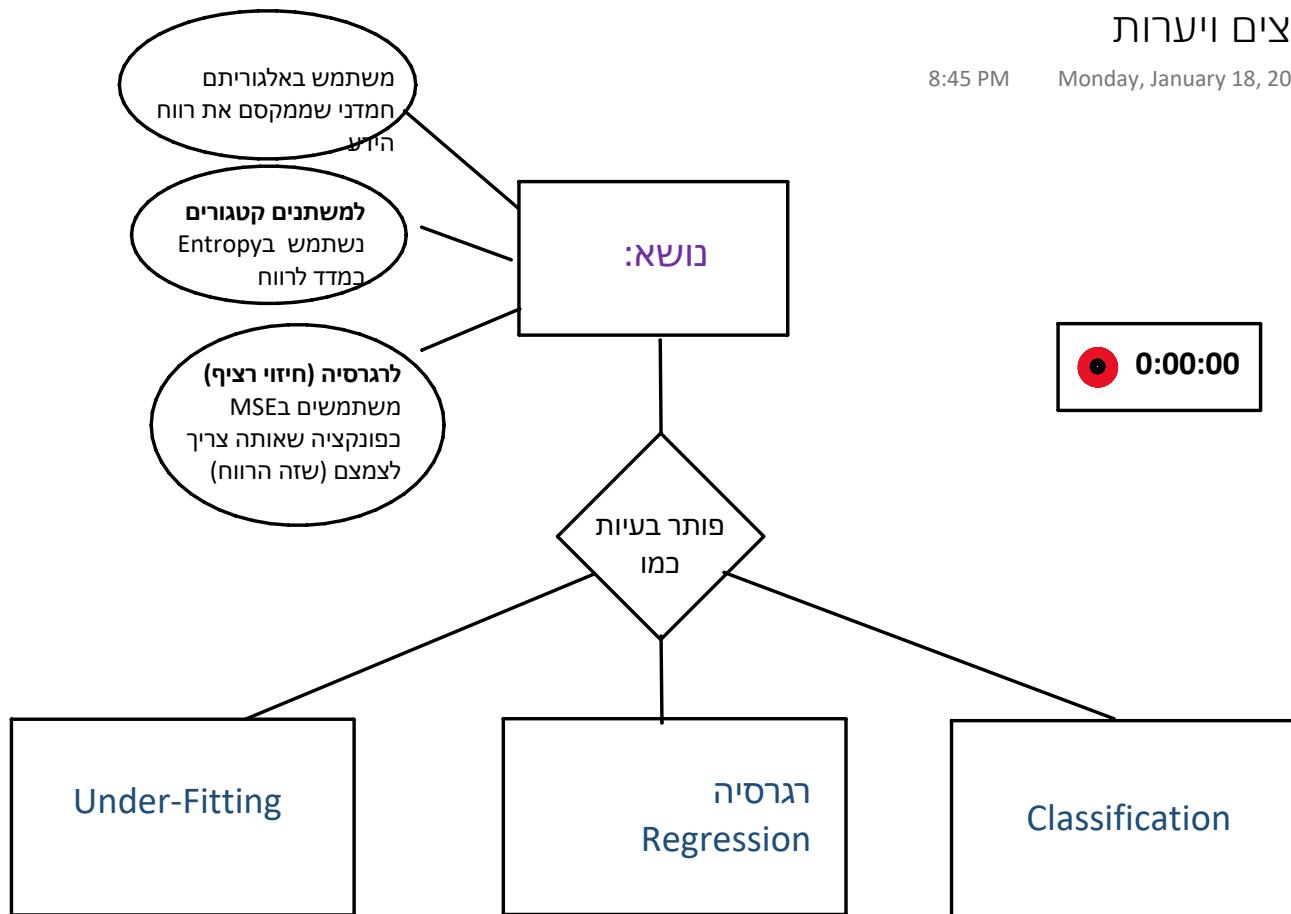
