MACHINE LEARNING TUTORIAL

What's in it for you?

- Life without Machine Learning
- Life with Machine Learning
- What is Machine Learning
- Types of Machine Learning
- The right Machine Learning solutions
- Machine Learning Algorithms
- Use case - Predicting the price of a house using Linear Regression

simplilearn

Life without Machine Learning

No facial recognition on Facebook and Instagram

World's best search engine wont be the best

No voice recognition

simplilearn

Life with Machine Learning

amazon

Shopping made easy

Connecting with old friends

NETFLIX

Get the best movie suggestions

simplilearn

Life with Machine Learning

VR glasses provide a highly immersive gaming experience

Gesture control gaming

FIFA 18

Opponent gets stronger as you progress in the game

simplilearn

Life with Machine Learning

amazon

Product suggestions

Price of a product varies based on the demand

Customer segmentation

simplilearn

Life with Machine Learning

UBER

Suggests drop, pickup locations automatically

Uber share finds customers travelling in similar routes

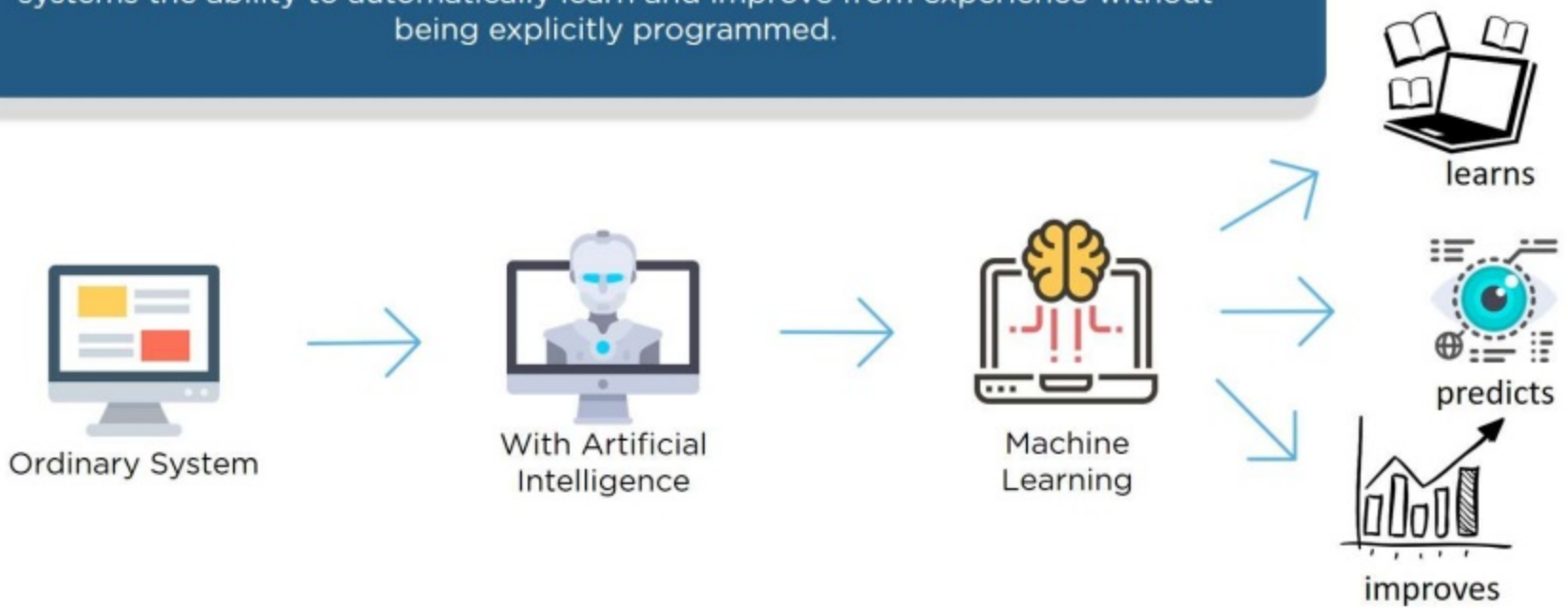
Allocates the bookings based on time, distance, price and ratings

simplilearn


What is Machine Learning?
simplilearn

What is Machine Learning?

Machine Learning is an application of Artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.



simplilearn

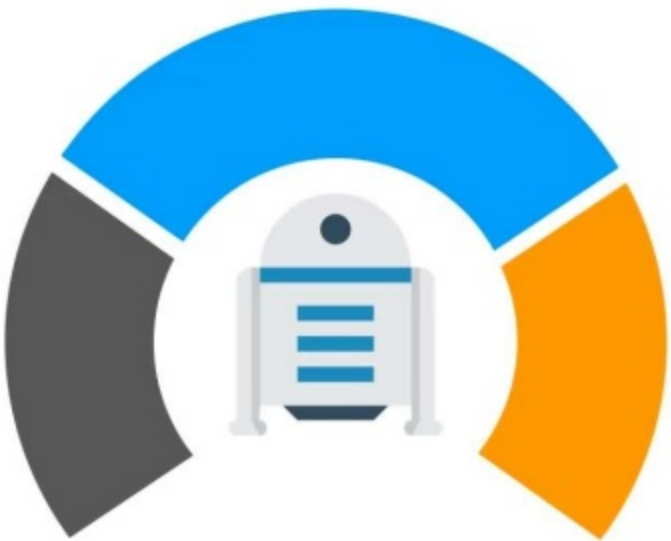
Machine Learning process


Types of Machine Learning
simplilearn

Types of Machine Learning

Unsupervised Learning
Non-labeled training data

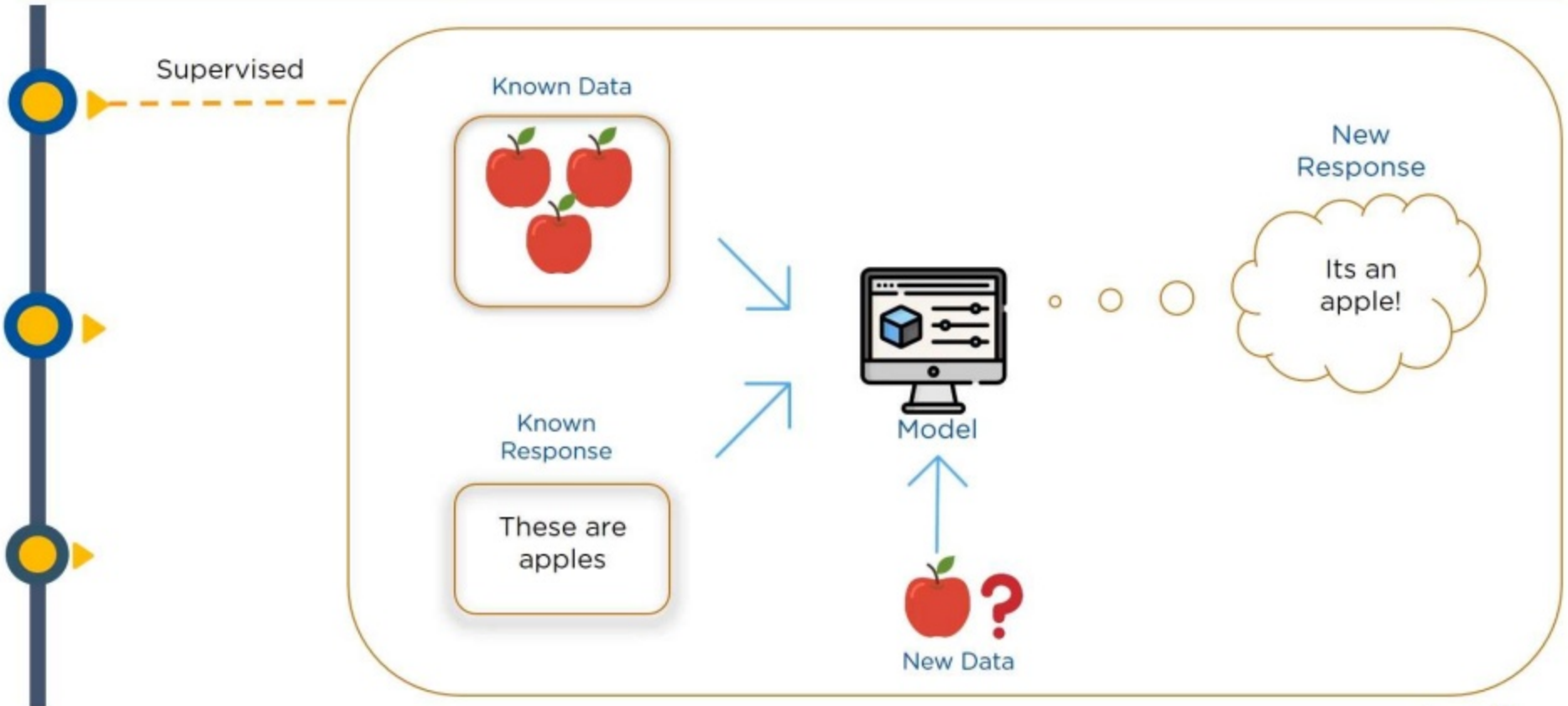
Supervised Learning
The machine learns from the training data that is labeled



