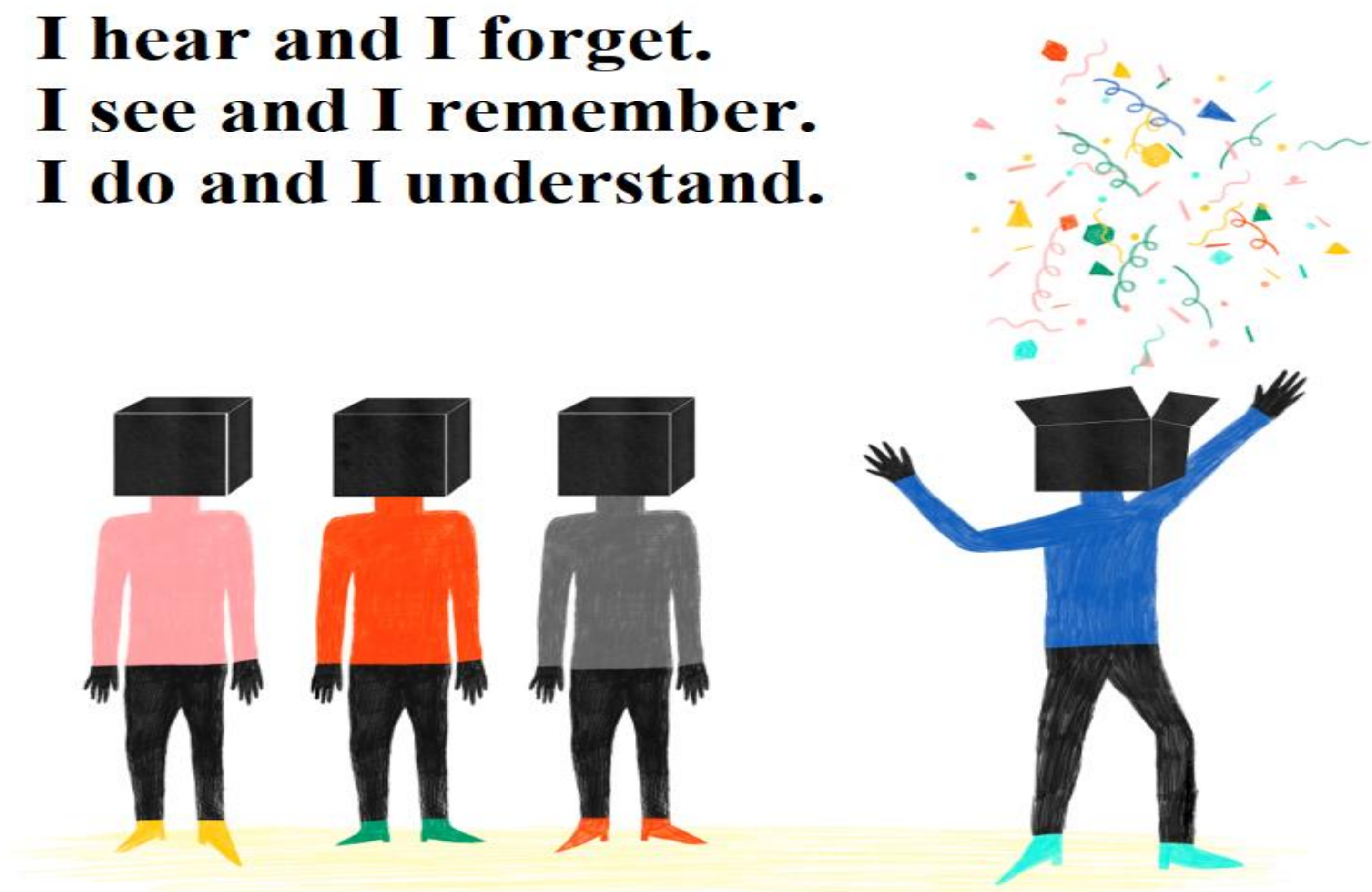
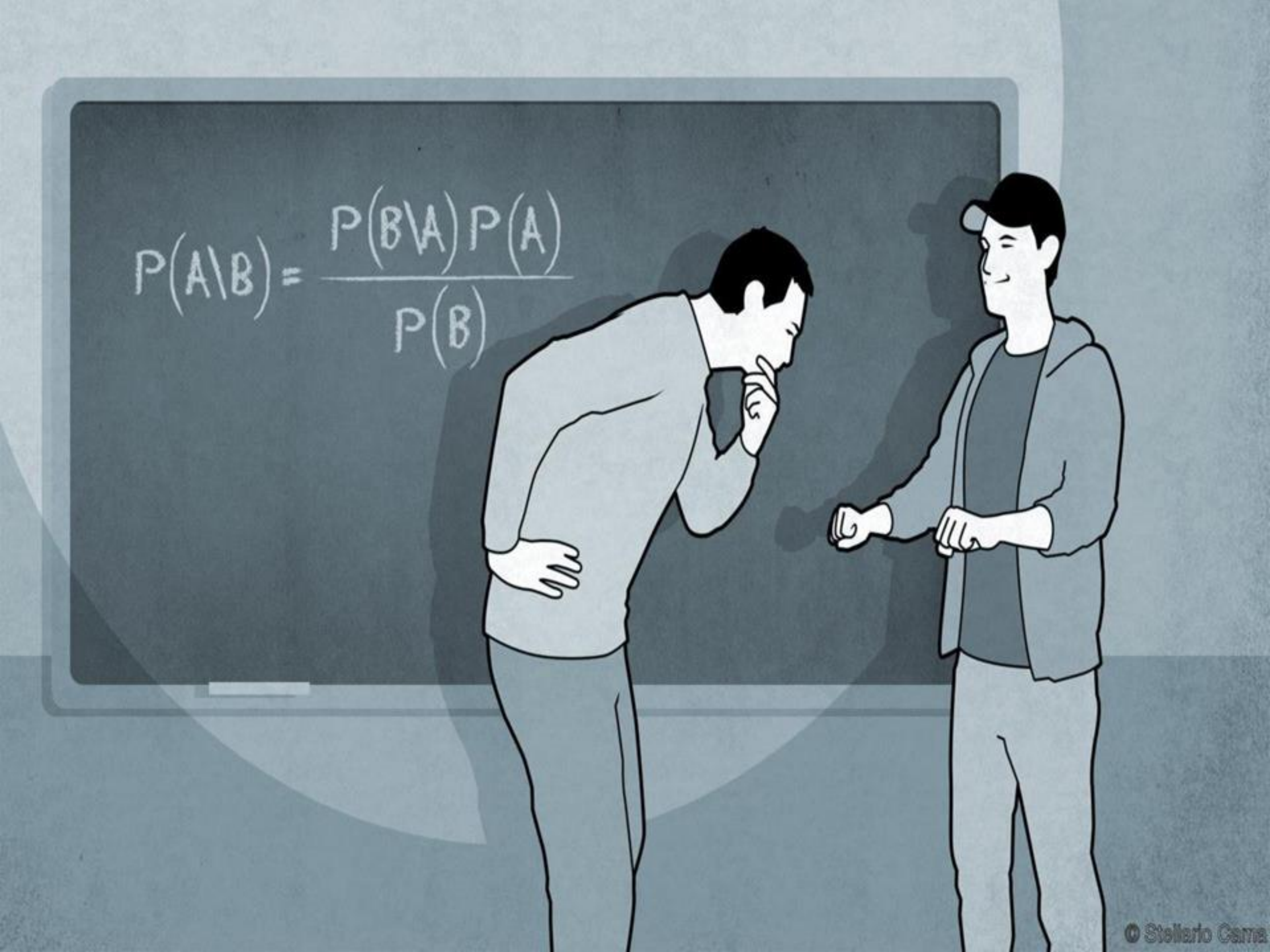


