

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:**

**PROGRAM ÇIKTILARI İLİŞKİ MATRİSİ**

**HAZIRLAYAN:**

**İBRAHİM BENER KARACA-220501019**

**https://github.com/BenerKaraca**

**ENES KÜÇÜK-220501017**

**https://github.com/EnessKucukk**

**AMİNE DERİN-220501004**

**https://github.com/aminederin**

**NAZLI SU KETÇİ-220501007**

**https://github.com/nzlktc02**

**DERS SORUMLUSU:**

**DR. ÖĞR. ÜYESİ ELİF PINAR HACIBEYOĞLU**

**TARİH: 21.12.2024**

1. **GİRİŞ**
   1. **Projenin amacı**

Bu proje, öğrencilerin ders çıktılarının, program çıktılarının ve değerlendirme kriterlerinin etkileşimli bir şekilde ilişkilendirilmesini sağlayarak, her öğrencinin başarısını daha objektif ve hesaplanabilir bir şekilde değerlendirmeyi amaçlamaktadır. Ders çıktılarının ve program çıktılarının başarı oranlarının hesaplanması, öğrencilerin öğrenme sürecini izlemek ve hangi alanlarda güçlü olduklarını, hangi alanlarda gelişim göstermeleri gerektiğini ortaya koymak için önemlidir. Proje, aynı zamanda her öğrenciye özel başarı oranları çıkararak, eğitimdeki kişiselleştirilmiş değerlendirme sistemini desteklemektedir.

**Projede Gerçekleştirilmesi Beklenenler:**

1. **Ders Çıktıları ve Program Çıktıları İlişki Matrisi Oluşturulması:**
   * Ders çıktıları ve program çıktıları arasında belirli bir ilişki kurarak, her bir ders çıktısının hangi program çıktılarıyla ne kadar ilintili olduğunu gösteren bir ilişki matrisi oluşturulacaktır.
   * Bu matriste, her bir program çıktısının dersin çıktıları ile ne kadar ilişkili olduğu 0 ile 1 arasında değerlendirilecektir. Bu, her ders çıktısının başarı oranına nasıl katkı sağladığını görmemizi sağlar.
2. **Öğrenci Notları Tablosu:**
   * Öğrenci başarılarını değerlendirmek için, öğrencilerin her bir değerlendirme kriterinden aldıkları notlar (örneğin; ödev, vize, final) 0-100 aralığında belirlenerek bir öğrenci notları tablosu oluşturulacaktır.
   * Bu tablo, öğrencinin her bir değerlendirme kriterindeki başarısını gösterir ve ilerleyen aşamalarda bu veriler kullanılarak başarı oranları hesaplanacaktır.
3. **Ders Çıktıları ve Değerlendirme Kriterleri İlişki Matrisi:**
   * Ders çıktılarının değerlendirme kriterleriyle (örneğin; ödev, vize, final gibi) ne kadar ilişkili olduğunu gösteren bir ilişki matrisi oluşturulacaktır.
   * Bu matriste, her ders çıktısının hangi değerlendirme kriterleriyle bağlantılı olduğu ve bu bağlantıların ağırlıkları (değerlendirme kriterlerinin yüzdelik dilimi) belirlenerek hesaplamalar yapılacaktır.
4. **Ağırlıklı Değerlendirme Tablosu:**
   * Değerlendirme kriterlerinin ağırlıkları, ders çıktılar / değerlendirme kriterleri ilişki matrisindeki değerlerle çarpılarak ağırlıklı değerlendirme tablosu oluşturulacaktır.
   * Bu tablo, her bir ders çıktısının başarı oranına olan katkıyı, ilgili değerlendirme kriterlerinin ağırlıkları ile ilişkilendirerek netleştirir.
5. **Başarı Oranı Hesaplamaları:**
   * Öğrencilerin her bir ders çıktısındaki başarı oranları, öğrencinin aldığı notlar ile ağırlıklı değerlendirme tablosundaki hesaplamalarla çarpılarak bulunacaktır.
   * Başarı oranları, öğrencinin belirli bir dersin çıktısı ile ilgili ne kadar başarılı olduğunu ifade eder. Aynı zamanda, toplam başarı oranı ile karşılaştırma yapılarak, öğrencilerin hangi ders çıktılarına ne kadar katkı sağladığı analiz edilecektir.
6. **Program Çıktıları Başarı Oranı Hesaplaması:**
   * Program çıktıları başarı oranı, ders çıktılarındaki başarı oranlarının öğrencinin genel program çıktılarındaki başarıya etkisini hesaplayarak elde edilecektir.
   * Her bir öğrenci için, program çıktıları başarı oranları hesaplanacak ve bu oranlar ile öğrencinin başarısı arasındaki ilişki değerlendirilecektir.
7. **Github ve Rapor Paylaşımı:**
   * Projede oluşturulan tüm dosyalar (Excel tabloları, kodlar, raporlar) öğrencilerin GitHub hesaplarına yüklenecek ve projeye ait Github bağlantıları raporda belirtilecektir.
   * Projeye dair tüm hesaplamalar, yapılan işlemler, kullanılan metotlar ve elde edilen sonuçlar rapor halinde sunulacaktır.
8. **Proje Sunumu:**

* Proje tamamlandığında, grup sunumu yapılacaktır. Bu sunumda, projedeki tüm hesaplamalar, kullanılan yöntemler, elde edilen sonuçlar ve yapılan analizler detaylı bir şekilde aktarılacaktır.

1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

**Kullanıcı Arayüzü Gereksinimleri:**

1. **Ekran Görüntüsü:**
   * Kullanıcıların Tablo1, Tablo2, ve NotYukleöğelerinin seçilebilmesi için bir dosya arayüzü bulunmalıdır.
   * Seçilen dosyaların geçerli olup olmadığı doğrulayan bir kontrol seçeneklerine bağlıdır.
2. **İşlem Durumu Göstergesi:**
   * Dosyaların okunması, işlenme ve sonuçların oluşturulma durumlarının sağlanması bir ilerlemenin aktarılması veya mesaj alanı önemlidir.
3. **Sonuçların Görüntülenmesi:**
   * Öğrenciye ait Tablo4ve Tablo5bunların içeriklerini listelemek için bir ön izleme özelliği sunulmalıdır.
   * Toplam başarı oranları gibi özet bilgi tablolarıyla birlikte görüntülenmesi gereklidir.
4. **Sonuçların İndirilmesi:**
   * Oluşturulan Excel hücrelerinin (ör. Tablo4, Tablo5, Tablo3) kullanıcı tarafından indirilebilmesi için bir indirme butonu bulunmalıdır.
5. **Hata Bildirimi:**
   * Kullanıcının herhangi bir hata ile karşılaşması durumunda, hata mesajı kullanıcının net ve anlaşılır bir şekilde gösterilmelidir.
6. **Tema ve Görsellik:**
   * Basit ve kullanıcı dostu bir program tasarımı tercih edilmelidir.
   * Açık ve koyu tema seçenekleri sunulabilir.

**Donanım Arayüzü Gereksinimleri:**

1. **İşlemci ve Bellek Kullanımı:**
   * Kodun veri işleme sırasında sistem kaynaklarının verimli kullanılması gereklidir. Büyük tablolar işlenirken sistemin çökmesini önleyecek görünümler yapılmalıdır.
2. **Depolama Gereksinimleri:**
   * İşlem sonucunda Excel'in saklanacağı geçici bir depolama alanı gereklidir.
   * Minimum 100 MB disk alanı yeterli olacaktır.
3. **Çalışma Platformu:**
   * Yazılım, Windows, macOS ve Linux gibi platformlarda çalışmalıdır.
   * Python'un çalıştırılabilen bir ortam (ör. Anaconda, PyCharm) gereklidir.
4. **Harici Giriş/Çıkış Cihazları:**
   * Kullanıcıların dosya yüklemesi için bir klavye ve fareye erişimi olmalıdır.
   * Sonuçların görüntülenmesi ve indirilmesi için ekran kaybı minimum 1024x768 olmalıdır.

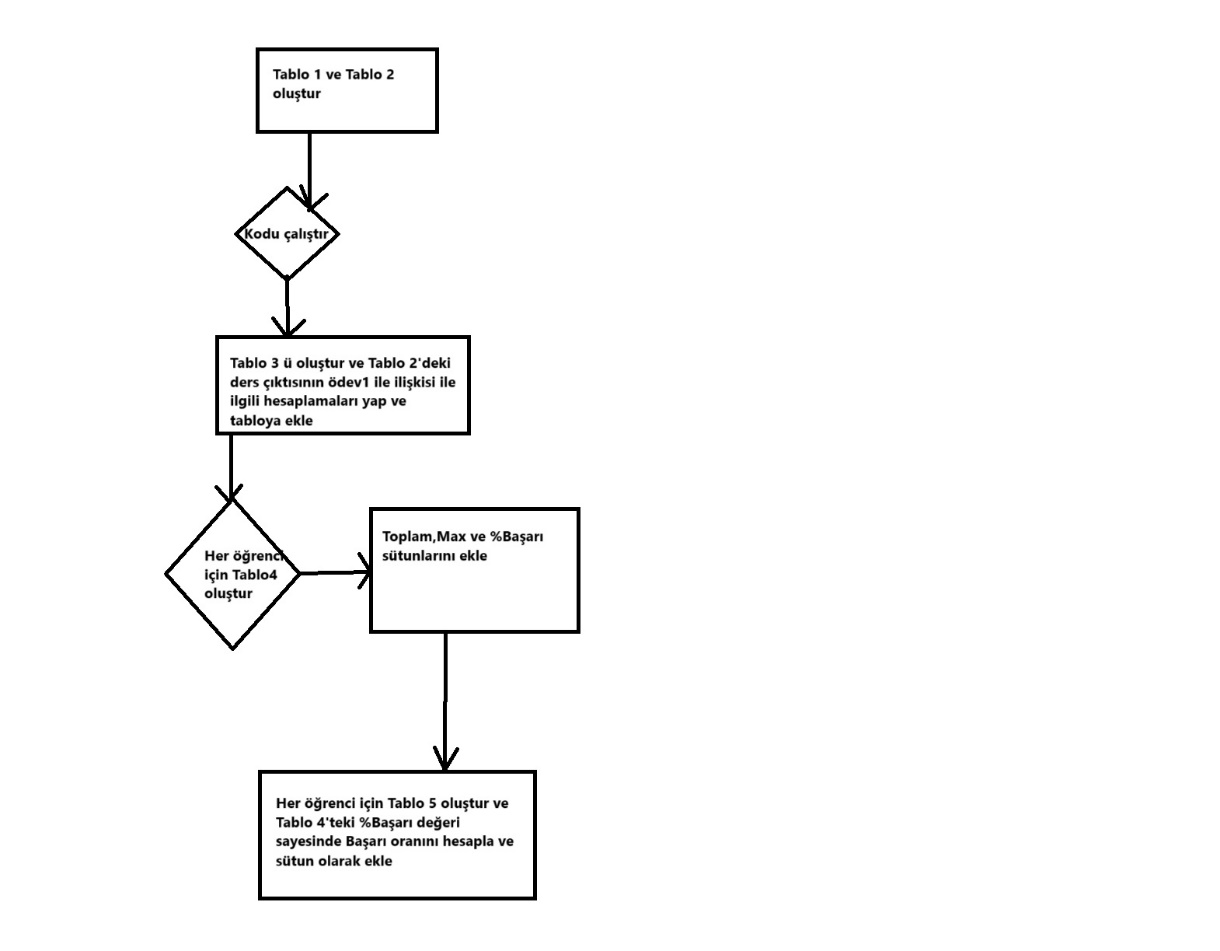
40

* 1. **Fonksiyonel gereksinimler**

1. **Sistem Bileşenleri**
   * **Giyilebilir Cihaz:**  
     Kullanıcının bileği, göğsü veya vücudunun uygun bir şekilde alınabilecek ergonomik ve hafif bir kapasitesine sahiptir. Kalp atış hızı, oksijen seviyesi, vücut özellikleri ve hareket durumu gibi sağlık sistemleri sürekli olarak toplar.
   * **Mobil Uygulama:**  
     Kullanıcı dostu bir programa sahip uygulama, cihazdan gelen verileri kullanıcının görsel, sesli veya metin tabanlı açıklamalarıyla sunar. Uygulama ayrıca, acil paketlerin saklandığı kişisel konum ve sağlık bölümlerine iletilir.
   * **Bulut Tabanlı Veri Depolama ve Analiz:**  
     Toplanan veriler güvenli bir şekilde bulut sunucularında saklanır ve analiz edilir. Yapay zeka programları, anormal durumlar algıları ve hızlı bir şekilde tepki verir.
2. **Entegrasyon**
   * **Kalp Atış Hızı Sensörü:**  
     Kullanıcının kalp atış ritmini sürekli izler ve anormallikler tespit edilir.
   * **Kan Oksijen Seviyesi Sensörü (SpO2):**  
     Kan oksijen seviyesini ölçerek hipoksemi gibi durumları erken tespit eder.
   * **Sıcaklık Sensörü:**  
     Kullanıcının vücut programını izleyerek Oliver veya hipotermi durumlarını belirler.
   * **Hareket ve Düşme Algılama Sensörü:**  
     Kullanıcının hareketlerini izler, düşme durumlarını tespit eder ve anında bildirim gönderir.
3. **Kablosuz İletişim**
   * **Bluetooth ve Wi-Fi:**  
     Giyilebilir cihaz, mobil uygulama ile Bluetooth üzerinden eşleşir ve buluta Wi-Fi ile veri aktarımı sağlar.
   * **GPS Modülü:**  
     Acil durumda elinizde tam olarak tespit edilir ve kontrol edilir.
4. **Yapay Zeka ve Veri Analizi**
   * **Makine Öğrenimi Algoritmaları:**  
     Toplanan biyometrik veriler analiz edilerek normalden sapmalar algılar. Anormal hareketleri hızla belirleyerek ilgili kişilerin bildirimlerini gönderir.
   * **Veri Görselleştirme:**  
     Mobil uygulama, sağlık açısından kendileriyle kullanıcılara sunar ve geçmiş sağlık sonuçlarını takip etmeyi sağlar.
5. **Enerjik ve Enerjik Olmayanlar**
   * **Şarj Edilebilir Batarya:**  
     Uzun süreli kullanım için optimize edilmiştir. Düşük enerji tüketimi sayesinde cihazın çalışma süresi artırılmıştır.
   * **Enerji Tasarruf Modu:**  
     Cihaz kullanıldığında minimum enerji tüketimi yapar.
6. **Kullanıcı Dostu**
   * **Ergonomi:**  
     Hafif ve taşınabilir tasarım, yaşlıların kullanabileceği şekilde kullanılabilir.
   * **Çoklu Geri Bildirim Seçenekleri:**  
     Kullanıcının dokunsal (titreşimli), sesli veya görsel bildirimler sunarak her türlü engelin değişmesine uyum sağlar.
7. **Güvenlik ve Veri Koruma**
   * **Uçtan Uca Şifreleme:**  
     Toplanan tüm veriler, kullanıcı gizliliğini korumak için şifrelenir.
   * **Yetkilendirme:**  
     Cihaza ve mobil uygulamaya yalnızca yetkili kişiler erişebilir.
8. **Saha Testleri ve Geri Bildirim Mekanizması**
   * Prototip, farklı yaş gruplarından ve sağlık durumlarından kullanıcılar üzerinde test edilir.
   * Kullanıcı geri bildirimlerine dayalı olarak cihazın performansı ve tasarımı optimize edilir.

40

* 1. **Use-Case diyagramı**

1. 
2. **TASARIM**
   1. **Mimari tasarım**

diyagram, metin, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

* 1. **Kullanılacak teknolojiler**

#### 1. ****Yazılımın Yazıldığı Dil:****

* **Python**: Proje, veri analizi, hesaplama ve otomasyon işlemleri için güçlü ve esnek bir dil olan Python kullanılarak yazılmıştır. Python, özellikle veri işleme ve matematiksel hesaplamalar için geniş bir kütüphane ekosistemine sahip olduğundan tercih edilmiştir.

#### 2. ****Kullanılan Harici Kütüphaneler:****

* **Pandas:**
  + Pandas, tablo benzeri verilerle çalışmak ve bu verileri analiz etmek için kullanılan bir Python kütüphanesidir. Bu projede Pandas, Excel dosyalarını okuma, işleme ve analiz etme işlemleri için kullanılmıştır.
  + Veri çerçeveleri (DataFrames) ile çalışılarak tablolar arasında hesaplamalar yapılmıştır.
* **OpenPyXL veya XlsxWriter (Opsiyonel):**
  + Excel dosyalarının yazılması ve düzenlenmesi için kullanılmıştır. Özellikle çıktıları Excel formatında kaydetmek için gereklidir.
  + Pandas'ın Excel yazma fonksiyonları aracılığıyla çalışır.
* **NumPy (Opsiyonel):**
  + Pandas ile birlikte matematiksel hesaplamalarda kullanılabilecek bir kütüphanedir. Büyük veri kümelerinin işlenmesinde hız avantajı sağlar.
* **OS (Opsiyonel):**
  + Dosya yollarını yönetmek veya kaydedilen dosyalarla ilgili işlemleri yapmak için kullanılabilir.

