



Vežbe br. 10

MAŠINSKO UČENJE



Zadatak za vežbu

- Modifikovati zadatak sa prethodnih vežbi, tako da se u procesu kreiranja različitih modela ne ponavljaju isti delovi koda. Za stablo odluke podesiti hiperparametre i prikazati model.


```
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score, f1_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn import tree

dataset = load_iris()
X = dataset.data[:, 2:]
y = dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y)

models = [KNeighborsClassifier(), DecisionTreeClassifier()]

for model in models:
    name = model.__class__.__name__
    print(name)
    model.fit(X_train, y_train)
    y_predict = model.predict(X_test)
    print(" Tacnost je: ", accuracy_score(y_test, y_predict))

if name == "DecisionTreeClassifier":
    plt.figure(figsize=(10, 10))
    _ = tree.plot_tree(model, feature_names=dataset.feature_names, class_names=dataset.target_names, filled=True)
    plt.show()

    hiperparametri = [
        {
            'criterion': ["gini", "entropy"],
            'ccp_alpha': [0, 0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 50, 100, 200]
        }
    ]

    grid = GridSearchCV(model, hiperparametri, cv=10)
    grid.fit(X_train, y_train)
    print("Najbolja kombinacija je: ", grid.best_params_)

    model.criterion = grid.best_params_["criterion"]
    model.ccp_alpha = grid.best_params_["ccp_alpha"]

    model.fit(X_train, y_train)
    y_predict = model.predict(X_test)
    print("Tacnost stabla (posle podesavanja) je: ", accuracy_score(y_test, y_predict))
```

Priprema za kolokvijum

- Za dati [skup podataka](#) izvršiti kreiranje modela mašinskog učenja koji će, na osnovu podataka o bankovnom nalogu osobe, predviđati da li će ta osoba napustiti banku (atribut „Exited“). Napraviti 3 modela, zasnovana na sledećim algoritmima: „logistička regresija“, „K najbližih suseda“, „stablo odluke“. Za svaki model prikazati njegovu tačnost. Za model zasnovan na algoritmu „K najbližih suseda“ prikazati i matricu konfuzije, a kod modela zasnovanog na logističkoj regresiji izvršiti podešavanje hiperparametra „C“ (definisati neke nasumične pozitivne vrednosti). Ispisati F1 skor modela koji se dobija nakon podešavanja hiperparametara.

```
y = dataset['Exited']  
X = dataset.drop(['Exited'], 1)
```