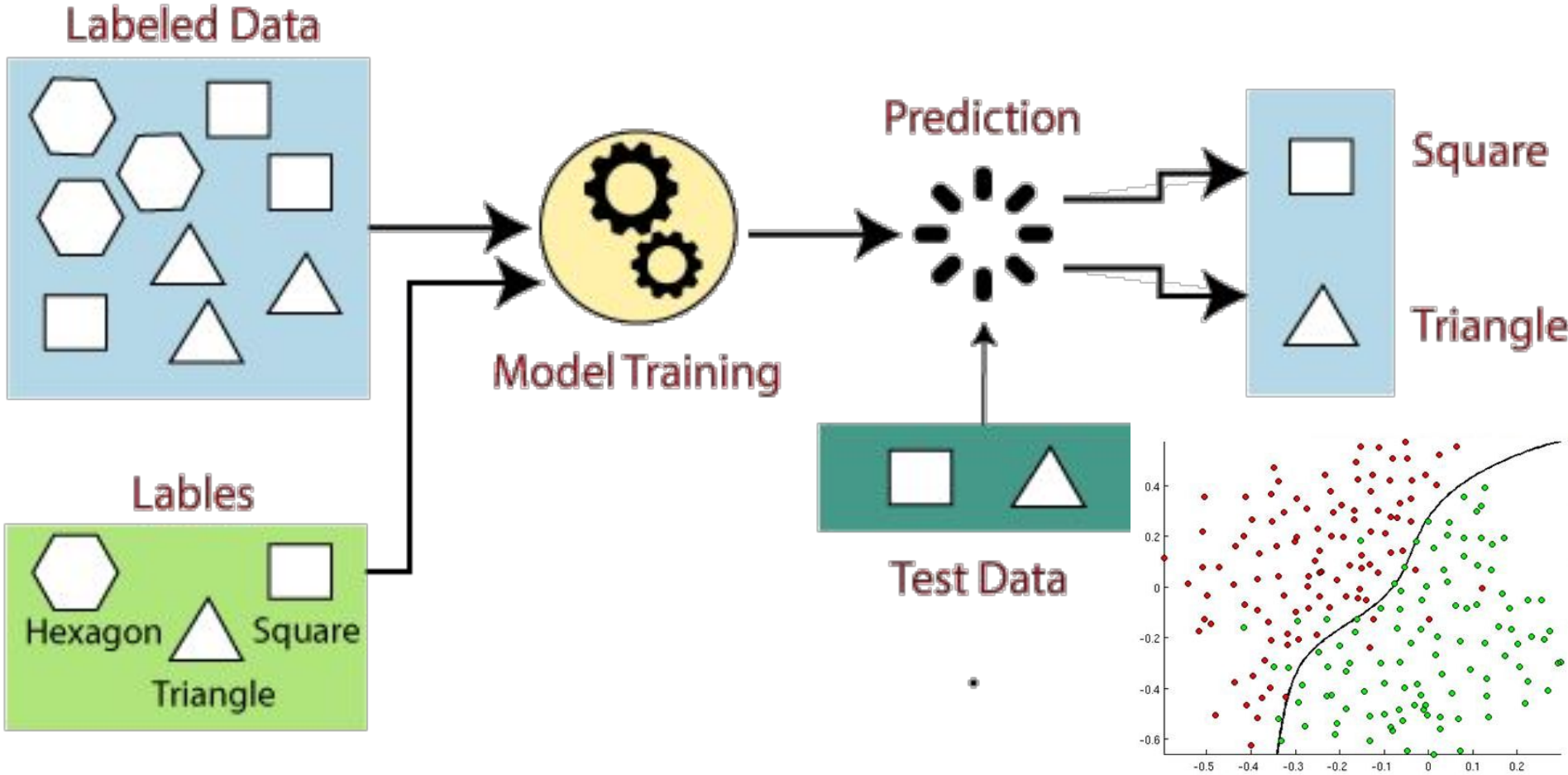
# תכנות פרספטרון

## תרגול בסיווג תמונות

גדי הרמן

# Classification

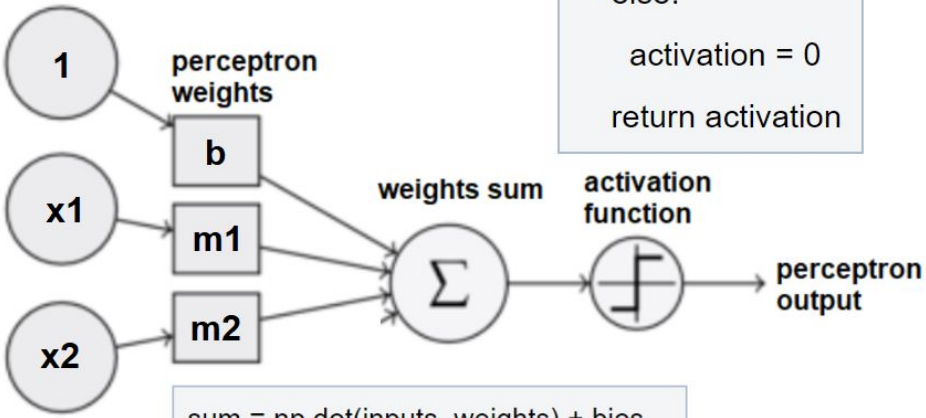


## שלבים בלמידת מכונה

- שלב 1: (הלמידה/אימון)
  - המכונה מקבלת DATA ו- LABELS.
  - המכונה מחשבת ערכים לפונקציה ידוע (לדוגמה: ערכים של m ו- b עבור פונקציה קווית)
- שלב 2: (החיזוי)
  - המכונה קבלת DATA חדש ומחשבת בעזרת הנוסחה והערכים שנקבעו בשלב האימון את הפלט.

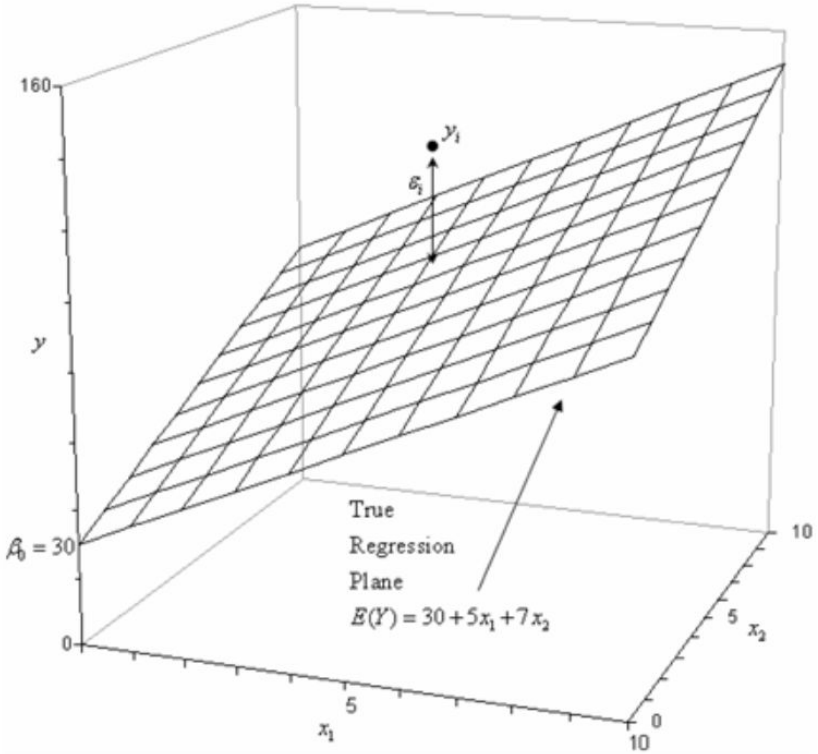
# פרספטרון כמסווג לינארי הכולל מספר מבואות

```
def Activation(s):
    if s > 0:
      activation = 1
    else:
      activation = 0
    return activation
```




```
sum = np.dot(inputs, weights) + bios
```

```
y=m1x1+m2x2+b
```



```python
import numpy as np
class Perceptron(object):
  def __init__(self, numOfInputs, epochs=100, learningRate=0.01):
    self.epochs = epochs
    self.learningRate = learningRate
    self.weights = np.zeros(numOfInputs)
    self.bios = 1

  def Activation(self, s):
    if s > 0:
      activation = 1
    else:
      activation = 0
    return activation

  def predict(self, inputs):
    sum = np.dot(inputs, self.weights) + self.bios
    out = self.Activation(sum)
    return out
```

# הפעולה train

$$m = m - err \cdot x \cdot Learning\ Rate$$

$$b = b - err \cdot Learning\ Rate$$

```
def train(self, inputs, labels):
    for _ in range(self.epochs):
        for i in range(len(inputs)):
            prd = self.predict(inputs[i])
            self.weights -= (prd - labels[i]) * inputs[i] * self.learningRate
            self.bios -= (prd - labels[i]) * self.learningRate
```

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

לצורך זיהוי פריטים בתמונה או בקיצור זיהוי תמונות על ידי מכונה לומדת יש צורך לארגן מערך של תמונות מתויגות.

נבחן את הסוגיה על ידי התבוננות על התמונה הבא:

האם כל הלבנים חתולים?

האם כל בעלי הפרווה כלבים?

האם לכל מי שיש אף בצורת משולש הוא חתול?

האם כל מי שיש לו אוזן ורודה הוא כלב?

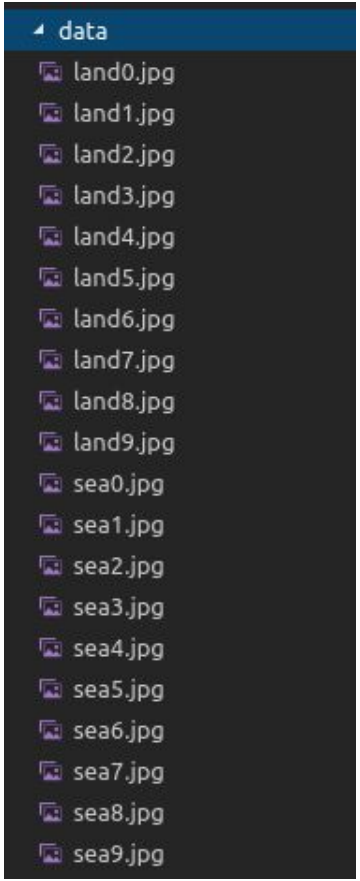
# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

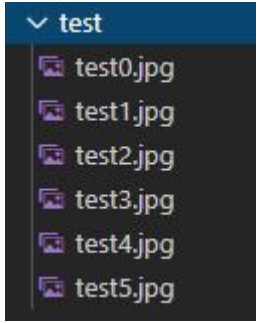
נראה שיש צורך לחדד את הנושא ולעזור לנו לאפיין את הגורמים המבדילים בין פריטים. תהליך זה נקראה במכונות לומדות: Feature Selection in Machine Learning והוא בא לספק לנו כלי להגדרת המאפיינים שיכולים להבדיל בין דברים שאנו רוצים לסווג.

לצורך התנסות בנושא ובמטרה לתרגל עבודה עם תמונות נחפש תכונות Features שיכולים להבדיל בין תמונות ים לבין תמונות יבשה. בקיצור משחק ים יבשה בגרסת מכונה לומדת (נו טוב לא ממש דומה למשחק הילדים ששיחקנו פעם).

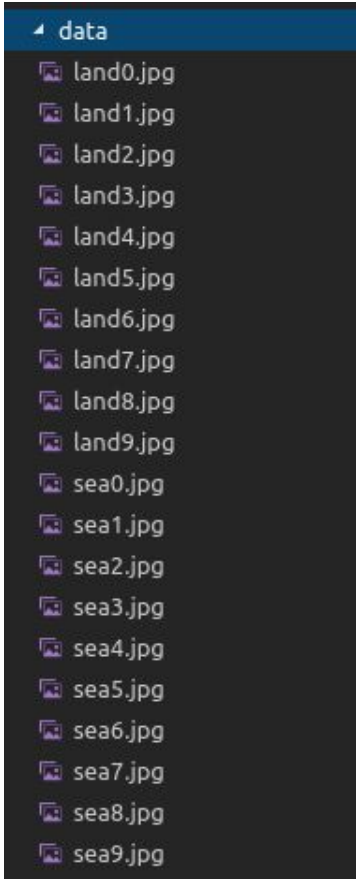
נוריד מהאינטרנט 10 תמונות של ים ו-10 תמונות של יבשה ונשמור אותם בתיקייה בשם data תחת השמות sea ומספר התמונה עבור כל התמונות המתארות ים ו- land ומספר התמונה עבוד כל התמונות המתארות יבשה.

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות




# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות



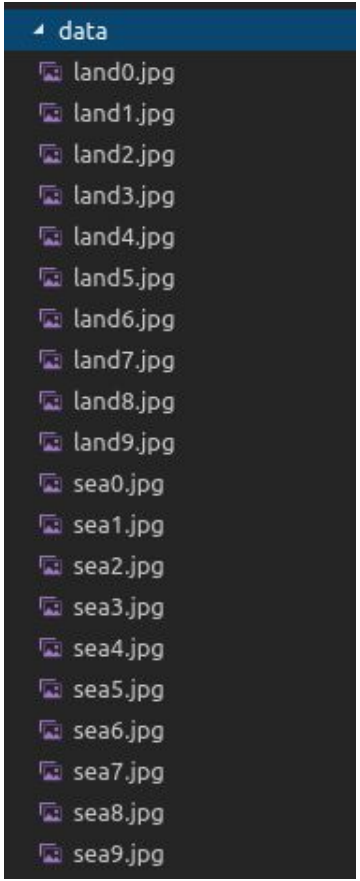

**שלב 1:**

נבנה יחד 2 רשימה הכוללת עבור כל תמונה את ממוצע הצבעים. רשימה ראשונה עבור תמונות הים והשנייה עבור תמונות היבשה.

להלן דוגמת קוד:

```
sea_colors = []
for i in range(10):
    img = Image.open("data/sea" + str(i) + ".jpg")
    img.load()
    data = np.array(img, dtype=np.uint8)
    t = []
    for i in range(3):
        t.append(data[:,:,i].sum() / data[:,:,i].size)
    sea_colors.append(t)
```

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות



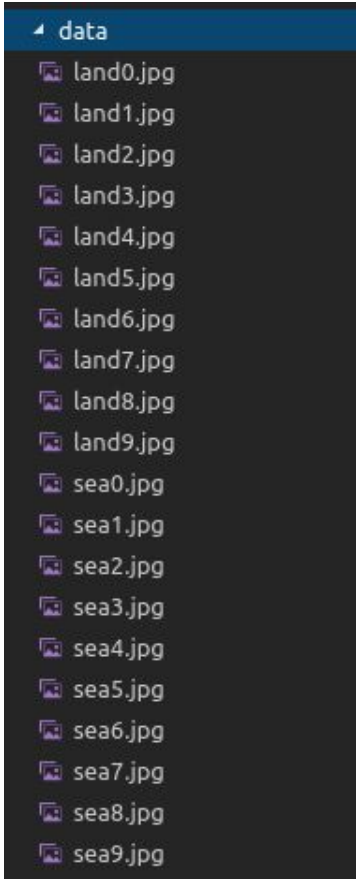

**שלב 1:**

נבנה יחד 2 רשימה הכוללת עבור כל תמונה את ממוצע הצבעים. רשימה ראשונה עבור תמונות הים והשנייה עבור תמונות היבשה.