Reinforcement Learning
The machine learns on its own

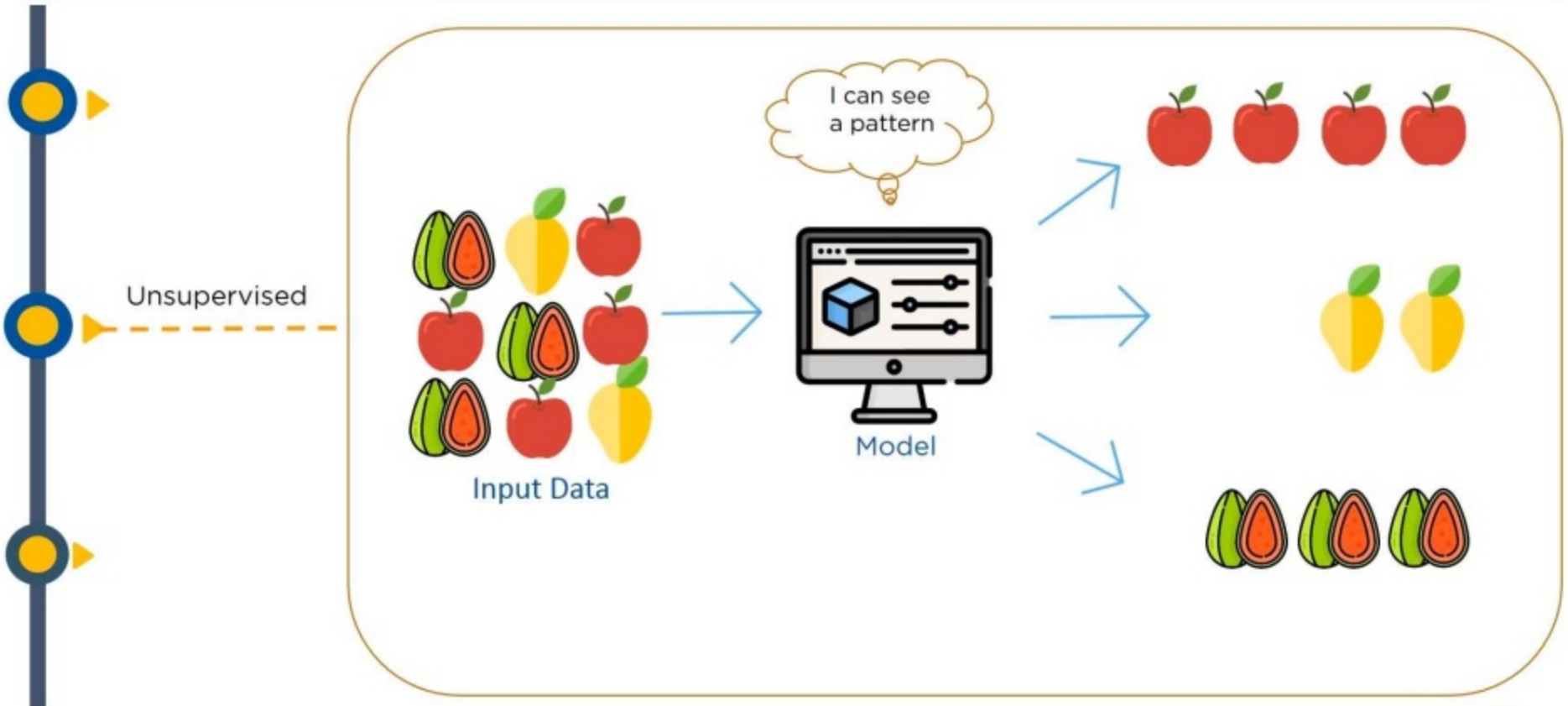
simplilearn

Types of Machine Learning



simplilearn

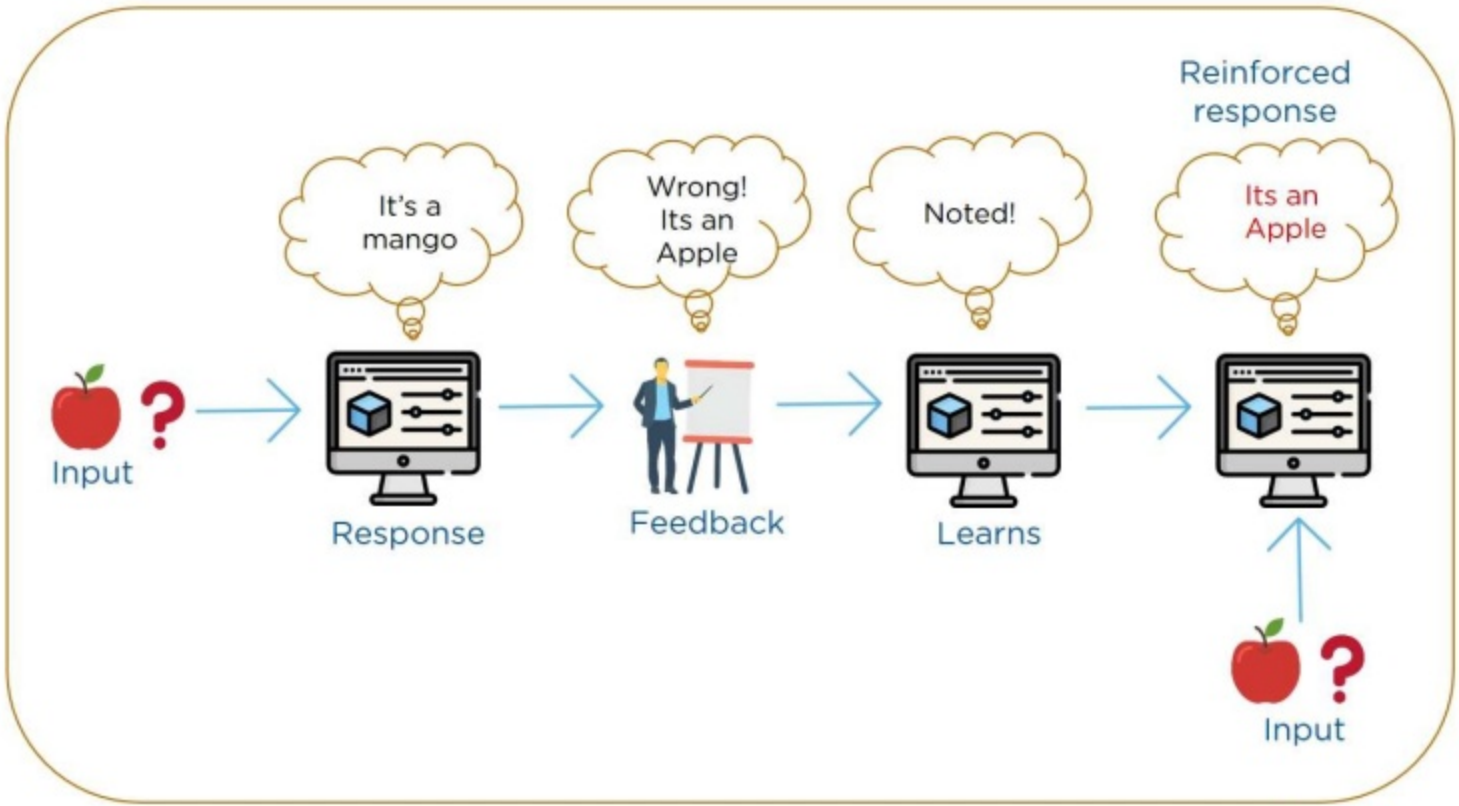
Types of Machine Learning



simplilearn

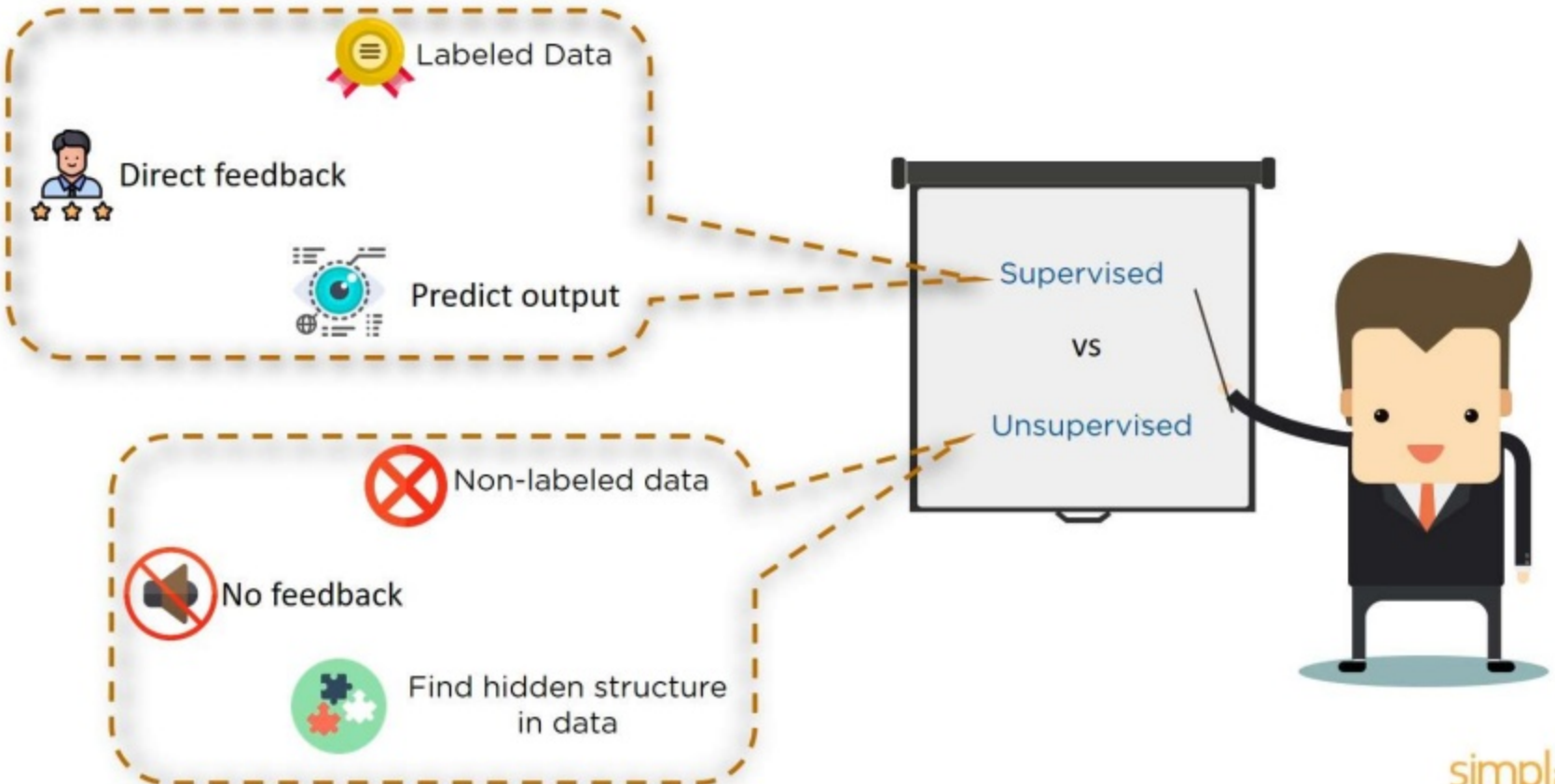
Types of Machine Learning

Reinforcement



simplilearn

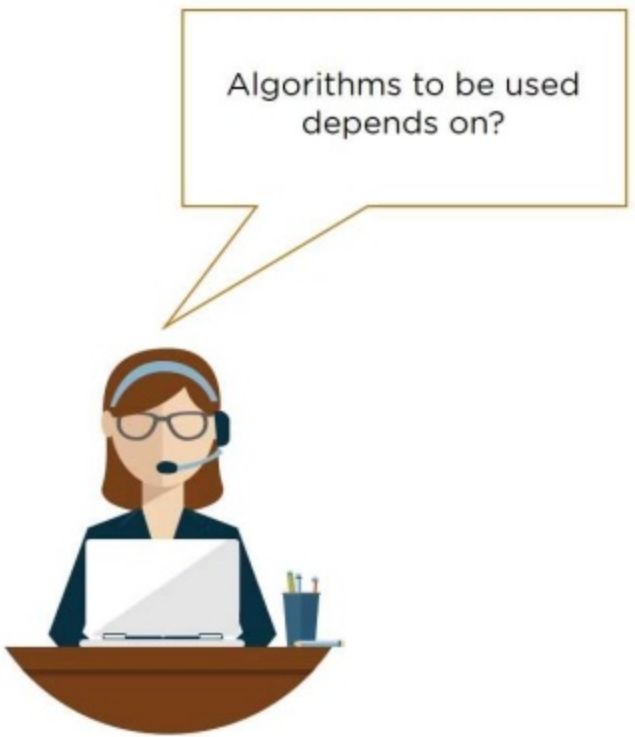
Supervised vs Unsupervised

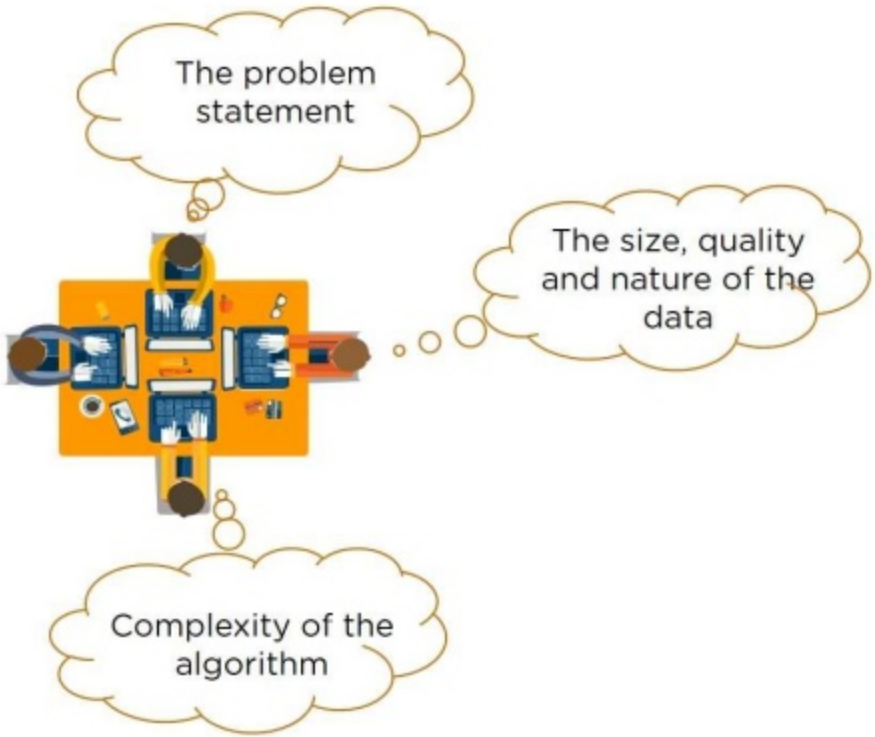


simplilearn


The right Machine Learning solution?
simplilearn

The Right Machine Learning Solution?





The right Machine Learning solution?

Classification

Used when the output is categorical like 'YES' or 'NO'

Algorithms used

- Decision Tree
- Naïve Bayes
- Random Forest
- Logistic regression
- KNN

Regression

Used when a value needs to be predicted like the 'stock prices'

Algorithms used

- Linear Regression

Clustering

Used when the data needs to be organized to find patterns in the case of 'product recommendation'

