ניתוח נתונים ולמידת מכונה

חלופה ליחידה 3

פיתוח התוכנית (תש"פ):

ד"ר תמיר חזן

קובי מייק

מרה קופלר

מהדורה מעודכנת (תשפ"א):

קובי מייק

מרה קופלר

אריאל בר-יצחק

שי פרח

כללי

נתונים והיכולת להפיק מהם תובנות וערך הפכו בשנים האחרונות לרכיב מהותי במחקר ובתעשייה. נתונים בכמות עצומה נוצרים מידי יום על ידי חיישנים, מערכות מחשב ופעילות אנושית במרחב המקוון. הנתונים כוללים תוכן במגוון רמות מורכבות, החל מנתונים מובנים כגון גלישה לאתרים, רכישות באינטרנט וכד' ועד לתוכן מורכב ולא מובנה כגון כתבות, תמונות וסרטים. היכולת לאסוף, לאגור לעבד ולהסיק מסקנות מנתונים הפכה ליכולת משמעותית במגוון גדול של יישומים לדוגמא נהיגה אוטונומית, רפואה מותאמת אישית ועוד.

על רקע עליית היקפם וחשיבותם של נתונים, צמח בשנים האחרונות תחום מדעי חדש – מדעי הנתונים. מדעי הנתונים הוא מדע בין תחומי שנוצר בשילובם של מדעי המחשב, סטטיסטיקה ותחומי הידע של הנתונים - תחומי הידע מהם נאספים הנתונים ובהם מיושמים תוצרי ניתוח הנתונים.

למידת מכונה היא אחד התחומים הצומחים באופן מואץ בשנים האחרונות. צמיחה זו נובעת הן בעקבות העלייה המואצת בכוח החישוב, הן בזכות שיפורים אלגוריתמים והן בזכות העלייה האקספוננציאלית בכמות הנתונים. אלגוריתמי למידת מכונה נמצאים לכן הן בליבה של הבינה המלאכותית והן במרכזו של מדעי הנתונים.

רציונל

לימודי מדעי הנתונים ולמידת מכונה בתיכון נפרשים על פני שלוש שנים. יחידה זו מהווה את החלק הראשון ויחידת מבוא ללימודי מדעי הנתונים ולמידת מכונה. מטרת התוכנית לחשוף את התלמידים והתלמידות למדעי הנתונים, סוגי היישומים האפשריים, עקרונות בעבודה עם נתונים, עקרונות למידת מכונה, תהליך העבודה במדעי הנתונים ויישום פרויקט.

יחידת ההמשך, התמחות בלמידה עמוקה במסגרת התמחות בהנדסת תוכנה, מעמיקה את הידע של התלמידים באלגוריתמי למידת מכונה מורכבים, רשתות נוירונים בכלל ורשתות עמוקות בפרט, הנמצאות בחזית הטכנולוגיה של למידת המכונה. ביחידה זו מעמיקים התלמידים והתלמידות אל העקרונות המתמטיים עליהם מבוססות רשתות נוירונים.

עקרונות להוראת התוכנית

עולם מדעי הנתונים ולמידת מכונה מבוסס כיום באופן כמעט בלעדי על שפת פיתון ולכן יש יתרון בהוראת התוכנית בשפת פיתון. עם זאת, חלק מהתלמידים כיום אינם לומדים פיתון בחטיבת הביניים ולכן יש צורך ללמד פיתון במסגרת התוכנית. קיימות מספר חלופות על מנת להתמודד עם סוגיה זאת:

1. ניתן ללמד את התוכנית בשפות C# או Java במקביל להוראת שפות אלו במסגרת יסודות. עם זאת, על מנת להכשיר לומדים עצמאיים בתחום מדעי הנתונים ולמידת מכונה המסוגלים להיעזר בתכנים החיצוניים הרבים הקיימים ברשת נדרש ללמד גם בסיס של שפת פיתון.
2. ניתן להקדיש את החודשיים הראשונים של השנה להוראת פיתון ורק לאחר מכן להתחיל את התוכנית במדעי הנתונים. ניתן ללמד פיתון בכיתה או באמצעות קורסים מקוונים כגון "מבוא למדעי המחשב בשפת פיתון" או self.py.
3. ניתן ללמד את התוכנית בכיתה י"א באופן מואץ לפני ובחפיפה חלקית להוראת תוכנית הלימודים בלמידה עמוקה.

הוראת התוכנית כוללת את כל השלבים של פרויקט עם נתונים כולל הגדרת שאלה, איסוף נתונים, חקר הנתונים, פיתוח מודלים ובניית דו"ח. במסגרת התוכנית נדרש ללמד לפחות אלגוריתם למידת מכונה אחד, KNN, לעומק, כך שהתלמידים יבינו את האלגוריתם על פרטיו המתמטיים במלואו.

בנוסף ניתן ללמד בסיום התוכנית אלגוריתמי למידת מכונה נוספים מבלי ללמד לעומק את מלוא הפרטים המתמטיים על אלגוריתמים אלו.

מקורות

סרטוני הסבר על היחידה:

https://www.youtube.com/playlist?list=PLUGwirBvkRns7QihJpzVt_u9ab2xVX2Z7

דוגמאות לעבודות גמר מופיעות בפרק 14.

יעדי למידה

- התלמיד/ה י/תסביר את הרעיונות המרכזיים של מדעי הנתונים כדיסציפלינה מדעית חדשה ואת מגוון האתגרים והפתרונות בהם עוסקים מדעי הנתונים
- התלמיד/ה י/תסביר הרעיונות המרכזיים בתחום למידת המכונה וי/תמנה את מגוון השיטות הקיימות בתחום.
- התלמיד/ה י/תישם שיטות העבודה במדעי הנתונים כולל איסוף נתונים, ניתוח נתונים וקבלת החלטות.
- התלמיד/ה י/תבצע מטלת ביצוע (תלקיט) כולל בחירת שאלת מחקר, איסוף נתונים, עיבודם ומידולם.

לוח זמנים

פרק	שם היחידה	שעות מעבדה
1	מבוא למדעי הנתונים	3
2	תכנות בפיתון (הוראה פרונטלית או קורס מקוון)	15
3	תכנות למדעי הנתונים	3
4	מבוא ללמידת מכונה	3
5	נתונים טבלאיים וספריית pandas	6
6	חקר נתונים (Exploratory data analysis)	6
7	אלגוריתם K Nearest Neighbors (KNN)	6
8	מדדי ביצוע של מסווג	3
9	מושגי יסוד בלמידת מכונה	3
10	הנדסת מאפיינים, ניקוי והכנת נתונים למודל מכונה	3-6
11	תהליך פיתוח פרויקט למידת מכונה	3
12	עקרונות עבודה עם נתונים לא מובנים – תמונות (רשות)	0-3
13	עקרונות אלגוריתמי למידת מכונה מודרניים	6
14	תלקיט סיכום – מטלת ביצוע (תוכנה)	30
סה"כ		**90**

הערה – ניתן להפוך את הסדר של פרקים 5,12 וללמד תחילה תמונה כנתונים ולאחר מכן נתונים טבלאיים

פרק 1: מבוא למדעי הנתונים

פרק זה פותח את היחידה ומטרתו הצגה רחבה של עולם מדעי הנתונים ויישומים במגוון תחומים. בפרק זה מוצגות דוגמאות ליישומים של מדעי הנתונים כגון מערכת להמלצה על סרטים, מערכת לזיהוי כלי רכב, מערכת לנהיגה אוטונומית, מערכת לזיהוי תרופות וכד'. בנוסף בפרק זה יוצג תהליך

העבודה במדעי הנתונים הכולל שאלת שאלות, איסוף נתונים, חקר נתונים, ניתוח ובניית מודלים, ניבוי, ויזואליזציה של התוצאות ובניית דוחות מבוססי נתונים.

