Data Science

Machine Learning

Agenda

01 Understanding Machine Learning

02 Supervised Learning

03 Unsupervised Learning

04 Linear Regression

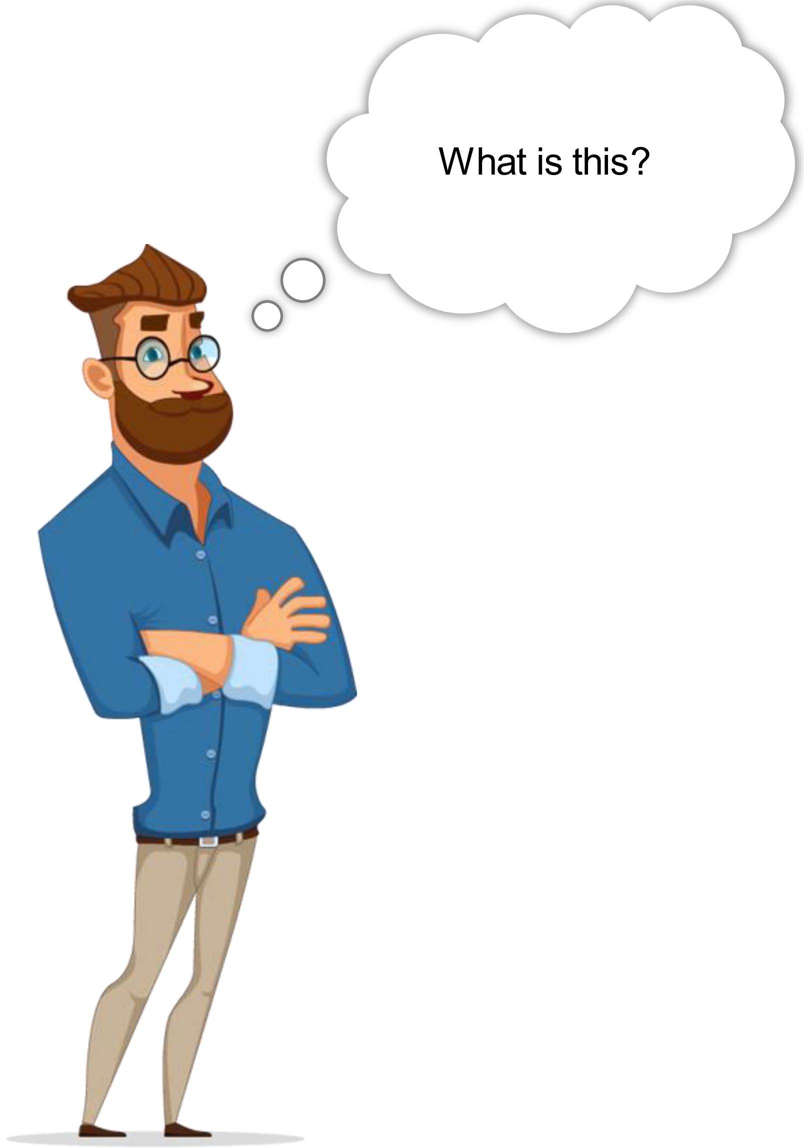
Understanding Machine Learning

Understanding Machine Learning

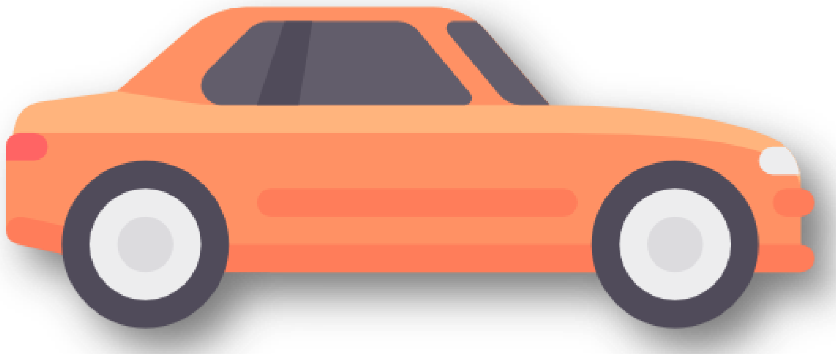
Machine learning is a field of study that provides systems the ability to automatically learn and improve from experience without being explicitly programmed

