Verisetimiz:

Pokemon.csv

Veri üzerinde bazı temizlik işlemleri yaptık, özellikle kategorik değerler için;

**Processes:**

Load and Inspect Data: Load the original dataset and inspect the first few rows and basic information.

Handle Missing Values: Fill any missing values.

Remove Non-Numeric Columns: Remove columns that are not numeric.

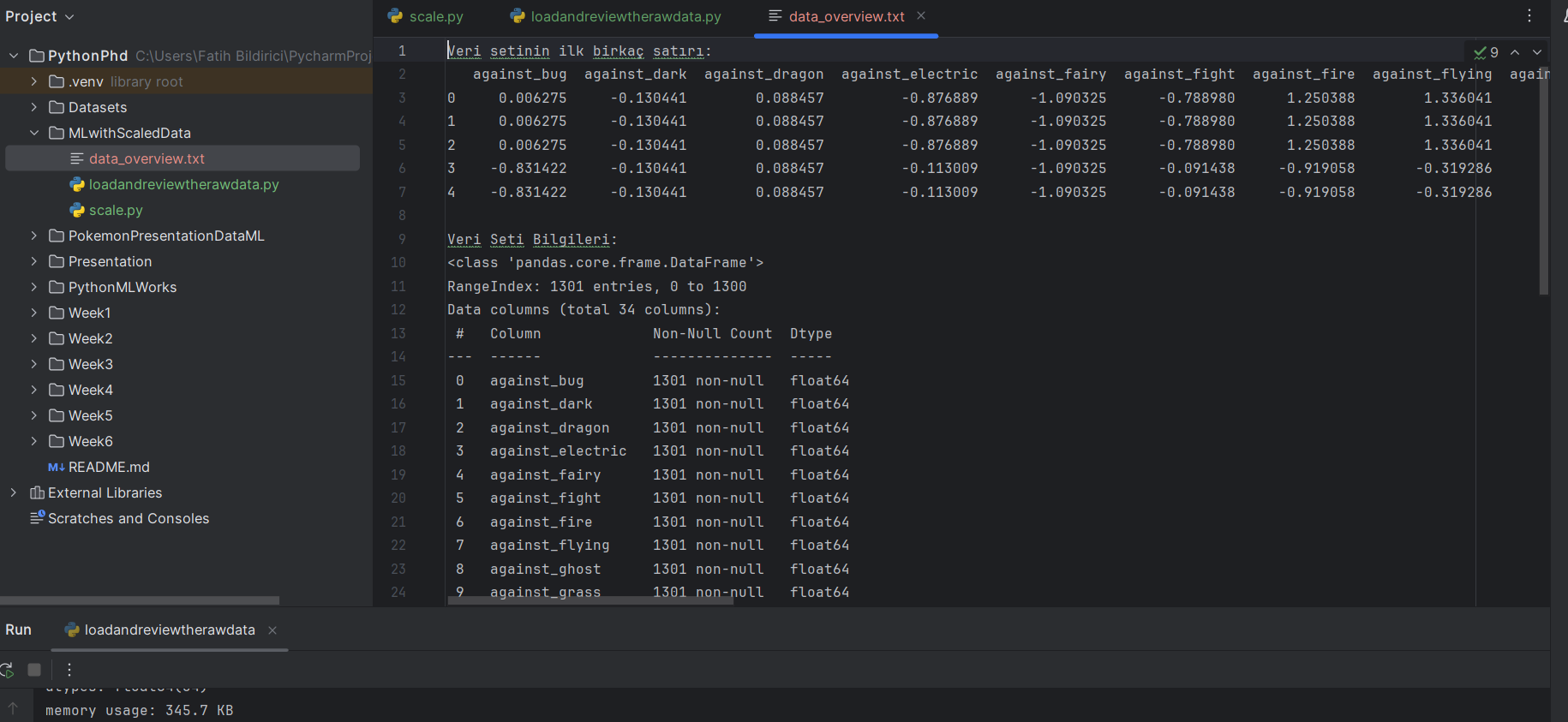
Scale Numeric Columns: Scale the numeric columns for better model performance.

