



## Uçtan Uca Machine Learning Projesi

Kendi belirlediğiniz bir sektördeki probleme uçtan uca makine öğrenmesi ile bir çözüm geliştirmenizi ve çözümünüzün her adımını dokümantete etmenizi istiyoruz.

Projenizi aşağıdaki gereksinimlere uygun olarak geliştirmeniz gerekiyor.

### 1) Sektör ve Problem Seçimi

Merak ettiğiniz veya çalışmaktan keyif alacağınız bir sektörü seçin. O sektördeki data science problemlerini araştırın. Hoşunuza giden bir tanesini seçin.

#### Örnek sektör ve problemler:

- **Bankacılık:** Credit scoring, fraud detection vb
- **E-Ticaret:** Recommendation, churn, search vb.
- **Oyun:** User segmantasyon, LTV Prediction vb.
- ...

### 2) Probleminize Uygun Dataset Seçimi

Dataseti devam eden veya geçmiş Kaggle yarışmalarından bulmanızı tavsiye ederiz. Yarışmalardaki datalar hem gerçeğe yakın hem de hacim olarak büyük oluyor.

#### Data seçim kriterleri:

- Tabular fomatlı bir dataset olmalı (.csv, .parquet, .xlsx)
- Tercihen sentetik bir dataset olmasın (kaggle description bakılabilir)
- Playground yarışmalarındaki dataları sentetik veri olmasından kaynaklı önermiyoruz
- Data boyutu 10k satırдан az olmamalı
- Datadaki feature sayısı ise en az 10 tane olmalı

Aynı zamanda oradaki bir çok discussion'dan ve kod kısmından faydalanabilirsiniz, submission atarak da modelinizin doğruluğunu test edebilirsiniz.

Yarışma ve data bulmak için seçtiğiniz problemi **Kaggle** da search edebilirsiniz. Tabii ki **Kaggle** dışında aşağıda olan veya olmayan başka kaynakları da kullanabilirsiniz

- UCI Machine Learning Repository

- Google Dataset Search
- Hugging Face Datasets vb.

### 3) Repo ve Proje Yapısı

#### Notebooks:

- 1) **EDA:** Problem tanımı, veriye ve değişkenlere ilk bakış, dağılımlar, korelasyonlar vb.
- 2) **Baseline:** En basit pipeline ve feature set ile temel bir model eğitmek
- 3) **Feature Eng:** Datada olan farklı featureleri içeri eklemek veya türetmek
- 4) **Model Optimization:** Seçtiğiniz modellere hiperparametre optimizasyonu uygulayın
- 5) **Model evaluation:** Feature importance, shap ve business gerekliliklerine göre en uygun feature seti belirleyin
- 6) **Pipeline notebook:** Final ön işleme ve final modelin train olduğu notebook

#### Notebooks/Markdown hücreleri:

- 1) **Eda bulguları docs:** Değişken açıklamaları, dikkat çeken analiz bulguları vb.
- 2) **Baseline docs:** Baseline için basit feature set ve kullanılan modelin başarısı vb.
- 3) **Feature eng docs:** Türetilen veya eklenen değişkenlerin modele olan etkisi vb.
- 4) **Model opt docs:** Optimize etmeye çalıştığın model ve parametreler ve sonuçları vb.
- 5) **Evaluation docs:** Hangi featureler modele nasıl katkı yaptı, model seçimi vb.
- 6) **Final Pipeline docs:** Final pipeline yapısı ve neden bu modeli ve feature setin seçildiği vb.

Final çıktılarınızı ekstra markdown dokümanlarına yazabilirsiniz.

#### Scripts:

- **config.py:** Pathler, business kuralları ve model ayarları vb.
- **inference.py:** Eğitilmiş modelden tahmin alma ve gerekli preprocessing işlemleri
- **app.py:** REST API (fast api veya flask) ve arayüz (gradio, streamlit, html/css/js vb.)
- **pipeline.py:** Tüm ml akışının gerçekleştiği final akış scripti.
- vb.

#### Readme:

- Proje Başlığı
- Projenin ne kapsamında yapıldığı ve çöz düğünüz problemin açıklaması
- Basit inference alabileceğimiz bir deploy linki
- Ekran resmi/videosu/gif
- Sektör, veriseti/pipeline/metrik hakkında kısa bilgi
- Kullanılan teknolojiler
- Local kurulum adımları
- İletişim
- Repo yapısı

**Repoda cevabı yazılı olarak bulunması gereken noktalar:**

- 1) Problem tanımı
- 2) Baseline süreciniz ve skorunuz
- 3) Feature engineering denemeleriniz ve sonuçlarınız
- 4) Seçtiğiniz validasyon şeması ve neden seçtiğiniz
- 5) Final pipelindaki feature setini ve ön işleme stratejisini nasıl seçtiğiniz
- 6) Final modeliniz ile baseline arasındaki başarı farkı nasıldır
- 7) Final model business gereksinimleri ile uyumlu mudur
- 8) Bu model canlıya nasıl çıkar, çıktığında hangi metrikler ile izlenmesi gereklidir

## Örnek repo yapısı

Buradaki tüm klasörler olmak zorunda değil, kendi kurgunuza göre ilgili klasörleri açabilirsiniz.

```
project-name/
  ├── .gitignore
  ├── README.md
  ├── requirements.txt
  ├── data/
  ├── notebooks/
  ├── src/
  ├── models/
  ├── docs/
  └── tests/
```

## 4) Deployment

Burada eğittiğiniz modelden basit bir inference alabileceğimiz bir rest api ve frontend bekliyoruz.

- Streamlit
- HuggingFace Space
- Render
- Cloud vb.

## 5) Olsa Güzel Olur

Zorunlu olmayan fakat yapıldığında projeyi bir adım öne taşıyan özellikler.

- **Git Geçmiş:** Projenin adım adım gelişimini gösteren düzenli commit'ler
- **Monitoring sistemi:** Oluşturduğunuz ML modelin sonuçlarının loglandığı bir database ve ona bağlı gerçek zamanlı çalışan monitoring ekran tasarımlı yapın.
- **Business kurgusu:** Gerçekten bir şirkette çalışan veri bilimci olduğunuzu hayal ederek bir sistem tasarımlı anlatın.
- **Üst yönetim sunumu:** Oluşturduğunuz akışın ve model mimarinizin daha az teknik olan üst yönetimdeki insanlara nasıl faydasını aktarısınız, sunum olarak hazırlayın.
- **YouTube videosu:** Projenin hangi amaçla yapıldığını ve nasıl çalıştığını anlatan kısa bir video.
- **Medium yazısı:** Projenin teknik kısımlarını adım adım anlatan bir Medium yazısı.

## Yardımcı Kaynaklar

- **Örnek proje:** <https://github.com/enesmanan/credit-risk-model>
- **Made with ML:** <https://madewithml.com/>
- **ML Eng Book:**  
[https://soclibrary.futa.edu.ng/books/Machine%20Learning%20Engineering%20\(Andriy%20Burkov\)%20\(Z-Library\).pdf](https://soclibrary.futa.edu.ng/books/Machine%20Learning%20Engineering%20(Andriy%20Burkov)%20(Z-Library).pdf)
- **Awesome Repo:**  
<https://github.com/Developer-MultiGroup/DMG-Data-Science-Awesome>

## Proje Teslim Formu

- **Form:** <https://forms.gle/UEQuUJinWjdu32kM8>

**Son teslim tarihi:** 9.12.2025