להלן דוגמה לפלט תקין:

```
sea_array [[ 58.78777751 130.41016153 186.70774648]
 [ 76.40140326 131.96297276 165.40403703]
 [122.35404023 154.34126688 196.69888385]
 [ 54.02722046 111.70695876 149.57194687]
 [ 92.76308246 115.0151928  128.01915012]
 [130.83236134 155.91324931 191.47257662]
 [ 82.18283177 129.21721615 151.26773854]
 [ 94.20739142 135.44244079 169.82846913]
 [ 96.15785348 137.06029745 168.99923701]
 [ 55.11920833 107.63084549 142.60709983]]
land_array [[128.87910584 121.23325928 100.96659791]
 [ 81.96936957  86.22154514  36.11645182]
 [114.53656012 147.22994865  87.28119651]
 [101.77011797 117.40772187  84.52560156]
 [120.73358401 122.99220265  95.46269543]
 [188.621389   206.63995731 150.15271712]
 [118.8398609  154.7234774  152.29637357]
 [ 87.06099206 111.65416667  15.36093254]
 [ 93.67001984 135.62204365  97.56781746]
 [ 74.21667163  93.52566319  98.54418281]]
```

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

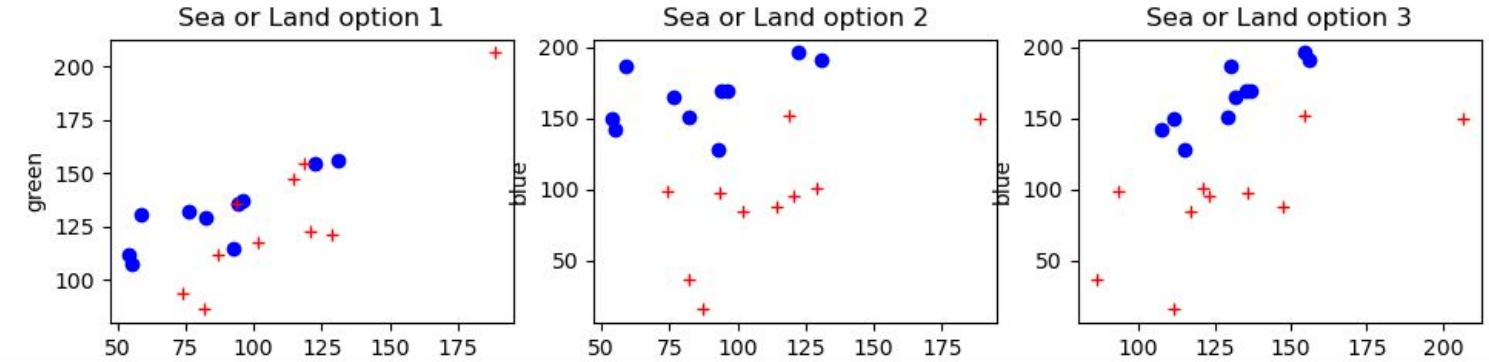



**שלב 2:**

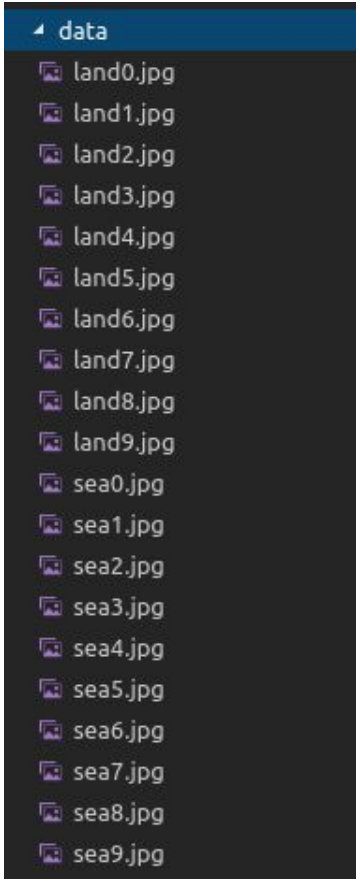
בנו 3 גרפים המסווגים באופן שונה את ממוצעי הצבעים באופן הבא:

1. אדום ירוק
2. אדום כחול
3. כחול ירוק

להלן דוגמה לפלט תקין:

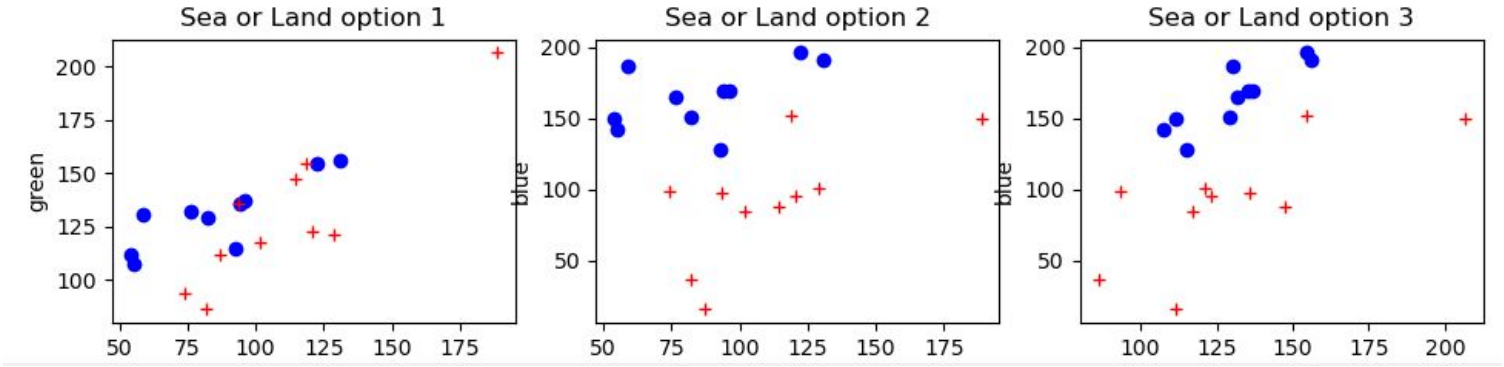


# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות




**שלב 3:**

תקבעו מהם התכונות Features שיכולים להבדיל בין תמונות ים לבין תמונות יבשה.



# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

```
#--------start Perceptron--------------------------------
class Perceptron(object):
  def __init__(self, numOfInputs, epochs=100, learningRate=0.01):
    self.epochs = epochs
    self.learningRate = learningRate
    self.weights = np.zeros(numOfInputs)
    self.bios = 1

  def Activation(self, s):
    if s > 0:
      activation = 1
    else:
      activation = 0
    return activation

  def predict(self, inputs):
    sum = np.dot(inputs, self.weights) + self.bios
    out = self.Activation(sum)
    return out

  def train(self, inputs, labels):
    for _ in range(self.epochs):
      for i in range(len(inputs)):
        prd = self.predict(inputs[i])
        self.weights -= (prd - labels[i]) * inputs[i] * self.learningRat
        self.bios -= (prd - labels[i]) * self.learningRate
#--------end Perceptron----------------------------------
```

**שלב 4:**

העתיקו לתוך קוד התוכנית שלכם את מימוש המחלקה Perceptron כפי שלמדנו.

