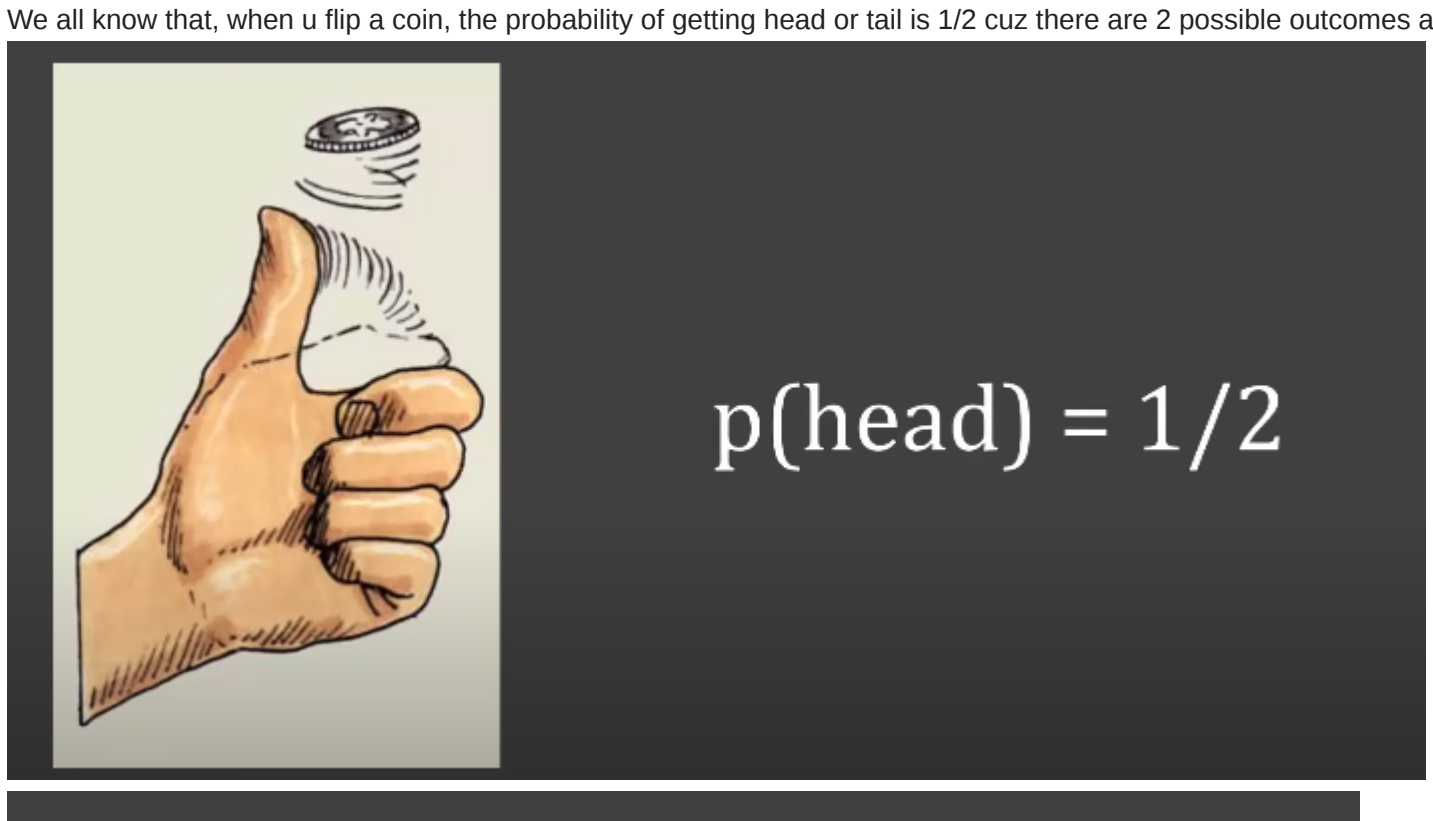
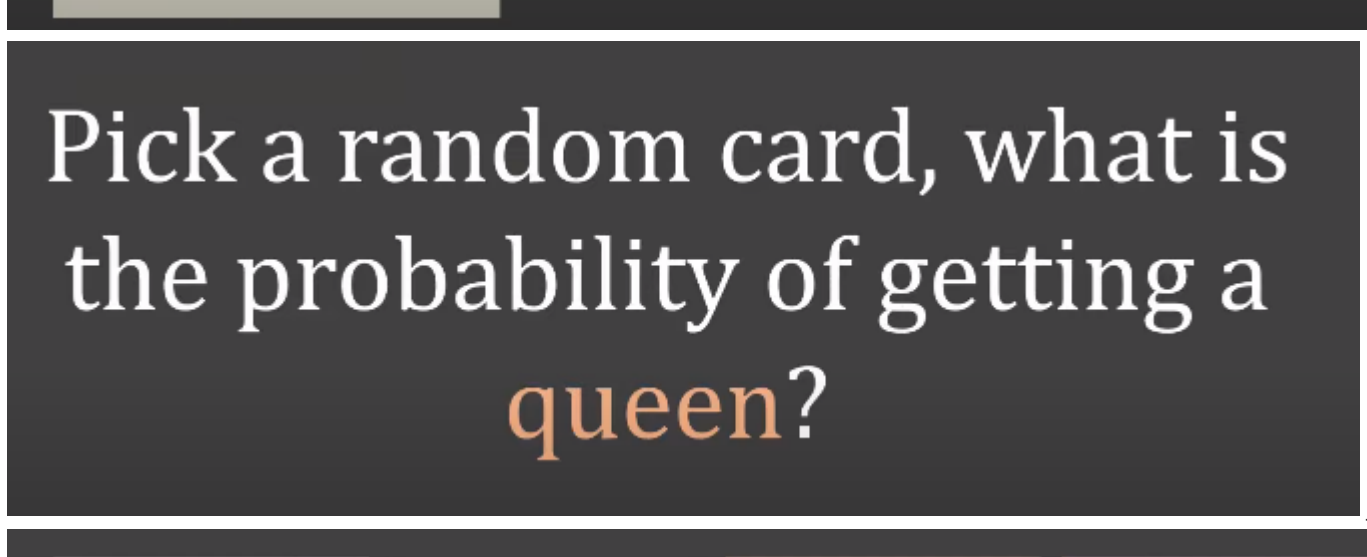
Naive Bayes part 1

Basics of Probability

We all know that, when u flip a coin, the probability of getting head or tail is 1/2 cuz there are 2 possible outcomes and the chance of getting head or tail is 50 50

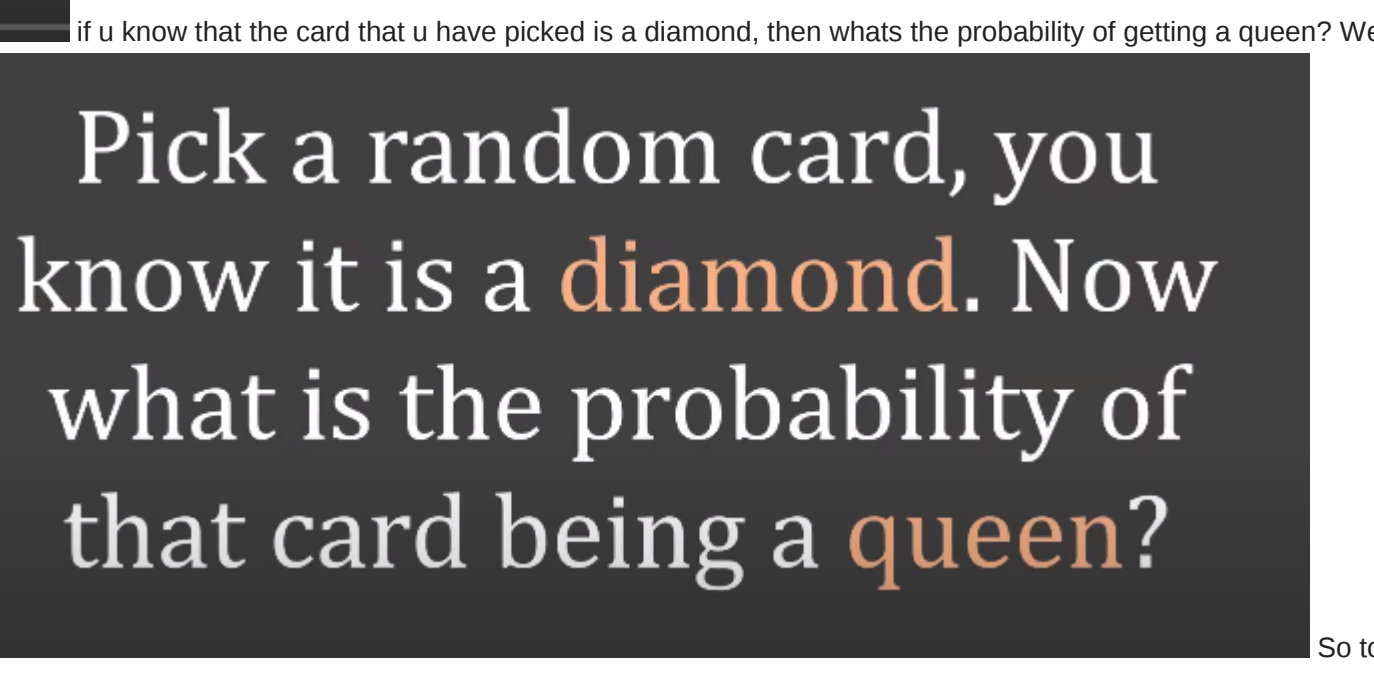


Similarly when u pick a random card, the probability of that being a queen is how much,



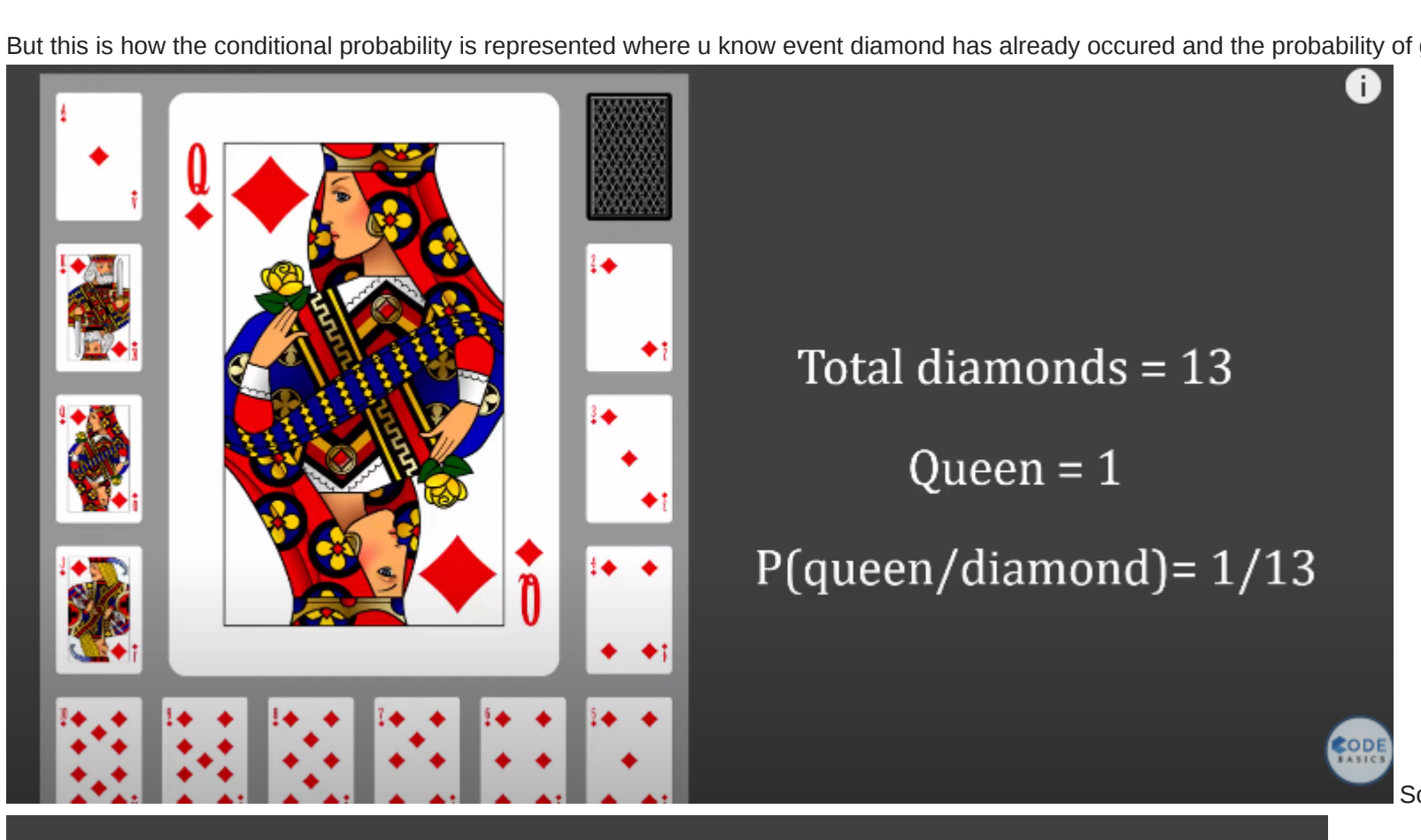
just think about it, its simple, there are total 4 queens and 52 total cards, hence the probability of getting a queen is 4/52 which is 1/13

