מערכות לומדות תשע"ז - 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.

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.

בבעית 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 האוכלוסיה 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 באוכלוסיה.

(**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.**

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.
בהמשך הקורס נתאר שיטות לטפל בסט לא מאוזן.

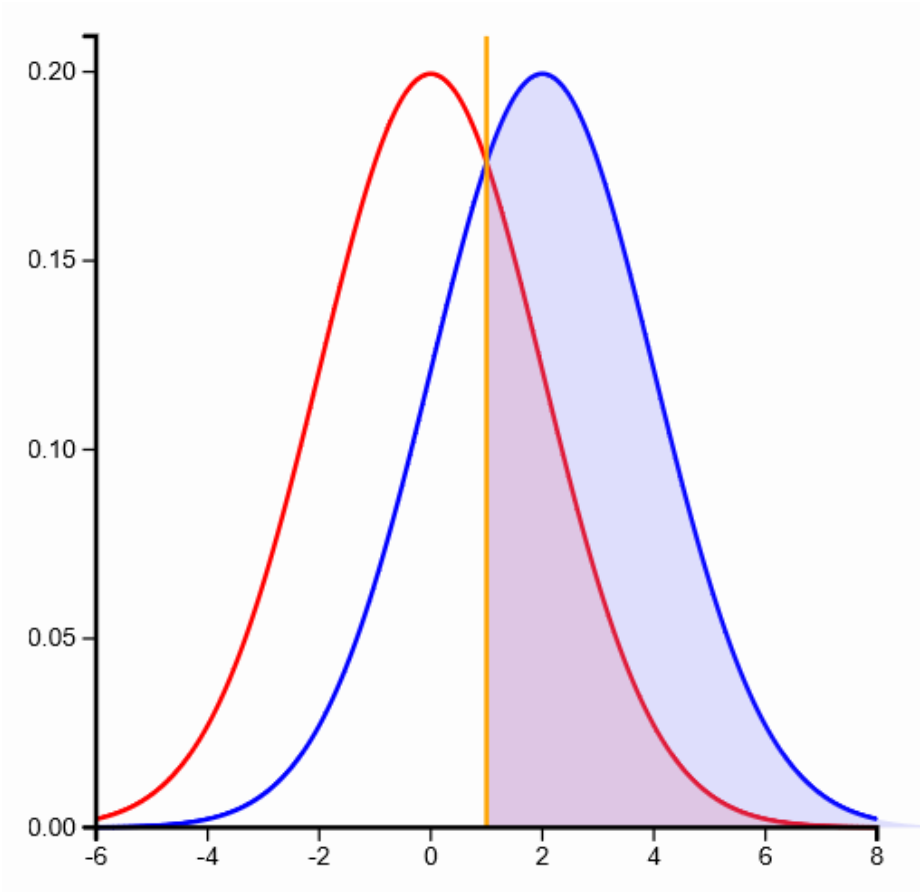
שיטה מקובלת להשוואת ביצועי מסווגים בינאריים היא עקומת ROC
(Receiver Operating Characteristics curve)

למעשה אנו משווים בעזרת שיטה זו את טיב ההפרדה בין הערכים של מאפיין עבור שני התיוגים.

קיראו על עקומת ROC.
שחקו בהדגמה האינטראקטיבית כאן
http://arogozhnikov.github.io/2015/10/05/roc-curve.html

כדי להבין את השיטה עליכם לברר (לעצמכם) על פי ההנחיות הבאות. הסתכלו על איור 1.

A. במשימת תיוג, מה זו העקומה הכחולה ומה זו העקומה האדומה ומה הוא ציר ה X?
רמז – העקומות הן דוגמאות לאחד המרכיבים שהזכרנו בחוק בייז בהקשר לתיוג.
B. מה השטח מתחת לעקומה הכחולה? לאדומה?
C. מה הוא הקו הצהוב? האם הזזה שלו משפיעה על הנתונים במשימת התיוג?
D. מה הוא השטח שמימין לקו הצהוב ומתחת לעקומה הכחולה. רמז – התשובה היא במונחים של שאלה 4.
E. מה הוא השטח שמימין לקו הצהוב ומתחת לעקומה האדומה? רמז – שאלה 4.
F. זהו את השטחים המייצגים את שני סוגי השגיאה במשימת תיוג.



