<u>קורס 'מבוא ללמידת מכונה'</u>

<u>דו"ח פרויקט</u>

מגישים: מיכאל לוינגר 204944144 רן טוכמן 201631678

תקציר מנהלים

בפרויקט זה בנינו מודלים אשר מטרתם לסווג רשומות לשתי קטגוריות (0/1) על סמך מספר פיצ'רים בסט נתונים (בעיית Binary Classification).

בשלב הראשון של הפרויקט חקרנו את סט הנתונים (שלב האקספלורציה) בכדי לקבל תפיסה רחבה יותר על המידע אשר בידינו. בעזרת הפונקציות השונות גילינו את האופי שבו כל פיצ'ר (עמודות המשתנים המסבירים) מתפלג, את ההתנהגות הקורלטיבית בין הפיצ'רים, נתונים סטטיסטיים על הפיצ'רים ועוד. בשלב זה של הפרויקט השתמשנו בפונקציות אשר מציגות את הנתונים בשלל וריאציות. ביצענו ויזואליזציה - ייצוג מידע בדרך סכמתית או ציורית, כולל פירוט המשתנים השונים ויחידות המידע.

השלב השני של הפרויקט עסק בעיבוד המקדים של סט הנתונים (pre processing) . טיפלנו במידע חסר, במידע לא ברור, הסרנו נתונים חריגים(outliers) , טיפלנו בנתונים קטגוריאליים וביצענו סטנדרטיזציה לנתונים. שלב עיבוד הנתונים נחוץ על מנת לאפשר בשלבים הבאים להסיק מסקנות מדויקות ואמינות.

בשלב השלישי בחרנו מודלים אשר על ידי החלתם על סט הנתונים בנינו תחזיות בינריות. כלומר, בשלב זה למעשה הרכבנו את ליבו של הפרויקט - בנינו מודלים אשר מטרתם לסווג רשומות לשתי קטגוריות (0/1) על סמך מספר פיצ'רים בסט נתונים מסוים. השתמשנו במודלים Naïve Bayes Classifier , KNN, Feed Forward Neural Network, Decision Tree

השלב האחרון של הפרויקט עסק בהערכת המודלים. בעזרת הפונקציות השונות הערכנו את טיב התחזיות וקיבלנו תפיסה לרמת הצלחתנו במשימה.

חלק א

בשלב האקספלורציה השתמשנו בפונקציות שונות בכדי לקבל מבט רחב על אופי הנתונים והאופן בו הם מתפלגים ובעיקר ביצענו שימוש בוויזואליזציה של הנתונים. ניתן לראות בנספח המצורף את הפונקציות השונות, הסכמות, וכלל פירוט המשתנים השונים ויחידות המידע. (בנספח המצורף רשמנו סיכומים והערות לגבי האופן בו הנתונים מתפלגים)

חלק ב - שלב העיבוד המקדים

ראשית חשוב לציין שבכדי לייצר קוד יותר מודולרי לחלק זה ולהבא אחריו בחרנו ליישם אותו בקובץ (.py) נפרד. הטמענו מחלקה (Preprocessingpipeline) אשר מטרתה לעבד את המידע. בחלק זה בצענו פעולות שונות והנחנו הנחות מסוימות, להלן הפירוט:

- בכדי לנקות מידע לא ברור, כמו 'unknown' או '?' השתמשנו בפונקציה בשם handle_unknown_features()

- הנתונים לא היו מנורמלים ולכן ביצענו סטנדרטיזציה לנתונים בעזרת feature_scaling(). כפי שלמדנו, המסווגים משתמשים בשיטות שונות (מרחק מנהטן, מרחק אוקלידי ועוד) לחישוב המרחקים בין המסווג לפיסות המידע ועל ידי ניתוח מרחקים אלו מבססים את המודל. כאשר הנתונים אינם מנורמלים לאותו טווח ערכים עלול הדבר להוביל לכך שלנתון אחד יהיה השפעה רבה יותר מנתון אחר ללא סיבה מוצדקת. לכן, יש צורך משמעותי לנרמול הנתונים על מנת לאפשר ניתוח נתונים בעל יכולת חיזוי מספקת.

