



**T.C**  
**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ**

**PROJE KONUSU: PROGRAM DERS İLİŞKİ MATRİSİ**  
**PROJESİ**

**ÖęRENCİ ADI/SOYADI, NUMARASI:**

**220502005 Aybüke Türidi**  
**220502033 Elif Beyza Beyaz**  
**220502043 Serhat Arslaner**  
**220501022 Dila Seray Tegün**  
**220501031 Ahmetcan Bostancı**

**DERS SORUMLUSU:**  
**PROF. DR./DR. ÖęR. ÜYESİ ELİF PINAR HACİBEYOęLU**

**TARİH: 29.12.2024**

# 1 GİRİŞ

## 1.1 Projenin amacı

Bu proje, öğrenci başarılarını ders ve program çıktıları üzerinden daha doğru ve detaylı bir şekilde analiz etmeyi amaçlamaktadır. Geleneksel not sistemleri öğrenci performansını değerlendirmede yetersiz kalabilirken, bu projede öğrencilerin ders ve program hedeflerine ne ölçüde ulaştıkları daha kapsamlı bir bakış açısıyla ele alınmıştır.

Projede, ders çıktıları ve program çıktıları arasındaki bağlantılar detaylı matrislerle tanımlanmış ve bu matrisler kullanılarak ağırlıklı değerlendirme tabloları oluşturulmuştur. Öğrencilerin ödev, quiz, vize ve final gibi farklı değerlendirme kriterlerinden aldıkları notlar, belirlenen ağırlıklar doğrultusunda analiz edilmiş ve her bir öğrencinin ders ve program hedeflerine katkısı ortaya konmuştur.

Proje sürecinde veri girişi, analiz ve çıktı işlemleri Python programlama dili kullanılarak otomatikleştirilmiş ve sonuçlar Excel tabloları olarak raporlanmıştır. Bu yaklaşım, hata oranını minimize etmiş ve analiz sürecini hızlandırmıştır. Proje sonunda elde edilen veriler, eğitim programlarının etkinliğini değerlendirme ve iyileştirme çalışmalarına katkı sağlamayı hedeflemektedir.

## 2 GEREKSİNİM ANALİZİ

Projenin amacına ulaşabilmesi için belirli teknik ve işlevsel gereksinimlerin yerine getirilmesi gerekmektedir. Bu gereksinimler; veri girişi, analiz süresi ve raporlama adımlarını kapsamaktadır.

### 2.1 Arayüz gereksinimleri

Projenin veri giriş ve çıktı işlemleri, kullanıcıların kolaylıkla erişebileceği Excel dosyaları üzerinden gerçekleştirilmiştir. Şuanda herhangi bir grafiksel arayüz tasarlanmamıştır Excel dosyaları üzerinden kullanıcılar, öğrenci notlarını ve değerlendirme kriterlerini ilgili Excel tablolarına girerek veri akışını başlatırlar.

- Kullanıcıların Excel tabloları üzerinden veri giriş ve güncelleme yapabilmesi.
- Python aracılığıyla analiz süreçlerinin başlatılabilmesi.
- Tabloların Excel formatında çıktı olarak sunulabilmesi.

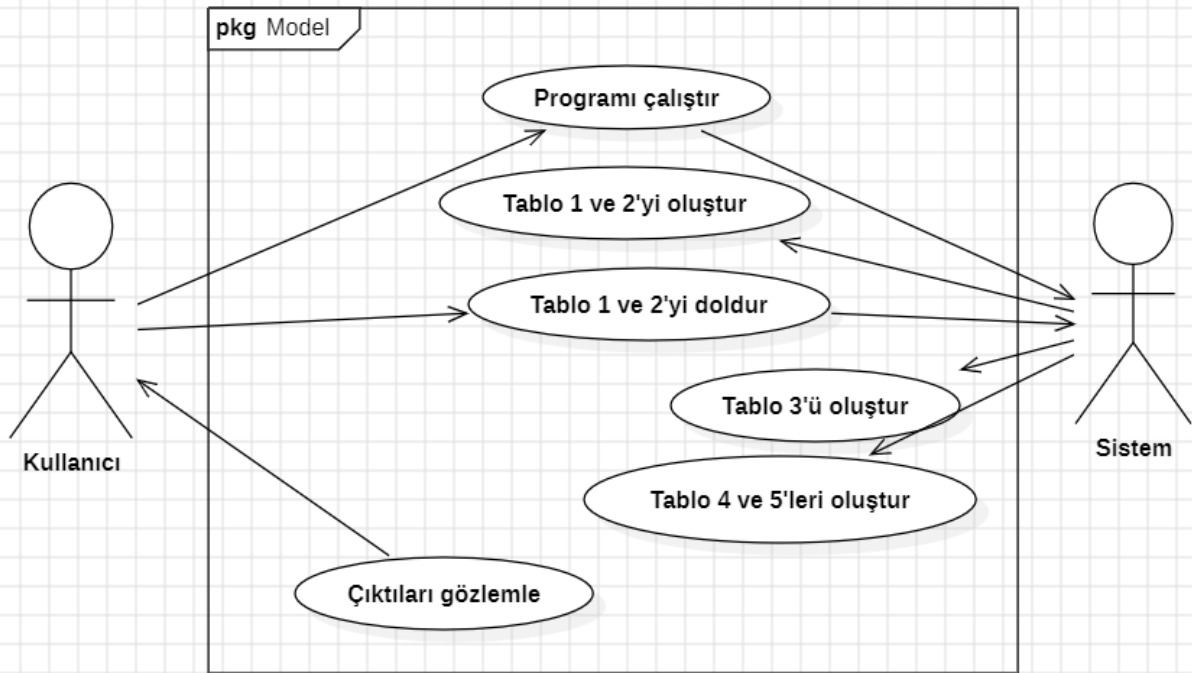
## 2.2 Fonksiyonel gereksinimler

Fonksiyonel gereksinimlerde projenin çalışma mantığına uygun olarak olacak özellikler belirtilir. Bu proje öğrenci not sistemi olmasından kaynaklı oluşacak tablo çeşitlerine göre fonksiyonel gereksinimler belirtilebilir.

- Tablo1 ve Tablo2'nin kod yardımıyla oluşturulması.
- Tablo3'te ders çıktılarına ait ağırlıklı değerlendirme sonuçlarının hesaplanması.
- Tablo4'te her bir öğrencinin ders çıktıları başarı oranlarının hesaplanması ve kaydedilmesi.
- Tablo5'te her bir öğrencinin program çıktıları başarı oranlarının hesaplanması
- Her öğrenci için bireysel Excel çıktılar oluşturularak dosya yapısının düzenli olması sağlanır.

## 2.3 Use-Case diyagramı

Aşağıda belirttiği üzere projeye uygun Use-Case diyagramı eklenmiştir.



Şekil 1

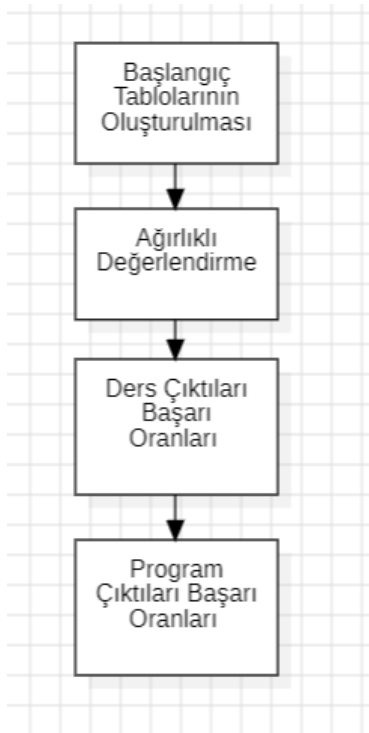
## 3 TASARIM

### 3.1 Mimari tasarım

Proje, verilerin Excel dosyaları aracılığıyla işlenmesi ve çıktıların organize edilmesine dayalı bir sistemdir. Sistem, veri girişi (Tablo1 ve Tablo2), veri işleme (Tablo3) ve veri çıktısı (Tablo4 ve Tablo5) olmak üzere 3 temel bölüme ayrılmıştır. Kodlar, modüler şeklinde tasarlanmış ve işlemler fonksiyonlara ayrılmıştır.

- Veri girişi için gerekli olan verilerin kullanıcılar tarafından Excel dosyaları aracılığıyla sisteme girilmesi sağlanmış olur. Girdiler alınarak tablo 1 ve 2'yi oluşturmuş olur.(Önceden Python scripti çalıştırılır.)
- Girdiğimiz verileri kullanmak için işleme yapmamız gerekmektedir. Bu katmanda Python scriptleri kullanılarak veri analizleri ve hesaplamalar gerçekleştirilir.
- İşlenen veriler tablo 3,4 ve 5 olarak Excel çıktısı halinde elde edilir.

Sistem akışını aşağıdaki gibi gözlemleyebiliriz.



Şekil 2

## 3.2 Kullanılacak teknolojiler

- **Python:** Veri işleme ve analiz süreçlerinin gerçekleştirilmesi için kullanılmıştır.
- **Pandas:** Tablo verilerinin yönetimi, düzenlenmesi ve analiz edilmesi için kullanılmıştır. Bunları DataFrame ile gerçekleştirmiştir.

```
df=pd.DataFrame(data)

df.to_excel("tablo1.xlsx",index=False)
```

Şekil 3

- **NumPy:** Hesaplamalar için kullanılmıştır. Boş değer gibi özel değerler ve hesaplamalar için NumPy kullanılmıştır.

```
StudentDataFrame=pd.DataFrame(np.ones((5,5)),c
```

Şekil 4

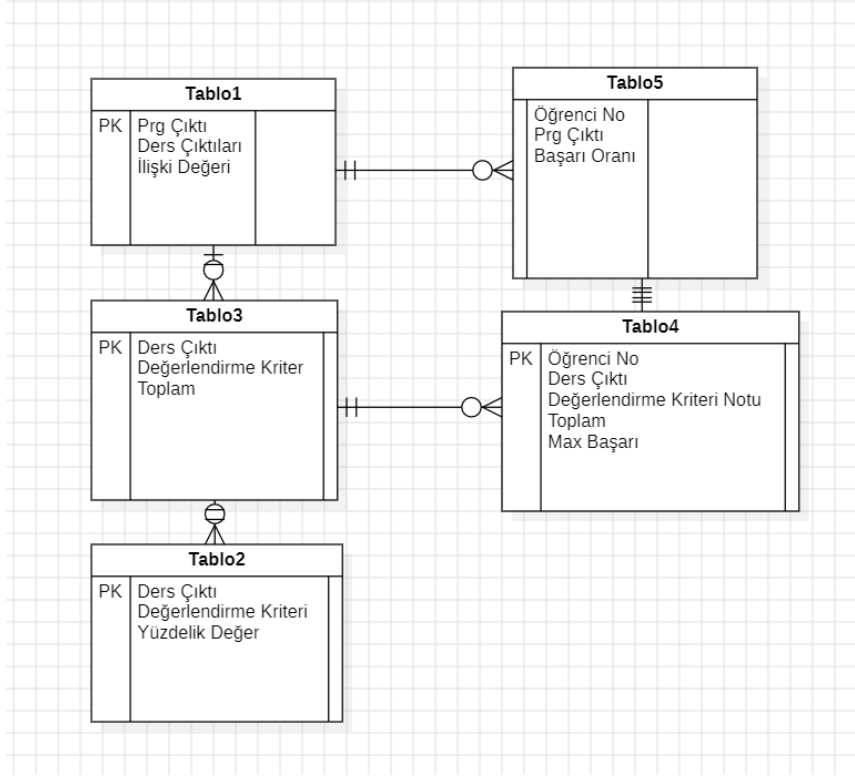
- **Excel:** Veri girişi ve raporlama.
- **OpenPyXL:** Excel dosyaları üzerinde işlem yapmak için Pandas tarafından arka planda kullanılmıştır.

## 3.3 Veri tabanı tasarımı

Proje kapsamında veriler Excel dosyaları üzerinden organize edilmiştir. Her bir tablo farklı bir analiz amacına hizmet etmektedir:

- Tablo1: Program çıktıları ve ders çıktıları arasındaki ilişki matrisini tutar.
- Tablo2: Ders çıktıları ve değerlendirme kriterleri arasındaki ağırlıkları kayıt eder.
- Tablo3: Ağırlıklı hesaplamaların sonucu olan toplam değerleri saklar.
- Tablo4 ve Tablo5: Her bir öğrenci bazında çıktıların hesaplandığı tabloları içerir.

Buna uygun ERD diyagramı aşağıdaki gibi oluşturulur.



Şekil 5

### 3.4 Kullanıcı arayüzü tasarımı

Proje, arayüz olarak doğrudan Excel dosyaları üzerinden veri giriş ve çıktı işlemlerini gerçekleştirir. Python scriptleri arka planda çalışarak analiz ve hesaplamaları otomatik hale getirir.

Proje, terminal veya bir Python IDE çalıştırılabilir. Kullanıcı, sırasıyla aşağıdaki Python dosyalarını çalıştırmalıdır:

- **tablo1\_tablo2\_create.py:** Bu dosya, temel veri tablolarını, Tablo1 ve Tablo2 oluşturur. Oluşturulan bu tablolara kullanıcı girişlerini gerçekleştirir.

Prg Çıktı	1	2	3	4	5	İlişki Değ.
1	1	1	0	1	1	0,8
2	0	0	1	0	0	0,24
3	0	0	1	1	0	0,30
4	0	0	0	1	1	0,36
5	0	1	0	1	0	0,40
6	1	0	0	0	1	0,40
7	0	0	0	1	1	0,40
8	0	1	1	1	1	0,80
9	1	0	1	1	0	0,60
10	0	1	1	1	0	0,60

Şekil 6

Ders Çıktı	Ödev1	Ödev2	Quiz	Vize	Final	
TABLO 2/ORAN	10	10	10	30	40	
1	1	0	1	0	0	
2	1	1	1	0	1	
3	0	1	0	1	1	
4	0	0	0	1	1	
5	0	0	1	1	1	

Şekil 7

- Düzenleme gerekli ise tablo1.xlsx ve tablo2.xlsx elle düzenlenebilir.
- tablo3\_tablo4\_tablo5\_create.py**: Ağırlıklı hesaplamaları yapar, buna göre Tablo3 oluşturulur ve öğrenci bazlı Tablo4 ve Tablo5 dosyalarını oluşturur.
- Tablo1,2 ve 3 tek bir dosya halinde çıktı verir. Tablo 4 ve 5 için her öğrencinin farklıdır. Örnek öğrenci çıktısını aşağıdaki gibi gözlemleyebiliriz.

tablo4_208796934.xlsx								
tablo4 > tablo4_208796934.xlsx								
	A	B	C	D	E	F	G	H
1	Ödev1	Ödev2	Quiz	Vize	Final	TOPLAM	MAX	% Başarı
2	8.10	0	7.20	0	0	15.30	20	76.50
3	8.10	5.90	7.20	0	21.60	42.80	70	61.14
4	0	5.90	0	22.20	21.60	49.70	80	62.13
5	0	0	0	22.20	21.60	43.80	70	62.57
6	0	0	7.20	22.20	21.60	51	80	63.75

Şekil 8

tablo5 > tablo5_208796934.xlsx							
	A	B	C	D	E	F	G
1	Prg Çıktı	76.5-a	61.1-b	62.1-c	62.6-d	63.7-e	Başarı Oranı
2	1	76.50	61.10	0	62.60	63.70	65.98
3	2	0	0	62.10	12.52	0	62.18
4	3	0	0	31.05	62.60	0	62.43
5	4	0	0	0	50.08	63.70	63.21
6	5	0	61.10	0	62.60	0	61.85
7	6	76.50	0	0	0	63.70	70.10
8	7	0	0	0	62.60	63.70	63.15
9	8	0	61.10	62.10	62.60	63.70	62.37
10	9	76.50	0	62.10	62.60	0	67.07
11	10	0	61.10	62.10	62.60	0	61.93

Şekil 9

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

#### TABLO1-TABLO2

##### 1. Kütüphane İçe Aktarımı

```
1 import pandas as pd
2
3 import numpy as np
```

- **Pandas (pd)**: Veri manipülasyonu ve analizi için kullanılır. Bu kodda, tabloları oluşturup Excel dosyasına yazmak için kullanılıyor.
- **NumPy (np)**: Sayısal işlemler için kullanılır. Burada, tabloların belirli hücrelerini boş değerle doldurmak için np.nan kullanılıyor.

##### 2. tablo2Create Fonksiyonu

Bu fonksiyon **Tablo 2**'yi oluşturur ve "tablo2.xlsx" adlı bir Excel dosyasına kaydeder.

##### 1. Veri Sözlüğü (data)

```
7 data = {
8     "Ders Çıktı": [1, 2, 3, 4, 5],
9     "Ödev1": [np.nan, np.nan, np.nan, np.nan, np.nan],
10    "Ödev2": [np.nan, np.nan, np.nan, np.nan, np.nan],
11    "Quiz": [np.nan, np.nan, np.nan, np.nan, np.nan],
12    "Vize": [np.nan, np.nan, np.nan, np.nan, np.nan],
13    "Final": [np.nan, np.nan, np.nan, np.nan, np.nan]
14 }
```

- **Sütunlar**: "Ders Çıktı", "Ödev1", "Ödev2", "Quiz", "Vize", "Final".
- "Ders Çıktı" sütunu 1-5 arasında numaralandırılmıştır.
- Diğer sütunlarda ilk başta değerler yoktur, bu nedenle np.nan ile doldurulmuştur.

##### 2. Ekstra Satır Eklenmesi

```
19 extra_row = ["TABLO 2/ORAN", "YÜZDELİK ÖDEV1", "YÜZDELİK ÖDEV2", "YÜZDELİK QUIZ", "YÜZDELİK VİZE", "YÜZDELİK FİNAL"]
20
21
22 df_with_extra_row = pd.DataFrame(data=[extra_row], columns=df.columns).append(df, ignore_index=True)
```

- "TABLO 2/ORAN" başlığı ve yüzde ifadelerini içeren bir satır eklenmiştir.



- **.append Metodu:** Bu ekstra satır, mevcut tabloya eklenir.

### 3. Excel'e Kaydetme

```
25 df_with_extra_row.to_excel( excel_writer: "tablo2.xlsx", index=False)
```

- Tablo, "tablo2.xlsx" adıyla kaydedilir.

### 3. tablo1Create Fonksiyonu

Bu fonksiyon **Tablo 1**'i oluşturur ve "tablo1.xlsx" adlı bir Excel dosyasına kaydeder.

#### Veri Sözlüğü (data)

```
31 data = {
32     "Prg Çıktı": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
33     1.0: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],
34     2.0: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],
35     3.0: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],
36     4.0: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],
37     5.0: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],
38     "İlişki Değ.": [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
39 }
40
41
42
43 }
```

- **Sütunlar:** "Prg Çıktı", 1.0, 2.0, ..., "İlişki Değ."
- "Prg Çıktı" sütunu 1-10 arasında numaralandırılmıştır.
- Diğer sütunlar başlangıçta np.nan ile doldurulmuştur.

### Excel'e Kaydetme

```
47 df.to_excel( excel_writer: "tablo1.xlsx", index=False)
```

- Tablo, "tablo1.xlsx" adıyla kaydedilir.

### TABLO3-TABLO4-TABLO5

#### 1. Kütüphane İçer Aktarımı

```
1  import pandas as pd
2
3  import numpy as np
```

- **Pandas (pd):** Veri manipülasyonu için kullanılır. Excel dosyalarını okuyup yazmak ve tabloları düzenlemek için kullanılmıştır.
- **NumPy (np):** Sayısal işlemler için kullanılır. Burada eksik hücreleri np.nan ile işaretlemek için kullanılmıştır.

## 2. Veri Yükleme

```
5     dataFrameNot = pd.read_excel("OgrenciNotlari.xlsx")
6     dataFrameNot.set_index(keys: "Ogrenci_No", drop=True, inplace=True)
7
8
9     dataFrameTablo2 = pd.read_excel("tablo2.xlsx")
10    dataFrameTablo1 = pd.read_excel("tablo1.xlsx")
11
12    dataFrameTablo1.index = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- **OgrenciNotlari.xlsx**: Öğrencilerin notlarının bulunduğu Excel dosyası. Öğrenci numarası, tablo için **index** olarak ayarlanmıştır.
- **tablo2.xlsx ve tablo1.xlsx**: Daha önce oluşturulmuş iki tablo dosyası yüklenir.
- **Tablo 1**: Program çıktıları 1-10 arasında olacak şekilde yeniden numaralandırılmıştır.

## 3. weightdict Hesaplama

```
16    weightlist = dataFrameTablo2.columns.to_list()
17
18
19    weightdict = dict()
20    newtablo2value = dataFrameTablo2.loc[0].to_list()
21
22    sayac = 2
23
24    for column in range(len(weightlist)):
25        weightdict[weightlist[column]] = newtablo2value[column]
26
27
28    dataFrameTablo2.drop(labels: 0, axis=0, inplace=True)
```

- **weightdict**: Her sütun için ağırlıkları (yüzdeleri) saklar. Bu bilgiler tablo2.xlsx dosyasından alınmıştır.
- İlk satırdaki yüzdeler weightdict içine yüklenir ve bu satır daha sonra tablodan silinir.

## 4. tablo3\_Create Fonksiyonu

Bu fonksiyon, **Tablo 3**'ü oluşturur ve Excel dosyasına kaydeder.

### 1. Orijinal Tabloyu Kopyalama

```
36    tablo3dataframe = originalFrame.copy()
```

Orijinal tablodan (dataFrameTablo2) bir kopya alınır.

## 2. Yüzdelik Ağırlıkların Uygulanması

```
43     for column in columns:
44         coefficient = int(weightdict[column])
45         tablo3dataframe[column] = tablo3dataframe[column].apply(lambda x: x * (coefficient / 100))
```

Her sütundaki değerler, ağırlık yüzdesine göre çarpılır.

## 3. TOPLAM Sütunu

```
48     tablo3dataframe["TOPLAM"] = tablo3dataframe[columns].sum(axis=1)
```

Her satırdaki tüm notların toplamı yeni bir sütunda gösterilir.

## 4. Kaydetme

```
51     tablo3dataframe.to_excel("tablo3.xlsx", index=False)
```

Tablonun Excel dosyasına yazılması.

## 5. Öğrenci\_tablo4\_Create Fonksiyonu

Bu fonksiyon, bir öğrenciye özel **Tablo 4** oluşturur.

### 1. Tablo 3'ün Güncellenmesi

```
64     tablo3_Create(dataFrameTablo2)
65     tablo3dataframeNew = tablo3_Create(dataFrameTablo2)
66     tablo3dataframeNew.drop("Ders Çıktı", axis=1, inplace=True)
```

### 2. Öğrenci Verilerinin İşlenmesi

```
71     StudentDatatablo4 = dataframeNot.loc[StudentNo]
72
73     StudentDataFrame = pd.DataFrame(np.ones((5, 5)), columns=["Ödev1", "Ödev2", "Quiz", "Vize", "Final"], index=[1, 2, 3, 4, 5])
```

### 3. Yüzdelik Not Hesaplama

```
84     for row in indexes:
85
86
87         for column in columns:
88
89             StudentDataFrame.loc[row, column] = tablo3dataframeNew.loc[row, column] * StudentDatatablo4[column]
```

### 4. Başarı Oranı

```
93     StudentDataFrame["% Başarı"] = (StudentDataFrame["TOPLAM"] / StudentDataFrame["MAX"]) * 100
94     global sayac
95     StudentDataFrame.to_excel("tablo4/tablo4_" + str(StudentNo) + ".xlsx", index=False)
```

## 6. Öğrenci\_tablo5\_Create Fonksiyonu

Bu fonksiyon, bir öğrenciye özel **Tablo 5** oluşturur.

### 1. Başarı Oranlarının Kullanılması

```
103     tablo5columnslst = tablo4frame["% Başarı"].to_list()
104
105     tablo5columnslst = [round(x * 10) / 10 for x in tablo5columnslst]
106
107     columns_to_rename = {1.0: tablo5columnslst[0], 2.0: tablo5columnslst[1], 3.0: tablo5columnslst[2], 4.0: tablo5columnslst[3],
```

### 2. Tablo 5'in Doldurulması

```
112     for index in DataFrameTablo5["Prg Çıktı"].to_list():
113
114         for column in DataFrameTablo5.columns.to_list()[1:-1]:
115
116             DataFrameTablo5.loc[index, column] = columns_to_rename[column] * dataFrameTablo1.loc[index, column]
```

### 3. Sonuçların Kaydedilmesi

```
145     DataFrameTablo5.to_excel("tablo5/tablo5_" + str(StudentNo) + ".xlsx", index=False)
```

## 7. Tüm Öğrenciler İçin Çalıştırma

```
157     for index in dataFrameNot.index.to_list():
158
159         row = dataFrameNot.loc[index]
160         has_null = bool(row.isnull().any())
161
162
163         if not has_null:
164             WhichStudent(index)
```

- Öğrenci listesi döngüye alınır.
- Her öğrencinin verileri işlenir ve Tablo 4 ile Tablo 5 oluşturulur.

## 4.2 Görev dağılımı

Bu projede, ekip üyeleri görevleri eşit şekilde paylaşmıştır.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

Tablo 5'te sütun adlarında tekrar eden değerler oluştu. Bu durum, sütun adlarını ayırmayı ve anlamlı bir şekilde değiştirmeyi zorlaştırdı. Özellikle aynı yüzdelik değerlere sahip sütunların birden fazla kez kullanılması, sütun isimlerini benzersiz hale getirme gerekliliğini doğurdu.

Bu zorluğu aşmak için sütun adlarına bir ek tanımlayıcı ("-a", "-b", "-c") eklendi. Böylece sütun adları birbirinden ayrılabilir hale geldi.