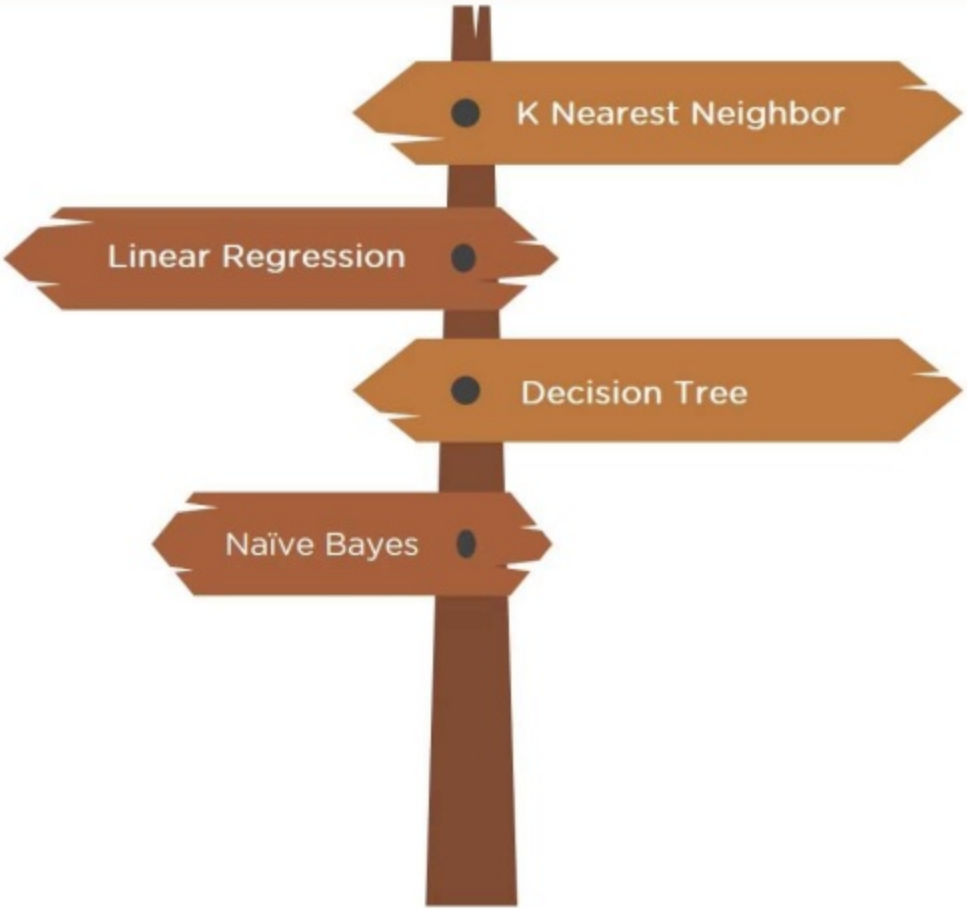
Algorithms used

- K Means

simplilearn

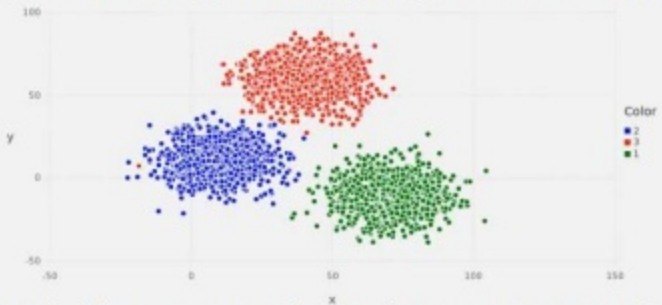

Machine Learning Algorithms
simplilearn

Machine learning Algorithms

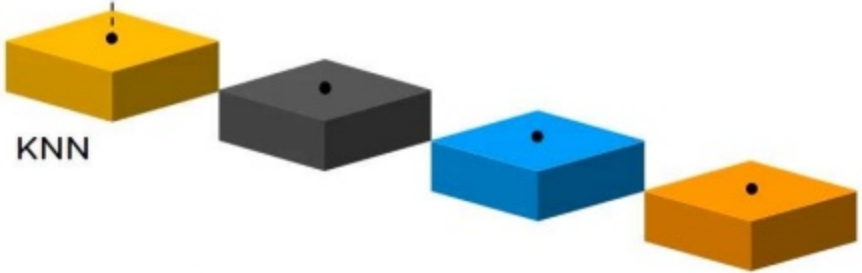


K Nearest Neighbors

K Nearest Neighbors algorithm works in a way that a new data point is assigned to a neighboring group it is most similar to.

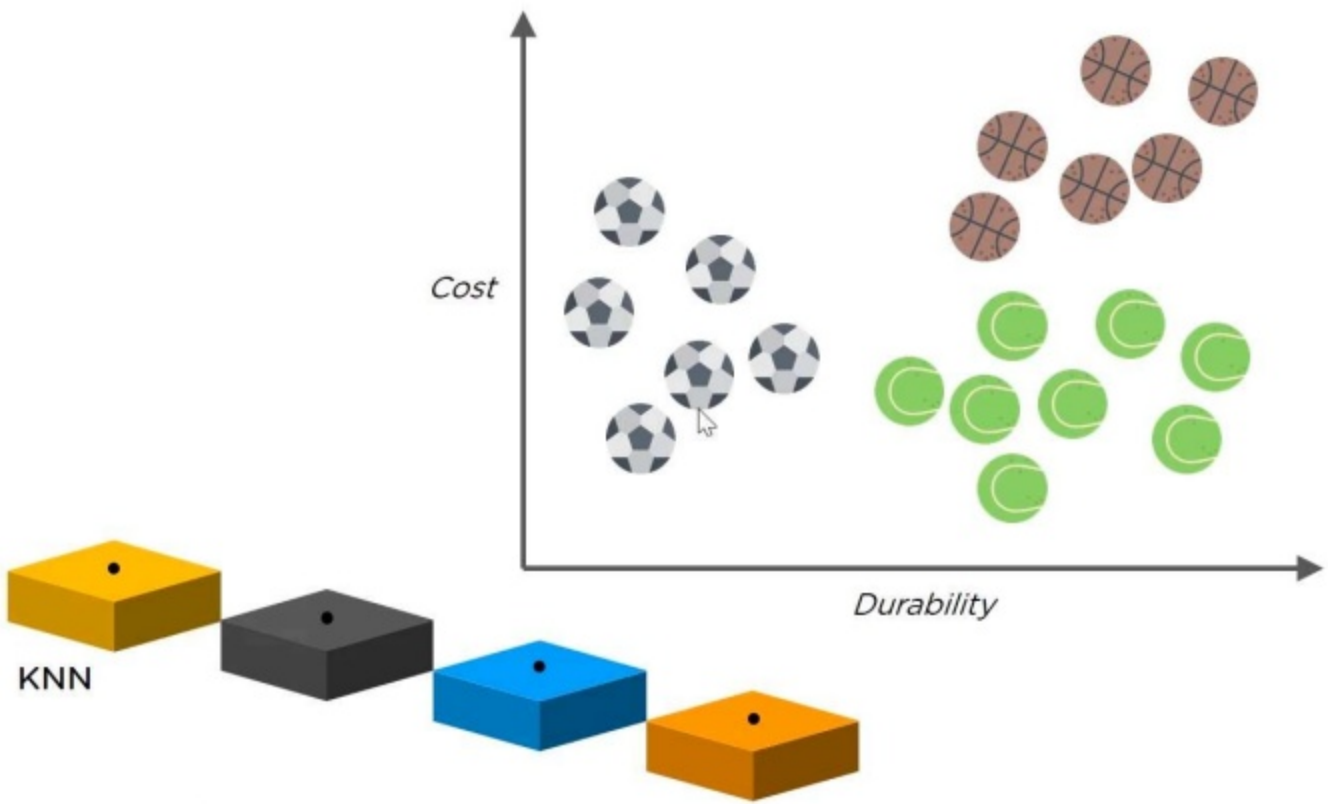


In K Nearest Neighbors, K can be an integer greater than 1. So, for every new data point we want to classify, we compute to which neighboring group it is closest to.



simplilearn

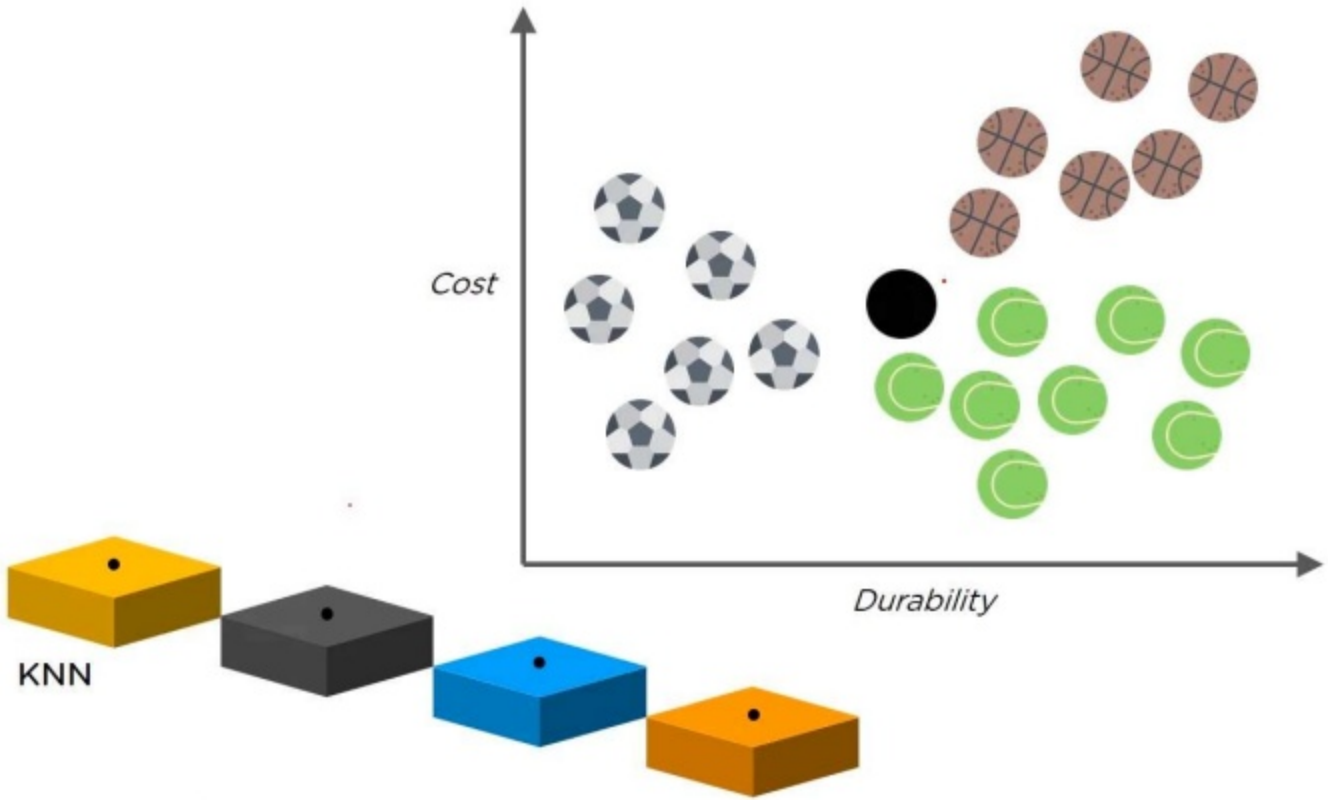
K Nearest Neighbors - Example



Consider there are 3 clusters:

- Football
- Basketball
- Tennis Ball

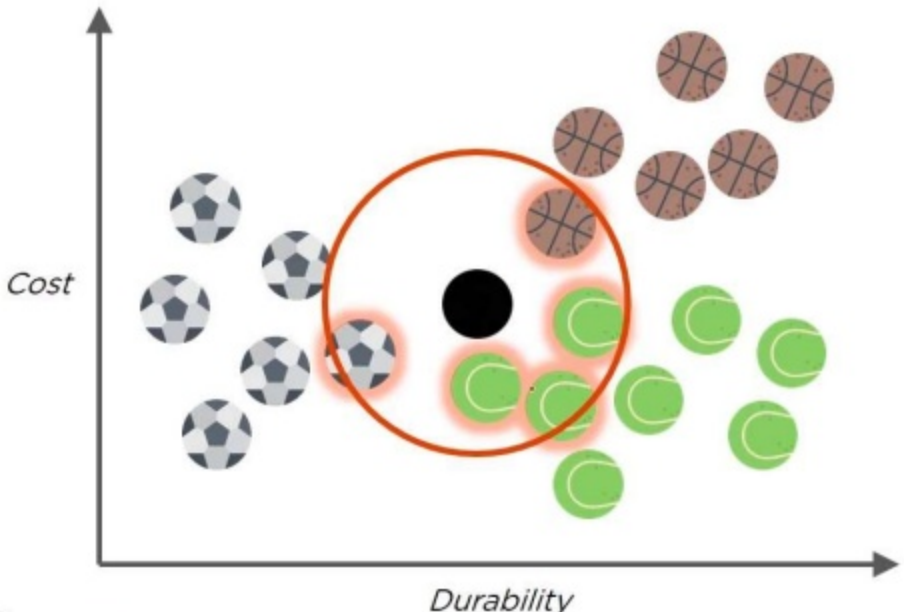
K Nearest Neighbors - Example



Now, we have a new data point (black ball)

We will now try to classify this using KNN

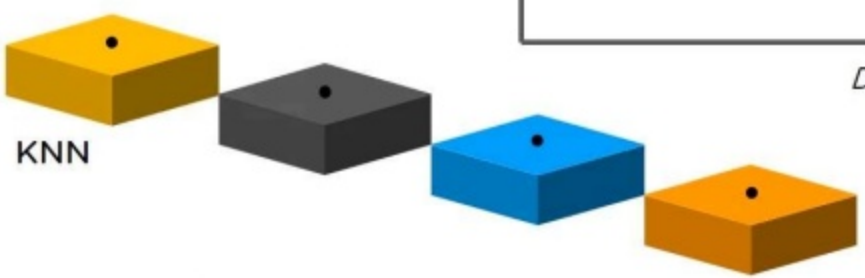
K Nearest Neighbors - Example



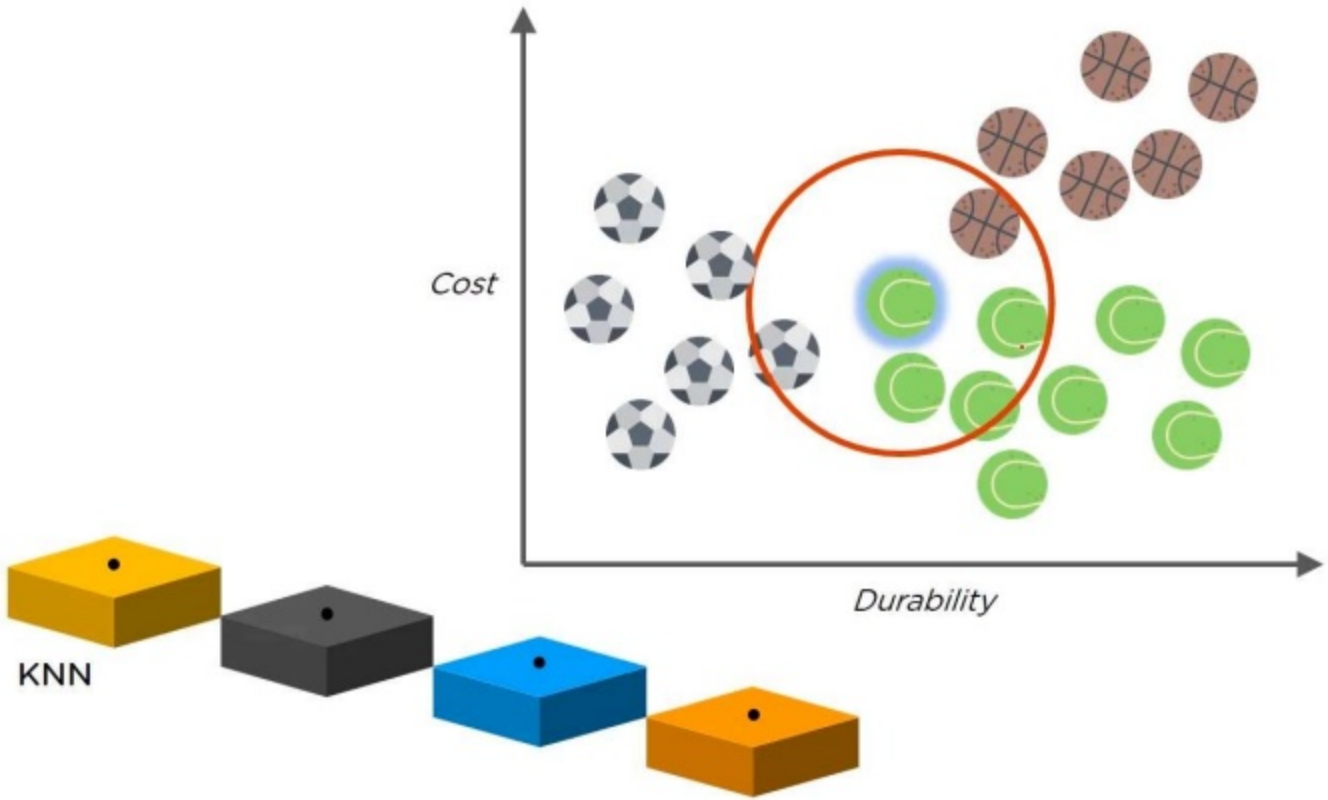
Let K = 5
So these glowing ones are the closest 5 balls to our new data point (black ball)

