

ORI

Python primeri modela

Klasterovanje

- Preporuka je koristiti sledeće Python biblioteke
 - Scikit-learn
 - za sve algoritme koje smo radili
 - Pandas
 - za učitavanje skupa podataka i manipulaciju nad njim
 - umesto pandas biblioteke možete koristiti i običan “csv” modul koji python poseduje, pa ručno učitavati kao na vežbama. Primer možete naći na [linku](#).

K-means klasterovanje

```
from sklearn.datasets import load_iris  
from sklearn.model_selection import train_test_split  
from sklearn.cluster import Kmeans
```

```
X, y = load_iris(return_X_y=True)  
km = KMeans(n_clusters = 3, n_init = 10, random_state=21)  
km.fit(X)
```

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

<https://medium.com/@belen.sanchez27/predicting-iris-flower-species-with-k-means-clustering-in-python-f6e46806aeee>

DB-Scan klasterovanje

```
from sklearn.datasets import load_iris  
from sklearn.model_selection import train_test_split  
from sklearn.cluster import DBSCAN
```

```
X, y = load_iris(return_X_y=True)  
clustering = DBSCAN(eps=3, min_samples=2)  
clustering.fit(X)
```

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>

Klasifikacija

- Preporuka je koristiti sledeće Python biblioteke
 - Scikit-learn
 - za sve algoritme koje smo radili
 - Keras
 - za modelovanje neuronskih mreža zbog fleksibilnosti, jednostavnosti i velikog broja primera kompleksnih modela implementiranih pomoću ovog framework-a
 - scikit-learn takođe ima implementaciju VNM
 - Pandas
 - za učitavanje skupa podataka i manipulaciju nad njim
 - umesto pandas biblioteke možete koristiti i običan “csv” modul koji python poseduje, pa ručno učitavati kao na vežbama. Primer možete naći na [linku](#).

Naïve Bayes klasifikator

```
from sklearn.datasets import load_iris  
from sklearn.model_selection import train_test_split  
from sklearn.naive_bayes import GaussianNB
```

```
X, y = load_iris(return_X_y=True)  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)  
gnb = GaussianNB()  
y_pred = gnb.fit(X_train, y_train).predict(X_test)
```

https://scikit-learn.org/stable/modules/naive_bayes.html

Neuronska mreža (scikit-learn implementacija)

```
from sklearn.datasets import load_iris
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.neural_network import MLPClassifier
```

```
X, y = load_iris(return_X_y=True)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
```

```
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1)
```

```
clf.fit(X_train, y_train)
```

https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Neuronska mreža (Keras implementacija)

- Keras je **preporuka** za implemtaciju VNM
- Primer klasifikacije nad IRIS datasetom se nalazi na sledećem linku

<https://gist.github.com/NiharG15/cd8272c9639941cf8f481a7c4478d525>