

FastAPI ile Haber Metni Kategorilendirme Servisi Geliřtirme

CAN ARI

Dosya Yapısı ve Genel Mimari

1 app/ – FastAPI Servisinin Ana Kodları

- API servisinin tüm kodlarını içerir.
- main.py dosyası bulunur

3 venv/ – Sanal Ortam (Virtual Environment)

- projenin bağımlılıklarını içeren Python sanal ortamını barındırır.

5 app.py – Streamlit Kullanıcı Arayüzü

- tahmin servisini bir web arayüzü ile erişilebilir hale getiren Streamlit uygulamasını içerir.

2 data/ – Modelin Eğitildiği Veri Seti

- Makine öğrenmesi modelimizin eğitildiği ham verilerin olduğu “news_data.csv” dosyası bulunur

4 news_analysis.ipynb – Model Eğitim Süreci

- makine öğrenmesi modelimizin eğitim sürecini içeren Jupyter Notebook dosyasıdır.

FastAPI Servisinin Çalışma Mantığı

```
1 from fastapi import FastAPI, HTTPException
2 from fastapi.middleware.cors import CORSMiddleware
3 from pydantic import BaseModel
4 import joblib
5 import os
6 import numpy as np
```

- ✓ FastAPI: API oluşturmak için kullanıyoruz.
- ✓ HTTPException: Hata yönetimi için kullanılıyor.
- ✓ CORS Middleware:
Tarayıcıdan API'ye gelen istekleri kontrol etmek için.
- ✓ Pydantic BaseModel:
Giriş ve çıkış verilerini şemaya uygun hale getirmek için.
- ✓ Joblib:
Önceden eğitilmiş makine öğrenmesi modelini yüklemek için.
- ✓ OS Modülü:
Model dosyalarının yollarını belirlemek için kullanılıyor.
- ✓ NumPy: Veri işlemleri için

```
8 # FastAPI app nesnesi
9 app = FastAPI()
10
11 # CORS Middleware
12 app.add_middleware(
13     CORSMiddleware,
14     allow_origins=["*"],
15     allow_credentials=True,
16     allow_methods=["*"],
17     allow_headers=["*"],
18 )
```

- ✓ FastAPI uygulamasının bir örneğini oluşturuldu.
- ✓ farklı kaynaklardan gelen web isteklerine izin vermek için CORS (Cross-Origin Resource Sharing), kullanıldı.
- ✓ herhangi bir web sitesinin API'ye istek göndermesine izin vermek için allow_origins=["*"] ifadesi yazıldı.

FastAPI Servisinin Çalışma Mantığı

```
20 # Model ve vectorizer yüklemek
21 current_dir = os.path.dirname(os.path.abspath(__file__))
22 model_path = os.path.join(current_dir, '..', 'model', 'news_classifier_model.pkl')
23 vectorizer_path = os.path.join(current_dir, '..', 'model', 'tfidf_vectorizer.pkl')
24
25 model = joblib.load(model_path)
26 vectorizer = joblib.load(vectorizer_path)
27
```

✓ Önceden eğitilmiş model (news_classifier_model.pkl) ve TF-IDF vektörizer (tfidf_vectorizer.pkl) yükleniyor.

```
28 # Giriş ve çıkış modelleri
29 class PredictionInput(BaseModel):
30     | text: str
31
32 class PredictionOutput(BaseModel):
33     | category: str
34
```

✓ Pydantic modelleri kullanıldı :

1. PredictionInput Giriş (Input) Modeli

- Bu model, doğru veri tipi olmasını sağlar ve verilerin doğru yapıya uygunluğunu denetler.

2. PredictionOutput Çıkış (Output) Modeli

- API'den dönecek olan çıktıyı tanımlar. Burada dönecek verinin kategori bilgisini string formatında döndürüyoruz.

FastAPI Servisinin Çalışma Mantığı

Root Endpoint ve Ping Endpoint

```
35 # Servis başlangıç kontrolü
36 @app.get("/")
37 async def root():
38     return {"message": "Prediction Service is running!"}
39
40 # Servisin çalışıp çalışmadığını test etmek için
41 @app.get("/ping")
42 async def ping():
43     return {"ping": "pong!"}
44
```

- ✓ Kullanıcı API'nin çalıştığını doğrulamak için / (ana dizin) endpoint'ine istekte bulunabilir.
- ✓ Yanıt olarak {"message": "Prediction Service is running!"} döndürülür.
- ✓ API'nin aktif olup olmadığını test etmek için kullanılan bir sağlık kontrol endpoint'idir.
- ✓ Yanıt olarak { "ping": "pong!" } döner.

