ANALIZA VELIKIH PODATAKA

školska 2024/2025 godina

Vežba 10: Višestruka regresija i značaj osobina

🔢 Višestruka linearna regresija

Višestruka linearna regresija (engl. *Multiple Linear Regression*) je model koji se koristi za predviđanje vrednosti jedne **zavisne promenljive** na osnovu više **nezavisnih promenljivih**. Matematički, model se izražava kao:

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

Gde su:

- y: zavisna promenljiva (target)
- xi: nezavisne promenljive (features)
- βi: regresioni koeficijenti
- ε: greška modela (engl. *residual*)

Cilj je proceniti koeficijente βi\beta_iβi tako da model što bolje predvidi ciljnu promenljivu.

📌 Pretpostavke linearnog modela

Da bi višestruka regresija bila validna, potrebno je da budu zadovoljene sledeće pretpostavke:

- **Linearnost**: Odnos između zavisne i nezavisnih promenljivih treba da bude linearan.
- **Independencija grešaka**: Reziduali treba da budu nezavisni jedni od drugih.
- **Homoskedastičnost**: Varijansa grešaka treba da bude konstantna kroz sve vrednosti prediktora.
- **Normalnost grešaka**: Reziduali bi trebalo da budu normalno distribuirani.

🧠 Značaj osobina (Feature Importance)

U kontekstu linearnog modela, značaj osobina se izražava kroz vrednosti regresionih koeficijenata. Veći apsolutni iznos koeficijenta ukazuje na veći uticaj te promenljive na izlaz. Međutim, ako promenljive imaju različite skale, direktna interpretacija koeficijenata može biti obmanjujuća. Zato se pre interpretacije najčešće primenjuje standardizacija (z-score).

Postoje i druge metode za merenje značaja osobina, poput:

- **Dominaciona analiza:** Ova metoda upoređuje doprinos svake osobine u različitim podskupovima modela i daje rangiranje osobina prema njihovom ukupnom značaju.
- **Permutacione važne osobine** (permutation importance): Procena značaja se vrši tako što se slučajno permutuje jedna osobina, a zatim meri pad performansi modela — što je veći pad, osobina je značajnija.
- **SHAP vrednosti:** SHAP (SHapley Additive exPlanations) vrednosti kvantifikuju doprinos svake osobine u konkretnim predikcijama i koriste se za interpretaciju kompleksnih modela poput ansambala i neuronskih mreža.

🔧 One-hot enkodiranje

One-hot enkodiranje je tehnika za konvertovanje kategorijalnih promenljivih u numerički format koji je pogodan za modele mašinskog učenja. Svaka kategorija dobija svoju binarnu kolonu. Na primer, promenljiva "Pol" sa vrednostima "muški" i "ženski" se transformiše u dve kolone: Pol_muški i Pol_ženski, koje sadrže vrednosti 0 ili 1.
Ova metoda sprečava model da pogrešno zaključi da postoji redosled ili odnos između različitih kategorija, što bi moglo da se desi kod običnog numeričkog kodiranja.

📏 Skaliranje podataka

Skaliranje se koristi kako bi se sve numeričke promenljive dovele na istu skalu. Ovo je posebno važno za modele koji koriste metričku udaljenost ili, kao u ovom slučaju, linearnu regresiju, gde različite skale mogu uticati na vrednosti koeficijenata. Najčešće korišćeno skaliranje je standardizacija, koja svaku promenljivu transformiše da ima srednju vrednost 0 i standardnu devijaciju 1.
Skaliranje takođe može pomoći u stabilizaciji i bržoj konvergenciji algoritama prilikom treniranja modela, što je naročito korisno kod modela sa velikim brojem osobina.

Napomena: One-hot enkodirane promenljive se obično **ne skaliraju**, jer su već binarne.

🧪 Praktični primer u Pythonu

U sledećem primeru koristićemo **Boston Housing** dataset da bismo predvideli prosečnu cenu kuća (medv) na osnovu više numeričkih i kategorijalnih karakteristika:

```
# Učitavanje biblioteka

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.metrics import mean_squared_error

import matplotlib.pyplot as plt

import seaborn as sns


# Učitavanje podataka

data = pd.read_csv("boston-housing.csv")


# Definisanje osobina i ciljne promenljive

X = data.drop("medv", axis=1)

y = data["medv"]


# Identifikacija kategorijalnih i numeričkih kolona

categorical_features = ['chas']

numerical_features = [col for col in X.columns if col not in

                      categorical_features]
```

```
# Kreiranje preprocesora

preprocessor = ColumnTransformer([

    ('num', StandardScaler(), numerical_features),

    ('cat', OneHotEncoder(), categorical_features)

])


# Pipeline: preprocesiranje + model

model = Pipeline([

    ('preprocessor', preprocessor),

    ('regressor', LinearRegression())

])


# Podela skupa

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


# Treniranje

model.fit(X_train, y_train)


# Predikcija i evaluacija

predictions = model.predict(X_test)

print("MSE:", mean_squared_error(y_test, predictions))


# Analiza značaja osobina

coefs = model.named_steps['regressor'].coef_


feature_names = numerical_features +
list(model.named_steps['preprocessor'].transformers_[1][1].get_feature_names_
out())
```

```
importance_df = pd.DataFrame({"Feature": feature_names, "Importance": coefs})

importance_df = importance_df.sort_values(by="Importance", key=abs,
ascending=False)



# Vizualizacija

sns.barplot(x="Importance", y="Feature", data=importance_df)

plt.title("Značaj osobina u višestrukoj regresiji")

plt.tight_layout()

plt.show()
```