Machine learning

Linear regression

Exercise VII

פיתוח:
ד"ר יהונתן שלר
משה פרידמן
עידן טוביס

קרדיט:
ד"ר יונתן רובין
,Dr. Andrew Ng,
ד"ר קובי מייק
ואחרים

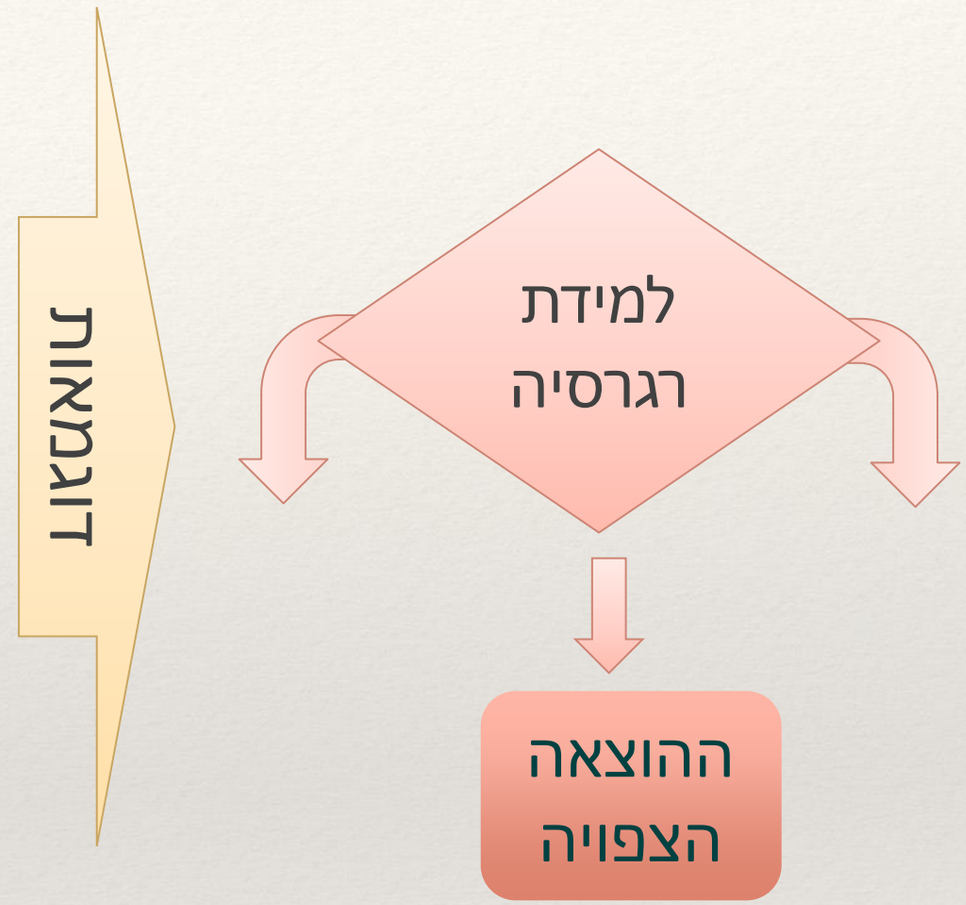
למידת מכונה: רגרסיה Regression

- חיזוי ערכו של משתנה רציף על פי ערכם של משתנים הקשורים אליו
- למשל: כמה כסף תוציא המשפחה בחו"ל (כמובן ברגע שתגמר תקופת הקורונה)
- שימו לב שזו אינה בעיית סיווג
- אין כאן מספר קטגוריות שהאלגוריתם צריך להכריע ביניהם
- כאן אנו מחפשים ערך חיובי כלשהו שיחזה בצורה הטובה ביותר את הוצאות המשפחה
- בעיה כזו נקראת בעיית רגרסיה (למה נקראת כך נראה בהמשך)

קרדיט: ד"ר אוהד סיוון

למידת רגרסיה מדוגמאות

הוצאות	ארץ יעד	ימים בטיול	מספר נפשות	שם
2000	איטליה	5	4	כהן
1000	תאילנד	12	2	רבין
1500	ניו יורק	2	5	שמיר
10000	בלגיה	10	12	שרון
1200	אנגליה	3	3	רמון
1500	מצרים	21	6	לוי
800	סין	7	3	בן דוד
2500	אוסטריה	10	1	מיכאלי



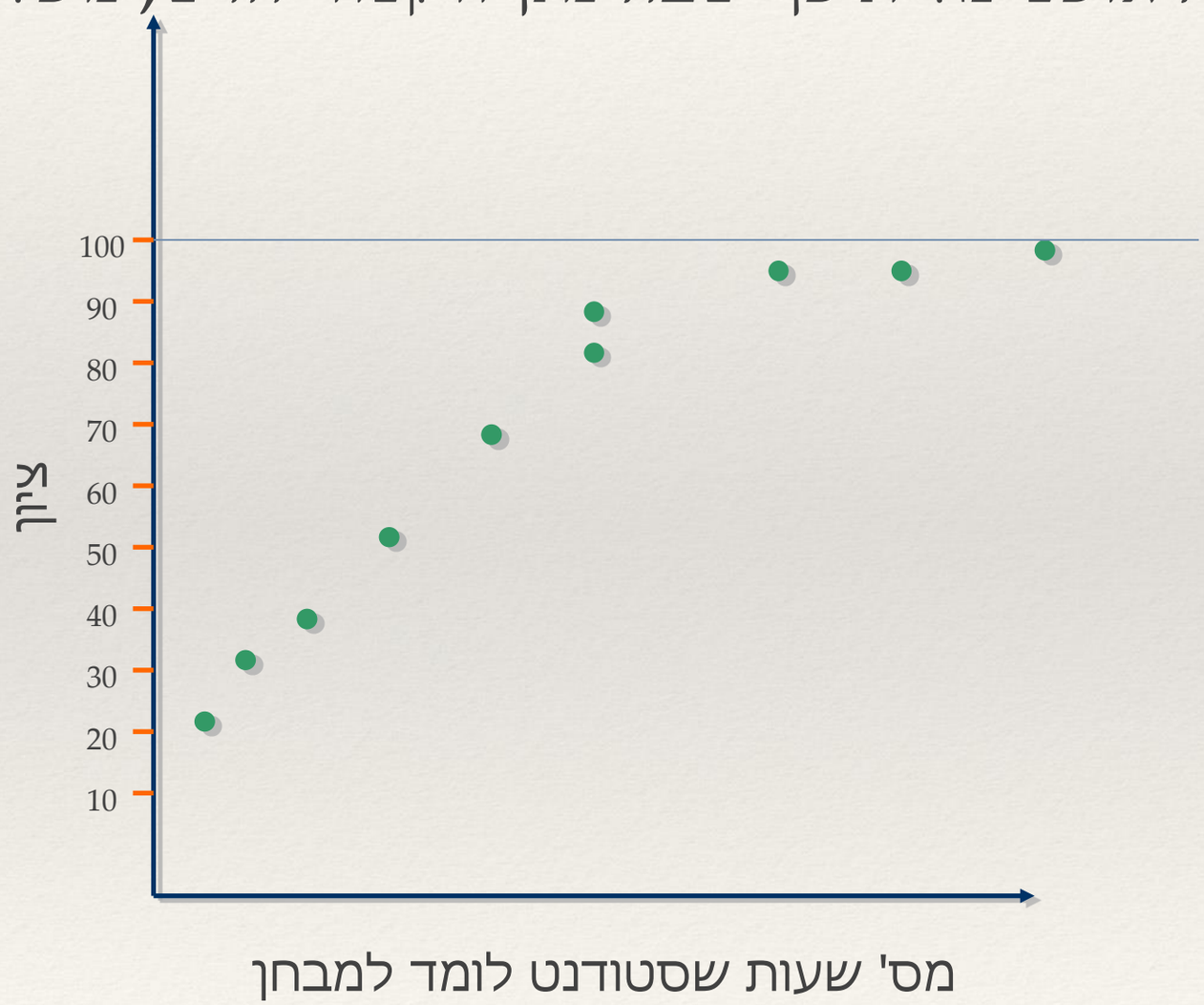
גם למידה כזו נקראת למידה מונחית (supervised learning)
כי עבור כל דוגמא אנחנו יודעים כמה כל משפחה הוציאה בחו"ל
והמחשב ישתמש בדוגמאות כדי ללמוד

קרדיט: ד"ר אוהד סיוון

דוגמא לבעיית רגרסיה כלמידה מונחית – חיזוי ציון של סטודנט

נתונים לנו ווקטורים במרחב עם ערך המוצמד לכל אחד מהם. (תצפיות)

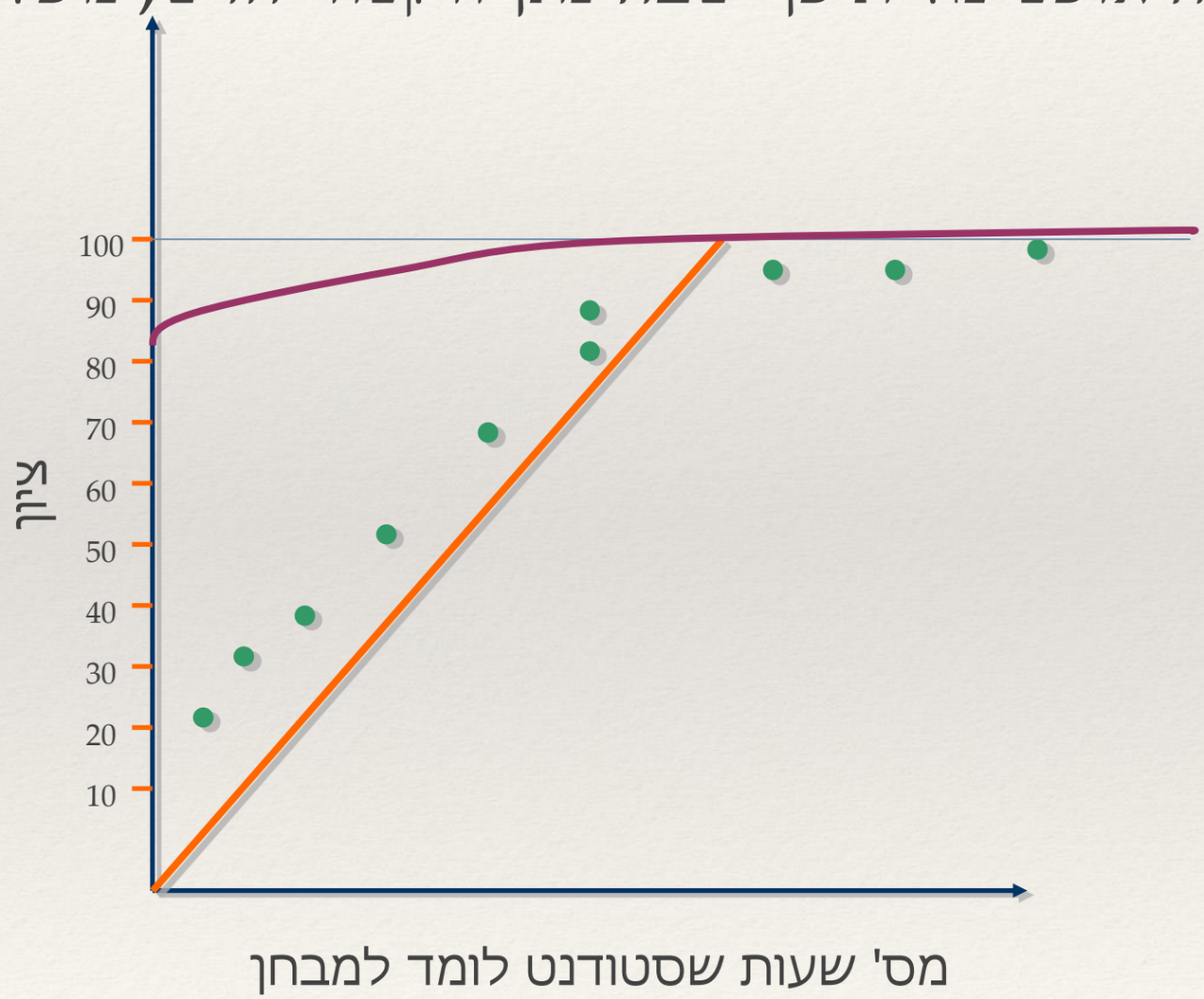
מצא משוואה אופטימלית כך שבהינתן ווקטור חדש, נוכל לשערך את ערכו.



דוגמא לבעיית רגרסיה כלמידה מונחית – חיזוי ציון של סטודנט

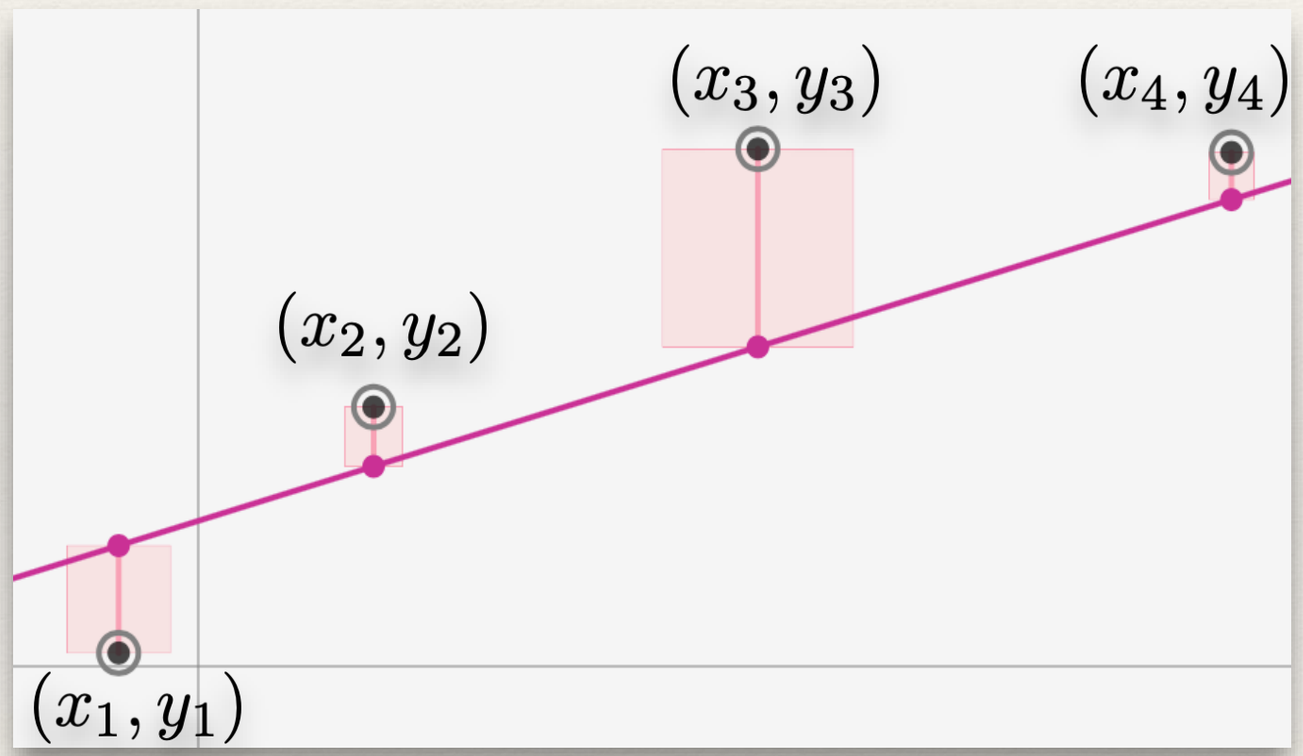
נתונים לנו ווקטורים במרחב עם ערך המוצמד לכל אחד מהם. (תצפיות)

מצא משוואה אופטימלית כך שבהינתן ווקטור חדש, נוכל לשערך את ערכו.



רגרסיה לינארית

ברגרסיה לינארית נחזה את הקשר בין המאפיינים לבין הערך אותו נרצה לחזות, כקשר לינארי.



