**סיכום לכריית נתונים:**

**מבוא לכריית נתונים:**

**יעדי הקורס:**  
-ללמוד טכניקות של כריית נתונים  
- לראות את השימוש בהם בעולם האמיתי או ביישומי מחקר  
- להבין את המגבלות של טכניקות סטטיסטיות סטנדרטיות ביישומי כריית נתונים   
- להבין את העקרונות המתודולוגיים מאחורי כריית נתונים  
- להיות מסוגל לקרוא על כריית נתונים בעיתונות הפופולארית בעין ביקורתית  
- ליישם ולהשתמש במודלים של כריית נתונים באמצעות תוכנה סטטיסטית(R, ראטל)

**מהו R?**-קוד פתוח לתוכנה סטטיסטית שצמחה מתוך S/Splus   
-הרבה משתמשים בחבילות ב-CRAN  
-יש עזרה למשתמש, לוח מודעות וכו'   
-יש הדרכות מקוונות זמינות על R באינטרנט  
-גרפיקה טובה  
 **מהי כריית נתונים?**  
כריית מידע או כריית נתונים באנגלית: (Data mining) היא הפעלת [אלגוריתם](http://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D) או [תוכנית מחשב](http://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%99%D7%AA_%D7%9E%D7%97%D7%A9%D7%91) לצורך גילוי מידע הטמון ב[בסיסי נתונים](http://he.wikipedia.org/wiki/%D7%91%D7%A1%D7%99%D7%A1_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) קיימים, ו[הסקת מסקנות](http://he.wikipedia.org/wiki/%D7%94%D7%99%D7%A7%D7%A9) מהצלבתו. גילוי ידע בבסיסי נתונים הוא תהליך שנועד לחקור ולנתח כמות גדולה של מידע באמצעים אוטומטיים ככל שניתן כדי לגלות דפוסים תקפים חדשים, שימושיים (מקבלי ההחלטות יכולים להשתמש בדפוסים שהתגלו כדי לשפר תהליכים קיימים) ובעלי משמעות (מובנים למקבל ההחלטות). עם ההתפתחות המתמדת בעולם טכנולוגיית המידע, כשכמות המידע הדיגיטלי גדלה כל הזמן, עולה החשיבות של תחום כריית המידע, שבאמצעותו ניתן להפוך [מידע](http://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%93%D7%A2) ל[ידע](http://he.wikipedia.org/wiki/%D7%99%D7%93%D7%A2).

המושג כריית מידע נגזר מפעולת הכרייה [הגאולוגית](http://he.wikipedia.org/wiki/%D7%92%D7%90%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%94) בה כורים באדמה כדי למצוא אוצרות טבע.

המונח "כריית נתונים" מתאר שלב אחד בתוך התהליך המכונה "גילוי ידע בבסיסי נתונים" - השלב בו מריצים אלגוריתמים במחשב. מלבד שלב כריית הנתונים כולל התהליך שלבים אחרים, חלקם קודמים לכריית הנתונים (כגון הכנת החיתוך) וחלקם עוקבים (כגון ניתוח התוצאות). הבעיות הנפוצות של כריית מידע הן בעיות סיווג, [ניתוח אשכולות](http://he.wikipedia.org/wiki/%D7%A0%D7%99%D7%AA%D7%95%D7%97_%D7%90%D7%A9%D7%9B%D7%95%D7%9C%D7%95%D7%AA), רגרסיה ולמידת חוקי אסוציאציה.

**תהליך כריית מידע:**

**עיבוד מוקדם**

מאחר שכריית מידע נסמכת על גילוי תבניות הנמצאות בנתונים, מאגר הנתונים שעליו נעשה התהליך צריך להיות גדול דיו כך שיכיל את התבניות מחד גיסא, ומאידך גיסא תמציתי מספיק כדי שניתן יהיה לחפש בו בזמן סביר. מקור מקובל למידע כזה הוא [מחסן נתונים](http://he.wikipedia.org/wiki/%D7%9E%D7%97%D7%A1%D7%9F_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) (Data warehouse) או [מחסן נתונים מחלקתי](http://he.wikipedia.org/w/index.php?title=%D7%9E%D7%97%D7%A1%D7%9F_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D_%D7%9E%D7%97%D7%9C%D7%A7%D7%AA%D7%99&action=edit&redlink=1) (DataMart).  
לאחר מכן נעשה סינון של הנתונים בו מנוקים נתונים עם רעש או מידע חסר. מן הנתונים המסוננים מיוצרים [וקטורים מאפיינים](http://he.wikipedia.org/w/index.php?title=%D7%95%D7%A7%D7%98%D7%95%D7%A8_%D7%9E%D7%90%D7%A4%D7%99%D7%99%D7%9F&action=edit&redlink=1) (Feature vector), וקטור אחד לכל תצפית. וקטור מאפיין הוא גרסה סיכומית של רשומת התצפית. כך לדוגמה תמונת פנים של 100 פיקסלים על 100 פיקסלים תכיל 10,000 ביטים של מידע גולמי. את התמונה ניתן להמיר לווקטור מאפיין על ידי זיהוי העיניים והפה בתמונה. בצורה זו ניתן להפחית את כמות המידע מ-10,000 ביטים לווקטור, לשלושה קודים המציינים את מיקומי העיניים והפה, וכך בשלב כריית המידע עצמו יידרש פחות עיבוד. בחירת המאפיין (או המאפיינים) תלויה במטרה; לפיכך בחירת המאפיינים הנכונים הכרחית לכריית מידע מוצלחת.  
הווקטורים המאפיינים מחולקים לשתי סדרות: סדרת אימון וסדרת בדיקה. סדרת האימון משמשת לאימון האלגוריתם, וסדרת הבדיקה משמשת לאימות הדיוק של התבניות שנמצאו.

