



İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi

Katkıda Bulunanların Listesi

- Ahmet Berkay Avan
- Mehmet Begun
- Ali Hakan Kıncal
- Barış Emre Ahi

Task Matrix

Görev	Atanan Kişi
RSS Kaynaklarından Veri Toplama Testi (Senaryo 1)	Ali Hakan Kıncal
Doğal Dil İşleme ile Kategorilendirme Testi (Senaryo 2)	Barış Emre Ahi
Öneri Üretimi Testi (Senaryo 3)	Mehmet Begun
Web Arayüzü Üzerinden Erişim ve Manuel Veri Girişi (Senaryo 4)	Ahmet Berkay Avan
Kullanıcı Hikayeleri ve Gereksinim Bağlantıları	Ali Hakan Kıncal, Barış Emre Ahi
Belgelerin Son Düzenleme ve Formatlaması	Ahmet Berkay Avan, Barış Emre Ahi, Ali Hakan Kıncal, Mehmet Begun

İÇİNDEKİLER

Katkıda Bulunanların Listesi	1
Giriş	2
Test Ön Koşulları ve Ortam Gereksinimleri	2
### Senaryo 1: RSS Kaynaklarından Veri Toplama	3
### Senaryo 2: Doğal Dil İşleme ile Kategorilendirme	3
### Senaryo 3: Öneri Üretimi	4
### Senaryo 4: Web Arayüzü Üzerinden Erişim ve Manuel Veri Girişi	4
Kullanıcı Hikayeleri ve Gereksinim Bağlantıları	5

Giriş

Bu bölüm, sistemin dört temel kullanım senaryosu için işlevsel ve işlevsel olmayan gereksinimlere dayalı olarak tanımlanan kabul kriterlerini ve bu kriterleri test eden kabul testlerini içerir. Her test, belirli bir senaryonun başarıyla tamamlandığını gösterecek şekilde tasarlanmıştır.

Test Ön Koşulları ve Ortam Gereksinimleri

Amaç: Testlerin sorunsuz yürütülebilmesi için gerekli yazılım, donanım ve konfigürasyon ayrıntılarının belirlenmesi.

- **Yazılım ve Bağımlılıklar:**
 - **Python Sürümü:** Python 3.10 ve üzeri
 - **Framework ve Kütüphaneler:**
 - Flask (web sunucusu ve arayüz)
 - SQLite (hafif veritabanı yönetim sistemi)
 - BeautifulSoup & Feedparser (RSS veri çekimi)
 - NLTK, Scikit-learn (doğal dil işleme ve LDA modeli)
 - CSS ve HTML (web arayüzü şablonlama)
 - **Bağımlılık Yönetimi:** Proje dizinindeki `requirements.txt` dosyası üzerinden tüm kütüphaneler kurulmalıdır.
 - **Ortam ve Altyapı:**
 - **İşletim Sistemi:** Windows 10 veya Ubuntu 22.04
 - **Geliştirme Ortamı:** VSCode, Terminal
 - **Ağ Gereksinimleri:** Stabil internet bağlantısı; özellikle RSS kaynaklarına erişim için
 - **Port Ayarları:** Uygulama, örneğin 5000 portu üzerinden çalıştırılmalıdır.
 - **Ön Hazırlık Adımları:**
 - Test veritabanının (news.db) oluşturulması
 - Gerekli konfigürasyon ve environment değişkenlerinin ayarlanması
-

Senaryo 1: RSS Kaynaklarından Veri Toplama

Acceptance Criteria:

- Sistem, tanımlı RSS kaynaklarından en az 10 haber çekebilmelidir.
- Çekilen veriler SQLite veritabanına başarıyla kaydedilmelidir.
- Veri çekme işlemi günlük olarak tekrarlanabilir olmalıdır.

Acceptance Test:

- **Test ID:** AT-01
 - **Test Adımları:**
 1. `fetch_rss()` fonksiyonu çalıştırılır.
 2. Veritabanı kontrol edilerek news tablosuna kayıt eklenip eklenmediği gözlemlenir.
 - **Beklenen Sonuç:** En az 10 haberin veritabanına kaydedilmiş olması gerekir.
 - **Kriterle Uyum Açıklaması:** Bu test, veri çekme modülünün işlevsel olduğunu ve günlük veri toplamaya uygun olduğunu doğrular.
-

Senaryo 2: Doğal Dil İşleme ile Kategorilendirme

Acceptance Criteria:

- Sistem, çekilen haberlerin en az %90'ını belirgin bir kategoriye ayırmalıdır.
- `category` alanı boş kalmamalıdır.
- LDA modeli düzgün çalışmalıdır.

