Proiect 3 TIA
Clasificarea culorilor

- Drăgan Pavel 331AA -

• Obiectivul proiectului

Acest proiect vine ca o completare a laboratorului 4. Urmează să comentez rezultatele obținute în urma rulării codului din laborator. Apoi, voi prezenta și modificările aduse setului de date și la ce concluzii am ajuns în urma acestui experiment.

• Obținerea și organizarea setului de date

Setul de date:

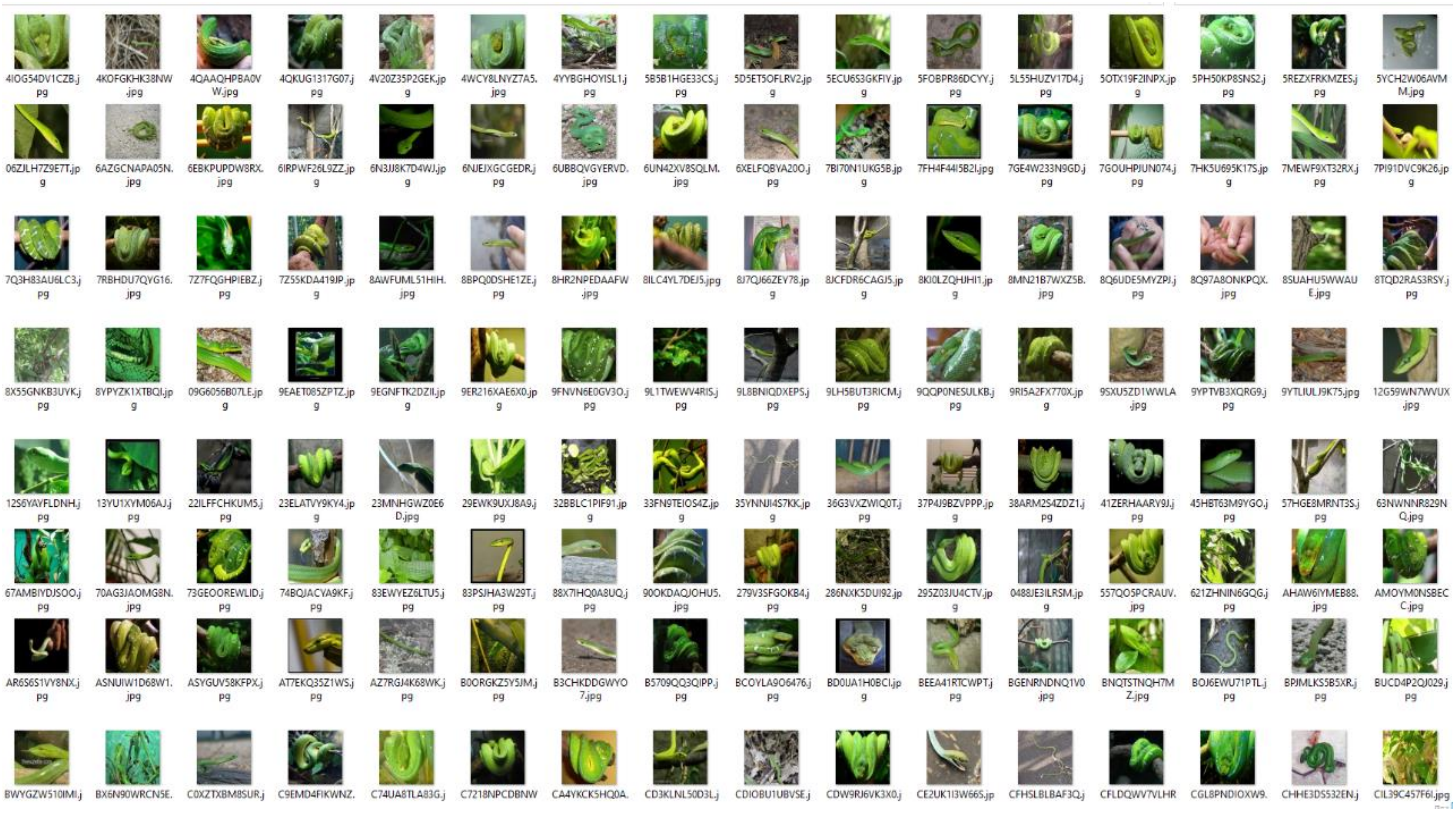
Setul de date folosit inițial este setul de date din laborator (set de date făcut de Google) împărțit în imagini roșii și albastre. O să fac o testare a algoritmilor prima dată cu setul de date nemodificat după care voi adăuga propriul meu folder cu imagini verzi pentru a vedea cum afectează asta performanța algoritmului.

Prelucrarea datelor și problemele întâmpinate:

Am ales inițial să folosesc imagini din setul de date Google-512, un set de date conceput în mod special pentru modele care vor să învețe să recunoască obiecte de o anumită culoare. O problemă întâmpinată la acest set de date a fost faptul că formatul imaginilor nu era uniform, în același set de date fiind prezente atât imagini jpg, png și gif. Acest lucru a dus la câteva erori în program așa că am decis să mă îndrept spre alt set de date găsit care conținea imagini cu șerpi verzi. Aceste imagini aveau în mod clar preponderent verde și mai important aveau același format deci am decis să merg mai departe cu el. De asemenea am redus dimensiunea setului de date la 430 de imagini în rând cu celelalte două foldere

331AA Drăgan Pavel

Un overview al imaginilor din noul set de date:



• Algoritmul utilizat

Etapa de cercetare:

Similar cu tema 2, algoritmii utilizați în această temă sunt algoritmii prezentați la laborator, și anume k-Nearest Neighbors (k-NN) și algoritmul Naive Bayes. Ambii algoritmi utilizați în învățarea supervizată pentru k-NN făcând în timpul fazei de predicție, clasificarea unui punct de date noi este determinată pe baza claselor majoritare ale vecinilor săi cei mai apropiați în spațiul caracteristicilor. Procedura de antrenare a algoritmului Naive Bayes implică calcularea probabilităților claselor și a probabilităților condiționate ale caracteristicilor date clasei. În timpul predicției, se utilizează Teorema lui Bayes pentru a calcula probabilitățile fiecărei clase date caracteristicilor, iar clasa cu cea mai mare probabilitate este atribuită noului punct de date.

Implementarea propriu-zisă a celor doi algoritmi a fost făcută în același program Python similar cu implementarea prezentată la ora de laborator.

Biblioteci Python utilizate:

- os: Furnizează metode de interacțiune cu sistemul de operare, precum citirea numelor de fișiere și directoare.

- numpy (np): Folosit pentru operații numerice și manipularea matricelor.

- sklearn: O bibliotecă de învățare automată care conține instrumente pentru clasificare, regresie, clusterizare, etc.

- skimage.io: Utilizată pentru citirea imaginilor.

- skimage.transform: Furnizează funcții de procesare a imaginilor, cum ar fi redimensionarea.

- matplotlib.pyplot (plt): Utilizată pentru crearea de vizualizări.

- pandas (pd): Utilizată pentru manipularea și analiza datelor.

Programul dezvoltat:

- Funcția de Încărcare a Setului de Date (load_dataset): Citește imaginile dintr-un director specificat și redimensionează fiecare imagine la o dimensiune specificată (64x64 pixeli în acest caz). Transformă imaginea într-un format liniar (aplatizează), creează o mapare a numelui claselor către etichete numerice, returnează imagini aplatizate, etichetele corespunzătoare și maparea claselor.

- Funcția de Salvare a Rezultatelor într-un Fișier CSV (save_results): Salvează etichetele reale, etichetele prezise și numele claselor corespunzătoare într-un fișier CSV și creează un director pentru rezultate dacă nu există.

- Funcția de Salvare a Imaginilor de Test cu Predicții (save_test_images): Salvează imaginile de test împreună cu etichetele reale și prezise și salvează, de asemenea, grafice ale distribuției probabilităților pentru fiecare imagine de test.

- Încărcarea Setului de Date: Specifică directorul care conține setul de date (dataset_folder) și apelează funcția load_dataset pentru a încărca imagini, etichete și maparea claselor.

- Împărțirea Setului de Date: Împarte setul de date în setul de antrenare și setul de test folosind train_test_split din sklearn.

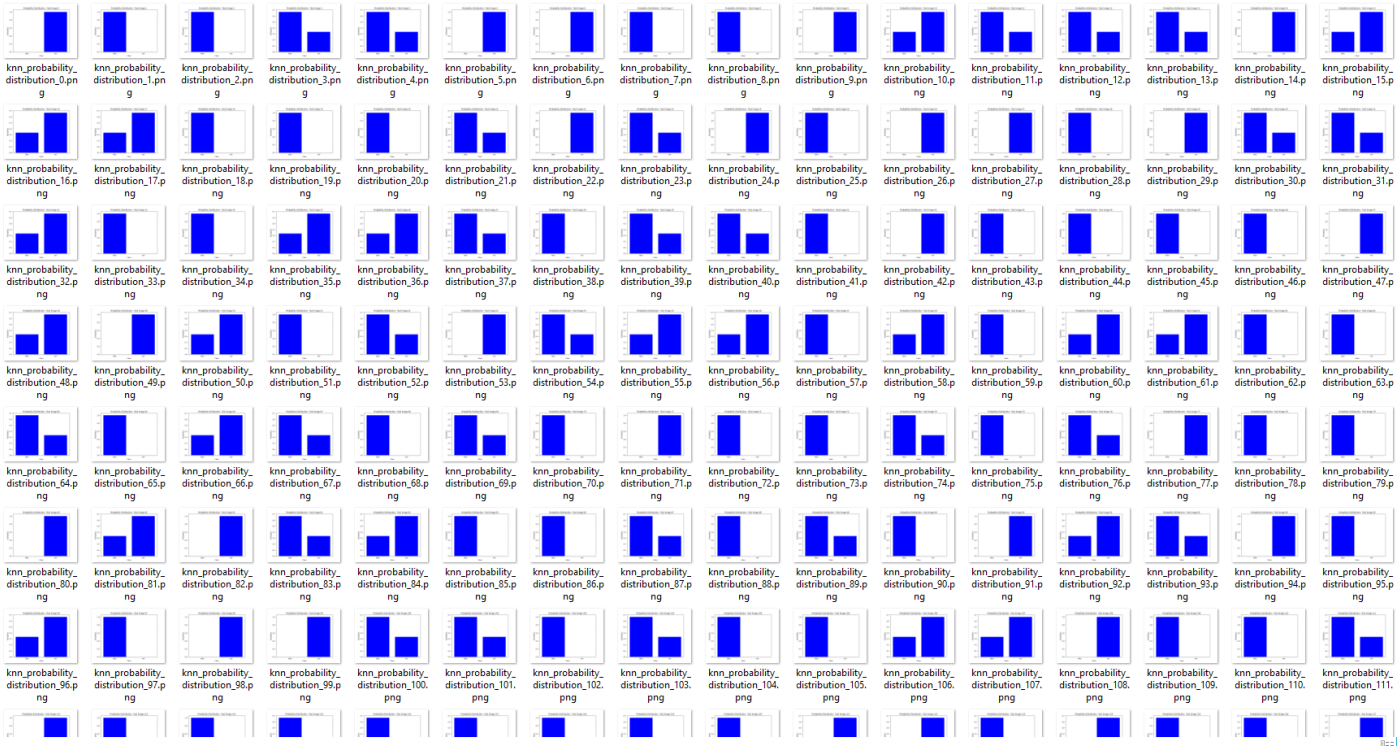
- k-Nearest Neighbors (k-NN): Inițializează un clasificator k-NN cu n_neighbors=3, antrenează modelul folosind setul de antrenare (X_train, y_train), face predicții pe setul de test (X_test) și salvează rezultatele și creează vizualizări.

- Naive Bayes (Gaussian Naive Bayes): Inițializează un clasificator Gaussian Naive Bayes, antrenează modelul folosind setul de antrenare, face predicții pe setul de test și salvează rezultatele și creează vizualizări.

• Rezultate

Pentru a vizualiza cât mai bine rezultatele programului am prelucrat fișierele cvs obținute, calculând câte predicții corecte au fost făcute și ce procentaj reprezintă acestea din toate predicțiile făcute. De asemenea am pus aceste date în câteva grafice pentru o mai bună vizualizare.

Rezultatele obținute pentru k-NN:

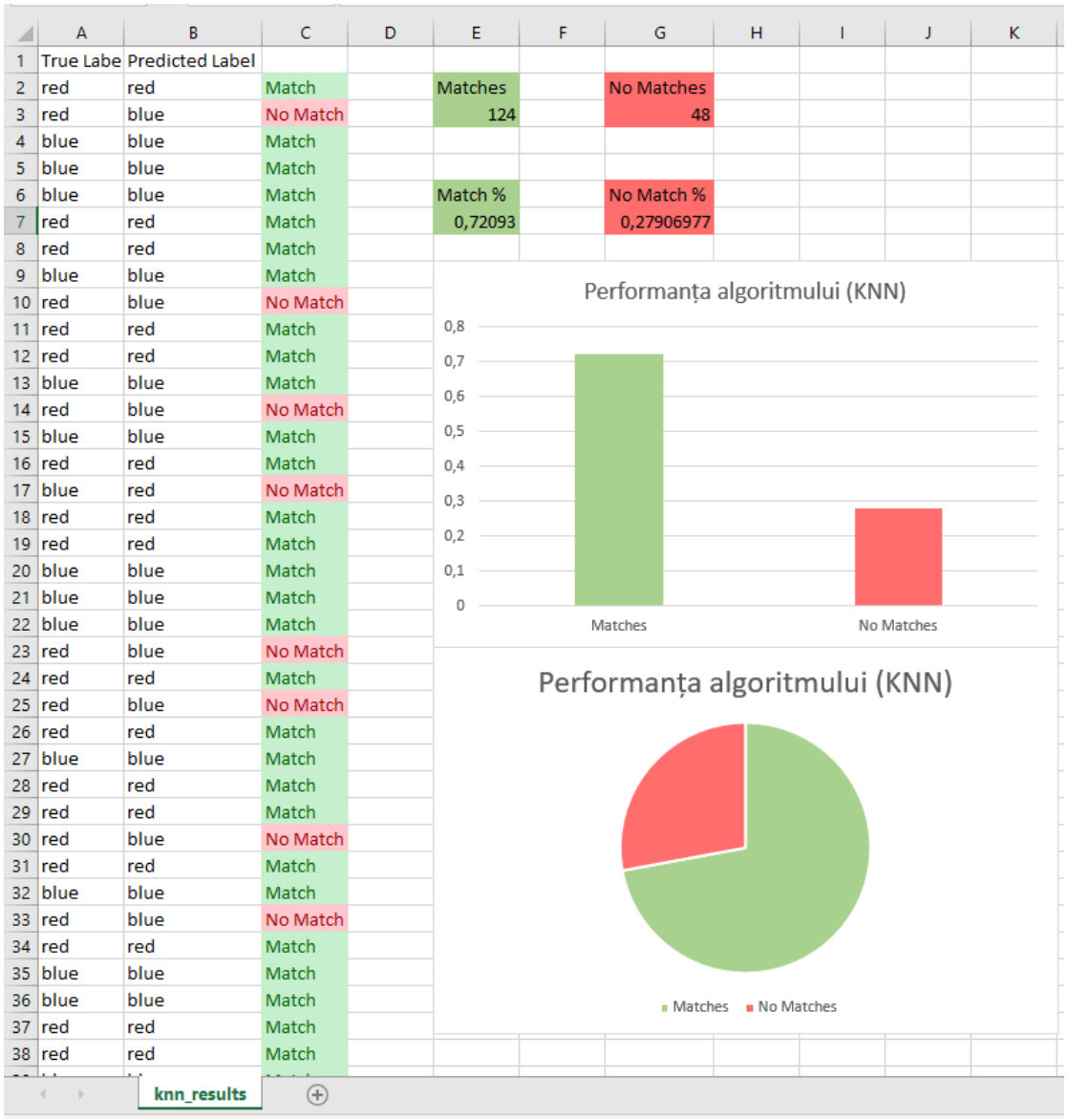


Pentru început observăm că algoritmul k-NN nu a reușit decât în puține cazuri să conveargă pe o singură predicție, existând multe cazuri în care ambele culori aveau probabilități ridicate.

