

Öne Çıkan Maddeler

1. **Giriş ve Platform:** Eğitimin amacı, Python'ı yapay zeka odağında, pratik bir bakış açısıyla ele almaktır. Uygulamalar **Google Colab** üzerinde gerçekleştirilmiştir.
2. **Temel Kavramlar (Değişkenler ve Veri Tipleri):** Python'ın dinamik tip yapısı (dynamically typed) vurgulanmış; integer, float, string ve boolean gibi temel veri tipleri ile **f-string** formatlama yöntemi tanıtılmıştır.
3. **Operatörler:** Aritmetik (+, *, **), karşılaştırma (==, >), mantıksal (and, or) ve atama (=, +=) operatörleri gibi temel işlemciler ele alınmıştır.
4. **Veri Yapıları (Listeler, Sözlükler):** Yapay zeka projelerinde sıkça kullanılan liste (değiştirilebilir, sıralı) ve sözlük (dictionary, key-value yapısı) gibi veri yapıları detaylandırılmıştır. **List Comprehension** gibi pratik kullanımlar gösterilmiştir.
5. **Kontrol Akışı (if, for, while):** Kodun karar almasını sağlayan if-elif-else koşul blokları ve tekrar eden işlemler için kullanılan for ve while döngüleri, break ve continue gibi kontrol ifadeleriyle birlikte işlenmiştir.
6. **Fonksiyonlar (def, lambda):** Kod tekrarını önleyen ve modülerlik sağlayan def ile fonksiyon tanımlama, parametre ve argüman kavramları, **lambda** ile tek satırlık anonim fonksiyonlar anlatılmıştır.
7. **Hata Yönetimi (try-except):** Kod çalışırken oluşabilecek hataları yönetmek, programın çökmesini engellemek ve hata ayıklama (debugging) için try-except bloklarının kullanımı gösterilmiştir.
8. **Dosya İşlemleri (with open):** Metin dosyalarını okuma (r), yazma (w) ve ekleme (a) modları ile açma işlemleri, kaynak yönetimini kolaylaştıran with open() yapısı ile ele alınmıştır.
9. **Nesne Yönelimli Programlama (OOP):** Kodun daha organize ve yeniden kullanılabilir olmasını sağlayan class (sınıf), init metodu ve self anahtar kelimesi gibi temel OOP kavramlarına giriş yapılmıştır.
10. **Öğrenme ve Pratik:** Eğitimin sonunda, öğrenilen bilgileri pekiştirmek için **FreeCodeCamp** gibi kaynaklar önerilmiş ve sürekli pratik yapmanın önemi vurgulanmıştır.
Hata Yönetimi (try-except): Programın çökmesini engellemek ve kod hatalarını yönetmek için **try-except** bloklarının kullanımı, spesifik hata türlerinin (**ValueError**, **ZeroDivisionError**) yakalanması ve **raise** ile bilinçli hata fırlatma konuları ele alınmıştır.
11. **Dosya İşlemleri (with open):** Metin dosyalarını okuma (r), yazma (w) ve ekleme (a) modları ile açma işlemleri, kaynak yönetimini kolaylaştıran **with open()** yapısı ve Türkçe karakterler için **encoding='utf-8'** kullanımı gösterilmiştir.
12. **Nesne Yönelimli Programlama (OOP):** Kodun daha organize ve yeniden kullanılabilir olmasını sağlayan **class** (sınıf), **__init__** metodu, **self** anahtar kelimesi ve sınıflar arası özellik aktarımı için **inheritance** (miras alma) gibi temel OOP kavramlarına giriş yapılmıştır.

13. **Öğrenme ve Pratik:** Python becerilerini geliştirmek için **FreeCodeCamp** gibi platformlar önerilmiş, sürekli pratik yapmanın ve kendi kendine proje geliştirmenin önemi vurgulanmıştır.
14. **Temel Veri Kütüphaneleri (Giriş):** Eğitimin ikinci bölümünde, bootcamp boyunca sıkça kullanılacak olan temel veri kütüphanelerine geçilmiştir.
15. **NumPy (Sayısal Python):** Hızlı ve verimli sayısal hesaplamalar için kullanılan **NumPy** kütüphanesi tanıtılmıştır. Python listelerinden farklı olarak **vektörel işlemler** yapabilme yeteneği ve **np.array()** ile **NumPy** dizisi oluşturma konuları işlenmiştir.
16. **Pandas (Veri Manipülasyonu):** Veri manipülasyonu ve analizi için endüstri standardı olan **Pandas** kütüphanesine giriş yapılmıştır. **DataFrame** ve **Seri** veri yapıları, **.head()**, **.info()**, **.describe()** gibi temel metotlar ve eksik verilerle başa çıkmak için **.isna().sum()**, **fillna()** gibi fonksiyonlar gösterilmiştir.
17. **Veri Görselleştirme (Matplotlib & Seaborn):** Veriyi anlamak ve içgörüler elde etmek için kullanılan **Matplotlib** ve **Seaborn** kütüphaneleri tanıtılmıştır. **countplot** ve **heatmap** gibi temel grafik türleri ile veri setinin dengesini ve eksik verilerin dağılımını görselleştirme pratikleri yapılmıştır.
18. **Scikit-learn (Makine Öğrenmesi):** Python'da makine öğrenmesi için endüstri standardı olan **Scikit-learn** kütüphanesine giriş yapılmıştır. Bir makine öğrenmesi projesinin 5 adımlık temel iş akışı (veri hazırlama, bölme, model seçme, eğitme, değerlendirme) anlatılmıştır.
19. **Temel ML Projesi:** Öğrenilen tüm kütüphaneler kullanılarak uçtan uca basit bir sınıflandırma projesi gerçekleştirilmiştir. Veri setini eğitim ve test olarak ayırmak için **train_test_split**, model olarak **KNeighborsClassifier**, modeli eğitmek için **model.fit()** ve tahmin yapmak için **model.predict(X_test)** gibi temel **Scikit-learn** fonksiyonları kullanılmıştır.

