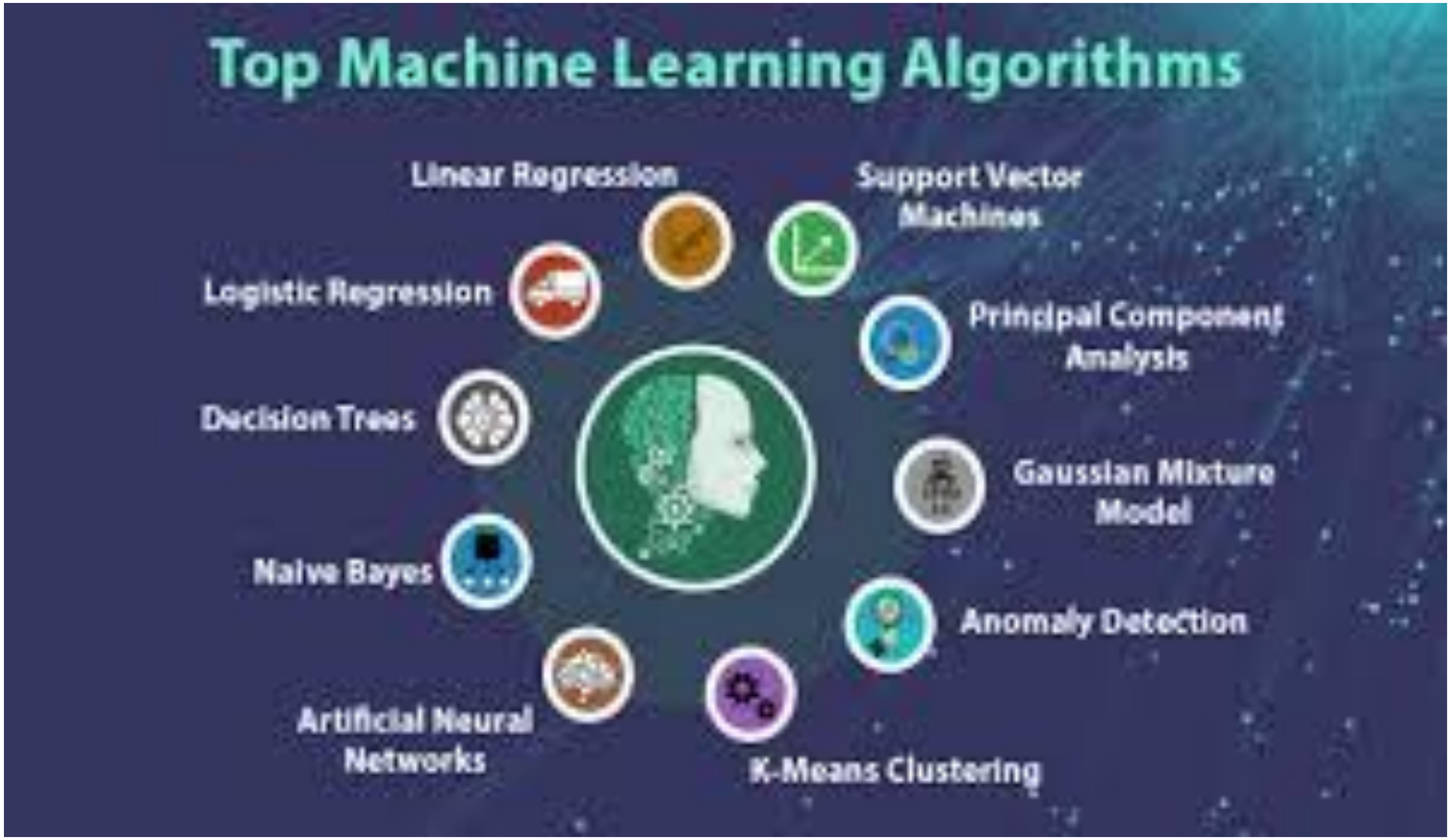


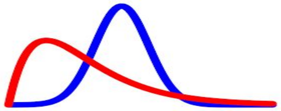
INTERNSHIPSTUDIO

Key ML Algorithms

Agenda

- Naive Bayes Algorithm
- Multinomial Naive Bayes
- Support Vector Regression
- R-Squared Intuition
- Adjusted R-Squared Intuition

Naive Bayes



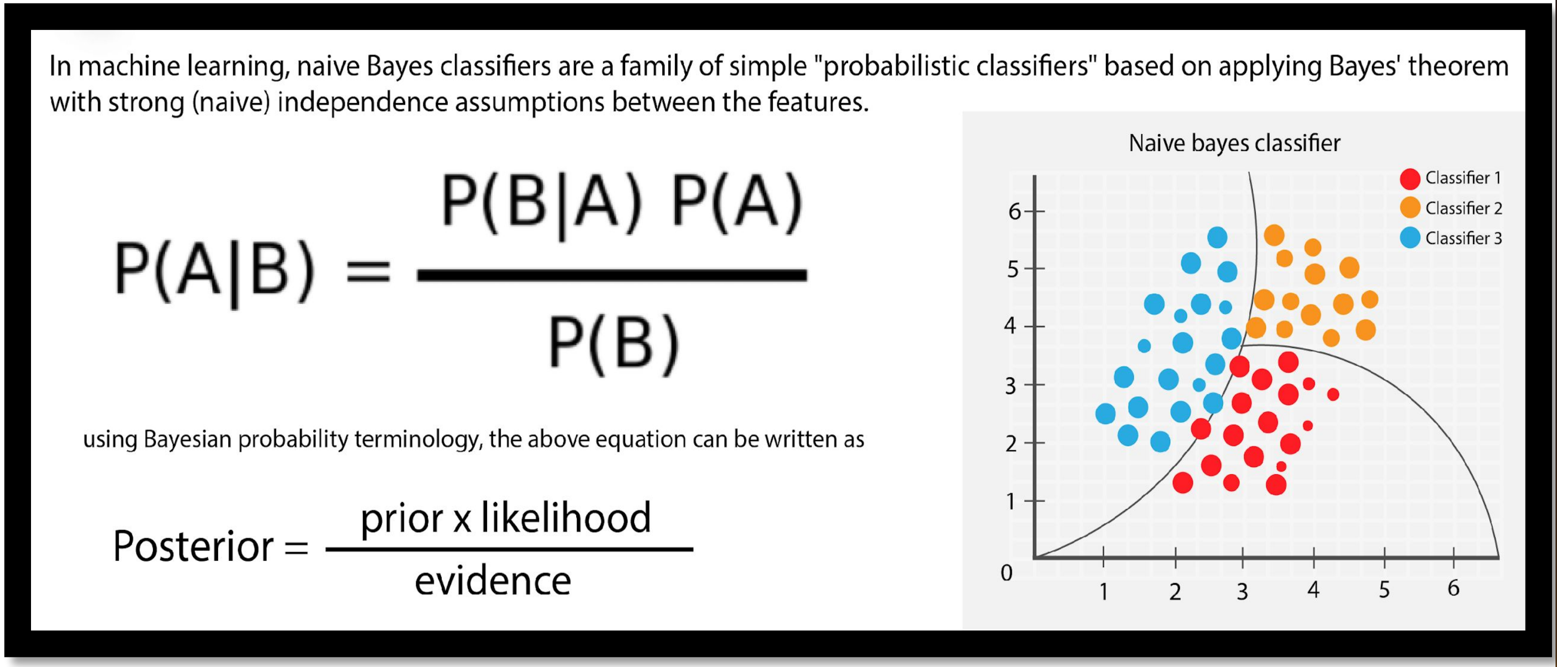
- **What is a classifier?**

  A classifier is a machine learning model that is used to discriminate different objects based on certain features.

- **Principle of Naive Bayes Classifier:-**

  A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

Bayes Theorem



In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

Naive bayes classifier
● Classifier 1
● Classifier 2
● Classifier 3
0
1
2
3
4
5
6
1
2
3
4
5
6

- Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred.
- Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent.
- That is presence of one particular feature does not affect the other. Hence it is called naive.

Conditional Probability:

- In the domain of probability theory, conditional probability is the measure of an event A occurring when another event say B has taken place. This is represented by and is read as "the conditional probability of A given B"…(A|B)
- One of the example in the domain of conditional probability is flipping the coin…the chances of having a head or tail are equal. IN other words, the probability of any of the events is 0.5. So, conditional probability talks about the chances of having a head once we already had a tail. Bayes theorem provides a mathematical model for calculating these.

