



Ankara Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
Bulut Bilişim ve Uygulamaları Proje Dokümanı

Timur Malkoç

22290719

<https://github.com/TimurMalkc/Cloud-Midterm-ML>

Tuğrul Özgün

22290082

<https://github.com/Forcibus/todoapp>

Proje 3: Akıllı Veri Analitiği ve Makine Öğrenmesi Uygulaması

Proje içerisinde back-end dili olarak Python, makine öğrenmesi kütüphanesi olarak Scikit-learn, veritabanı olarak BigQuery ve bulut platformu olarak Google Cloud kullanılmıştır. Kullanılacak veri seti olarak Kaggle üzerinden alınmış “Mushroom Classification” seçilmiştir.



Veri seti indirildikten sonra bir dataset ve table oluşturulup veri seti, BigQuery üzerine aktarılmıştır. BigQuery hali hazırda Google Cloud üzerinde bulunmaktadır ve bağlantı buradan sağlanmıştır.

The image shows the Google Cloud BigQuery interface. The 'maintable' dataset is selected in the Explorer. The table is displayed in a grid view with columns: Row, class, cap-shape, cap-surface, cap-color, bruises, odor, and gill-at. The table contains 14 rows of data. The interface includes a search bar, a 'Query' button, and various action buttons like 'Open in', 'Share', 'Copy', 'Snapshot', 'Delete', and 'Export'. The 'Schema' tab is active, showing the table's structure. The 'Table Explorer' tab is also visible, showing the table's details. The 'Insights' tab is also visible, showing the table's lineage. The 'Data Profile' and 'Data Quality' tabs are also visible. The 'Results per page' is set to 50, and the table shows 1 - 50 of 8124 results.

Row	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-at
1	p	x	y	e	false	m	f
2	p	k	y	c	false	m	a
3	p	k	y	n	false	m	f
4	p	f	y	e	false	m	a
5	p	k	y	e	false	m	f
6	p	k	y	c	false	m	f
7	p	x	y	n	false	m	a
8	p	f	y	n	false	m	f
9	p	x	y	c	false	m	a
10	p	k	y	n	false	m	a
11	p	f	y	c	false	m	f
12	p	f	y	n	false	m	f
13	p	f	y	e	false	m	f
14	p	f	y	n	false	m	a

Python programlaması için ise tekrardan Google Cloud üzerinde bulunan Jupyter Notebook kullanılmıştır. Jupyter Notebook kullanımı için Vertex AI platformu üzerinden bir workbench instance açılmıştır.

The image shows the Google Cloud Vertex AI Workbench Instances page. The page displays a table of instances. The table has columns: Filter, Instance name, Zone, Auto upgrade, Version, Machine Type, GPUs, and Owner. The table contains one instance named 'proj-instance' in the 'us-central1-a' zone. The instance is configured with 'M129' version, 'Efficient Instance: 4 vCPUs, 16 GB RAM' machine type, and 'None' GPUs. The owner is '911470216392-compute@developer.gserviceaccount.com'. The page also includes a 'Filter' button and a 'Learn more' link.

Filter	Instance name	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner
<input checked="" type="checkbox"/>	proj-instance	us-central1-a	—	M129	Efficient Instance: 4 vCPUs, 16 GB RAM	None	911470216392-compute@developer.gserviceaccount.com

Kod, verinin her seferinde tekrar tekrar yüklenmemesi için ve okunabilirlik açısından 4 Jupyter Notebook hücresine bölünmüştür. Birinci hücrede gerekli kütüphaneler ve metotlar import edilip veri tabanı BigQuery üzerinden çekilmiştir.

```
[26]: import pandas as pd
import pandas_gbq as pdq
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

sql = "SELECT * FROM `cloud-midterm-ml.MidtermMLDataset.maintable`"
mushroomData = pdq.read_gbq(sql)

mushroomData

Downloading: 100%|██████████|
```

```
[26]:
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	p	x	y	e	False	m	f	c	b	w	...	y	c	c	p	w	n	n	w	c	d
1	p	k	y	c	False	m	a	c	b	w	...	y	c	c	p	w	n	n	w	c	d
2	p	k	y	n	False	m	f	c	b	w	...	y	c	c	p	w	n	n	w	c	d
3	p	f	y	e	False	m	a	c	b	w	...	y	c	c	p	w	n	n	w	c	d
4	p	k	y	e	False	m	f	c	b	y	...	y	c	c	p	w	n	n	w	c	d
...
8119	e	f	y	n	True	n	f	c	b	w	...	s	e	e	p	w	t	e	w	c	w
8120	e	k	s	e	True	n	f	c	b	e	...	s	e	e	p	w	t	e	w	c	w
8121	e	x	y	n	True	n	f	c	b	w	...	s	e	w	p	w	t	e	w	c	w

İkinci hücrede kategorik verinin işlenebilmesi için LabelEncoder metodu kullanılarak veriler her birine karşılık gelen sayılar ile kodlanmıştır. Hangi verinin hangi sayıya karşılık geldiği de enumerator aracılığı ile çıktı olarak gösterilmiştir. Veri setinin sütunlarının tamamının kullanılması overfitting bir modele sebep olmasından ve benzer özelliklerin modeli gereksiz karmaşılaştırmasından dolayı tüm sütunların içerisinde 6 tanesi seçilip modelde kullanılmıştır.

```
[27]: selectedCols = ['class', 'bruises', 'gill-attachment', 'gill-spacing', 'gill-size', 'stalk-shape']
LabelEncoder = LabelEncoder()

for col in selectedCols:
    mushroomData[col] = LabelEncoder.fit_transform(mushroomData[col])
    print(f"\nEncoding for column: {col}")
    for i, class_ in enumerate(LabelEncoder.classes_):
        print(f"{class_} => {i}")

Encoding for column: class
e => 0
p => 1

Encoding for column: bruises
0.0 => 0
1.0 => 1

Encoding for column: gill-attachment
a => 0
f => 1

Encoding for column: gill-spacing
c => 0
w => 1

Encoding for column: gill-size
b => 0
n => 1

Encoding for column: stalk-shape
e => 0
t => 1
```

Üçüncü hücrede Sci-kit learn makine öğrenmesi metotları kullanılarak model eğitilmiştir.

```
X = mushroomData[['bruises', 'gill-attachment', 'gill-spacing', 'gill-size', 'stalk-shape']]
y = mushroomData['class']
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=42)
```

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)
```

```
y_pred = knn.predict(x_test)
```

```
print("\nKNN Accuracy:", accuracy_score(y_test, y_pred))
```

```
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
KNN Accuracy: 0.9098966026587888
```

```
Classification Report:
```

	precision	recall	f1-score	support
0	0.89	0.94	0.92	2123
1	0.93	0.88	0.90	1939
accuracy			0.91	4062
macro avg	0.91	0.91	0.91	4062
weighted avg	0.91	0.91	0.91	4062

Son hücrede kullanıcıdan girdi alma sistemi eklenmiştir ve belirtilen özelliklere göre model bir mantarın zehirli mi değil mi olduğunu söyleyebilmektedir.

```
bruises = int(input("Bruises (1 for Yes, 0 for No): "))
gillA = int(input("Gill Attachment: (1 for Free, 0 for Attached): "))
gillSP = int(input("Gill Spacing: (1 for Crowded, 0 for Close): "))
gillSZ = int(input("Gill Size: (1 for Narrow, 0 for Broad): "))
stalk = int(input("Stalk Shape: (1 for Tapering, 0 for Enlarging): "))

mushroomInput = pd.DataFrame({'bruises': [bruises], 'gill-attachment': [gillA], 'gill-spacing': [gillSP],
                              'gill-size': [gillSZ], 'stalk-shape': [stalk]})
```

```
pred = knn.predict(mushroomInput)
```

```
if pred == 0:
    print("This mushroom is poisonous")
else:
    print("This mushroom is edible")
```

```
Bruises (1 for Yes, 0 for No): 1
Gill Attachment: (1 for Free, 0 for Attached): 0
Gill Spacing: (1 for Crowded, 0 for Close): 0
Gill Size: (1 for Narrow, 0 for Broad): 1
Stalk Shape: (1 for Tapering, 0 for Enlarging): 0
This mushroom is edible
```

Proje 1: Çift Katmanlı Web Uygulaması (Web API + Frontend)

Task Manager Uygulaması (Frontend + Backend Deployment)

Bu proje, kullanıcıların görev (task) oluşturmalarını, güncellemesini, silmesini ve filtrelemesini sağlayan çift katmanlı bir web uygulamasıdır. Arka uç RESTful API olarak Node.js (Express) ile geliştirilmiş, ön yüz ise React.js kullanılarak inşa edilmiştir. Veriler MongoDB veritabanında saklanmaktadır ve AWS de deploylanmaktadır. Veriler güvenlik için JWT encryption ile şifrelenmektedir. Proje, ölçeklenebilirlik ve erişilebilirlik açısından AWS EC2 ortamında barındırmaya uygun olarak yapılandırılmıştır.

Uygulama Özellikleri

- Görev ekleme (başlık ve tarih ile)
- Görev düzenleme (başlık, tamamlanma durumu, tarih)
- Görev silme
- Tamamlanma durumuna ve son teslim tarihine göre filtreleme
- Karanlık/Aydınlık mod desteği
- Tarih bilgilerini Türkçe formatta görüntüleme

Görevler

Yeni görev ekle

gg . aa . yyyy



Ekle

Duruma göre filtrele: Hepsi

Son tarihe göre filtrele:

gg . aa . yyyy



ee



Tamamlandı

01.05.2025

Durumu Değiştir

Sil

dd



Bekliyor

07.05.2025

Durumu Değiştir

Sil

cc



Bekliyor

12.05.2025

Durumu Değiştir

Sil

bb



Tamamlandı

24.05.2025

Durumu Değiştir

Sil

aa



Tamamlandı

03.05.2025

Durumu Değiştir

Sil

Bulut Sunucuları (1/2) Bilgi

Son güncelleme
1 minute önce



Bağlan

Bulut sunucusu durumu

Eylemler

Bulut sunucularını başlatın



Bulut Sunucusu Özniteliğe veya etikete göre (case-sensitive) bulun

Tüm duru...

< 1 >



	Name	Bulut sunucusu kimliği	Bulut sunucusu...	Bulut sunuc...	Durum kontrolü	Alarm durumu	Erişilebilirlik AL...
<input checked="" type="checkbox"/>	todo-app--inst...	i-05984ae76aae52490	Çalışıyor	t3.micro	3/3 denetim geçti	Alarmları görünt	eu-north-1a
<input type="checkbox"/>	todo1	i-004ecbef3e375bab2	Sonlandırıldı	t3.micro	-	Alarmları görünt	eu-north-1a