Understanding Machine Learning through an Example

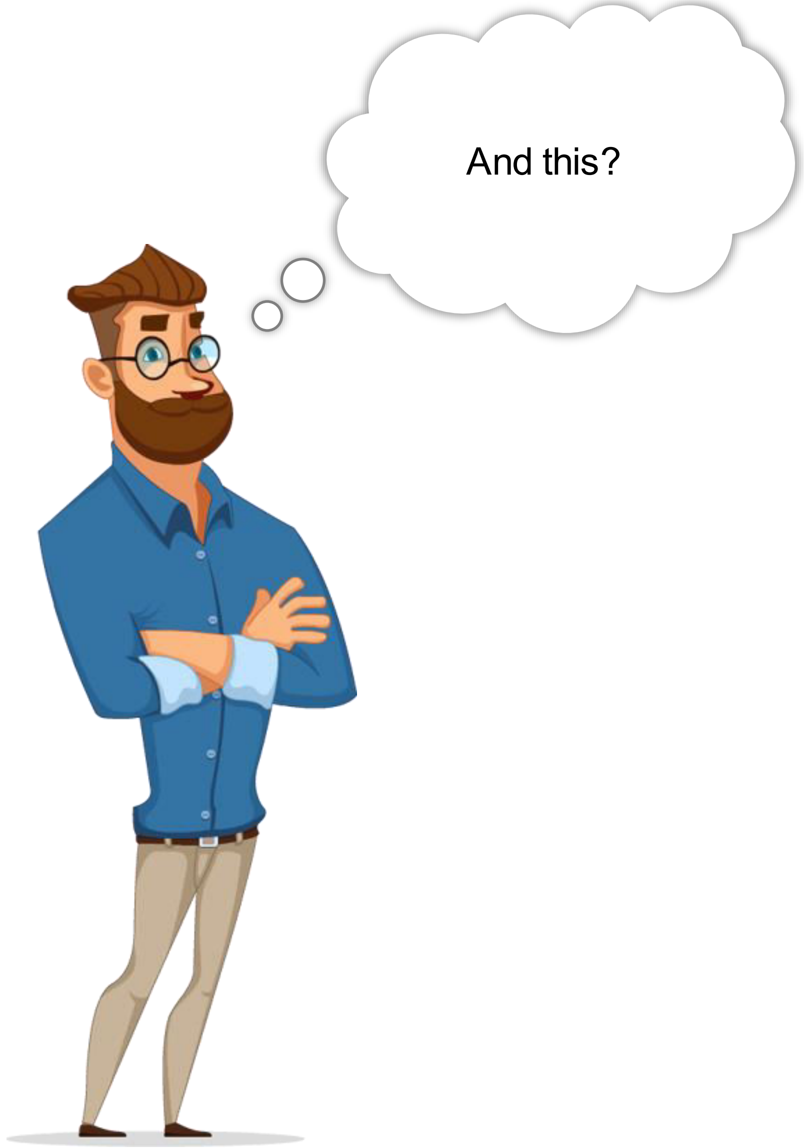
Understanding Machine Learning



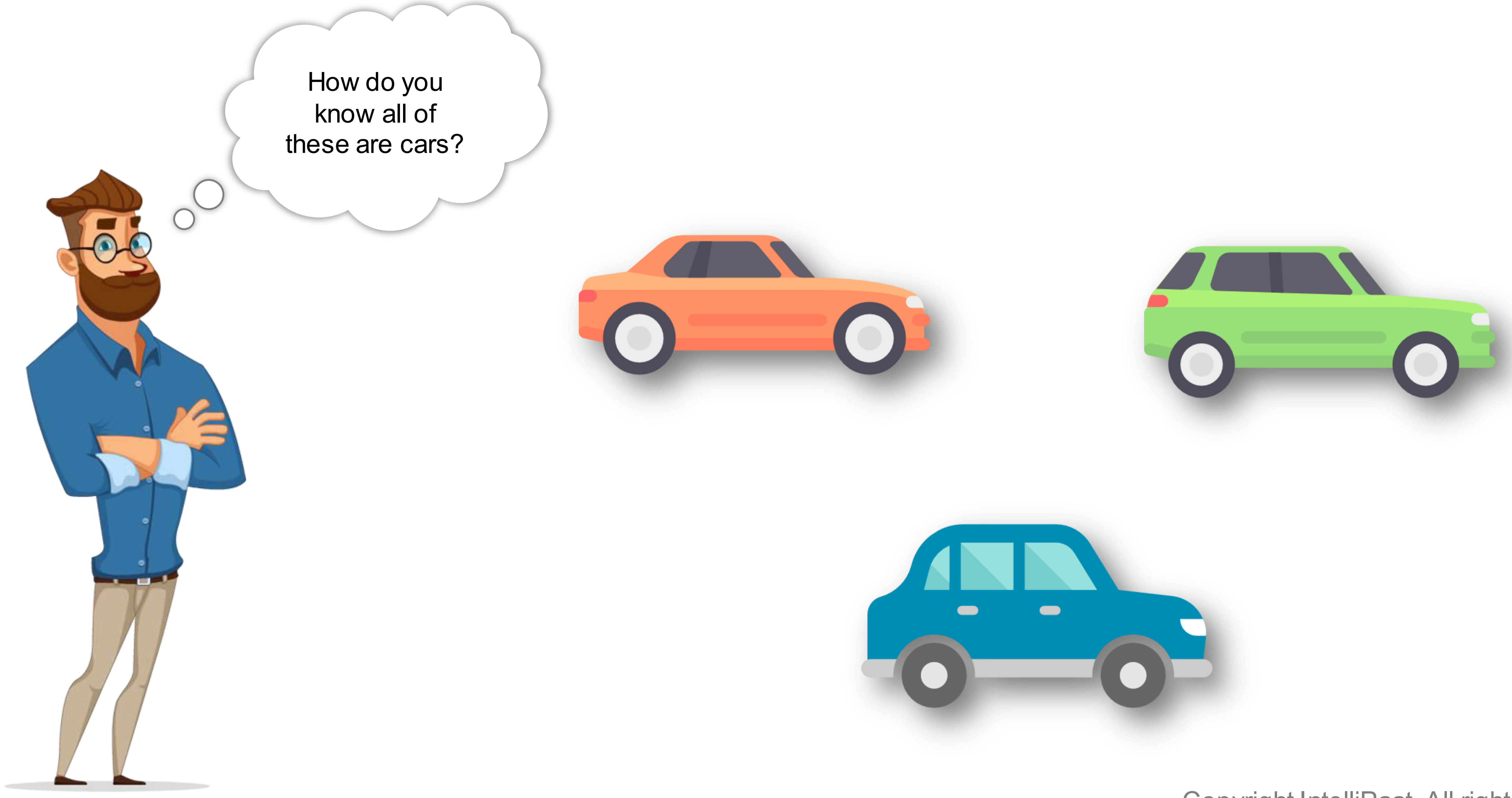
Understanding Machine Learning



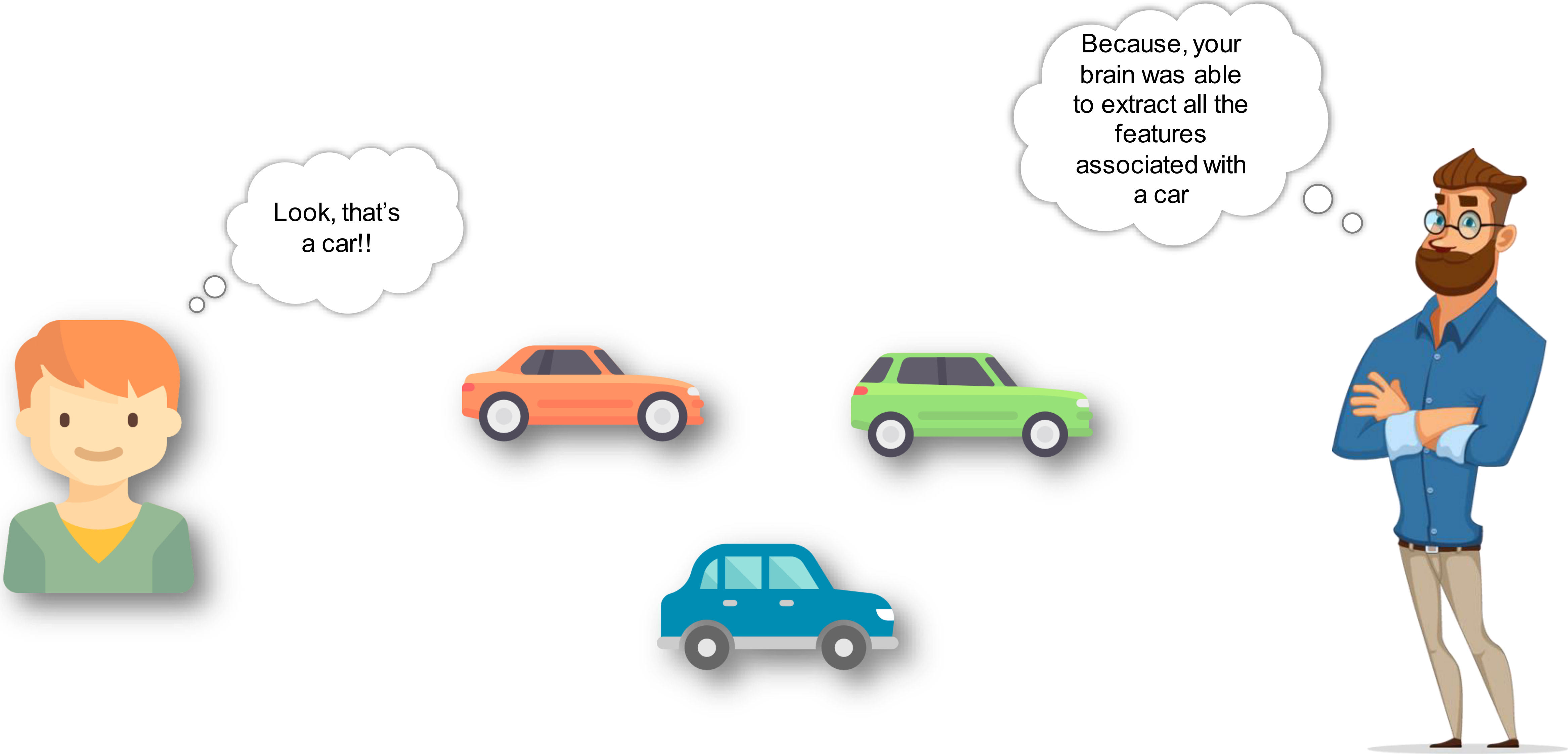
Understanding Machine Learning



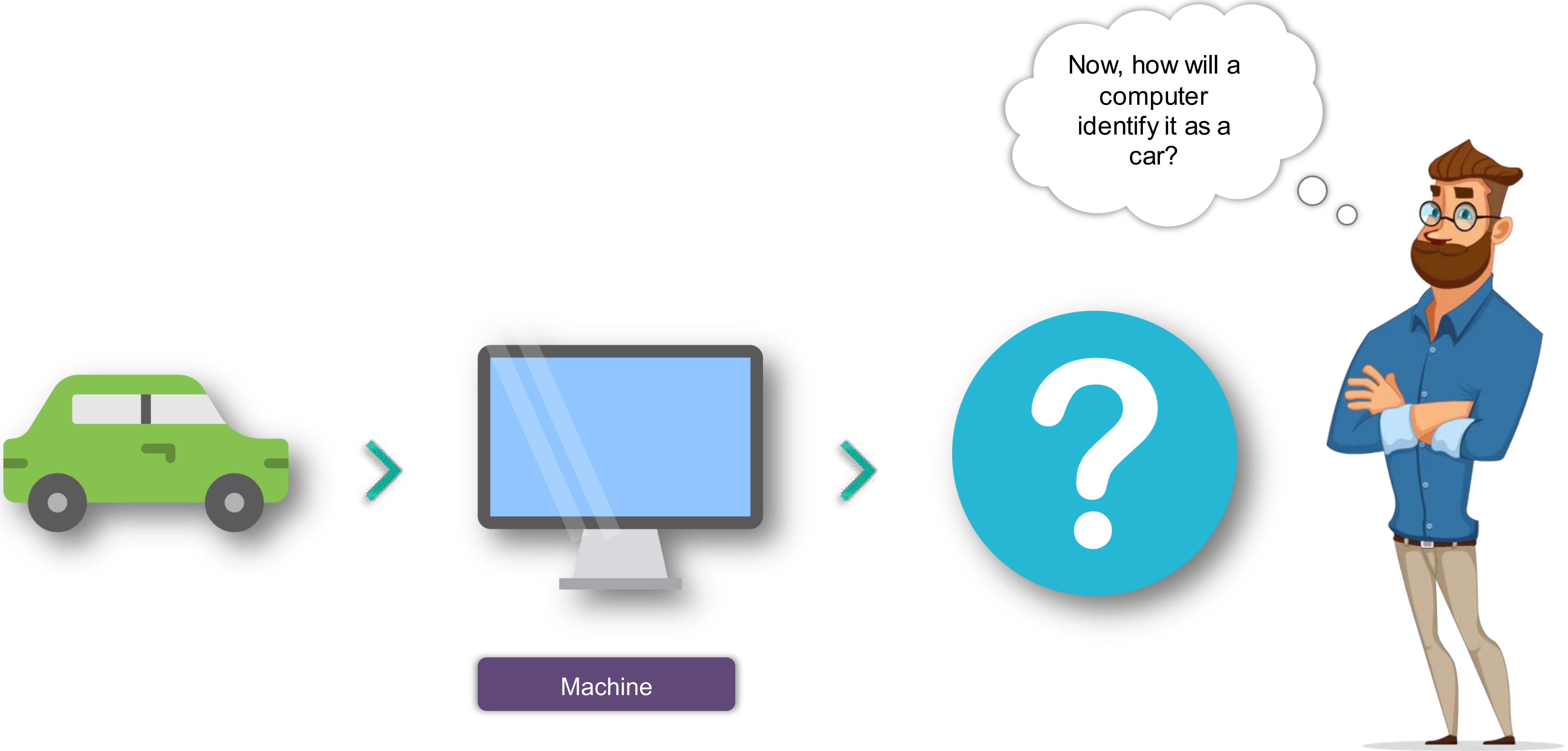
Understanding Machine Learning



Understanding Machine Learning

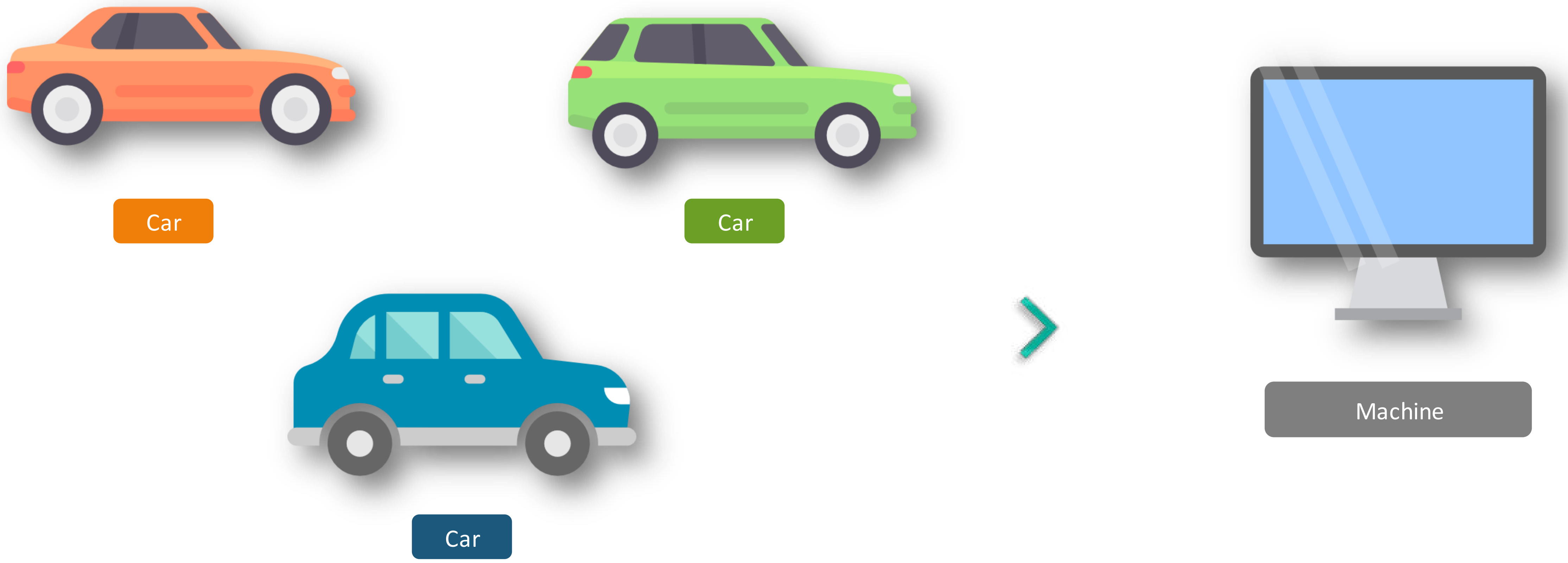


Understanding Machine Learning



Understanding Machine Learning

The machine would be fed data, so that it can learn it's features



Understanding Machine Learning

Evaluating how much the machine has learnt from the data

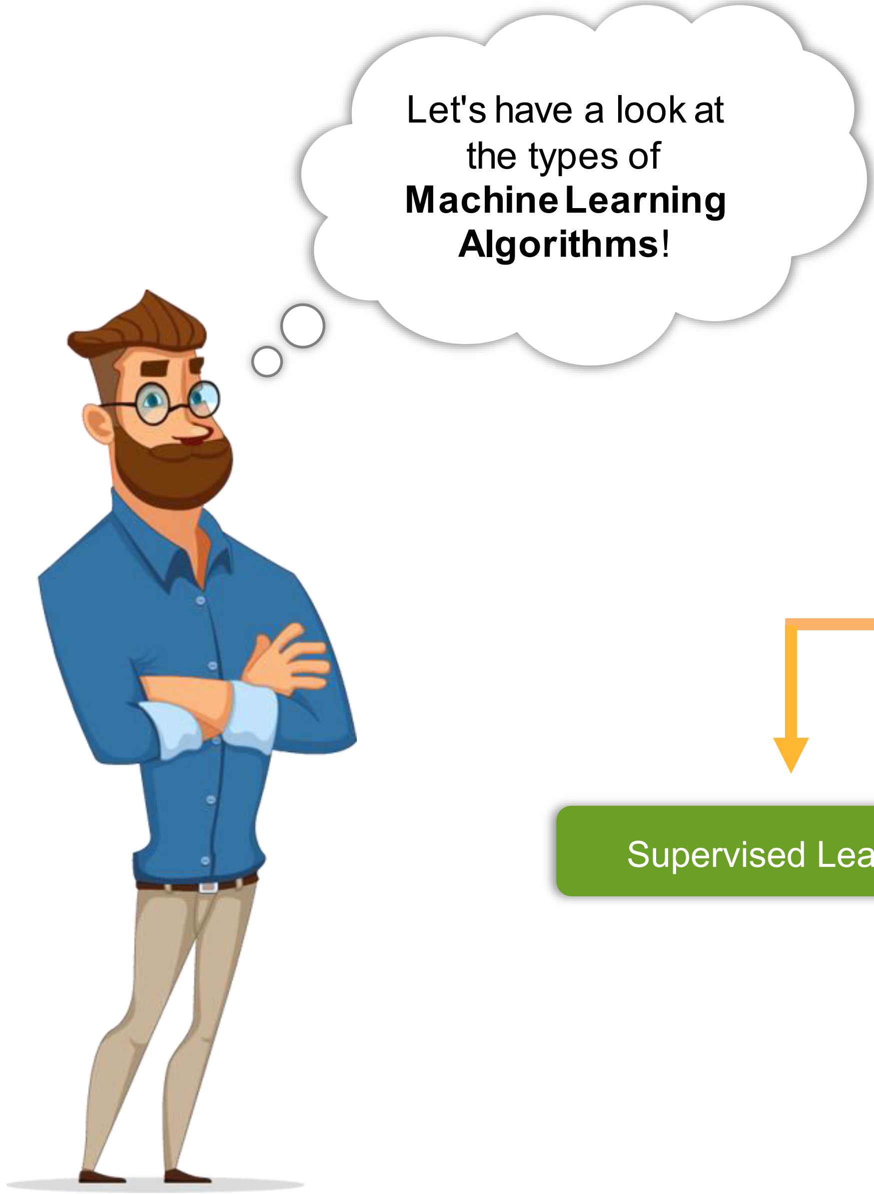
New Data

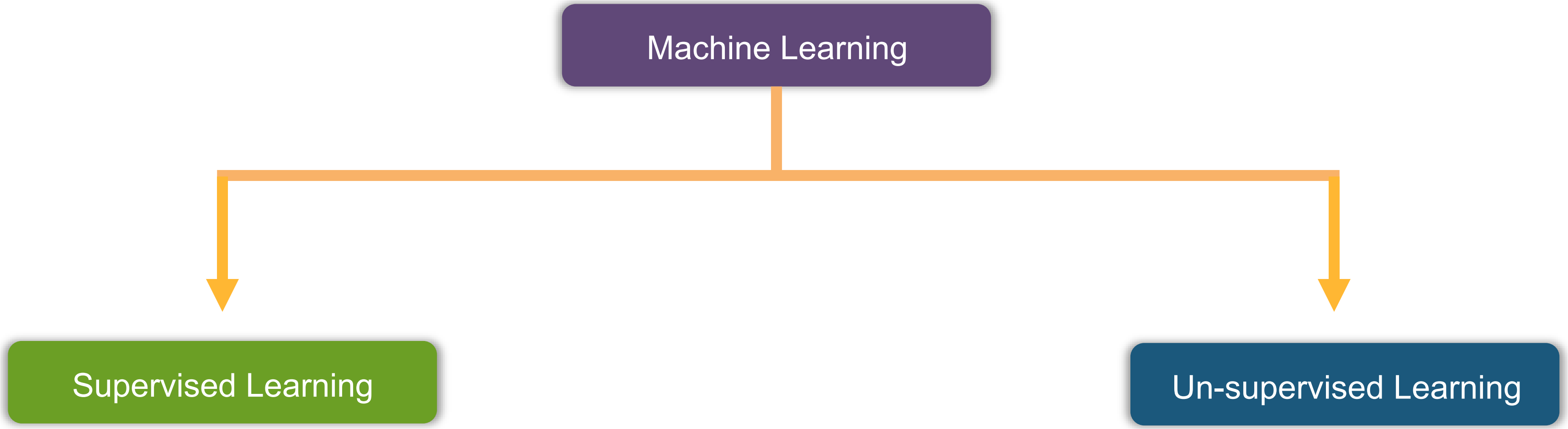
Machine

Result

Types of Machine Learning

Types of Machine Learning Algorithms





Supervised Learning Models

Supervised Learning Models

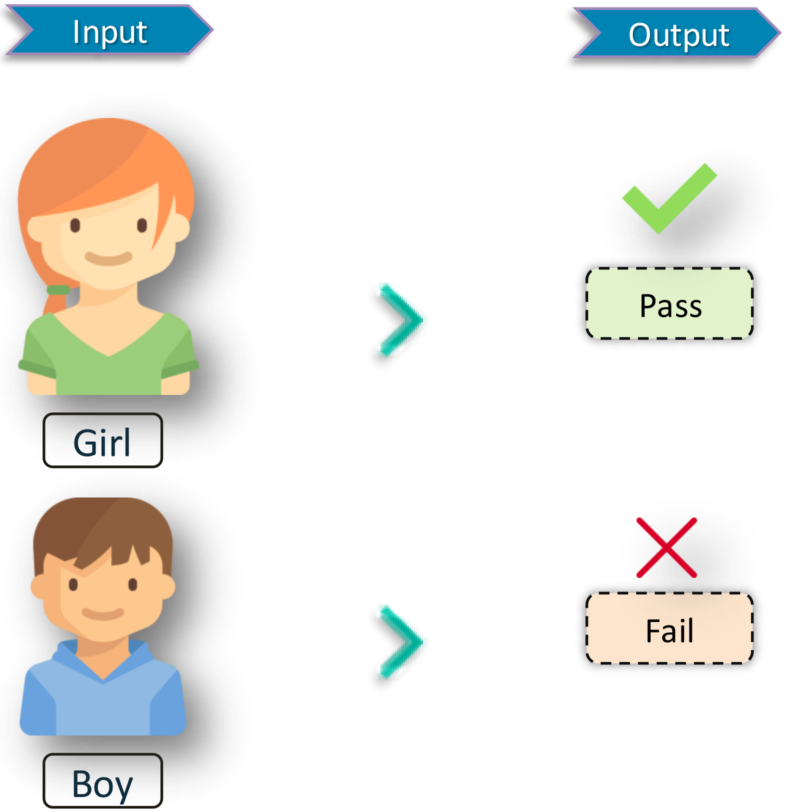
Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data

Supervised Learning

