Name: Upadhyay Akarsh
Company: Spartificial
Intern type: ML Research Intern

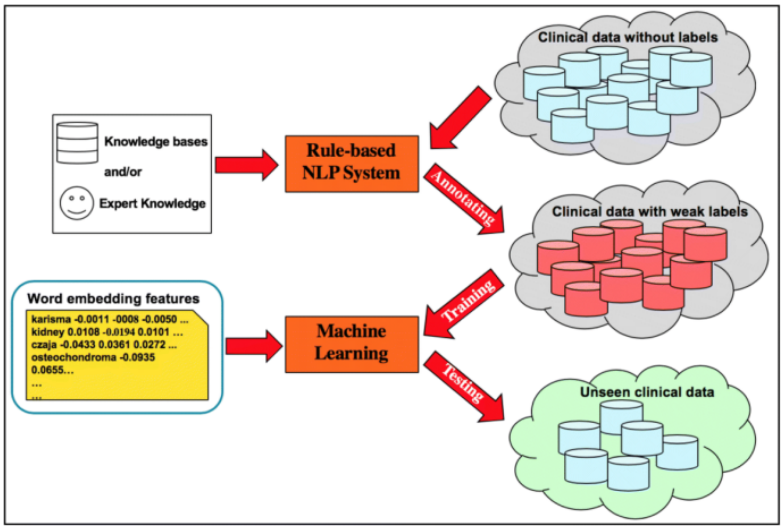
Problem Statement: Given a raw text which contains the prescription given to a patient suffering from some kind of disease, develop a Machine Learning algorithm which would help classify the given prescribed text into one of the Medical Specialties.

The Medical Specialities includes:

- Cardiovascular / Pulmonary
- Consult - History and Phy.
- Discharge Summary
- Gastroenterology
- General Medicine
- Neurology Obstetrics / Gynecology
- Orthopedic
- Radiology
- SOAP / Chart / Progress Notes
- Surgery
- Urology

ABOUT THE PROBLEM STATEMENT:

- We know that social media is growing day by day, and more and more platforms are coming online to provide medical treatment to the people who are located in remote areas and do not have the facility to reach the hospital.
- Also, people sometimes post about their health on the internet, for example, people tweet about their mental health or post which are related to it.



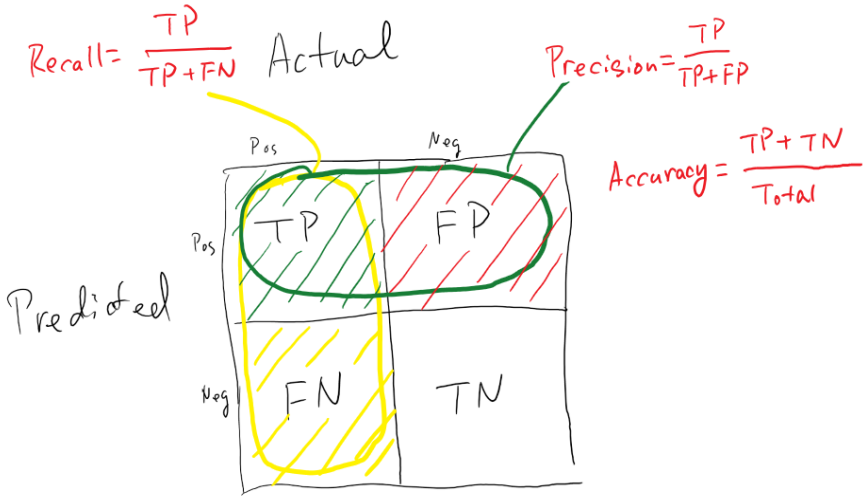
OUR MOTIVE:

- The target labels are represented by the name of needed diagnostics procedure
- The value of the solution might be found in helping a doctor to find the optimal solution for diagnostics order. Patients can save time and money, and doctors can serve a patient more efficiently on sparing time for unnecessary diagnostics.
- Moreover, in difficult cases the algorithm may help a doctor to find a diagnosis faster, which in some cases may be extremely valuable, up to saving lives.