**כריית המידע מטפלת לרוב באחת מהבעיות הבאות:**

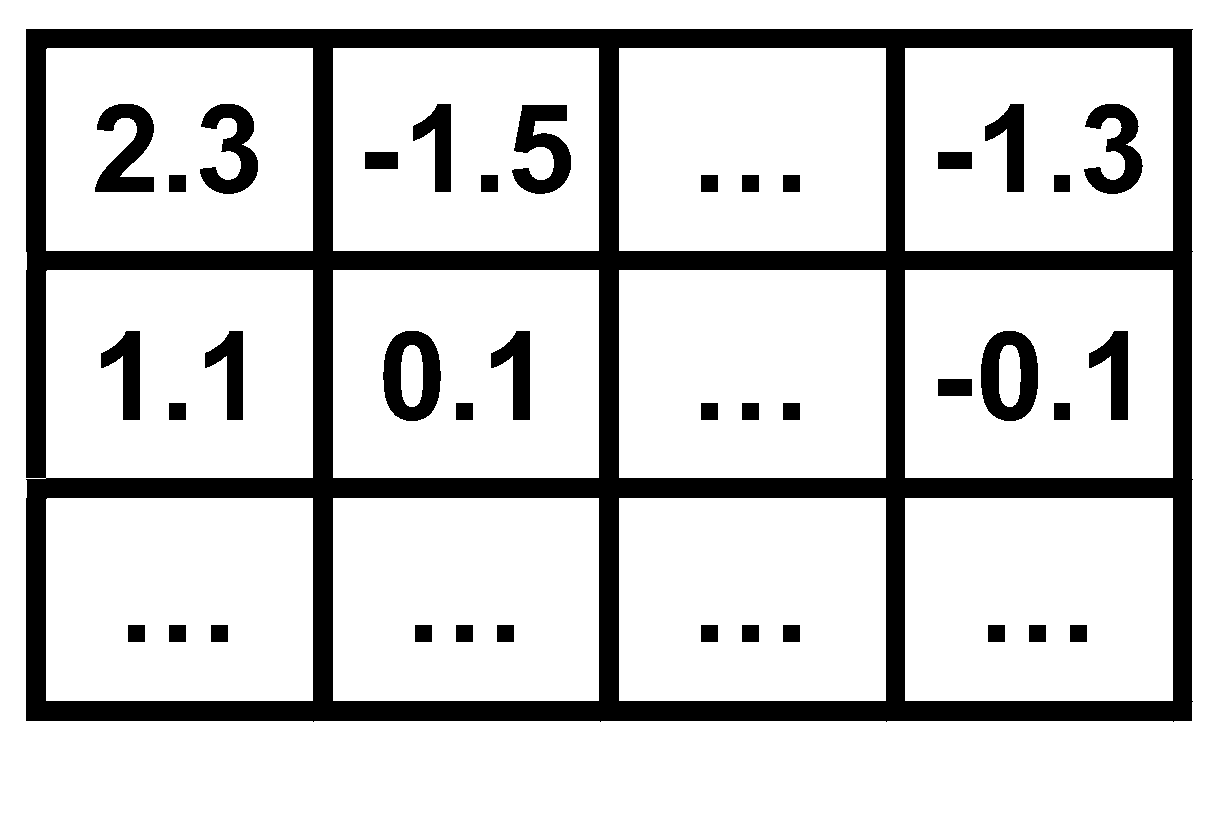
* [**בעיות סיווג**](http://he.wikipedia.org/w/index.php?title=%D7%A1%D7%99%D7%95%D7%95%D7%92_%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99&action=edit&redlink=1)**-** בהתאם לדוגמאות שנאספו בעבר, האלגוריתם לסיווג מסיק מודל סיווג המשמש לסווג מקרים חדשים. למשל [בנק](http://he.wikipedia.org/wiki/%D7%91%D7%A0%D7%A7) יכול לבנות מודל סיווג של לווי [משכנתא](http://he.wikipedia.org/wiki/%D7%9E%D7%A9%D7%9B%D7%A0%D7%AA%D7%90) בהתאם לדוגמאות שנאספו בעבר. כאשר מגיע לקוח חדש המבקש משכנתא הוא מסווג לפי מידת הוודאות שהוא אכן יצליח לעמוד בעתיד בהחזרי ההלוואה. אלגוריתמים נפוצים לכך הם [עץ החלטה](http://he.wikipedia.org/wiki/%D7%A2%D7%A5_%D7%94%D7%97%D7%9C%D7%98%D7%94) (Decision Tree), [שיטת השכן הקרוב](http://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D_%D7%A9%D7%9B%D7%9F_%D7%A7%D7%A8%D7%95%D7%91) (Nearest neighbor),[סיווג בייס נאיבי](http://he.wikipedia.org/w/index.php?title=%D7%A1%D7%99%D7%95%D7%95%D7%92_%D7%91%D7%99%D7%99%D7%A1_%D7%A0%D7%90%D7%99%D7%91%D7%99&action=edit&redlink=1) (Naive Bayes classifier) ו[רשת עצבית מלאכותית](http://he.wikipedia.org/wiki/%D7%A8%D7%A9%D7%AA_%D7%A2%D7%A6%D7%91%D7%99%D7%AA_%D7%9E%D7%9C%D7%90%D7%9B%D7%95%D7%AA%D7%99%D7%AA).
* **ניתוח אשכולות** - פילוח את הרשומות לאשכולות, כך שבכל אשכול נמצאות רשומות בעלות מאפיינים דומים. למשל ניתן להיעזר באלגוריתם לניתוח אשכולות כדי לפלח את לקוחות החברה.
* **רגרסיה** - ניסיון למצוא פונקציה הממדלת את הנתונים תוך מזעור שגיאת הניבוי. לדוגמה תוך שימוש ב[רגרסיה לינארית](http://he.wikipedia.org/wiki/%D7%A8%D7%92%D7%A8%D7%A1%D7%99%D7%94_%D7%9C%D7%99%D7%A0%D7%90%D7%A8%D7%99%D7%AA).
* **למידת חוקי אסוציאציה** - נועד למצוא אסוציאציה או קשר מעניין בין פריטים שונים של נתונים. לרוב משמש כדי להציג חוקיות בין שדות שונים בבסיס הנתונים. הדוגמה הנפוצה ביותר היא ניתוח סל קניות. במקרה זה מנסים למצוא האם יש חוקיות בין מוצרים שונים אשר נרכשו על ידי אותו לקוח באותה קניה. למשל ניתן לדעת אילו מוצרים נרכשים בדרך כלל יחדיו (ממרח שוקולד ולחם). בהתאם לתוצאות ניתן לסדר את המוצרים במרכול בצורה האופטימלית.

### אימות תוצאות השלב האחרון בגילוי ידע מנתונים הוא אימות של התבניות שנמצאו על ידי האלגוריתמים של כריית המידע גם בטווח נתונים רחב יותר. לא כל התבניות הנמצאות על ידי האלגוריתמים כורי המידע הן בהכרח תקפות. לעתים האלגוריתמים מוצאים תבניות בסדרת האימון שאינן מופיעות בסדרת המידע הכללית, תופעה הידועה כ[Overfitting](http://he.wikipedia.org/w/index.php?title=Overfitting&action=edit&redlink=1)- . על מנת להתגבר על בעיה זו, בשלב ההערכה נבדקים הדפוסים מול סדרת הבדיקה שעליה לא אומנו האלגוריתמים. התבניות שנלמדו מחולצות מסדרת הבדיקה והפלט מושווה לפלט הרצוי.

**ההבדל בין סטטיסטיקה לכריית נתונים:**  
בסטטיסטיקה מסורתית התהליך הוא: השערה, איסוף נתונים, ניתוח מודל מונחה, התמקדות בהבנה.  
כריית נתונים(Machine Learning) התהליך הוא: יש כמה השערות, הנתונים נאספים בדרך כלל בהתחלה, הניתוח הוא בדרך כלל מונחה-נתונים ולא השערה. לעיתים קרובות מונחה אלגוריתם ולא מונחה מודל, התמקדות בניבוי.   
רעיונות סטטיסטיים מאוד שימושיים בכריית נתונים, למשל ניתן לבצע אימות האם הידע שהתגלה הוא שימושי.

כריית נתונים עוסקת בתחומים שונים: טכנולוגיית מסד נתונים, מכונת למידה, לימודי מידע, סטטיסטיקה, ראייה.

**סוגים של נתונים: קובץ שטוח או נתוני וקטור**



שורות = אובייקטים, תצפיות, tuples.  
עמודות = מדידות על אובייקטים  
מייצג כל שורה כווקטור p-ממדי, שבו p היא ממדית  
גם P וגם N יכולים להיות גדולים מאוד בכריית נתונים   
מטריצה יכולה להיות די דלילה.

**מודלים שונים:**

המטרה היא לבנות מודל מתאים שיכול לדמות את הנתונים במידת הצורך.

**Predictive Modeling:**

במודל חיזוי מתבצע שימוש בשיטות סטטיסטיות בשביל לחזות משתנים בעתיד. ניתן לבצע חיזוי על כל סוג אירוע, ללא קשר מתי בוצע האירוע. יש ניבוי להתנהגות בעתיד. לדוגמא, משתמשים במודל זה פעמים רבות כדי לזהות פשעים וחשודים לאחר שהפשע התרחש.   
משמש במערכות דואר אלקטרוני לזיהוי האם ההודעה היא דואר זבל או לא.

חיזוי משתנה X באמצעות Y. X יכול להיות כאן וקטור P-ממדי.  
סיווג: Y הוא קטגורי  
רגרסיה: Y הוא ערך אמיתי.  
ניתן ללמוד את הקשר בין X ל-Y. בכריית נתונים הדגש הוא על דיוק הניבוי ולא על הבנת המודל.

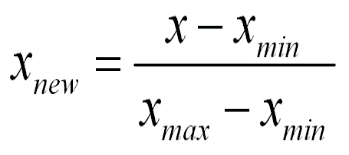
המטרה היא גילוי של דפוס התנהגות דרך הנתונים.

**Exploratory Data Analysis- ניתוח נתונים:**

נקרא גם בשם EDA. גישה לניתוח נתונים ע"י סיכום המאפיינים העיקריים שמשתמשת בשיטות חזויות(לרוב שיטות גרפיות). המטרות העיקריות הן:   
-מקסום של הבנה מערכי הנתונים  
-חשיפת מבנה בסיסי  
-חילוץ משתנים חשובים  
-זיהוי חריגים וחריגות  
-בדיקת הנחות יסוד  
-פיתוח מודלים חסכנים  
-קביעת הגדרות אופטימליות.  
  
לרוב הנתונים שאנו מקבלים הם עם טעויות ולכן אנחנו משתמשים בשיטות סטטיסטיות לניתוח הנתונים. עושים חציון, ממוצע ופרמטרים סטטיסטיים.   
**Dataset**- אוסף של נתונים שמורכב מתצפיות. בכל תצפית יש משתנים(X1,X2), וסה"כ Xd משתנים.   
משתנה מטרה(לא חובה שיהיה)- משתנה שתלוי מהמשתנים האחרים. המשתנים האחרים בכל תבנית קובעים את Y, ו-Y הוא משתנה שניתן לחזות( לדוגמא, האם יהיה גשם מחר, כמה מילימטרים ירדו וכו'..).  
הדרך הנפוצה ביותר היא להגדיר את הנתונים בתוך טבלה או מטריצה שבו כל **עמודה** מייצגת משתנה מסוים, וכל שורה מציגה את חיבור הנתונים בשאלה של המשתנים. כל ערך הוא נתון, וניתן להכניס כמה ערכות נתונים אשר יתאימו למספר השורות. בודקים עבור כל משתנה מה הוא. כל תצפית זה X ו-X הוא בעל התפלגות המתאימה לטבלה( אסור שתהיה סטייה בין הנתונים בטבלה להתפלגות X). ניתן ללמוד האם ההתפלגות היא סימטרית או לא.  
**קורלציה-** מתאם או קורלציה הוא מדד סטטיסטי המעריך את העקביות ביחס בין מספר משתנים. האם ישנה עקביות בין שינוי במשתנה אחד לשינוי במשתנה השני. כאשר יש התאמה גבוהה ביחס בין המשתנים הדבר רומז על קשר של סיבתיות, אך לא בהכרח. אם יש התאמה המתאם יראה על 1, אם לא אז 1-  
קיימים מספר מדדי קשר, בהתאם ל[סולם המדידה](http://he.wikipedia.org/wiki/%D7%A1%D7%95%D7%9C%D7%9D_%D7%9E%D7%93%D7%99%D7%93%D7%94) של המשתנים הנמדדים. במקרה של אי התאמה בין הסולמות של שני משתנים, יש להשתמש במדד המתאים לסולם הנמוך מבין השניים.  
אם משתנה אחד יותר גדול, גם המשתנה השני יהיה יותר גדול. אם יש קשר בין המשתנים הנקודות יהיו סביב הקו ואם לא הן יהיו מפוזרות.

**Data Preparation- נתוני הכנה:**

הנתונים בעולם האמיתי הם מלוכלכים(עם טעויות) ולא תמיד מתאימים לנו, ולכן צריך לבצע ניתוח ולבדוק האם הם מתאימים לנו למשימה. מתבצע על בסיס חקירת הנתונים. לוקחים כמה נתונים, מחלקים אותם ובונים נתון יחיד. את הנתונים שלא משפיעים על התוצאה אנחנו מסננים. המטרה היא תמיד לנסות לנרמל את הנתונים. לאחר נרמול לא אמורים להיות נתונים עם N/A(נתונים לא זמינים).  
**נרמול**- החלפת נתונים באחת מ-2 האופציות הבאות:

*  נורמליזציה- כאשר כל X יהיה בין 0-1- אם X הוא מספר גדול וחורג נקבל שכל הנתונים הם קרובים וקטנים ולכן גם נקבל התפלגות אחידה. האופציה הזו היא פחות טובה לנו.
* – תקינה- אופציה טובה יותר ובהתפלגות נורמלית עדיף להשתמש בה.

**חוסר שלמות**: ערכים חסרי תכונות, או חסרי תכונות הנוגעות לנו, או שמכילים רק נתונים מצרפיים.   
**רועש:** מכיל שגיאות או נתונים חריגים.  
**עולה בקנה אחד**: מכיל סתירות או חריגות בקודים או בשמות.

אם אין נתונים באיכות, אין תוצאות כריית איכות. ההחלטות חייבות להיות מבוססות על נתונים איכותיים ולכן מחסן הנתונים צריך אינטגרציה בין נתונים בעקביות. כאשר יש נתונים איכותיים, התוצאה בנתונים אמינה ומשקפת.

**משימות בהכנה למשימה:**1. ניקוי נתונים- בדיקה עבור איכות הנתונים וניקוי נתונים חסרים.  
2. שינוי נתונים  
3. הפחתת נתונים- נותן רק התוצאות האנליטיות הזהות.   
4- צמצום והפיכת נתונים, חשוב במיוחד עבור נתונים מספריים.

**נתונים חסרים:**

לא תמיד כל הנתונים זמינים, יכול להיות ממצב שלא נרשמו נתונים בסקרים או שהנתונים נלקחו ממקורות שונים. יכול לנבוע בגלל: תקלה בציוד, הנתונים לא נכנסו כראוי, נתונים אינם זמינים, גרסאות שונות שמוזגו.

**איך מתמודדים עם זה?**

* **התעלמות-** ניתן לעשות את זה רק לאחוז קטן מהנתונים החסרים.
* **שימוש במספר קבוע גלובלי( למשל, משתנה ממוצע) למילוי ערך חסר-** נמצא בקטגוריה "לא ידוע", לנתונים רצופים זה יקטין משמעותי את השונות.
* **שימוש בערך אקראי למילוי ערך חסר-** שומרים את השונות, וכך אין נזק.
* **שימוש בזקיפה-** ניתן לקחת את השכן הקרוב ביותר או להשתמש במודל מבוסס (רגרסיה או בייס)

**נתונים רועשים:**

רעש מוגדר כשגיאה אקראית בנתונים, או שונות במשתנה הנמדד.  
**ערכים שגויים נוצרים בעקבות**: אוסף נתונים שגוי, הכנסת נתונים שגויה, טכנולוגיה מוגבלת.  
**עוד בעיות אשר דורשות ניקוי נתונים**: רשומות כפולות, נתונים חלקיים, נתונים שלא עולים בקנה אחד(לא מותאמים).

