

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.