Bayes Theorem:

- It describes the probability of occurrence of an event, based upon the existing knowledge about the conditions that are related to that specific event.
- Say, diabetes happens at some particular age X, and then by using the Bayes theorem, the age of a person can be used to forecast the chances that they will have diabetes and the results will be much better as compared to a situation when we had no idea about their age.

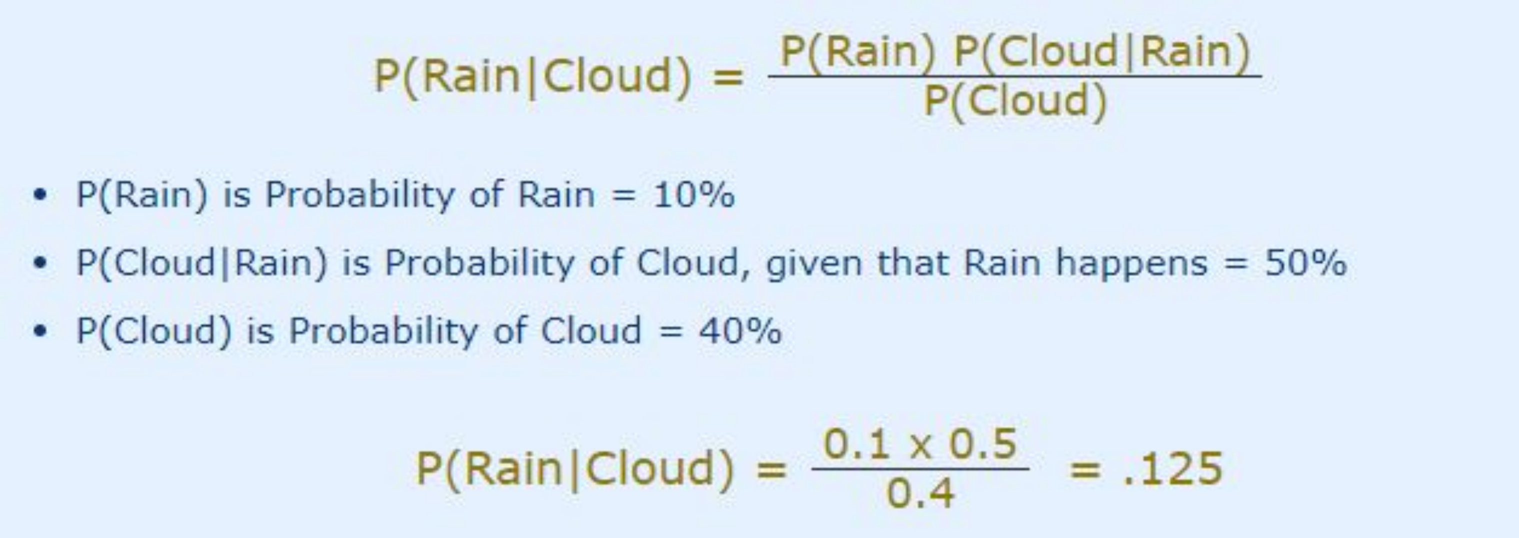
Example- Picnic Day

You are planning a picnic today, but the morning is cloudy

- Oh no! 50% of all rainy days start off cloudy!
- But cloudy mornings are common (about 40% of days start cloudy)
- And this is usually a dry month (only 3 of 30 days tend to be rainy, or 10%)

What is the chance of the rain?

- We will use Rain to mean rain during the day, and Cloud to mean cloudy morning.
- The chance of Rain given Cloud is written P(Rain|Cloud)
- So let's put that in the formula:



$$P(\text{Rain}|\text{Cloud}) = \frac{P(\text{Rain})\, P(\text{Cloud}|\text{Rain})}{P(\text{Cloud})}$$

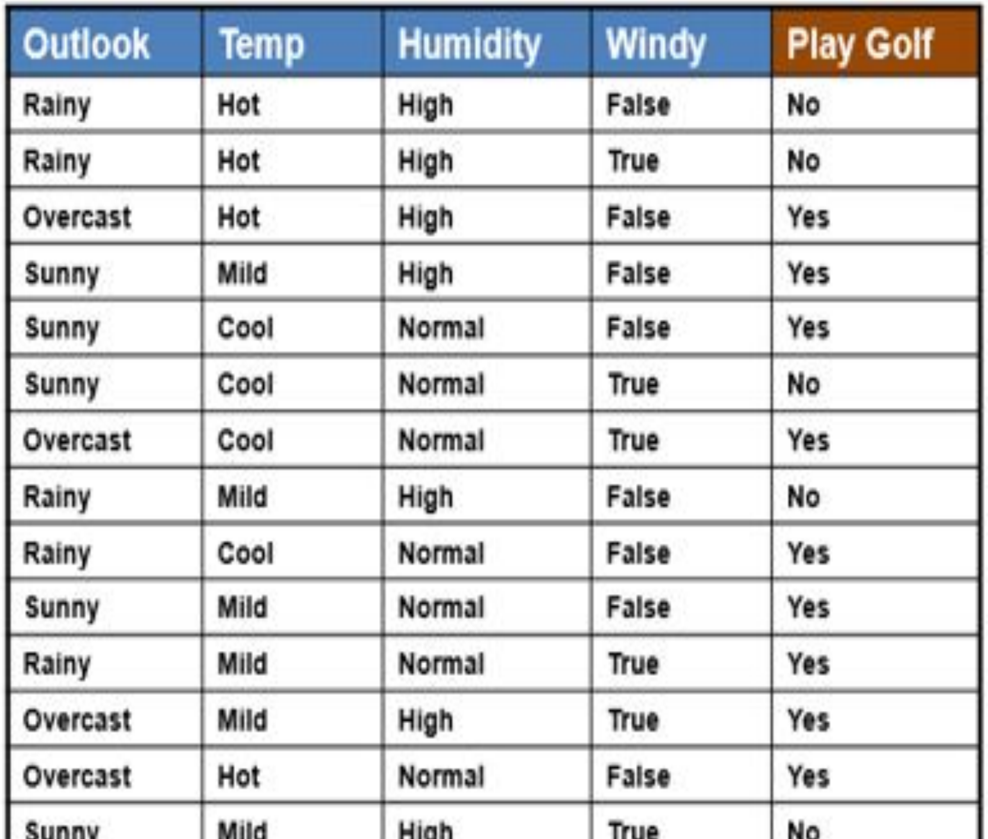
- P(Rain) is Probability of Rain = 10%
- P(Cloud|Rain) is Probability of Cloud, given that Rain happens = 50%
- P(Cloud) is Probability of Cloud = 40%

$$P(\text{Rain}|\text{Cloud}) = \frac{0.1 \times 0.5}{0.4} = .125$$

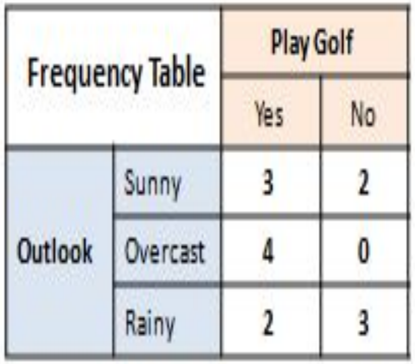
Or a 12.5% chance of rain. Not too bad, let's have a picnic!

How Naive Bayes algorithm works?