#### 3. ****Kullanılan Teknolojiler ve Ek Açıklamalar:****

* **Excel Formatı:**
  + Tabloların işlenmesi için Excel formatı (XLSX) kullanılmıştır. Öğrencilerin notlarının, ders çıktıları ilişki tablolarının ve başarı oranlarının hem okunması hem de yazılması bu formatta gerçekleştirilmiştir.
* **Yazılım Mimarisi:**
  + **Modüler Yapı:** Proje, farklı işlevlere sahip fonksiyonlar şeklinde modüler bir yapıda tasarlanmıştır. Bu yapı, yazılımın sürdürülebilirliğini ve yeniden kullanılabilirliğini artırır.
  + **Ana İşlev (Main Function):** Tüm süreci başlatan ve diğer fonksiyonları çağıran bir ana işlev (ana\_islev) bulunmaktadır. Bu işlev sayesinde kod, basit ve anlaşılır bir şekilde çalıştırılabilir.
* **Dosya Giriş/Çıkış:**
  + Proje, girdileri Excel dosyalarından okur ve işlenen verileri yine Excel formatında çıktı olarak kaydeder. Bu, kullanıcıların verilere kolayca erişebilmesini sağlar.
* **Harici Araçlar (Opsiyonel):**
  + Eğer gerekli olursa, bir **web crawler** veya otomatik veri toplama aracı projeye entegre edilebilir. Bu, öğrenci notlarını veya değerlendirme verilerini çevrimiçi bir sistemden çekmek için kullanılabilir.

#### 4. ****Çalışma Ortamı:****

* **Python Sürümü:**
  + Minimum Python 3.8 veya daha üst bir sürüm önerilir. Bu sürüm, Pandas ve diğer kütüphanelerin en son sürümleriyle uyumludur.
* **Çalıştırma Ortamı:**
  + Yazılım herhangi bir modern Python IDE (örneğin, PyCharm, VSCode, Jupyter Notebook) veya komut satırı (terminal) üzerinden çalıştırılabilir.
* **Bağımlılıkların Kurulumu:**
  + Gerekli kütüphanelerin kurulumu, Python'un **pip** paket yöneticisi kullanılarak yapılabilir:

#### 5. ****Diğer Önemli Notlar:****

* **Hata Yönetimi:**
  + Kodun birçok kısmında hata yönetimi (try-except blokları) kullanılarak programın kararlılığı artırılmıştır. Örneğin, dosya okunamadığında kullanıcıya bilgi verilmesi gibi özellikler eklenmiştir.
* **Esneklik:**
  + Ders çıktıları ve değerlendirme kriterlerinin ağırlıkları, ihtiyaçlara göre kolayca güncellenebilir. Oranlar bir Python sözlüğü (dictionary) içinde tutulduğu için yeni kriterler eklemek veya ağırlıkları değiştirmek oldukça kolaydır.

#### 6. ****Projenin Çalışma Süreci:****

1. Gerekli Excel dosyalarının hazırlanması (Tablo1, Tablo2, NotYukle).
2. Python scriptinin çalıştırılması.
3. İşlenen verilerin Excel dosyaları olarak çıktı alınması.
4. Elde edilen sonuçların yorumlanması ve raporlanması.
   1. **Veri tabanı tasarımı**
5. diyagram, metin, plan, taslak içeren bir resim

   Açıklama otomatik olarak oluşturuldu
   1. **Kullanıcı arayüzü tasarımı**

* Proje için herhangi bir arayüz kullanılmadı.

1. **UYGULAMA**
   1. **Kodlanan bileşenlerin açıklamaları**

* Bileşenlerin (fonksiyonlar, sınıflar ya da modüller) açıklanması
* Kullanılan Kütüphaneler
* pandas:
* Veri manipülasyonu ve analiz için kullanılan güçlü bir Python kütüphanesi. Bu kodda, Excel dosyalarını okumak, tablo işlemleri yapmak ve yeni Excel dosyaları oluşturmak için kullanılmıştır.
* os:
* Dosya ve dizin işlemleri yapmak için kullanılan Python’un yerleşik bir kütüphanesidir. Bu kodda, belirli dosyaların mevcut olup olmadığını kontrol etmek için kullanılmıştır.
* Fonksiyonlar ve Görevleri
* 1. tablo\_dosyasi\_oku(dosya\_adi)
* Görev: Verilen dosya adını kontrol eder, eğer dosya mevcutsa pandas kullanarak Excel dosyasını okur ve bir DataFrame olarak döndürür.
* İşleyiş:
* os.path.exists ile dosyanın var olup olmadığı kontrol edilir.
* pd.read\_excel ile Excel dosyası okunur.
* Eğer bir hata oluşursa, hata mesajı ile birlikte None döner.
* 2. program\_ders\_iliski\_matrisi\_olustur(tablo)
* Görev: Program çıktıları ve ders çıktıları arasındaki ilişkiyi ifade eden bir matrisi oluşturur.
* İşleyiş:
* Eğer tablo içinde İlişki Değ. adında bir sütun yoksa, her satırın geri kalan sütunlarının toplamını alarak yeni bir İlişki Değ. sütunu oluşturur.
* Bu sütun, her program çıktısının derslerle olan toplam ilişki derecesini ifade eder.
* 3. agirlikli\_degerlendirme\_tablosu\_olustur(tablo2, oranlar)
* Görev: Ders çıktıları ile değerlendirme kriterleri (ödev, quiz, vize, final) arasındaki ağırlıkları dikkate alarak ağırlıklı bir tablo oluşturur.
* İşleyiş:
* Tablo2’nin her sütununu, verilen oranlar sözlüğüne göre çarpar.
* Sonuçta her satırın toplam ağırlıklı puanını içeren bir TOPLAM sütunu ekler.
* Örneğin, bir dersin ağırlığı şu şekilde hesaplanır:
* python
* Kodu kopyala
* toplam = (Ödev1 \* 0.1) + (Ödev2 \* 0.1) + (Quiz \* 0.1) + (Vize \* 0.3) + (Final \* 0.4)
* 4. tablo4\_ve\_5\_olustur(tablo3, student\_grades, tablo1)
* Görev: Her öğrenci için Tablo 4 ve Tablo 5’i oluşturur ve Excel dosyası olarak kaydeder.
* İşleyiş:
* Tablo 4:
* Öğrencinin her değerlendirme kriteri (ödev, quiz, vb.) için başarı oranı hesaplanır.
* Başarı oranları, ağırlıklı değerlendirme tablosu (Tablo 3) ve öğrenci notlarının normalize edilmiş hali kullanılarak hesaplanır.
* Başarı oranları yüzdelik olarak yazılır.
* Tablo 5:
* Program çıktıları için başarı oranı hesaplanır.
* Tablo 1’deki ilişki değerleri ve Tablo 4’ten gelen başarı oranları dikkate alınır.
* Program çıktılarının başarı yüzdesi, ilgili ders çıktılarının başarı oranlarının ortalamasına dayanır.
* Tüm bu bilgiler ayrı ayrı Excel dosyalarına yazdırılır.
* 5. ana\_islev()
* Görev: Tüm işlemleri sırayla yürüten ana fonksiyondur.
* İşleyiş:
* Gerekli Excel dosyalarını (Tablo1, Tablo2, NotYukle) okur.
* Fonksiyonları sırayla çağırarak:
* Program çıktıları ve ders çıktıları ilişkisini oluşturur (Tablo 1).
* Ağırlıklı değerlendirme tablosunu oluşturur (Tablo 3).
* Öğrenci başarı oranlarını analiz ederek Tablo 4 ve Tablo 5’i oluşturur.
* Sonuçlar Excel dosyaları olarak kaydedilir.
* Kullanım Akışı
* Excel Dosyalarının Okunması:
* Tablo1.xlsx: Program çıktıları ile ders çıktıları arasındaki ilişkiyi içerir.
* Tablo2.xlsx: Ders çıktıları ve değerlendirme kriterleri arasındaki ilişkiyi içerir.
* NotYukle.xlsx: Öğrencilerin ders değerlendirmelerine ilişkin notlarını içerir.
* Tablo 1:
* Program çıktıları ve ders çıktıları ilişkisi analiz edilir ve ilişki değerleri hesaplanır.
* Tablo 3:
* Değerlendirme kriterlerinin ağırlıkları uygulanarak ağırlıklı bir tablo oluşturulur.
* Tablo 4 ve Tablo 5:
* Her öğrenci için ayrı ayrı:
* Tablo 4: Ders çıktıları başarı oranları hesaplanır.
* Tablo 5: Program çıktıları başarı oranları hesaplanır.
* Her bir öğrenci için ayrı Excel dosyaları olarak kaydedilir.
* Kodu Neden Böyle Yazdık?
* Modülerlik: Kod, birden fazla küçük ve bağımsız fonksiyon içeriyor. Bu, kodun bakımını kolaylaştırır ve tekrar kullanılabilir hale getirir.
* Hata Kontrolü: Dosyaların varlığı ve okunabilirliği kontrol ediliyor. Hata durumunda kullanıcıya bilgi veriliyor.
* Excel İşlemleri: pandas kullanılarak veri manipülasyonu hızlı ve etkili bir şekilde yapılıyor.
* Ağırlık Hesaplaması: Derslerin ve değerlendirme kriterlerinin önemini ifade etmek için ağırlıklar uygulanıyor.
* Otomasyon: Tüm işlemler otomatik olarak sıralı şekilde yapılıyor ve sonuçlar Excel dosyalarına kaydediliyor.
* import pandas as pd  
  import os  
    
