AULA 11- Exercício teórico PLN

November 16, 2023

Vitor Albuquerque de Paula

1 Considere o seguinte conjunto de treinamento. Classifique com kNN (k = 1) a sentença:

"I always like foreign films". Compare a distância Cosseno com Euclidiana.

Classe	Texto
Negativo (-)	Just plain boring
Negativo (-)	Entirely predictable and lacks energy
Negativo (-)	No surprises and very few laughs
Positivo (+)	Very powerful
Positivo (+)	The most fun film of the summer

[8]:
```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity, euclidean_distances
import numpy as np

# Sentenças do conjunto de treinamento e a sentença a ser classificada
sentences = [
    "Just plain boring",
    "Entirely predictable and lacks energy",
    "No surprises and very few laughs",
    "Very powerful",
    "The most fun film of the summer",
    "I always like foreign films"
]

# Vectorizando as sentenças
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(sentences).toarray()

# Calculando as distâncias para a sentença a ser classificada
target_sentence_vector = X[-1]
training_vectors = X[:-1]
```

```
# Distância Cosseno
cosine_distances = cosine_similarity([target_sentence_vector], training_vectors)

# Distância Euclidiana
euclidean_distances = euclidean_distances([target_sentence_vector],␣
 ↪training_vectors)

(cosine_distances, euclidean_distances)
```

```
Frase: 'Just plain boring'
   Distância Cosseno: 0.0
   Distância Euclidiana: 2.6457513110645907
Frase: 'Entirely predictable and lacks energy'
   Distância Cosseno: 0.0
   Distância Euclidiana: 3.0
Frase: 'No surprises and very few laughs'
   Distância Cosseno: 0.0
   Distância Euclidiana: 3.1622776601683795
Frase: 'Very powerful'
   Distância Cosseno: 0.0
   Distância Euclidiana: 2.449489742783178
Frase: 'The most fun film of the summer'
   Distância Cosseno: 0.0
   Distância Euclidiana: 3.605551275463989
```

A análise das distâncias entre a sentença "I always like foreign films" e as sentenças do conjunto de treinamento revelou os seguintes resultados:

Distância Cosseno: Todas as distâncias coseno são 0, indicando que, sob essa métrica, a sentença de interesse não tem semelhança com nenhuma das sentenças do conjunto de treinamento. Isso pode ser devido à escolha de palavras únicas na vetorização, que não captura bem a semelhança semântica.

Distância Euclidiana: As distâncias variam, com a menor distância sendo para a sentença "Very powerful" (2.44948974). Portanto, usando a distância Euclidiana, a sentença "I always like foreign films" seria classificada como "Positivo (+)".

Devido à discrepância entre as métricas, é importante notar que a escolha da técnica de vetorização e a métrica de distância podem ter um impacto significativo nos resultados do kNN. A distância Cosseno é geralmente preferida para dados de texto, pois é mais sensível ao ângulo entre os vetores (ou seja, a direção das palavras no espaço vetorial) do que à sua magnitude. No entanto, a vetorização simples usada aqui pode não ser adequada para capturar a semântica das sentenças de forma eficaz. Métodos mais avançados, como TF-IDF ou incorporações de palavras (word embeddings), poderiam oferecer uma representação mais rica e talvez resultar em uma classificação mais precisa.

2 Considere o seguinte conjunto de treinamento. Classifique com Naive Bayes a sentença: "eu gosto deste lugar"

Classe	Texto
Negativo (-)	eu não gosto deste restaurante
Negativo (-)	estou cansado dessas coisas
Positivo (+)	eu me sinto bem com essas cervejas
Positivo (+)	eu amo esse sanduíche
Positivo (+)	este é um lugar incrível!

[7]:
```
from collections import defaultdict
import math

# Dados de treinamento
training_data = [
    ("Negativo", "eu não gosto deste restaurante"),
    ("Negativo", "estou cansado dessas coisas"),
    ("Positivo", "eu me sinto bem com essas cervejas"),
    ("Positivo", "eu amo esse sanduíche"),
    ("Positivo", "este é um lugar incrível!")
]

# Tokenização e contagem de frequência de palavras por classe
word_freq = defaultdict(lambda: defaultdict(int))
class_count = defaultdict(int)

for label, text in training_data:
    class_count[label] += 1
    for word in text.split():
        word_freq[label][word] += 1

# Cálculo das probabilidades
total_samples = sum(class_count.values())
class_probabilities = {label: count / total_samples for label, count in␣
 ↪class_count.items()}

# Probabilidades condicionais de palavras dadas as classes
word_probabilities = {}
for label in word_freq:
    total_words = sum(word_freq[label].values())
    word_probabilities[label] = {word: (count / total_words) for word, count in␣
 ↪word_freq[label].items()}

# Classificando a nova sentença
new_sentence = "eu gosto deste lugar"
new_sentence_words = new_sentence.split()

# Probabilidades de cada classe para a nova sentença
sentence_probabilities = {}
```

```
for label in class_probabilities:
    sentence_prob = math.log(class_probabilities[label])  # Uso do log para␣
 ↪evitar underflow
    for word in new_sentence_words:
        # Adicionando uma suavização para evitar multiplicação por zero se a␣
 ↪palavra não estiver presente
        word_prob = word_probabilities[label].get(word, 1 / (total_words +␣
 ↪len(word_freq[label])))
        sentence_prob += math.log(word_prob)
    sentence_probabilities[label] = sentence_prob

sentence_probabilities
```

```
A sentença 'eu gosto deste lugar' foi classificada como: Negativo
```

De acordo com o modelo Naive Bayes treinado com os dados fornecidos, a sentença "eu gosto deste lugar" é classificada como Negativo (-). Isso é determinado pela comparação das probabilidades logarítmicas, onde a sentença obteve uma maior probabilidade de pertencer à classe Negativa em comparação com a classe Positiva.

É importante notar que o Naive Bayes assume independência entre as características (neste caso, palavras), o que nem sempre é verdadeiro em dados de linguagem natural. Além disso, a ausência de algumas palavras na classe Positiva pode ter influenciado o resultado