

Korištenje Python paketa za statističku analizu i vizualizaciju podataka, te analizu rezultata algoritama strojnog učenja i dubinske analize podataka

Matija Piškorec

Institut Ruđer Bošković

10. lipanj, 2026.

Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

Jasnoća

Vizualizacija rezultata strojnog učenja

Zaključak

Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

Jasnoća

Vizualizacija rezultata strojnog učenja

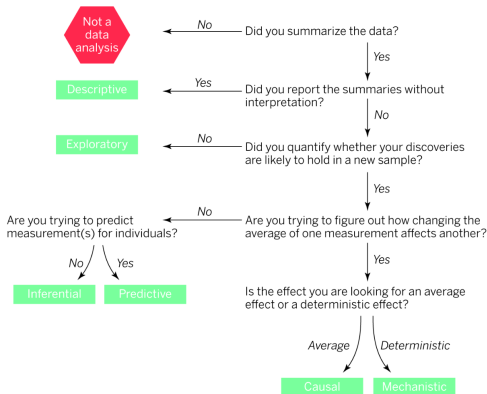
Zaključak

- Python Data Science Handbook - poglavlje 4 i relevantni dijelovi poglavlja 5
<https://jakevdp.github.io/PythonDataScienceHandbook/>
- Fundamentals of Data Visualization by Claus O. Wilke
<https://clauswilke.com/dataviz/>
- Think Python: How to Think Like a Computer Scientist by Allen B. Downey
<https://greenteapress.com/wp/>
<https://alldowney.github.io/ThinkPython/>
- Python for Data Analysis by Wes McKinney
<https://www.lkhibra.ma/books/Python-for-Data-Analysis.pdf>
- PMF Strojno učenje 2023 - vježbe
<https://github.com/pmf-strojnoucenje/Vjezbe>

Analiza podataka - dijagram

U ovom predavanju ćemo se baviti **deskriptivnom** i **eksploratornom** analizom podataka koristeći vizualizaciju u Pythonu.

Data analysis flowchart



Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

Jasnoća

Vizualizacija rezultata strojnog učenja

Zaključak

Zašto Python za vizualizaciju podataka?



- **Matplotlib** - temeljna biblioteka za izradu grafova; sve se gradi oko objekata *figure* (slika) i *axes* (koordinatni sustav).
- **Seaborn** - sloj iznad Matplotliba s gotovim statističkim grafovima i skupovima podataka.
- **pandas** - učitavanje i obrada podataka u tablicama (*DataFrame*).

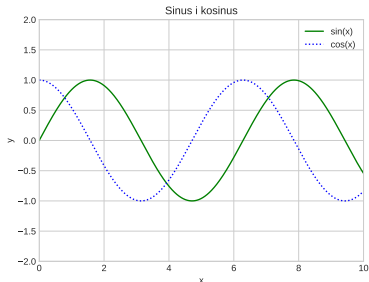
Tipičan uvodni predložak za sve primjere koji slijede:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

Linijski graf: ax.plot

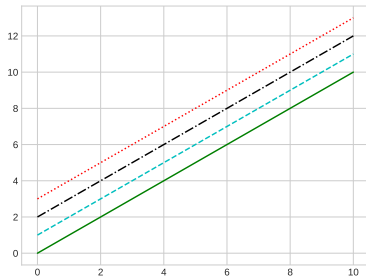
```
1 x = np.linspace(0, 10, 1000)
2
3 fig, ax = plt.subplots()
4 ax.plot(x, np.sin(x), label="sin(x)")
5 ax.plot(x, np.cos(x), label="cos(x)")
6 ax.set(xlabel="x", ylabel="y")
7 ax.legend()
8 plt.show()
```

Linijski graf (line graph) je jednostavan način za prikazati parove (x, y) numeričkih podataka u slučaju da *postoji* implicitna interpolacija podataka!



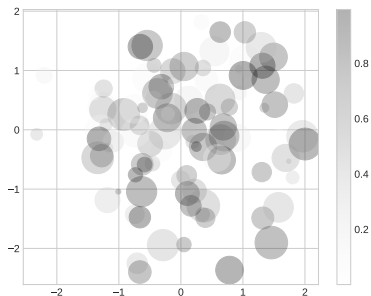
Boje i stilovi linija

```
1 # format "boja + stil linije"  
2 # puna zelena  
3 plt.plot(x, x + 0, '-g')  
4 # crtkana cijan  
5 plt.plot(x, x + 1, '--c')  
6 # crta-tocka crna  
7 plt.plot(x, x + 2, '-.k')  
8 # tockasta crvena  
9 plt.plot(x, x + 3, ':r')
```



Scatterplot: `ax.scatter`

```
1  rng = np.random.default_rng(0)
2  x = rng.normal(size=100)
3  y = rng.normal(size=100)
4
5  # kodiranje 5 varijabli:
6  # - pozicija (x,y)
7  # - boja i transparentnost
8  # - veličina kružića
9  plt.scatter(x, y, c=rng.random(100),
10             s=1000 * rng.random(100),
11             alpha=0.3)
12  plt.colorbar()
```



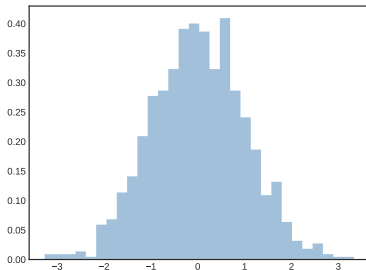
Scatterplot je jednostavan način za prikazati parove (x, y) numeričkih podataka u slučaju da *ne postoji* implicitna interpolacija podataka!

Histogram: ax.hist



```
1 data = np.random.default_rng(1701).  
   normal(size=1000)  
2  
3 plt.hist(data, bins=30, density=True,  
4          alpha=0.5,  
5          histtype="stepfilled",  
6          color="steelblue")  
7 plt.show()
```

Histogram je jednostavan i direktan način za prikazati empirijsku distribuciju podataka.



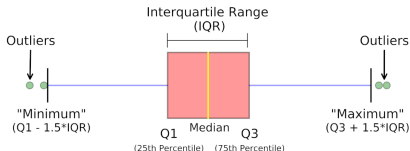
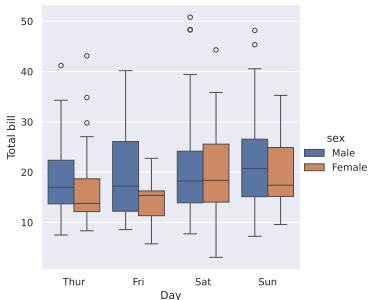
Boxplot (Seaborn paket): sns.boxplot

```

1 # dataset dostupan unutar Seaborna
2 tips = sns.load_dataset("tips")
3
4 sns.boxplot(x="day", y="total_bill",
5             hue="sex", data=tips)

```

Boxplot je jedan od najsžažetijih načina za prikazati distribuciju podataka preko nekoliko ključnih statističkih vrijednosti.



Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

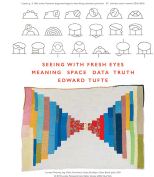
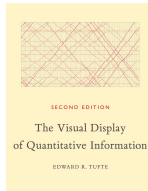
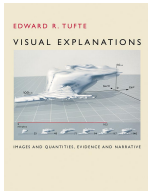
Jasnoća

Vizualizacija rezultata strojnog učenja

Zaključak

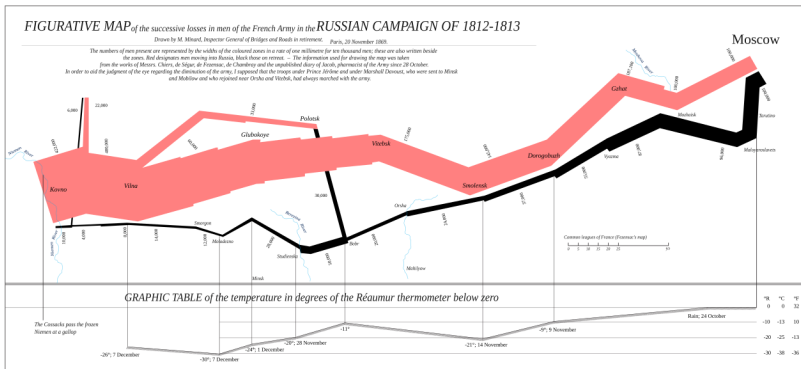
Edward Tufte

Edward Tufte - američki profesor na sveučilištu Yale i autor brojnih knjiga o vizualizaciji informacija.



Edward Tufte - primjer iz knjiga

Infografika Charlesa Josepha Minarda iz 1869. koja prikazuje gubitke Napoleonove vojske u ruskoj kampanji 1812. - 1813.



- **Integritet** - vizualizacija treba biti vjerodostojna i ne smije obmanjivati gledatelja.
- **Jasnoća** - vizualizacija treba biti jasna i lako razumljiva, bez nepotrebnih elemenata koji odvlače pažnju.

Ideja je da **estetika** proizlazi direktno iz navedena dva principa, i ne moramo se dodatno truditi je postizati!

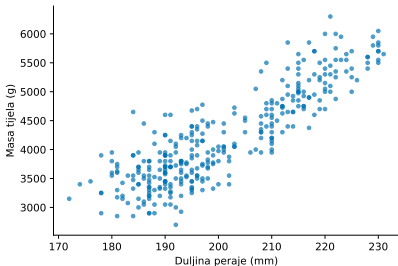
1. **“Prikaži podatke umjesto ukrasa (dekoracija)”** - Primarna svrha grafike je *otkriti* podatke, a ne ih *ukrašavati* (dekorirati).
2. **“Prikaži izvorne podatke umjesto njihovih sažetaka”** - Prikažite izvorne podatke pomoću vizualizacija visoke rezolucije umjesto sažetaka podataka.
3. **“Izbjegavaj iskrivljavanje podataka”** - Predstavi podatke vjerodostojno izbjegavajući obmanjujuće vizualizacije.
4. **“Prikaži multivarijantne podatke”** - Prikaži više varijabli istovremeno koristeći različite vizualne elemente.

“Prikaži podatke umjesto ukrasa (dekoracija)” - Primarna svrha grafike je *otkriti* podatke, a ne ih *ukrašavati* (dekorirati).

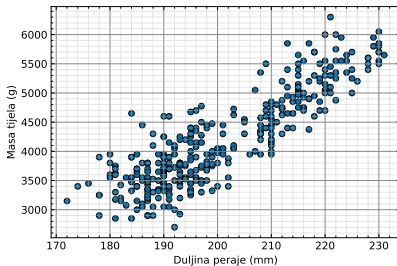
Kako to postići - Izbjegavaj nepotrebne vizualne elemente koji se natječu s podacima za pažnju gledatelja (debele linije, pretjerana boja ili stilizacija). Svaki vizualni element (bez obzira je li povezan s podacima ili dekoracijom) pridonosi kognitivnom opterećenju gledatelja.

Integritet 1/4 - primjer (Palmer Penguins)

Preporučeno - čisto



Nije preporučeno - pretrpano



```
1 df = sns.load_dataset("penguins").dropna()
2 x = df["flipper_length_mm"]
3 y = df["body_mass_g"]
4
5 fig, ax = plt.subplots()
6 ax.scatter(x, y, s=12, alpha=0.7)
7
8 # uklanjamo nepotrebne okvire
9 for s in ("top", "right"):
10     ax.spines[s].set_visible(False)
```

Isti podaci, manje dekoracije: uklonjeni okviri, mreža i suvišne oznake.

Integritet 1/4 - primjer (tablica)

Preporučeno - čista tablica

Role	Name	Year of the...	Debut	Thousands of Fans	Takedown Rate
Face (The Hero)	The Ultimate Warrior	Tiger	May-2011	97.3	86.2
	Hulk Hogan	Oxen	Jan-2008	988.6	62.0
	Macho Man Randy Savage	Monkey	Feb-2008	157.6	59.3
	Hacksaw Jim Duggan	Pig	Mar-2008	30.3	53.4
	Superfly Jimmy Snuka	Dragon	Mar-2008	12.3	52.7
Heel (The Bad Guy)	Rowdy Roddy Piper	Rooster	Jun-1968	71.6	45.4
	The Million Dollar Man Ted DiBiase	Rat	Apr-1975	449.3	43.8
	Mr. Perfect Curt Hennig	Rat	May-1980	13.8	38.0
	Jake the Snake Roberts	Snake	Jul-1975	5.6	38.0
Jobber (The Unknown)	Brad Smith	Sheep	Aug-2008	1.1	36.3
	Ted Duncan	Sheep	Aug-2008	0.2	33.6
	Joey the Uber Nerd Cherdarchuk	Snake	Aug-2008	0.0	21.0

Nije preporučeno - pretrpana tablica

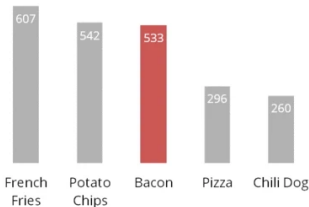
Role	Name	Year of the...	Debut	Number of Fans	Takedown Rate
Face (The Hero)	The Ultimate Warrior	Tiger	May-2011	97320.00	86.2
Face (The Hero)	Hulk Hogan	Oxen	Jan-2008	988551.00	61.978
Face (The Hero)	Macho Man Randy Savage	Monkey	Feb-2008	157618.00	59.29
Face (The Hero)	Hacksaw Jim Duggan	Pig	Mar-2008	30300.00	53.4332
Face (The Hero)	Superfly Jimmy Snuka	Dragon	Mar-2008	12341.00	52.7
Heel (The Bad Guy)	Rowdy Roddy Piper	Rooster	Jun-1968	71645.00	45.4
Heel (The Bad Guy)	The Million Dollar Man Ted DiBiase	Rat	Apr-1975	449342.00	43.7689
Heel (The Bad Guy)	Mr. Perfect Curt Hennig	Rat	May-1980	13773.00	38
Heel (The Bad Guy)	Jake the Snake Roberts	Snake	Jul-1975	5609.00	37.99
Jobber (The Unknown)	Brad Smith	Sheep	Aug-2008	1103.00	36.316
Jobber (The Unknown)	Ted Duncan	Sheep	Aug-2008	200.00	33.61
Jobber (The Unknown)	Joey the Uber Nerd Cherdarchuk	Snake	Aug-2008	5.00	21.0196

Objektive tablice sadrže *identičnu* informaciju, no pretrpana verzija ima puno nepotrebnih dekorativnih elemenata (obojani retci, pune linije)!

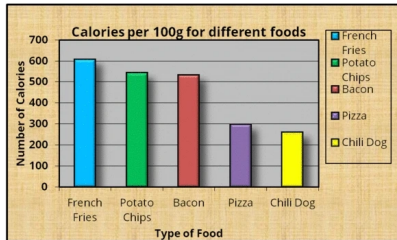
Integritet 1/4 - primjer (barchart)

Preporučeno - čisti barchart

Calories per 100g



Nije preporučeno - pretrpani barchart



Još jedan primjer - oba barcharta sadrže *identičnu* informaciju, no pretrpana verzija ima puno nepotrebnih dekorativnih elemenata!

“Prikaži izvorne podatke umjesto njihovih sažetaka” -

Prikažite izvorne podatke pomoću vizualizacija visoke rezolucije umjesto sažetaka podataka.

Kako to postići - Ljudski vizualni sustav je jako moćan i često je efektivnije prikazati izvorne podatke u visokoj rezoluciji nego sažetke podataka (primjerice, histograme, boxplotove). Sažeci uzrokuju gubitak informacije i to ima smisla samo ako smo sigurni da tako nećemo izgubiti važne uvide.

Integritet 2/4 - primjer (Anscombeov kvartet)



```
1 ans = sns.load_dataset("anscombe")
2 groups = ans.groupby("dataset")
3
4 fig, axes = plt.subplots(2, 2,
5 sharex=True, sharey=True)
6 for ax, (name, g) in zip(
7     axes.flat, groups):
8     ax.scatter(g["x"], g["y"])
9     ax.set_title(f"Skup {name}")
```

Anscombeov kvartet: Četiri posve različita skupa podataka koja imaju identične sumarne statistike:

srednja vrijednost (x) = 9

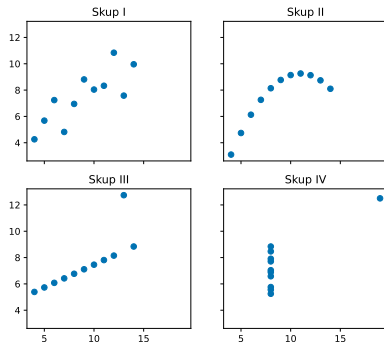
srednja vrijednost (y) = 7.50

varijanca (x) = 11

varijanca (y) = 4.125

korelacija (x i y) = 0.816

Iste sumarne statistike, različiti podaci

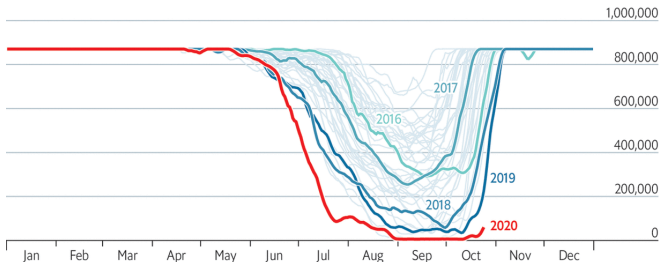


Integritet 2/4 - primjer (The Economist)

Aiming to freeze

Laptev Sea, total ice extent, km²

By year, 1978-2020



Source: NOAA

The Economist

Primjer grafa iz The Economista gdje je prikaz sirovih podataka puno efektivniji od bilo kakve sumarijacije.

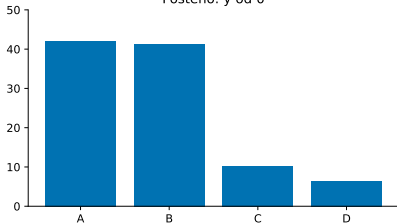
“Izbjegavaj iskrivljavanje podataka” - Predstavi podatke istinito izbjegavajući obmanjujuće vizualizacije.

Kako to postići - Izbjegavajte skraćene osi, obmanjujuće skale, iskrivljene omjere, dodavanje lažnih podataka (na primjer, dodavanje interpoliranih podataka tamo gdje ih nema) ili uklanjanje podataka - selektivni odabir podataka (*cherry picking*) ili uklanjanje anomalnih podataka (*outliers*) bez opravdanog razloga.

Integritet 3/4 - primjer (skraćena os)

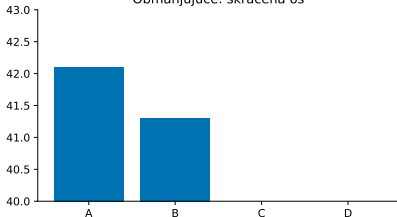
Preporučeno - y počinje od 0

Pošteno: y od 0



Nije preporučeno - skraćena os

Obmanjujuće: skraćena os



```

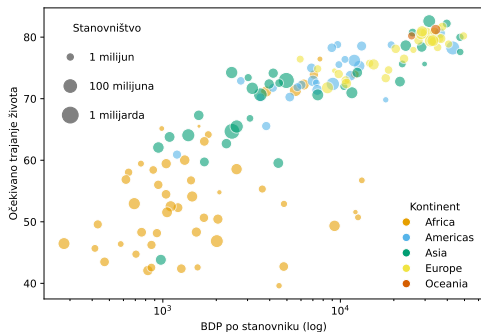
1 parties = ["A", "B", "C", "D"]
2 votes = [42.1, 41.3, 10.2, 6.4]
3
4 fig, ax = plt.subplots()
5 ax.bar(parties, votes)
6 ax.set_ylim(0, 50) # dobro!
7 # ax.set_ylim(40, 43) # loše!
```

Ista dva najveća stupca izgledaju gotovo jednako (gore) ili dramatično različito (dolje) - razlika je samo u rasponu y-osi.

“Prikaži multivariantne podatke” - Prikaži više varijabli istovremeno koristeći različite vizualne elemente.

Kako to postići - Koristite poziciju, boju, veličinu, i višestruke grafove (*faceting*) za prikaz više varijabli istovremeno.

Integritet 4/4 - primjer (Gapminder)



```
1 from gapminder import gapminder
2 g = gapminder
3 g07 = g[g.year == 2007]
4
5 for c, sub in g07.groupby("continent"):
6     plt.scatter(sub["gdpPercap"], sub["lifeExp"],
7                 s=sub["pop"] / 2e5, # velicina
8                 label=c, alpha=0.6)
9 plt.xscale("log")
10 plt.legend(title="Kontinent")
```

Četiri varijable na jednom grafu: položaj (x, y), veličina kružića i boja.

1. **“Maksimiziraj omjer podataka i tinte”** - Svaki piksel treba biti opravdan, pomaže li razumijevanju podataka ili ga ometa?
2. **“Potakni usporedbu”** - Usporedba olakšava razumijevanje jer čuva kontekst.
3. **“Integriraj tekst i grafiku”** - Koristi tekst za pomoć u razumijevanju grafike koristeći izravno označavanje.
4. **“Razumij gledatelja”** - Koristi odgovarajuće vizualne elemente za podatke koje imaš i gledatelja kojeg imaš na umu.

“Maksimiziraj omjer podataka i tinte” - Svaki piksel treba biti opravdan, pomaže li razumijevanju podataka ili ga ometa?

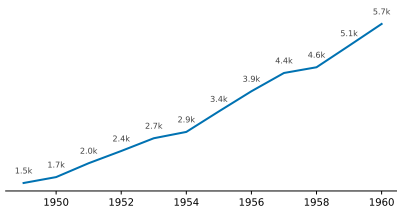
Kako to postići - Uklonite okvire, pozadine, suvišne oznake i uvijek koristite najprikladnije vizualne elemente za podatke (na primjer, položaj se može učinkovitije prikazati jednostavnom točkom nego stupcem).

Jasnoća 1/4 - primjer (putnici po godini)



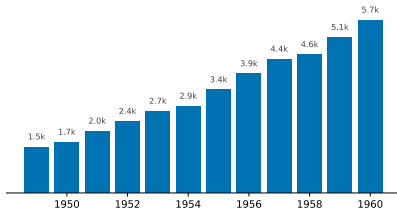
Preporučeno - linija (malo tinte)

Putnici po godini (1949-1960)



Nije preporučeno - stupci (previše tinte)

Putnici po godini (1949-1960)



```
1 fl = sns.load_dataset("flights")
2 ts = fl.groupby("year")["passengers"].sum()
3
4 fig, ax = plt.subplots()
5 ax.plot(ts.index, ts.values, lw=2) # umjesto: ax.bar(ts.index, ts.values)
6
7 # skidamo tri osi i y-oznake
8 for s in ("top", "right", "left"):
9     ax.spines[s].set_visible(False)
10 ax.set_yticks([])
```

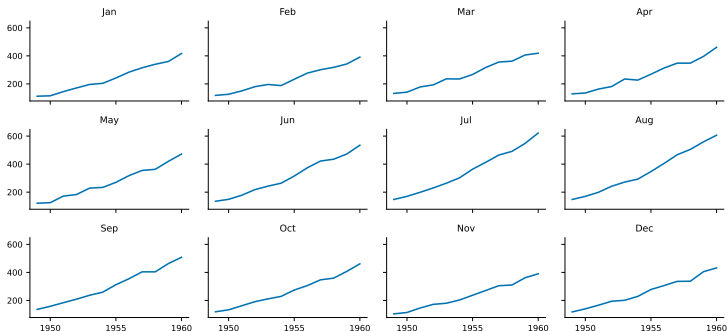
Linija i stupci nose *istu* informaciju, no stupci troše puno više tinte. Vrijednosti ispisane uz podatke čine os suvišnom.

“Potakni usporedbu” - Usporedba olakšava razumijevanje jer čuva kontekst.

Kako to postići - Poravnaj osi, koristi dosljedne ljestvice i sheme boja, postavi usporedive elemente blizu jedan drugome.

Jasnoća 2/4 - primjer (potakni usporedbu)

Putnici po mjesecu kroz godine — zajednička skala

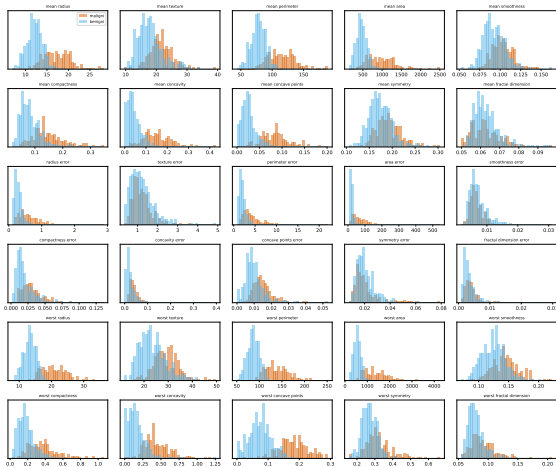


```
1 fl = sns.load_dataset("flights")
2
3 fig, axes = plt.subplots(3, 4, sharex=True, sharey=True)
4 for ax, (month, g) in zip(axes.flat, fl.groupby("month")):
5     ax.plot(g["year"], g["passengers"])
6     ax.set_title(month, fontsize=9)
```

Zajednička skala (`sharex=True` i `sharey=True`) omogućuje izravnu usporedbu.

Jasnoća 2/4 - primjer (potakni usporedbu)

Distribucija značajki: maligni vs benigni tumori



Distribucija (histogrami) svake od 30 značajki u `breast_cancer` skupu podataka za maligne i benigne tumore - brzi pregled koje značajke razdvajaju klase.

