מבוא למערכות לומדות – תרגיל בית 5

מגישים:

שם: אסף חאייק ברוך **ת.ז:** 206783441

שם: בן בנוז **ת.ז:** 207570573

כללי:

באופן כללי בתרגיל הזה השתמשנו בעיקר בקוד הישן שלנו מתרגילים 2, 3 ו-4.
בגלל זה פשוט נציין בכל חלק של המטלה מה שינינו מהתרגילים המקוריים, וזאת בכדי להראות רק את מה שרלוונטי לתרגיל הזה ושלא ראיתם קודם.

בחירת סטי האימון, הולידציה, והמבחן

באזור הזה דבר אחד עיקרי השתנה – נתנו לנו סט מבחן חדש ולא מסווג שעליו היינו צריכים לעבוד.
בגלל זה כבר אין צורך בסט המבחן הישן שלנו, במיוחד בגלל שכבר השתמשנו בו בשביל לבדוק ביצועים בחלקים הקודמים בכל מקרה.
לכן הוספנו לסט האימון את סט המבחן הישן, כך שעכשיו הוא מכיל 85% מתוך המידע המסווג שניתן לנו בכל התרגילים הקודמים, וסט הולידציה מכיל כ-15% מאותו המידע, כמו בתרגילים הקודמים. סט המבחן כמובן מכיל את כל המידע החדש שניתן לנו בתרגיל הזה (הלא מסווג).

השלמת ערכים חסרים וגילוי outliers:

בחלק הזה היו לנו 2 שינויים שעשינו, אחד באופן ההשלמה והשני בערכים שאנחנו משתמשים בהם בהשלמה.

מבחינת אופן ההשלמה, מכיוון שקודם השתמשנו בתווית (הסיווג\המפלגה) של דוגמא במקרה שבו השלמנו ערך חסר לפי ממוצע (עשינו ממוצע\ערך הכי נפוץ לפיצ'ר מתוך אלו שיש להם את אותה התווית) לא יכולנו לעשות זאת על סט המבחן החדש, שאינו מסווג (בעיקרון גם לא היינו צריכים לעשות את זה כשהיה לו סיווג, אבל זה כבר תרגיל שעבר).
לכן שינינו את הקוד כך שבמקרים שבהם עושים השלמת ערך באופן הזה בסט המבחן ניקח ממוצע כולל ללא התייחסות לתווית.
במקרים שבהם אנו עושים השלמת ערכים לפי קורלציה של פיצ'רים לא היה שינוי.

מבחינת הערכים שאנחנו משתמשים בהם בהשלמה, מכיוון שנתנו לנו במשוב לתרגיל 2 את הרשימה של הפיצ'רים ואנו יודעים איזה פיצ'רים נדגמו באופן אקראי ואילו לא, החלטנו להשלים ערכים ולחפש ערכים לא חוקיים רק בפיצ'רים שאינם אקראיים – כלומר אלה שרלוונטיים לבעיית הסיווג ואלה שהם פונקציות שלהם (עדיין רלוונטיים, פשוט חולקים מידע עם אחרים).
זה מסיר רעש וזריקת ערכים במקומות שלא צריך.

בעיות הסיווג:

בחלק הזה לא שינינו שום דבר מבחינת האופן שהקוד מתפקד, רק 2 שינויים טכניים.

הראשון הוא שעשינו כיוונון מחדש של ההיפר-פרמטרים מכיוון שאנחנו עובדים עכשיו עם סט אימון יותר גדול ואפשר להגיע לערכים יותר טובים.
כמובן שלאחר הכיוונון בדקנו שוב לפי סט הולידציה איזה מהמסווגים מבין SVM ו-RandomForest נותן את התוצאות הכי טובות ובחרנו להשתמש בו על סט המבחן.
זה נעשה באופן אוטומטי כמו שעשינו בתרגיל 3, רק עם סט האימון המורחב.

השני היה שהקוד שלנו הדפיס השוואות של החיזויים על סט המבחן למציאות, אך הפעם סט המבחן שלנו לא היה מסווג ולכן היינו צריכים לוודא שהקוד לא יעשה אותן במקרה הזה.

ההיפר-פרמטרים החדשים שמצאנו לכל אחת מהבעיות הם:

בבעיית סיווג המצביעים קיבלנו מסווג מסוג RandomForest עם היפר-פרמטרים:

```
RandomForestClassifier(bootstrap=False, class_weight='balanced',
            criterion='gini', max_depth=60, max_features='sqrt',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=4,
            min_samples_split=10, min_weight_fraction_leaf=0.0,
            n_estimators=1027, n_jobs=None, oob_score=False,
            random_state=None, verbose=0, warm_start=False)
```

בבעיית מנצח הבחירות קיבלנו מסווג מסוג RandomForest עם היפר-פרמטרים:

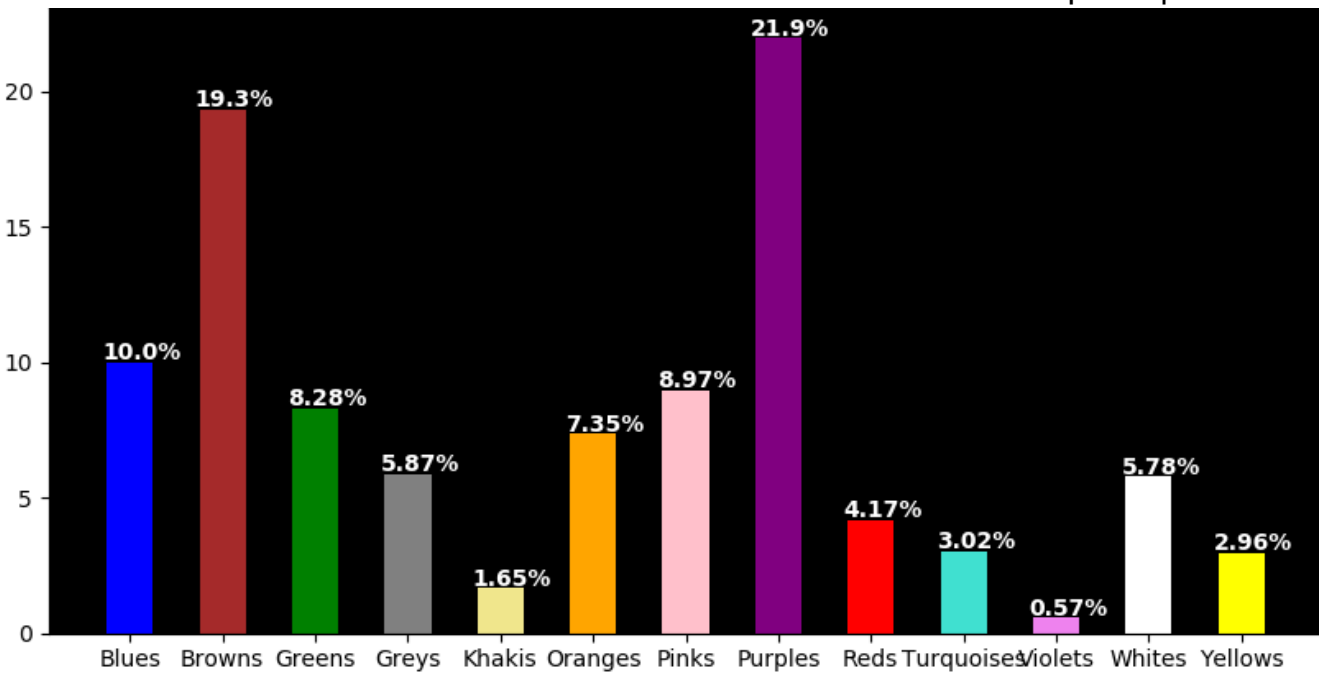
```
ElectionsWinnerWrapper(model=RandomForestClassifier(bootstrap=True, class_weight='balanced',
            criterion='gini', max_depth=None, max_features=None,
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=5, min_weight_fraction_leaf=0.0,
            n_estimators=1370, n_jobs=None, oob_score=False,
            random_state=None, verbose=0, warm_start=False))
```

בבעיית התפלגות הקולות בין המפלגות קיבלנו מסווג מסוג RandomForest עם היפר-פרמטרים:

```
ElectionsResultsWrapper(model=RandomForestClassifier(bootstrap=True, class_weight='balanced',
            criterion='gini', max_depth=None, max_features=None,
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=5, min_weight_fraction_leaf=0.0,
            n_estimators=1370, n_jobs=None, oob_score=False,
            random_state=None, verbose=0, warm_start=False))
```

בכל הבעיות הניקוד של 2 סוגי המסווגים היה קרוב, אבל תמיד RandomForest היה קצת יותר טוב.

תוצאות הסיווג של המצביעים שמורות בקובץ test_results.csv בפורמט שנדרש.
המפלגה שחזינו שתנצח היא מפלגת Purples.
התפלגות הקולות בין המפלגות היא:



יצירת קואליציה יציבה:

גם לחלק הזה לא היינו צריכים לשנות שום דבר משמעותי בקוד עצמו, אבל כן היינו צריכים להתייחס שונה לסט המבחן מכיוון שהוא ללא תוויות.
מכיוון שהקוד שלנו צריך תוויות בשביל ליצור קואליציה (אחרת איך נוכל לתרגם דמיון בין מצביעים לדמיון בין מפלגות?), השתמשנו בתוויות שחזינו עבור סט המבחן בחלק החיזוי.
כך יכולנו להריץ את הקוד שלנו מתרגיל 4, כאשר הקלט היה סט המבחן ביחד עם הסיווגים שלנו (ללא המזהה "IdentityCard_Num" כמובן).
שינוי נוסף שעשינו הוא תיקון באג שאינו משנה את אופן פעולת הקוד (רק מונע exception במקרה קצה מסויים).

כמו בתרגיל הקודם בדקנו דרך מתודת ה-clustering שלנו עם MiniBatchKMeans ו-BayesianGaussianMixture (GMM משופר), ובמתודה ה-generative שלנו עם LDA ו-QDA, ובחרנו את הקואליציה עם ניקוד ה-Davies-Bouldin הכי נמוך (הכי טוב) מתוך ה-4 שנוצרו.

הניקוד שיצא לנו בכל מקרה הוא:

MiniBatchKMeans: 4.59, BayesianGaussianMixture: 3.144, LDA: 3.965, QDA: 3.253.

הקואליציה הכי טובה נוצרה ע"י BayesianGaussianMixture והיא מכילה את המפלגות הבאות:

Blues, Greens, Greys, Khakis, Oranges, Pinks, Purples, Reds, Turquoises, Whites, Yellows.

פה ניתן לראות את הקואליציה ואת האופוזיציה לפי PCA בתלת מימד: