

## דו"ח תרגיל 2 למידת מכונה

עבור כל אלגוריתם בנית פונקציה שתבדוק את הערכים של ההיפר-פרמטרים שמניבים את אחוזי ההצלחה הטובים ביותר.

כאשר השתמשתי ב-10 אחוזים מהקלט שנתון לי כ-90 validation אחוז מהקלט שלחתי לאימון של האלגוריתם. לאחר שקיבלתי את  $w$  חזרה מהמודל בדקתי על סט validation את אחוזי ההצלחה. כך ביצעתי וערכתי כל פעם את הvalidation להיות 10 אחוזים אחרים מהקלט כדי לקבל ממוצע של אחוזי ההצלחה בין כל 10 הפעמים עבור ערכי ההיפר-פרמטרים ששלחתי.

עבור knn האלגוריתם חיפש את הערך הטוב ביותר עבור ההיפר-פרמטר  $k$  וקיבלתי את ממוצע אחוז ההצלחה הגבוה ביותר עבור  $k=3$   
the best percentage success is : 93.51851851851852% for  $k = 3$

```
def k_parm_knn(x_train, y_train):
    per_valid = 10
    len_valid = int(len(x_train)/per_valid)
    max_success = 0
    max_k = 1
    for k in range(1, len(x_train)):
        i = 0
        success = 0
        while i < per_valid:
            begin = i * len_valid
            end = begin + len_valid - 1
            tmp_x = x_train.copy()
            tmp_y = y_train.copy()
            validation_x = tmp_x[begin:end]
            validation_y = tmp_y[begin:end]
            for j in range(len_valid):
                tmp_x = np.delete(tmp_x, begin, axis=0)
                tmp_y = np.delete(tmp_y, begin, axis=0)
                for x,y in zip(validation_x,validation_y):
                    y_hat = knn_model(k, tmp_x, tmp_y, x)
                    if y == y_hat:
                        success += 1
                i += 1
            success /= per_valid
            if success > max_success:
```

```

max_success = success
max_k = k
print(f" the percentage success is : {(success/len(validation_x))*100}% for k = {k} ")

print(f" the best percentage success is : {(max_success/len(validation_x))*100}% for k = {max_k} ")
return max_k

```

עבור אלגוריתם perceptron ניסיתי למצוא את הערכים הכי טובים עבור כמות epochs וערך הלמידה  $\eta$ .

```

def parms_perceptron(x_train, y_train):
    per_valid = 10
    len_valid = int(len(x_train)/per_valid)
    max_n2 = 1
    max_epoch = 1
    max2_success = 0
    for epoch in range(1, 200):
        max_n = 1
        n = 0.01
        max_success = 0
        while n < 1:
            success = 0
            i = 0
            while i < per_valid:
                begin = i * len_valid
                end = begin + len_valid - 1
                tmp_x = x_train.copy()
                tmp_y = y_train.copy()
                validation_x = tmp_x[begin:end]
                validation_y = tmp_y[begin:end]
                for j in range(len_valid):
                    tmp_x = np.delete(tmp_x, begin, axis=0)
                    tmp_y = np.delete(tmp_y, begin, axis=0)

```

```

w = per_train(n, epoch, tmp_x, tmp_y)
for x,y in zip(validation_x, validation_y):
    y_hat = np.argmax(np.dot(w, x))
    if y == y_hat:
        success += 1
    i += 1
success /= per_valid
if success > max_success:
    max_success = success
    max_n = n
n += 0.1
if max_success > max2_success:
    max2_success = max_success
    max_n2 = max_n
    max_epoch = epoch
print(f" the percentage success is : {(max_success / len(validation_x)) * 100}% with n = {max_n} for epoch = {epoch} ")
print(f" the best percentage success is : {(max2_success/len(validation_x))*100}% with n = {max_n2} for epoch = {max_epoch} ")
return max_epoch,max_n2

```

לאחר הרצה ממושכת קיבלתי שעבור  $n=0.83$  וכמות epochs=102 האלגוריתם נותן את האחוזים הגבוהים ביותר. אך בהסתכלות בנתונים עבור שאר ההדפסות גילתי שהפער כמעט לא ניכר עבור ערכי n שונים ובאחוזים מאוד מאוד קטנים אז ניתן לומר שערך הלמידה לא כל כך שינה.