Add Synthetic Data: Use make\_classification to generate synthetic data for better balance and variety. This synthetic data will help in comparative analysis.

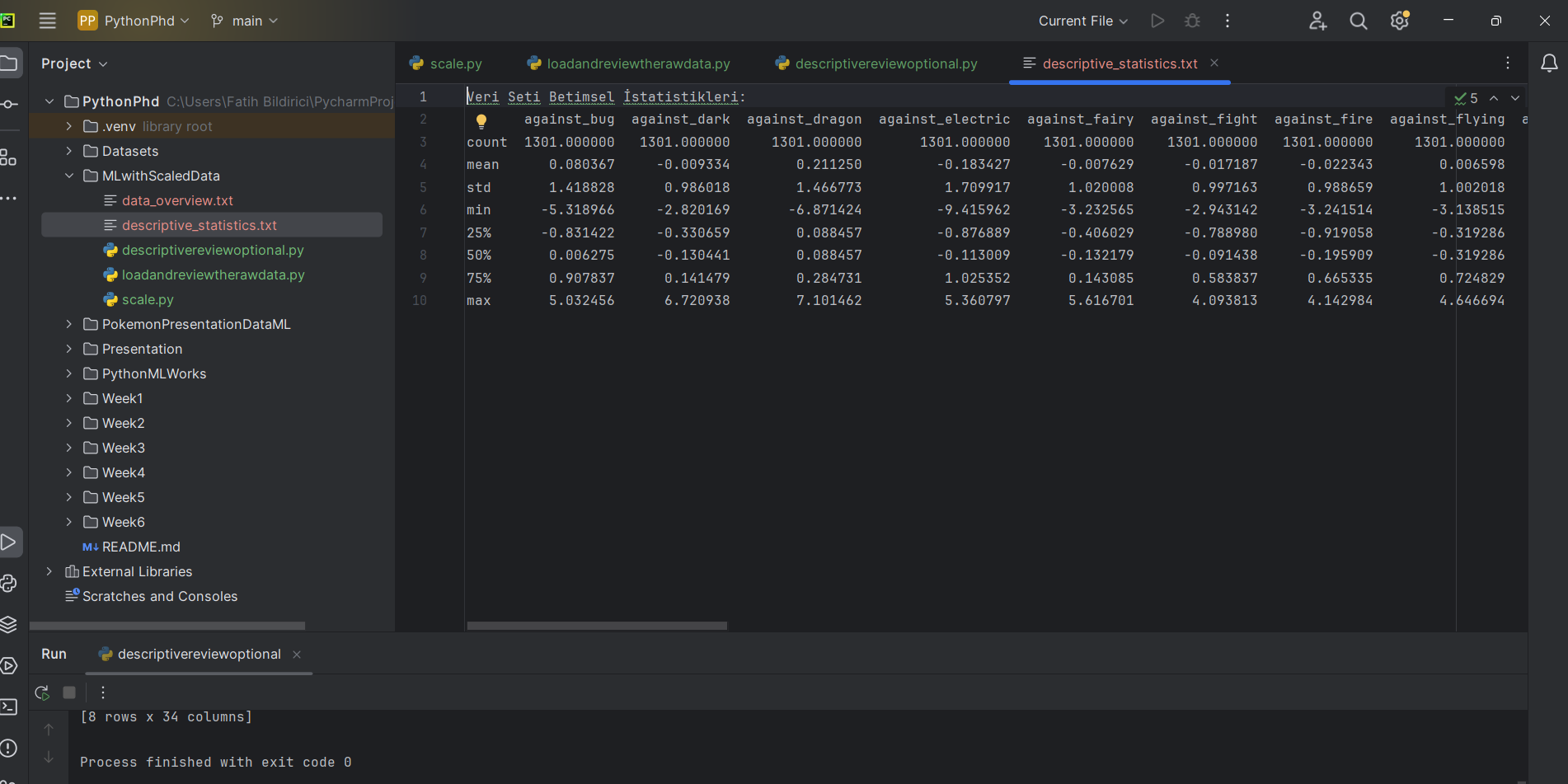
Combine Datasets: Combine the original and synthetic datasets.

Save Final Dataset: Save the combined dataset to a new CSV file.

1. Adım veriyi yükleme ve inceleme



1. Veri için bir descriptive statistics bakalım (optional)



Betimsel İstatistiklerin Açıklaması

count: Her bir sütundaki geçerli (NaN olmayan) değerlerin sayısı.

mean: Ortalama değer.

std: Standart sapma.

min: Minimum değer.

25%: İlk çeyrek değeri (Q1).

50% (medyan): Ortanca değer (Q2).

75%: Üçüncü çeyrek değeri (Q3).

max: Maksimum değer.

Kodu Açıklama

Veriyi Yükleme: CSV dosyasını pd.read\_csv kullanarak yüklüyoruz.

Betimsel İstatistikleri Hesaplama: data.describe(include='all') ile betimsel istatistikleri çıkarıyoruz. include='all' parametresi, hem sayısal hem de kategorik değişkenleri dahil eder.

Dosyaya Kaydetme: Betimsel istatistikleri descriptive\_statistics.txt dosyasına yazıyoruz.

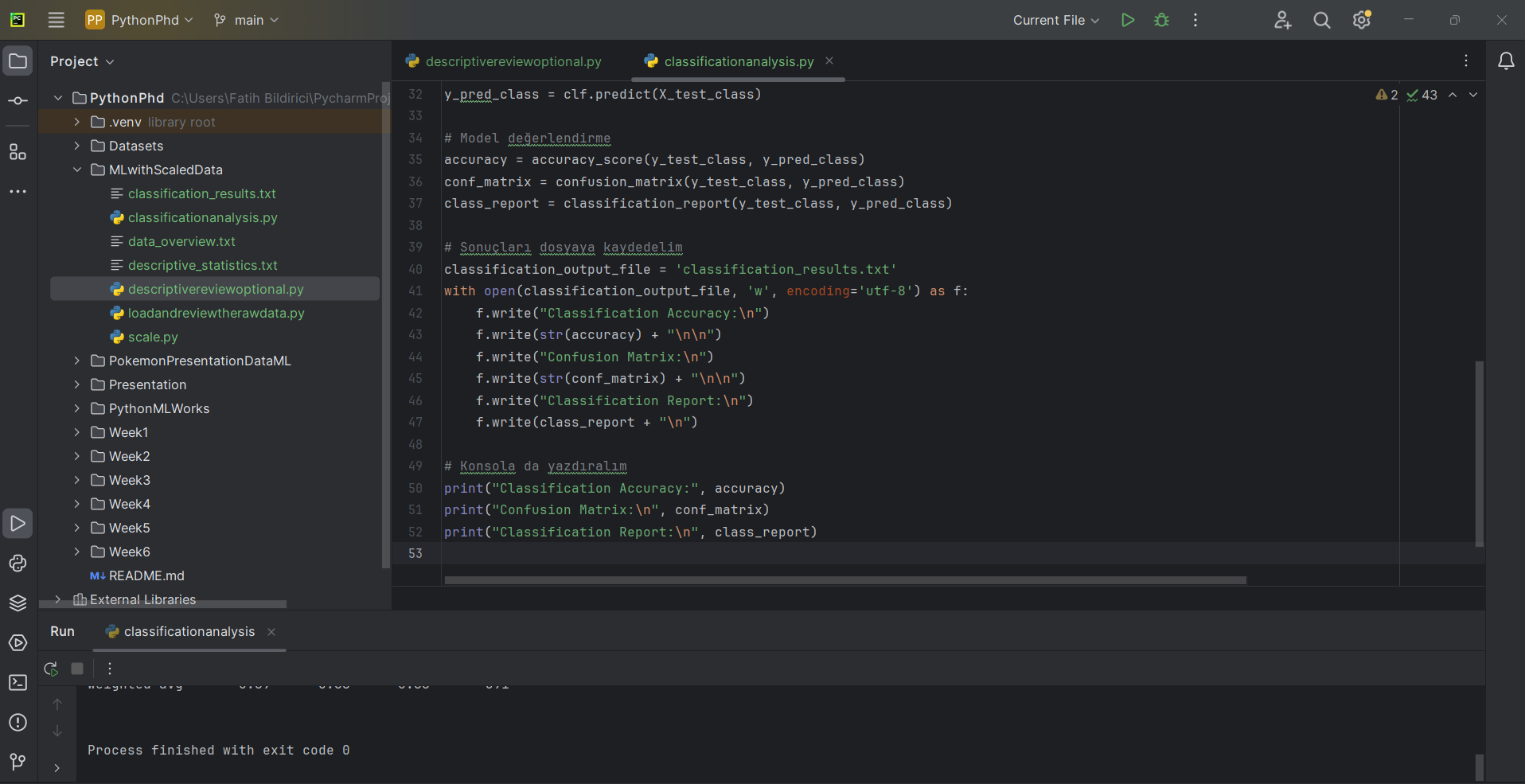
Konsola Yazdırma: Betimsel istatistikleri konsola da yazdırıyoruz.