We use Weather and play golf dataset here

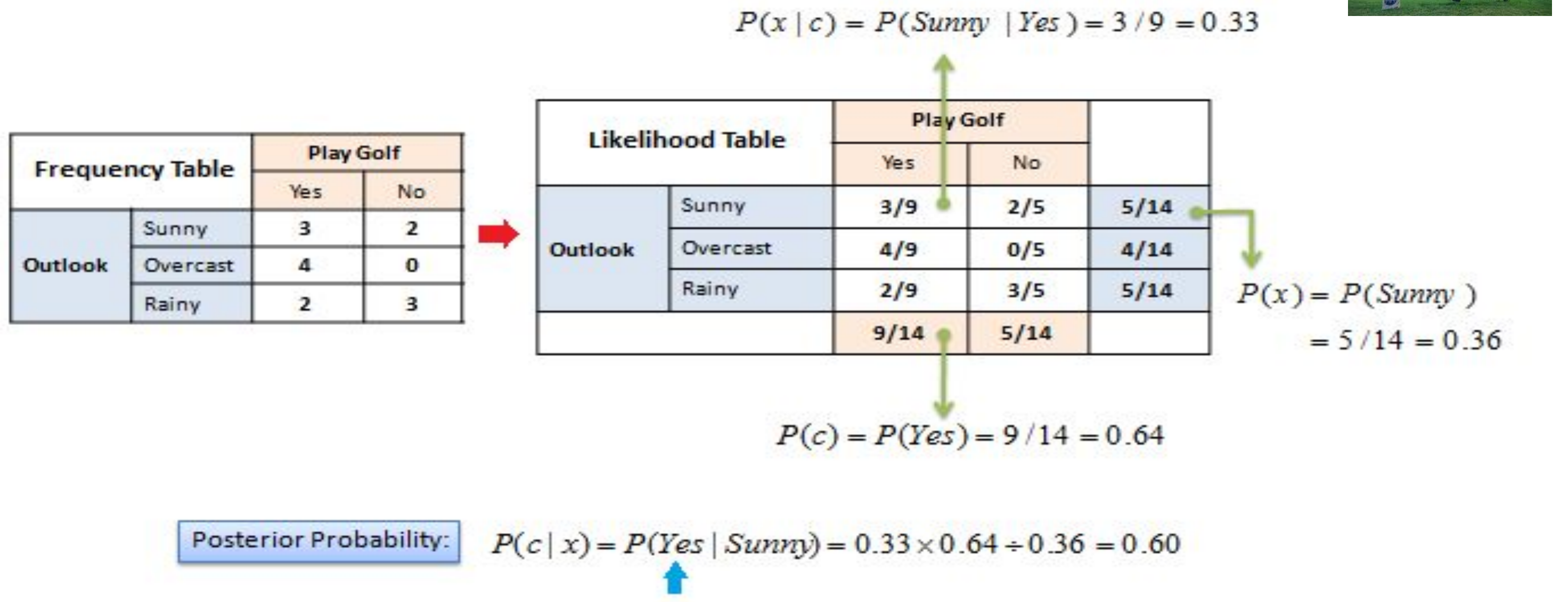


Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Frequency table for each attribute



$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$

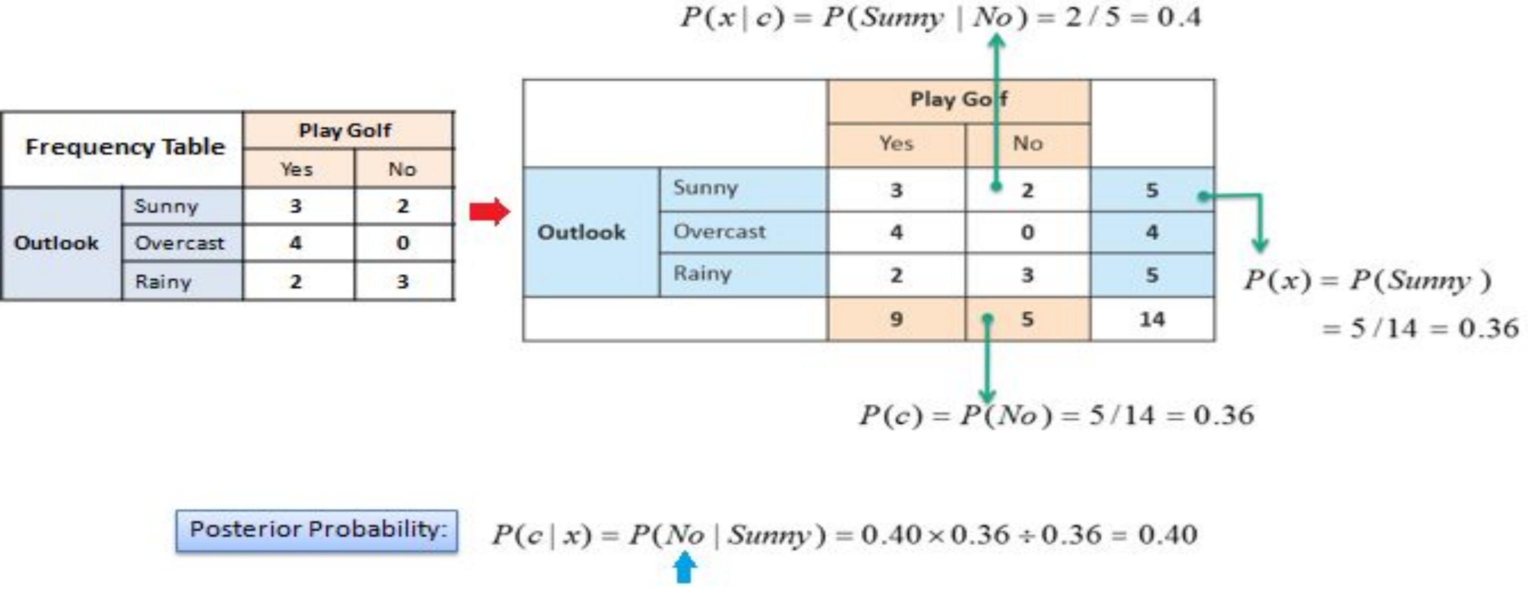
Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$P(x) = P(Sunny)$
$= 5/14 = 0.36$

$P(c) = P(Yes) = 9/14 = 0.64$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$



$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

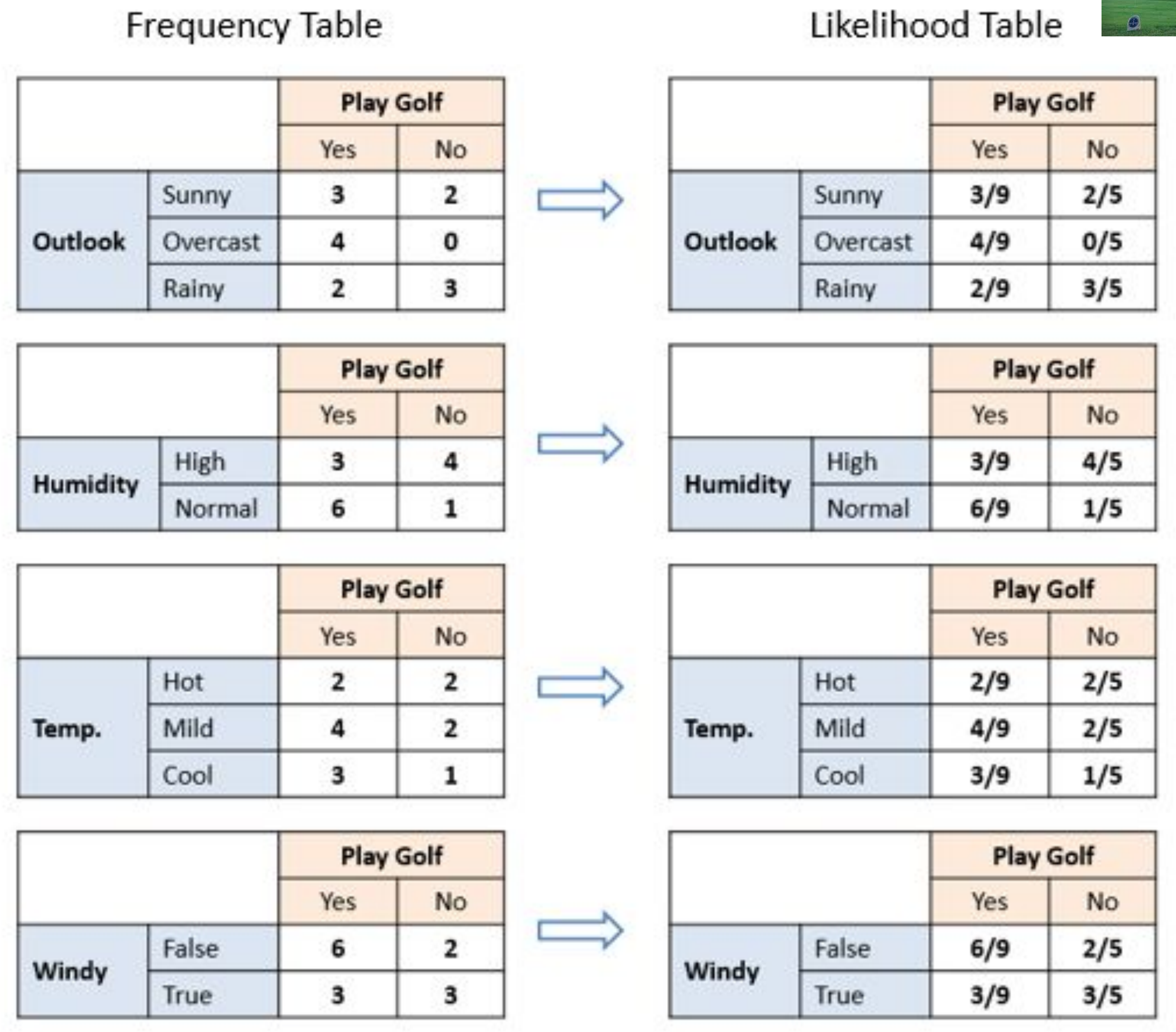
		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

$P(x) = P(Sunny)$
$= 5/14 = 0.36$

$P(c) = P(No) = 5/14 = 0.36$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

The likelihood tables for all four predictors



Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

The class with the highest posterior probability is the outcome of prediction.