Chandan Verma

Corporate Trainer(Machine Learning,AI,Cloud Computing,IOT)

www.facebook.com/verma.chandan.070


P(A\B) = P(B\A) P(A) / P(B)
© Stellario Cama

TOM

JAK

[dog, ball wonderful]

[love, great, wonderful]

Lets assume you received and anonymous email
“I love beach sand. Additionally the sunset at beach offers wonderful view”

TOM	PROBABILITY	JAK	PROBABILITY
LOVE	.1	WONDERFUL	.5
WONDERFUL	.1	LOVE	.3
GREAT	.8.	DEAL	.2

"Wonderful Love."

Naive Bayes

Naive Bayes Classifier is a **supervised** and probabilistic learning method.

Naïve Bayes classifier works effectively for classifying emails, texts, symbols, and names.

The naive Bays algorithm is called "naive" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other feature.

Bayes

He is the statistician and Philosopher.

It calculates the probability that one event (A) is true, given that another event (B) is also true.

Random Experiment

Random Experiment: An experiment is said to be a random experiment, if its out-come can't be predicted with certainly.

The outcomes of a random experiment are called events connected with the experiment.

Events

Simple or elementary event: An event, consisting of a single point is called a simple event.

Compound or mixed event: A subset of the sample space which has more than one element is called a mixed event.

Equally likely events: Events are said to be equally likely, if we have no reason to believe that one is more likely to occur than the other.

Certain Events: An event which is sure to occur at every performance of an experiment is called a certain event connected with the experiment.

Impossible Even: An event which cannot occur at any performance of the experiment is called an possible event.

Exhaustive events: When every possible outcome of an experiment is considered, the observation is called exhaustive events.

All the possible outcomes of the experiments are known as exhaustive events.

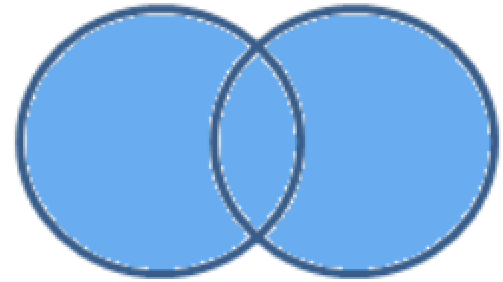
Mutually Exclusive Events: If there be no element common between two or more events, i.e., between two or more subsets of the sample space, then these events are called mutually exclusive events.

If two or more events can't occur simultaneously, i.e. no two of them can occur together.

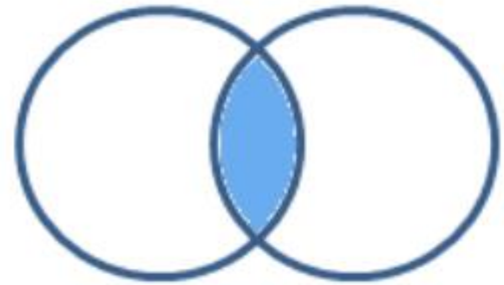
Independent or Mutually independent Events: Two or more events are said to be independent, if occurrence of non-occurrence of any of them does not affect the probability of occurrence or non-occurrence of other event.

Mutually exclusiveness is used, when the events are taken from the same experiment, whereas the independence is used, when the events are taken from different experiment

