# דו"ח מסכם

# פרויקט מסכם בלמידת מכונה

# בעיית סיווג קבלה לעבודה

12/08/2024

Ofir Shalhevet

## חלק 1- אקספלורציה (EDA):

חלקו הראשון של הפרויקט יעסוק בניתוח נתונים ובחינת הרלוונטיות שלהם. את האקספלורציה ביצענו על העתק נתוני ה-train, עליהם נבנה ונאמן את המודל. את הבדיקות החלטנו לבצע על סט הנתונים של הtrain ללא validation לאחר הסרה של תוצאות (שורות) בהן חסרים יותר מ3- ערכים (פיצ'רים) שונים ( ערכים בשורת התוצאה בהם יש NULL). ההנחה היא שעל משתנה עם יותר משלושה ערכים חסרים עלולה להטות את המודל ולא לסייע לו בעקבות חוסר אינפורמציה (2504 שורות). **תחילה, ביצענו בדיקת ערכים מצויים (ייחודיים) בכל עמודה.** פעולה זו אפשרה לנו להבין אילו עמודות בסט האימון מכילות נתונים מספריים (כגון int64 ו-float64) ואילו עמודות מכילות נתונים קטגוריאליים. הבנה זו הייתה קריטית מכיוון שהיא השפיעה על הדרך בה התמודדנו עם כל סוג של נתונים בהמשך הפרויקט. הבדיקה הבאה היתה **בדיקת היסטוגרמה של המשתנים הנומריים,** זיהינו כי רוב הפיצ'רים אינם מתפלגים בצורה נורמלית, למעט פיצ'ר D. ואף לרוב הפיצ'רים יש outliers נדרש לבחון בהמשך כיצד נכון להסיר את הנתונים הקיצוניים לטובת בניית המודל והסקת המסקנות.

לאחר מכן, בחנו את הפיצ'רים הקטגוריאליים כדי להבין האם הערכים המצויים חוזרים על עצמם בצורה משמעותית/ניתן ליצור מהם קבוצות של דוגמאות העולות ברוב הבדיקות, זאת לטובת בניית המודל בהמשך. התמקדנו בפיצ'ר 'C' וביצענו ניתוח כדי להבין את ההתפלגות שלו ביחס ל-label (תווית הסיווג) בעזרת תרשים עמודות. בעמודת sex, מצאנו שלושה ערכים ייחודיים (Man, Woman, Non-binary), אך ראינו כי כמות הגברים היא עצומה לעומת כמות הנשים בDATA ולעומת OTHER, ולכן בהמשך בשלב העיבוד איחדנו בין others לנשים- לבינארי (נספח 3). ובעמודה country, זיהינו 165 ערכים ייחודיים המייצגים מדינות שונות , שלאחר מכן הפכנו ליבשות (הרחבה בפרק הבא). לאחר מכן, ניסינו לנתח את הקשר בין היבשות לתוויות הסיווג (labels) באמצעות גרפים ובדיקת Chi-Square (שלא נלמד בכיתה). בדיקת **Chi-Square** אומנם, הראתה קשר סטטיסטי בין היבשות לתוויות (p-value קטן מ-0.05), אומנם מציינת כי יש קשר בין היבשת לתוצאה הסופית, אך לפי הסתכלות בגרף נראה כי החלוקה בין הלייבלים לכל יבשת היא מאוזנת, (אומנם באפריקה קצת פחות אך יתכן שנובעת מDATA פחותה) ועל כן, כרגע לא ניתן לדעת בוודאות אם יש השפעה לכל מדינה בתוך כל יבשת על הלייבל (נספח 2).

לאחר מכן, ה**חלטנו לבצע קורלציה בין המשתנים לlabel. תחילה, בין הבוליאניים:** כש P VALUE קטן מ-0.05 מצביע על מובהקות סטטיסטית, כלומר שיש קשר בין המשתנה לבין הלייבל. הקטגוריות של משתנה C לא הראו השפעה משמעותית על הלייבל, ככל הנראה שמכיוון לכל אדם יש אפשרות להשתייך רק לאחת מ6 קטגוריות. מצב זה יוצר חלוקה לא מאוזנת בין 0 ל-1 בכל קטגוריה של C, מה שגורם לערכי p-value גבוהים ולא מובהקים סטטיסטית. לעומת זאת, פיצ'רים בינאריים כמו worked_in_the_past או age_group, מציגים חלוקה ברורה יותר של הלייבלים ,מה שמוביל לערכי p-value נמוכים ומובהקים, וסביר שהם ישפיעו יותר על הלייבל (נספח 4). לפי הP VALUE ניתן לראות כי העובדה שאדם מתכנת או עבד בעבר סביר שמשפיעה יותר מאשר בעיות נפשיות.

**לאחר מכן בחנו את המשתנים הקטגוריאליים והנומריים, ביחס לLABEL וביחס לעצמם.** המסקנה העקרית היא שהמשתנים **stack_experience** ו-**D** נראים כבעלי חשיבות גבוהה יותר בניבוי הלייבל בשל יכולתם להבחין בין הקבוצות. משתנים אחרים כמו **prev_salary**, **years_of_experience**, ו-**education** עשויים להיות פחות משפיעים בשל החפיפה הגדולה יותר בהתפלגויות שלהם בין הלייבלים (נספח 5). **לבסוף, החלטנו לבצע קורלציות בין הפיצ'רים עצמם,** החלטנו לחקור את האינטראקציות בין כל שני פיצ'רים בוליאניים (באמצעות "וגם"), והגענו למסקנה שההשפעה שלהם משתנה בהתאם ל-label באמצעות מפות חום- להרחבה במחברת . בהמשך, הצגנו מטריצת קורלציה בין הפיצרים הנומריים. במטריצת הקורלציה ניתן לראות קשר חזק של 0.9 בין years_of_experience ל-B, מה שמרמז על כך שיותר שנות ניסיון קשורות לדירוג גבוה יותר ב-B. בשל הקשר הליניארי הזה, החלטנו לאחד בין הפיצ'רים הללו בעיבוד מקדים (ראו נספח 5 ו-6). בנוסף, ישנו קשר חזק של 0.58 בין stack_experience ללייבל, וקשר של 0.4 בין D ללייבל, מה שתומך בהשערה המוקדמת שלנו על חשיבותם של משתנים אלו בניבוי הלייבל (וגם הגיוני מבחינה פרקטית) (נספח 6).

## חלק 2- עיבוד מקדים

במהלך חלק זה טייבנו את הדאטה ואפשרנו שימוש בו בצורה מדויקת יותר בהמשך לבניית המודלים. החלק כלל התמודדות עם המשתנים הקטגוריאליים, הבוליאניים והנומריים, התמודדות עם תוצאות חריגות (Outliers), התמודדות עם נתונים חסרים בעמודות השונות, נרמול הנתונים וניתוח מידתיות הבניה. כל אלו בוצעו על נתוני ה-Train, ולבסוף מומשו על נתוני ה-Test. מהתהליך עלו מספר מסקנות עיקריות שראינו לציין בדו"ח זה בנוסף להסברים המפורטים במחברת הקוד.

ראשית, החלטנו לבצע One-Hot Encoding על פיצ'ר "C" כדי להמיר אותן לפורמט מספרי שהמודלים יכולים לעבוד איתו בצורה יעילה. שיטה זו מתאימה לפיצ'רים שאינם בעלי סדר טבעי (במקרה של עמודה "C",גם אם יש סדר, הוא לא ידוע לנו) וכן נרצה להימנע מלתת משמעות מספרית לקטגוריות, לאחר מכן, שינינו את הערכים True ו-False בערכת הנתונים לערכים 1 ו-0,

בהתאמה. במקביל, ויתרמו על הפיצ'ר 'C' לאחר ההמרה, מכיוון שהוא הפך ללא רלוונטי עבור הניתוחים הבאים. בנוסף, התמודדנו עם העמודה 'country' שהכילה מדינות רבות ושונות. מכיוון שהיו 165 מדינות שונות בעמודה זו, החלטנו לאחד את המדינות ליבשות. חישבנו את ממוצע התוויות (labels) עבור כל קטגוריה בעמודת 'country_to_continent' והכנסנו אותם לעמודה חדשה בשם 'continent_encoded'. פעולה זו מאפשרת לנו למפות את הממוצעים הללו חזרה לתוך ה-DataFrame המקורי, ומספקת מידע נוסף למודל על המאפיינים הגיאוגרפיים של הנתונים.

בפיצ'ר stack_experience. בתחילה, שקלנו לקודד את הפיצ'ר הזה לפי מספר השפות בהן יש ניסיון (אורך המחרוזת בלבד), אך הנחנו שזה עלול להטות את הנתונים, מכיוון שאדם שיודע לעבוד עם המון טכנולוגיות שאינן שפות תכנות סביר שלא יתקבל כמתכנת, אך אורך המחרוזת ארוכה. לכן, כדי לשמר את החשיבות של כל שפת תכנות, יצרנו מילון ערכים (dictionary) המכיל ניקוד עבור כל שפה בהתאם לדירוג באתר: [most popular programming languages 2023: When and how to use 40 them - Part 1 · Raygun Blog](), כאשר הניקוד משקף את החשיבות היחסית של כל שפה בתעשייה. בהמשך, כאשר גילינו כי פיצ'ר זה הכי משמעותי, פעולה זו אף העלתה לנו את ה AUC.

התמודדנו עם הפיצ'ר education בעזרת ייצוג משתנה אורדינלי שבו הקטגוריות מסודרות לפי סדר בעל משמעות (לדוגמה, תיכון, תואר ראשון, תואר שני, וכו') . כדי לשמר את הסדר הזה בצורה נכונה, השתמשנו בקידוד אורדינלי (ordinal label encoder), שבו כל דרגת השכלה קיבלה ערך מספרי בהתאם לסדר ההיררכי שלה (מ0 עד 4). בנוסף, הוספנו קטגוריה בשם "אחר" ( =other -1) עבור כל ערך שאינו מתאים לאחת הקטגוריות שהגדרנו, כדי לוודא שכל הנתונים מקודדים בצורה המתאימה למודלים. בעמודה sex, ראינו כי כמות הגברים היא עצומה לעומת כמות הנשים בDATA ולעומת OTHER, ולכן איחדנו בין others לנשים- לבינארי (נספח 3) ובעמודות בינאריות אחרות הפכנו מTRUE וFALSE ל0 ו1.

