**פרויקט**

**Machine Learning**

**חלק ב'**



מגישים :

עומר ויצמן – 205505928

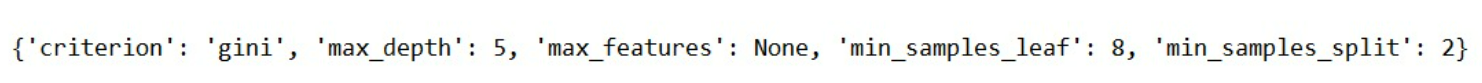
מיכאל אלפסי - 308307776

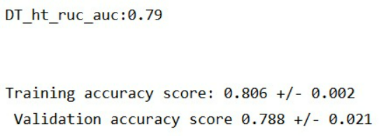
**Decision Trees**

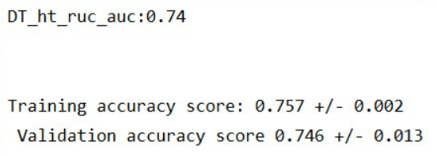
1. בתהליך hyperparameter tuning בוחרים את הצירוף של הפרמטרים שיביאו לאופטימיזציה את דיוק של אלגוריתם הלמידה. החלטנו להשתמש בשיטת CVgrid בשיטה זו מחפשים את צירוף ה- hyperparameter באופן אקראי על ידי הגדרת מספר האיטרציות הרצויות. בחרנו בשיטה זו כי הגדרנו טווח ערכים גדול בפרמטרים עם מספר רב של צירופים האפשריים ולכן מעבר על כל הצירופים האפשריים היה זמן ריצה ארוך. מעבר לשימוש בCVgrid עשינו בדיקה על כל פרמטר בנפרד על מנת לבחור בצורה פרטנית כל ערך במודל ולהשפיע עליו. לאחר מכן בדקנו את תוצאות המודל עם ערכים שקיבלנו באמצעות CVgrid לעומת ערכים שבחרנו באופן פרטני ולקחנו את הגבוה בינהם. להלן הפירוט על כיוונון הערכים במודל בצורה פרטנית:

* **Max depth** -העומק המקסימלי של העץ. נכוון את Max depth כדי לשלוט בגודל של העץ ולכן במורכבות של העץ. ככל ש Max depth גדול יותר העץ מורכב יותר יש יותר פיצולים ועלול להיווצר מצב של overfitting . אך ככל ש Max depth קטן העץ קטן ועלול להיווצר underfitting. ל hyperparameter זה בחנו את הערכים [1,10]. הציון מקסימלי התקבל הערך 7.
* **Criterion**- ה hyperparameterבוחר את הקריטריון לחישוב איכות ה split.

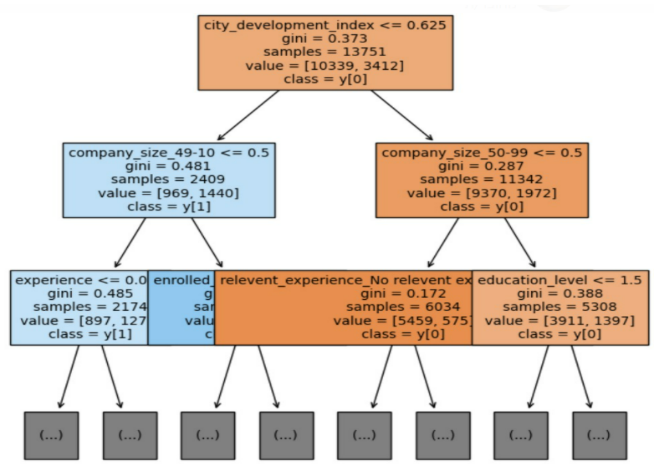
Gini Impurity- בחרנו בקריטריון זה כי קיבלנו ציון גבוה יותר מאשר השימוש באינטרופיה מדד ג'יני מחשב כמה פעמים sample שנבחר באופן אקראי מהdataset יתויג עם ה label הלא נכון.