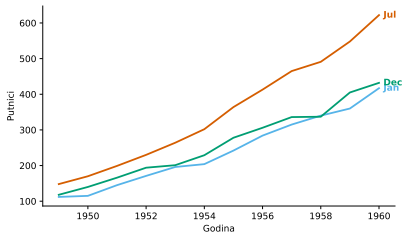
“Integriraj tekst i grafiku” - Koristi tekst za pomoć u razumijevanju grafike koristeći izravno označavanje.

Kako to postići - Označavanje osi i grafikona je minimalni zahtjev, a informativne legende su također potrebne, ali najbolje je izravno označiti važne elemente podataka (linije, točke) jer to smanjuje potrebu za unakrsnim referenciranjem (*crossreferencing*).

Jasnoća 3/4 - primjer (izravne oznake)



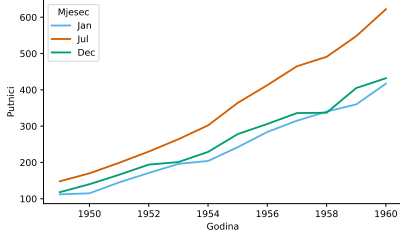
Preporučeno - izravne oznake



```
1 fig, ax = plt.subplots()
2 for m in ["Jan", "Jul", "Dec"]:
3     ax.plot(wide.index, wide[m])
4     # oznaka na kraju linije
5     ax.text(wide.index[-1] + 0.15,
6             wide[m].iloc[-1], m,
7             va="center",
8             fontweight="bold")
9     # umjesto: ax.legend()
```

Uvijek je bolje direktno označiti elemente na grafu (u ovom slučaju linije) nego se oslanjati na legende!

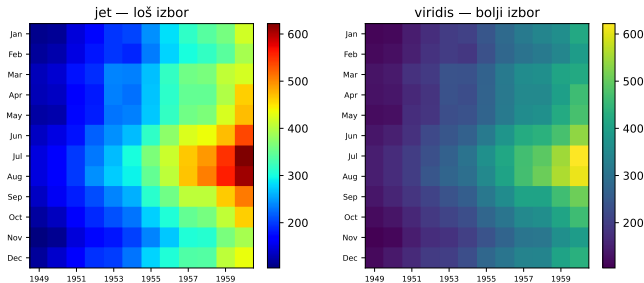
Nije preporučeno - zasebna legenda



“Razumij gledatelja” - Koristi odgovarajuće vizualne elemente za podatke koje imaš i gledatelja kojeg imaš na umu.

Kako to postići - Na primjer, položaj može kodirati informacije s mnogo većom preciznošću nego boja, debljina linije ili kutovi. Također, neki mediji zahtijevaju veću razlučivost (tisak) nego drugi (web).

Jasnoća 4/4 - primjer (izbor palete boja)



```
1 mat = fl.pivot(index="month", columns="year", values="passengers")
2
3 fig, (ax1, ax2) = plt.subplots(1, 2)
4 ax1.imshow(mat, cmap="jet") # loše
5 ax2.imshow(mat, cmap="viridis") # bolje
```

Paleta *viridis* je perceptivno ujednačena i čitljiva i osobama s daltonizmom. Standardna *jet* paleta unosi nepostojeće "granice" između boja.

Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

Jasnoća

Vizualizacija rezultata strojnog učenja

Zaključak

Vizualizacija nije korisna samo za prikaz podataka, već i za **analizu i prikaz rezultata** algoritama strojnog učenja. Isti principi (integritet i jasnoća) vrijede i ovdje.

U nastavku prikazujemo nekoliko tipičnih primjera:

- matrica korelacije značajki
- ROC krivulja i površina ispod krivulje (AUC)
- matrica konfuzije
- grupiranje (*clustering*)
- prilagodba modela i prenaučenost (polinomna regresija)
- krivulje učenja i validacije modela

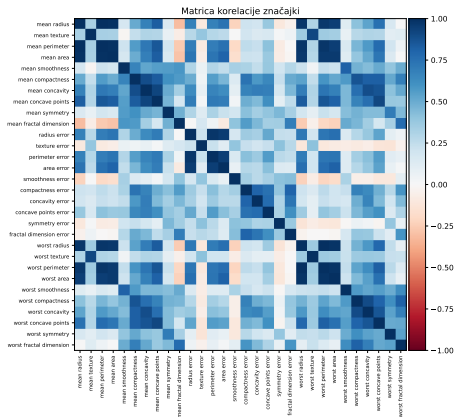
Matrica korelacije značajki

Matrica korelacije otkriva grupe međusobno povezanih značajki.

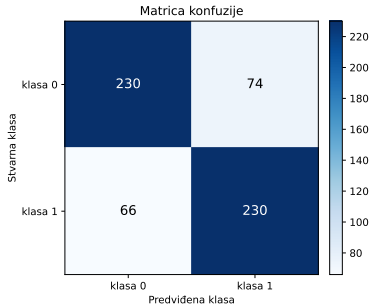
Toplinska karta (*heatmap*) s divergentnom paletom:

plavo = pozitivna korelacija

crveno = negativna korelacija



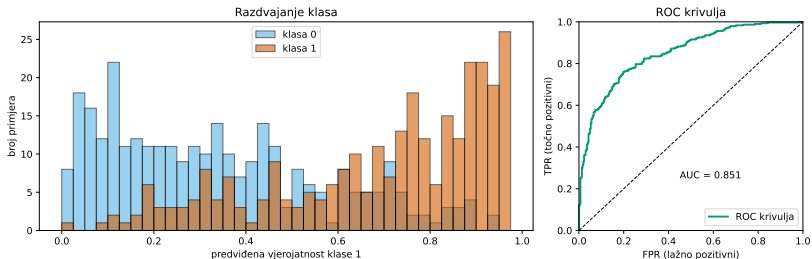
```
1 from sklearn.metrics import
  confusion_matrix
2
3 cm = confusion_matrix(
4     y_test, model.predict(X_test))
5
6 im = ax.imshow(cm, cmap="Blues")
7 # upisivanje brojeva u celije
8 for i in range(2):
9     for j in range(2):
10        ax.text(j, i, cm[i, j],
11               ha="center",
12               va="center")
```



Dijagonala = točne klasifikacije

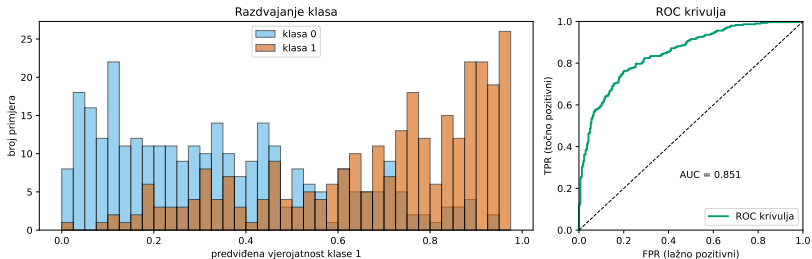
Izvan dijagonale = pogreške

ROC krivulja i AUC



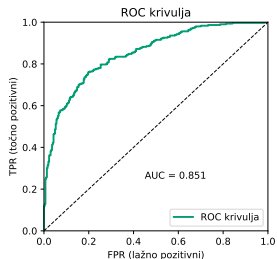
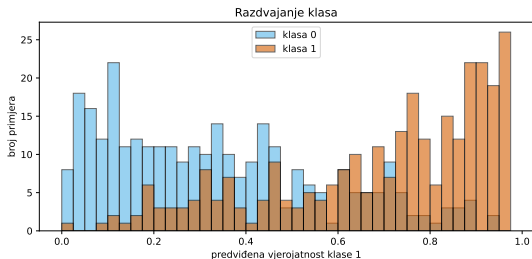
- **TPR** (*True Positive Rate*) = osjetljivost = udio točno klasificiranih pozitivnih primjera
- **FPR** (*False Positive Rate*) = udio negativnih primjera pogrešno klasificiranih kao pozitivni.

ROC krivulja i AUC



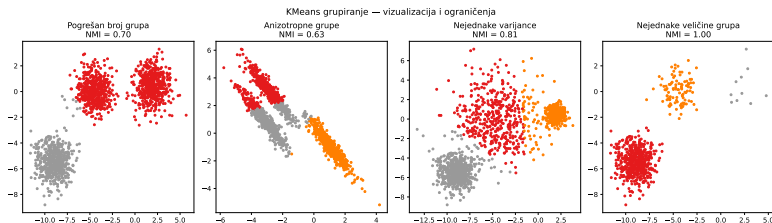
- **ROC** (*Receiver Operating Characteristic*) krivulja pokazuje kompromis između TPR i FPR pri različitim pragovima klasifikacije.
- **AUC** (*Area Under the Curve*) - površina pod ROC krivuljom, mjeri ukupnu sposobnost modela da razlikuje klase, bez obzira na prag. AUC sažima krivulju u jedan broj: 1.0 = savršen klasifikator, 0.5 = slučajan klasifikator.

ROC krivulja i AUC



```
1 from sklearn.metrics import (roc_curve, roc_auc_score)
2
3 # model je već treniran
4 y_proba = model.predict_proba(X_test)
5 fpr, tpr, _ = roc_curve(y_test, y_proba[:, 1])
6 auc = roc_auc_score(y_test, y_proba[:, 1])
7
8 ax.plot(fpr, tpr)
9 ax.plot([0, 1], [0, 1], "--")
```

Grupiranje (*clustering*)

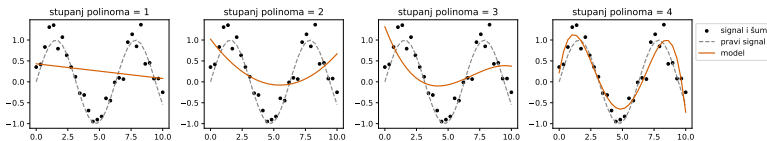


```
1 from sklearn.cluster import KMeans
2 y_pred = KMeans(n_clusters=3).fit_predict(X)
3 plt.scatter(X[:, 0], X[:, 1], c=y_pred, cmap="Set1") # boja = grupa
```

k -srednjih vrijednosti (*k-means*) dobro radi za okrugle, jednako velike grupe, a griješi kod anizotropnih ili nejednakih grupa.

Normalized Mutual Information (NMI) mjeri slaganje sa stvarnim grupama.