Here, there are 3 tennis balls and one each of basketball and football

Thus we will classify the black ball as a tennis ball



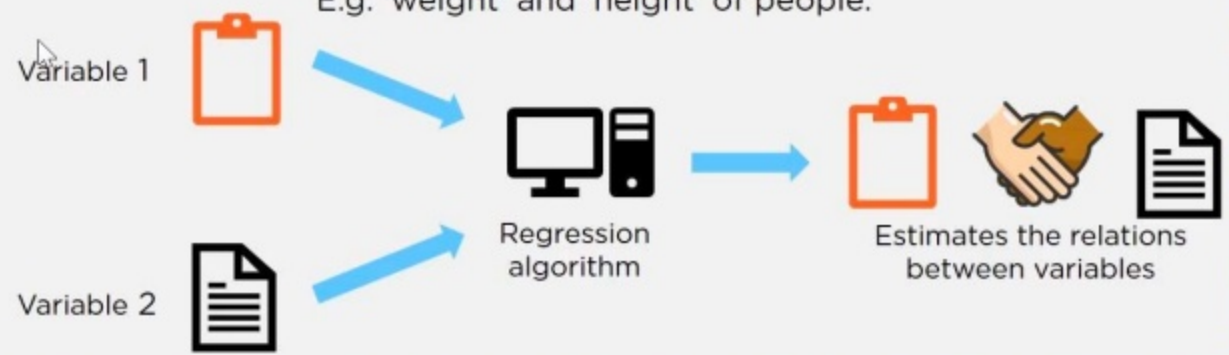
K Nearest Neighbors - Example

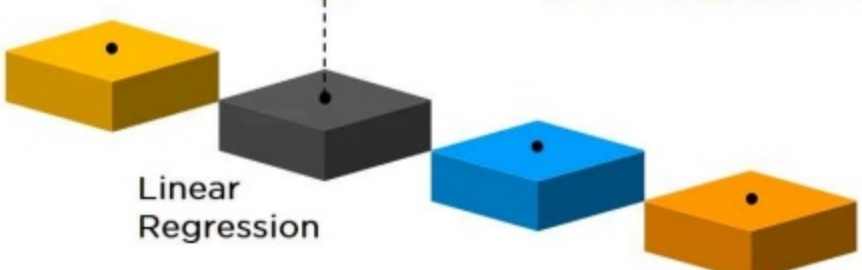


Thus our new data point is a tennis ball

Linear Regression

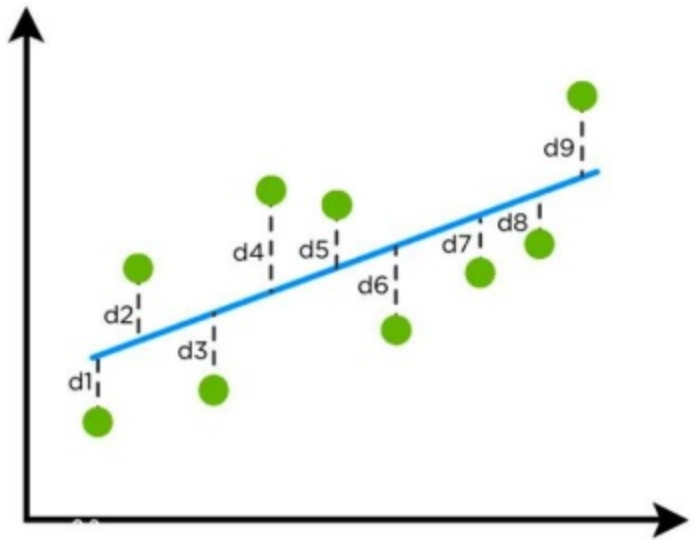
Linear Regression is a process used for estimating the relationships among variables. Here, one of the variables is dependent on one or more independent variables.
E.g. 'weight' and 'height' of people.





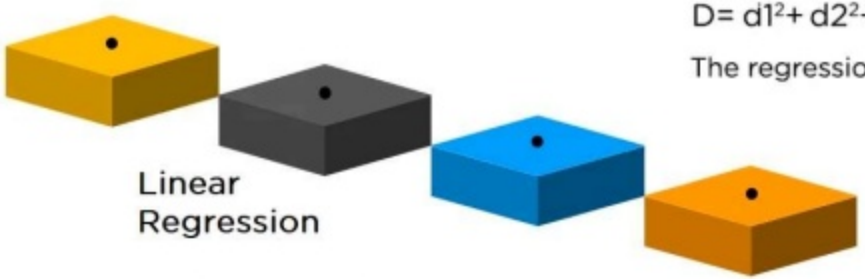
simplilearn

Linear Regression



$D = d1^2 + d2^2 + d3^2 + d4^2 + d5^2 + d6^2 + d7^2 + d8^2 + d9^2$

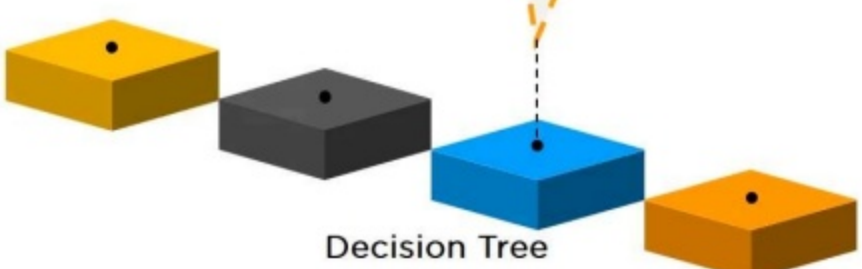
The regression line (blue) has the least value of D



simplilearn

Decision Tree

A decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision

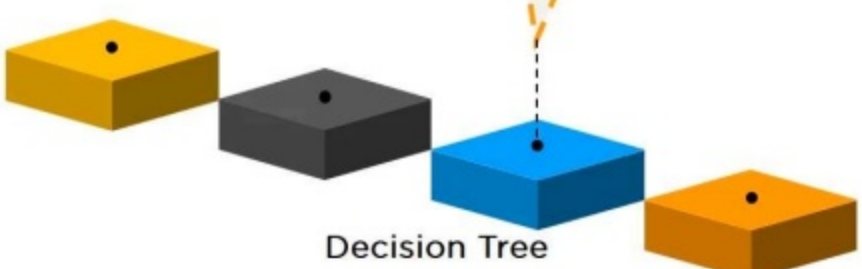


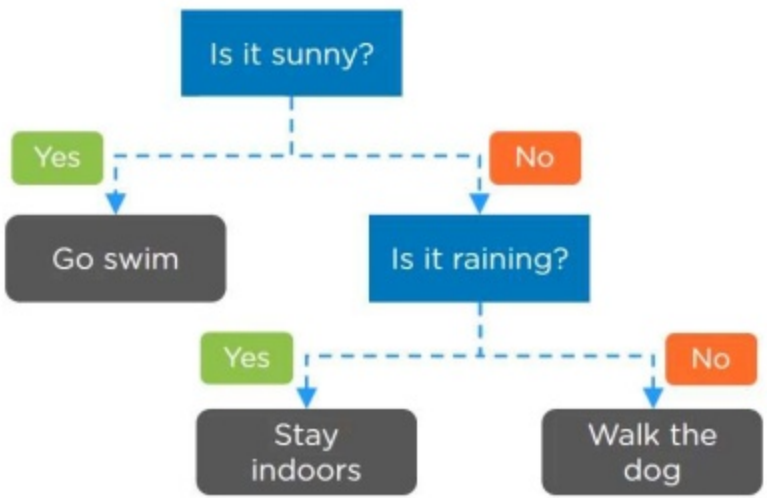
Decision Tree

simplilearn

Decision Tree

A decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision

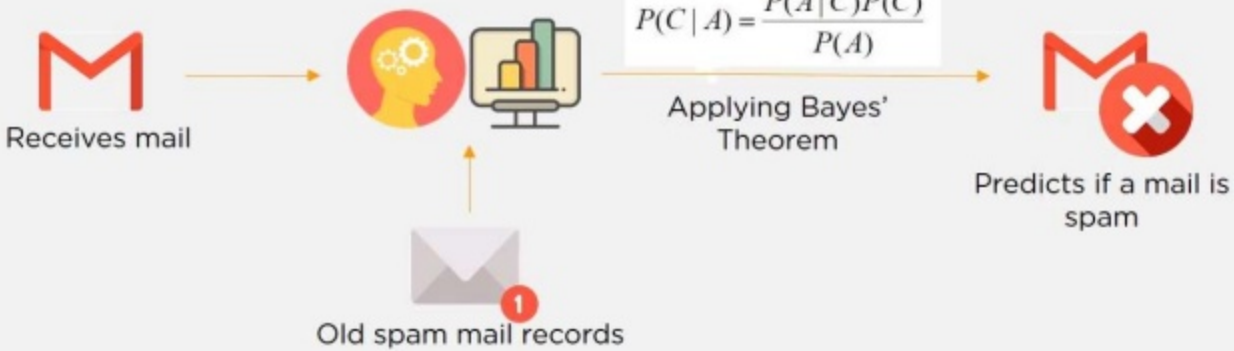


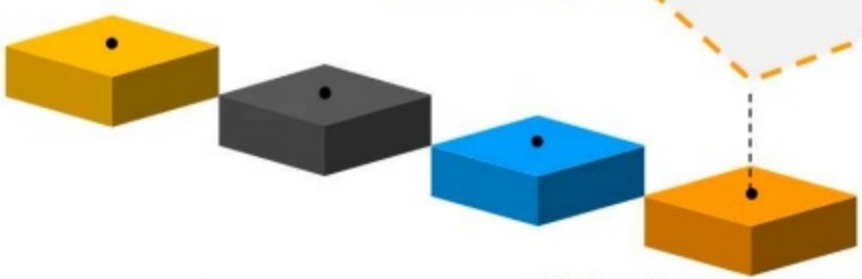


simplilearn

Naive Bayes

The Naive Bayes Classifier technique is based on conditional probability and is particularly suited when the complexity of the inputs is high



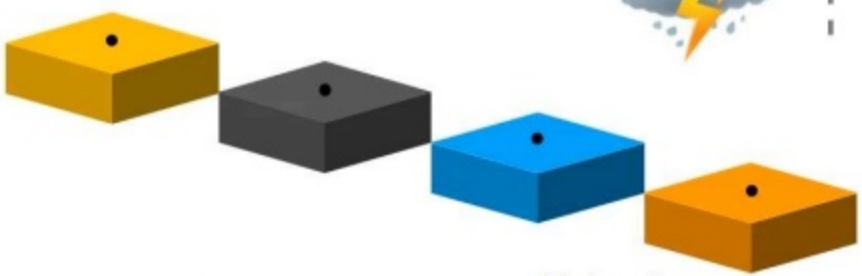


Naive Bayes

simplilearn

Naive Bayes

Naive Bayes can be used to determine on which days to play cricket



Naive Bayes

simplilearn


Use case
simplilearn

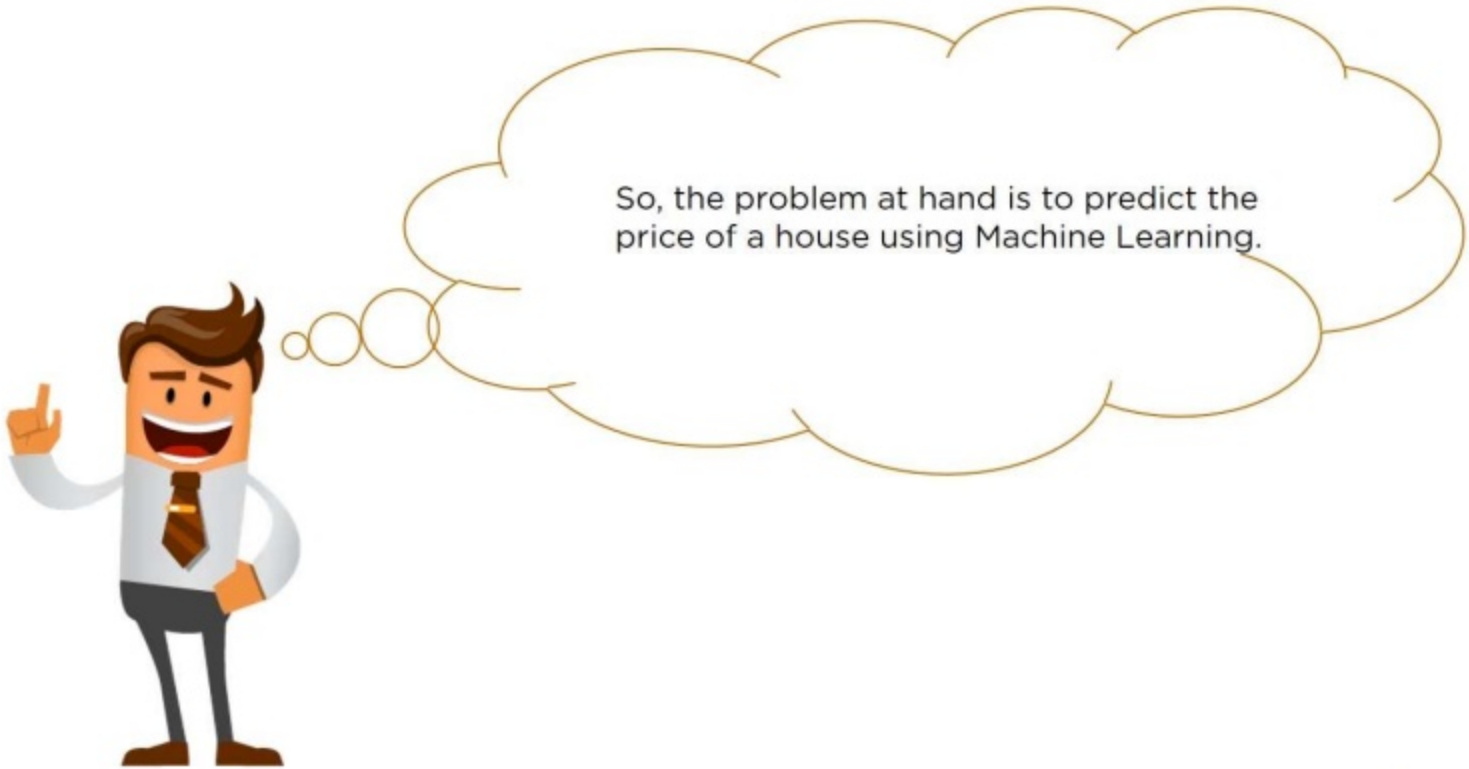
Use case

Use case

Use case

Use case

Use case - Problem statement



Use case - Problem statement

PROBLEM STATEMENT

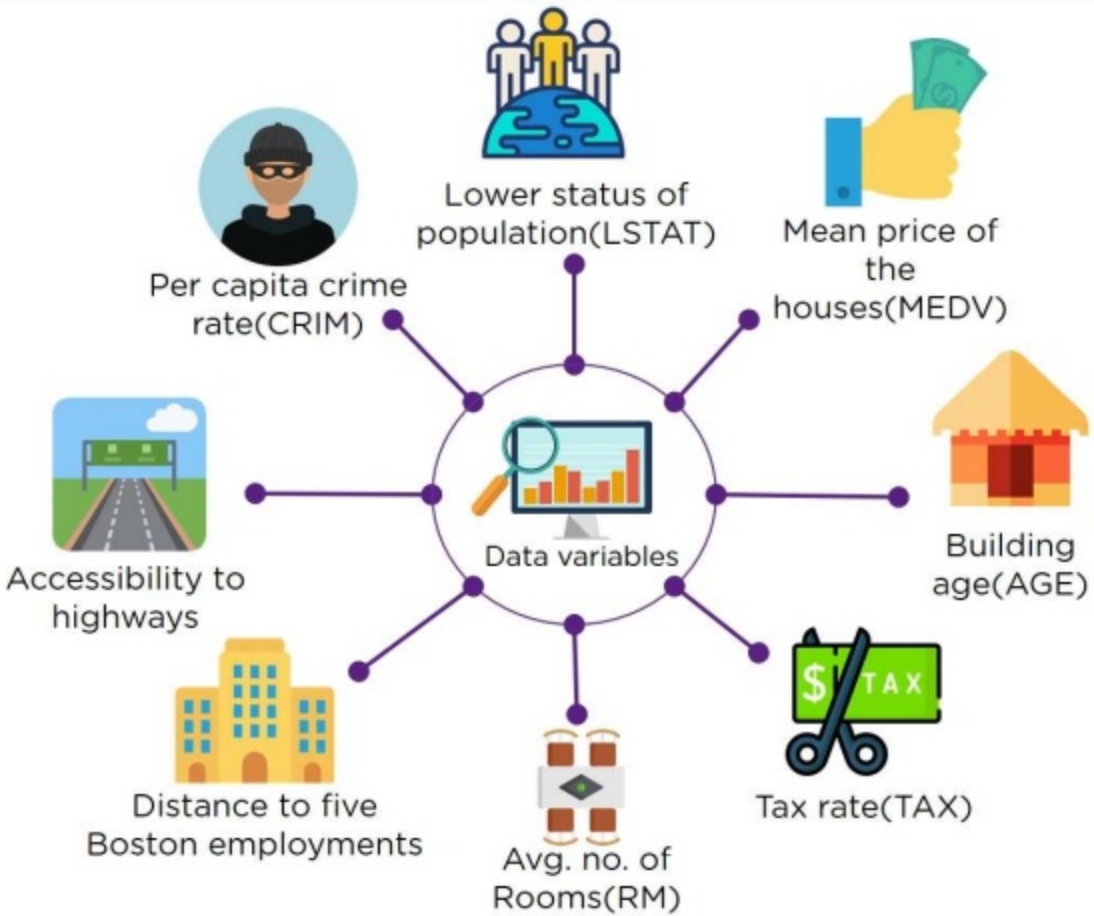
To predict the Price of a house based on a given housing dataset using Machine Learning

simplilearn

Why Linear Regression?

