MAKİNE ÖĞRENMESİ OPERASYONLARI

Giriş:

[Students Exam Scores: Extended Dataset | Kaggle](https://www.kaggle.com/datasets/desalegngeb/students-exam-scores) adlı veri setimiz bir okuldaki öğrencilerin girdikleri 3 sınava etki eden sosyo-ekomomik faktörlerle verilmiş bir datasettir.

Feature’lar:

Gender: Öğrencinin Cinsiyeti

EthnicGroup: Öğrencinin etnik grubu (A-E)

ParentEduc: Ebevenylerin eğitim düzeyi (lise, lisans, yüksek lisans , )

LunchType: Okul öğle yemeği türü (standart, ücretsiz,indirimli)

TestPrep: Sınava hazırlık kursu (tamamlandı, yok)

ParentMartialStatus: Ebevenylerin medeni durumu (evli, bekar, dul, boşanmış)

PracticeSport: Öğrencinin spor yapma sıklığı (hiç, bazen,düzenli)

IsFirstChild: Ailenin ilk çocuğu olup olmadığı (evet, hayır)

NrSiblings: Öğrencinin kardeş sayısı (0, 7 arasında)

TransportMeans Okula ulaşım aracı (servis, özel araç)

WklyStudyHours: Haftalık ders çalışma saati (5 saatten az , 5-10 saat arasında, 10 satten fazla)

MathScore: Matematik sınavı puan (0-100)

ReadingScore: Okuma sınavı puanı (0-100)

WritingScore: Yazma sınavı puanı (0-100)

Analitik sorular

1)Feature’lardan hangisi test scoruna en çok etki eder

2)Test puanlarını etkileyecek etkileşimli özellikler nelerdir?

1. Öğrenci Sınav Sonucu Verileri (2 puan)

I. Her özelliği (sütun) analiz edin ve ardından istatistiklerini verin. (1 puan)

1)GENDER

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, renklilik içeren bir resim

Açıklama otomatik olarak oluşturuldu

2)ETNİK GROUP

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

EĞİTİM DURUMU:

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

LUNCH TYPE

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

TEST PREP

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

MARTİAL STATUS

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazılım, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

SPORTS:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, diyagram, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

FIRST CHİLD

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

SİBLİNGS

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

WEEKLY HOURS

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

TRANSPORTATİON

metin, ekran görüntüsü, ekran, görüntüleme, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

MATH SCORE

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, diyagram, daire içeren bir resim

Açıklama otomatik olarak oluşturuldu

READİNG SCORE

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

WRİTİNG SCORE

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

PLOT SUMMARY

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

II. Verileri temizleyin ve verileri eğitim için hazırlayın. Gereksiz sütunu silme, eksik verileri doldurma vb. ön işlem adımlarını uygulayın (1 puan)

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

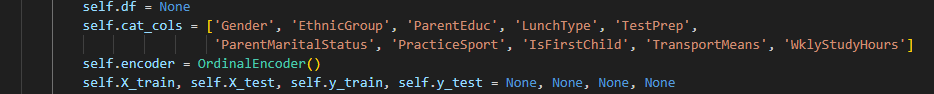
Bu fonksiyon dropna() ile eksik verilere sahip satırları veri çerçevesinden siler.

“Cat\_cols” ile kategorik olarak ayrılıyor.

“OrdinalEncoder” ile kategorik sütunlar sayısal değerlere dönüştürülüyor

2. MLOps Tasarım Kalıpları (3 puan)

I. Öğrenci sınav sonucu verilerine göre model tasarımı uygulayın. (1,5puan)



metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Numerical Inputs (Sayısal Girişler): Sayısal veriler olan 'MathScore', 'ReadingScore' ve 'WritingScore' sütunları kullanılmıştır.

Categorical Inputs (Kategorik Girişler): Kategorik veriler olan 'Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild', 'TransportMeans' ve 'WklyStudyHours' sütunları kullanılmıştır.

II. Öğrenci sınav sonucu verilerine göre yazılım ürün tasarımı uygulayın.(1,5 puan)

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

load\_data() fonksiyonu, belirtilen dosya yolundan veri setini yükleyerek bir Pandas DataFrame'e aktarır ve daha sonra bu DataFrame'in ilk beş satırını ekrana yazdırır.

preprocess\_data() fonksiyonu, veri setindeki eksik değerleri çıkarır ve kategorik sütunları ordinal kodlamaya tabi tutar. Bu işlem, kategorik verileri sayısal değerlere dönüştürerek makine öğrenimi algoritmalarının veri setini işleyebilmesini sağlar. Ardından, işlenmiş veri setinin ilk beş satırını ekrana yazdırır.

split\_data() fonksiyonu, veri setini eğitim ve test setlerine böler. Bu işlemde, bağımlı değişkenler (özellikler) ve hedef değişken (etiket) ayrılır. Veri setinin %80'ini eğitim için kullanırken, %20'sini test için ayırır. Bölünmüş veri setlerini X\_train, X\_test, y\_train ve y\_test değişkenlerine atar.

train\_model() fonksiyonu, Random Forest Regressor modelini eğitir. Bu, veri setinin eğitim kısmını kullanarak modelin öğrenmesini sağlar.

predict() fonksiyonu, eğitilmiş modeli kullanarak test veri seti üzerinde tahmin yapar. Tahminleri y\_pred değişkenine atar.

calculate\_rmse() fonksiyonu, gerçek ve tahmin edilen değerler arasındaki ortalama kare hatasını (RMSE) hesaplar. RMSE'yi hesaplar ve ekrana yazdırır.

analyze\_results() fonksiyonu, gerçek ve tahmin edilen değerleri içeren bir DataFrame oluşturur. Bu DataFrame, gerçek ve tahmin edilen değerleri MathScore, ReadingScore ve WritingScore sütunları şeklinde içerir. İlk 10 satırını ekrana yazdırır.

visualize\_predictions() fonksiyonu, gerçek ve tahmin edilen değerleri kullanarak bir scatter plot grafiği oluşturur. X ekseni gerçek değerleri, Y ekseni tahmin edilen değerleri temsil eder. Ayrıca, doğru bir tahmin hattı çizilir. Oluşturulan grafiği ekranda gösterir.

visualize\_feature\_importance() fonksiyonu, modelin özelliğe olan önemini görselleştirir. Özellik önem sıralamasını bir çubuk grafik olarak çizer ve en önemli 10 özelliği gösterir.

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, görüntüleme, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ve böylece yazılım ürünü tasarlamış oluruz

3) MLOPS (5 PUAN)

1)Veri üzerine açıklanabilir yapay zeka tekniklerini uygulayın.

VERİ ÖN İŞLEME:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Veri setinin yüklenmesi ve eksik değerlerden arındırılması, kategorik özelliklerin ordinal kodlamaya dönüştürülmesi

MODEL EĞİTİMİ

metin, ekran görüntüsü, yazı tipi, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Split data ve train model fonksiyonları ile veri eğitim ve test verilerine bölündü. Veri seti bir random forest üzerinde eğğitildi.

TAHMİNLERİN YAPILMASI

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Eğitilmiş modelin test veri seti üzerine tahminde bulunur

SONUÇ ANALİZİ:

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Tahminler gerçek değerlerle karşılaştırıldı ve analiz edildi. Rmse hesaplandı. Ve analiz görüntülendi. Önem sırasına göre sıralandı

2)Veri güvenliğini nasıl sağlarsınız veri üzerinden örnekler ile açıklayın

VERİ DOSYASININ YÜKLENMESİ:

Veri kaynağı yüklenirken load data fonksiyonu ile yüklenir. Veri dosyasının kyanağına ve erişim izinlerine dikkat edilmelidir.

ÖN İŞLEME ADIMLARI:

Veri ön işleme adımları preprocess data fonsiyonu ile gerçekleştirilir. Veri kümesi kişilere ait özel veriler içeridğinden bilgilerin korunması önemlidir. Hassas bilgiler içeren sütunlar gizlenebilir.

ANALİZ

Analiz gerçekleştirilirken veri çerçevesindeki hassas bilgiler gizlenebilir.

Veri güvenliğini sağlamak için çeşitli araçlar ve teknikler kullanılabilir. Bunlar şunlardır:

Veri güvenliğiyle ilgili olarak, aşağıdaki önlemler alınabilir:

Veri şifreleme: Hassas verilerin yetkisiz erişimlere karşı korunması için simetrik veya asimetrik şifreleme yöntemleri kullanılabilir.

Veri anonimleştirme: Kişisel verilerin gizliliğini korumak için veri anonimleştirme teknikleri uygulanabilir. Bu teknikler, kişisel bilgilerin gizlenmesi veya kimlikle ilişkilendirmenin zorlaştırılmasıyla veri setinin anonimleştirilmesini sağlar.

Veri maskeleme: Hassas verilerin geçici veya gizli değerlerle değiştirilmesini sağlayan maskeleme teknikleri kullanılabilir. Böylece, veri setinin güvenliği korunurken işlem yapılabilirlik sağlanır.

Veri erişim kontrolü: Verilere erişimi kontrol etmek ve yetkilendirmek için veri erişim kontrol araçları kullanılabilir. Veritabanı yetkilendirmesi, kullanıcı rolleri ve izinleri belirleme gibi yöntemlerle verilere sınırlı erişim sağlanabilir.

Güvenli veri depolama: Veri setinin güvenli bir şekilde depolanması için güvenli veri depolama araçları kullanılabilir. Veritabanları, bulut tabanlı hizmetler veya güvenli sunucu çözümleri bu amaçla tercih edilebilir.

Veri güvenlik denetimleri: Veri güvenliği politikalarının uygulanmasını ve denetlenmesini sağlamak için veri güvenlik denetim araçları kullanılabilir. Bu araçlar, veri güvenliği açıklarını tespit etmek, saldırıları izlemek ve güvenlik ihlallerini önlemek için kullanılabilir.

3)Sürekli entegrasyon nasıl sağlarsınız, örnekler ile açıklayın. (1 puan)

Sürekli entegrasyon sağlanırken aşağıdaki adımlar izlenir:

Kod, bir versiyon kontrol sistemi kullanılarak yönetilir. Örneğin, popüler bir versiyon kontrol sistemi olan Git kullanılabilir. Bu şekilde kodun geçmiş versiyonları takip edilebilir, değişiklikler geri alınabilir ve farklı bir dalda çalışılabilir.

Kod, otomatik testlerle sürekli olarak kontrol edilir. Örneğin, kendi kodunuzda bir login() fonksiyonu olduğunu varsayalım. Bu fonksiyonun doğru çalıştığından emin olmak için bir test senaryosu oluşturulabilir. Örneğin, geçerli bir kullanıcı adı ve şifreyle doğru bir şekilde giriş yapılıp yapılmadığı kontrol edilebilir.

Sürekli entegrasyonu otomatikleştirmek için bir sürekli entegrasyon aracı kullanılır. Bu araç, kodu belirli bir zaman aralığında veya her bir değişiklik yapıldığında otomatik olarak derler, test eder ve dağıtır. Örneğin, CircleCI veya GitLab CI/CD gibi sürekli entegrasyon araçları kullanılabilir.

Kod, sürekli entegrasyon aşamasında otomatik olarak derlenir ve dağıtılır. Bu, kodun hızlı bir şekilde test edilmesini ve dağıtım sürecinin otomatikleştirilmesini sağlar. Örneğin, kendi kodunuzda bir deploy() fonksiyonu olduğunu düşünelim. Bu fonksiyon, kodu hedef sunucuya otomatik olarak yüklemek ve uygulamayı güncellemek için kullanılabilir.

Hataların ve hatalı durumların hızlı bir şekilde tespit edilmesi için hata izleme sistemleri kullanılır. Örneğin, kendi kodunuzda bir log\_error() fonksiyonu olduğunu varsayalım. Bu fonksiyon, hataları yakalayarak bir hata günlüğüne kaydedebilir ve geliştiriciye bildirim gönderebilir.

Sürekli ortam yüklemesi sağlanırken ise aşağıdaki adımlar izlenir:

Yapılandırmalar farklı ortamlar için yönetilir. Örneğin, kendi kodunuzda bir config.py dosyası kullanarak farklı ortamlar için yapılandırmaları yönetebilirsiniz. Bu şekilde, veritabanı bağlantıları ve API anahtarları gibi yapılandırmaları dinamik olarak belirleyebilirsiniz.

Kod, otomatik olarak hedef ortama dağıtılır. Bunun için bir deploy() fonksiyonu kullanılabilir. Bu fonksiyon, güncellenmiş kodu sunuculara veya bulut hizmetlerine otomatik olarak dağıtabilir.

Ortamlara özgü yapılandırmalar kullanılır. Örneğin, get\_db\_connection() fonksiyonunda, veritabanı bağlantısını belirlemek için ortama özgü yapılandırmalar kullanılabilir. Bu sayede, geliştirme, test veya üretim ortamlarına dinamik olarak bağlantı yapılabilir.

Test ortamı kullanılarak sürekli ortam yüklemesi yapılır. Bir run\_tests() fonksiyonu kullanılarak otomatik test senaryoları çalıştırılabilir ve her güncelleme sonrasında kod test edilip istikrarı sağlanabilir.

Hataların yakalanması, loglama ve izleme mekanizmalarıyla sağlanır. Örneğin, log\_error() fonksiyonu ile hatalar loglanabilir ve loglar izlenebilir. Bu sayede, hatalar erken tespit edilebilir ve performans sorunları izlenebilir.

4)Sürekli ortam yüklemesi nasıl sağlarsınız, örnekler ile açıklayın. (1 puan)

Bağımlılıklar yönetilmelidir: Kullanılan tüm kütüphaneler ve bağımlılıklar güncel ve tutarlı bir şekilde yönetilmelidir. Versiyon uyumsuzluklarını önlemek için requirements.txt veya Pipfile gibi bir bağımlılıklar dosyası oluşturulmalı ve bu dosyada bağımlılıklar listelenmelidir.

Ortam parametreleri yapılandırılmalıdır: Kodun doğru bir şekilde çalışabilmesi için gereken ortam parametreleri (dosya yolları, API anahtarları, veritabanı bağlantı bilgileri vb.) yapılandırılabilir hale getirilmelidir. Çevre değişkenleri veya yapılandırma dosyaları kullanılarak farklı ortamlarda (geliştirme, test, üretim) kodun sorunsuz bir şekilde çalıştırılması sağlanmalıdır.

Otomatik testler eklenmelidir: Kodun hata ayıklanmış ve istikrarlı olduğundan emin olmak için otomatik testler oluşturulmalıdır. Farklı test türlerinden (unit testleri, entegrasyon testleri, kabul testleri) yararlanarak kodun doğru çalıştığını ve potansiyel hataların tespit edildiğini sağlamalıdır.

Hata işleme ve günlükleme doğru bir şekilde gerçekleştirilmelidir: Uygulama, hata durumlarıyla başa çıkabilmeli ve kullanıcı dostu hata mesajları veya günlük kayıtları üretebilmelidir. Bu sayede hatalar hızlı bir şekilde tespit edilebilir ve düzeltilebilir.

Performans optimizasyonuna dikkat edilmelidir: Büyük veri setleri veya yoğun iş yükleriyle çalışan bir uygulama geliştiriliyorsa, performans optimizasyonuna önem verilmelidir. Kod içerisinde gereksiz döngüler veya yavaş çalışan işlemler kontrol edilmeli ve gerekli iyileştirmeler yapılmalıdır.

Güvenlik önlemleri alınmalıdır: Uygulamanın güvenliği için gerekli önlemler alınmalıdır. Kullanıcı girişleri doğru bir şekilde denetlenmeli, kimlik doğrulama ve yetkilendirme mekanizmaları kullanılmalı ve güvenlik testleriyle olası güvenlik açıkları tespit edilmelidir.

İzleme ve hata takibi yapılmalıdır: Uygulama sürekli olarak izlenmeli ve performans, kullanılabilirlik ve hata durumları gibi önemli metrikler takip edilmelidir. Böylece uygulamanın sağlığı kontrol edilebilir ve hızlı müdahalelerde bulunulabilir.

Dağıtım süreci otomatikleştirilmelidir: Continous Delivery sürecinde, kodun otomatik olarak test edilmesi, paketlenmesi ve dağıtılması için bir otomatikleştirme aracı kullanılmalıdır. Bu, insan hatalarını azaltır, süreci hızlandırır ve dağıtımların tutarlı olmasını sağlar.

Yukarıdaki adımlar takip edilerek, Continous Delivery sürecinde kod daha güvenilir, ölçeklenebilir ve hata toleranslı hale getirilebilir. Ancak, her zaman uygulamanın özel gereksinimlerine ve organizasyonun politikalarına uygun olarak süreci uyarlamak önemlidir.

4. Dersin Geri Bildirimi (3 puan)

I. Lütfen dönem boyunca kursta neleri iyi yaptığımızı belirtin (1 puan)

Anlatımınız samimi, öğrencilerle etkileşiminiz mükemmel, ders içinde kahoot yapılması pekiştirici

II. Lütfen dönem boyunca kursta neleri kötü yaptığımızı belirtin. (1puan)

Dersin süresini uzatmak lazım konular arasında yaptığınız muhabbetler fazla sarıyor 😊

III. Sonraki dönemler için kursu nasıl iyileştirebiliriz? (1 puan)

Hocam bu sorun sizden kaynaklı değil ama bir dahaki sefere öğrenci işlerine söyleyebilirseniz lütfen 2. Sınıf makine öğrenmesi görmemiş bir öğrenciye makine öğrenme operasyonları vermesinler.

Ek:

