למידה חישובית – תרגיל 3

מיכל אברמוב, 301834297
בוריס בורשבסקי, 311898746

מה מצורף

1. מחברת המפרטת בשלבים את מה שעשינו (modeling_notebook.ipynb)
2. קובץ modeling.py המכיל את כל הקוד של התרגיל
3. קובץ html שמאפשר צפיה נוחה יותר במחברת
4. קבצי הdata.
5. קובץ test_pridictions.csv עם תוצאות הפרדיקציה

התרגיל

התרגיל מחולק ל3 שלבים עיקריים:

1. בחינה בסיסית של מספר מודלים
2. בחינה מעמיקה של 3 מודלים בולטים
3. הפעלה ופרדיקציה בעזרת המודל שנבחר

בחינה בסיסית

במהלך הבחינה הבסיסית רצנו על מספר רב של מודלים על הdata של הtrain וחישבנו עליהם cross_val_score על מנת להעריך את הפרדיקציה, את כולם ניתן לראות במחברת, אנו נפרט פה על כמה עיקריים:

- ראינו ש LinearSVC וSVC נתנו לנו ניקוד באזור ה 0.9 ולכן לא בחרנו אותם
- ראינו שOneVsOneClassifier נתן ניקוד של 0.92.
- ראינו ש GaussianNB נתן ניקוד של 0.86, גם Perceptron וLinearDiscriminantAnalysis נתנו ציונים יחסית נמוכים
- עבור KNeighborsClassifier רצנו על כל מאפשרויות מ2 עד 20 ע״מ למצוא את הk האופטימלי, מבחינותינו הk האופטימלי היה 5 אך גם הוא נתן רק ניקוד של 0.92:

```
minimum n_neighbors = 2, score = 0.905779
minimum n_neighbors = 3, score = 0.916074
minimum n_neighbors = 4, score = 0.912584
minimum n_neighbors = 5, score = 0.919978
minimum n_neighbors = 6, score = 0.916100
minimum n_neighbors = 7, score = 0.915899
minimum n_neighbors = 8, score = 0.913968
minimum n_neighbors = 9, score = 0.914135
minimum n_neighbors = 10, score = 0.912396
minimum n_neighbors = 11, score = 0.909670
minimum n_neighbors = 12, score = 0.910648
minimum n_neighbors = 13, score = 0.907146
minimum n_neighbors = 14, score = 0.906557
minimum n_neighbors = 15, score = 0.905389
minimum n_neighbors = 16, score = 0.902088
minimum n_neighbors = 17, score = 0.901114
minimum n_neighbors = 18, score = 0.899555
minimum n_neighbors = 19, score = 0.898974
Best n_neighbors size: 5
KNeighborsClassifier with best N param: 0.919978
```

- עבור DecisionTreeClassifier רצנו על כל האפשרויות של עצים עם גודל split משתנה וראינו שעבור split מינימאלי בגודל 5 אנו מקבלים ציון של 9.3

```
minimus splitter = 2, score = 0.925955
minimus splitter = 3, score = 0.926707
minimus splitter = 4, score = 0.929093
minimus splitter = 5, score = 0.930633
minimus splitter = 6, score = 0.928047
minimus splitter = 7, score = 0.929036
minimus splitter = 8, score = 0.929990
minimus splitter = 9, score = 0.928648
minimus splitter = 10, score = 0.928674
minimus splitter = 11, score = 0.929825
minimus splitter = 12, score = 0.928855
minimus splitter = 13, score = 0.925516
minimus splitter = 14, score = 0.925342
minimus splitter = 15, score = 0.924767
minimus splitter = 16, score = 0.923404
minimus splitter = 17, score = 0.923196
minimus splitter = 18, score = 0.922429
minimus splitter = 19, score = 0.922043
Best Splitter size: 5
DecisionTreeClassifier with best splitter: 0.930633
DecisionTreeClassifier Default score: 0.925955
```

- ניסינו אותו דבר עם RandomForestClassifier וראינו שעבורו אנו מקבלים ציונים מעל 9.4 כאשר עבוד splitter בגודל 4 מקבלים אפילו 9.5

```
minimum splitter = 2, score = 0.946670
minimum splitter = 3, score = 0.948827
minimum splitter = 4, score = 0.951943
minimum splitter = 5, score = 0.947105
minimum splitter = 6, score = 0.947287
minimum splitter = 7, score = 0.946150
minimum splitter = 8, score = 0.945923
minimum splitter = 9, score = 0.949397
minimum splitter = 10, score = 0.944574
minimum splitter = 11, score = 0.946851
minimum splitter = 12, score = 0.943054
minimum splitter = 13, score = 0.946655
minimum splitter = 14, score = 0.945529
minimum splitter = 15, score = 0.947304
minimum splitter = 16, score = 0.947055
minimum splitter = 17, score = 0.946091
minimum splitter = 18, score = 0.942185
minimum splitter = 19, score = 0.946858
Best Splitter size: 4
RandomForestClassifier with best splitter: 0.951943
RandomForestClassifier Default score: 0.946670
```

בחינה מעמיקה של 3 מודלים בולטים

- בשלב זה בחרנו 3 מודלים שאותם נבחן ע״י פרדיקציה על הtrain והצגה של classification report:
  - **RandomForestClassifier(min_samples_split=4, max_features=None)**

```
***** RandomForestClassifier *****
             precision    recall  f1-score   support

      Blues    0.54545   0.44444   0.48980        27
     Browns    0.90359   0.96175   0.93177      1072
     Greens    0.99780   0.99561   0.99671       912
      Greys    0.98489   0.94220   0.96307       346
    Oranges    0.90826   0.96429   0.93543       308
      Pinks    0.89805   0.84318   0.86975       491
    Purples    0.97107   0.96867   0.96987      1213
       Reds    0.97419   0.96486   0.96950       313
     Whites    0.86310   0.71782   0.78378       202
    Yellows    0.94553   0.96047   0.95294       253

avg / total    0.94437   0.94471   0.94389      5137
```

  - **KNeighborsClassifier(n_neighbors=5)**

```
***** KNeighborsClassifier *****
             precision    recall  f1-score   support

      Blues    0.50000   0.03704   0.06897        27
     Browns    0.86484   0.97295   0.91572      1072
     Greens    0.99232   0.99123   0.99177       912
      Greys    0.86479   0.88728   0.87589       346
    Oranges    0.89247   0.80844   0.84838       308
      Pinks    0.94203   0.79430   0.86188       491
    Purples    0.97930   0.97527   0.97728      1213
       Reds    0.84384   0.89776   0.86997       313
     Whites    0.82781   0.61881   0.70822       202
    Yellows    0.90647   0.99605   0.94915       253

avg / total    0.92093   0.92174   0.91814      5137
```

  - **DecisionTreeClassifier(min_samples_split=5)**

```
***** DecisionTreeClassifier *****
             precision    recall  f1-score   support

      Blues    0.40000   0.37037   0.38462        27
     Browns    0.89744   0.91418   0.90573      1072
     Greens    0.98905   0.99013   0.98959       912
      Greys    0.93696   0.94509   0.94101       346
    Oranges    0.88599   0.88312   0.88455       308
      Pinks    0.85062   0.83503   0.84275       491
    Purples    0.96595   0.95878   0.96235      1213
       Reds    0.93949   0.94249   0.94099       313
     Whites    0.73367   0.72277   0.72818       202
    Yellows    0.94048   0.93676   0.93861       253

avg / total    0.92301   0.92330   0.92312      5137
```

ניתן לראות ש**RandomForestClassifier** מציג תוצאות טובות משמעותית משני הclassifierים האחרים

הפעלה ופרדיקציה בעזרת המודל שנבחר

המודל הנבחר:

(**RandomForestClassifier**(min_samples_split=4, max_features=None

בשלב זה אימננו את המודל בעזרת RepeatedStratifiedKFold וקיבלנו ציון של 0.942263 בדיוק!

לבסוף הרצנו פרדיקציה על הטסט וקיבלנו שהמפלגה הזוכה היא ה**סגולה**

אך ניתן לראות הפער מהחומים מאוד קטן:

```
Vote distribution
Blues, 10.000000, 0.587199%
Browns, 400.000000, 23.487962%
Greens, 309.000000, 18.144451%
Greys, 94.000000, 5.519671%
Oranges, 104.000000, 6.106870%
Pinks, 167.000000, 9.806224%
Purples, 404.000000, 23.722842%
Reds, 104.000000, 6.106870%
Whites, 45.000000, 2.642396%
Yellows, 66.000000, 3.875514%
```

בשלב זה ברצנו בדיקה של הclassification שלנו מול המידע test המקורי וראינו דיוק של 94% - למעשה טעינו ב101 מתוך 1703 ניסויים.

Confusion matrix:

```
array([[  6,   0,   0,   0,   0,   0,   0,   0,   0,   5],
       [  0, 364,   2,   0,   0,   4,   1,   0,   6,   0],
       [  0,   0, 307,   0,   0,   3,   2,   0,   0,   0],
       [  0,   0,   0,  92,   7,   0,   0,   0,   0,   0],
       [  0,   0,   0,   2,  90,   0,   0,   5,   0,   0],
       [  0,  10,   0,   0,   0, 153,   2,   0,   2,   0],
       [  0,   2,   0,   0,   0,   4, 393,   0,   0,   0],
       [  0,   0,   0,   0,   7,   0,   0,  99,   0,   0],
       [  0,  24,   0,   0,   0,   3,   6,   0,  37,   0],
       [  4,   0,   0,   0,   0,   0,   0,   0,   0,  61]])
```

בשלב זה החלטנו לבדוק את התפלגות הטעויות שלנו וראינו שבמרבית המקרים אנו טועים לטובת החומים, דבר שעשוי להטות את התוצאות:

{'Reds': 8, 'Greens': 2, 'Whites': 11, 'Yellows': 4, 'Greys': 2, 'Oranges': 11, '**Browns': 33**, 'Pinks': 27,'
{'Blues': 4, 'Purples': 13

בשלב זה ניסנו לראות האם **DecisionTreeClassifier** בכל זאת נותן תוצאות טובות יותר, אך ראינו כי לא כך והחלטנו להשאר עם **RandomForestClassifier.**

לבסוף את הפרדיקציה ניתן לראות בקובץ test_pridictions.csv