מערכות לומדות תשפ"ג - תרגיל 2

מסווגים ומאפיינים

בתרגיל זה תשתמשו בספריה scikit-learn ותממשו קוד בעצמכם כדי ללמוד על תכונות שונות של מסווגים, על מדדי איכות שונים ועל מאפיינים.
סעיפים להגשה עם ניקוד מסומנים בצבע חום. סעיפים לבירור עצמי ללא ניקוד מסומנים בירוק עם קו תחתי.

נושא 1 – מדדי איכות

בשיעור הזכרנו את הטבלה הנקראת confusion matrix המשמשת ככלי להצגת ביצועי אלגוריתמים של סיווג. כעת נתעמק יותר בנושא מדדי האיכות של אלגוריתמי סיווג ובדרכי ההשוואה בין האלגוריתמים.

Confusion matrix

ראו גם הערך בוויקיפדיה https://en.wikipedia.org/wiki/Confusion_matrix#Table_of_confusion.
בבעיית סיווג בינארי ישנן שתי מחלקות אותן נכנה חיובית ושלילית. כאשר אנו מפתחים מסווג, עליו לתת כמובן תשובה אחת משתיים לכל נתון (חיובי או שלילי). על כן אפשר לתאר את תשובות המסווג ביחס לנתון בו אנו יודעים את התשובה האמיתית, זאת בעזרת ארבעה סוגי התשובה הבאים:

1. תשובת חיובית נכונה – True Positive (המסווג השיב **חיובי** וזו תשובה **נכונה**).
2. תשובה שלילית נכונה – True Negative (המסווג השיב **שלילי** וזו תשובה **נכונה**).
3. תשובה חיובית כוזבת – False Positive (המסווג השיב **חיובי** וזו תשובה **שגויה**).
4. תשובה שלילית כוזבת – False Negative (המסווג השיב **שלילי** וזו תשובה **שגויה**).

נציג את התשובות הללו בטבלה

		תשובת המסווג	
		Positive	Negative
סיווג האמת	Positive	**True Positive (TP)**	**False Negative (FN)**
	Negative	**False Positive (FP)**	**True Negative (TN)**

בעזרת הטבלה אפשר לתאר גדלי שגיאה עבור אלגוריתם סיווג מסוים על סט מבחן מסוים.

נעזר בדוגמה של המחלה הממארת שבה תיארנו את השימוש בחוק בייס באבחנה רפואית (שיעור 2). השיטה לאבחון שהזכרנו היא למעשה אלגוריתם סיווג, ובו נדון כעת.
במונחי הטבלה, כאשר האלגוריתם מחליט "חולה" זו תשובה חיובית, וכאשר האלגוריתם מחליט "לא חולה" זו התשובה השלילית.

נזכור גם כי ההסתברות האפריורית להיות חולה היא 1 ל 1000.

כעת נשווה כמה כללי החלטה (אלגוריתמי סיווג).

א. כלל "תמיד בריא".
(**שאלה 1**) מלאו את הטבלה עבור מסווג זה במספרים הצפויים עבור מדגם מייצג של האוכלוסייה בגודל 1000 איש (999 בריאים, 1 חולה):

		תשובת המסווג "תמיד בריא"	
		Positive	Negative
סיווג האמת	Positive P=1	**True Positive (TP)**	**False Negative (FN)**
	Negative N=999	**False Positive (FP)**	**True Negative (TN)**

(**שאלה 2**) התיוגים באלכסון המשני (האדום) הם התיוגים השגויים. כמה שגיאות עשה האלגוריתם זה?

(**שאלה 3**) נגדיר את דיוק המסווג כיחס בין

מספר התיוגים הנכון ל ***מספר השאלות הכולל*** =

accuracy (ACC)

$$ACC = \frac{TP + TN}{P + N}$$

מה דיוק המסווג?

אנו רואים כי מסווג כזה, שאינו מתאמץ כלל להבחין בין בריאים לחולים מגיע לאחוזי דיוק מרשימים במיוחד – כל זאת הודות לנדירות הרבה של החולים באוכלוסייה.

(**שאלה 4**) עבור מסווג זה חשבו את הערכים המתוארים בטבלה

		תשובת המסווג	
		Positive	Negative
סיווג האמת	Positive P	**True Positive rate** **TP/P**	**False Negative rate** **FN/P**
	Negative N	**False Positive rate** **FP/N**	**True Negative rate** **TN/N**

ב. כלל "תמיד חולה".

(**שאלות 5 עד 8**) עבור מסווג זה חשבו והציגו את אותם מדדים כמו בשאלות 1 עד 4.

ג. כלל "מחליט בעזרת מטבע הוגן".

(**שאלות 9 עד 12**) עבור מסווג זה חשבו והציגו את אותם מדדים כמו בשאלות 1 עד 4. כדי לקבל בטבלה בשאלה 9 ערכים שלמים, הכפילו את כמות הנתונים: P=2 ו N = 1998. **הסבירו.**

Confusion matrix , precision, recall, f1-score

הסתכלו על תוצאות ההרצה של הדוגמה של סיווג ספרות (ראינו אותה בעבר):
https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digits-classification-py

תוצאות סיווג הספרות מוצגות בעזרת כמה מדדי איכות. נתחיל עם Confusion matrix:

```
[[87  0  0  0  1  0  0  0  0  0]
 [ 0 88  1  0  0  0  0  0  1  1]
 [ 0  0 85  1  0  0  0  0  0  0]
 [ 0  0  0 79  0  3  0  4  5  0]
 [ 0  0  0  0 88  0  0  0  0  4]
 [ 0  0  0  0  0 88  1  0  0  2]
 [ 0  1  0  0  0  0 90  0  0  0]
 [ 0  0  0  0  0  1  0 88  0  0]
 [ 0  0  0  0  0  0  0  0 88  0]
 [ 0  0  0  1  0  1  0  0  0 90]]
```

בעזרת הצגת ה confusion matrix ענו:

(שאלה 13)

כיצד מחושבים המספרים בשורה ה i? הסבירו בצורה מפורשת.

(שאלה 14)

כיצד מחושבים המספרים בעמודה ה j? הסבירו בצורה מפורשת.

נסתכל כעת על המדדים precision, recall, f1-score.

העזרו בהסברים בקישורים האלה

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support

https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

והסתכלו על התוצאות המוצגות בעזרת מדדים אלה

```
Classification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        88
           1       0.99      0.97      0.98        91
           2       0.99      0.99      0.99        86
           3       0.98      0.87      0.92        91
           4       0.99      0.96      0.97        92
           5       0.95      0.97      0.96        91
           6       0.99      0.99      0.99        91
           7       0.96      0.99      0.97        89
           8       0.94      1.00      0.97        88
           9       0.93      0.98      0.95        92

    accuracy                           0.97       899
   macro avg       0.97      0.97      0.97       899
weighted avg       0.97      0.97      0.97       899
```

(שאלה 15)

לספרה 9 תוצאת precision הנמוכה ביותר.

לספרה 3 תוצאת ה recall הנמוכה ביותר.

כתבו את הנוסחה לחישוב של כל מדד, והעזרו בתוצאות שמוצגות ב confusion matrix כדי לרשום במפורש את החישוב המספרי של תוצאות אלה.

(שאלה 16)
תנו דוגמה למקרה ממשי מן העולם בו עדיף לקבל precision מקסימלי, והסבירו.

(שאלה 17)
תנו דוגמה למקרה ממשי מן העולם בו עדיף לקבל recall מקסימלי, והסבירו.

(שאלה 18)
עבור המסווג של שאלה 1 "תמיד בריא" חשבו את ערך ה precision ואת ערך ה recall לשתי המחלקות.

סיכום ביניים:
ראינו כי המדד של דיוק של המסווג עשוי להטעות. אם סט הנתונים שלנו אינו מאוזן (unbalanced), אזי בחירה תמיד במחלקה הנפוצה עשויה להניב מסווג עם דיוק גבוה. מדדים אחרים שהוצגו כאן פחות רגישים לבעיה זו. בהמשך הקורס נתאר שיטות נוספות לטפל בסט לא מאוזן.

נושא 2 – הנדסת מאפיינים

כאשר רוצים לבצע משימת למידה במרחב קלט מממד גבוה (למשל תמונות) יש צורך לרוב בהרבה מאוד נתונים. כדי לצמצם את הצורך בכמות נתונים גדולה מדי מנסים להוריד את הממד של מרחב הקלט (dimensionality reduction) על ידי המרת הקלט למאפיינים. משימת הלמידה תבוצע על קלט שהומר למאפייניו.
כמובן שההמרה למרחב מאפיינים צריכה לשמר את האינפורמציה הנחוצה למשימת הלמידה.

במקרים רבים משימת הלמידה דורשת הפרדה בין קבוצות. לעיתים הקבוצות הללו נפרדות זו מזו, אך גבולות ההפרדה ביניהן אינן קו ישר (או על-מישור). אזי ניתן לבצע המרה לא לינארית של הקלט (למרחב מאפיינים) כך שבמרחב החדש הקבוצות יופרדו על ידי על מישור. לפעמים מרחב המאפיינים דווקא יהיה מממד גבוה מאשר ממד הקלט. כאן נשתמש ב"טריק" מתמטי המאפשר לבצע זאת ללא חסרונות של ממד גבוה. עוד על כך בהמשך הקורס כשנזכיר Support Vector Machines – SVM ו Kernel Methods.

אם כך הקריטריונים לבחירת סט מאפיינים טוב הם:

א. אם מרחב הקלט מממד גבוה מדי, הסט מצמצם את ממד הקלט בצורה רבה.

לדוגמה: בניסיון לסיווג בננה \ תפוח מתמונות בגודל 100 X 100, נמיר ל 2 ערכים בלבד: צבע הפרי ויחס אורך רוחב של הפרי.

ב. אם הקבוצות בקלט לא ניתנות להפרדה על ידי על-מישור, סביר שיש טרנספורמציה לא לינארית הממירה למרחב מאפיינים בו כן ניתן לבצע הפרדה על ידי על-מישור.

ג. סט המאפיינים משמר את האינפורמציה הרלוונטית לבעיית הלמידה (וזורק את כל השאר). לדוגמה: צבע הפרי בבעיה של אומדן הבשלות של עגבניות מתמונות.

ד. רצוי – המאפיינים בעלי תכונות של invariances המתאימות לבעיה.
לדוגמה: נניח שנרצה להפריד בין משולשים למרובעים בתמונות, נחפש מאפיינים שלא תלויים במיקום הצורה בתמונה, בכיוון שלה או בגודלה, כלומר מאפיינים בעלי אדישות למיקום, סיבוב וגודל (invariant to translation, rotation and scale).

ה. רצוי – סט המאפיינים קל למימוש ומהיר לשימוש.
לדוגמה: מציאת הצבע הממוצע של תמונה קלה למימוש ומהירה יחסית לביצוע.

שימוש בקוד קיים

כעת נתמקד בתהליך בחירת המאפיינים לצורך זיהוי ספרות הכתובות בכתב יד. נשתמש ב data set של הספרות הכלול בספריה scikit-learn. במשימה זו ננסה להוריד את ממד הבעיה מ 64 (מספר הפיקסלים בתמונה) ל <= 10.
נטען את הנתונים באמצעות הקוד

```
from sklearn import datasets

# The digits dataset
digits = datasets.load_digits()
```

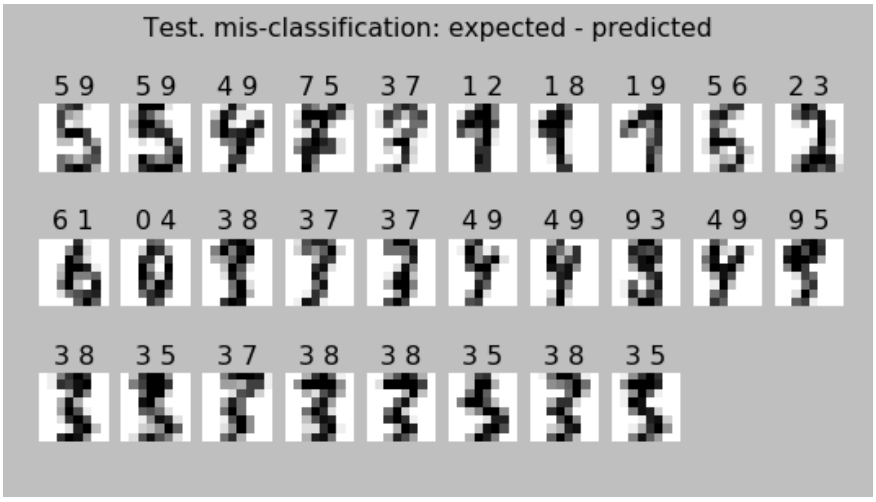
http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits

הנה דוגמה של הספרות עצמן:

נמשיך להיעזר בקוד עליו הסתכלנו בשאלות הקודמות
http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

הורידו את הקוד והריצו.

(**שאלה 19**) שנו את הקוד כך שיציג (בסוף, במקום ההצגה הקיימת) את כל הספרות שתויגו לא נכון. לכל ספרה כזו הציגו: את התיוג המקורי, את התיוג השגוי, ואת התמונה שסווגה לא נכון.
צרפו את התוצאה הגרפית למסמך ההגשה. התוצאה צריכה להראות כך:



כתיבת קוד למציאת מאפיינים

בדוגמה שהרצתם הקלט הן תמונות קטנות (מטריצות 8 X 8).
כעת נתרגל עליהן בחירה של מאפיינים וסיווג בעזרתם.

(**שאלה 20**) הנחיות:

א. השתמשו בקוד של הדוגמה כבסיס לקוד שלכם.

ב. כתבו קוד המחלץ מאפיינים שונים (לפחות 5 ולא יותר מ 10). כל מאפיין ימומש בפונקציה המקבלת מטריצת תמונה ומחזירה ערך יחיד (סקאלר). תנו שמות משמעותיים לפונקציות.

רעיונות למאפיינים:

- סכום כל הערכים במטריצה.
- מדד סימטריה אנכית (למשל סכום ההפרשים בין המטריצה להיפוך ימין שמאל שלה).
- מדד סימטריה אופקית.
- שונות של סכום שורות המטריצה.
- שונות של סכום עמודות המטריצה.
- סכום האזור המרכזי במטריצה.
- השוני בין מרכז המטריצה להיקפה.
- מספר מדדי שוני בין רביעים של המטריצה (רביע ראשון לעומת שני, ראשון לעומת שלישי וכו).

ג. השתמשו במספר מערכים חד ממדיים כמספר המאפיינים לאיסוף המאפיינים עבור כל התמונות בסט הנתונים. תנו שמות משמעותיים למערכים.

ד. לצורך פשטות הסיווג, בחלק זה של התרגיל נתייחס רק לספרות "0" ו "1". כדי להמשיך רק עם הערכים הרלוונטיים מצאו את האינדקסים של הערכים 0 ו 1 בתוך digits.target כך

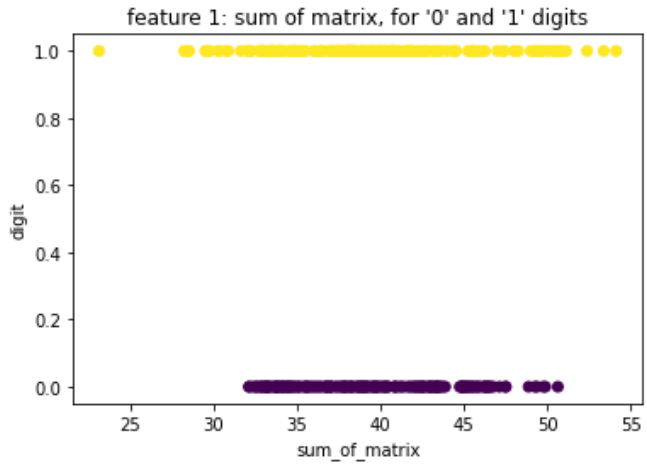
```
indices_0_1 = np.where(np.logical_and(digits.target >=0 , digits.target <= 1))
```