Here, the input variable is “Gender” and the output variable is “Result” and we are mapping a function between “Result” & “Gender”



$$Y=f(x)$$

Supervised Learning Models

Supervised learning problems can be further grouped into regression and classification problems

Classification

A classification problem is when the output variable is a category

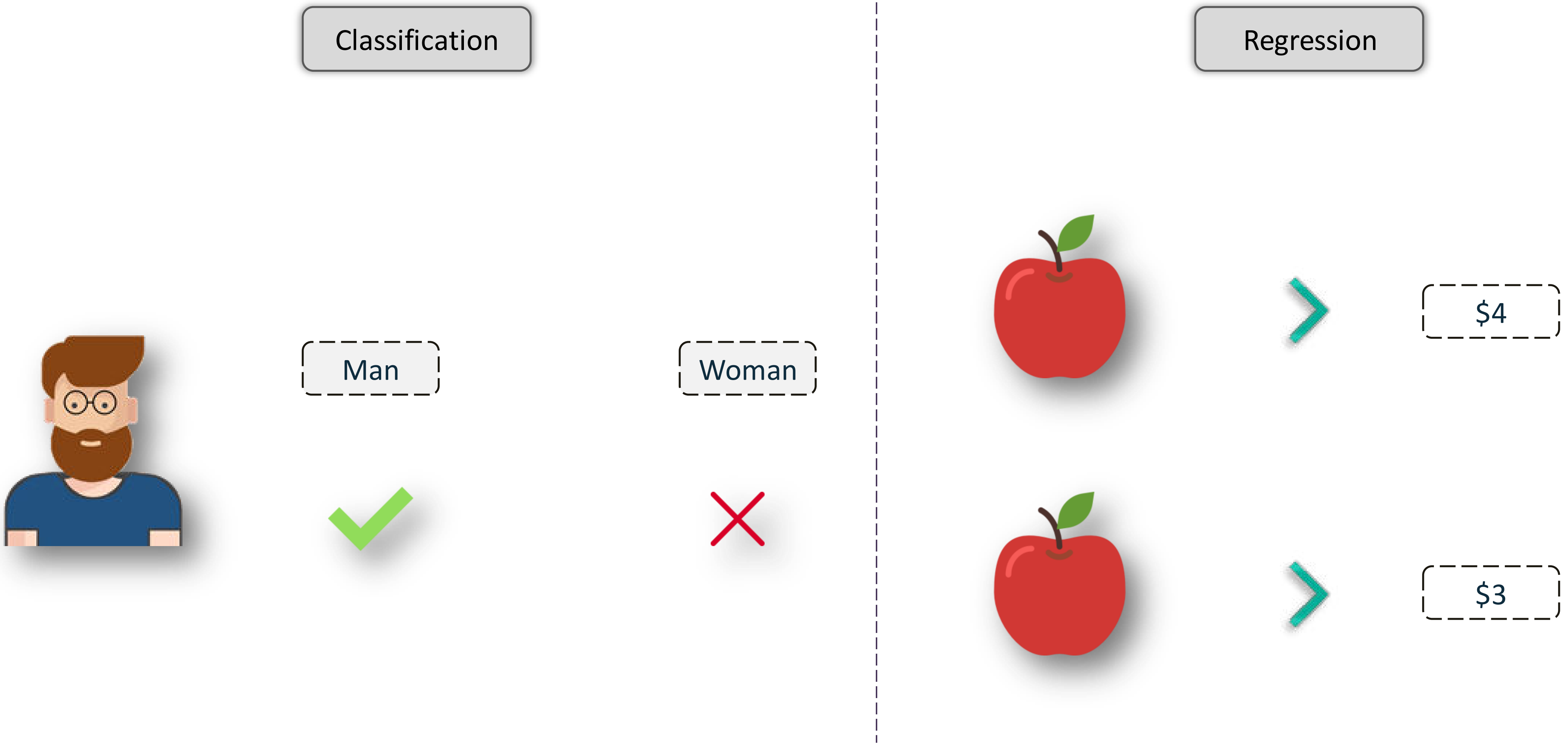
A classification problem is when the output variable is a category

Regression

A regression problem is when the output variable is a real value, such as “dollars” or “weight”

Given a size of the house predict the price (real value)

Types of Supervised Learning



Unsupervised Learning Models

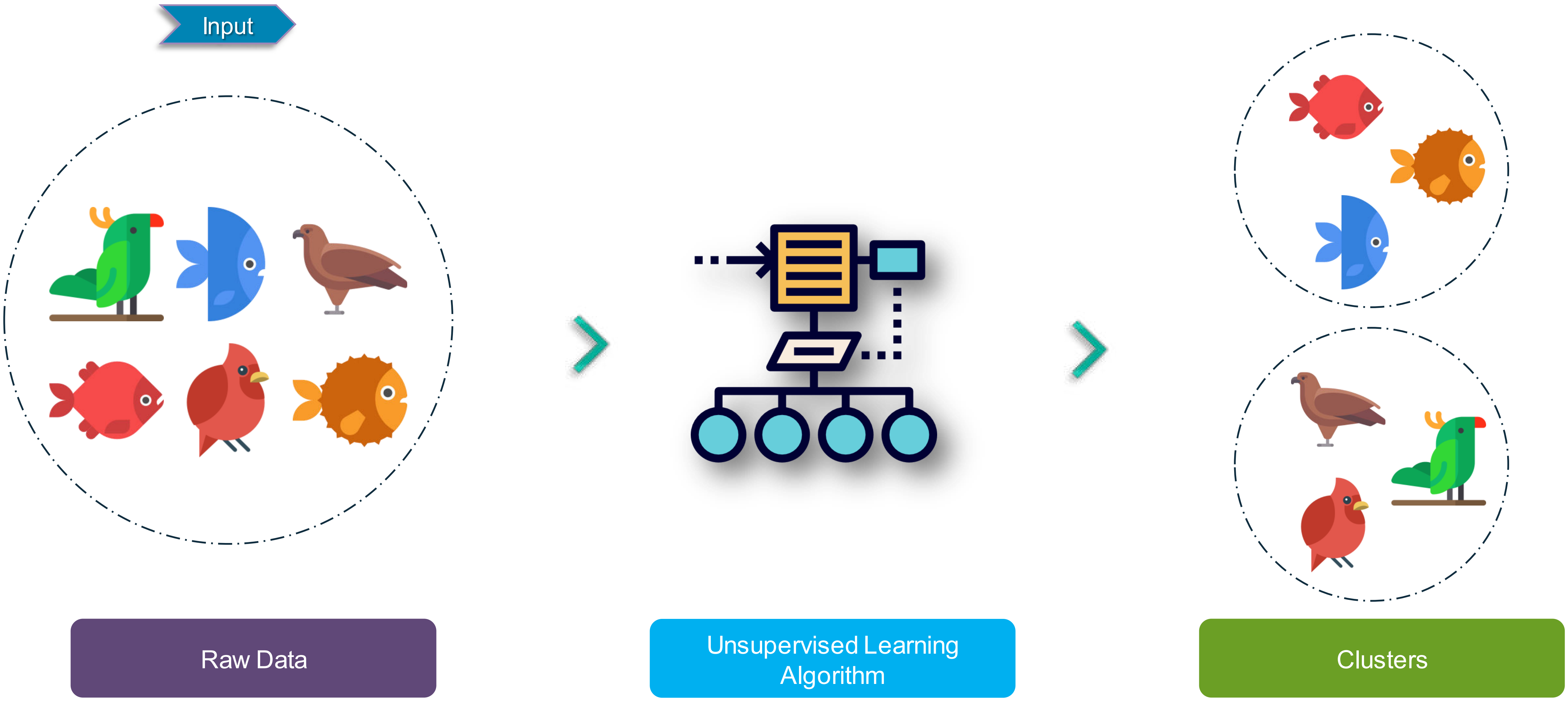
Unsupervised Learning Models

Unsupervised learning is where you only have input data (X) and no corresponding output variables

The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data

Some common use-cases for unsupervised learning are **exploratory analysis, clustering and dimensionality reduction**

