SENTIMENT ANALYSIS IN MARKETING

PHASE-3

Loading and preprocessing datasets:

Data loading with pandas:

- ❖ Choose your content.
- ❖ Gather your dataset.
- ❖ Split your dataset.
- ❖ Train a machine learning model.
- ❖ Validate your model.
- ❖ Deploy your model.
- ❖ Monitor your models performance.

Loading data:

For example:

Loading data using python

```
import time
import pandas as pd
import numpy as np
import os
from IPython.display import display
from nimbusml.datasets import get_dataset
```

```
from nimbusml.feature_extraction.text import NGramFeaturizer
from nimbusml.feature_extraction.text.extractor import Ngram
from nimbusml.linear_model import AveragedPerceptronBinaryClassifier
from nimbusml.decomposition import PcaTransformer
from nimbusml import Pipeline
# Load data from package
trainDataFile = get_dataset('gen_twittertrain').as_filepath()
testDataFile = get_dataset('gen_twittertest').as_filepath()
print("Train data file path: " + str(os.path.basename(trainDataFile)))
print("Test data file path: " + str(os.path.basename(testDataFile)))

trainData = pd.read_csv(trainDataFile, sep = "\t")
testData = pd.read_csv(testDataFile, sep = "\t")

trainData.head()
```

```
Train data file path: train-twitter.gen-sample.tsv
Test data file path: test-twitter.gen-sample.tsv
```

Sentiment	Text	Label	
0	Negative	Oh you are hurting me	0
1	Positive	So long	1
2	Positive	Ths sofa is comfortable	1
3	Negative	The place suck. No?	0
4	Positive	@fakeid "Chillin" I love it!!	1

Processing dataset :

The NGramFeaturizer transform produces a bag of counts of sequences of consecutive words, called n-grams, from a given corpus of text. The word counts are then normalized using term frequency-inverse document frequency (TF-IDF) method.

In NimbusML, the user can specify the input column names for each operator to be executed on. If not, all the columns from the previous operator or the origin dataset will be used. In , the column syntax of nimbusml will be discussed in more details.

For text featurizer, since the output has multiple columns, for visualization, the names for those will become "output_col_name.[word sequence] " to represent the count for word sequence [word sequence] after normalization. In this example, we train the model with only one column, column "Text".

For example using python:

featurizer = NGramFeaturizer(word_feature_extractor=Ngram(weighting = 'TfIdf'))

Then we can call .fit_transform() to train the featurizer.

text_transformed = featurizer.fit_transform(trainData["Text"].to_frame()) # Using one column as input

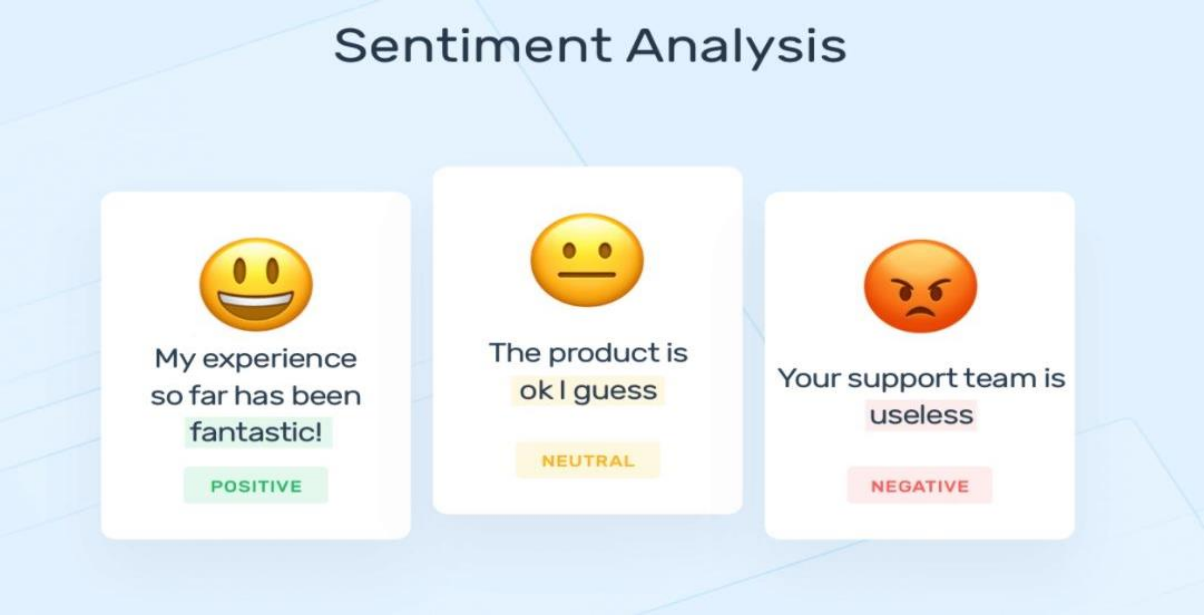
print(text_transformed.shape)

Major steps in preprocessing:

- *Acquire the dataset.*
- *Import all crucial lilibrarie*
- *Import the dataset.*
- *Identifying and handling the mossing values.*
- *Encoding the categorial data.*
- *Splitting the dataset.*
- *Feature scaling.*

Methods in preprocessing:

- ✓ **Data quality assessment.**
- ✓ **Data cleaning.**
- ✓ **Data transformation.**
- ✓ **Data reduction.**



The above diagram is an example for loading data in marketing.

Example for sentiment analysis: