- **Conditional Probability**
  - Conditional probability is a probability of an event occurring, given that another event (by assumption, presumption, assertion or evidence) has already occurred.

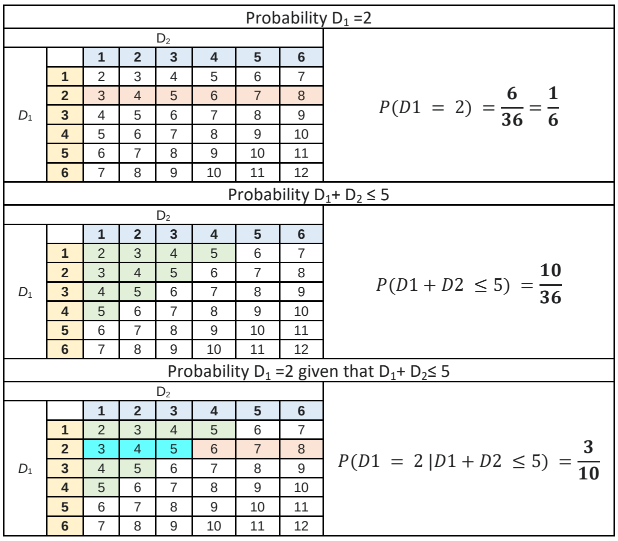
$$P(A|B) = \frac{P(A \,\cap\, B)}{P\,(\,B\,)}\,; P\,(\,B\,) \neq 0$$

P (A|B) = Probability of occurrence of event A given event B has already occurred

P (A∩B) = Probability of occurrence of A AND B

P(B)   = independent probability of B

  - Example: If two fair dice are rolled, compute probability that the face-up value of the first dice is 2, given that their sum is no greater than 5.
    - Event- rolling 2 dice. Sample space = 36 outcomes



Probability D_1 =2

D_1 \ D_2	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

$$P(D1 \;=\; 2) \;= \frac{\mathbf{6}}{\mathbf{36}} = \frac{\mathbf{1}}{\mathbf{6}}$$

Probability D_1+ $D_2 \leq 5$

D_1 \ D_2	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

$$P(D1 + D2 \;\leq 5) \;= \frac{\mathbf{10}}{\mathbf{36}}$$

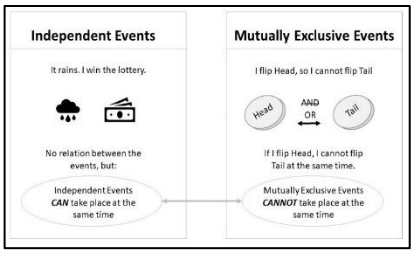
Probability D_1 =2 given that D_1+ $D_2 \leq$ 5

D_1 \ D_2	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

$$P(D1 \;=\; 2\,|D1 + D2 \;\leq 5) \;= \frac{\mathbf{3}}{\mathbf{10}}$$

    - $$\boxed{P(A|B) = \frac{P(A \cap B)}{P\,(\,B\,)} = \frac{\frac{3}{36}}{\frac{10}{36}} = \frac{3}{36} * \frac{10}{36} = \frac{3}{10}}$$

- **Independent vs Mutually exclusive events**
  - **Mutually exclusive event** is a situation when two events cannot occur at same time E.g., a fair coin toss will be a head or a tail.


Independent Events
It rains. I win the lottery.
No relation between the events, but:
Independent Events CAN take place at the same time
Mutually Exclusive Events
I flip Head, so I cannot flip Tail
Head
AND
OR
Tail
If I flip Head, I cannot flip Tail at the same time.
Mutually Exclusive Events CANNOT take place at the same time

    - $P(A \cap B) = P(A) * P(A|B)$
    - $P(A|B) = \frac{P(A\cap B)}{P(B)} = 0$
    - $P(B|A) = \frac{P(A\cap B)}{P(A)} = 0$
  - **Independent event** occurs when one event remains unaffected by the occurrence of the other event. E.g., if we take two separate coins and flip them, then the occurrence of Head or Tail on both the coins are independent to each other.
    - $P(A \cap B) \;=\; P(A) * P(B)$
    - $P(A|B) = \frac{P(A\cap B)}{P(B)} = \frac{P(A)*P(B)}{P(B)} = P(A)$
    - $P(B|A) = \frac{P(A\cap B)}{P(A)} = \frac{P(A)*P(B)}{P(A)} = P(B)$

- **Bayes Theorem**

⇨ By conditional probability,

⇨ $P(A|B) \;=\;$ Posterior $\;=\;$ Probability of A given B $\;=\; \frac{P(A \cap B)}{P\,(\,B\,)}$

⇨ $P(B|A) \;=\;$ Likelihood $\;=\;$ Probability of B given A $\;=\; \frac{P(B \cap A)}{P(A)}$

⇨ $P(A \;\cap\; B) = P(B \;\cap\; A)$

⇨ $P(A|B) * P\,(\,B\,) = P(B|A) * P(A)$

⇨ $P(A|B) = \frac{P(B|A)\,.\,P(A)}{P(B)}; \; P\;(B) \;\; \neq 0$

  - P(B) = independent probability of B
  - P(A) = independent probability of B

Naive Bayes Classifiers

- Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem.
- The fundamental assumption is that each feature makes an independent & equal contribution to the outcome.
  Independent E.g., the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds.
  Equal contribution E.g., knowing only temperature and humidity alone can't predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing equally to the outcome.
- **Note:** the independence assumption is never correct in real world situations but often works well in practice.
- **Example:** Find probability of Not playing golf, given that the weather conditions are "Sunny outlook", "Temperature is hot", "Normal humidity" and "no wind".

Feature matrix				Target
Outlook	**Temperature**	**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

- Solution: By Bayes Theorem,
  - ⇨ $\boxed{P(y \mid X) = \frac{P(X|y)\ .\ P(y)}{P(X)}}$
  - ⇨ Now, it's time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into the independent parts.
  - ⇨ For independent variables, $P(A \cap B)\ =\ P(A) * P(B)$
  - ⇨ $\boxed{P(y \mid X) = \frac{P(x_1|y)P(x_2|y)P(x_3|y)....P(x_{n-1}|y)P(x_n|y).\ P(y)}{P(x_1)P(x_2)P(x_3)...P(x_{n-1})P(x_n)}}$
  - ⇨ $\boxed{P(y \mid X) = \frac{P(y).\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)P(x_3)...P(x_{n-1})P(x_n)}}$
  - ⇨ $\boxed{P(y \mid X) \propto P(y)\ .\prod_{i=1}^{n} P(x_i|y)}$ As the denominator remains constant
  - ⇨ We find the probability of given features for all possible values of the class variable y and pick up the output with maximum probability. This can be expressed mathematically as:
  - ⇨ $\boxed{y = argmax_y\ P(y).\prod_{i=1}^{n} P(x_i|y)}$
  - ⇨ To apply above formula to the golf dataset, we need to find $P\left(x_i \mid y_j\right)$for each x_i in X and y_j in y.

Outlook	Yes	No	P(Yes)	P(No)
Rainy	2	3	22.2%	60.0%
Overcast	4	0	44.4%	0.0%
Sunny	3	2	33.3%	40.0%
Total	9	5	100%	100%

Temp	Yes	No	P(Yes)	P(No)
Hot	2	2	22.2%	40.0%
Mild	4	2	44.4%	40.0%
Cool	3	1	33.3%	20.0%
Total	9	5	100%	100%

Play	Count	Prob
Yes	9	64.3%
No	5	35.7%
Total	14	100%

Humidity	Yes	No	P(Yes)	P(No)
High	3	4	33.3%	80.0%
Normal	6	1	66.7%	20.0%

Windy	Yes	No	P(Yes)	P(No)
FALSE	6	2	66.7%	40.0%
TRUE	3	3	33.3%	60.0%

- Problem: Find whether to play golf or not given following conditions today = (Sunny, Hot, Normal, False)
- $\boxed{P(Yes \mid today) = \frac{P(Sunny\ outlook\ |Yes)P(Hot\ Temperature|Yes)P(Normal\ Humidity|Yes)P(False\ Wind|Yes).\ P(Yes)}{P(Today)}}$
- $\boxed{P(Yes \mid today) \propto \frac{2}{9} * \frac{2}{9} * \frac{6}{9} * \frac{6}{9} * \frac{9}{14} \approx \frac{1296}{91584} \approx 0.0141}$
- $\boxed{P(No| \ today) = \frac{P(Sunny\ outlook\ |No)P(Hot\ Temperature|No)P(Normal\ Humidity|No)P(No\ Wind|No).\ P(No)}{P(Today)}}$
- $\boxed{P(No \mid today)\ \propto \frac{3}{5} * \frac{2}{5} * \frac{1}{5} * \frac{2}{5} * \frac{5}{14} \approx \frac{60}{8750} \approx 0.0068}$
- Since we have ignored denominator term the sum of probabilities is not coming 1. We will normalize this value

$$\boxed{P(Yes \mid today) = \frac{0.0141}{0.0141 + 0.0068} = 0.67}$$

$$\boxed{P(No \mid today) = \frac{0.0068}{0.0141 + 0.0068} = 0.33}$$

- Since,
  - P (Yes | today) > P (No | today)
  - So, prediction that golf would be played is 'Yes'.

- **Naive Bayes on text classification**
  - **Text Corpus**

Sr. No.	Xq	No of words	Y		
1	A Great Game	3	sports		
2	The Election Was Over	4	non-sports		Word Count
3	A Very Clear Match	4	sports	Sports	11
4	Clean but Forgettable Game	4	sports	Non-Sports	9
5	It Was A Close Election	5	non-sports	Unique	14

Problem- Classify New = a very close game whether it is **sports** or **non-sports**

$$\boxed{P(New) = P(a\,) * P(very) * P(close) * P(game)}$$

$$\boxed{P(New \mid Sports) \approx P(a\ |sports) * P(very|sports) * P(close|sports) * P(game|sports).*\ P(sports)}$$

$$\boxed{P(New \mid sports) \approx \frac{2}{11} * \frac{1}{11} * \frac{0}{11} * \frac{2}{11} * \frac{3}{5} \approx 0}$$

$$\boxed{P(New \mid NonSports) \approx P(a\ |NonSports)P(very|NonSports)P(close|NonSports)P(game|NonSports) *\ P(NonSports)}$$

$$\boxed{P(New \mid NonSports) \approx \frac{1}{9} * \frac{0}{9} * \frac{0}{9} * \frac{1}{9} * \frac{2}{5} \approx 0}$$

- **Laplace/Additive Smoothing**
  - Since a query word "Close" doesn't exist in our training data. Due to that numerator becomes zero
  - Now, there are equal chances that it belongs to an either category. This equal probability is calculated by

$$\boxed{P(Word|y) = \frac{\#word + \alpha}{\#Word\ Corpus|y\ +\ \alpha K.} = \frac{Word\ Count\ \ +\ \alpha}{(Total\ number\ of\ words\ where\ class\ is\ y) + Total\ number\ of\ unique\ words * \alpha}}$$

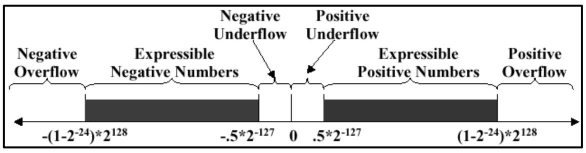
  - Majority of the time alpha =1, as the alpha increases this likelihood $\approx$ 0.5 e.g., $a = 10000$

$$\boxed{P(w_i|\ y = 1\,) = P(w_i|\ y = 0\,)\ \frac{2 + 10000}{1000 + 10000 * 2} = \frac{1}{2}}$$

$$\boxed{P(New \mid Sports) \approx \frac{2+1}{11+14} * \frac{1+1}{11+14} * \frac{0+1}{11+14} * \frac{2+1}{11+14} * \frac{3}{5} \approx 0.000027648}$$

$$\boxed{P(New \mid NonSports) \approx \frac{1+1}{9+14} * \frac{0+1}{9+14} * \frac{0+1}{9+14} * \frac{1+1}{9+14} * \frac{2}{5} \approx 0.00000571753}$$

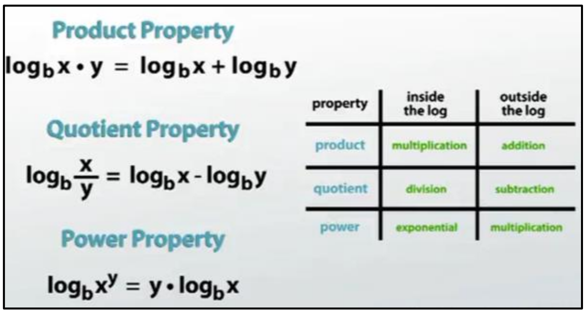
- **Log-probabilities for numerical stability**
  - Since, we are multiplying all the probabilities the output number is significantly small, if there are d features and if d becomes 100. The output value is so less.



  - To solve this problem, the equation is modified as below.

$$\boxed{\log P(y \mid X)\ = \log\left(P(y)\ \prod_{i=1}^{n} P(x_i|y)\right)}$$

- This is done considering,



Product Property

$\log_b x \cdot y = \log_b x + \log_b y$

Quotient Property

$\log_b \frac{x}{y} = \log_b x - \log_b y$

Power Property

$\log_b x^y = y \cdot \log_b x$

property	inside the log	outside the log
product	multiplication	addition
quotient	division	subtraction
power	exponential	multiplication

- **Bias-Variance Trade-Off (overfitting Underfitting cases)**
  - low variance but high bias – Under Fitting.
  - high variance but low bias – Overfitting
  - Case1(α = 0)
    - Suppose in the corpus of 1000 words, a word occurs vary rarely say 2. So, if I remove them from training data the probability becomes 0 from 2/1000
    - This implies a small change in the training dataset results in large model changes.
    - This is high variance & overfitting
  - Case2(α = 10000)
    - In above case, when α is 10000

$$\boxed{P(w_i|\, y = 1\,) = P(w_i|\, y = 0\,)\ \frac{2 + 10000}{1000 + 10000 * 2} = \frac{1}{2}}$$

    - When α is large, the classifier will be giving similar probabilities that a query point belongs to both classes

$$\boxed{P(w_i|\, y = 1\,)\ \approx\ P(w_i|\, y = 0\,)}$$

    - This happens as the Likelihood gives similar value for both the classes. The final probability will be dependent on prior.

$$\boxed{P(y \mid X) \propto\ P(y) \prod_{i=1}^{n} P(x_i|y)}$$

  - The probability of Prior will be dependent on no of data points lying in dataset corresponding to the specific class. If more points lie in one class it's probability will be higher. This is underfitting.
  - in naïve base the hyperparameter will be α as change in this value changes fitting of the model.
- **Feature importance and interpretability**
  - After classification, we will have summary table as follows

Word	P (W_i| Y =1)	P (W_i| Y =0)
W_1		
.		
.		
W_n		

  - From just looking at higher probabilities, we can conclude which features will be important in classification. The top values belonging to the column 1 & column2 will represent the most important words or features for particular class
- **Imbalanced data**
  - Imbalanced data problem can be solved by following 2 ways. Suppose we have an imbalanced dataset such that n1 points belongs to class 1 & n2 points belong to class2.
  - Up sampling or Down sampling
    - By doing so, $\boxed{n_1 \approx\ n_2\, P(\,y = 1) =\ P(\,y = 0) = \frac{1}{2}}$
  - By dropping the P(y=1) & P(y=0) i.e., making the values 1 as we only consider majority vote
  - The other problem with Imbalanced data is that α will be affecting minority class more than that of majority class.
- **Outlier**
  - Outlier in text or training data means that the text occurs very less times.
  - A simple solution is to set a threshold such that if that word occurs less than τ(tau), we will ignore the word
  - Another solution is to apply Laplace smoothing.

- **Missing Values**
  - For missing values in categorical data like NaN. the simplest solution is to consider NaN itself as a category.
  - Numerical imputations can be applied
- **Handling Numerical Features (Gaussian Naïve Bayes)**

X	F1	F2	F3	.	.	Fj	.	Fd	Y
X1									1
X2									1
X3									1
X4									1
X5									1
X6									1
X7									0
X8									0
Xi	Xi1	Xi2	Xi3	.	.	Xij	.	Xid	0
.									0
.									0
Xn									0

  - For the query point x_i

$$\boxed{P(y=1\mid x_i) \propto P(y=1)\prod_{i=1}^{d} P(x_{ij}|y_i=1)}$$

  - Probability of $P(y=1) = \frac{n1}{n1+n2}$ = i.e., n1 & n2 are the no of points belonging to class 1 & class2
  - To compute probability of specific numeric value from a feature we need to draw pdf of that feature
  - Here, the base assumption is this feature follows a gaussian distribution
- **Other Types**
  - Gaussian: It is used in classification and it assumes that features follow a normal distribution.
  - Multinomial**:** It is used for discrete counts. Multinomial Naive Bayes says know that each $P(x_i|y)$ is a multinomial distribution, rather than some other distribution. This works well for data which can easily be turned into counts, such as word counts in text.
  - Bernoulli**:** The binomial model is useful if feature vectors are binary (i.e. 0 and 1). One application would be text classification with 'binary bag of words' model

- **Similarity or distance matrix**
  - Since naïve bayes is a probability-based classification method, we can't use similarity or distance matrix in naïve bayes
- **Large Dimensionality**
  - When the dimensionality is high Likelihood Probability is so less since we multiply lot of values lying between 0 & 1, This is called as numerical stability issues.
  - To avoid this problem, it is always recommended to use log probabilities
- **Summary**
  - The basic assumption of Naive bayes is Conditional independence of features, as long as this assumption is true, NB performs fairly well. When It starts becoming false, the performance starts deteriorating.
  - Some studies also show, even if some part of features is dependant, naive base performs fairly well
  - For text classification, Naive base is very useful. Naïve bayes is considered as baseline or benchmark for text classification.
  - Just like naive bayes is applied for binary classification we can extend the same concept for multiclass classification
  - Naive bayes is extensively used for categorical features
  - Naive bayes is not often used for real-valued or numerical features
  - The interpretability of NB model is good
  - Space & Time complexity
    - Runtime complexity = Low
    - Train Time complexity = Low
    - Runtime space = Low
  - The model will easily overfit if Laplace smoothing is not used