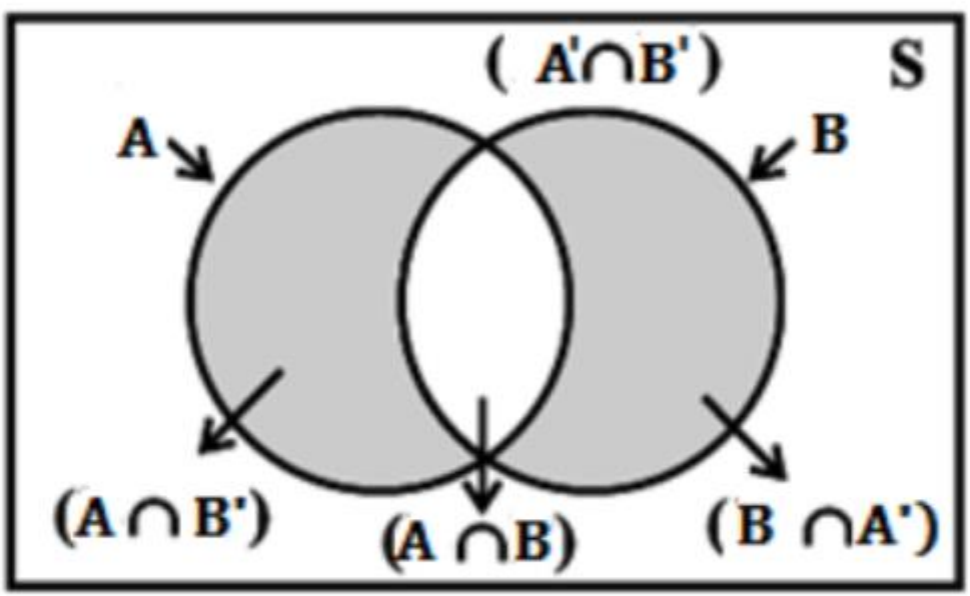
Union of Events



Intersection of Events



Disjoint Events



Probability

Probability tells us how likely something is to happen.

Probability is a way of expressing what the chances are that an event will occur.

$$\boxed{\text{Probability}(\text{Head})} = \frac{1}{2}$$

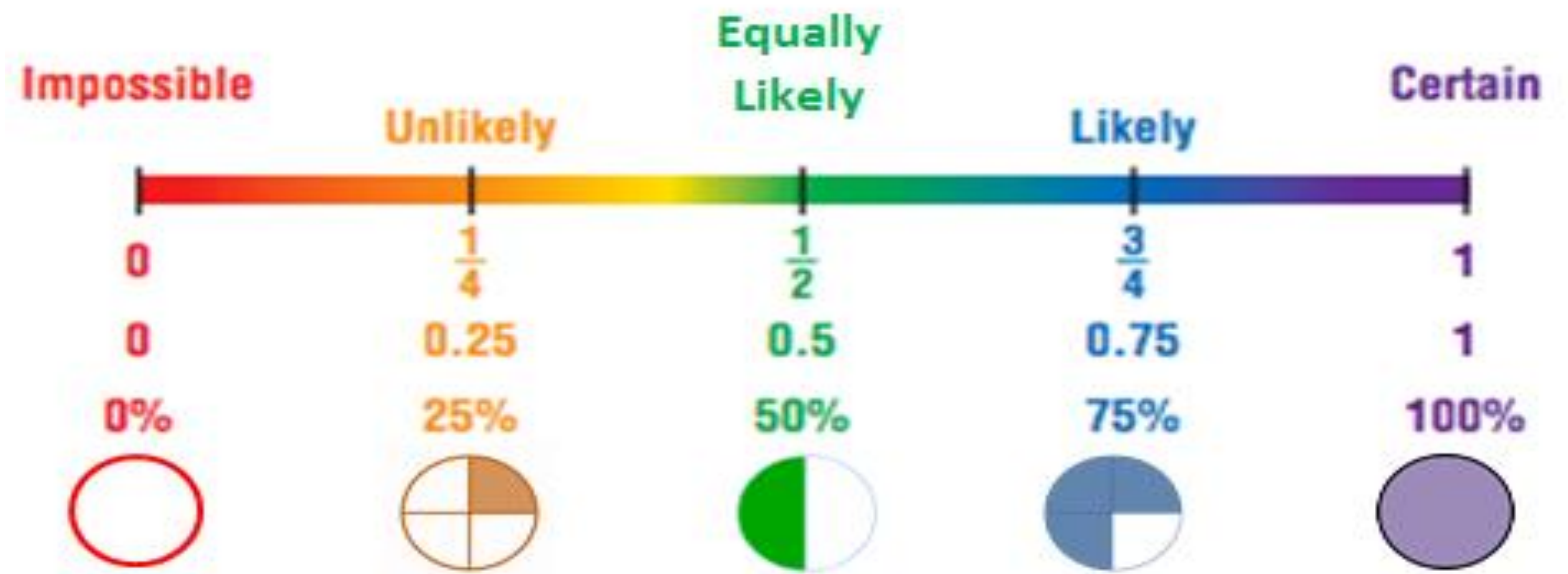
One way

All possibilities

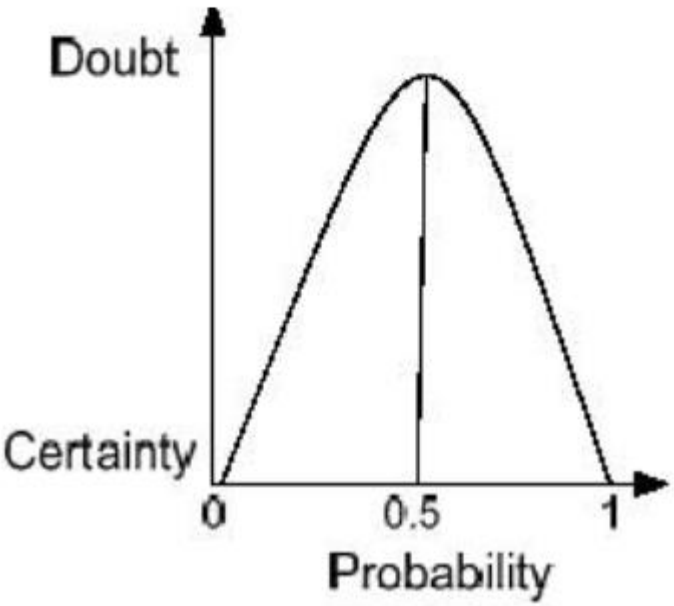
Random Variable is a real-valued function of the outcome of the Experiment

$$\text{Probability of event} = \frac{\text{Number of ways event can happen}}{\text{Total number of outcomes}}$$

Probability Line



Theorems related to Probability



The probability of an event can only be between 0 and 1 and can also be written as a percentage.

