The election challenge – part 2

בחנו את המסווגים

* Decision Tree
* Perceptron
* Linear SVM OVO
* Linear SVM OVR

ניסינו להריץ אותם עם פרמטרים שונים ובחרנו בסופו של דבר לאתחל אותם באופן הבא:

* DecisionTreeClassifier(max\_depth=10)
* Perceptron(n\_iter=100)
* SVC(kernel="linear", C=1)
* LinearSVC(C=1)

גילינו שz-scaling משפר מאוד את הדיוק של הperceptron ו- Linear SVM OVR

ובמידה מסוימת גם את הדיוק של Linear SVM OVO.

ניסינו לשמור את ערכי ה-log של העמודות Yearly\_IncomeK ו- Yearly\_ExpensesK במקום הערכים המקוריים, אבל זה רק פגע בדיוק ולכן החלטנו לעבוד עם הערכים המקוריים.

ניסינו להוסיף מסווג נוסף שהוא שקלול של 3 המסווגים שהשיגו את התוצאות הטובות ביותר. אם יש הסכמה בין 2 מהם לוקחים את הערך שהם מסכימים עליו, ואם כל אחד אומר משהו אחר לוקחים את הערך מהמסווג שהשיג את הציון הכי גבוה. בסופו של דבר המסווג הזה השיג תוצאות פחות טובות מהמסווג שהשיג את התוצאה הכי טובה בשלב הראשון (Decision Tree).

מדידת ביצועים- לכל מסווג בדקנו באיזה אחוז מהמקרים הוא צודק (הפונקציה המובנית במסווגים- score()), ובנוסף בחנו עוד 2 מדדים- חישבנו לכל מפלגה את ה- sensitivity וה- precision. אם, לדוגמא, מפלגה גדולה רוצה לדעת איזה בוחרים הכי כדאי לה להסיע לקלפי ביום הבחירות, כדאי לעבוד עם המסווג שה- precisionשלו עבורה הוא הכי טוב (בהנחה שהיא לא יכולה להסיע את כולם, ועדיף לפספס מישהו שכן מצביע לה מאשר להסיע מישהו שלא מצביע לה). עבור מפלגה קטנה כדאי דווקא לבחור במסווג שה-sensitivity שלו הכי גבוה עבורה, מפני שבמילא אין לה הרבה מצביעים, ולכן יותר חשוב להביא כמה שיותר אנשים שמציביעם לה לקלפי, גם אם יהיו בתוכם אחוז יותר גבוה של אנשים שלא מצביעים לה. מכיוון שאנחנו לא מוכוונים למפלגה מסוימת, הדפסנו את הערך המינימלי בכל אחד מהמדדים האלה.