  def tablo\_dosyasi\_oku(dosya\_adi):  
   *"""  
   Verilen Excel dosyasını okur ve tabloyu döndürür.  
   """* if os.path.exists(dosya\_adi):  
   try:  
   tablo = pd.read\_excel(dosya\_adi, index\_col=0)  
   print(f"{dosya\_adi} başarıyla okundu.")  
   return tablo  
   except Exception as e:  
   print(f"Hata: {dosya\_adi} okunamadı. ({e})")  
   return None  
   else:  
   print(f"Hata: {dosya\_adi} bulunamadı.")  
   return None  
    
  def program\_ders\_iliski\_matrisi\_olustur(tablo):  
   *"""  
   Program çıktıları ve ders çıktıları ilişki matrisi oluşturur.  
   """* if "İlişki Değ." not in tablo.columns:  
   tablo["İlişki Değ."] = tablo.iloc[:, :-1].sum(axis=1) / (tablo.shape[1] - 1)  
   return tablo  
    
  def agirlikli\_degerlendirme\_tablosu\_olustur(tablo2, oranlar):  
   *"""  
   Ağırlıklı değerlendirme tablosu oluşturur.  
   """* tablo3 = tablo2.mul(oranlar, axis=1)  
   tablo3["TOPLAM"] = tablo3.sum(axis=1)  
   return tablo3  
    
  def tablo4\_ve\_5\_olustur(tablo3, student\_grades, tablo1):  
   *"""  
   Her öğrenci için Tablo 4 ve Tablo 5'i oluşturur ve kaydeder.  
   """* for student, grades in student\_grades.iterrows():  
   # Tablo 4: Ders çıktıları başarı oranları hesaplama  
   tablo4 = tablo3.drop(columns="TOPLAM").mul(grades / 100, axis=1) # Notlar normalize edilir  
   toplam = tablo4.sum(axis=1) # Her ders çıktısı için toplam puan  
   max\_puan = tablo3["TOPLAM"] \* 100 # Maksimum puan ağırlıklarla hesaplanır  
   basari\_yuzdesi = (toplam / max\_puan) \* 100 # Yüzdelik başarı oranı  
    
   tablo4\_result = pd.DataFrame({  
   "Ödev1": (tablo4["Ödev1"] \* 100).round(0).astype(int),  
   "Ödev2": (tablo4["Ödev2"] \* 100).round(0).astype(int),  
   "Quiz": (tablo4["Quiz"] \* 100).round(0).astype(int),  
   "Vize": (tablo4["Vize"] \* 100).round(0).astype(int),  
   "Final": (tablo4["Final"] \* 100).round(0).astype(int),  
   "TOPLAM": toplam.round(1),  
   "MAX": max\_puan.round(1),  
   "%Başarı": basari\_yuzdesi.round(1).astype(str) + "%"  
   })  
   tablo4\_result.to\_excel(f"Tablo4\_{student}.xlsx")  
   print(f"Tablo 4 başarıyla kaydedildi: Tablo4\_{student}.xlsx")  
    
   # Tablo 5: Program çıktıları başarı oranları hesaplama  
   program\_cikti\_oranlari = {}  
   for prg\_cikti in tablo1.index:  
   ilgili\_ders\_ciktilari = [col for col in tablo1.columns if col != "İlişki Değ." and tablo1.at[prg\_cikti, col] > 0]  
   ilgili\_basari\_oranlari = basari\_yuzdesi.loc[ilgili\_ders\_ciktilari]  
   iliski\_degeri = tablo1.at[prg\_cikti, "İlişki Değ."]  
    
   if iliski\_degeri > 0:  
   program\_cikti\_oranlari[prg\_cikti] = (ilgili\_basari\_oranlari.mean() / iliski\_degeri).round(1)  
   else:  
   program\_cikti\_oranlari[prg\_cikti] = 0  
    
   tablo5\_result = pd.DataFrame({  
   "Program Çıktıları": list(program\_cikti\_oranlari.keys()),  
   "Başarı Oranı (%)": [str(val) + "%" for val in program\_cikti\_oranlari.values()]  
   }).set\_index("Program Çıktıları")  
   tablo5\_result.to\_excel(f"Tablo5\_{student}.xlsx")  
   print(f"Tablo 5 başarıyla kaydedildi: Tablo5\_{student}.xlsx")  
    
  def ana\_islev():  
   *"""  
   Ana işleyiş fonksiyonu.  
   """* tablo1\_path = "Tablo1.xlsx"  
   tablo2\_path = "Tablo2.xlsx"  
   student\_grades\_path = "NotYukle.xlsx"  
    
   tablo1 = tablo\_dosyasi\_oku(tablo1\_path)  
   tablo2 = tablo\_dosyasi\_oku(tablo2\_path)  
   student\_grades = tablo\_dosyasi\_oku(student\_grades\_path)  
    
   if tablo1 is not None and tablo2 is not None and student\_grades is not None:  
   # Program çıktıları ve ders çıktıları ilişki matrisi  
   tablo1 = program\_ders\_iliski\_matrisi\_olustur(tablo1)  
   tablo1.to\_excel("Tablo1\_Program\_Ders\_Iliski.xlsx")  
   print("Tablo 1 başarıyla kaydedildi: Tablo1\_Program\_Ders\_Iliski.xlsx")  
    
   # Ders çıktıları ve değerlendirme kriterleri ilişki matrisi  
   oranlar = {"Ödev1": 0.1, "Ödev2": 0.1, "Quiz": 0.1, "Vize": 0.3, "Final": 0.4} # Ağırlıklar doğrulandı  
   tablo3 = agirlikli\_degerlendirme\_tablosu\_olustur(tablo2, oranlar)  
   tablo3.to\_excel("Tablo3\_Agirlikli\_Degerlendirme.xlsx")  
   print("Tablo 3 başarıyla kaydedildi: Tablo3\_Agirlikli\_Degerlendirme.xlsx")  
    
   # Tablo 4 ve 5 sonuçlarını kaydetme  
   tablo4\_ve\_5\_olustur(tablo3, student\_grades, tablo1)  
    
   print("Tüm işlemler başarıyla tamamlandı.")  
   else:  
   print("Tablo1, Tablo2 veya öğrenci notları okunamadığı için işlemler yapılamadı.")  
    
  if \_name\_ == "\_main\_":  
   ana\_islev()

#### 4.2 Görev dağılımı

#### **Proje Yönetimi ve Genel Koordinasyon:**

* + **Sorumlu:** [Enes Küçük]
  + Görevler: Tüm proje bileşenlerinin ilerleyişini takip etmek, gerekli dokümanların güncel tutulmasını sağlamak, teslim tarihlerini organize etmek ve ekibin koordinasyonunu sağlamak.
* **Program Çıktıları / Ders Çıktıları İlişki Matrisi (Tablo 1):**
  + **Sorumlu:** [Amine Derin]
  + Görevler: Program çıktıları ile ders çıktıları arasındaki ilişki matrisini oluşturmak ve oranları hesaplamak.
* **Ders Çıktıları / Değerlendirme Kriterleri İlişki Matrisi (Tablo 2):**
  + **Sorumlu:** [Bener Karaca]
  + Görevler: Değerlendirme kriterlerinin ders çıktılarıyla olan ilişki oranlarını belirlemek ve tabloyu oluşturmak.
* **Ağırlıklı Değerlendirme Tablosu (Tablo 3):**
  + **Sorumlu:** [Enes Küçük]
  + Görevler: Ağırlıklı oranların hesaplanması ve ders çıktıları ile değerlendirme kriterlerinin ilişkilendirilmesi.
* **Öğrenci Notlarına Göre Ders Çıktıları Başarı Oranları (Tablo 4):**
  + **Sorumlu:** [Nazlı Su Ketçi]
  + Görevler: Öğrenci notları üzerinden ders çıktılarının başarı oranlarının hesaplanması ve tabloya dökülmesi.
* **Öğrenci Notlarına Göre Program Çıktıları Başarı Oranları (Tablo 5):**
  + **Sorumlu:** [Amine Derin]
  + Görevler: Program çıktılarının öğrenci notları üzerinden başarı oranlarının analiz edilmesi ve tablonun oluşturulması.
* **Kodlama ve Test:**
  + **Sorumlu:** [Bener Karaca]
  + Görevler: Python kodlarının geliştirilmesi, hata ayıklama ve testlerin yapılması.

#### ****2. Raporun Hazırlanması Sürecindeki Görev Dağılımı****

* **Giriş ve Proje Amacı:**
  + **Sorumlu:** [Nazlı Su Ketçi]
  + Görevler: Projenin genel açıklamasını, amacı ve hedeflerini yazmak.
* **Yazılım ve Teknolojiler Hakkında Bilgi:**
  + **Sorumlu:** [Nazlı Su Ketçi]
  + Görevler: Yazılımda kullanılan dil, kütüphaneler ve diğer teknolojilerle ilgili açıklamaların hazırlanması.
* **Bileşenlerin Teknik Açıklamaları:**
  + **Sorumlu:** [Enes Küçük]
  + Görevler: Tabloların mantığını ve oluşturulma süreçlerini açıklamak.
* **Hesaplama Süreçlerinin Açıklanması:**
  + **Sorumlu:** [Amine Derin]
  + Görevler: Hesaplama yöntemlerinin detaylı açıklanması ve örneklerle gösterilmesi.
* **Görev Dağılımı ve Ekip Çalışması:**
  + **Sorumlu:** [Amine Derin]
  + Görevler: Projedeki görev paylaşımını detaylandıran bu bölümün yazılması.
* **Sonuç ve Değerlendirme:**
  + **Sorumlu:** [Bener Karaca]
  + Görevler: Projenin sonuçlarının değerlendirilmesi ve önerilerin yazılması.
* **Düzenleme ve Son Kontrol:**
  + **Sorumlu:** [Bener Karaca]
  + Görevler: Raporun düzenlenmesi, gramer ve yazım hatalarının kontrolü, teslim öncesi son düzenlemelerin yapılması.

#### ****3. Sunum Hazırlığı****

* **Sunum İçeriğinin Hazırlanması:**
  + **Sorumlu:** [Amine Derin]
  + Görevler: Raporun içeriğinden faydalanarak sunum slaytlarını hazırlamak.
* **Sunum Provası:**
  + **Sorumlu:** Tüm ekip üyeleri
  + Görevler: Her ekip üyesinin sunacağı bölümlerin prova edilmesi ve sunum sırasında oluşabilecek sorulara hazırlık yapılması.
  1. **Karşılaşılan zorluklar ve çözüm yöntemleri**

#### ****1. Geliştirme Sürecinde Karşılaşılan Problemler****

1. **Veri Uyumsuzluğu:**
   * **Açıklama:** Tablo1, Tablo2 ve öğrenci notları dosyaları arasında veri uyumsuzlukları yaşandı. Farklı formatlarda veya eksik veri içeren dosyalar nedeniyle analizlerde sorunlar ortaya çıktı.
   * **Çözüm:** Veriler Python'daki pandas kütüphanesi kullanılarak temizlendi ve eksik veriler için uygun varsayılan değerler (örneğin "0") belirlendi. Ayrıca dosyaların formatlarını standartlaştırmak için bir ön işlem aşaması eklendi.
2. **Karmaşık Hesaplama Mantığı:**
   * **Açıklama:** Ders çıktıları ile değerlendirme kriterlerinin ağırlıklarının birleştirilmesi ve başarı oranlarının hesaplanmasında karışıklıklar meydana geldi.
   * **Çözüm:** Kod yapısı daha modüler hale getirildi. Fonksiyonlar daha küçük alt görevler için ayrılarak hata ayıklaması kolaylaştırıldı. Ayrıca her fonksiyon için örnek veri setleriyle test yapılarak doğru sonuçlar elde edildi.
3. **Harici Kütüphanelerin Kullanımıyla İlgili Sorunlar:**
   * **Açıklama:** pandas ve openpyxl kütüphanelerinde sürüm uyumsuzluğu nedeniyle dosyaların okuma ve yazma işlemlerinde hatalar oluştu.
   * **Çözüm:** Kütüphanelerin güncel sürümleri yüklendi ve belgeleri incelenerek doğru yöntemler kullanıldı. Kütüphane sürümleri proje dokümanına not edildi.
4. **Ekip İçi Koordinasyon:**
   * **Açıklama:** Görev dağılımında zamanlama uyuşmazlıkları nedeniyle bazı görevler gecikti.
   * **Çözüm:** Haftalık durum toplantıları düzenlendi ve görevlerin takibi için bir proje yönetim aracı (örneğin, Trello veya Google Sheets) kullanıldı. Görev öncelikleri yeniden değerlendirilerek ekip üyeleri arasında iş bölümü optimize edildi.
5. **Sonuçların Doğruluğunun Kontrolü:**
   * **Açıklama:** Öğrenci başarı yüzdeleri ve program çıktıları başarı oranlarının hesaplanmasında sonuçların doğruluğu konusunda şüpheler oluştu.
   * **Çözüm:** Kodlama sırasında her aşamada ara çıktılar saklandı ve manuel kontrol yapıldı. Ayrıca, doğru sonuçlar elde edilmesini sağlamak için örnek veriler üzerinde kontrol testleri yapıldı.
6. **Raporlama ve Çıktıların Formatlanması:**
   * **Açıklama:** Hesaplama sonuçlarını Excel'e yazarken biçimlendirme sorunları yaşandı, özellikle karmaşık tabloların okunabilirliği düşük kaldı.
   * **Çözüm:** openpyxl kütüphanesiyle çıktı formatları düzenlendi ve başlıklar, tablo renkleri gibi iyileştirmeler yapıldı. Ayrıca çıktıların kolay okunabilmesi için gereksiz sütunlar kaldırıldı.

#### ****2. Çözüm Yöntemleri****

1. **Kod ve Verilerin Modüler Hale Getirilmesi:**
   * Geliştirilen fonksiyonlar daha basit ve bağımsız görevleri yerine getirecek şekilde düzenlendi.
   * Hata oluşabilecek yerlerde daha fazla hata yakalama mekanizması (try-except blokları) kullanıldı.
2. **Veri Kontrol Süreci:**
   * Tüm Excel dosyaları kodlama öncesi manuel olarak kontrol edildi.
   * Veri temizleme işlemi için otomatik bir modül geliştirildi.
3. **Dokümantasyon ve Test Süreci:**
   * Geliştirilen her fonksiyon detaylı bir şekilde belgelenerek ekip üyelerinin anlaması kolaylaştırıldı.
   * Örnek veri setleri oluşturularak test süreçleri otomatikleştirildi.
4. **Ekip İçi İletişim ve Koordinasyon:**
   * Görevlerin netleştirilmesi ve ilerlemenin takibi için proje toplantıları düzenlendi.
   * İletişim için birden fazla kanal (örneğin, WhatsApp ve e-posta) aktif kullanıldı.
5. **Hata Yönetimi ve Çözüm Önerileri:**
   * Karşılaşılan sorunlar detaylı şekilde kayıt altına alındı ve benzer durumlarda kullanılmak üzere çözüm önerileri oluşturuldu.
6. **Yedekleme ve Versiyon Kontrolü:**
   * Veriler ve kodlar düzenli olarak yedeklendi.
   * Versiyon kontrolü için GitHub gibi bir platform kullanılarak iş birliği kolaylaştırıldı
   1. **Proje isterlerine göre eksik yönler**

**%Başarı Hesaplamaları Hatalı:**

Tablo 4'te %Başarı değerleri bazen doğru normalize edilmiyor veya yuvarlama hatası var.