Polinomna regresija raznih stupnjeva (ručno)

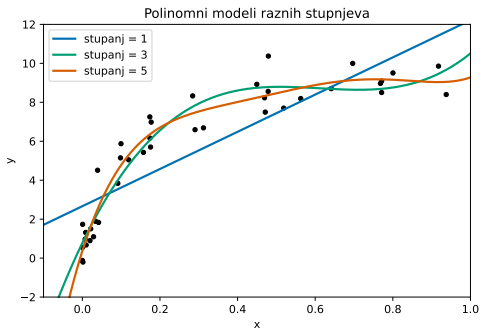


```
1 from sklearn.preprocessing import PolynomialFeatures
2 from sklearn.linear_model import LinearRegression
3 from sklearn.pipeline import Pipeline
4
5 for degree in [1, 2, 3, 4]:
6     model = Pipeline([("poly", PolynomialFeatures(degree)),
7                       ("linear", LinearRegression())])
8     model.fit(X, y)
9     ax.plot(X, model.predict(X)) # prilagodjena krivulja
```

Premali stupanj = podnaučenost (*underfitting*)

Preveliki stupanj = prenaučnost (*overfitting*)

Polinomni modeli raznih stupnjeva



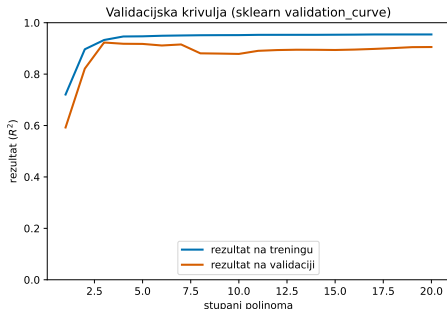
```
1 from sklearn.pipeline import make_pipeline
2
3 def poly_model(degree):
4     return make_pipeline(PolynomialFeatures(degree), LinearRegression())
5
6 for d in [1, 3, 5]:
7     m = poly_model(d).fit(X, y)
8     plt.plot(x_grid, m.predict(x_grid), label=f"stupanj = {d}")
```

make_pipeline povezuje transformaciju značajki i model u jedan objekt.

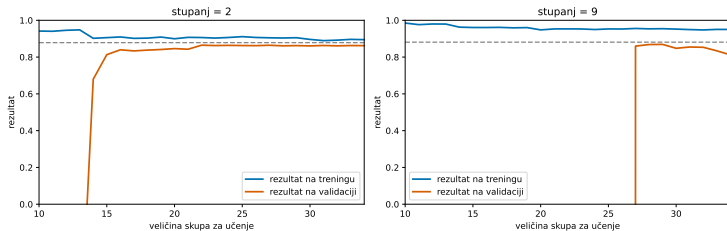
Odabir složenosti modela (validacija)

Validacijska krivulja prikazuje uspješnost na skupu za učenje i validaciju u ovisnosti o složenosti modela (ovdje je to stupanj polinoma).

Velik razmak između krivulja = *prenaučenost* (overfitting).



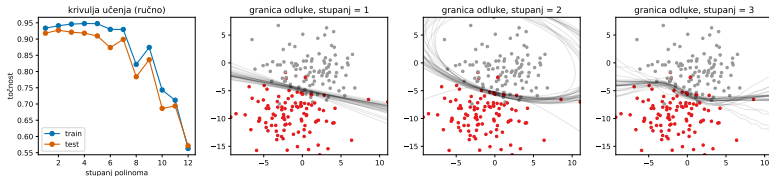
Krivulja učenja (sklearn learning_curve)



```
1 from sklearn.model_selection import learning_curve
2
3 N, train_lc, val_lc = learning_curve(
4     model, X, y, cv=7,
5     train_sizes=np.linspace(0.3, 1, 25))
6
7 plt.plot(N, train_lc.mean(1), label="trening")
8 plt.plot(N, val_lc.mean(1), label="validacija")
```

Uspješnost u ovisnosti o veličini skupa za učenje - pokazuje pomaže li više podataka.

Ručna konstrukcija krivulje učenja



```
1 from sklearn.model_selection import train_test_split
2
3 for degree in range(1, 13):
4     train = test = 0.0
5     for _ in range(30): # usrednjavanje preko 30 podjela
6         X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.33)
7         model.fit(X_tr, y_tr)
8         train += model.score(X_tr, y_tr)
9         test += model.score(X_te, y_te)
10    accuracy_train.append(train / 30)
11    accuracy_test.append(test / 30)
```

Isti princip bez gotove funkcije: za svaki stupanj usrednjavamo točnost preko 30 nasumičnih podjela (lijevo), a desno je raspršenost granica odluke za stupnjeve 1-3.

Uvod

Uvod u Python i Matplotlib

Principi Edwarda Tuftea

Integritet

Jasnoća

Vizualizacija rezultata strojnog učenja

Zaključak

- Python (Matplotlib, seaborn, pandas) omogućuje izradu kvalitetnih vizualizacija s malo koda.
- Vizualizacijski principi Edwarda Tuftea - **integritet** i **jasnoća**, pomažu nam izraditi vjernije, razumljive i vizualno privlačnije grafove.
- Isti principi vrijede i pri vizualizaciji **rezultata strojnog učenja** (ROC, matrica konfuzije, grupiranje, krivulje učenja).

Hvala na pažnji!