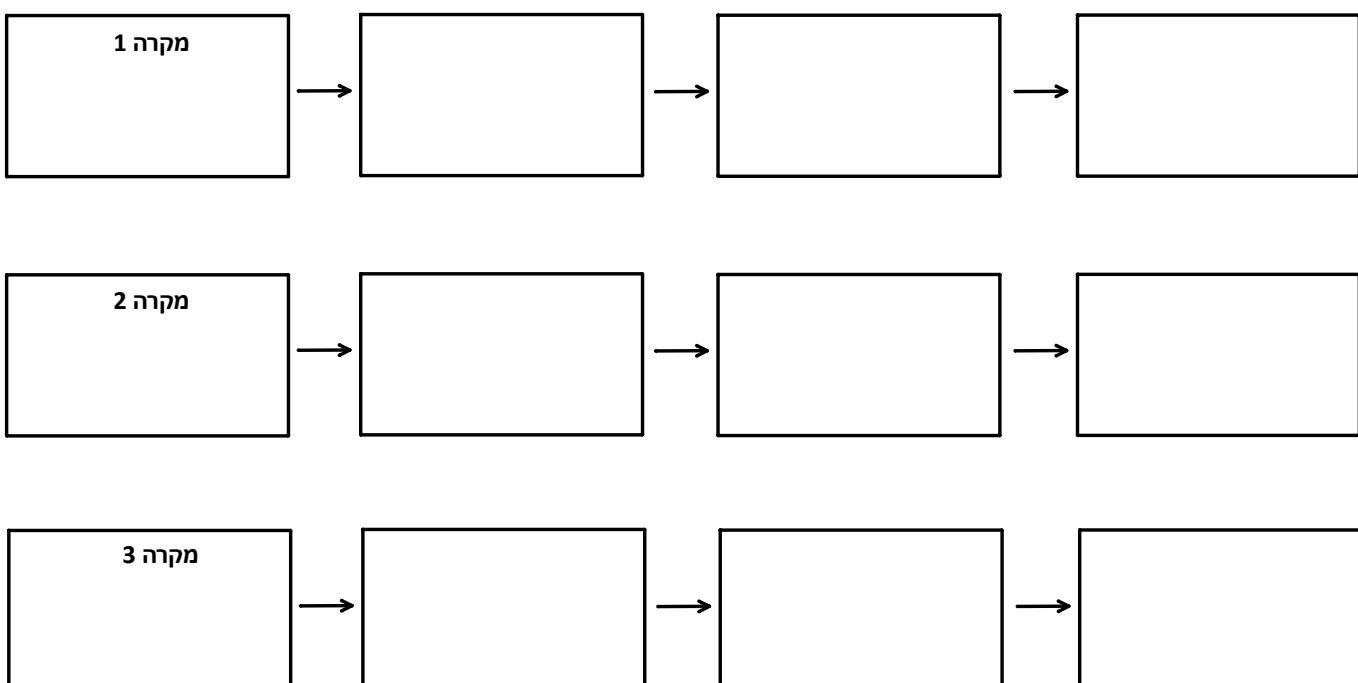


0:00:00



אור נפתרו?



# Decision Trees

## for Classification & Regression

יתרונות:

- קל להבין ולהסביר לא מומחים
  - ישנם המאמינים כי עצי החלטה משקפים טוב את אופן קבלת החלטות האנושי
  - ניתן להפיק חוקים (די מובנים לבני אדם) מענפי העץ ולשים קבלת החלטות שקל להסביר
  - ע"י הפעלת החוקים בזאת אחר זה
  - שגיאת ביאס קטנה: גמישים מספיק כדי לקרב כל פונקציה
  - ניתן לדרג (rank) את ה features לפי חשיבותם (כמה אינפורמציה הם תורמים)
  - ניתן להפעיל על נתונים עתק: scale to big data
  - ההיסטוריה ארוכה והרבה הרחבות שימושיות:
- cart(77), ID3 (83), c4.5 (94), VeryFastDT(2000), C5(2007), sprint parallel data mining

חרוגות:

- בקלות מבצע התאמת יתר, שגיאת ואריאנס (שליטה מוגבלת בעזרת רגוליזציית מרכיבות – על עומק העץ)
- שינויים קטנים בתנומות עלולים לגרום לשינויים בעז
- עלולים להשתמש במאפיינים לא רלוונטיים ולהיות גדולים שלא לצורך
- הዮיריסטיקות לבחירת מאפיין, כמו : IG, Gini, לא תמיד מיצירות את העץ הכי טוב
- האלגוריתם מעדיף עצים קטנים. ולא תמיד מייצר לנו את העץ האופטימלי
- שינוי קטן בקבוצת האימון יכול להשפיע דראסטית על מבנה העץ

### פונקציית Gain

#### Entropy:

$$I_D = -\left(\frac{1}{2} * \log_2 \frac{1}{2}\right) - \left(\frac{1}{2} * \log_2 \frac{1}{2}\right) = 1$$

$$I_1 = -\left(\frac{1}{3} * \log_2 \frac{1}{3}\right) - \left(\frac{2}{3} * \log_2 \frac{2}{3}\right) = 0.918$$

$$I_2 = -(1 * \log_2 1) - (0 * \log_2 0) = 0$$

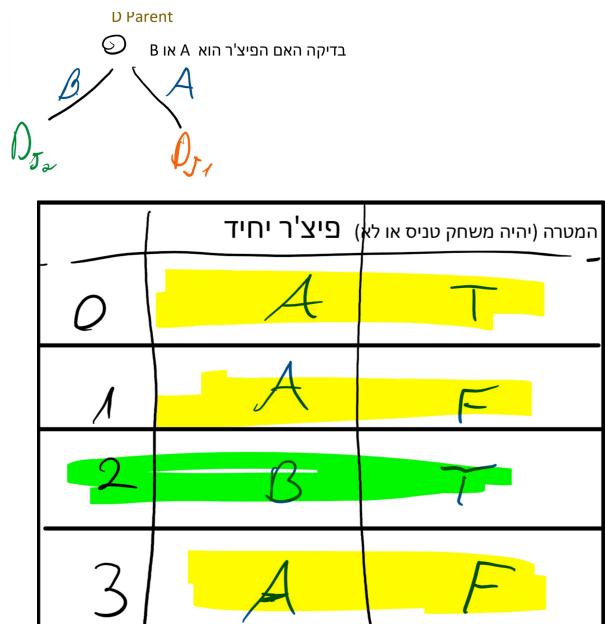
$$IG_{entropy} = 1 - \left(0 * \frac{1}{4} + 0.918 * \frac{3}{4}\right) = 0.3115$$

#### חישוב לפי Gini

$$I_D = \frac{1}{2} * \frac{1}{2} + \frac{1}{2} * \frac{1}{2} = \frac{1}{2}$$

$$I_1 = \left(\frac{1}{3} * \frac{2}{3}\right) + \left(\frac{2}{3} * \frac{1}{3}\right) = \frac{4}{9} = 0.444$$

$$I_2 = 1 * (1 - 1) + 0 * (0 - 1) = 0$$



$P(i|t)$  - ההסתברות של  $i$  נגיד  $true$  בקבוצה  $t$ .

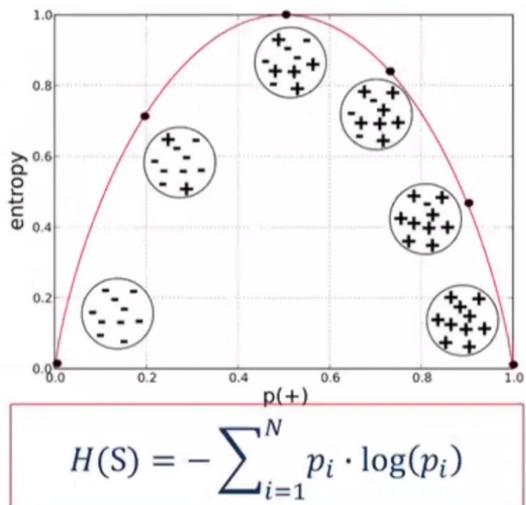
נגיד בקבוצה שיש פה בטבלה

ההסתברות ל $True$  שווה ל  $\frac{2}{4}$  שזה שווה גם

להסתברות ל $false$  (במקרה זה)

$$IG_{gini} = \frac{1}{2} - \left( \left(0 * \frac{1}{4}\right) + 0.444 \left(\frac{3}{4}\right) \right) = 0.167$$