איור 1

G. באיור 2, העקומה הכחולה זזה ימינה. המשמעות היא שהערכים של המאפיין השתנו (ועל כן הנתונים השתנו). ודאו שכל תשובותיכם לסעיפים הקודמים (A-F) הגיוניות ומתאימות לשינוי.

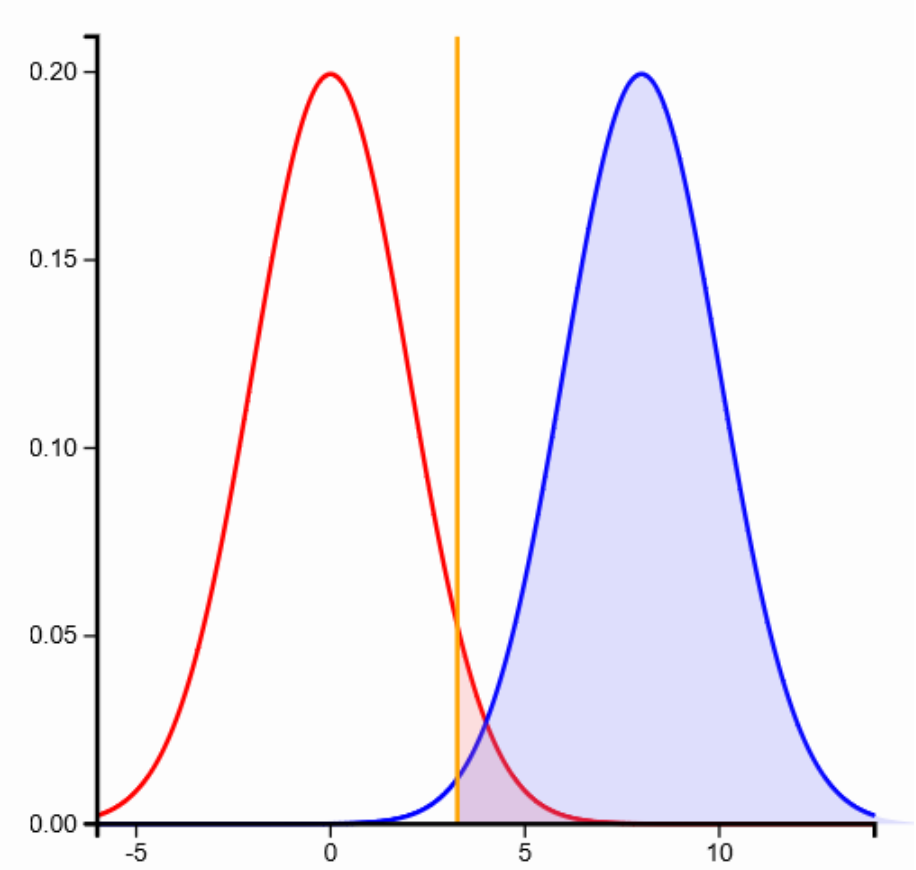
H. שימו לב שכאן בחרתי ערך אחר לקו הצהוב. זהו את שגיאות התיוג כעת. האם מיקום הקו הצהוב הוא זה הממזער את שגיאות התיוג?

I. שימו לב שבאיור 1 הייתה חפיפה גדולה בין העקומות, ובאיור 2 החפיפה קטנה בהרבה. "זו תכונה של המאפיין שבחרנו ולא של החלטת הסיווג!" האם אתם מסכימים עם משפט זה?