For Mutually Exclusive

$$P(A \text{ or } B) = P(A) + P(B)$$

$$P(A \cup B) = P(A) + P(B)$$

Non-Mutually Exclusive

$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$

$$P(A \cup B) = P(A) + P(B) - P(AB)$$

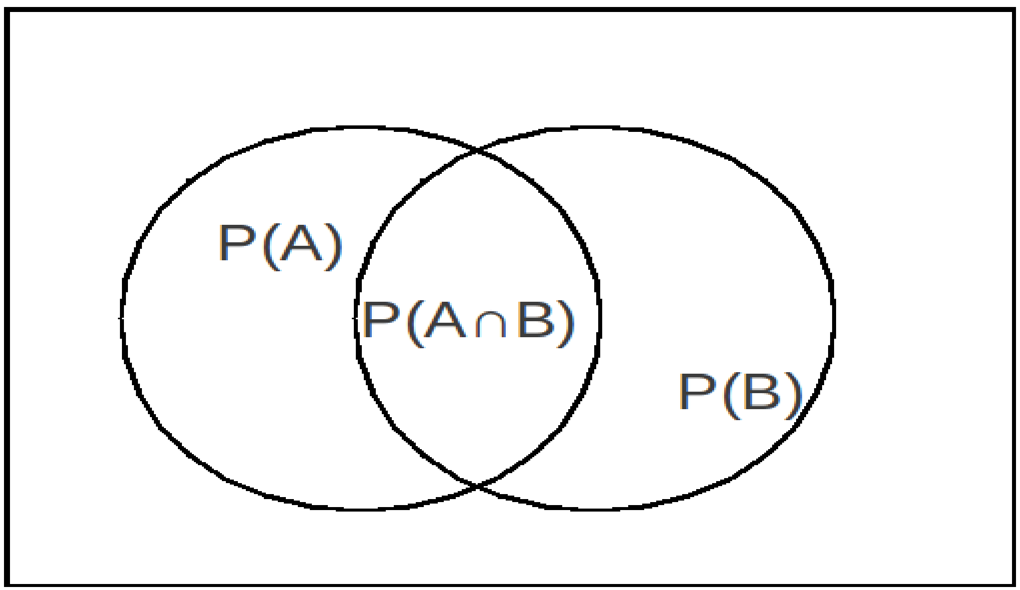
For Independent Events

$$P(A \text{ and } B) = P(A) \times P(B)$$

$$P(AB) = P(A) \times P(B)$$

Conditional Probability

A conditional probability is a probability that a certain event will occur given some knowledge about the outcome or some other event.



A Trying to Find ,B is Given

P(A|B)

Probability of A given B

Conditional Probability

- How likely is that a person has a disease given that a medical test was negative.
- Buy new car |Started New Job.
- We toss a fair coin three successive time . we wish to find probability P(A|B) when A and B are the event

  A={more heads than tails come up),

  B={1st toss is head}

Formula for Conditional Probability

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B/A) = \frac{P(A \cap B)}{P(A)}$$

$$P\left(\frac{A}{B}\right) = \frac{\text{P(A and B)}}{P(B)} = \frac{\text{Probability of the occurence of both A and B}}{\text{Probability of B}}$$

In a sample of 40 vehicles, 18 are red, 6 are trucks, and 2 are both. Suppose that a randomly selected vehicle is red. What is the probability it is a truck?

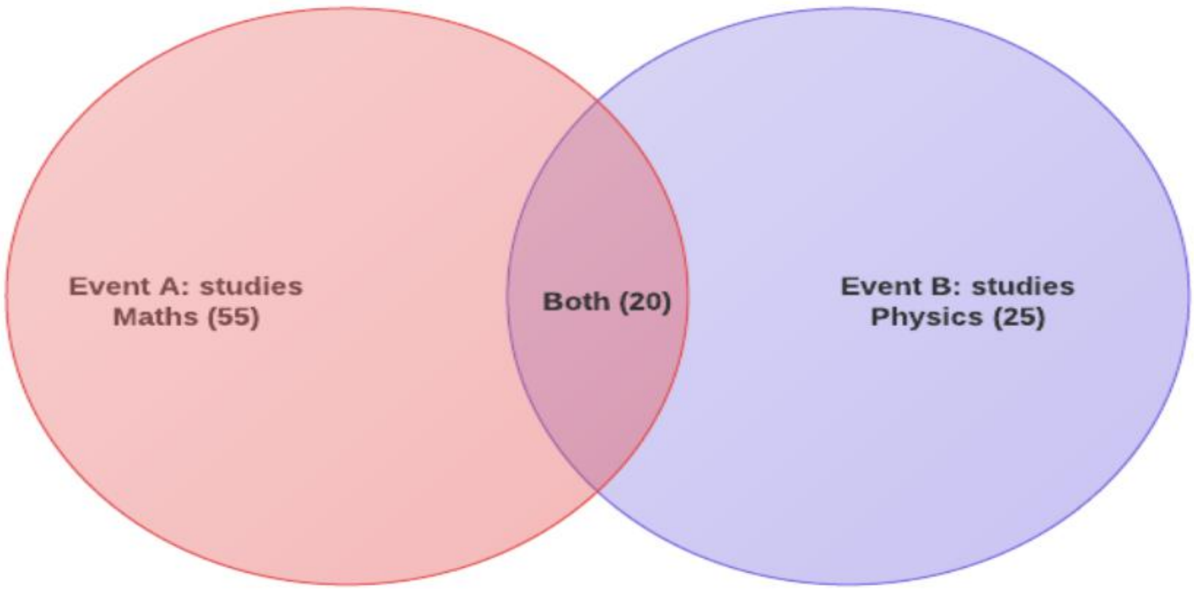
$$\mathrm{P}(\mathrm{truck}|\mathrm{red}) = \frac{\mathrm{P}(\mathrm{truck\ and\ red})}{\mathrm{P}(\mathrm{red})}$$

A fair die is rolled.

1. Find the probability that the number rolled is a five, given that it is odd.
2. Find the probability that the number rolled is odd, given that it is a five.

There are two urns containing colored balls. The first urn contains 50 red balls and 50 blue balls. The second urn contains 30 red balls and 70 blue balls. One of the two urns is randomly chosen (both urns have probability of being chosen) and then a ball is drawn at random from one of the two urns. If a red ball is drawn, what is the probability that it comes from the first urn?

There are two urns containing colored balls. The first urn contains 50 red balls and 50 blue balls. The second urn contains 30 red balls and 70 blue balls. One of the two urns is randomly chosen (both urns have probability of being chosen) and then a ball is drawn at random from one of the two urns. If a red ball is drawn, what is the probability that it comes from the first urn?



what's the probability that a student picked at random studies math GIVEN that we know he studies physics?

$P(A|B) = P(A \cap B) / P(B)$

$P(A|B) = 20 / 25$

$P(A|B) = 0.8$

$P(A|B)$ = probability that student studies math GIVEN he studies physics.

$P(B|A)$ = probability that student studies physics GIVEN he studies math.

Now we just calculated $P(A|B)$. What if from that resulted we wanted to calculate the inverse probability, that is $P(B|A)$?

That's where Bayes' Theorem comes into place.

Bayes' Theorem

The Bayes theorem describes the probability of an event based on the prior knowledge of the conditions that might be related to the event.

If we know the conditional probability $P(A|B)$, we can use the bayes rule to find out the reverse probabilities $P(B|A)$.

Bayes' Theorem

Bayes' Theorem is stated as Probability of the event B is equal to the Probability of the event A given B multiplied by the probability of A upon probability of B.

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

Bayes' Theorem

Given a Hypothesis **(H)** and evidence **(E)**, Bayes' Theorem states that the relationship between the probability of the hypothesis before getting the evidence, **P(H)**, and the probability of the hypothesis after getting the evidence, **P(H|E)**, is

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

- **P(H)** is called the **prior probability**
- **P(H|E)** is called the **posterior probability**
- **P(H|E)/P(E)**, is called the **likelihood ratio**

Bayes' Theorem

P(A|B): the conditional probability that event A occurs , given that B has occurred. This is also known as the posterior probability.

P(A) and P(B) : *probability of A and B without regard of each other*

P(B|A) : the conditional probability that event B occurs , given that A has occurred.

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

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

Prior probability

Likelihood

Posterior probability

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Bayes' Theorem Example

suppose we have a Deck of Cards and we wish to find out the probability of the card we picked at random to being a king, given that it is a face card.

- **P(King)** which is **4/52** as there are 4 Kings in a Deck of Cards.
- **P(Face|King)** is equal to **1** as all the Kings are face Cards.
- **P(Face)** is equal to **12/52** as there are 3 Face Cards in a Suit of 13 cards and there are 4 Suits in total.

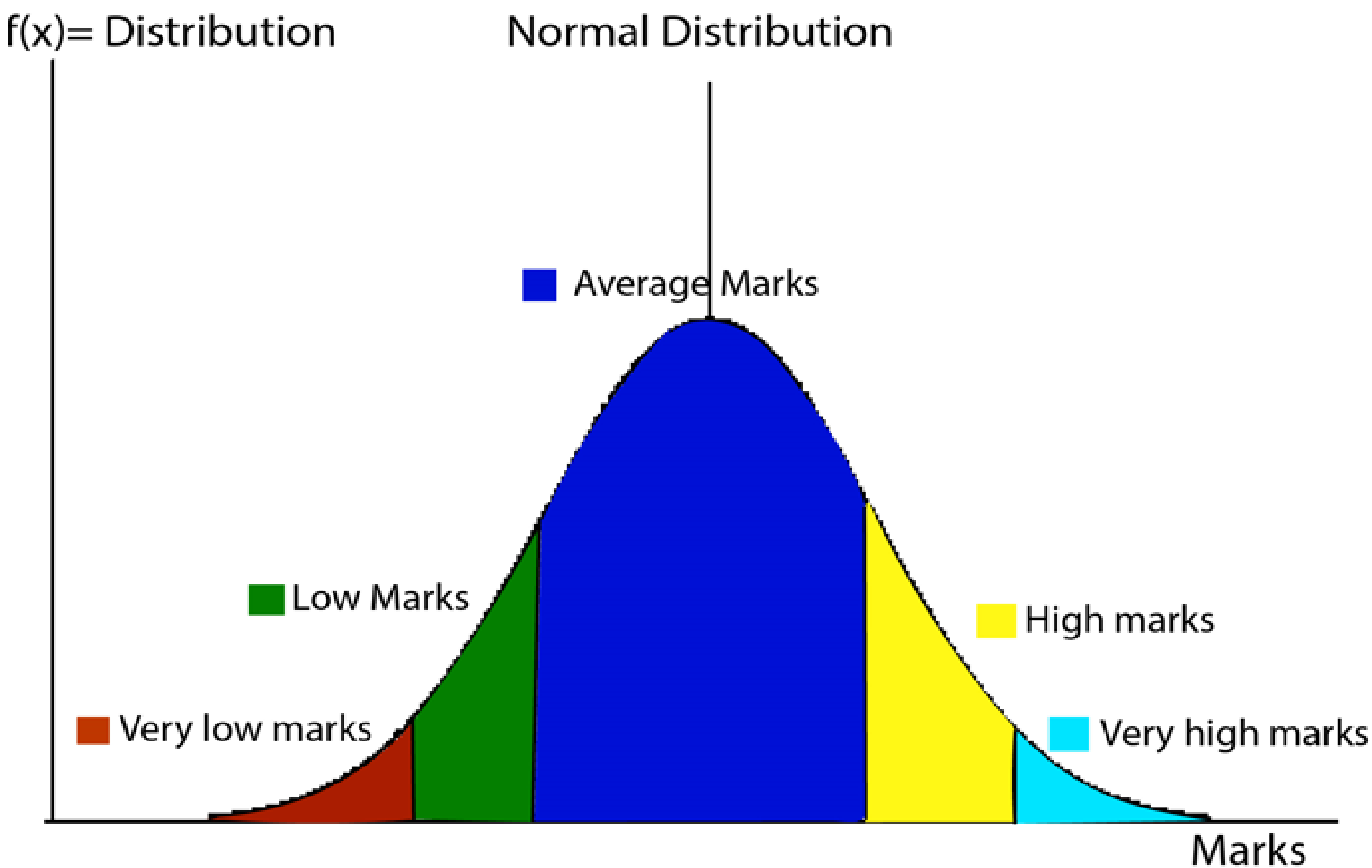
Types of Naive Bayes Algorithm

Gaussian: It's specifically used when the features have **continuous values.**

Multinomial: It is used for **discrete Feature.**

Bernoulli: The binomial model is useful if your **feature vectors are binary** (i.e. zeros and ones)