- בנוגע לממדיות הבעיה, למדנו שבקירוב צריך שתהיה לפחת כמות דגימות של n בריבוע כאשר n מייצג את כמות הפיצ'רים. קירוב זה מתקיים ולכן לפי הבנתינו ממדיות הבעיה איננה קטנה מדי. במידה וקיימות עמודות אשר לא מועילות לנו כן עדיף להסיר אותם על מנת לחסוך במשאבים לעיבוד המידע. עוד בשלב הוויזואליזציה היה ניתן להבחין בקורלציה שבין צמדי העמודות הבאים - (14,13), (1,13), (1,14), (9,11) ואכן ה FeatureSelector (כלי שבעזרתו הבחנו אילו הן עמודות אשר מוסיפות הכי פחות מידע לתהליך החיזוי) המליץ להסיר את העמודות 11,13,14.
בכדי למפות את העמודות אשר נותנות הכי פחות מידע ביחס לשאר העמודות השתמשנו במתודות הבאות:

- identify_collinear() - מתבסס על הקורלציה שבין העמודות השונות
- identify_zero_importance() - מתבסס על מודל הלמידה (GBM)
- identify_low_importance() - (PCA model - percentage of the variance 95%)

בנוסף, בהקשר להורדת העמודות ה'מיותרות' , חשוב לציין כי בעזרת cross validation ביצענו הערכה על מנת לבחון את ההבדלים בין ביצועי המודל על הסט ה'מלא' אל מול סט הנתונים ה'חלקי'. לאחר בדיקה, התוצאות היו דומות ולכן ניתן להסיק שהסרת המידע לא פגעה בניתוח המידע.

- הסרנו נתונים חריגים(outliers) על בסיס ה'Z-score ' של כל ערך בעמודה בהתאם לתוחלת ולשונות העמודה.

- מתוך הנחה כי יש צורך לשנות את הנתונים ה'קטגוריאליים' לנומריים, בעזרת one hot encoding המרנו בצורה נכונה את הנתונים באופן הרצוי.

- טיפול במידע חסר – עבור הנתונים החסרים בעמודות עם המשתנים הנומריים בחרנו למלא את החסר עם ממוצע העמודה. עבור נתונים קטגוריאליים בחרנו למלא את החסר באותו באופן לאחר שהמרנו את המשתנים הקטגוריאליים לנומריים.

• עוד בשלב האקספלורציה הבחנו כי בסט הנתונים ישנם הרבה יותר שורות אשר מסווגות כ0-ים בעמודת המשתנה התלוי מאשר שורות אשר מסווגות כ1-ים. לכן, בכדי לאזן את סט הנתונים השתמשנו בשיטה הנקראת under sampling ובכך איזנו את סט הנתונים. (לאחר בדיקה ולאחר הורדת עמודות מיותרות בדקנו ואכן ישנם מספיק נתונים לפי הכלל ההיוריסטי n בריבוע נתונים לכל n עמודות) .

אם לא היינו מבצעים את הדגימה מחדש אומנם היינו מקבלים accuracy כולל גבוה יותר ממה שקיבלנו, אך תוצאה זו הייתה משקרת שכן יש הרבה יותר 0-ים מ1-ים ולכן המודל היה מסווג טוב מאוד 0-ים ורע מאוד 1-ים. במקרה בו סיווג ה1-ים קריטי, כמו לדוגמא בזיהוי סרטן, זהו מצב שאנו לא יכולים להרשות.

חלק ג - שלב הרצת המודלים

כמו בחלק הקודם, גם כאן בחרנו לממש חלק מהקוד בקובץ py. הנקרא LearningModels.py. בקובץ הנ"ל מימשנו מחלקה ()RunUtils שתעזור לנו לחלק את סט הנתונים לtrain, validation וכן ליצור ולהדפיס למסך confusion_matrix וגרפים נוספים.

