Introduction to NLP feature engineering

FEATURE ENGINEERING FOR NLP IN PYTHON

Rounak Banik
Data Scientist

datacamp

Numerical data

Iris dataset

sepal length	sepal width	petal length	petal width	class
6.3	2.9	5.6	1.8	Iris-virginica
4.9	3.0	1.4	0.2	Iris-setosa
5.6	2.9	3.6	1.3	Iris-versicolor
6.0	2.7	5.1	1.6	Iris-versicolor
7.2	3.6	6.1	2.5	Iris-virginica

One-hot encoding

sex
female
male
female
male
female
...

One-hot encoding

sex	one-hot encoding
female	→
male	→
female	→
male	→
female	→
...	...

One-hot encoding

sex	one-hot encoding	sex_female	sex_male
female	→	1	0
male	→	0	1
female	→	1	0
male	→	0	1
female	→	1	0
...	...	...	...

One-hot encoding with pandas

```
# Import the pandas library
import pandas as pd

# Perform one-hot encoding on the 'sex' feature of df
df = pd.get_dummies(df, columns=['sex'])
```

Textual data

Movie Review Dataset

review	class
This movie is for dog lovers. A very poignant...	positive
The movie is forgettable. The plot lacked...	negative
A truly amazing movie about dogs. A gripping...	positive

Text pre-processing

- Converting to lowercase
  - **Example:** `Reduction` to `reduction`
- Converting to base-form
  - **Example:** `reduction` to `reduce`

Vectorization

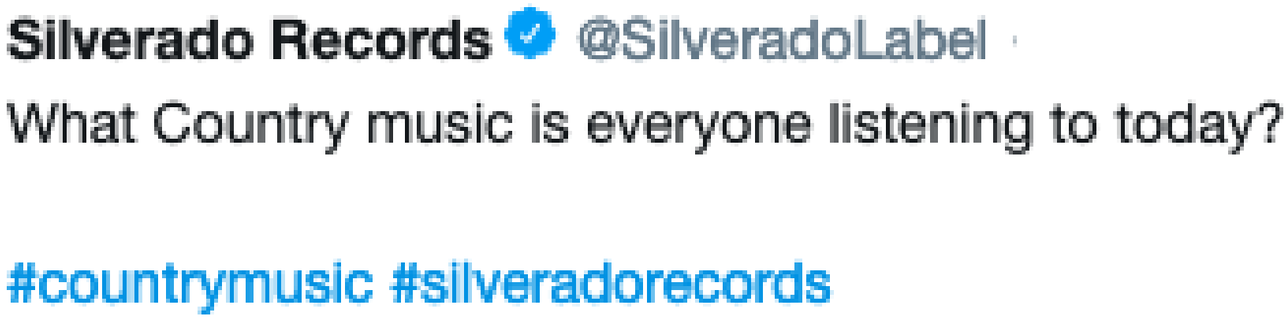
review	class
This movie is for dog lovers. A very poignant...	positive
The movie is forgettable. The plot lacked...	negative
A truly amazing movie about dogs. A gripping...	positive

Vectorization

0	1	2	...	n	class
0.03	0.71	0.00	...	0.22	positive
0.45	0.00	0.03	...	0.19	negative
0.14	0.18	0.00	...	0.45	positive

Basic features

- Number of words
- Number of characters
- Average length of words
- Tweets



POS tagging

Word	POS
I	Pronoun
have	Verb
a	Article
dog	Noun

Named Entity Recognition

- Does noun refer to person, organization or country?

TED

Noun	NER
Brian	**Person**
DataCamp	**Organization**

Concepts covered

- Text Preprocessing
- Basic Features
- Word Features
- Vectorization

Let's practice!

FEATURE ENGINEERING FOR NLP IN PYTHON

Basic feature extraction

FEATURE ENGINEERING FOR NLP IN PYTHON

Rounak Banik
Data Scientist

datacamp

Number of characters

```
"I don't know." # 13 characters
```

```
# Compute the number of characters
text = "I don't know."
num_char = len(text)

# Print the number of characters
print(num_char)
```

```
13
```

```
# Create a 'num_chars' feature
df['num_chars'] = df['review'].apply(len)
```

Number of words

```
# Split the string into words
text = "Mary had a little lamb."
words = text.split()

# Print the list containing words
print(words)
```

```
['Mary', 'had', 'a', 'little', 'lamb.']
```

```
# Print number of words
print(len(words))
```

```
5
```

