

Istraživanje podataka 1

Vežbe 5

28. März 2021

Outline

- 1 Istraživanje podataka u programskom jeziku Python
- 2 Unakrsna validacija

Outline

- 1 Istraživanje podataka u programskom jeziku Python
- 2 Unakrsna validacija

Programski jezik Python

- Za programiranje u pj Python može se koristiti integrisano razvojno okruženje PyChram
- ekstenzija programa *.py*
- Pri istraživanju podataka često se koristi Jupyter notebook dokument koji može da sadrži kombinaciju koda napisanog u pj Python, rezultat koda, tekst i slike. Ekstenzija dokumenta je *.ipynb*.

Istraživanje podataka u programskom jeziku Python

- biblioteke
 - pandas - obezbeđuje strukture podataka za rad sa relacionim ili označenim podacima.
 - *scikit-learn* - biblioteka za mašinsko učenje

NumPy

Biblioteka *NumPy* osnovna biblioteka za naučno izračunavanje u programskom jeziku Python.

- **ndarray** (alias `array`) klasa za nizove
`import numpy as np`
`b = np.array([6, 7, 8])`
- dimenzije se nazivaju osama (eng. `axes`)

pandas

Biblioteka *pandas* obezbeđuje strukture podataka za rad sa relacionim ili označenim podacima.

Osnovne strukture podataka su:

- **Series** za rad sa 1D podacima
- **DataFrame** za rad sa 2D podacima

Series - primer

```
import pandas as pd
import numpy as np

s1 = pd.Series([1,2,3,5])
```

```
0    1
1    2
2    3
3    5
```

```
dtype: int64
```

```
s1[2]
```

```
3
```

```
s1.get(1)
```

```
2
```

```
s1.get(8, np.nan)
```

```
nan
```


Series - primer

```
s1 = pd.Series([1,2,3,5], index=['a', 'b', 'c', 'd'])
```

```
a    1
```

```
b    2
```

```
c    3
```

```
d    5
```

```
dtype: int64
```

```
s1['b']
```

```
2
```

Series

Neki atributi:

- **index** - oznake vrednosti
- **values** - vrednosti

Neke metode:

- **keys** - oznake vrednosti
- **value_counts** - broj pojavljivanja za svaku vrednost

Operacije (+, -, *, /) se vrše nad elementima sa istim vrednostima indeksa

Series - primer

```
s1.index
```

```
Index(['a', 'b', 'c', 'd'], dtype='object')
```

```
s1.values
```

```
array([1, 2, 3, 5], dtype=int64)
```

```
s1.value_counts()
```

```
5    1
3    1
2    1
1    1
dtype: int64
```

DataFrame

DataFrame je 2D označena struktura podataka sa kolonama koje mogu biti različitih tipova.

- **index** - oznaka reda
- **column** - oznaka kolone

```
pandas.DataFrame( data, index, columns)
```

DataFrame - primer

```
d1= {'prva': pd.Series([1,2,3, 4], index=['a', 'b', 'c', 'd']),  
     'druga': pd.Series(['x', 'y', 'z'], index=['a', 'b', 'c'])  
    }
```

```
df1=pd.DataFrame(d1)
```

```
df1
```

	prva	druga
a	1	x
b	2	y
c	3	z
d	4	NaN

DataFrame

Za tipove kolona *pandas* koristi

- nizove biblioteke NumPy (float, int, bool)
- pandas.CategoricalDtype - kategorički podaci (sa kategorijama i poretkom)

```
t = pd.CategoricalDtype(categories=['b', 'a'], ordered=True)  
pd.Series(['a', 'b', 'a', 'c'], dtype=t)
```
- za čuvanje niski koristi dtype *object*
- ...

DataFrame - primer

```
df2 = pd.DataFrame({'cat' : pd.Categorical(['a', 'b', 'a', 'x'],  
                                          categories=['b', 'a'], ordered=True),  
                   'num' : [1,2,2,3],  
                   'obj' : ['c', 'd', 'e', 'c']  
                   })
```

df2

	cat	num	obj
0	a	1	c
1	b	2	d
2	a	2	e
3	NaN	3	c

DataFrame - primer

```
df2.dtypes
```

```
cat    category  
num      int64  
obj     object  
dtype: object
```


DataFrame

Izdvajanje redova i kolona:

- **iloc** - izdvajanje redova i kolona prema poziciji
iloc[izbor reda, izbor kolone]
izbor: pozicija (jedan red), [lista pozicija], opseg (donja pozicija : gornja pozicija)
- **loc** - izdvajanje redova i kolona prema oznaci/indeksu
loc[izbor reda, izbor kolone]
izbor: labela, [lista labela], opseg (donja labela : gornja labela)
- **df[col]** - izdvajanje kolone
- **df[uslov]** - izdvajanje redova prema uslovu

DataFrame - primer

```
df1['prva']
```

```
d    4  
c    3  
b    2  
a    1  
Name: prva, dtype: int64
```

```
df1['prva']['c']
```

```
3
```

DataFrame - primer

```
df2
```

	cat	num	obj
0	a	1	c
1	b	2	d
2	a	2	e
3	NaN	3	c

```
df2.iloc[0,1] #[pozicija reda, pozicija kolone]
```

```
1
```

DataFrame - primer

```
df2.iloc[1,1:]
```

```
num    2  
obj    d  
Name: 1, dtype: object
```

```
df2.iloc[[1],1:]
```

	num	obj
1	2	d

```
df2.iloc[[1,3],1:]
```

	num	obj
1	2	d
3	3	c

DataFrame - primer

```
df1
```

	druga	prva
d	NaN	4
c	z	3
b	y	2
a	x	1

```
df1.loc['a']
```

```
druga    x  
prva     1  
Name: a, dtype: object
```

DataFrame - primer

```
df1.loc['c':'a']
```

	druga	prva
c	z	3
b	y	2
a	x	1

```
df1.loc[['a', 'c'], ['prva', 'druga']]
```

	prva	druga
a	1	x
c	3	z

DataFrame - primer

```
df1[df1['prva']>1]
```

	druga	prva
d	NaN	4
c	z	3
b	y	2

```
df1[df1['prva']>1]['druga']
```

```
d    NaN  
c     z  
b     y  
Name: druga, dtype: object
```

```
df1[df1['prva']>1][['druga']]
```

	druga
d	NaN
c	z
b	y

DataFrame

Neke metode:

- **sort_index** - sortiranje redova/kolona prema oznakama
- **sort_values** - sortiranje redova/kolona prema vrednostima
- **describe** - sortiranje redova/kolona prema vrednostima
- **any** - proverava da li je neka vrednost *True*
- **all** - proverava da li su sve vrednosti *True*
- **drop** - brisanje reda

DataFrame - primer

```
df1.sort_index(ascending=False, inplace=True)
```

```
df1
```

	prva	druga
d	4	NaN
c	3	z
b	2	y
a	1	x

DataFrame - primer

```
df1.sort_index(ascending=True, inplace=True, axis=1)
```

```
df1
```

	druga	prva
d	NaN	4
c	z	3
b	y	2
a	x	1

DataFrame - primer

```
df2.describe(include='all')
```

	cat	num	obj
count	3	4.000000	4
unique	2	NaN	3
top	a	NaN	c
freq	2	NaN	2
mean	NaN	2.000000	NaN
std	NaN	0.816497	NaN
min	NaN	1.000000	NaN
25%	NaN	1.750000	NaN
50%	NaN	2.000000	NaN
75%	NaN	2.250000	NaN
max	NaN	3.000000	NaN

DataFrame - primer

```
df2['new'] = df2['num'] * 2
```

```
df2
```

	cat	num	obj	new
0	a	1	c	2
1	b	2	d	4
2	a	2	NaN	4
3	NaN	3	c	6

DataFrame - primer

```
df2.isna()
```

	cat	num	obj	new
0	False	False	False	False
1	False	False	False	False
2	False	False	True	False
3	True	False	False	False

```
df2.isna().any()
```

```
cat      True
num      False
obj      True
new      False
dtype: bool
```

```
df2.isna().any().any()
```

```
True
```

DataFrame

Statistike:

- **mean**
- **quantile**
- **mode** - sortiranje redova/kolona prema vrednostima
- **nunique** - sortiranje redova/kolona prema vrednostima
- ...

DataFrame - primer

```
df2.mean()
```

```
num    2.0  
dtype: float64
```

```
df2.mean(axis=1)
```

```
0    1.0  
1    2.0  
2    2.0  
3    3.0  
dtype: float64
```

DataFrame - primer

```
df2.quantile(0.25)
```

```
num    1.75  
Name: 0.25, dtype: float64
```

```
df2.quantile([0.25, 0.5, 0.75])
```

	num
0.25	1.75
0.50	2.00
0.75	2.25

Obrada nedostajućih vrednosti

numpy.nan - nedostajuća vrednost

Metode za rad sa nedostajućim vrednostima:

- **isnull** - detekcija nedostajućih vrednosti (za *numpy.nan* vraća *True*)
- **notnull** - detekcija nedostajućih vrednosti (za *numpy.nan* vraća *False*)
- **fillna** - zamena *numpy.nan* sa zadatim argumentom
- **dropna** - brisanje redova/kolona sa *numpy.nan* vrednostima
- **replace** - zamena jedne vrednosti sa drugom

DataFrame - primer

```
df1
```

	druga	prva
d	NaN	4.0
c	z	NaN
b	y	2.0
a	x	1.0

```
df1['prva'].replace(np.NaN, df1['prva'].mean(), inplace=True)
```

```
df1
```

	druga	prva
d	NaN	4.000000
c	z	2.333333
b	y	2.000000
a	x	1.000000

DataFrame - primer učitavanja skupa

```
iris = pd.read_csv('iris.csv')
```

```
iris.shape
```

```
(146, 5)
```

```
iris.head(5)
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Klasifikacija: podela na skup za treniranje i testiranje

- *model_selection.train_test_split* - podela skupa podataka na deo za treniranje modela i deo za testiranje
 - *train_size* - veličina skupa za treniranje
 - ako je realan broj onda je procenat instanci u trening skupu
 - ako je ceo broj onda je broj instanci u trening skupu
 - *test_size* - veličina skupa za testiranje, default=0.25
 - ako je realan broj onda je procenat instanci u test skupu
 - ako je ceo broj onda je broj instanci u test skupu
 - *shuffle* - da li se vrši mešanje instanci u skupu pre podele, default=True
 - *random_state* - seme za generisanje slučajnih brojeva
 - *stratify* - stratifikovana podela, navodi se lista oznaka klasa

Klasifikacija: drveta odlučivanja u scikit-learn

Klasa `tree.DecisionTreeClassifier`

- neki parametri
 - *criterion* - kriterijum za podelu : gini, entropy , default=gini
 - *max_depth* - maksimalna dubina drveta: ako nije zadata, drvo se širi do čistih listova ili dok čvorovi imaju više od *min_samples_split* instanci.
 - *min_samples_split* - minimalan broj instanci u čvoru da bi došlo do podele čvora : default=2

Klasifikacija: drveta odlučivanja u scikit-learn

- neki parametri
 - *min_samples_leaf* - minimalan broj instanci u listu: default=1
 - *max_leaf_nodes* - maksimalan broj listova
 - *min_impurity_decrease* - minimalno smanjenje nečistoće.
Čvor će biti podeljen ako je smanjenje nečistoće veće ili jednako od zadate vrednosti.

Klasifikacija: drveta odlučivanja u scikit-learn

- neki atributi
 - *classes_* - oznake klasa
 - *feature_importances_* - značajnost atributa
 - *tree_* - drvo odlučivanja

Klasifikacija: drveta odlučivanja u scikit-learn

- metode
 - *fit(x,y)* - za pravljenje drveta odlučivanja na osnovu skupa (x,y)
 - *predict(x)* - za predviđanje klase za x
 - *predict_proba(x)* - vraća verovatnoću klase za x. Predviđena verovatnoća klase je procenat instanci te klase u listu. Redosled klase odgovara onome u atributu *classes_*.

Klasifikacija: drveta odlučivanja u scikit-learn

- *tree.plot_tree* - funkcija za iscrtavanje drveta odlučivanja
 - *decision_tree*
 - *feature_names* - imena atributa radi lepšeg prikaza test-uslova, default=None
 - *class_names* - imena klasa radi prikaza imena najzastupljenije klase, default=None
 - *filled* - svakoj klasi se dodeljuje boja i čvorovi se boje prema najzastupljenijom klasi, default=None
 - *impurity* - prikaz nečistoće čvora, default=True
 - *rounded* - prikaz čvora sa zaobljenih uglovima, default=False

Klasifikacija: mere performansi

- `confusion_matrix(y_true, y_pred)` - matrica konfuzije
- `accuracy_score(y_true, y_pred, normalize=True)` - preciznost, ako je `normalize=False` vraća broj tačno klasifikovanih instanci
- `precision_score(y_true, y_pred, average)` - preciznost
- `recall_score(y_true, y_pred, average)` - odziv
- `f1_score(y_true, y_pred, average)`
- `classification_report(y_true, y_pred)` - tekstualni izveštaj sa glavnim merama za klasifikaciju

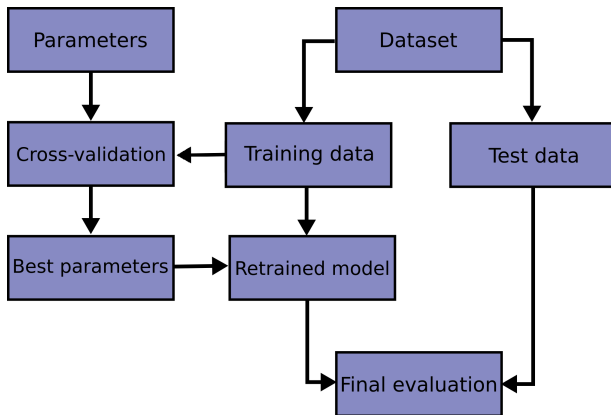
Klasifikacija: mere performansi

- opcije za *average*
 - `average=None`, rezultat je skor za svaku klasu
 - `average=binary`, vraća rezultat samo za klasu navedenu preko parametra *pos_label*
 - `average=macro`, mera se računa globalno, brojanjem ukupnih tp, fn i fp instanci.
 - `average=weighted`, računa meru za svaku klasu posebno i vraća srednju vrednost ponderisanu podrškom.

Outline

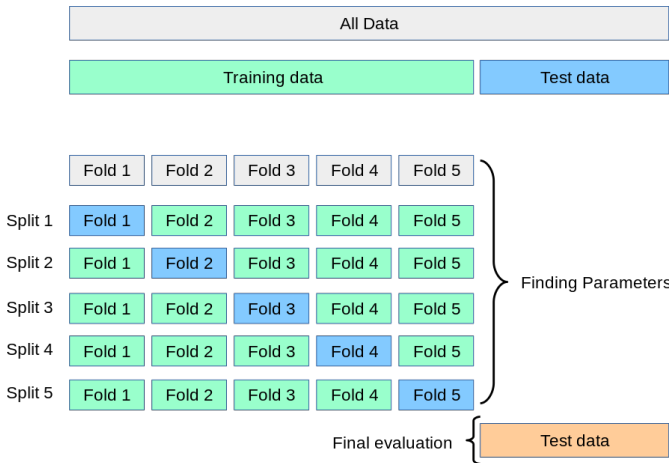
- 1 Istraživanje podataka u programskom jeziku Python
- 2 Unakrsna validacija

Unakrsna validacija



- Podešavanje hiper-parametara procenjivača

Unakrsna validacija



Unakrsna validacija

- `sklearn.model_selection.GridSearchCV`
- iscrpna pretraga korišćenjem zadatih vrednosti parametara za procenjivača
- parametri
 - *estimator* - procenjivač (npr. `KNeighborsClassifier`)
 - *param_grid* - rečnik ili lista rečnika sa definisanim mogućim vrednostima za parametre procenjivača
 - *scoring* - mera za proveru modela
 - *cv* - broj podskupova za unakrsnu validaciju, `default= 3-fold`
 - *refit* - da li ponovo napraviti model sa najboljim parametrima nad celim skupom podataka. Da bi mogla da se radi predikcija nad drugim skupom potrebno je staviti `True`. (`default=True`)

Unakrsna validacija

- atributi
 - *cv_results_* - rečnik sa podacima o zadatim parametrima i rezultatima
 - *best_estimator_* - najbolji klasifikator
 - *best_score_* - najbolji skor
 - *best_params_* - parametri koji daju najbolji rezultat
 - *scorer_* - funkcija za skor

Unakrsna validacija

- metode
 - *fit(x,y)* - pravljenje modela sa optimalnim parametrima na osnovu skupa (x,y)
 - *predict(x)* - predviđanje klase za test instancex