נושאי הלימוד

- מהם מדעי הנתונים
- יישומים של מדעי הנתונים
- תהליך העבודה במדעי הנתונים

מטרות ביצועיות

- התלמיד/ה י/תסביר את הרעיונות המרכזיים של מדעי הנתונים כדיסציפלינה מדעית חדשה
- התלמיד/ה י/תסביר מגוון האתגרים והפתרונות בהם עוסקים מדעי הנתונים
- התלמיד/ה י/תציג יישום של מדעי הנתונים
- התלמיד/ה י/תנסח שאלת מחקר שניתן לחקור באמצעות נתונים

פרק 2: תכנות בפיתון

עולם מדעי הנתונים ולמידת מכונה מבוסס כיום באופן כמעט בלעדי על שפת פיתון ולכן על מנת להכשיר לומדים עצמאיים בתחום מדעי הנתונים ולמידת מכונה המסוגלים להיעזר בתכנים החיצוניים הרבים הקיימים ברשת נדרש ללמד גם בסיס של שפת פיתון. בנוסף, במידה והשפה הנבחרת להוראת התוכנית היא פיתון, נדרשת רמה מספקת של תכנות. במידה והתלמידים לא למדו תכנות בכיתות נמוכות יותר או למדו תכנות ברמה שאינה מספקת יש להקדיש את השעות הנדרשות על מנת ליישר קו בנושא התכנות.

נושאי הלימוד:

- משתנים וטיפוסי נתונים בסיסיים
- הוראות תנאי
- לולאות
- רשימות ומערכים
- פונקציות
- רשות: tuples, dictionary, classes

מטרות ביצועיות

- התלמיד/ה י/תבצע השמה, פעולות אריתמטיות ופעולות השוואה למשתנים מסוג שלם, נקודה צפה, בוליאני, תו ומחרוזת
- התלמיד/ה י/תבצע השמה, מנייה ופעולות השוואה לרשימות
- התלמיד/ה י/תבצע השמה, מנייה ופעולות השוואה למערכים (בפיתון – numpy arrays)
- התלמיד/ה י/תכתוב קוד הכולל תנאים מסוג if-else
- התלמיד/ה י/תכתוב קוד הכולל לולאות מסוג for each
- התלמיד/ה י/תריץ את הקוד בסביבת העבודה הנבחרת וי/תוודא את נכונותו
- התלמיד/ה י/תתמודד עם שגיאות תחביר ושגיאות לוגיות וי/תקן אותן

פרק 3: תכנות למדעי הנתונים

הגישה המקובלת כיום בעולם לעבודה עם נתונים היא גישת notebooks. בגישה זו הקוד, תוצאות הביצוע, גרפים ומידע נוסף נשזרים יחד במסמך אחד המהווה מחברת עבודה דינאמית. גישה זו

מתאימה מאוד לחקר ועיבוד של נתונים. בנוסף עבודה עם נתונים מחייבת קלט של נתונים לסביבת העבודה מקבצים.

מטרת פרק זה להשלים את פרק 2, ולהוסיף את התכנים הנדרשים לצורך עבודה עם נתונים

נושאי הלימוד:

- Notebooks כסביבת עבודה.
- קלט נתונים לסביבת העבודה. ניתן ללמד קלט תמונות על ידי ספריית matplotlib או קלט נתונים טבלאיים על ידי ספריית pandas או ספריה תואמת בשפות אחרות.

<u>מטרות ביצועיות</u>

- התלמיד/ה י/תריץ את הקוד בסביבת notebooks
- התלמיד/ה י/תטען נתונים לסביבת העבודה.

פרק 4: מבוא ללמידת מכונה

פרק זה כולל סקירה של תחום למידת המכונה והסוגים השונים של מכונות לומדות.

<u>נושאי הלימוד</u>

- Supervised learning and classification
- Unsupervised learning
- Reinforcement learning

<u>מטרות ביצועיות</u>

- התלמיד/ה י/תמנה את הסוגים השונים של מכונות לומדות
- התלמיד/ה י/תגדיר סיווג, רגרסיה, למידה מפוקחת, למידה לא מפוקחת ולמידה מחיזוקים
- התלמיד/ה י/תנסח בעיית סיווג ותגדיר מהם הנתונים הנדרשים לצורך האימון

פרק 5 – נתונים טבלאיים וספריית pandas

בפרק זה נלמד על נתונים טבלאיים. התלמידים יתנסו בחיפוש נתונים טבלאיים באתרים המציעים מסדי נתונים (לדוגמא Kaggle.com) ובטעינת נתונים טבלאיים לסביבת העבודה.

<u>נושאי הלימוד</u>

- איסוף נתונים ברשת (לדוגמא מאתר Kaggle)
- טעינת נתונים מקבצי אקסל וקבצי CSV באמצעות ספריית pandas
- סוגי משתנים: מספריים, קטגוריאליים, טקסטואליים, תאריך וכד'
- פקודות בסיסיות בpandas:
  - Head
  - Info
  - Describe
  - חיתוך עמודות על ידי ['col name']
  - חיתוך שורות על ידי [Boolean condition]
  - חיתוך שורות ועמודות על ידי loc,iloc

מטרות ביצועיות

- התלמיד/ה י/תחפש נתונים באתר kaggle.com (https://www.kaggle.com/), (או אתר אחר עם אותם אפשרויות)
- התלמיד/ה י/תציע שאלת מחקר ותאתר נתונים רלוונטיים לשאלת מחקר באתר kaggle
- התלמיד/ה י/תוריד מהאתר kaggle.com (או אתר אחר עם אותם אפשרויות) את בסיס הנתונים iris.csv (https://www.kaggle.com/uciml/iris)
- התלמיד/ה י/תטען את בסיס הנתונים iris.csv לסביבת העבודה באמצעות ספריית pandas
- התלמיד/ה י/תחתוך שורות ועמודות מתוך מאגר נתונים באמצעות אופרטור [], loc, iloc.

פרק 6 – חקר נתונים (Exploratory data analysis)

בפרק זה נלמד כיצד ניתן לחקור, להבין ולפתח אינטואיציה ביחס לנתונים על ידי סטטיסטיקה תיאורית והמחשות ויזואליות. נלמד מהם מדדי מרכוז ומדדי פיזור של נתונים, נלמד המחשות שונות בהתאם לסוג המשתנים. נשתמש בשתי ספריות מרכזיות לצורך ההמחשה: matplotlib, seaborn.

נושאי הלימוד

- מדדי מרכוז:
  - ממוצע
  - חציון
- מדדי פיזור:
  - סטיית תקן
  - שונות
  - רבעונים וטווח בין רבעוני
- התפלגות נורמלית
- סוגי גרפים שונים להמחשה של משתנה יחיד בהתאם לסוג הנתונים:
  - Count plot למשתנה בדיד
  - Histogram למשתנה רציף
  - Box plot למשתנה רציף
- סוגי גרפים שונים להמחשה של שני משתנים בהתאם לסוג הנתונים:
  - Bar plot – למשתנה אחד בדיד ואחד רציף
  - Box plot - למשתנה אחד בדיד ואחד רציף
  - Scatter plot – למשתנים רציפים
- שיטות להמחשה של יותר משני משתנים:
  - Hue – שינוי הצבע בהתאם למשתנה בדיד
  - Size – שינוי גודל הנקודה בהתאם למשתנה רציף

מטרות ביצועיות

- התלמיד/ה ת/יממש גרפים מסוג countplot, boxplot, displot, barplot, scatterplot באמצעות ספריית seaborn
- התלמיד/ה ת/יסיק מסקנות מההמחשות הויזואליות
- התלמיד/ה ת/יוסיף באמצעות ספריית matplotlib כותרות ותיאור לצירים לגרפים.
- התלמיד/ה ת/ישנה את גודל הגרף באמצעות ספריית matplotlib

פרק 7 – אלגוריתם K Nearest Neighbors (KNN)

בפרק זה נלמד אלגוריתם לומד ראשון – מסווג מסוג KNN. לאחר הוראת החלק התיאורטי של מסווג KNN התלמידים יממשו את המסווג ויבחנו את פעולתו על נתונים. באופן זה מודגם תהליך שלם: איסוף נתונים, חילוץ מאפיינים, אימון מודל ובחינת ביצועיו.

<u>נושאי הלימוד</u>

- פונקציית מרחק אוקלידית בין וקטורי מאפיינים
- אלגוריתם KNN
- מימוש האלגוריתם באמצעות ספריה
- חלוקת נתונים לסט אימון וסט מבחן
- הערכת ביצועים על ידי מדד accuracy
- משמעות היפר-פרמטר K על האלגוריתם ועל הביצועים.

<u>מטרות ביצועיות</u>

- התלמיד/ה י/טען נתונים וי/תחלץ או י/תבחר מתוכם מאפיינים ותגיות לצורך סיווג.
- התלמיד/ה י/תכיר קוד פיתון המממש את אלגוריתם KNN.
- התלמיד/ה י/תחלק את הנתונים לtrain/test באמצעות פונקציית ספריה.
- התלמיד/ה י/תסווג את הנתונים באמצעות אלגוריתם KNN מתוך ספרייה.
- התלמיד/ה י/תחשב את מדד accuracy של המסווג.
- התלמיד י/תכתוב לולאה המחשבת את הביצועים עבור מספר K שונים.

פרק 8 – מדדי ביצוע של מסווג

בפרק זה נעמיק במדדי ביצוע של מסווגים, נראה כיצד לחשב confusion matrix וכיצד להסיק מסקנות על ביצועי האלגוריתם באמצעות מדדים הנגזרים מהמטריצה.

<u>נושאי הלימוד</u>

- Confusion matrix
  - True positive
  - True negative
  - False positive
  - False negative
- Accuracy
- Precision
- Recall
- F1

<u>מטרות ביצועיות</u>

- התלמיד/ה י/תחשב confusion matrix באמצעות ספריה.
- התלמיד/ה י/תציג תצוגה גרפית של confusion matrix.
- התלמיד/ה י/תבין את המשמעות של התאים השונים בconfusion matrix.
- התלמיד/ה י/תחשב את המאפיינים precision, recall, F1.

פרק 9 - מושגי יסוד בלמידת מכונה

בפרק זה נלמד על מושגי יסוד של מכונה לומדת: תת אימון ואימון יתר. נבין כיצד ניתן לזהות וכיצד ניתן להימנע ממצבים אלו.

נושאי הלימוד

- שגיאת אימון (bias) ושגיאת מבחן (variance)
- מצב Overfit and underfit.
- מורכבות המודל והקשר בין מורכבות המודל למצבי תת-התאמה והתאמת יתר.
- התמודדות עם overfit על ידי פיקוח על מורכבות המודל: בחירת היפר-פרמטר K.

מטרות ביצועיות

- התלמיד/ה י/תחשב גרף שגיאת אימון ושגיאת מבחן של אלגוריתם KNN עבור K שונים
- התלמיד/ה י/תזהה את אזורי overfit ו-underfit בגרף

פרק 10 – הנדסת מאפיינים, ניקוי והכנת נתונים למודל מכונה

בפרק זה נלמד כיצד להתמודד עם תופעות שונות הקיימות בעולם האמיתי ומשפיעות על איכות הנתונים וכיצד לטפל בהן.

נושאי הלימוד

- זיהוי נתונים חסרים וחריגים והתמודדות איתם:
  - מחיקת שורות
  - מחיקת עמודות
  - החלפת ערכים חריגים או חסרים בערכים ממוצעים או חציוניים
- המרת נתונים קטגוריאליים לנתונים מספריים:
  - משתנים חסרי סדר על ידי המרה לone hot encoding
  - משתנים בעלי סדר על ידי החלפה למספר
- נרמול נתונים לצורך שיפור ביצועי המסווג:
  - MinMaxScaler
  - StandardScaler
- יצירת משתנים חדשים (לדוגמא יצירת משתנה בדיד מתוך משתנה רציף לצורך סיווג)

מטרות ביצועיות

- התלמיד/ה י/זהה נתונים חסרים וחריגים במאגר נתונים
- התלמיד/ה י/תמיר נתונים קטגוריאליים למספריים
- התלמיד/ה י/תנרמל נתונים באמצעות שיטות נירמול נתונים וי/תבחן את השפעת הנרמול על ביצועי המסווג

פרק 11 –תהליך פיתוח פרויקט למידת מכונה

בפרק זה נלמד על עקרונות הפיתוח של מכונה לומדת, כולל אימון, ולידציה, כיוון היפר פרמטרים ובחינה של ביצועי המכונה.