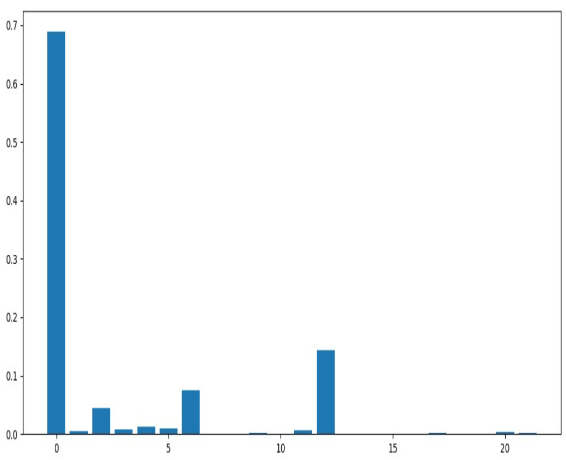
* **min\_samples\_leaf**- המספר המינימלי של samples הנדרשים להיות בעלה. המוטיבציה לכוון את פרמטר זה הוא בשביל להימנע ממצב של overfitting.ערך גבוהה יותר יגדיל את מספר התצפיות הנדרשות בעלה ולכן יגביל את תהליך הלמידה של העץ, ילמד פחות את ה train set ויעזור למנוע מצב של overfitting. בחרנו לבדוק את הערכים [1,40]. הציון מתקבל כאשר מספר ה samples המינימלי הוא 35.
* **min\_samples\_split**- המספר המינימלי של samples כדי לבצע פיצול ב node internal. המוטיבציה בלכוון את המשתנה כדי להימנע overfitting . ככל hyperparameter גדל כך העץ מוגבל ביותר samples לכל node והוא מונע מהמודל ללמוד קשרים ספציפיים לtrain set. עם זאת ערך גבוהה יותר עלול לגרום ל underfitting. בחרנו לבדוק את הפרמט בערכים בין [2,40], הציון המקסימלי יתקבל בערך 40.
*  הקונפיגורציה הטובה ביותר שהתקבלה בעזרת CVgrid:

הציון שהתקבל בעזרת הקונפגרציה הנ"ל:

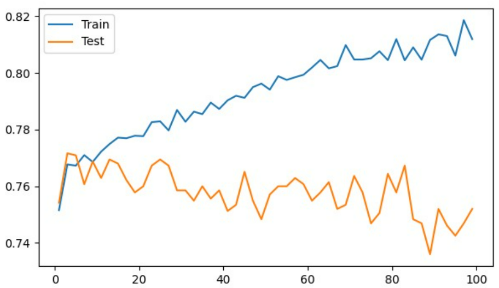
* הקונפיגורציה הטובה ביותר שהתקבלה בעזרת פונקציות לכל פרמטר:
*  הציון שהתקבל בעזרת הקונפגרציה הנ"ל:

2. אחד ההיתרונות של עצי החלטה על פני מודלים אחרים הוא יכולת הסברה של המודל. המודל מורכב מחוקים של if- then שקל לבני אדם לעקוב אחר החלטות שעץ החלטה עושה, לפרש אותן ולהבין למה המודל קיבל החלטה כזו או אחרת. ניתן להשתמש גם בהצגה גרפית של המודל כדי להבין את החלטותיו.

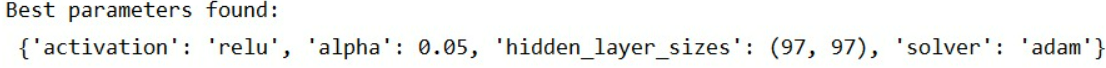
3. קיבלנו עץ עם max\_depth של 5 ולכן כדי שיהיה ניתן לראות בירור את הגרף חתכנו את העץ ל max\_depth של 3.

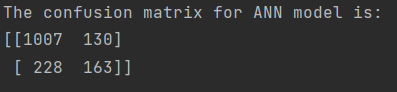
* על ידי התבוננת במבנה העץ ניתן להבין את חשיבות של כל Feature. זאת בגלל שהאלגוריתם לבניית העץ הוא אלגוריתם חמדן כלומר Feature עם חשיבות גבוהה יותר ימצא קרוב יותר לשורש. ניתן לראות כי City\_development\_index הוא השורש ולכן לו החשיבות הגבוהה ביותר במודל וכי הsplit הטוב ביותר הוא 0.625 אחריו ה- company\_size כאשר הsplit הטוב ביותר הוא בערך 0.5.
* כדי להבין מה החשיבות של כל feature ניתן להשתמש בפונקציה של feature\_importances. הפונקציה מחזירה את החשיבות של כל Feature. ככל שהמספר גבוה יותר כך הפיצ'ר חשוב יותר. ניתן לראות בגרף כי ל-City\_development\_index יש חשיבות של 0.684 והוא החשוב ביותר וכי company\_size (50-99) הוא השני בחשיבתו עם ציון של 0.14 לשאר הפיצ'רים יש חשיבות יחסית דומה. תוצאת אלו מתיישבות עם התוצאות של הסעיף הקודם זאת בגלל שהאלגוריתם איתו בנינו את העץ ההחלטה הוא אלגוריתם חמדן.