Kodda, MAX değeri doğru hesaplanmış olsa da, öğrenci notlarının toplamı (TOPLAM) ile karşılaştırması doğru yapılmamış olabilir.

Tablo 5'teki Başarı Oranı Yanlış Hesaplanabilir:

Başarı Oranı = Ortalama (Tablo 5 satır değerleri) / İlişki Değeri(Tablo 1) formülü uygulanmış ancak:

Tablo 1 ve Tablo 3 arasında veri eşleştirme sırasında sütunlar yanlış kullanılabilir.

İlişki Değ. sütununun eksik olduğu veya yanlış hesaplandığı durumlar oluşabilir.

Yuvarlama ve Formatlama Sorunları:

Tablo 4 ve Tablo 5'te yüzdelik değerler string formatında (95% gibi) isteniyor, ancak tam bu şekilde olmayabilir.

Özellikle 0.5 gibi değerlerin 50% yerine 5% gibi hatalı görünmesi olası.

Test Verisi ve Gerçek Verilerle Uyumsuzluk:

Kod genel olarak doğru bir mantık üzerine kurulu olsa da, kullanılan örnek verilerde olabilecek uyumsuzluklar (örneğin, Tablo 1'deki sütunların Tablo 3 ile eşleşmemesi) hatalara neden olabilir.

Hatalı Sütun ve Satır Erişimleri:

Kodda bazı sütunların veya satırların isimlerine dayanarak işlem yapılmakta. Eğer verilerde sütun isimleri farklıysa (İlişki Değ. yerine İlişki Degeri yazılmışsa gibi), hata alınabilir.

1. **TEST VE DOĞRULAMA**
   1. **Yazılımın test süreci**

**import pandas as pd**

**def test\_kodlari():**

**# Tablo 1: Program çıktıları ve ders çıktıları ilişkisi**

**tablo1\_data = {**

**"1": [1, 0, 0, 1, 1],**

**"2": [0, 1, 0, 0, 1],**

**"3": [1, 0, 1, 0, 0],**

**"4": [0, 1, 0, 1, 1],**

**"5": [1, 0, 1, 1, 0]**

**}**

**tablo1 = pd.DataFrame(tablo1\_data, index=["Prg1", "Prg2", "Prg3", "Prg4", "Prg5"])**

**# Tablo 2: Ders çıktıları ve değerlendirme kriterleri**

**tablo2\_data = {**

**"Ödev1": [1, 0.5, 0, 0.8, 0.3],**

**"Ödev2": [0, 1, 0.2, 0.6, 0],**

**"Quiz": [1, 0, 0.5, 0, 0.4],**

**"Vize": [0.5, 0, 0.6, 1, 0.4],**

**"Final": [0.8, 1, 0.7, 0.9, 0.5]**

**}**

**tablo2 = pd.DataFrame(tablo2\_data, index=["DC1", "DC2", "DC3", "DC4", "DC5"])**

**# Öğrenci notları**

**student\_grades\_data = {**

**"Ödev1": [90, 70, 85],**

**"Ödev2": [80, 60, 75],**

**"Quiz": [70, 90, 65],**

**"Vize": [85, 95, 80],**

**"Final": [95, 85, 100]**

**}**

**student\_grades = pd.DataFrame(student\_grades\_data, index=["Öğrenci A", "Öğrenci B", "Öğrenci C"])**

**# Test işlemleri**

**from yaz import program\_ders\_iliski\_matrisi\_olustur, agirlikli\_degerlendirme\_tablosu\_olustur, tablo4\_ve\_5\_olustur**

**print("### Test Başlıyor ###")**

**# Tablo 1 işleme**

**tablo1 = program\_ders\_iliski\_matrisi\_olustur(tablo1)**

**print("Tablo 1 İşlendi:\n", tablo1)**

**# Tablo 3 oluşturma**

**oranlar = {"Ödev1": 0.1, "Ödev2": 0.1, "Quiz": 0.1, "Vize": 0.3, "Final": 0.4}**

**tablo3 = agirlikli\_degerlendirme\_tablosu\_olustur(tablo2, oranlar)**

**print("Tablo 3 İşlendi:\n", tablo3)**

**# Tablo 4 ve 5 oluşturma**

**tablo4\_ve\_5\_olustur(tablo3, student\_grades, tablo1)**

**print("### Test Tamamlandı ###")**

**# Test kodunu çalıştır**

**test\_kodlari()**

* 1. **Yazılımın doğrulanması**

KAYNAKÇA:

 **Pandas Kütüphanesi:**

* Veri işleme ve analizinde kullanılan pandas kütüphanesi için kaynak:  
  McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media.  
  Resmi Web Sitesi: https://pandas.pydata.org/

 **OpenPyXL Kütüphanesi:**

* Excel dosyalarıyla çalışmak için kullanılan kütüphane:  
  OpenPyXL Documentation, <https://openpyxl.readthedocs.io/>, Erişim Tarihi: Aralık 2024.

 **Excel ile Veri Analizi:**

* Veri işleme süreçlerinde kullanılan Excel ile ilgili kaynak:  
  Walkenbach, J. (2015). *Excel Bible*. Wiley.

 **Proje Yönetim Metodolojileri:**

* Yazılım projelerinde kullanılan temel proje yönetim yöntemleri:  
  Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.

 **Excel Dosyalarının İşlenmesi:**

* Excel dosyalarının analizi ve Python ile entegrasyonu:  
  VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

 **Öğrenci Performansının Değerlendirilmesi:**

* Eğitim süreçlerinde öğrenci performansı ölçüm metrikleri üzerine kaynak:  
  Guskey, T. R. (2003). *Evaluating Professional Development*. Corwin Press.