Number of words

```
# Function that returns number of words in string
def word_count(string):
    # Split the string into words
    words = string.split()

    # Return length of words list
    return len(words)
```

```
# Create num_words feature in df
df['num_words'] = df['review'].apply(word_count)
```

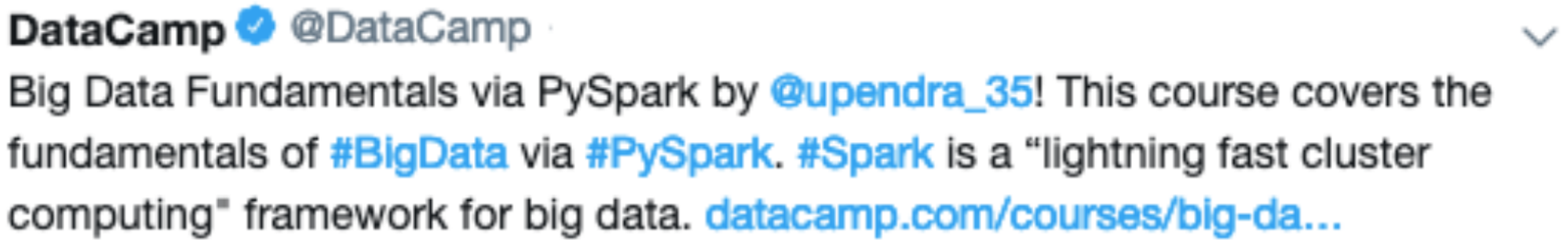
Average word length

```
#Function that returns average word length
def avg_word_length(x):
    # Split the string into words
    words = x.split()
    # Compute length of each word and store in a separate list
    word_lengths = [len(word) for word in words]
    # Compute average word length
    avg_word_length = sum(word_lengths)/len(words)
    # Return average word length
    return(avg_word_length)
```

Average word length

```
# Create a new feature avg_word_length
df['avg_word_length'] = df['review'].apply(doc_density)
```

Special features



Hashtags and mentions

```
# Function that returns number of hashtags
def hashtag_count(string):
    # Split the string into words
    words = string.split()
    # Create a list of hashtags
    hashtags = [word for word in words if word.startswith('#')]
    # Return number of hashtags
    return len(hashtags)
```

```
hashtag_count("@janedoe This is my first tweet! #FirstTweet #Happy")
```

```
2
```

Other features

- Number of sentences
- Number of paragraphs
- Words starting with an uppercase
- All-capital words
- Numeric quantities

Let's practice!

FEATURE ENGINEERING FOR NLP IN PYTHON

Readability tests

FEATURE ENGINEERING FOR NLP IN PYTHON

Rounak Banik
Data Scientist

datacamp

Overview of readability tests

- Determine readability of an English passage
- Scale ranging from primary school up to college graduate level
- A mathematical formula utilizing word, syllable and sentence count
- Used in fake news and opinion spam detection

Readability text examples

- Flesch reading ease
- Gunning fog index
- Simple Measure of Gobbledygook (SMOG)
- Dale-Chall score

Readability test examples

- **Flesch reading ease**
- **Gunning fog index**
- Simple Measure of Gobbledygook (SMOG)
- Dale-Chall score

Flesch reading ease

- One of the oldest and most widely used tests
- **Greater the average sentence length, harder the text is to read**
  - "This is a short sentence."
  - "This is longer sentence with more words and it is harder to follow than the first sentence."
- **Greater the average number of syllables in a word, harder the text is to read**
  - "I live in my home."
  - "I reside in my domicile."
- Higher the score, greater the readability

Flesch reading ease score interpretation

Reading ease score	Grade Level
90-100	5
80-90	6
70-80	7
60-70	8-9
50-60	10-12
30-50	College
0-30	College Graduate

Gunning fog index

- Developed in 1954
- Also dependent on average sentence length
- Greater the percentage of complex words, harder the text is to read
- Higher the index, lesser the readability

Gunning fog index interpretation

Fog index	Grade level
17	College graduate
16	College senior
15	College junior
14	College sophomore
13	College freshman
12	High school senior
11	High school junior

Fog index	Grade level
10	High school sophomore
9	High school freshman
8	Eighth grade
7	Seventh grade
6	Sixth grade

The readability library

```
# Download nltk punkt module
import nltk
nltk.download('punkt_tab')
```

```
# Import the Readability class
from readability import Readability

# Create a Readability Object
readability_scores = Readability(text)

# Generate scores
gf = readability_scores.gunning_fog()
print(gf.score())
```

```
16.26
```

Let's practice!

FEATURE ENGINEERING FOR NLP IN PYTHON