**Artificial Neural Networks**

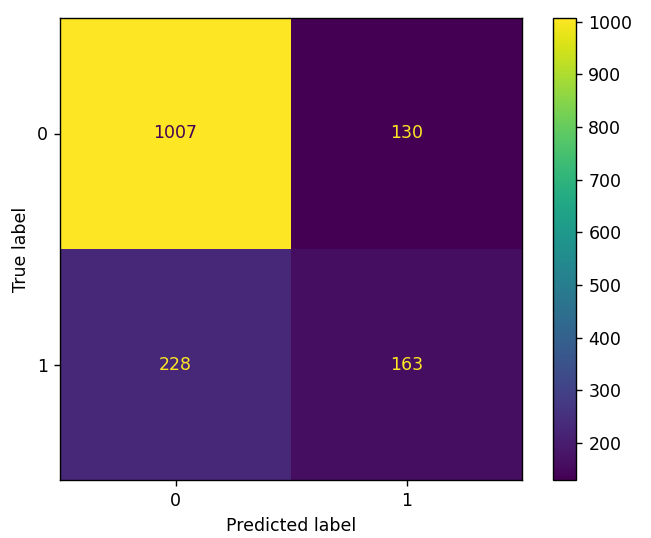
1. לפי ערכי ברירת המחדל אנו יכולים לראות כי המודל יבחר ליצור שכבת נוירונים אחת שהיא בעצם גם שכבת הכניסה המורכבת מכ-100 נוירונים. כלומר יהיו כ-100 נוירונים שכל אחד מהם יהיה מורכב ממכפלה וקטורית של וקטור ערכי התצפית של כל פיצ'ר אל וקטור המשקולות בהתאמה. קיבלנו כי רמת דיוק של ערכי ברירת המחדל על הtrain set הוא: אנו יודעים כי ערכי ברירת המחדל ייצרו תנאי עצירה מסוימים שימנעו מהמודל להגיע לרמת דיוק מירבית. על Validation set קבלנו כי הדיוק הוא:
2. החלטנו לבצע Hyperparameter Tuning על הפרמרטים הבאים:

* **Hidden layer size**: פרמטר זה בוחן גם את מספר השכבות הפנימיות, השכבות בין הinput layer- לoutput layer של המודל וגם את מספר הנוירונים לכל שכבה. על מנת להחליט את הצירוף הטוב ביותר הרצנו לולאה הבודקת את הדיוק לכל צירוף של שכבה ומספר נוירונים שווים על train set ועל Validation set . ניתן להסיק מהגרף כי הנקודות (17,57, 63, 97, 77) יכולות להביא אותנו למודל עם הדיוק הגבוה ביותר. לאחר מכן, השתמשנו בgrid search על מנת למצוא מתוך מספר נקודות ששמנו לב אליהן קודם את הטובה ביותר. בחרנו את הנקודה (97,97) מכיוון שהיא הביאה אותנו לציון הגבוה יותר. המוטיבציה בכיוון פרמטר זה נובעת מכך שבעזרת כמות נוירונים גבוהה ניתן להגיע למימוש פיתרון מסובך ע"י כך שכל נוירון מבצע משימה אחת ספציפית (הפעלת פונקציה על מכפלת וקטורי הפיצ'רים) וכך ניתן להגיע לדיוק גבוה. הסיכון הוא שכל תוספת של נוירון מעלה את זמן הריצה ואת הסיכוי לoverfitting ולכן יש צורך לכוון אותו.
* **Activity**: פרמטר זה מגדיר את Activation function כדי לעבד את הinput שמגיע מן הנוירון. החלטנו בעזרת grid search לבחון איזו פונקציה תעלה הכי הרבה את הדיוק של המודל. בחרנו לבדוק את Activation functions relu, tanh, identity ו- logistic. בבדיקה מצאנו כי הפונקציה ‘relu’ תמקסם את הדיוק .המוטיבציה בכוונון Activation function היא היכולת לעזור לנו לסווג את התצפיות בצורה הנכונה ביותר.
* **Solver**: פרמטר זה קובע את שיטת תיקון המשקולות שתיעשה על מנת למזער את הטעויות. בחרנו לבחון את השיטות adam, lbfgs ו-sgd. על פי grid search שביצענו קיבלנו כי נבחר בadam- שזוהי שיטה שמתאימה יותר כאשר יש אלפי תצפיות כמו במקרה שלנו. המוטיבציה בכיוון פרמטר זה היא שכיוונון המשקולות היא הדרך של המודל למזער את הטעויות וכך להשתפר. לכן, כדאי לנו להתאים את צורת כוונון המשקולות אל הדאטה שלנו.

