**Feature selection**

**מטרה**

הדגמת חשיבות ה-feature selection לביצועים. הבנת השיטות השוונת הקיימות וה-tradeoff ביניהן.

**תקציר**

במאמר זה מתואר שימוש בשתי שיטות feature selection למציאת פיצ'רים רלוונטיים אופטימליים עבור מודל עץ החלטה בודד על דאטה עם הרבה פיצ’רים לא רלוונטיים. בהשוואה נבחנו ההבדלים המרכזיים בין השיטות מבחינת אופטימליות הפיצ'רים שנמצאו וזמני הריצה. שימוש בשתי השיטות משפר את יעילות המודל, אולם אחת אינה feasible מבחינת זמן ריצה.

**רקע תיאורטי**

בחירת פיצ'רים הינה התהליך של לבחור את הפיצ'רים הרלוונטים מתוך מידע טבלאי עם הרבה פיצ'רים שונים, להורדת סיבוכיות המודל וזמן הריצה באימון וה-testing. שני סוגי שיטות שונות קיימות, והן שיטות פילטור (נותנות לכל פיצ'ר ציון בנפרד ומפלטרות את אלו עם ציונים נמוכים) ושיטות עוטפות (מחפשות קומבינציה של פיצ'רים ובוחרות את הטובה ביותר). בניסוי נבחנו רק שיטות עוטפות.

**מהלך ניסוי**

בניסוי נבחן מידע עם 100 פיצ'רים, מתוכם רק 5 רלוונטיים לפונקציית המטרה שעל המודל ללמוד:

y = 10sin(pi\*x1\*x2) + 20\*(x3-0.5)^2 + 10\*x4 +5\*x5 + e

כאשר e הוא רעש גאוסי נורמלי. 1000 דגימות ב-train,test כל אחד.

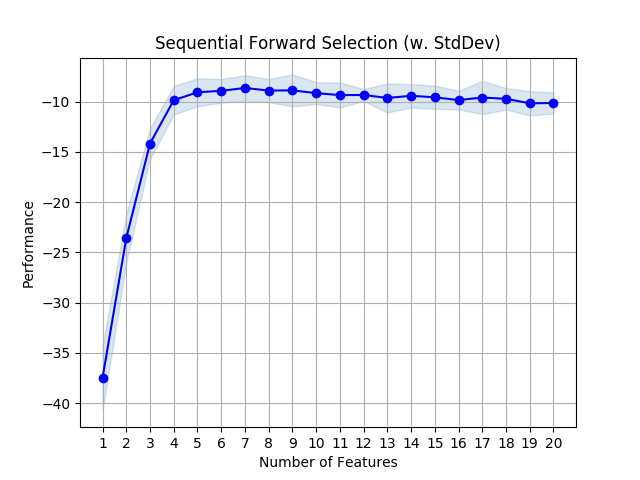
המודל שנבחר לשמש למטרת הרגרסיה הוא עץ החלטה בודד. עץ שכזה יכול ללמוד פונקציות שאינן לינאריות ולינאריות כאחת – ולכן נצפה שיוכל ללמוד את פונקציית המטרה עד כדי הרעש הגאוסי, ולספק MSE נמוך.

תחילה הותאם מודל העץ עם Cross Validation עם 5 folds. לאחר מכן נבחרה שיטת forward feature selection (מתוך ספריית הפייתון mlxtend) ובוצע חיפוש עבור 20 פיצ'רים מרכזיים. שיטת exhaustive search נבחנה לאחר מכן.

**תוצאות ניסוי**

עבור המודל הבסיסי עם כל הפיצ'רים התקבל MSE=16.8.

עבור שיטת ה-forward התקבלה שגיאה נמוכה (הגעה לרוויה) כבר עבור 5 features. מבחינה שלהם אלו אכן 5 הפיצ'רים הרלוונטיים לפונקציית המטרה. גרף המתאר את ה-Negative MSE למול מס' הפיצ'רים:



והגענו במצב זה ל-MSE=8.83 עם 7 features, אפשר לראות שנבחרו 2 פיצ'רים עודפים אם בוחרים לפי הציון הנמוך ביותר.

עבור שיטת ה-exhaustive search החיפוש כלל לא התכנס אפילו עבור 5 פיצ'רים לאחר 20 דקות – שיטה זו כלל אינה feasible.

**מסקנות ודיון**

בניסוי נבדקה חשיבות ה-feature selection לביצועי המודל שלנו על דאטה עם הרבה פיצ'רים שאינם רלוונטיים. הודגמה שיטת ה-forward feature selection, בעזרתה אכן נמצאו הפיצ'רים הרלוונטיים ושגיאת המודל ירדה משמעותית. 2 פיצ'רים עודפים נבחרו – לכן חשוב לשים לב כי גם שיטה זו עלולה לבצע overfit בהתאם לדאטה שאנו עובדים עליו, ונדרש לפתור overfit זה בשיטות שונות. כמו כן הוראה כי שימוש בשיטת ה-brute force למציאת הפיצ'רים אומנם תביא פיצ'רים אופטימליים – אך אינה אפשרית מבחינת זמני ריצה.