

## רגרסיה ליניארית עם המשוואות הנורמליות

היום אנחנו נתרגל את הפתרון של רגרסיה ליניארית בעזרת Normal Equation.

לצורך כך, אנחנו נצור שלושה סטים של דטה סינטטי שכל אחד מהם מורכב מ 10 או 20 נקודות רנדומליות שמקיימות משוואות שונות **בתוספת רעש**, ואז נבנה את המשוואות:

- הלייבלים של הדטהסט הראשון יהיו נקודות שמקיימות את משוואת הישר שעוברת דרך ראשית הצירים בתוספת רעש.
- הלייבלים של הדטהסט השני יהיו נקודות שמקיימות את משוואת הישר שלא עוברת דרך ראשית הצירים בתוספת רעש.
- הלייבלים של הדטהסט השלישי יהיו נקודות שמקיימות את משוואת הפרבולה בתוספת רעש.

1. תחילת נתרגל בניית מערכים של נקודות אקראיות. היעזרו ב:

<https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.random.normal.html>

- a. בנו וקטור של עשרה מספרים שלמים רנדומליים
- b. בנו וקטור של עשרה floats – אפשר להשתמש בו כדטה שלנו, כלומר ה- $X$ , בהשמך התרגיל
- c. בנו וקטור של חמישה מספרים אקראיים שהם כפולה של שלוש
- d. שאלת אתגר: בחרו באקראיות מספר ששייך לעשרת המספרים הראשונים של סדרת פיבונאצ'.

2. הכנת הלייבלים:

### עבור הדטהסט הראשון:

- a. בנו סט של עשר נקודות אקראיות הנמצאות על קו ישר אחד העובר דרך ראשית הצירים. רמז: בחרו בשיפוע (לדוגמא 2) והכפילו בסט של 10 נקודות אקראיות על ציר ה- $y$  (אלדוגמא בנקודות מ. 1b) שמרו את התוצאה במשתנה בשם `first_array`.
- b. הוסיפו רעש גאוסיאני לכל אחד מהנקודות. רמז:

<https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.random.normal.html>

### עבור הדטהסט השני:

- c. בנו וקטור אחר של עשר נקודות אקראיות על ישר אחד, שלא עובר דרך ראשית הצירים (בחרו קבוע אחר שהוא השיפוע והכפילו בו ועוד קבוע והוסיפו אותו לכל הנקודות). שימרו את התוצאה בשם `second_array` והוסיפו רעש גאוסיאני.

### עבור הדטהסט השלישי:

d. בנו וקטור של עשרים נקודות שנמצאות על פרבולה. שימרו את התוצאה בשם `third_array` והוסיפו רעש גאוסיאני.

3. תרגול פעולות במטריצות:

a. בנו שתי מטריצות של  $4 \times 4$

b. הכפילו אותם זו בזו

c. מצאו `inverse` ו `transpose` של התוצאה

4. חישוב רגרסיה:

### התאמה לדטהסט הראשון:

מצאו התאמה לינארית לנקודות של `first_array` ע"י הצבה במשוואת הרגרסיה הלינארית שלמדנו

$$\hat{h} = (x'x)^{-1} x'y$$

כאשר  $x$  היא מטריצה באורך כמות הדגימות לאימון (10) וברוחב כמות הפיצ'רים: 1 עבור `first_array` ויותר עבור השאר. (אם בחרתם להשתמש ב `b` בשאלה 2 אז מטריצת  $x$  תהיה מורכבת מ וקטור `b` וכמות הוקטורים `ones` בהתאם למספר הפיצ'רים).  $y$  - הוא ווקטור הנקודות שיצרתם ב 12. זהו הפתרון שמגדיר את הקבועים של ההתאמה הלינארית. שימו לב, אם  $x$  הוא ווקטור חד מימדי, צרו ממנו ווקטור דו מימדי בעזרת פקודת `reshape` של `numpy`.

### התאמה לדטהסט השני:

5. עשו התאמה לישר ב). `second_array` רמז: חישובו בכמה פיצ'רים השתמשתם בשביל לייצר את הישר).

בשביל לייצר את ווקטור ה אהיעזרו ב `column stack` של `numpy`.

[https://docs.scipy.org/doc/numpy/reference/generated/numpy.column\\_stack.html](https://docs.scipy.org/doc/numpy/reference/generated/numpy.column_stack.html)

שיאפשר לחבר את ווקטור ה א עם ווקטור ה `ones`.

### התאמה לדטהסט השלישי:

6. עשו התאמה של הנקודות ב `third_array` לפרבולה. רמז: הדרך לעשות את זה היא להרחיב לשני פיצ'רים שאחד הוא  $x$  והשני הוא  $x^2$ . אהציגו התוצאה בעזרת `matplotlib`.

7. הצגת התוצאות: הציגו בנפרד את הנקודות של `first_array` קו ההתאמה שקיבלתם בעזרת ספריית `Matplotlib`. לינק להסברים: [http://matplotlib.org/users/pyplot\\_tutorial.html](http://matplotlib.org/users/pyplot_tutorial.html)

רמז: קו ישר ניתן לייצר בעזרת `linspace` ומשוואת הישר, בדומה לאיך שייצרתם את `second_array` רק בלי להוסיף רעש.

### 8. שאלת אתגר

המשוואה הבאה מייצרת את סט הנקודות:

$$y = ae^{bx^2+cx}$$

x=[0.08750722,0.01433097,0.30701415,0.35099786,0.80772547,0.16525226,0.46913072,0.69021229,0.84444625,0.2393042,0.37570761,0.28601187,0.26468939,0.54419358,0.89099501,0.9591165,0.9496439,0.82249202,0.99367066,0.50628823]

y=[4.43317755,4.05940367,6.56546859,7.26952699,33.07774456,4.98365345,9.93031648,20.68259753,38.74181668,5.69809299,7.72386118,6.27084933,5.99607266,12.46321171,47.70487443,65.70793999,62.7767844,35.22558438,77.84563303,11.08106882]

מצאו את a,b,c במשוואה.

רמז 1: האם צריך את האקספוננט, איך נפטרים ממנו?

רמז 2: מה יקרה עם נפעיל לוג על שני האגפים?