## מדד אי-סדר Entropy



אם לדוגמה יש לנו מטבע מיוחד שבו אנחנו תמיד מקבל עץ - האנתרופיה תהיה קטנה:  
 $-0 * \log(1) - 1 * \log(0) = 0$

אם יש לנו מטבע אמיתי (ההסתברות היא חצי-חצי)  
 $entropy = -0.5 * \log(0.5) - 0.5 * \log(0.5) = \frac{1}{2} + \frac{1}{2} = 1$

נחשב entropy לדוגמה של 4 (למשל הטלת שתי מטבעות, קיבל כל צירוף זה וגע)

$$Entropy = -\left(\frac{1}{4} * \log_2\left(\frac{1}{4}\right)\right) * 4 = -\left(\frac{1}{4} * 2\right) * 4 = 2$$

כלומר -> ככל שיש יותר בלאגון, כך הentropy יותר גדולה

לאחר פיצול בודקים את H של 2 הקבוצות שנוצרו

| Name   | age | gender | Balance (\$) | Employed | Default |
|--------|-----|--------|--------------|----------|---------|
| Mike   | 42  | M      | 200,000      | Yes      | No      |
| John   | 37  | M      | 35,000       | No       | Yes     |
| Mary   | 40  | F      | 115,000      | No       | No      |
| Robert | 23  | M      | 72,000       | Yes      | No      |
| Dora   | 31  | F      | 29,000       | No       | Yes     |

ההסתברות לקבל יلد מסוים) \* (האנתרופיה שלו)

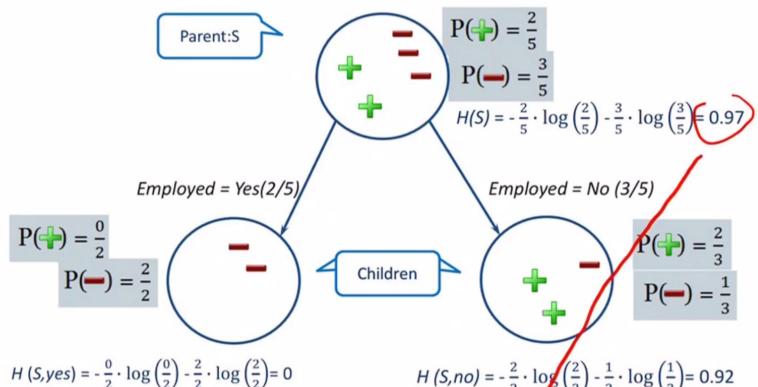
$$H(D) = -[P_{Yes} \cdot \log(P_{Yes}) + P_{No} \cdot \log(P_{No})] = -\frac{2}{5} \cdot \log\left(\frac{2}{5}\right) - \frac{3}{5} \cdot \log\left(\frac{3}{5}\right) = 0.97$$

נפוץ לפני : Employed

$$Employed = Yes: \quad H(D, E=yes) = -\frac{0}{2} \cdot \log\left(\frac{0}{2}\right) - \frac{2}{2} \cdot \log\left(\frac{2}{2}\right) = 0$$

$$Employed = No: \quad H(D, E=no) = -\frac{2}{3} \cdot \log\left(\frac{2}{3}\right) - \frac{1}{3} \cdot \log\left(\frac{1}{3}\right) = 0.92$$

האנתרופיה המשוקלلت של הילדיים:



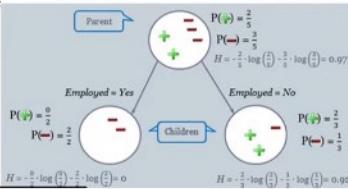
אנתרופיה משוקלلت של 2 הקבוצות שהופרדו:

$$P(S, Emp=Yes) \cdot H(S, Emp=Yes) + P(S, Emp=No) \cdot H(S, Emp=No) \\ = \frac{2}{5} * 0 + \frac{3}{5} * 0.92 = 0.552$$

