Dokumentacija koda za projektni zadatak

Kod za projektni zadatak je pisan kao Jupyter Notebook., a programski jezik koji je korišten je Python, verzija 3.

## Potrebno napraviti za sve modele

Za ispravan rad potrebno je uključiti sljedeće pakete:

* from sklearn import preprocessing
* import matplotlib.pyplot as plt
* import numpy as np
* import pandas as pd
* from sklearn.model\_selection import train\_test\_split
* from sklearn.naive\_bayes import GaussianNB
* from sklearn.metrics import roc\_curve, auc, roc\_auc\_score
* import matplotlib.pyplot as plt
* from sklearn.preprocessing import MinMaxScaler
* from sklearn.tree import DecisionTreeClassifier
* from sklearn.ensemble import RandomForestClassifier
* from sklearn import svm
* from sklearn.pipeline import Pipeline

Mapa u kojoj se pokreće, izvršava kod mora sadržavati i mapu bank-additional koja u sebi ima sljedeće datoteke:

* bank-additional-full.csv – datoteka koja sadrži dataset u csv formatu
* bank-additional.csv – sadrži podatke za testiranje

Nakon što se učitaju potrebni podaci (iz datoteka bank-additional-full.csv i bank-additional.csv), podaci se transformiraju u pogodan oblik za korištenje modela, algoritama strojnog učenja:

df01 = data.append(test, ignore\_index=True)

…

data\_X = data\_X\_ALL[0:41188]

data\_y = data\_y\_ALL[0:41188]

test\_X = data\_X\_ALL[41188:45307]

test\_y = data\_y\_ALL[41188:45307]

scaler = MinMaxScaler()

scaler.fit(test\_X)

test\_X\_trs = scaler.transform(test\_X)

## Korištenje pojedinog modela

Za korištenje pojedinog modela prvo se pripremaju podaci specifični za taj model:

X\_train\_NB, X\_test\_NB, y\_train\_NB, y\_test\_NB = train\_test\_split(data\_X, data\_y, test\_size=0.33, random\_state=42)

scaler = MinMaxScaler()

scaler.fit(X\_train\_NB)

X\_train\_trs\_NB =scaler.transform(X\_train\_NB)

X\_test\_trs\_NB =scaler.transform(X\_test\_NB)

Dalje slijedi treniranje, izrada modela:

modelNB = GaussianNB()

modelNBFIT = modelNB.fit(X\_train\_trs\_NB, y\_train\_NB)

probasNB\_ = modelNBFIT.predict\_proba(X\_test\_trs\_NB)

fprNB, tprNB, thresholdsNB = roc\_curve(y\_test\_NB, probasNB\_[:, 1])

roc\_aucNB = auc(fprNB, tprNB)

…

Te na kraju kod grafički prikaz uspješnosti modela:

plt.clf()

plt.figure(figsize=(10,5))

plt.subplot(1, 2, 1)

plt.plot(fprNB, tprNB, label='ROC curve (area = %0.4f)' % roc\_aucNB)

plt.plot([0, 1], [0, 1], 'k--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.0])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

…

Modeli su „neovisni“ jedan o drugome. Nakon prva dva koraka, kad se uključe potrebni paketi, učitaju i transformiraju podaci (što je zajedničko za sve modele), kod za pojedini model se može izvršavati neovisno o kodu za ostale modele.