השתמשו בזה לשליפת הערכים מתוך מערכים אחרים. למשל:

```
digits.target[indices_0_1]
```

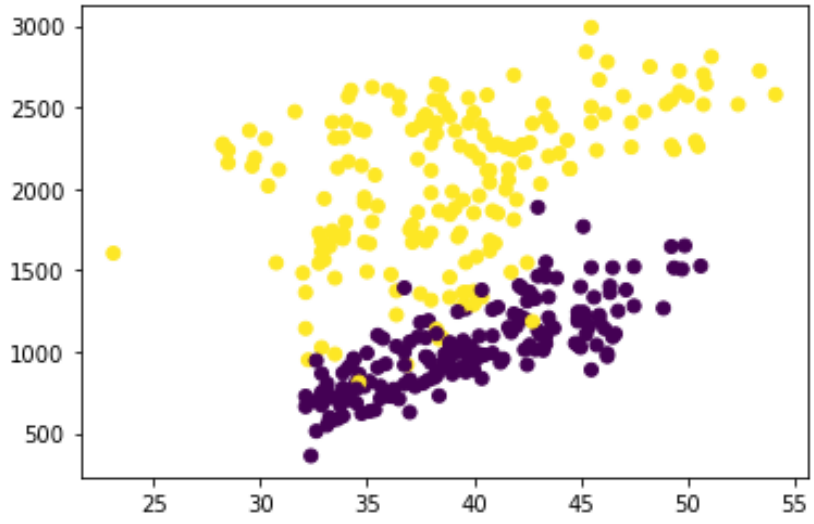
ה. באיזה צרופים של מאפיינים כדאי להשתמש לסיווג? לצורך בדיקה של המאפיינים ובחירת צרופים שלהם, הציגו את ערכי המאפיינים בצורה ויזואלית. צרפו למסמך ההגשה (Word) דוגמאות של figures כאלה.

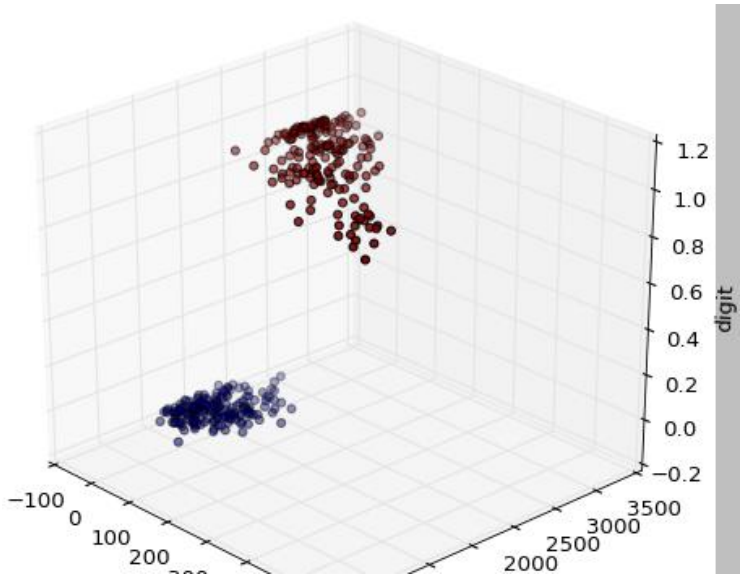
- הצגה של כל מאפיין בנפרד (חובה), עבור שתי קבוצות הספרות. בציר X הציגו את ערכי המאפיין, ובציר Y את התיוג (0 או 1). הוסיפו כיתוב לצירים וכותרות (שם המאפיין). **כאן מספר הfigures יהיה כמספר המאפיינים שמימשתם. הוסיפו לקובץ ההגשה הסבר מה כל מאפיין עושה.** דוגמה להצגה זו:



- הצגה של זוגות של מאפיינים עבור שתי הקבוצות. a) בציר X הציגו מאפיין ראשון, בציר Y מאפיין שני, ואת התיוג על ידי צבע הנקודות. או b) בציר X הציגו מאפיין ראשון, בציר Y מאפיין שני, ובציר Z את התיוג (0 או 1). כדאי להשתמש בצבע שונה לנקודות מכל קבוצה. **כיוון שמספר הצרופים גדול יותר, אין צורך לצרף את כל ה figures למסמך ההגשה. בחרו וצרפו לפחות 3 דוגמאות בהן נראית הפרדה טובה בין הקבוצות. הכותרת תהיה שמות המאפיינים.**

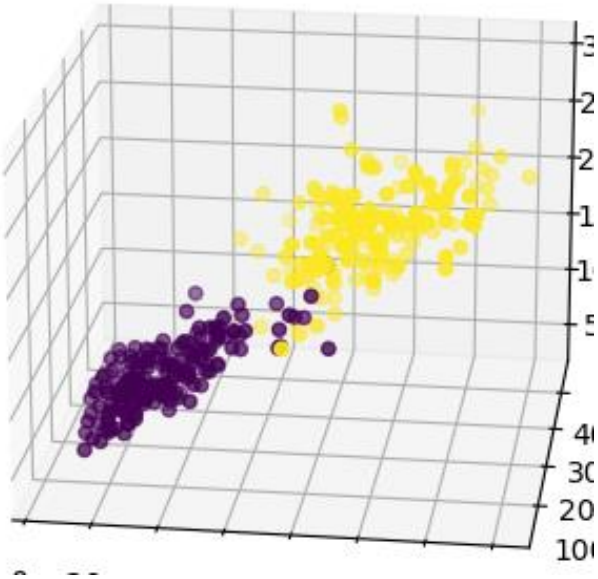
הנה דוגמאות לשתי האפשרויות שתוארו (זהות המאפיינים הוסתרה בכוונה):





- הצגת שלשות של מאפיינים עבור שתי הקבוצות. בציר X הציגו מאפיין ראשון, בציר Y מאפיין שני, ובציר Z מאפיין שלישי. את התיוג הציגו בעזרת צבע. **גם כאן אין צורך לצרף את כל האפשרויות למסמך ההגשה. בחרו וצרפו לפחות 3 דוגמאות בהן נראית הפרדה טובה בין הקבוצות. הכותרת תכלול את שמות שלושת המאפיינים.**

הנה דוגמה להצגה של שלושה מאפיינים (זהות המאפיינים הוסתרה בכוונה):



השתמשו בקוד הבא ליצירת תצוגה תלת ממדית (שנו את הקוד על פי צרוף המאפיינים שאתם בוחרים). שימו לב שהפונקציות שמייצרות את המאפיינים הופעלו על כל הסט, כך שיצרו מערכים בשמות המתאימים (בקוד כאן אלו המערכים featureA, featureB, featureC. אלה הם שמות גנריים, אל תשמשו בהם: **תנו שמות משמעותיים לפונקציות ולמערכים**).

```
fig = plt.figure()
fig.suptitle('YOUR TITLE', fontsize=14)
ax = fig.gca(projection='3d')
ax.scatter(featureA[indices_0_1], featureB[indices_0_1], featureC[indices_0_1],
c=digits.target[indices_0_1])
ax.set_xlabel('featureA')
ax.set_ylabel('featureB')
ax.set_zlabel('featureC')
fig.show()
```

להצגה אינטראקטיבית של ה plots רשמו ב console:

```
%matplotlib auto
```

ואז כל plot יוצג בחלון נפרד ותוכלו לסובב את התצוגה התלת ממדית ולמצוא זווית מבט נוחה להצגה.
לחזרה למצב הקודם רשמו ב console:

```
%matplotlib inline
```

