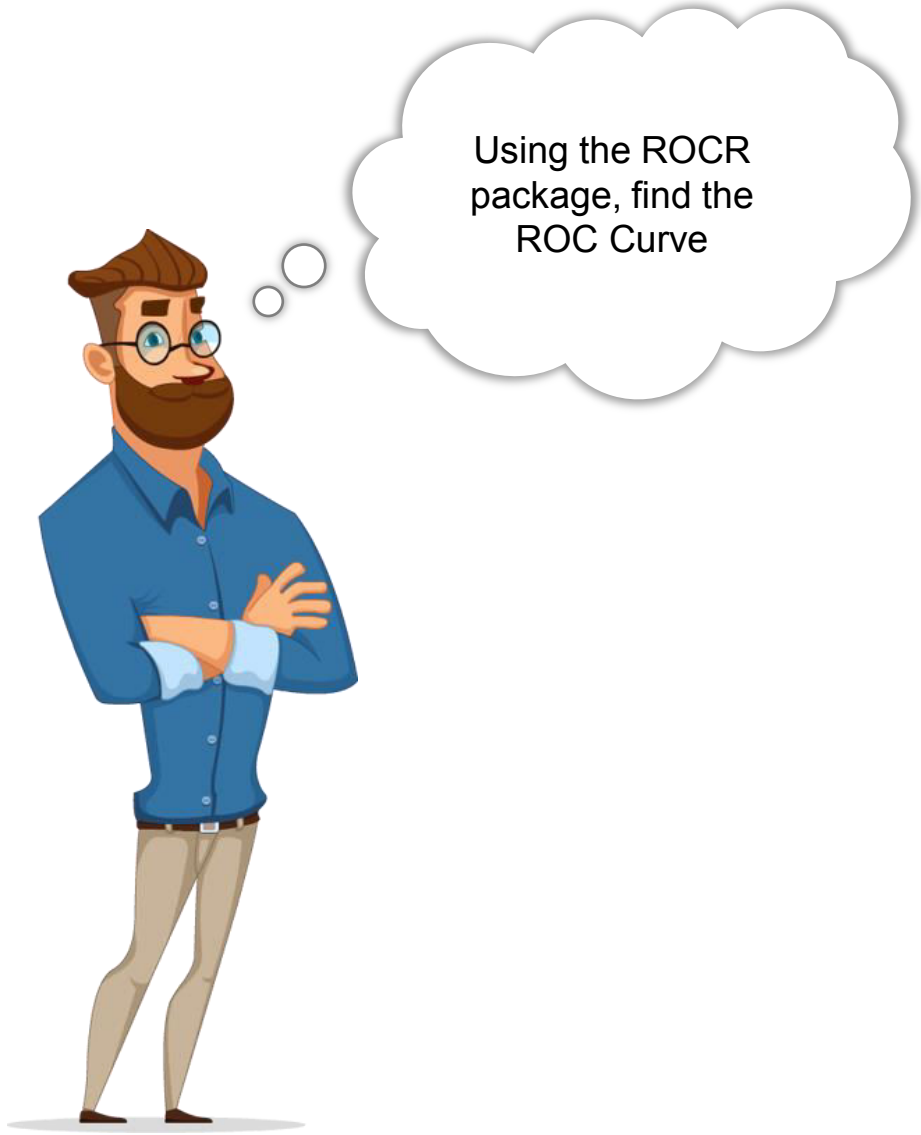

```
library(ROCR)

prediction(result_log,test$Churn) -> predict_log

performance(predict_log,"acc")->acc

plot(acc)
```

ROCR Package



performance(predict_log,"tpr","fpr") -> roc_curve

plot(roc_curve)

plot(roc_curve, colorize=T)

ROCR Package


```
performance(predict_log,"auc")->auc

auc
```

Quiz

Quiz

Is logistic regression a supervised machine learning algorithm?

A. True

B. False

Quiz

Is logistic regression a supervised machine learning algorithm?

Solution:

A. True

Quiz

Is logistic regression mainly used for regression?

A. True

B. False

Quiz

Is logistic regression mainly used for regression?

Solution:

B. False

Quiz

Which of the following methods do we use to best fit the data in logistic regression?

A. Least Square Error

B. Maximum Likelihood

C. Jaccard Distance

D. Both A and B

Quiz

Which of the following methods do we use to best fit the data in logistic regression?

Solution:

B. Maximum Likelihood

Quiz

How would you create a simple logistic regression model where the dependent variable is ‘gender’ & the independent variable is ‘MonthlyCharges’?

A. lm(gender=MonthlyCharges, data= customer_churn, family="binomial")

B. glm(gender~MonthlyCharges, data= customer_churn, family="logistic")

C. glm(gender~MonthlyCharges, data= customer_churn, family="binomial")

D. glm(gender~MonthlyCharges, data= customer_churn)

Quiz

How would you create a simple logistic regression model where the dependent variable is ‘gender’ & the independent variable is ‘MonthlyCharges’?

Solution:

C. glm(gender~MonthlyCharges, data= customer_churn, family="binomial")

Quiz

Which function is used to create the ROC curve?

A. ROC()

B. Predict()

C. Performance()

D. Roc_plot()

Quiz

Which function is used to create the ROC curve?

Solution:

A. Performance()

Thank You

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24/7 Chat with Our Course Advisor