## Information Gain

$$IG(\text{Parent}, \text{children}) = H(\text{parent}) - \sum_{i=1}^{\#\text{children}} P(\text{child}_i) \cdot H(\text{child}_i)$$

- $IG(\text{parent}, \text{children}) = \text{entropy}(\text{parent}) - [P(\text{Emp}=\text{Yes}) \cdot \text{entropy}(\text{Emp}=\text{Yes}) + P(\text{Emp}=\text{No}) \cdot \text{entropy}(\text{Emp}=\text{No})]$   
 $= 0.97 - \left[ \frac{2}{5} \cdot 0 + \frac{3}{5} \cdot 0.92 \right] = 0.97 - 0.552 \approx 0.42$
- Employed feature added 0.42 amount of information, i.e. it reduced the amount of uncertainty (entropy) in target variable, from 0.97 to 0.55



• מאפיין מולטי-קטגוריאלי - למשל צבע : מפרידים ל 2 או יותר תת קבוצות של ערכים.

למשל:

• ערך אחד מול כל השאר: Color = Red?

• נפוץ ל 2 או יותר קבוצות זרות של ערכים:

Color is in {red, pink, orange}, {green, blue}, {white, black, green}

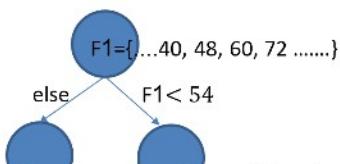
## כיצד נפוץ Real value feature מחפשים סף של פו נפוץ

נבדוק את ה GO למספר ספים שבuzzרתם ניתן לבצע פיזול בינהו...

|             |    |    |    |    |    |    |
|-------------|----|----|----|----|----|----|
| Temp:       | 40 | 48 | 60 | 72 | 80 | 90 |
| PlayTennis: | No | No | Ye | Ye | Ye | No |

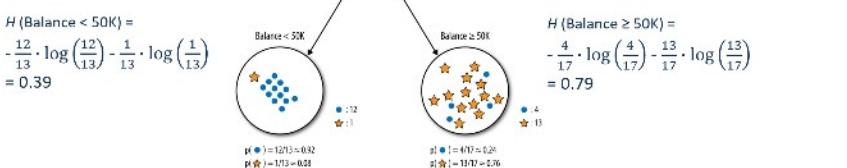
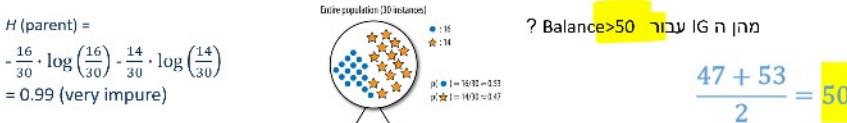
נסדר את ערכי feature עליה (וברשימה נפרדת את ערכי המתאים).  
נבדוק את הערך infoGain רק בנקודות שבהם מתחלף ה-target

- $(48+60)/2=54$
- $(80+90)/2=84$
- מוגבהת לחישוב Info gain המקסימלי להיות באחד המעברים  
(פיזול שאיןנו במעבר תמיד ייתן פחות הומוגניות מאשר פיזול במעבר)



## Example: Predicting mortgage default using balance (continuous feature)

|          |     |     |     |     |     |     |     |
|----------|-----|-----|-----|-----|-----|-----|-----|
| Balance: | 27K | 28K | 30K | 39K | 47K | 53K | 57K |
| Default: | Ye  | Ye  | Ye  | No  | Ye  | No  | No  |



$$IG = H(\text{parent}) - [P(C_L) \cdot H(C_L) + P(C_R) \cdot H(C_R)] = 0.99 - \left[ \frac{13}{30} \cdot 0.39 + \frac{17}{30} \cdot 0.79 \right] = 0.37$$

**GrowTree(S):**

```

if (t=0) for all <x,t> ∈ S, return (new leaf(0))
Else if (t=1) for all <x,t> ∈ S, return (new leaf(1))
Else
choose "best" feature  $x_j$ 
 $S_0 = \{x, t\} \in S \text{ with } x_j = 0$ 
 $S_1 = \{x, t\} \in S \text{ with } x_j = 1$ 
return (new node ( $x_j < 0.5$ , GrowTree( $S_0$ ),
GrowTree( $S_1$ ))

