BSM622

Veri Analitiği ve İstatistik

5. Hafta: Regresyon ve Tahmin

Doç. Dr. Onur Polat

Kaynak: Bruce, P., Bruce, A., Gedeck, P. (2020). Practical Statistics for Data Scientists, 2. Baskı, O'Reilly — Bölüm 4

Bu Hafta Neler Öğreneceğiz?

- Basit Doğrusal Regresyon (Simple Linear Regression)
- Çoklu Doğrusal Regresyon (Multiple Linear Regression)
- Model Değerlendirme ve Seçimi
- Ağırlıklı Regresyon (Weighted Regression)
- Faktör Değişkenleri (Factor Variables)
- Regresyon Denkleminin Yorumlanması
- Regresyon Tanıları (Diagnostics)
- Polinom ve Spline Regresyonu, GAM

Regresyon Analizi — Anahtar Terimler

- Yanıt Değişkeni (Y): Tahmin edilmek istenen bağımlı değişken (hedef, sonuç)
- Bağımsız Değişken (X): Tahminde kullanılan özellik/nitelik (predictor, feature)
- Uyumlanmış Değerler (Fitted Values): Modelin ürettiği tahmin değerleri $\hat{y}$
- Artıklar (Residuals): Gerçek değer ile tahmin arasındaki fark: $e_i = y_i - \hat{y}_i$
- En Küçük Kareler (Least Squares): Artıkların kareleri toplamını minimize eden yöntem
- RMSE (Root Mean Squared Error): Tahmin hatasının standart ölçüsü
- R^2 (Belirlilik Katsayısı): Modelin açıkladığı varyans oranı

BÖLÜM I

Basit Doğrusal Regresyon

Regresyon Denklemi

- Bir bağımsız değişken (X) ile bir bağımlı değişken (Y) arasındaki doğrusal ilişkiyi modeller
- b_o (kesişim): X = 0 olduğunda Y'nin beklenen değeri
- b_1 (eğim): X'deki bir birimlik artışın Y üzerindeki etkisi

$$Y = b_o + b_1X + e$$

- En Küçük Kareler (OLS): $\Sigma e_i^2 = \Sigma(y_i - \hat{y}_i)^2$ toplamını minimize eder
- Örnek Veri Seti: LungDisease.csv — Maruz kalma (Exposure) vs. PEFR

Python ile Basit Doğrusal Regresyon

```
from sklearn.linear_model import LinearRegression
import pandas as pd

lung = pd.read_csv('LungDisease.csv')
model = LinearRegression()
model.fit(lung[['Exposure']], lung['PEFR'])

print(f'Kesisim (b0): {model.intercept_:.3f}')
print(f'Egim (b1): {model.coef_[0]:.3f}')

# Tahmini degerler ve artiklar
fitted = model.predict(lung[['Exposure']])
residuals = lung['PEFR'] - fitted
```

Çıktı: $b_0 \approx 424.58$, $b_1 \approx -4.19$ → Maruz kalma arttıkça PEFR azalır

Uyumlanmış Değerler ve Artıklar

- Uyumlanmış değer ($\hat{y}_i$): Her gözlem için modelin ürettiği tahmin
- Artık ($e_i = y_i - \hat{y}_i$): Gözlenen değer ile tahmin arasındaki fark
- Artıklar, modelin yakalayamadığı bilgiyi temsil eder
- İyi bir modelde artıklar rastgele dağılmalı, sistematik bir örüntü göstermemelidir
- Tahmin vs. Açıklama: Regresyon hem tahmin hem de değişkenler arası ilişkiyi anlamak için kullanılabilir

BÖLÜM II

Çoklu Doğrusal Regresyon

Çoklu Regresyon Denklemi

$$Y = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p + e$$

- Birden fazla bağımsız değişkenle tahmin yapma
- Her katsayı (b_j), diğer değişkenler sabit tutulduğunda X_j'nin Y üzerindeki marjinal etkisini gösterir
- Örnek: King County Konut Satış Verisi (house_sales.csv)
- Bağımlı değişken: AdjSalePrice (düzeltilmiş satış fiyatı)
- Bağımsız değişkenler: SqFtTotLiving, SqFtLot, Bathrooms, Bedrooms, BldgGrade

King County Konut Verisi ile Çoklu Regresyon

```
house = pd.read_csv('house_sales.csv', sep='\t')

predictors = ['SqFtTotLiving', 'SqFtLot', 'Bathrooms',
              'Bedrooms', 'BldgGrade']
outcome = 'AdjSalePrice'

house_lm = LinearRegression()
house_lm.fit(house[predictors], house[outcome])

print(f'Kesisim: {house_lm.intercept_:.3f}')
for name, coef in zip(predictors, house_lm.coef_):
    print(f'  {name}: {coef:.3f}')
```

Her 1 ft² yaşam alanı artışı → satış fiyatında ~$228 artış

Model Değerlendirme: RMSE ve R^2

- RMSE (Root Mean Squared Error): Tahminlerin ortalama hata büyüklüğü
- R^2 (Belirlilik Katsayısı): Modelin varyansın ne kadarını açıkladığı (0–1 arası)
- Düzeltilmiş R^2 (Adjusted R^2): Değişken sayısı için cezalandırılmış R^2

```
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

fitted = house_lm.predict(house[predictors])
RMSE = np.sqrt(mean_squared_error(house[outcome], fitted))
r2 = r2_score(house[outcome], fitted)
print(f'RMSE: {RMSE:.0f}')   # ~261,000
print(f'R2: {r2:.4f}')       # ~0.54
```

statsmodels ile Detaylı Analiz

```
import statsmodels.api as sm

model = sm.OLS(house[outcome],
               house[predictors].assign(const=1))
results = model.fit()
print(results.summary())

# Cikti:
# - Her katsayinin t-istatistigi ve p-degeri
# - F-istatistigi (model butun olarak anlamlilik)
# - AIC / BIC bilgi kriterleri
# - R-squared ve Adj. R-squared
```

statsmodels, scikit-learn'e göre çok daha detaylı istatistiksel çıktı sağlar

BÖLÜM III

Model Seçimi ve Ağırlıklı Regresyon

Adımsal Regresyon (Stepwise Regression)

- İleri Seçim (Forward Selection): Boş modelden başla, en iyi değişkeni ekle
- Geriye Eleme (Backward Elimination): Tam modelden başla, en az katkılıyı çıkar
- AIC (Akaike Information Criterion): Model karmaşıklığını cezalandıran kriter
- Daha düşük AIC → daha iyi model (doğruluk ve sadelik dengesi)

```
from dmba import stepwise_selection, AIC_score

def train_model(variables):
    model = LinearRegression()
    model.fit(X[variables], y)
    return model

def score_model(model, variables):
    return AIC_score(y, model.predict(X[variables]), model)

best_model, best_vars = stepwise_selection(
    X.columns, train_model, score_model)
```

Ağırlıklı Regresyon (Weighted Regression)

- Bazı gözlemlere daha fazla ağırlık verilerek model kurulur
- Örnek: Daha güncel satışlara daha yüksek ağırlık
- Kullanım alanları: Heteroskedastisite düzeltme, zaman ağırlıklandırma

```
# Yila gore agirlik: yeni veriler daha agir
house['Year'] = house.DocumentDate.apply(
    lambda d: int(d.split('-')[0]))
house['Weight'] = house.Year - 2005

house_wt = LinearRegression()
house_wt.fit(house[predictors], house[outcome],
             sample_weight=house.Weight)
```

BÖLÜM IV

Faktör Değişkenleri

Kukla Değişken Kodlaması (Dummy Variables)

- Kategorik değişkenler doğrudan regresyona giremez → sayısal kodlama gerekir
- P seviyeli kategorik değişken → P−1 kukla (binary) değişken
- Referans seviye: Kukla değişken oluşturulmayan kategori (drop_first=True)
- Her kukla değişkenin katsayısı, o kategorinin referansa göre farkını gösterir

```
# PropertyType: Multiplex, Single Family, Townhouse
print(pd.get_dummies(house['PropertyType'],
                     drop_first=True, dtype=int).head())

# Cikti: Single Family ve Townhouse icin 0/1 sutunlar
# Referans kategori: Multiplex
```

Çok Seviyeli Faktör Değişkenleri

- ZipCode gibi çok seviyeli değişkenler → çok fazla kukla değişken
- Çözüm: Seviyeleri gruplandırma (binning)
- Kitaptaki yaklaşım: Artıkların medyanına göre posta kodlarını 5 gruba ayırma
- Bu yaklaşım, konum bilgisini az sayıda değişkenle yakalar

```
# Posta kodlarini artiklar uzerinden grupla
zip_groups['ZipGroup'] = pd.qcut(
    zip_groups['cum_count'], 5,
    labels=False, retbins=False)

# Modele ZipGroup olarak ekle
house = house.join(
    zip_groups[['ZipCode','ZipGroup']].set_index('ZipCode'),
    on='ZipCode')
```

BÖLÜM V

Regresyon Denkleminin Yorumlanması

İlişkili Tahmin Değişkenleri ve Çoklu Doğrusallık

- İlişkili Tahmin Değişkenleri: Bağımsız değişkenler arası korelasyon, katsayıları değiştirir
- Çoklu Doğrusallık (Multicollinearity): Tahmin değişkenleri birbirine çok bağımlı olduğunda katsayılar kararsızlaşır
- Belirti: Bir değişken eklendiğinde/çıkarıldığında diğer katsayılar çok değişir
- Örnek: SqFtTotLiving modele girdiğinde Bedrooms katsayısı negatife dönebilir

Karıştırıcı Değişkenler ve Etkileşimler

- Karıştırıcı Değişken (Confounding Variable): Hem X hem Y ile ilişkili olup gerçek etkiyi maskeleyen değişken
- Örnek: Konum (ZipGroup) → hem fiyatı hem ev özelliklerini etkiler
- Etkileşim (Interaction): Bir değişkenin etkisi diğerine bağlıdır
- Örnek: SqFtTotLiving × ZipGroup etkileşimi → büyük evin getirisi konuma göre değişir

```
import statsmodels.formula.api as smf

# Etkilesim terimi: SqFtTotLiving * ZipGroup
model = smf.ols(
    formula='AdjSalePrice ~ SqFtTotLiving*ZipGroup + '
            'SqFtLot + Bathrooms + Bedrooms + BldgGrade + '
            'PropertyType',
    data=house)
results = model.fit()
```

BÖLÜM VI

Regresyon Tanıları (Diagnostics)

Aykırı Değerler ve Etkili Gözlemler

- Aykırı Değer (Outlier): Modelin beklentisinden çok uzak gözlem
- Standardize Artıklar: $|e_i^*| > 2.5$ ise potansiyel aykırı değer
- Kaldıraç (Leverage / Hat Value): Bir gözlemin X-uzayındaki uçsallığı — Eşik: $2(p+1)/n$
- Cook's Distance: Bir gözlemin hem aykırılık hem kaldıraç açısından model üzerindeki etkisi — Eşik: $4/(n-p-1)$
- Etki Grafiği: Hat values vs. studentized residuals, nokta boyutu = Cook's distance

Aykırı Değer Analizi — Python

```
from statsmodels.stats.outliers_influence import OLSInfluence

# ZipCode 98105 alt kumesi uzerinde model
house_98105 = house.loc[house['ZipCode'] == 98105]
house_outlier = sm.OLS(house_98105[outcome],
                       house_98105[predictors].assign(const=1))
result_98105 = house_outlier.fit()

# Etkili gozlem analizi
influence = OLSInfluence(result_98105)
sresiduals = influence.resid_studentized_internal

# Cook's distance ile filtreleme
mask = [d < .08 for d in influence.cooks_distance[0]]
house_infl = house_98105.loc[mask]
```

Heteroskedastisite ve Kısmi Artık Grafikleri

- Heteroskedastisite: Artıkların varyansının tahmin değerine göre değişmesi
- Tanı: Tahmin vs. |artık| grafiğinde sistematik örüntü aranır
- Normal Dağılım: Artıkların histogramı simetrik ve çan şekline yakın olmalı
- Kısmi Artık Grafikleri (Partial Residual Plots): Her bağımsız değişkenin etkisini ayrı ayrı inceler
- Doğrusal olmayan ilişki varsa → polinom veya spline regresyon düşünülmeli

BÖLÜM VII

Polinom, Spline Regresyon ve GAM

Polinom Regresyon

- Doğrusal olmayan ilişkiler için X'in üst kuvvetlerini modele ekleme
- Quadratik (2. derece): $Y = b_0 + b_1X + b_2X^2$
- Kitap örneği: $SqFtTotLiving^2$ terimi eklenerek eğrisellik yakalanır
- Dikkat: Yüksek dereceli polinomlar aşırı uyum (overfitting) riski taşır

```
model_poly = smf.ols(
    formula='AdjSalePrice ~ SqFtTotLiving + '
            'np.power(SqFtTotLiving, 2) + '
            'SqFtLot + Bathrooms + Bedrooms + BldgGrade',
    data=house_98105)
result_poly = model_poly.fit()
print(result_poly.summary())
```

Spline Regresyon

- Spline: Veri aralığını parçalara bölerek her parçaya ayrı polinom uydurma
- Düğüm noktalarında (knots) eğriler sürekli ve türevlenebilir biçimde birleşir
- B-splines: En yaygın spline tabanı, df (serbestlik derecesi) ile esneklik kontrol edilir
- Polinom regresyona göre avantaj: Uçlarda aşırı salınım yapmaz

```
formula = ('AdjSalePrice ~ '
           'bs(SqFtTotLiving, df=6, degree=3) + '
           'SqFtLot + Bathrooms + Bedrooms + BldgGrade')
model_spline = smf.ols(formula=formula,
                       data=house_98105)
result_spline = model_spline.fit()
```

GAM — Genelleştirilmiş Eklemeli Modeller

- GAM: Her bağımsız değişken için ayrı düzgünleştirme fonksiyonu (smoothing spline)
- $Y = s(X_1) + f(X_2) + ...$ → Her değişken doğrusal veya eğrisel olabilir
- pygam kütüphanesi: s() spline terimi, l() doğrusal terim
- Kısmi bağımlılık grafikleri → her değişkenin etkisini görselleştirme
- Avantaj: Doğrusal olmayan ilişkileri otomatik yakalar, yorumlanabilirlik korunur

```
from pygam import LinearGAM, s, l

gam = LinearGAM(s(0, n_splines=12) + l(1) + l(2) + l(3) + l(4))
gam.gridsearch(X, y)
print(gam.summary())
# Kismi bagimlilik grafikleri ile her degiskenin
# etkisini ayri ayri gorsellestirin
```

ÖZET

Bu Hafta Öğrendiklerimiz

- Basit ve çoklu doğrusal regresyon ile tahmin modelleri kurma
- Model değerlendirme: RMSE, R^2, düzeltilmiş R^2, AIC
- Adımsal regresyon ile otomatik değişken seçimi
- Kategorik değişkenlerin kukla kodlamayla modele dahil edilmesi
- Katsayı yorumlama: ilişkili değişkenler, karıştırıcılar, etkileşimler
- Regresyon tanıları: aykırı değerler, etkili gözlemler, heteroskedastisite
- Doğrusal olmayan ilişkiler: polinom, spline regresyon ve GAM

Kaynaklar ve Önerilen Okumalar

- Bruce, P., Bruce, A., Gedeck, P. (2020). Practical Statistics for Data Scientists, 2. Baskı, O'Reilly — Bölüm 4
- GitHub: github.com/gedeck/practical-statistics-for-data-scientists
- scikit-learn Dokümantasyonu: scikit-learn.org/stable/modules/linear_model.html
- statsmodels Dokümantasyonu: statsmodels.org/stable/regression.html
- Veri Seti: King County House Sales (house_sales.csv)

Gelecek Hafta: Sınıflandırma

- Lojistik Regresyon
- Sınıflandırma metrikleri (doğruluk, kesinlik, duyarlılık, F1)
- ROC eğrisi ve AUC
- Naive Bayes ve K-En Yakın Komşu (KNN)

Sorularınız?