Raport Tema 3

Data Mining

Bogdan Rusu MOC1

Motroi Valeriu MSAI 1

1. **Descrierea modului de lucru**
   Setul de date **Forest Cover Type**, a fost împărțit în 5 bucketuri. Patru din ele au fost folosite pentru antrenament și unul pentru testare (realizând un round robin pentru fiecare bucket în parte). De asemenea, observațiile asociate fiecărei clase au fost distribuite uniform în cele 5 mulțimi. Aceaste bucketuri ne-au ajutat ca să comparam 8 algoritmi (Arbori de decizie, Bayes Naiv, K-NN, Retele Neurale, SVM, XGBoost, Retele Neurale Stacked, Random Forest). Mai jos sunt rezultatele succint descrise. In fișierul **classifiers.html** se pot găsi rezultatele complete.

   Metoda de alegere a hiperparametrilor: S-a folosit un algoritm genetic implementat în biblioteca **scipy.optimize.differential_evolution.** Funcția de fitness fiind simple modificări ale funcției exp(acc^-2), unde acc este acuratețea pe setul de test.

2. **Arbori de decizie**
   - **Implementarea:** sklearn.tree.DecisionTreeClassifier (python)
   - **Parametri:**
     Înălțimea maxima: **13**
     Criteriul: **Entropia**
     Impuritatea minima: **2.79316138e-04**
   - **Acuratețea pentru fiecare bucket:** 78.77%, 77.98%, 78.34%, 78.77%, 78.44%.
   - **Media acurateții**: **78.46%**
   - **Varianta: 0.09**
3. **Bayes Naiv**
   - **Implementarea:** sklearn.tree.GaussianNB (python)
   - **Parametri:** Default
   - **Acuratețea pentru fiecare bucket:** 41.24%, 42.79%, 45.07%, 42.96%, 42.63%
   - **Media acurateții: 42.94%**
   - **Varianta: 1.5**
4. **K-NN**
   - **Implementarea:** sklearn.tree.KneighborsClassifier (python)
   - **Parametri:**
     K: **3**
   - **Acuratețea pentru fiecare bucket:** 83.13%, 82.11%, 82.24%, 83%, 81.48%
   - **Media acurateții: 82.39%**
   - **Varianța: 0.37**
5. **Retele Neurale**
   - **Implementarea:** Folosind Keras (python)

- **Parametri:**
  Activare: **ReLU**, cu excepția ultimui layer unde este folosit **SoftMax**
  Functia de loss: **CrossEntropy**
  Optimizator: **Adam**
  Numarul de epoci de antrenament: **12**
  Batch Size: **70**
  Ponderea claselor: [1, 2] -> **3/16**, [3, 4, 5, 6, 7] -> **2/16**
  Reteaua: **4** straturi ascunse fully connected (dimensiunile **256, 196, 128, 96**) toate urmate de **Batch Normalization**.
- **Acuratețea pentru fiecare bucket:** 81.88%, 83.07%, 82.54%, 83.43%, 82.18%
- **Media acurateții: 82.62%**
- **Varianța: 0.32**

6. **SVM**
   - **Implementarea:** sklearn.svm.SVC (python)
   - **Parametri:**
     Kernel: **Polinomial** de grad **3**
     Cache Size: 2048
     gamma: Scale
   - **Acuratețea pentru fiecare bucket:** 71.16%, 70.14%, 70.44%, 71.49%, 72.42%
   - **Media acurateții: 71.13%**
   - **Varianța: 0.65**
7. **XGBoost**
   - **Implementarea:** xgboost (R)
   - **Parametri:**
     Funcția obiectiv: **multi:softmax**
     Funcția de loss: **mean**
     Gamma: **0.501, 1.0**
     Eta: **0.3, 0.025, 0.0025** (descreste in functie de iteratie)
     Înălțimea maxima: **5, 10, 15** (descreste in functie de iteratie)
     Numărul de iterații: **200**
     Metoda: **xgbTree**
   - **Acuratețea pentru fiecare bucket:** 87.14%, 86.14%, 86.84%, 86.21%, 85.42%
   - **Media acurateții: 86.35%**
   - **Varianța: 0.36**
8. **Retele Neurale Stacked**
   - **Implementarea:** Folosind Keras (python)
   - **Descriere:** 20 de modele sunt antrenate specializat pe clase. Outputul celor 20 de rețele se combină și se transmit unei noi rețele care face predicția finală.
   - **Parametri la modelele specializate pe clase:**
     Activare: **ReLU**, cu excepția ultimui layer unde este folosit **Sigmoid**
     Functia de loss: **BinaryCrossEntropy**
     Optimizator: **Adam**
     Numarul de epoci de antrenament: **12**
     Batch Size: **64**

Reteaua: **2** straturi ascunse fully connected (dimensiunile **156, 96**) toate urmate de **Batch Normalization**.

- **Parametri la modelul care face predicția**:
  Activare: **ReLU**, cu excepția ultimui layer unde este folosit **SoftMax**
  Funcția de loss: **CrossEntropy**
  Optimizator: **Adam**
  Numărul de epoci de antrenament: **10**
  Batch Size: **64**
- **Reteaua**: **3** straturi ascunse fully connected (dimensiunile **256, 128, 96**) toate urmate de **Batch Normalization** si **Dropout** de **0,1**.
- **Acuratețea pentru fiecare bucket:** 82.51%, 82.34%, 82.21%, 83.13%, 82.84%
- **Media acurateții: 82.6%**
- **Varianța: 0.11**

9. **Random Forest**
   - **Implementarea:** sklearn.ensemble.RandomForestClassifier (python)
   - **Parametri:**
     n_estimators: **70**
     Criteriul: **Entropia**
     Înălțimea Maxima: **13**
     min_impurity: **2.79316138e-04**
   - **Acuratețea pentru fiecare bucket:** 82.67%, 81.94%, 81.85%, 81.18%, 82.77%
   - **Media acurateții: 82.082%**
   - **Varianța: 0.34**