שיערוך מודל רגרסיה
(regression model evaluation)

שיערוך מודל רגרסיה

SAE: $SAE = \sum_{I=1}^{n} |y_i - \hat{y}_i|$

MAE: $MAE = \frac{1}{n}\sum_{I=1}^{n} |y_i - \hat{y}_i|$

SSE: $SSE = \sum_{I=1}^{n} (y_i - \hat{y}_i)^2$

MSE: $MSE = \frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2$

RMSE: $RMSE = \sqrt{\frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2}$

$\bar{y}$: ממוצע הערכים המונחים, כלומר $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

SST (Sum of Squared Total): $SST = \sum_{I=1}^{n} (y_i - \bar{y})^2$

R-SQARE: $\boldsymbol{R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{MSE}{\sigma^2}}$

שאלה 1 (סקר)

SAE: $SAE = \sum_{I=1}^{n} |y_i - \hat{y}_i|$ **MAE:** $MAE = \frac{1}{n}\sum_{I=1}^{n} |y_i - \hat{y}_i|$

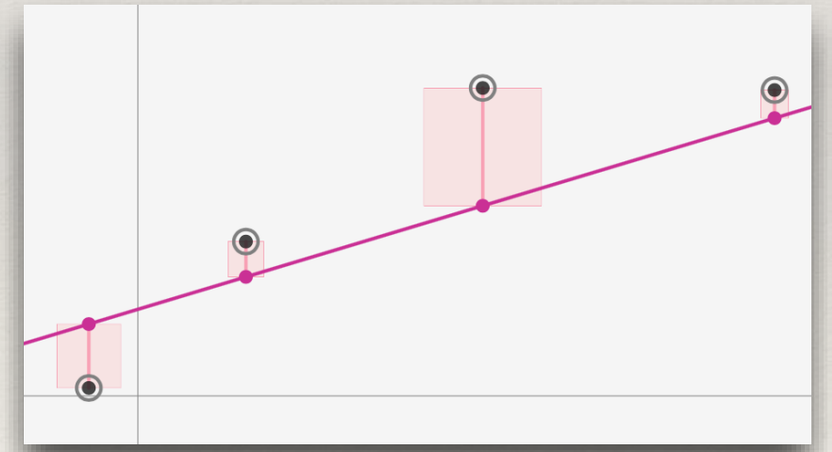
SSE: $SSE = \sum_{I=1}^{n} (y_i - \hat{y}_i)^2$ **MSE:** $MSE = \frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2$ **RMSE:** $RMSE = \sqrt{\frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2}$

$\bar{y}$: ממוצע הערכים המונחים, כלומר $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ **SST (Sum of Squared Total):** $SST = \sum_{I=1}^{n} (y_i - \bar{y})^2$ =

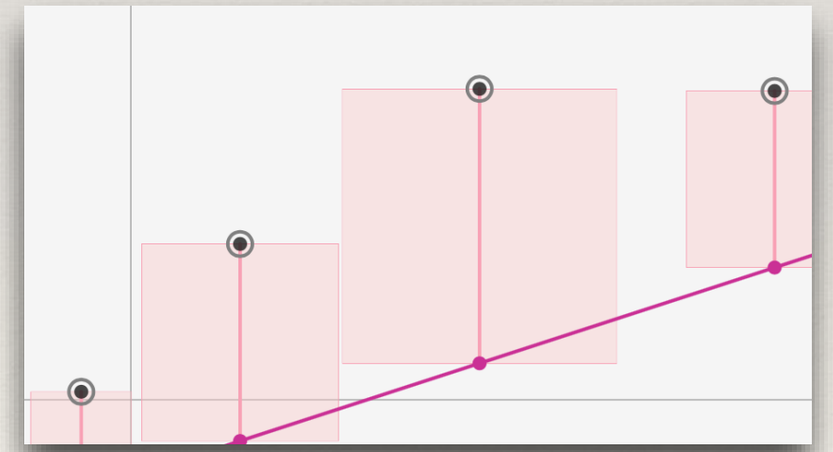
R-SQARE: $\boldsymbol{R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{MSE}{\sigma^2}}$

לאיזו מהפונקציות הלינאריות הבאות הטעות המינימלית?

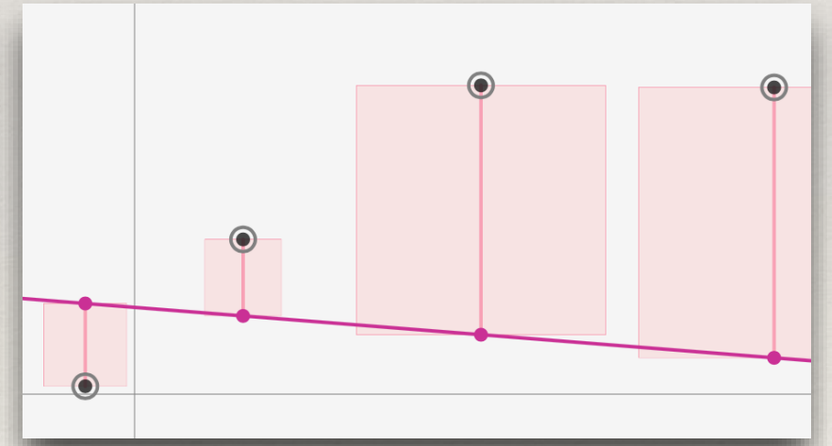
A.



B.



C.



שאלה 1 (סקר)

SAE: $SAE = \sum_{I=1}^{n} |y_i - \hat{y}_i|$ **MAE:** $MAE = \frac{1}{n}\sum_{I=1}^{n} |y_i - \hat{y}_i|$

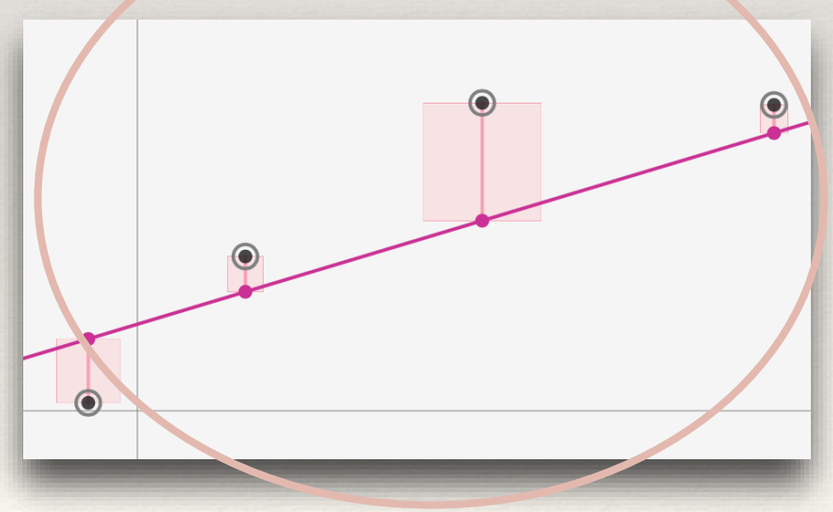
SSE: $SSE = \sum_{I=1}^{n} (y_i - \hat{y}_i)^2$ **MSE:** $MSE = \frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2$ **RMSE:** $RMSE = \sqrt{\frac{1}{n}\sum_{I=1}^{n} (y_i - \hat{y}_i)^2}$

$\bar{y}$: ממוצע הערכים המונחים, כלומר $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ **SST (Sum of Squared Total):** $SST = \sum_{I=1}^{n} (y_i - \bar{y})^2$ =

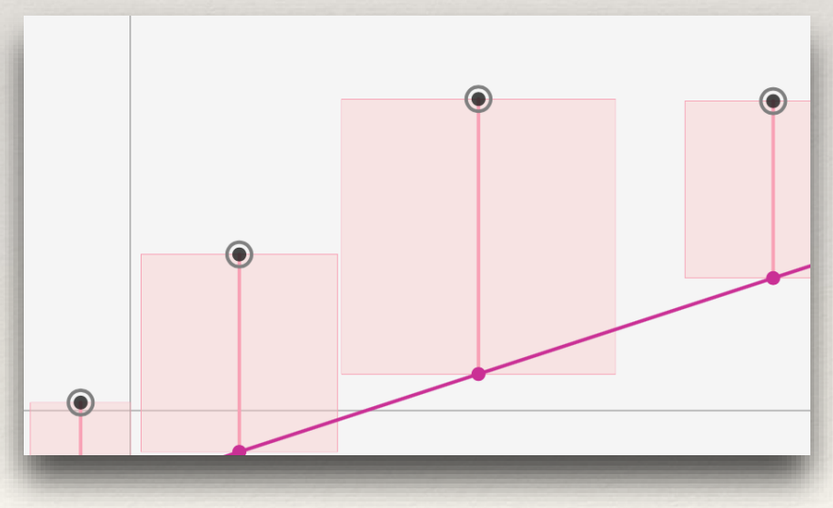
R-SQARE: $\boldsymbol{R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{MSE}{\sigma^2}}$

לאיזו מהפונקציות הלינאריות הבאות הטעות המינימלית?

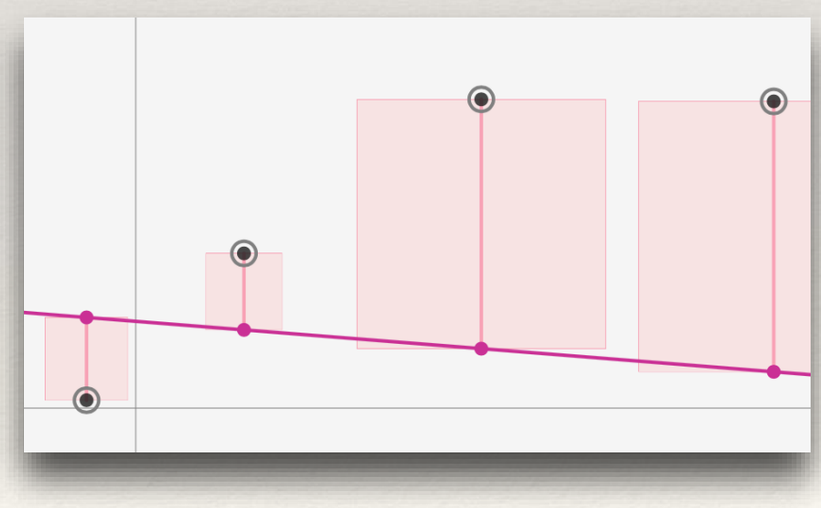
A.



B.



C.



SSE Illustration – the 1st model (with y_pred1 predictions) has better (lower) SSE

	y	y_pred1	y_pred1-y	(y_pred1-y)^2	y_pred2	y_pred2-y	(y_pred2-y)^2
SSE				1000			31000
0	120	130	10	100	50	-70	4900
1	140	120	-20	400	200	60	3600
2	150	160	10	100	150	0	0
3	200	220	20	400	350	150	22500

שאלה 2א – שיערוך מודל רגרסיה

Height (X)	Weight (Y)	Predicted (Ŷ)	Error (Y-Ŷ)	Absolute-Error |(Y-Ŷ)|
43	41	43.6	-2.6	2.6
44	45	44.4	0.6	0.6
45	49	45.2	3.8	3.8
46	47	46	1	1
47	44	46.8	-2.8	2.8
Regression line = y=9.2+0.8x				

תרגיל –

נתונים 2 משתנים – גובה ומשקל. רוצים לשערך את המשקל כתלות בגובה, ומצאו קו רגרסיה לינארי:

$\hat{y}=9.2+0.8x_1$

חשבו את ה- SAE וה- MAE:

$$SAE = \sum_{I=1}^{n}|y_i - \hat{y}_i| \quad :\textbf{SAE}$$

$$MAE = \frac{1}{n}\sum_{I=1}^{n}|y_i - \hat{y}_i| \quad :\textbf{MAE}$$

פתרון

- קודם נשערך את Y
- כעת נחשב את הטעות המוחלטת
- נסכום ונקבל SAE = 10.8
- נחלק בכמות הדו' ונקבל

$$\text{MAE} = \frac{1}{5} \cdot 10.8 = 2.16$$

שאלה 2ב – שיערוך מודל רגרסיה

Height (X)	Weight (Y)	Predicted ($\hat{Y}$)	Error ($Y-\hat{Y}$)	squared error $(Y-\hat{Y})^2$
43	41	43.6	-2.6	6.76
44	45	44.4	0.6	0.36
45	49	45.2	3.8	14.44
46	47	46	1	1
47	44	46.8	-2.8	7.84
Regression line = y=9.2+0.8x				

SSE: $SSE = \sum_{I=1}^{n}(y_i - \hat{y}_i)^2$

MSE: $MSE = \frac{1}{n}\sum_{I=1}^{n}(y_i - \hat{y}_i)^2$

RMSE: $RMSE = \sqrt{\frac{1}{n}\sum_{I=1}^{n}(y_i - \hat{y}_i)^2}$

תרגיל –

נתונים 2 משתנים – גובה ומשקל. רוצים לשערך את המשקל כתלות בגובה, ומצאו קו רגרסיה (הנ"ל)

חשבו את ה- SSE, ה-MSE וה-RMSE:

פתרון:

- נשתמש בשערוך הקודם של Y ($\hat{Y}$) ובחישוב הטעות ($Y-\hat{Y}$)
- כעת נחשב את הטעות הריבועית
- נסכום ונקבל SSE=30.4
- נחלק בכמות הדו' ונקבל

MSE=$\frac{1}{5} \cdot 30.4 = 6.08$

RMSE ≈ 2.46

שאלה 2ג – שיערוך מודל רגרסיה

Height (X)	Weight (Y)	Y−Ȳ	squared dist from avg (Y-Ȳ)²
43	41	-4.2	17.64
44	45	-0.2	0.04
45	49	3.8	14.44
46	47	1.8	3.24
47	44	-1.2	1.44
Regression line = y=9.2+0.8x			

תרגיל –
נתונים 2 משתנים – גובה ומשקל.
רוצים לשערך את המשקל כתלות בגובה, ומצאו קו רגרסיה (הנ"ל)

חשבו את ה- R-Squared:

פתרון:

- נשתמש בחישוב הקודם של SSE:
  קיבלנו SSE=30.4
- נחשב את הממוצע $\bar{y}$
  $\bar{y}$=45.2
- ונחשב את המרחקים הריבועיים מהממוצע $(Y-\bar{Y})^2$
- נחשב את SST (סכום המרחקים הריבועיים הנ"ל): SST=36.8
- נחשב את R-Squared

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{30.4}{36.8} \approx 0.174$$

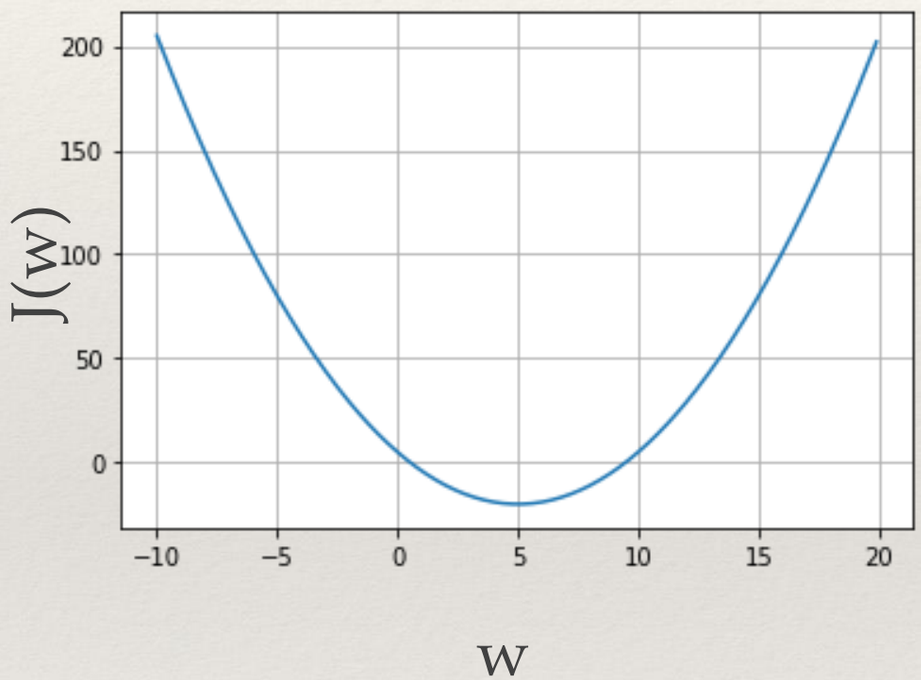
SSE: $SSE = \sum_{I=1}^{n}(y_i - \hat{y}_i)^2$

$\bar{y}$: ממוצע הערכים המונחים, כלומר $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

SST (Sum of Squared Total): $SST = \sum_{I=1}^{n}(y_i - \bar{y})^2$

R-Squared: $R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{MSE}{\sigma^2}$

מציאת מינימום - השוואת הנגזרת לאפס



- ניתן למצוא מינימום על ידי השוואת הנגזרת לאפס
- לא תמיד ניתן לפתור את המשוואה $\frac{df(x)}{dx} = 0$
- במקרים אלו ניתן להתקרב אל נקודת המינימום באופן איטרטיבי

הכללה - התקדמות בכיוון הנגזרת

- בוחרים x התחלתי.
אנחנו רוצים לקדם את x לכיוון נקודת המינימום.
- נמשיך להתקדם לכיוון בו הפונקציה יורדת
- לכן, כדאי לעדכן את x (להגדיל או להקטין) נגד כיוון הנגזרת
- כלומר לכיוון $-\frac{df(x)}{dx}$

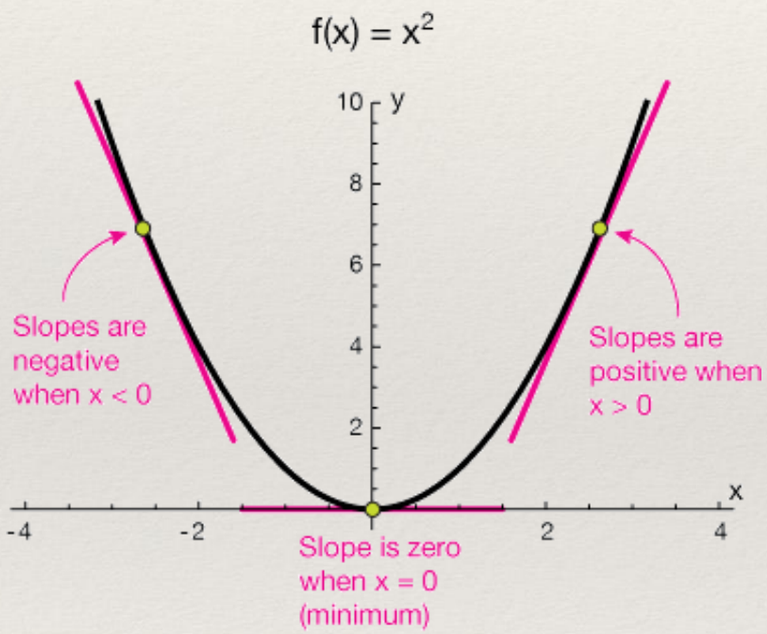


Image source: https://xaktly.com/TheDerivative.html

אלגוריתם gradient descent

- על מנת למצוא את נקודת המינימום של פונקציה f(x):
  - נגריל נקודת התחלה x
  - נקדם את ערך x לפי הנוסחה:

$$x \leftarrow x - \alpha \frac{\partial f(x)}{\partial x}$$

- α – קצב הלמידה (נקבע על ידנו. לדוגמא 0.1 או 0.01)
- $\frac{\partial f(x)}{\partial x}$ - נגזרת הפונקציה

תרגיל חישוב Gradient descent

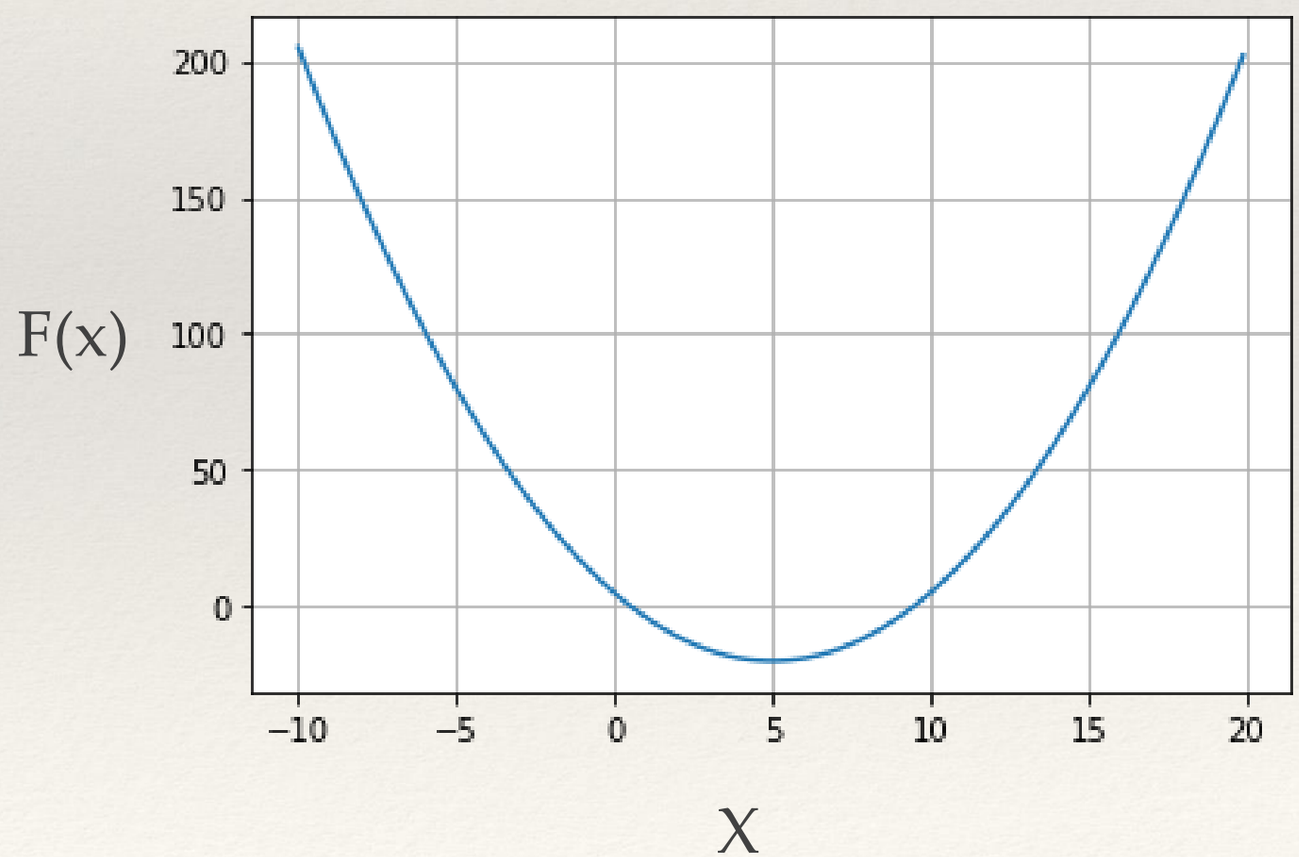
- נתונה הפונקציה $f(x)=x^2-10x+5$
- עקבו אחר פעולתו של אלגוריתם gradient descent עבור $\alpha=0.1$

Step	x	$\frac{\partial f(x)}{\partial x}$	$\alpha\frac{\partial f(x)}{\partial x}$
0	0		
1			
2			
3			
4			

תרגיל חישוב Gradient descent - פתרון

epoch	x	dx	α·dx
0	0.00	-10.00	-1.00
1	1.00	-8.00	-0.80
2	1.80	-6.40	-0.64
3	2.44	-5.12	-0.51
4	2.95	-4.10	-0.41
5	3.36	-3.28	-0.33
6	3.69	-2.62	-0.26
7	3.95	-2.10	-0.21
8	4.16	-1.68	-0.17
9	4.33	-1.34	-0.13
10	4.46	-1.07	-0.11
11	4.57	-0.86	-0.09
12	4.66	-0.69	-0.07
13	4.73	-0.55	-0.05
14	4.78	-0.44	-0.04
15	4.82	-0.35	-0.04
16	4.86	-0.28	-0.03
17	4.89	-0.23	-0.02
18	4.91	-0.18	-0.02
19	4.93	-0.14	-0.01
20	4.94	-0.12	-0.01

- נתונה הפונקציה $f(x)=x^2-10x+5$
- עקבו אחר פעולתו של אלגוריתם gradient descent עבור $\alpha=0.1$



רגרסיה לינארית (linear regression)

רגרסיה לינארית

ברגרסיה לינארית – הקשר בין וקטור המאפיינים, לערך אותו רוצים לחזות הוא פונקציה לינארית

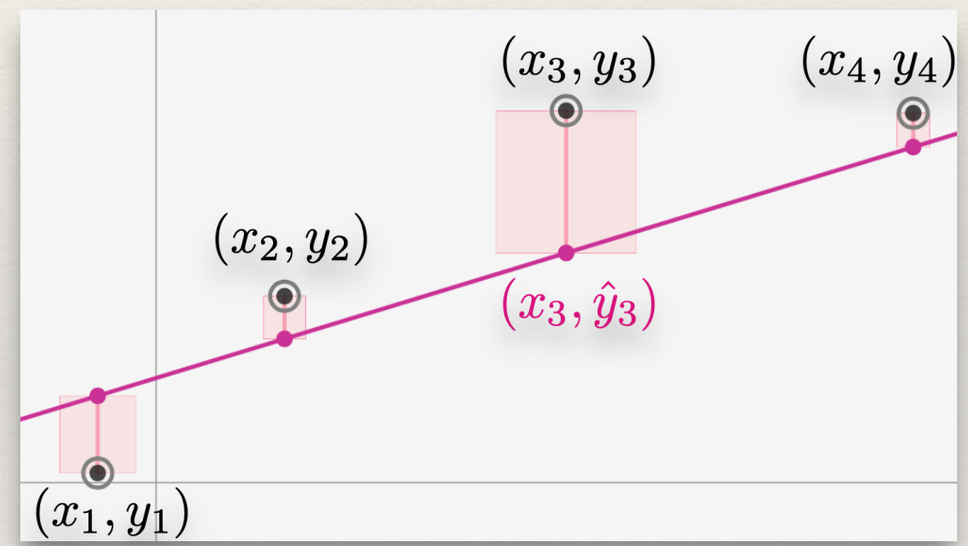
מקרה פשוט (כמו בדוגמה) : יש רק מאפיין אחד בווקטור המאפיינים

שכר יומי בש"ח	סכום הציונים
460	2104
232	1416
315	1534
178	852
...	...

פונקצית מחיר (Cost Function)

המודל הלינארי:

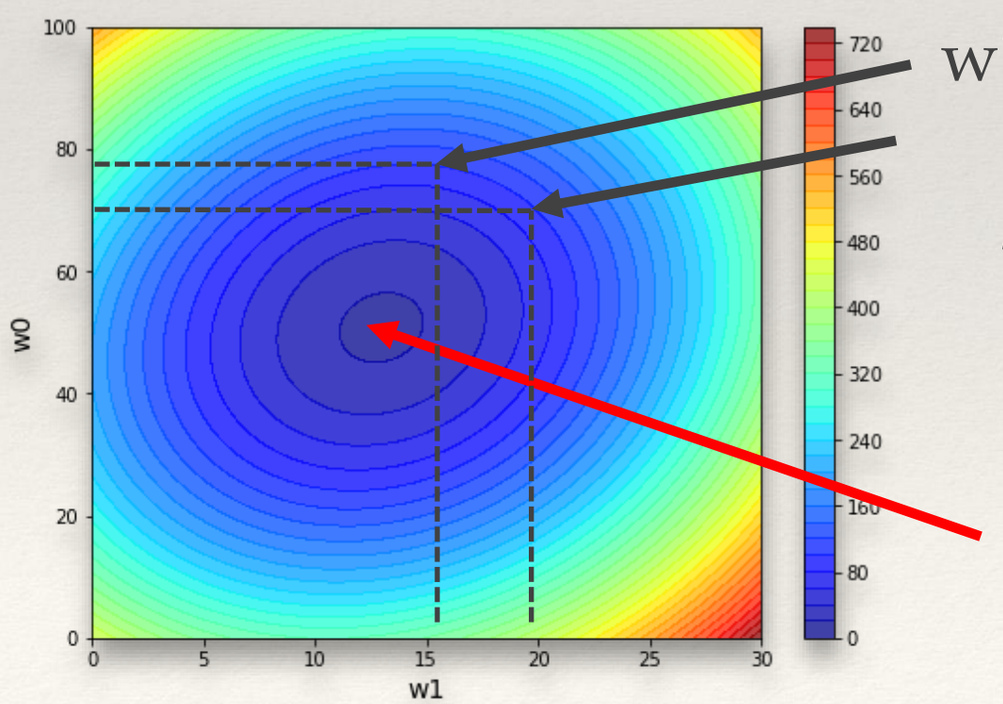
$$\hat{y} = f\left(x; \vec{w}\right) = w_0 + w_1 x$$



פונקצית המחיר – מוגדרת ע"י ממוצע הטעות הריבועית

$$J\left(\vec{w}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_i - y_i\right)^2$$

הטעות: $MSE = \frac{1}{n}\sum_{I=1}^{n}(\hat{y}_i - y_i)^2$

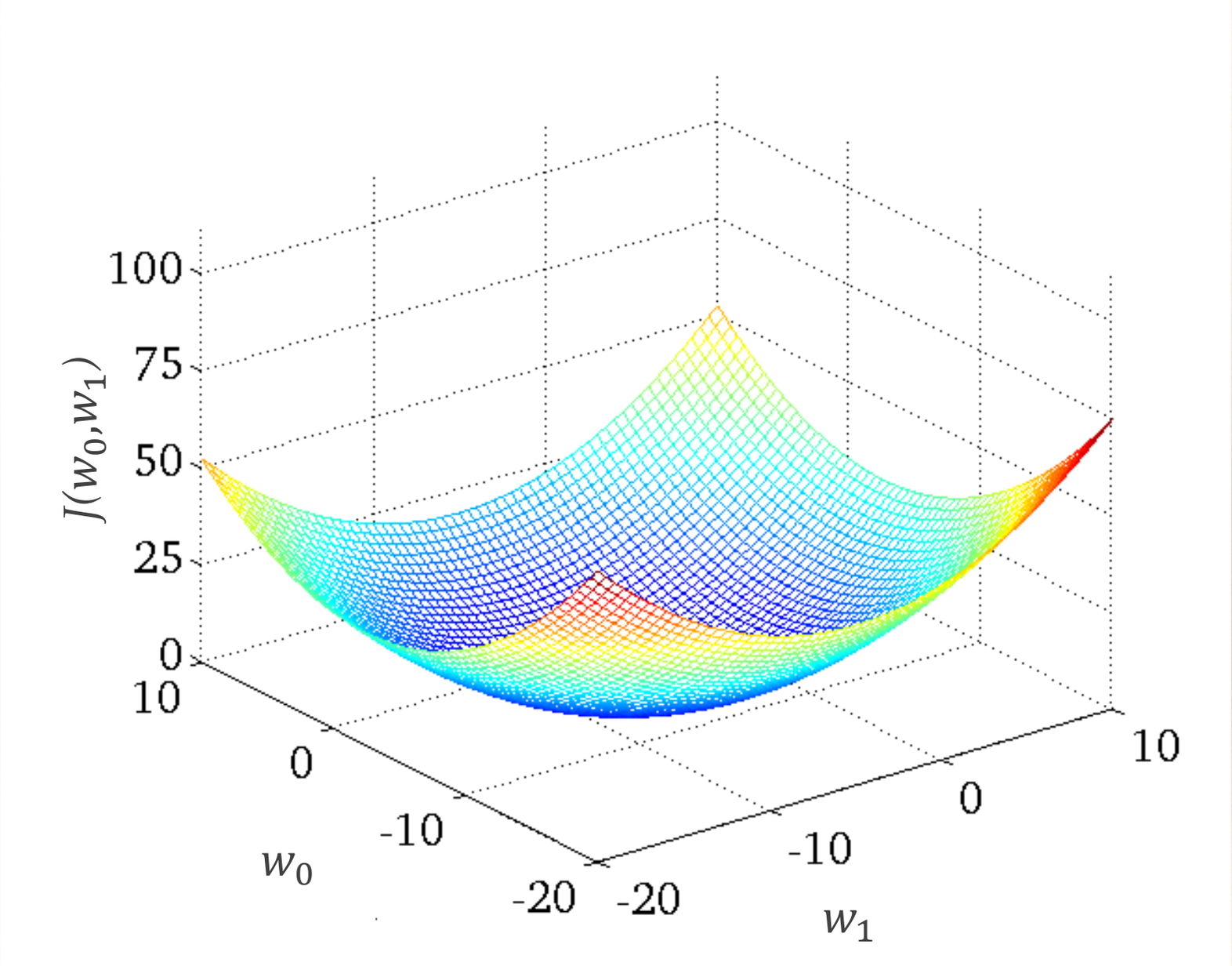


ערכי w0, w1 שונים, בעלי אותה הטעות

מטרה: למצוא היפותזה עם טעות מינימלית ע"י שימוש בפו' J

$$\min_{w}\left[J\left(\vec{w}\right)\right]$$

פונקצית מחיר עבור 2 פרמטרים w_0,w_1



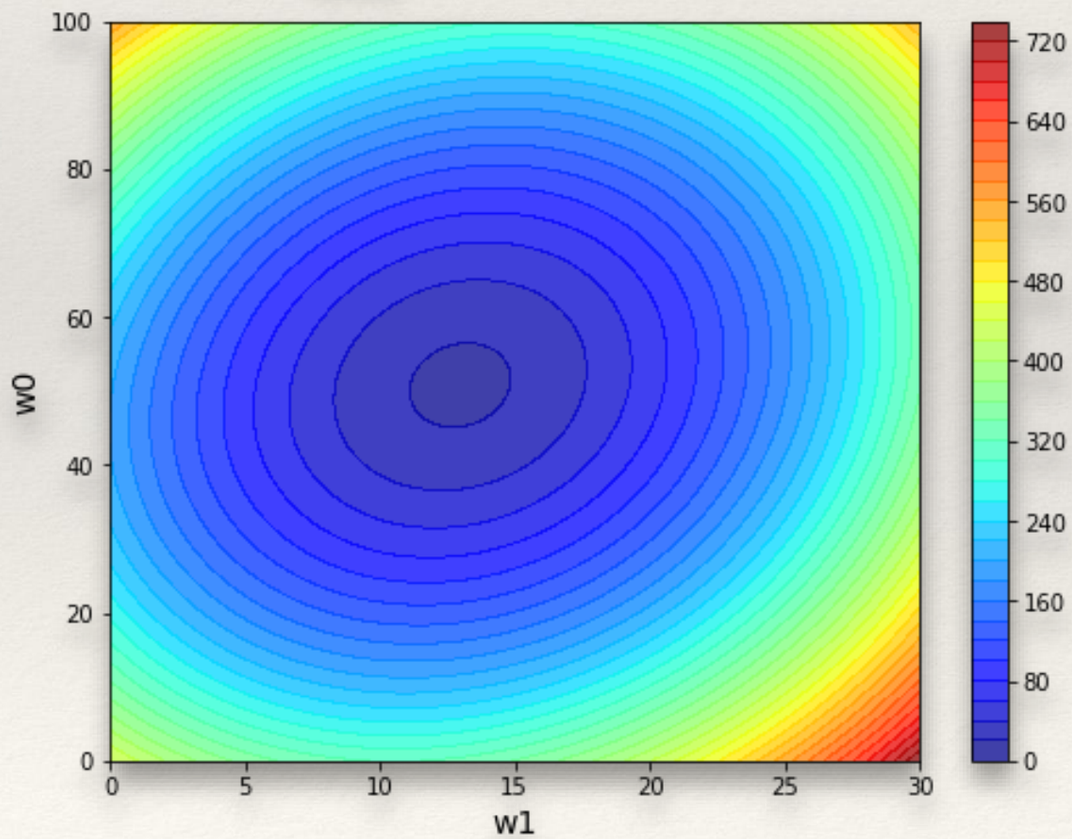
פונקצית מחיר (Cost Function)

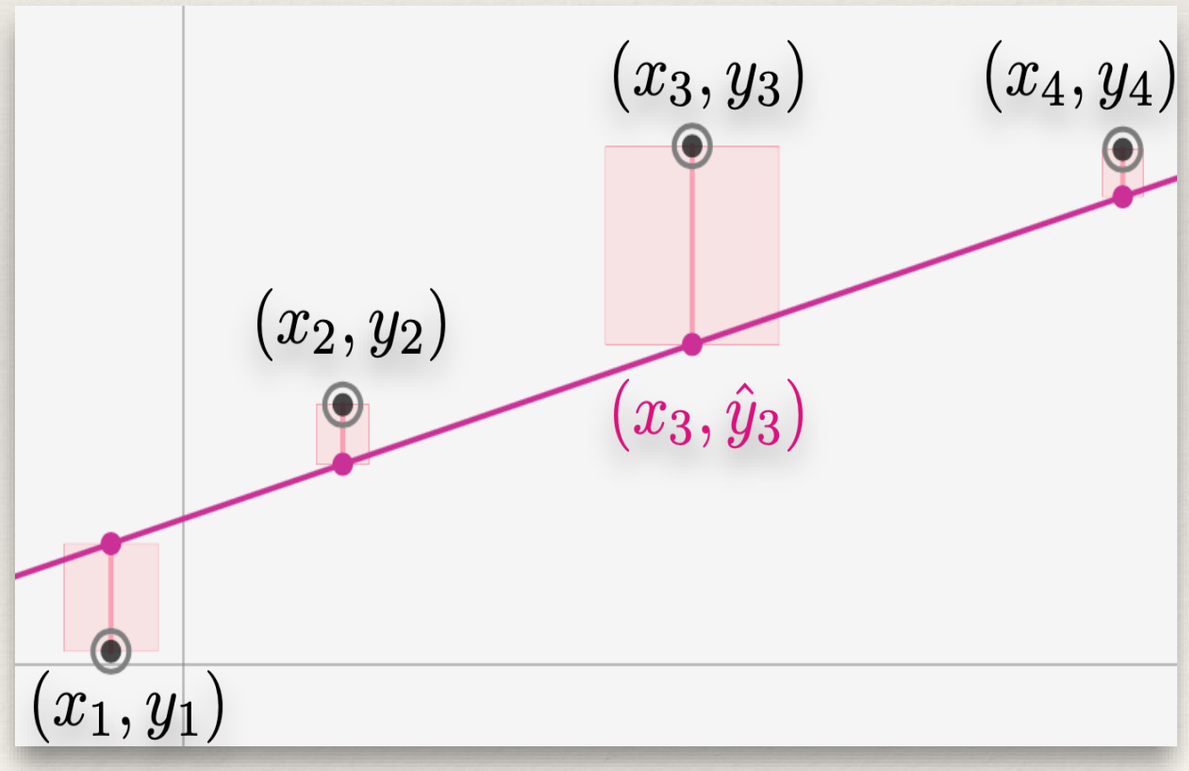
המודל הלינארי:

$$\hat{y} = f\left(x; \vec{w}\right) = w_0 + w_1 x$$

פונקצית המחיר – מוגדרת ע"י ממוצע הטעות הריבועית

$$J\left(\vec{w}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_i - y_i\right)^2$$





Linear Regression via Gradient Descent

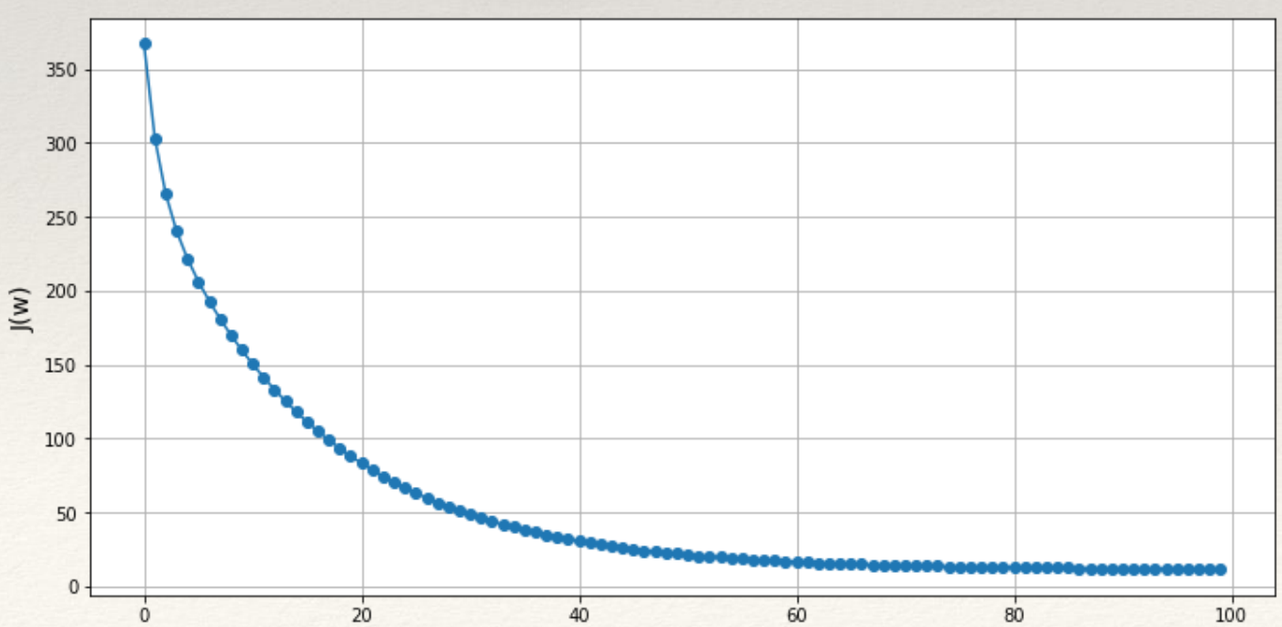
פונקצית המחיר:

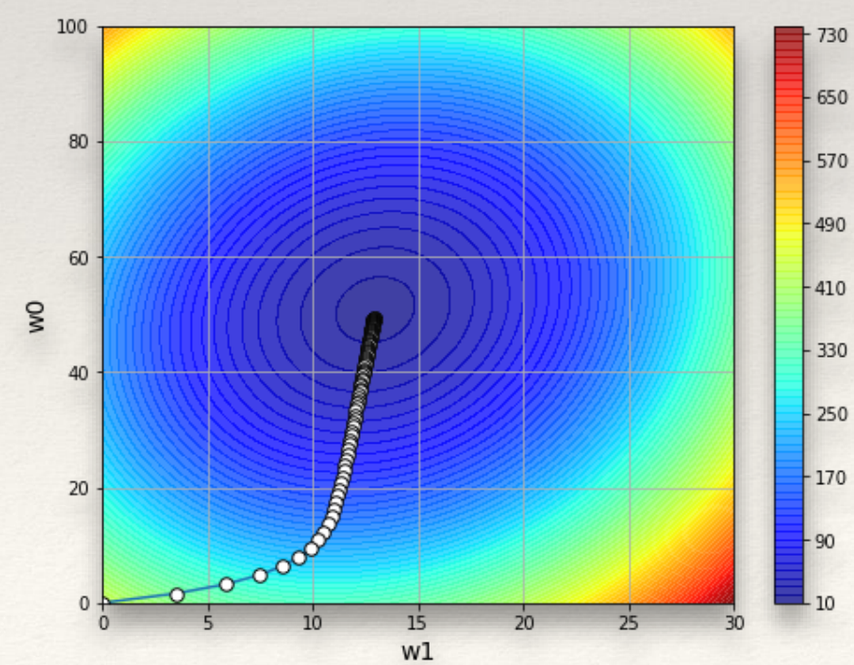
$$J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_i - y_i\right)^2$$

$$\frac{\partial J}{\partial w_0} = \frac{2}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_i - y_i\right)\cdot 1$$

$$\frac{\partial J}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_i - y_i\right)\cdot x_i$$

הנגזרות החלקיות



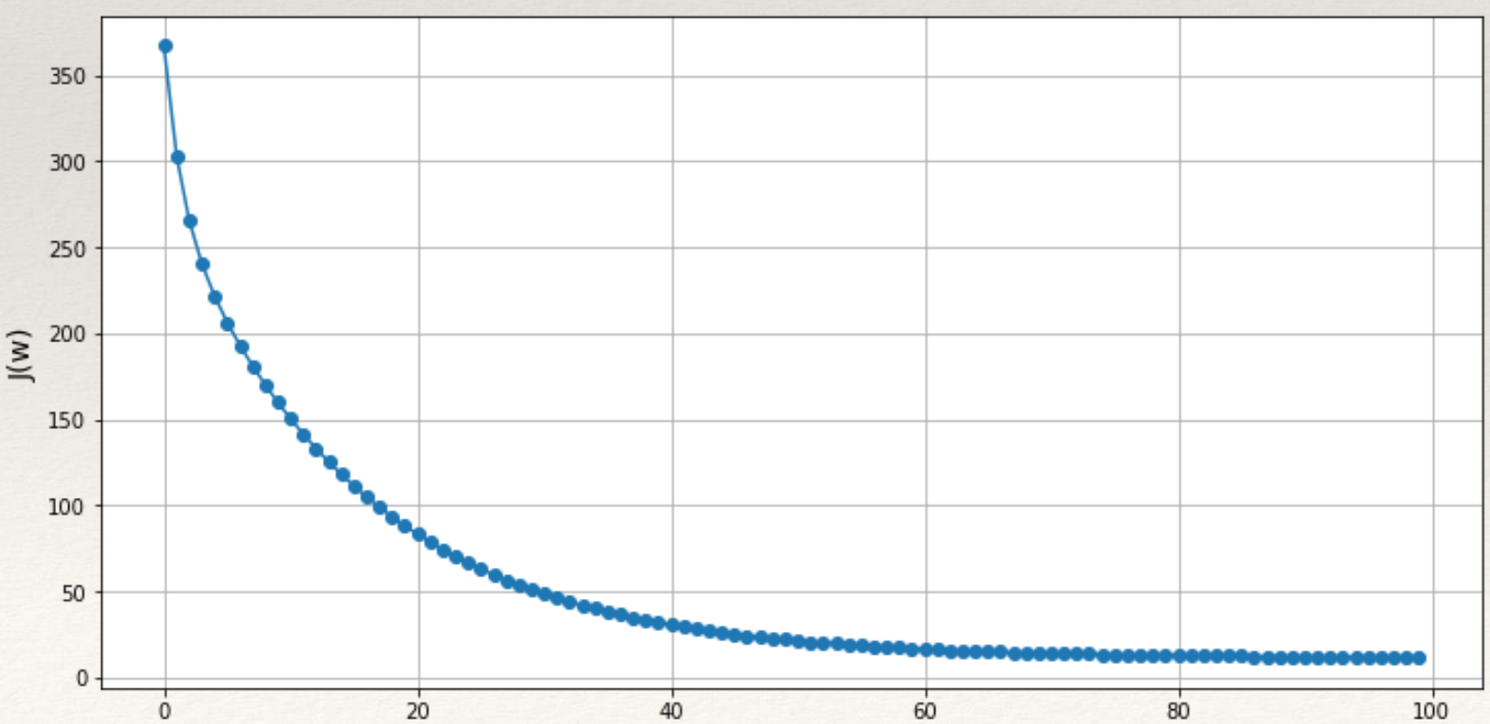


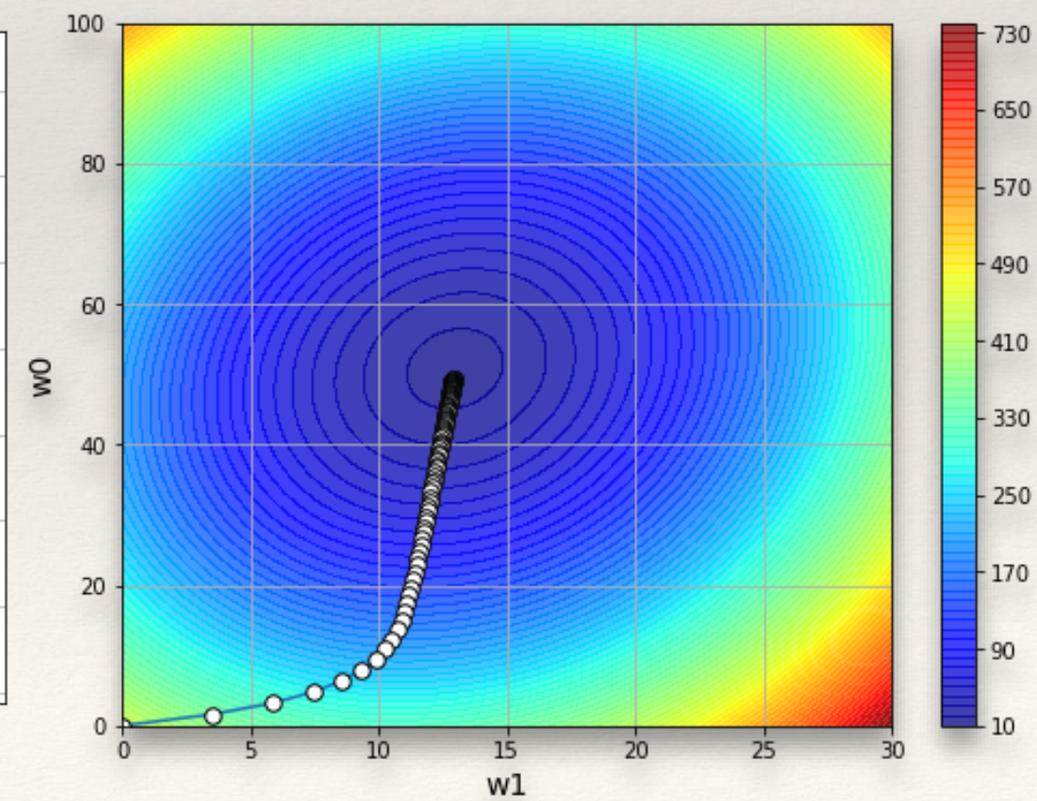
Linear Regression via Gradient Descent

$$w_0 := w_0 - \frac{2\alpha}{n}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)$$

$$w_1 := w_1 - \frac{2\alpha}{n}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)\cdot x_i$$

עדכון הפרמטרים





Linear Regression with Gradient Descent

Gradient-Descent(*S: training-examples & targets, α*)

