Day 5

Agenda :

1.- Foundation of NLP
2. text preprocessing
3. Bag of Words
4. TFIDF
5. Stopwords
6. Stemming
7. Tokenization
8. Performance Matrix, Precesion, Recall, AUC score
9. Sentiment analysis on textual data
10. Naive bays algorithem

1. Categorical Data

- Categorical data are variables that contain label values rather than numeric values.

	gender	class	city
0	Male	A	Delhi
1	Female	B	Gurugram
2	Male	C	Delhi
3	Female	D	Delhi
4	Female	A	Gurugram

1. One-Hot Encoding

One hot encoding is a binary encoding applied to categorical values. To increase performance one can also first perform label encoding then those integer variables to binary values which will become the most desired form of machine-readable

-Pandas get_dummies() converts categorical variables into dummy/indicator variables.

	gender	class	city
0	Male	A	Delhi
1	Female	B	Gurugram
2	Male	C	Delhi
3	Female	D	Delhi
4	Female	A	Gurugram

	A	B	C	D	E	F	G	H
1	Male	Female	A	B	C	D	Delhi`	Gurugram
2	1	0	1	0	0	0	1	0
3	0	1	0	1	0	0	0	1
4	1	0	0	0	1	0	1	0
5	0	1	0	0	0	1	1	0
6	0	1	1	0	0	0	0	1
7								

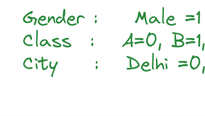
2. Label Encoding

Label encoding can uniquely number the different categories from 0 to n-1.

1. Thus also termed as Integer encoding. Label Encoder class from the scikit-learn library is used for this purpose.
2. LabelEncoder cannot handle missing values so it's important to impute them.
3. LabelEncoder can be used to store values using less disk space. This is simple to use and works well on tree-based algorithms.
4. It cannot work for linear models, SVMs, or neural networks as their data needs to be standardized.

Gender : Male =1 , Female =0
Class : A=0, B=1, C=2, D= 3
City : Delhi =0, Gurugram =

	gender	class	city
0	Male	A	Delhi
1	Female	B	Gurugram
2	Male	C	Delhi
3	Female	D	Delhi
4	Female	A	Gurugram



gender	class	city
1	0	0
0	1	1
1	2	0
0	3	0
0	0	1

Foundation of Natural Language Processing (NLP)

Text Data -------- d- dimension vector

There are several technique to convert text to vector:

1. BOW (Bag of words)
2. TF - IDF
3. W2V

1. BOW

Steps:

1. Construct a dictionary: a set of all the unique words in your reviews.
   (d- unique words)
   Ex. This pasta is very tasty and pasta is affordable.

   Unique words : [This, pasta, is, very, tasty, and, affordable]

2. write word count:

This	pasta	is	very	tasty	and	affordable
1	2	2	1	1	1	1

#Each word have diff dimension or number of times word occurred in text.

objective of BOW :

v1 - This pasta is very tasty and affordable.
v2 - This pasta is not tasty and is affordable.

	This	pasta	is	very	tasty	not	and	is	affordable
v1 :	1	1	1	1	1	0	1	0	1
v2 :	1	1	1	0	1	1	1	2	1

$$\text{Distance} = (v1 - v2) = \sqrt{(1)^2 + (1)^2 + (2)^2}$$

$$= \sqrt{6}$$

Problem With BOW : The sentence are opposite to each other but the distance between then is less. Its should be high.

Similarity ↑ Distance ↓

Stopwords:

- Pronoun, conjunction, Model verb, Article, Helping verb etc.
- Not useful words
- Remove stopwords

Stemming

- convert the words into basewords.

Ex: Taster, tasty, Tasteful → Tasty

Beautiful, beauty → Beauty

Algorithms used for stemming: 1. PorterStemmer
2. Snowballstemmer

Tokenization

- Breaking the sentences inti words.

Ex : This pasta is very tasty and pasta is affordable.

TF IDF(Term Frequency and Inverse document frequency)

Term Frequency :

$$TF = \frac{\text{Number of times Wi occurred in Vj}}{\text{Total Number of words in Vj}}$$

v1 - This pasta is very tasty and affordable.
v2 - This pasta is not tasty and is affordable.

	This	pasta	is	very	tasty	not	and	is	affordable
v1 :	1	1	1	1	1	0	1	0	1
v2 :	1	1	1	0	1	1	1	2	1

$$TF(\text{pasta}, v1) = \frac{1}{7}$$

$$TF(\text{is}, v2) = 2$$

$$0 <= TF(W_i, V_j) <= 1$$

Inverse Document Frequency (IDF) :

$$IDF(W_i, D_j) = \log(N/n_i)$$

N: Number of documents
n: Number of documents containing the word

Example :

Let's cover an example of 3 documents -

Document 1: It is going to rain today.
Document 2: Today I am not going outside.
Document 3: I am going to watch the season premiere.

Step 1 Clean data and Tokenize:

Word	Count
going	3
to	2
today	2
i	2
am	2
it	1
is	1
rain	1

Step 2 Find TF

Document 1— It is going to rain today.

Words/ Documents	Document 1
going	0.16
to	0.16
today	0.16
i	0
am	0
it	0.16
is	0.16
rain	0.16

TF for sentence 1

Continue for rest of sentences -

Words/ Documents	Document 1	Document 2	Document 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

Step 3 Find IDF

Words	IDF Value
going	log(3/3)
to	log(3/2)
today	log(3/2)
i	log(3/2)
am	log(3/2)
It	log(3/1)
is	log(3/1)
rain	log(3/1)

IDF for document

Step 4 Build model i.e. stack all words next to each other —

Words	IDF Value
going	0
to	0.41
today	0.41
i	0.41
am	0.41
It	1.09
is	1.09
rain	1.09

Words/ Documents	Document 1	Document 2	Document 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

IDF Value and TF value of 3 documents.

Step 5 Compare results and use table to ask questions

Words/ Documents	going	to	today	i	am	it	is	rain
Document 1	0	0.07	0.07	0	0	0.17	0.17	0.17
Document 2	0	0	0.07	0.07	0.07	0	0	0
Document 3	0	0.05	0	0.05	0.05	0	0	0

Remember, the final equation = TF-IDF = TF * IDF

Key Notes:

- More importance to rare words in my Dataset.
- More importance if a word is frequent in a particular documents
- TF and IDf both are kind of probabilities.
- No need of stemming stowords it will give zero values like is, am, the, etc.

Text Preprocessing Steps:

1. Begin with removing the HTML tags.
2. Remove punctuation or limited set of special characters like @ , " # etc.
3. check if the word is made up of English word letter and is not alpha numeric.
4. Check if the length of the word is greater then 2 (a it was researched that there is no adjective in 2 letter)
5. Convert the word to lowercase.
6. Remove stopwords.
7. Finally Snowball stemming the word(it was overserved to be better then porterstemming)\
8. TFIDFVectorizer() --- Class for TFIDF
9. CountVectorizer()

Naive Bays Algorithem

Conditional Probability :

Conditional probability is a measure of the probability of an event occurring given that another event has (by assumption, presumption, assertion, or evidence) occurred.

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

Probability of event A occured and event B occured

Probability of event A given B has occured

Probability of event B

P(B) != 0

Bays Theorem

Bayes' Theorem is a simple mathematical formula used for calculating conditional probabilities.

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

Probability of A occurring given evidence B has already occurred

Probability of B occurring given evidence A has already occurred

Probability of A occurring

Probability of B occurring

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

Y: class of the variable

X: dependent feature vector (of size n)

Naive Bays

-- Bayes' rule provides us with the formula for the probability of Y given some feature X. In real-world problems, we hardly find any case where there is only one feature.

-- When the features are independent, we can extend Bayes' rule to what is called Naive Bayes which assumes that the features are independent that means changing the value of one feature doesn't influence the values of other variables and this is why we call this algorithm "NAIVE"

$$P(Y = k|X1, X2 \ldots . Xn) = \frac{P(X1|Y = k) * P(X2|Y = k) \ldots \ldots * P(Xn|Y = k) * P(Y = k)}{P(X1) * P(X2) \ldots . * P(Xn)}$$

This formula can also be understood as

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Example :

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Frequency and Likelihood tables of 'Color'

Frequency Table

		Stolen?	
		Yes	No
Color	Red	3	2
	Yellow	2	3

Likelihood Table

		Stolen?	
		P(Yes)	P(No)
Color	Red	3/5	2/5
	Yellow	2/5	3/5