the best percentage success is : 97.39130434782608% with  $\eta = 0.8300000005$  for epoch = 102

עבור אלגוריתם svm הצטרכתי למצוא את הערכים הכי טובים עבור ערכי הלמדא, קצת הלמידה וכמות epochs

```

def parms_svm(x_train, y_train):
    per_valid = 10
    len_valid = int(len(x_train) / per_valid)
    max_ep=1
    max3_success=0
    max_3n=1

```

```

max_lamda3=1
for epoch in range(1,200):
    max_lamda2 = 1
    max2_success = 0
    max_n = 1
    n = 1
    while n > 0.0000001:
        max_lamda = 1
        lamda = 0.1
        max_success = 0
        while lamda >= 0.000000000001:
            success = 0
            i = 0
            while i < per_valid:
                begin = i * len_valid
                end = begin + len_valid - 1
                tmp_x = x_train.copy()
                tmp_y = y_train.copy()
                validation_x = tmp_x[begin:end]
                validation_y = tmp_y[begin:end]
                for j in range(len_valid):
                    tmp_x = np.delete(tmp_x, begin, axis=0)
                    tmp_y = np.delete(tmp_y, begin, axis=0)
                w = svm_train(lamda, n, epoch, tmp_x, tmp_y)
                for x, y in zip(validation_x, validation_y):
                    y_hat = np.argmax(np.dot(w, x))
                    if y == y_hat:
                        success += 1
                i += 1
            success /= per_valid
            if success > max_success:
                max_success = success
                max_lamda = lamda
            lamda /= 10
        if max_success > max2_success:
            max2_success = max_success
            max_lamda2 = max_lamda

```

```

        max_n = n
    n /= 10
    if max2_success > max3_success:
        max3_success = max2_success
        max_3n = max_n
        max_lamda3 = max_lamda2
        max_ep = epoch
    print(f" the percentage success is : {(max2_success / len(validation_x)) * 100}% with n = {max_n} for lamda = {max_lamda2} for epoch = {epoch} ")

    print(f" the best best percentage success is : {(max3_success / len(validation_x)) * 100}% with n = {max_3n} for lamda = {max_lamda3} for epoch = {max_ep} ")
    return max_lamda3, max_3n, max_ep

```

the best best percentage success is : 96.52173% with n = 0.001 for lamda = 0.1 for epoch = 89

ועבור אלגוריתם pa היה צריך רק למצוא את כמות הepochs

```

def epoch_parm_pa(x_train, y_train):
    per_valid = 10
    len_valid = int(len(x_train) / per_valid)
    max_epoch = 1
    max_success = 0
    for epoch in range(1, 200):
        success = 0
        i = 0
        while i < per_valid:
            begin = i * len_valid
            end = begin + len_valid - 1
            tmp_x = x_train.copy()
            tmp_y = y_train.copy()
            validation_x = tmp_x[begin:end]
            validation_y = tmp_y[begin:end]
            for j in range(len_valid):
                tmp_x = np.delete(tmp_x, begin, axis=0)
                tmp_y = np.delete(tmp_y, begin, axis=0)
            w = pa_train(epoch, tmp_x, tmp_y)
            for x, y in zip(validation_x, validation_y):

```

```

y_hat = np.argmax(np.dot(w, x))
if y == y_hat:
    success += 1
i += 1
success /= per_valid
if success > max_success:
    max_success = success
    max_epoch = epoch
print(f" the percentage success is : {(success / len(validation_x)) * 100}% with for epoch = {epoch} ")
print(f" the best percentage success is : {(max_success / len(validation_x)) * 100}% for epoch = {max_epoch} ")
return max_epoch

```

the best percentage success is : 96.52173913043478% for epoch = 119

לגבי אלגוריתמי הלמידה השתמשתי בדברים הבאים בשביל להגדיל את אחוז ההצלחה:

Feature selection כאשר בניסוי וטעיה החלטתי להוריד את הפיצר האחרון

בחירת היפר-פרמטרים שמניבים אחוזים טובים ביותר על סט הלמידה

הוספת bias

נרמול הנתונים לפי פונקציית z-score