Unsupervised Learning Models

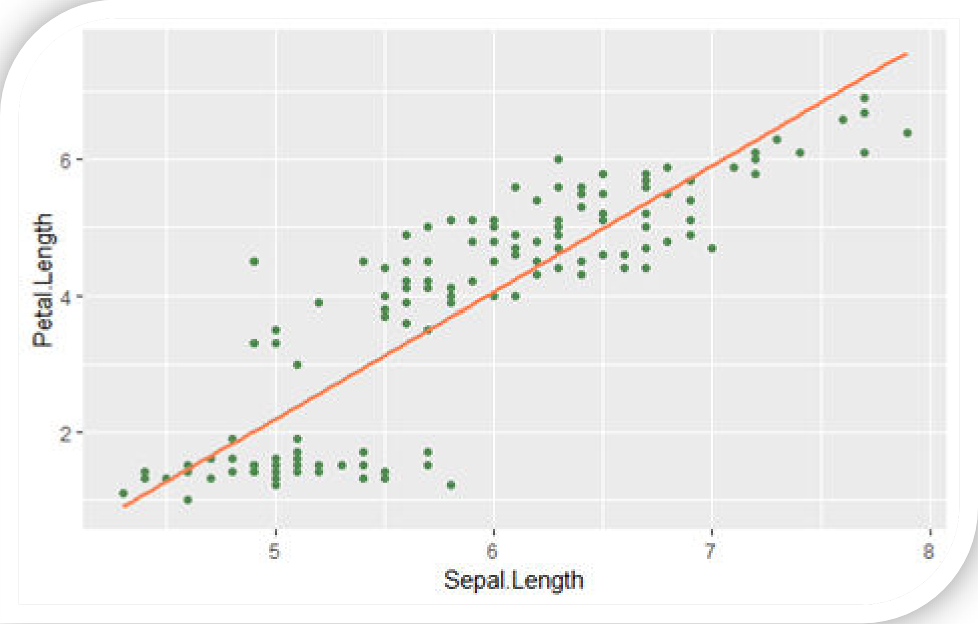


Introduction to Linear Regression

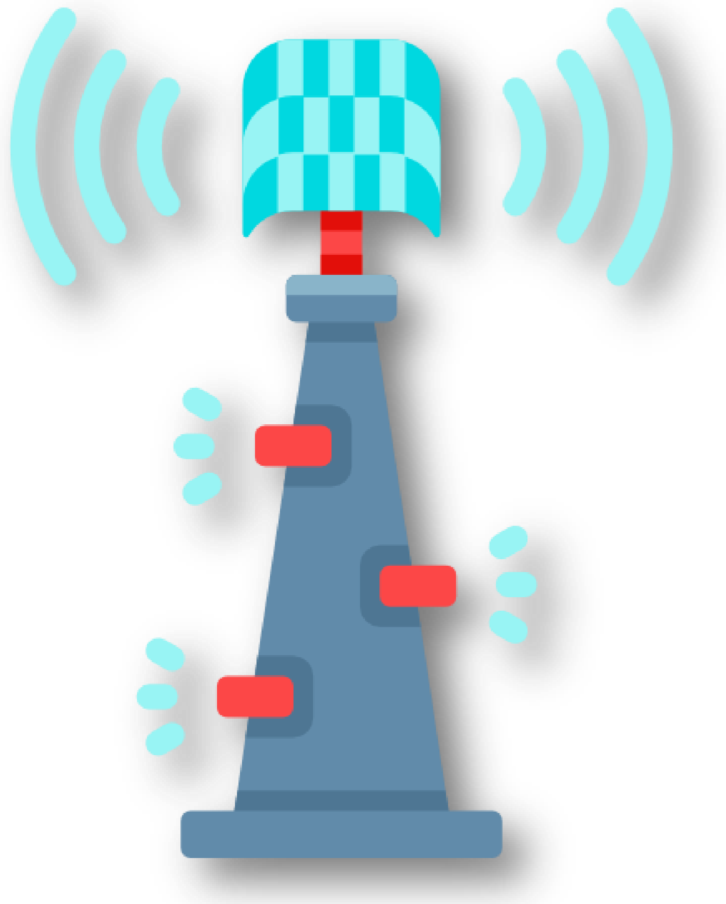
Linear Regression

It helps in understanding the ***linear*** relationship between **dependent** & **independent** variables

$$y = b_0 + b_1 x$$



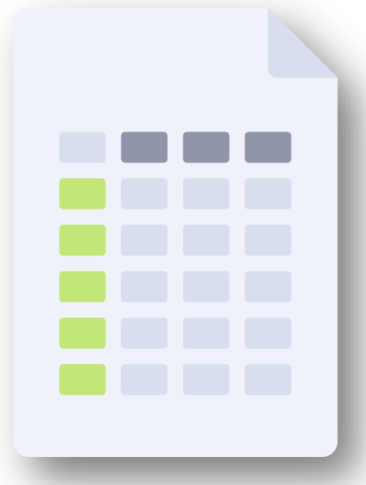
Linear Regression Example

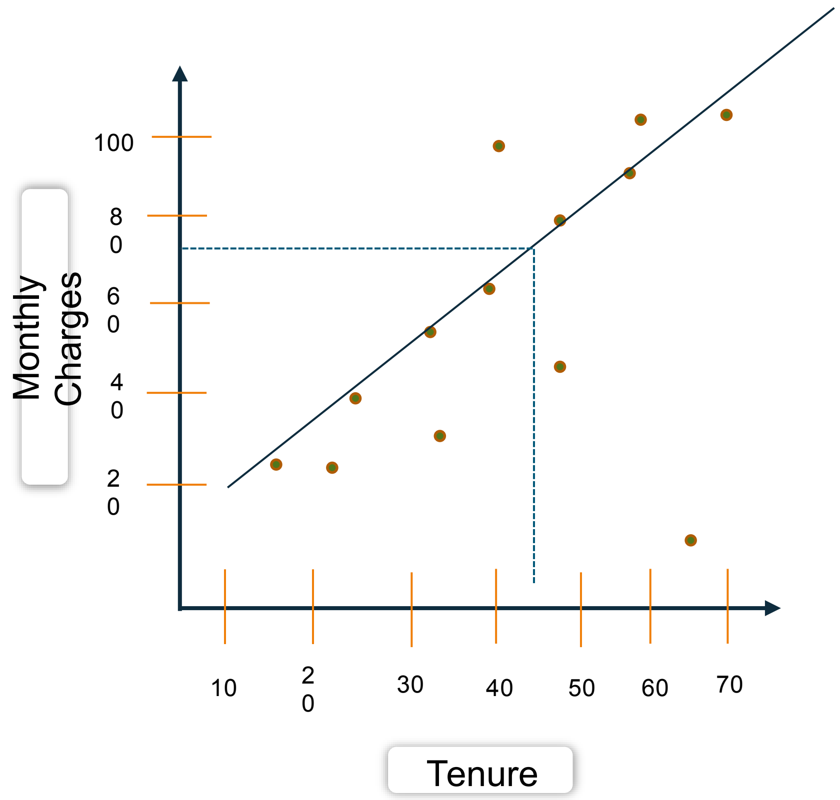


Neo

Linear Regression Example

Estimating the value of “Monthly Charges” when “Tenure” of the customer changes





Error Term in Linear Regression Equation

In general, the data doesn't fall exactly on a line, so the regression equation should include an implicit **error term**

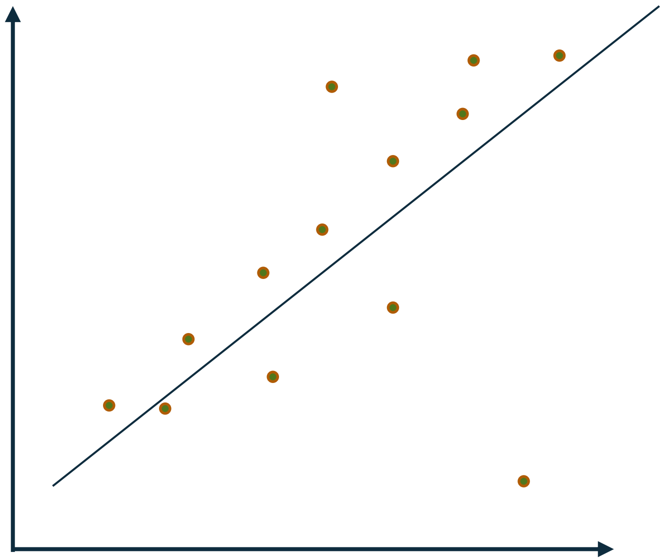
The **fitted values** (predicted values) are typically denoted by Y-hat

$$Y_i = b_0 + b_1 X_i + e_i$$

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i$$

Exploring b1

If b1 > 0, then x(predictor) and y(target) have a positive relationship. That is increase in x will increase y



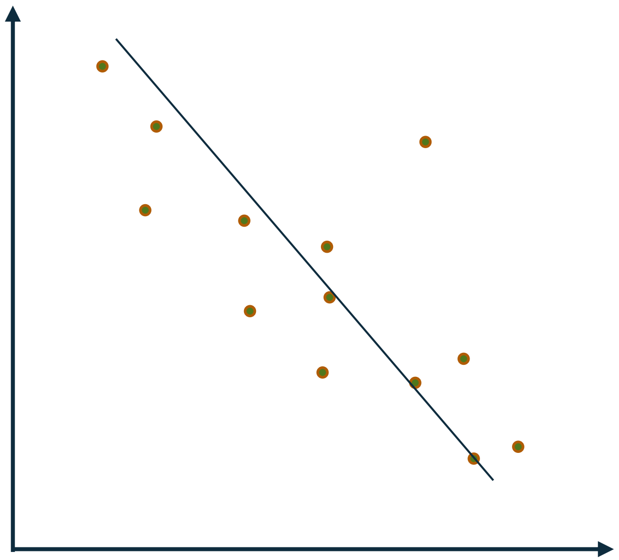
$$y = b_0 + b_1 x$$

b1 > 0

Positive Relationship

Exploring b1

If b1 < 0, then x(predictor) and y(target) have a negative relationship. That is increase in x will decrease y

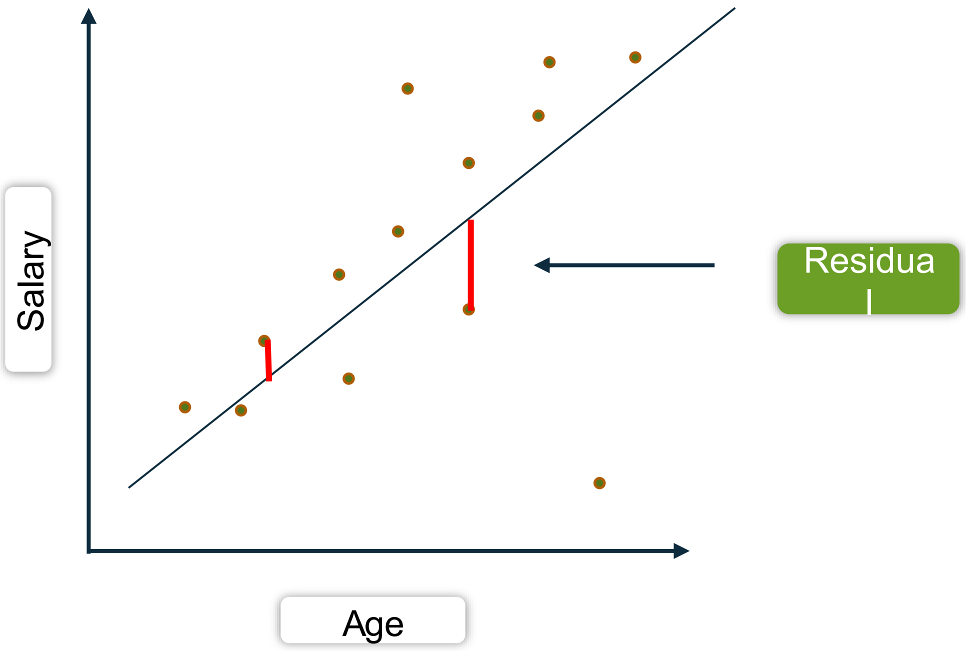


$$y = b_0 + b_1 x$$

b1 < 0 → Negative Relationship

Residuals

The difference between the observed value of the dependent variable (y) and the predicted value ($\hat{y}$) is called the residual. Each data point has one residual

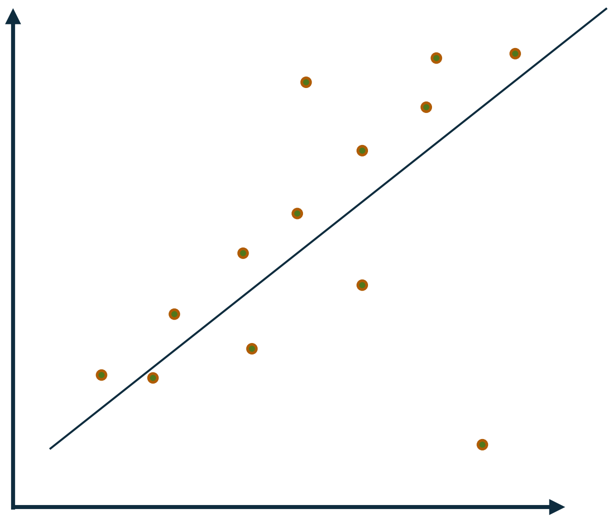


Residual = Observed value - Predicted value

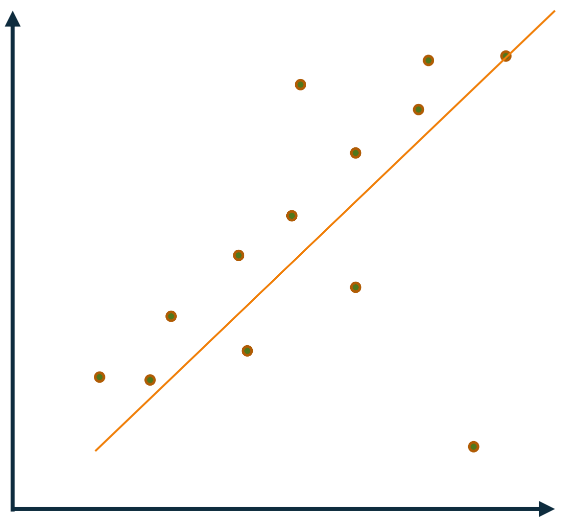
Line of Best Fit

Line of Best Fit

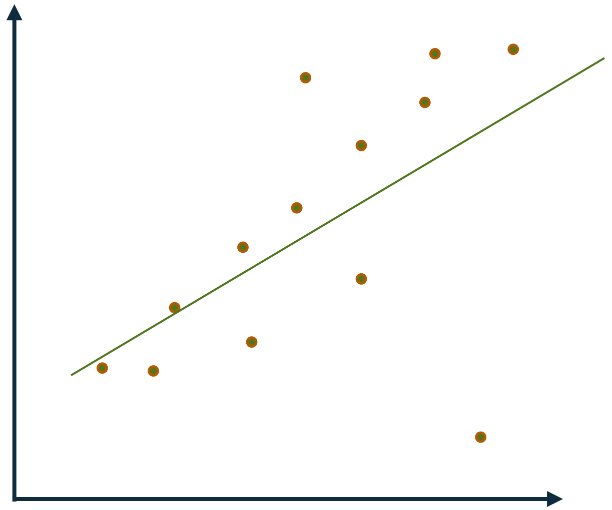
There could be multiple fit lines passing through the points



Fit Line 1



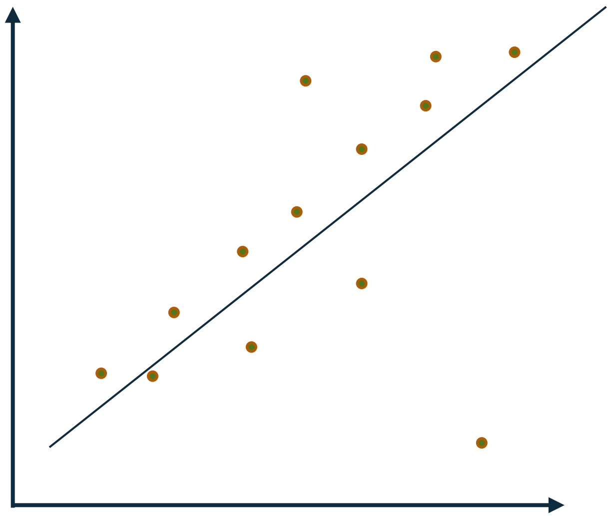
Fit Line 2



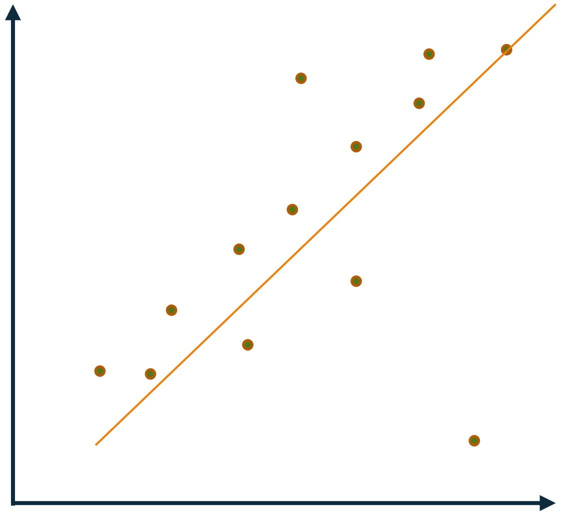
Fit Line 3

Line of Best Fit

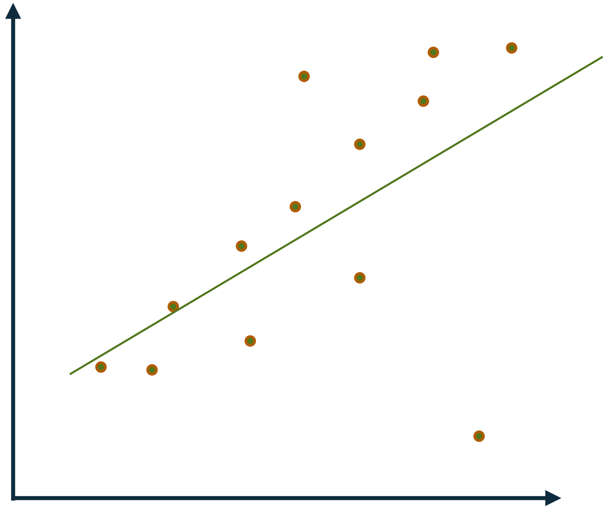
The line with the lowest value of Residual sum of Squares would be the best fit line



RSS=120



RSS=80



RSS=132

$$\text{RSS} = \sum_{k=1}^{n} (Actual - Predicted)^2$$

Math Behind Linear Regression

Math Behind Linear Regression

In practice, the regression line is the estimate that minimizes the sum of the squared residual values, also called the **residual sum of squares (RSS)**

$$RSS = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$= \sum_{i=1}^{n}(Y_i - \hat{b}_0 - \hat{b}_1 X_i)^2$$

Math Behind Linear Regression

Sum of squared Errors is given by:

$$SS_{(residuals)} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

If $\hat{y} = b_0 + b_1 x$ then error in estimate for x_i is $e_i = y_i - \hat{y}_i$

Minimize Sum of Squared Errors (SSE)

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$$

To minimize the error, 1st order derivative should be equal to zero:

$$\delta e / \delta b_0 = 0$$

$$\delta e / \delta b_1 = 0$$

So, we get 2 equations and 2 unknowns – b_0 and b_1

Math Behind Linear Regression

So we get:

$$\delta e/\delta b_0 = \sum_{i=1}^{n} 2\,(y_1 - b_0 - b_1x_1)\,(-1) = 0 \;............(1)$$

$$\delta e/\delta b_1 = \sum_{i=1}^{n} 2\,(y_1 - b_0 - b_1x_1)\,(-x_1) = 0 \;...........(2)$$

Expanding these equations, can calculate the b_0 & b_1

Math Behind Linear Regression

$$\hat{b}_1 = \frac{\sum_{i=1}^{n}(Y_i - \overline{Y})(X_i - \overline{X})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$$

$$\hat{b}_0 = \overline{Y} - \hat{b}_1\overline{X}$$

Coefficient of Determination (R Squared)

Coefficient of Determination(R Squared)

R Squared is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable

An **R Squared of 0** means that the dependent variable cannot be predicted from the independent variable

An **R Squared of 1** means the dependent variable can be predicted without error from the independent variable

An R Squared between 0 and 1 indicates the extent to which the dependent variable is predictable. An R Squared of 0.10 means that 10 percent of the variance in *Y* is predictable from *X*; an R Squared of 0.20 means that 20 percent is predictable; and so on

Coefficient of Determination(R Squared)

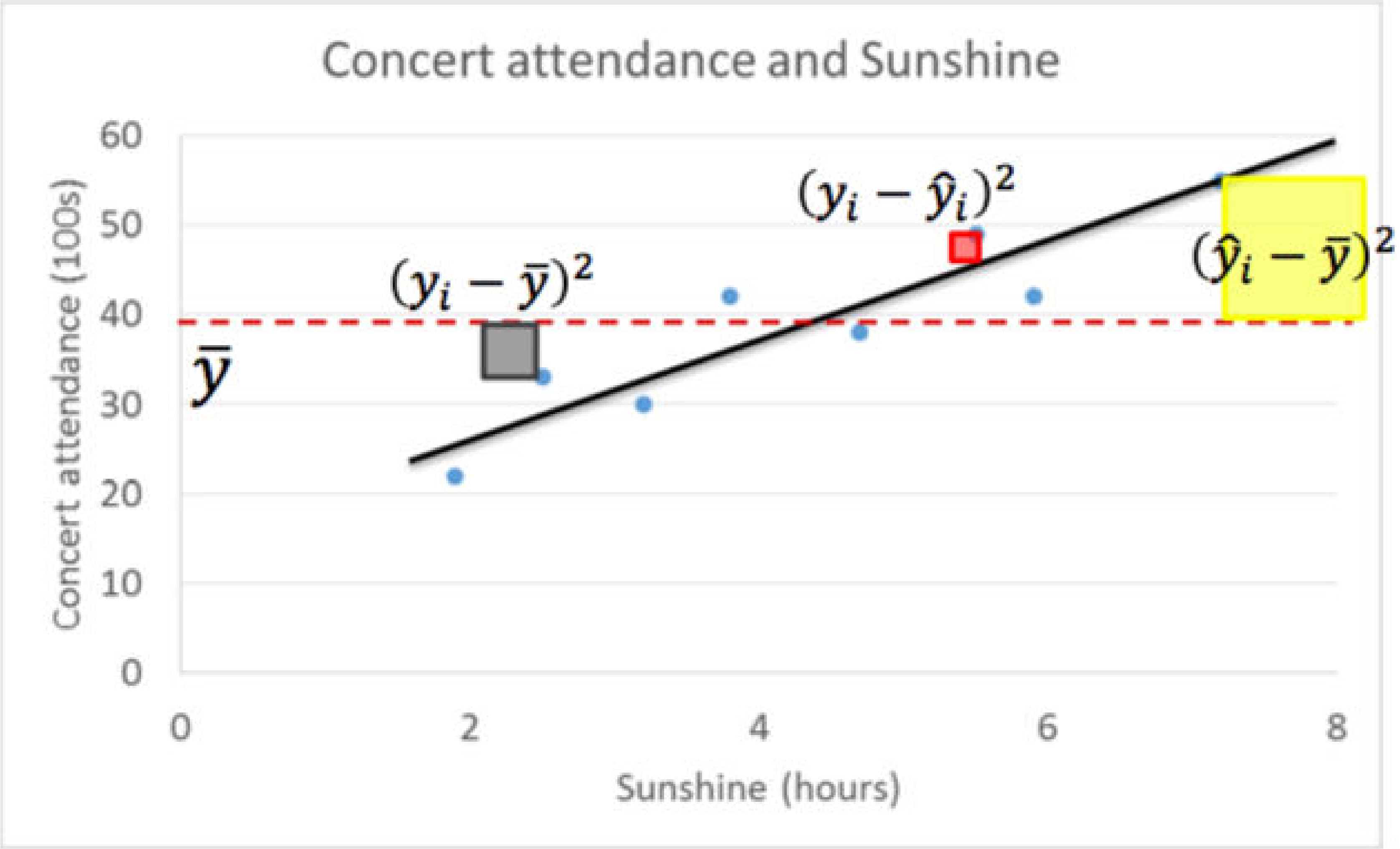
SST (Total Sum of Squares) =

SSR (Sum of Squares Regression) + SSE (sum of squared errors of prediction)

$$SST = SSR + SSE \Rightarrow \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = R^2$$

$$SST = \sum (y_i - \bar{y})^2 \quad SSR = \sum (\hat{y}_i - \bar{y})^2 \quad SSE = \sum (y_i - \hat{y}_i)^2$$

Coefficient of Determination(R Squared)



Adjusted R Squared

Adjusted R Squared

1. The **adjusted R-squared** is a modified version of R-squared that has been adjusted for the number of predictors in the model
2. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance
3. Adjusted R Squared will always be less than or equal to R Squared

Simple Linear Regression in R

Problem Statement

Building simple linear 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. Divide the “customer_churn” data into train & test sets with split ratio 65:35

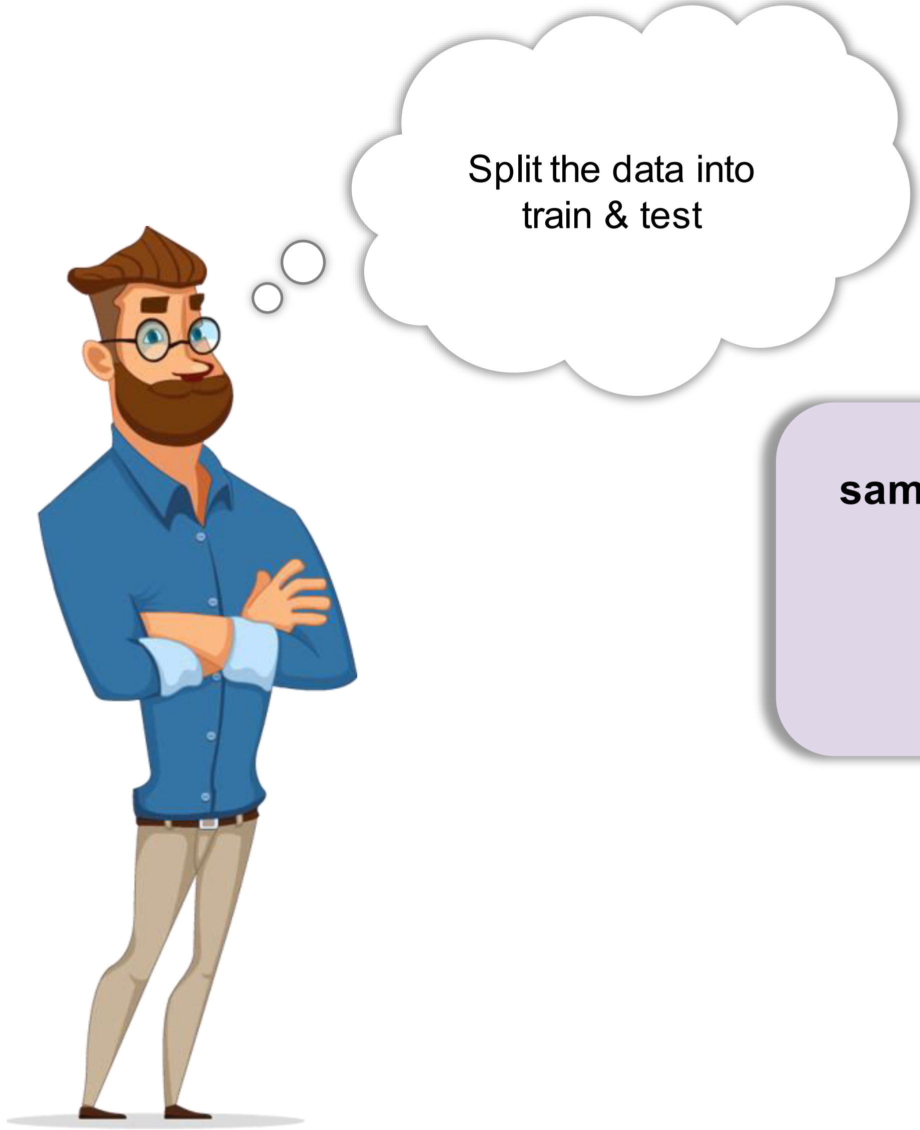
2. Build a Simple Linear Regression model where the dependent variable is “Monthly_Charges” and the independent variable is “tenure”

3. Build a Simple Linear Regression model where the dependent variable is “Monthly_Charges” and the independent variable is “InternetService”

Simple Linear Regression in R


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

Simple Linear Regression 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
```

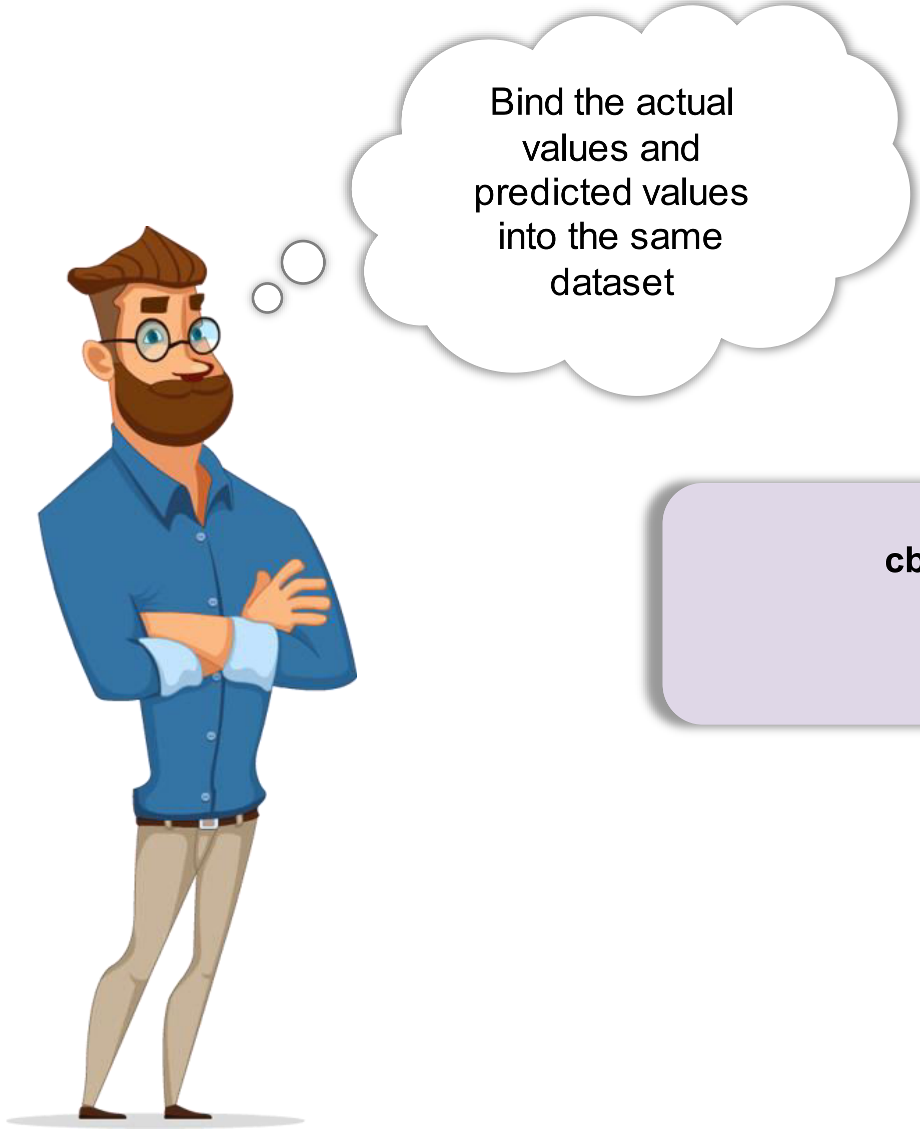
Simple Linear Regression in R

lm(MonthlyCharges~tenure, data=train)-> model1

↓

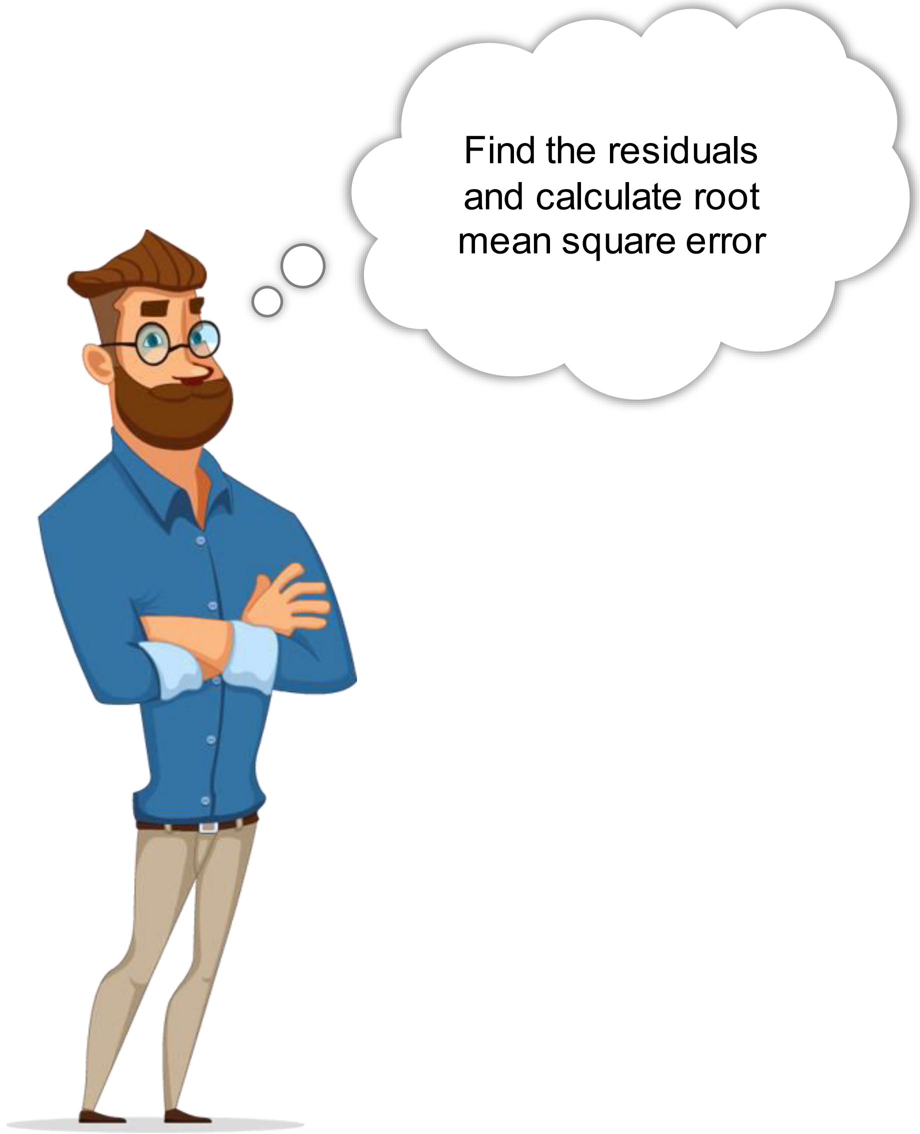
predict(model1, newdata=test)->predicted_values

Simple Linear Regression in R




```
cbind(Actual=test$MonthlyCharges,Predicted=predicted_values)->final_data

as.data.frame(final_data)->final_data
```

Simple Linear Regression in R




```
final_data$Actual - final_data$Predicted ->error

cbind(final_data,error)-> final_data
```

```
sqrt(mean((final_data$error)^2))->rmse1
```

Simple Linear Regression in R

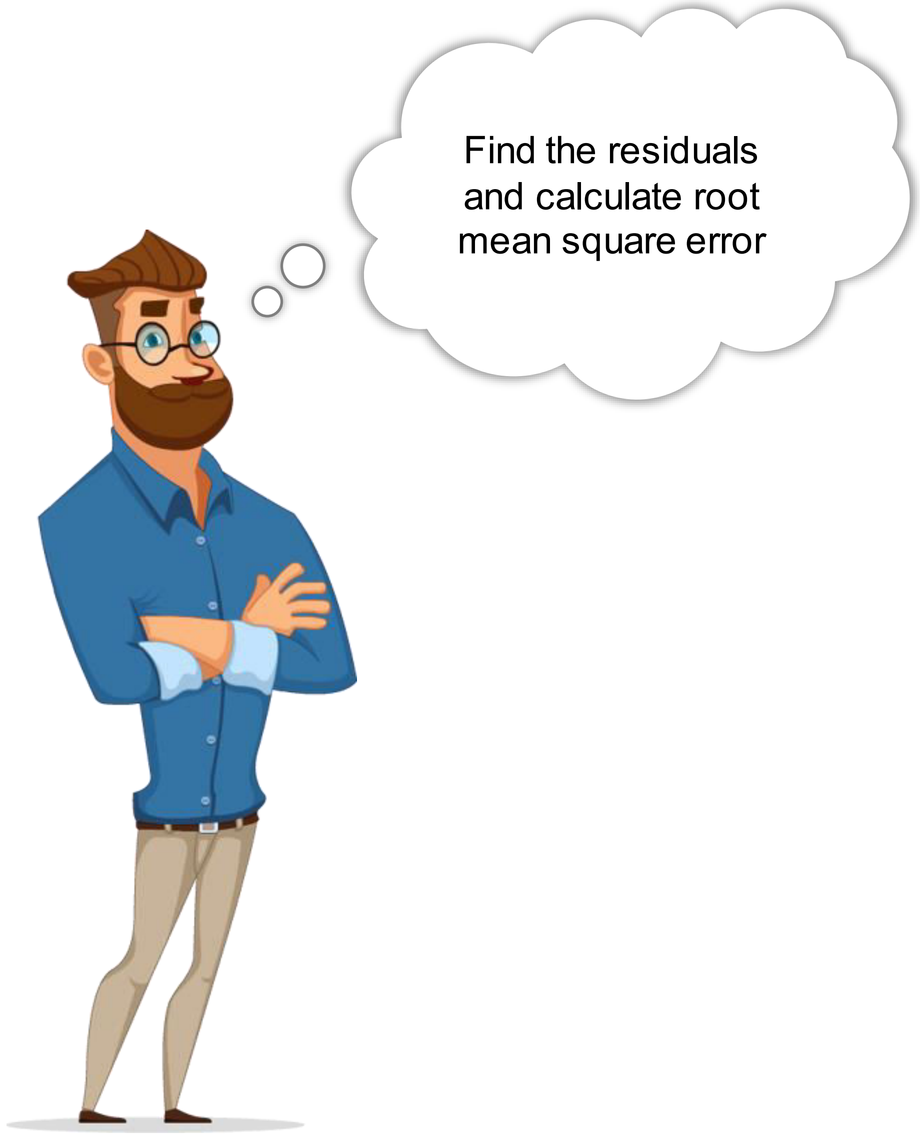
lm(MonthlyCharges~InternetService, data=train)-> model2

predict(model2, newdata=test)->result

Simple Linear Regression in R


```
cbind(Actual=test$MonthlyCharges,Predicted=result)->final_data2

as.data.frame(final_data2)->final_data2
```

Simple Linear Regression in R




```
final_data2$Actual - final_data2$Predicted ->error2

cbind(final_data2,error2)-> final_data2
```

```
sqrt(mean((final_data2$error2)^2))->rmse2
```

Multiple Linear Regression in R

Multiple Linear Regression in R

Multiple Linear Regression helps in modelling a relationship between ***two or more explanatory variables*** & ***a response variable***!

$$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3$$

Multiple Linear Regression in R

1. Divide the “customer_churn” data into train & test sets with split ratio 65:35

2. Build a Multiple Linear Regression model where the dependent variable is “tenure” and the independent variables are “Monthly_Charges”,”gender”,”InternetService” & “Contract”

3. Build a multiple Linear Regression Model where the dependent variable is “tenure” & the independent variables are “Partner”,”PhoneService”,”TotalCharges” & “PaymentMethod”

Multiple Linear Regression in R

lm(tenure~MonthlyCharges**+**gender**+**InternetService**+**Contract, data=train)-> mod1

predict(mod1,test)->result1

Multiple Linear Regression in R


```
cbind(Actual=test$tenure,Predicted=result1)->final_data1

as.data.frame(final_data1)->final_data1
```

Multiple Linear Regression in R

Find the residuals and calculate root mean square error

```
final_data$Actual - final_data$Predicted ->error1

cbind(final_data1,error1)-> final_data1
```

```
sqrt(mean((final_data1$error1)^2))->rmse1
```

Multiple Linear Regression in R

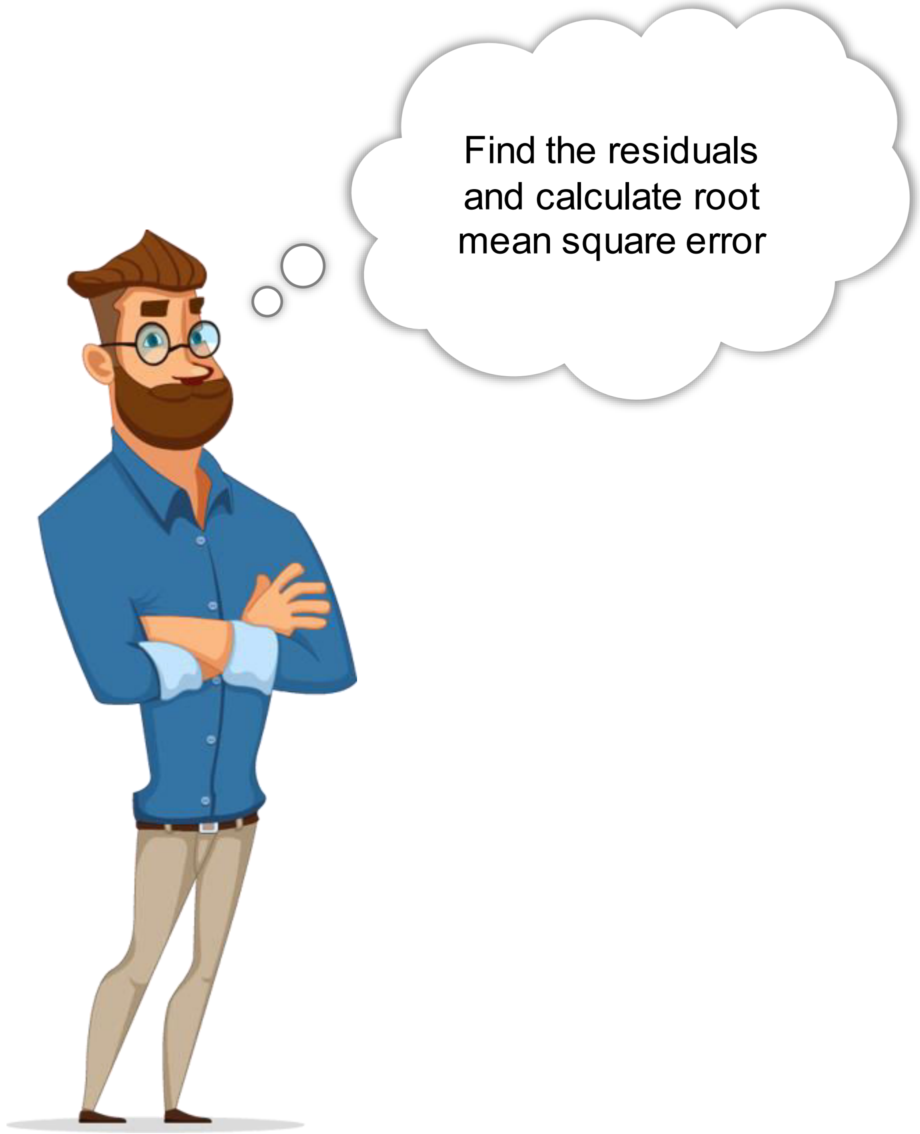
lm(tenure~Partner**+**PhoneService**+**TotalCharges**+** PaymentMethod,data=train)-> mod2

predict(mod2,test)-> result2

Multiple Linear Regression in R


```
cbind(Actual=test$tenure,Predicted=result2)->final_data2

as.data.frame(final_data2)->final_data2
```

Multiple Linear Regression in R



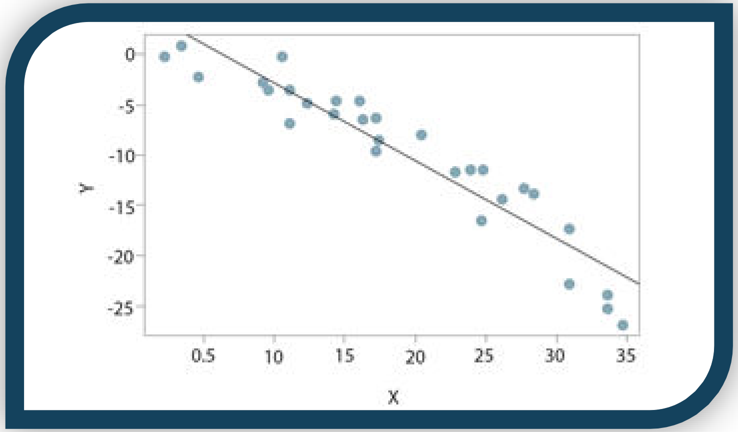

```
final_data2$Actual - final_data2$Predicted ->error2

cbind(final_data2,error2)-> final_data2
```

```
sqrt(mean((final_data2$error2)^2))->rmse2
```

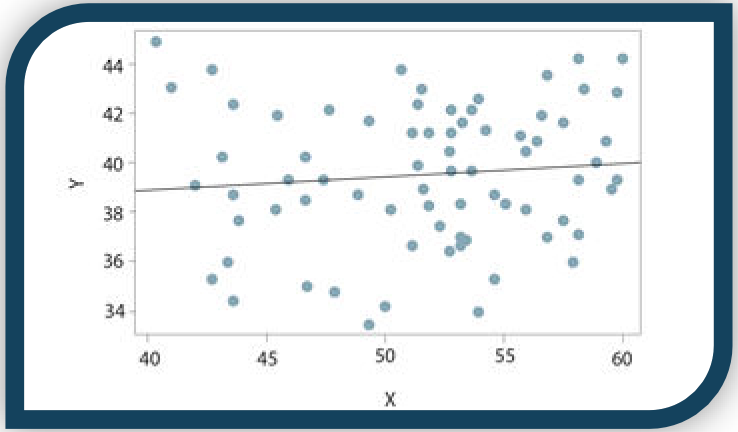
Assumptions in Linear Regression

Assumptions in Regression - Linearity

There should be a ***linear*** & ***additive*** relationship between the dependent & independent variables!



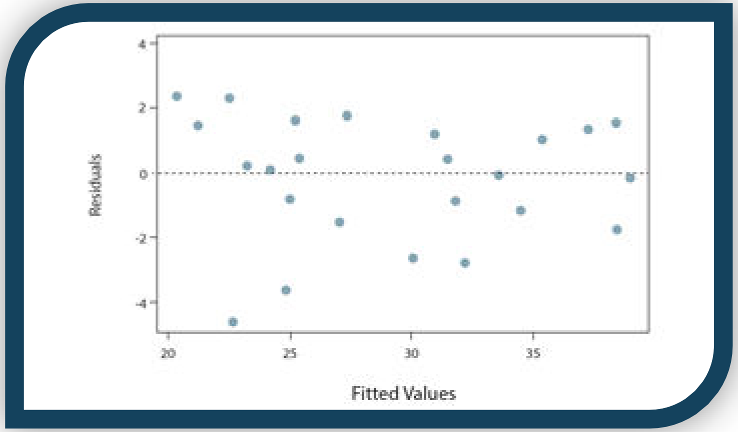
Satisfies the assumption



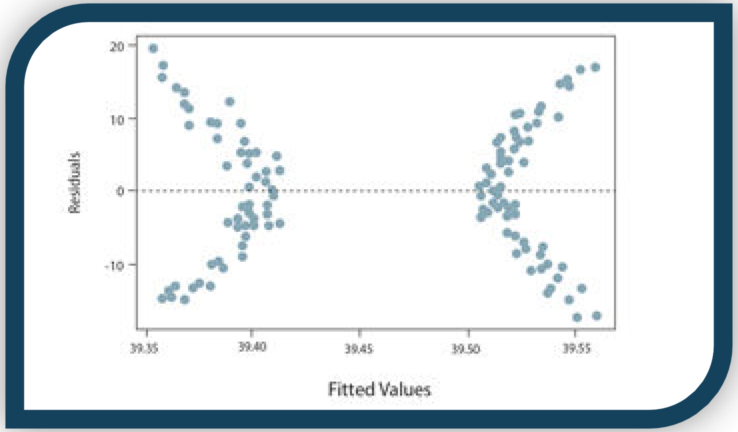
Doesn't Satisfy the assumption

Assumptions in Regression – Equal Error Variance

The residuals must have ***constant variance***!



If there is no pattern, data is random and hence, satifies the condition

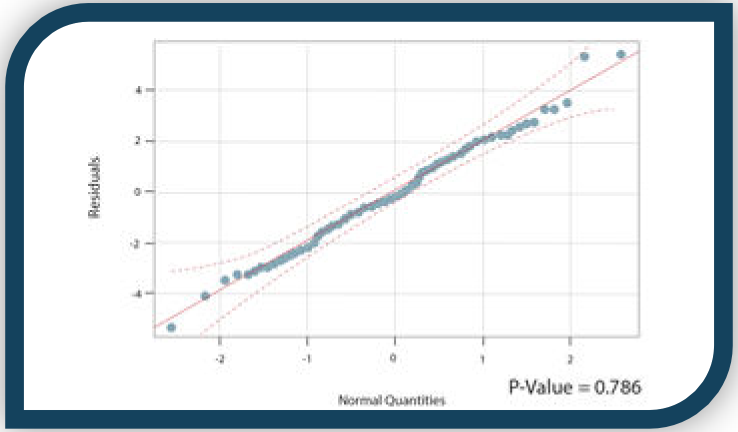


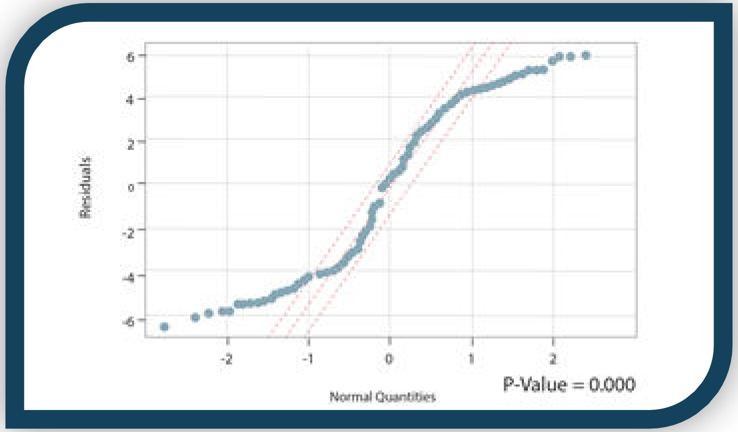
If there is a pattern, the data is not random and hence, doesn’t satify the condition

Assumptions in Regression – Normality of Errors

The residuals must have ***normally distributed***!

Build a normal probability plot. If the residuals are closer to the fit line, the more normal they are





Checking the Assumptions in R

Checking Assumptions in R

1. Make a scatter-plot between “tenure” & “TotalCharges” and find out if there’s a linear relationship between them

2. Make a “residual vs fit” graph to check for equal error variance

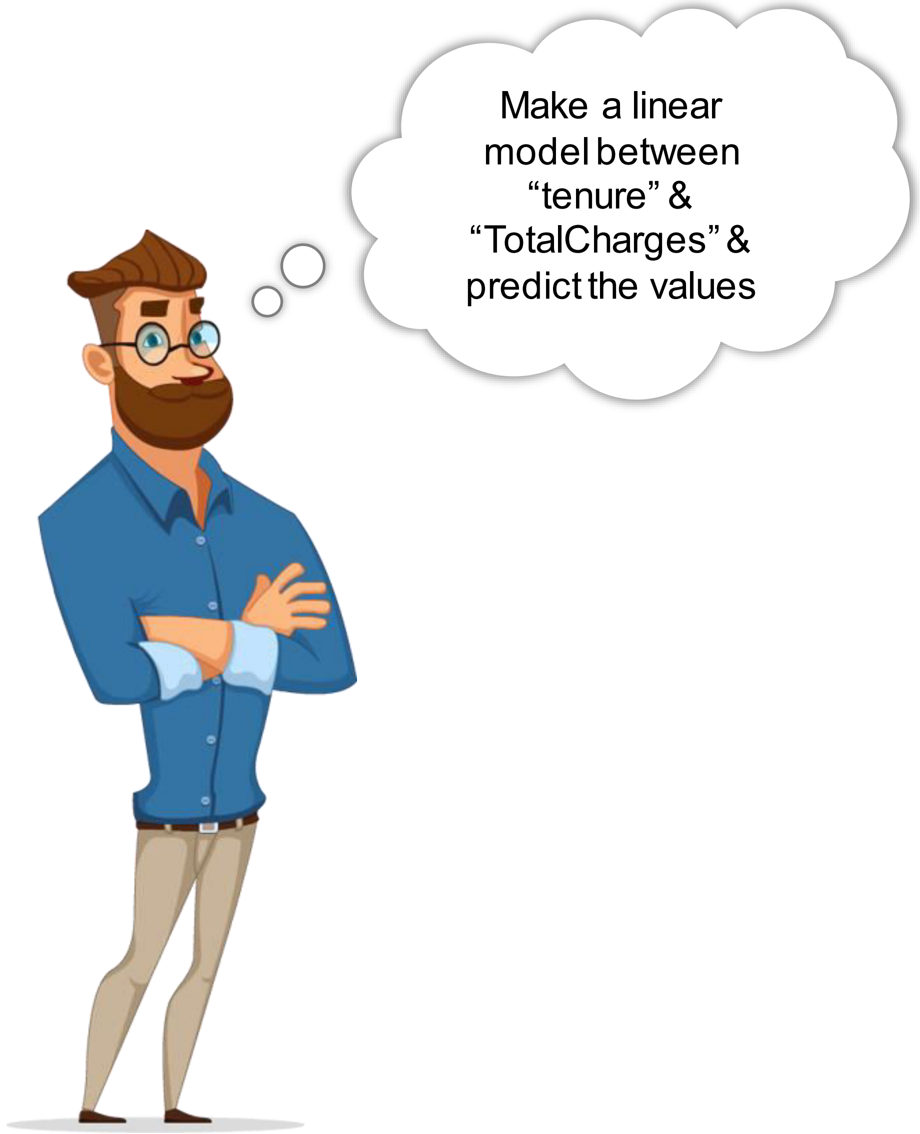
3. Build a “normal probability plot” to check for normality of errors

Checking for Linearity

ggplot(data= customer_churn, **aes**(x=tenure, y=TotalCharges))
+ geom_point()

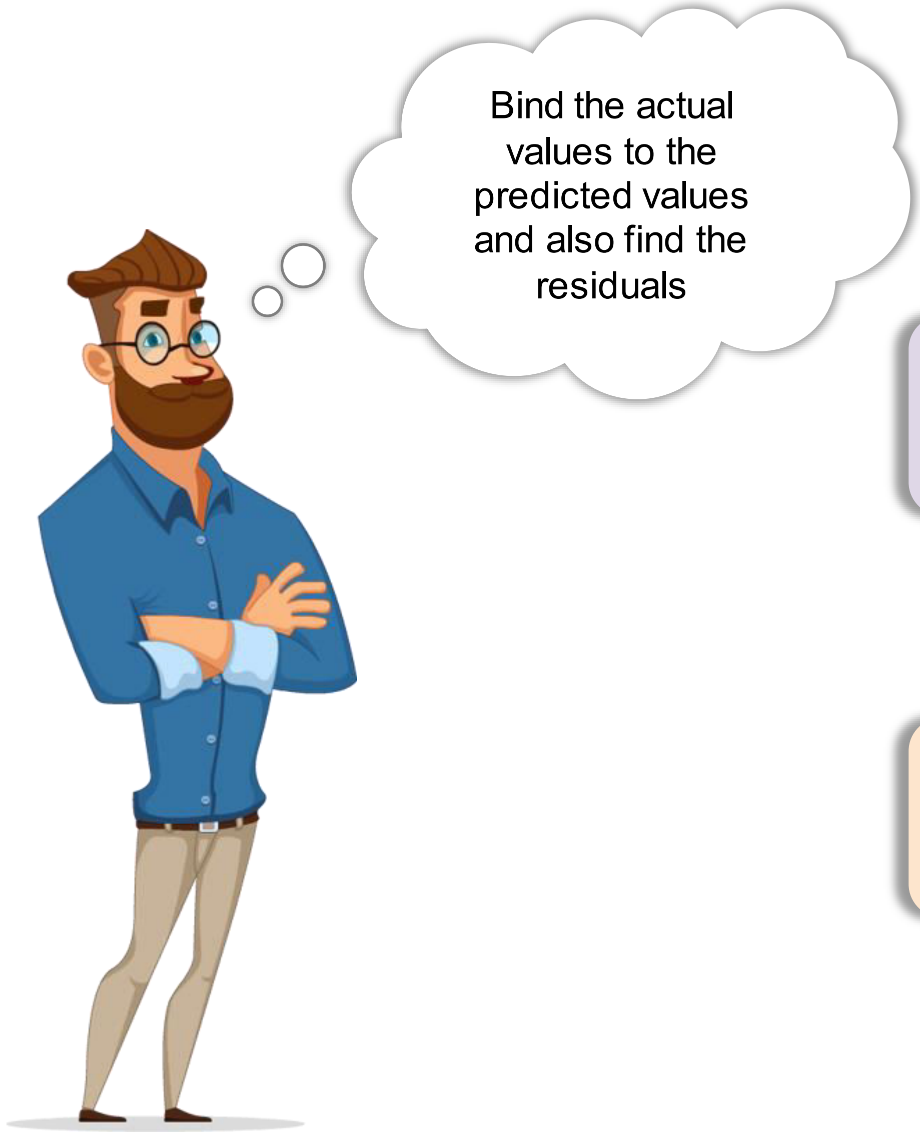
ggplot(data= customer_churn, **aes**(x=tenure, y=TotalCharges))
+ geom_point()**+geom_smooth**(method = "lm")

Checking for Equal Error Variance