Naïve Bayes Classifier – Pros & Cons

- Advantages
    - Easy to implement
    - Good results obtained in most of the cases
- Disadvantages
    - Assumption: class conditional independence, therefore loss of accuracy
    - Practically, dependencies exist among variables
        - E.g., hospitals: patients: Profile: age, family history, etc.
          Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
        - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks

Pros and Cons of Naive Bayes?

Pros:

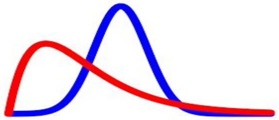
- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s).
- For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

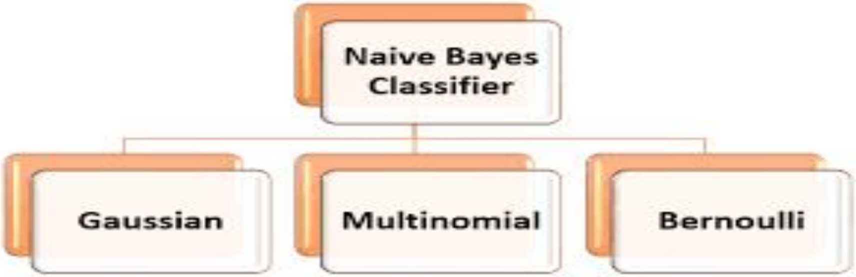
What are the Pros and Cons of Naive Bayes?

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”.
  - To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Types of Naive Bayes





Naive Bayes Classifier

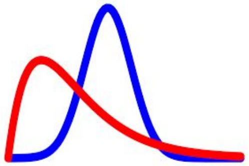
Gaussian

Multinomial

Bernoulli

Scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of model under the scikit-learn library :-

- Gaussian: It is used in classification and it assumes that features follow a normal distribution.
- Multinomial: It is used for discrete counts. For example, a text classification problem.
- Bernoulli: The binomial model is useful if your feature vectors are binary (i.e. zeros and ones).



Q.1 Define classifier in terms of ML?

Q.2 What is Bayes Theorem. Please write its formula?

Q.3 Write the Probability for picnic example in Bayes theorem?

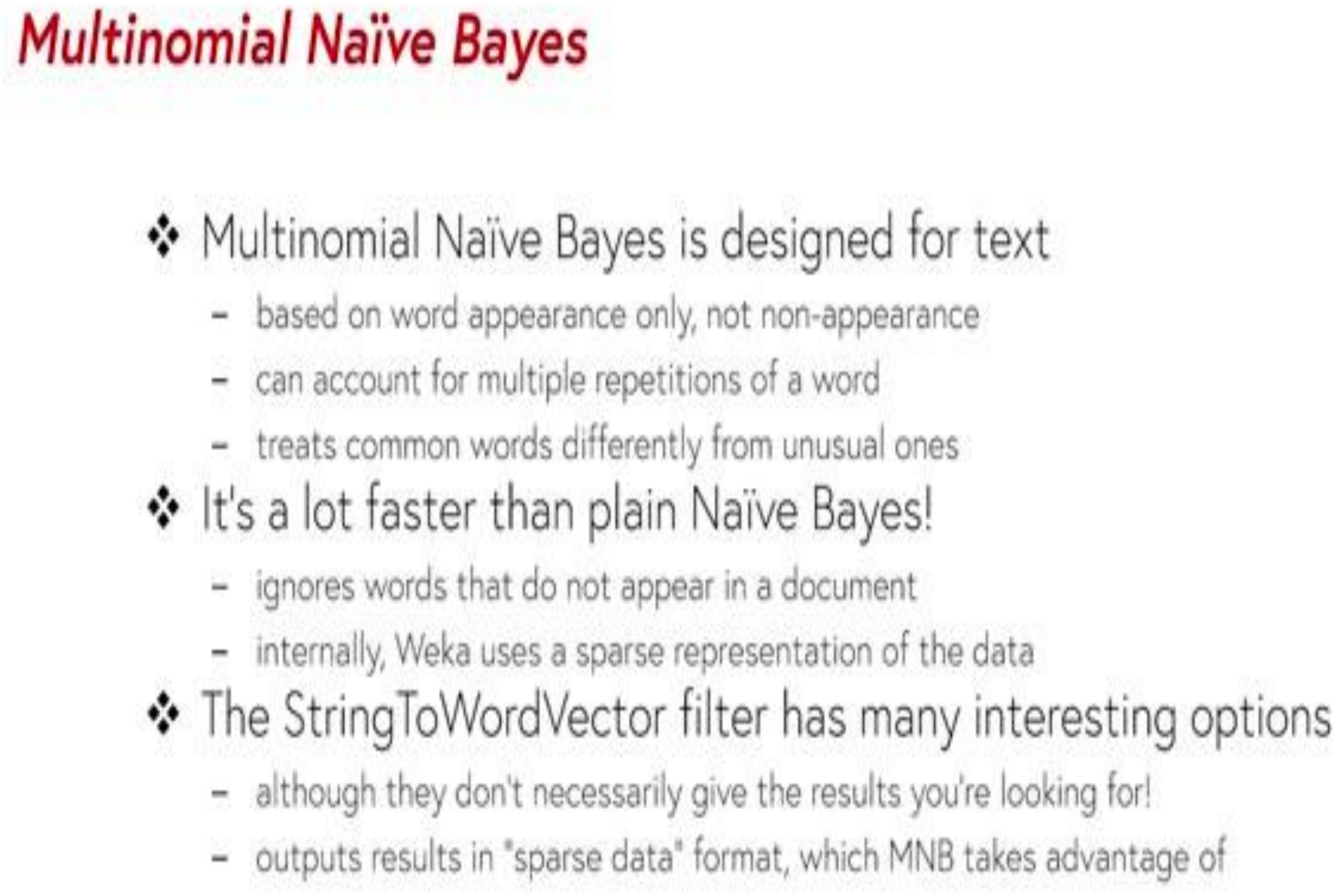
Q.4 Explain how Naïve Bayes Classifier works?

Q.5 What are the advantages and disadvantages of Naïve Bayes Classifier ?

Q.6 Explain in brief the types of Naïve Bayes Classifier ?

Q.7 Write the code for adding Naïve Bayes Classifier from sklearn library?