ו. השתמשו במסווג logistic regression והציגו את ביצועי המסווג כך:

```
# creating the X (feature) matrix
X = np.column_stack((featureA[indices_0_1], featureB[indices_0_1]))

# scaling the values for better classification performance
X_scaled = preprocessing.scale(X)

# the predicted outputs
Y = digits.target[indices_0_1]

# Training Logistic regression
logistic_classifier = linear_model.LogisticRegression(solver='lbfgs')
logistic_classifier.fit(X_scaled, Y)

# show how good is the classifier on the training data
expected = Y
predicted = logistic_classifier.predict(X_scaled)


print("Logistic regression using [featureA, featureB] features:\n%s\n" % (
    metrics.classification_report(
        expected,
        predicted)))

print("Confusion matrix:\n%s" % metrics.confusion_matrix(expected, predicted))
```

```
# estimate the generalization performance using cross validation
predicted2 = cross_val_predict(logistic_classifier, X_scaled, Y, cv=10)

print("Logistic regression using [featureA, featureB] features cross
validation:\n%s\n" % (
    metrics.classification_report(
        expected,
        predicted2)))

print("Confusion matrix:\n%s" % metrics.confusion_matrix(expected, predicted2))
```

ז. הנדסו את המאפיינים הטובים ביותר, ובחרו את צרוף המאפיינים הטוב ביותר (מותר לבחור כמה מאפיינים שתרצו מתוך אלה שייצרתם).

במסמך ההגשה (Word): ציינו את שמות המאפיינים שבחרתם והסבירו מה הם עושים. העתיקו למסמך ההגשה את ביצועי המסווג שלכם בעזרת cross validation, בפורמט הבא. **ציון מלא לסעיף זה יתקבל עבור מספר שגיאות כולל 2 או פחות.**

```
Logistic regression using [featureA, featureB, featureC] features cross validation:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       178
           1       0.99      1.00      1.00       182

    accuracy                           1.00       360
   macro avg       1.00      1.00      1.00       360
weighted avg       1.00      1.00      1.00       360


Confusion matrix:
[[177   1]
 [  0 182]]
```

ח. תחרות!!! **(רשות)**

הפעילו את המסווג שלכם (עם עד 10 המאפיינים הטובים ביותר שתצליחו ליצור) על **כל הספרות**. הקפידו לציין את שמות המאפיינים ולהסביר מה הם עושים. דווחו על התוצאות בעזרת cross validation בפורמט של הסעיף הקודם.

בונוס עד 10 נקודות יינתן כתלות באיכות הסיווג.

החבילות השימושיות לתרגיל זה:

```
import matplotlib.pyplot as plt
import numpy as np

from sklearn import datasets, metrics
from sklearn import linear_model
from sklearn.model_selection import cross_val_predict
from sklearn import preprocessing
```

הגשה

א. תאריך הגשה:
נושא 1 – מדדי איכות, שאלות 1 עד 18:
עד יום ראשון, 4.12.22, בשעת חצות הלילה (בין ראשון לשני).

נושא 2 – הנדסת מאפיינים, שאלות 19 – 20:
עד יום שני 19.12.22, בשעת חצות הלילה (בין שני לשלישי).

ב. ניתן להגיש בזוגות.
ג. יש לכתוב **שם \ שמות + ת"ז** בראשית כל מסמך מוגש (**כולל בקבצי הקוד**).
ד. כל מגיש (ביחיד או בזוג) צריך לדעת להסביר כל מה שנעשה בפתרון המוגש. חלק מן המגישים ידרשו להסביר את הפתרון שלהם למרצה.
ה. לכל נושא יש להגיש מסמך Word המכיל את התשובות לתיבת ההגשה.
שם המסמכים יהיה **ex2b.docx , ex2a.docx**.
לכל נושא תהיה תיבת הגשה משלו.
הקפידו שמספור סעיפי התשובות שלכם יהיה זהה למספור סעיפי השאלות.
ו. לכל פונקציה צריך להיות תיעוד.
ז. יש להגיש את כל הקוד לתרגיל בקובץ יחיד בשם **ex2.py**. בתחילת הקובץ יבואו הגדרות כל הפונקציות. בהמשך הקובץ יבוא חלק ההרצה. חלק זה **יופרד על ידי הערות לכל אחד מסעיפי השאלות.**