

ПИиКТ

Современные инструменты анализа данных

Лабораторная работа №2

Выполнили: Нуруллаев Даниил Романович и

Балтабаев Дамир Темиржанович

Группы:P3214 и P3210

Преподаватель: Кустова Е. С

1)Обучим модель на тестовых данных

import pandas as pd

data = pd.read\_csv("/content/sample\_data/dataset.csv")

columns = data[['genre', 'lyrics']]

columns = columns[(columns.genre == 'Country') | (columns.genre == 'R&B')]

2) Приведем к нижнему регистру

lowered = columns['lyrics'].str.lower()

columns['lowered'] = lowered

3) Токенизация

import nltk

from nltk.tokenize import word\_tokenize

nltk.download('punkt')

tokened = columns.apply(lambda row: nltk.word\_tokenize(row['lowered']), axis=1)

columns['tokened'] = tokened

4) Удаляем стоп-слова

nltk.download('stopwords')

from nltk.corpus import stopwords

noise = stopwords.words('english')

withoutstop = columns['tokened'].apply(lambda x: [item for item in x if item not in noise])

without\_stop = []

for a in withoutstop:

  without\_stop.append(", ".join(a))

columns['without\_stop'] = without\_stop

5) Лемматизация

from nltk.stem import WordNetLemmatizer

nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()

lemmatized = columns['without\_stop'].apply(lambda x: [lemmatizer.lemmatize(x)])

lemma = []

for a in lemmatized:

  lemma.append(", ".join(a))

columns['lemmatized'] = lemma

6) Делим данные на обучающую и тестовую сборки

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(columns.lemmatized, columns.genre, train\_size = 0.7)

//x\_train - тексты, на которых мы обучаем модель. В данном случае мы используем столбец lemmatized, так как он содержит данные, прошедшие все этапы подготовки

y\_train - жанры, соответствующие текстам, на которых модель обучается - столбец genre

x\_test - точно такие же тексты из набора данных, на которых мы будем проверять, насколько модель научилась предсказывать жанр

y\_test - жанры, соответствующие x\_test. т.е. мы смотрим, насколько предсказания соответствуют содержимому этой переменной, чтобы оценить качество обучения

КодТекст

7) Векторизация

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer = CountVectorizer(ngram\_range=(1, 3))

vectorized\_x\_train = vectorizer.fit\_transform(x\_train)

//n = 1 - униграмма (для обучения используются слова по отдельности)

n = 2 - биграмма (для обучения используются пары слов)

n = 3 - триграмма (для обучения используются тройки слов)

8) Классификация

//Для задачи классификации используем Наивный Байесовский Классификатор - простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими предположениями о независимости.

#импортируем байесовский классификатор

from sklearn.naive\_bayes import MultinomialNB

clf = MultinomialNB()

clf.fit(vectorized\_x\_train, y\_train)

# тестовую выборку просто векторизировали

vectorized\_x\_test = vectorizer.transform(x\_test)

Изображение выглядит как текст

Автоматически созданное описание

Оценка классификации

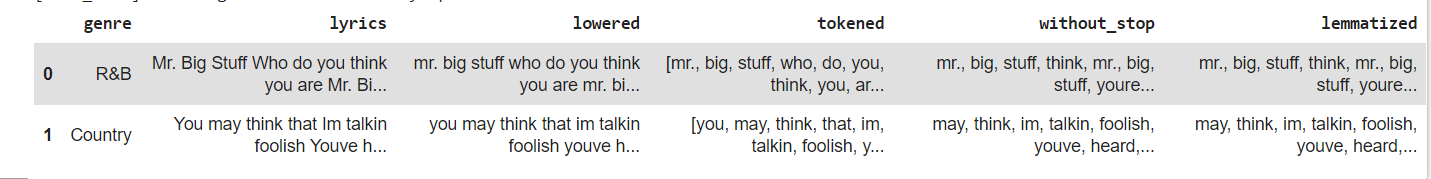
Изображение выглядит как стол

Автоматически созданное описание

2 Задание

Изображение выглядит как текст

Автоматически созданное описание



Векторизуем тексты песен и передаем их обученной модели

Изображение выглядит как текст

Автоматически созданное описание

Оценка классификации

Изображение выглядит как стол

Автоматически созданное описание

3 Задание

data = pd.read\_csv("/content/sample\_data/dataset-lyrics-musics-mini.csv")

columns = data[['cantorNome', 'letra']]

columns = columns[(columns.cantorNome == 'david-bowie') | (columns.cantorNome == 'paul-mccartney')]

lowered = columns['letra'].str.lower()

columns['lowered'] = lowered

nltk.download('punkt')

tokened = columns.apply(lambda row: nltk.word\_tokenize(row['lowered']), axis=1)

columns['tokened'] = tokened

nltk.download('stopwords')

noise = stopwords.words('english')

withoutstop = columns['tokened'].apply(lambda x: [item for item in x if item not in noise])

without\_stop = []

for a in withoutstop:

  without\_stop.append(", ".join(a))

columns['without\_stop'] = without\_stop

nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()

lemmatized = columns['without\_stop'].apply(lambda x: [lemmatizer.lemmatize(x)])

lemma = []

for a in lemmatized:

  lemma.append(", ".join(a))

columns['lemmatized'] = lemma

columns



Предсказания для выборки

x\_train, x\_test, y\_train, y\_test = train\_test\_split(columns.lemmatized, columns.cantorNome, train\_size = 0.7)

vectorized\_x\_test = vectorizer.transform(columns['letra'])

vectorizer = CountVectorizer(ngram\_range=(1, 3))

vectorized\_x\_train = vectorizer.fit\_transform(x\_train)

clf = MultinomialNB()

clf.fit(vectorized\_x\_train, y\_train)

# тестовую выборку просто векторизировали

vectorized\_x\_test = vectorizer.transform(x\_test)

clf.predict(vectorized\_x\_test)

array(['david-bowie', 'david-bowie', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'paul-mccartney',

'david-bowie', 'david-bowie', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney', 'david-bowie',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'david-bowie', 'paul-mccartney', 'david-bowie', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'david-bowie', 'david-bowie',

'david-bowie', 'david-bowie', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney', 'david-bowie',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'david-bowie', 'paul-mccartney', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'paul-mccartney', 'david-bowie',

'david-bowie', 'david-bowie', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney',

'paul-mccartney', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'david-bowie', 'david-bowie', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'paul-mccartney', 'paul-mccartney',

'david-bowie', 'david-bowie', 'david-bowie', 'david-bowie',

'david-bowie', 'paul-mccartney', 'david-bowie', 'david-bowie',

'paul-mccartney', 'david-bowie', 'paul-mccartney', 'david-bowie',

'paul-mccartney', 'david-bowie', 'david-bowie'], dtype='<U14')

Оценка классификации Изображение выглядит как стол

Автоматически созданное описание

Вывод: В ходе выполнения данной лабораторной работы, мы познакомились с основами анализа текстовой информации. Мы познакомились с этапами предварительной подготовки данных, также применили машинное обучение для задачи классификации