Grigoras Teodora
Grupa 334AA
14/12/2023

Clasificare cu algoritmul Random Forest

Detalii:
Tema: Clasificarea radiografiilor la plămâni in doua categorii (NORMALE & PNEUMONIE);
Set de date: 1874 fișiere in format JPEG : 1500 Antrenare, 187 Validare, 187 Testare ;
Algoritm folosit: Random Forest.

Descriere:

Setul de date este compus din 1874 de fișiere in format JPEG procurat de pe kaggle.com. Inițial setul de date era mult mai mare si disproporționat asa ca am fost nevoita sa il aduc la dimensiunile preferate. Am împărțit datele in setul de antrenare (train1), validare (val) si testare (test). Procentajul cantitativ este de 80-10-10.

Pentru antrenare am folosit algoritmul Random Forest. Inițial am încercat cu K-means, dar rezultatele performantelor erau destul de slabe. Cu Random Forest, algoritmul de clasificare are rezultate performante pe setul de validare: acuratețe 92%, precizie 91%, sensibilitate 95%, scor F1 93%.

La partea de testare am folosit un set de date diferit, neetichetate, pe care am aplicat algoritmul Random Forest. Rezultatele sunt satisfăcătoare. Am afișat vectorul test_prediction in care se afla etichetele prezise (0 pentru NORMAL si 1 pentru PNEUMONIE). Cum in folderul de test nu sunt separate cele doua cazuri si eu nu sunt un specialist in domeniul medical, am așezat categoriile in ordinea următoare pentru o înțelegere mai buna a rezultatelor: primele 93 NORMALE, ultimele 94 PNEUMONIE. Astfel, test_prediction trebuie sa aibă cat mai multe valori de 1 pe primele 93 de poziții si cat mai multe valori de 0 pe ultimele 94 de poziții.

Descrierea codului:

Bibliotecile folosite:

```
import os
import cv2
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
# from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

Grigoras Teodora
Grupa 334AA
14/12/2023

Funcțiile de încărcare a datelor (asemenea celor din laboratorul 2):

```
11   # Function to load train images from a folder and assign labels
12 > def load_train_images_from_folder(folder, target_shape=None): ···
35
36   # Function to load test images from a folder and assign labels
37 > def load_test_images_from_folder(folder, target_shape=None): ···
60
61 > def load_val_images_from_folder(folder, target_shape=None): ···
```

Calea folderelor de antrenare, validare si testare:

```
# Folder paths
data_folder = './train1'  # Folder with training data
test_folder = './test'  # Folder with test data
val_folder = './val'  # Folder with validation data
```

Încărcarea datelor de antrenare si validare:

```
# Load images and labels from the 'dataset' folder and resize them to (200, 200)
images, labels = load_train_images_from_folder(data_folder, target_shape=(250, 250))

# Load validation images and labels from the 'val' folder
val_images, val_labels = load_val_images_from_folder(val_folder, target_shape=(250, 250))

# Combine training and validation data
images += val_images
labels += val_labels
```

Modelarea datelor de antrenare si validare:

```
121   # Convert labels to binary (0 or 1)
122   labels_binary = [1 if label == 'NORMAL' else 0 for label in labels]
123
124   # Reshape the images and convert them to grayscale
125   image_data = [cv2.cvtColor(image, cv2.COLOR_BGR2GRAY).flatten() for image in images]
126
127   # Convert the list of 1D arrays to a 2D numpy array
128   image_data = np.array(image_data)
129
130   # Scale the data
131   scaler = StandardScaler()
132   scaled_data = scaler.fit_transform(image_data)
```

Grigoras Teodora
Grupa 334AA
14/12/2023

Încărcarea si modelarea datelor de test:

```
153  # Load TEST images from the 'test' folder
154  test_images, test_labels = load_test_images_from_folder(test_folder, target_shape=(250, 250))
155
156  # Convert labels to binary (0 or 1), reshape and convert to grayscale and scale dthe TEST data
157  test_labels_binary = [1 if label == 'NORMAL' else 0 for label in test_labels]
158  test_image_data = [cv2.cvtColor(image, cv2.COLOR_BGR2GRAY).flatten() for image in test_images]
159  test_image_data = np.array(test_image_data)
160  scaled_test_data = scaler.transform(test_image_data)
```

Antrenarea cu Random Forest si crearea vectorului de predicții pentru setul de validare:

```
162  # Split the data into training and validation sets
163  X_train, X_val, y_train, y_val = train_test_split(scaled_data, labels_binary, test_size=0.2, random_state=42)
164
165  # Train a Random Forest model
166  random_forest_model = RandomForestClassifier(random_state=42, n_estimators=100)
167  random_forest_model.fit(X_train, y_train)
168
169  # Predictions on the validation set
170  validation_predictions = random_forest_model.predict(X_val)
171
```

Evaluarea performantelor pe setul de validare:

```
243  # Evaluate performance on the validation set
244  accuracy_val = accuracy_score(y_val, validation_predictions)
245  precision_val = precision_score(y_val, validation_predictions)
246  recall_val = recall_score(y_val, validation_predictions, zero_division=1)
247  f1_val = f1_score(y_val, validation_predictions)
248  confusion_matrix_val = confusion_matrix(y_val, validation_predictions)
249
250  print("Performance on Validation Set:")
251  print(f"Accuracy: {accuracy_val:.4f}")
252  print(f"Precision: {precision_val:.4f}")
253  print(f"Recall: {recall_val:.4f}")
254  print(f"F1 Score: {f1_val:.4f}")
255  print("Confusion Matrix:")
256  print(confusion_matrix_val)
```

Grigoras Teodora
Grupa 334AA
14/12/2023

Aplicarea modelului pe setul de testare si crearea vectorului de predicții + plotarea imaginilor cu predicția lor:

```
# Predictions on the test set
test_predictions = random_forest_model.predict(scaled_test_data)
#Afisez vectorul de predictie
print("test_prediction: ")
print(test_predictions)

# Plotarea imaginilor din test cu labelul asociat (din 5 in 5 pt ca sunt multe(187))
for i, test_image in enumerate(test_images[::5]):

    if test_predictions[i] == 1:
        print(f"Test Image {i + 1} - ==NORMAL==")
    else:
        print(f"Test Image {i + 1} - ==PNEUMONIA==")

    # Visualize the test image with its predicted value
    plt.figure()
    plt.imshow(cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB))
    if test_predictions[i] == 1:
        plt.title(f"Test Image {i + 1} - ==NORMAL==")
    else:
        plt.title(f"Test Image {i + 1} - ==PNEUMONIA==")
    plt.axis('off')
    plt.show()
```

Evaluarea performantelor pe setul de testare:

```
283   # Evaluate performance on the test set
284   accuracy_test = accuracy_score(test_labels_binary, test_predictions)
285   precision_test = precision_score(test_labels_binary, test_predictions)
286   recall_test = recall_score(test_labels_binary, test_predictions, zero_division=1)
287   f1_test = f1_score(test_labels_binary, test_predictions)
288   confusion_matrix_test = confusion_matrix(test_labels_binary, test_predictions)
289
290   print("\nPerformance on Test Set:")
291   print(f"Accuracy: {accuracy_test:.4f}")
292   print(f"Precision: {precision_test:.4f}")
293   print(f"Recall: {recall_test:.4f}")
294   print(f"F1 Score: {f1_test:.4f}")
295   print("Confusion Matrix:")
296   print(confusion_matrix_test)
```

Grigoras Teodora
Grupa 334AA
14/12/2023

INTERPRETAREA REZULTATELOR:

- Rezultate pentru setul de VALIDARE:

```
Performance on Validation Set:
Accuracy: 0.9290
Precision: 0.9130
Recall: 0.9545
F1 Score: 0.9333
Confusion Matrix:
[[146  16]
 [  8 168]]
```

Se poate observa că performantele sunt destul de bune, aproape maxime.

- Rezultate pentru setul de TESTARE:

```
test_prediction:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
 0 0]
```

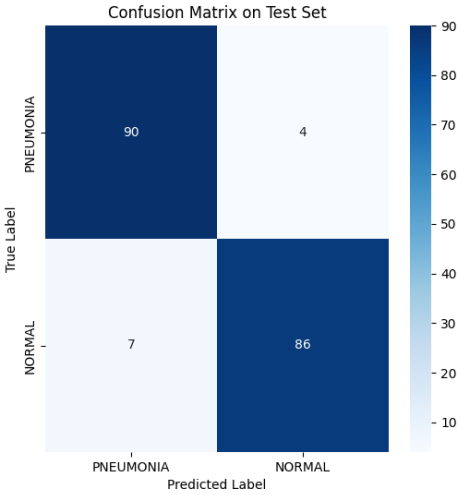
Se poate observa ca in vectorul test_prediction majoritatea primelor 93 de valori sunt 1 adică NORMAL si majoritatea ultimelor 94 de valori sunt 0 adică PNEUMONIE.

Primele 93 de valori: 7 valori eronate
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1

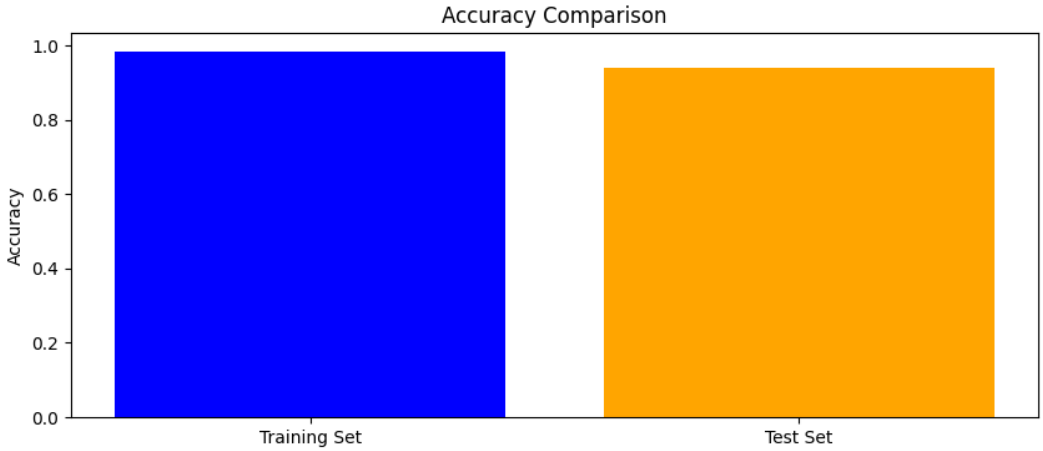
Ultimele 94 de valori: 4 valori eronate
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0

Rezulta :

- matricea de confuzie: [90 4
  7 86]
- acuratețea: 95,7 %
- precizia: 95,5%
- sensibilitatea: 92,4%
- specificitatea:95,7%
- scorul F1: 93,4%



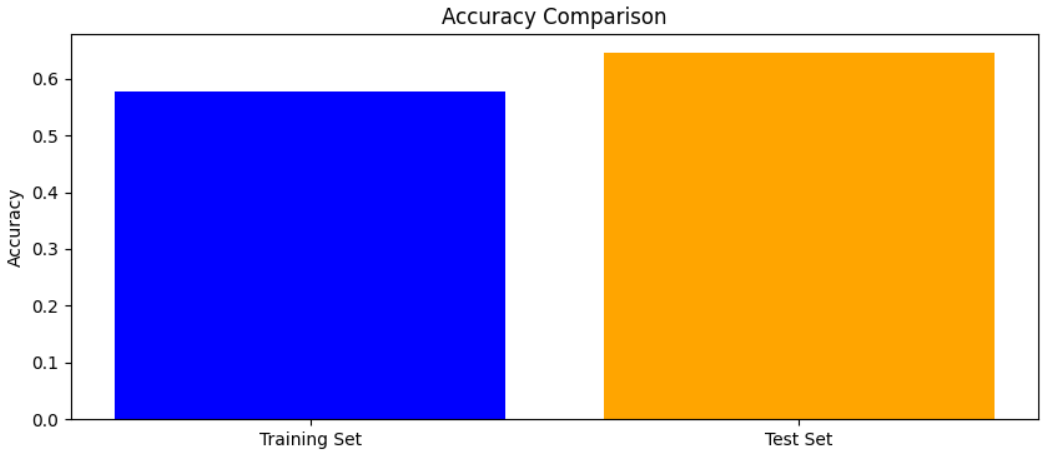
Grigoras Teodora
Grupa 334AA
14/12/2023
În diagrama de mai jos este afișat nivelul acurateței pe setul de antrenare si pe cel de testare. Se poate observa ca aceste valori sunt foarte ridicate (0.98 antrenare și 0.94 testare)



COMPARAREA REZULTATELOR CU K-MEANS

Am încercat rezolvarea problemei clasificării și cu algoritmul K-Means, dar rezultatele au fost semnificativ mai slabe.

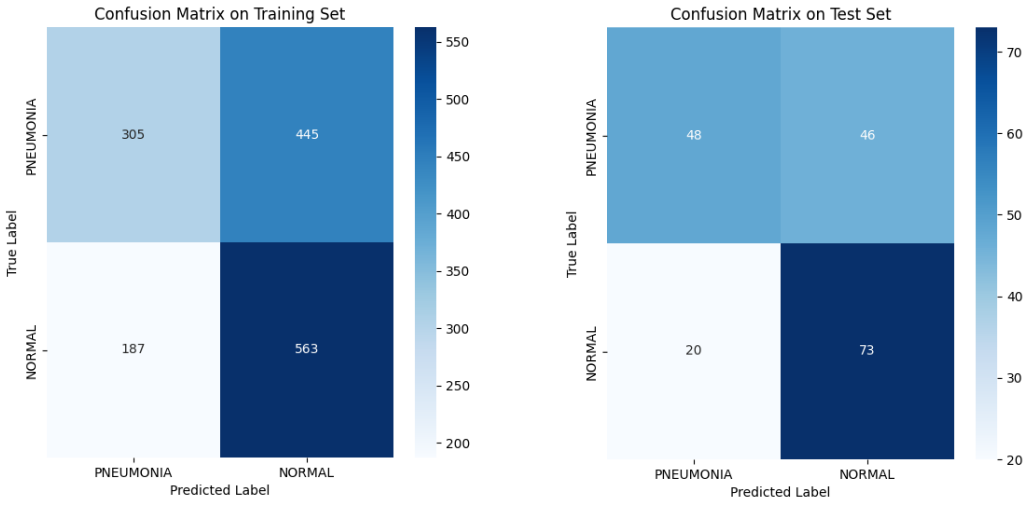
Mai jos se afla diagrama ce compara acuratețea antrenare-testare oferită de algoritmul k-Means.



Se poate observa că acestea sunt vizibil scăzute față de rezultatele cu Random Forest, mai precis o acuratețe de 0.57 la antrenare și 0.64 la testare.

Grigoras Teodora
Grupa 334AA
14/12/2023

Aceste rezultate se mai pot observa și din vizualizarea matricelor de confuzie



CONCLUZIE

Problema clasificării poate depinde de diferiți factori precum dimensiunea datelor, calitatea acestora, preprocesarea, dar și algoritmul ales. Din rezultatele obținute în urma utilizării celor doi algoritmi (k-Means și Random Forest) se pot observa clar diferențele uriașe date de alegerea algoritmului. Menționez că aceștia au fost aplicați pe același set de date.

Problema clasificării radiografiilor pulmonare este una destul de dificilă care necesită, indiferent de algoritm sau de date, un expert bine pregătit în domeniul medical, dar consider că aceste metode de clasificare pot aduce o mare mână de ajutor în eficientizarea procesului de diagnosticare, fiind ușor de implementat si neavând nevoia de o capacitate mare de procesare.