```
DataFrameTablo5.columns=['Prg Çıktı', "{}-a".format(newcolumns[0]), "{}-b".format(newcolumns[1]), "{}-c".format(newcolumns[2]), "{}-d".format(newcolumns[3]), "{}-e".format(newcolumns[4]), 'Başarı Oranı']
newcolumns1=DataFrameTablo5.columns.to_list()
```

## 4.4 Proje isterlerine göre eksik yönler

- Bonus özellik gerçekleştirilmedi.

# 5 TEST VE DOĞRULAMA

## 5.1 Yazılımın test süreci

Projenin test süreci, sağlanan bir girdi tablosu ve bu girdilere dayalı olarak önceden hazırlanmış bir referans sonuç tablosunun kullanılmasıyla yürütüldü. Girdi tablosun içerdiği veri kümesi yazılımın başlangıç girdisi olarak kullanıldı. Test süreci boyunca kodun algoritmaları çalıştırıldı ve sırasıyla Tablo 1, Tablo 2 ve Tablo 3 gibi ara çıktılar oluşturuldu. Nihayetinde Tablo 4 ve Tablo 5 gibi sonuç tabloları elde edildi.


Test sırasında, oluşturulan sonuç tabloları sağlanan referans sonuç tablosuyla karşılaştırılarak kodun doğruluğu değerlendirildi. İlk çalıştırmalarda, Tablo 5'in sütun adlandırmalarında aynı yüzdelik değerlere sahip sütunların tekrarı gibi sorunlarla karşılaşıldı. Bu problem, sütun adlarına ek tanımlayıcı eklenerek çözüldü. Test sürecinde bu tür düzenlemeler yapıldıktan sonra yazılımın, beklenen çıktıları ürettiği doğrulandı.

## 5.2 Yazılımın doğrulanması

Yazılımın doğrulanması, test sürecinde elde edilen çıktılar ile sağlanan referans sonuç tablosunun karşılaştırılması üzerine odaklandı. Kodun her bir adımı sırasında oluşturulan tablolar, referans sonuç tablosuyla karşılaştırıldı. Örneğin, İlk testlerde fark edilen sütun isimlendirme problemleri gibi durumlar düzeltilerek tekrar testler yapıldı.

Sonuç olarak, yazılım, sağlanan girdilerle tutarlı şekilde çalışarak doğru sonuçlar üretti. Sağlanan referans sonuç tablosu, test sürecinin başarısını ölçmek için kullanıldı.

Tablo 5 için kullanılan çözüm:

tablo5 >  tablo5_208796934.xlsx							
	A	B	C	D	E	F	G
1	Prg Çıktı	76.5-a	61.1-b	62.1-c	62.6-d	63.7-e	Başarı Oranı
2	1	76.50	61.10	0	62.60	63.70	65.98
3	2	0	0	62.10	12.52	0	62.18
4	3	0	0	31.05	62.60	0	62.43
5	4	0	0	0	50.08	63.70	63.21
6	5	0	61.10	0	62.60	0	61.85
7	6	76.50	0	0	0	63.70	70.10
8	7	0	0	0	62.60	63.70	63.15
9	8	0	61.10	62.10	62.60	63.70	62.37
10	9	76.50	0	62.10	62.60	0	67.07
11	10	0	61.10	62.10	62.60	0	61.93

## Github Bağlantıları:

<https://github.com/220502005> AYBÜKE TÜRİDİ

<https://github.com/beyzabeyaz0> ELİF BEYZA BEYAZ

<http://github.com/serhatarslaner> SERHAT ARSLANER

<https://github.com/dilaseray> DİLA SERAY TEGÜN

<https://github.com/bozokhalat> AHMETCAN BOSTANCI