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

**שלב 5:**

הכינו את הנתונים למבנה המתאים להיכנס

כ- data ו- label לתוך הפרספטרון.

```
data (all training data (green and blue)):
 [[130.41016153 186.70774648]
 [131.96297276 165.40403703]
 [154.34126688 196.69888385]
 [111.70695876 149.57194687]
 [115.0151928  128.01915012]
 [155.91324931 191.47257662]
 [129.21721615 151.26773854]
 [135.44244079 169.82846913]
 [137.06029745 168.99923701]
 [107.63084549 142.60709983]
 [121.23325928 100.96659791]
 [ 86.22154514  36.11645182]
 [147.22994865  87.28119651]
 [117.40772187  84.52560156]
 [122.99220265  95.46269543]
 [206.63995731 150.15271712]
 [154.7234774  152.29637357]
 [111.65416667  15.36093254]
 [135.62204365  97.56781746]
 [ 93.52566319  98.54418281]]
```

להלן דוגמה לפלט תקין:

```
lbl:
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

**שלב 6:**

הכינו מערך נתונים נוסף לשלב הבדיקה, מערך המבוסס על 6 תמונות מתוך תיקיית Test

```
test_data (all testing data (green and blue)):
 [[101.40009948 133.04162356]
 [142.22322273  70.25658182]
 [184.54570305 228.31970038]
 [133.08846498 105.62144064]
 [155.4776512   96.15760848]
 [132.79631621  86.55477991]]
```

להלן דוגמה לפלט תקין:

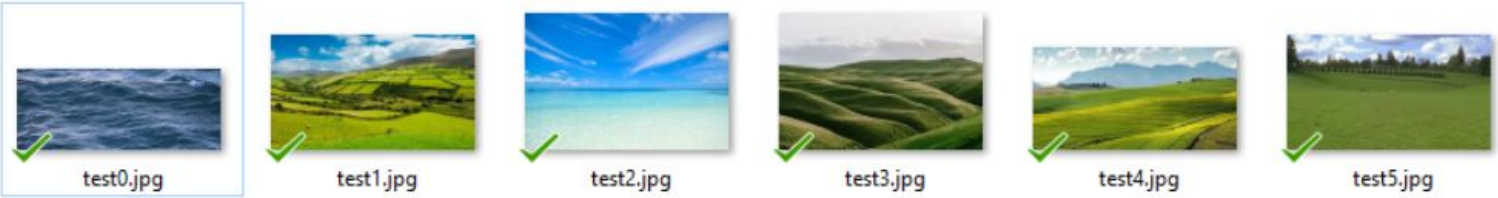
```
testlbl:
 [0, 1, 0, 1, 1, 1]
```

# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

**שלב 7:**

בצעו אימון של פרספטרון על מערך נתוני האימון:

```
#--------start run machine learned-----
perceptron = Perceptron(2)
perceptron.train(data, lbl)
```

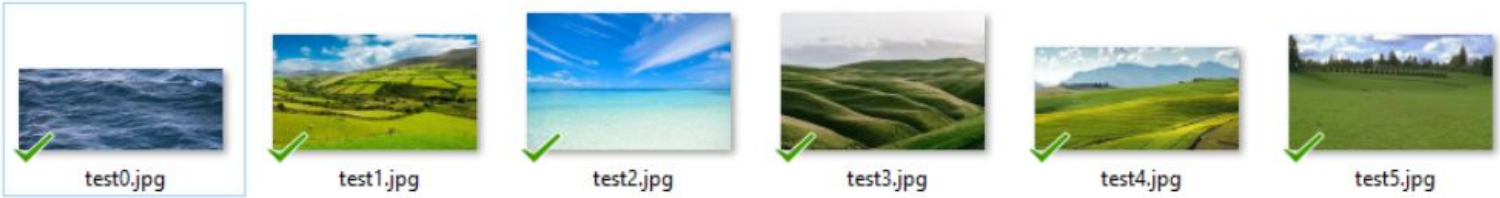


# מכונה לומדת מבוססת פרספטרון לזיהוי תמונות

**שלב 8:**

לאחר שלב האימון ביצעו בדיקה על ידי הפעולה predict.

```
[101.40009948 133.04162356] SEA Picture !!!
[142.22322273  70.25658182] LAND Picture !!!
[184.54570305 228.31970038] SEA Picture !!!
[133.08846498 105.62144064] LAND Picture !!!
[155.4776512   96.15760848] LAND Picture !!!
[132.79631621  86.55477991] LAND Picture !!!
```




# משימה 3 במטלה