

סיכום עבודת גמר בהדמיית נתונים – חיים גולדפישר

חלק א':

• מחברת 1: שיפור מחברת מסמסטר א' (סיווג יין לבן\אדום לפי צבע) -

במחברת זאת השתמשתי בעיקר בשיטות שלמדנו לאורך הסמסטר מבלי לבצע עבודה המצריכה חשיבה יצירתית ע"ג הדאטה. המודל בסמסטר שעבר היה מוצלח. השגתי בו הצלחה של 98.7% ע"י שימוש ב KNN. הפעם הורדתי מימדים ע"י שימוש ב PCA, שהוריד לי את כמות המימדים למספר יותר נמוך. מכאן נותר להשתמש במודלים של ensemble learning.

השימוש במודל Catboost שהוא מודל מסוג gradient boosting נתן תוצאה נהדרת של 95.5% הצלחה תוך שימוש ב 9 פיצ'רים מתוך 12 ובכך בעצם נתן הוכחה לכך שלרוב, המודלים של ensemble learning עדיפים על המודלים שנלמדו בסמסטר שעבר.

עיקר המטרה במחברת זאת היה להציג חלק מן היכולות שרכשתי לאורך הסמסטר, ולכן לא התעכבתי יותר מדי בהעלאת הדיוק מעבר למה שקיבלתי.

• מחברת 2: Fashion Mnist -

במחברת זאת האתגר היה להבין כיצד עליי לסדר את הדאטה כך שע"י שימוש בכמה שפחות פיצ'רים נוכל לקבל תוצאות מספקות דיין ע"מ שהמודל יהיה גם קומפקטי וגם יעיל. חשוב לציין שבמקרה זה, הדאטה היה מאורגן היטב וכל התמונות סודרו באותו מיקום ע"ג התמונה, כך שהיה נוח לעבוד עם הדאטה מבלי להצטרך לגעת בו. ניסיתי לבצע בתחילה כל מיני הורדות של פיקסלים ריקים בכל התמונות. אך לאחר מכן הבנתי שהדבר מיותר משום ש PCA עושה את אותו הדבר ביעילות מרובה.

מודל Voting עם ההיפר פרמטר soft voting היה המודל המוצלח ביותר ע"י שימוש במספר מודלים - Catboost, XG Boosting, Random Forest. חשוב לציין ש Catboost היה הטוב ביותר מבין כולם, אך לא עבור כל פרמטר. מסתבר שיש לייבלים שלשני המודלים הנוספים יותר פשוט לסווג מאשר ל Catboost. הדבר היה די פשוט לזיהוי ע"י הסתכלות ב classification report והשוואת אחוזי דיוק פר לייבל מתוך עשרת הלייבלים. עובדה זו מסקרנת מאוד כשלעצמה משום ששלושתם בסופו של דבר מודלים המסתמכים על עצים.

אחוז הדיוק הסופי היה מספק, אך לא יכולתי לבצע grid search משום שהתהליך נמשך יותר מדי זמן, ולא הסתיים גם לאחר שעות רבות של הרצה. אני בטוח שעם אמצעים מתאימים (מעבד, שרת) הייתי מגיע לתוצאות טובות בהרבה.

• מחברת 3: Dogs VS Cats -

במחברת זו, עיקר האתגר שחוויתי היה בדרך שבה העלתי את הדאטה. כלומר מהו גודל התמונה האידיאלי ע"מ שהמודל יהיה יעיל מצד אחד, אך לא עמוס מדי מצד שני - **העלתי 45*45**. הייתה לי דילמה מרכזית האם להעלות תמונות בשחור לבן, או להעלות תמונות צבעוניות אך "לשלם" באמצעות שילוש כמות הפיצ'רים בעקבות ערכי RGB - **בחרתי להעלות**

תמונות צבעוניות, מתוך מחשבה שבPCA גם ככה רוב הפיצ'רים ירדו, ואז אשאר רק אם הפיצ'רים החשובים באמת. הרי אם הצבע בפיקסל כלשהו לא ירד בהליך הPCA, כנראה שהוא חשוב דיו כדי לשפר את המודל. אחד מהדברים שסיקרנו אותי במיוחד והובילו אותי לאורך ניסיון סידור הדאטה, היה להבין מה יש בשיטת למידה עמוקה שמוצלח כ"כ בהקשרי עיבוד תמונה וחיזוי תמונות ע"י רשת נוירונים (הצלחה של 97%). המסקנה שלי לאחר קריאה בנושא הייתה שאם אבצע מניפולציות כלשהן ע"ג הדאטה עצמו, ע"י הסטות וסיבובים כלשהם של התמונות, אוכל לעזור למודל שלי לכסות כמה שיותר מקרים ובכך להגדיל את אפשרויות החיזוי שלו (מבלי להצטרך להבין לעומק עיבוד תמונה). אחרי מספר נסיונות (היפוכים, הפיכות וכו' - מוזכר במחברת) החלטתי להכפיל את הדאטה ע"י הוספת תמונות מראה של התמונות עצמן, כך שהמודל יצליח לסווג גם כאשר בעל החיים לא נמצא בדיוק באותו מיקום בתמונה.