Bu kodu çalıştırarak, veri setinizin betimsel istatistiklerini descriptive\_statistics.txt dosyasına kaydedebilir ve inceleyebilirsiniz.

1. Analizler

Classification

Is\_legendary seçildi, açıklama yazılacak integera çevirildi.



2. Regresyon analizi

Regresyon analizi genellikle sürekli (continuous) hedef değişkenlerle yapılır. is\_legendary değişkeni, kategorik bir değişkendir (0 veya 1 değerleri alır) ve bu tür bir değişken için sınıflandırma algoritmaları daha uygundur. attack gibi sürekli bir değişken, regresyon için daha doğru bir seçimdir.

Comparative Analysis için İki Farklı Algoritma ile Çalışmak

Ödevin gerekliliklerini yerine getirmek için iki farklı algoritma kullanarak comparative analysis yapmamız gerekmektedir. Bu durumda, sınıflandırma ve regresyon algoritmaları kullanarak is\_legendary ve attack değişkenleri üzerinde çalışabiliriz.

Sınıflandırma Analizi (is\_legendary için):

Hedef: is\_legendary

Algoritma: Decision Tree Classifier

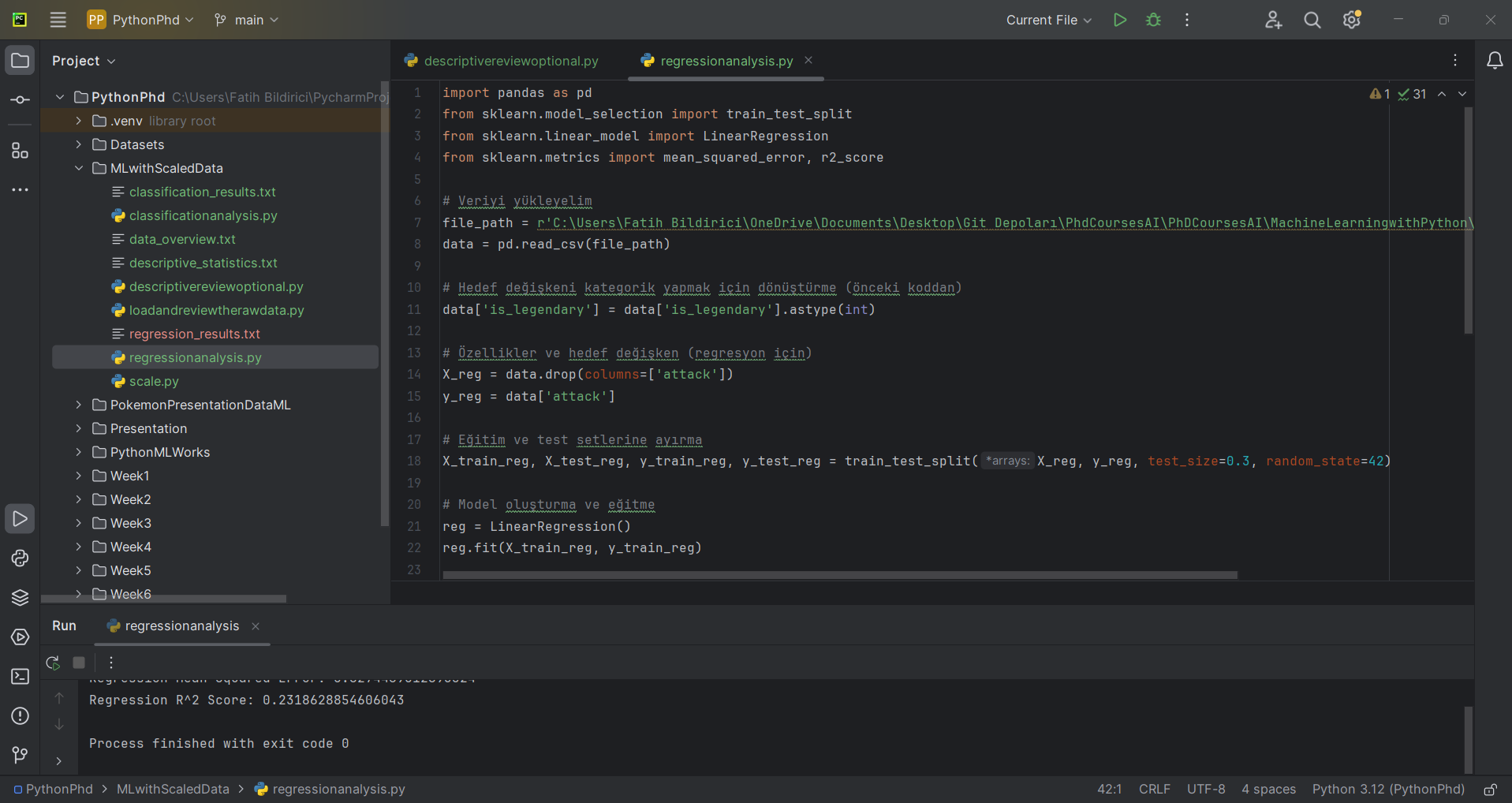
Regresyon Analizi (attack için):

Hedef: attack

Algoritma: Linear Regression

Bu iki analiz sonucunda elde edeceğimiz performans metriklerini karşılaştırarak veri setinin ve model performanslarının analizi yapılabilir. Şimdi, regresyon ve sınıflandırma analizlerinin sonuçlarını kaydederek bir comparative analysis yapalım.

Regresyon:



Comparative Analysis

Sınıflandırma Sonuçları: Dosyada classification\_results.txt olarak kaydedildi.

Accuracy

Confusion Matrix

Classification Report (Precision, Recall, F1-Score)

Regresyon Sonuçları: Dosyada regression\_results.txt olarak kaydedildi.

Mean Squared Error

R^2 Score

Bu iki modelin sonuçlarını karşılaştırarak veri setinin farklı yönlerini ve model performanslarını değerlendirebiliriz. Örneğin, is\_legendary değişkeni üzerinde sınıflandırma modeli kullanarak doğru şekilde tahmin yapma oranımızı (accuracy) ve hata oranımızı (confusion matrix) görebiliriz. Benzer şekilde, attack değişkeni üzerinde regresyon modeli kullanarak tahmin hatalarımızı (MSE) ve modelin açıklayıcılık oranını (R^2) değerlendirebiliriz.