**טיפול בערכים חסרים:** בקטע זה התמודדנו עם הערכים החסרים בנתונים באמצעות הפונקציה SimpleImputer. עבור העמודות המספריות השתמשנו בחציון (median) כדי למלא את הערכים החסרים, מכיוון שהחציון פחות רגיש לחריגים (הרחבה במחברת). עבור העמודות הבינאריות השתמשנו בערך השכיח ביותר (most frequent) כדי לשמור על התפלגות הנתונים ולמנוע הטיה, מכיוון שברוב הפיצ'רים הבינאריים קיים פער כל כך גדול בין הערכים הנפוצים. בחירה זו מבטיחה שהחסרים יתמלאו בערך שמייצג את הרוב המוחלט של הנתונים בעמודה למלא את הערכים החסרים בצורה חכמה מבלי לאבד חלק משמעותי מהנתונים.

**טיפול בערכים חריגים:** בקטע זה בנינו פונקציה שמבצעת ווינסוריזציה (Winsorization) על פיצ'רים מספריים כדי לטפל בערכים חריגים. הפונקציה מציגה את ההתפלגות של הפיצ'רים לפני ואחרי התהליך, שבו הערכים הקיצוניים מוחלפים בערכים קרובים יותר לתחום הנתונים העיקרי. לדוגמה, אם בפיצ'ר prev_salary היו ערכים חריגים גבוהים במיוחד, הווינסוריזציה תחסום אותם בטווח מסוים כדי להפחית את השפעתם על המודל. פעולה זו מאפשרת לנו להתמודד עם ערכים חריגים מבלי להסירם, מה שעוזר לשמור על שלמות הנתונים ולהפחית את השפעתם השלילית על המודלים. לאחר הפעולה, ניתן לראות את השינוי בהיסטוגרמות (נספח 7). למשל, הקצה הימני של הגרף מסמל את הערכים הגבוהים ביותר של הפיצ'ר stack_experience, כלומר מקרים עם ניסיון רב בטכנולוגיות. לפני הווינסוריזציה, היו שם ערכים חריגים גבוהים. לאחר הווינסוריזציה, הערכים הגבוהים האלה הוגבלו והוזזו לכיוון המרכז, מה שהקטין את השפעתם על הנתונים (נספח 8), חשוב לנו לציין שניתן לראות שגם לאחר הטיפול בחריגים ניתן לראות בboxplot נקודות שאינן בטווח אך אנו מניחים שזה נובע מנתונים לגיטמיים, בזנבות ההתפלגות, או בגלל מאפייני הBOXPLOT (נספח 9). להרחבה ניתן לראות במחברת הקוד.

**בניית פיצ'רים חדשים:** הוספנו פיצ'ר חדש לנתונים על ידי שילוב של הפיצ'ר B עם years_of_experience, תוך יצירת יחס בין שני הפיצ'רים. תהליך זה נועד לסייע בזיהוי טוב יותר של קשרים משמעותיים בנתונים ולהפחית את הבעיה של מימדיות (dimensionality) (להרחבה במחברת הקוד).

**נורמליזציה:** ראינו כי כל פיצ'ר נמדד בסקלה שונה ולכן כדי לנטרל את ההשפעה השונה של כל נתון בפיצ'ר על המסקנה הכוללת במודל, צריך לבצע נרמול. ביצענו נרמול של הנתונים בעזרת MinMaxScaler . בחרנו בשיטה זו כי בשיטה זו ניתן לשמור על היחס בין הדוגמאות השונות למרות השינוי לערכים בין 0 ל 1, למעשה שינוי הערך לא פוגע במשמעות היחסית. את הנ"ל ביצענו לאחר התמודדות עם ה- Outliers, מכיוון שערכן משפיע על תהליך הנירמול של שאר הנתונים המתייחס לדוגמא הגדולה והקטנה ביותר וכך מחשב את היחסים בין הדוגמאות, ניתן לראות את הדמיון בין הגרפים של הפיצרים הנומריים לפני ואחרי נורמליזציה (מה שאנו מניחים שמעיד על תקינות) (נספח 10). בנוסף לנ"ל ישנן עוד מסקנות המפורטות ב- MarkDown במחברת הקוד.

## חלק 3 – הרצת המודלים

בחלק זה, ביצענו הרצה של המודלים השונים על הדאטה שלנו. הנ"ל בוצע עד הגעה לתוצאה סופית ובחינת הרלוונטיות של השינויים שביצענו (ביול היפר -פרמטרים לכל מודל).

**KNN:**

מודל KNN משמש למשימות סיווג ורגרסיה, בכך שהוא מבצע תחזיות על סמך הדוגמאות הקרובות ביותר לדוגמה החדשה במאגר האימונים. במודל זה, ההיפר פרמטר המרכזי הוא מספר השכנים k, כלומר מספר הדוגמאות הקרובות ביותר שבהן נעשה שימוש כדי לקבוע את הסיווג או את התוצאה. בניתוח שלנו, בדקנו את הביצועים של המודל עם ערכים שונים של k בטווח של 5 עד 20. מצאנו כי הערך האופטימלי של k הוא 14, עם MSE (שגיאת ריבוע ממוצעת) של 0.2772 על סט הוולידציה, מה שמראה על ביצועים טובים של המודל. ממוצע ה-AUC (שטח תחת העקומה) שהתקבל מחמישה מחזורי קרוס-וולידציה היה 0.8166, מה שמעיד על יכולת טובה להבחין בין המעמדות. בנוסף, ה-AUC על סט הוולידציה עבור k=14 היה מעט נמוך יותר - 0.8024. בניתוח עקומות ה-MSE, ניתן לראות שככל שמספר השכנים גדל, השונות קטנה עד לנקודה מסוימת (בסביבות k=14), ומשם היא מתחילה לעלות שוב. זה מצביע על האיזון האופטימלי בין שונות להטיה. למרות שיש סימנים קלים של Overfitting, הפער בין ה-MSE של האימון לבין ה-MSE של הוולידציה קטן יחסית (כ-0.05), מה שמעיד שהמודל לא סובל מהתאמה יתר חמורה. בנוסף, נראה כי לאחר יישום PCA והפחתת מימדים, הערך האופטימלי של k השתנה ל-20, וה-AUC הממוצע היה 0.8031. למרות שה-MSE של הוולידציה עם הערך k=20 היה נמוך יחסית (0.2859), ניתן לראות ירידה קלה בביצועים בהשוואה למודל ללא PCA, במיוחד בהתחשב ב-AUC על סט הוולידציה שהיה 0.7981.באופן כללי, התוצאות מצביעות על כך שהפחתת המימדים באמצעות PCA יכולה להוביל לשינויים בביצועים, ו**במקרה הזה נראה שהמודל ללא PCA מספק תוצאות מעט טובות יותר (נספח 10).**

**Naive BAYES**

מודל Naive Bayes מבוסס על ההנחה של אי-תלות מותנית בין המשתנים, כאשר כל פיצ'ר תורם לסיווג באופן עצמאי. ב-Gaussian Naive Bayes, ההנחה היא שהפיצ'רים המספריים מתפלגים נורמלית. זהו מודל פשוט ויעיל לסיווג, ונמצא שימושי במקרים רבים. השתמשנו ב-Gaussian Naive Bayes לחיזוי האם מועמד יתקבל לעבודה. בחרנו להשתמש **בערך ברירת המחדל של ההיפר-פרמטר var_smoothing: 1e-9** מתוך הנחה שאם המודל עם ערכי ברירת המחדל יספק ביצועים טובים יחסית, יוכל להוות מדד כללי לדיוק המודל, ואז נשקיע באופטימיזציה נוספת. לאחר הרצה ראשונית של המודל, קיבלנו ממוצע AUC של כ-0.8216 על פני חמישה מחזורי קרוס-וולידציה. על סט הוולידציה, ה-AUC היה 0.8000, מה שמעיד על כך שהמודל מצליח להבחין בין הקבוצות. לאחר מכן, בחנו את השימוש ב-PCA עם המודל, ממוצע ה-AUC עם PCA היה 0.7910, ועל סט הוולידציה ה-AUC היה 0.7677. ובנוסף השימוש ב pca לא הפחית את ה mse על ה validation. **(נספח 11)** בשל הירידה בביצועים, החלטנו שלא להשתמש ב-PCA עבור המודל הזה. בשל ביצועיו הנמוכים יחסית של Naive Bayes למודלים אחרים שניסינו, כמו Random Forest ו-Decision Tree, (AUC גבוה יותר בכ-0.6 נקודות), החלטנו שלא לבצע אופטימיזציה נוספת למודל זה. בחרנו ב-Gaussian Naive Bayes כי זיהינו שהפיצ'רים המשפיעים בנתונים היו בעיקר קטגוריאליים, כמו EDUCATION, או מספריים כמו stack_experience, D, ו-A, שמתפלגים בקירוב בצורה נורמלית. הבחירה הזו התחזקה בעקבות הניתוח שלנו שהראה שלפיצ'רים הללו יש חשיבות גבוהה בניבוי התוצאות.