Acceptance Test:

- **Test ID:** AT-02
 - **Test Adımları:**
 1. `categorize_news()` fonksiyonu çalıştırılır.
 2. Her haber için `category` alanı kontrol edilir.
 - **Beklenen Sonuç:** Tüm haberler en az bir kategoriye atanmış olmalıdır.
 - **Kriterle Uyum Açıklaması:** Haberlerin doğru şekilde sınıflandırılması, öneri motorunun işlevselliği için ön koşuldur.
-

Senaryo 3: Öneri Üretimi

Acceptance Criteria:

- Her kategorilendirilmiş haber için en az bir öneri sunulmalıdır.
- Öneriler, ilgili kategorinin "action_suggestions" listesine uygun olmalıdır.
- Öneri kullanıcı arayüzünde görüntülenebilir olmalıdır.

Acceptance Test:

- **Test ID:** AT-03
 - **Test Adımları:**
 1. Uygulama başlatılır (python app.py).
 2. Tarayıcıdan anasayfa (http://127.0.0.1:5000/) açılır.
 3. Rastgele bir haber seçilir ve öneri varlığı kontrol edilir.
 - **Beklenen Sonuç:** İlgili haberin altında en az bir öneri görünmelidir.
 - **Kriterle Uyum Açıklaması:** Bu test, öneri motorunun doğru ve anlamlı çalıştığını garanti eder.
-

Senaryo 4: Web Arayüzü Üzerinden Erişim ve Manuel Veri Girişi

Acceptance Criteria:

- Kullanıcı, web arayüzü üzerinden sisteme erişebilmelidir.
- Kullanıcı, tarayıcıdan sistemin sunduğu haber listesini görebilmelidir.
- Her haber başlığı, bağlantıya tıklanabilir olmalı ve ilgili kaynağa yönlendirmelidir.
- Web arayüzü, her haber için özet, kategori ve öneri bilgilerini açıkça göstermelidir.
- Kullanıcı, <http://localhost:5000/admin> bağlantısı üzerinden yeni haber kaydı oluşturabilmelidir.

Acceptance Test:

- **Test ID:** AT-04
- **Test Adımları:**
 1. Flask uygulaması çalıştırılır (python app.py).
 2. Tarayıcıdan <http://127.0.0.1:5000> adresine erişilir.
 3. Haber listesi ve öneriler kontrol edilir, kategori filtresi denenir.
 4. Rapor oluşturma butonu test edilir.
- **Beklenen Sonuç:** Sayfa yüklenmeli, öneriler doğru görünmeli ve manuel veri girişi başarılı bir şekilde gerçekleştirilmeli.
- **Kriterle Uyum Açıklaması:** Bu test, son kullanıcı deneyimini ve sistemin erişilebilirliğini değerlendirir.

Kullanıcı Hikayeleri ve Gereksinim Bağlantıları

Amaç: Her test senaryosunun ilgili kullanıcı ihtiyaçları ve iş gereksinimleriyle doğrudan ilişkilendirilmesini sağlamak.

- **Örnek Kullanıcı Hikayeleri:**

- *Hikaye 1:* "Kullanıcı olarak, güncel haberleri sistem üzerinden görüntülemek istiyorum."
 - **Bağlantı:** Senaryo 1 (RSS veri çekme)
- *Hikaye 2:* "Kullanıcı olarak, haberlerin doğru kategorilendirilmesini görmek istiyorum."
 - **Bağlantı:** Senaryo 2 (Doğal dil işleme ile kategorilendirme)
- *Hikaye 3:* "Kullanıcı olarak, haberlere ilişkin anlamlı eylem önerileri almak istiyorum."
 - **Bağlantı:** Senaryo 3 (Öneri üretimi)
- *Hikaye 4:* "Yönetici olarak, sistem üzerinden haber kayıtlarını manuel olarak ekleyebilmek istiyorum."
 - **Bağlantı:** Senaryo 4 (Web arayüzü üzerinden erişim ve manuel veri girişi)