נושאי הלימוד

- חלוקת נתונים לtrain/validation/test, הסיבות לחלוקה זו וכיצד לבצעה באופן מעשי.
- כיצד ביצוע ולידציה באופן שגוי תוך שימוש במדגם האימון יכול לגרום לקבלת החלטות שגויה והערכה לא נכונה של איכות המודל
- שיטת cross validation והמקרים בהם משתמשים בשיטה זו.
- חיפוש שיטתי של היפר-פרמטרים על validation set או בגישת cross validation.
- המשמעות של היפר פרמטרים שונים באלגוריתמים שונים (בהתאם לאלגוריתמים שנלמדו)
  - KNN: K, p (distance type), weights
  - SVM: C, kernel

<u>מטרות ביצועיות</u>

- התלמיד/ה י/תחלק את הנתונים לtrain/validation/test
- התלמיד/ה י/תחפש היפר פרמטרים על ידי לולאה או על ידי GridSearchCV()

<u>פרק 12: עקרונות עבודה עם נתונים לא מובנים – תמונות (רשות)</u>

סיווג תמונות היא אחת האפליקציות המעניינות של למידת מכונה. בפרק בה נלמד כיצד מיוצגת תמונה במחשב וכיצד לחלץ מאפיינים של תמונה.

<u>נושאי הלימוד</u>

- מערכים תלת ממדיים
- ייצוג תמונה במחשב כמערך תלת ממדי של נקודות (pixels) בגודל (hight,width,3)
- טעינה של תמונות מקובץ
- תצוגה של תמונות
- קריאה וכתיבה של ערכי נקודות (pixels) בתמונה

<u>מטרות ביצועיות</u>

- התלמיד/ה י/תטען תמונה לסביבת העבודה ויציג את התמונה
- התלמיד/ה י/תשנה נקודות (pixels) בתמונה לפי תבנית רצויה באמצעות לולאת for
- התלמיד/ה י/תשנה נקודות (pixels) בתמונה לפי תבנית רצויה באמצעות גישה ישירה למערך
- התלמיד/ה י/תחשב מאפיינים של תמונה כגון ממוצע העוצמה של הצבעים בתמונה
- התלמיד/ה י/סווג תמונות על פי מאפייני RGB של התמונה

<u>פרק 13 - עקרונות אלגוריתמי למידת מכונה מודרניים</u>

אלגוריתם KNN הנלמד בתוכנית הינו אלגוריתם בסיסי ביותר והוראתו בתוכנית הינה בעיקר משיקולים פדגוגיים על מנת לאפשר לתלמידים להבין לעומק את מלוא הפרטים המתמטיים של האלגוריתם. בפרק זה התלמידים יחשפו לאלגוריתמי למידת מכונה מודרניים בעלי כושר למידה ויישומיות גבוהים. ניתן לבחור אחד או יותר מהאלגוריתמים הבאים:

- רגרסיה לינארית
- Perceptron
- SVM
- רגרסיה לוגיסטית

- רשתות נוירונים

נושאי הלימוד

- עקרונות פעולה של אלגוריתמי למידת מכונה מודרניים
- השוואה בין אלגוריתמי למידת מכונה מבחינת אופן הפעולה וביצועים
- יישום אלגוריתם למידת מכונה

מטרות ביצועיות

- התלמיד/ה ת/יישם אלגוריתם למידת מכונה נוסף
- התלמיד/ה ת/ישווה ביצועים בין שני אלגוריתמי למידת מכונה

פרק 14 - תלקיט סיכום – מטלת ביצוע (תוכנה)

התלמידים יבצעו פרויקט סיכום בהיקף של 30 שעות לימוד.

מטרות ביצועיות

- התלמיד/ה י/תציע נושא מקורי לפרויקט.
- התלמיד/ה י/תזהה מקורות נתונים.
- התלמיד/ה י/תאסוף נתונים.
- התלמיד/ה י/תאמן מספר מודלים של למידת מכונה על הנתונים שאסף.
- התלמיד/ה י/תנתח וישווה את הביצועים של האלגוריתמים שאימן באמצעות confusion matrix ומדדי ביצוע נוספים בהתאמה לבעיית הסיווג.
- התלמיד/ה י/תציע גישות לשיפור הביצועים.
- התלמיד/ה י/תכתוב דו"ח מסכם.

דרישות מהפרויקט

הפרויקט יכול להיות מוגש כמחברת עם כל התיעוד מובנה בתוך תאי טקסט או מחברת + קובץ word נפרד

- שער
  - לוגו
  - שם המחקר
  - שם התלמיד/ה
  - תאריך
- מטרת המחקר
- תיאור הנתונים
  - איפה נמצא מאגר הנתונים, מי הכין אותו, מתי, למה...
  - אילו נתונים קיימים בו (סוגי נתונים, כמות, אילו מאפיינים)
  - קישור למאגר
- חקר נתונים
  - טעינת נתונים (pandas)
  - ויזואליזציה של הנתונים – (seaborn)
- למידת מכונה

- הכנת הנתונים (בחירת מאפיינים, בחירת מטרה, הפיכת משתנים קטגוריאליים למספריים, חיפוש וטיפול בנתונים חסרים/חריגים, נרמול)
- חלוקה הנתונים ל3 חלקים: train, validation, test
- אימון אלגוריתם אחד לפחות
- כיוון היפר-פרמטרים
- בחינת ביצועים

- סיכום
  - סיכום העבודה – מסקנות לגבי הנתונים, מסקנות לגבי האלגוריתמים
  - רפלקציה אישית (מה למדת מהעבודה)

<u>פרויקטים לדוגמא</u>

- ניבוי הצלחה של שחקני כדורסל
- זיהוי תמונות של תמרורים
- ניבוי מחלות על פי סימפטומים
- ניבוי הצלחה במכירות של משחקי מחשב
- ניבוי הצלחה של סרטים
- ניבוי מזג אויר

מחוון וקריטריונים להערכה מופיעים באתר הפיקוח על הוראת מדעי המחשב.

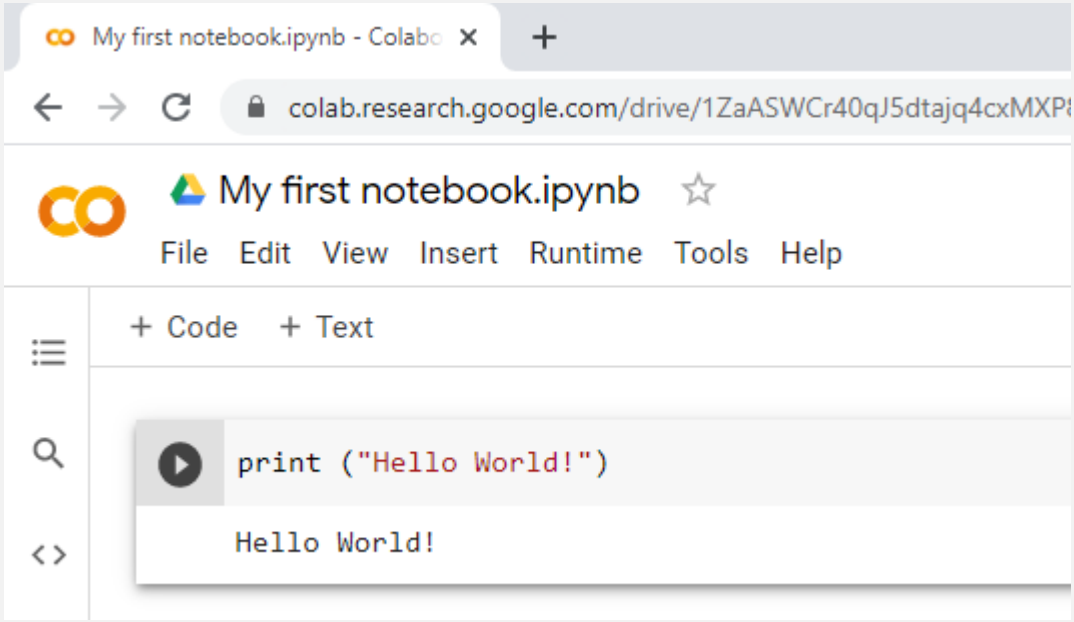
<u>דוגמאות לפרויקטים</u>

- חקר נתונים פרויקט Titanic
  - Titanic <u>לינק למחברת</u>
  - סרטוני הסבר <u>כאן</u>
- חקר נתונים פרויקט Diamonds
  - Diamonds <u>לינק למחברת</u>
  - סרטוני הסבר <u>כאן</u>
- חקר נתונים פרויקט Titanic & Diamonds Advanced
  - Titanic Advanced <u>לינק למחברת</u>
  - Diamonds Advanced <u>לינק למחברת</u>
  - סרטוני הסבר <u>כאן</u>

נספח א' – דוגמאות קוד בשפת פיתון

פרק 3 – תכנות למדעי הנתונים

- התלמיד/ה י/תריץ את הקוד בסביבת notebooks



- התלמיד/ה י/תטען נתונים לסביבת העבודה.

```
import pandas as pd
pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

- התלמיד/ה י/תחתוך שורות ועמודות מתוך מאגר נתונים באמצעות אופרטור [], loc, iloc.

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')

# Slice columns
df[['SepalLengthCm','SepalWidthCm','PetalLengthCm', 'PetalWid
thCm']]
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

```
# Slice column
df['Species']
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

```
# Slice rows
df[df['Species'] != 'Iris-virginica']
```

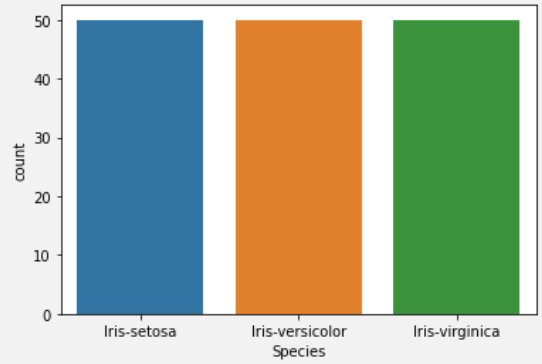
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
95	96	5.7	3.0	4.2	1.2	Iris-versicolor
96	97	5.7	2.9	4.2	1.3	Iris-versicolor
97	98	6.2	2.9	4.3	1.3	Iris-versicolor
98	99	5.1	2.5	3.0	1.1	Iris-versicolor
99	100	5.7	2.8	4.1	1.3	Iris-versicolor

```
# Slice rows and columns
df.loc[df['Species'] == 'Iris-virginica','SepalLengthCm']
```

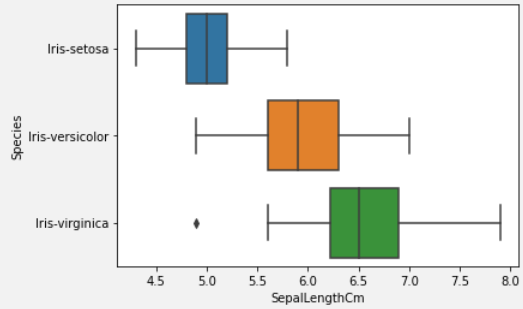
```
100    6.3
101    5.8
102    7.1
103    6.3
104    6.5
105    7.6
106    4.9
107    7.3
108    6.7
109    7.2
110    6.5
111    6.4
112    6.8
113    5.7
114    5.8
115    6.4
116    6.5
117    7.7
118    7.7
119    6.0
120    6.9
121    5.6
122    7.7
123    6.3
124    6.7
125    7.2
```

- התלמיד/ה ת/יממש גרפים מסוג countplot, boxplot, displot, barplot, scatterplot באמצעות ספריית seaborn

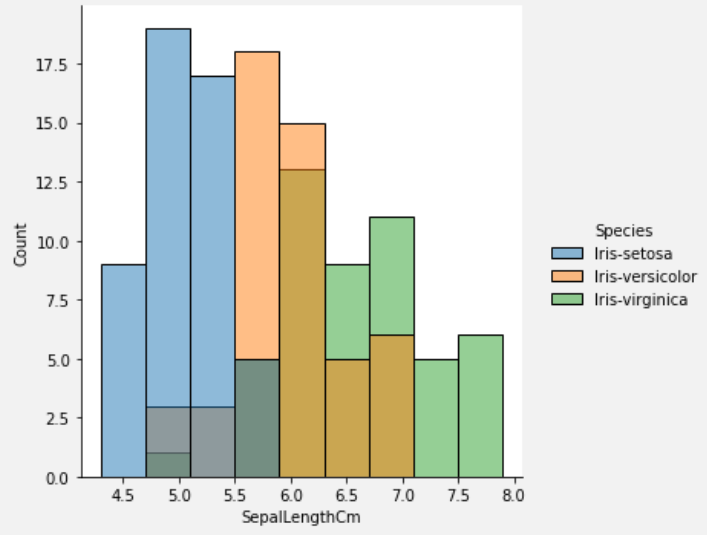
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(df['Species'])
```



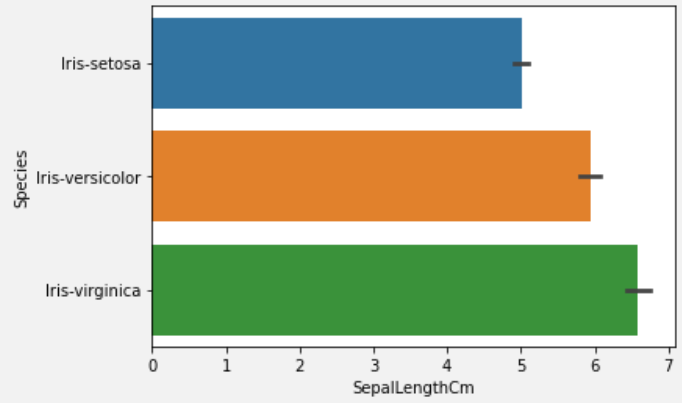

```
sns.boxplot(data=df,x='SepalLengthCm',y='Species')
```



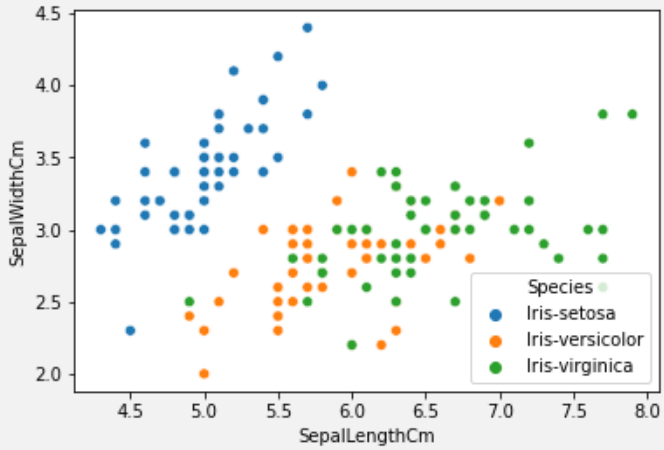

```
sns.displot(data=df,x='SepalLengthCm',hue='Species')
```




```
sns.barplot(data=df,x='SepalLengthCm',y='Species')
```




```
sns.scatterplot(data=df,x='SepalLengthCm',y='SepalWidthCm',hu
e='Species')
```



- התלמיד/ה ת/יסיק מסקנות מההמחשות הויזואליות
- התלמיד/ה ת/יוסיף באמצעות ספריית matplotlib כותרות ותיאור לצירים לגרפים.
- התלמיד/ה ת/ישנה את גודל הגרף באמצעות ספריית matplotlib

פרק 7 - אלגוריתם K Nearest Neighbors (KNN)

- התלמיד/ה י/תטען נתונים וי/תחלץ או י/תבחר מתוכם מאפיינים ותגיות לצורך סיווג.

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')

X = df[['SepalLengthCm','SepalWidthCm','PetalLengthCm',
'PetalWidthCm']]
y = df['Species']
```

- התלמיד/ה י/תכיר קוד פיתון המממש את אלגוריתם KNN.

```
import numpy as np
```

```
def KNN(k,X,y,u):
  # k number of neighbors
  # X train data
  # y train labels
  # u new sampleto be classified

  # find distance between u and each sample in the training s
et
  dist_vector = []
  for m in range(len(X)):
    dist = 0
    for i in range(len(u)):
      dist += (u[i]-X[m,i]) ** 2
    dist = np.sqrt(dist)
    dist_vector.append(dist)

  # find k nearest neighbors
  labels = []
  y_copy=list(y)
  for i in range(k):
    min_dist = min(dist_vector)
    min_dist_index = dist_vector.index(min_dist)
    labels.append(y_copy[min_dist_index])
    dist_vector.pop(min_dist_index)
    y_copy.pop(min_dist_index)

  # find most common label within k-nearest neighbors
  labels_unique = list(set(labels))
  labels_count = []
  for label in labels_unique:
    labels_count.append(labels.count(label))
  max_count = max(labels_count)
  max_count_index = labels_count.index(max_count)
  max_count_label = labels_unique[max_count_index]

  return max_count_label

k=3
X = df[['SepalLengthCm','SepalWidthCm','PetalLengthCm', 'Peta
lWidthCm']].to_numpy()
y = df['Species'].to_numpy()
u=[5.7, 3.0,  4.2,  1.2]
```

```
'Iris-versicolor'
```

- התלמיד/ה י/תחלק את הנתונים לtrain/test באמצעות פונקציית ספריה.

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')
```

```
X = df[['SepalLengthCm','SepalWidthCm','PetalLengthCm',
'PetalWidthCm']]
y = df['Species']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

- התלמיד/ה י/תסווג את הנתונים באמצעות אלגוריתם KNN מתוך ספרייה.

```
from sklearn.neighbors import KNeighborsClassifier
knn3 = KNeighborsClassifier(n_neighbors=3)
knn3.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

- התלמיד/ה י/תחשב את מדד accuracy של המסווג.

```
knn3.score(X_test,y_test)
```

```
0.9210526315789473
```

- התלמיד י/תכתוב לולאה המחשבת את הביצועים עבור מספר K שונים.

```
for k in range(1,55,2):
  knn = KNeighborsClassifier(n_neighbors=k)
  knn.fit(X_train, y_train)
  accuracy = knn.score(X_test,y_test)
  print (k,accuracy)
```

```
1 0.9210526315789473
3 0.9210526315789473
5 0.9210526315789473
7 0.8947368421052632
9 0.9473684210526315
11 0.9473684210526315
13 0.9473684210526315
15 0.9473684210526315
17 0.9473684210526315
19 0.9473684210526315
21 0.9473684210526315
23 0.9473684210526315
25 0.9473684210526315
27 0.9473684210526315
29 0.9473684210526315
31 0.9736842105263158
```

```
33 0.9736842105263158
35 0.9473684210526315
37 0.9473684210526315
39 0.9473684210526315
41 0.9473684210526315
43 0.9473684210526315
45 0.9473684210526315
47 0.9473684210526315
49 0.9473684210526315
51 0.9210526315789473
53 0.9210526315789473
```

<u>פרק 8 – מדדי ביצוע של מסווג</u>

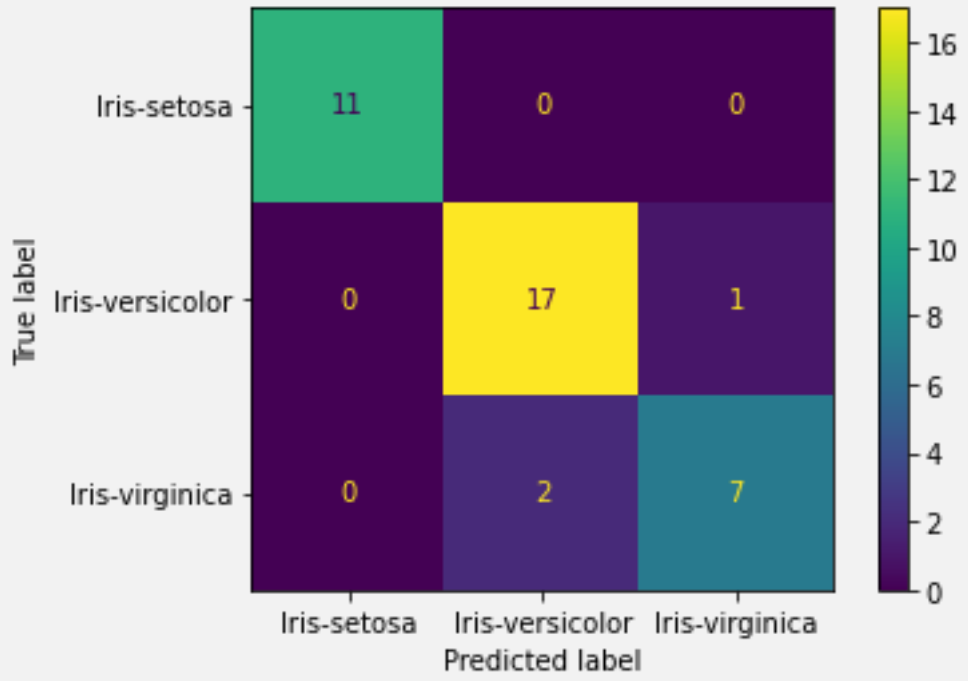
- התלמיד/ה י/תחשב confusion matrix באמצעות ספריה.

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, knn.predict(X_test))
```

```
array([[11,  0,  0],
       [ 0, 17,  1],
       [ 0,  2,  7]])
```

- התלמיד/ה י/תציג תצוגה גרפית של confusion matrix.

```
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(knn, X_test, y_test)
```



- התלמיד/ה י/תחשב את המאפיינים precision, recall, F1.

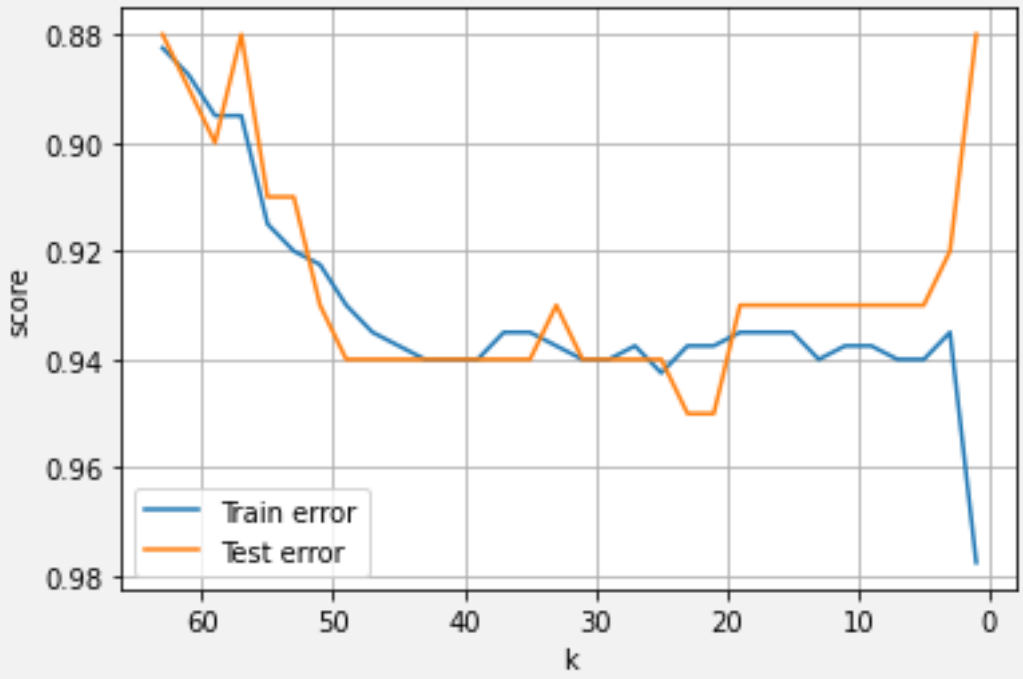
```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')

X = df[df['Species']!='Iris-
setosa'][['SepalLengthCm','SepalWidthCm','PetalLengthCm',
'PetalWidthCm']]
y = df[df['Species']!='Iris-setosa']['Species']

X_train, X_test, y_train, y_test = train_test_split(X, y)

knn33 = KNeighborsClassifier(n_neighbors=33)
knn33.fit(X_train, y_train)

tn, fp, fn, tp = confusion_matrix(y_test,
knn33.predict(X_test)).ravel()

precision = tp / (tp+fp)
recall = tp / (tp+fn)
f1 = 2*precision*recall / (precision+recall)

print ('precision:',precision,'recall:',recall,'F1:',f1)

precision: 0.9375 recall: 0.9375 F1: 0.9375
```

<u>פרק 9 - מושגי יסוד בלמידת מכונה</u>

- התלמיד/ה י/תחשב גרף שגיאת אימון ושגיאת מבחן של אלגוריתם KNN עבור K שונים

```
df = pd.read_csv('/content/drive/MyDrive/Data/Iris.csv')
df = df[df['Species']!='Iris-setosa']
df = df.sample(frac=1).reset_index(drop=True)
X = df[['SepalWidthCm','PetalWidthCm']].to_numpy()
y = df['Species'].to_numpy()

def accuracy(classifier,X,y):
  k_fold = KFold(n_splits=5)

  train_score_t = 0
  test_score_t = 0

  for train_indices, test_indices in k_fold.split(X):
      classifier.fit(X[train_indices], y[train_indices])
      train_score_t += classifier.score(X[train_indices],
y[train_indices])
      test_score_t += classifier.score(X[test_indices],
y[test_indices])

  return train_score_t/5,test_score_t/5

k_list=[]
```

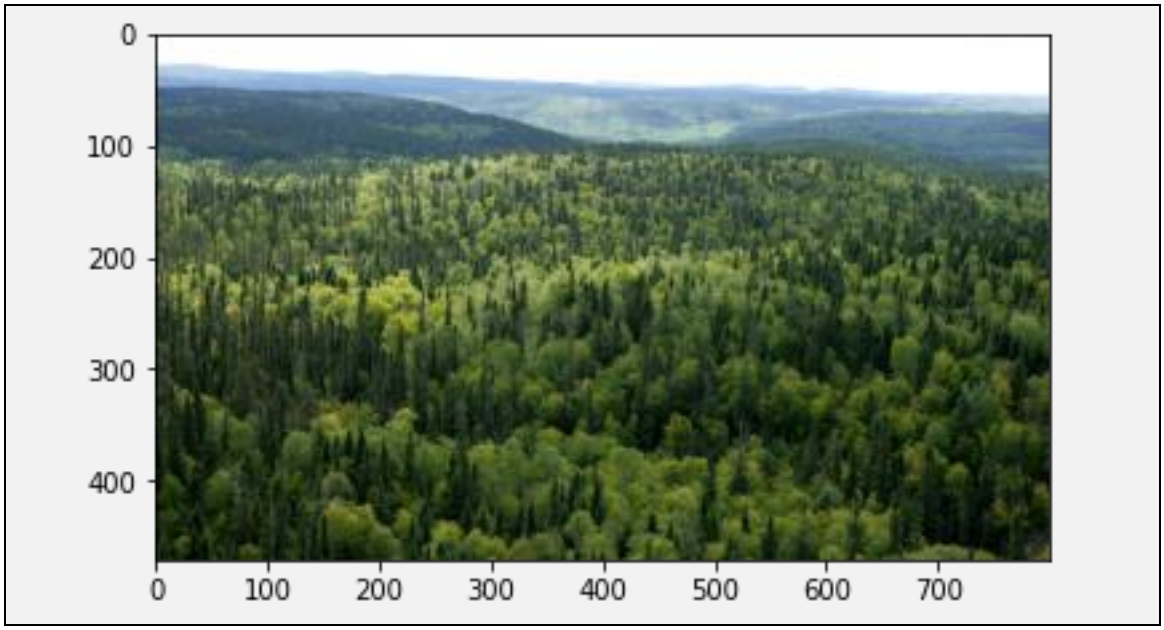
```
train_score=[]
test_score=[]

for k in range(1,65,2):
  knn = KNeighborsClassifier(n_neighbors=k)
  acc = accuracy(knn,X,y)
  k_list.append(k)
  train_score.append(acc[0])
  test_score.append(acc[1])

plt.plot(k_list,train_score,label='Train error')
plt.plot(k_list,test_score,label='Test error')
plt.legend()
plt.xlabel('k')
plt.gca().invert_xaxis()
plt.ylabel('score')
plt.gca().invert_yaxis()
plt.grid()
```



פרק 12 – עקרונות עבודה עם נתונים לא מובנים - תמונות

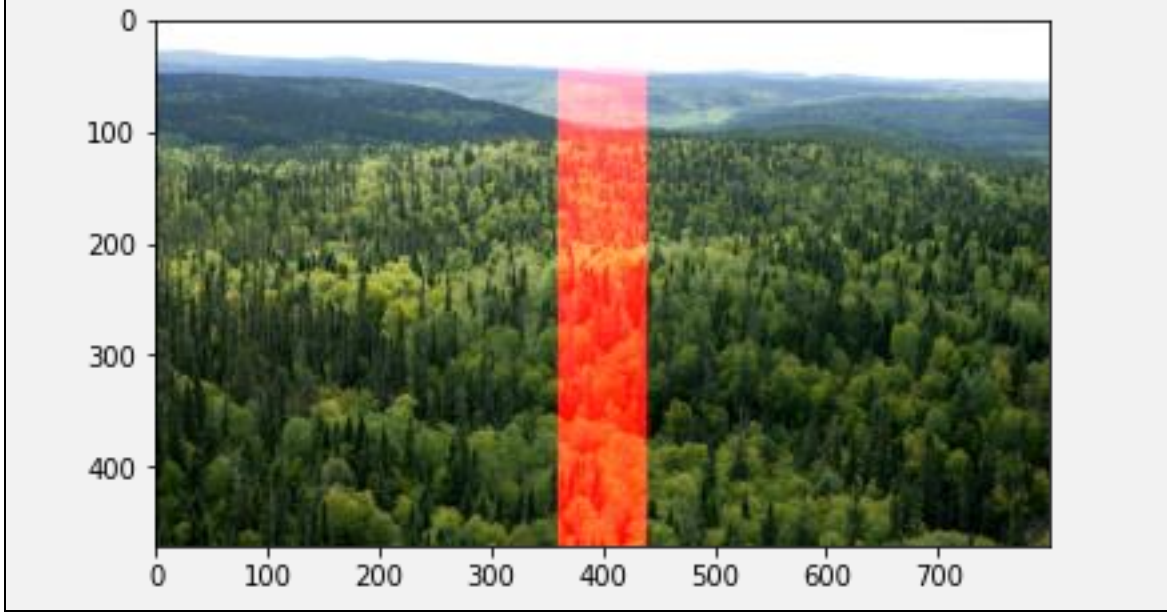
- התלמיד/ה י/תטען תמונה לסביבת העבודה ויציג את התמונה

```
forest0 = imglib.imread('/content/drive/My
Drive/Data/images/forest0.jpg')
plt.imshow(forest0)
```



- התלמיד ישנה נקודות (pixels) בתמונה לפי תבנית רצויה באמצעות לולאת for

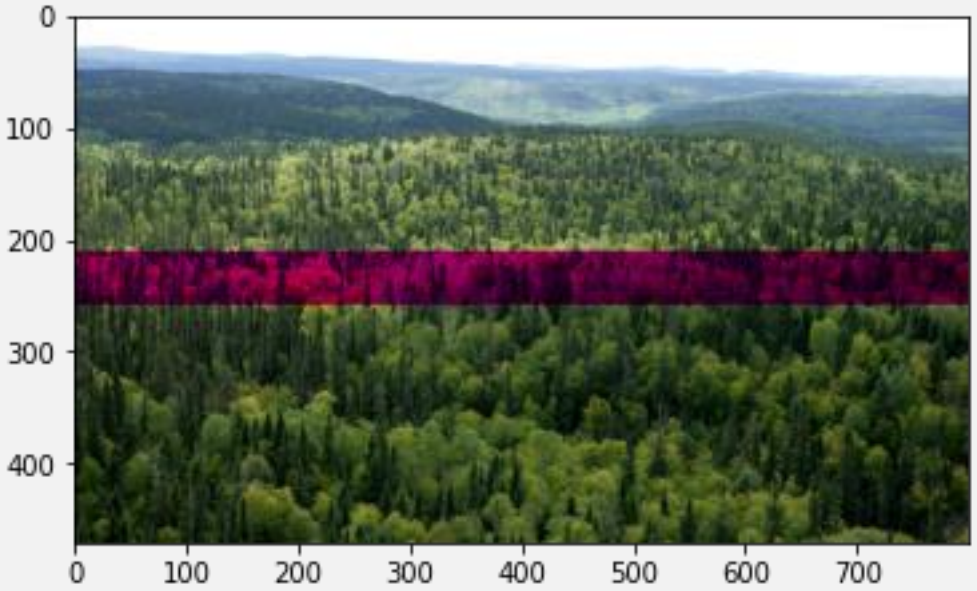
```
(hight,width,colors) = forest0.shape
forest1 = forest0.copy()
for h in range(hight):
    for w in range(int(width*.45),int(width*.55)):
      forest1[h,w,0]=255
plt.imshow(forest1);
```



- התלמיד/ה י/תשנה נקודות (pixels) בתמונה לפי תבנית רצויה באמצעות גישה ישירה למערך

```
forest2 = forest0.copy()
h_start = int(hight*0.45)
h_end   = int(hight*0.55)
```

```
forest2[h_start:h_end,:,1]=0
plt.imshow(forest2);
```



- התלמיד/ה י/תחשב מאפיינים של תמונה כגון ממוצע העוצמה של הצבעים בתמונה

```
def img_to_features(image):
  (hight,width,colors) = image.shape
  n_pixels = hight * width
  average_red   = image[:,:,0].sum() / n_pixels
  avarage_green = image[:,:,1].sum() / n_pixels
  avarage_blue  = image[:,:,2].sum() / n_pixels
  features = [average_red,avarage_green,avarage_blue]
  return features
print (img_to_features(forest0))
```

```
[86.43915605095542, 103.66260350318471, 79.31247876857749]
```