J. הסתכלו על הצירים של עקומת ROC. איך הם קשורים לשטחים המוצגים באיורים?

K. הזיזו את הקו הצהוב ימינה ושמאלה וראו כיצד הזזתו מובחנת על גבי עקומת ROC.

L. מתי מתקבלת עקומת ROC של קו אלכסוני מ 0,0 ל 1,1 ?



איור 2

כעת, אחרי שאתם מבינים את המרכיבים של העקומה, ענו:

(שאלה 13) כיצד נראית העקומה כאשר הנתונים (של שתי המחלקות) חופפים בצורה מושלמת? **הסבירו.**

(שאלה 14) כיצד נראית העקומה כאשר הנתונים של שתי המחלקות **כלל** אינם חופפים? **הסבירו.**

(שאלה 15) מה המשמעות של עקומת ROC הנמצאת מתחת לאלכסון? מה זה אומר על יכולת הסיווג בעזרת המאפיינים שבחרנו? כיצד אפשר לתקן זאת בקלות ולהעביר את העקומה להיות מעל לאלכסון? **הסבירו.**

נושא 2 – הנדסת מאפיינים

כאשר רוצים לבצע משימת למידה במרחב קלט ממימד גבוה (למשל תמונות) יש צורך לרוב בהרבה מאוד נתונים. כדי לצמצם את הצורך בכמות נתונים גדולה מדי מנסים להוריד את המימד 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 ממימד גבוה מאשר מימד הקלט. כאן נשתמש ב"טריק" מתמטי המאפשר לבצע זאת ללא חסרונות של מימד גבוה. עוד על כך בהמשך הקורס כשנזכיר SVM – Support Vector Machines ו Kernel Methods.

אם כך הקריטריונים לבחירת סט מאפיינים טוב הם:

א. אם מרחב הקלט ממימד גבוה מדי, הסט מצמצם את מימד הקלט בצורה רבה.
לדוגמה: בנסיון 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 לבעית 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 מימד 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:

A selection from the 64-dimensional digits dataset

הסתכלו בקוד הנמצא כאן
http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html
קטע קוד זה משתמש במסווג מסוג support vector machine לביצוע הסיווג.

הורידו את הקוד והריצו. הסתכלו על התוצאות - שתי הטבלאות המוצגות בסוף הרצת הקוד - בראשונה מדדי איכות שונים לכל ספרה ובשניה Confusion matrix.

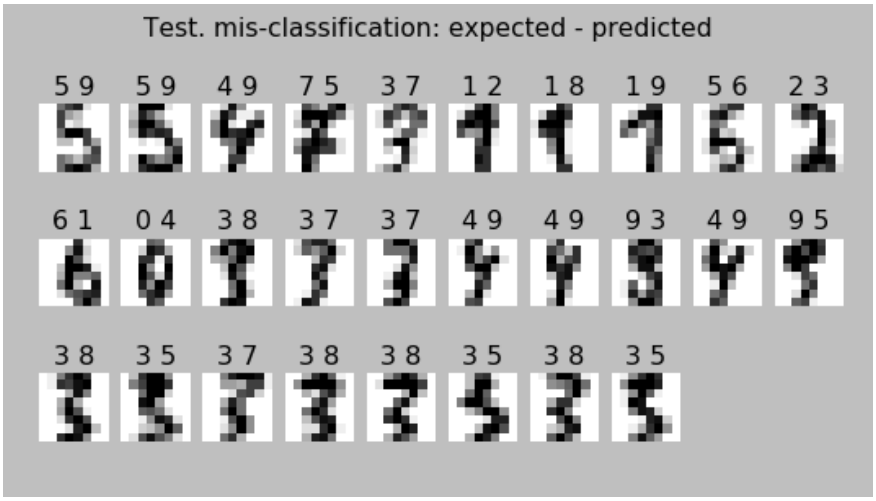
(**שאלה 16**) מה ה recall של הספרה 8? מה המשמעות וכיצד רואים זאת ב confusion matrix ?

(**שאלה 17**) מה ה precision של הספרה 0? מה המשמעות וכיצד רואים זאת ב confusion matrix ?

(**שאלה 18**) כמה ספרות 5 סווגו בטעות כספרה אחרת? לאילו ספרות הן סווגו?