Initialize w_0 & w_1 with small random numbers

Until <u>TERMINATION</u> Do

- Initialize each Δw_0 & Δw_1 to zero
- For each $<x_i, y_i>$ *in S* Do
  - Compute $\hat{y}_i = w_{0+} w_1 x_{i1}$
  - Update Δw_0 & Δw_1 values for example i as following:
    - $\Delta w_0 = \Delta w_0 - \alpha \frac{2}{n}(\hat{y}_i - y_i)$
    - $\Delta w_1 = \Delta w_1 - \alpha \frac{2}{n}(\hat{y}_i - y_i)\, x_{i1}$
- $w_0 = w_0 + \Delta w_0$
- $w_1 = w_1 + \Delta w_1$

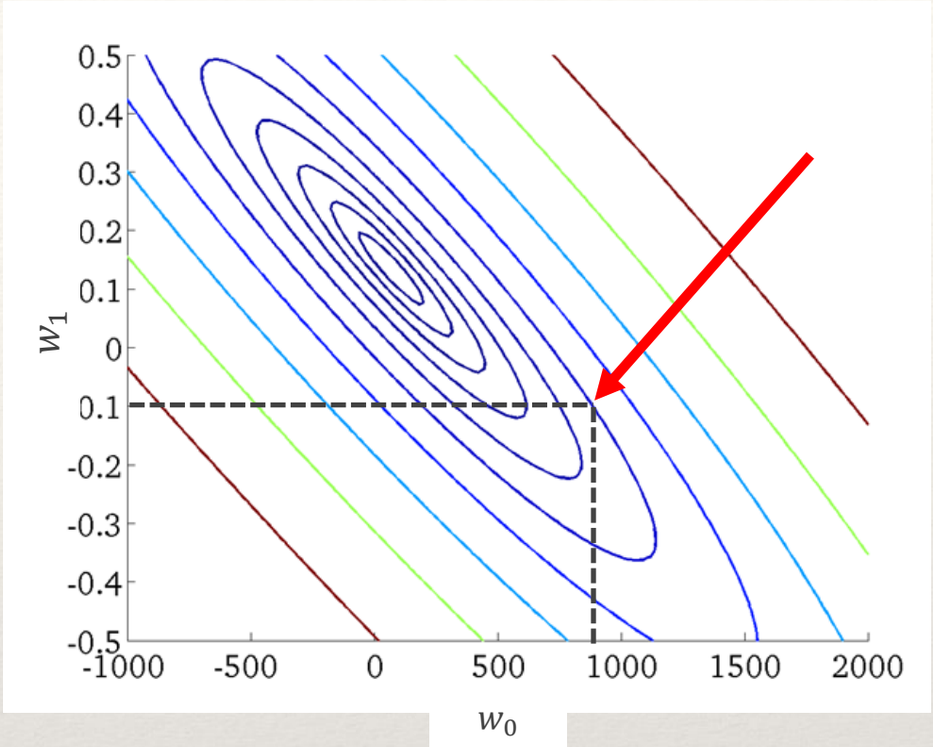
$\hat{y}$ –predicted value
y – actual target value
α - learning rate

שאלה 3

$J(w_0,w_1)$

- פונקצית המחיר (מדמה גרף תלת מימדי)

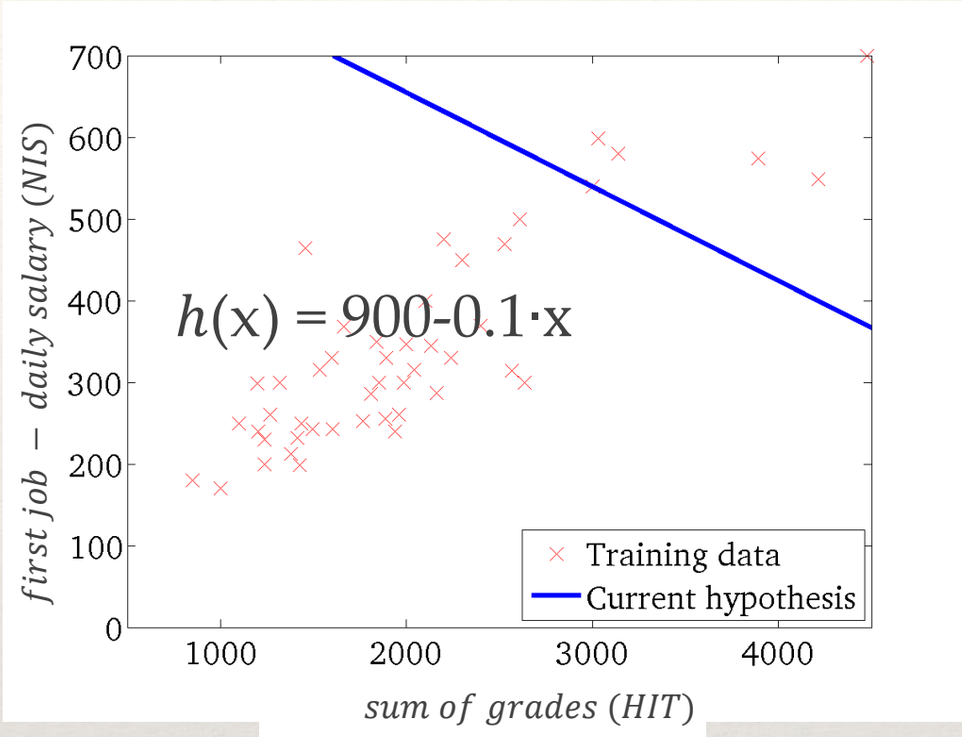
- פונקצית המחיר - פונקציה של w_0,w_1



$H_W(x)$

- ישר הרגרסיה, מאפיין אחד

- פונקציה של x, עבור w_0,w_1 מסוימים



תרגיל – בתחילת האיטרציה (הסבב) הרביעית בtraining, קיבלנוה את המשקולות המתאימות למשוואה הבאה: $h(x) =$ $900-0.1\cdot x$, נתונים $\alpha= 0.0000001$, ועבור ה- train-set

<u>חשבו</u> את w_0,w_1, לאיטרציה הבאה בעזרת Gradient Descent עבטר הרגרסיה הלינארית

- חשבו סבב נוסף

שכר יומי בש"ח	סכום הציונים
460	2104
315	1534
178	852

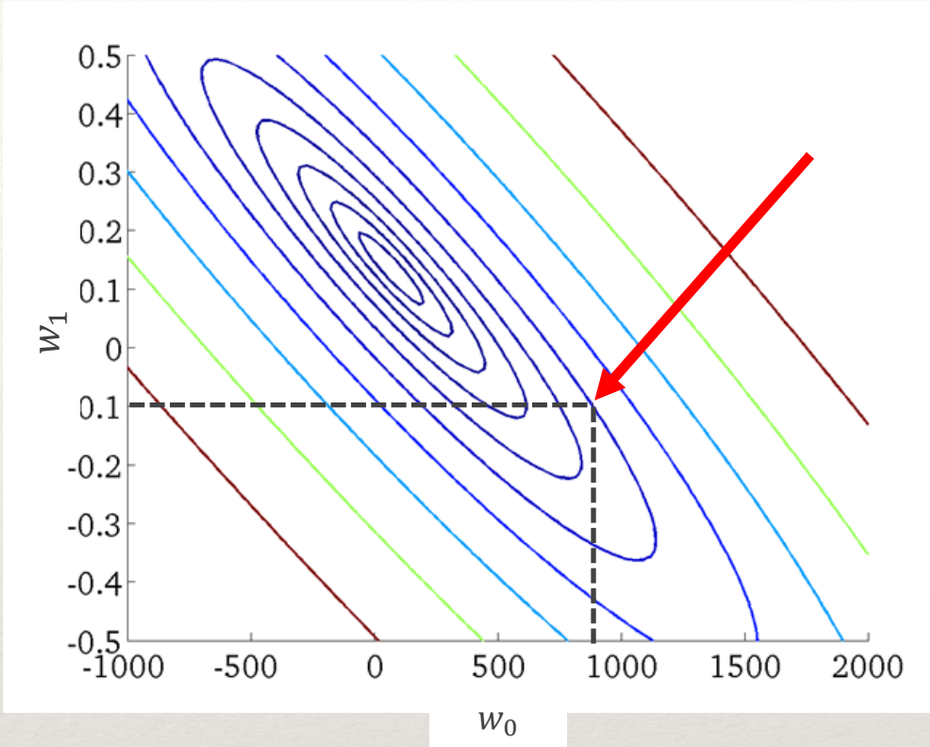
Train set

שאלה 3

$J(w_0,w_1)$

- פונקצית המחיר (מדמה גרף תלת מימדי)

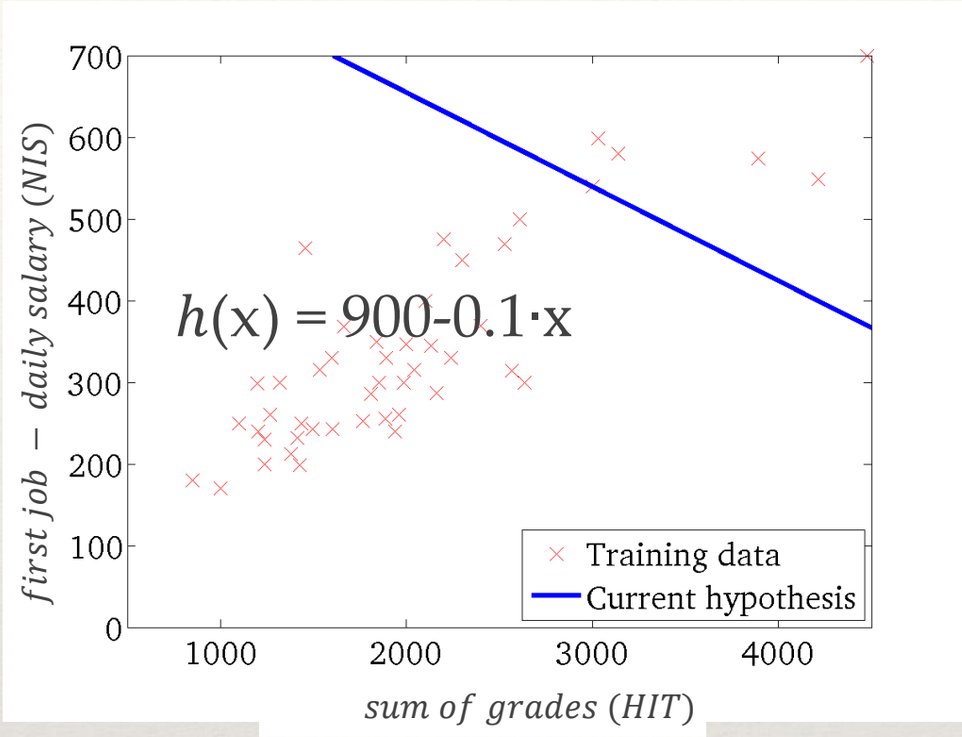
- פונקצית המחיר - פונקציה של w_0,w_1



$H_W(x)$

- ישר הרגרסיה, מאפיין אחד

- פונקציה של x, עבור w_0,w_1 מסוימים



שכר יומי בש"ח	סכום הציונים
460	2104
315	1534
178	852

Train set

פתרון:

חישוב w_0 חדש

$$\boldsymbol{w_0} = w_0 - \alpha \cdot \frac{\partial J}{\partial w_0}$$

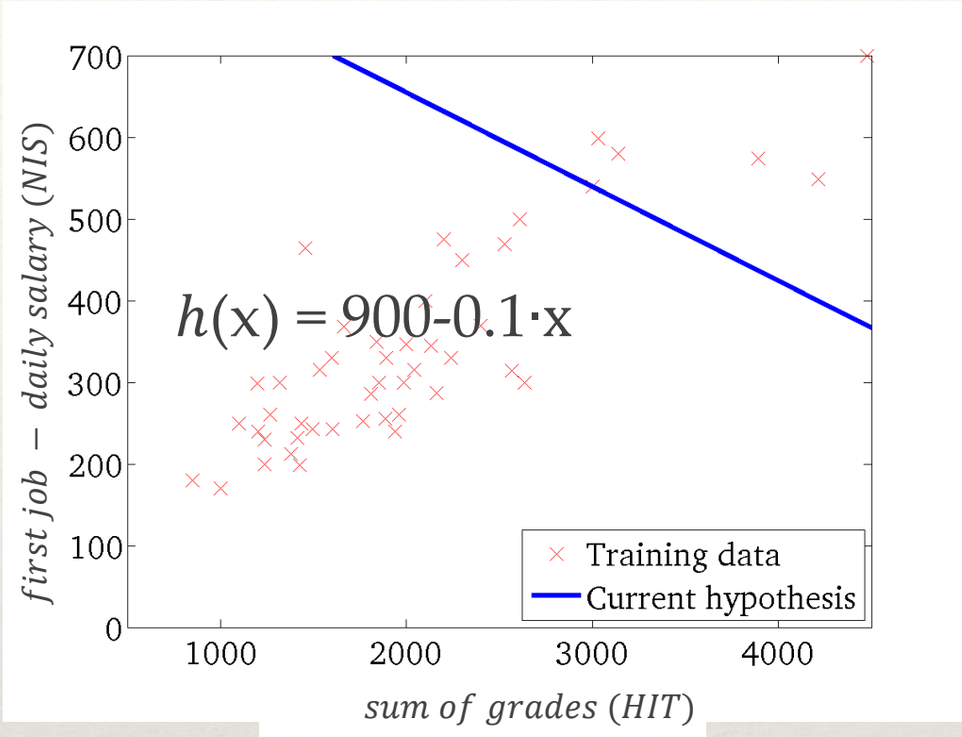
$$= 900 - 0.0000001 \cdot \frac{2}{3}$$
$$\cdot [(900 - 0.1 \cdot 2104 - 460) \cdot 1$$
$$+ (900 - 0.1 \cdot 1534 - 315) \cdot 1$$
$$+ (900 - 0.1 \cdot 852 - 178) \cdot 1]]$$
$$= 900 - 0.0000001 \cdot \frac{2}{3} \cdot 1{,}298 \approx \mathbf{899.99}$$

שאלה 3

$H_W(x)$

- ישר הרגרסיה, מאפיין אחד

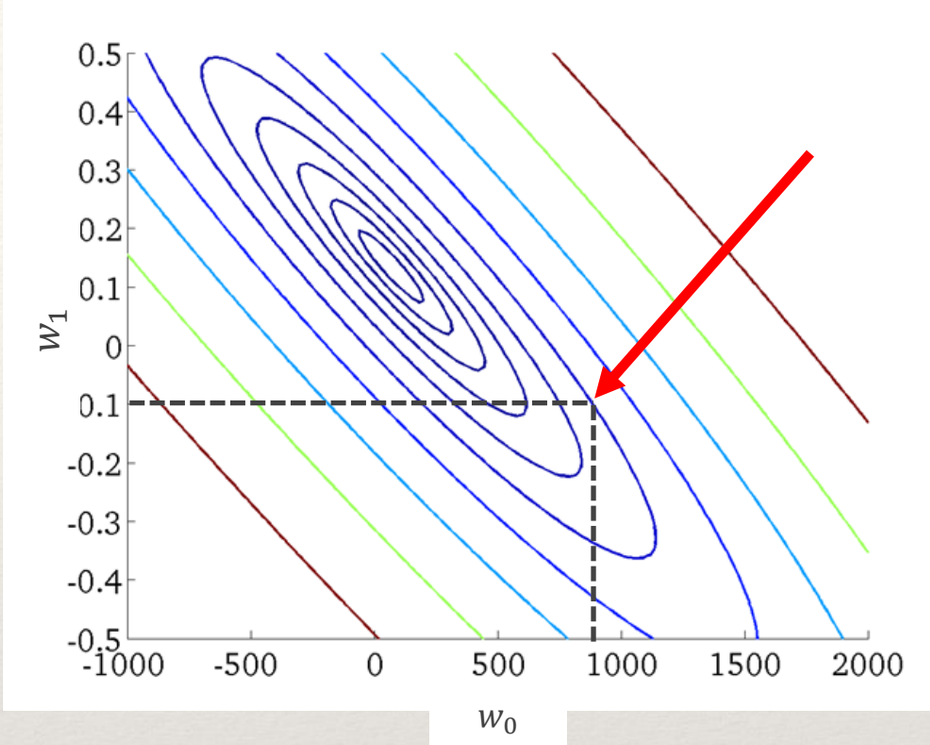
- פונקציה של x, עבור w_0,w_1 מסוימים



$J(w_0,w_1)$

- פונקצית המחיר (מדמה גרף תלת מימדי)

- פונקצית המחיר - פונקציה של w_0,w_1



שכר יומי בש"ח	סכום הציונים
460	2104
315	1534
178	852

Train set

המשך פתרון:

חישוב w_1 חדש

$$\boldsymbol{w_1} = w_1 - \alpha \cdot \frac{\partial J}{\partial w_1}$$

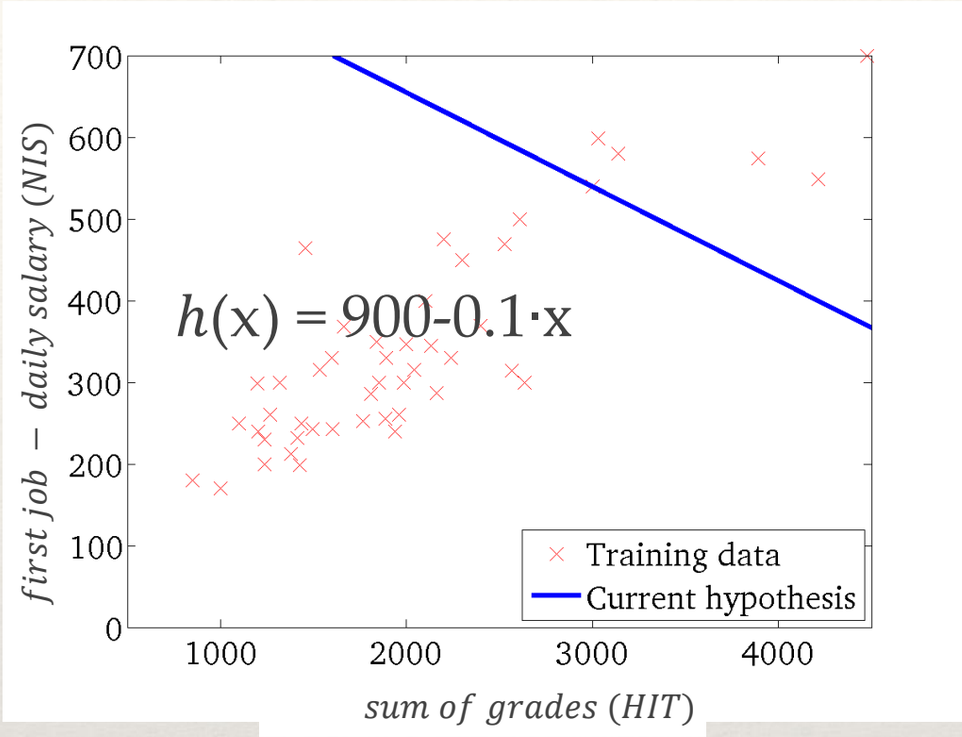
$$= -0.1 - 0.0000001 \cdot \frac{2}{3}$$
$$\cdot [(900 - 0.1 \cdot 2104 - 460) \cdot 2104$$
$$+ (900 - 0.1 \cdot 1534 - 315) \cdot 1534$$
$$+ (900 - 0.1 \cdot 852 - 178) \cdot 852]]$$
$$= -0.1 - 0.0000001 \cdot \frac{2}{3} \cdot 2{,}212{,}864.8 \approx -0.147$$

שאלה 3

$H_W(x)$

- ישר הרגרסיה, מאפיין אחד

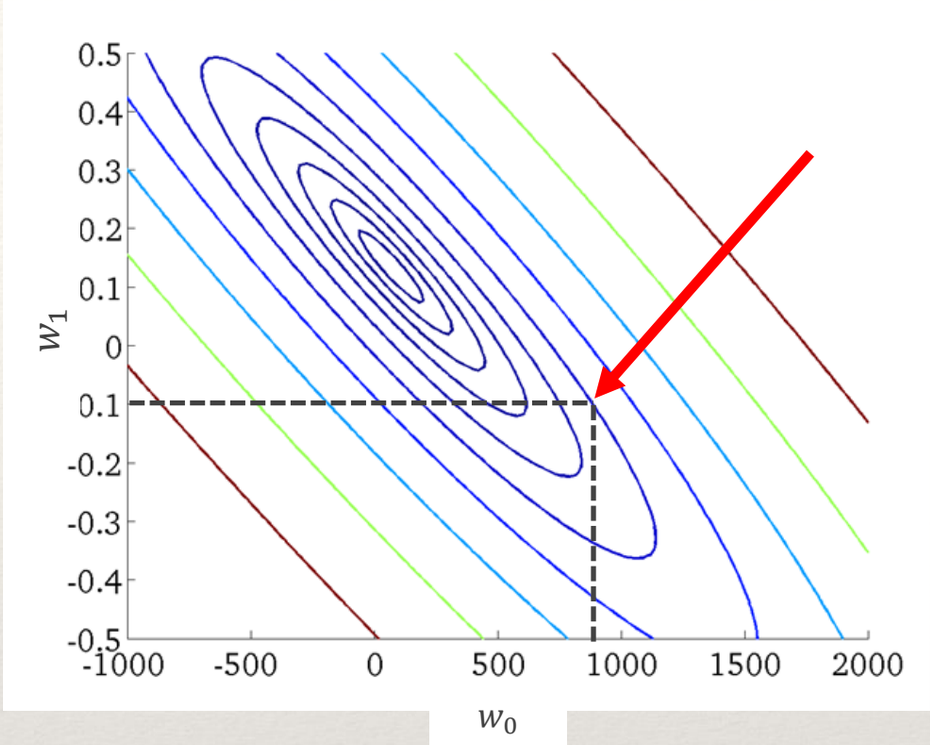
- פונקציה של x, עבור w_0,w_1 מסוימים



$J(w_0,w_1)$

- פונקצית המחיר (מדמה גרף תלת מימדי)

- פונקצית המחיר - פונקציה של w_0,w_1



שכר יומי בש"ח	סכום הציונים
460	2104
315	1534
178	852

Train set

<u>פתרון לאחר איטרציה אחת:</u>

w_0 = **899.99**

w_1 =**-0.147**

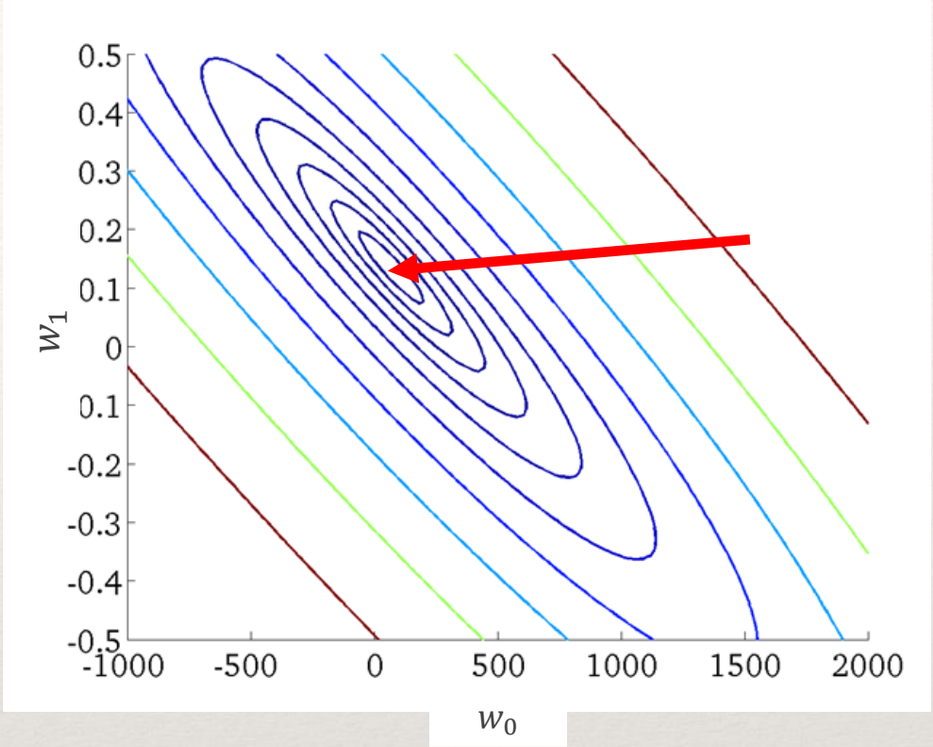
$H_W(x)$= 899.99- 0.147x

שאלה 3

$J(w_0,w_1)$

- פונקצית המחיר (מדמה גרף תלת מימדי)

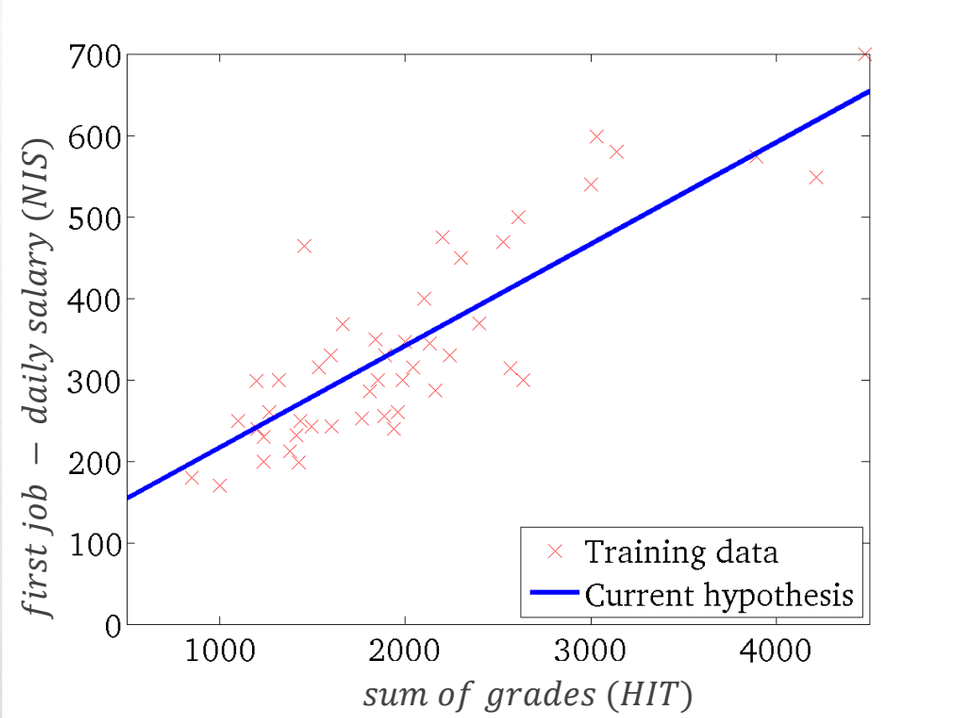
- פונקצית המחיר - פונקציה של w_0,w_1



$H_W(x)$

- ישר הרגרסיה, מאפיין אחד

- פונקציה של x, עבור w_0,w_1 מסוימים



שכר יומי בש"ח	סכום הציונים
460	2104
315	1534
178	852

Train set

... לאחר כמה סבבים

רגרסיה לינארית (linear regression)
ריבוי משתנים (multivariate) - מוטיבציה

Gradient Descent – for linear regression - summary

univariate linear regression

$$\left\{(x_i, y_i)\right\}_{i=1}^{n}$$

$$\hat{y}_i = w_0 + w_1 x_i$$

$$J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

$$\min_{w}\left[J(\vec{w})\right]$$

$$w_0 := w_0 - \frac{2\alpha}{n}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)$$

$$w_1 := w_1 - \frac{2\alpha}{n}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)\cdot x_i$$

multivariate linear regression

$$\left\{(\vec{x}_i, y_i)\right\}_{i=1}^{n}$$

$$\hat{y}_i = \vec{w}\cdot\vec{x}_i$$

$$J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

$$\min_{w}\left[J(\vec{w})\right]$$

$$\vec{w} := \vec{w} - \frac{2\alpha}{n}\sum_{i=1}^{n}(\vec{w}\cdot\vec{x}_i - y_i)\cdot\vec{x}_i$$

Linear Regression with Gradient Descent

Gradient-Descent(*S: training-examples & targets, α*)

Initialize each w_j with small random number

Until <u>TERMINATION</u> Do

- Initialize each Δw_j to zero
- For each $<x_i, y_i>$ *in S* Do
  - Compute $\hat{y}_i = \vec{w}\,\vec{x}_i$
  - Update Δw_j values for example i as following (for each Δw_j):
    - $\Delta w_j = \Delta w_1 - \alpha \frac{2}{n}(\hat{y}_i - y_i)\, x_{ij}$
- For each weight w_j Do
  - $w_j = w_j + \Delta w_j$

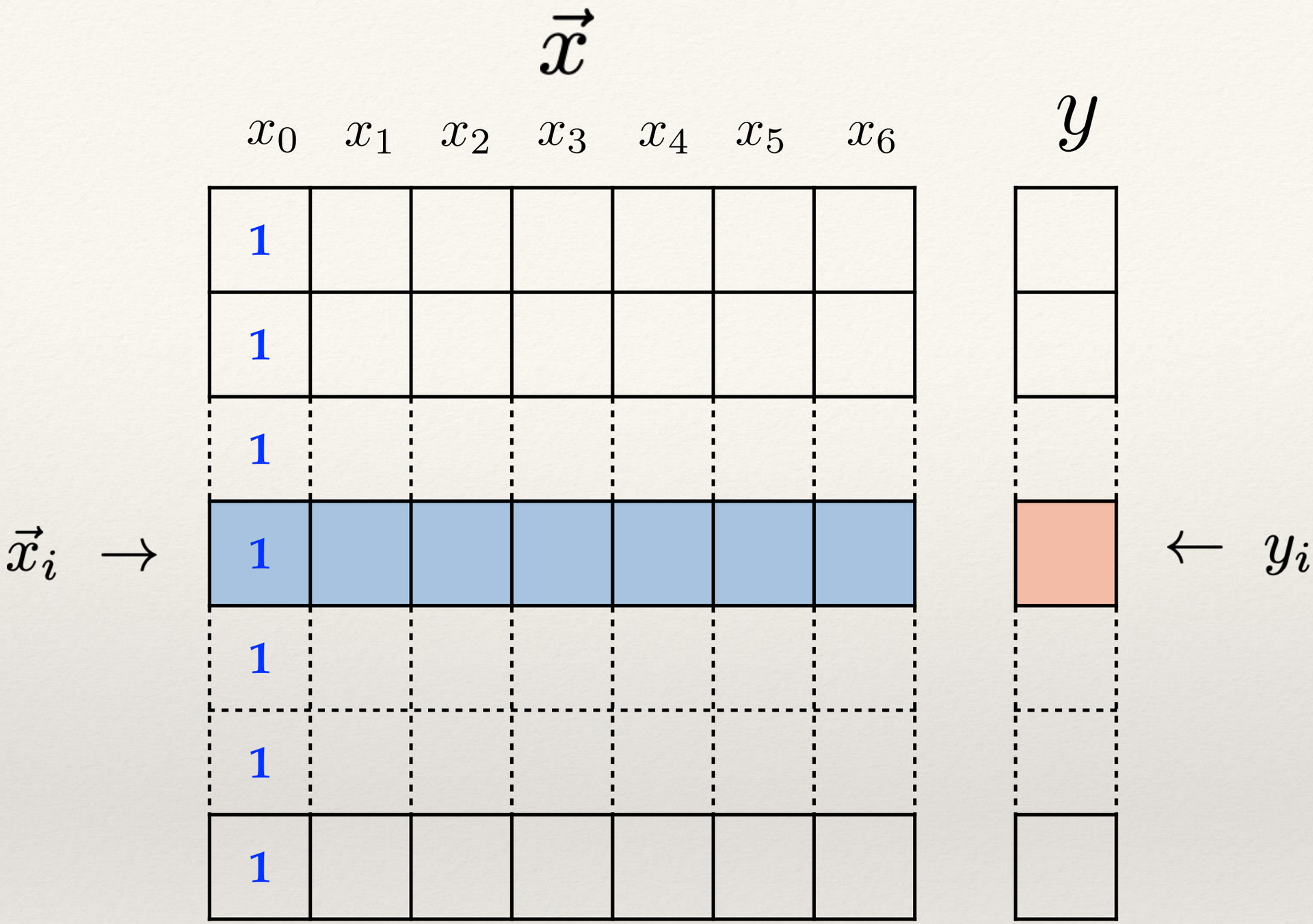
$\hat{y}$ –predicted value
y – actual target value
α - learning rate

multivariate linear regression

במקרה זה, לכל וקטור באימון יש יותר ממאפיין אחד (למשל: גודל הדירה, קומה, כיווני-אוויר, וכו')

Price ($K) **(y)**	Size (meter^2) **(x4)**	Number of bedrooms **(x3)**	Number of floors **(x2)**	Age of home (years) **(x1)**
460	2104	5	1	45
232	1416	3	2	40
315	1534	3	2	30
178	852	2	1	36

$$H_W(x) = \vec{w}^T \cdot \vec{x} = w_0 \cdot x_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_d \cdot x_d$$



המודל הלינארי -עבור רגרסיה מרובת משתנים:

$$\hat{y}_i = \vec{w} \cdot \vec{x}_i$$

קומבינציה לינארית של המאפיינים

פונקצית מחיר (Cost Function)