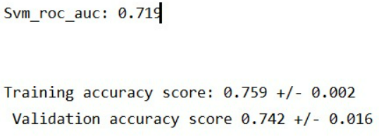
הקונפיגורציה הטובה ביותר והציון שהתקבל בעזרת הקונפגרציה:



1. מטריצת מבוכה עבור 2 המחלקות:

מטריצת מבוכה נותנת לנו מידע על ביצועי המודל, כמה הוא חזה נכון וכמה לא נכון ומהווה סיכום של תוצאות חיזוי על בעיית הסיווג. מספר התחזיות הנכונות והלא נכונות מסוכמים עם ערכי ספירה ומחולקים לפי כל מחלקה (0,1). ניתן לראות כי המודל חוזה בצורה יחסית מדוייקת את מחלקה 0 (לא נשאר בעבודה) עם false negative של 228 מתוך 1238 שחזה שיעזבו, לעומת זאת המודל התקשה בחיזוי של מחלקה 1 (נשאר בעבודה) עם false positive של 130 מתוך 293 שחזה שיאשרו.

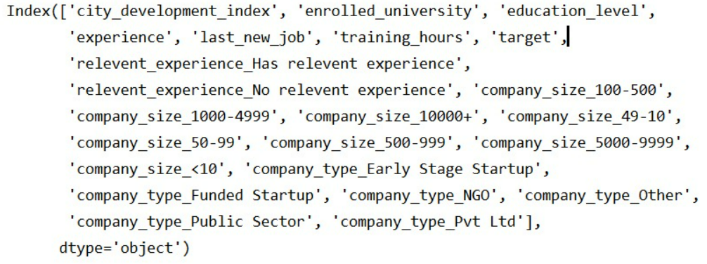
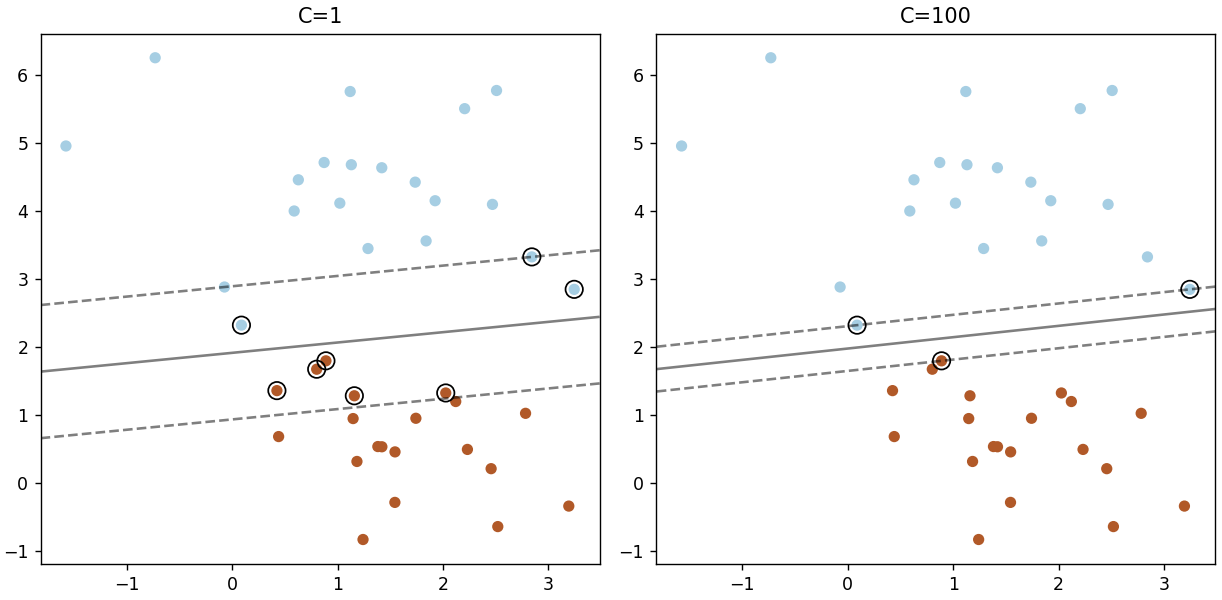
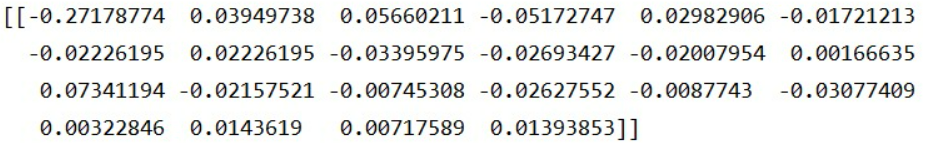
המטריצה תעזור לנו להבין כי המודל יחסית יכול לסייע לחברה למרות הציון שנשחשב נמוךך (0.76) כי על פי דעתנו ישנה חשיבות יותר גדולה לחברה לדעת מי רוצה לעזוב (כאמור המודל דיי מוצלח במשימה זו) לעומת המידע עבור עובדים שרוצים גם ככה להישאר בחברה.

**SVM**

1. תוצאת המודל עם פרמרטרים דיפולטיביים:

על מנת למצוא את הקונפיגורציה המיטבית נבצע Hyperparameter Tuning בשיטת random\_search כדי למקסם את הדיוק של תהליך הלמידה.   
כיוונו את הפרמטרים הבאים:

* **Kernel**- נשתמש בפרמטר זה כברירת מחדל בLinear על מנת למצוא בהמשך את משוואת הישר המפריד.
* **C**- מייצג את ה"קנס" שניתן לכל sample הנמצאת predict class zone של הclass השני, ככל שהערך גבוהה יותר כך המודל יסווג את הdata נכון יותר תוצאות רנדום והציון שהתקבל בעזרת הקונפגרציה:

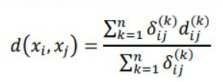
1. 

על מנת לקבל את משוואת הישר השתמשנו בפונקציית linearSVC אשר מחזירה את השיפוע והחותך לכל פיצ'ר. נבחן את השפעת הפיצ'רים על בניית הקו הישר במודל. הפיצ'רים שהשיפוע שלהם יהיה גדול יותר בערך מוחלט נסיק כי יש להם יותר השפעה. ניתן לראות גם פה כי City\_development\_index (שיפוע 0.27) הוא

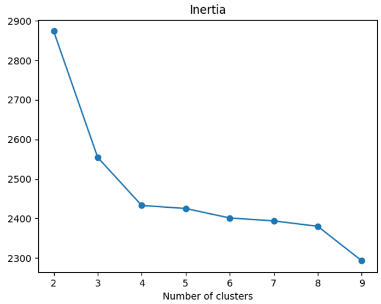
המשפיע ביותר כמו בתוצאות הקודמות שקיבלנו.

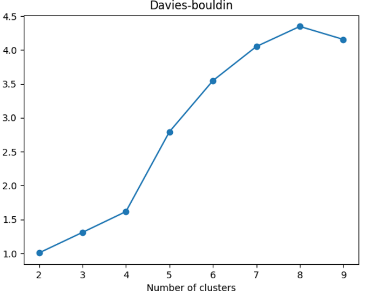
**Clustering**

1. במודלים שהרצנו עד עכשיו השתמשנו בעובדה כי ה data set הוא supervised , כלומר שלכל sample קיים label. בשונה משאר המודלים במקרה הזה ה data set שנריץ עליו את האלגוריתם יהיה ללא labeling ויכלול אך ורק את ערכי הפיצ'רים השונים לכל sample ללא שיוך ל class (unsupervised). באלגוריתם זה בשונה משאר המודלים בהם התבקשנו לסווג כל sample בודדת לאחת מתוך 2 class, אנחנו ננסה ע"פ "דימיון" ו"אי דימיון" של תצפיות לתצפיות אחרות, לסווג אותן למספר מחלקות שונות ונמצא את מספר ערכי k שיהוו את מספר הclusters אליהן יחולקו samples שלנובצורה האופטימלית.
2. בחרנו סט פיצ'ירים מעורב המכיל פיצ'רים רציפים ( training\_hours, city\_development\_index)

ופיצ'רים קטגוריאליים (experience, education\_level וכו'), כיוון שהסט שבחרנו מעורב נשתמש במטריקת מדידת המרחק .Gower distance תחילה עבור מדדי המרחק הבינאריים (relevent\_experience וכו')/ קטגוריאליים (company\_size וכו') / אורדינליים(education\_level וכו') נשתמש במדד מרחק Hamming ,(הפיצ'רים הרציפים שלנו כבר מנורמלי בטווח [0,1] כנדרש) ולבסוף נשלב בין המדיים בעזרת מדד Gower:

נצפה לראות מספר אפיונים שונים שיתבטאו בקלסטרים שיצאו לנו, כגון: קלסטר המאפיין עובדים עם מאפיינים איכותיים (השכלה גבוהה, ניסיון עשיר, שייכות לעיר איכותית וכו'), עובדים עם מאפיינים נורמטייבים, עובדים עם מאפיינים חלשים.

1. ****על מנת להחליט מהו מספר ה-clusters הנכון ביותר לחלוקת הדאטה שלנו נבחר מספר קריטריוני החלטה:

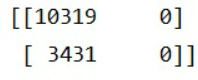
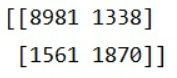
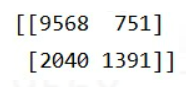
* **Inertia**- קריטריון זה מודד את המרחק של התצפיות ממיקום נקודת מרכז המחלקה. כלומר, ככל שהמרחק קטן יותר ככה נדע שהסיווג נעשה בצורה טובה יותר. ע"פ הגרף של מדד זה ניתן לראות כי כדאי לנו לחלק את הדאטה לכ-9 clusters שונים.
* **Silhouette**- קריטריון זה מודד מה המרחק של הנקודות ששייכות לcluster מסוים אל ה-cluster שלהן ביחס למרחק שלהן לcluster אחר הקרוב ביותר. המדד נע בטווח בין [-1,1] כך שככל שהערך גדול יותר ככה המרחק בין הclusters גדול יותר שזאת השאיפה שלנו. ע"פ הגרף של מדד זה ניתן לראות כי כדאי לנו לחלק את הדאטה לכ-2 clusters שונים.
* **Davies\_Bouldin**- קריטריון זה משווה בין cluster אל ה-cluster שהכי "דומה" לו כאשר הדימיון נמדד ביחס בין המרחקים בין הנקודות שבתוך ה-cluster (הצפיפות שלו) לבין המרחק בין שני ה-clusters. טווח הערכים הוא [0,∞] כאשר השאיפה שלנו היא לערך הנמוך ביותר הניתן. ע"פ הגרף של מדד זה ניתן לראות כי כדאי לנו לחלק את הדאטה לכ-2 clusters שונים.

- לסיכום ניתן לראות כי ע"פ המדדים שלנו אין החלטה חד משמעית אבל 2 קטגוריות הצביעו על חלוקה ל- 2 clusters ולכן נבחר K=2.

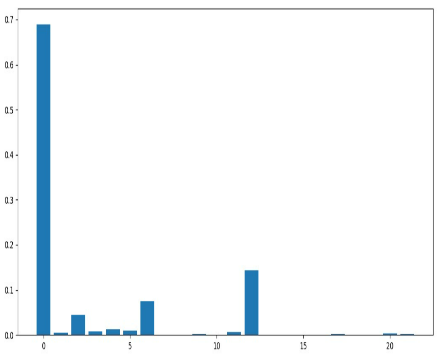
**Evaluation**

ישנן מטריקות שונות לבחור את מודל הלמידה האופטימלי מבין המודלים השונים שבצענו על dataset. משימת הלמידה שבצענו היא מסוגClassification . נבצע Classification Metrices לכל אחד משלושת המודלים, מטריצה זו נותנת לנו מידע על ביצועי המודל, כמה הוא חזה נכון וכמה לא נכון. ניתן לראות כי המודל שדייק בדרך הטובה ביותר בסיווג שנתן עם כמות false positive וfalse negative הקטנה ביותר הוא מודל עצי החלטה. בחרנו להסתמך על המטריקה ruc\_auc (דאטה לא מאוזן) המודדת את הרמה בה המודל יכול להבדיל בין classes שונים. כמה שהמדד גבוהה יותר כך יכולת הזיהוי של המודל טובה יותר הערכים נעים בין [0,1]. ניתן לראות כי העץ קיבל את ערכי המדד הגדולים ביותר מכל שלושת המודלים.

* בנוסף אנו רואים כי מודל הsvm שלנו מסווג רק את המחלקה 0 (לא נשאר בעבודה) ולא מצליח לסווג את מחלקה 1 כלל וגם קיבל ציון נמוך במדד ruc\_auc לכן אנו מניחים שישנה בעיה בהתאמת המודל לסוג הנתונים שלנו / יש לנו טעות מסויימת שלא הצלחנו לפתור בעצמנו.

ANN: DT: SVM:

**Improvement**

1. במהלך העבודה התגלה כי חלק מהדאטה הנמצאת בפיצ'רים אינה תורמת ללימוד המשימה ואף גורמת לסרבול המודל, זמן ריצה ארוך, רעש מיותר ועוד. החלטנו לאחד קטגוריות מסויימות שלא תרמו ללימוד המשימה על פי הגרף של feature\_importances. הפיצ'רים שבחרנו להשאיר הינם: city\_development\_index, enrolled\_university, education\_level, experience, last\_new\_job, relevent\_experience בנוסף עמודות 11, 12 מתייחסות לגודל החברה 10-49 ו50-99 אותם נאחד לעמודה אחת וכל שאר האפוציות של גדלי החברה נאחד יחד גם כן ובכך נוריד את מספר הפצ'ירים המיותרים. אנו משערים שפעולה זו תעזור להקטנת העץ ובכך תשפר את תהליך הלמידה.

לשיפור המודל החלטנו להשתמש ב random forest, מודל זה הוא טוב יותר ממודל העץ החלטה הרגיל מכיוון שהוא מונע over-fitting בצורה טובה בכך שהוא מורכב ממספר עצים שמבוססים על חלקים שונים מהדאטה שלנו. בעיה נוספת במודל עצי ההחלטה הוא היותו מודל גרידי אשר בוחר פיצ'ר ולא חוזר אחורה ומפצל באמצעותו ענפים נוספים וממשיך הלאה לפיצ'ר הבא ולכן לא בטוח שימצא תמיד את הפתרון האופטימלי, לעומתו היתרון ב random forest הוא שהוא בונה המון עצי החלטה "קטנים" ולכן פחות גרידי ויכול לתת יותר חשיבות למידע בכל פיצ'ר ויכול לסייע בהגעה לתוצאה אופטימלית. אנו משערים שהשימוש במודל יביא לעלייה במדד הruc\_auc שלנו.

2. בצענו את השינויים בדאטה באופן שתארנו בסעיף הקודםת בנוסף נחפש שוב עבור מודל ה random forest את הערכים האופטימליים עבורו. בעזרת CVgrid נכוון את הפרמטרים הבאים:

* **N\_estimators**: מספר העצים הסופי
* **Max\_features**: קובע כמה features יילקחו בכל חלוקה אקראית של features.
* **Max\_depth**: עומק מקסימלי של העץ
* **Min\_samples\_split**: מספר מינימילי של פיצול בין תצפיות בכל צומת בעץ
* **Min\_samples\_leaf**: מספר מינימילי של תצפיות כדי להיחשב כעלה
* **Bootstrap**: שיטת חלוקת הדאטה. ביכולתינו להחליט אם להשתמש בשיטה זו או לא

הקונפגרציה הטובה ביותר הינה:



תוצאת המודל החדש הינה:

כפי שניתן לראות אכן הצלחנו לשפר מעט את המודל ולעלות את מדד הruc\_auc ב2%. לא הצלחנו לשפר משמעותית כי לדעתנו מיצוי הדאטה האפשרי לטובת החיזוי שלנו התבצע יחסית בצורה טובה כבר במודל עצי ההחלטה שהרצנו בסעיף הראשון.