

תכנית לימודים במקצוע מדעי המחשב

"בינה מלאכותית בשילוב מדעי המחשב"
לחטיבת הביניים

2AI בשילוב אלגוריתמיקה
באמצעות Python

חלק ב'

תכנית לימודים חט"ב

נימוקים לפיתוח תכנית לימודים בינה מלאכותית בשילוב מדעי המחשב לכיתות ח' בחטיבות הביניים

תכנית זו שמה לעצמה למטרה לחנך את תלמידי חטיבת הביניים ליהנות מהעיסוק במדעי המחשב ובבינה המלאכותית באופן מושכל, ולהצליח לפתור בעיות אלגוריתמיות על ידי תכניות מחשב בהתאם לפוטנציאל כל תלמיד ותלמידה. התכנית מעניקה לתלמידים כרטיס כניסה לעולם הטכנולוגיה של המאה ה-21, עולם שבו מערכות בינה מלאכותית כבר אינן תוספת אלא מרכיב שגרתי בחיי היום יום ובעולם התעסוקה, מעודדת אותם לעבודה בקבוצות, לחקור טכנולוגיות מתקדמות ולבקר אותן.

מסיבה זו, נושאי הבינה המלאכותית בתכנית אינם נלמדים כיחידה נפרדת מן התכנות, אלא שזורים בתוך שגרת הלימוד של פייתון: כל אבן בניין אלגוריתמית מקבלת מקבילה יישומית בעולם ה-AI, כך שנושא הפעולות מלווה בעבודה מול API, מחרוזות מלוות בהיכרות עם טוקנים ואופן פעולתו של מודל שפה, ורשימות מלוות בבניית מערכת המלצה בסיסית. מבנה זה מבטיח שהתלמידים אינם משתמשים בכלי AI ללא הבנה, אלא מיישמים אותו על בסיס מיומנויות תכנות מוצקות שרכשו בעצמם, ובד בבד מתרגלים שוב ושוב את ההבחנה בין אלגוריתם קלאסי, שקוף וקבוע, לבין מערכת בינה מלאכותית המבוססת על הסתברות ולומדת מנתונים. הבחנה זו היא תנאי הכרחי לחשיבה ביקורתית: להבין מדוע מודל שפה עלול "להזות", מאין נובעות הטיות במערכת סיווג או המלצה, ומהי האחריות המוטלת על מי שכותב או משתמש בקוד שמפעיל מערכות כאלה, ועל כן שולב בתכנית נושא האתיקה העוסק בהטיות, בפרטיות ובזכויות יוצרים.

למימוש עקרונות האלגוריתמיקה, התכנות ושילוב הבינה המלאכותית בתכנית נעשה שימוש בשפת Python, מהסיבות הבאות: מבנה השפה מאפשר המשך ישיר לקו המחשבה שהחל בהוראת אלגוריתמיקה בכיתה ז' והשפה מאפשרת תכנות פרוצדורלי. צורת הכתיבה קלה ללימוד הן בכיתה והן בחקר עצמי, וקלה לקריאה אלגוריתמית, והשפה הוגדרה באוניברסיטאות מובילות בעולם כשפה ראשונית. ייחודיות השפה מתבטאת בכך שהיא מאפשרת ממשק גרפי (GUI) קל ונוח ללימוד, ואומצה על-ידי הקהילה המדעית לביצוע מגוון רחב של חישובים, ובהם תחומי הבינה המלאכותית עצמם, מה שהופך אותה לבחירה טבעית במיוחד עבור תכנית המשלבת AI.

תכנית לימודים זו נעזרת בצורת הכתיבה הקלה שבשפה כדי לפתח אצל התלמידים והתלמידות תחושת מסוגלות למידה עצמית גבוהה, הן במדעי המחשב והן בבינה מלאכותית. התכנית נעזרת במבנה השפה כדי לתרגל חשיבה מסדר גבוה ולפתח יכולות למידה עצמית, וייחודיות השפה, לצד השילוב המתמיד של AI בתוכה, מעודדת יצירתיות, למידה בקבוצה והעצמה כיתתית מחד, ואחריות אישית ויכולות חקר וביקורת מאידך.

סילבוס תכנית "AI2 בשילוב פייתון ב"

מס' פרק	שם הפרק	נושאי הפרק	שעות התנסות	שעות עיוניות	סה"כ שעות
1	חזרה על נושאי פייתון א'	מפגש מחודש עם מושגים אלגוריתמיים, סביבת עבודה, משתנים, משפטי תנאי וביצוע חוזר (for, while)	5	1	6
2	שיעור פתיחה לשנה ב' 2AI	מהפכת ה-AI תכנות קלאסי לעומת AI נתונים, אחריות והסכנה בהטיות "מצרכנים ליצרנים": החזון השנתי פיתוח קוד המשלב בינה מלאכותית	2	2	4
3	פעולות	- מפגש מחודש: פעולה בלי פרמטרים שאינה מחזירה ערך - מפגש מחודש: פעולה עם פרמטרים שאינה מחזירה ערך - פעולה בלי פרמטר המחזירה ערך - פעולה עם פרמטר המחזירה ערך	4	2	6
4	API - להכניס בינה לקוד	שילוב AI בקוד שימוש במפתח API (בדגש על נושא פונקציות) שרשור פרומפט פרויקטון: פתרון בעיה המשלבת תנאים, לולאות, פונקציות, API	3	1	4
5	מחרוזות	- המחרוזת הריקה - אופרטורים מחרוזתיים (* +) - אופרטור היחס in - חיתוך מחרוזת ([], ::) - פעולות מרכזיות: len, find, upper, lower,	6	2	8

			count, startswith, endswith, isalpha, isnumeric, replace - ביצוע חוזר מוגבל מראש (הדגשת ההבדל בין אינדקס לבין תוכן) for i in st: for ch in st:		
6	2	4	Tokens, מפת משמעויות, הקשב העצמי (הקשר).	מודל שפה - מתקדם	6
10	3	7	- מהי רשימה, גישה לאיברי הרשימה - שרשור רשימות - אורך רשימה (הפעולה len) - פעולות על רשימה: count, in, remove, pop, index, append, insert, join split, sort, reverse - פתרון בעיות אלגוריתמיות המשלבות את מבנה הנתונים רשימה	רשימה list	7
8	4	4	מהו סיווג, מהי תכונה, מהו קו מפריד. אימון ומבחן, סיווג באמצעות Teachable Machines, מדד הדיוק.	סיווג	8
8	4	4	2 סוגים של מערכות המלצה: מבוססת תוכן, מבוססת משתמשים.	מערכות המלצה	9

פרק 1: מפגש מחודש עם מבנים אלגוריתמים בסיסיים וביצוע חוזר

מטרת הפרק:

מטרת הפרק: מפגש מחודש עם תנאים לוגיים, ביצוע מותנה וביצוע חוזר.

מטרות ביצועיות - מדעי המחשב:

תנאים לוגיים:

1. התלמיד ידע להסביר מהם תנאים לוגיים מורכבים ואת שלוש הפעולות הבוליאניות הבסיסיות ("וגם", "או", "שלילה").
2. התלמיד ידע לפתח בגישה מודולרית פתרונות אלגוריתמיים המשתמשים בפעולות בוליאניות ותנאים מורכבים ("וגם", "או" ושלילה).

ביצוע מותנה:

3. התלמיד ידע להסביר מהו ביצוע מותנה, ובפרט מה תפקיד התנאי הלוגי המשולב בו, ומהי מטרתו של ביצוע מותנה.
4. התלמיד ידע לפתח בגישה מודולרית פתרון אלגוריתמי הכולל שימוש בביצוע מותנה עם חלופה (else).
5. התלמיד ידע לפתח בגישה מודולרית פתרון אלגוריתמי הכולל שימוש בביצוע מותנה ללא חלופה.
6. התלמיד ידע לפתח בגישה מודולרית פתרון אלגוריתמי הכולל קיבוצ של ביצועים מותנים ביצוע מותנה מתגלגל (elif).
7. התלמיד ידע לפתח בדיקת תקינות קלט בעזרת הוראת ביצוע מותנה.

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לכתוב תנאים לוגיים מורכבים ולשלבם בהוראות מותנות.
 2. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשלב פעולות על מספרים ועל תנאים לוגיים.
 3. התלמיד ידע לממש ביצוע מותנה עם חלופה.
 4. התלמיד ידע לממש ביצוע מותנה ללא חלופה.
 5. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט שכולל ביצועים מותנים מסוגים שונים וקיבוצ של ביצועים מותנים.
- מושגי מדעי המחשב: טיפוס נתונים בוליאני, תנאים בוליאניים פשוטים ומורכבים, ביצוע מותנה, בדיקת תקינות קלט.

מושגים תכנותיים ב-Python: אופרטורים בוליאניים, הוראת ביצוע מותנה מתגלגל.

דרכי ההוראה:

1. הפרק יילמד תוך ביצוע משימות מובנות ומשימות התנסות חופשית בסביבה. כל מושג, גם הפשוט ביותר, צריך להילמד בהקשר של פתרון בעיה מסוימת, בעקבות צורך שמתעורר במהלך פתרון הבעיה.

2. הצגת (או הרחבת הבעיה הקודמת אל) בעיה המזמנת פתרון אלגוריתמי שכולל שימוש בקשרים לוגיים. פיתוח הפתרון האלגוריתמי ומימוש ב-Python.
3. הצגת בעיה המזמנת פתרון אלגוריתמי שכולל ביצוע מותנה עם חלופה. פיתוח הפתרון האלגוריתמי ומימוש ב-Python.
4. הצגת (או הרחבת הבעיה הקודמת אל) בעיה המזמנת פתרון אלגוריתמי שכולל ביצוע מותנה ללא חלופה. פיתוח הפתרון האלגוריתמי ומימוש ב-Python.
5. הצגת (או הרחבת הבעיה הקודמת אל) בעיה המזמנת פתרון אלגוריתמי שכולל קינן של ביצועים מותנים. פיתוח הפתרון האלגוריתמי ומימוש ב-Python.

דרכי הערכה:

1. מעקב אחר ביצוע אלגוריתמים נתונים הכוללים תנאים לוגיים פשוטים ומורכבים.
2. זיהוי מטרתו של אלגוריתמים נתונים הכוללים תנאים לוגיים פשוטים ומורכבים.
3. פיתוח פתרון אלגוריתמי עבור בעיה נתונה תוך שימוש בתנאים לוגיים פשוטים או מורכבים.
4. מעקב אחר ביצוע אלגוריתם נתון הכולל ביצוע מותנה (עם/בלי חלופה או מקונן).
5. זיהוי מטרתו של אלגוריתם נתון הכולל בתוכו ביצוע מותנה (עבור סוגי ביצועים מותנים שונים וקינן ביצועים מותנים).
6. פיתוח פתרון אלגוריתמי עבור בעיה נתונה תוך שימוש בביצוע מותנה עם חלופה.
7. פיתוח פתרון אלגוריתמי עבור בעיה נתונה תוך שימוש בביצוע מותנה ללא חלופה.
8. פיתוח פתרון אלגוריתמי עבור בעיה נתונה תוך שימוש בביצוע מותנה מקונן.
9. מימוש פתרון אלגוריתמי המשתמש בביצוע מותנה עם חלופה.
10. מימוש פתרון אלגוריתמי המשתמש בביצוע מותנה ללא חלופה.
11. מימוש פתרון אלגוריתמי המשתמש בביצוע מותנה מקונן.

ביצוע חוזר:

מטרות ביצועיות - מדעי המחשב:

1. התלמיד יזהה בעיות שפתרון דורש ביצוע חוזר.
2. התלמיד יבחין בין בעיה הדורשת פתרון בביצוע חוזר באורך ידוע מראש לבין בעיה הדורשת פתרון ביצוע חוזר מותנה.
3. התלמיד יסביר את הצורך בתנאי הסיום של לולאה מותנית.
4. התלמיד יסביר במילים שלו את נכונות האלגוריתם המתבסס על ביצוע חוזר.
5. התלמיד יכתוב קטע קוד ובו ביצוע חוזר מוגבל מראש.
6. התלמיד יכתוב קטע קוד ובו ביצוע חוזר מותנה.
7. התלמיד יסביר את ההבדלים בין פעולות ביצוע חוזר (ביצוע חוזר מוגבל מראש, ביצוע חוזר מותנה).
8. התלמיד יכתוב לולאה מותנית שתנאי העצירה שלה תלוי משתנה.

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לממש הוראת ביצוע חוזר מוגבל מראש.
 2. התלמיד ידע לממש הוראת ביצוע חוזר מותנה.
 3. התלמיד ידע לממש פונקציה ללא פרמטרים המכילה ביצוע חוזר.
- מושגי מדעי המחשב: ביצוע חוזר מוגבל מראש (גודל קפיצה כלשהו), טווח משתנה לולאה, ביצוע חוזר מותנה.
- מושגים תכנותיים ב-Python: נקודותיים, הזחה, while, for...range().

דרכי ההוראה:

1. יש להדגים ביצוע חוזר באמצעות פסאודו-קוד נייטרלי שאין לו תרגום ישיר לשפת התכנות, למשל "חזור n פעמים". במקרים כאלה ניתן לתרגל גם באמצעים לא ממוחשבים.
 3. יש להדגיש את משתנה הלולאה ושימושיו:
- for item in range(n) / for item in range(start, stop, step).
4. יש להדגיש כי יתכן ביצוע שאינו מסתיים (דהיינו בעל אורך אינסופי).
 5. יש להדגיש את המשמעות המיוחדת של מושג הנכונות בהקשר של ביצוע חוזר (מילוי אחר התנאים הנדרשים בסיום הביצוע החוזר והבטחת סיום הביצוע).
 6. יש לתרגל באופן מעמיק את כל נושא הביצוע החוזר תוך פתרון בעיות אלגוריתמיות.

דרכי הערכה:

1. מעקב אחר אלגוריתם (תכנית וגם אלגוריתם מילולי) המבוסס על ביצוע חוזר.
2. מעקב אחר בעיה זיהוי סוג הביצוע החוזר הנדרש - אורך ידוע מראש או מותנה.
3. זיהוי מספר הפעמים שתתבצע לולאה נתונה.
4. כתיבת פסאודו קוד או קוד לפתרון בעיה הדורשת ביצוע חוזר.
5. כתיבת תכנית הכוללת ביצוע חוזר מוגבל מראש.
6. כתיבת תוכנית הכוללת ביצוע חוזר מותנה.

חלוקת השעות לפרק 1

הנושא	סך שעות	שעות עיוניות	שעות התנסות
תנאים לוגיים וביצוע מותנה	2	0	2
ביצוע חוזר באורך ידוע מראש	1	1	1

1	1	1	ביצוע חוזר מותנה
4	2	6	סה"כ שעות

פרק 2: שיעור פתיחה לשנה ב' AI 2

מטרת הפרק:

הבנת המעבר מתכנות קלאסי מבוסס חוקים ללמידת מכונה הלומדת מדוגמאות, פיתוח חשיבה ביקורתית על הטיות בנתונים והבנת מושג האחריות האתית של מפתחים.

מטרות ביצועיות:

- התלמיד ידע להסביר את ההבדל העקרוני בין תכנות קלאסי (אלגוריתם מבוסס חוקים קבועים וצפויים מראש שנכתבו על ידי מפתח) לבין למידת מכונה (מודל הלומד דפוסים ומשתפר מתוך דוגמאות ונתונים).
- התלמיד ידע להגדיר מהי "הטיה" בלמידת מכונה ולהסביר כיצד מאגר נתונים מוטה או לא מגוון משפיע על החלטות המודל בעולם האמיתי (כגון: תוכנת קבלה לעבודה, מכונת אוטונומית, מערכת רפואית).
- התלמיד ידע להסביר את החשיבות של מגוון נתונים (דאטה) במאגר הלמידה לשם יצירת מערכת הוגנת ומדויקת יותר לכולם.

מושגי מדעי המחשב / AI: למידת מכונה (Machine Learning) לעומת תכנות קלאסי (Rule-based), מאגר נתונים (Dataset), נתונים מוטים (Bias), הטיות תרבותיות וחברתיות, מגוון נתונים, זיהוי דפוסים.

דרכי ההוראה:

- פתיחת השיעור באמצעות משחק ודיון סביב Quick, Draw! של גוגל לשם הבנת הדרך שבה המחשב מנחש את הציור בזמן אמת.
- עבודה עצמאית עם דף העבודה "איך מחשב רואה בלי עיניים?" ובילוש בתוך מאגר הנתונים הציבורי של הציורים.
- קיום דיון כיתתי בנושא אתיקה והטיות במערכות מבוססות AI בעולם האמיתי (כגון רובוטים רפואיים, מכונות אוטונומיות ותוכנות מיון קורות חיים) והאחריות המקצועית של המפתח.

דרכי הערכה:

1. מענה על שאלות ניתוח בדף העבודה לגבי סיבות לכישלון זיהוי של המחשב (למשל, היעדר ייצוג לקומקומים מודרניים/חשמליים במאגר).
2. הצגת דוגמאות למקרים שבהם נתונים מוטים עלולים להוביל להחלטות שגויות, לא הוגנות או מסוכנות.
3. השלמת משפטי סיכום מונחים בנושא השפעת גודל ומגוון מאגר הנתונים על רמת הדיוק וההגינות של המודל.

חלוקת השעות לפרק 2:

שעות התנסות	שעות עיוניות	סך שעות	הנושא
2	1	2	תכנות קלאסי לעומת AI, מהפכת ה-AI והתנסות ב-Quick, Draw!
2	1	2	סה"כ שעות

פרק 3 - פעולות

מטרת הפרק:

הבנת הצורך בשילוב פעולות כחלק מעקרון המודולריות, ההכמסה. הבחנה בין פעולה עם/בלי פרמטרים ופעולה שאינה מחזירה ערך ומימוש פעולות בהתאם לדרישות.

מטרות ביצועיות - מדעי המחשב:

הצגת בעיה אשר פתרונה מצריך שימוש בהפשטה פרוצדורלית והגעה לפתרון אלגוריתמי עבורה תוך כדי דיון כיתתי (אפשר גם להתחיל בבעיה ראשונית ולהרחיב אותה באופן הדרגתי לבעיות נוספות כך שהמושגים יוצגו בהדרגה):

1. אלגוריתם כאבן בניין אלגוריתמית.
2. שימוש באבן הבניין האלגוריתמית, תיאור פנימי שלה.
3. שימוש מול מימוש, הסתרת פרטי המימוש.
4. רפלקציה דרך ההפשטה הפרוצדורלית על תהליך פתרון הבעיות ביחידה כפיתוח על פי עקרון bottom-up והאבחנה בין "מה" ל"איך".

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד יכתוב פעולה שאינה מקבלת פרמטרים ואינה מחזירה ערך ויזמנה.
 2. התלמיד יכתוב פעולה המקבלת פרמטרים ואינה מחזירה ערך ויזמנה.
 3. התלמיד יכתוב פעולה המקבלת פרמטרים ומחזירה ערך ויזמנה.
 4. התלמיד יבחין בין טווח הכרה של משתנה מחוץ ובתוך הפעולה. מושגי מדעי המחשב: פעולה, פרמטרים, טווח הכרה של משתנה - משתנה מקומי, משתנה בעל מאפיין global.
- מושגים תכנותיים ב-Python: נקודותיים, הזחה, def.

דרכי הערכה:

1. מעקב אחר אלגוריתם אשר משתמש בהפשטה פרוצדורלית.
2. פיתוח ומימוש של פתרון אלגוריתמי עבור בעיה נתונה, אשר משתמש בהפשטה פרוצדורלית.

דרכי ההוראה:

1. הפרק יילמד תוך ביצוע משימות מובנות ומשימות התנסות חופשית בסביבה. כל מושג, גם הפשוט ביותר, צריך להילמד בהקשר של פתרון בעיה מסוימת, בעקבות צורך שמתעורר במהלך פתרון הבעיה.
2. הצגת בעיה המזמנת שימוש בהפשטה פרוצדורלית דרך קופסה שחורה.
3. פיתוח הדרגתי של פתרון אלגוריתמי לבעיה, אשר משתמש באבני בניין פרוצדורליות, עם ובלי פרמטרים.
4. מימוש הפתרון האלגוריתמי ב-Python.

חלוקת השעות לפרק 3

הנושא	סך שעות	שעות עיוניות	שעות התנסות
פעולה בלי/ עם פרמטרים שאינה מחזירה ערך	2	0	2
פעולה בלי/ עם פרמטרים שאינה מחזירה ערך	4	1	3
סה"כ שעות	6	1	5

פרק 4: API - להכניס בינה לקוד

מטרת הפרק:

הקניית היכולת לשלב מודלי שפה מתקדמים ישירות בתוך קוד פייתון באמצעות קריאות API, תוך יישום פונקציות עטיפה, בנייה דינמית של פרומפטים ופיתוח פרויקטים אינטראקטיביים.

מטרות ביצועיות - AI

- התלמיד ידע להסביר מהו ממשק תכנות יישומים (API) וכיצד הוא מאפשר לתכנית פייתון לתקשר עם שרתים ולקבל יכולות חישוב ועיבוד מוכנות.
- התלמיד יבין את מגבלות ה-API (כמו מגבלות קצב - Rate Limits ושגיאות עומס) ואת השפעתן הכלכלית והתפעולית על עיצוב התוכנה.
- התלמיד יבין את החשיבות של חוויית משתמש חסינה (Robust UX) באמצעות בדיקות תקינות קלט וטיפול בסיסי במצבי עומס ושגיאה.
- התלמיד יסביר את הלוגיקה של צבירת נתונים לאורך זמן ושליחתם המרוכזת לקבלת מענה אלגוריתמי מה-API.

מטרות ביצועיות - תכנות בסביבת Python:

- התלמיד ידע לייבא ספריות צד-שלישי מתאימות ולהגדיר לקוח ייעודי באמצעות מפתח API אישי מאובטח.

2. התלמיד יכתוב פונקציית עטיפה (def) המקבלת פרומפט כפרמטר, מבצעת קריאת API ומחזירה (return) את תשובת המודל לתוכנית הראשית.
3. התלמיד ישתמש במחרוזות מעוצבות (f-strings) לבניית פרומפטים דינמיים המשלבים קלטים משתנים מהמשתמש בזמן ריצה.
4. התלמיד יממש לוגיקת לולאה מותנית (while) לצבירת נתונים ושרשור מחרוזות לקראת שליחה אחת מרוכזת מחוץ ללולאה.
5. התלמיד יממש מנגנוני בדיקת תקינות קלט (באמצעות len ותנאים מורכבים), סינון מחרוזות (באמצעות in) והגדרת ערכי ברירת מחדל בקוד.

דרכי ההוראה:

1. הצגת מושג ה-API כ"מלצר" שמקשר בין הקוד (הלקוח) לבין המוח של Gemini (המטבח) באמצעות המצגת.
 2. עבודה עצמאית על דפי תרגול תוך שילוב הדרגתי: שליחה פשוטה, פונקציית עטיפה, פרומפט דינמי, לולאת צבירה ופיצ'רים מתקדמים.
 3. עבודה על פרויקטון "יום מאוזן" – בניית אפליקציית ניהול זמן חכמה המשלבת לולאות, תנאים ופנייה למודל השפה לקבלת עצות מותאמות אישית.
- מושגי מדעי המחשב / AI: ממשק תכנות יישומים (API), מפתח API Key, מגבלות קצב/קריאות (Rate Limits), שגיאות עומס (Rate errors / Service unavailable), פרומפט (Prompt), פרומפט דינמי.
- מושגים תכנותיים ב-Python: ייבוא ספריות חיצוניות (import), הגדרת פונקציה (def), החזרת ערך (return), מחרוזת מעוצבת (f-string), לולאה מותנית (while), שרשור/צבירת מחרוזות, אופרטור היחס (in) לסינון מחרוזות, בדיקת אורך (len).

דרכי הערכה:

1. בדיקה שהתלמידים משתמשים ב-return בתוך פונקציית העטיפה שלהם, ושמירת מפתח ה-API בצורה בטוחה.
2. הערכת הקוד של לולאת צבירת המצרכים לוודא שאין קריאות API בתוך הלולאה (למניעת בזבז קריאות) אלא קריאה אחת בלבד מחוץ ללולאה.
3. מימוש לפחות פיצ'ר מתקדם אחד (סינון מילים אסורות, בדיקת אורך או ערך ברירת מחדל).

חלוקת השעות לפרק 4:

שעות התנסות	שעות עיוניות	סך שעות	הנושא
1	1	2	מבוא ל-API, מפתח API, ייבוא ספריות וכתובת פונקציית עטיפה
2	0	2	הנדסת פרומפטים דינמית, לולאות צבירה ופרויקטון "יום מאוזן"
3	1	4	סה"כ שעות

פרק 5: מחרוזות

מטרת הפרק:

היכרות עם טיפוס נתונים מורכב, הכולל פעולות מובנות בשפה ושילובו בפתרון בעיות אלגוריתמיות.

מטרות ביצועיות - מדעי המחשב:

1. התלמיד ידע להסביר מהי מחרוזת ומה מטרתה.
2. התלמיד יכיר את אופרטור השרשור + ואופרטור השכפול * (חזרה מפייתון א')
3. התלמיד יכיר את מאפייני המחרוזת: מציין מיקום תו במחרוזת [] ויבחין בין מיקום תו במחרוזת לבין ערך תו במחרוזת על פי מציין מיקום.
4. התלמיד יכיר את מנגנון חיתוך מחרוזות על פי טווח (התחלה, סוף, קפיצה)
5. התלמיד יכיר את הפעולות המרכזיות על מחרוזת, וביניהן: len, in, startswith, endswith, in, upper, lower, count, find, isalpha, isnumeric, len, replace, indexOf
6. התלמיד ידע לפתח בגישה מודולרית פתרונות אלגוריתמיים המשתמשים באופרטורים חשבוניים.
7. התלמיד ידע לפתח בגישה מודולרית פתרונות אלגוריתמיים המשתמשים בפעולות מתקדמות על מחרוזות.

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשתמש בפעולות ומתודות על מחרוזות.

2. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשתמש בפעולות חיתוך על מחרוזות כגון: [start:end:step].
3. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשתמש באופרטורים על מחרוזות כגון: +, *.
4. התלמיד יכיר את הפעולות המרכזיות על מחרוזת, וביניהן: len, in, find, upper, lower, count, startswith, endswith, isalpha, isnumeric, replace.

מושגי מדעי המחשב: מחרוזות כטיפוס נתונים מורכב. התייחסות ראשונית אל מחרוזת כישות הנושאת איתה תוכן ופעולות שניתן לבצע עליו.

מושגים תכנותיים ב-Python: מחרוזות, אופרטורים ופעולות על מחרוזות, תחום ערכים סדורים (range), הוראת ביצוע חוזר.

דרכי ההוראה:

1. הפרק יילמד תוך ביצוע משימות מובנות ומשימות התנסות חופשית בסביבה. כל מושג, גם הפשוט ביותר, צריך להילמד בהקשר של פתרון בעיה מסוימת, בעקבות צורך שמתעורר במהלך פתרון הבעיה.
2. הצגת בעיה המזמנת פתרון אלגוריתמי הכולל ביצוע חוזר מוגבל מראש. פיתוח הפתרון האלגוריתמי ומימושו ב-Python.
3. הצגת בעיה המזמנת שימוש במחרוזות.
4. פיתוח הדרגתי של פתרון אלגוריתמי לבעיה, שמבצע פעולות על מחרוזות ומימושו ב-Python.

דרכי הערכה:

1. מעקב אחר ביצוע אלגוריתמים נתונים הכוללים שימוש במחרוזות.
2. זיהוי מטרתם של אלגוריתמים נתונים הכוללים מחרוזות.
3. פיתוח ומימוש של פתרון אלגוריתמי עבור בעיה המצריכה שימוש במחרוזות.

חלוקת השעות לפרק 5

הנושא	סך שעות	שעות עיוניות	שעות התנסות
פעולות על מחרוזת	2	1	1
חיתוך מחרוזת	3	1	2

2	1	3	מעבר על מחרוזת בלולאה
5	3	8	סה"כ שעות

פרק 6: מודל שפה - מתקדם

מטרת הפרק:

- הבנת אופן הפעולה הפנימי של מודלי שפה גדולים (LLMs) דרך פירוק טקסט לטוקנים, מיקום סמנטי במפת משמעויות ומנגנון הקשב העצמי, ומימוש משחק הניחושים "סמנטעל" בפיתוח.
- התלמיד יסביר מהי טוקניזציה (Tokenization) וכיצד מודלי שפה מפרקים טקסט ליחידות מידע ומזהים מספריים (IDs) מתוך מילון המודל.
 - התלמיד יסביר את מושג "חלון ההקשר" (Context Window) ומגבלות הזיכרון הנגזרות מכמות הטוקנים המקסימלית.
 - התלמיד יתאר מהי "מפת משמעויות" (Embeddings) וכיצד מילים מיוצגות ככתובות וקטוריות במרחב רב-ממדי שבו קרבה פיזית מעידה על קשר סמנטי.
 - התלמיד יסביר את עקרון הפעולה של מנגנון "הקשב העצמי" (Self-Attention) וכיצד הוא מאפשר לשנות את מיקום המילה במפת המשמעויות בצורה דינמית בהתאם לשאר המילים במשפט.

מטרות ביצועיות - תכנות בסביבת Python:

- התלמיד ידע להתקין ולייבא מודלים רב-לשוניים ייעודיים לעבודה עם מפת משמעויות של שפה טבעית בפיתוח.
- התלמיד ידע לממש פונקציה המחשבת מרחק סמנטי בין שתי מחרוזות ומחזירה ערך מספרי.
- התלמיד ידע לממש פרויקט המשלב לולאת ניחושים מותנית, מעקב אחר מונים (כמו מונה ניסיונות) והצגה גרפית דינמית של התקדמות (כגון שכפול תווים/אימוג'ים).

דרכי ההוראה:

1. העברת חלק ראשון (טוקנים) באמצעות מצגת ראשונה, וחקירה מעשית בדף עבודה "טוקניזר" (ניסוי גילוי שיאי טוקנים, השוואת שפות והשפעת אותיות גדולות).
 2. מעבר לחלק שני (מפת משמעותיות וקשב) באמצעות מצגת שניה, הדגמת מיקום מילים על מערכת צירים דו-ממדית והסבר על תזוזת המילים לפי הקשר.
 3. פיתוח משותף והדרגתי של משחק "סמנטעל" (Semantle) שלב אחר שלב: קודם לוגיקת קירבה בסיסית, לאחר מכן הוספת מונה ניסיונות, המרה לאחוזים ולבסוף עיצוב המדחום הגרפי עם אימוג'ים.
- מושגי מדעי המחשב / AI: טוקנים (Tokens), מפת משמעותיות (Embeddings), מרחב סמנטי רב-ממדי, מנגנון הקשב העצמי (Self-Attention), הקשר (Context)
- מושגים תכנותיים ב-Python: התקנת ספריות חיצוניות, מונה ניסיונות, המרה לאחוזים, לולאת ניחושים, הדפסה דינמית ושכפול תווים (*).

דרכי הערכה:

1. מילוי דף המעקב וביצוע הניסויים בטוקניזר.
2. השלמת השלד של פונקציית calculate_similarity בצורה תקינה.
3. הצגת משחק "סמנטעל" עובד המציג מונה ניסיונות מדויק ומדחום קירבה גרפי אינטראקטיבי.

חלוקת השעות לפרק 6:

שעות התנסות	שעות עיוניות	סך שעות	הנושא
2	1	3	טוקנים, טוקניזציה ומזהים, התנסות מעשית וכתובת טוקניזר פשוט
2	1	3	מפת משמעותיות, קשב עצמי, מרחק סמנטי ופרויקט "סמנטעל"
4	2	6	סה"כ שעות

פרק 7: - רשימה - טיפוס נתונים מורכב (list)

מטרת הפרק:

היכרות עם רשימה (מערך דינמי) כמבנה נתונים אבסטרקטי. היכרות עם הפשטה פרוצדורלית - הגדרת קופסה אלגוריתמית שחורה המממשת פעולה מופשטת ושימוש בקופסאות אלגוריתמיות שחורות. הכרת מבנה הבקרה של ביצוע חוזר מותנה.

מטרות ביצועיות - מדעי המחשב:

1. התלמיד ידע להסביר מהי רשימה ומה מטרתה.
2. התלמיד ידע להסביר את הפעולות הבסיסיות על רשימה (אורך רשימה, שייכות לרשימה, גישה למקום מסוים ברשימה, מחיקת פריט, הוספת פריט בסוף רשימה, הוספת פריט במקום מסוים ברשימה, עדכון פריט במקום מסוים ברשימה).
3. התלמיד ידע לפתח בגישה מודולרית פתרונות אלגוריתמיים המשתמשים ברשימה ובפעולות על רשימה.
4. התלמיד ידע להסביר מהי תבנית חיפוש סדרתי ברשימה ממוינת.
5. התלמיד ידע לפתח בגישה מודולרית פתרון אלגוריתמי המשתמש בתבנית סכימה.
6. התלמיד ידע לפתח בגישה מודולרית פתרון אלגוריתמי המשתמש בתבנית חיפוש.
7. התלמיד ידע להסביר מהי הפשטה פרוצדורלית ומה מטרתה.
8. התלמיד ידע לפתח בגישה מודולרית פתרונות אלגוריתמיים המשתמשים בהפשטה פרוצדורלית כקופסה שחורה (עם או בלי פרמטרים).
9. התלמיד ידע לכתוב אלגוריתמים למימוש קופסאות שחורות.
10. התלמיד ידע ליישם את ההליך של פתרון בעיות בפרדיגמה הפרוצדורלית תוך שימוש במושגים וברעיונות שנלמדו כדי לפתור בעיות אלגוריתמיות ברמות משתנות.

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשתמש ברשימה סדורה ובפעולות סטנדרטיות על רשימה סדורה (אורך רשימה, שייכות לרשימה, גישה למקום מסוים ברשימה, מחיקת פריט, הוספת פריט בסוף רשימה, הוספת פריט במקום מסוים ברשימה, עדכון פריט במקום מסוים ברשימה).
2. התלמיד ידע לממש תבנית חיפוש סדרתי ברשימה.
3. התלמיד ידע לממש הגדרת פעולה עם ובלי פרמטרים.
4. התלמיד ידע להשתמש בפעולה שהוגדרה ומומשה על ידו.
5. התלמיד ידע לממש פתרון לבעיה ע"י פרויקט המשתמש בהפשטה פרוצדורלית.

מושגי מדעי המחשב: מערך דינמי (רשימה) כמבנה נתונים אבסטרקטי. התייחסות ראשונית אל רשימה כישות שנושאת איתה תוכן ופעולות שניתן לבצע עליו. ביצוע חוזר מותנה. תבנית סכימה. תבנית חיפוש.

מושגים תכנותיים ב-Python: הוראת ביצוע חוזר מותנה. רשימה ופעולות (, append, count, split, join, extend, find, insert, pop, remove, reverse, sort). פעולות - הגדרה ושימוש.

דרכי ההוראה:

1. הפרק יילמד תוך ביצוע משימות מובנות ומשימות התנסות חופשית בסביבה. כל מושג, גם הפשוט ביותר, צריך להילמד בהקשר של פתרון בעיה מסוימת, בעקבות צורך שמתעורר במהלך פתרון הבעיה.
2. הצגת בעיה המזמנת שימוש באוסף סדור של ערכים.
3. פיתוח הדרגתי של פתרון אלגוריתמי לבעיה, שמבצע פעולות על רשימה סדורה ומימושו ב-Python.
4. הצגת בעיה המזמנת חיפוש אחר ערך באוסף סדור של איברים.
5. פיתוח פתרון אלגוריתמי לבעיה אשר משתמש בתבנית חיפוש ומימושו ב-Python.
6. הצגת בעיה המזמנת שימוש בהפשטה פרוצדורלית דרך קופסה שחורה.
7. פיתוח הדרגתי של פתרון אלגוריתמי לבעיה, אשר משתמש באבני בניין פרוצדורליות, עם ובלי פרמטרים.
8. מימוש הפתרון האלגוריתמי ב-Python.

דרכי הערכה:

1. מעקב אחר ביצוע אלגוריתמים נתונים הכוללים שימוש ברשימות.
2. זיהוי מטרתם של אלגוריתמים נתונים הכוללים רשימות.
3. פיתוח ומימוש של פתרון אלגוריתמי עבור בעיה המצריכה שימוש ברשימות.
4. מעקב אחר אלגוריתם אשר משתמש בהפשטה פרוצדורלית.
5. פיתוח ומימוש של פתרון אלגוריתמי עבור בעיה נתונה, אשר משתמש בהפשטה פרוצדורלית.

חלוקת השעות לפרק 7

הנושא	סך שעות	שעות עיוניות	שעות התנסות
מהי רשימה	2	1	1
פעולות על רשימה	3	1	2
סכימת איברים ברשימה	3	1	2
חיפוש סדרתי ברשימה	2	1	1
סה"כ שעות	10	4	6

פרק 8: סיווג

מטרת הפרק:

הבנת מושג הסיווג (קלסיפיקציה), ההבדל בין סיווג נאיבי מבוסס חוקים קוד לבין למידת מכונה, אימון, מבחן, ייצוא של מודל סיווג חזותי באמצעות Teachable Machine של גוגל, וייבוא של המודל חזרה לתוך קוד פייתון.

מטרות ביצועיות - AI:

1. התלמיד יגדיר מהי בעיית סיווג (Classification) ומהו "קו מפריד" (Decision Boundary) במרחב התכונות.
2. התלמיד יבחין בין תכונות (Features) יעילות ומבדילות לבין תכונות שאינן רלוונטיות למשימת הסיווג.
3. התלמיד יתאר את שלבי העבודה המרכזיים בלמידת מכונה: איסוף נתונים, חלוקה לסט אימון (Training set) וסט מבחן (Test set), ואימון המודל.
4. התלמיד יסביר את המושג "מדד הדיוק" (Accuracy) של מודל סיווג וידע לחשב אותו על סמך תוצאות הסיווג של סט המבחן.
5. התלמיד יזהה הטיות (Bias) פוטנציאליות במודלים חזותיים (כגון השפעות רקע, תאורה וזווית) ויציע דרכים להתגבר עליהן באמצעות תכנון נתונים מגוון.

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לממש סיווג "נאיבי" מבוסס חוקים מורכבים (if-else וקשרים לוגיים) בפייתון.
2. התלמיד יזהה וינמק את המגבלות של קוד חוקים קשיח בהתמודדות עם מקרי קצה ומצבים משתנים בעולם האמיתי (כמו בננה ירוקה וקשה).
3. התלמיד ידע להפעיל סביבת עבודה ייעודית לאימון מודלים, להגדיר קטגוריות, לאסוף ולארגן דוגמאות אימון, ולאמן את המודל.
4. התלמיד ידע לבחון את המודל המאומן בעזרת סט תמונות מבחן חיצוניות שלא שימשו לאימון, לחשב את מדד הדיוק שלו, ולייצא את קובצי המודל הסופיים לשימוש עתידי.
5. התלמיד ידע לייבא את קובצי המודל לתוך קוד פייתון ולשלב קוד נתון בשביל לעבוד עם המודל מתוך הקוד.

דרכי ההוראה:

1. הצגת נושא הסיווג וחקירת "תכונות טובות" דרך המצגת הראשונה ודף התרגול הראשון (סיווג שירים לפי שנה, וסיווג תפוח/בננה).
 2. התנסות ראשונית חווייתית ב-Teachable Machine ללא הכנה מראש, לשם למידה עצמאית וגילוי הטיות הרקע והתאורה (מצגת עזר, מצגת לתלמידים, ודף עבודה).
 3. תכנון מודע ובניית פרויקט סיווג מתקדם (אגרוף מול OK) תוך הגדרת מקרים מאתגרים מראש, איסוף +100 תמונות לקטגוריה, בדיקה מול 7 תמונות מבחן חיצוניות, וחישוב מדד הדיוק (מצגת ודף עבודה). לבסוף, חיבור המודל לפייתון.
 4. עבודה מעשית ושימוש באביזרים פיזיים (בריסטולים, כפפות, טושים) ליצירת סביבות צילום מגוונות בכיתה.
- מושגי מדעי המחשב / AI: סיווג (Classification), תכונה (Feature), תכונה טובה ומבדילה לסיווג, קו מפריד (Decision Boundary), מרחב תכונות, תהליך למידת מכונה, סט אימון (Training set), סט מבחן (Test set), אימון מודל (Model Training), מדד הדיוק (Accuracy), הטיות חזותיות (השפעת רקע, תאורה, זווית).
- מושגים תכנותיים ב-Python: סיווג נאיבי מבוסס תנאים, תנאים לוגיים מורכבים (if-else, and), רשימות מקבילות, פונקציית סיווג (def), קריאה לפעולה, שמירת מודל וייצוא (Savedmodel / zip), טעינת מודל ושימוש בקוד.

דרכי הערכה:

1. מעקב אחר אלגוריתם המבצע סיווג נאיבי מבוסס חוקים ומציאת מקרי קצה בהם מערכת החוקים נכשלת.
2. זיהוי מטרתו ותכונותיו של מודל סיווג נתון, ובחירת תכונות (Features) טובות ומבדילות להפרדה במרחב התכונות.
3. פיתוח ומימוש של פתרון אלגוריתמי (או תהליך למידת מכונה) הכולל איסוף נתונים מתוכנן, הגדרת קטגוריות, אימון מודל ובחינתו בעזרת סט מבחן.
4. חישוב מדד הדיוק (Accuracy) של מודל סיווג על סמך תוצאות זיהוי של נתוני מבחן חדשים.
5. ניתוח והסבר של הטיות (Bias) פוטנציאליות במודל חזותי והצעת דרכים מעשיות להתמודדות עמן (כגון גיוון הרקע והתאורה).

חלוקת השעות לפרק 8:

שעות התנסות	שעות עיוניות	סך שעות	הנושא
1	1	2	מבוא לסיווג, תכונות טובות וסיווג נאיבי בקוד פייתון
1	1	2	תהליך למידת מכונה (אימון ומבחן) והיכרות עם Teachable Machine
2	2	4	תכנון פרויקט סיווג (אגרוף/OK), התמודדות עם הטיות, חישוב מדד הדיוק, שמירת המודל וטעינה שלו לקוד פייתון
4	4	8	סה"כ שעות

פרק 9: מערכות המלצה

מטרת הפרק:

הבנת האלגוריתמים העומדים בבסיס מערכות המלצה, פיתוח מערכת המלצה מבוססת משתמשים מקצה לקצה בפייתון תוך איסוף נתונים מ-Google Forms, וקיום דיון ביקורתי בהשלכות חברתיות והאתיות שלהן.

מטרות ביצועיות - AI:

1. התלמיד יבחין בין שני סוגי מערכות המלצה מרכזיות: מערכת מבוססת תוכן (Content-based) ומערכת מבוססת משתמשים (Collaborative Filtering).
2. התלמיד יסביר כיצד מחושב מדד הדמיון בין משתמשים לשם איתור "תאום דיגיטלי" (השוואה לוגית של דירוגים משותפים תוך התעלמות מפריטים שאינם מוכרים).

3. התלמיד יגדיר ויסביר את המושגים "בועת סינון" (Filter Bubble) ו"תיבת תהודה" (Echo Chamber) ואת השפעתן על שבירת המציאות המשותפת וקיטוב חברתי.
4. התלמיד יסביר את "המודל העסקי של הקשב" וכלכלת תשומת הלב, ויציע כלים מעשיים לשליטה בפייד (כמו כפתור "לא מעניין" וחיפוש יזום).

מטרות ביצועיות - תכנות בסביבת Python:

1. התלמיד ידע לתכנן סקר איסוף נתונים דיגיטלי (Google Form) המבוסס על ערכים מוסכמים ומובנים שקוד פייתון יכול לקרוא.
2. התלמיד יממש פונקציה לחישוב מדד דמיון (Similarity Score) בין רשימות דירוגים של משתמשים שונים בעזרת לולאות ותנאים לוגיים מורכבים.
3. התלמיד יממש פונקציית המלצה המזהה פריטים שאינם מוכרים למשתמש הנוכחי אך אהובים ומומלצים על ידי התאום הדיגיטלי שלו.
4. התלמיד יכתוב תוכנית ראשית המשלבת קריאת נתונים, פנייה לפעולות החישוב והדפסת הפריט המומלץ למשתמש.

דרכי ההוראה:

1. פתיחה והכרות עם סוגי מערכות המלצה באמצעות מצגת ראשונה, וביצוע שלב א' של הפרויקט – בניית סקר Google Forms בנושא לבחירה (סדרות, טיולים, שירים) והפצתו בכיתה/קבוצה.
2. כתיבת קוד המערכת בפייתון (שיעור שני) – עבודה עם קובץ CSV המיובא ישירות מה-Google Sheet, והשלמת 3 הפעולות (קליטה, דמיון, המלצה) הנתונות בדף העבודה.
3. שיעור שלישי – דיון בסכנות דרך המצגת וביצוע הניסוי המעשי המרתק "בונים בועה ב-5 דקות" (בחלון גלישה בסתר ביוטיוב) להמחשת המהירות שבה האלגוריתם יוצר סביבת "קירות שקופים".

מושגי מדעי המחשב / AI: מערכת המלצה מבוססת תוכן (Content-based Filtering), מערכת המלצה מבוססת משתמשים (Collaborative Filtering), תאום דיגיטלי (Digital Twin), ציון דמיון (Similarity Score), דירוג משותף, בועת סינון (Filter Bubble), תיבת תהודה (Echo Chamber), המודל העסקי של הקשב, כלכלת תשומת הלב, הנדסת התמכרות, אחריות טכנולוגית.

מושגים תכנותיים ב-Python: לולאות על רשימות מקבילות, השלמת לוגיקה תנאית מורכבת, חיפוש והחזרת אינדקסים, תוכנית ראשית המשלבת קריאת נתונים וקריאה לפונקציות.

דרכי הערכה:

1. מעקב אחר אלגוריתם (באמצעות טבלת מעקב) המחשב מדד דמיון (Similarity Score) בין רשימות דירוגים של משתמשים שונים.
2. זיהוי מטרתו של אלגוריתם נתון או קטע קוד המשווה בין דירוגי משתמשים ומאתר את ה"תאום הדיגיטלי" בעל רמת הדמיון הגבוהה ביותר.
3. פיתוח ומימוש של פתרון אלגוריתמי למציאת פריט מומלץ אשר אינו מוכר למשתמש הנוכחי (דירוג "0") אך אהוב ומומלץ על ידי התאום הדיגיטלי שלו (דירוג "V").
4. **הבחנה** בין גישת המלצה מבוססת תוכן (Content-based) לבין גישת המלצה מבוססת משתמשים (Collaborative Filtering).
5. ניתוח ביקורתי של השלכות חברתיות ואתיות הנגרמות על ידי אלגוריתמי המלצה, לרבות זיהוי קיומן של "בועות סינון" ו"תיבות תהודה" והסבר מנגנון פעולתן.

חלוקת השעות לפרק 9:

שעות התנסות	שעות עיוניות	סך שעות	הנושא
1	1	2	מבוא למערכות המלצה, תכנון פרויקט ואיסוף נתונים באמצעות גוגל פורמס
2	1	3	תכנות מערכת ההמלצה בפייטון (מציאת תאום דיגיטלי והפקת המלצה)
1	2	3	מלכודות האלגוריתמים: ניסוי "בונים בועה ביוטיוב", דיון באתיקה ובמודל העסקי של הקשב
4	4	8	סה"כ שעות