המודל הלינארי: $\hat{y}_i = \vec{w} \cdot \vec{x}_i$

פונקצית המחיר

$$J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\vec{w}\cdot\vec{x}_i - y_i)^2$$

$(n \geq 1)$

Multivariate Gradient Descent Algorithm - for linear regression:

Repeat until done:

We want w_0 to be partialy derived as the rest of $\vec{w}$, so if j=0, $x_{i,0} = 1$

$$w_j = w_j - \alpha \cdot \frac{\partial J}{\partial w_j} = w_j - \alpha \cdot \frac{2}{n} \cdot \sum_{i=1}^{n}\left[(\vec{w}\cdot\vec{x}_i - y_i)\cdot x_{i,j}\right]$$

(simultaneously update w_j for j=0,…,d)

Gradient Descent - שאלה 4

Multivariate Gradient Descent Algorithm - for linear regression:

Repeat until done:

$$w_j = w_j - \alpha \cdot \frac{\partial J}{\partial w_j} = w_j - \alpha \cdot \frac{2}{n} \cdot \sum_{i=1}^{n} \left[(\vec{w} \cdot \vec{x}_i - y_i) \cdot x_{i,j} \right]$$

תרגיל –

נתונים $\alpha=0.1$, $w_j=0$, לכל j

ונתון ה-train set

בצעו סבב אחד של Gradient Descent

תשובה:

Train set

Size (meter^2)	Number of bedrooms	Age of home (years)	Price ($K)
2104	5	45	460
1416	3	40	232
852	2	36	178

Gradient Descent - שאלה 4

Multivariate Gradient Descent Algorithm - for linear regression:

Repeat until done:

$$w_j = w_j - \alpha \cdot \frac{\partial J}{\partial w_j} = w_j - \alpha \cdot \frac{2}{n} \cdot \sum_{i=1}^{n} \left[(\vec{w} \cdot \vec{x}_i - y_i) \cdot x_{i,j} \right]$$

תרגיל –

נתונים α=0. 1, w_j=0, לכל j

ונתון ה-train set

בצעו סבב אחד של Gradient Descent

Train set

Size (meter^2)	Number of bedrooms	Age of home (years)	Price ($K)
2104	5	45	460
1416	3	40	232
852	2	36	178

Predicted (Ŷ)	(Ŷ-Y)xi,0	(Ŷ-Y)xi,1	(Ŷ-Y)xi,2	(Ŷ-Y)xi,3
0	-460	-967840	-2300	-20700
0	-232	-328512	-696	-9280
0	-178	-151656	-356	-6408
Sum	-870	-1448008	-3352	-36388
α*2*1/n*Sum	-58	-96533.9	-223.4667	-2425.86667
new w vals	58	96533.87	223.46667	2425.866667

תשובה:

נחשב את ערכי הנגזרות החלקיות ונכפיל ב-α

נחשב את ערכי וקטור המשקולות ($\vec{w}$) החדשים ...

שאלה: האם ערכי $\vec{w}$ המעדוכן נראים סבירים? מדוע זה קרה?

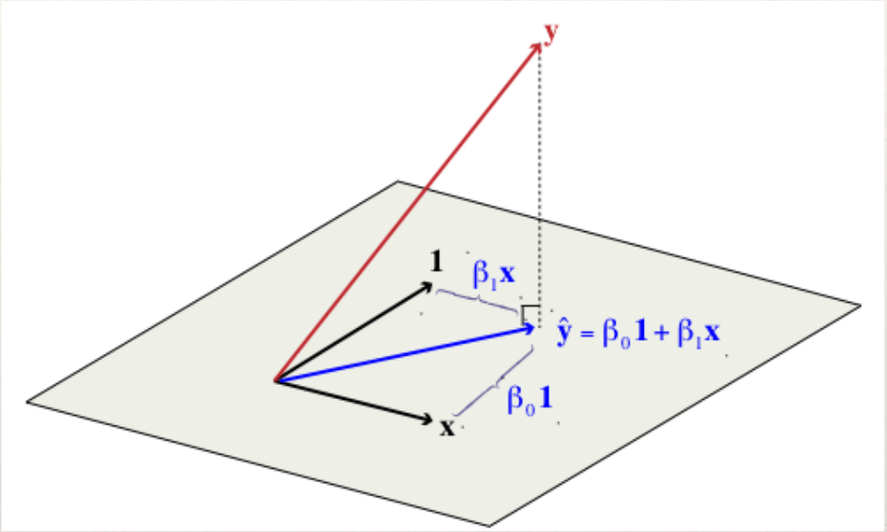
רגרסיה לינארית (linear regression)
- היבטים שונים ב-flow

חזרה על סוף מצגת ההרצאה על רגרסיה

Linear regression – features should not be correlated

Linear regression - predicted value is a linear combination of the feature values

→ Doesn't work well if features are dependent.



$$\text{NMI} = \frac{I\,(f1; f2)}{|H(f1) + H(f2)|/2}$$

Possible solution – look for correlation between features

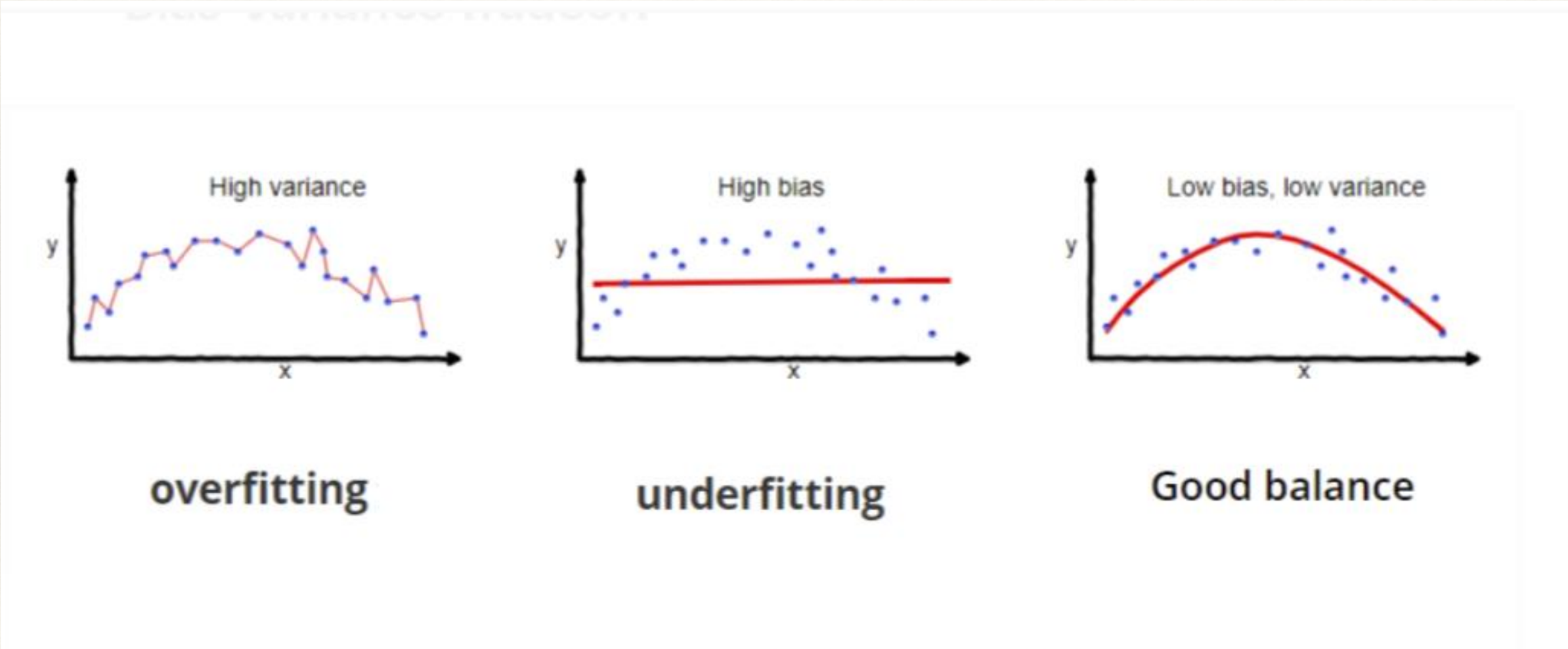
- Pearson correlation coefficient
- Normalized Mutual information

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

$$= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Overfitting vs Underfitting

Linear regression tends to overfit, mainly if there are many features involve (even after removal of dependent features)



Credit: geeksforgeeks.org

Overfitting vs Underfitting - solutions

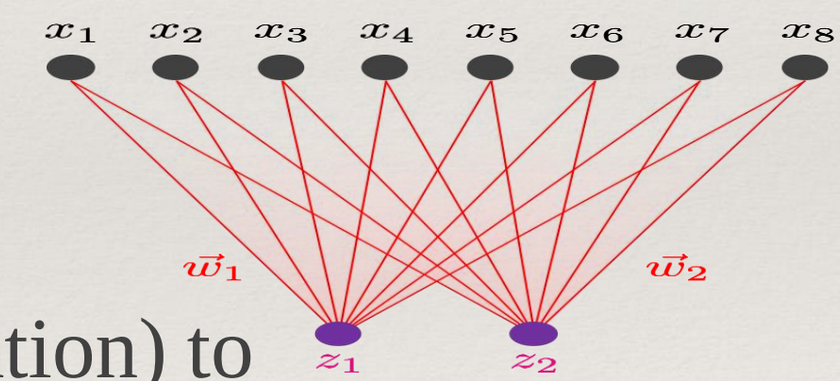
Linear regression tends to overfit, mainly if there are many features involve (even after removal of dependent features)

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

$$= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Solution 1 – Reduce correlated features

- ❖ Remove correlated features (e.g., Pearson correlation)
- ❖ Remove correlation (e.g., PCA)



Solution 2 – Stop before convergence

- ❖ Use validation set (e.g., k-fold cross-validation) to choose a stop point

Regularization - different values of lambda

Solution 3 – Regularization

Keep all the features but reduces the magnitudes of the hypothesis parameters.

- works well when all features contribute to the target prediction.

L_1 Regression (Lasso regression)

cost function: $J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \cdot \sum_{j=1}^{d}|w_j|$

We want to keep weights small

L_2 Regression (Ridge regression)

cost function: $J(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \cdot \sum_{j=1}^{d} {w_j}^2$

Objective: min $J(\vec{w})$,

- With $very\ large$ λ causes only w_0 to matter, we want a trade-off between them for a trade-off between overfitting and underfitting

$\lambda = 0$: We'll get the same coefficients as simple linear regression.

$\lambda = \infty$: The coefficients will be zero.

$0 < \lambda < \infty$: We want our values in between

Credit: Ria Nag @ analyticsvidhya.com

סילום (Scaling)

Gradient descent

- Gradient descent convergence more quickly
  - Might not converge if no scaling and alpha is not tuned.

Linear regression - predicted value is a linear combination of the feature values

- Scaling assists for understanding (importance of features)

Regularization

- Significant for regularization

Popular Scaling actions

- Popular range: [-1,1]
- Standardization
- centralization is popular (after, mean=0)

Hyperparameters and tuning

- α – קבוע הלמידה
- Epocs – מספר האיטרציות המקסימלי
- λ – קבוע הרגולריזציה
- שיטת הרגולריזציה

גם עבור רגרסיה לנארית נוכל לבצע hyperparameter tuning בשיטות שלמדנו

רגרסיה לינארית – יתרונות וחסרונות

יתרונות:

- קל למימוש, להבנה והסבר
- ניתן להסיק על חשיבות המאפיינים
- עובד די טוב גם עם train-set קטן
- זמן אימון מהיר
- סיבוכיות מקום נמוכה
- רוב החסרונות ברות טיפול (למשל טיפול ב-overfitting בעזרת regularization)

חסרונות:

- לא מתאים כשאין קשר לינארי ומתקשה שההיפותזה מורכבת
- נטייה ל-overfitting – במיוחד בריבוי מאפיינים
- מתקשה לטפל במאפיינים לא רלוונטיים וברעש
- לא עובד טוב ללא סילום (scaling)
- צריך לוודא חוסר תלות בין המאפיינים
- הטעות צריכה להתפלג נורמלית

Other regression algorithms

Other regression algorithms

- **KNN Regressor**

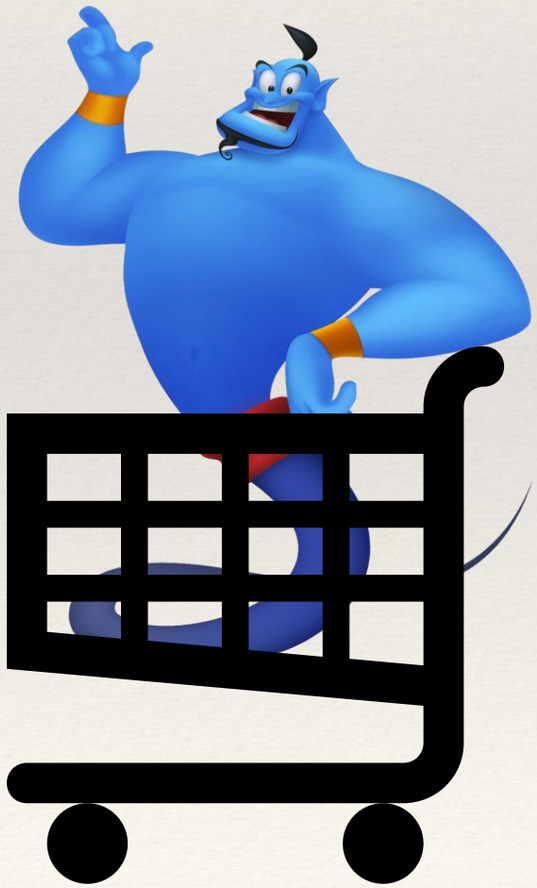
And many more

- CART – Classification and regression trees
- Ensembles with week CARTs (e.g., Random forest regressor)
- Support Vector Regressors (we will learn SVM for classification)
- Artificial Neural Networks (we will learn ANN for classification)

CART - Classification and Regression Trees

CART - Classification and Regression Trees

- ❖Only Binary splits
- ❖use of Gini Index instead of Information Gain/Entropy
- ❖More popular than ID3/C4.5
- ❖Uses pruning



Gini impurity

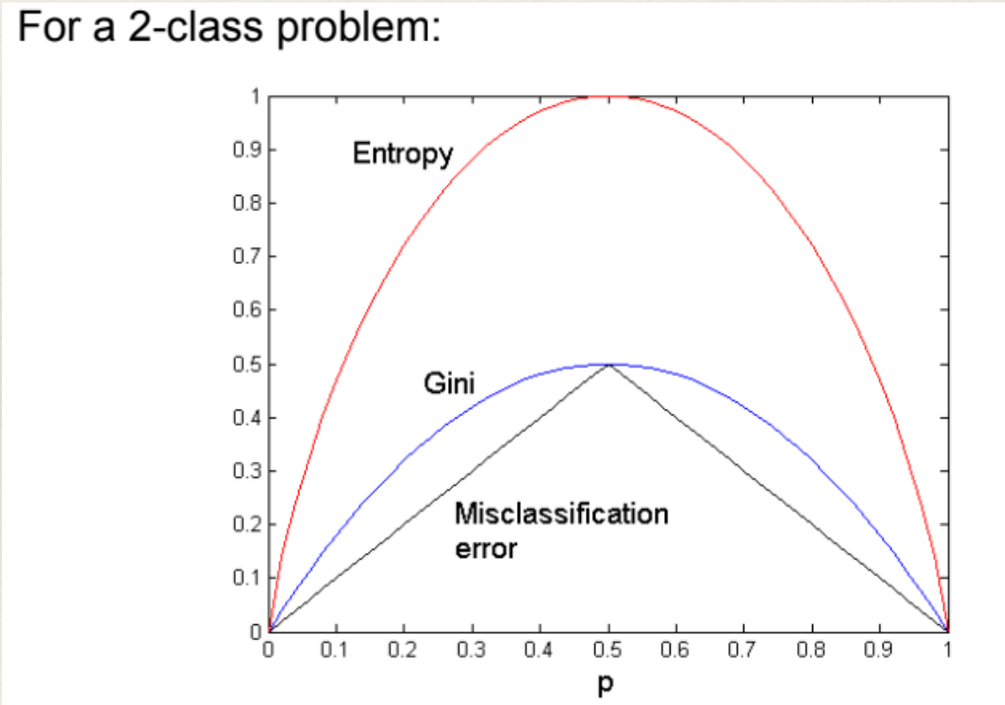
CART - Classification and Regression Trees

