## Software fault prediction

Install and dependencies

הקוד כתוב כולו בפייתון ואני עובד על גרסא 2.7 של אנקונדה 64 ביט שאפשר להוריד פה:  
<https://www.continuum.io/downloads>

הקוד שלי נמצא כאן:  
<https://github.com/amir9979/Debugger>

לפני שמריצים בפעם הראשונה צריך לבצע התקנה, ההתקנה מתבצעת בפשטות על ידי הפקודה:

python setup.py develop

מתוך התיקייה learner.

בנוסף צריך לוודא שמותקן java על המחשב וjavadoc נמצא בpath ( ניתן לבדוק על ידי הרצת הפקודהwhere Javadoc )

## Run SFL

כדי להריץ את המערכת יש ליצור קובץ קונפיגורציה עבור הלמידה.

קובץ הקונפיגורציה מגדיר את הערכים הבאים:

workingDir - התיקייה המקומית שבה ישמר המידע

git - הנתיב לגיט של הפרוייקט ( נתיב לוקאלי בלבד)

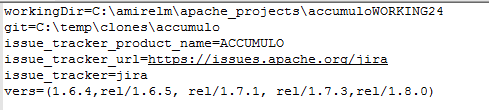
issue\_tracker - באיזה issue\_tracker להשתמש ( כרגע נתמכים bugzilla, jira, github)

issue\_tracker\_product\_name - מהו הרכיב עבורו רוצים להוריד את הבאגים

issue\_tracker\_url - הכתובת של האתר ממנו יש להוריד את נתוני הבאגים

vers - רשימת הגרסאות עבורן מתבצעת הלמידה

זהו קובץ קונפיגורציה לדוגמא עבור הפרוייקט [accumulo](https://github.com/apache/accumulo)



לאחר מכן אפשר להתחיל להריץ את השיטה על ידי הפקודה הבאה:

Python wrapper.py <Conf\_path> learn

## Results

לאחר סיום ההרצה נקבל קבצי arff תחת התיקייה weka בנתיב שהוגדר workingDir.  
קבצים אלו הם קבצי קלט עבור מערכת הלמידה weka. שמות הקבצים שיווצרו יהיו בפורמט הבא: {BugType}\_{LearningType}\_{Granularity}.arff

כאשר BugType יהיה All שמשמעותו היא שכל מה שהשתנה בcommit שמתקן את הבאג היה תקול וMost שמשמעותו היא שרק הרכיב ששונה הכי הרבה בcommit היה תקול.  
LearningType- training עבור קובץ האימון וtesting עבור קובץ הבדיקה  
Granularity- files כאשר מתיחסים לקבצים וmethods כאשר מתייחסים למתודות במחלקות