SENTIMENT ANALYSIS FOR MARKETING

Phase 3

Development part 1

Introduction:

Sentiment Analysis is a use case of **Natural Language Processing** (NLP) and comes under the category of **text classification**. To put it simply, Sentiment Analysis involves classifying a text into various sentiments, such as positive or negative, Happy, Sad or Neutral, etc. Thus, the ultimate goal of sentiment analysis is to decipher the underlying mood, emotion, or sentiment of a text. This is also known as **Opinion Mining**.

Loading data:

IN:

Import pandas as pd

Df = pd.read_csv("Tweeter.csv")

Load the dataset...

Print(df.head())

OP:

∞ tweet_id =	▲ airline_sen =	# airline_sen =	▲ negativere =	# negativere =	A airline =
570301031407624 196	negative	1.0	Bad Flight	0.7033	Virgin America
570300817074462 722	negative	1.0	Can't Tell	1.0	Virgin America
570300767074181 121	negative	1.0	Can't Tell	0.6842	Virgin America

∞ tweet_id Ξ	A airline_sen =	# airline_sen =	A negativere =	# negativere =	A airline =
57030613367776 513	00 neutral	1.0			Virgin America
570301130888122 positive 368		0.3486		0.0	Virgin America
57030108367281 571	3 neutral	0.6837			Virgin America
⇔ tweet_id =	A airline_sen =	# airline_sen =	▲ negativere =	# negativere =	∆ airline =
tweet_id = 570300767074181 121	A airline_sen =	# airline_sen = 1.0	▲ negativere =	# negativere = 0.6842	∆ airline = Virgin America
570300767074181	_	_	-	-	

As the dataset about tweeter in airline sentiment and have 16,424 datas,let check the few data in 6 columns.

Preprocessing dataset:

1. Text Cleaning (or) preprocessing:

Remove special characters, URLs, and other unwanted elements from the text.

IN[]:

Import re

```
Def clean_text(text):
```

Text = re.sub(r'http\S+', ", text) # Remove URLs

Text = $re.sub(r'[^A-Za-z0-9]+', '', text)$ # Remove special characters

Text = text.lower() # Convert to lowercase

Return text

Clean the dataset..,

Cleaned text = clean text(text)

OP[]:

⇔ tweet_id =	▲ airline_sen =	# airline_sen =	▲ negativere =	# negativere =	A airline <u></u>
570306133677760 513	neutral	1.0			Virgin America
570301130888122 368	positive	0.3486		0.0	Virgin America
570301083672813 571	neutral	0.6837			Virgin America

2. Tokenization:

Now we will tokenize all the cleaned tweets in our dataset. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens

IN[]:

From nltk.tokenize import word_tokenize

Def tokenize_text(text):

Tokens = word_tokenize(text)

Return tokens

```
Splits the datasets,...
```

```
tokenized_tweet = combi['tidy_tweet'].apply(lambda x:
x.split())
tokenized_tweet.head()
```

Tokens = tokenize_text(cleaned_text)

OP[]:

	airline_sentiment	Text
0	neutral	@VirginAmerica What @dhepburn said.
1	positive	VirginAmerica plus youve added commercials t
2	neutral	VirginAmerica I didnt today Must mean I n
3	negative	VirginAmerica its really aggressive to blast
4	negative	VirginAmerica and its a really big bad thing

3.Stop Word Removal:

Remove common stop words like"and"."the","is", that do not provide significant information.

IN[]:

From nltk.corpus import stopwords

Def remove_stopwords(tokens):

Stop_words = set(stopwords.words('english'))

Filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

Return filtered_tokens

Filtered_tokens = remove_stopwords(tokens)

OP[]:

∞ tweet_id =	▲ airline_sen =	# airline_sen =	▲ negativere =	# negativere =	A airline =
570306133677760 513	neutral	1.0			Virgin America
570301130888122 368	positive	0.3486		0.0	Virgin America
570301083672813 571	neutral	0.6837			Virgin America

4.Lemmatization:

Reduce words to their base form (lemmas). Remove prefixs and suffixs.