HOW WOULD WE APPROACH THIS PROBLEM:

- Perform the data review:
    - This steps includes:
        - Loading the data, and observing the entries present in the data, and this is an important step in the understanding of the data, and getting familiar with it.
        - Also, we would, along with entries, would also see the target label (in our case, it is the medical speciality)
        - Also, since this is not a uniform dataset, there would be cases of the class imbalance, and we would apply some statistical techniques, which would help us in dealing with the class imbalance.

- Why is class Imbalance an issue?
    - Most machine learning algorithms assume data is equally distributed. So when we have a class imbalance, the machine learning classifier tends to be more biased towards the majority class, causing bad classification of the minority class.
    - Now, let us understand this problem through an example.Suppose there is a task of anomaly detection, where you need to tell whether the given value is an anomaly or not.
    - Now, in real life scenarios, there are very few anomalies present in the dataset.
    - Let us assume that the dataset contains 1000 samples, and there are 50 anomalies. Now, suppose that you make a model, and model learnt the parameters,and outputs any input as non-anomaly.
    - Now, the accuracy of the model on the total dataset is 95%, which is great, but our main aim was not to predict any input as non-anomaly, but we developed it as a predictor to output the given data point as anomalous or non-anomalous.



    - So, here the use of other metrics such as precision or recall comes into play. This metrics are class specific, which means that it does not consider the average correct prediction (which is the case of average)
    - However, the weighted precision and recall are used for overall precision and recall, for a lot of classes.

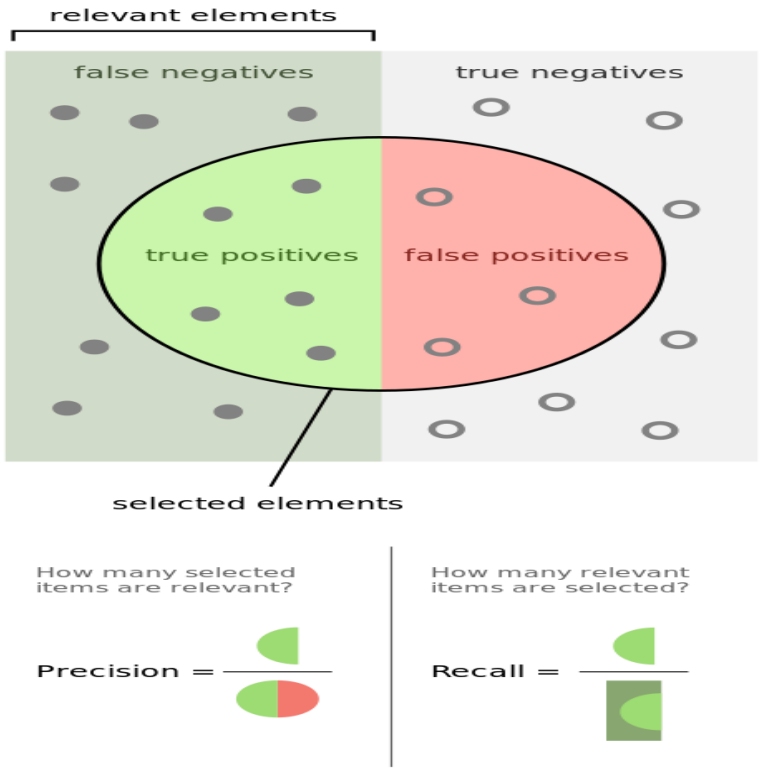
■ PRECISION AND RECALL:

- In pattern recognition, information retrieval and classification (machine learning), precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances.

- Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance.

- The below image describes the meaning of precision and recall, in terms of the true positive, true negative, false positive and false negative.

- Terminologies: (with the help of example)
    - A true positive is an outcome where the model correctly predicts the positive class. Similarly, a true negative is an outcome where the model correctly predicts the negative class.
    - A false positive is an outcome where the model incorrectly predicts the positive class. And a false negative is an outcome where the model incorrectly predicts the negative class.



        - Also, in order to take into account both the precision and the recall metrics into a single metric, we use the F1-Score, which is the harmonic mean of precision and the recall metrics.

- Perform the Text Pre-processing:

    - Why is text preprocessing important?
        - In simple words, the computer does not understand english or any other language. All the computer understands are the numbers.

        - So, It transforms text into a more digestible form so that machine learning algorithms can perform better.
        - 

        - Now, to convert the given sentence into a form that the computer can understand, there are some methods, which helps in text preprocessing. Let us talk about each one of them in detail:
            - **WORD TOKENIZE:**

                - Word tokenization is the process of splitting a large sample of text into words.
                - This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis like classifying and counting them for a particular sentiment etc.
                - Eg: Suppose there is a statement: "It originated from the idea”, then the corresponding word tokenized form is: ['It', 'originated', 'from', 'the', 'idea']

            - **SENTENCE TOKENIZE:**
                - This is also a form of tokenization, we can tokenize the sentences in a paragraph like we tokenized the words.

    - Consider the example: "Sun rises in the east. Sun sets in the west.", then the tokenized form would be ['Sun rises in the east.', 'Sun sets in the west.']

- Now, having done the tokenization, we just assign a number to each of this word, and then convert it into its one-hot encoding, and then as we have converted into the feature vector, we can pass it through the machine learning algorithm, and get the prediction.

- Now, there are also some issues with the sentence. Maybe there could be a word whose different forms are present in the sentence, meaning maybe used in its past perfect or present tense, however the meaning remains the same.

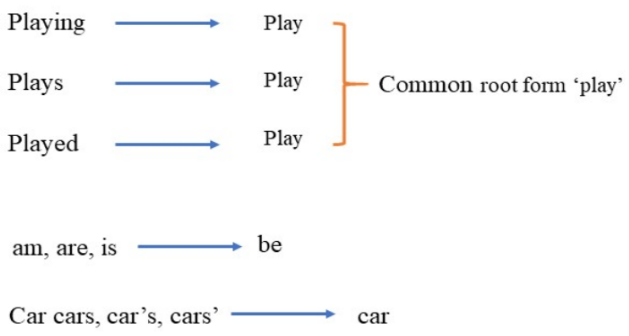
- So, here the role of Stemming and Lemmatization comes into play, whose main work includes converting the original word into its root form, so that the tokenization becomes easy to understand.
- Now let us understand what are Stemming and Lemmatization are:

- **STEMMING:**

    - First of all, let us understand the scenario in which stemming is used, and then we would approach to applying stemming technique for the purpose of text normalization, so that, further preprocessing can be done to convert them into numbers so that ML algorithm can be applied to those numbers

    - In our common language, when we speak or write something, it contains a lot of words that are derived from its root word.

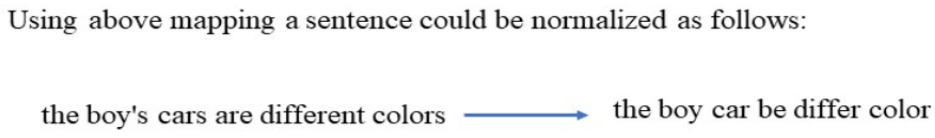
    - Let us understand this with the help of the following diagram:



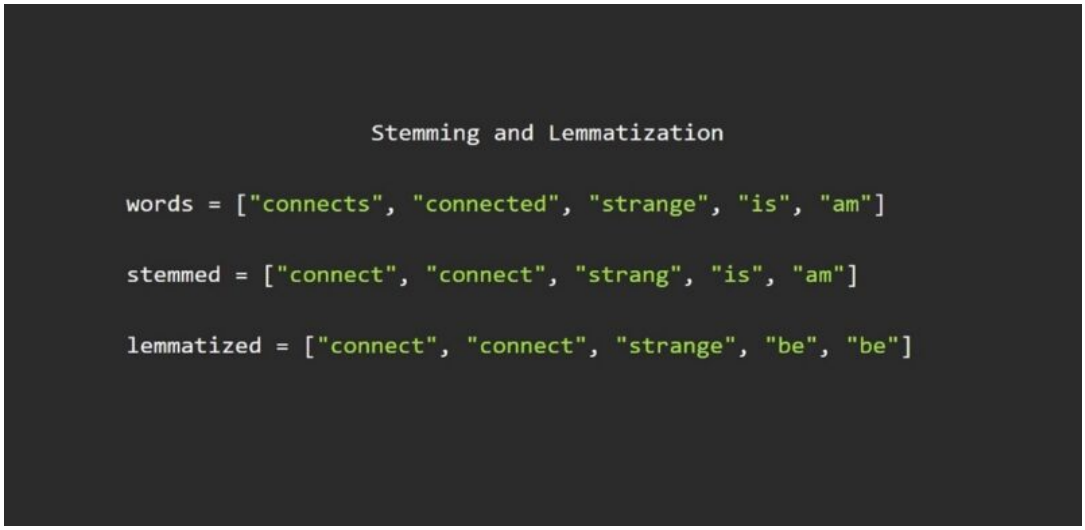
- This is the meaning of the infected sentence. Infected sentence means, the sentence which contains some words, which are derivatives from its root word.

- Why convert such an infected sentence into a pure sentence?
    - This is because of the fact that, ultimately all the derived words have the same meaning as that of the root word, just the suffix or prefix has been added to the sentence.

- **Role of Stemming:**
    - So, where does stemming come into picture in this type of sentence?
    - The main role of stemming is to convert all the infected words into its root word by snipping the suffix from it, so that the text can be considered as a base sentence.
    - Base sentence means the sentence which doesn't contain any infected words.
    - So, with some code, the method of stemming can be applied, which takes an input sentence and also outputs a sentence, which contains all the base words.
    - A figure shown below would help you understand the thing:



- **Role of Lemmatization:**
    - You have got an understanding of what Stemming is all about. Now, what if I said that, the context of the word, just not depends upon the root word, but also the words which are neighbors of it?
    - The below image makes it clear, what Lemmatization does:



- If we think of stemming as just taking the best guess of where to snip a word based on how it looks, lemmatization is a more calculated process.

- It involves resolving words to their dictionary form. In fact, a lemma of a word is its dictionary or canonical form!

- For lemmatization to resolve a word to its lemma, it needs to know its part of speech.

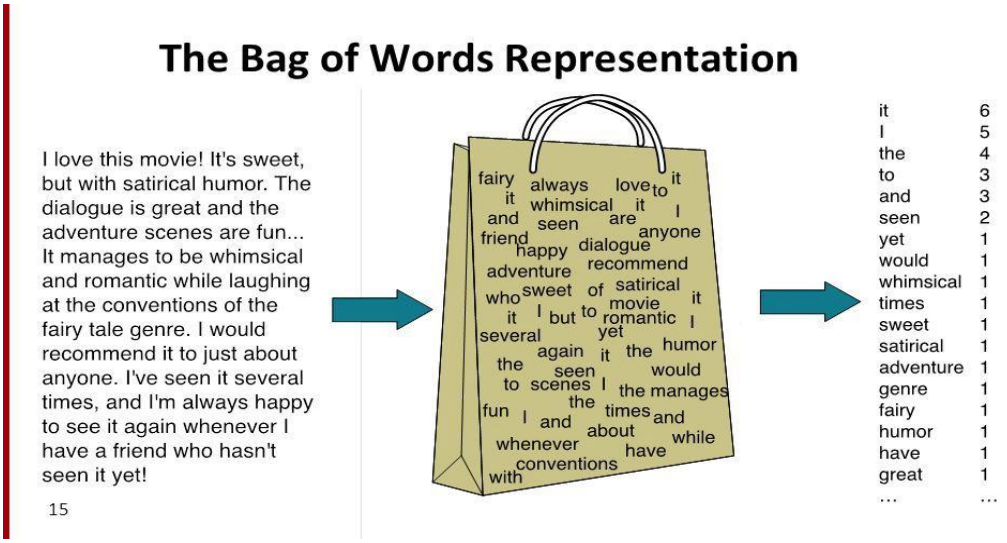
- That requires extra computational linguistics power such as a part of speech tagger. This allows it to do better resolutions (like resolving is and are to “be”).

