תרגול 8 - אלגוריתם KNN ככלי לפיתוח מכונה לומדת

תרגיל 1

שלב א' (מימוש מרחק אוקלידי בין נקודות במרחב דו-מימדי)

ממשו קוד בשפת python המחשב את המרחק האוקלידי בין הנקודה הראשונה במערך data לבין כל אחד משאר הנקודות.

לרשותכם הקוד הבא הכולל את מערך הנתונים data עליו יש לבצע את הבדיקה.

```
import matplotlib.pyplot as plt
import numpy as np
data = np.array( [      [ 6 , 7],
                        [ 2 , 3],
                        [ 3 , 7],
                        [ 4 , 4],
                        [ 5 , 8],
                        [ 6 , 5],
                        [ 7 , 9],
                        [ 8 , 5],
                        [ 8 , 2],
                        [10 , 2]   ])
categories = np.array([0,1,1,1,1,2,2,2,2,2])
colormap = np.array(['r', 'g', 'b'])
plt.scatter(data[:,0], data[:,1], s=100, c=colormap[categories])
plt.show()
```

פלט צפוי:

```
All Euclidean Distance from point:  [6. 7.]

0.0
5.656854249492381
3.0
3.605551275463989
1.4142135623730951
2.0
2.23606797749979
2.8284271247461903
5.385164807134504
6.4031242374328485
```

שלב ב' (מימוש מרחק אוקלידי ב- N מימדים)

ממשו קוד בשפת python המחשב את המרחק האוקלידי בין הנקודה הראשונה במערך data לבין כל אחד משאר הנקודות.

לרשותכם הקוד הבא הכולל את מערך הנתונים data עליו יש לבצע את הבדיקה.

```
data = np.array( [      [ 5.0 , 5.0, 5.0],
                        [ 0.0 , 0.0, 0.0],
                        [ 3.0 , 7.0, 2.0],
                        [ 4.0 , 4.0, 8.0],
                        [ 5.0 , 8.0, 9.0],
                        [ 6.0 , 5.0, 7.0],
                        [ 7.0 , 9.0, 4.0],
                        [ 8.0 , 5.0, 1.0],
                        [ 8.0 , 2.0, 3.0],
                        [10.0 , 2.0, 5.0]    ])
```

כדי להציג בגרף 3D את הנקודות יש לכתוב את הקוד הבא:

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(data[:,0], data[:,1],data[:,2], s=150)
plt.show()
```

שלב ג' (מימוש פעולה KNN עבור N מימדים)

ממשו קוד בשפת python המחשב מהם הנקודות הכי קרובות עבור K=1 ו- K=3 בין הנקודה test_data לבין הנקודות train_data.

לרשותכם הקוד הבא הכולל את מערך הנתונים data

```
train_data = np.array( [
                [ 2.0 , 3.0 ],
                [ 3.0 , 7.0 ],
                [ 4.0 , 4.0 ],
                [ 5.0 , 8.0 ],
                [ 6.0 , 5.0 ],
                [ 7.0 , 9.0 ],
                [ 8.0 , 5.0 ],
                [ 8.0 , 2.0 ],
                [10.0 , 2.0 ]   ])

test_data = np.array( [[ 6.0 , 7.0 ]])
```

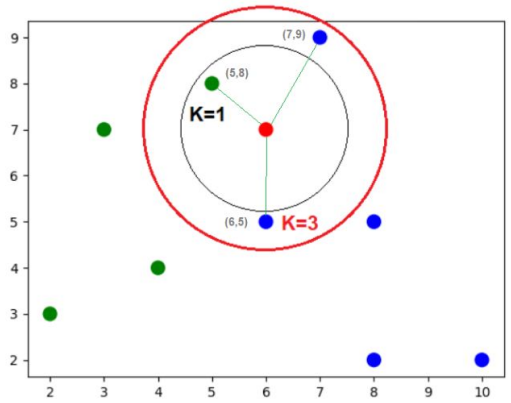
פלט רצוי:

נקבל את הפלט הבא עובר K=3:

```
[5. 8.]
[6. 5.]
[7. 9.]
```

ואת פלט הבא עובר K=1:

```
[5. 8.]
```



שלב ד' (מימוש כל הקוד יחד)

ממשו קוד בשפת python הכולל את הפעולה predict כדי להציג לאיזה קבוצה שייכת הנקודה test_data על מסך תיוג הנקודות שב train_data.

בדקו את הקוד שלכם עבור K=1 ו- K=3

לרשותכם קוד הכולל את מערך הנתונים data ו- lbl