Multinomial Naive Bayes

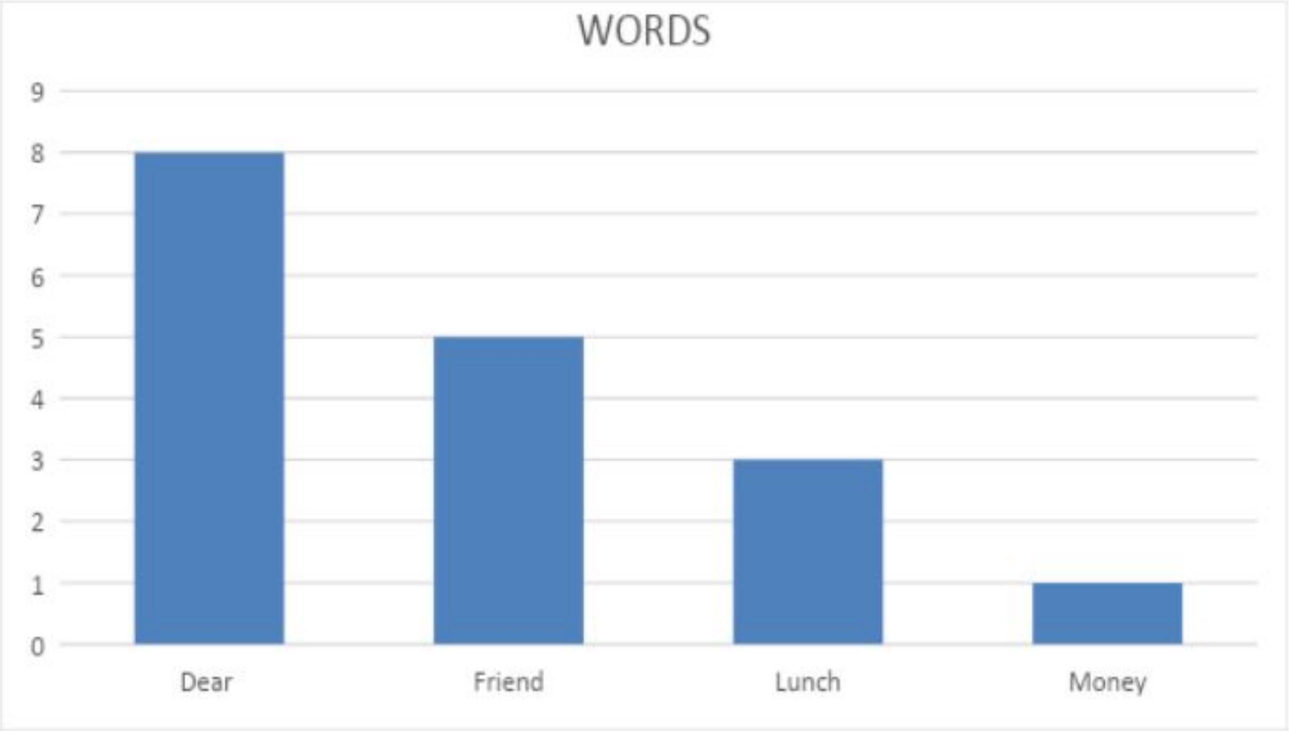


Multinomial Naïve Bayes

- ❖ Multinomial Naïve Bayes is designed for text
  - based on word appearance only, not non-appearance
  - can account for multiple repetitions of a word
  - treats common words differently from unusual ones
- ❖ It's a lot faster than plain Naïve Bayes!
  - ignores words that do not appear in a document
  - internally, Weka uses a sparse representation of the data
- ❖ The StringToWordVector filter has many interesting options
  - although they don't necessarily give the results you're looking for!
  - outputs results in "sparse data" format, which MNB takes advantage of

Example- Normal Vs Spam mail

- We received a lot of emails from friends, family, office and we also receive spam mails. Initially, we consider eight normal messages and four spam messages.
- Let see the histogram of all the words that occur in the normal messages from family and friends.



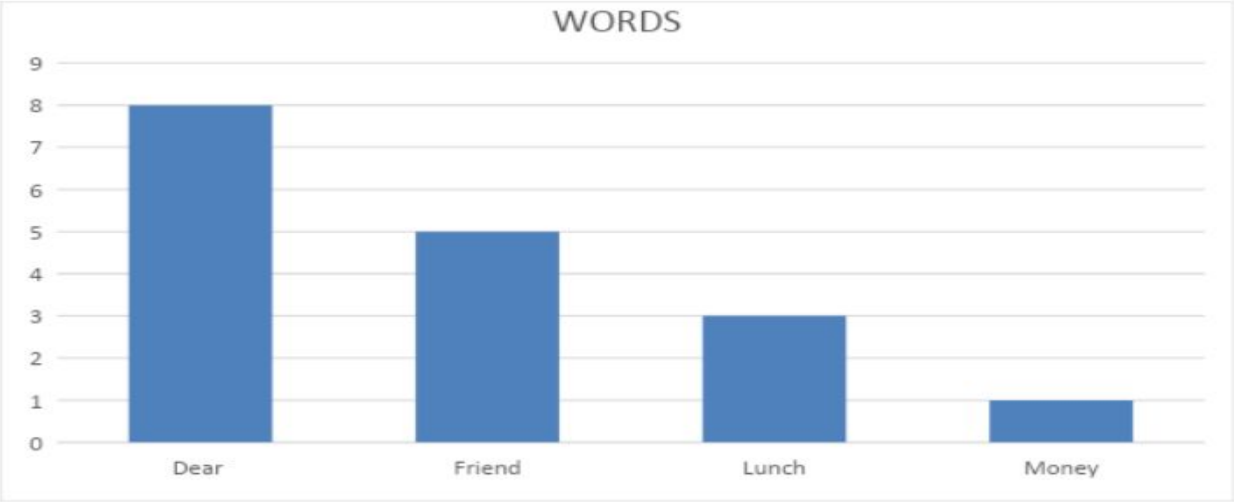
Normal mail

- We can use the histogram to calculate the probabilities of seeing each word, given that it was a normal message.
  The probability of word dear given that we saw in normal message is-

- **Probability (Dear|Normal) = 8 /17 = 0.47**

Similarly, the probability of word Friend is-

- Probability (Friend/Normal) = 5/ 17 =0.29
- Probability (Lunch/Normal) = 3/ 17 =0.18
- Probability (Money/Normal) = 1/ 17 =0.06



Normal Vs Spam mail

Now, let’s say we have received a normal message as **Dear Friend** and we want to find out if it’s a normal message or spam.

- We start with an initial guess that any message is a Normal Message.
- From our initial assumptions of 8 Normal messages and 4 Spam messages, 8 out of 12 messages are normal messages. The prior probability, in this case, will be:
- **Probability (Normal) = 8 / (8+4) = 0.67**
- We multiply this prior probability with the probabilities of **Dear Friend** that we have calculated earlier.
- **0.67 * 0.47 * 0.29 = 0.09**

0.09 is the probability score considering **Dear Friend** is a normal message.

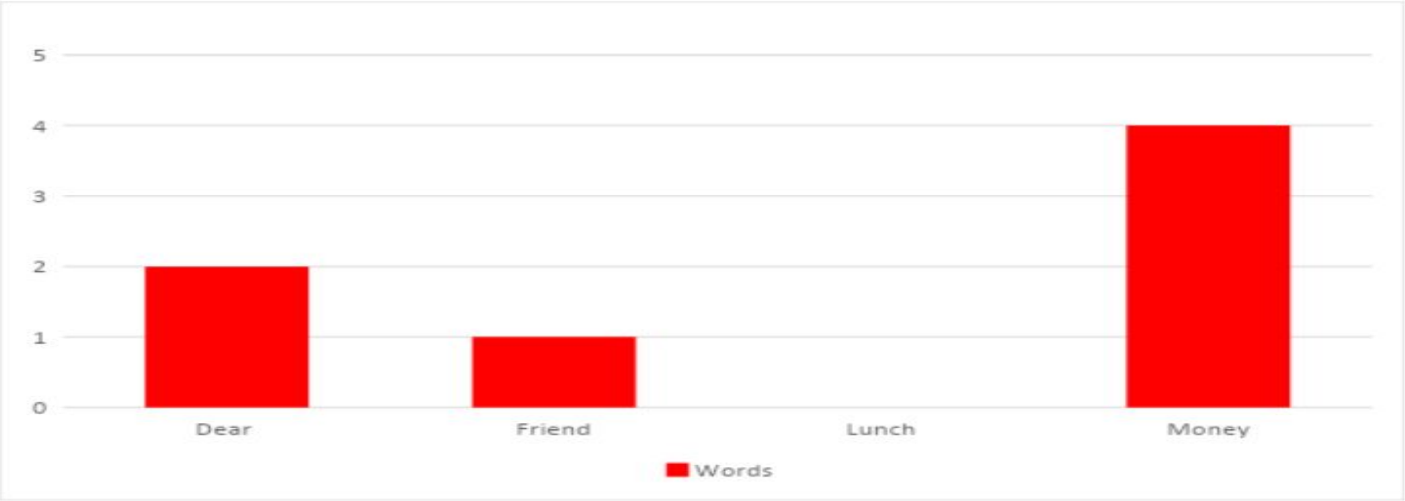
Spam mail

- The probability of word dear given that we saw in spam message is-

Probability (Dear|Spam) = 2 /7 = 0.29

- Similarly, the probability of word Friend is-

Probability (Friend/Spam) = 1/ 7 =0.14
Probability (Lunch/Spam) = 0/ 7 =0.00
Probability (Money/Spam) = 4/ 7 =0.57