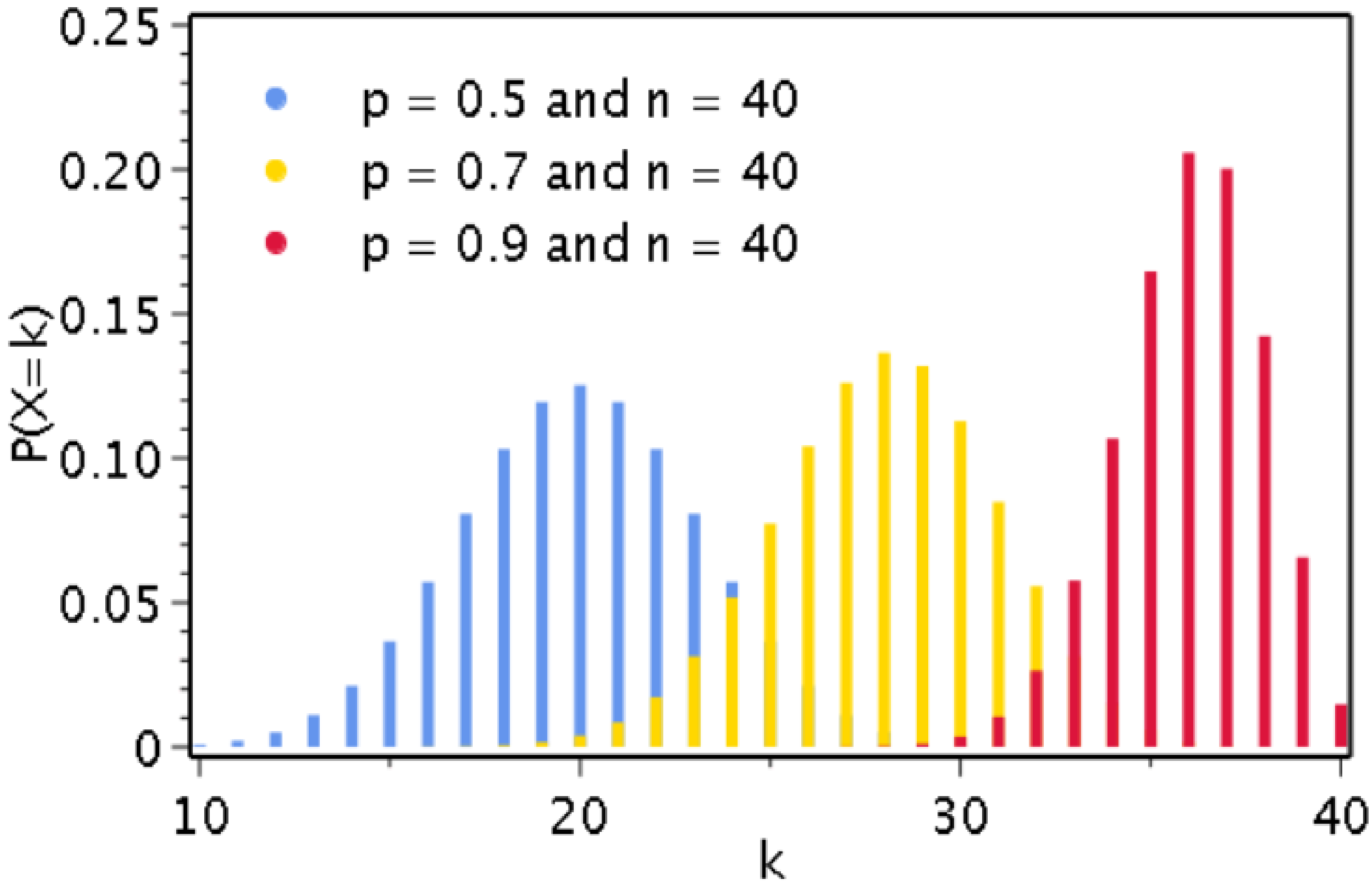
Gaussian Distribution



Binomial Distribution

The binomial distribution is a common discrete distribution used in statistics, as opposed to a continuous distribution such as the normal distribution. This is because the binomial distribution only counts two states, typically represented as 1 (for a success) or 0 (for a failure) given a number of trials in the data.

$$P = \binom{N}{x} p^x (1-p)^{N-x}$$



p = 0.5 and n = 40
p = 0.7 and n = 40
p = 0.9 and n = 40
P(X=k)
0.25
0.20
0.15
0.10
0.05
0
10
20
30
40
k

Applications

Text classification: It is used as a probabilistic learning method for text classification. The Naive Bayes classifier is one of the most successful known algorithms when it comes to the classification of text documents, i.e., whether a text document belongs to one or more categories (classes).

Spam filtration: It is an example of text classification.

This has become a popular mechanism to distinguish spam email from legitimate email.

Sentiment Analysis: It can be used to analyze the tone of tweets, comments, and reviews—whether they are negative, positive or neutral.

Data_Set

1	Fruit	Sweetness	Soureness	Fruit Type
2	Lemon	1	9	Sour
3	Grapfruit	2	8	Sour
4	Orange	3	7	Sour
5	Raspberry	2	8	Sour
6	Cherry	6	4	Sweet
7	Banana	9	1	Sweet
8	Grapes	8	2	Sweet
9	Watermelon	9	1	Sweet
10	Avacado	1	1	None
11	Strawberry	5	5	Sour

Data_Set

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	FALSE	No
1	Rainy	Hot	High	TRUE	No
2	Overcast	Hot	High	FALSE	Yes
3	Sunny	Mild	High	FALSE	Yes
4	Sunny	Cool	Normal	FALSE	Yes
5	Sunny	Cool	Normal	TRUE	No
6	Overcast	Cool	Normal	TRUE	Yes
7	Rainy	Mild	High	FALSE	No
8	Rainy	Cool	Normal	FALSE	Yes
9	Sunny	Mild	Normal	FALSE	Yes
10	Rainy	Mild	Normal	TRUE	Yes
11	Overcast	Mild	High	TRUE	Yes
12	Overcast	Hot	Normal	FALSE	Yes
13	Sunny	Mild	High	TRUE	No

Feature: ‘Outlook’, ‘Temperature’, ‘Humidity’ and ‘Windy’.

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

•Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
•P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
•P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

X = (Rainy, Hot, High, False)
y = No

if any two events A and B are independent, then,
P(A,B) = P(A)P(B)

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)...P(x_n)}$$

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

we find the probability of given set of inputs for all possible values of the class variable *y* and pick up the output with maximum probability. This can be expressed mathematically as:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

P(y) is also called **class probability** and $P(x_i \mid y)$ is called **conditional probability**.
The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$.

```
Windy_df
```

	Yes	No	P(Yes)	P(No)
TRUE	3	3	0.333333	0.6
FALSE	6	2	0.666667	0.4

```
Humidity_df
```

	Yes	No	P(Yes)	P(No)
High	3	4	0.333333	0.8
Low	6	1	0.666667	0.2

```
Outlook_df
```

	Yes	No	P(Y)	P(N)	P(YES)	P(NO)
Sunny	3	2	0.333333	0.4	0.333333	0.4
Overscast	4	0	0.444444	0.0	0.444444	0.0
Rainy	2	3	0.222222	0.6	0.222222	0.6

```
Temperature_df
```

	Yes	No	P(YES)	P(N0)
Mild	4	2	0.444444	0.4
Cool	3	1	0.333333	0.2
Hot	2	2	0.222222	0.4

```
Glofplayclass
```

	Play	P(Yes/No)
Yes	9	0.642857
No	5	0.357143

Feature

#Outlook Feature

Ol_Sunny=my_data[my_data["Outlook"]=="Sunny"]

Ol_Overcast=my_data[my_data["Outlook"]=="Overcast"]

Ol_Rainy=my_data[my_data["Outlook"]=="Rainy"]

#Temperature Feature

Tem_Mild=my_data[my_data["Temperature"]=="Mild"]

Tem_Cool=my_data[my_data["Temperature"]=="Cool"]

Tem_Hot=my_data[my_data["Temperature"]=="Hot"]

#Humidity Feature

Hum_High=my_data[my_data["Humidity"]=="High"]

Hum_Low=my_data[my_data["Humidity"]==“Normal"]

#Wind Feature

Windy_False=my_data[my_data["Windy"]==False]

Windy_True=my_data[my_data["Windy"]==True]

```
#Overcast Feature
#Total yes and no for sunny
Osunny_yes=len(Ol_sunny[Ol_Sunny["Play Golf"]=="Yes"])
Osunny_no=len(Ol_sunny[Ol_Sunny["Play Golf"]=="No"])
#Total Yes and no For Overcast
Ocast_yes=len(Ol_Overcast[Ol_Overcast["Play Golf"]=="Yes"])
Ocast_no=len(Ol_Overcast[Ol_Overcast["Play Golf"]=="No"])
#Total Yes and no for Rainy
Orainy_yes=len(Ol_Rainy[Ol_Rainy["Play Golf"]=="Yes"])
Orainy_no=len(Ol_Rainy[Ol_Rainy["Play Golf"]=="No"])
#Total yes and No in Overcast Feature
Ol_yes=[Osunny_yes,Ocast_yes,Orainy_yes]
Ol_no=[Osunny_no,Ocast_no,Orainy_no]
```

```python
#Temperature Feature
#Yes and No  ,For Mild
TMild_yes=len(Tem_Mild[Tem_Mild["Play Golf"]=="Yes"])
TMild_no=len(Tem_Mild[Tem_Mild["Play Golf"]=="No"])
# Yes and No,For Cool
Tcool_yes=len(Tem_Cool[Tem_Cool["Play Golf"]=="Yes"])
Tcool_no=len(Tem_Cool[Tem_Cool["Play Golf"]=="No"])
#Yes and No,For Hot
Thot_yes=len(Tem_Hot[Tem_Hot["Play Golf"]=="Yes"])
Thot_no=len(Tem_Hot[Tem_Hot["Play Golf"]=="No"])
#Total number of yes and no in Temperature Feature
Tem_yes=[TMild_yes,Tcool_yes,Thot_yes]
Tem_No=[TMild_no,Tcool_no,Thot_no]
```

```
#Humidty Feature
#yes and no for High
Humhigh_yes=len(Hum_High[Hum_High["Play Golf"]=="Yes"])
Humhigh_no=len(Hum_High[Hum_High["Play Golf"]=="No"])
#yes and no for low
Humlow_yes=len(Hum_Low[Hum_Low["Play Golf"]=="Yes"])
Humlow_no=len(Hum_Low[Hum_Low["Play Golf"]=="No"])
#Total Number of yes and no
Hum_yes=[Humhigh_yes,Humlow_yes]
Hum_no=[Humhigh_no,Humlow_no]
```

```
#Windy Feature
#Yes and No for True
WindyTrue_Yes=len(Windy_True[Windy_True["Play Golf"]=="Yes"])
WindyTrue_No=len(Windy_True[Windy_True["Play Golf"]=="No"])
#Yes and No for False
WindyFalse_Yes=len(Windy_False[Windy_False["Play Golf"]=="Yes"])
WindyFalse_No=len(Windy_False[Windy_False["Play Golf"]=="No"])
#Toal number of Yes and no for for Windy
Windy_Yes=[WindyTrue_Yes,WindyFalse_Yes]
Windy_No=[WindyTrue_No,WindyFalse_No]
```

```
#Data frame for all feature
Outlook_df=pd.DataFrame({"Yes":Ol_yes,"No":Ol_no},index=["Sunny","Overscast","Rainy"])
Temperature_df=pd.DataFrame({"Yes":Tem_yes,"No":Tem_No},index=["Mild","Cool","Hot"])
Humidity_df=pd.DataFrame({"Yes":Hum_yes,"No":Hum_no},index=["High","Low"])
Windy_df=pd.DataFrame({"Yes":Windy_Yes,"No":Windy_No},index=["TRUE","FALSE"])
```

```
#Probability of each Yes and NO for all feature
#Outlook Feature probability
Outlook_df["P(YES)"]=[3/9,4/9,2/9]
Outlook_df["P(NO)"]=[2/5,0/5,3/5]
#Temperature Feature Probability
Temperature_df["P(YES)"]=[4/9,3/9,2/9]
Temperature_df["P(N0)"]=[2/5,1/5,2/5]
#Humidity Feature Probability
Humidity_df["P(Yes)"]=[3/9,6/9]
Humidity_df["P(No)"]=[4/5,1/5]
#Wind Feature Probability
Windy_df["P(Yes)"]=[3/9,6/9]
Windy_df["P(No)"]=[3/5,2/5]
```