**שינוי נתונים:**

* **יכול להפחית את ההשפעה של ערכים קיצוניים**
* **הפחתת שונות:** שימושי כאשר מתעסקים עם נתונים משתנים (כמו הכנסות)**,** הופך הסתברויות מ-0 ל-1לאמיתיות, יש שימוש בשורש ריבועי, גומלים ואלגוריתם.
* **נורמליזציה:** כאשר אנחנו רוצים לקבל את כל המשתנים באותו קנה מידה, דקויות של מינימום-מקסימום, וצריך שיהיה תקין.
* **בניית תכונות ומאפיינים חדשים מהנתונים הקיימים**

**התמודדות עם נתונים מסיביים:**

אם הנתונים לא מתאימים לנו ניתן לבצע:

* מדגם- אך צריך להיזהר עם הנתונים האקראיים
* ניסוח שאלה מחדש- הפחתת בסיס הנתונים ע"י שימוש בכלים מתאימים וניסוח השאלה מחדש.
* שימוש במסד נתונים- לדוגמא, my sql
* חקירה אסטרטגית לצמצום הנתונים- יכול להפחית את P או N

**הפחתת נתונים:**

כרוך באובדן מידע של X. אם P יהיה מאוד גדול( למשל 1000), ההפחתה תוכל לסייע ללמידה אמינה יותר. לדוגמא בלימוד אשכולות: יש 100 נתונים ממדיים. אם האשכול קיים רק ב-2 מהממדים, כל האחרים הם רעש. אם ה-98 האחרים הם רעש, אז יהיה קל יותר לגלות אם אנחנו צריכים להתמקד רק ב-2 המרחבים האחרים. הפחתת ממד יכולה לספק גם פרשות או תובנה.

**בחירת הנתונים להפחתה תעשה ע"י:**

* שימוש ב-EDA למציאת משתנים חסרי תועלת
* שימוש בחיפוש במודל פשוט(למשל רגרסיה)- יכול להיות יקר
* שימוש בשיטות האוריסטיות כמו שיטות של בחירה בקדימה/אחורה- יכול להיתפס במינימום מקומי.

שימוש במדגם- בוחרים קבוצת של נתונים לצורך מדגם. ניתן לבחור קבוצה אקראית של נתונים אך צריך להיזהר מנתונים משתנים. ניתן לבחור שיטות דגימה מרובות שנותנות את הקירוב של כל תת אוכלוסייה באתר הכללי, זה פעול ביחד עם נתונים משתנים, ויכול לעזור כאשר יש חוסר איזון בין התגובות.

**Linear Regression- רגרסיה ליניארית**

רגרסיה ליניארית היא גישה למודלים של היחסים בין משתנה תלוי(Y) למשתנה בלתי תלוי(X). המקרה של משתנה אחד שתלוי במשתנה אחר נקרא רגרסיה ליניארית פשוטה, להבדיל מרגרסיה ליניארית מרובת משתנים שבו משתנה אחד תלוי במשתנים רבים ע"י קורלציה.   
אנו מניחים שניתן לחשב את Y מתוך X.

X0=1 תמיד, לצורך כתיבה מקוצרת.

יש לרגרסיה לינארית שימושים מעשיים רבים. רוב היישומים נופלים לתוך אחת משתי הקטגוריות הבאות:

* אם המטרה היא ניבוי, חיזוי, או הפחתה, רגרסיה ליניארית יכולה לשמש כדי להתאים מודל חיזוי לנתונים שנצפו מסט של ערכי *y וX.* לאחר פיתוח מודל כזה, אם ערך נוסף של *X* ניתן ללא ערך הנלווה של *y,* המודל יכול לשמש להכנת תחזית של הערך *y.*
* כאשר נותנים משתנה y ומספר משתנים   *X*1, ..., *Xp*שעשויים להיות קשורים *לy,* ניתוח רגרסיה ליניארית יכול לכמת את עוצמתה של מערכת היחסים בין *j y וX,* כדי להעריך אם אולי אין להם מערכת יחסים עם *y* בכל, וכדי לזהות איזה תת קבוצות של *j X* מכילות מידע מיותר על *y.*

**- Perceptron מודל לינארי:**

זהו אלגוריתם מיון שמבצע תחזיות שמבוססות על פונקציות ניבוי ליניאריות המשלבות משקלות עם וקטור. בעזרתו ניתן לחשב את Y ולהגיד אם Y כן או לא. משימה הזו נקראת classification. אלגוריתם זה נותן פתרון לדברים הבאים:

* משימה ליניארית נפרדת
* בעולם האמיתי- סוג כזה של משימות לא מעניינות ולכן יש לנו מודל מתמטי למקרה כזה.

הלמידה מתבצעת כאשר:

* יש דפוס קיים
* אנחנו לא יכולים להגדיר באופן מתמטי
* יש לנו מספיק נתונים על זה

**Learning Algorithm-למידה:**

אלגוריתם לשינוי משקלות כדי לקבל Ein מינימלי

* משנים את w כך Ein (סכום כל השגיאות) יהיה מינימלי.במודל ליניארי ניתן לחשב את הערך האופטימלי בצורה ישירה
* מכיוון שערכת הנתונים היא מדגם מכל הנתונים האפשריים שגיאה זו נקראת במדגם error Ein(h).
* כאשר אנו משתמשים בהשערות בנתונים אחרים אנחנו מגלים את השגיאה.
* היעד הוא להיות קטן מ- Eout(g) → 0

**שגיאות ברגרסיה לינארית:**

אנחנו משתמשים בשגיאה בריבוע: (h(x)-f(x)) 2 . מחפשים קו הפרדה שיביא לכמות שגיאות מינימלית.h(Xi) - אם שני המשתנים קטנים מ-0 או גדולים מ-0 אז התוצאה היא 0. אם המשתנים שונים(אחד גדול מ-0 והשני קטן מ-0) אז התוצאה היא 1 וזה נחשב שגיאה.

Yi- אם הוא אותו דבר כמו ה-Ei הראשון אז התוצאה היא 0. אם לא, התוצאה היא 1. כל נקודה זה עמודה.

**מה זה ליניארי?**

* מכשירי לרגרסיה ליניארית
* כלי לסיווג ליניארי
* האלגוריתם עובד בגלל הליניאריות במשקלות

רגרסיה לינארית למען ניבוי:

1. יש לך אוסף של נתונים
2. מכניסים קו שהוא מצמצם איזשהו מדד של טעות
3. אם הצלחנו, זה כלי טוב לניבוי

**Machine learning- למידה חישובית**

מטרת הלמידה יכולה להיות [מידול](http://he.wikipedia.org/w/index.php?title=%D7%9E%D7%99%D7%93%D7%95%D7%9C&action=edit&redlink=1), [חיזוי](http://he.wikipedia.org/wiki/%D7%97%D7%99%D7%96%D7%95%D7%99) או גילוי של עובדות לגבי העולם האמיתי. לדוגמה: עבור [זיהוי תווים אופטי](http://he.wikipedia.org/wiki/%D7%96%D7%99%D7%94%D7%95%D7%99_%D7%AA%D7%95%D7%95%D7%99%D7%9D_%D7%90%D7%95%D7%A4%D7%98%D7%99) ניתן להשתמש בלמידה חישובית כדי לגלות מהי האות הכתובה או המודפסת. [מערכות זיהוי דיבור](http://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%96%D7%99%D7%94%D7%95%D7%99_%D7%93%D7%99%D7%91%D7%95%D7%A8) יכולות גם הן להשתמש בלמידה חישובית כדי ללמוד, בהינתן אותות קוליים כלשהם, מהי ה[הברה](http://he.wikipedia.org/wiki/%D7%94%D7%91%D7%A8%D7%94) שיצרה אותם.  
קיימים משתני X ולכן X קיים Y.

**Error measures- צעדי שגיאה:**

קיימות 2 הנחות:

1. נלמד את Y באמצעות X
2. בתוך קבוצת פונקציות יש פונקציות שנותנות לחשב את ה-Y בצורה איכותית.

אם 2 ההנחות נכונות, ניתן למצוא את X.

**Eout –** שגיאה בנקודה שעדיין לא נמצאת בתוך ערכת הנתונים. לא ניתן לדעת מהו ה- Eout. ככל שיש יותר נקודות והמודל יותר מורכב כך גם n יותר גדול. כמות הנקודות צריכה להיות גדולה משמעותית מכמות הפרמטרים. המשמעות היא ש- Ein יהיה בסדר גודל של Eout. אם אין לנו הרבה נקודות ונוסיף עוד פרמטרים Ein יהיה שונה מ- Eout.

**Ein-** שגיאה בנקודה שנמצאת בתוך ה-dataset (ערכת הנתונים).

כאשר מספר הנקודות יהיה גדול פי 10 ממספר הפרמטרים-Ein יהיה שווה ל-Eout ואז נוכל לקבל הערכה טובה. לכן, חשוב מאוד לעשות מודל פשוט ולא מורכב מדי.

**Dvc-** מספר הפרמטרים (נכון לכל מודל)

**ציון מידת השגיאה:**

* מידת השגיאה צריכה להיות מוגדרת ע"י המשתמש.
* לא תמיד אפשרי- יש חלופות אפשריות כמו square error

**Generalization- הכללה:**

המטרה הבסיסית של מכונה לומדת היא היכולת להכליל מתוך הניסיון. כלומר היכולת לבצע (חיזוי, סיווג, רגרסיה וכו') באופן מדויק ככל האפשר על מידע (דוגמאות\משימות) שעדיין לא נצפה, על בסיס צבירת ניסיון ממידע שכבר קיים. בדרך כלל דוגמאות המשמשות ללימוד נוצרות מאיזושהי התפלגות לא ידועה והמכונה הלומדת בונה מודל מוכלל מאותו מרחב של ההתפלגות, מה שמאפשר ביצוע מדויק באופן מספק על דוגמאות חדשות.

**כלל ההכללה:**

* N צריך להיות משמעותית גדול ממספר הפרמטרים.
* **כלל אצבע: N >= 10Dvc**
* בשביל מודל מורכב צריכות להיות מספיק נקודות.

**ממדVC:**   
מדד בתחום [הלמידה החישובית](http://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%94_%D7%97%D7%99%D7%A9%D7%95%D7%91%D7%99%D7%AA) המתאר את רמת כושר ההפרדה של סיווג הנוצר על ידי [אלגוריתם למידה](http://he.wikipedia.org/w/index.php?title=%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D_%D7%9C%D7%9E%D7%99%D7%93%D7%94&action=edit&redlink=1). ממד VC של מסווג כלשהו C מוגדר כגודל הקבוצה הגדולה ביותר של דוגמאות למידה ש C יכול לנפץ (shatter). ניפוץ, או shattering, לצורך הגדרה זו מתקיים כאשר בהינתן קבוצה H המכילה דוגמאות למידה, לכל חלוקה אפשרית שלה לשתי מחלקות סוג, קיים פרמטר α עבור מסווג C כך ש C(α) מסוגל להפריד בהתאם את הדוגמאות שב- h. יש לשים לב שעל פי ההגדרות הנ"ל מתקיים שמימד VC של מסווג C הוא לפחות n אם קיימת קבוצת דוגמאות H כלשהי בגודל n כך שלכל חלוקה אפשרית שלה לשני סוגי דוגמאות, מסווג C מסוגל להפריד נכונה את הדוגמאות שב-H בהתאם.

**Bias and Variance- הטיה ושונות:**

קיימות 2 סיבות לשגיאה:

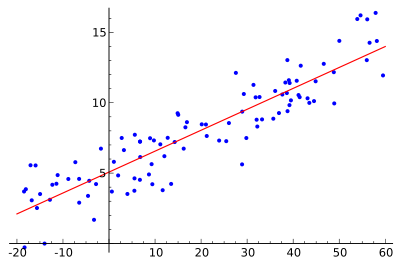
1. **Bias**- שגיאה שנובעת מבחירה של מודל פשוט ולא נכון ומהנחות שגויות בלמידת האלגוריתם. גם במקרה N גדול מאוד תהיה שגיאה. שגיאה זו נקראת הטיה של המודל. הטיה יכולה להיות 0 רק אם F נמצא ב-H.
2. **Variance**- שגיאה אשר נובעת מכך שהנקודות לא מתאימות למודל האמיתי. קיימת רגישות לתנודות ולרעש ולכן משגיאה זו לא ניתן להתעלם.

ככל שיש יותר נקודות כך יהיה ניתן להתאים טוב יותר את המודל.

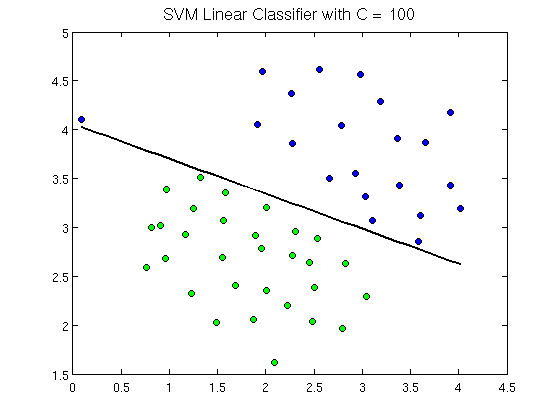
**Linear Models- מודלים ליניאריים**

המודלים נחשבים ליניאריים לפי W. ניתן לעלות את כל ה-X-ים בחזקות ועדיין המודל יישאר ליניארי. ניתן גם להחליף את X ב-Z (קומבינציה של נתונים לא ליניאריים).

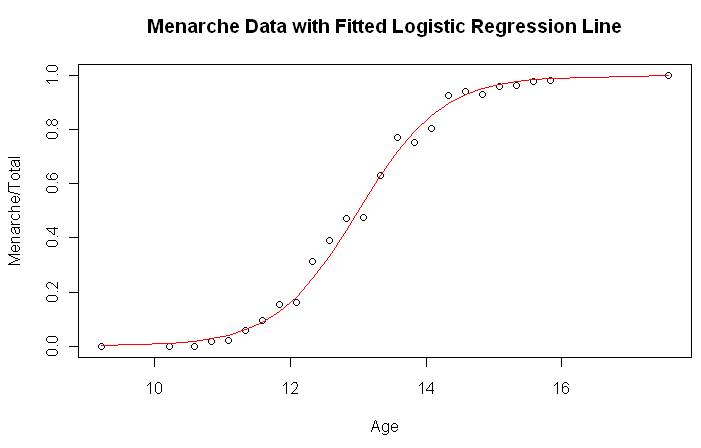
**Linear Regression- רגרסיה ליניארית:** רגרסיה ליניארית היא גישה למודלים של היחסים בין משתנה תלוי(Y) למשתנה בלתי תלוי(X). המקרה של משתנה אחד שתלוי במשתנה אחר נקרא רגרסיה ליניארית פשוטה, להבדיל מרגרסיה ליניארית מרובת משתנים שבו משתנה אחד תלוי במשתנים רבים ע"י קורלציה.   
אנו מניחים שניתן לחשב את Y מתוך X.



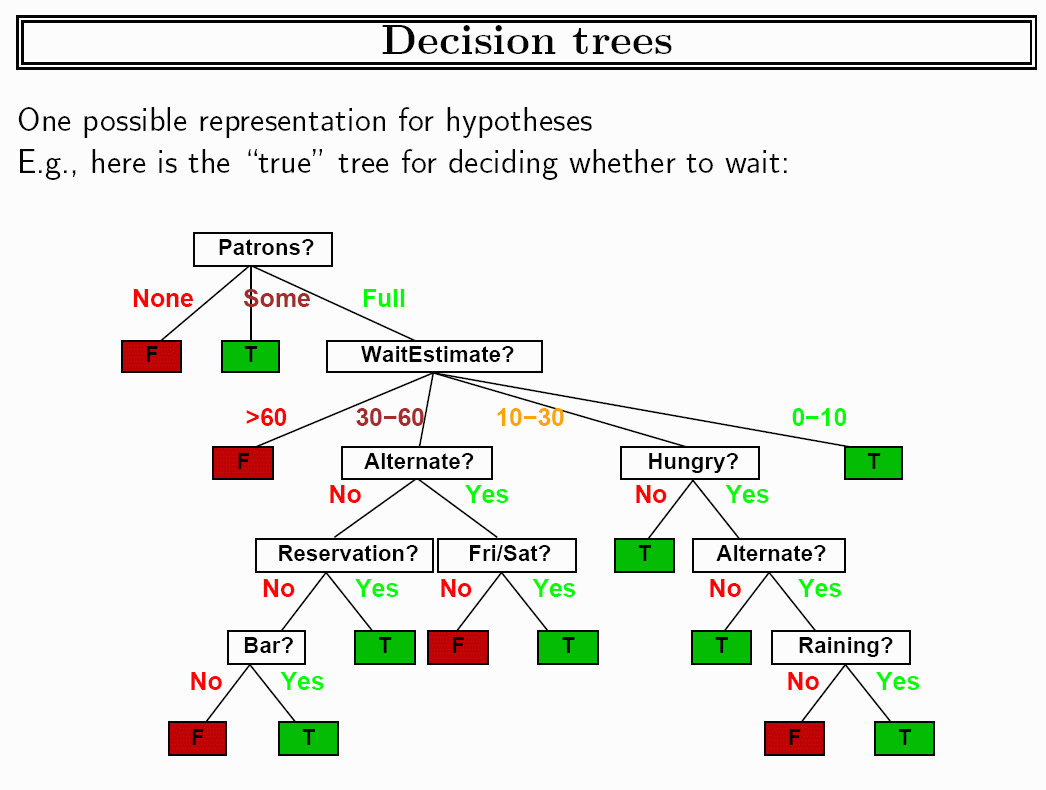
**Classification- סיווג:** אלגוריתם מיון שמבצע תחזיות שמבוססות על פונקציות ניבוי ליניאריות המשלבות משקלות עם וקטורים. בעזרתו עושים סיווג ומחשבים את Y , ואז ניתן להגיד כן או לא.



**Logistic Regression– רגרסיה לוגיסטית:** רגרסיה לוגיסטית אומדת את ההסתברות לאירוע שהתרחש. אנחנו רוצים לחזות מידע של משתנים בלתי תלויים רלוונטיים באזור קו ההפרדה שם ניתן להגיד אם כן או לא, ולא את הערך המספרי מדויק של משתנה תלוי. ההסתברות (p)  שזה 1 (אירוע שהתרחש) ולא 0 (אירוע לא מתרחש).



**Classification (Decision) Trees- עצי החלטה**



עצים הם אחד המודלים הכי פופולאריים והשימושיים בכריית נתונים. ישנו אלגוריתם המבוסס על סיווג.   
**יתרונות המודל:**

* פשוט לבנייה והסברה
* יכולת התמודדות טובה עם ערכים ריאליים
* מהיר
* יכולת התמודדות עם נתונים חריגים או נתונים חסרים
* מבוסס על יכולות סיווג ולכן תמיד בסוף מגיעים להחלטה
* אין הנחות בחלוקה
* דורש הכנת נתונים קטנה, טכניקות אחרות לעתים קרובות דורשות נרמול לנתונים.

**חסרונות המודל:**

* גבול ההחלטה הוא לא רציף
* לא תמיד ניתן לבנות עץ יעיל, זה תלוי איך התחלנו את העץ. אם הפרמטר בהתחלה שונה, הנתונים גם יצאו שונים.
* יש מספר כיוונים לפרמטרים שונים עם הדרכה קטנה

**הערה**: גבולות העץ הם פונקציה ליניארית וציר מקביל.

בתחילה בוחרים מי המשתנה הטוב ביותר וכך ממשיכים להתקדם. הבחירה נעשית בכדי ליצור הפרדה בין הנתונים. פיצול טוב נקבע ע"י "טוהר" העלים(איכות הנתונים). ניתן לבנות יער ע"י חלוקת הנתונים לעצים שונים. לאחר מכן עוברים על כל עץ ומחליטים לפי הרוב שהעצים הצביעו. אם 9 מתוך 10 עצים הצביעו "כן", אנחנו נחליט "כן". אם החלוקה שווה( 5 כן ו- 5 לא) נגיד שלא ניתן לקבוע ע"פ הנתונים.

**אינדוקציית עץ החלטה:**

אלגוריתם בסיסי:

* העץ בנוי בצורה רקורסיבית מלמעלה למטה.
* בתחילה כל הדוגמאות נמצאות בשורש.
* הדוגמאות מחולקות למחיצות רקורסיביות כדי למקסם את ההצלחה.
* הערכה של כל הפיצולים האפשריים. החלטת הפיצול הטובה ביותר היא לפצל את הנתונים עם המידע הגדול ביותר כדי לחלק את הנתונים לתתי קטגוריות, והשיטה נעשית באופן רקורסיבי.

תנאים להפסקת מחיצות:

* כל הדוגמאות שייכות לאותו המעמד
* העלה קטן מהסף שנקבע
* יש סחר בין מורכבות והכללה

**Regression Trees- עצי רגרסיה:**

כאשר התגובה היא ערך אמיתי, ניתן להשתמש בעצים גם לרגרסיה. תחזית של עלה היא ערך ממוצע במקום הערכת הסתברות. ניתן למשתמש בשונות כאמצעי החלטה ומסייע לניבוי קטגוריות. עצים נמצאים היום בשימוש רחב מכיוון:

* ניתן לטפל בממדים גבוהים
* בונה מודל בשימוש ממד 1 בזמן
* ניתן לטפל בכל סוגי המשתנים
* מנבאים קטגוריות
* התמורות קבועות למשתני הקלט
* ניתן לפרש עצים
* מומחה יכול להגדיר את הגיון העצים ככללים
* אלגוריתם העצים קל יחסית לכתיבה ובדיקה

**מגבלות של עצים:**

* קושי במבנה של מודל ליניארי
* מחסור בהחלטות חלקות
* שונות גבוהה
  + עצים יכולים להיות "לא יציבים" כפונקציה של המדגם. לדוגמא, שינוי קטן בנתונים מביא לעץ שונה לחלוטין ודבר זה גורם ל-2 בעיות:

1. שונות גבוהה תורמת לשגיאות בניבוי
2. שונות גבוהה מפחיתה פרשנות

* טובים למודלים משולבים
* משמש לעיתים קרובות לחיזוק החלטה

**Neural Networks- רשתות עצביות מלאכותיות**

**Neural network model- מודל רשת עצבי מלאכותי:**

משפחה של אלגוריתמי למידה סטטיסטיים בהשראת [רשתות עצביות ביולוגיות](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Biological_neural_network&usg=ALkJrhhEEdMqK8myQo4GYghFVP9jC7gDRg) [(מערכות העצבים המרכזיות](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Central_nervous_system&usg=ALkJrhgqmn_IgOnnuJTut-yPUKjAMtIdlw) של בעלי חיים, במיוחד במוח) המשמשות להערכה או אמידת [פונקציות](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Function_(mathematics)&usg=ALkJrhh7uluxdbGnor2mWVGkmaDo5LXu7A) שתלויות במספר גדול של [קלטים](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Argument_of_a_function&usg=ALkJrhj2pdDMdZ0W4j_DX2lxUzCgdVq-VA) שבדרך כלל לא ידועים. רשתות עצביות מלאכותיות בדרך כלל מוצגות כמערכות של "מקושרים [נוירונים"](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Artificial_neuron&usg=ALkJrhiG9DQIUio4cUdyV04t0_1j9jpCmA) אשר יכול לחשב ערכים מקלטים, ומסוגלים [ללמוד](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Machine_learning&usg=ALkJrhiAEUyHXm7FoDnBEuk7woQbAKBRdA) מכונה, כמו גם [זיהוי תבניות](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Pattern_recognition&usg=ALkJrhgKHN6yRjs2GpO3TgSYj86WtCWXpw) הודות לטבע ההסתגלות שלהם.

לדוגמא, רשת עצבית [לזיהוי כתב יד](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Handwriting_recognition&usg=ALkJrhh4sVycmo351C_tw0Te_T2ffChrHQ) המוגדרת על ידי קבוצה של תאי עצב וקלט שעלול להיות מופעלים על ידי פיקסלים של תמונת קלט. לאחר ששוקלל והשתנה על-ידי [פונקציה](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Function&usg=ALkJrhj8zlI0HqOVhZ7tfTpohSSskPNPoA) (שנקבעה על ידי המעצב של הרשת), ההפעלות של נוירונים אלה לאחר מכן מועברים לתאי עצב אחרים. תהליך זה חוזר על עצמו, עד שלבסוף, נוירון פלט מופעל. זה קובע איזה מאפיין לקרוא.

כמו שיטות למידת מכונה אחרות - מערכות שלומדות מנתונים - רשתות עצביות כבר בשימוש כדי לפתור מגוון רחב של משימות שקשה לפתור באמצעות [תכנות רגיל המבוסס על כללים,](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Rule-based_programming&usg=ALkJrhhfnVd-NjwFyd4ib8EsPN8yB6ExBg) הכוללים [ראייה ממוחשבת](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Computer_vision&usg=ALkJrhhDH9MJSQQRg07llCneCah7I9ZT_A) [וזיהוי דיבור.](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Speech_recognition&usg=ALkJrhjQE6n3Rna2GPe_dxoVtI0vfS4iUw)

הקלט מתקבל מהמודל הקודם. יש סדרת נקודות חישוב שכולן הן רגרסיה ליניארית. **הבעיה** במודל זה הוא חישוב המשקלים. מכיוון שיש המון נקודות, המשקלים אקראיים ולכן רוב הסיכויים ש-Y לא יהיה אמיתי. לוקחים את ההפרש שמקבלים ומנסים להקטין אותו.

**חסרונות:**

* צריך המון נקודות. אם אין, אין מה להשתמש במודל זה.
* לא מתרגם מספיק טוב משפה לשפה
* לא מזהה מספיק טוב תמונות

**יתרון:** מודל מאוד פשוט

**Stochastic gradient descent-SGD:**

היתרונות של SGD:

* חישוב זול
* עזרה רנדומלית
* פשוט- מיושם עבור טכניקות שונים

**Back Propagation- רביה חוזרת:**

שיטה נפוצה של אימון [רשתות עצביות מלאכותיות](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Artificial_neural_networks&usg=ALkJrhhOgnri6kXpDwYRC3CQ0p3nnqQkGg) הפועלות בשיתוף עם [שיטת אופטימיזציה](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Mathematical_optimization&usg=ALkJrhgWNUnMxPkcrRHjzs4EVMMmITvyjg) כגון [מציאת שיפוע.](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Gradient_descent&usg=ALkJrhgNEBYy4tgGctBxhoYxZkGrBFr7Fw) השיטה מחשבת את השיפוע של [פונקציית הפסד](http://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.co.il&sl=en&u=http://en.wikipedia.org/wiki/Loss_function&usg=ALkJrhikwo5r_5SIRtHb46hUzn7hyrVRhA) עם כבוד לכל המשקולות ברשת. השיפוע מוזן לשיטת אופטימיזציה אשר בתורו משתמשת בו כדי לעדכן את המשקולות, בניסיון למזער את פונקציית ההפסד.

**יתרונות:**

* רשת פרספטרון רב שכבתית יכולה להיות מאומנת ע"י אלגוריתם הרבייה כדי לבצע מיפוי בין הקלט והפלט.

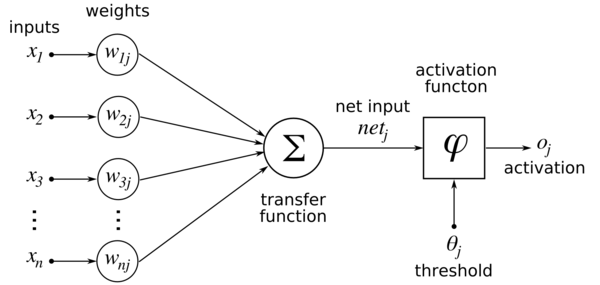
**חסרונות:**

* דורש הרבה הכנות לנתונים- כמעט כל הנתונים לא מתויגים.
* זמן למידה ארוך- מאוד איטי ברשתות עם מספר רב של שכבות נסתרות
* יכול להיתקע במקומות קטנים

**Deep Neural Networks- רשתות עצביות עמוקות:**

רשתות עצביות ולמידה עמוקה מספקות את הפתרונות הטובים ביותר כיום לבעיות רבות בזיהוי תמונה, זיהוי דיבור, ועיבוד שפה טבעי.

אסטרטגיית למידה סטנדרטית: מאתחל באקראי את המשקלות של הרשת, החלה ירידה הדרגתית בשימוש ברביה חוזרת.  
שיטת הרבייה לא עובדת טוב אם יש אתחול באופן אקראי.



**Overfitting. Regularization. Validation- הסדרה, אימות:**

אם ניקח יותר מדי נקודות ומעט מדי פרמטרים, ההתאמה מתחילה להיות לרעשים ולא לנקודות. צריך לבצע התאמה אידיאלית לכל נקודה.

**Overfitting-** התאמה לנקודה ספציפית שלא מקטינה את Eout. קורה כאשר המודל לומד לתאר רעש בנוסף לתלות האמיתית בין הקלט לפלט. במונחים של הטיה ושונות, למודל מורכב שיכול להשיג שגיאה קטנה מאוד בכל מערך נתונים(יש לו הטיה נמוכה לכל בעיה) יכולה להיות שגיאה גדולה בדוגמאות חדשות(ניסויים חדשים) כי השונות(תלויה בתצפית במערך הנתונים) גבוהה מדי. **Overfitting** הוא מזיק. 2 דרכים להתמודדות עם זה:

1. **Regularization- הסדרה:**

* שיטה להימנעות מ- Overfitting ע"י מניעת מזעור של Ein. מוסיפים ל-Ein איבר שלא נותן לעשות התאמה לנקודות. לא נותן ל-Ein להיות 0, הוא משבש את ההתאמה.
* זה יכול להתבצע ע"י עצירה מוקדמת של התהליך הרקורסיבי לחישוב W ע"י מזעור Ein
* שיטה יעילה נוספת של הסדרה היא פונקציית שגיאה משלימה עם טווח קטן הנקראת הסדרה. בד"כ היא מתאימה ל-2w||

1. **Validation set- אימות:**

* לוקחים חלק מהנתונים ומשתמשים בהם לבדיקה. נוצרים 3 סטים שניתן להעריך באמצעותם את Eout. בוחרים את המודל הטוב ביותר מבין הסטים.
* אמידת ההטיה:
  + סט הדרכה- מזוהם לחלוטין
  + סט אימות- מזוהם מעט
  + סט בדיקה- נקי לחלוטין

**Clustering and Unsupervised Learning- למידה ללא פיקוח והשגחה**

**Clustering-** זיהוי אוטומטי של מבנה קבוצת נתונים.

* בדרך כלל נעשה ע"י חלוקה לאשכולות כך שהנקודות בכל קבוצה דומות יותר לאחד השני מאשר לנקודות בקבוצה אחרת.
* מתבצע שימוש בטכניקת תיאורים(בניגוד לחיזוי).
* זיהוי קבוצות "טבעיות" של אובייקטי נתונים- תיאור קבוצות איכותיות בנתונים.
* שימושי לעיתים קרובות.
* לערכים אמיתיים של וקטורים האשכולות יכולים להיחשב כעננים של נקודות במרחב P-ממדי.

**מהו ניתוח אשכולות?**

* תוצאות ניתוח טובות מתבטאות בדמיון לנקודות הקרובות באותו אשכול, ושוני מהאובייקטים באשכולות אחרים. במילים אחרות, הדמיון בתוך האשכול גבוה(שונות תוך-אשכול נמוכה) והדמיון בין אשכולות נמוך(שונות בין-אשכול גבוהה).

**שימוש באשכולות עוזר ב:**

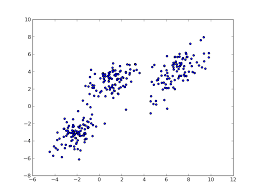
* גילוי של ידע חדש מהנתונים
* ניגוד עם סיווג לא מפוקח(שהתוויות ידועות בו).
* שימוש מאוד לסיכום ערכות נתונים גדולות
* ל-N גדול או גבוה ממדית
* יישומים של אשכולות
* מסמכים שמיוצרים ע"י מנועי חיפוש
* פילוח לקוחות
* ניתוח נתונים מרחביים
* אשכולות גיאוגרפיים של אירועים: שיעורי סרטן, מכירות וכ'ו
* אשכולות של גנים עם פרופילים דומים

**מדדי מרחק:** כדי לעשות אשכולות אנחנו זקוקים ל"סוג" של מרחק בין הנקודות. לעיתים המרחקים אינם ברורים, אבל אנחנו יכולים ליצור אותם.

**שיטות אשכולות:**

* עובד על מרחקים
* מטריצת NXN ל מרחקים
* היררכי- כל נקודה באשכול שלה, כל הנתונים באשכול אחד, איטרטיבי לניתוח.
* יש שני סוגות אשכולות עיקריים:
  + לחלק את הסט למחיצות של אשכולות עם גבולות מוגדרים
  + להציב כל נקודה באשכול הקרוב ביותר שלה.

**: K-means**



* בתחילה בוחרים K, שהוא מספר האשכולות.
* מאתחלים את האשכולות ע"י בחירת נקודה אחת לכל אשכול.
  + - בדרך כלל K הן נקודות אקראיות.

**אלגוריתם K-means:**

1. בחירה שרירותית של אובייקט K מהנתונים (לדוגמא, לקוח K) להיות מרכז האשכול
2. לכל אחד מהאובייקטים שנותרו: הקצאת כל אובייקט לאשכול שהמרכז בו הכי קרוב אליו.
3. חזור על 3 הצעדים עד שהאשכולות הבאים מתכנסים (ללא שינוי באשכולות)
4. חיבור של המרכז החדש של האשכולות הנוכחיים

**יתרונות:**

* יעיל יחסית: קל ליישום- לעיתים מגיע לפתרונות הטובים ביותר.
* אינטואיטיבי

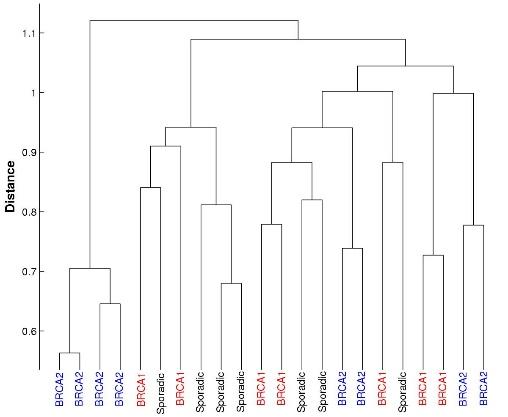
**חסרונות:**

* צריך לבחור את K האשכולות מראש
* אין אפשרות לטיפול בנתונים חריגים רועשים
* לא מתאים לגילוי אשכולות עם צורות לא קמורות
* די רגיש לנקודת התחלה ראשונית

**וריאציות שונות של K-means:**

* ניתן לשלב בין K-means ל- K-Medoids וכך לעשות את הניתוח טוב וחזק יותר.
  + K-Medoids- המיקום המרכזי ביותר באשכול
* הפוך את האתחול לטוב יותר
  + לקיחת מדגם אקראי קטן ואשכול כדי למצוא נקודת התחלה
  + אסיפת נקודות K ברשת
  + לבצע כמה ריצות עם נקודות התחלה שונות

**Hierarchical clustering- היררכית אשכולות:**



אינו מחייב שמספר האשכולות יהיו כקלט. ישנן 2 קצוות:

* כל הנתונים באשכול אחד
* כל נקודת נתונים באשכול שלה
* ייצוג: עץ של אשכולות מקוננים
* **אלגוריתם חמדן:** 
  + מצא את 2 הנקודות הקרובות ביותר
  + מחברים ביניהן
  + חוזרים על אותה פעולה
  + בין הנקודות הקרובות מחברים במרכז וככה נבנה העץ
* ניתן גם להריץ אחורה
* הדמיה אפקטיבית באמצעות " "dendrograms
  + מראה מבנה מקונן
  + מיזוגים ופיצולים בצמתי העץ. ניתן לחלק ולחתוך.

האלגוריתם דורש מדידת מרחק למרחק בין אשכולות, או בין נקודה ואשכול.

**מרחק בין אשכולות:**

* **קישור יחיד:**
  + המרחק הקטן ביותר בין נקודות
  + השכן הקרוב ביותר
  + יכולה להיות רגישות לחריגים
* **קישור מלא:**
  + המרחק הגדול ביותר בין נקודות
  + אוכף קומפקטיות
* **קישור ממוצע:**
  + ממוצע- מקבל ממוצע של התנהגות
  + נקודת המרכז- חזקה יותר

**יתרונות:**

* לא צריך לציין K אשכולות מראש
* ייצוג חזותי של אשכולות שונים
* מאפיינים מdendrogram
* ניתוח יותר איכותי ויפה להבנה של אדם

**חסרונות:**

* אפשרויות חיבורים שונים מקבלות תוצאות שונות מאוד

**לסיכום**- אם קשה לבחור את K, משתמשים בהיררכיה.   
אם לא קשה לבחור את K- משתמשים ב-K-means