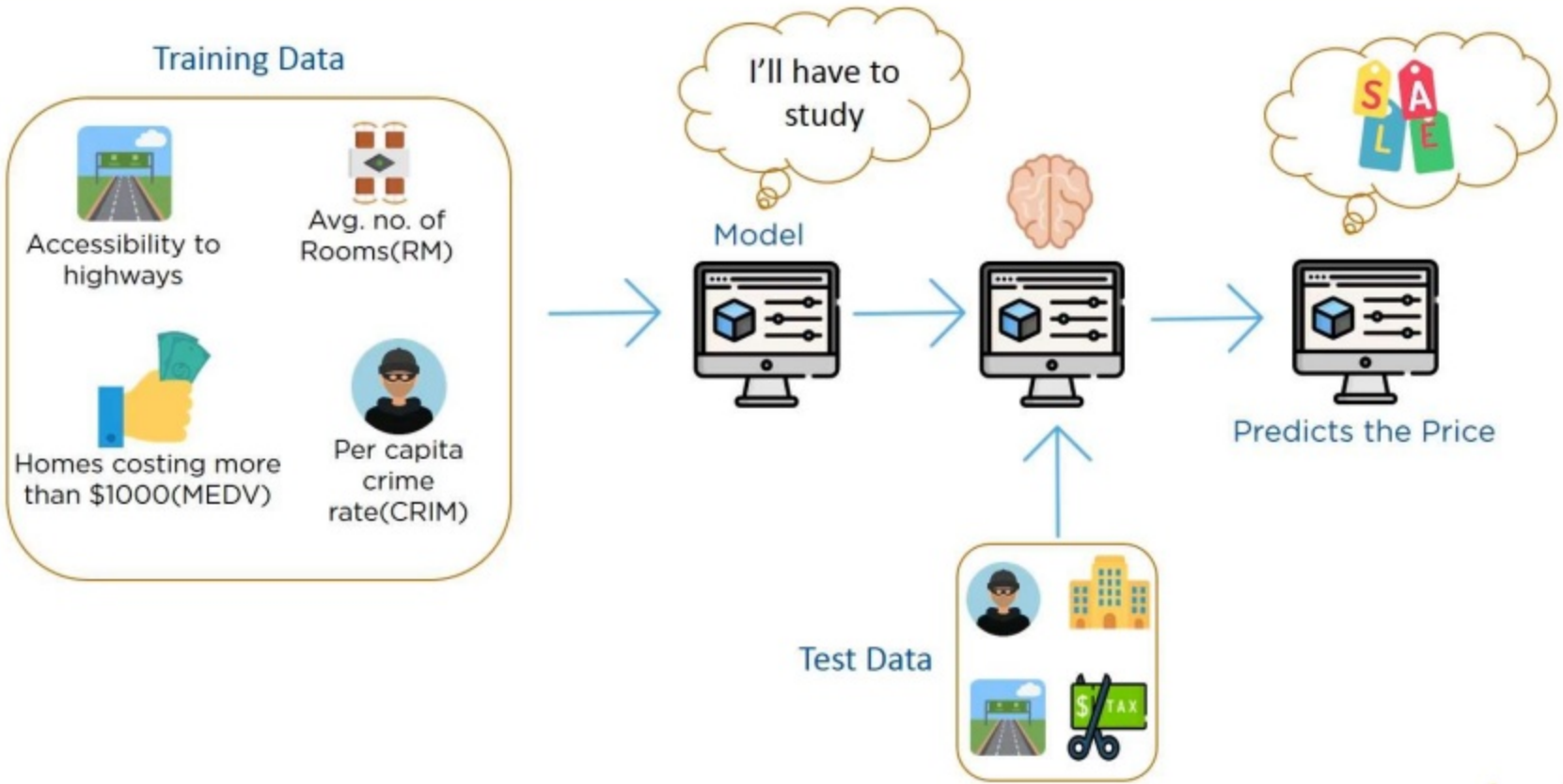

Use case - Dataset
simplilearn

Use Case - Dataset

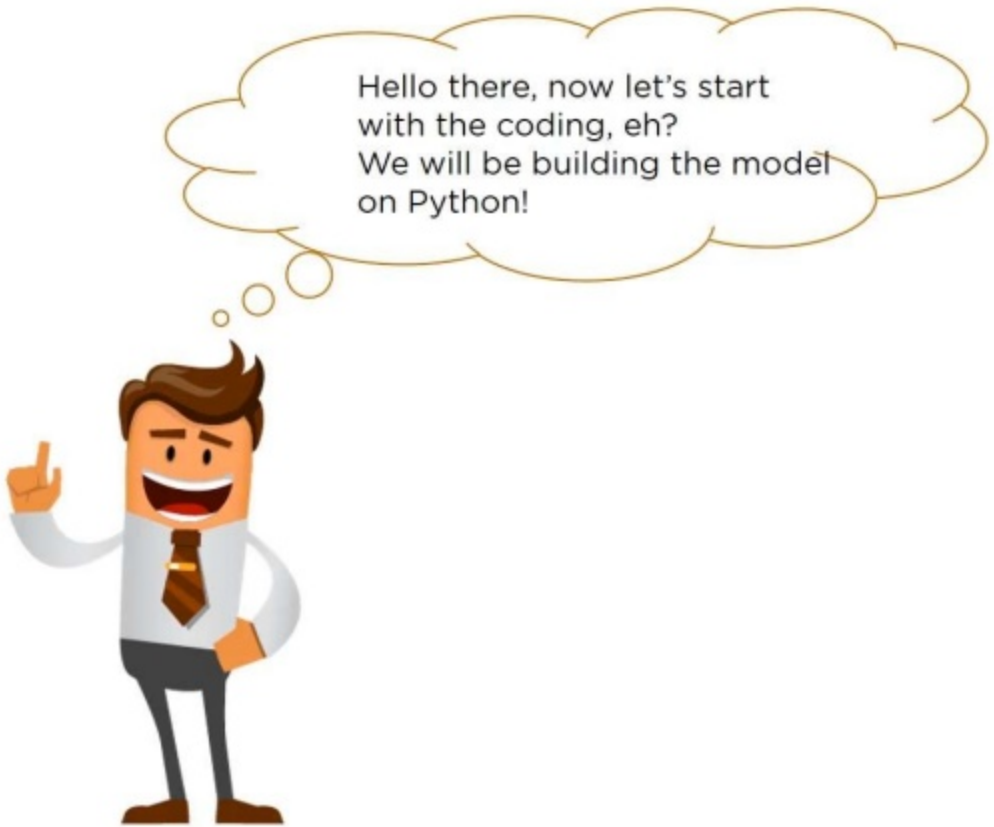


simplilearn

What is Linear Regression?



Use Case - Implementation



Use Case - Implementation

```
## importing libraries from sklearn
import numpy as np
import pandas as pd
from sklearn import linear_model
from sklearn.cross_validation import train_test_split
import numpy as np
import pandas as pd

#loading Boston dataset from datasets library
from sklearn.datasets import load_boston
boston = load_boston()

#transferring to dataframe
df_x=pd.DataFrame(boston.data,columns=boston.feature_names)
df_y=pd.DataFrame(boston.target)
```

simplilearn

Use Case - Implementation

```
#describing the boston dataset
df_x.describe()
```

Output

Out[73]:

	CRIM	ZN	INDUS	CHAS	NOX	RM
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.593761	11.363636	11.136779	0.069170	0.554695	6.284634
std	8.596783	23.322453	6.860353	0.253994	0.115878	0.702617
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500
75%	3.647423	12.500000	18.100000	0.000000	0.624000	6.623500
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000

simplilearn

Use Case - Implementation

```
#training the regression model
reg=linear_model.LinearRegression()

#splitting the data
x_train, x_test, y_train,
y_test=train_test_split(df_x,df_y,test_size=0.2,random_state=4)

#fitting the data into the model
reg.fit(x_train, y_train)

#calculating coefficients
reg.coef_
```

Output

```
Out[77]: array([[ -1.14743504e-01,   4.70875035e-02,   8.70282354e-03,
                   3.23818824e+00,  -1.67240567e+01,   3.87662996e+00,
                  -1.08218769e-02,  -1.54144627e+00,   2.92604151e-01,
                  -1.33989537e-02,  -9.07306805e-01,   8.91271054e-03,
                  -4.58747039e-01]])
```

simplilearn

Use Case - Implementation

```
#predicting the prices
a=reg.predict(x_test)
```

```
In [82]: a[]
```

```
#viewing the test data
y_test
```

Out[80]:

	0
8	16.5
289	24.8
68	17.4
211	19.3
226	37.6
70	24.2

simplilearn

Use Case - Implementation

```
#predicting the prices
a=reg.predict(x_test)
```

```
In [83]: a[2]
Out[83]: array([ 17.59242607])
```

```
Out[80]:
```

	0
8	16.5
289	24.8
68	17.4
211	19.3
226	37.6
70	24.2

```
#viewing the test data
y_test
```

Use Case - Implementation

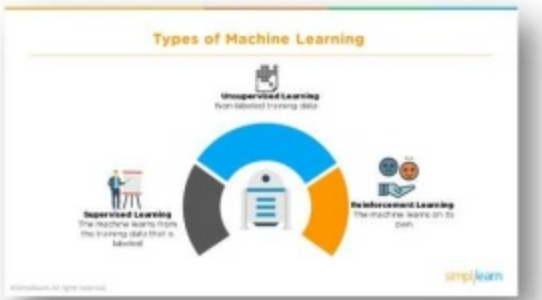
```
#finding Mean Square Error(MSE)
np.mean((a-y_test)**2)
```

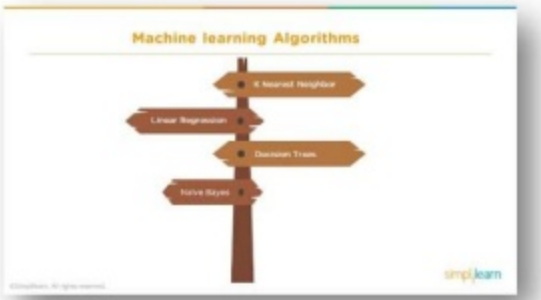
```
Out[19]: 0    25.407977
```

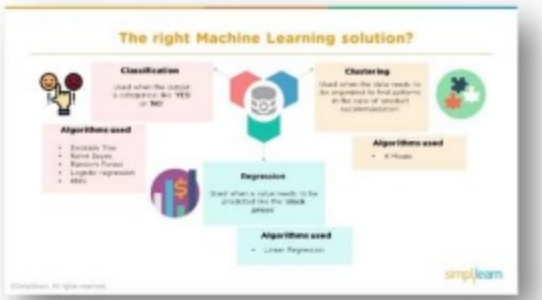
So, the MSE= 25
and
Which is quite good for our model

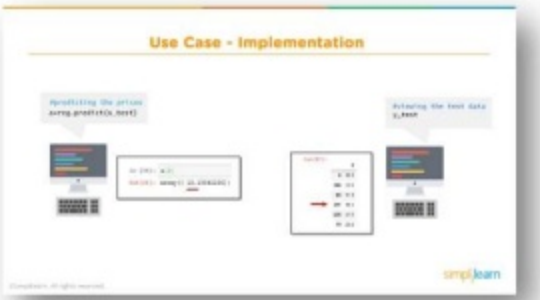
simplilearn

Key Takeaways









simplilearn