FastAPI Servisinin Çalışma Mantığı

/predict Endpoint'i

```
45 # Tahmin endpoint
46 @app.post("/predict", response_model=PredictionOutput)
47 async def predict(input_data: PredictionInput):
48     try:
49         # Giriş metnini vektörleştirmek için
50         text_vector = vectorizer.transform([input_data.text])
51
52         # Model ile tahmin yapmak
53         prediction = model.predict(text_vector)
54
55         # Tahmini döndürmek
56         return PredictionOutput(category=prediction[0])
57     except Exception as e:
58         raise HTTPException(status_code=500, detail=str(e))
59
```

✓ Bu endpoint, bir POST isteği alarak metni sınıflandırmak için makine öğrenmesi modelini çalıştırır.

Streamlit ile Kullanıcı Arayüzü

Kullanıcıların API'yi Kolayca Kullanabilmesi İçin Oluşturulan Arayüz

```
1 import streamlit as st
2 import requests
3 import matplotlib.pyplot as plt
4 from collections import Counter
5
6 # FastAPI'nin URL'si
7 API_URL = "http://127.0.0.1:8000/predict"
8
9 st.title("Haber Kategorisi Tahmin Uygulaması")
10
11 # Kullanıcıdan metin girişi
12 news_texts = st.text_area("Birden fazla haber metni girin (her birini yeni satırda yazınız):")
13
```

Uygulama, kullanıcıdan haber metinlerini alır, API'ye gönderir ve API'den gelen tahminleri görsel olarak kullanıcıya sunar. Ayrıca, tahmin edilen kategorilerin dağılımını görselleştirir ve kategori başına haber sayısını gösterir.

Streamlit ile Kullanıcı Arayüzü

```
33     if categories:
34         # Kategorileri sayalım
35         category_counts = Counter(categories)
36
37         # Kategorilerin dağılımını görselleştirelim
38         st.subheader("Kategori Dağılımı")
39         fig, ax = plt.subplots()
40         ax.bar(category_counts.keys(), category_counts.values(), color='skyblue')
41         ax.set_xlabel('Kategori')
42         ax.set_ylabel('Adet')
43         ax.set_title('Her Kategorideki Haber Sayısı')
44         st.pyplot(fig)
45
46         # Kategorilerin detaylarını gösterelim
47         st.subheader("Tahmin Edilen Kategoriler")
48         for category, count in category_counts.items():
49             st.write(f"{category}: {count} adet haber")
50     else:
51         st.warning("Hiçbir metin girilmedi.")
52 else:
53     st.warning("Lütfen haber metinlerini girin.")
54
```

```
14 # Tahmin butonu
15 if st.button("Tahmin Et"):
16     if news_texts:
17         # Metinleri satırlara ayıralım
18         news_list = news_texts.split("\n")
19
20         categories = [] # Kategorileri saklamak için liste
21
22         # Her bir metni FastAPI'ye gönderip tahminleri alalım
23         for news in news_list:
24             if news.strip(): # Boş satırları atla
25                 response = requests.post(API_URL, json={"text": news.strip()})
26
27                 if response.status_code == 200:
28                     prediction = response.json()
29                     categories.append(prediction['category'])
30                 else:
31                     st.error(f"'{news}' için tahmin yapılamadı.")
32
```

Yukarıdaki kodlar ile oluşturduğumuz arayüz sayesinde kullanıcılar birden fazla haber metnini tek seferde girip, "Tahmin Et" butonuna tıklayarak tahmin edilen kategoriler ile ilgili sonuçları görebilir ve her kategori için haber sayısını görsel olarak izleyebilirler.

