ANALIZA VELIKIH PODATAKA

školska 2024/2025 godina

Vežba 8: Linearna regresija i analiza zavisnosti između varijabli

Linearna regresija predstavlja osnovni model prediktivne analize koji nam omogućuje da razumemo **odnos između jedne zavisne promenljive** i **jedne ili više nezavisnih promenljivih**. U najjednostavnijem slučaju, koristimo je za predikciju numeričke vrednosti, npr. kolika će biti cena proizvoda ako znamo njegovu veličinu, ili koliko će vremena trebati korisniku da izvrši neku akciju u aplikaciji ako znamo broj koraka koje je prethodno preduzeo.

U praksi, linearna regresija se koristi kada želimo da testiramo hipotezu o **postojanju linearnog trenda** između promenljivih, kao i da kvantifikujemo snagu tog odnosa. Osim toga, regresija služi i kao **osnova za mnoge kompleksnije modele**, pa je razumevanje njenih temelja ključno za sve koji se bave analizom podataka ili mašinskim učenjem.

Linearnu regresiju je uveo Francis Galton krajem 19. veka u kontekstu **nasledne visine kod ljudi**, i otuda potiče termin „regresija“ (nazadovanje ka srednjoj vrednosti). Danas se koristi u mnogim oblastima:

- **Ekonomija** – predviđanje plata, BDP-a, potrošnje.
- **Zdravstvo** – uticaj terapije na zdravlje.
- **Marketing** – efekat budžeta na prodaju.
- **Inženjering** – uticaj temperature na stabilnost materijala

Dodatna prednost linearne regresije je njena **interpretabilnost** – lako možemo interpretirati svaki koeficijent kao promenu u zavisnoj promenljivoj uz jednu jediničnu promenu nezavisne promenljive (uz pretpostavku da su ostale konstantne). Ovo je veoma korisno u domenima gde je važno objasniti uticaj faktora, a ne samo dobiti tačnu predikciju.

Zbog jednostavnosti i široke primene, linearna regresija je često prvi model koji se koristi prilikom analize podataka, kako bi se stekao osnovni uvid u odnose među varijablama

Matematička osnova

Linearna regresija se zasniva na ideji da postoji **linearna funkcija** koja može da objasni (ili predvidi) vrednosti zavisne promenljive. Drugim rečima, možemo pretpostaviti da postoji **linearan odnos** između varijabli: ako se jedna menja, i druga se menja proporcionalno.

📐 Jednostavna linearna regresija

Kod jednostavne regresije imamo **samo jednu nezavisnu promenljivu (x)** i želimo da predvidimo jednu zavisnu promenljivu (y). Matematika iza modela izgleda ovako:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Ovde:

- β_0 predstavlja **presretanje** prave sa y-ose (intercept).
- β_1 je **koeficijent nagiba**, tj. koliki je uticaj promenljive x na vrednost y.
- ε je **greška modela** (nepredviđeni deo y).

Cilj je da se pronađu takvi β_0 i β_1 koji će **minimizovati ukupnu grešku predikcije**. Najčešće se koristi **metod najmanjih kvadrata (OLS)**.

📊 Višestruka linearna regresija

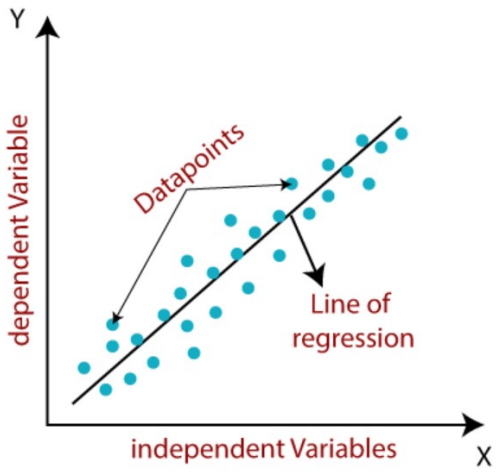
Ako imamo više ulaznih varijabli (npr. kvadratura, broj soba, sprat), onda govorimo o **višestrukoj (multiple) linearnoj regresiji**:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

Ovaj pristup omogućava složeniju analizu i modelovanje realističnijih problema, gde zavisna varijabla zavisi od kombinacije faktora.

📏 Geometrijska interpretacija

U jednostavnoj regresiji, rešenje je prava linija koja najbolje „leži“ u prostoru tačaka na 2D grafikonu. U višestrukoj regresiji, rešenje je hiperravnina u višedimenzionalnom prostoru, koja pokušava da minimizira ukupnu udaljenost (grešku) između predikcija i stvarnih tačaka.



Pretpostavke linearne regresije

Pre nego što primenimo linearnu regresiju na neki skup podataka, važno je proveriti da li su zadovoljene osnovne **statističke pretpostavke** koje ovaj model podrazumeva. Linearna regresija funkcioniše optimalno samo kada su ove pretpostavke ispunjene. U suprotnom, koeficijenti modela mogu biti pristrasni, predikcije nepouzdane, a tumačenje rezultata pogrešno.

1. Linearni odnos između nezavisnih i zavisne promenljive

Model pretpostavlja da postoji **linearan odnos** između svake nezavisne promenljive X i zavisne promenljive y. Ovo znači da promena u X uzrokuje proporcionalnu promenu u y.

🔍 *Kako proveriti?*
Vizualizacijom pomoću scatter plot-a (tačkastog dijagrama) ili korišćenjem korelacionih koeficijenata. Ako odnos nije linearan, može se razmotriti transformacija podataka (npr. logaritamska, kvadratna).

2. Normalna raspodela reziduala (grešaka)

Razlika između stvarnih i predikovanih vrednosti, tj. **reziduali**, treba da budu približno **normalno distribuirani**. Ovo je naročito važno kada želimo da primenjujemo statističke testove značajnosti regresionih koeficijenata.

🔍 *Kako proveriti?*
Korišćenjem histograma, Q-Q plot-a (kvantil-kvantil dijagrama), ili Shapiro-Wilk testa. Ako reziduali nisu normalno distribuirani, to može ukazivati na nedostatke u modelu.

3. Konstantna varijansa grešaka

Greške (reziduali) treba da imaju približno **istu varijansu** za sve vrednosti nezavisnih promenljivih. Ako varijansa grešaka raste ili opada sa promenom u X, govorimo o **heteroskedastičnosti**, što može učiniti procene koeficijenata nepouzdanim.

🔍 *Kako proveriti?*
Grafikom raspršenja reziduala naspram predikcija. Ako se vidi obrazac "levka" (sužava se ili širi), to je znak heteroskedastičnosti. U tom slučaju se može koristiti transformacija ili robustna regresija.

4. Nezavisnost reziduala

Reziduali treba da budu **nezavisni** jedni od drugih. Ovo je posebno važno kod vremenskih serija i panel podataka. Ako su greške korelisane, postoji **autokorelacija**, što znači da prethodne greške utiču da model više nije validan u klasičnoj formi.

🔍 *Kako proveriti?*
Durbin-Watson test je standardan način za proveru autokorelacije, dok vizuelno možemo koristiti grafik reziduala kroz vreme.

5. Odsustvo multikolinearnosti

U višestrukoj regresiji, nezavisne promenljive ne bi smele biti **međusobno visoko korelisane**. Ako jesu, teško je izolovati njihov individualni uticaj na zavisnu promenljivu, jer se one ponašaju kao da "prenose slične informacije".

🔍 *Kako proveriti?*
Korišćenjem **VIF (Variance Inflation Factor)**. Ako je VIF za neku promenljivu veći od 5 (ili 10), postoji značajna multikolinearnost i ta promenljiva bi mogla biti isključena ili transformisana.

🚨 **Zašto su ove pretpostavke važne?**

Kršenje ovih pretpostavki ne mora nužno da znači da model neće raditi, ali rezultati mogu postati **nepouzdani**:

- Procene koeficijenata mogu biti pristrasne ili nestabilne.
- Statistički zaključci (npr. da li je neki prediktor značajan) mogu biti pogrešni.
- Predikcije mogu imati visoku grešku u realnim uslovima.

U svakodnevnoj praksi, podaci često **ne zadovoljavaju sve pretpostavke savršeno**, ali razumevanje stepena odstupanja pomaže u izboru alternativnih metoda.

Izgradnja modela korak po korak (Python)

📁 Učitavanje biblioteka i podataka

Koristićemo **pandas**, **matplotlib**, **seaborn**, i **scikit-learn**.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

Primer dataset-a: predikcija **cene stana** na osnovu kvadrature.

```
# Generisanje jednostavnog skupa podataka
data = {
    'Kvadratura': [30, 40, 50, 60, 70, 80, 90],
    'Cena': [50000, 65000, 70000, 85000, 95000, 110000, 125000]
}
df = pd.DataFrame(data)
```

📉 **Vizualizacija podataka**

```
sns.scatterplot(x='Kvadratura', y='Cena', data=df)
plt.title("Zavisnost cene od kvadrature")
plt.show()
```

🔧 **Treniranje modela**

```
X = df[['Kvadratura']]
y = df['Cena']

model = LinearRegression()
model.fit(X, y)

# Ispis koeficijenata
print("Koeficijent (β1):", model.coef_[0])
print("Intercept (β0):", model.intercept_)
```

Model sada ima oblik:

$$\hat{y} = \beta_0 + \beta_1 x$$

📊 Predikcija i evaluacija

```
y_pred = model.predict(X)
```

```
plt.scatter(X, y, color='blue', label='Stvarne vrednosti')
plt.plot(X, y_pred, color='red', label='Predikcija')
plt.legend()
plt.title("Regresiona linija")
plt.show()
```

🔪 **Evaluacija modela**

📌 Mean Absolute Error (MAE)

Srednja apsolutna greška:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

```
mae = mean_absolute_error(y, y_pred)
print("MAE:", mae)
```

📌 Mean Squared Error (MSE)

Srednja kvadratna greška:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
mse = mean_squared_error(y, y_pred)
print("MSE:", mse)
```

📌 R-squared (R^2)

Koeficijent determinacije, meri **procenat varijanse y** koju model objašnjava.

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

SS_{res} – zbir kvadrata reziduala (rezidualna suma)
SS_{tot} – ukupna suma kvadrata (varijansa stvarnih vrednosti)

```
r2 = r2_score(y, y_pred)
print("R²:", r2)
```

$R^2 = 1$ → savršeno, $R^2 \approx 0$ → loš model

🔎 Tumačenje koeficijenata

U jednostavnoj regresiji, koeficijent β_1 označava **koliko se y menja kada se x poveća za 1 jedinicu**.

Ako je β_1=1500, to znači da se cena stana uvećava za 1500€ po svakom dodatnom kvadratu.

U situacijama kada su greške velike ili sistematski raspoređene, može biti korisno razmotriti nelinearne modele, transformacije podataka ili dodatne promenljive koje bolje objašnjavaju zavisnost.

U **višestrukoj linearnoj regresiji**, svaki regresioni koeficijent β_i predstavlja **marginalni efekat** odgovarajuće promenljive x_i, dok su **sve ostale promenljive fiksirane**. To znači da koeficijent pokazuje koliko se očekuje da će zavisna promenljiva y da se promeni kada se x_i poveća za jednu jedinicu, dok sve ostale promenljive ostaju nepromenjene.

🔍 **Primer višestruke regresije**

```
# Višestruka regresija – primer sa veštačkim podacima
data = {
    'Kvadratura': [30, 40, 50, 60, 70],
    'Sprat': [1, 2, 3, 4, 5],
    'Cena': [55000, 65000, 75000, 85000, 95000]
}
df = pd.DataFrame(data)

X = df[['Kvadratura', 'Sprat']]
y = df['Cena']

model = LinearRegression()
model.fit(X, y)

print("Koeficijenti:", model.coef_)
print("Intercept:", model.intercept_)
```

Model sada ima oblik:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

U ovom primeru vidimo kako se višestruka linearna regresija primenjuje na skup podataka sa dve ulazne promenljive – kvadraturom i spratom stana. Model uči koeficijente koji kvantifikuju uticaj svake od ovih karakteristika na cenu stana, omogućavajući nam da predviđamo cene na osnovu kombinacije faktora.

Još jedan praktični primer – jednostavna linearna regresija

Koristićemo `scikit-learn` i `matplotlib` za analizu skupa podataka.

📦 Instalacija biblioteka (ako je potrebno):

```
pip install pandas scikit-learn matplotlib seaborn
```

🔽 Učitavanje primera:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Učitavanje primera skupa podataka
df = pd.read_csv('../input/health-insurance-dataset/Health_insurance.csv')
df.head()
```

🎯 Cilj: Predikcija charges na osnovu bmi

```
X = df[['bmi']]
y = df['charges']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

📊 Vizualizacija i tumačenje rezultata

```
plt.scatter(X_test, y_test, color='blue', label='Stvarne vrednosti')
plt.plot(X_test, y_pred, color='red', label='Regresiona linija')
plt.xlabel("BMI")
plt.ylabel("Charges")
plt.title("Linearna regresija: charges vs. bmi")
plt.legend()
plt.show()
```

📈 Metrike uspešnosti modela

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("MAE:", mae)
print("MSE:", mse)
print("R²:", r2)
```

Dodatno objašnjenje metrika:

Metod	Značenje
MAE (Mean Absolute Error)	Prosečna apsolutna greška. Lako za interpretaciju.
MSE (Mean Squared Error)	Kvadratna greška – kažnjava velike greške.
R^2 (R-squared)	Objašnjava koliko dobro se model uklapa. Vrednosti bliže 1 su bolje.

🔍 Tumačenje koeficijenata

```
print(f"Koeficijent (b1): {model.coef_[0]}")
print(f"Intercept (b0): {model.intercept_}")
```

Koeficijent kaže za koliko će se charges promeniti ako bmi poraste za 1 jedinicu (ako ostale promenljive ostanu iste).

📐 Višestruka linearna regresija

Dodajemo više promenljivih:

```
X_multi = df[['age', 'bmi', 'children']]
y = df['charges']

X_train, X_test, y_train, y_test = train_test_split(X_multi, y,
test_size=0.2, random_state=42)

model_multi = LinearRegression()
model_multi.fit(X_train, y_train)

y_pred_multi = model_multi.predict(X_test)
print("R² score (višestruka regresija):", r2_score(y_test, y_pred_multi))
```

🔍 Provera pretpostavki vizualno (reziduali)

Za pravilno tumačenje i poverenje u rezultate regresije, važno je proveriti da li su osnovne pretpostavke zadovoljene:

```
# Računanje reziduala (grešaka)
residuals = y_test - y_pred

# Distribucija reziduala
sns.histplot(residuals, kde=True)
plt.title("Distribucija reziduala")
plt.xlabel("Reziduali")
plt.show()

# Reziduali vs. predikcije
plt.scatter(y_pred, residuals)
plt.axhline(0, color='red', linestyle='--')
plt.title("Predikcije vs. reziduali")
plt.xlabel("Predikcija")
plt.ylabel("Rezidual")
plt.show()
```

Simetrična i normalna raspodela reziduala ukazuje da je greška slučajna, što je poželjno. **Nasumičan raspored tačaka oko nule** u scatter dijagramu (predikcije vs. reziduali) sugeriše da je linearni model dobar fit. Ako postoji obrazac (npr. zakrivljenost), to ukazuje na nelinearnost koju model nije uhvatio.

⚠️ Problemi i ograničenja linearne regresije

1. **Osetljivost na outliere** – ekstremne vrednosti mogu značajno promeniti model i uticati na tačnost predikcija.
2. **Ne može modelovati nelinearne odnose bez transformacija** – ako zavisnost između promenljivih nije linearna, model neće biti precizan.
3. **Multikolinearnost** – kada se dve ili više promenljivih ponašaju slično, teško je odrediti njihov pojedinačan uticaj na zavisnu promenljivu.
4. **Pretreniranje (overfitting)** – model može naučiti šum iz podataka umesto prave veze, što smanjuje tačnost na novim podacima.
5. **Pretpostavke modela mogu biti narušene** – linearna regresija pretpostavlja normalnu raspodelu reziduala, homoskedastičnost i nezavisnost posmatranja, što često nije slučaj u stvarnim podacima.

Razumevanje ovih ograničenja pomaže u pravilnoj interpretaciji rezultata i u donošenju odluke da li je linearna regresija odgovarajući alat. Kada pretpostavke nisu zadovoljene, korisno je razmotriti alternativne modele, kao što su polinomska regresija, regresija sa regularizacijom (Ridge, Lasso) ili nelinearne metode poput Random Forest-a.