# ANKARA ÜNİVERSİTESİ

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**PROJE RAPORU**

**(BLM3522-A) BULUT BİLİŞİM VE UYGULAMALARI**

**Perizat SAGYNBEKOVA (21290895)**

**Derda SİNA GÜNAY (20291274)**

**Rüstem TUKHBETOV (21291001)**

**GitHub (**<https://github.com/Impasbaa/BulutBilisim>**)**

**YouTube (**<https://youtu.be/MvMmtNjSvO0>**)**

**12.05.2025**

**PROJE 2: AKILLI VERİ ANALİTİĞİ VE MAKİNE ÖĞRENMESİ UYGULAMASI**

**Problem:** Makine Öğrenimi APİ’si (Application Programming Interface)

**Backend:** Python

**Makine Öğrenmesi Kütüphaneleri:** Scikit-learn

**Veritabanı:** Veri, doğrudan bir CSV dosyasından alınıyor

**Bulut Platformu:** AWS (Amazon Web Services) - Sagemaker

**Veri Seti:** Iris

**AMAÇ**

Bu projenin amacı, makine öğrenmesi ile oluşturulmuş bir modeli kullanarak tahmin işlemlerini gerçekleştirebilen bir sistem kurmaktır. Bu sistemde model, AWS SageMaker üzerinde barındırılmakta ve bu modele veri gönderip sonuç almak için AWS Lambda fonksiyonu kullanılmaktadır. Lambda fonksiyonu, bir API Gateway ile bağlantılıdır. Böylece kullanıcılar, oluşturulan API üzerinden veri göndererek tahmin sonucunu alabilmektedir.

Proje sayesinde, eğitimli bir modeli doğrudan internet üzerinden erişilebilecek şekilde yayınlama, bu modeli Lambda ve API Gateway ile bağlayarak otomatik çalışan bir sistem oluşturma hedeflenmiştir. Bu yapı sayesinde hem sunucusuz (serverless) bir ortamda çalışılmış hem de ölçeklenebilir ve düşük maliyetli bir çözüm elde edilmiştir.

**GİRİŞ**

Bu projede, gerçek bir veri kümesi üzerinde çalışarak bir makine öğrenmesi modeli geliştirme süreci gerçekleştirilmiştir. Modelin eğitimi, Amazon SageMaker gibi bulut tabanlı bir platform üzerinde gerçekleştirilmiş ve ardından yine bulut ortamında modelin dağıtımı (deployment) sağlanmıştır. Böylece model yalnızca eğitim amacıyla değil, aynı zamanda canlı ortamlarda da kullanılabilir hale getirilmiştir.

Veri ön işleme, analiz ve model eğitimi adımlarının ardından, veriden anlamlı bilgiler çıkarma ve belirli girişlere karşılık tahminlerde bulunma gibi temel makine öğrenmesi hedeflerine ulaşılmıştır. Ayrıca, Lambda ve API Gateway servisleri kullanılarak modelin web üzerinden erişilebilir olması sağlanmış ve uçtan uca bir makine öğrenmesi çözümü tamamlanmıştır.

Bu çalışma sayesinde hem makine öğrenmesi algoritmalarının pratikte nasıl kullanıldığı hem de bulut bilişim teknolojilerinin bu süreçlerdeki önemi konusunda değerli deneyimler kazanılmıştır.

**Amazon SageMaker ve Notebook Instance:**

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

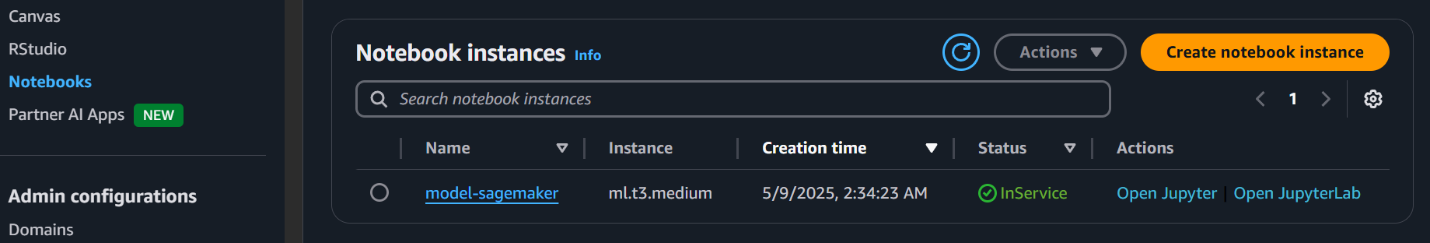
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.***Şekil 1. Amazon SageMaker AI Hizmeti*

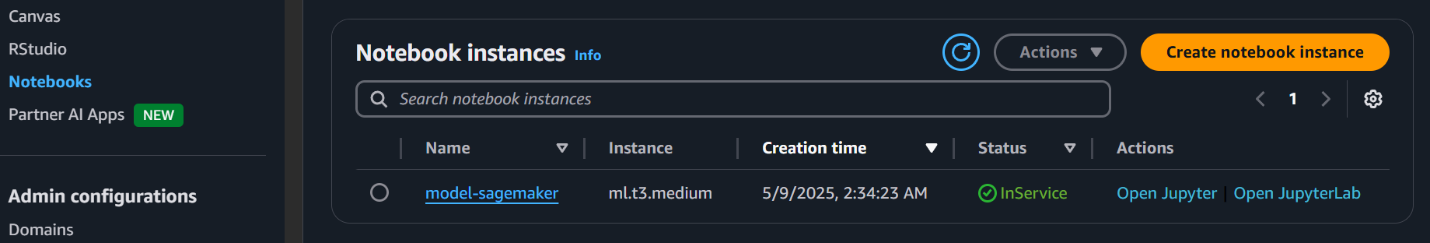
AWSSageMaker, makine öğrenmesi projelerini baştan sona yönetmemizi sağlayan bir bulut hizmetidir.

* Veri hazırlama
* Model Eğitme
* Hiperparametre Optimizasyonu
* Model Dağıtımı

İşlemlerinin hepsini AWS SageMaker hizmetiyle yapmamız mümkündür.

Proje kapsamında amacımız bir model eğitip tahmin işlemleri gerçekleştirebilldiğimiz bir sistem kurmak ve model dağıtımı yapmak olduğu için AWS SageMaker hizmetini kullanmaya karar verdik.

*Şekil 2. Notebook Instance Oluşturmak.*

*Şekil 3. Jupyter Notebook*

Amazon SageMaker’ın en önemli özelliklerinden birisi içerisinde notebook instance oluşturma imkânımız olmasıdır.   
Notebook instance oluşturmayı AWS’nin avantajlarından yararlanarak kullanabildiğimiz bir jupyter notebook oluşturmak olarak ifade edebiliriz, bu sayede bulut üzerinde veri seti yükleme, veri işleme ve model eğitme işlemlerini yapabileceğimiz bir ortama sahip olabiliyoruz.

**Veri Hazırlığı:**

metin, yazı tipi, sayı, numara, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 4. Veri Setini Çekme İşlemi*

*Bu aşamada:*

* Urllib kütüphanesini kullanarak yukarıdaki görselde gözüken web sitesine istekte gönderdik ve model eğitmek için kullanacak olduğumuz iris veri setini indirdik.
* İndirdiğiğimiz veri setini unzip komutunu kullanarak ayıkladık.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 5. Veri Önişleme Süreci*

* Model eğitmeden önce veri setini inceledik ve gerekli veri önişleme işlemlerini gerçekleştirerek eğtim ve doğrulama veri setlerini oluşturduk.

**S3 ve Bucket Düzenlemeleri(Depolama Düzenlemeleri):**

*ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Şekil 6. S3’e Giriş*

Amazon S3(Simple Storage Service), AWS’nin nesne tabanlı depolama hizmetidir.

Eğitim ve doğrulama dosyalarımızı SageMaker’ın eğitim işlerinin doğrudan okuyabileceği şekilde buluta yüklemek model eğitim sürecini sorunsuz yürütmemizi sağlayacak.

S3’e girerek “sagemaker-kurma-ve-dagitma-modeli" isminde bir bucket oluşturduk.

(Bucket AWS S3’te kullandığımız klasör benzeri bir yapıdır.)

*Şekil 7. AWS S3’e Veri Taşıma*

* Boto3 kütüphanesiyle AWS S3’e bağlandık.
* bucket\_name değişkeniyle hedef bucket’ı belirledik, data üzerinde sütun düzenlemesi yaptıktan sonra train\_data ve val\_data’yı data.csv dosyasına kaydettik.
* “data/train/data” ve “data/val/data” anahtarlarıyla S3’teki bucket’ımıza yükleme işlemini gerçekleştirdik.
* Boto3, IAM rolü üzerinden otomatik kimlik doğrulama sağladı.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 8. Data Klasörü*

Ekran görüntülerimizi süreci tamamladıktan sonra aldığımız için model klasörü de gözüküyor ancak normalde bu aşamadayken sadece data klasörüne sahiptik.

**Docker ve Eğtilecek Modelin Şablonu:**

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

*Şekil 9. Docker ve Estimator*

Sagemaker’ın içinde bullunan Estimator, eğitim sürecini tanımladığımız bir python nesnesidir.

* Container oluşturarak stabil bir çalışma ortamı kurguladık ve model eğitiminde hangi algoritmanın kullanılacağını tanımladık.
* Estimator ile oluşturduğumuz Container ortamında hangi ayarlarla model eğitimini yapacağımızı belirttik.

Bu aşamada Hangi algoritma, hangi veri, kaç tane ve ne tip sunucu kullanılacağı, hiperparametreler gibi tanımlamalarımızı yaptık.

**Model Eğitim ve Dağıtım Süreci:**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 10. Model Eğitimi*

Birkaç veri düzenleme işlemi yaptık ve modelimizi eğittik

metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 11. Model Dağıtımı*

Gerçek zamanlı bir tahmin sunucusu kurduk (Endpoint oluşturduk).

Bu sayede dışarıdan istek atılıp, tahmin sonucunun elde edilebileceği sistemi oluşturmuş olduk.

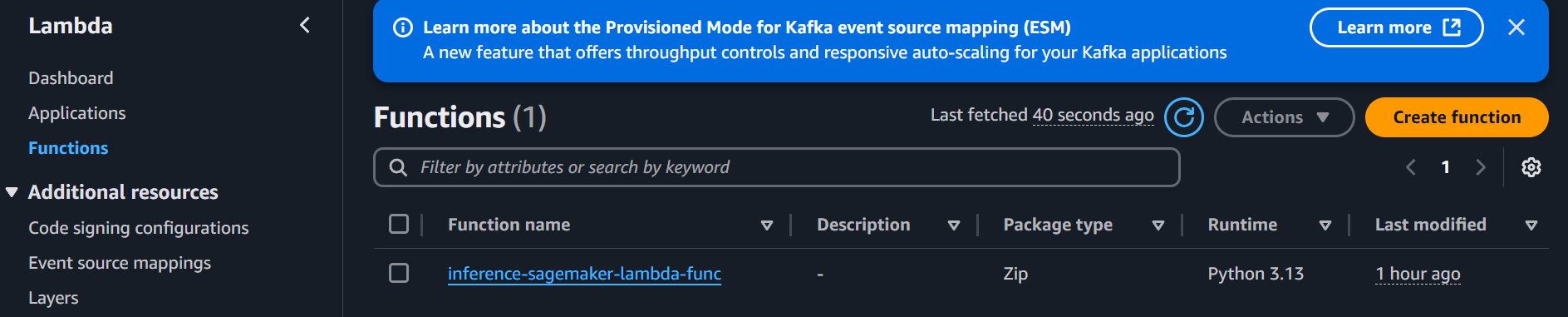
metin, ekran görüntüsü, çizgi, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 12. Oluşan Dosya ve Klasörler*

*metin, yazılım, multimedya yazılımı, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Şekil 13. İki Aşama Önce Oluşturduğumuz Endpoint*

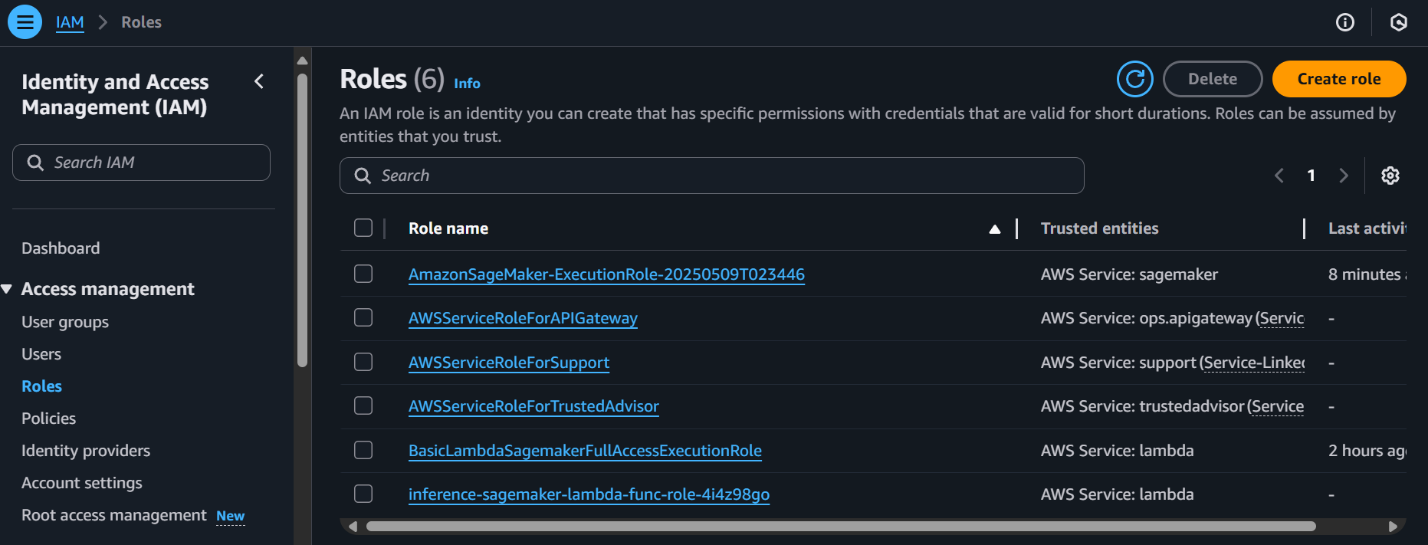
**API Gateway Düzenlemeleri ve Lambda Fonksiyonu:**

*Şekil 14. Oluşturulan Lambda Fonksiyonu*

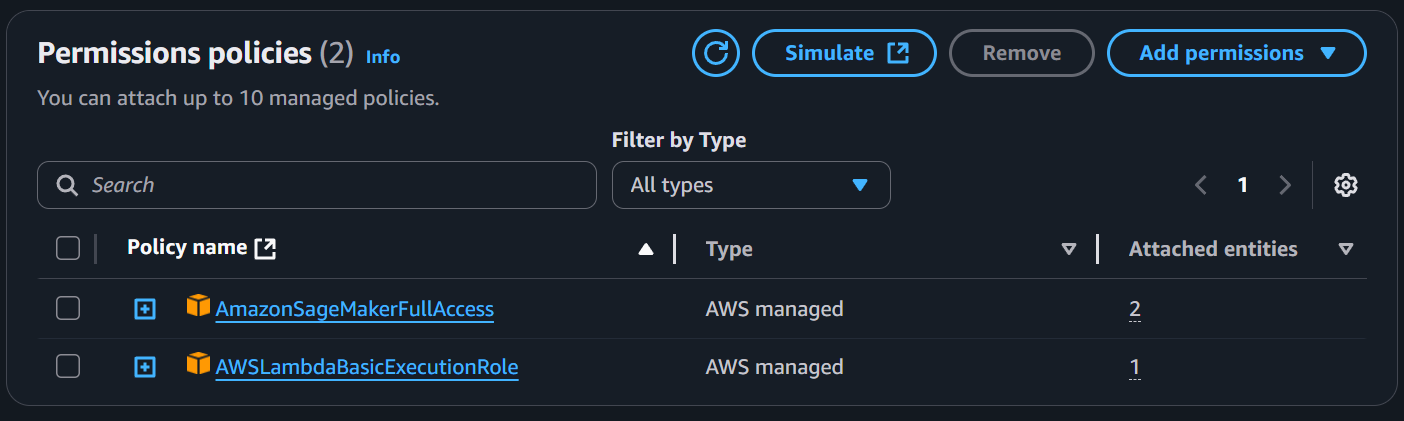
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 15. Lambda Fonksiyonunun Kodu*

Lambda Foknksiyonu yazdığımız kod sayesinde gelen HTTP isteklerinin nasıl yönetileceğini belirler.  
Helen isteği alır eğittiğimiz modelin olduğu endpointe yönlendirir, tahmin sonucunu alır uygun formata çevirir ve istemciye tahmin sonucunu geri gönderir.  
Ayrıca sunucuyu sadece istek atıldığı zamanlarda aktifleştirerek maliyetleri ve enerji tüketimini azaltır.

*Şekil 16. IAM Rol Oluşturma*

Lambda kodu çalıştığında SageMaker endpoint’ine istek atabilsin, S3’ten veri okuyup CloudWatch’a log yazabilsin diye bu tam erişim yetkilerine sahip rolü oluşturduk.

*Şekil 17. Rol için Seçilen Politikalar*

**Uygulama:**

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Şekil 18. API Gateway POST Metodu*

* Oluşturduğumuz endpoint’e istek attık.

*metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Şekil 19. Tahmin Çıktısı*

*metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Şekil 20. All-In-One API Platformu (Postman) Üzerinde Deneme*

Sonuç olarak, bulut ortamında dağıtık çalışabilen ve kendisine istek gönderildiğinde tahmin sonuçları döndürebilen bir model geliştirdik.

**KAYNAKLAR**

1. **Veri Seti:** <https://archive.ics.uci.edu/dataset/53/iris>
2. **XGBoost Hiperparametreleri:** <https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost_hyperparameters.html>