Zero2end ML Bootcamp kapsamında Göker Güner tarafından verilen "Python'a Giriş ve Temel Veri Kütüphaneleri" eğitimi, Python programlama dilini yapay zeka perspektifiyle ele almayı amaçlamıştır. Eğitim, teorik anlatımdan ziyade pratik uygulamalara odaklanmış ve tüm kodlama işlemleri için web tabanlı ve erişimi kolay bir ortam olan **Google Colab** platformunu kullanmıştır. Amaç, katılımcılara sadece Python'ın sentaksını öğretmek değil, aynı zamanda veri odaklı projelerde hangi yapıların neden önemli olduğuna dair bir "kulak aşinalığı" kazandırmaktır.

Eğitimin gelişme bölümü, Python'ın temel yapı taşlarından başlayarak sistemli bir şekilde ilerlemiştir. İlk olarak, **değişkenler**, temel **veri tipleri** (string, integer, boolean) ve çeşitli **operatörler** tanıtılmıştır. Ardından, veri bilimi projelerinin temelini oluşturan **veri yapıları** detaylandırılmış; özellikle liste ve sözlük (dictionary) yapıları ile **List Comprehension** gibi verimli kullanım yöntemleri üzerinde durulmuştur. Kodun akışını kontrol etmek için if-else gibi **koşul blokları** ve for, while gibi **döngüler** işlenmiştir. Kod tekrarını önlemek ve modüler bir yapı oluşturmak amacıyla def ile **fonksiyon** tanımlama, parametreler, argümanlar ve lambda fonksiyonları gibi konular ele alınmıştır. Son olarak, programın dayanıklılığını artıran try-

except ile **hata yönetimi**, with open() ile **dosya işlemleri** ve daha organize kod yazmayı sağlayan **Nesne Yönelimli Programlama (OOP)** temellerine giriş yapılmıştır.

Eğitim, Python'ın sadece bir programlama dili olmadığını, aynı zamanda yapay zeka ve veri bilimi gibi alanlarda karmaşık problemleri çözmek için esnek ve güçlü bir araç olduğunu vurgulayarak sona ermiştir. Göker Güner, öğrenilen bilgilerin kalıcı hale gelmesi için **sürekli pratik yapmanın** kritik önem taşıdığını belirtmiş ve katılımcıları **FreeCodeCamp** gibi kaynaklardan yararlanarak kendi projelerini geliştirmeye teşvik etmiştir. Sonuç olarak, bu oturum, katılımcılara Python ile veri odaklı projeler geliştirmek için gerekli olan temel bilgi ve becerileri kazandırırken, onları kendi öğrenme yolculuklarında ilerlemeleri için motive etmiştir.

Göker Güner tarafından sunulan "Python'a Giriş ve Temel Veri Kütüphaneleri" eğitiminin ikinci bölümü, Python'ın temel konularından sonra, veri bilimi ve makine öğrenmesi projelerinin temelini oluşturan kütüphanelere odaklanmıştır. Bu bölümün amacı, bootcamp boyunca sıkça kullanılacak olan **NumPy, Pandas, Matplotlib, Seaborn** ve **Scikit-learn** gibi temel araçları tanıtmak ve bu kütüphanelerin bir makine öğrenmesi projesinin iş akışında nasıl bir araya geldiğini göstermektir.

Eğitimin gelişme bölümü, Python'ın ileri konularından başlayarak veri kütüphanelerine doğru bir geçiş yapmıştır. İlk olarak, kodun dayanıklılığını artıran **try-except** ile hata yönetimi, **with open()** ile güvenli dosya işlemleri ve daha organize kod yazmayı sağlayan **Nesne Yönelimli Programlama (OOP)** temelleri ele alınmıştır. Ardından, hızlı sayısal hesaplamalar için temel olan **NumPy** ve vektörel işlem yetenekleri tanıtılmıştır. Veri manipülasyonu için endüstri standardı olan **Pandas** kütüphanesine geçilmiş; **DataFrame** yapısı, **.head()**, **.info()** gibi keşifsel veri analizi metotları ve eksik verilerle başa çıkma teknikleri gösterilmiştir. Veriyi daha iyi anlamak için **Matplotlib** ve **Seaborn** ile veri görselleştirme pratikleri yapılmıştır. Son olarak, tüm bu adımlar, makine öğrenmesi için temel kütüphane olan **Scikit-learn** kullanılarak bir araya getirilmiş ve **train_test_split**, **model.fit()** ve **model.predict()** gibi fonksiyonlarla uçtan uca basit bir sınıflandırma projesi gerçekleştirilmiştir.

Eğitim, Python ve temel veri kütüphanelerinin bir makine öğrenmesi projesini hayata geçirmek için nasıl bir bütün olarak çalıştığını göstererek sona ermiştir. Katılımcılar, veri hazırlama, analiz, görselleştirme ve modelleme adımlarından oluşan temel iş akışını pratik bir örnek üzerinden deneyimlemişlerdir. Göker Güner, bu eğitimin bootcamp boyunca yapılacak daha karmaşık projeler için sağlam bir temel oluşturduğunu belirtmiş ve öğrenilen bilgileri pekiştirmek için **FreeCodeCamp** gibi kaynaklardan yararlanarak sürekli pratik yapmanın önemini vurgulamıştır. Sonuç olarak, bu oturum, katılımcılara veri bilimi araç setinin temel bileşenlerini tanıtırken, onları kendi projelerini geliştirmeleri için gerekli olan temel bilgi ve motivasyonla donatmıştır.