

Приложение статьи.

Часть сервера:

```
from fastapi import FastAPI
import pickle
from pydantic import BaseModel
import pandas as pd
import numpy as np
import string
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pymorphy3

app = FastAPI()

with open('model_lr.pkl', 'rb') as file:
    model = pickle.load(file)

with open('vectorizer1.pkl', 'rb') as file:
    vectorizer = pickle.load(file)

def fun_punctuation_text(text):
    text = text.lower()
    text = ''.join([ch for ch in text if ch not in string.punctuation])
    text = ''.join([i if not i.isdigit() else ' ' for i in text])
    text = ''.join([i if i.isalpha() else ' ' for i in text])
    text = re.sub(r'\s+', ' ', text, flags=re.I)
    text = re.sub('[a-z]', '', text, flags=re.I)
    st = '>\xa0'
    text = ''.join([ch if ch not in st else ' ' for ch in text])
    return text

def fun_lemmatizing_text(text):
    tokens = word_tokenize(text)
    res = list()
    for word in tokens:
        p = pymorphy3.MorphAnalyzer(lang='ru').parse(word)[0]
        res.append(p.normal_form)
    text = " ".join(res)
    return text

def fun_tokenize(text):
    nltk.download('stopwords')
    russian_stopwords = nltk.corpus.stopwords.words('russian')
    russian_stopwords.extend(['и', 'в', 'во', 'не', 'что', 'он', 'на', 'я',
                              'с', 'со', 'как', 'а', 'то', 'все',
                              'она', 'так', 'его', 'но', 'да', 'ты', 'к',
                              'у', 'же', 'вы', 'за', 'бы', 'по', 'только',
                              'ее', 'мне', 'было', 'вот', 'от', 'меня',
                              'еще', 'нет', 'о', 'из', 'ему', 'теперь',
                              'когда', 'даже', 'ну', 'вдруг', 'ли', 'если',
                              'уже', 'или', 'ни', 'быть', 'был', 'него',
                              'до', 'вас', 'нибудь', 'опять', 'уж', 'вам',
                              'ведь', 'там', 'потом', 'себя', 'ничего', 'ей',
                              'может', 'они', 'тут', 'где', 'есть', 'надо',
```

```

'ней', 'для', 'мы', 'тебя', 'их', 'чем', 'была',
    'сам', 'чтоб', 'без', 'будто', 'чего', 'раз',
'тоже', 'себе', 'под', 'будет', 'ж', 'тогда',
    'кто', 'этот', 'того', 'потому', 'этого',
'какой', 'совсем', 'ним', 'здесь', 'этом', 'один',
    'почти', 'мой', 'тем', 'чтобы', 'нее',
'сейчас', 'были', 'куда', 'зачем', 'всех', 'никогда',
    'можно', 'при', 'наконец', 'два', 'об',
'другой', 'хоть', 'после', 'над', 'больше', 'тот',
    'через', 'эти', 'нас', 'про', 'всего', 'них',
'какая', 'много', 'разве', 'три', 'эту', 'моя',
    'впрочем', 'хорошо', 'свою', 'этой', 'перед',
'иногда', 'лучше', 'чуть', 'том', 'нельзя', 'такой',
    'им', 'более', 'всегда', 'конечно', 'всю',
'между']]
    t = word_tokenize(text)
    tokens = [token for token in t if token not in russian_stopwords]
    text = " ".join(tokens)
    return text

def fun_pred_text(text):
    text = fun_punctuation_text(text)
    text = fun_lemmatizing_text(text)
    text = fun_tokenize(text)
    return text

def predict_cluster(text):
    cluster_description = {
        0: "0 - Облачные технологии и IT-инфраструктура",
        1: "1 - Образование в IT и искусственном интеллекте",
        2: "2 - Технологии в образовании и медицине",
        3: "3 - Креативные агентства и дизайн",
        4: "4 - Креативные индустрии и корпоративные проекты",
        5: "5 - Облачные сервисы и ИИ-разработки",
        6: "6 - Киберспорт и IT-сервисы"
    }

    processed_text = fun_pred_text(text)
    text_vectorized = vectorizer.transform([processed_text])

    cluster_prediction = model.predict(text_vectorized)[0]
    probabilities = model.predict_proba(text_vectorized)[0]
    main_cluster = cluster_description[cluster_prediction]

    probability_lines = [
        f"{cluster_description[cluster_idx]}- {prob:.2f}"
        for prob, cluster_idx in sorted(
            zip(probabilities, range(len(probabilities))),
            key=lambda x: -x[0]
        )
    ]

    return main_cluster, "\n".join(probability_lines)

class Item(BaseModel):
    text: str

@app.post("/predict")
def post_pred_text(item: Item):
    return {'cluster': predict_cluster(item.text)}