Yclass

#Total Number of Paly Golf "Yes and No"

```
GolfPlay_Yes=my_data[my_data["Play Golf"]=="Yes"]
GolfPlay_No=my_data[my_data["Play Golf"]=="No"]
Num_play=len(GolfPlay_Yes)
Num_Noplay=len(GolfPlay_No)
Yclass=[Num_play,Num_Noplay]
```

#DataFrame of GolfPlay Yes and No Class

```
Glofplayclass=pd.DataFrame({"Play":Yclass},index=["Yes","No"])
#Probability of yes and no
Glofplayclass["P(Yes/No)"]=[9/14,5/14]
```

```
Windy_df
```

	Yes	No	P(Yes)	P(No)
TRUE	3	3	0.333333	0.6
FALSE	6	2	0.666667	0.4

```
Humidity_df
```

	Yes	No	P(Yes)	P(No)
High	3	4	0.333333	0.8
Low	6	1	0.666667	0.2

```
Outlook_df
```

	Yes	No	P(Y)	P(N)	P(YES)	P(NO)
Sunny	3	2	0.333333	0.4	0.333333	0.4
Overscast	4	0	0.444444	0.0	0.444444	0.0
Rainy	2	3	0.222222	0.6	0.222222	0.6

```
Temperature_df
```

	Yes	No	P(YES)	P(N0)
Mild	4	2	0.444444	0.4
Cool	3	1	0.333333	0.2
Hot	2	2	0.222222	0.4

Probability of playing golf given that the temperature is cool.
P(temp. = cool | play golf = Yes) = ??.

```
Glofplayclass
```

	Play	P(Yes/No)
Yes	9	0.642857
No	5	0.357143

So now, we are done with our pre-computations and the classifier is ready!Let us test it on a new set of features (let us call it today)

today = (Sunny, Hot, Low, False)

$$P(Yes|today) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(today)}$$

and probability to not play golf is given by:

$$P(No|today) = \frac{P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)}{P(today)}$$

$$P(Yes|today) > P(No|today)$$

email-spam-detection

Data Set

https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

SMS Spam Collection Data Set

Download: Data Folder, Data Set Description

Abstract: The SMS Spam Collection is a public set of SMS labeled messages that have been collected for mobile phone spam research.

Data Set Characteristics:	Multivariate, Text, Domain-Theory	**Number of Instances:**	5574	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	N/A	**Date Donated**	2012-06-22
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	242742

	Class	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
mydata=pd.read_table("C:/Users/SHUBH_RAM/Desktop/Tool_PPT_019/Ml_Data/Data_2019/SMSSpamCollection",
                     sep='\t',header=None,names=["Class","Message"])
```

```python
mydata['Class'] = mydata.Class.map({'ham':0, 'spam':1})
print(mydata.shape)
```

```
(5572, 2)
```

```python
mydata.head()
```

	Class	Message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Bag of words

```
['Hello, how are you!',
'Win money, win from home.',
'Call me now',
'Hello, Call you tomorrow?']
```

	are	call	from	hello	home	how	me	money	now	tomorrow	win	you
0	1	0	0	1	0	1	0	0	0	0	0	1
1	0	0	1	0	1	0	0	1	0	0	2	0
2	0	1	0	0	0	0	1	0	1	0	0	0
3	0	1	0	1	0	0	0	0	0	1	0	1

convert this set of text to a frequency distribution matrix

Implementing Bag of Words from scratch

- Convert all strings to their lower case form
- Removing all punctuations
- Tokenization
- Count frequencies

```
'''
Solution:
'''
documents = ['Hello, how are you!',
             'Win money, win from home.',
             'Call me now.',
             'Hello, Call hello you tomorrow?']

lower_case_documents = []
for i in documents:
    lower_case_documents.append(i.lower())
print(lower_case_documents)
```

```python
'''
Solution:
'''
sans_punctuation_documents = []
import string

for i in lower_case_documents:
    sans_punctuation_documents.append(i.translate(str.maketrans('','',string.punctuation)))

print(sans_punctuation_documents)
```

```python
'''
Solution:
'''
preprocessed_documents = []
for i in sans_punctuation_documents:
    preprocessed_documents.append(i.split(' '))
print(preprocessed_documents)
```

```python
'''
Solution
'''
frequency_list = []
import pprint
from collections import Counter

for i in preprocessed_documents:
    frequency_list.append(Counter(i))

pprint.pprint(frequency_list)
```