if u know that the card that u have picked is a diamond, then whats the probability of getting a queen? Well this is called a **Conditional probability** where u know that event A has occured and now u are trying to predict the probaility of event B

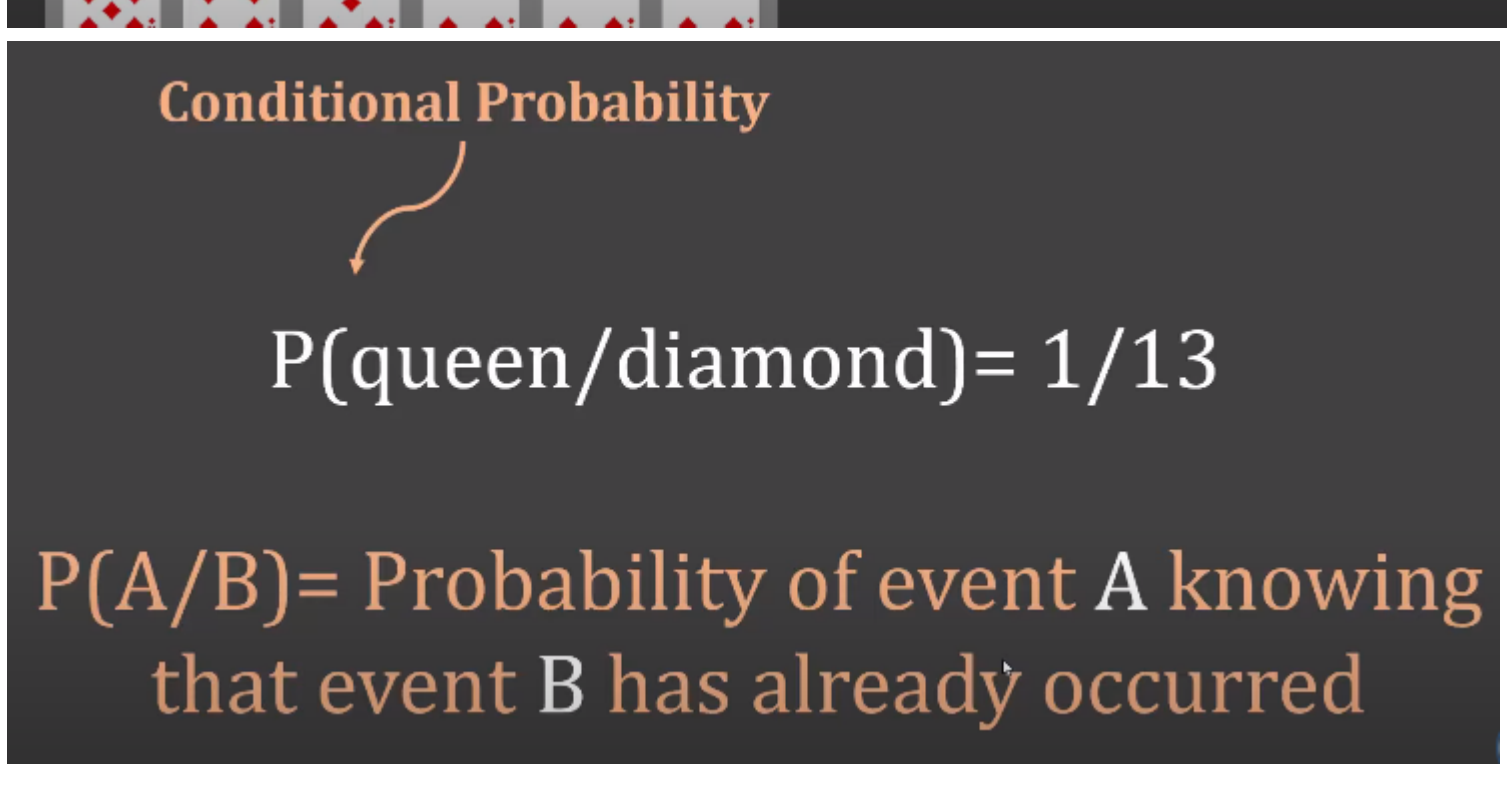


So total diamonds are 13 and queen is 1, we all know by just simple intuition that probability here us 1/13.

But this is how the conditional probability is represented where u know event diamond has already occured and the probability of gettting a queen is something u are calculating

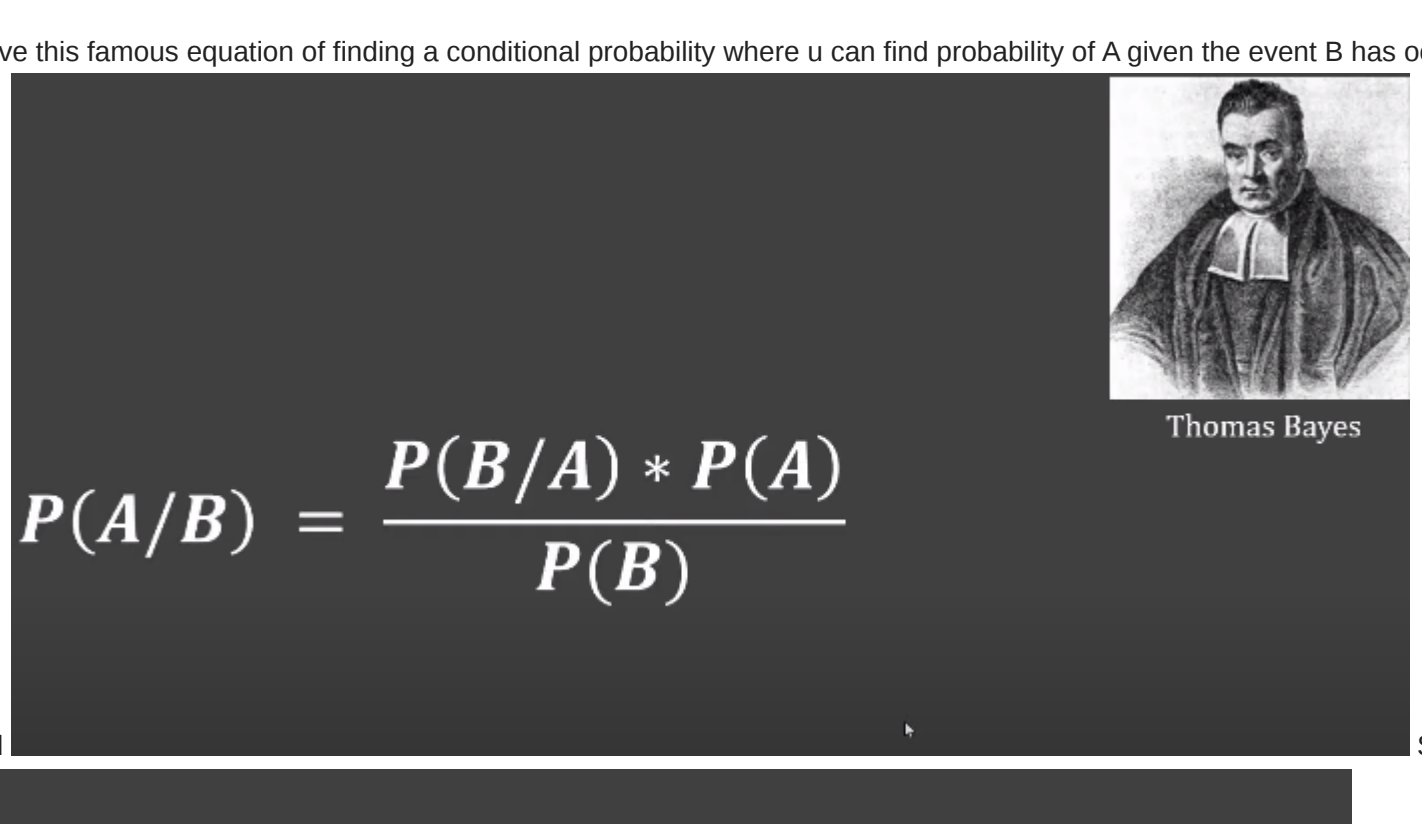


So thats called a Conditonal probability and the way that its represented is as below.