```

נתונה קבוצה  $S$  של דוגמאות עם מאפיינים בינהירים ותווית מטרית ביןארית  $0 \leq y \leq 1$

אם הקבוצה היא הומוגנית מבחינת  $y$ , ניצור ממנה עלה, ונסמן על פי ה `target`. סימנו

אחרת (S איננה הומוגנית): נבחר מאפיין שיפצל "הכי טוב" ל 2 קבוצות נפרדות. כמה שייתר הומוגניות ב  $y$

נחזיר עץ שהשורש שלו הוא השאלה שנבחרה כדי לפצל,  
ויש לו 2 ילדים שבכורה ורקורסיבית ממשיכים להתפצל לעוד קבוצות יותר וייתר הומוגניות עד שהקבוצות "מספיק" הומוגניות או שלא ניתן יותר לפצל

אם ניתן להפריד את קבוצת האימון לקבוצות הומוגניות, האלגוריתם יבנה עץ שיפריד

לחילופין,  
ניתן להפסיק לפצל אם השיפור בהומוגניות (או בשגיאה) איןנו משמעותי או שיש מגבלות גודל לעץ.  
כך מסתפקים בעצים קטנים יותר שנוטנים שגיאת או אריאנס קטנה (בהתואгу לעצים גדולים)

תנאי העצירה,  
אם באחד הענפים כל הדוגמאות הן 0 (לא יהיה משחק טניס) נחזיר 0  
כנ"ל גם ל 1 (יהיה משחק) אם לכל הדוגמאות שבקבוצה יהיה משחק -> נחזיר 1

הערה: הקבוצה לא חייבת להיות הומוגנית. יש מצבים שבהם נוצר כאשר יש גם דוגמאות שאומרות שהיא משחק טניס, וגם באלה שלא.

ואז אם הגיע לענף זהה, נגיד שב 98% נגיד הוא חודה שהיא משחק. אז נרצה שיש משחק ונגיד "בטוחן של .... אחותו"  
מתי זה יקרה?

1. אם הגיעו לUMB שאותו פיצ'ר שיפור את התחזית שלנו,
2. שהשגיאה על העלה קטנה
3. שהעץ יהיה עמוק מדי.

### הבעיה עיקרית של העצים זה **overFitting**

טיפול אפשרי זה לקטץ את הגובה  
כל שהעץ יותר קטן, בר סביר יותר  
שנקבל שגיאת bias גדולה יותר

# בעז - איך ניתן להוריד את שגיאת הוויריאנס (Over Fitting)

## שיטת להפחית שגיאות Variance בעזרת Ensembles



### שיטת 1 Bagging

הננו אכן מוריד את השגיאות, אבל בכלל שהעצים הם ד"י דומים אחד לשני (לפחות הצמתי הקרובים לשורש) ולכן איננו לא מצליחים ממש להוריד את השגיאת Variance בצורה משמעותית.

תזכורת: BootStrap לוקח הרובה פעמים את אותן דוגמאות (ולכן יש בין 2p, D1 .. חפיפה גדולה)

תזכורת 2: הסיכוי שדוגמה לא תכנס ל-set training, לומר שהיא תהיה בולידציה הוא 1/e (קרוב לשיש)

### Bagging using Bootstrap Aggregation

- נבצע bootstrapping (דגימה עם חזרות) כדי לגדל N עצים שונים מ- B מודלים שונים קצת זה מזה.
- נמציע את תוצאות החיזוי של העצים השונים:
  - ברגראטייה ממצאים את החיזוי
  - בקלסיפיקציה משתמשים ב-vote majority.

אפשר לגדל עצים לא מקוצצים שעולים לעשوت התאמת יתר: לכל עץ תיתן שגיאת ואריאנס גבואה אשר תתמצע ע"יensemble

בפרקтика מגדים אפילו אלפי עצים ומחברים לפופולריות חיזוי אחת

חיסרון?