```

Часть клиента:

```
import streamlit as st
import requests
import os

st.set_page_config(
    page_title="Предсказание вероятности тем статей",
    page_icon="📄",
)

os.environ['HTTP_PROXY'] = ''
os.environ['HTTPS_PROXY'] = ''

st.title("📄 Предсказание тематического кластера статей")
input_text = st.text_area("Введите описание статьи", height=200)

if st.button("Предсказать"):
    if input_text == "":
        st.write("Введите описание фильма")
    else:
        with st.spinner("Анализируем описание..."):
            data = {"text": input_text}
            url = "http://127.0.0.1:8000/predict"
            response = requests.post(url, json=data)
            result = response.json()
            clust = result.get("cluster")

            st.markdown(f"""
            #### Предсказанный кластер
            """)
            st.write(f"Кластер: {clust[0]}")

            st.markdown(f"""
            #### Вероятности тем
            """)
            st.text(clust[1])
```

Результат:

Статья с сайта хабр:



Предсказание тематического кластера статей

Введите описание статьи

Привет! Меня зовут Андреева Саша, я веб-разработчик в компании iSpring. Два года назад мы столкнулись с тем, что всё больше и больше страниц нашего ведущего сайта начали падать в выдаче, а рост органического трафика заметно уменьшился. Основной причиной было то, что просели показатели сайта — мы постоянно обновляли страницы, но не уделяли должное внимание оптимизации. Мы проанализировали показатели, выбрали инструмент для отслеживания данных по всем страницам, в нём же настроили алерты и взялись за активную работу по оптимизации.

Перед вами рабочий чек-лист, в нём собраны основные наработки и советы по оптимизации, которые мы реализовали и продолжаем применять. Если вы работаете с CMS, то помимо перечисленных, есть дополнительные способы улучшить показатели — спрашивайте в комментариях.

Предсказать

Предсказанный кластер

Кластер: 1 - Образование в IT и искусственном интеллекте

Вероятности тем

- 1 - Образование в IT и искусственном интеллекте- 0.29
- 3 - Креативные агентства и дизайн- 0.20
- 2 - Технологии в образовании и медицине- 0.16
- 5 - Облачные сервисы и ИИ-разработки- 0.10
- 6 - Киберспорт и IT-сервисы- 0.10
- 0 - Облачные технологии и IT-инфраструктура- 0.08
- 4 - Креативные индустрии и корпоративные проекты- 0.06

Статья из Pdf файла:

Предсказание тематического кластера статей

Введите описание статьи

Можно аннотировать типы аргументов и возвращаемого значения функции, чтобы сделать код более читаемым и понятным. Например, `def add(a: int, b: int) → int`: чётко говорит, что оба аргумента должны быть `int`, а результат тоже будет `int`. Такие аннотации помогают статическим анализаторам, вроде `mypy`, находить ошибки до выполнения кода. Начнём с простого примера. Есть функция, которая принимает два числа и возвращает их сумму

Предсказать

Предсказанный кластер

Кластер: 2 - Технологии в образовании и медицине

Вероятности тем

- 2 - Технологии в образовании и медицине- 0.48
- 1 - Образование в IT и искусственном интеллекте- 0.19
- 3 - Креативные агентства и дизайн- 0.09
- 0 - Облачные технологии и IT-инфраструктура- 0.08
- 5 - Облачные сервисы и ИИ-разработки- 0.08
- 6 - Киберспорт и IT-сервисы- 0.06
- 4 - Креативные индустрии и корпоративные проекты- 0.04

Статья из Json файла:



Предсказание тематического кластера статей

Введите описание статьи

В прошлой статье, рассказывая про GPT-J-6B, я упоминал, что современные алгоритмы обработки естественного языка вызывают немалый ажиотаж даже среди людей, мало слышащих про машинное обучение. И вот, не успел ещё стихнуть шум обсуждений про возможности GPT-3 от OpenAI, как в начале 2021-го года нам показали ещё одну работу их команды в области ИИ, которую назвали в честь Сальвадора Дали и робота ВАЛЛ-И – DALL-E. Архитектурно DALL-E это версия GPT-3, к которой был добавлен хитрый способ токенизации изображений, позволяющий создавать мультимодальный словарь, в котором часть токенов отвечает за текст, а вторая часть за изображение. Что означает мультимодальность? Это модальность в разных сочетаниях, таких как: видео и текст, аудио и текст. Таким образом, вы можете представить это себе как классическую задачу для компьютера в сфере искусственного интеллекта, который может обрабатывать происходящее на изображении, интерпретировать и описывать все происходящие события, учитывая фон, изменения положений вещей в пространстве и контекст происходящего.

Предсказать

Предсказанный кластер

Кластер: 5 - Облачные сервисы и ИИ-разработки

Вероятности тем

- 5 - Облачные сервисы и ИИ-разработки- 0.43
- 2 - Технологии в образовании и медицине- 0.17
- 1 - Образование в IT и искусственном интеллекте- 0.13
- 0 - Облачные технологии и IT-инфраструктура- 0.07
- 3 - Креативные агентства и дизайн- 0.07
- 6 - Киберспорт и IT-сервисы- 0.06
- 4 - Креативные индустрии и корпоративные проекты- 0.06