Veri Madenciliğine Giriş Vize Ödevi

Şeyma Nur Mutlu
170201004@kocaeli.edu.tr

***Özet*— Veri Madenciliğine Giriş dersi kapsamında yapmış olduğum bu projede Melbourne'deki ev fiyatları veri setini kullanarak veri ön işleme, lineer regresyon, karar ağaçları uygulamaları yaptım.**

Anahtar Kelimeler—Lineer Regresyon, Karar Ağaçları, Veri Ön İşleme

Giriş

Bu dokümanda, hazır veri seti üzerinde yapmış olduğum uygulamalar adım adım açıklanacaktır.

Temel Bilgiler

Uygulamalar, Python yazılım dili ile Jupyter Notebook üzerinde yapılmıştır. Ödev raporu IEEE formatında yazılmıştır. Veri seti, Kaggle'dan temin edilmiştir.

Uygulamada kullanılan hazır kütüphaneler;

- Seaborn
- Numpy
- Pandas
- Matplotlib

I. Veri Ön İşleme(Data Preprocessing)

Veri madenciliği uygulamalarında veri kalitesi, kilit konulardan biridir. Veri kalitesini arttırmak ve bu bağlamda uygulamalarımızdan daha güvenilir sonuçlar almak için veri ön işleme adımları uygularız.

Veri Ön İşleme Teknikleri;

- Veri Dönüştürme(Data Transformation)
    - Min-Max Normalizasyon
    - Z-Skor Normalizasyonu
    - Ondalık Normalizasyonu
- Veri Birleştirme
- Veri Temizleme
    - Eksik veriler içeren kayıtların göz ardı edilmesi
    - Eksik verilerin değişken ortalamasıyla doldurulması
    - Bir değişkenin tüm örneklemeleri için ortalama değer kullanılması
    - Eldeki verilere dayalı olarak en uygun değer seçilmesi
- Veri İndirgeme(Data Reduction)
    - Boyut İndirgeme(Dimensionality Reduction)
    - Veri Birleştirme(Aggregation)
    - Kesikli Hale Getirme(Discretization and Binarization)
    - Veri Sıkıştırma

Projede kullanılan veri ön işleme adımları ;

A. Özellikler Üzerinde İyileştirme

Elimizdeki veri üzerinde evlerin lokasyonunu tanımlamak için birden fazla özellik olduğunu görüyoruz. Sadece lokasyon tanımlamak için kullanılan sütunlar; Sburb, Address, PostCode, Lattitude, Longtitude. Aynı özelliği belirten sütunlar içerisinden özelliği spesifik olarak tanımlamaya yeten sutünları seçerek diğerlerini silebiliriz.

Method, SellerG, Propertycount ve Date sutünları da bu analiz için gereksiz görünüyor, bu bağlamda özellikleri silerek daha doğru bir analiz edebiliriz.

Binanın inşaa yılı(YearBuilt) özelliği yerine binanın yaşı için Yaş = Bulunduğumuz Yıl - Binanın İnşaa Yılı tanımlamasını yapabiliriz.

Silmek için **df.drop( )** metodunu kullanıyoruz. Bu fonksiyon, spesifik olarak silmek istediğiniz sütun veya satırları özelleştirerek kaldırabilmemizi mümkün kılar.

Yeni sütün eklemek içinse dataframe['Age'] = 2020 - dataframe['YearBuilt'] tanımlamasını yapıyoruz.

B. Kayıp Verilerin Doldurulması

Kayıp verileri doldurmak için öncelikle hangi özelliğimizde ne kadar kayıp olduğuna bakmalıyız. Kayıp verilerin sayılarını görmek için **dataframe.isnull().sum()** fonksiyonlarını kullanıyoruz. Bu noktada sonuçlara bakarak çıkarım yapmak ve veri üzerinde bazı değişiklikler yapmak durumundayız.

- Verimizi kullanarak evlerin fiyatları üzerinde çalışacağamızdan dolayı, fiyat özelliği boş olan veriler bizim işimize yaramıyor. Fiyat bilgisi olmayan tüm satırları sileceğiz.

C. Veri Bölme

Modeli, daha önce model tarafından görülmemiş yeni verilerle değerlendirmeniz gerekir. Bu verileri kullanmadan önce veri kümemizi bölerek bu gereksinimi gidereceğiz.

Bu işlem için **sklearn.model_selection** kütüphanesinden **train_test_split** fonksiyonunu kullanacağız.

Eğitilmiş veri üzerinde **dataframe_train.isnull().sum()** fonksiyonunu kullanarak eksik verilerimize bakıyoruz. Eksik değerleri **dropna()** fonksiyonu ile kaldırıyoruz.

D. Kategorik Verileri İşleme

İçlerinde kategorileri olan özelliklere bir göz atıp, önem derecelerini inceleyeceğiz.

II. Kullanılacak Veri Madenciliği Algoritmaları

A. Lineer Regresyon

Regresyon, veri kümesi içerisindeki değişkenler arasındaki ilişkileri incelemek için kullanılan istatistiksel bir yöntemdir.

Doğrusal regresyon, farklı değişkenler arasındaki ilişkiyi modellemek için kullanılan en temel tekniktir.

Temel olarak "basit" ve "çoklu regresyon" olarak iki kısımda incelenmektedir. Basit regresyon bir tane bağımlı değişken bir tane de bağımsız değişkenden oluşmaktadır.

$$y = \beta 0 + \beta 1x + e$$

- β0 kesişim noktası (intercept),
- β1 bağımsız değişkenin regresyon katsayısı (coefficient),
- e değeri ise hata değerini göstermektedir.

B. *Karar Ağaçları*

Karar Ağaçları (DT), hem classification (Sınıflandırma) hem de regression (regresyon) problemlerinde kullanılan makine öğrenmesinin en popüler algoritmalarından biridir. Karar ağaçlarında önce bir karar ağacı oluşturulmakta, daha sonra karar ağacından üretilen kurallar ile veri tabanında bulunan kayıtlar sınıflandırılmaktadır [8].

Karar ağaçlarında ID3, C4.5, Sliq, Sprint, CART, REP Tree, Random Forest, Logistic Model Tree gibi birçok algoritma kullanılabilmektedir. Veriler önce bu karar ağacı algoritmalarından birine gönderilir. Algoritma bu verileri işler ve bir karar ağacı oluşturur. Oluşturulan bu karar ağacı sınıfı bilinmeyen veriler üzerine uygulanarak bu verilerin sınıflarının tespit edilmesi sağlanır.

Karar ağacı metotlarında en ayırt edici niteliği belirlemek için her nitelik için bilgi kazancı ölçülür. Bilgi Kazancı ölçümünde Entropy kullanılır. Entropy rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir.

$$H = -\Sigma\, p(x) \log p(x) \text{ (Entropi Denklemi)}$$

Burada, p (x) belirli bir sınıfa ait grubun yüzdesini ve H ise entropiyi belirtmektedir.

Karar ağacımızın entropi değerini en aza indirgeyen bölünmeler yapmasını isteriz. En iyi bölünmeyi belirlemek içinde bilgi kazancını kullanırız. Bilgi kazancı aşağıdaki eşitlik ile hesaplanır:

$$Gain(S,D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$

Burada, S orijinal veri kümesidir ve D ise kümenin bölünmüş bir parçasıdır. Her V, S'nin bir alt kümesidir. V'nin tümü ayrıktır ve S'yi oluşturmaktadır. Bu durumda bilgi kazancı, bölünmeden önceki orjinal veri setinin entropisi ile her bir özniteliğin entropi değeri arasındaki fark olarak tanımlanmaktadır.

Karar ağaçlarının avantajlarından bazıları ise, kullanılan ağaç yapıları görselleştirilebilir dolayısıyla yorumlamanın nispeten kolay sayılabilmesi, hem sayısal hem de kategorik verileri işleyebilmesi, istatistiksel testler kullanılarak bir modelin doğrulanabilmesdir.

Karar ağaçlarının dezavantajlarından bazıları ise, veriyi iyi bir şekilde açıklamayan karmaşık ağaçlar üretilebilmesi, bu durumda ağaç dallanması takibinin zorlaşması, ezbere öğrenme ("over-fitting") yaşanabilmesidir.

III. ALGORITMALA SONUÇLARININ YORUMLANMASI

1. *Ortalama Mutlak Hata*

Veri kümesindeki tüm veri noktalarının mutlak hatalarının ortalamasıdır.

2. *Ortalama Hata Karesi*

Veri kümesindeki tüm veri noktalarının hatalarının karelerinin ortalamasıdır. En popüler metriklerden biridir.

3. *Medyan Mutlak Hata*

Veri kümesindeki tüm hataların medyanıdır. Bu metriğin temel avantajı, aykırı değerlere karşı sağlam olmasıdır. Test veri kümesindeki tek bir kötü nokta, ortalama bir hata ölçütünün aksine tüm hata ölçüsünü çarpıtmaz

4. *Açıklanan Varyans Puanı*

Modelimizin veri setimizdeki varyasyonu ne kadar iyi açıklayabildiğini ölçer. 1.0 puan, modelimizin mükemmel olduğunu gösterir.

5. *R2 Puanı*

Belirleme katsayısını ifade eder. Bu bize bilinmeyen örneklerin modelimiz tarafından ne kadar iyi tahmin edileceğini gösterir. Olası en iyi puan 1.0'dır, ancak puan negatif de olabilir.

Running LinearRegression()

Mean Absolute Error: 276907.10834066453

Mean Squared Error: 190378911450.44968

Explained Variance Error: 0.5535609893875975

RMSE: 436324.31911417644

R2 Score: 0.5535466961582458

Running DecisionTreeRegressor(max_depth=9, min_samples_split=4, random_state=1)

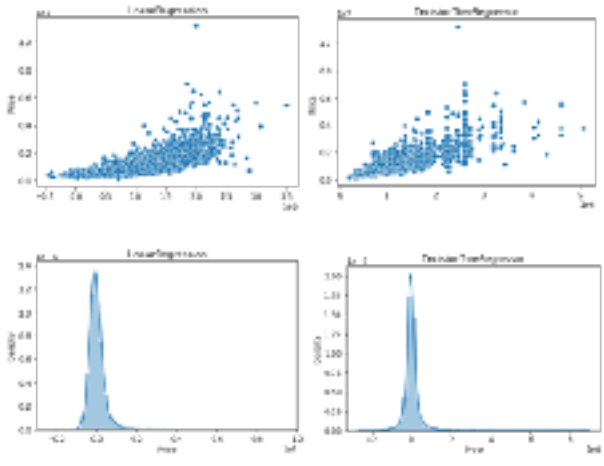
Mean Absolute Error: 210354.67170614714

Mean Squared Error: 135222152610.54773

Explained Variance Error: 0.6829068495830195

RMSE: 367725.64856227767

R2 Score: 0.6828935709022286



SONUÇ

R2 puanının 1'e yakınlığı, modelin verileri çok iyi tahmin edebildiği anlamına gelir. Her bir metriği takip etmek yorucu olabilir, bu nedenle modelimizi değerlendirmek için bir veya iki ölçüm seçeriz. İyi bir uygulama, ortalama hata karesinin düşük olduğundan ve açıklanan varyans puanının

yüksek olduğundan emin olmaktır. Bu bağlamda analiz sonuçlarına bakarak Karar Ağacı algoritmasının daha iyi sonuç verdiğini söyleyebiliriz.

REFERENCES

1. E.K. Ulgen, Makine Öğrenimi Bölüm-5, Medium
2. E.K. Ulgen, Regresyon Bölüm-6, Medium
3. B.Y. Tekin, Python ile Veri Ön İşleme – Data Preprocessing, Glabal AI Hub
4. A. Oguzlar, VERİ ÖN İŞLEME, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Sayı: 21, Temmuz-Aralık 2003, ss. 67-76.
5. N. Ilkbahar, Data Preprocessing(Veri Ön İşleme), Medium
6. Tan, Steinbach, Karpatne, Kumar, Introduction to Data Mining, Model Overfitting
7. T. Pino, Price Analysis and Linear Regression, Kaggle
8. Silahtaroğlu, G., Veri madenciliği. Papatya Yayınları, İstanbul, 2008