

Proje Raporu: Futbol İstatistik Dashboard ve Tahmin Modelleri

1. Proje Tanımı

Bu rapor, futbola dair geçmiş maç verilerini kullanarak:

- **İstatistiksel Gösterim Dashboard'u** oluşturmayı,
- **Lig bazlı tahmin modelleri** geliştirip bunları bir "Forecast" sekmesinde kullanıma sunmayı,

amaçlayan iki aşamalı proje için yol haritasını ve gereksinimleri detaylandırmaktadır.

2. Kapsam ve Modüller

| Modül | Açıklama | Çıktı/Kullanıcı Deneyimi |
|------------------------------|--|--|
| A. Veri Katmanı | - PostgreSQL veri tabanı- Lig, takım, maç ve skor tabloları- ETL pipeları | Güvenilir, sorgulanabilir ham veri |
| B. Backend API | - FastAPI ile REST uç noktaları: • Takım istatistikleri • Lig genel istatistikleri | JSON formatında veriyi frontend'e sunan API |
| C. Frontend Dashboard | - Next.js + React- Recharts veya Chart.js ile grafikler- Arama & filtreleme UI | Kullanıcı dostu arayüz: takım/lig/ sezon seç, grafik & tablo |
| D. Tahmin Modelleri | - XGBoost/LightGBM (ML)- LSTM/Transformer (DL)- Model kayıt altyapısı | Her lig için ayrı model dosyası ve performans raporu |
| E. Forecast Tab | - Frontend'de tahmin sayfası- Backend inference uç noktası | Seçilen ligde gelecek maç için skor/olasılık tahminleri |
| F. DevOps & MLOps | - Docker/Docker Compose- GitHub Actions CI/CD- MLflow/DVC model versiyonlama | Otomatik build, test, deploy; model takibi ve güncelleme |

3. Teknoloji ve Araç Seçimleri

| Katman | Teknoloji / Araç | Açıklama |
|-------------|--------------------------------|---|
| Veri | PostgreSQL | Zaten mevcut, güçlü SQL desteği |
| Backend | FastAPI | Yüksek performans, asenkron destek |
| Frontend | Next.js + React | SSR + SEO avantajı, modern React ekosistemi |
| Modelleme | scikit-learn, XGBoost, PyTorch | Hızlı prototipleme, yaygın topluluk |
| Container | Docker + docker-compose | İzolasyon, taşınabilir geliştirme ortamı |
| CI/CD | GitHub Actions | Kolay entegrasyon, ücretsiz |
| Model Vers. | MLflow veya DVC | Modelin metadata ve versiyon takibi |

4. Detaylı Yol Haritası

♦ Hazırlık & Planlama (1 hafta)

1. Gereksinim dokümanı oluşturma
2. ER diyagramı ve API sözleşmelerini (OpenAPI) tasarlama
3. GitHub repo şablonu ve klasör yapısını belirleme
4. Başlangıç Dockerfile ve README hazırlığı

♦ Aşama 1: Dashboard MVP (2-3 hafta)

1. Backend: Takım ve lig istatistik API uç noktaları
2. Frontend: Temel sayfalar, arama, grafik bileşenleri
3. Docker Compose: Backend + Frontend + PostgreSQL (geliştirme ortamı)
4. CI/CD: Lint, unit test, security scan iş akışı

♦ Aşama 2: İleri Dashboard Özellikleri (2 hafta)

1. Detaylı filtreler ve tarih aralığı seçimi
2. Tablo görüntüleme ve CSV indirme
3. Kullanıcı kimlik doğrulama (opsiyonel)

♦ Aşama 3: Model Geliştirme & Eğitim (3-4 hafta)

1. Özellik mühendisliği (takım formu, içerik dışı veri)
2. Baseline modeller (lojistik regresyon) ve ileri modeller (XGBoost, LSTM)
3. Hiperparametre optimizasyonu (Optuna)
4. Eğitim kodunun hazırlanması ve MLflow ile model kayıt
5. Performans raporlarının oluşturulması

♦ Aşama 4: Forecast Tab Entegrasyonu (2 hafta)

1. Inference API uç noktası
2. Frontend Forecast sayfasının geliştirilmesi
3. Model güncelleme stratejisi (cronjob/GitHub Actions)
4. CI/CD - Model Deployment iş akışı

♦ Aşama 5: Üretim & İzleme (1 hafta)

1. Production ready Docker Compose veya Kubernetes manifestleri
2. Canlıya deploy (VPS veya bulut)
3. Logging & Monitoring (Grafana, Prometheus)
4. Kullanım kılavuzu ve tam dokümantasyon

5. CI/CD & Konteynerizasyon Detayları

• Pull Request İş Akışı

• Linting (flake8/isort)

• Unit test (pytest)

- Docker image build (PR tag ile)

- **Ana Dal Merge**

- Image push (ghcr.io/org/football-app\latest)

- Staging deploy

- **Model Pipeline**

- Eğitim sonrası GitHub Actions job: model eğitimi, artifact olarak model dosyası

- Inference API'da test edilmesi

```
# Örnek: .github/workflows/ci.yml
name: CI
on: [pull_request]
jobs:
  test-and-build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Setup Python
        uses: actions/setup-python@v4
        with: python-version: '3.11'
      - name: Install dependencies
        run: pip install -r requirements.txt
      - name: Lint
        run: flake8 src/
      - name: Test
        run: pytest --maxfail=1 --disable-warnings -q
      - name: Docker Build
        run: docker build -t football-app:pr-${{ github.event.number }} .
```

6. Sonuç

Bu rapor, projenin tüm gereksinimlerini, teknolojik altyapısını ve aşamalı yol haritasını özetlemektedir. Bir sonraki adım olarak sprint planlaması ve ilk modülün geliştirmesine geçebiliriz.