```
train_data = np.array( [
            [ 2.0 , 3.0 ],
            [ 3.0 , 7.0 ],
            [ 4.0 , 4.0 ],
            [ 5.0 , 8.0 ],
            [ 6.0 , 5.0 ],
            [ 7.0 , 9.0 ],
            [ 8.0 , 5.0 ],
            [ 8.0 , 2.0 ],
            [10.0 , 2.0 ]   ])

train_lbl = np.array( [
            [ 1 ],
            [ 1 ],
            [ 1 ],
            [ 1 ],
            [ 2 ],
```

```
            [ 2 ],
            [ 2 ],
            [ 2 ],
            [ 2 ]          ])

test_data = np.array( [[ 6.0 , 7.0 ]])
```

תרגיל 2: סיווג פרחי אירוס (בדיקת ביצועי KNN בסיווג פרחי אירוס)

בפעילות זו נערוך בדיקת ביצועים לאלגוריתם KNN במטרה לבצע סיווג פרחי אירוס ל-3 משפחות.

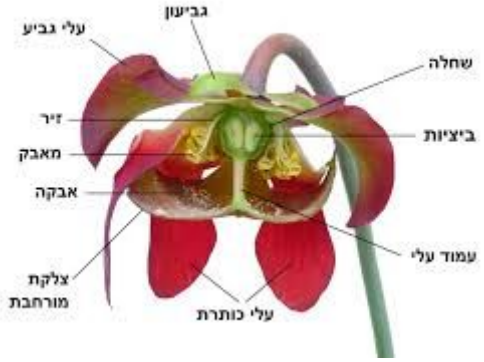
שלב א' (היכרות עם מבנה מערך הנתונים)

מערך נתונים הכולל מידע מתויג על 3 סוגים שונים של אירוסים. להלן תמונות של שלשות האירוסים:

מאגר הנתונים כולל נתונים מתוקפים מחקרית המסווגים שלושת פרחים על פי 4 מאפיינים שהם:

- אורך ה- petel (אורך עלי כותרת של איריס)
- רוחב ה- petal (רוחב עלי כותרת של איריס)
- אורך ה- sepal (אורך עלי הגביע של איריס)
- רוחב ה- sepal (רוחב עלי הגביע של איריס)

להלן תמונה שתסביר את המאפיינים כל גבי תמונה של הפרח



ניתן להוריד את קובץ הנתונים ישירות מהקישור הבא:

https://www.neuraldesigner.com/files/datasets/iris_flowers.csv

נפתח את קובץ הנתונים על ידי תוכנת Excel ונקבל את הנתונים הבאים:

	A	B	C	D	E
1	sepal_length	sepal_width	petal_length	petal_width	class
2	5.1	3.5	1.4	0.2	iris_setosa
3	4.9	3	1.4	0.2	iris_setosa
4	4.7	3.2	1.3	0.2	iris_setosa
5	4.6	3.1	1.5	0.2	iris_setosa
6	5	3.6	1.4	0.2	iris_setosa
7	5.4	3.9	1.7	0.4	iris_setosa
8	4.6	3.4	1.4	0.3	iris_setosa
9	5	3.4	1.5	0.2	iris_setosa
10	4.4	2.9	1.4	0.2	iris_setosa
11	4.9	3.1	1.5	0.1	iris_setosa
12	5.4	3.7	1.5	0.2	iris_setosa
13	4.8	3.4	1.6	0.2	iris_setosa
14	4.8	3	1.4	0.1	iris_setosa
15	4.3	3	1.1	0.1	iris_setosa
16	5.8	4	1.2	0.2	iris_setosa
17	5.7	4.4	1.5	0.4	iris_setosa

קיבלנו קובץ המכיל דגימות של 150 פרחים (50 מכל סוג) כאשר לכל פרח מדדו את 4 המאפיינים שקבענו כדי להבדיל בין השלושה. כמובן הקובץ מכיל עמודה חמישית הכוללת את שם הפרח שאותו מדדו. מכאן שיש לנו 150 שורות של מידע מתויג.

בשלב הבא יש צורך להתאים את מערך הנתונים כדי שיכנס לתוך אלגוריתם KNN באופן הבא:

- נשנה את שם הפרח למספרים 1 עד 3 בהתאמה.
- נערבב את סוגי הפרחים.
- נחלק את המערך ל- 4 מערכים על פי הפירוט הבא:
  a. מערך אימונים הכולל רק את הנתונים.
  b. מערך הכולל את התגיות של מערך האימונים.
  c. מערך בדיקה הכולל רק את הנתונים.
  d. מערך הכולל את התגיות של מערך הבדיקה.

ממשו קוד בשפת python המבצע זאת.

להלן דוגמה לפלט תקין:

```
test_data:
 [[5.7 3.8 1.7 0.3]
 [6.  2.9 4.5 1.5]
 [6.9 3.1 4.9 1.5]
 [5.9 3.  5.1 1.8]
 [6.3 2.7 4.9 1.8]
 [5.4 3.7 1.5 0.2]
 [4.6 3.2 1.4 0.2]
 [5.1 3.8 1.9 0.4]
 [4.9 3.1 1.5 0.1]
 [5.1 3.4 1.5 0.2]]
train_data:
 [[5.7 2.6 3.5 1. ]
 [5.5 2.4 3.7 1. ]
 [6.6 2.9 4.6 1.3]
 [7.1 3.  5.9 2.1]
 [5.  3.3 1.4 0.2]
 [7.7 3.  6.1 2.3]
 [4.9 2.5 4.5 1.7]
 [4.7 3.2 1.3 0.2]
 [4.9 2.4 3.3 1. ]
 [5.7 2.8 4.5 1.3]
 [7.9 3.8 6.4 2. ]
 [4.9 3.1 1.5 0.1]
 [6.1 2.8 4.7 1.2]
 [4.8 3.4 1.9 0.2]
 [6.9 3.2 5.7 2.3]
test_lbl:
 [[1.]
 [2.]
 [2.]
 [3.]
 [3.]
 [1.]
 [1.]
 [1.]
 [1.]
 [1.]]
train_lbl:
 [[2.]
 [2.]
 [2.]
 [3.]
 [1.]
 [3.]
 [3.]
 [1.]
 [2.]
 [2.]
 [3.]
 [1.]
 [2.]
 [1.]
 [3.]
 [1.]
```

כדי להמיר קובץ CSV למערך Numpy ניתן להשתמש בקוד הבא:

```
import numpy as np
vir_iris_data = np.genfromtxt('iris_for_ML.csv', delimiter=',')
```

שלב ב' (מימוש סיווג פרחי אירוס תוך שימוש בפעולות שכתבנו)

ממשו קוד בשפת python העושה שימוש בפעולות שכתבנו כדי לבצע סיווג לשלושת סוגי הפרחים

להלן בסיס לצורך מימוש התרגיל:

```
def euclidean_distance(p1, p2):
    d = 0.0
    for i in range(len(p1)):
        a = float(p1[i])
        b = float(p2[i])
        d += np.power((a-b),2)
    d = np.sqrt(d)
    return d

def takeSecond(elem):
    return elem[1]

def predict(train, test, lbl, K):
```

```
        distances = []
        for t, l in zip(train,lbl):
            dist = euclidean_distance(test, t)
            distances.append([t, dist, l[0]])
        distances.sort(key=takeSecond)
        neighbors = []
        for i in range(K):
            neighbors.append(distances[i])
        out = [row[-1] for row in neighbors]
        return max(out,key=out.count)
```

שלב ג' (מימוש סיווג פרחי אירוס תוך שימוש ב- sklearn)

ממשו קוד בשפת python העושה שימוש בספריה sklearn כדי לבצע סיווג לשלושת סוגי הפרחים תוך שימוש במחלקה KNeighborsClassifier. להלן בסיס לצורך מימוש התרגיל:

```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix

….

classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(train_data, train_lbl)
lbl_pred = classifier.predict(test_data)
print(confusion_matrix(test_lbl, lbl_pred))
```

דוגמה לפלט תקין:

```
[[ 8  0  0]
 [ 0  9  0]
 [ 0  1 12]]
```

שלב ד' (מימוש סיווג פרחי אירוס תוך שימוש ברשת נוירונים ב- sklearn)

ממשו קוד בשפת python העושה שימוש בספריה sklearn כדי לבצע סיווג לשלושת סוגי הפרחים תוך שימוש במחלקה MLPClassifier. להלן בסיס לצורך מימוש התרגיל:

```
import numpy as np
import sklearn.neural_network
from sklearn.metrics import confusion_matrix
```

```
……..

mlp = sklearn.neural_network.MLPClassifier(
            hidden_layer_sizes=(5),
            solver='sgd',
           learning_rate_init=0.01,
            max_iter=1000)

mlp.fit(train_data, train_lbl)
```