העצים דומים מדי אחד לשני. ולכן שגיאת הוויריאנס אינה יורדת ממשמעותית מכיוון שדגימות ה-bootstrapping היקן קורלטטיבית והצמתים (המשמעותיים) בסמוך לשורש, נשרים לרוב קבועים במקומם בעז

## שיטת 2 - יער Random Forest

### Random Forest

איננו לא מרים לאלגוריתם לקבל את כל הפיצ'רים ולבחר את ההבי טוב מביניהם, אלא: איננו בוחרים פיצ'רים בצורה רנדומית. ורק אוטם איננו בודקים ( רקUberם איננו בודקים את החוג gain information )

בכל שלב איננו מגירים את הפיצ'רים שניתן לבדוק אותם מחדש

1. מננים לחזות משתנה בדיד או רצוף
2. ככל ש'וח יותר קטן, כך שגיאת הביאס תהיה יותר גדולה  
(הביאס יהיה יותר נמוך)
3. ככל שיש יותר עצים, שגיאת הביאס יורדת וזמן החישוב עולה  
לינארית
3. MSE עבר גרסיה,  
Entropy - עבר קלסיפיקציה
4. יותר עמוק - יותר וריאנס, יותר גמיש
5. עדיף להשתמש עם החלקה (חלוקת לפלים למשל)

היפר פרמטרים עיקריים:

1. גרסיה/קלסיפיקציה
2. 'מ' (משפיע על הביאס- וויריאנס)
3. מספר העצים ביער (משפיע על הוויריאנס וגם על זמן החישוב)
4. קריטריון לבחירת מאפיין "טוב": IG, Gini, MSE ...
5. עומק העץ / רגוליזציה של מרכיבות העץ
6. אופן חילוק ההסתברויות בעלים ...

# Random Forests:

## מגבלה רנדומית על ה Features הניתנים לבחירה

ב- Fset מה?

כפי' פיצ'ר קטgoriy שבסח'ר, לא יבחר יותר רצים לבחר מאפיין עבור צומת (בהינתן קבוצת דוגמאות S, רשימה מאפיינים מותרים I 'm): הפונקציה מחזירה את המאפיין שבסח'ר גם קבוצת מאפיינים שניית לה שימוש בהם כדי להמשיך ולפצל בעומק העץ.

(בפי'צ'ר נומריה זה לא משנה)

[ChooseBestFeature\(S, Fset, m'\)](#) returns (BestFeature, UpdatedFeatureSet)

If Fset is empty return (Null,Null)

בהתחלחה קבוצה המאפיינים המקוריים בעומק העץ Fset לא כוללת את המאפיינים שנבחרו ע"י האבות

RandFset= Randomly select m' features from Fset (if |Fset|<m', select all features in Fset)

BestF=FindBestF(S, RandFset) // select the feature with the largest IG

remove from Fset,  
return(BestF, Fset)

בוחרים רנדומית קבוצה של 'm' רק חלק קטן מה Features שעומדים לבדיקה.

דוגמיה:

1. אין טעם לשאול בעמיהם האם ירד גשם

2. יש טעם לשאול אם ירד מתחת 20 מ"מ - ואז אם ירד מעל 10 מ"מ

הבחירה במגבלה 'm' מאפיינים היא היפר-פארמטר:

- מה יקרה כאשר 'm'=m ?
- מייצר bootstrap bagging רגיל: העצים יהיו דומים אחד לשני בתלות בשוני במדגים.
- מיקובל (ברירת המחדל בהרבה חבילות תוכנה)  $m = \sqrt{m}$

כאשר יש הרובה מאפיינים והם קורלטיבים אחד עם השני, נבחר  $m < m'$

## יער - אנסמבל של עצי החלטה

### מושיבציה

עצים יכולים להיות עם שונות מאוד גודלה Over Fitting ולכן אנחנו לא ממש סובבים על ההחלטה שלהם.

### הרעינו

ניקח אוסף גדול של עצי החלטה,

1. כל אחד מהם נגדל מעט (או שוחתו אותו). לא ניתן לו להגיע לגובה גדול.

1. נאמן כל אחד מהם על תחת קבוצה של פיצ'רים.  
נכיח שיש לנו 1000 פיצ'רים : אז שבל אחד מהם יתאים על 5-6

1. כל אחד מהם אנחנו נאמן רק על חלק מהדעתה  
נכיח שיש מיליון דוגמאות.. נתן לכל אחד מהם רק עשרה-אלפים או אולי בחירה בעזרת BootStrap

בר שבל אחד מהעצים הקטנים שלנו יהיה מומחה בפי'צ'ר מסוים שונה.

1. ניקח את אוסף העצים האלה (נגדיר בין 50 ל-200 עצים)  
ונעשה להם Majority Vote. כל אחד מהעצים יגיד מה דעתו, אנחנו ניקח את התוצאה שיש לה הכى הרובה הצבעות.

## האלגוריתם המלא לבניית עץ (בינארי) רנדומי

בכל צומת, שמים מגבלה אקראית על ה Features הנחוצים לבחירה

**GrowRandBTree(S, Fset,m')**

```
If t= 0 for all <x,t> in S, new leaf(0)
Else if t=1 for all <x,t> in S , new leaf(1)
Else
  BestF, Fset =ChooseBestFeature(S, Fset, m')
  if BestF ==Null, return
  S0=all <x,t> in S, where XBest=0
  S1=all <x,t> in S, where XBest=1
  returnNew node(XBest,GrowRandBTree(S0, Fset, m'), GrowRandBTree(S1, Fset, m'))
```

בחירה ה features מוגבלת  
לרשימת רנדומית של m'  
features

**ChooseBestFeature(S,Fset, m') returns (BestFeature, UpdatedFeatureSet)**

```
If Fset is empty return (Null,Null)
RandFset= Randomly select m' features from Fset (if |Fset|<m', select all features in Fset)
BestF=FindBestF(S, RandFset) // select the feature with the largest IG
  remove from Fset,
  return(BestF, Fset)
```

## סיכום: Random Forest bagging & bootstrap with random feature sets

אם  $m < m'$  בכל פעם שבוחרים צומת לפיצול, האלגוריתם לא מורה להסתכל על רוב המאפיינים

- לא מגבלה על בחירה המאפיינים, סביר שהרוב העצים היה נבחר אותו מאפיין (דומיננטי ב GO) עברו השורש
- הסיכוי לבחירה לא ליהיבר הוא  $m/(m-n)$  ולכן ככל ש 'm' נמוך, יש סיכוי רב שהצמתים של עצים שונים לא יהיו זהים.  
ז.א. המאפיין ה"חזק" ביותר לא ילקח בחשבון בסיכוי של  $m/(m-n)$ .
- כמקרה פרטי, ב 2 עצים שונים, אפילו השורש לא יהיה זהה בהסתברות  $m/m'$

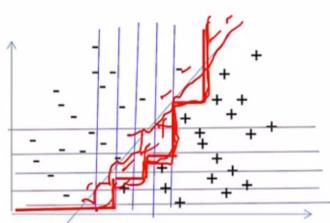
ב bagging ללא מגבלת מאפיינים, אכן רואים כי העצים קורלטיביים  
ב random Forest העצים פחות קורלטיביים ככל ש 'm' קטן יחסית ל 'm'

בחירה במגבלת 'm' מאפיינים היא היפר-פארמטר: ותלויה בד"כ בקורסציה שיש בין המאפיינים

כאשר יש הרבה מאפיינים והם קורלטיביים אחד עם השני, נבחר  $m < m'$

נפוץ בחבילות תוכנה להשתמש בברירת מחדל:  $m = \sqrt{m'}$

עדי החלטה יכולים להפריד כל קבוצת דוגמאות (שלא מכילה סטייה)!



אבל האם נוכל ללמידה הפרדה לינארית (קו ישר)  
בעזרת עץ החלטה?

מה אי אפשר  
לייצר על ידי עצי  
ההחלטה? הפרדה  
LINEARIT

בעדי החלטה ניתן רק לחזור אופקי או אנכי...!! אבל,

ניתן לקרב כל פונקציה (להפרדה מושלמת) בעזרת חיתוכים אופקיים ואנכיים.