Normal Vs Spam mail

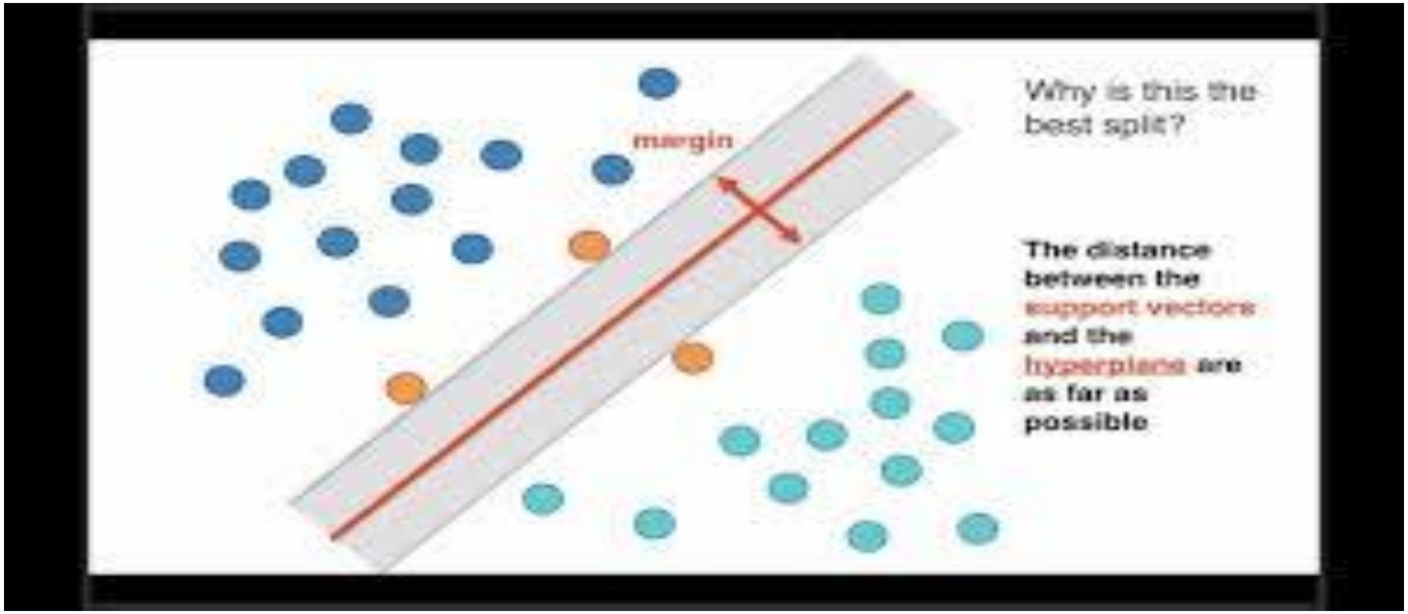
- Alternatively, let's say that any message is a Spam.
  - 4 out of 12 messages are Spam. The prior probability in this case will be:
  - **Probability (Normal) = 4 / (8+4) = 0.33**
  - Now we multiply the prior probability with the probabilities of **Dear Friend** that we have calculated earlier.
  - **0.33 * 0.29 * 0.14 = 0.01**

0.01 is the probability score considering **Dear Friend** is a Spam.

Conclusion- The probability score of **Dear Friend** being a normal message is greater than the probability score of **Dear Friend** being spam. We can conclude that **Dear Friend** is a normal message.

What is a Support Vector Machine?

- **Support Vector Machine** is a discriminative algorithm that tries to find the optimal hyperplane.
- In a 2D space, a hyperplane is a line that optimally divides the data points into two different classes.
- In a higher-dimensional space, the hyperplane would have a different shape rather than a line.



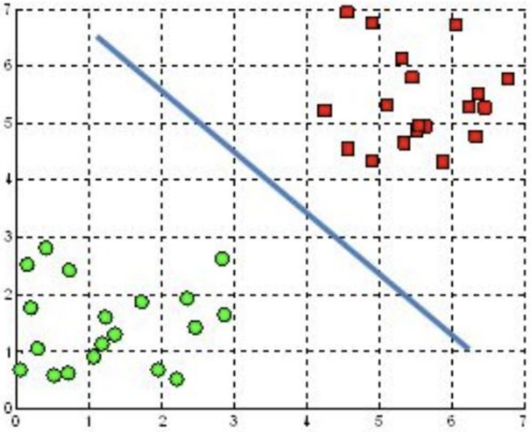
Why is this the best split?

The distance between the support vectors and the hyperplane are as far as possible

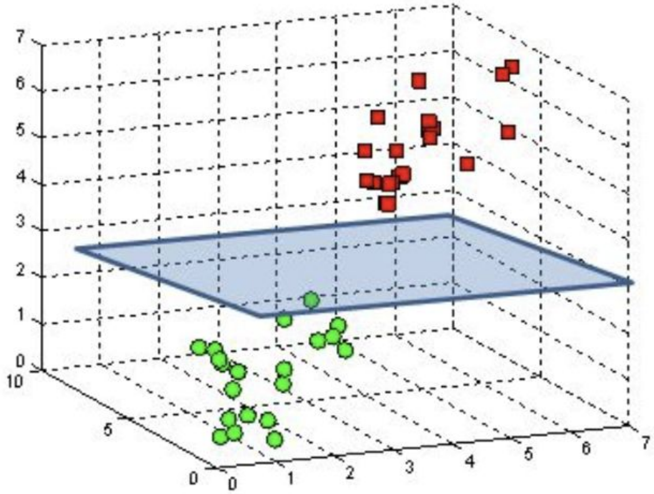
Support Vector Machine

- **Hyperplane**: A hyperplane is a plane which is used to divide categories based on their values.
- A hyperplane is always 1 dimension less than the actual plane used for plotting the outcomes or for analyses.
  - Linear Regression with 1 feature and 1 outcome we can make a 2-D plane to depict the relationship and the regression line fitted to that is a 1-D plane.
  - Similarly, for a 3-D relationship, we get a 2-D hyperplane.