שלב הPCA והמודלים יחסית זהים לאלו שב Fashion Mnist. מודל Catboost היה המוצלח ביותר מבין כל המודלים שניסיתי. כמו במודל הקודם, לא הצלחתי לבצע אופטימיזציה של היפרפרמטרים משום שהמחשב שלי לא הצליח להשלים את תהליך ה grid search. לכן עשיתי קרוס ולידציה מבלי להריץ על יותר מדי היפרפרמטרים. הייתה בעיה להשתמש בקובץ הטסט משום שהוא נטול לייבלים. על כן, בתחילת העבודה העברתי 1000 תמונות של חתולים ו 1000 תמונות של כלבים עם label לתיקיית 'test' אשר היתה טסט עבור המודל שלי. כלומר ה train הכיל 23,000 תמונות וה test הכיל 2,000 תמונות. המודל היה יחסית מוצלח בעיניי והגיע ל 70.15% דיוק לאחר קרוס ולידציה על הtrain.

חלק ב':

● מחברת 4: למידת מכונה עבור סיווג תנועות ידיים -

במחברת זו, ההליך המרכזי והמשמעותי ביותר היה ללא ספק העלאת הדאטה וסידורו. לפני שהתחלתי להעלות את הדאטה, הסרתי את הקבצים המיותרים מכל תיקייה כך שרק שלושת הקבצים האחרונים ישארו. בניתי לשם העלאת הקבצים שלוש פונקציות, הראשונה והמרכזית העלתה בנפרד כל CSV, נתנה לו לייבל (מתוך שלוש אפשרויות) ושם הנסיין עפ"י שם התיקייה והקובץ בהתאמה. כמו כן, הפונקציה ניקתה ממנו את 7 השניות הראשונות וחיברה את כל ה DF הללו ל DF אחד. הפונקציה כללה בתוכה עוד שתי פונקציות פנימיות. אחת מוסיפה לקבצי Alone את HandRight, ע"י בחירת גודל ה DF הקטן יותר והוספת id, Time של כל ערכי HandRight אל Alonen. הפונקציה השנייה מסדרת את הדאטה כך שכל שורת יד ימין תתחבר עם שורת יד שמאל לאותה השורה באמצעות merge ע"ב פרמטרים משותפים (למשל: Time, id). כמו כן, הפונקציה מסירה יד ללא בת זוג ובוחרת מכל שנייה בדאטה 5 דגימות. הדבר נובע מכך שיש בכל שנייה 60 צילומים. לכן ע"י בחירת 12% דגימות מתוך הדאטה, ניתן לקבל 5 דגימות מתוך 60 פר שנייה.

אחת הבעיות המרכזיות בהן נתקלתי הייתה אי הלימה משמעותית בין תוצאות המודל שלי (95%) לתוצאות המודל על הטסט (89%). לאחר מחשבה הגעתי למסקנה שעשיתי טעות בכך שהמודל שלי ראה את כל האנשים שיש בtrain, ולכן הצליח לסווג אותם בקלות בvalidation שלי. כאשר המודל נתקל ב test באנשים שטרם ראה, הוא התקשה לסווג אותם. הסקתי מכך שאם הייתי עושה את ה train_test_split כך שב X_train, לא יופיעו חלק מהנסיינים, המודל שלי היה לומד לסווג טוב יותר אנשים שטרם ראה. ניסיתי לקחת טסט באופן ידני (10 שניות אחרונות מכל ניסוי), אבל זה לא שיפר את המודל, ולכן ויתרתי על הרעיון. כמו כן, חשוב לי לציין

כי מאחר וההוראה הייתה להוסיף את HandRight עבור כל Alone, הליך הזיהוי של Alone הופך למשחק די מכור. פשוט מאוד להגיע ל100% הצלחה עבור label זה.

מספר נקודות אחרונות:

1. מודל CatBoost - כפי שניתן לראות, לאורך כל העבודה השתמשתי רבות בספרייה וקיבלתי תוצאות נהדרות (קיבלתי על כך אישור, למרות שלא למדנו את הספרייה הזאת במסטר).
2. ספריית Plotly - התקנת CatBoost דורשת גם את התקנת ספריית Plotly, משום שלמשל, כאשר מריצים grid search על CatBoost, תוך כדי הרצה מופיע גרף של Plotly אשר מציג את השפעת הפרמטר הנתון על תוצאות המודל, כולל הצגה גרפית של הפרמטרים הטובים ביותר (מתעדכן תוך כדי ריצה). מתוך עניין בדקתי אילו גרפים יש ל Plotly להציע וגיליתי שיש שם גרפים מעולים, יפים בהרבה משל matplotlib ו seaborn, וכן מציגים את הדאטה בצורה בהירה מאוד. דבר נוסף הוא שהרצת גרפים די מורכבים נעשים תוך כתיבת שורות קוד בודדות בלבד. ניתן לעשות מגוון פעולות על גרפים של Plotly, החל מהיכולת להניח את העכבר על הגרף ולקבל פרטים שונים על הדאטה, ועד האפשרות לנוע בתוך גרף תלת מימדי. מאחר ולא ניתן להעלות את הגרפים של Plotly לגיטהאב, הם עלו בתור תמונות png. הגרפים המקוריים מופיעים ב Kaggle (מצורפים לינקים ב ReadMe ובתוך כל אחת מהמחברות).
3. K-Means - הדבר אמנם לא מצויין בתוך המחברות עצמן, אך ניסיתי לאורך כל העבודה לעשות מגוון שימושים ב k-means, הן בשלב ניתוח הדאטה ע"י יצירת פיצ'רים (הרצת האלגוריתם על הדאטה ללא הלייבלים, חיזוי בהתאם לn_clusters, והוספת העמודה לדאטה), והן ע"י שילוב האלגוריתם במודלים (באמצעות pipeline). לא משנה מה ניסיתי לעשות, kmeans במקרה הטוב לא השפיע על תוצאות המודל ובמקרה הגרוע רק גרע אותן. לכן, ויתרתי על הרעיון לשלב אותו במחברות.