מתוך האפשרויות בחרנו במודלים הבאים :
Support Vector Machine, Naïve Bayes classifier,
Logistic Regression, Artificial neural network
בכדי למצוא את ה 'הייפר פרמטרים' האופטימליים, הפעלנו grid search עבור כל מודל. לאחר שקיבלנו את הפרמטרים, הטמענו אותם בארבעת המודלים.

במידה והיינו מנסים לעבור על כל הקומבינציות האפשריות עבור ה 'הייפר פרמטרים' במודל Artificial neural network זה היה לוקח הרבה מאוד זמן ולכן הרצנו את ה grid search בכמה איטרציות ועל ידי כך, בכל שלב באופן מודולרי נשארנו עם הפרמטרים הטובים ביותר. לבסוף, בחרנו להשתמש בפרטרים הבאים:

- נבחר להשתמש ב3 שכבות של 50 נוירונים.
- Dropout Rate 12% - drop out היא שיטת רגולריזציה שמכבה באופן רנדומלי נוירונים (במקרה שלנו 12%) בכל epoch ובכך מאלצת את הרשת לא להסתמך יתר על המידה על נוירונים ספציפיים. שיטה זו עוזרת לשפר את יכולת ההכללה.
- נבחר לבצע שימוש ב - Adam max(https://arxiv.org/abs/1412.6980) , שיטה לאופטימיזציה סטוכסטית שבדרך כלל נותנת תוצאות טובות עבור רשתות נוירונים מלאכותיות.
- נבצע שימוש בפונקציית האקטיבציה 'SoftPlus' על פי תוצאה של grid search.
- לאופן שבו מאתחלים את המשקולות יש חשיבות רבה ולכן השתמשנו ב Xavier uniform initializer - אתחול משקולות לפי התפלגות נורמלית עם תוחלת 0 ושונות אשר תלויה בכמות הנוירונים (number of incoming neurons).

- פונקציית ה loss נבחרה באופן טיבעי להיות binary cross entropy.

חלק ד שלב הערכת המודלים

- בכדי לתאר את הצלחת המודלים השונים יישמנו confusion matrix עבור כל אחד מהם. כל אחד מתאי המטריצה מציג אחוז הצלחה מסוים בהקשר לתחזית מול תוצאות האמת. התא השמאלי העליון TP מציג את אחוז הפעמים בהם חזינו תוצאה חיובית (1) כאשר הנתון "המציאותי" אכן היה חיובי. התא הימני עליון FP מציג את אחוז הפעמים בהם חזינו תוצאה חיובית כאשר התוצאה האמיתית הייתה למעשה שלילית (0).
התא השמאלי תחתון FN מציג את אחוז הפעמים בהן חזינו כי התוצאה תהיה שלילית כאשר בפועל הייתה חיובית. התא הימני התחתון TN מציג את אחוז הפעמים בהן חזינו שהתוצאה שלילית (0) ואכן הייתה כזאת.
לדוגמא, עבור מודל ANN אשר בנינו המטריצה הניבה את התוצאות הבאות:
TP = 0.8916, FP = 0.1084, TN = 0.7800, FN = 0.2200

- מתוך בחינה של פערי הביצועים בין הרצת המודל הנבחר ANN על ה Train או על הValidation מן הגרף אשר מציג את רמת הדיוק (accuracy) ביחס למספר ההרצות עבור ה train set, validation set - ניתן לראות כי אין פערים משמעותיים בין הביצועים ולכן להסיק כי המודל אכן אמין ואין אובר פיטינג, כלומר יכולת ההכללה של המודל מספקת. ברשת הנוירונים השתמשנו בשיטת רגולריזציה בשם dropout אשר רנדומלית מתעלמת מחלק מן הנוירונים ובכך נוצר מצב שבו הרשת לא מסתמכת יותר מדי על נוירונים ספציפיים מה שמשפר את יכולת ההכללה ומונע אובר פיטינג.

- בהקשר של בניית פלט ROC על כל Fold-K עבור כל אחד מהמודלים שהורצו – בחרנו באופן אקראי בk=10 .

לסיכום

בפרויקט זה, בעזרת סט נתונים (קובץ csv) הכולל בתוכו עמודות משתנים מסבירים (בלתי תלויים) ועמודת משתנה מוסבר (עמודת משתנה תלוי אשר ערכיו קטגוריאליים - 0,1) יצרנו מודלים אשר בעזרתם, בהינתן סט נתונים "חדש" אשר מכיל עמודות בעלות משתנים בלתי תלויים בלבד (עם אותם מאפיינים כמו סט העמודות אשר בעזרתם יצרנו את המודל), ניתן לחזות באופן מספק את ערכיי עמודת המשתנה התלוי.

מתוך האפשרויות השונות בחרנו במודלים הבאים:

SVM - עבור בעיות סיווג, בשלב האימון מתאימים מסווג שמפריד נכון ככל האפשר בין דוגמאות אימון חיוביות ושליליות. המסווג שנוצר ב SVM הוא המפריד הליניארי אשר יוצר מרווח גדול ככל האפשר בינו לבין הדוגמאות הקרובות לו ביותר בשתי הקטגוריות. כאשר מוצגת נקודה חדשה, האלגוריתם יזהה האם היא ממוקמת בתוך הקו המגדיר את הקבוצה, או מחוצה לו. SVM אינו מוגבל רק לסיווג ליניארי, ויכול לבצע גם סיווג לא ליניארי באמצעות הוספת קרנל (kernel) שבו ממופה הקלט למרחב בממד גבוה.

ANN - מודל אשר מדמה רשת אשר מכילה בדרך כלל מספר רב של יחידות מידע (קלט ופלט) המקושרות זו לזו, קשרים שלעיתים קרובות עוברים דרך יחידות מידע "חבויות".

NB Classifier - סיווג נאיב בייס בלמידת מכונה הוא אוסף שיטות סיווג המבוססות על חוק בייס ועל ההנחה ה"נאיבית" שאין תלות בין תכונות האובייקטים המסווגים כאשר כבר ידוע סיווגם.

Linear Regression – מודל זה מתבסס על שיטה מתמטית למציאת הפרמטרים של הקשר בין משתנה בלתי תלוי X למשתנה תלוי Y, בהנחה שהקשר ביניהם ליניארי.

העברנו את סט הנתונים 4 שלבים עיקריים: שלב האקספלורציה, שלב העיבוד המקדים, שלב הרצת המודלים ושלב הערכת המודלים; כאשר כל שלב התבסס על השלב הקודם לו ובכך, באופן מודולרי, הגענו לתוצאות מספקות והצלחנו ליישם מודלים אשר מניבים תוצאות אמינות.

תוצאות המודלים:

Model Name	Accuracy	Misclass	**AUC**
Logistic Regression	0.8282	0.1718	0.90
NB Classifier	0.6887	0.3113	0.86
SVM	0.8119	0.1881	0.89
ANN	0.8384	0.1616	0.92

אפשר לראות שחוץ ממודל ה-Naïve Bayes רוב המודלים הגיעו לתוצאות יחסית טובות. שלושת המודלים SVM, Logistic Regression, ANN נותנים תוצאות דומות כאשר רשת הנוירונים נותנות תוצאה רק מעט יותר טובה מהמודלים האחרים.
נשאלת השאלה מדוע מודל מתוחכם כמו רשת נוירונים לא מנצחת בפער מודלים פשוטים כמו SVM ו-Logistic Regression. במקרים בהם מודל פשוט מספק תוצאות דומות לאלה שמספק מודל מורכב, אפשר להסיק שהמודל המורכב אינו נדרש וכנראה שהמידע ניתן להפרדה בעזרת מודלים פשוטים. טענה זו מתחזקת כשאר אנו בוחנים

את הקרנל שנבחר כאשר הרצנו grid search על ה-SVM. הקרנל שנתן את התוצאה הטובה ביותר היה קרנל לינארי. לכן, אפשר להסיק שהמידע כנראה ניתן להפרדה לינארית.