



Spor Öneri Platformu

Sunan: Şöbiyet Ekibi

12.02.2025



Biz Kimiz ?



Ayşenur YÖRÜR
Medeniyet Üniversitesi Bilgisayar Mühendisliği
4. Sınıf Öğrencisi



Emine Sude ASLAN
Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği
4. Sınıf Öğrencisi



Eda USLU
Düzce Üniversitesi Yönetim Bilişim Sistemleri
2. Sınıf Öğrencisi



Elif ARSLAN
Korkut Ata Üniversitesi Bilgisayar Mühendisliği
3. Sınıf Öğrencisi



Ayşe Gül KILIÇ
Pamukkale Üniversitesi Yönetim Bilişim Sistemleri
4. Sınıf Öğrencisi

Projemiz

Spor Öneri projemiz, GSB Biz uygulamasını kullanan öğrencilere fiziksel özelliklerine uygun spor dallarını önerir. Önerinin ardından, kullanıcı büyük dil modeliyle iletişime geçerek kendisi için bir antrenman programı oluşturabilir veya önerilen spor hakkında detaylı bilgi edinebilir.



Sağlık ve Motivasyon Desteği

Kullanıcıyı spor yapmaya teşvik eder, spor alışkanlığı kazandırır ve motivasyonunu artırır.

PROJENİN YARARLARI

Kişiselleştirilmiş Spor Önerileri

Kullanıcının yaş, kilo ve boyuna uygun spor dallarını önererek en verimli ve güvenli egzersizleri yapmasına yardımcı olur.

Sakatlanma Riskini Azaltma

Kullanıcının fiziksel özelliklerine uygun sporları önererek sakatlanma riskini en aza indirir.

Verimlilik ve Süreklilik

Sporu daha bilinçli yapmalarını sağlayarak uzun vadede sağlıklı yaşam alışkanlıkları kazanmalarına destek olur.

Erişilebilir Spor Danışmanı

Kullanıcı, dilediği zaman sporla ilgili sorularını sorarak yapay zekadan öneriler alabilir.

Spor Öneri Projesinin BİZ Uygulamasına Sağlayacağı Yararlar

Kullanıcı Etkileşimini Artırma

Gençler yapay zeka ile etkileşim kurdukları için daha rahat soru sorabilirler.

Daha İyi Kullanıcı Deneyimi

Modelin kişilerin bilgilerine göre öneride bulunması kişiye ait cevaplar sunar

Özgün Spor Tercihleri

Önerilen sporlar sadece çevrede görülenlerle sınırlı kalmadığı için gençlere yeni spor dallarını keşfetmesine dair bir ufuk sunar.

Projemizde Ne Yaptık?

Model Eđitimi

20 yıllık olimpiyat verileri
kullanarak spor öneri
modeli geliřtirdik.

LLM Entegrasyonu

Hugging face dil modeline
eriřerek kullanıcının sorularını
sorabileceđi bir bot imkanı
sunduk.

Flask Uygulaması

Uygulama geliřtiriciler arasında
daha kolay anlařılabilmesi iin
flask yapısını tercih ettik.

Spor Öneri Uygulaması Örnek Kullanım Videosu

Yoga veya Pilates

merhaba. Bana haftalık yoga takvimi çıkarır mısın? haftada iki gün ve yoğun antrenmanlar halinde olsun.

merhaba. Bana haftalık yoga takvimi çıkarır mısın? haftada iki gün ve yoğun antrenmanlar halinde olsun.

Gönder

| Model (Base) | Accuracy | F1 |
|------------------------------|----------|--------|
| Logistic Regression | 0.2283 | 0.1306 |
| Support Vector Machine (SVC) | 0.2207 | 0.1070 |
| Gradient Boosting | 0.3256 | 0.2689 |
| Naive Bayes | 0.0702 | 0.0357 |
| Random Forest | 0.2627 | 0.2327 |

| Model (with new features) | Accuracy |
|---------------------------|----------|
| Random Forest | 0.6706 |

**Öncesi
Accuracy**

0.23

**Sonrası
Accuracy**

0.67

Teknik Detaylar

Problem Tanımı:

Kullanıcıların yaş, boy ve kilo gibi fiziksel özelliklerine göre en uygun spor dallarını belirleme hedeflenmiştir.

- Bağımsız değişkenler: Yaş, cinsiyet, boy, kilo, ülke
- Bağımlı değişken: Spor dalı

Veri Seti

Olimpiyat sporcularına ait 1896-2016 yıllarını kapsayan veri seti.
Spor branşlarını belirlemek için geniş kapsamlı, gerçek dünya verisi

Sporcu Yönlendirme

Kullanıcıların yaş, boy ve kilo gibi fiziksel özelliklerine göre en uygun spor dallarını belirleme hedeflenmiştir.

Problem tipi: Denetimli öğrenme (Supervised Learning) sınıflandırma problemi.

Spor Branşlarının Kategorilendirilmesi

Daha iyi öznitelik mühendisliği

Modelin daha anlamlı ve genelleştirilebilir öğrenmesini sağlar.

Veri dengesini artırır

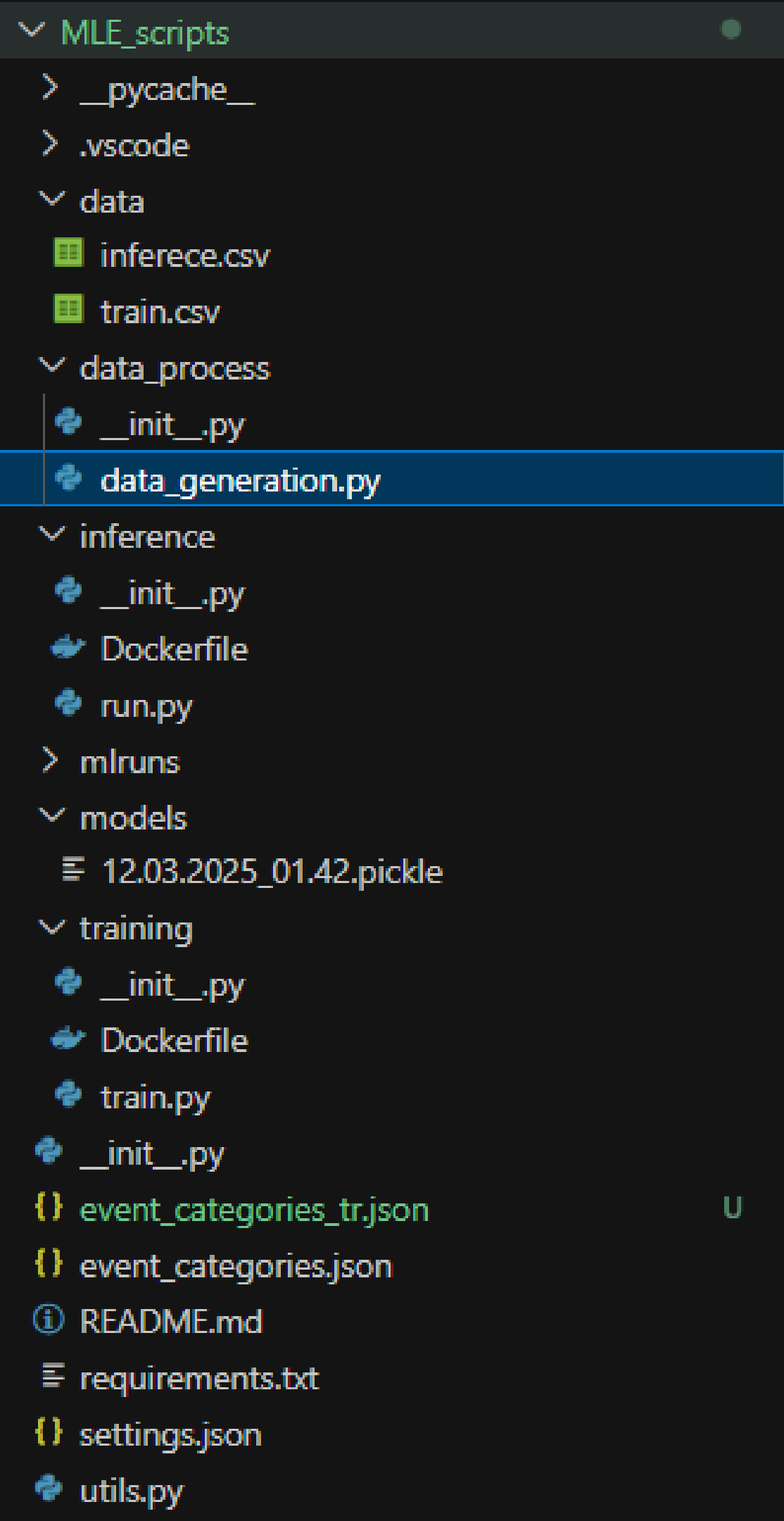
Az temsiliyeti olan sporları daha geniş kategorilerle ele alır.

Modelin açıklanabilirliğini güçlendirir

Model tahminlerini "sen bisiklet sporuna uygunsun" yerine, "sen dayanıklılık gerektiren spor dallarında başarılı olabilirsin" şeklinde açıklayabilir.

Böylece son kullanıcı için daha anlamlı bir çıktı üretmiş olur.

```
134 "endurance_stamina": {  
135   },  
136 },  
137 "road_cycling": [  
138   "Cycling Men's Road Race, Individual",  
139   "Cycling Men's 100 kilometres Team Time Trial",  
140   "Cycling Men's Mountainbike, Cross-Country",  
141   "Cycling Women's Road Race, Individual",  
142   "Cycling Women's Mountainbike, Cross-Country"  
143 ],  
144 "biathlon": [  
145   "Biathlon Men's 20 kilometres",  
146   "Biathlon Men's 10 kilometres Sprint",  
147   "Biathlon Women's 7.5 kilometres Sprint"  
148 ],  
149 },  
150 "precision_marksmanship": {  
151   "archery": [  
152     "Archery Men's Individual",  
153     "Archery Women's Individual"  
154   ],  
155   "shooting": [  
156     "Shooting Men's Air Rifle, 10 metres",  
157     "Shooting Women's Air Rifle, 10 metres",  
158     "Shooting Men's Air Pistol, 10 metres",  
159     "Shooting Women's Air Pistol, 10 metres",  
160     "Shooting Men's Rapid-Fire Pistol, 25 metres",  
161     "Shooting Men's Free Pistol, 50 metres",  
162     "Shooting Men's Small-Bore Rifle, Three Positions, 50 metres",  
163     "Shooting Men's Small-Bore Rifle, Prone, 50 metres",  
164     "Shooting Mixed Small-Bore Rifle, Three Positions, 50 metres",  
165     "Shooting Mixed Small-Bore Rifle, Prone, 50 metres",  
166     "Shooting Mixed Free Pistol, 50 metres",  
167     "Shooting Men's Skeet",  
168     "Shooting Mixed Skeet",  
169     "Shooting Mixed Trap",  
170     "Shooting Men's Trap"  
171   ],  
172 },  
173 "artistic_aesthetic_judged": {  
174   "figure_skating": [  
175     "Figure Skating Men's Singles",  
176     "Figure Skating Women's Singles",  
177     "Figure Skating Mixed Pairs",  
178     "Figure Skating Mixed Ice Dancing"  
179   ],  
180 },  
181 }
```

```
4 import os
5 import sys
6 import json
7 from sklearn.model_selection import train_test_split
8
9 # Create logger
10 logger = logging.getLogger(__name__)
11 logger.setLevel(logging.INFO)
12
13 # Define directories
14 ROOT_DIR = os.path.dirname(os.path.abspath(__file__))
15 sys.path.append(ROOT_DIR)
16
17 from utils import get_project_dir
18
19 DATA_DIR = get_project_dir('data')
20 if not os.path.exists(DATA_DIR):
21     os.makedirs(DATA_DIR)
22
23 # Change to CONF FILE = "settings.json" if you want to use it
24 CONF_FILE = "settings.json"
25
26 # Load config
27 logger.info(f"Loading config from {CONF_FILE}")
28 with open(CONF_FILE, 'r') as f:
29     conf = json.load(f)
30
31 # Define paths
32 logger.info("Defining paths...")
33 DATA_DIR = get_project_dir(conf['general']['data_dir'])
34 TRAIN_PATH = os.path.join(DATA_DIR, conf['training']['train_path'])
35 INFERENCE_PATH = os.path.join(DATA_DIR, conf['inference']['inference_path'])
```



PoC (Proof of Concept) Tamamlandı: Başarılı bir temel model oluşturulduktan sonra MLOps adımları uygulanmaya başlandı.



- **data_generation.py:** Veri ön işleme ve veri seti oluşturma
- **training.py:** Model eğitimi
- **inference.py:** Model tahmini (belirli bir noktaya kadar geliştirildi)



- **MLflow:** *Artifact Management*, Veri seti, model ve tahmin sonuçları kaydedildi. *Version Control*, Modelin farklı versiyonları saklanmasını ve karşılaştırılmasını sağlar.
- **Docker:** Modelin bağımlılıklardan arındırılmış, her bilgisayarda çalışabilir hale getirilmesi sağlandı.



- **Hedef:**
- Cloud (AWS, GCP veya Azure) üzerinde MLflow ile model ve veri yönetimi sağlanacak.
- CI/CD Pipeline ile model güncellemeleri otomatik hale getirilecek.
- Modelin API olarak deploy edilmesi için Flask planlanıyor

Dinlediğiniz İçin Teşekkür Ederiz.