PYTHON İLE SINIFLANDIRMA - REGRESYON - KÜMELEME İŞLEMLERİ

Ali Burhan GÜNCAN

Siliconmade Akademi

SA003 - Python ile İleri Seviye Yapay Zeka Uzmanlığı Eğitimi

Utku KÖSE - Emre YAZICI

Mart 3, 2024

PYTHON İLE SINIFLANDIRMA - REGRESYON - KÜMELEME İŞLEMLERİ

GİRİŞ

Yapay Zeka, endüstrileri, teknolojiyle etkileşimleri ve dolaylı olarak insan hayatını yeniden şekillendiren bir teknolojidir. Temelde, sistemlerin verilerden öğrenmesini sağlayan makine öğrenimi yatmaktadır. Makine öğreniminin yapay zekadaki önemi, büyük verilerden görüş elde etme, veriye dayalı karar alma ve bu kararlar ile çeşitli avantajlar elde etme yeteneğinden kaynaklanmaktadır.

Makine öğrenimi, görevleri otomatikleştirerek, iş akışlarını düzene sokarak ve insan müdahalesini en aza indirerek endüstriler genelinde otomasyonu ve verimliliği arttırmaktadır. Örneğin bir üretim hattında önceki ürünlerden edindiği bilgiler ile yeni ürünün kalite kontrol süreçlerinde insan müdahalesini ortadan kaldırırken; bir diğer yandan sağlık sektöründe yıllardır biriken hastalık tanı verileri ile yeni muayene süreçlerinde hızlı ve doğru hastalık tanımlaması yaparak, kişiye özel tedaviler önerebilir.

Bunların yanında makine öğrenimi, doğal dil işleme, görüntü tanımlama - işleme, ses tanımlama - işleme gibi alanlarda akıllı sistemlere güç sağlar. Böylece sanal asistanların, sohbet robotlarının, öneri sistemlerinin vs. dil çevirisinden yüz tanımaya kadar kullanıcı deneyimlerini geliştirmesine ve insan-bilgisayar etkileşimini kolaylaştırmasına yardımcı olur.

Makine öğrenimi sistemlerinin yinelemeli bir doğası vardır. Bu durum yapay zeka sistemlerinde sürekli iyileştirmeye ve adaptasyona imkan sağlar. Modeller zaman içinde gelişerek tahminlerini ve performanslarını iyileştirebilir. Böylece her bir aşamada daha karmaşık, daha kaotik bir sorunun veya görevin üstesinden gelebilecek bir kapasiteye ulaşır.

Özetle, makine öğrenimi yapay zekanın temel taşıdır. Makine öğrenimi veri analizini, süreçlerin otomasyonunu, insan - bilgisayar etkileşimini ve sürekli öğrenmeyi mümkün kılar. Etkisi dolaylı veya direk bir şekilde doğayı, insanları, endüstrileri kapsamaktadır. Gelecekteki inovasyon süreçleri için neredeyse limitsiz bir potansiyele sahiptir.

UYGULAMALAR

Sınıflandırma İşlemleri

Bu raporda sınıflandırma işlemleri için Diabetes Prediction Dataset veriseti kullanılmıştır.

Veri setinin ilk durumu proje dosyaları içinde "diabetes_prediction_dataset_org.csv" ismi ile bulunabilir. Veri seti, ilk durumunda 9 sütundan oluşmaktadır:

- gender: Bireyin cinsiyetini ifade eden sütundur.
- age: Bireyin yaşını ifade eden sütundur.
- hypertension: Atardamarlardaki kan basıncının sürekli olarak yükseldiği tıbbi durumun bireyde olup olmadığını gösteren sütundur.
- heart_disease: Bireyin kalp rahatsızlığına sahip olup olmadığını gösteren sütundur.
- smoking_history: Bireyin sigara içme durumu tarihçesini temsil eden sütundur.
- bmi: Bireyin vücut kitle endeksini gösteren sütundur.
- HbA1c_level: Bireyin son 2-3 aydaki ortalama kan şekeri düzeyinin ölçüsünü gösteren sütundur.
- blood_glucose_level: Belirli bir zamanda kan dolaşımında bulunan glikoz miktarını ifade eden sütundur.
- diabetes: Bireyin diyabet olup olmadığını ifade eden sütundur. Hedef değişkendir.

"diabetes_prediction_dataset_org.csv" isimli veri seti "classification_data_prep.py" isimli python dosyasında bulunan süreçlerden geçirilerek, yapay zeka kullanımına hazır hale getirilmiştir.

Veri hazırlama sürecinde 18 adet boş satır silinmiştir. "Dummy Index" - "Mapping" gibi işlemler kullanılmıştır. Sonuç sütunu ile korelasyonu 5%'nin altında olan sütunlar PCA ile tek sütun haline çevrilmiştir. Yeni sütunun da korelasyonu 5%'nin altında kalınca işlem iptal edilmiştir. Son durumda korelasyonu 5%'nin altında olan bütün sütunlar silinmiştir. Verinin son hali "diabetes_prediction_dataset_edited.csv" olarak dosyalar içine kaydedilmiştir.

Son durumda yapay zeka süreçleri için **"MLPClassifier"** ve **"DecisionTreeClassifier"** kullanılmıştır. Bu veri setinde DecisionTreeClassifier bir miktar daha iyi sonuç vermektedir.

Dosyalar içinde bulunan "classification_fit.py" yapay zeka kullanım süreçlerini, "classification_run_file.py" arayüz kullanımını gerçekleştirmektedir.

Regresyon İşlemleri

Bu raporda regresyon işlemleri için House Price Prediction Challenge veriseti kullanılmıştır.

Veri setinin ilk durumu proje dosyaları içinde "House_Price_Prediction_challenge_org.csv" ismi ile bulunabilir. Veri seti, ilk durumunda 12 sütundan oluşmaktadır:

- POSTED_BY: Mülkün kim tarafından ilana verildiğini gösteren sütundur.
- UNDER_CONSTRUCTION: Mülkün yapım aşamasında olup olmadığını belirten sütundur.
- RERA: Mülkün "RERA" onayını temsil eden sütundur.
- BHK_NO.: Mülkün oda sayısını gösteren sütundur.
- BHK_OR_RK: Mülk tipini ifade eden sütundur.
- SQUARE_FT: Mülkün yüz ölçümünü gösteren sütundur.
- READY_TO_MOVE: Mülke taşınabilirliği ifade eden sütundur.
- RESALE: Mülkün ikinci el olup olmadığını belirten sütundur.
- ADDRESS: Mülkün adresini belirten sütundur.
- LONGITUDE: Mülkün boylam değerini ifade eden sütundur.
- LATITUDE: Mülkün enlem değerini ifade eden sütundur.
- TARGET(PRICE_IN_LACS): Mülkün fiyatını ifade eden sütundur. Hedef değişkendir.

"House_Price_Prediction_challenge_org.csv" isimli veri seti "scoring_data_prep.py" isimli python dosyasında bulunan süreçlerden geçirilerek, yapay zeka kullanımına hazır hale getirilmiştir.

Veri hazırlama sürecinde hedef sütunundaki anomaliler veri setinden çıkarılmıştır. Hedef sütunu, "hedef sütunu - hedef sütununun ortalama değeri" ile değiştirilmiştir. Adres sütunundan şehir ve ilçeler ayıklanmıştır. Veri seti üzerinde "Dummy Index" - "Mapping" - "Target Average" gibi işlemler kullanılmıştır. Son durumda korelasyonu 5%'nin altında olan sütunlar silinmiştir. Verinin son hali "House_Price_Prediction_challenge_edited.csv" olarak kaydedilmiştir.

Son durumda yapay zeka süreçleri için **"RandomForestRegressor"** ve **"GradientBoostingRegressor"** kullanılmıştır. Bu veri setinde GradientBoostingRegressor bir miktar daha iyi sonuç vermektedir.

Dosyalar içinde bulunan "scoring_fit.py" yapay zeka kullanım süreçlerini, "scoring_run_file.py" arayüz kullanımını gerçekleştirmektedir.

Kümeleme İşlemleri

Bu raporda kümeleme işlemleri için Shop Customer Data veriseti kullanılmıştır. Bu veri setinin bir hedef değer sütunu bulunmamaktadır.

Veri setinin ilk durumu proje dosyaları içinde "Customers_org.csv" ismi ile bulunabilir. Veri seti, ilk durumunda 8 sütundan oluşmaktadır:

- CustomerID: Müşteri kimlik numarasını ifade eden sütundur.
- Gender: Müşteri cinsiyetini ifade eden sütundur.
- Age: Müşteri yaşını ifade eden sütundur.
- Annual Income ($): Müşterinin yıllık gelirini ifade eden sütundur.
- Spending Score (1-100): Müşterinin harcama skorunu ifade eden sütundur.
- Profession: Müşterinin mesleğini ifade eden sütundur.
- Work Experience: Müşterinin toplam çalışma yılını ifade eden sütundur.
- Family Size: Müşterinin toplam aile bireyi sayısını ifade eden sütundur.

"Customers_org.csv" isimli veri seti "clustering_data_prep.py" isimli python dosyasında bulunan süreçlerden geçirilerek, yapay zeka kullanımına hazır hale getirilmiştir.

Veri hazırlama sürecinde yaş ve gelir sütunlarına, kümelendirme sonucu doğruluk oranlarını arttırabilmek için gruplandırma yapılmıştır. Cinsiyet sütunu 1 - 0 olarak "Map"lenmiştir. ID sütunu silinmiştir. Gruplandırılan yaş ve gelir sütunları ile meslek sütununa "Dummy Index" uygulanmıştır. Verinin son hali "Customers_edited.csv" olarak kaydedilmiştir.

Son durumda yapay zeka süreçleri için **"2 kümeli PCA Metodu"** ve **"10 kümeli PCA Metodu"** kullanılmıştır. 10 kümeli PCA Metodu daha iyi sonuç vermektedir.

Dosyalar içinde bulunan "clustering_fit.py" yapay zeka kullanım süreçlerini, "clustering_run_file.py" arayüz kullanımını gerçekleştirmektedir.

BULGULAR

Sınıflandırma İşlemleri

Yapay zeka süreçleri için **"MLPClassifier"** ve **"DecisionTreeClassifier"** kullanılmıştır.

MLPClassifier (Multi-Layer Perceptron Classifier): Sınıflandırma görevleri için kullanılan bir tür sinir ağı algoritmasıdır. Her biri bir sonraki katmana bağlanan birden fazla düğüm katmanından oluşur. Eğitim için "backpropagation" tekniğini kullanır.

DecisionTreeClassifier: Tahmin yapmak için karar ağacını kullanan bir sınıflandırıcı türüdür. Karar ağaçları, verileri özelliklere dayalı olarak alt kümelere böler ve her düğümde bilgi kazanımını en üst düzeye çıkarmak veya kirliliği en aza indirmek için kararlar alır. Yorumlanması kolaydır ve hem sayısal hem de kategorik verileri işleyebilir.

Uygulamada işlemler öncesi "Shuffle (Karıştırma)" yapıldığı için her çalışmasında farklı sonuçlar verecektir. "diabetes_prediction_dataset_edited.csv" veri setinin 100 kez çalıştırılması sonucu ulaşılan sonuçlar şu şekildedir:

- **MLPClassifier doğruluk oranları:**
  - Ortalama değer: 0.94595
  - Standart sapma: 0.00497
  - En yüksek değer: 0.95536
  - En düşük değer: 0.93152
- **DecisionTreeClassifier doğruluk oranları:**
  - Ortalama değer: 0.95352
  - Standart sapma: 0.00113
  - En yüksek değer: 0.95669
  - En düşük değer: 0.95066

DecisionTreeClassifier kullanımının bu veri setinde daha iyi sonuç verdiği söylenebilir.

Regresyon İşlemleri

Yapay zeka süreçleri için **"RandomForestRegressor"** ve **"GradientBoostingRegressor"** kullanılmıştır.

RandomForestRegressor: Karar ağaçlarına dayalı bir topluluk öğrenme yöntemidir. Eğitim sırasında birden fazla karar ağacı oluşturur ve regresyon görevleri için ayrı ayrı ağaçların ortalama tahminini çıkarır. Büyük veri kümeleriyle iyi çalışır.

GradientBoostingRegressor: Her ağacın bir öncekinin yaptığı hataları düzelttiği karar ağaçlarını sırayla oluşturan bir topluluk öğrenme tekniğidir. Ortalama kare hata gibi bir kayıp fonksiyonunu gradyan inişiyle en aza indirmeyi amaçlar. Hem regresyon hem de sınıflandırma görevleri için etkilidir.

Uygulamada işlemler öncesi "Shuffle (Karıştırma)" yapıldığı için her çalışmasında farklı sonuçlar verecektir. "House_Price_Prediction_challenge_edited.csv" veri setinin 100 kez çalıştırılması sonucu ulaşılan sonuçlar Tablo 1'de görülebilir.

Tablo 1

Sınıflandırma sonuçları

Analiz tipi	Ortalama d.	Standart s.	En yüksek d.	En düşük d.
R Square sonuçları				
RandomForestRegressor	0.84384	0.00390	0.85284	0.83496
GradientBoostingRegressor	0.85108	0.00358	0.86107	0.84320
Root Mean Square Error sonuçları				
RandomForestRegressor	15.49069	0.20394	16.05814	15.02269
GradientBoostingRegressor	15.12754	0.18862	15.56602	14.71059
Mean Absolute Error sonuçları				
RandomForestRegressor	10.72449	0.11507	11.08674	10.41774
GradientBoostingRegressor	10.50539	0.10827	10.78218	10.25409

GradientBoostingRegressor kullanımının bu veri setinde daha iyi sonuç verdiği söylenebilir.

Kümeleme İşlemleri

Yapay zeka süreçleri için **"2 kümeli PCA Metodu"** ve **"10 kümeli PCA Metodu"** kullanılmıştır.

PCA (Principal Component Analysis): Orijinal bilgilerin çoğunu korurken veri kümelerini daha düşük boyutlu bir alana dönüştürerek basitleştirmek için kullanılan bir boyut azaltma tekniğidir. Veri analizi ve makine öğreniminde görselleştirme, gürültü azaltma ve özellik çıkarma için yaygın olarak kullanılır.

Kümeleme işleminde işlemler öncesi "Shuffle (Karıştırma)" yapılmadığı için her çalışmasında aynı sonucu verecektir.

Kümeleme işleminde hedef sütunu olmadığı için şu şekilde bir hata modellemesi yapılmıştır:

- Elde edilen PCA verisi "detransform" edilmiştir.
- "Detransform verisi" ile orjinal veri setinin farkının mutlak değeri alınmıştır.
- Her bir satır için bu mutlak değerlerin karelerinin toplamının karekökü alınıp veri setinin orjinal haline yeni sütunlar olarak eklenmiştir.
- Son durumda her satırda PCA'in 2 ve 10 kümeli hali için hatayı temsil eden yeni iki sütun oluşmuştur.
- Bu sütunların ortalaması da bu çalışmada hata olarak tanımlanmıştır.

"Customers_edited.csv" veri setinin PCA ile kümelenmesi sonrası yapılan hata analizinde karşılaşılan sonuçlar şu şekildedir:

- **2 kümeli PCA için hata: 2.6309**
- **10 kümeli PCA için hata: 1.1127**

10 kümeli PCA kullanımının bu veri setinde daha iyi sonuç verdiği söylenebilir.

SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu raporda sınıflandırma işlemleri için Diabetes Prediction Dataset, regresyon işlemleri için House Price Prediction Challenge, kümeleme işlemleri için Shop Customer Data veri setleri kullanılmıştır.

Sonuçlarda şu maddeler görülmüştür:

- Sınıflandırma işlemlerinde bireylerin diyabet olma durumu 90%'nin üzerinde bir doğruluk ile tahmin edilmektedir.
- Regresyon işlemlerinde veri setindeki ortalama ev fiyatı 65 - 70'dir. Analizler sonucu Mean Absolute Error 10 mertebesindedir. Yani bir evin fiyatı ortalama 15% hata ile tahmin edilebilmektedir.
- Kümeleme işlemlerinde verinin PCA'ye sokulmadan önceki hali ile "detransform" edilen veri karşılaştırıldığında, verinin kümeleme işleminde doğru temsil edilme oranının yüksek olduğu rahatlıkla görülebilmektedir.

Tüm bunlar göz önüne alındığında, makine öğrenmesinin hayatın çok farklı alanlarında, birbirlerinden farklı amaçlar doğrultusunda, yüksek doğrulukla hizmet edebildiği söylenebilir. Bu güçlü teknolojinin kullanımında en temel limit insanın hayal gücü olabilir.

Gelecek çalışmalarda:

- Sınıflandırma ve regresyon konusunda daha karmaşık veri setleri kullanılabilir.
- Kümeleme işlemlerinde hedef sütun olan veri setleri denenebilir.
- Bu program için hazırlanan arayüz çok daha profesyonel bir hale getirilebilir.

KAYNAKLAR

-. (2020). House price prediction challenge [March 3, 2024]. https://www.kaggle.com/datasets/anmolkumar/house-price-prediction-challenge/data

-. (2023a). Diabetes prediction dataset [March 3, 2024]. https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

-. (2023b). Shop customer data [March 3, 2024]. https://www.kaggle.com/datasets/datascientistanna/customers-dataset/data

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Lundh, F. (1999). An introduction to tkinter. *URL: www. pythonware. com/library/tkinter/introduction/index. htm*.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *9th Python in Science Conference*.

Utku KÖSE. (n.d.). *Yapay zeka temelleri – önemli / yönlendirici konular*.