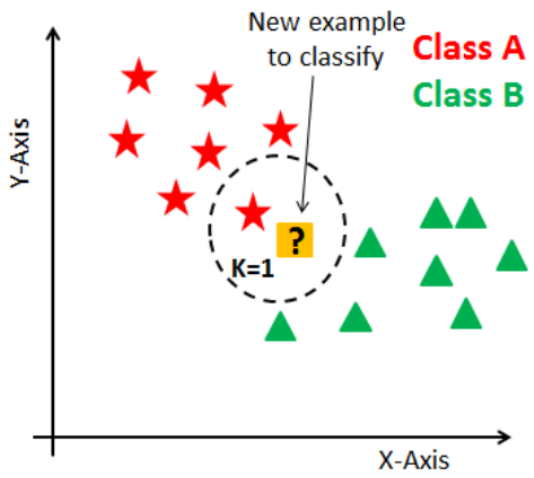
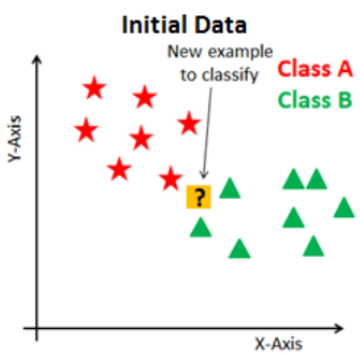
# אבן דרך 1 - מסווג KNN
## תאריך הגשה: 9/8/21

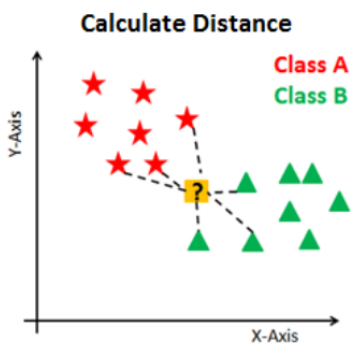
בתרגיל זה תממשו את האלגוריתם **K-nearest-neighbors** בשפת C++.
KNN הוא אלגוריתם פשוט מאוד, קל להבנה, תכליתי ואחד האלגוריתמים הנפוצים ביותר של למידת מכונה.
באלגוריתם נעשה שימוש במגוון יישומים כגון פיננסים, בריאות, מדע המדינה, זיהוי כתבי יד, זיהוי תמונות וזיהוי וידאו. האלגוריתם KNN משמש הן לבעיות סיווג והן לנסיגה.

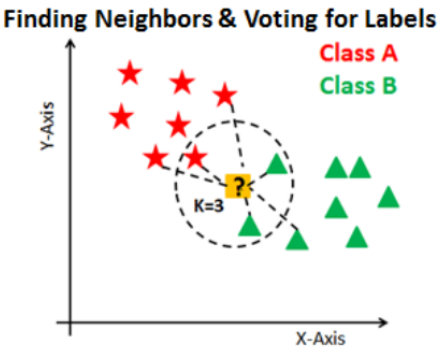
באלגוריתם זה K הוא מספר השכנים הקרובים. לרוב נבחר את K להיות מספר אי-זוגי. עבור K=1 נקבל את המקרה הפשוט ביותר - אלגוריתם השכן הקרוב ביותר. נניח כי P היא נקודה במרחב הדאטא שלנו עבורה נרצה להעריך את המחלקה אליה היא משתייכת. תחילה נמצא את השכן הקרוב ביותר לנקודה P ונסווג אותה לפי המחלקה אליה משתייך השכן. כפי שניתן לראות באיור:




באופן דומה עבור K שכנים. תחילה נמצא את K השכנים הקרובים ביותר לנקודה P. לאחר מכן נסווג את P על פי המחלקה אליה משתייכים המספר הגדול ביותר של שכנים מבין הK השכנים שמצאנו.




על מנת לקבוע את מי הם השכנים הקרובים ביותר נשתמש במטריקת מרחק כלשהי. מטריקה זו יכולה להיות מרחק אוקלידי, האמינג, מנהטן, מינקובסקי ועוד. בתרגיל זה תצטרכו לממש עבור מרחק אוקלידי. כלומר בהינתן שתי נקודות במרחב בעלי הקואורדינטות $(x_1, y_1), (x_2, y_2)$ המרחק האוקלידי ביניהן יהיה:

$$D_{EUC} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

לסיכום - שלבי האלגוריתם בהינתן נקודה לסיווג ודאטא עבור נקודות מסווגות:

1. לחשב מרחקים לנקודות המודל
2. למצוא K שכנים קרובים ביותר
3. לסווג את הנקודה לפי הסיווג של רוב השכנים

בתרגיל זה תצטרכו לבנות מסווג KNN עם מרחק אוקלידי. מצורפים 2 קבצי דאטא בפורמט CSV.

**Classified.csv -** דאטאסט של אירוסים מ3 מינים שונים: setosa, virginica, versicolor. לכל אירוס אספו 4 מאפיינים: רוחב / אורך עלי גביע ורוחב / אורך עלי כותרת. במאגר זה כל האירוסים מסווגים לפי המינים הנכונים.

**Unclassified.csv -** דאטאסט של אירוסים לא מסווגים. לכל אירוס נתונים ארבעת המאפיינים הנ"ל.

על המסווג להחזיר קובץ CSV בשם output.csv עם רשימת מיני האירוסים לפי הסדר המקורי בו הופיעו בדאטאסט Unclassified.csv .

## דרישות התרגיל

על התרגיל להתקמפל ולרוץ על שרתי האוניברסיטה ( U2 / Planet). אין להשתמש בספריות אלגוריתמים מוכנים אלא לממש בעצמכם את האלגוריתם והפונקציות שלו. יש להשתמש בפונקציונליות הקיימת ב C++ (לדוגמא namespaces) בעת כתיבת הקוד. יש לממש במבנה קוד תקין כלומר הצהרות בקובץ .hpp בנפרד מהמימוש בקבצי .cpp .

יש להגיש את התרגיל בזוגות.
כחלק מדרישות התרגיל עליכם לעבוד בגיט.
את התרגיל יש להגיש על ידי הגשת קובץ טקסט פשוט במודל שמכיל את השמות ומספרי תעודות הזהות שלכם ולינק לגיט המכיל את התרגיל שלכם.

על הגיט להכיל קובץ README שמסביר איך להריץ את המסווג וכן מסביר את המימוש שלכם ואילו אופטימיזציות הכנסתם אם בכלל. חשוב שתהיה דוקומנטציה מלווה גם בקוד עצמו.

*** לא חובה אך מומלץ ויקצר לכם את זמן הכתיבה של אבן דרך 2: שמרו על עיקרון Open / Close - שמשמעותו העיצוב שלנו צריך להיות פתוח להרחבה אך סגור לשינויים. נבין מתוך דוגמה. אם נממש את המסווג לעיל במחלקה כלשהי, מן הסתם יהיו לו מתודות ספציפיות המתאימות לדרישות האלגוריתם הזה.
כשנרצה לנסות כל מיני וריאציות למימוש נצטרך לשנות קוד קיים. אם נרצה לסווג לפי אלגוריתם אחר או לייבא אחד, אז המתודות שלו יהיו שונות, והטמעתו בפרויקט לא תהיה פשוטה.

לעומת זאת, אילו נתאמץ להגדיר ממשק המתאר את הפונקציונאליות הכללית של מסווג, ובכל שאר הפרויקט יעבדו עם ממשק זה כטיפוס, אז נוכל להחליף אפילו בזמן ריצה מימושים שונים לממשק מבלי שזה ישפיע על שאר הקוד. כלומר העיצוב סגור לשינויים. ואם נרצה להוסיף עוד מימוש פשוט נוסיף מחלקה הממממשת את הממשק וכך היא תתחבר לפרויקט, או במילים אחרות, פתוח להרחבה.

**בהצלחה!**