- Instead of entropy, impurity can be measured by the Gini index

$$Gini(S) = 1 - \sum_i p_i^2$$

- average Gini index

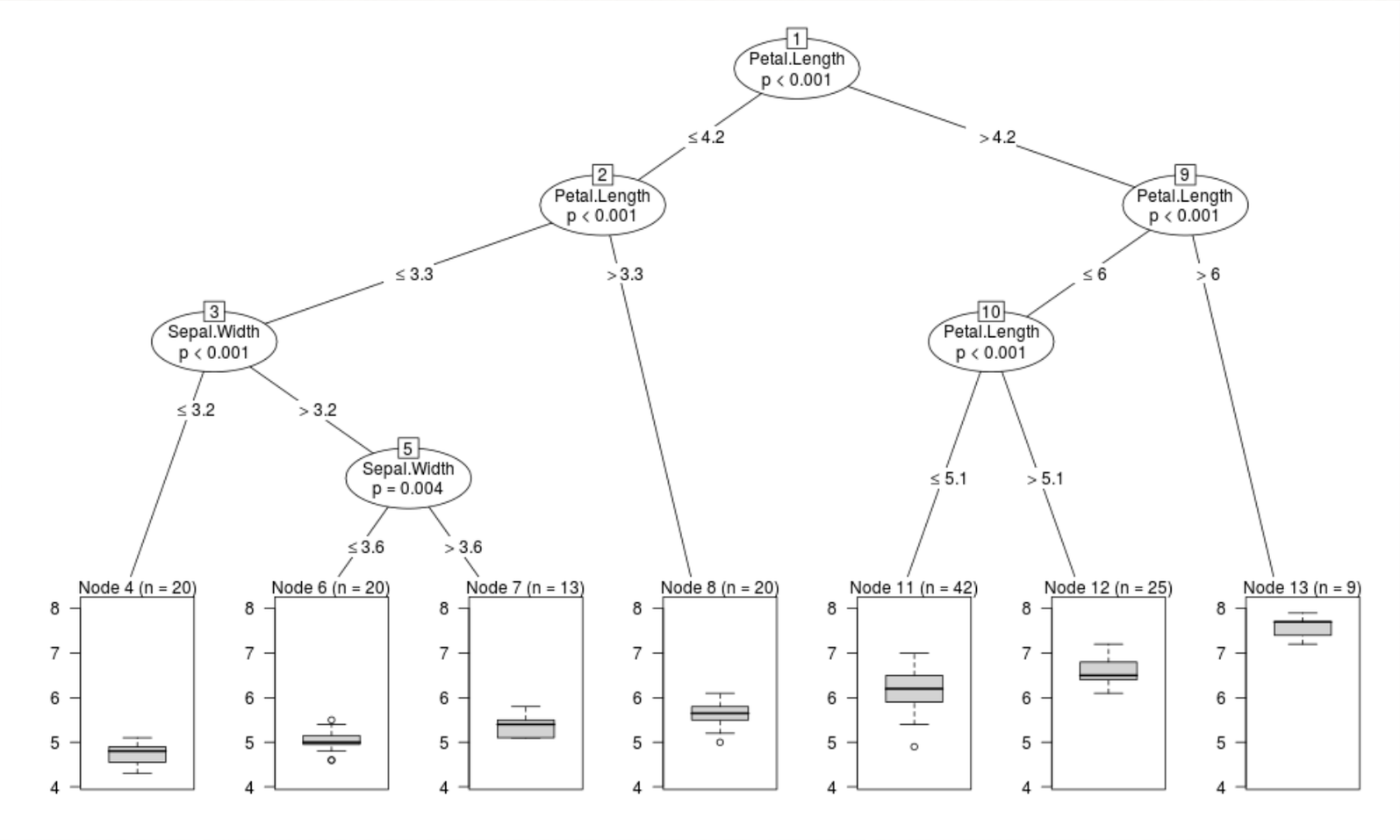
$$Gini(S, A) = \sum_i \frac{|S_i|}{|S|} \cdot Gini(S_i)$$



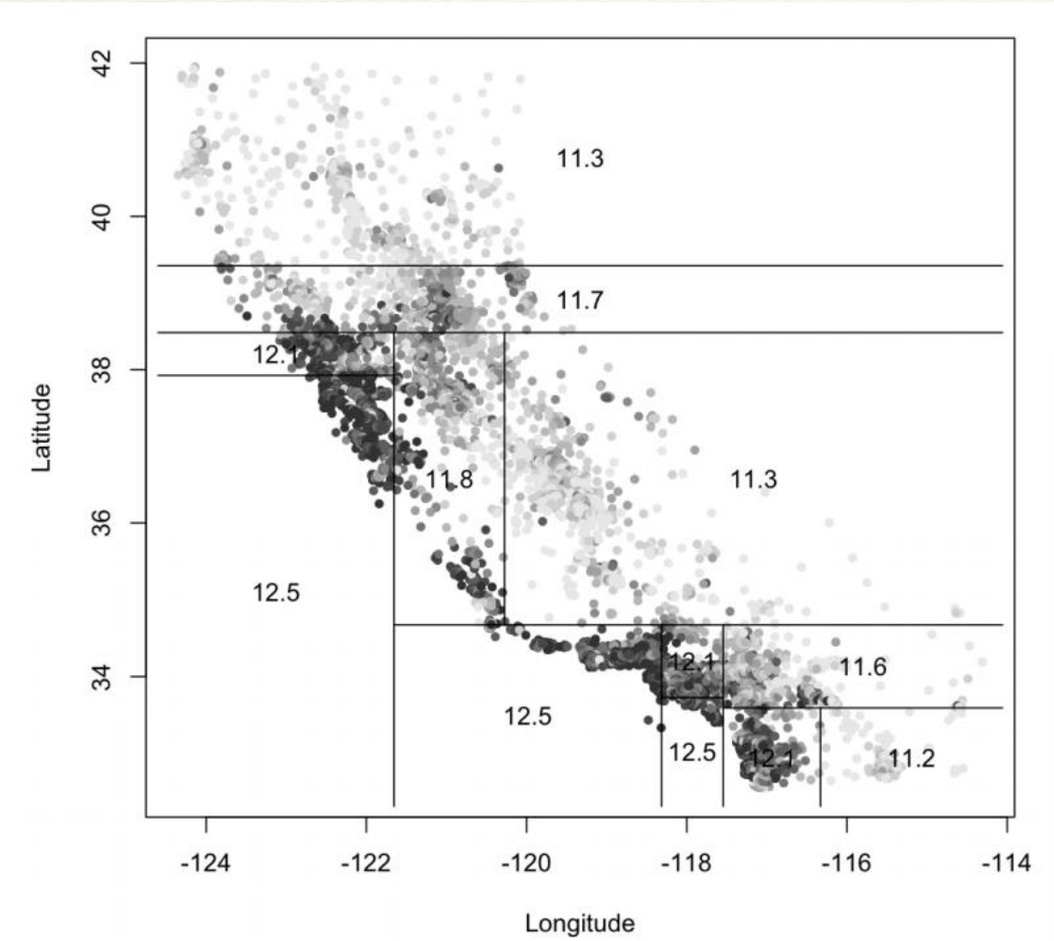
Regression Trees

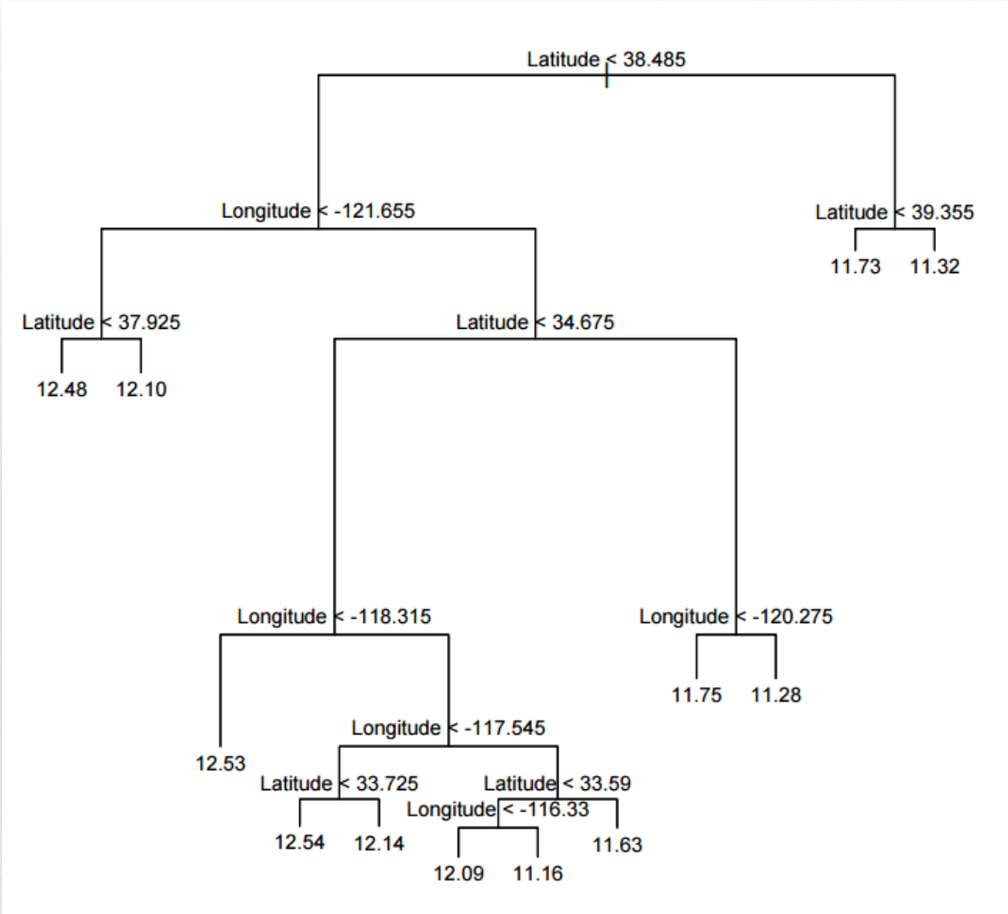
How to choose attributes?

- ❖Instead of entropy/gini use variance
- ❖Early stopping - must
- ❖Use mean, median etc.



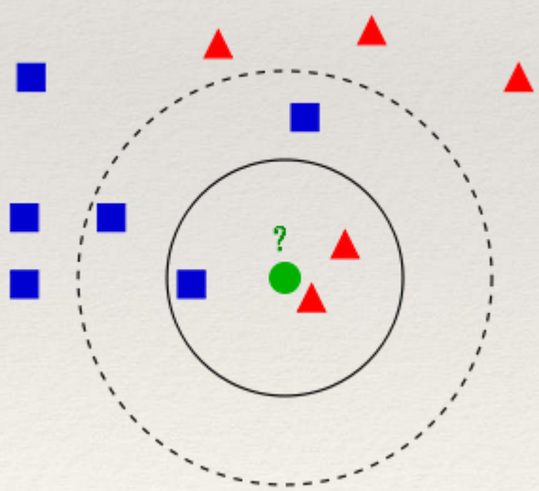
Example: location -> house price

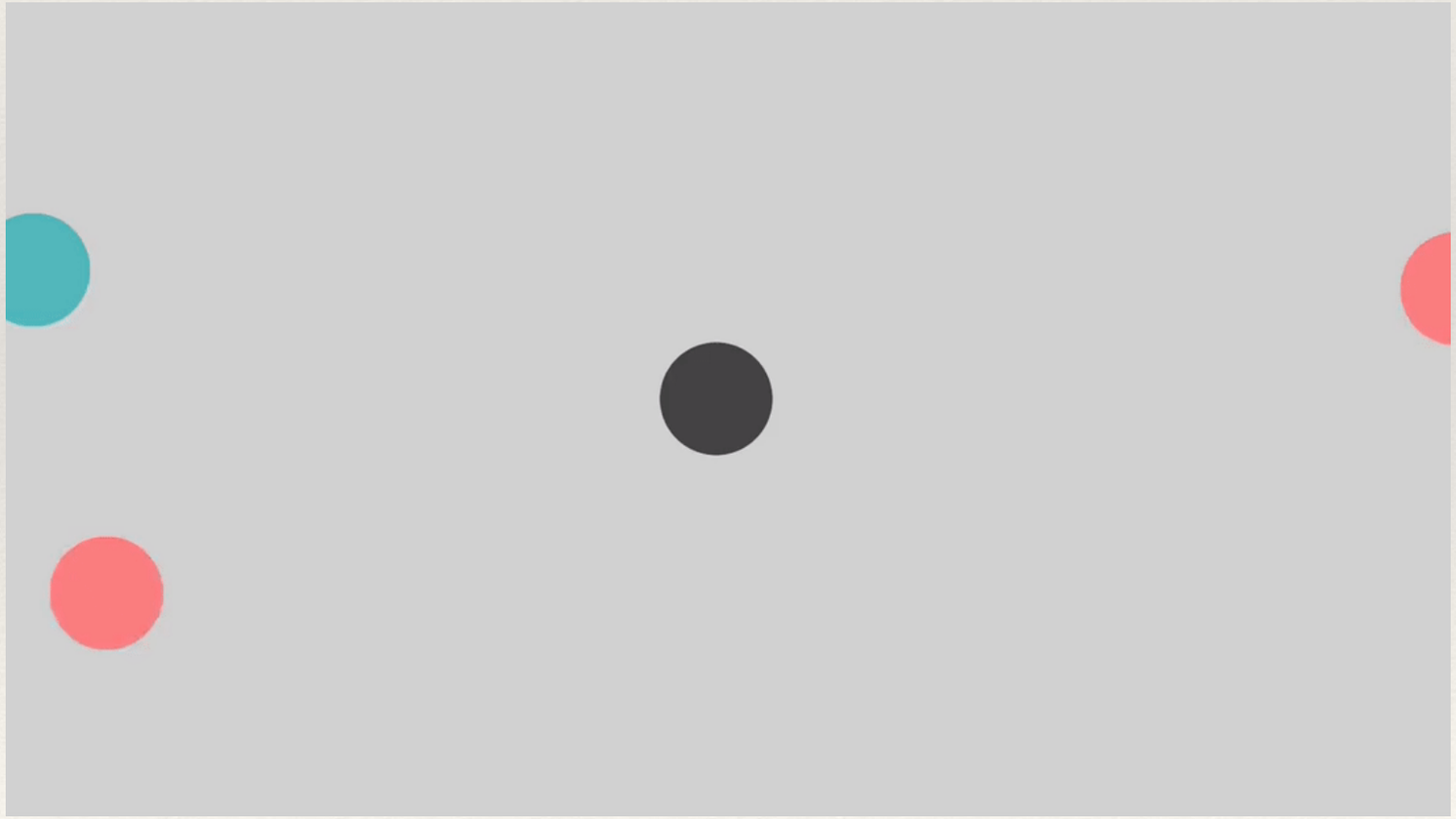




KNN – both for classification and regression

- The new example, is marked with a black circle
- The examples from the training are marked in blue and red (2 classes) circles
- In example K = 3





KNN for Regression – the algorithm

Input:

- Hyper parameters: k – the number nearest neighbors; d – the distance metric; Averaging – uniform/weighted
- Training set (<X,y>)

The KNN Algorithm:

- for test instance x_j in the test-set:
  - Calculate $d(x_j,x_i)$
  - Select the k closest training examples, $d(x_j,x_i)$ sorted
  - Predict value = average of target values from nearest neighbors

Notations and Terms:

$\mathbf{x_j}$ – example from the test-set

$\mathbf{x_i}$ – example from the train-set

$\mathbf{d(x_j,x_i)}$ – distance between x_j and x_i.

KNN for Regression –Exercise

<u>notations</u>:

A feature vector with two features X1,X2, will be written as (x1,x2)

<(x1,x2)|t> denotes the example and its expected target (for training set examples)

X1=num of people, x2=num of days, target=price of trip

<u>Our training set</u>:

<(3,2)|50>, <(4,2)|100>, <(4,3)|400>, <(7,4)|1000>, <(6,7)|2000>, <(4,6)|800>, <(6,2)|300>

Question: Predict the value for the new feature vector (6,4), using KNN

Use Manhattan distance and K=1. What is the closest distance to the new example and how will it be classifier?

A. Distance=2, target =300

B. Distance=1, target =1000

C. Distance=3, target =400

D. Distance=4, target =800

KNN for Regression –Exercise

<u>notations</u>:

A feature vector with two features X1,X2, will be written as (x1,x2)

<(x1,x2)|t> denotes the example and its expected target (for training set examples)

X1=num of people, x2=num of days, target=price of trip

<u>Our training set</u>:

<(3,2)|50>, <(4,2)|100>, <(4,3)|400>, <(7,4)|1000>, <(6,7)|2000>, <(4,6)|800>, <(6,2)|300>

Question: Predict the value for the new feature vector (6,4), using KNN

Use Manhattan distance and K=1. What is the closest distance to the new example and how will it be classifier?

A. Distance=2, target =300

B. Distance=1, target =1000

C. Distance=3, target =400

D. Distance=4, target =800

להתראות בשבוע הבא ☺