A hyperplane in $\mathbb{R}^2$ is a line

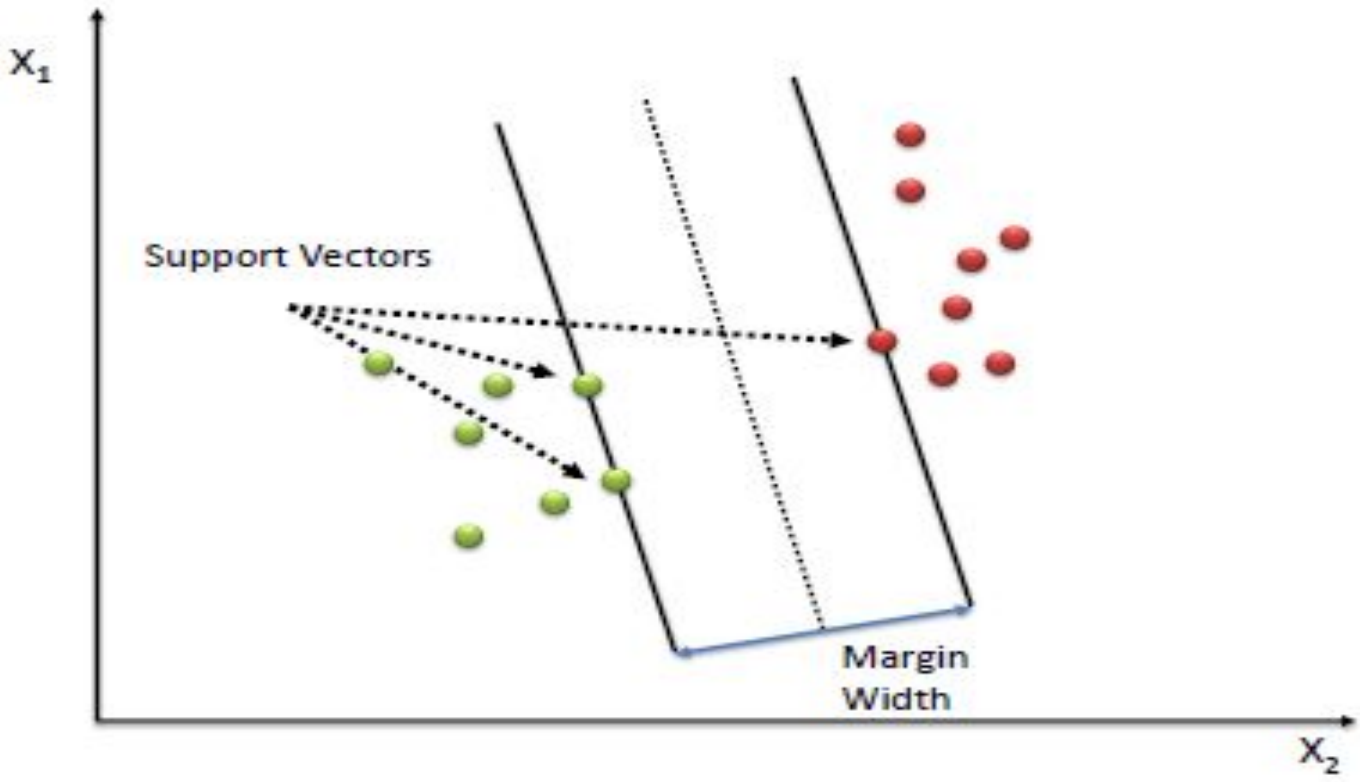


A hyperplane in $\mathbb{R}^3$ is a plane



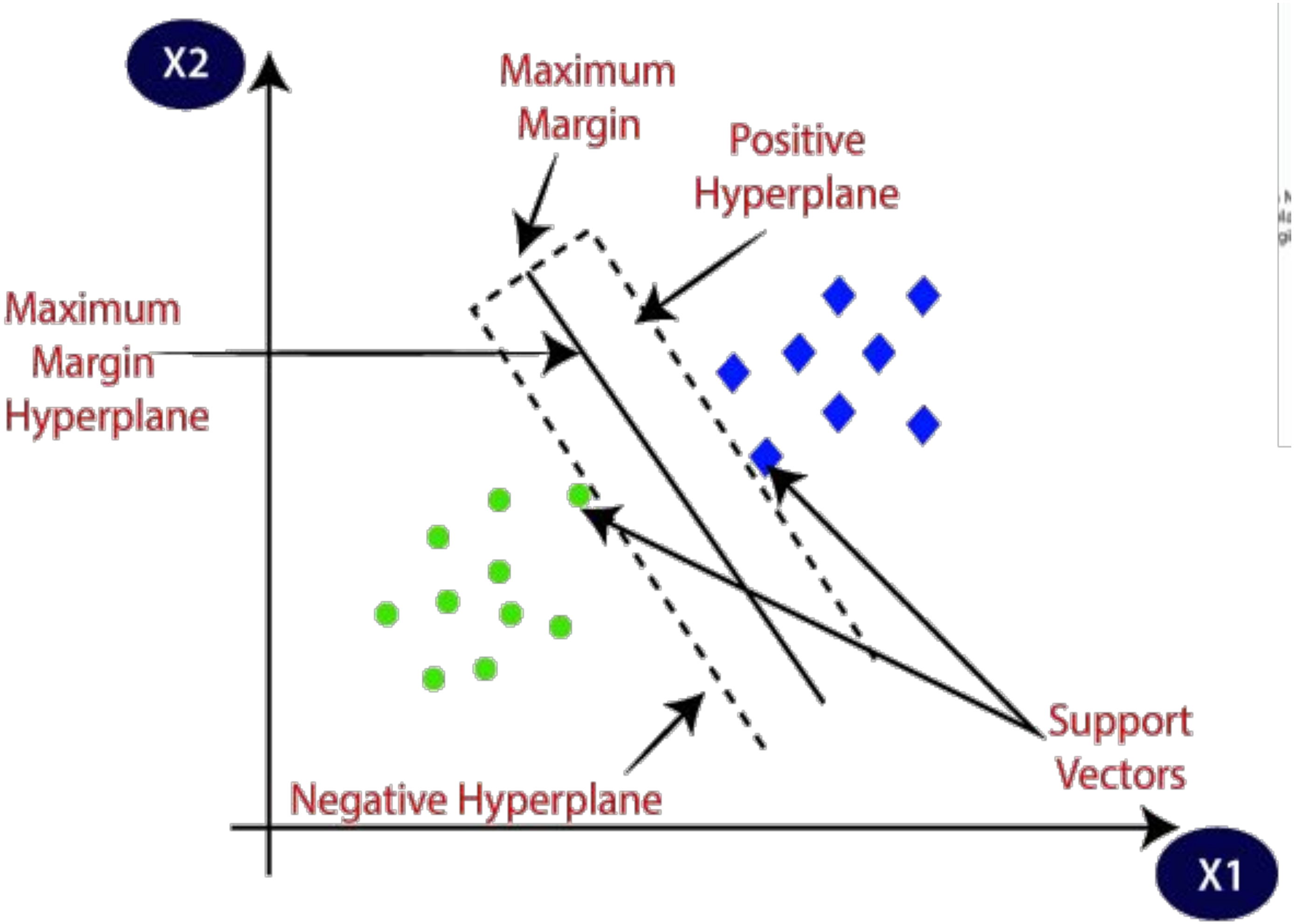
Support Vector Machine

- **Support Vectors**: Support Vectors are those points in the space that are closer to the hyperplane and also decide the orientation of the hyperplane.
- The lines or planes drawn is called Support Vector Lines or Support Vector Planes.



Margin Width

The perpendicular distance between the 2 support vector lines or planes is called Margin Width.

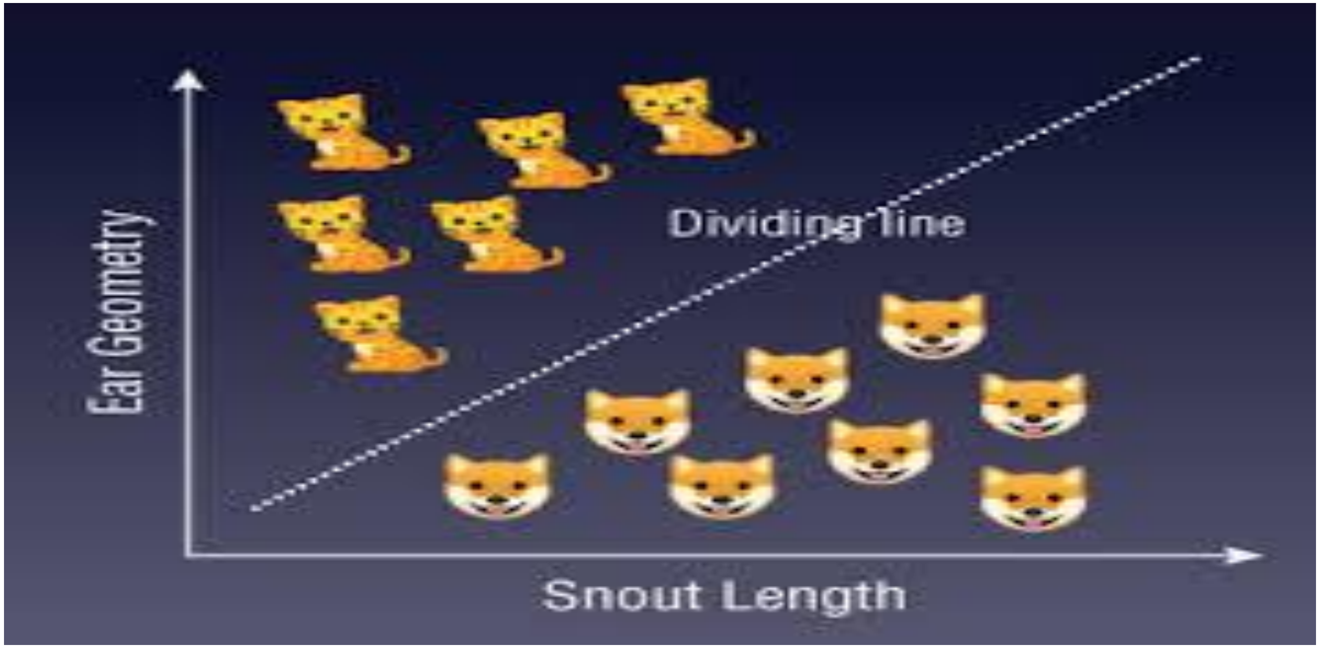


SVM example

Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog

- We will first train our model with lots of images of cats and dogs so that it can learn about different features
- and then we test it with this strange creature.

- So as support vector creates a decision boundary between these two data and choose extreme cases (support vectors).



Support Vector Regression

The key advantages

- SVM works really well with high dimensional data. If your data is in higher dimensions, it is wise to use SVR.
- For data with a clear margin of separations, SVM works relatively well.
- When data has more features than the number of observations, SVM is one of the best algorithms to use.
- As a discriminative model, it need not memorize anything about data. Therefore, it is memory efficient.

Some drawbacks

- It is a bad option when the data has no clear margin of separation i.e. the target class contains overlapping data points.
- It does not work well with large data sets.
- For being a discriminative model, it separates the data points below and above a hyperplane. So, you will not get any probabilistic explanation of the output.
- It is hard to understand and interpret SVM as its underlying structure is quite complex.

QUIZ!

Q.1 Explain Multinomial Naïve Bayes?

Q.2 Write the probabilities of different normal words coming in Multinomial Naïve Bayes example?

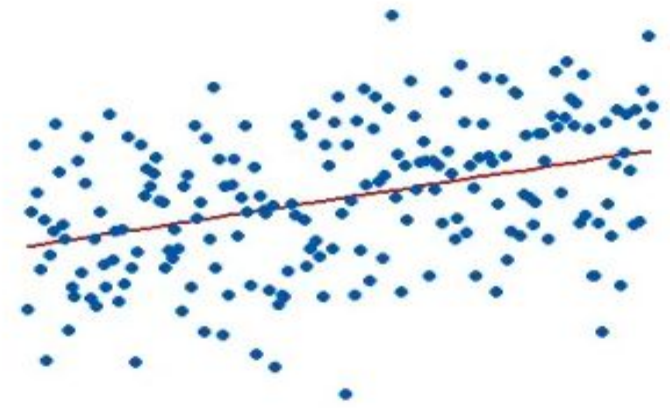
Q.3 Define support boundaries and margin in SVM?

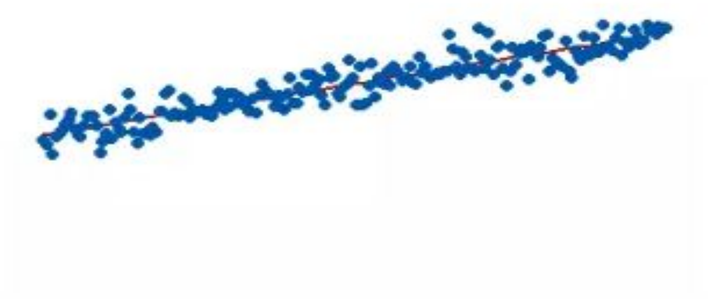
Q.4 Define support vectors and hyperplane.

Q.5 What are the key advantages/disadvantages of SVR.

What is R-squared?

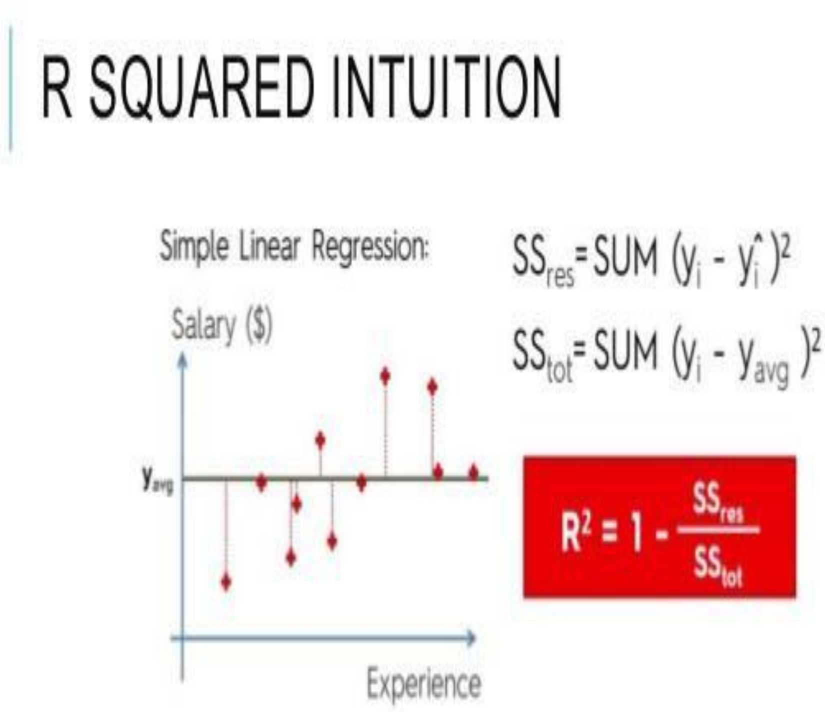
- We evaluate our algorithms (especially the regression algorithms) to see which does the work best for us.
- At times one algorithm might have a edge over the other one but it might not in other cases so to check this we use a method called R-squared intuition which helps to predict how effective our algorithm is
- The R-squared for the regression model on the left is 15%, and for the model on the right it is 85%.





How good is our model?

- Take the sum of differences between the points and our linear model line
- Check if its really less than the differences that the average line



R SQUARED INTUITION

Simple Linear Regression:

$$SS_{res} = SUM\ (y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM\ (y_i - y_{avg})^2$$

$$\boxed{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}}$$

Sum Squared Regression Error

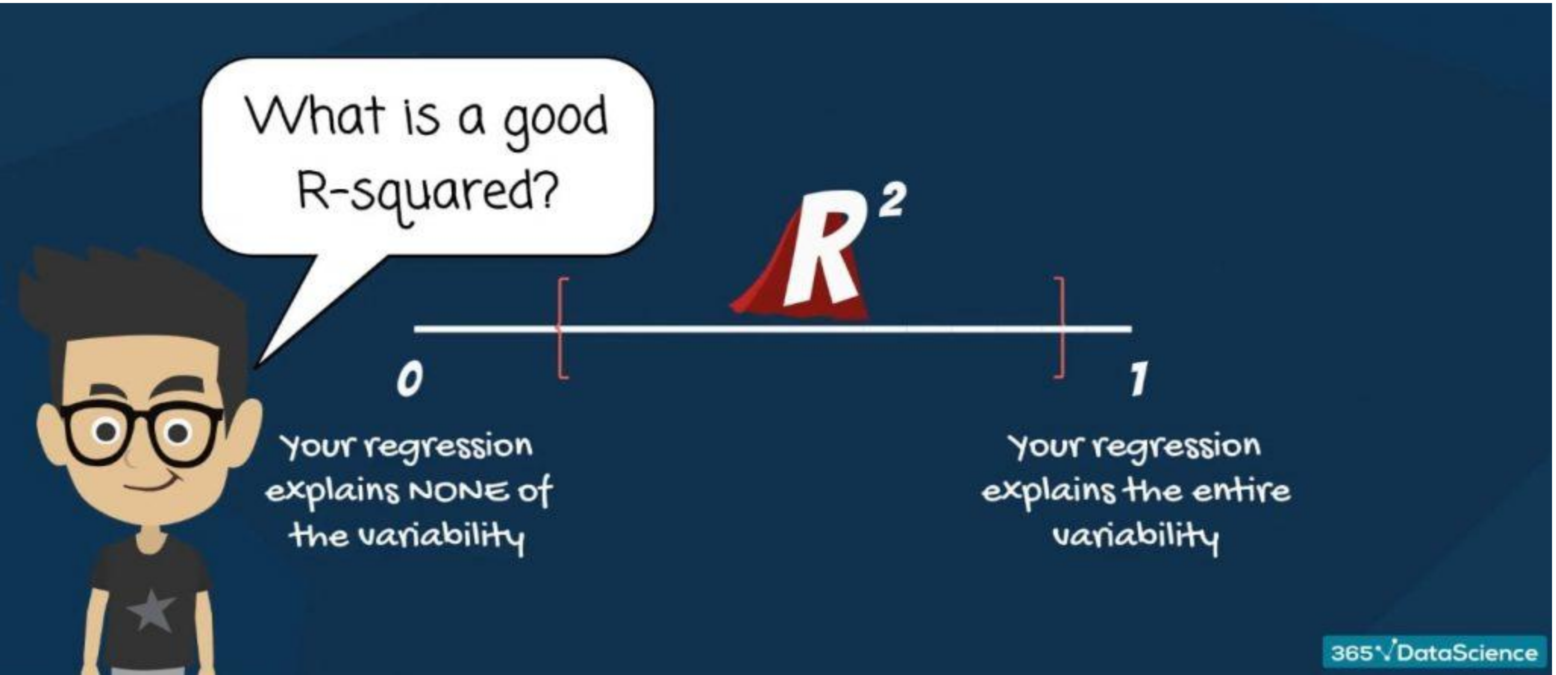
$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Sum Squared Total Error

Good R-Squared

R-squared is always between 0 and 100% :-

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.
- The value of R-square can also be negative when the models fitted is worse than the average fitted model.


What is a good R-squared?
R^2
0
1
Your regression explains NONE of the variability
Your regression explains the entire variability
365√DataScience

Problems with R-Squared

- R^2 increases with every predictor added to a model.
- As R^2 always increases, with the more terms you add to the model. This can be completely misleading.
- Similarly, if the model has too many terms and too many high-order polynomials you can run into the problem of over-fitting the data and can lead to misleading projections.
- The idea behind adjusted R-squared is to penalize the score as we add more features to our model.
- Let’s look at the formula of adjusted R-squared.

Adjusted R-square

$$adj\,R^2 = 1 - (1 - R^2)\frac{n-1}{n-m-1}$$

n is the number of data points; m is the number of independent features

- Adjusted r-square is a modified form of r-square whose value increases if new predictors tend to improve model's performance and decreases if new predictors does not improve performance as expected.
- It is always lower than the R-squared.
- In the simplified Best Subsets Regression output, where the adjusted R-squared peaks, and then declines. Meanwhile, the R-squared continues to increase.

Q.1 What is of R-squared?

Q.2 What is good R-squared?

Q.3 Pls elaborate the p**roblems with** R-Squared?

Q.4 How Adjusted R-square overcomes the problems?