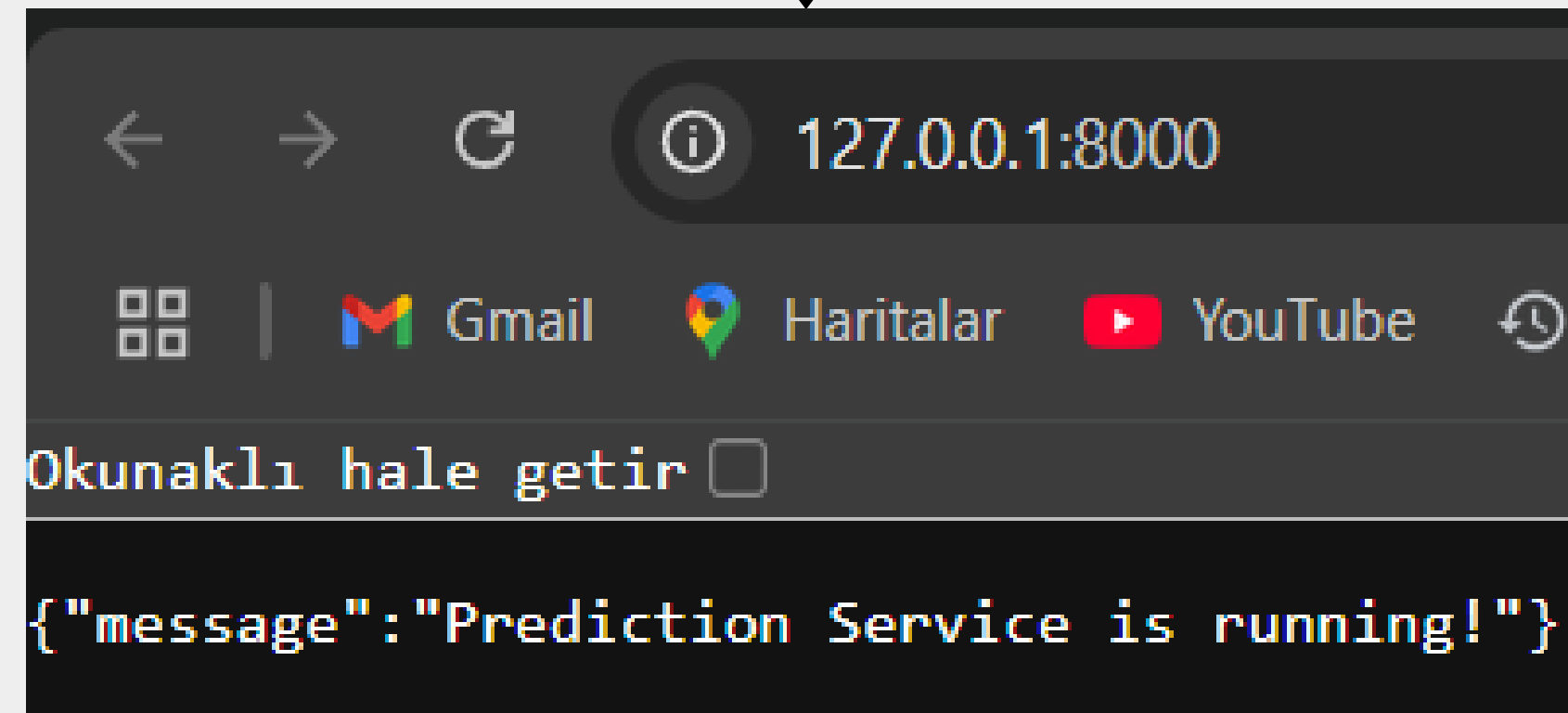
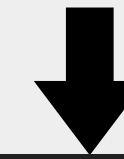
Projenin Test Edilmesi & Çalıştırma Adımları

Sanal Ortamın Aktifleştirilmesi ve FastAPI Sunucusunu Çalıştırma

```
PS C:\Users\Hp\OneDrive\Masaüstü\V.\venv\Scripts\Activateler\jupyterNotebook_Projects\fastapi_prediction_service>  
(venv) PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\fastapi_prediction_service> cd app  
(venv) PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\fastapi_prediction_service\app> uvicorn main:app --reload
```



```
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)  
INFO: Started reloader process [21048] using StatReload  
INFO: Started server process [13040]  
INFO: Waiting for application startup.  
INFO: Application startup complete.  
█
```



http://127.0.0.1:8000/Docs

default

GET	/	Root	▼
GET	/ping	Ping	▼
POST	/predict	Predict	▼

Schemas

HTTPValidationError	>	Expand all	object
PredictionInput	>	Expand all	object
PredictionOutput	>	Expand all	object
ValidationError	>	Expand all	object

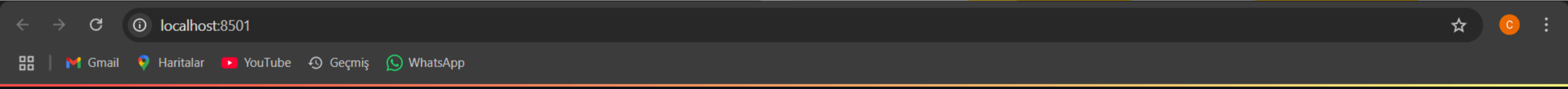
Streamlit Arayüzünü Başlatma

```
PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\fastapi_prediction_service>  
PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\fastapi_prediction_service>
```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.1.106:8501>



Haber Kategorisi Tahmin Uygulaması

Birden fazla haber metni girin (her birini yeni satırda yazınız):

Tahmin Et

Model Test

test verisi

"Türk lirası, dolar karşısında değer kazandı."
"Fenerbahçe yeni transferiyle kadrosunu güçlendirdi."
"NASA Marsa yeni bir uzay aracı gönderdi."
"Kadın hakları için dünya çapında bir yürüyüş düzenlendi."
"Yapay zeka, sağlık alanında büyük bir devrim yaratıyor."
"Çin ekonomisi son yıllarda hızlı bir büyüme gösterdi."
"Popüler bir film, tüm zamanların en çok izlenen yapımlarından biri oldu."
"Küresel ısınma tehdidi, dünya çapında tartışılıyor."
"Microsoft, yeni işletim sistemi sürümünü duyurdu."
"Dünya futbolunun en büyük takımları, transfer dönemiyle ilgili görüşmelerde."
"Büyüyen e-ticaret sektörü, geleneksel perakendeciliği zorluyor."
"Birçok ülkede okullar çevrim içi eğitime geçti."
"Uzay turizmi, geleceğin en büyük endüstrisi olabilir."
"Yeni bir çevre yasası, plastik atıkları azaltmayı hedefliyor."
"Öğrenciler, yeni okul yılı için hazırlıklarını tamamladı."
"Bilim insanları, evrim teorisinin yeni bir yönünü keşfetti."
"Tarih boyunca büyük savaşlar, dünya tarihini şekillendirdi."
"Bitcoin, dijital para birimlerinin geleceği olabilir."
"Türkiyede turizm sektörü, pandemi sonrası toparlanma sürecinde."
"Yeni bir mobil uygulama, kullanıcı deneyimini dönüştürmeyi vaat ediyor."
"Geçtiğimiz yıl en yüksek ekonomik büyümeyi Asya ülkeleri gerçekleştirdi."
"Ünlü bir şarkıcı, dünya turnesine çıkıyor."
"Havacılık sektörü, yeni nesil uçaklarla hız kazanacak."
"Ülkeler arasında yapılan ticaret anlaşmaları, küresel ekonomiyi şekillendiriyor."
"Yüksek hızda internet bağlantısı, dünya çapında erişilebilir hale geliyor."



Haber Kategorisi Tahmin Uygulaması

Birden fazla haber metni girin (her birini yeni satırda yazınız):

