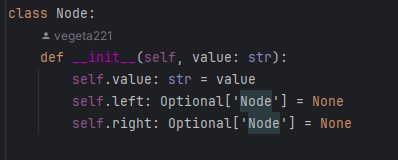
**מעבדה בבינה מלאכותית משימה 5**



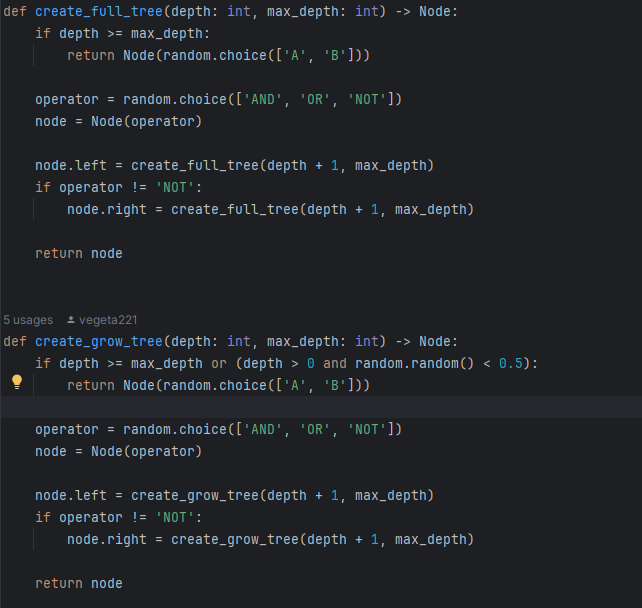
יצירת אוכלוסיה אקראית: אוכלוסיית הגנים (עצים) נוצרת באופן אקראי.

מבנה עץ: כל עץ מורכב מבחירה אקראית של אופרטורים (AND, OR, NOT) ואופרנדים (A, B) בסימון Prefix. צמתים פנימיים הם אופרטורים, ועלים הם אופרנדים.

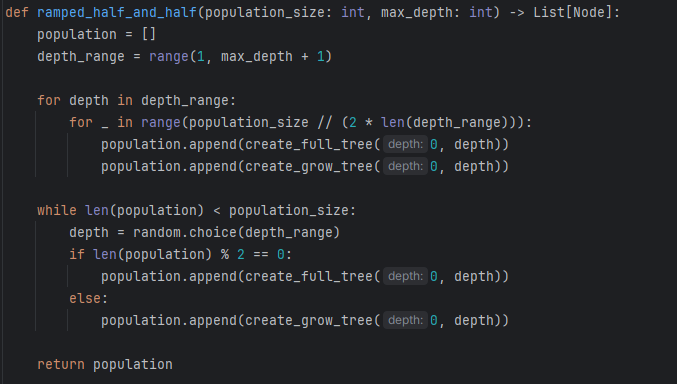
סימון Prefix: מבנה העץ עוקב מטבעו אחר Prefix מכיוון שהאופרטורים נמצאים בצמתים הפנימיים והאופרנדים נמצאים על העלים.

על מנת לענות על החלק הזה נכתב הקוד הבא:  


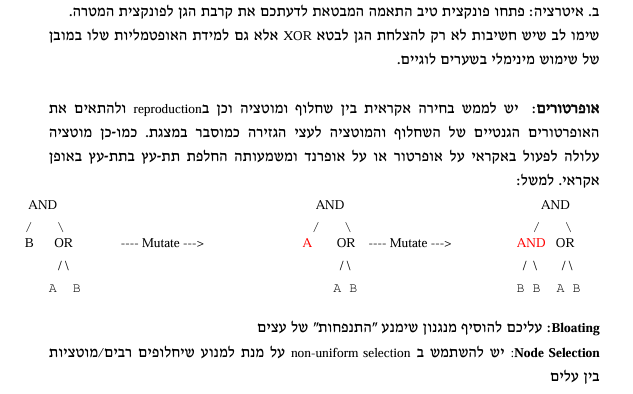
עץ רנדומלי:



Ramped Half-and-Half



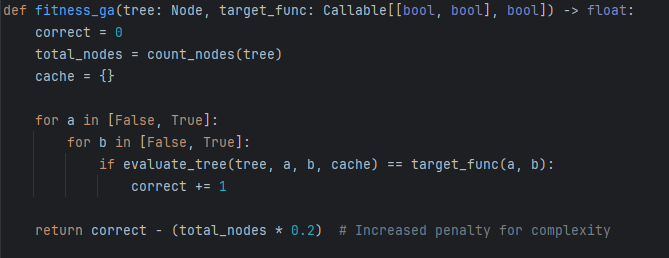
העומק נשלט לפי max\_depth ולא לפי הנדרש אך זוהי אותה הכוונה.



פונקציית הפיטנס: פונקציית הפיטנס מעריכה עד כמה גן (עץ) תואם את פונקציית המטרה (XOR במקרה זה). הוא גם מתייחס למורכבות של העץ כדי להעניש שימוש מופרז בשערים לוגיים.

פונקציית קרבה למטרה: פונקציית הפיטנס מודדת את נכונות העץ בהבעת פונקציית XOR.

אופטימליות (שימוש מינימלי בשערים לוגיים): פונקציית הפיטנס כוללת קנס על מספר הצמתים בעץ, ומעודדת עצים פשוטים יותר.



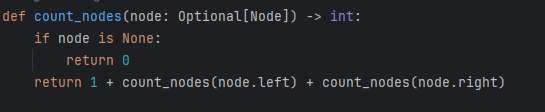
נכונות: פונקציית הפיטנס מחשבת את מספר התפוקות הנכונות על ידי השוואת תפוקת העץ לפונקציית המטרה עבור כל שילובי הקלט האפשריים של a ו-b.

מורכבות ענישה: הפונקציה סופרת את המספר הכולל של צמתים בעץ ומפחיתה עונש פרופורציונלי למספר הצמתים (total\_nodes \* 0.2).

כדי להעריך את העץ נכתבה הפונקציה הבאה:



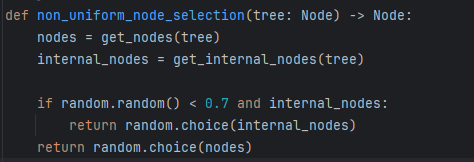
הפונקציה count\_nodes סופרת את המספר הכולל של צמתים בעץ, המשמשת בפונקציית הכושר כדי להעניש את המורכבות.

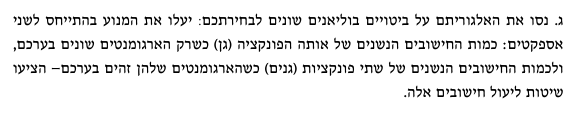


על מנת למנוע המצב של Bloating למעשה ניתן עונש מוגבר: לפונקציית הכושר יש כעת עונש מוגבר (total\_nodes \* 0.2) למספר הצמתים בעץ כדי למנוע התנפחות. יכלנו בפונקציית הפיטנס פשוט לעשות total\_nodes \* 0.1 וזה היה מספר אבל לא מונע התנפחות.

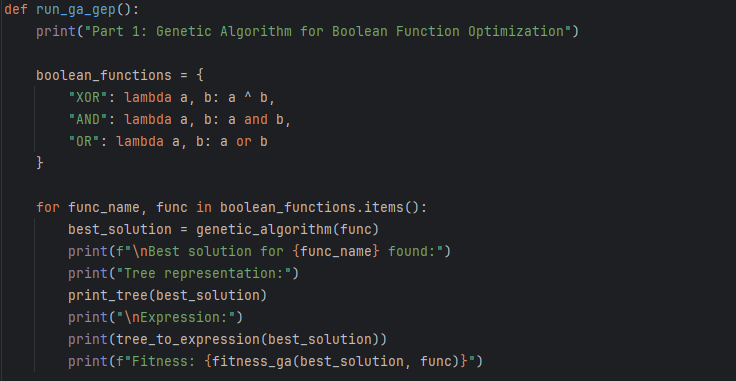
בחירה לא אחידה: פונקציה זו בוחרת צמתים עם הטיה לצמתים פנימיים, ומפחיתה את הסבירות למוטציות מוגזמות או הצלבות ברמת העלים.

הסתברות הבחירה מותאמת כדי להעדיף צמתים פנימיים 70% מהזמן (random.random() < 0.7), מה שמבטיח סבירות גבוהה יותר לשנות את מבנה העץ.





הפונקציה הראשית מטפלת באופציות רבות לניסוי:



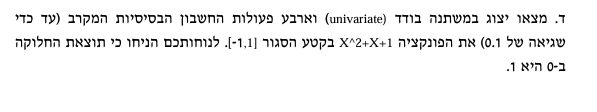
השיטות שניתן להציע הן:  
שינון: הפונקציה evaluate\_tree משתמשת מילון כדי לאחסן תוצאות שחושבו בעבר. זה מפחית חישובים מיותרים כאשר אותו תת-עץ מוערך מספר פעמים עם אותן כניסות.

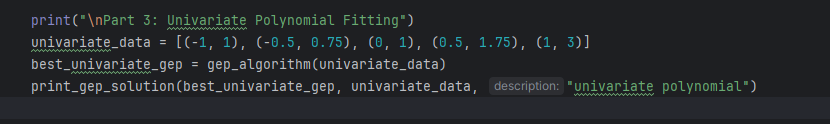
הורדות של תתי-ביטויים משותפים: על ידי שימוש בפונקציית tree\_to\_expression ליצירת מפתח ייחודי עבור כל תת-עץ, האלגוריתם יכול לזהות ולעשות שימוש חוזר בתוצאות של תת-ביטויים נפוצים.

בנוסף לאלו אפשר להשתמש בתור שיפור ב:  
הערכה מקבילית: על ידי מנוף עיבוד מקבילי כדי להעריך מספר פרטים באוכלוסייה במקביל. זה יכול להיות מועיל במיוחד באלגוריתמים אבולוציוניים שבהם הערכות הפיטנס אינן תלויות זו בזו.

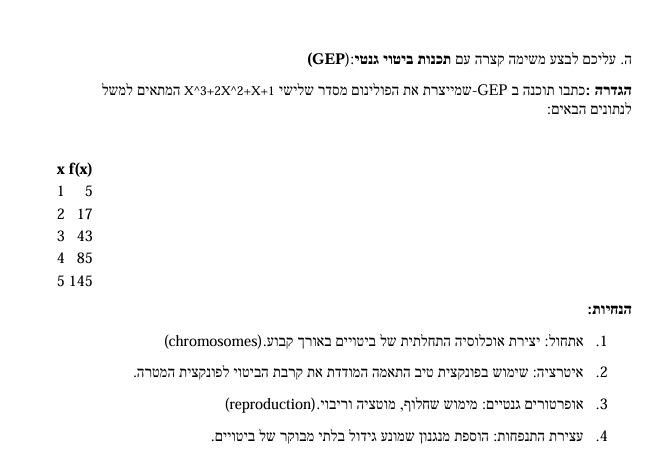
הערכה מצטברת: כאשר נעשה שינוי קטן באינדיבידואל (למשל באמצעות מוטציה או הצלבה), רק צריך להעריך מחדש את החלק שמושפע בעץ ולא את העץ כולו.

ישנן עוד אופציות רבות.

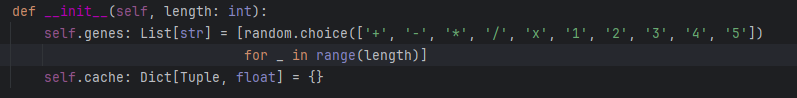


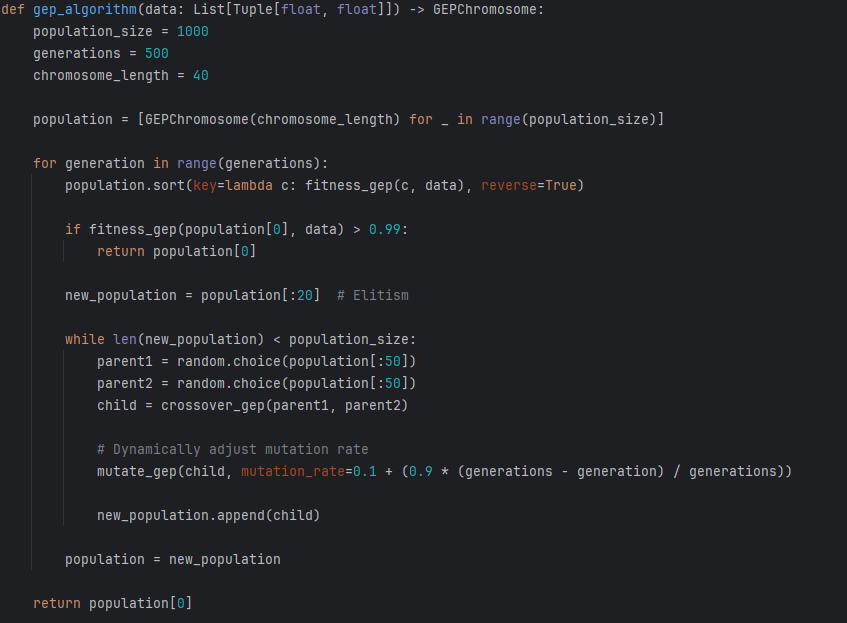
מבחינת חלק זה נכתב הקוד הבא שמטפל בו:  


זה בפונקציה הראשית.



1. שלב האתחול כולל יצירת אוכלוסייה ראשונית של כרומוזומים, כאשר כל כרומוזום הוא ביטוי באורך קבוע. אוכלוסייה זו תשמש כנקודת המוצא לאלגוריתם GEP.

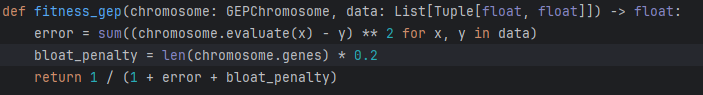


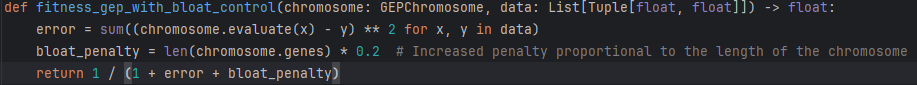
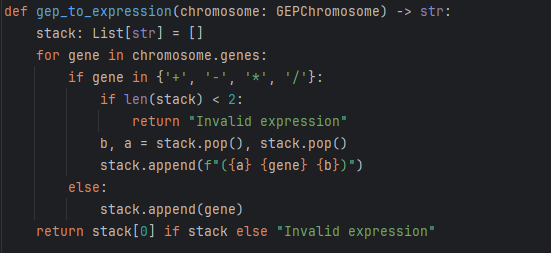


GEPChromosome: כל כרומוזום מיוצג על ידי מחלקת GEPChromosome, שמתחילה עם רשימה של גנים (ביטויים) באורך קבוע.

אתחול ב-gep\_algorithm: הפונקציה gep\_algorithm מאתחלת אוכלוסייה של מופעי GEPChromosome, כל אחד עם אורך קבוע (chromosome\_length).

1. שלב האיטרציה כולל הערכת פיטנס של כל כרומוזום באוכלוסייה. פונקציית הפיטנס מודדת עד כמה הביטוי (כרומוזום) תואם את פונקציית המטרה.



פונקציה fitness\_gep: פונקציה זו מחשבת את הפיטנס של כרומוזום על ידי השוואת הפלט שלו לערכי פונקציית המטרה בנקודות שונות. פרמטרים הנתונים מכיל tuples של ערכי קלט ואת ערכי פונקציית היעד המתאימים.

חישוב שגיאות: פונקציית הפיטנס מחשבת את סכום השגיאות בריבוע בין הכרומוזום המוערך וערכי היעד.

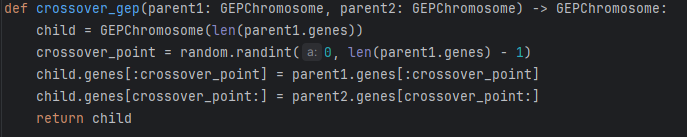
עונש Bloat: כדי למנוע נפיחות (צמיחה בלתי מבוקרת של ביטויים), מתווסף עונש פרופורציונלי לאורך הכרומוזום.

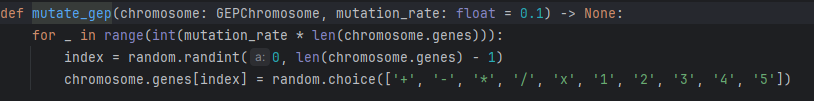
ציון פיטנס: ציון הפיטנס גבוה יותר עבור כרומוזומים עם שגיאה נמוכה יותר וגודל קטן יותר.

פונקציית fitness\_gep\_with\_bloat\_control: זוהי גרסה של פונקציית fitness\_gep המדגישה שליטה בנפיחות.

gep\_to\_expression: פונקציה זו ממירה את הגנים של כרומוזום לביטוי מתמטי הניתן לקריאה.

1. האופרטורים הגנטיים ממלאים תפקיד מכריע בהתפתחות האוכלוסייה לאורך דורות. הם כוללים הצלבה, מוטציה ורבייה (אליטיזם). אופרטורים אלו מציגים גיוון ומאפשרים לאוכלוסייה לחקור את מרחב הפתרונות בצורה יעילה.





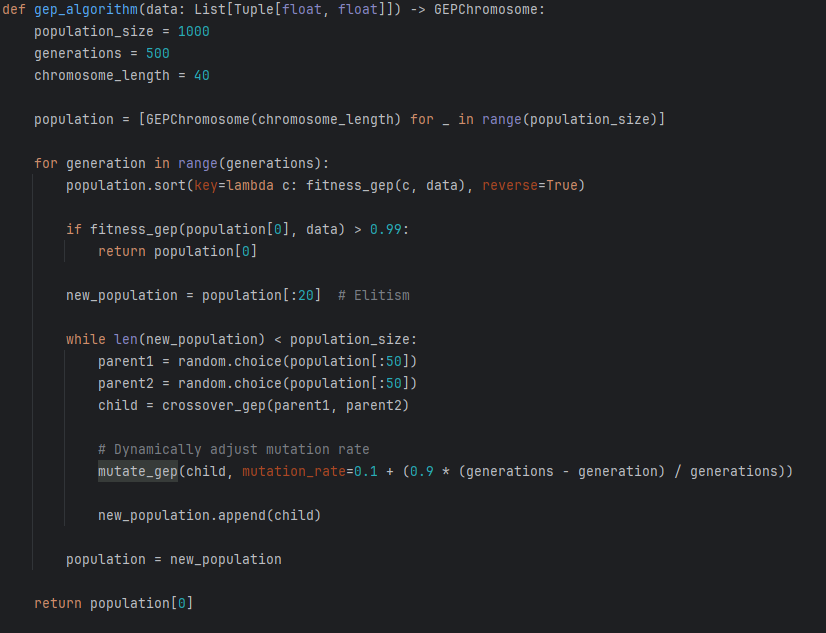
פונקציית הצלבה: פונקציית crossover\_gep בוחרת נקודת הצלבה אקראית ומשלבת את הגנים משני כרומוזומי אב ליצירת כרומוזום ילד.

crossover\_point: אינדקס אקראי בכרומוזום שבו נלקחים גנים מההורה הראשון לפני נקודה זו וגנים מההורה השני נלקחים לאחר נקודה זו.

פונקציית מוטציה: פונקציית mutate\_gep חוזרת על הגנים של כרומוזום ומעבירה מוטציה אקראית של אחוז מסוים מהם.

mutation\_rate: פרמטר זה שולט באחוז הגנים שיש לעבור מוטציה. הפונקציה חוזרת על חלק מהכרומוזום ומחליפה גנים נבחרים בגנים אקראיים חדשים.

רבייה (אליטיזם): בפונקציית gep\_algorithm, הכרומוזומים הטובים ביותר מועתקים ישירות לאוכלוסייה החדשה כדי להבטיח שימור של חומר גנטי איכותי.



1. טופל כבר בסעיף 2.