```
lm(TotalCharges~tenure, data = customer_churn)->mod1

predict(mod1,data=customer_churn)-> result1
```

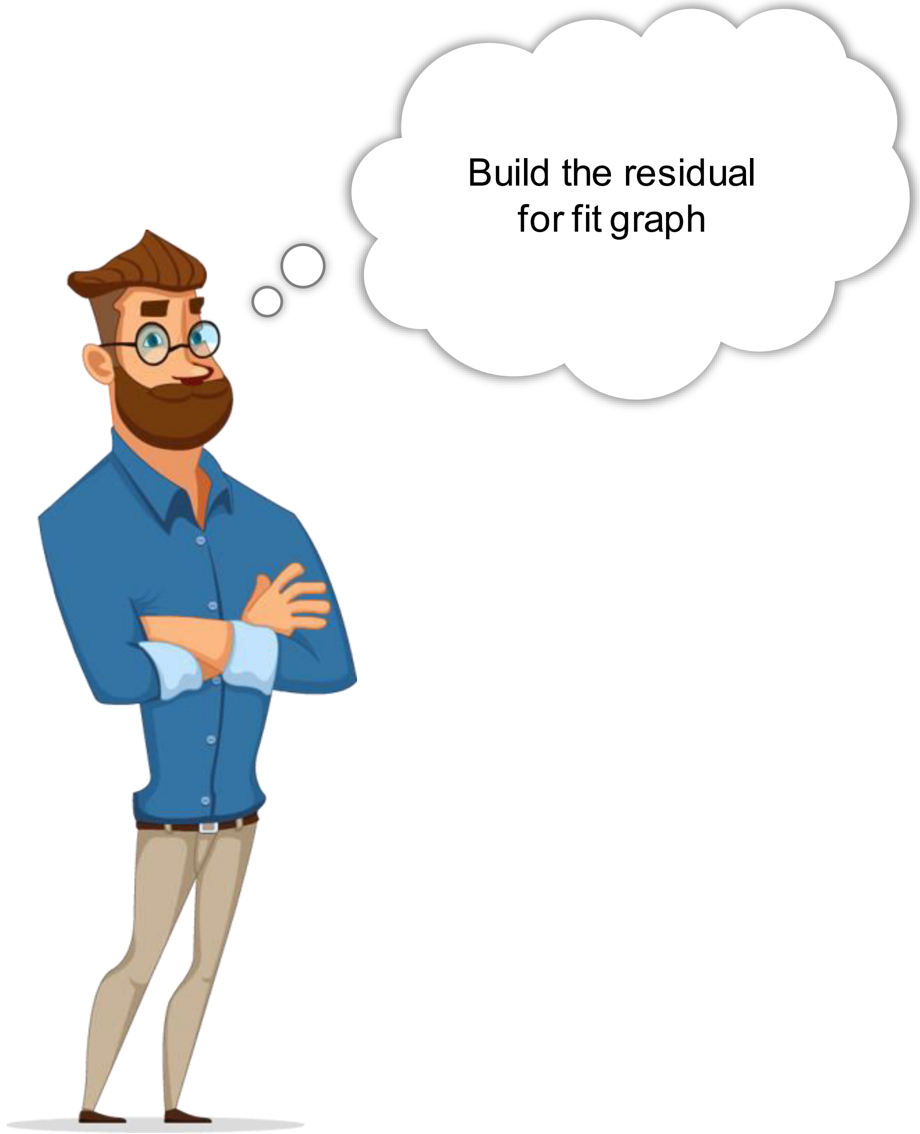
Checking for Equal Error Variance




```
cbind(Actual=customer_churn$TotalCharges, Predicted=result1)-> final_data1

as.data.frame(final_data1)->final_data1
```

```
final_data1$Actual -final_data1$Predicted  -> error1

cbind(final_data1,error1)-> final_data1
```

Checking for Equal Error Variance




```
ggplot(data= final_data1, aes(x=Predicted, y=error1)) + geom_point()
```

Checking for Normality of Errors


```
qqnorm(final_data1$error1)
```

Linear Regression on ‘Boston’ dataset

Problem Statement

Building linear regression model on top of the 'Boston' dataset

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black
0.00632	18.0	2.31	0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90
0.02731	0.0	7.07	0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90
0.02729	0.0	7.07	0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83
0.03237	0.0	2.18	0	0.4580	6.998	45.8	6.0622	3	222	18.7	394.63
0.06905	0.0	2.18	0	0.4580	7.147	54.2	6.0622	3	222	18.7	396.90
0.02985	0.0	2.18	0	0.4580	6.430	58.7	6.0622	3	222	18.7	394.12
0.08829	12.5	7.87	0	0.5240	6.012	66.6	5.5605	5	311	15.2	395.60
0.14455	12.5	7.87	0	0.5240	6.172	96.1	5.9505	5	311	15.2	396.90
0.21124	12.5	7.87	0	0.5240	5.631	100.0	6.0821	5	311	15.2	386.63

Task 1

Task 1

str(Boston)

↓

summary(Boston)

Task 1

Divide the data into
train & test tests

```
library(caTools)
sample.split(Boston$medv,SplitRatio = 0.65) -> split_tag

subset(Boston,split_tag==T) -> train
subset(Boston,split_tag==F) -> test
```

Task 1

Build the first model and predict the values

```
lm(medv~crim,data = train) -> lin_mod1
```

```
predict(lin_mod1,test) -> lin_result1
```

Task 1

Bind the actual and predicted values and find the RMSE value

```
cbind(Actual=test$medv,Predicted=lin_result1) -> final_data1
as.data.frame(final_data1) -> final_data1
```

```
(final_data1$Actual - final_data1$Predicted)  -> error1
cbind(final_data1,error1) -> final_data1

sqrt(mean((final_data1$error1)^2))
```

Task 2

Task 2


```
lm(medv~.,data = train) -> lin_mod2
plot(lin_mod2)
```

Task 2

Predict the values and bind actual and predicted results

```
predict(lin_mod2,test) -> lin_result2
```

```
cbind(Actual=test$medv,Predicted=lin_result2) -> final_data2
as.data.frame(final_data2) -> final_data2
```

Task 2


```
(final_data1$Actual - final_data1$Predicted)  -> error1
cbind(final_data1,error1) -> final_data1
```

```
sqrt(mean((final_data1$error1)^2))
```

Quiz

Quiz

A correlation between age and health of a person was found to be -1.09. On the basis of this you would tell the doctors that:

a) A. The age is good predictor of health

b) B. The age is poor predictor of health

c) C. None of these

Quiz

Which of the following metrics can be used for evaluating regression models?

1) R Squared 2) Adjusted R Squared 3) F Statistics 4) RMSE / MSE / MAE

a) 2 and 4

b) 1 and 2

c) 2, 3 and 4

d) All of the above

Quiz

A residual is defined as:

a. Observed value - Predicted value
b. Error sum of square
c. Regression sum of squares
d. Type I Error

Thank You

IntelliPaat

India : +91-7847955955

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

sales@intellipaat.com

24X7 Chat with our Course Advisor