**Decision Tree**

מודל עץ החלטה עובד על ידי פיצול הנתונים בצמתים שונים על בסיס תנאים מסוימים, ויוצר עץ של החלטות שמוביל לתוצאות הסופיות. כל צומת בעץ מייצג בדיקה על תכונה מסוימת, וכל ענף מייצג את תוצאות הבדיקה, עד שמגיעים לעלי העץ שמייצגים את התחזית או הסיווג הסופי. עץ החלטה נחשב קל להבנה ופירוש, אך רגיש ל overfitting-אם לא מווסתים את עומק העץ או משתמשים בשיטות להקטנת מורכבותו. מודל עץ ההחלטה (ללא PCA) מציג את הביצועים הטובים ביותר שלו בעומק עץ של 7 (היפר פרמטר), כאשר דיוק הוולידציה הוא 0.7666. ה AUC-הממוצע מחמשת מחזורי הקרוס-וולידציה הוא 0.8606, דבר שמעיד על יכולת גבוהה של המודל להבחין בין הקבוצות. כאשר מעמיקים את העץ ומגדילים את העומק המקסימלי, דיוק האימון עולה, מה שמצביע על כך שהמודל מתאים יותר לנתוני האימון. עם זאת, במקביל, דיוק הוולידציה יורד בעומקים גבוהים יותר, מה שמעיד על היווצרות מצב של overfitting, שבו המודל מתאים את עצמו יותר מדי לפרטי הנתונים באימון ואינו מצליח להתאים את עצמו בצורה טובה לנתונים חדשים. מבחינת שונות והטיה, ככל שהעומק המקסימלי של העץ גדל מעבר ל-7, השונות במודל יורדת מכיוון שהמודל מתאים עצמו בצורה טובה יותר לנתוני האימון, אך ההטיה גדלה כי המודל לומד יותר מדי את פרטי הנתונים במקום לתפוס את המאפיינים הכלליים. מצב זה מודגם בעליה בדיוק האימון וירידה בדיוק הוולידציה, מה שמעיד על כך שהמודל מאבד את יכולתו להכליל על נתונים חדשים.

כאשר מבצעים הורדת מימדים בעזרת PCA, עדיין נצפים הביצועים הטובים ביותר עם עומק עץ של 7, כאשר דיוק הוולידציה הוא 0.7220 וה AUC-הממוצע הוא 0.8173. אומנם, המודל עם PCA יכול להיות פחות רגיש ל overfitting-מאשר המודל ללא PCA (שכן יש פחות פער בין הטעויות על סט האימון מאשר על סט הוולידציה), אך נראה כי התוצאות נמוכות יותר מאשר במודל ללא PCA. זה מצביע על כך ש PCA-אומנם מפחית את בעיית ה overfitting, אך ייתכן שגם גורם לאובדן של מידע חשוב. הפער בין ביצועי המודל ללא PCA לבין המודל עם PCA הוא משמעותי, ולכן סביר שלא נבחר בPCA, **(נספח 12).**

**<u>Random Forest- The chosen model</u>**

בחרנו במודל Random Forest וביצענו אופטימיזציה של ההיפרפרמטרים על מנת להגיע לתוצאות הטובות ביותר. לאחר ניסיונות שונים, מצאנו את השילוב האופטימלי של ההיפרפרמטרים אשר הביא לתוצאה הטובה ביותר ב-AUC בעזרת Grid search**.** **(נספח 15**). תחילה בדקנו מודל ללא PCA, הביצועים הטובים ביותר התקבלו בעומק מקסימלי של 35. המודל השיג דיוק של 0.82 על סט האימון ודיוק של 0.77 על סט הוולידציה. ה-AUC הממוצע בקרוס-וולידציה עמד על 0.8682, בעוד שבסט הוולידציה ה-AUC היה 0.8633. לאחר מכן, ניסינו לשפר את המודל בעזרת שימוש ב-PCA, אך התוצאות שהתקבלו היו פחות טובות מאשר במודל ללא PCA (ירידה טריוויאלית). במודל עם PCA, הביצועים הטובים ביותר התקבלו בעומק מקסימלי של 20. המודל השיג דיוק של 0.82 על סט האימון ודיוק של 0.74 על סט הוולידציה. ה-AUC הממוצע בקרוס-וולידציה עמד על 0.8392, ובסט הוולידציה ה-AUC היה 0.8359. נראה כי מודל ה-PCA עזר להפחית את עומק העץ אך איבדנו מידע.

ניסינו לבצע אופטימיזציה נוספת על ידי שימוש בData ללא PCA. ראינו, כי בגרף ה-Accuracy, החל מעומק עץ של 10, ה-Validation Accuracy מתייצב בעוד שה-Train Accuracy ממשיך לעלות. הנחנו כי סימן זה מעיד על התחלה של Overfitting, ולכן החלטנו להריץ את המודל מחדש עם עומק מקסימלי של 10. לאחר ההתאמה הזו, התוצאה השתפרה, עם AUC ממוצע בקרוס-וולידציה של 0.8682 ו-AUC על סט הוולידציה של 0.8638.

לאחר ניתוח Feature Importance, ניסינו אופטימיזציה נוספת, גילינו כי מספר מצומצם של פיצ'רים משפיעים משמעותית על המודל (כפי שהנחנו). כתוצאה מכך, החלטנו להוריד פיצ'רים פחות חשובים ולהתמודד עם "קללת המימד" על ידי השארת 15 הפיצ'רים החשובים ביותר בלבד, זאת בהתאם לממצאי ה-PCA, וגם כדי לא לפשט את המודל יתר על המידה. לאחר מכן, ניסינו כלי חדש בשם CalibratedClassifierCV, שהצליח לשפר עוד יותר את המודל ל-AUC של 0.8644. (להרחבה עליו בחלק 6), <u>ובך אנו מניחים שלמרות שהמודל היה overfitted הצלחנו להגדיל את יכולת ההכללה שלו (**נספח 17-13).**</u>

## :Feature Importance

במהלך הפרויקט רצינו ל הבין את מידת התרומה של הפיצ'רים למודל שבחרנו. החלטנו כי לא נכון לנתח את המשמעות של כל פיצ'ר. שמנו לב כי בגרף אנו רואים כי הקידודים בינאריים, קיבלו חשיבות נמוכה יותר, ככל הנראה כי ברמה האיכותית קשה להסיק מסקנות על תוצאות בינאריות. ולכן בחרנו להתמקד ב- TOP7 , זאת לטובת הסקת מסקנות איכותית על המודל ולא רק כמותית כמו שמבוצע בשאר הפרויקט **(נספח 16).**

**stack_experience:** ניתן לראות שזהו הפיצ'ר החשוב ביותר במודל ה- Random Forest, מה שמחזק את הממצאים הקודמים שלנו בניתוח ה-EDA, פיצ'ר הזה הראה מתאם חזק עם התווית (label) ועם ההגיון שלנו, מה שמעיד על כך שמועמדים עם יותר ניסיון טכנולוגי נוטים יותר לעמוד בקריטריונים הנדרשים לקבלה לעבודה.

**D:** במהלך ה-EDA, הפיצ'ר הזה הראה מתאם (0.4) עם התווית ובהבדלה בין הlabel. מה שהצביע לנו על כך שישפיע משמעותית על תחזיות המודל. סביר להניח שהחשיבות שלו נובעת מכך שהוא משקף אינדיקטור כלשהו להצלחה בתפקיד, אף על פי שאין לנו ידע מדויק לגבי מה הוא מייצג.

**Prev salary:** למרות שמצאנו חפיפה משמעותית בין התוויות עבור פיצ'ר זה במהלך ה-EDA, הוא עדיין תורם לתחזיות המודל. הסיבה לכך יכולה להיות היכולת של הפיצ'ר להבדיל בין מועמדים כאשר הוא משולב עם משתנים אחרים, גם אם הוא לא הגורם המכריע ביותר בפני עצמו. זה נשמע הגיוני, שכן משכורת יכולה לשקף ניסיון ובכירות.

**A:** התרומה של פיצ'ר זה מודגשת על ידי מיקומו ברשימת החשיבות. למרות שלא ברור מהו הפיצ'ר המקורי שהוא מייצג.

**education:** אף על פי שפיצ'ר זה לא הראה השפעה משמעותית כמו פיצ'רים אחרים, הוא עדיין משחק תפקיד במודל. אנו מניחים כי הוא תורם לתחזיות הכלליות בכך שהוא מספק מידע רקע על המועמד, אך הוא פחות מכריע בהשוואה לפיצ'רים כמו stack_experience או D.

**B_years_ratio:** השילוב בין משתני B ו-years_of_experience הראה מתאם חזק במהלך ניתוח הקורלציות, מה שהוביל להחלטה למזג אותם לפיצ'ר אחד. ונראה כי השילוב אכן משפר את יכולת החיזוי של המודל. אומנם B לא ידוע לנו, אך בראי מציאותי, שנות נסיון אכן מהווה אינדיקציה לקבלה לתפקיד.

**IS_Dev**: למרות שהפיצ'ר בינארי, הוא נמצא במקום השביעי, כלומר גם לפי ההגיון וגם לפי המודל, לנסיון של אדם בתכנות ספציפית, (לעומת רק אם עבד בעבר) יש השפעה, וזה אף מסתדר לנו עם ההגיון המציאותי.

## חלק 4 - הערכת המודל

הערכנו את המודל באמצעות K-Fold cross validation בחלק הקודם, ותיארנו פערי ביצוע ו overfittingו ואיך בכל זאת הצלחנו להגדיל את יכולת ההכללה שלו. כאן נבצע ניתוח על מטריצת הבלבול (Confusion Matrix) כפי שניתן לראות בגרף המצורף.

המטריצה מחולקת לארבע אפשרויות:

1. TP (True Positive) - המקרים שבהם המודל חזה את התוצאה הנכונה של מועמד שהתקבל לעבודה, ואכן כך קרה בפועל. במטריצה זו, 88.5% מהמועמדים שחזו שיתקבלו לעבודה, אכן התקבלו.
2. FP (False Positive) - המקרים שבהם המודל חזה שמועמד יתקבל לעבודה, אך בפועל הוא לא התקבל. כאן מדובר ב-18.45% מהמקרים.
3. TN (True Negative) - המקרים שבהם המודל חזה שמועמד לא יתקבל לעבודה, ואכן הוא לא התקבל בפועל. המודל חזה נכון ב-64.1% מהמקרים שבהם מועמדים לא התקבלו.
4. FN (False Negative) - המקרים שבהם המודל חזה שמועמד לא יתקבל לעבודה, אך בפועל הוא כן התקבל. המודל טעה ב-11.5% מהמקרים.

למעשה, האלכסון הראשי של המטריצה (המאונך בין ה-TN וה-TP) מעיד על איכות המודל, כשהנתונים מראים שאחוז הדיוק (Precision) של המודל הוא 64.1% עבור מועמדים שלא התקבלו, ו-88.5% עבור מועמדים שכן התקבלו. הרגישות (Recall) של המודל, המודדת את אחוז המקרים החיוביים שנחזו נכון, עומדת על 88.5%. לדעתנו, המקרה החמור ביותר הוא הFP מכיוון שהוא "מאשר" מועמד שאינו מתאים, מה שעלול לגרום לעלויות לארגון, כמו חוסר פרודוקטיביות או הצורך להחליף את העובד במהירות.

## חלק 5 – ביצוע פרדיקציה

בשלב זה הרצנו את המודל על נתוני סט ה Test-את נתוני ה Test-העברנו את אותו תהליך עיבוד מקדים כמו בנתוני ה ,Train-שכלל סידור והתאמת הדאטה למודל Random Forest- אך כמובן ללא הסרת שורות עם ערכים חסרים כפי שביצענו על ה train(לא על testולא על validation). בנוסף, ניתן לראות בנספחים את ההתפלגויות, הפרדיקציה בוצעה לאחר שיפור התוצאות באמצעות כלי שלא נלמד, ה CalibratedClassifierCV. (להרחבה במחברת הקוד).

## חלק 6- שימוש בכלים שלא נלמדו בקורס

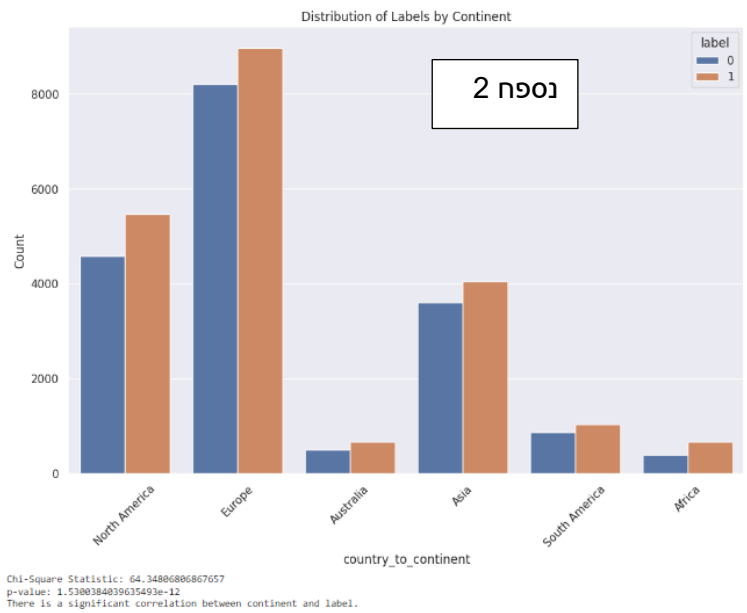
CalibratedClassifierCV: **(נספח 17)**

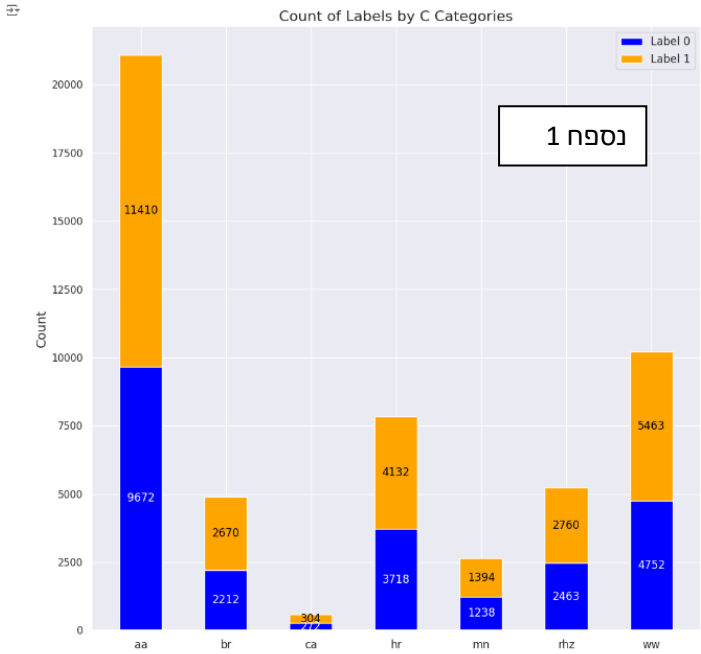
את הכלי בחרנו לממש בשלב הערכת המודל, זאת לאחר הבנה של התוצאות הקיימות ורצון לשפר את התוצאות של המודל הטוב ביותר. הבנו כי אנחנו טועים יותר במצב של FP (השגיאה החמורה ביותר לדעתנו) ורצינו לנסות להקטין למינימום את כמות הטעויות במצב זה .הפונקציה הזו מקבלת את המודל הנבחר במקרה שלנו (Random Forest – לאחר שאומן, ובעזרת שיטת Sigmoid, היא מטייבת את ההסתברויות של אי ההתאמות, כלומר, הלייבלים שסווגו בצורה שגויה, על ידי שימוש בהסתברויות של הדוגמאות שסווגו נכון. בחנו את התוצאות בעזרת שימוש ב Regression Isotonic-מול Sigmoid, והמתודה שהורידה את ה-FPלמינימום הייתה .Sigmoid לאחר מכן, מבוצע אימון למודל מחדש, שמטרתו לשפר את ההסתברויות שכל לייבל יסווג נכון. כך הפונקציה משפרת את תוצאות המודל מ0.8636 ל0.8644 ומקטינה את כמות ה FP-מ-0.369 ל 0.359.

# נספחים:

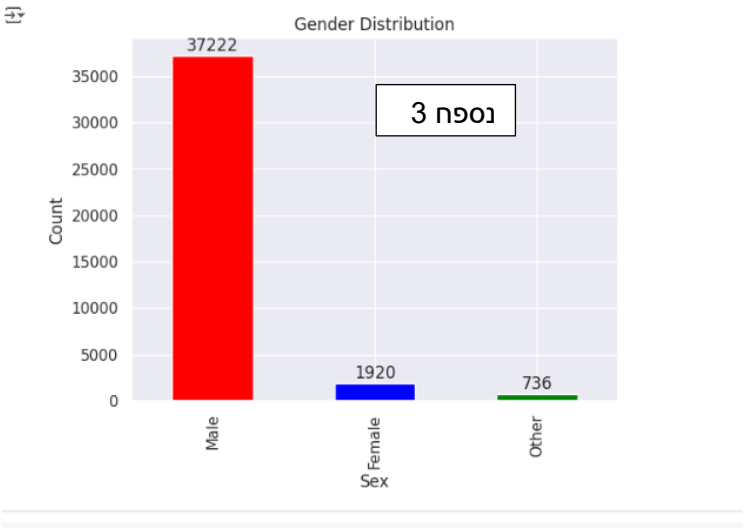
## חלק 1 וחלק 2- אקספלורציה ועיבוד מקדים:

התפלגות הפיצ'ר "C" לפי קטגוריות והתפלגות לפי יבשות לאחר ה"טיפול" בפיצ'ר "country".

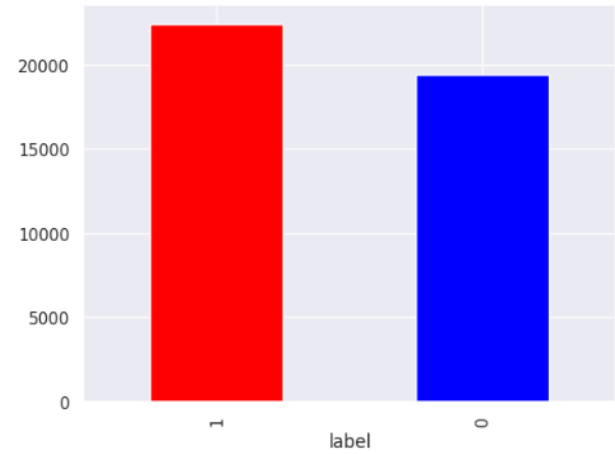



היחס בין הלייבלים, וגרף המשתנה הקטגוריאלי "SEX" אשר בהמשך הפכנו למשתנה בינארי.

