

**Bulut Bilişim ile Akıllı Veri Analitiği ve**

**Makine Öğrenmesi Uygulaması**

**BULUT BİLİŞİM VE UYGULAMALARI BLM3522 DERSİ VİZE ÖDEVİ**

**3.PROJE: AKILLI VERİ ANALİTİĞİ VE MAKİNE ÖĞRENMESİ UYGULAMASI**

**Hazırlayanlar:**

**Muhammed Faruk Gözay 22290673**

**Batuhan Dede 22290502**

**Yavuz Selim Demirhan 22290565**

**Github Repo Adresi:**

**https://github.com/orgs/BLM3522-Grubu/repositories?**

**Projenin Amacı:**

Bu projenin amacı, Python programlama dili ve Scikit-learn makine öğrenmesi kütüphanesi kullanılarak temel bir sınıflandırma modeli geliştirip, bu modeli Google Cloud Platform'un (GCP) BigQuery servisi üzerinde saklanan verilerle entegre ederek çalıştırmaktır. Proje, bulut ortamında veri işleme, analiz etme ve makine öğrenmesi modellerinin uygulanmasını göstermektedir.

Projenin temel hedefleri:

Basit bir makine öğrenmesi modeli geliştirmek.

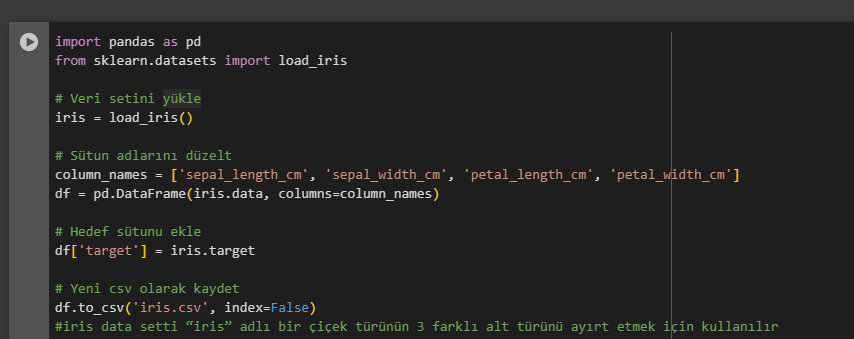
Veriyi Google Cloud üzerinde BigQuery ile işlemek ve analiz etmek.

Modeli Google Colab ortamında eğitmek.

Bulut ortamındaki veri ile model arasında bağlantı kurmak.

Verilerden bilgi çıkarımı yapmak ve sınıflandırma (tahminleme) gerçekleştirmek.

| **Gereksinim** | **Projede Kullanıldı Mı?** | **Açıklama** |
| --- | --- | --- |
| Python, .NET, Java,Node.js, PHP | Python | Python dili kullanıldı. |
| Makine Öğrenmesi Kütüphaneleri | Scikit-learn | Model eğitimi için kullanıldı. |
| Veritabanı: BigQuery | BigQuery | Veriler burada saklandı ve işlendi. |
| Bulut Platformları: Google Cloud Platform | Google Cloud | BigQuery üzerinden veri işleme yapıldı. |
| Model Eğitimi | Evet | Iris veri seti ile sınıflandırma modeli geliştirildi. |
| Bulutta Model Kullanımı / Veri Çekimi | Evet | BigQuery'den veri çekildi ve analiz yapıldı. |



* pandas, veri analizi ve veri işleme için kullanılan güçlü bir Python kütüphanesidir. Veri çerçeveleri (DataFrame) oluşturarak tablo gibi veri yapılarıyla çalışmamıza olanak tanır.
* sklearn.datasets içindeki load\_iris() fonksiyonu, içinde hazır gelen meşhur *Iris* veri setini yüklememizi sağlar.
* İris değişkeni artık *Iris veri seti*ni içeriyor.
* Bu veri seti içinde:
  + iris.data: Giriş özellikleri (4 sütun – sepal/petal ölçüleri)
  + iris.target: Etiket (çıktı sınıfı – 0, 1, 2)
  + iris.feature\_names: Özellik adları
  + iris.target\_names: Sınıf adları (çicek türleri: setosa, versicolor, virginica)

Iris veri setindeki özelliklerin isimlerini daha okunabilir ve standart hale getiriyoruz:

* **Sepal** (çanak yaprak): çiçeğin dışındaki büyük yapraklar
* **Petal** (taç yaprak): içteki renkli yapraklar

DataFrame oluşturularak iris verisi tablo biçimine sokuluyor:

* sepal\_length\_cm: Çanak yaprak uzunluğu
* sepal\_width\_cm: Çanak yaprak genişliği
* petal\_length\_cm: Taç yaprak uzunluğu
* petal\_width\_cm: Taç yaprak genişliği

target sütunu eklenerek her gözlemin hangi çiçek türüne ait olduğu belirtiliyor.

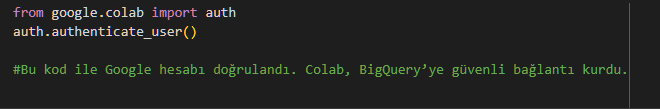
Tüm tablo .csv formatında dışa aktarılıyor.

iris.csv adlı dosya artık iris verisinin düzenli halini içeriyor.

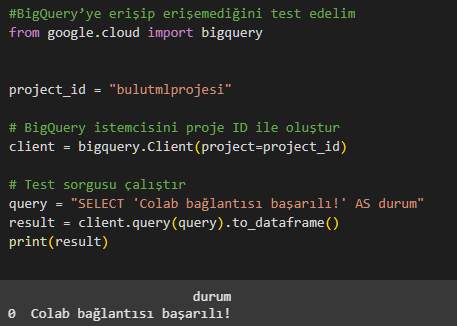
index=False ifadesi sayesinde satır numaraları dosyaya yazılmıyor.



Gerekli kütüphaneleri kurduk



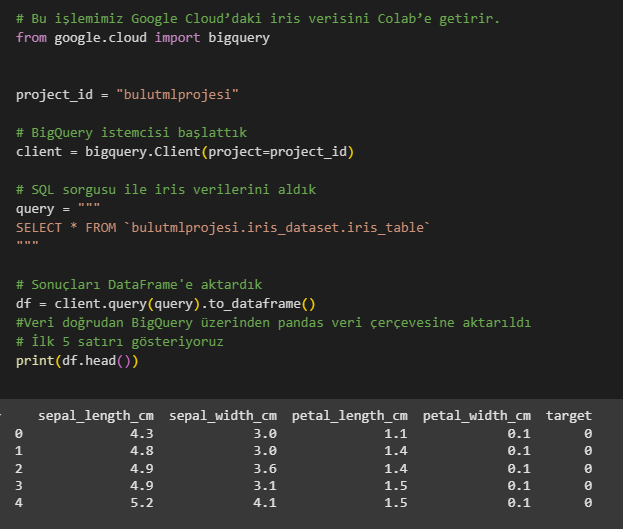
Bu kod, Google Colab ile Google hesabımı bağlamamı sağladı. Böylece Colab, Google Cloud servislerine erişim hakkı kazandı



Bu kod bloğu, Google Colab ile Google Cloud üzerindeki BigQuery servisi arasında bağlantı kurup kurulamadığını test etmek için kullanılır. İlk olarak google.cloud içinden BigQuery istemci kütüphanesi içe aktarılır. Ardından, daha önce Google Cloud Console üzerinden oluşturulmuş olan projenin ID’si tanımlanır ve bu ID ile BigQuery istemcisi (client) oluşturulur. Bu istemci sayesinde BigQuery üzerinde SQL sorguları çalıştırılabilir.

Kodda kullanılan örnek SQL sorgusu, sadece “Colab bağlantısı başarılı!” metnini döndüren basit bir sorgudur ve test amacıyla yazılmıştır. Bu sorgu, BigQuery’ye başarıyla erişildiğini gösteren bir onay mesajı üretir. client.query(...).to\_dataframe() kısmı, bu sorgunun sonucunu tablo (DataFrame) formatında elde eder ve print(...) ile ekrana yazdırır. Eğer bu çıktı sorunsuz gelirse, BigQuery bağlantısı doğru şekilde kurulmuş demektir ve artık veri analiz işlemlerine başlanabilir.

.



**from google.cloud import bigquery**  
BigQuery kütüphanesini içe aktarır. Bu, Google Cloud'daki veri setlerine erişmek için gereklidir.

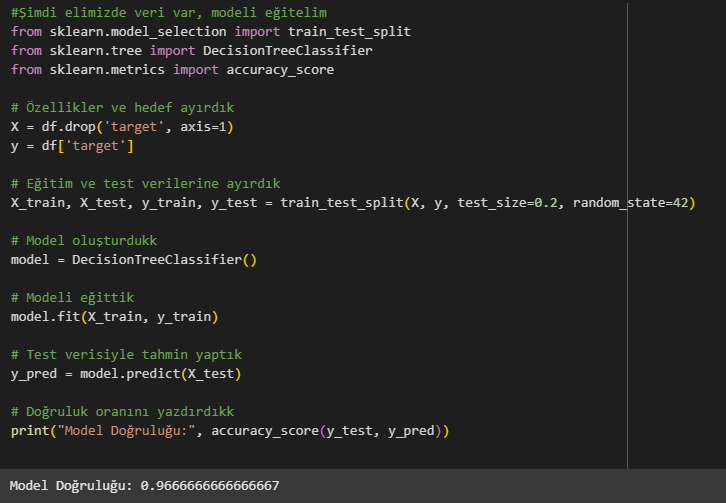
🔸 **project\_id = "bulutmlprojesi"**  
Bu satırda, BigQuery projemizin kimliği (ID'si) tanımlanır. Google Cloud Console'da oluşturduğun proje adı buraya yazılır.

🔸 **client = bigquery.Client(project=project\_id)**  
Burada bigquery.Client() sınıfı kullanılarak bir istemci nesnesi (client) oluşturulur. Bu istemci, Colab ile BigQuery arasında bağlantı kurar.

🔸 **query = """ SELECT \* FROM bulutmlprojesi.iris\_dataset.iris\_table """**  
Bu SQL sorgusu, BigQuery içindeki iris\_dataset veri kümesinde yer alan iris\_table adlı tablonun **tüm verilerini almak** için yazılmıştır. SELECT \* ifadesi tüm sütunları getir demektir.

🔸 **df = client.query(query).to\_dataframe()**  
Bu satırda, yukarıda yazılan SQL sorgusu BigQuery’ye gönderilir ve sonuçlar df adlı bir **Pandas DataFrame** (veri tablosu) olarak alınır. Artık bu veri Python ortamına gelmiştir ve üzerinde analiz yapılabilir.

🔸 **print(df.head())**  
Veri çerçevesinin (df) ilk 5 satırını ekrana yazdırır. Bu, verinin başarılı şekilde geldiğini görmemizi sağlar.



from sklearn.model\_selection import train\_test\_split  
Veri kümesini eğitim ve test setlerine ayırmak için gerekli fonksiyonları içe aktarır.

from sklearn.tree import DecisionTreeClassifier  
Karar ağacı (Decision Tree) sınıflandırıcısını (yani makine öğrenmesi algoritmasını) içe aktarır.

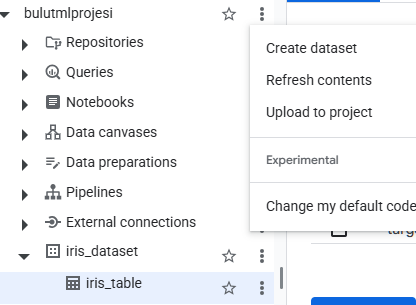
from sklearn.metrics import accuracy\_score  
Modelin doğruluğunu ölçmek için kullanılan fonksiyonu getirir.

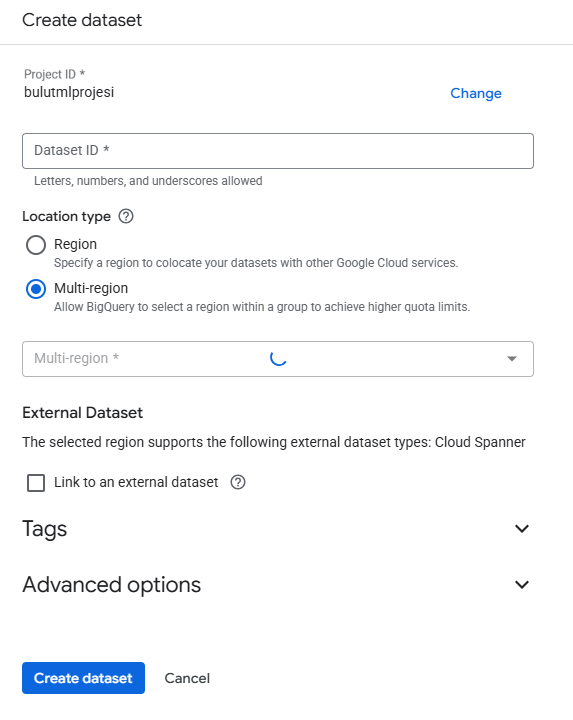
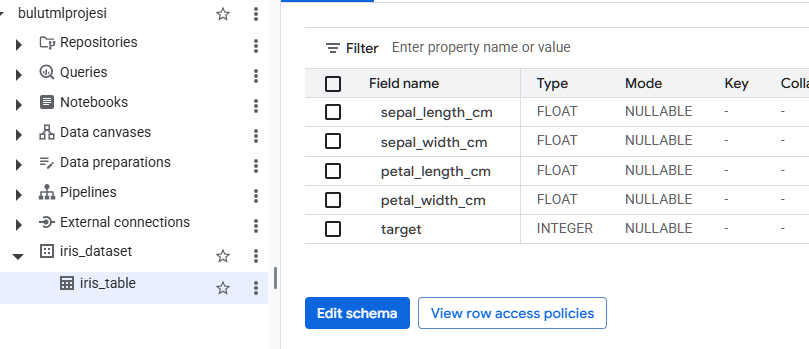
🔷 **Veri Hazırlığı**:

* X = df.drop('target', axis=1)  
  Bağımsız değişkenler yani özellikler (sepal ve petal ölçüleri) alınır.
* y = df['target']  
  Bağımlı değişken yani tahmin edilmek istenen sınıf bilgisi (0, 1, 2) alınır.
* X\_train, X\_test, y\_train, y\_test = train\_test\_split(...)  
  Veri %80 eğitim, %20 test olmak üzere ayrılır. random\_state=42 ise her çalıştırmada aynı sonucu vermesi için sabitlenmiş bir değerdir.

🔷 **Model Eğitimi ve Değerlendirme**:

* model = DecisionTreeClassifier()  
  Bir karar ağacı sınıflandırıcısı nesnesi oluşturulur.
* model.fit(X\_train, y\_train)  
  Model, eğitim verisi ile eğitilir. Yani çiçek ölçülerine göre türünü tahmin etmeyi öğrenir.
* y\_pred = model.predict(X\_test)  
  Eğitilen model, test verisi üzerinde tahmin yapar.
* print("Model Doğruluğu:", accuracy\_score(y\_test, y\_pred))  
  Gerçek test verileriyle tahmin sonuçları karşılaştırılır ve modelin doğruluk oranı yazdırılır.



* Yeni bir proje oluşturuldu (örneğin: bulutmlprojesi).
* Bu proje, BigQuery dahil tüm işlemleri barındıracak ana yapıdır.
* Google Cloud Console üzerinden BigQuery servisi etkinleştirildi.
* Sol menüde “BigQuery” sekmesine tıklanarak platform açıldı.
* Veri Kümesi Kimliği olarak iris\_dataset yazıldı.
* Lokasyon “US” olarak bırakıldı.
* “Oluştur”a tıklandı.
*  iris\_dataset üzerine tıklandı.
*  “Tablo Oluştur” seçildi.
*  Kaynak olarak “Dosyadan yükle” (CSV olarak iris.csv dosyası yüklendi).
*  Tablo adı: iris\_table olarak girildi.
* 
* from google.colab import auth ile kimlik doğrulama yapıldı.
* Kullanıcı Google hesabıyla oturum açarak Colab’in Google Cloud kaynaklarına erişmesine izin verdi.

iris\_table tablosundaki veriler SQL ile çekildi

Sonuçlar pandas.DataFrame olarak Python ortamına aktarıldı.

Bu işlemler sayesinde:

* Bulutta veri saklama ✔
* Buluttan veri işleme ve analiz ✔
* Python + Google Cloud entegrasyonu ✔

| **Gereksinim** | **Projede Kullanıldı Mı?** | **Açıklama** |
| --- | --- | --- |
| Python, .NET, Java,Node.js, PHP | Python | Python dili kullanıldı. |
| Makine Öğrenmesi Kütüphaneleri | Scikit-learn | Model eğitimi için kullanıldı. |
| Veritabanı: BigQuery | BigQuery | Veriler burada saklandı ve işlendi. |
| Bulut Platformları: Google Cloud Platform | Google Cloud | BigQuery üzerinden veri işleme yapıldı. |
| Model Eğitimi | Evet | Iris veri seti ile sınıflandırma modeli geliştirildi. |
| Bulutta Model Kullanımı / Veri Çekimi | Evet | BigQuery'den veri çekildi ve analiz yapıldı. |