- All things are okay, how to convert the words into numbers?
  - **Bag of Words (BoW):**
    - The Bag of Words (BoW) model is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words vector (a string of numbers)
    - Let’s recall the three types of movie reviews we saw earlier:
      - Review 1: This movie is very scary and long
      - Review 2: This movie is not scary and is slow
      - Review 3: This movie is spooky and good
    - We will first build a vocabulary from all the unique words in the above three reviews. The vocabulary consists of these 11 words: ‘This’, ‘movie’, ‘is’, ‘very’, ‘scary’, ‘and’, ‘long’, ‘not’, ‘slow’, ‘spooky’, ‘good’.

- We can now take each of these words and mark their occurrence in the three movie reviews above with 1s and 0s. This will give us 3 vectors for 3 reviews:

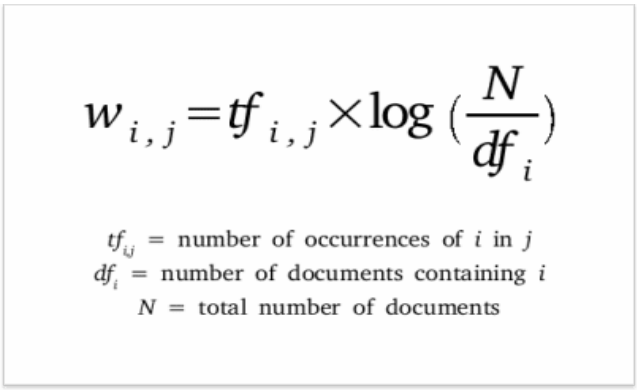
	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

- So, the vectors becomes:
  - Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]
  - Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]
  - Vector of Review 3: [1 1 1 0 0 0 1 0 0 1 1]
- We got the vectors, right, now we can pass this through the machine learning algorithm, and we can predict the category.
- However, there are some problems with this approach:
  - If the new sentences contain new words, then our vocabulary size would increase and thereby, the length of the vectors would increase too.
  - Additionally, the vectors would also contain many 0s, thereby resulting in a sparse matrix (which is what we would like to avoid)
  - We are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.



- **TF-IDF (Term Frequency- Inverse Document Frequency)**
  - We have got how to convert a sentence to a vector, however, sometimes if there are words that are too frequent, due to which, the value of those words in the sentence becomes less worthy.

  - This concept is covered by TF-IDF, where it counts the frequency of the word, and then applies the logarithm to the value which is equal to (total document/document which contains the word), and multiplies it by term frequency.



$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

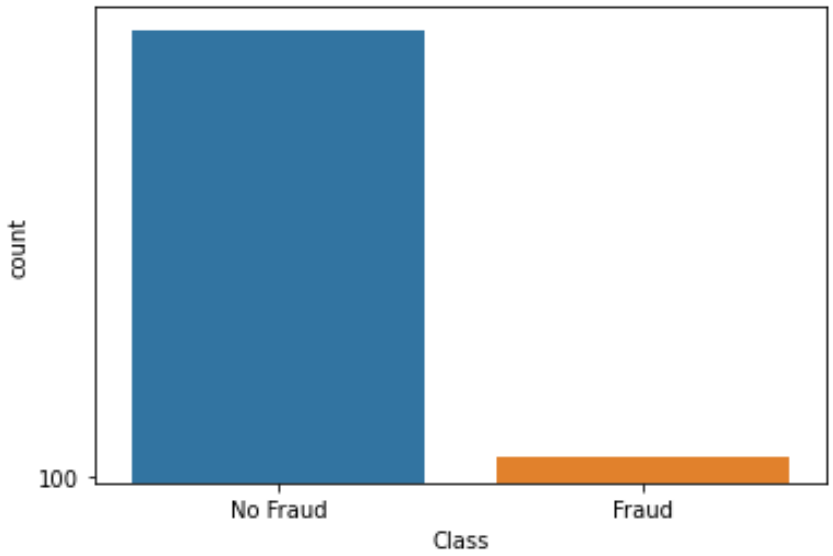
$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

- So, you got to know how to preprocess the text, and then pass it through the ML algorithm. We would talk about the ML algorithm a bit later, butfor now, let us focus on the labels

- Practically, you know that, there are some diseases and their specialty as for example it could be allergy, which are more common, and there are also some specific medical situations, which are rare.

- Similar is the case, here, there are some labels, which are common, while others are not that common.

- This situation is known as class imbalance, and could prove to be a bad thing for our ML algorithm, as if most of the time, it would predict the majority class label.



- For handling the class imbalance, there are some statistical techniques, out of which we would, see the two techniques, as mentioned below:

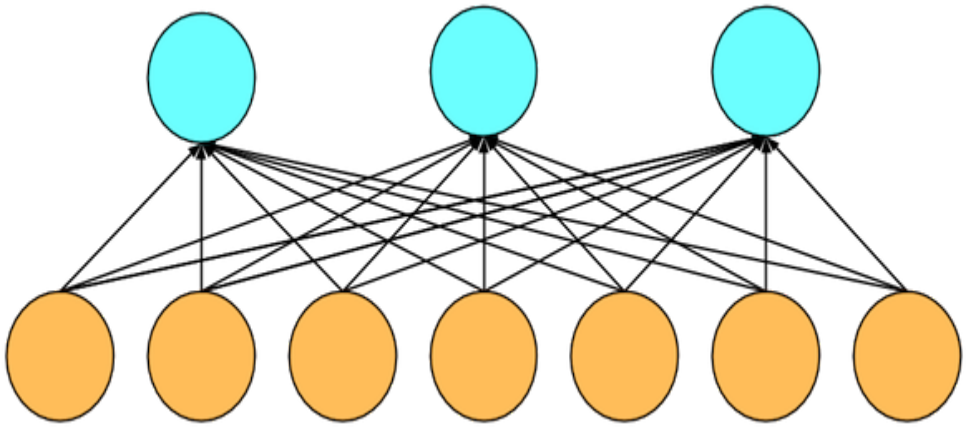
    - **Oversampling:**
        - Duplicating samples from the minority class
        - As in our case, the increasing the number of Fraud labelled data points.

    - **Undersampling:**
        - Deleting samples from the majority class.
        - As in our case, the reducing the number of Non-fFraud labelled data points.

    - In other words, both oversampling and undersampling involve introducing a bias to select more samples from one class than from another, to compensate for an imbalance that is either already present in the data, or likely to develop if a purely random sample were taken.

- Okay, so having done the sampling part for handling the class imbalance, now let us come back to the Machine Learning algorithm.
- We would be using three machine learning algorithms.
    - They are:
        - Logistic Regression
        - Naive Bayes
        - LightGBM
- So let us understand these three algorithms in detail.

- **Logistic Regression:**
    - Multiclass logistic regression is also called multinomial logistic regression and softmax regression. It is used when we want to predict more than 2 classes.



    - Logistic regression uses a sigmoid function to predict the output. The sigmoid function returns a value from 0 to 1. Generally, we take a threshold such as 0.5.

    - If the sigmoid function returns a value greater than or equal to 0.5, we take it as 1, and if the sigmoid function returns a value less than 0.5, we take it as 0.

- $$h = \frac{1}{1 + e^{-z}}$$

    - How to solve this?
    - We will treat each class as a binary classification problem the way we solved a heart disease or no heart disease problem.
    - This approach is called the one vs all method
    - In the one vs all method, when we work with a class, that class is denoted by 1 and the rest of the classes becomes 0.
    - For example, if we have four classes: cars, trucks, bikes, and boats. When we will work on the car, we will use the car as 1 and the rest of the classes as zeros.
    - Again, when we will work on the truck, the element of the truck will be one, and the rest of the classes will be zeros.

- **Naive Bayes:**

    - Let us first understand, what the Bayes Theorem means:
        - 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, the presence of one particular feature does not affect the other. Hence it is called naive.

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

        - The variable y is the class variable, which represents if it is suitable to clinical text labels given the conditions. Variable X represents the parameters/features.

        - X is given as, x_1,x_2….x_n represent the features, i.e they can be mapped to the text features.By substituting for X and expanding using the chain rule we get,

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

    - Now, you can obtain the values for each by looking at the dataset and substitute them into the equation.

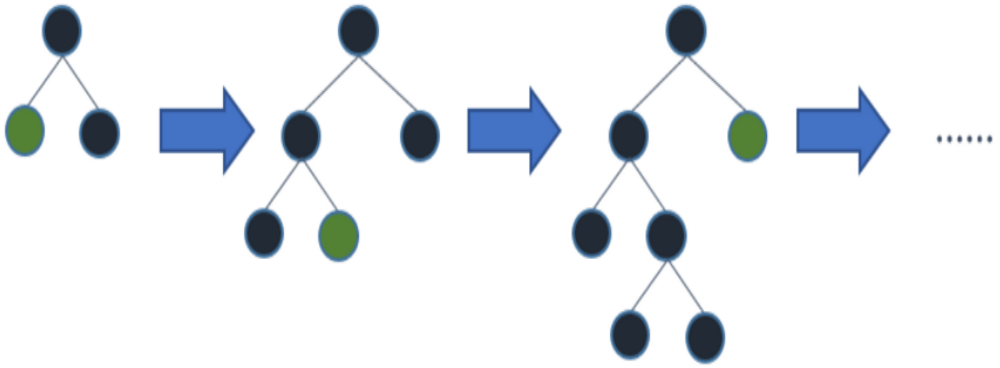
    - For all entries in the dataset, the denominator does not change, it remains static. Therefore, the denominator can be removed and a proportionality can be introduced.

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

- **LightGBM:**
    - Light GBM is a gradient boosting framework that uses tree based learning algorithms.

    - Light GBM grows tree vertically while other algorithms grow trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithms grow level-wise.

    - It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.



Leaf-wise tree growth

- **And finally, let us understand confusion Matrix:**
  - Well, it is a performance measurement for machine learning classification problems where output can be two or more classes. It is a table with 4 or more different combinations of predicted and actual values.

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
Predicted Values	Negative (0)	FN	TN

  - True Positive:
    - Interpretation: You predicted positive and it's true.
    - You predicted that a woman is pregnant and she actually is.
  - True Negative:
    - Interpretation: You predicted negative and it's true.

        - You predicted that a man is not pregnant and he actually is not.
    - False Positive: (Type 1 Error)
        - Interpretation: You predicted positive and it's false.
        - You predicted that a man is pregnant but he actually is not.
    - False Negative: (Type 2 Error)
        - Interpretation: You predicted negative and it's false.
        - You predicted that a woman is not pregnant but she actually is.
- Results:

]:

	Preprocessing	Model	Precision	Recall	F1-score	Accuracy
0	Count Vectorize	Naive Bayes	0.776678	0.235234	0.361101	0.235234
1	Count Vectorize	Naive Bayes	0.820368	0.237271	0.367560	0.237271
2	Count Vectorize	LightGBM	1.000000	0.236253	0.382208	0.236253
3	TF-IDF 1-grams	Naive Bayes	0.780723	0.237271	0.363481	0.237271
4	TF-IDF 1-grams	Naive Bayes	0.984988	0.436864	0.556071	0.436864
5	TF-IDF 1-grams	LightGBM	0.977667	0.427699	0.540830	0.427699
6	Word2vec	Naive Bayes	0.974696	0.219938	0.351502	0.219938

- This is the result, which we got after applying the technique of oversampling, and these were the results, which could be considered as final results, since we induced the data, which was not suffering from the class imbalance.
- We see, that the recall is quite low, and hence, this can be reasoned as follows:
    - Maybe, the data has a distribution, which cannot be captured by these models.
    - Also, there are a lot of dimensions, and hence curse of dimensionality plays an important role, which should be taken into account