Q Why NLP?

**Roadmap for NLP**

1. Text Pre-processing  
   a. Tokenization , stop words, lemmatization , stemming   
   b. BOW,TFIDF, unigrams, bigrams   
   c. word2vec, avgword2vec
2. Solve good number of ml use cases
3. RNN, LSTM RNN, GRU RNN -> Deep Learning
4. Advance text pre-processing  
   a. word embedding   
   b. word2vec
5. Bi directional LSTM, Encoder, Decoder , Attention Model
6. Transformer
7. BERT

Libraries:-

1. NLTL
2. Spacy
3. Tax blob
4. Tensor flow
5. Pytorch
6. Huggigface

**Tokenization** ->

In order to get our computer to understand any text, we need to break that word down in a way that our machine can understand.

Creating Vocabulary is the ultimate goal of Tokenization

* **Word tokenizer** -> One of the major issues with word tokens is dealing with Out Of Vocabulary (OOV) words.
* **Subword** ->   
  Split piece of text into subwords (or n-gram characters). Ex smartest -> smart-est, lower -> low-er  
  a. Byte Pair Encoding
* **Characters Tokenizer** ->   
  **Merits**-> It overcomes the drawback which we have with word tokenizer.

**Demerits** ->   
a. length of the input and output sentences increases rapidly as we are representing a sentence as a sequence of characters.  
b. it becomes challenging to learn the relationship between the characters to form meaningful words.

**Text Pre-processing -1**

Stemming ->

Process of reducing words to its base word stem/ root word / base form.

Historical **Histori** -> Not Have any meaning

History

Finally

Final **Fina** -> Meaning is gone

Finalized

Going

Goes go

Gone

Advantages:-

1. It is very fast

Disadvantages:-  
 1. It is removing the meaning of words.

Lemmatization:-

It overcome the problem with stemming. It is reduces the words to its base form also words are meaning full.

It does not remove the meaning for words.

Ex-

Finally

Final -> Final

Finalized

Disadvantages -;  
1. It is very slow because it compares each word with defined dictionary.

Use cases ->

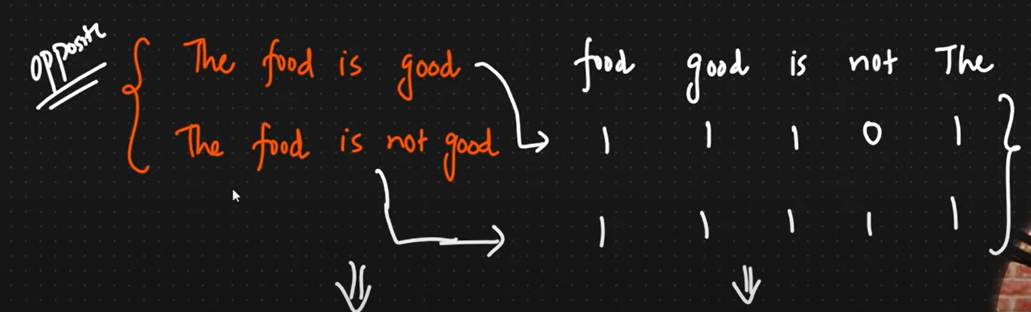
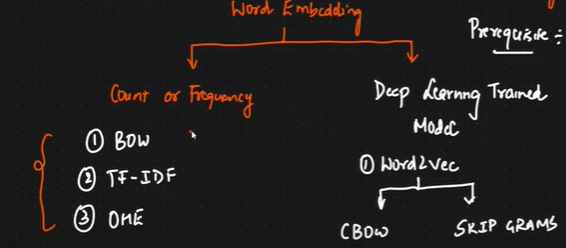
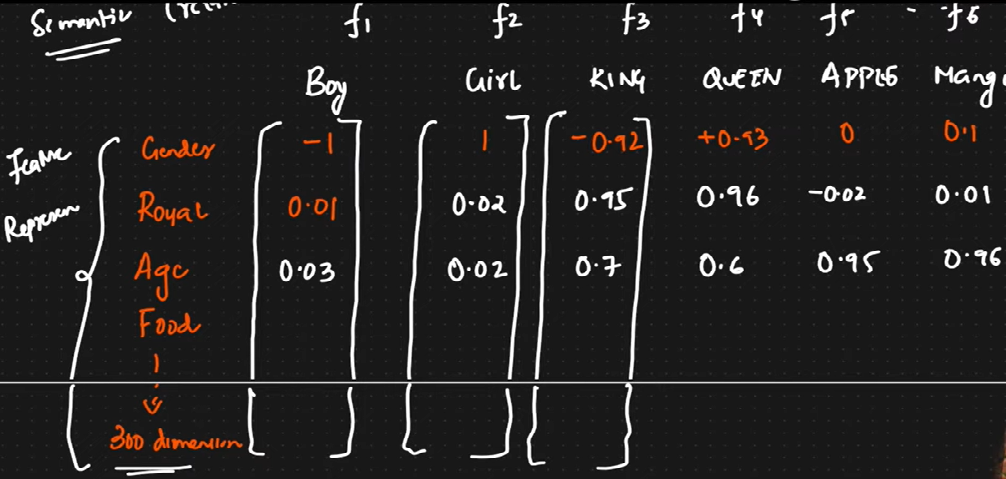
**Stemming** **Lemmatization**

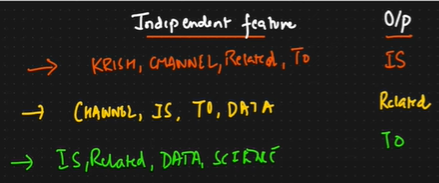
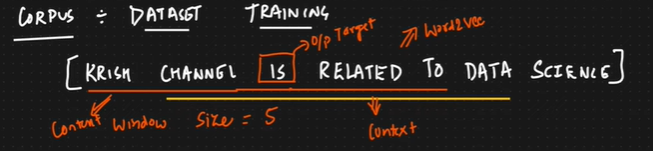
Spam classification Text summarization

Review classification Language Translation   
 Chatbots

**Text Pre-processing 2 ->**

**Word -> vector**

1. **One Hot encoding  
   merits**:-  
   a. simple  
   b. Intuitive  
   **demerits**:-  
   a. Sparse metrics which is computational expensive  
   b. Out of vocabulary   
   c. Input size is not fixed  
   d. Semantic meaning of b/w words is not captured
2. **Bag of words/n-grams -> Binary Bow  
   merits**:-  
   a. Simple and intuitive   
   **demerits**:-  
   a. Sparsity  
   b. OOV  
   c. order of words does not matter / Not able to capture the word ordering  
   d. Not able to capture semantic meaning of docs  
     
   **if we calculate the similarity of tow sentence will get both sentence are very much similar but both sentence are apposite to each other.**
3. **TD-IDF**Give more weightage to rare words.  
   **Term Frequency** -> will find rare words in sent. Term frequency computed for every sentence. TF = No. of rep of words / total number of words in sentence.  
   **Inverse Document Frequency** -> IDF computed for each words. IDF = Number of sentences / number of sentence in which word present. **Merits**:-  
   intuitive  
   word importance is getting captured  
   **demerits**:-  
   Sparsity problem is there.  
   OOV
4. **Word2vev -> Feature Representation**  **Merits:-  
   1. Limited dimensions  
   2. Reduced sparsity  
   3. Sematic Meaning**

**Feature representation is created by trained model.  
Cosine Similarity -> Value towards 0 is more similar.  
Distance = 1-cosine similarity   
1. COW (Continuous bag of words):-  
We will generate training data.** **Window size -> Will be odd number ex-5  
Hidden layers is equal to window size   
2. Skip Gram  
It is apposite of CBOW  
Avg Word2Vec**

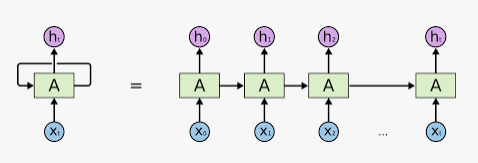
IVQ-> what is the diff bw train, test and validation data.   
Validation data is used for hyper parameter tuning.  
IVQ-> why we use Random forest instead of Decision tree.

DT-> Low bias, high variance   
Random Forest -> Low bias, Low variance, Prevent overfitting, high variance

RNN- Recurrent Neural Network  
Chatbot -> Sequence of words is very important

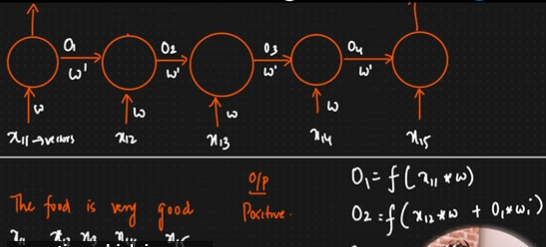
Language Translation  
Time series Data

Text Generation -> Suggestions

  
Types of RNN:-

1. One to one -> One input and one output ex- Image classification
2. One to Many -> Single input and multiple output ex – text generation , Music generation, google search suggestion, Movie recommendation
3. Many to one -> ex. Sentiment analysis , Sales prediction
4. Many to Many -> Language translation, QNA, Chatbot

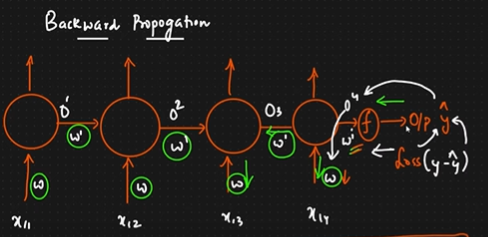
Forward Propagation in RNN:-

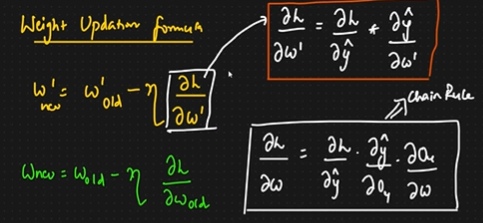


IVQ. Fit\_transform and transform

IVQ. What is the relationship bw left and right skewed data.

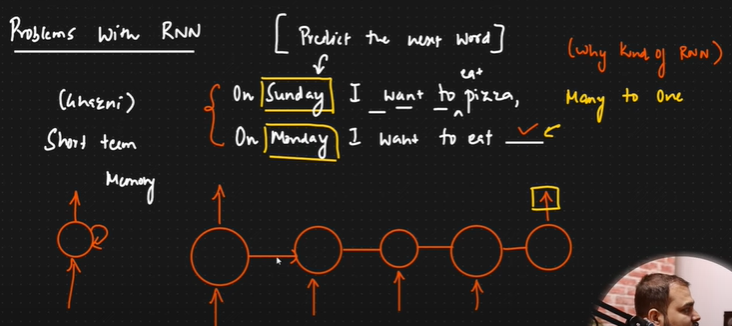
IVQ. How do we check normal distribution -> QQ plot

Backward Propagation in RNN:-  




Suppose we are using sigmoid activation function (o/p 0 to 1) and if we calculate derivative of sigmoid then o/p will be 0 to 0.25. So here weights updates are very small. So all the time weights will be same.

Demerit ->   
1. Gradient Vanishing Problem or dead neuron  
2. Context info   
  
Problem with RNN:-  
When we have longer sentence the RNN unable to capture context of sentence because it has short term memory. It will work fine of smaller sentences (2-3 words).



LSTM RNN->