IN[]:

Import spacy

Nlp = spacy.load("en_core_web_sm")

Def lemmatize_text(text):

Doc = nlp(text)

Lemmatized_text = ' '.join([token.lemma_ for token in doc])

Return lemmatized_text

Lemmatized_text = lemmatize_text(" ".join(filtered_tokens))

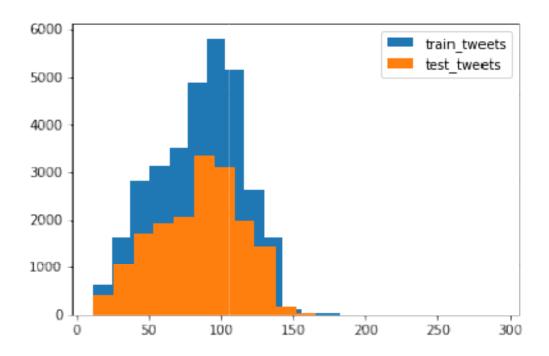
OP[]:

	tweet_ id	airli ne_s enti ment	airline_s entiment _confide nce	neg ativ erea son	negativ ereason _confid ence	ai rli ne	airline _senti ment_ gold	na me	negati verea son_g old	ret wee t_c oun t	text	twe et_ coo rd	twe et_c reat ed	twe et_l ocat ion	user _ti mez one
0	57030 61336 77760 513	neutr al	1.0000	Na N	NaN	Vi rg in A m er ic a	NaN	cai rdi n	NaN	0	@Vi rgin Ame rica Wha t @dh epbu rn said.	Na N	201 5- 02- 24 11: 35: 52- 080 0	Na N	East ern Tim e (US & Can ada)
1	57030 11308 88122 368	posit ive	0.3486	Na N	0.0000	Vi rg in A m er ic a	NaN	jna rdi no	NaN	0	@Vi rgin Ame rica plus you' ve adde d com merc ials t	Na N	201 5- 02- 24 11: 15: 59 - 080 0	Na N	Paci fic Tim e (US & Can ada)

	tweet_ id	airli ne_s enti ment	airline_s entiment _confide nce	neg ativ erea son	negativ ereason _confid ence	ai rli ne	airline _senti ment_ gold	na me	negati verea son_g old	ret wee t_c oun t	text	twe et_ coo rd	twe et_c reat ed	twe et_l ocat ion	user _ti mez one
2	57030 10836 72813 571	neutr al	0.6837	Na N	NaN	Vi rg in A m er ic a	NaN	yv on nal yn n	NaN	0	@Vi rgin Ame rica I didn' t toda y Must mea n I n	Na N	201 5- 02- 24 11: 15: 48- 080 0	Lets Play	Cen tral Tim e (US & Can ada)
3	57030 10314 07624 196	nega tive	1.0000	Bad Flig ht	0.7033	Vi rg in A m er ic a	NaN	jna rdi no	NaN	0	@Vi rgin Ame rica it's reall y aggr essiv e to blast 	Na N	201 5- 02- 24 11: 15: 36- 080 0	Na N	Paci fic Tim e (US & Can ada)
4	57030 08170 74462 722	nega tive	1.0000	Can' t Tell	1.0000										

In [2]:

train = pd.read_csv('train_E6oV3IV.csv')
test = pd.read_csv('test_tweets_airlines.csv')

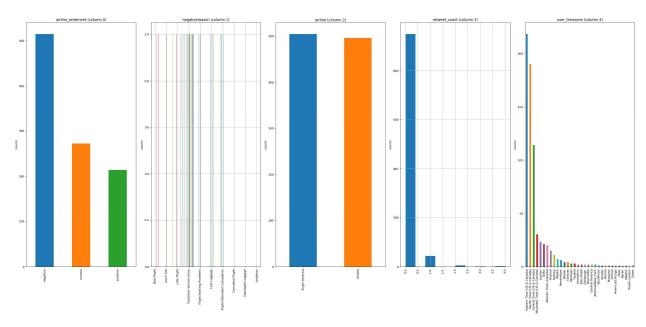


Data Inspection

Let's check out a few tweets.

In [3]:

train[train['label'] == 0].head(10)



In [18]: sent_data.head()

Out[18]:

	airline_sentiment	text
0	neutral	@VirginAmerica What @dhepburn said.
1	positive	@VirginAmerica plus you've added commercials t
2	neutral	@VirginAmerica I didn't today Must mean I n
3	negative	@VirginAmerica it's really aggressive to blast
4	negative	@VirginAmerica and it's a really big bad thing

Data preprocessing is a critical step in sentiment analysis as it lays the foundation for building effective sentiment classification models. By transforming raw text data into a clean, structured format, preprocessing helps in extracting meaningful features that reflect the sentiment expressed in the text.