331AA Drăgan Pavel

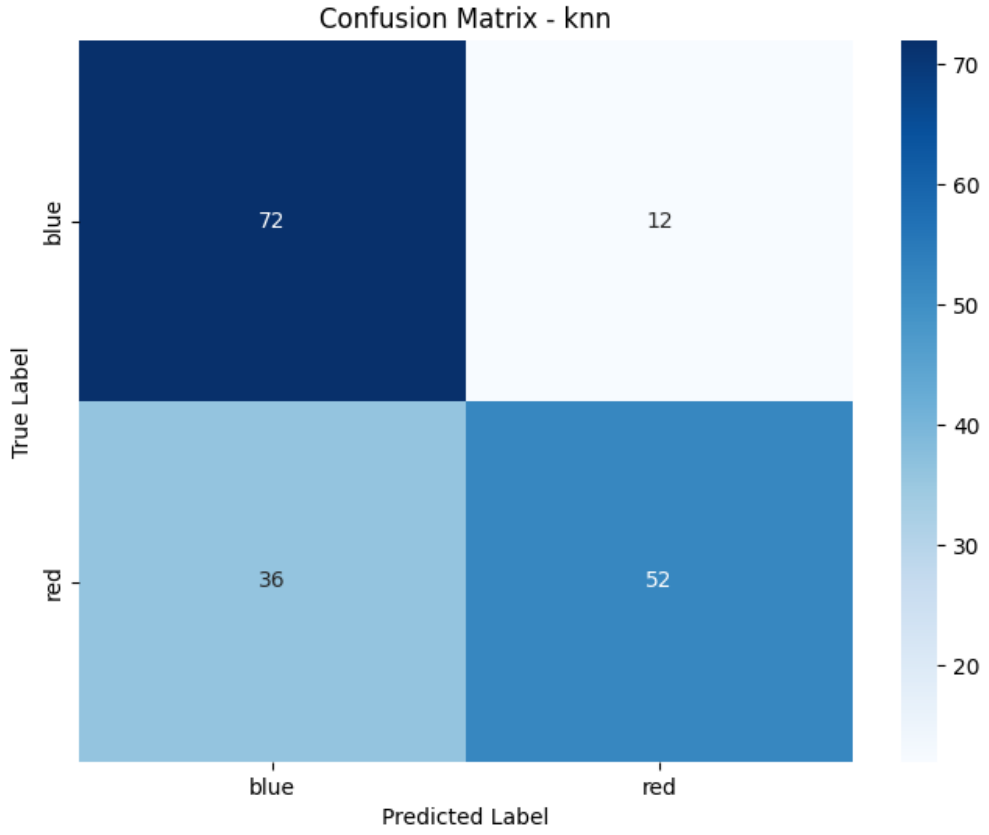
Performanța algoritmului k-NN:

Performanța algoritmului pentru această problemă de sortare a obiectelor după culori este cu mult mai ridicată decât performanța obținută la tema doi ceea ce nu face decât să susțină concluzia că algoritmii folosiți nu erau cei mai potriviți pentru task-ul ales.



	A	B	C	D	E	F	G	H	I	J	K
1	True Labe	Predicted Label									
2	red	red	Match		Matches		No Matches				
3	red	blue	No Match		124		48				
4	blue	blue	Match								
5	blue	blue	Match								
6	blue	blue	Match		Match %		No Match %				
7	red	red	Match		0,72093		0,27906977				
8	red	red	Match								
9	blue	blue	Match								
10	red	blue	No Match								
11	red	red	Match								
12	red	red	Match								
13	blue	blue	Match								
14	red	blue	No Match								
15	blue	blue	Match								
16	red	red	Match								
17	blue	red	No Match								
18	red	red	Match								
19	red	red	Match								
20	blue	blue	Match								
21	blue	blue	Match								
22	blue	blue	Match								
23	red	blue	No Match								
24	red	red	Match								
25	red	blue	No Match								
26	red	red	Match								
27	blue	blue	Match								
28	red	red	Match								
29	red	red	Match								
30	red	blue	No Match								
31	red	red	Match								
32	blue	blue	Match								
33	red	blue	No Match								
34	red	red	Match								
35	blue	blue	Match								
36	blue	blue	Match								
37	red	red	Match								
38	red	red	Match								

Am modificat codul pentru a putea obține și o matrice de convoluție și mai mulți parametrii care să facă mai ușoară interpretarea rezultatelor.



2knn_metrics (1).txt - Notepad

File Edit Format View Help

```
Confusion Matrix:
[[72 12]
 [36 52]]

Accuracy: 0.72

Classification Report:
{'blue': {'precision': 0.6666666666666666, 'recall': 0.8571428571428571, 'f1-score': 0.75, 'support': 84.0},

 'red': {'precision': 0.8125, 'recall': 0.5909090909090909, 'f1-score': 0.6842105263157896, 'support': 88.0},

 'accuracy': 0.7209302325581395,

 'macro avg': {'precision': 0.7395833333333333, 'recall': 0.724025974025974, 'f1-score': 0.7171052631578948, 'support': 172.0},

 'weighted avg': {'precision': 0.7412790697674418, 'recall': 0.7209302325581395, 'f1-score': 0.7163402692778458, 'support': 172.0}}
```

Matrice de confuzie:

- Pentru clasa 'blue':

72 de exemple au fost clasificate corect ca 'blue' (True Positive - TP).

12 exemple au fost clasificate greșit ca 'red' (False Positive - FP).

- Pentru clasa 'red':

36 de exemple au fost clasificate greșit ca 'blue' (False Negative - FN).

52 de exemple au fost clasificate corect ca 'red' (True Negative - TN).

Accuracy (Acuratețe):

Acuratețea este proporția totală a exemplelor clasificate corect, în acest caz, 0.72 sau 72%.

Acuratețea se calculează folosind formula: (TP + TN) / (TP + TN + FP + FN).

Raportul de clasificare:

- Pentru clasa 'blue':

Precizia (Precision): 0.67 - 67% dintre exemplele clasificate ca 'blue' au fost corecte.

Recall (Sensibilitate): 0.86 - 86% dintre exemplele reale de 'blue' au fost clasificate corect.

F1-score: 0.75 - o măsură echilibrată între precizie și recall.

Support: 84 de exemple de 'blue' în setul de date.

- Pentru clasa 'red':

Precizia (Precision): 0.81 - 81% dintre exemplele clasificate ca 'red' au fost corecte.

Recall (Sensibilitate): 0.59 - 59% dintre exemplele reale de 'red' au fost clasificate corect.

F1-score: 0.68 - o măsură echilibrată între precizie și recall.

Support: 88 de exemple de 'red' în setul de date.

Acuratețe Generală:

Acuratețea generală este 0.72, ceea ce indică o performanță relativ bună a modelului pentru cele două clase.

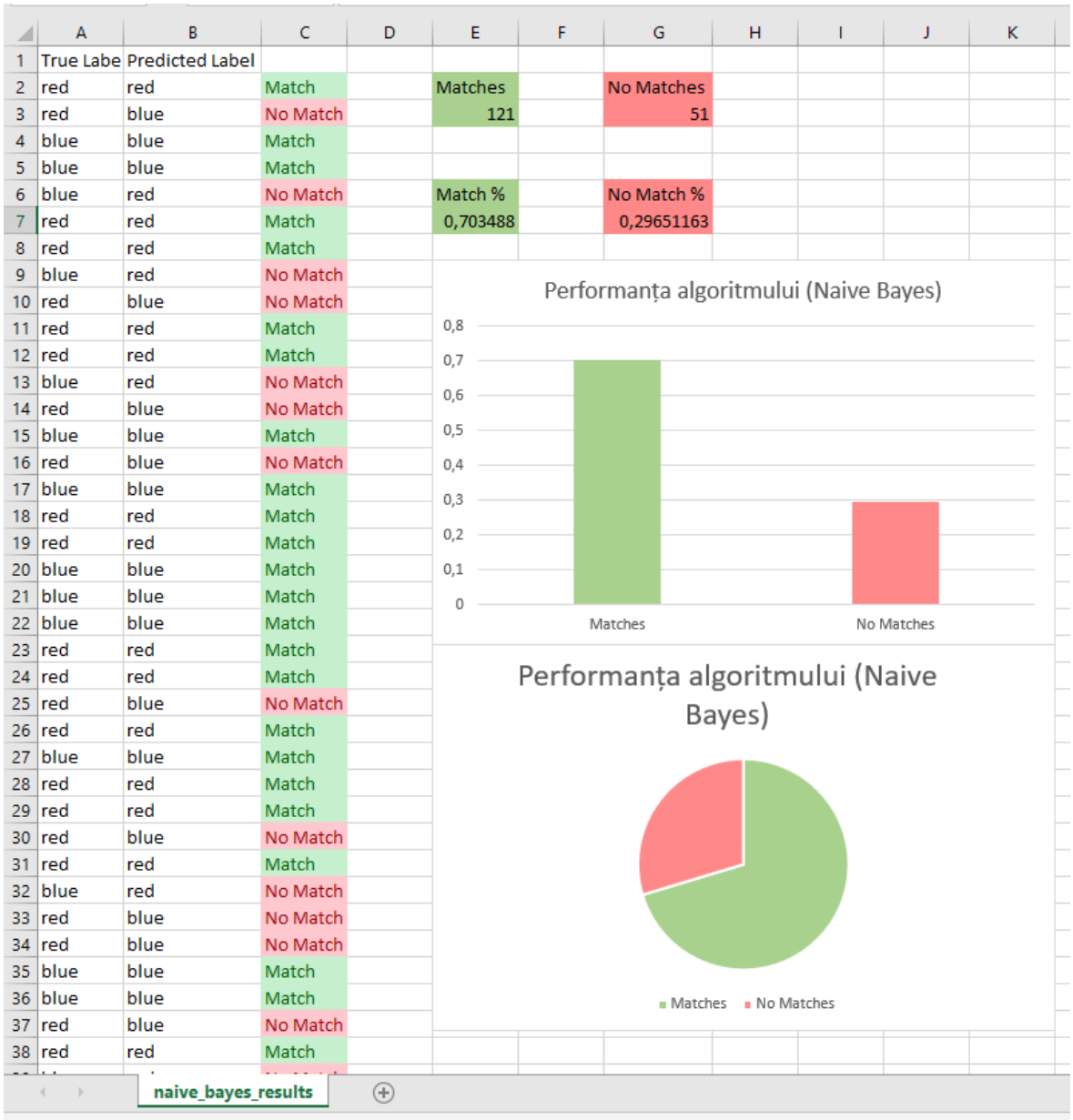
331AA Drăgan Pavel

Rezultatele obținute pentru Naive Bayes:

Observăm că spre deosebire de k-NN acest algoritm a dat rezultate mult mai „definitive”,cazurile în care o imagine să aibă probabilitatea de a conține două culori fiind mult mai rară.

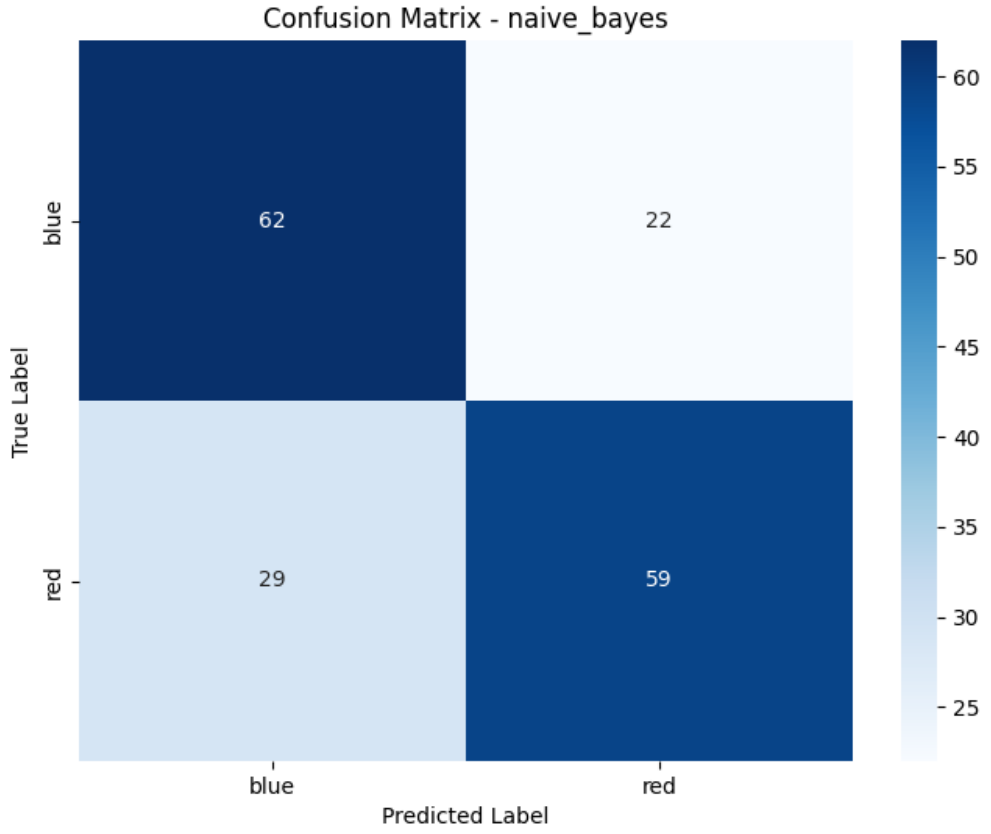
Performanța algoritmului Naive Bayes:

În mod similar rezultatelor înregistrate la tema 2, algoritmul Naïve Bayes a obținut rezultate similar cu cele ale algoritmului precedent (acuratețe aproximativ 70% vs 72% pentru k-NN).



	A	B	C
1	True Labe	Predicted Label	
2	red	red	Match
3	red	blue	No Match
4	blue	blue	Match
5	blue	blue	Match
6	blue	red	No Match
7	red	red	Match
8	red	red	Match
9	blue	red	No Match
10	red	blue	No Match
11	red	red	Match
12	red	red	Match
13	blue	red	No Match
14	red	blue	No Match
15	blue	blue	Match
16	red	blue	No Match
17	blue	blue	Match
18	red	red	Match
19	red	red	Match
20	blue	blue	Match
21	blue	blue	Match
22	blue	blue	Match
23	red	red	Match
24	red	red	Match
25	red	blue	No Match
26	red	red	Match
27	blue	blue	Match
28	red	red	Match
29	red	red	Match
30	red	blue	No Match
31	red	red	Match
32	blue	red	No Match
33	red	blue	No Match
34	red	blue	No Match
35	blue	blue	Match
36	blue	blue	Match
37	red	blue	No Match
38	red	red	Match

E	G
Matches	No Matches
121	51
Match %	No Match %
0,703488	0,29651163



*2naive (2).txt - Notepad

File Edit Format View Help

```
Confusion Matrix:
[[62 22]
 [29 59]]

Accuracy: 0.70

Classification Report:
{'blue': {'precision': 0.6813186813186813, 'recall': 0.7380952380952381, 'f1-score': 0.7085714285714286, 'support': 84.0},

 'red': {'precision': 0.7283950617283951, 'recall': 0.6704545454545454, 'f1-score': 0.6982248520710058, 'support': 88.0},

 'accuracy': 0.7034883720930233,

 'macro avg': {'precision': 0.7048568715235382, 'recall': 0.7042748917748918, 'f1-score': 0.7033981403212173, 'support': 172.0},

 'weighted avg': {'precision': 0.7054042712957442, 'recall': 0.7034883720930233, 'f1-score': 0.7032778312921425, 'support': 172.0}}
```

Matrice de confuzie:

- Pentru clasa 'blue':

62 de exemple au fost clasificate corect ca 'blue' (TP).

22 de exemple au fost clasificate greșit ca 'red' (FP).

- Pentru clasa 'red':

29 de exemple au fost clasificate greșit ca 'blue' (FN).

59 de exemple au fost clasificate corect ca 'red' (TN).

Accuracy (Acuratețe):

Acuratețea este 0.70 sau 70%, indicând proporția totală a exemplelor clasificate corect.

Se calculează folosind formula: (TP + TN) / (TP + TN + FP + FN).

Raportul de clasificare:

- Pentru clasa 'blue':

Precizia (Precision): 0.68 - 68% dintre exemplele clasificate ca 'blue' au fost corecte.

Recall (Sensibilitate): 0.74 - 74% dintre exemplele reale de 'blue' au fost clasificate corect.

F1-score: 0.71 - o măsură echilibrată între precizie și recall.

Support: 84 de exemple de 'blue' în setul de date.

- Pentru clasa 'red':

Precizia (Precision): 0.73 - 73% dintre exemplele clasificate ca 'red' au fost corecte.

Recall (Sensibilitate): 0.67 - 67% dintre exemplele reale de 'red' au fost clasificate corect.

F1-score: 0.70 - o măsură echilibrată între precizie și recall.

Support: 88 de exemple de 'red' în setul de date.

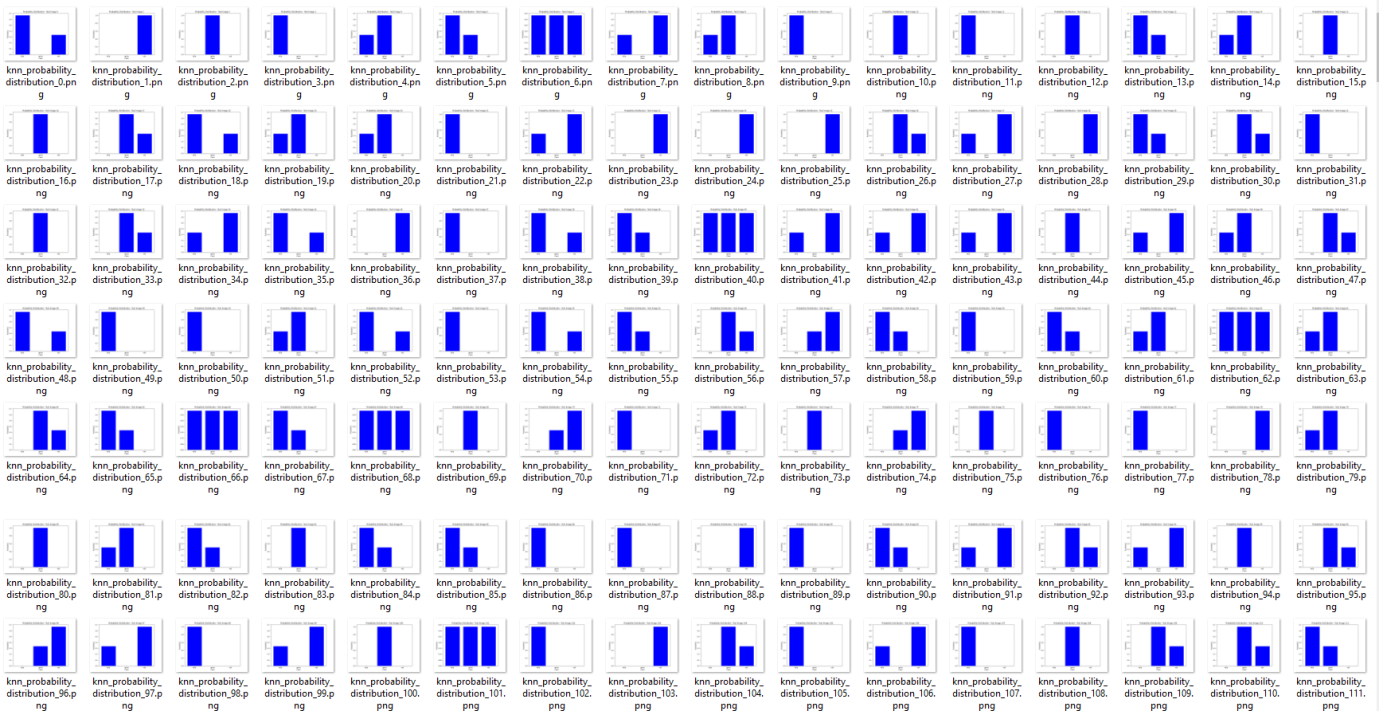
Acuratețe generală este 0.70, indicând o performanță relativ bună a modelului pentru cele două clase.

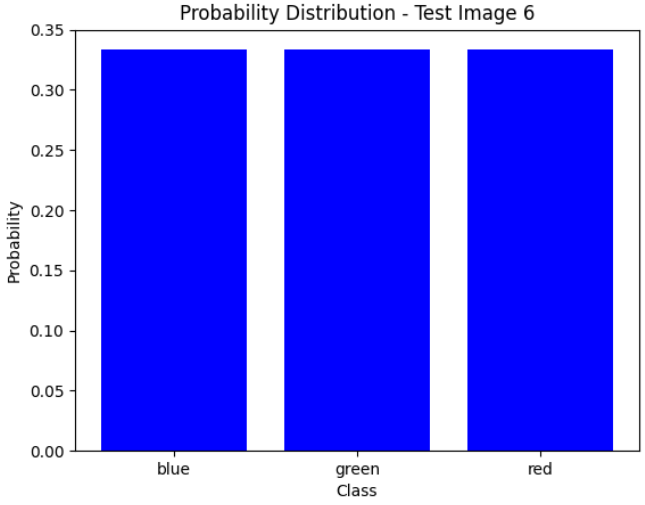
• Modificări

Urmează să expun și rezultatele obținute după adăugaraea folderului cu imagini verzi.

Rezultatele obținute pentru k-NN cu trei culori:

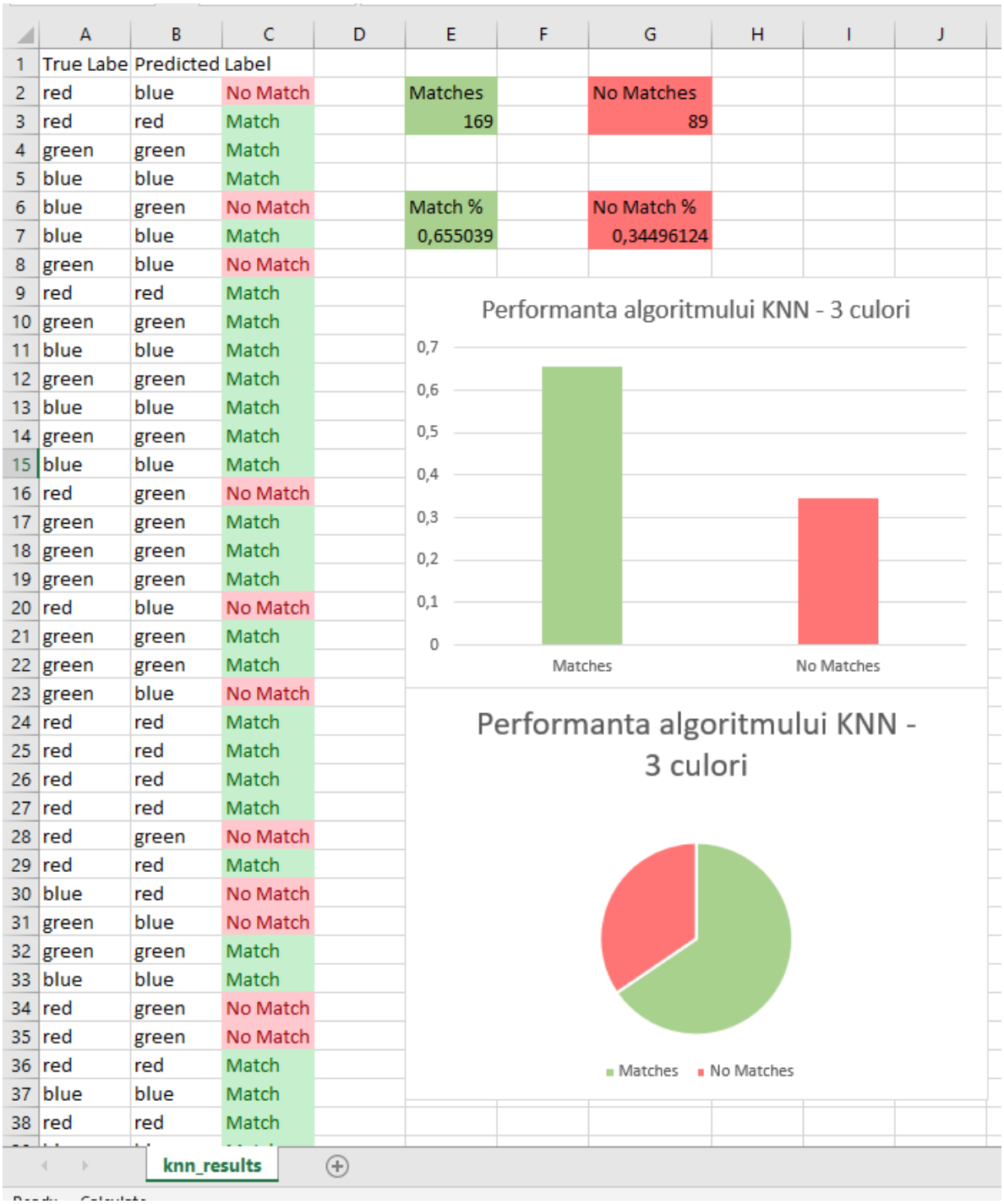
Am obținut rezultate similar cu cele de la doar două culori însă acum există mai multe cazuri în care probabilitatea este egală și maximă pentru toate cele trei culori, lucru care nu era așa de des întâlnit în cazul anterior.

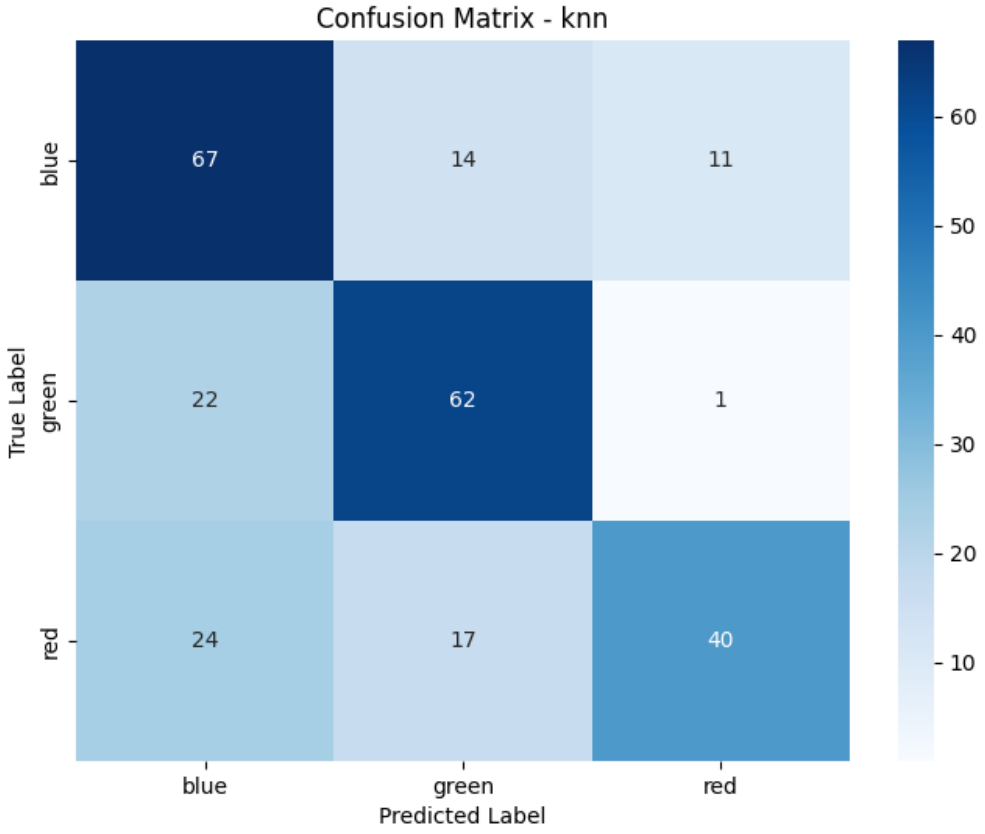




Performanța algoritmului k-NN cu trei culori:

Observăm o ușoară scădere a performanței pentru varianta cu 3 culori de la o acuratețe a predicțiilor de aproximativ 72% la una de 65%. Deși mică, această scădere arată faptul că adăugarea unui noi culori a avut un impact asupra performanței. Acest lucru poate fi un factor în plus și pentru rezultatele mai proaste obținute la tema doi deoarece acolo pe lângă faptul că sarcina de a diferenția expresiile în sine era ușor mai dificilă mai aveam și 7 expresii de clasificat și nu e greu de imaginat ca această scădere a performanței odată cu mărirea obiectelor de clasificat să fie o scădere exponențială.





3knn_metrics.txt - Notepad

File Edit Format View Help

```
Confusion Matrix:
[[67 14 11]
 [22 62  1]
 [24 17 40]]

Accuracy: 0.66

Classification Report:
{'blue': {'precision': 0.5929203539823009, 'recall': 0.7282608695652174, 'f1-score': 0.6536585365853659, 'support': 92.0},

 'green': {'precision': 0.6666666666666666, 'recall': 0.7294117647058823, 'f1-score': 0.696629213483146, 'support': 85.0},

 'red': {'precision': 0.7692307692307693, 'recall': 0.49382716049382713, 'f1-score': 0.6015037593984963, 'support': 81.0},

 'accuracy': 0.6550387596899225,

'macro avg': {'precision': 0.6762725966265789, 'recall': 0.650499931588309, 'f1-score': 0.6505971698223361, 'support': 258.0},

'weighted avg': {'precision': 0.6725698896927546, 'recall': 0.6550387596899225, 'f1-score': 0.651441368306974, 'support': 258.0}}
```

Matrice de confuzie:

- Pentru clasa 'blue':

67 de exemple au fost clasificate corect ca 'blue' (TP).

14 exemple au fost clasificate greșit ca 'green' (FP).

11 exemple au fost clasificate greșit ca 'red' (FP).

- Pentru clasa 'green':

22 de exemple au fost clasificate greșit ca 'blue' (FN).

62 de exemple au fost clasificate corect ca 'green' (TN).

1 exemplu a fost clasificat greșit ca 'red' (FP).

- Pentru clasa 'red':

24 de exemple au fost clasificate greșit ca 'blue' (FN).

17 exemple au fost clasificate greșit ca 'green' (FN).

40 de exemple au fost clasificate corect ca 'red' (TN).

Accuracy (Acuratețe):

Acuratețea este 0.66 sau 66%, indicând proporția totală a exemplelor clasificate corect.

Se calculează folosind formula: (TP + TN) / (TP + TN + FP + FN).

Raportul de clasificare:

- Pentru clasa 'blue':

Precizia (Precision): 0.59 - 59% dintre exemplele clasificate ca 'blue' au fost corecte.

Recall (Sensibilitate): 0.73 - 73% dintre exemplele reale de 'blue' au fost clasificate corect.

F1-score: 0.65 - o măsură echilibrată între precizie și recall.

Support: 92 de exemple de 'blue' în setul de date.

- Pentru clasa 'green':

Precizia (Precision): 0.67 - 67% dintre exemplele clasificate ca 'green' au fost corecte.

Recall (Sensibilitate): 0.73 - 73% dintre exemplele reale de 'green' au fost clasificate corect.

F1-score: 0.70 - o măsură echilibrată între precizie și recall.

Support: 85 de exemple de 'green' în setul de date.

- Pentru clasa 'red':

Precizia (Precision): 0.77 - 77% dintre exemplele clasificate ca 'red' au fost corecte.

Recall (Sensibilitate): 0.49 - 49% dintre exemplele reale de 'red' au fost clasificate corect.

F1-score: 0.60 - o măsură echilibrată între precizie și recall.

Support: 81 de exemple de 'red' în setul de date.

Acuratețe generală:

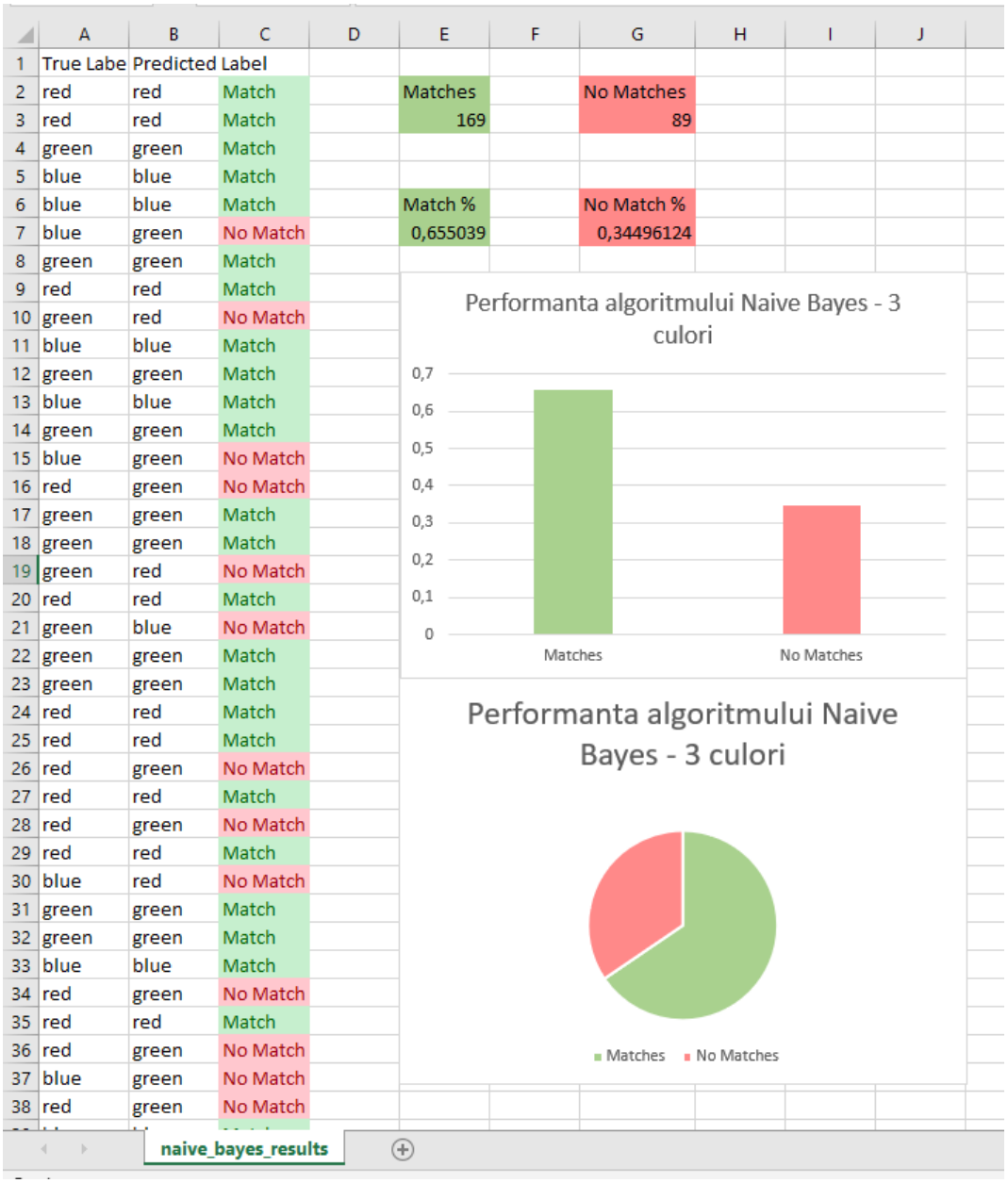
Acuratețea generală este 0.66, indicând o performanță moderată a modelului pentru cele trei clase.

Rezultatele obținute pentru Naive Bayes cu trei culori:

Observăm aceleași caracteristici ale clasificării prezente și la varianta cu doar două culori.

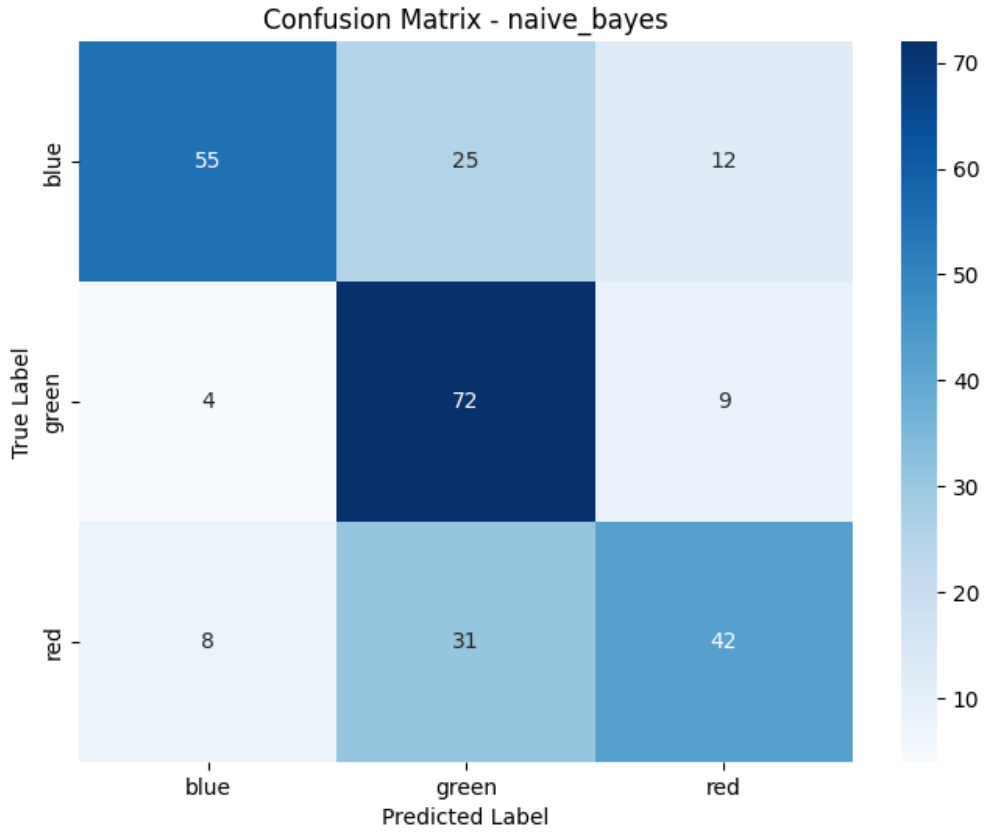
Performanța algoritmului Naive Bayes: cu trei culori

Spre deosebire de primul caz cu două culori, acum performanța algoritmului nu a scăzut deloc față de cea a algoritmului k-NN cu 3 culori însă observăm totuși și aici o scădere de la 70% la aproximativ 65%.



	A	B	C
1	True Labe	Predicted Label	
2	red	red	Match
3	red	red	Match
4	green	green	Match
5	blue	blue	Match
6	blue	blue	Match
7	blue	green	No Match
8	green	green	Match
9	red	red	Match
10	green	red	No Match
11	blue	blue	Match
12	green	green	Match
13	blue	blue	Match
14	green	green	Match
15	blue	green	No Match
16	red	green	No Match
17	green	green	Match
18	green	green	Match
19	green	red	No Match
20	red	red	Match
21	green	blue	No Match
22	green	green	Match
23	green	green	Match
24	red	red	Match
25	red	red	Match
26	red	green	No Match
27	red	red	Match
28	red	green	No Match
29	red	red	Match
30	blue	red	No Match
31	green	green	Match
32	green	green	Match
33	blue	blue	Match
34	red	green	No Match
35	red	red	Match
36	red	green	No Match
37	blue	green	No Match
38	red	green	No Match

E	G
Matches	No Matches
169	89
Match %	No Match %
0,655039	0,34496124



3naive_bayes_metrics.txt - Notepad

File Edit Format View Help

```
Confusion Matrix:
[[55 25 12]
 [ 4 72  9]
 [ 8 31 42]]

Accuracy: 0.66

Classification Report:
{'blue': {'precision': 0.8208955223880597, 'recall': 0.5978260869565217, 'f1-score': 0.6918238993710691, 'support': 92.0},

 'green': {'precision': 0.5625, 'recall': 0.8470588235294118, 'f1-score': 0.6760563380281689, 'support': 85.0},

 'red': {'precision': 0.6666666666666666, 'recall': 0.5185185185185185, 'f1-score': 0.5833333333333334, 'support': 81.0},

 'accuracy': 0.6550387596899225,

 'macro avg': {'precision': 0.6833540630182421, 'recall': 0.6544678096681507, 'f1-score': 0.6504045235775239, 'support': 258.0},

 'weighted avg': {'precision': 0.6873445273631841, 'recall': 0.6550387596899225, 'f1-score': 0.6525681685059408, 'support': 258.0}}
```

331AA Drăgan Pavel

Matrice de confuzie:

- Pentru clasa 'blue':

55 de exemple au fost clasificate corect ca 'blue' (TP).

25 exemple au fost clasificate greșit ca 'green' (FP).

12 exemple au fost clasificate greșit ca 'red' (FP).

- Pentru clasa 'green':

4 exemple au fost clasificate greșit ca 'blue' (FN).

72 de exemple au fost clasificate corect ca 'green' (TN).

9 exemple au fost clasificate greșit ca 'red' (FP).

- Pentru clasa 'red':

8 exemple au fost clasificate greșit ca 'blue' (FN).

31 de exemple au fost clasificate greșit ca 'green' (FN).

42 de exemple au fost clasificate corect ca 'red' (TN).

Accuracy (Acuratețe):

Acuratețea este 0.66 sau 66%, indicând proporția totală a exemplelor clasificate corect.

Se calculează folosind formula: (TP + TN) / (TP + TN + FP + FN).

Raportul de clasificare:

- Pentru clasa 'blue':

Precizia (Precision): 0.82 - 82% dintre exemplele clasificate ca 'blue' au fost corecte.

Recall (Sensibilitate): 0.60 - 60% dintre exemplele reale de 'blue' au fost clasificate corect.

F1-score: 0.69 - o măsură echilibrată între precizie și recall.

Support: 92 de exemple de 'blue' în setul de date.

331AA Drăgan Pavel

- Pentru clasa 'green':

Precizia (Precision): 0.56 - 56% dintre exemplele clasificate ca 'green' au fost corecte.

Recall (Sensibilitate): 0.85 - 85% dintre exemplele reale de 'green' au fost clasificate corect.

F1-score: 0.68 - o măsură echilibrată între precizie și recall.

Support: 85 de exemple de 'green' în setul de date.

- Pentru clasa 'red':

Precizia (Precision): 0.67 - 67% dintre exemplele clasificate ca 'red' au fost corecte.

Recall (Sensibilitate): 0.52 - 52% dintre exemplele reale de 'red' au fost clasificate corect.

F1-score: 0.58 - o măsură echilibrată între precizie și recall.

Support: 81 de exemple de 'red' în setul de date.

Acuratețe generală:

Acuratețea generală este 0.66, indicând o performanță moderată a modelului pentru cele trei clase.

• Concluzii și observații

În primul rând performanța obținută în această temă a fost mult mai ridicată decât cea obținută în tema 2. Acest lucru se datorează algoritmului mai potrivit pentru task-ul dat și a setului de date mai contrastant care a facilitat o antrenare și testare mai optimă.

Deși nu au fost înregistrate scăderi extrem de mari în performanță ele au fost totuși prezente ceea ce oferă încă un motiv; așa cum am mai menționat, pentru performanța scăzută înregistrată în tema 2. Am înregistrat o scădere de aproximativ 6% în performanță la adăugarea unei singure culori în plus deci este de așteptat ca la adăugarea a încă șase culori scăderea să fie și mai mare.

K-Nearest Neighbors (kNN):

Acuratețea a variat între 66% și 72%, în funcție de setul de date și clasificarea dorită. Pentru clasificarea a două culori (blue și red), modelul a obținut o acuratețe de 72%, indicând o performanță bună în acest scenariu. În cazul clasificării pe trei culori, performanța a fost echilibrată, cu precizii și recall-uri acceptabile pentru fiecare clasă.

Naive Bayes: Acuratețea modelului a variat între 65% și 66% pentru scenariile prezentate. A arătat o performanță mai bună în clasificarea culorilor 'blue' și 'green', dar a avut dificultăți în identificarea corectă a culorii 'red'. Performanța a fost mai consistentă în comparatie cu kNN, dar modelul poate beneficia de ajustări suplimentare.

• Bibliografie

https://images.cv/dataset/green-snake-image-classification-dataset

https://cvhci.anthropomatik.kit.edu/~bschauer/datasets/google-512/

https://excelchamps.com/conditional-formatting/cell-contains-text/