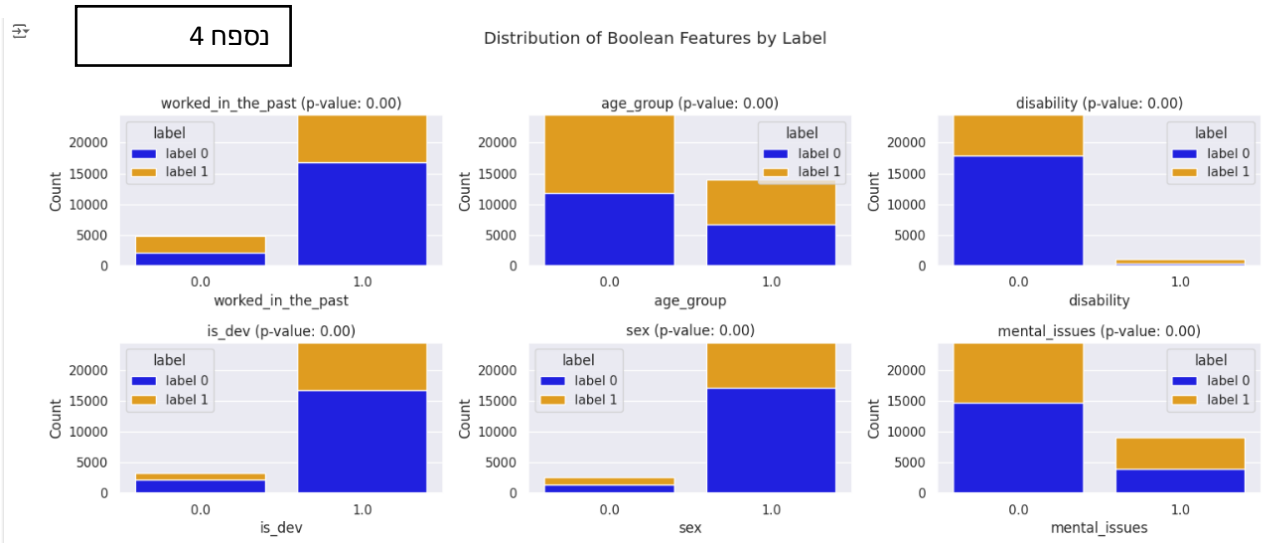



```
label
1    0.5362
0    0.4638
Name: proportion, dtype: float64
<Axes: xlabel='label'>
```



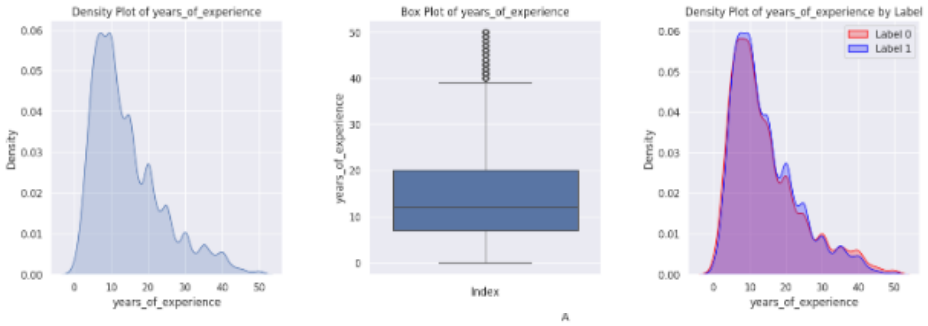
כחלק מהאקספלורציה, רצינו לראות איך כל פיצ'ר מתפלג ואת החלוקה לפי הלייבלים למשתנים הבינאריים. בנוסף ביצענו מבחן CHI למובהקות סטטיסטית, תוצאות במחברת הקוד.

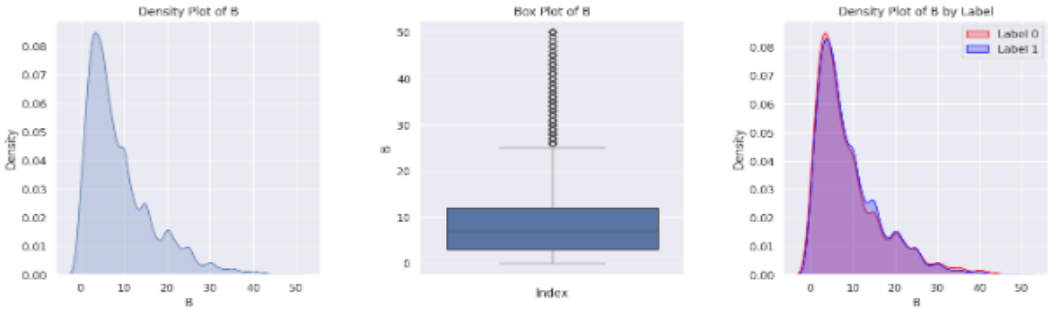
נספח 4

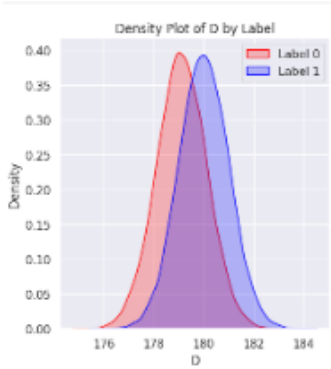


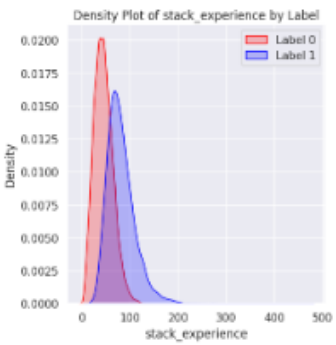

לפיצ'רים הקטגוריאליים והנומריים יצרנו density plot, BOX PLOT ו Density Plot by label זאת כדי לזהות התפלגות, זנבות, חריגים, ואיזה פיצ'ר מצליח להפריד בין הלייבלים בצורה טובה יותר (פחות חפיפה של שני הגרפים) לשם טיוב המודלים בהמשך ולמידה על הפיצ'רים. למשל בגרפים של D ושל stack experiences . בנוסף, ניתן לראות את הדמיון בהתפלגות בין B ל Years of experience. לדוגמא:

נספח 5




לעוד דוגמאות של גרפי התפלגות הצפיפות

ולתובנות נוספות, ניתן להסתכל במחברת הקו

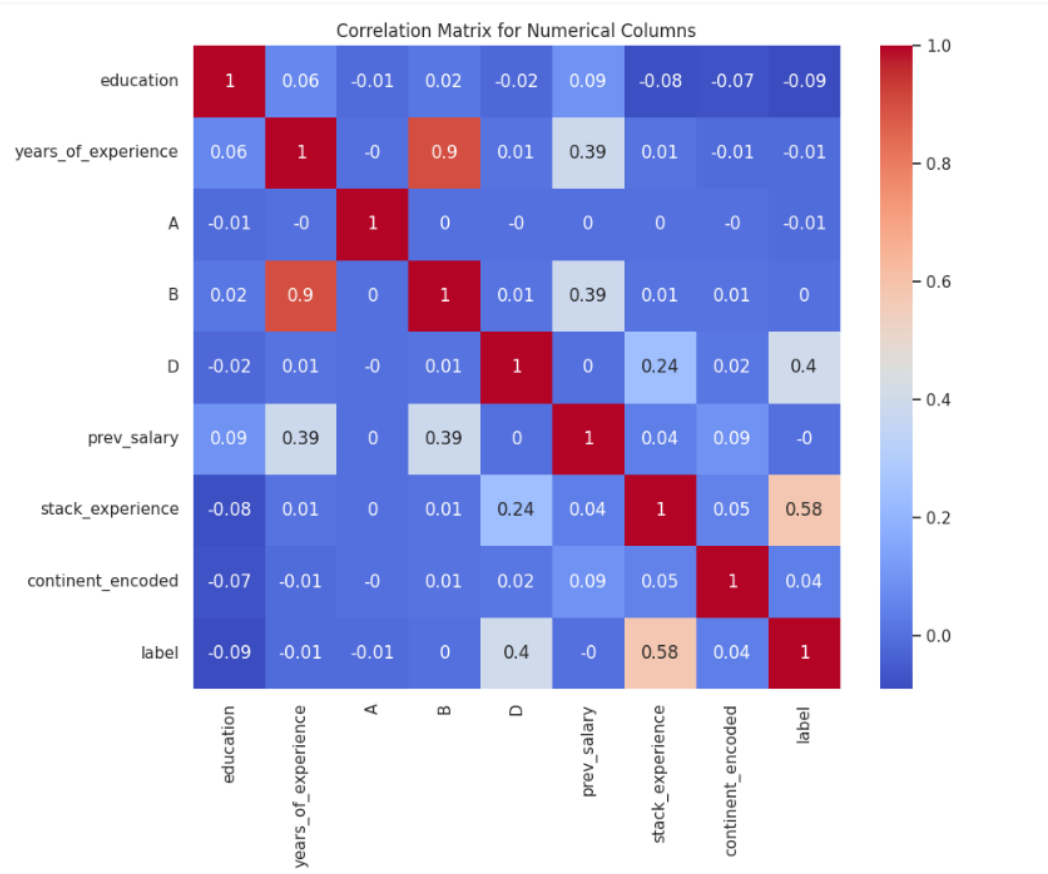
ניתן לראות שפיצ'ר D מתפלג נורמלי,

וכי שני הפיצ'רים מפרידים במידה מה

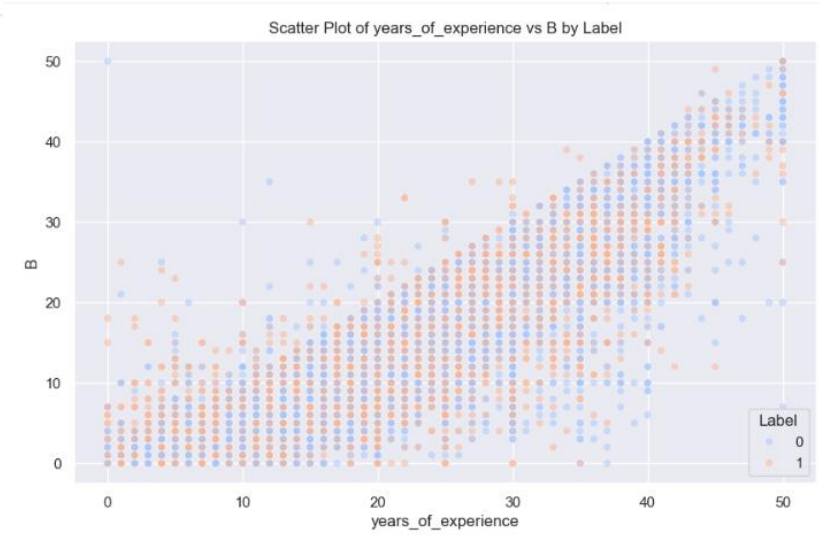
בין הלייבלים .

**גרף קורלציה בין משתנים נומריים וקטגוריאליים ללייבל,** גם כאן ניתן לראות קורלציה גבוהה בין B years of experience

ובין D & stack of experience ללייבל.



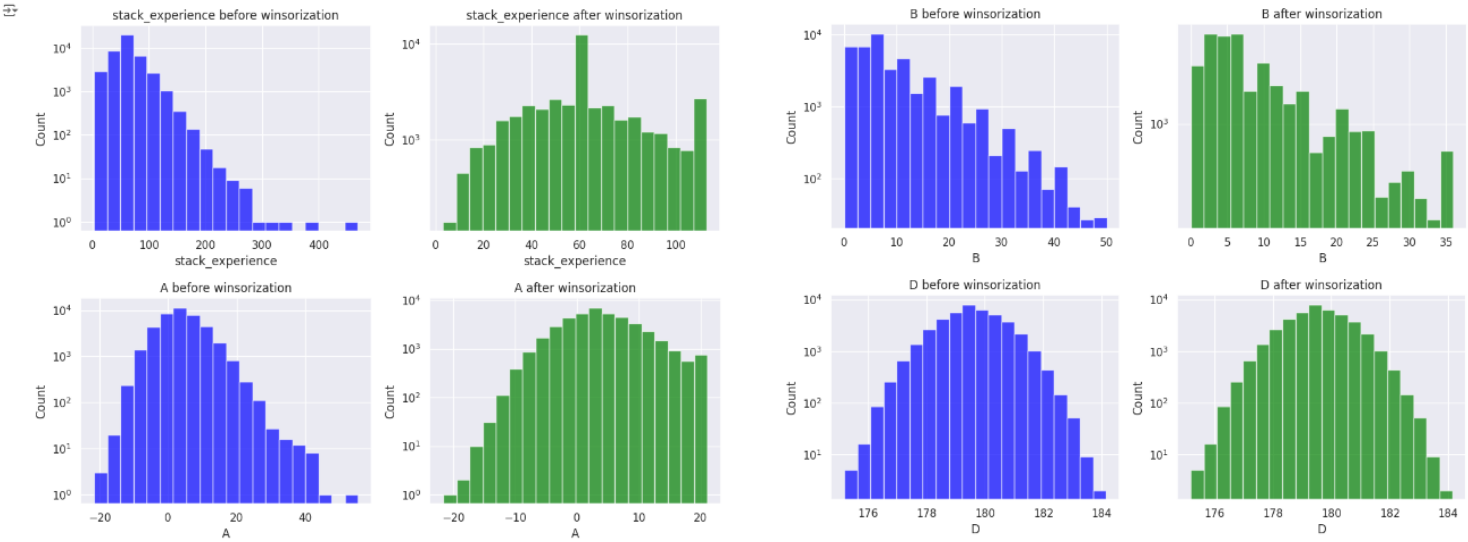

נספח 6




נספח 7

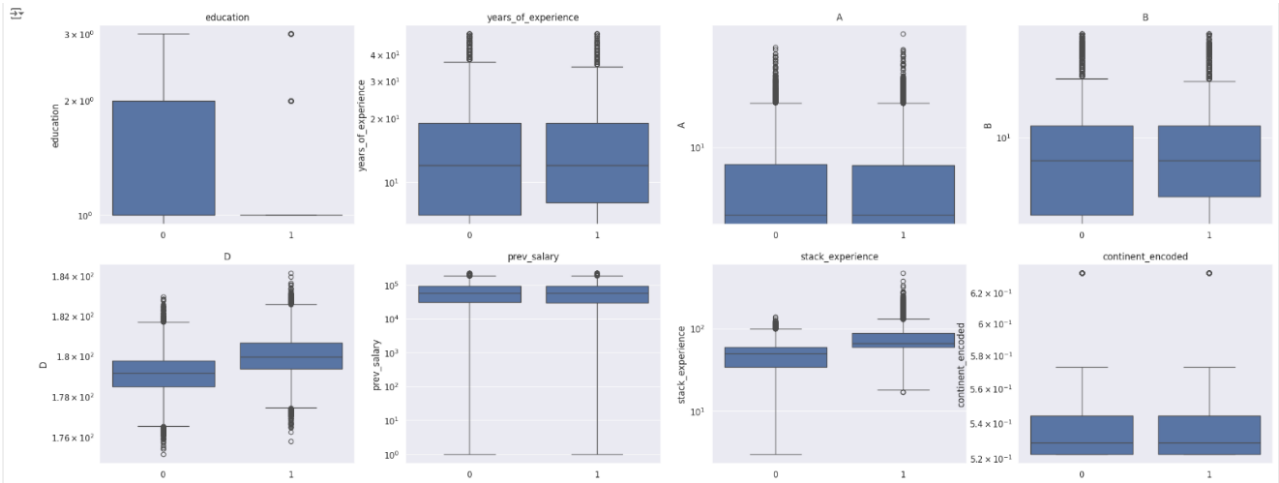
**גרפים מדגמיים לפני ואחרי התמודדות עם "חריגים" בשיטת winsorization:**

נספח 8



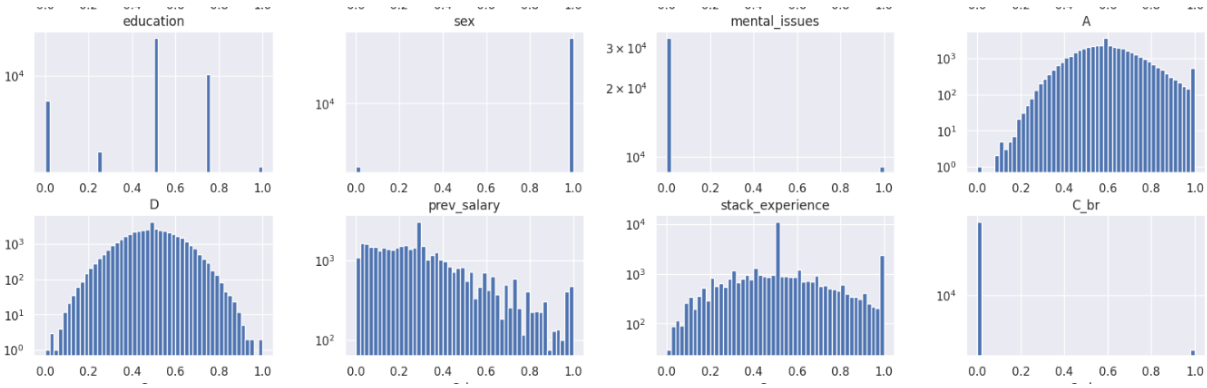

**ערכים חריגים \ "זנבות" לפיצ'רים הנומריים:**

נספח 9




**הגרפים לאחר נורמליזציה בשיטת MIN MAX:**

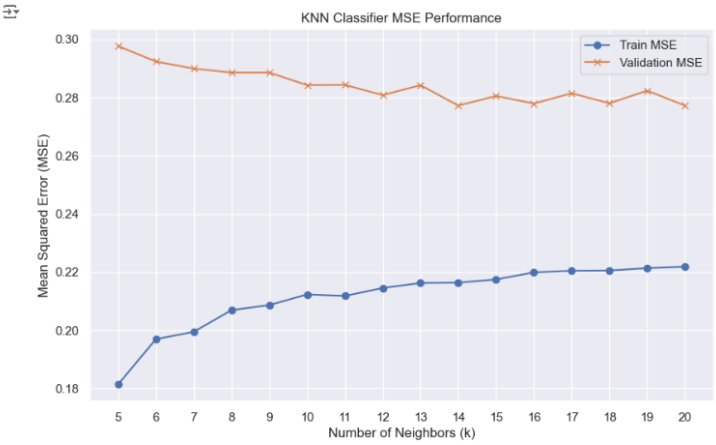
נספח 10

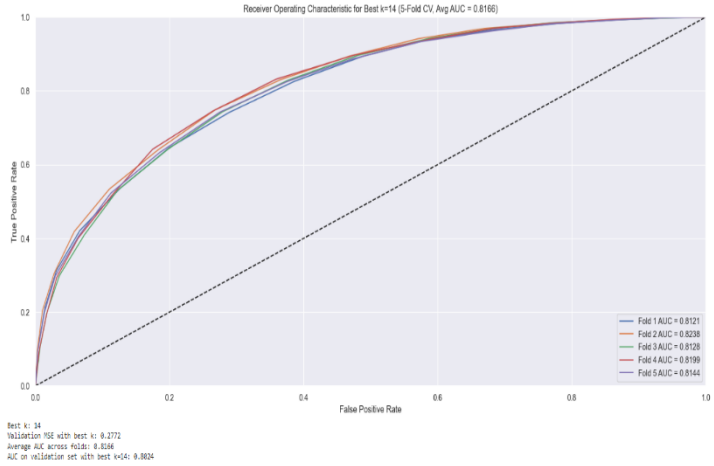


## חלק 3- מודלים:

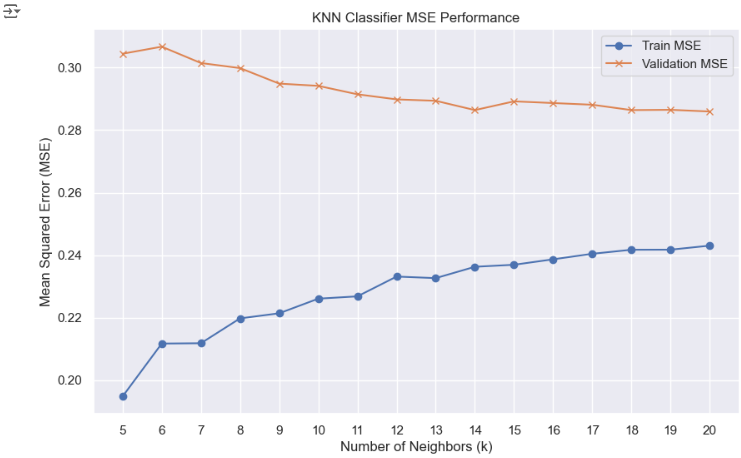
נספח 11

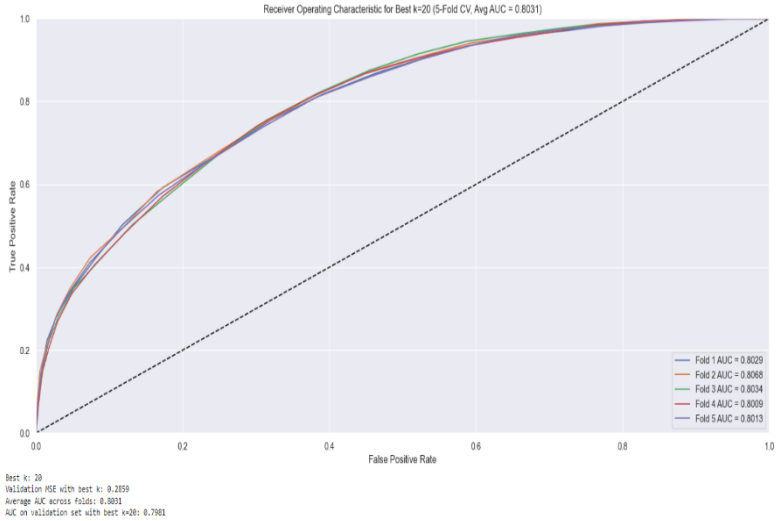
### Knn without PCA:



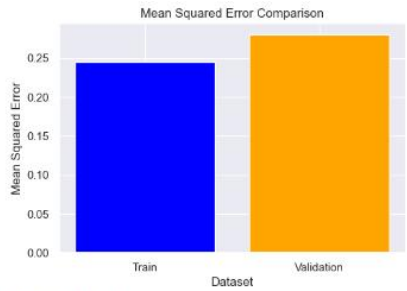




### Knn with PCA:

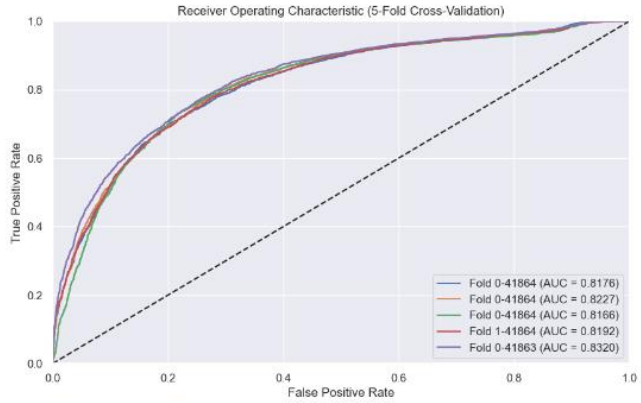


## Naïve Bayes without PCA



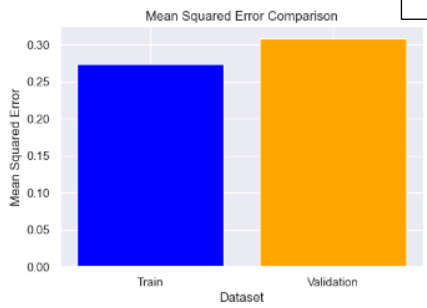

```
AUC on Validation Set: 0.8090
```



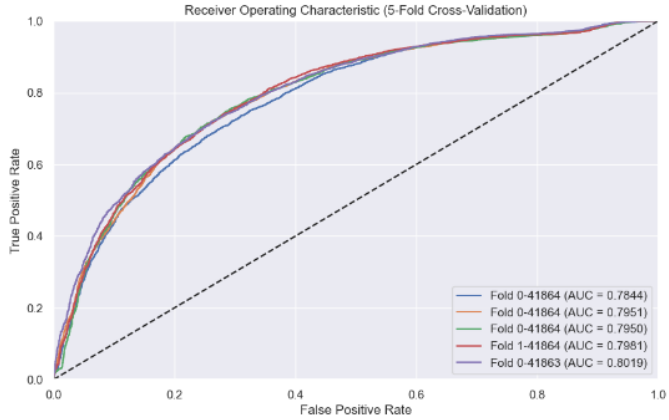

```
Average AUC across folds: 0.8216
```

## Naïve Bayes with PCA

נספח 12



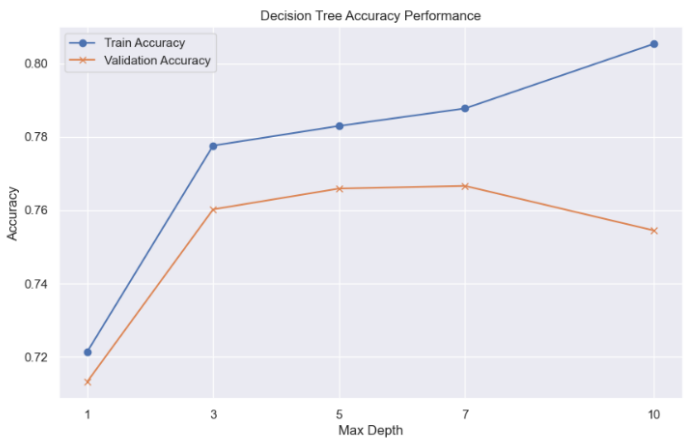

```
AUC on Validation Set: 0.7677
```

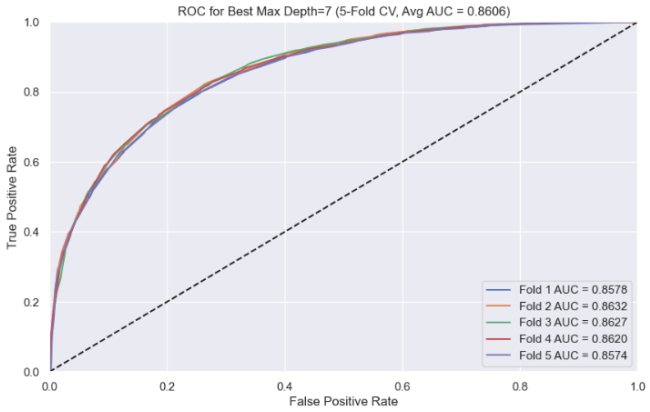



```
Average AUC across folds: 0.7949
```

נספח 13

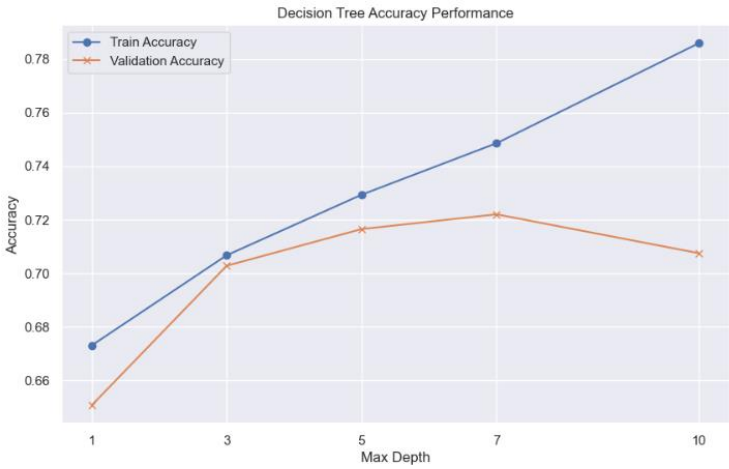
## Decision Tree without PCA:

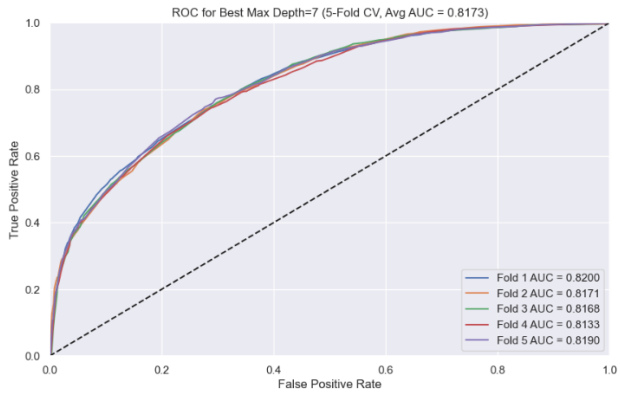



```
Best max depth: 7
Validation Accuracy with best max depth: 0.7666
Validation AUC with best max depth: 0.8576
Average AUC across folds: 0.8606
```

## Decision Tree with PCA:

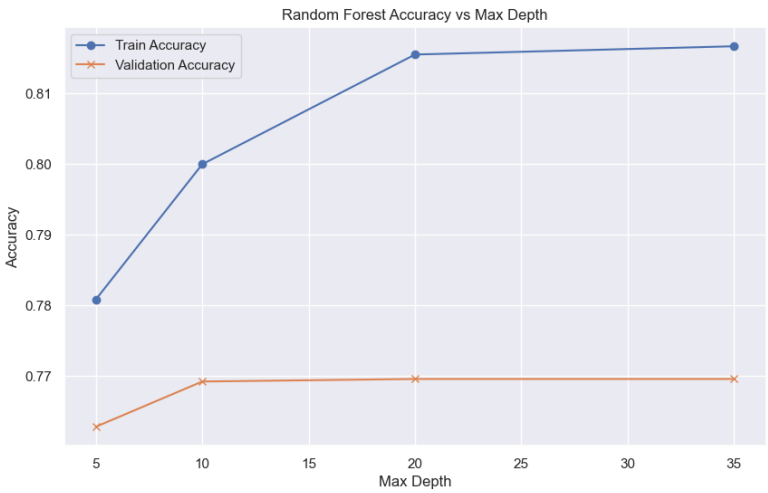


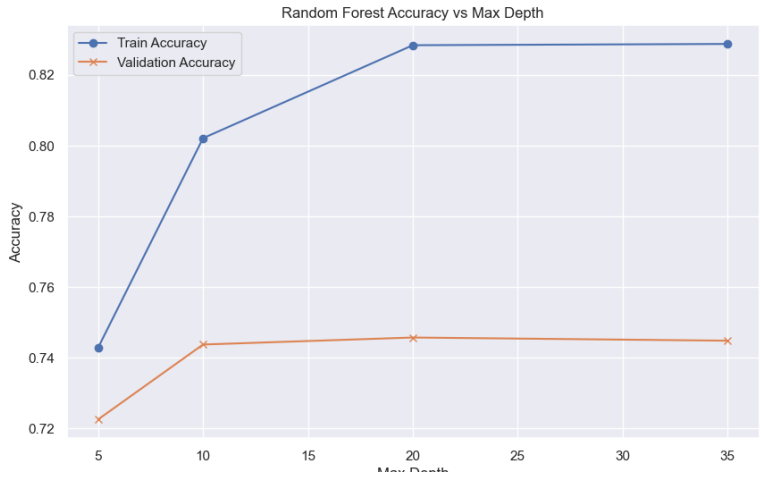

```
Best max depth: 7
Validation Accuracy with best max depth: 0.7220
Validation AUC with best max depth: 0.8092
Average AUC across folds: 0.8173
```

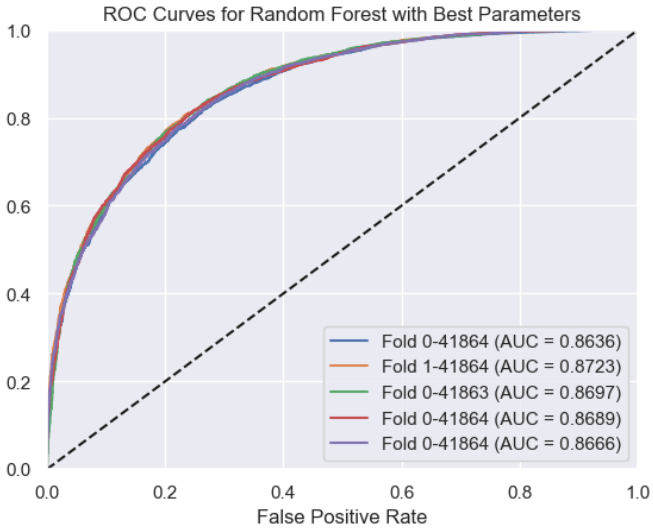
נספח 14

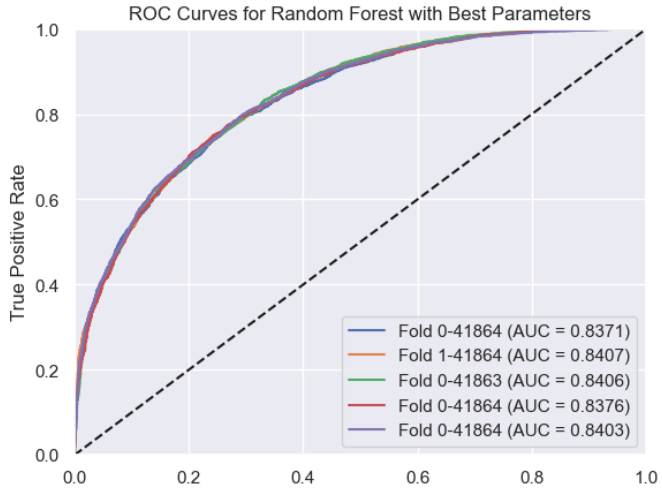
**Random Forest without PCA**




**Random Forest with PCA**

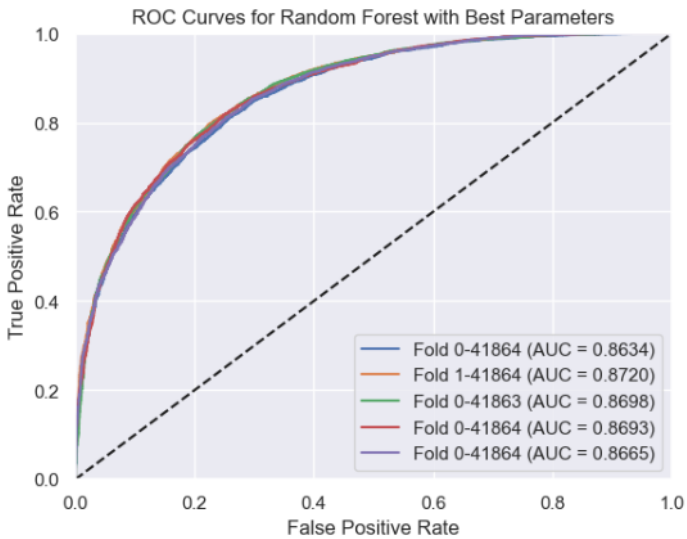



**ROC after using best hyperparameters from GridSearch**: 'entropy', 'max_depth': 10, 'max_leaf_nodes': 650, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 600.

נספח 15

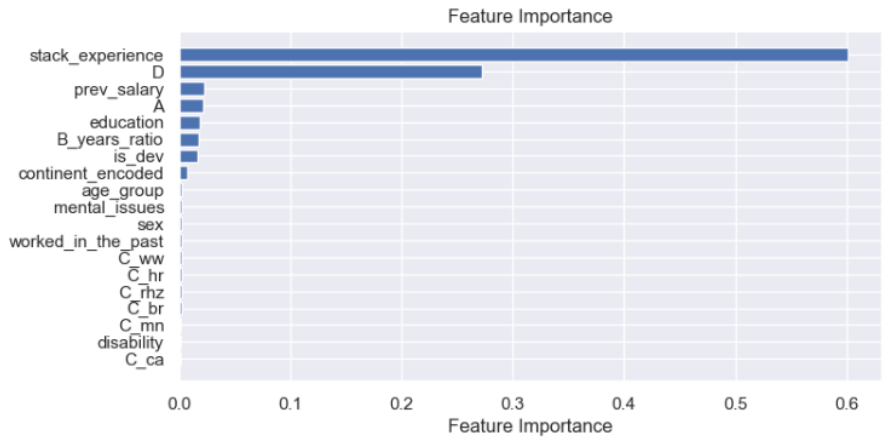



```
Average AUC across folds: 0.8682
[0.768877   0.44856198 0.37528742 ... 0.87625795 0.47044872 0.93396244]
Test AUC on validation set: 0.8638
```

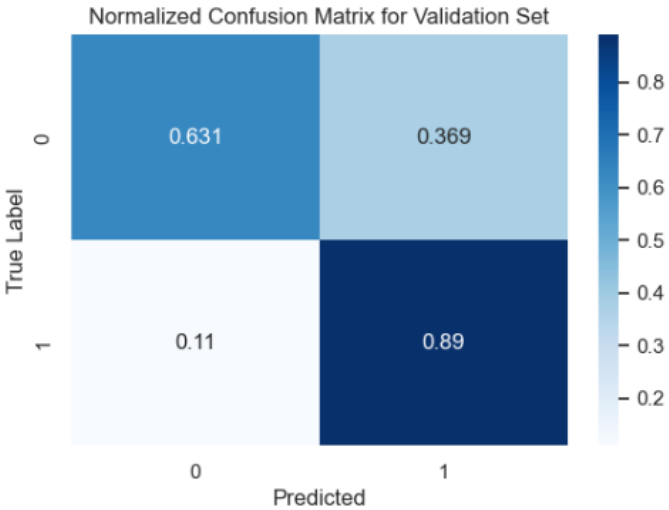
במהלך בחירת ההיפרפרמטרים למודל, בדקנו את ההשפעה של מספר עצים שונים (400, 500, ו-600) על הביצועים. עם פחות עצים (400-500), המודל הראה שונות גבוהה יותר ופחות יציבות בתוצאות, מה שהוביל לדיוק נמוך יותר. ככל שהוספנו יותר עצים, המודל נעשה יציב ומדויק יותר, ולכן בחרנו ב-600 עצים, למרות עלות החישוב הגבוהה יותר. לאחר שבחנו את הגרף שמציג את הקשר בין עומק העץ לדיוק המודל, החלטנו להוריד את העומק המקסימלי של העץ מ35 ל-10 למרות הבחירה של הGrid. הבחירה הזו נעשתה מכיוון שראינו שמעל לעומק זה, דיוק הוולידציה מתייצב ולא משתפר יותר, בעוד שדיוק האימון ממשיך לעלות. מצב זה מעיד על תחילת התופעה של Overfitting, שבה המודל מתאים את עצמו יתר על המידה לנתוני האימון ואינו מצליח להכליל היטב על נתונים חדשים. באמצעות הקטנת עומק העץ, הצלחנו לשמור על איזון טוב יותר בין דיוק המודל לבין יכולתו להכליל על נתונים שלא נראו באימון, ובכך לשפר את ביצועי המודל הכלליים.

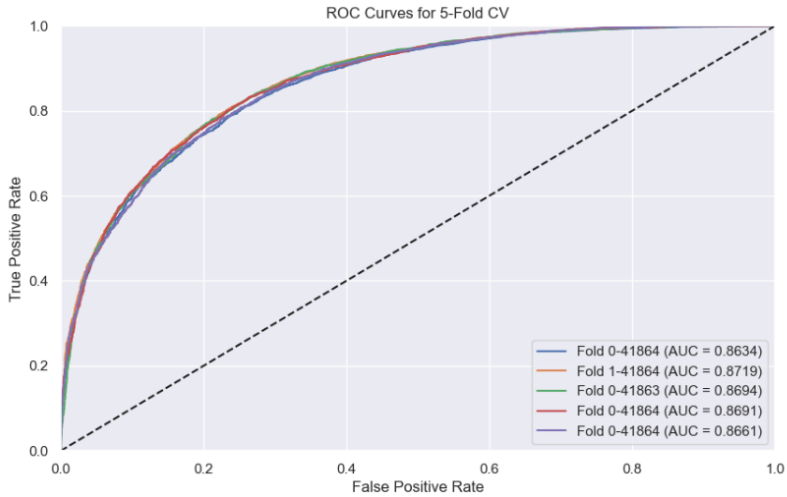
**Feature importance on data while using Random Forest:**

נספח 16




## Roc ant correlation Matrix after using 15 important features




```
Average AUC across folds: 0.8680
Validation Accuracy: 0.7700
Test AUC on validation set: 0.8636
```

## Roc ant correlation Matrix after using 15 important features and CalibratedClassifierCV:

נספח 17