(**שאלה 19**) איזו ספרה סווגה הכי הרבה פעמים בטעות כספרה אחרת? כמה טעויות כאלה היו? לאילו ספרות סווגה?

(**שאלה 20**) 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.
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 21**) הנחיות:

א. השתמשו בקוד של הדוגמה כבסיס לקוד שלכם.
ב. כיתבו 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 האיזור 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 מימדיים 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 הרלוונטים מיצאו 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:

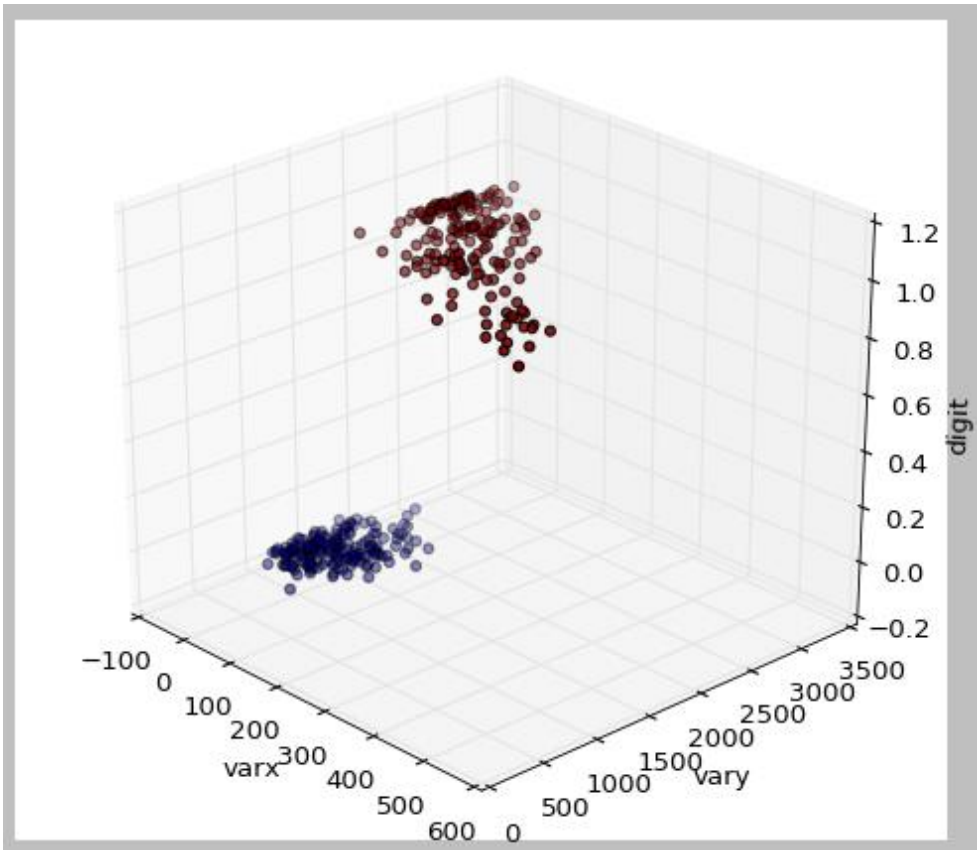
```
digits.target[indices_0_1]
```

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. השתמשו באחת האפשרויות:

- הצגה של כל מאפיין בנפרד, עבור שתי קבוצות הספרות. בציר X הציגו את ערכי המאפיין, ובציר Y את התיוג (0 או 1). הוסיפו כיתוב לצירים וכותרות. כאן מספר הfigures יהיה כמספר המאפיינים שמימשתם.

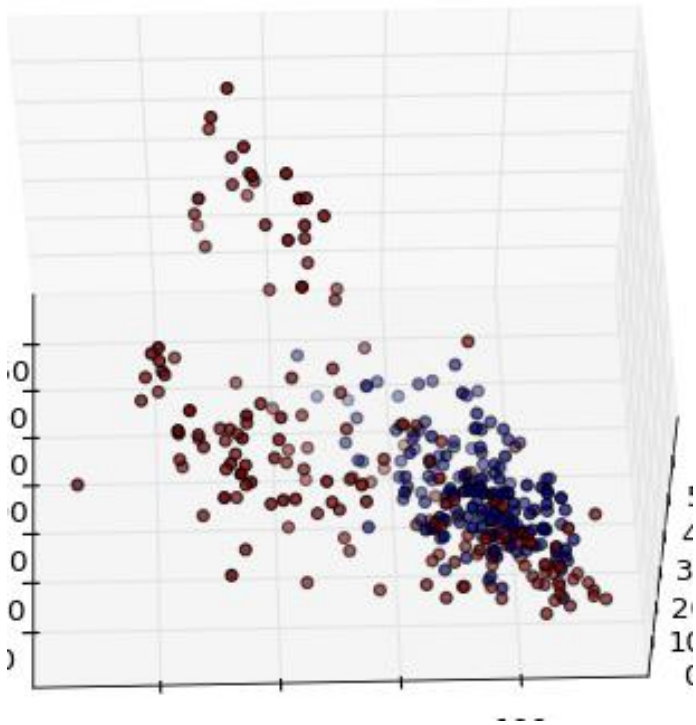
- הצגה של זוגות של מאפיינים עבור שתי הקבוצות. a) בציר X הציגו מאפיין ראשון, בציר Y מאפיין שני, ואת התיוג על ידי צבע הנקודות. או b) בציר X הציגו מאפיין ראשון, בציר Y מאפיין שני, ובציר Z את התיוג (0 או 1). כדאי להשתמש בצבע שונה לנקודות מכל קבוצה. כיוון שמספר הצרופים גדול יותר, אין צורך לצרף את כל ה figures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

הנה דוגמה להצגה כזו:



- 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 דוגמאות בהן נראית הפרדה טובה בין הקבוצות.

הנה דוגמה להצגה של שלושה מאפיינים שבה אין הפרדה טובה:



השתמשו בקוד זה ליצירת תצוגה תלת מימדית (שנו את הקוד על פי צרוף המאפיינים שאתם בוחרים)

```
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()
```

ו. השתמשו במסווג logistic regression:

```
# 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()
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).
דווחו (במסמך ההגשה - Word) על ביצועי המסווג שלכם בעזרת cross validation, בפורמט הבא. הקפידו לציין את שמות המאפיינים

```
 Logistic regression using [A,B,C, ... ] features cross validation:
             precision    recall  f1-score   support

          0       0.99      0.99      0.99       178
          1       0.99      0.99      0.99       182

avg / total       0.99      0.99      0.99       360


Confusion matrix:
[[177   1]
 [  2 180]]
```

ח. תחרות!!! **(רשות)**
הפעילו את המסווג שלכם (עם עד 10 המאפיינים הטובים ביותר שתצליחו ליצור) על **כל הספרות**. דווחו על התוצאות בעזרת cross validation בפורמט של הסעיף הקודם.
בונוס ינתן כתלות באיכות הסיווג.

החבילות שיש לכלול בתרגיל זה:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

from sklearn import datasets, svm, metrics
from sklearn import linear_model
from sklearn.cross_validation import cross_val_predict
from sklearn import preprocessing
```

הגשה

א. תאריך הגשה: עד יום ראשון, 1.1.17, בשעת חצות הלילה.
ב. ניתן להגיש בזוגות.
ג. יש לכתוב **שם \ שמות + תז** 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. יש להגיש מסמך Word המכיל את כל התשובות לתרגיל. שם מסמך זה יהיה **ex2.docx**.
הקפידו שמיספור סעיפי התשובות שלכם יהיה זהה למיספור 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.**
ח. שני הקבצים ישכנו בתוך תיקייה הכוללת את שמכם.
שם התיקיה למגיש יחיד:
EX2FamilyName
שם התיקיה לשני מגישים:
EX2Family1Family2
התיקיה תארז לקובץ **zip** בעל אותו שם כשל התיקיה (לא rar).