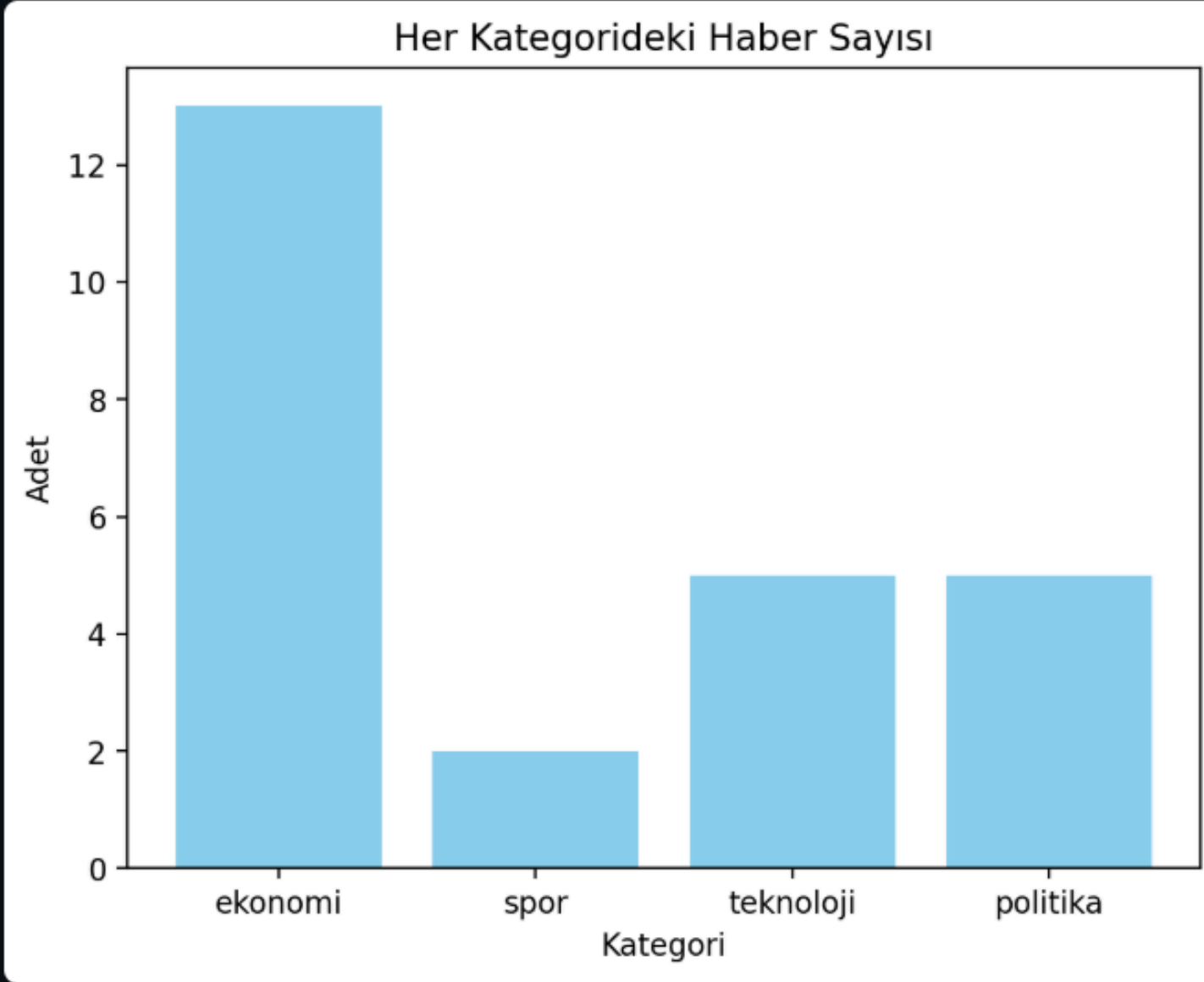
"Türk lirası, dolar karşısında değer kazandı."
"Fenerbahçe yeni transferiyle kadrosunu güçlendirdi."
"NASA Marsa yeni bir uzay aracı gönderdi."
"Kadın hakları için dünya çapında bir yürüyüş düzenlendi."
"Yapay zeka, sağlık alanında büyük bir devrim yaratıyor."
"Çin ekonomisi son yıllarda hızlı bir büyüme gösterdi."
"Popüler bir film, tüm zamanların en çok izlenen yapımlarından biri oldu."
"Küresel ısınma tehdidi, dünya çapında tartışılıyor."
"Microsoft, yeni işletim sistemi sürümünü duyurdu."
"Dünya futbolunun en büyük takımları, transfer dönemiyle ilgili görüşmelerde."
"Büyüyen e-ticaret sektörü, geleneksel perakendeciliği zorluyor."
"Birçok ülkede okullar çevrim içi eğitime geçti."
"Uzay turizmi, geleceğin en büyük endüstrisi olabilir."
"Yeni bir çevre yasası, plastik atıkları azaltmayı hedefliyor."
"Öğrenciler, yeni okul yılı için hazırlıklarını tamamladı."
"Bilim insanları, evrim teorisinin yeni bir yönünü keşfetti."
"Tarih boyunca büyük savaşlar, dünya tarihini şekillendirdi."
"Bitcoin, dijital para birimlerinin geleceği olabilir."

Tahmin Et

SONUÇ

Kullanıcı Arayüzü

Kategori Dağılımı



Tahmin Edilen Kategoriler ⇄

ekonomi: 13 adet haber

spor: 2 adet haber

teknoloji: 5 adet haber

politika: 5 adet haber

Terminal Çıktı

```
INFO: 127.0.0.1:62527 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62528 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62529 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62530 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62531 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62532 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62533 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62534 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62535 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62536 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62537 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62538 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62539 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62540 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62541 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62542 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62543 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62544 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62545 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62546 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62547 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62548 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62549 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62550 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:62551 - "POST /predict HTTP/1.1" 200 OK
```



Teşekkürler.

CAN ARI