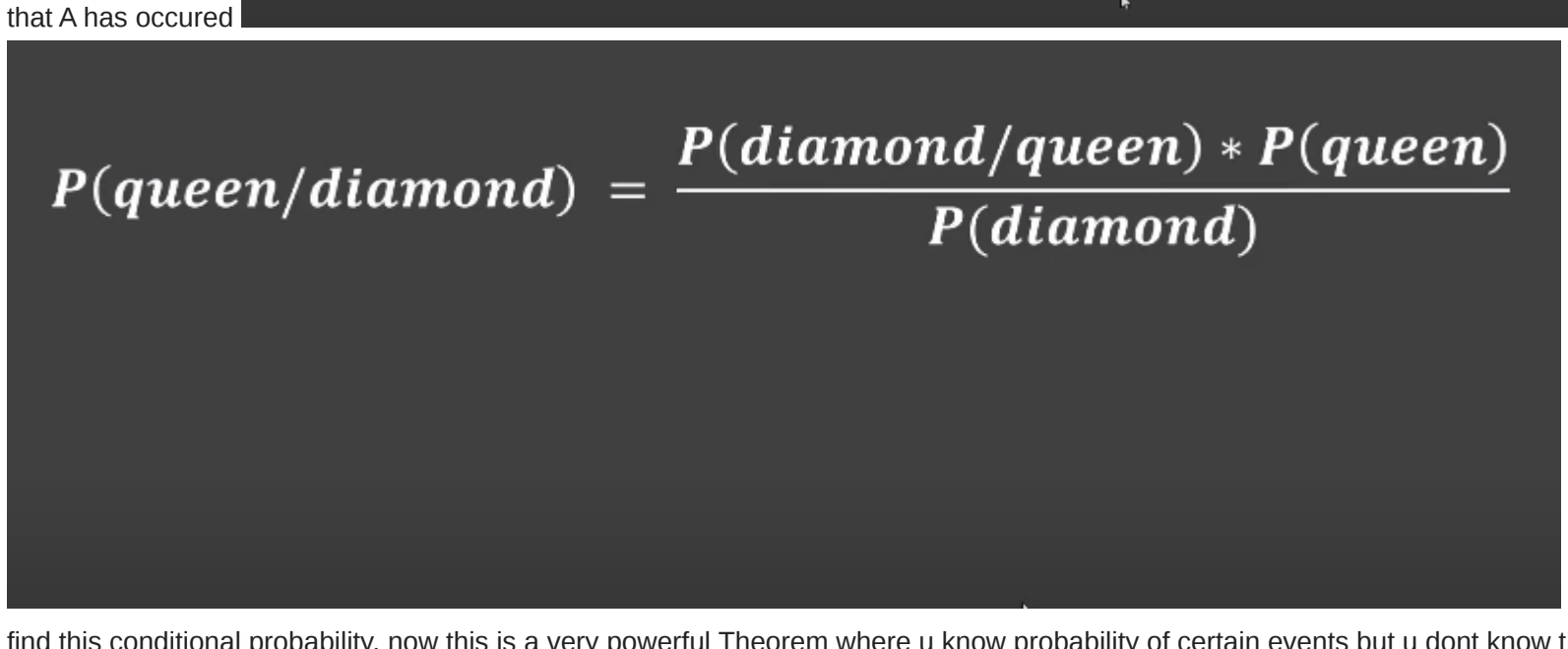


Bayes Theorem

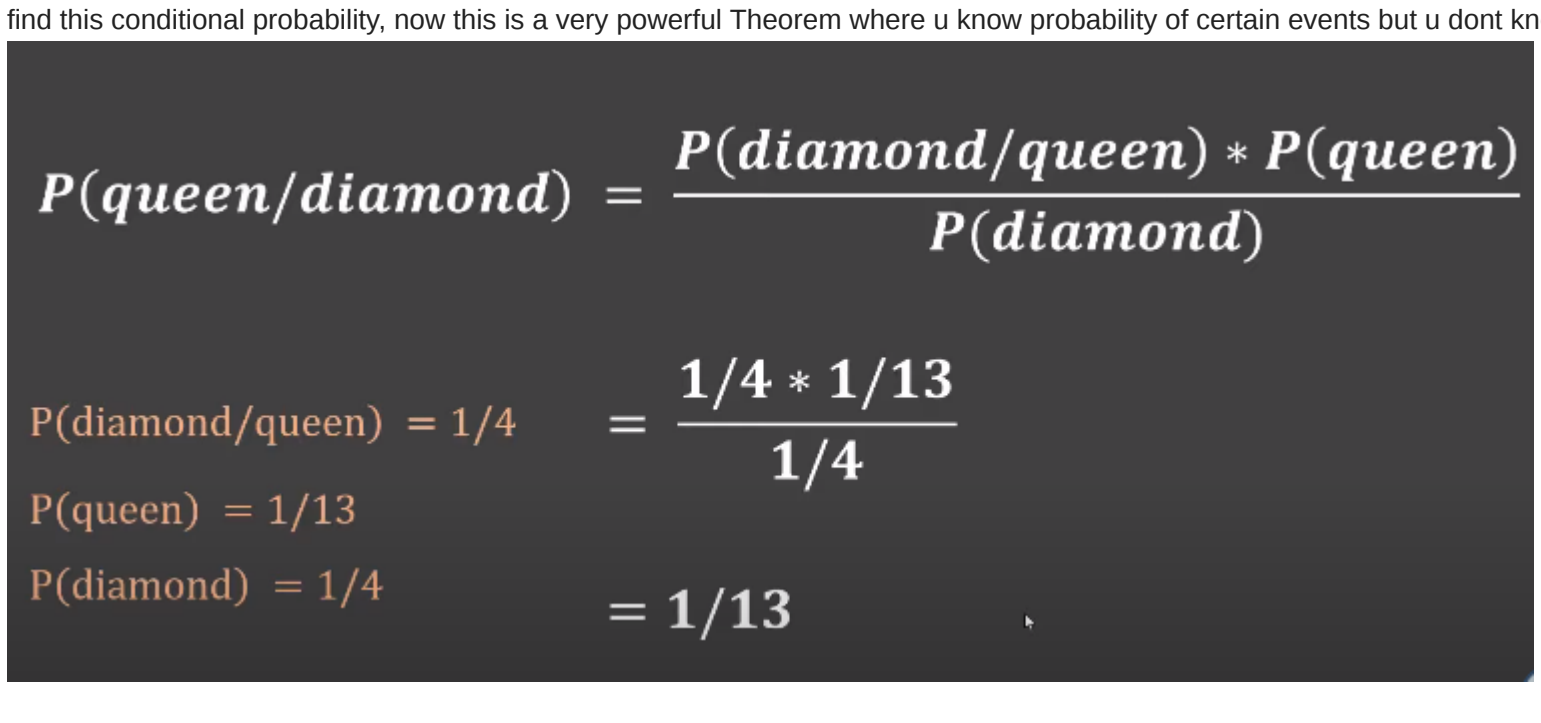
Thomas Bayes gave this famous equation of finding a conditional probability where u can find probability of A given the event B has occured by knowing some parameters which is the individual probabilities of A and B and knowing the probaility of B given that A has occured



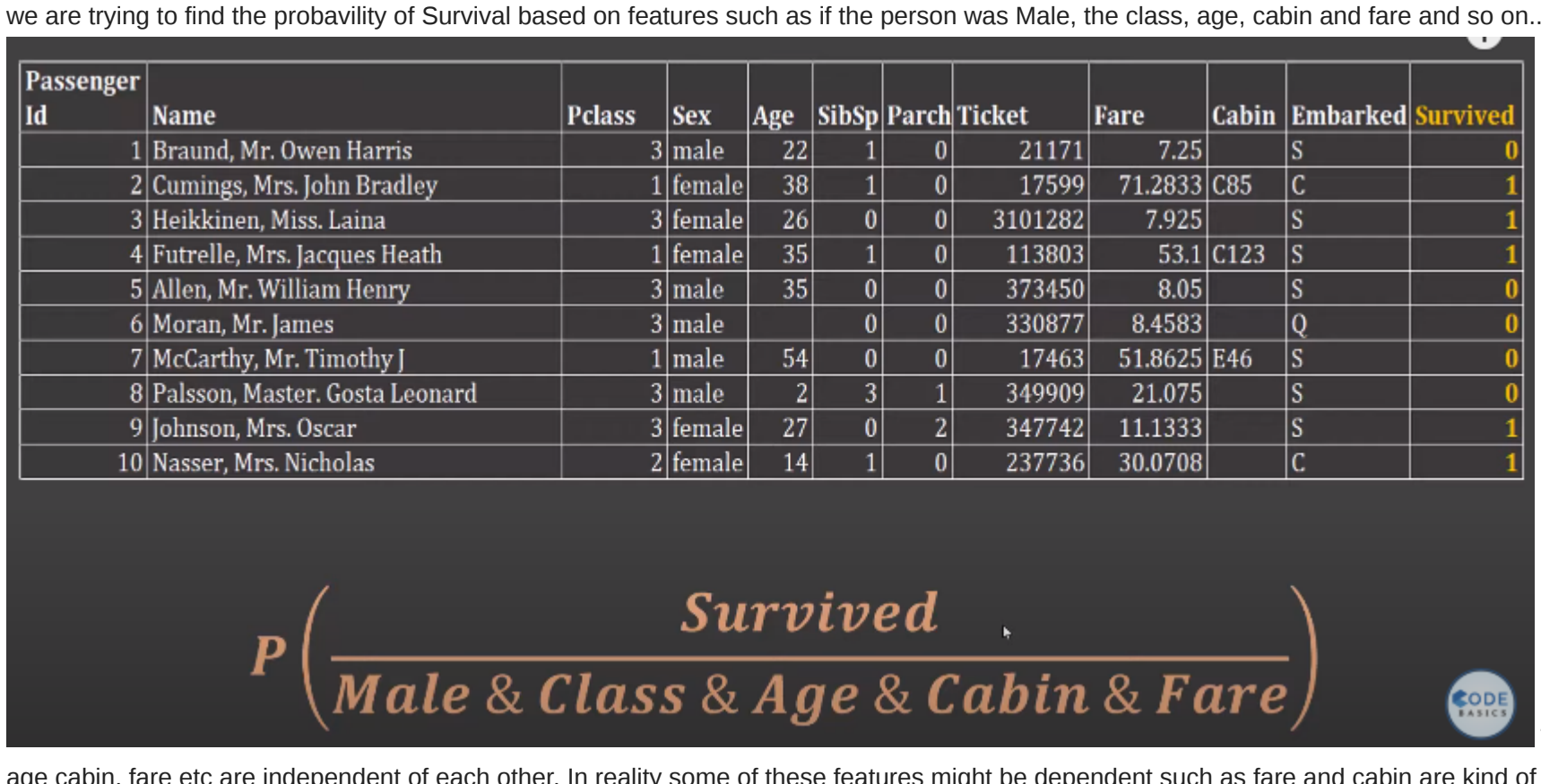
So lets look at it in the context of our queen and diamond problem



So here, this is the same equation we have represented for our specific use case. If u put them into our equation, u can easily find this conditional probability, now this is a very powerful Theorem where u know probability of certain events but u dont know the probability of some other events and using those certain events u can find other event probabilities

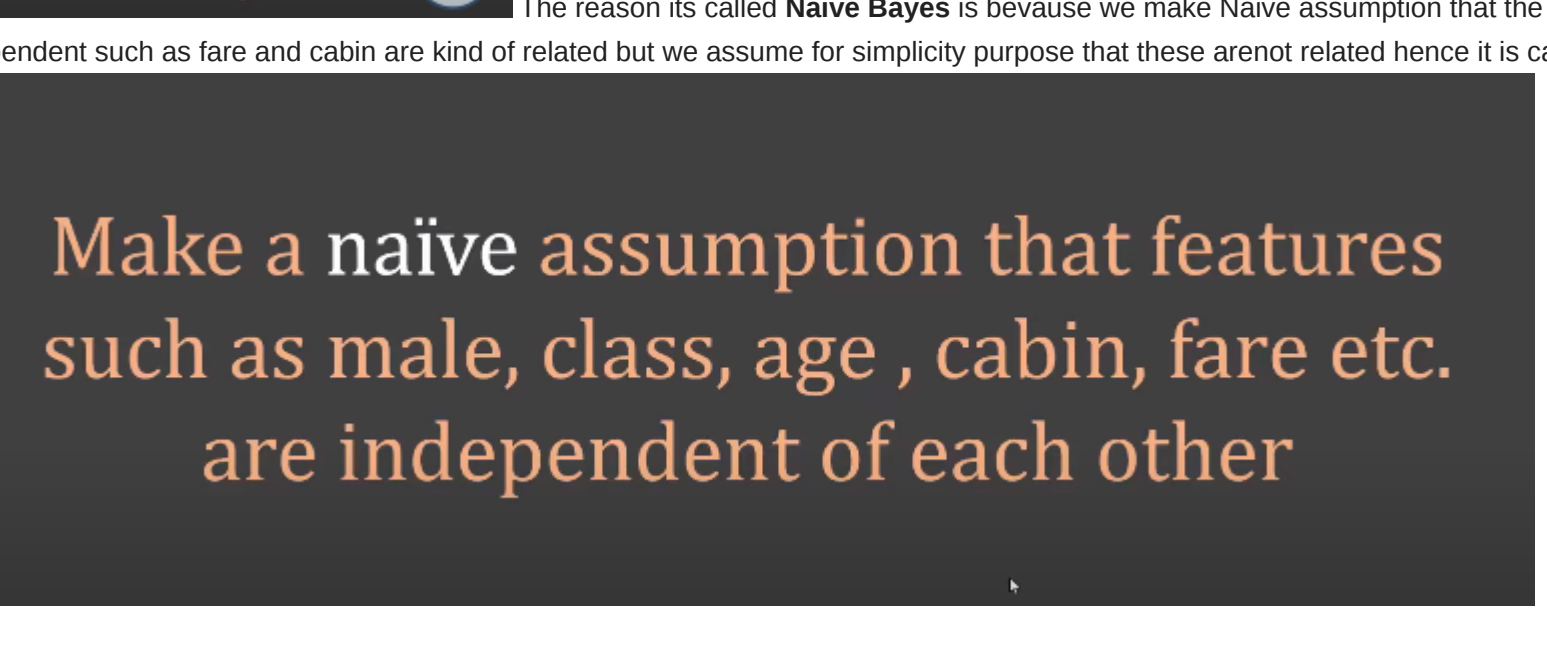


You all know about Titanic crash, we have a dataset of tis titanic crash where there are name of ppl along with certain features like Fare, ticket, Cabin, etc based on that we are trying to find out the Survival rate and here we can use **Bayes Theorem** where we are trying to find the probavility of Survival based on features such as if the person was Male, the class, age, cabin and fare and so on..

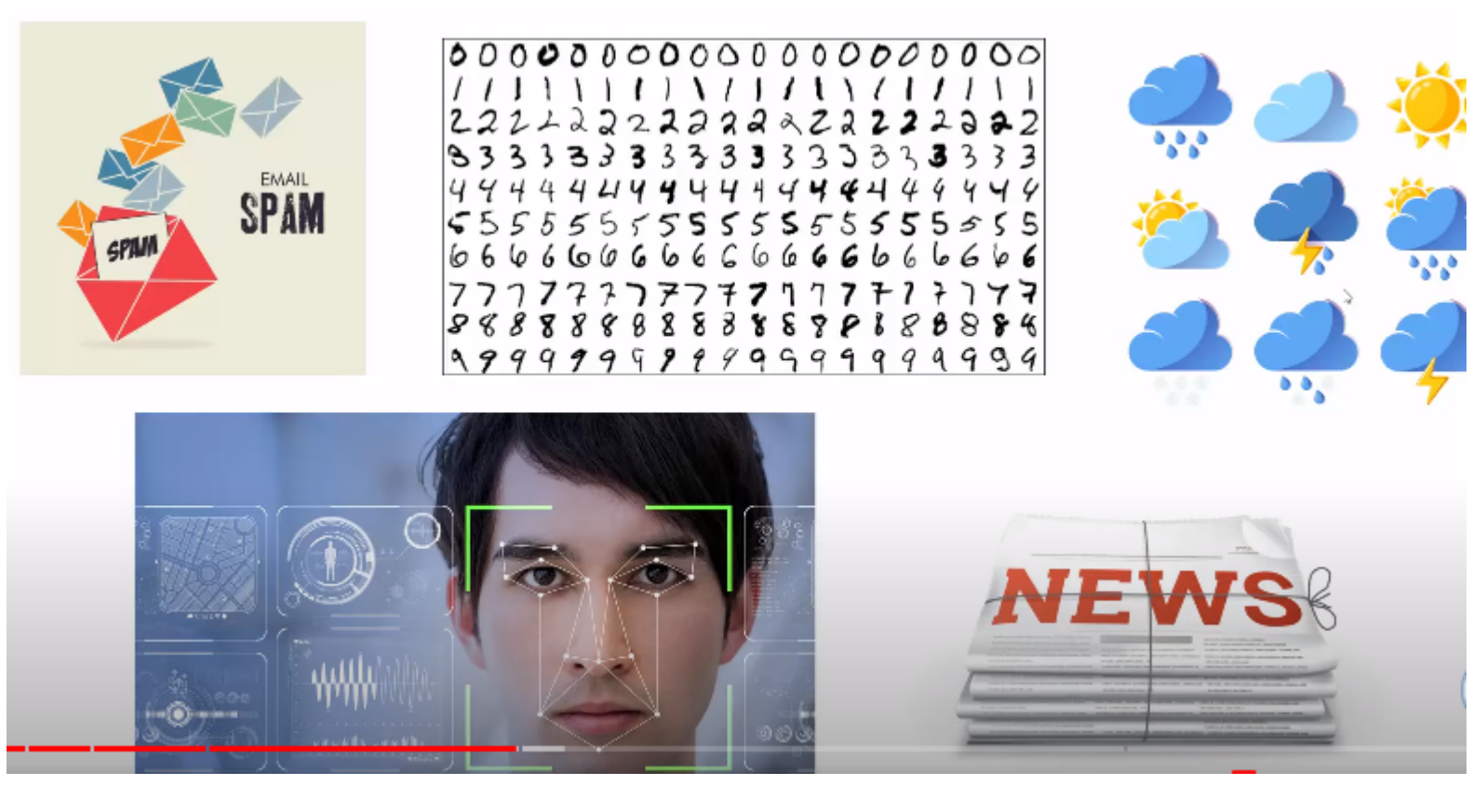


Passenger Id	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
1	Braund, Mr. Owen Harris	3	male	22	1	0	21171	7.25		S	0
2	Cumings, Mrs. John Bradley	1	female	38	1	0	17599	71.2833	C85	C	1
3	Heikkinen, Miss. Laina	3	female	26	0	0	3101282	7.925		S	1
4	Futrelle, Mrs. Jacques Heath	1	female	35	1	0	113803	53.1	C123	S	1
5	Allen, Mr. William Henry	3	male	35	0	0	373450	8.05		S	0
6	Moran, Mr. James	3	male		0	0	330877	8.4583		Q	0
7	McCarthy, Mr. Timothy J	1	male	54	0	0	17463	51.8625	E46	S	0
8	Palsson, Master. Gosta Leonard	3	male	2	3	1	349909	21.075		S	0
9	Johnson, Mrs. Oscar	3	female	27	0	2	347742	11.1333		S	1
10	Nasser, Mrs. Nicholas	2	female	14	1	0	237736	30.0708		C	1

The reason its called **Naive Bayes** is bevause we make Naive assumption that the features such as Male, class, age cabin, fare etc are independent of each other. In reality some of these features might be dependent such as fare and cabin are kind of related but we assume for simplicity purpose that these arenot related hence it is called **Naive Bayes** and it is a simple assumption which reduces our calculation and makes the Algorithm simple yet effective



Naive Bayes is used in Email Spam detection, hand digit character recognization, whether predictions, face detections and news article categorization



Coding Part

Download the titanic dataset from Kaggle

In [13]:
```python
import pandas as pd
```

In [14]:
```python
data = pd.read_csv('titanic2.csv')
data.head()
```

Out[14]:

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
0	1	Braund, Mr. Owen Harris	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	female	38.0	1	0	PC 17599	71.2833	C85	C	1
2	3	Heikkinen, Miss. Laina	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	1	0	113803	53.1000	C123	S	1
4	5	Allen, Mr. William Henry	3	male	35.0	0	0	373450	8.0500	NaN	S	0

In [15]:
```python
data.shape
```

Out[15]: (891, 12)

So we can see that some vairables doesnt haave any impact and it doesnt matter if we drop them, for example Name, it doesnt have any impact, so we will drop those variables which doesnt seem to be imprttant

In [16]:
```python
data.drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis='columns'
          ,inplace=True)
data.head()
```

Out[16]:

	Pclass	Sex	Age	Fare	Survived
0	3	male	22.0	7.2500	0
1	1	female	38.0	71.2833	1
2	3	female	26.0	7.9250	1
3	1	female	35.0	53.1000	1
4	3	male	35.0	8.0500	0

In [19]:
```python
X = data.drop(['Survived'], axis=1)
y = data['Survived']
```

Now we will turn the 'Sex' column into a numerical value using Dummy variable

In [22]:
```python
dummies = pd.get_dummies(data.Sex).astype(int)
dummies.head()
```

Out[22]:

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1

In [23]:
```python
X = pd.concat([X, dummies], axis=1)
X
```

Out[23]:

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	0	1
1	1	female	38.0	71.2833	1	0
2	3	female	26.0	7.9250	1	0
3	1	female	35.0	53.1000	1	0
4	3	male	35.0	8.0500	0	1
...	...	...	...	...	...	...
886	2	male	27.0	13.0000	0	1
887	1	female	19.0	30.0000	1	0
888	3	female	NaN	23.4500	1	0
889	1	male	26.0	30.0000	0	1
890	3	male	32.0	7.7500	0	1

891 rows × 6 columns

In [26]:
```python
X.drop(['Sex'], axis=1, inplace=True)
X
```

Out[26]:

	Pclass	Age	Fare	female	male
0	3	22.0	7.2500	0	1
1	1	38.0	71.2833	1	0
2	3	26.0	7.9250	1	0
3	1	35.0	53.1000	1	0
4	3	35.0	8.0500	0	1
...	...	...	...	...	...
886	2	27.0	13.0000	0	1
887	1	19.0	30.0000	1	0
888	3	NaN	23.4500	1	0
889	1	26.0	30.0000	0	1
890	3	32.0	7.7500	0	1

891 rows × 5 columns

Lets also find out if there is NaN value

In [27]:
```python
X.columns[X.isna().any()]
```

Out[27]: Index(['Age'], dtype='object')

We can see that it tells us that the 'Age' column has some NaN value. lets chheck it

In [29]:
```python
X.Age.head(30)
```

Out[29]:
```
0     22.0
1     38.0
2     26.0
3     35.0
4     35.0
5      NaN
6     54.0
7      2.0
8     27.0
9     14.0
10     4.0
11    58.0
12    20.0
13    39.0
14    14.0
15    55.0
16     2.0
17     NaN
18    31.0
19     NaN
20    35.0
21    34.0
22    15.0
23    28.0
24     8.0
25    38.0
26     NaN
27    19.0
28     NaN
29     NaN
Name: Age, dtype: float64
```

We can see above that there is some NaN values.

Lets just fill the NaN value with the mean value of the Age column

In [31]:
```python
X.Age = X.Age.fillna(X.Age.mean())
X.head(30)
```

Out[31]:

	Pclass	Age	Fare	female	male
0	3	22.000000	7.2500	0	1
1	1	38.000000	71.2833	1	0
2	3	26.000000	7.9250	1	0
3	1	35.000000	53.1000	1	0
4	3	35.000000	8.0500	0	1
5	3	29.699118	8.4583	0	1
6	1	54.000000	51.8625	0	1
7	3	2.000000	21.0750	0	1
8	3	27.000000	11.1333	1	0
9	2	14.000000	30.0708	1	0
10	3	4.000000	16.7000	1	0
11	1	58.000000	26.5500	1	0
12	3	20.000000	8.0500	0	1
13	3	39.000000	31.2750	0	1
14	3	14.000000	7.8542	1	0
15	2	55.000000	16.0000	1	0
16	3	2.000000	29.1250	0	1
17	2	29.699118	13.0000	0	1
18	3	31.000000	18.0000	1	0
19	3	29.699118	7.2250	1	0
20	2	35.000000	26.0000	0	1
21	2	34.000000	13.0000	0	1
22	3	15.000000	8.0292	1	0
23	1	28.000000	35.5000	0	1
24	3	8.000000	21.0750	1	0
25	3	38.000000	31.3875	1	0
26	3	29.699118	7.2250	0	1
27	1	19.000000	263.0000	0	1
28	3	29.699118	7.8792	1	0
29	3	29.699118	7.8958	0	1

So we took care of the NaN value

Train Test split

In [67]:
```python
from sklearn.model_selection import train_test_split
```

In [68]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [69]:
```python
len(X_train)
```

Out[69]: 712

In [70]:
```python
len(X_test)
```

Out[70]: 179

Naive Bayes model

Now there are couple of Naive bayes classes but we are going to use 'GaussianNB'. u use this when ur data distribution is normal

In [71]:
```python
from sklearn.naive_bayes import GaussianNB
```

In [72]:
```python
model = GaussianNB()
```

In [73]:
```python
model.fit(X_train, y_train)
```

Out[73]:
```
▾ GaussianNB
GaussianNB()
```

In [74]:
```python
model.score(X_test, y_test)
```

Out[74]: 0.7318435754189944

Checking first 10 X_test data

In [75]:
```python
X_test.head(10)
```

Out[75]:

	Pclass	Age	Fare	female	male
239	2	33.000000	12.2750	0	1
309	1	30.000000	56.9292	1	0
432	2	42.000000	26.0000	1	0
126	3	29.699118	7.7500	0	1
748	1	19.000000	53.1000	0	1
251	3	29.000000	10.4625	1	0
120	2	21.000000	73.5000	0	1
477	3	29.000000	7.0458	0	1
230	1	35.000000	83.4750	1	0
758	3	34.000000	8.0500	0	1

Checking first 10 y_test data

In [76]:
```python
y_test.head(10)
```

Out[76]:
```
239    0
309    1
432    1
126    0
748    0
251    0
120    0
477    0
230    1
758    0
Name: Survived, dtype: int64
```

Predicting first 10 X_test data and u can see we got it all right

In [77]:
```python
model.predict(X_test[:10])
```

Out[77]: array([0, 1, 1, 0, 0, 1, 0, 0, 1, 0], dtype=int64)

Checking the probability of the datas being a survival or not

In [78]:
```python
model.predict_proba(X_test[:10])
```

Out[78]:
```
array([[9.83958927e-01, 1.60410733e-02],
       [2.45557815e-03, 9.97544422e-01],
       [1.76518799e-02, 9.82348120e-01],
       [9.92406951e-01, 7.59304926e-03],
       [8.78398888e-01, 1.21601112e-01],
       [3.98224222e-02, 9.60177578e-01],
       [9.25205509e-01, 7.47944908e-02],
       [9.92323338e-01, 7.67666156e-03],
       [7.62491088e-04, 9.99237509e-01],
       [9.92634182e-01, 7.36581847e-03]])
```

In [ ]: