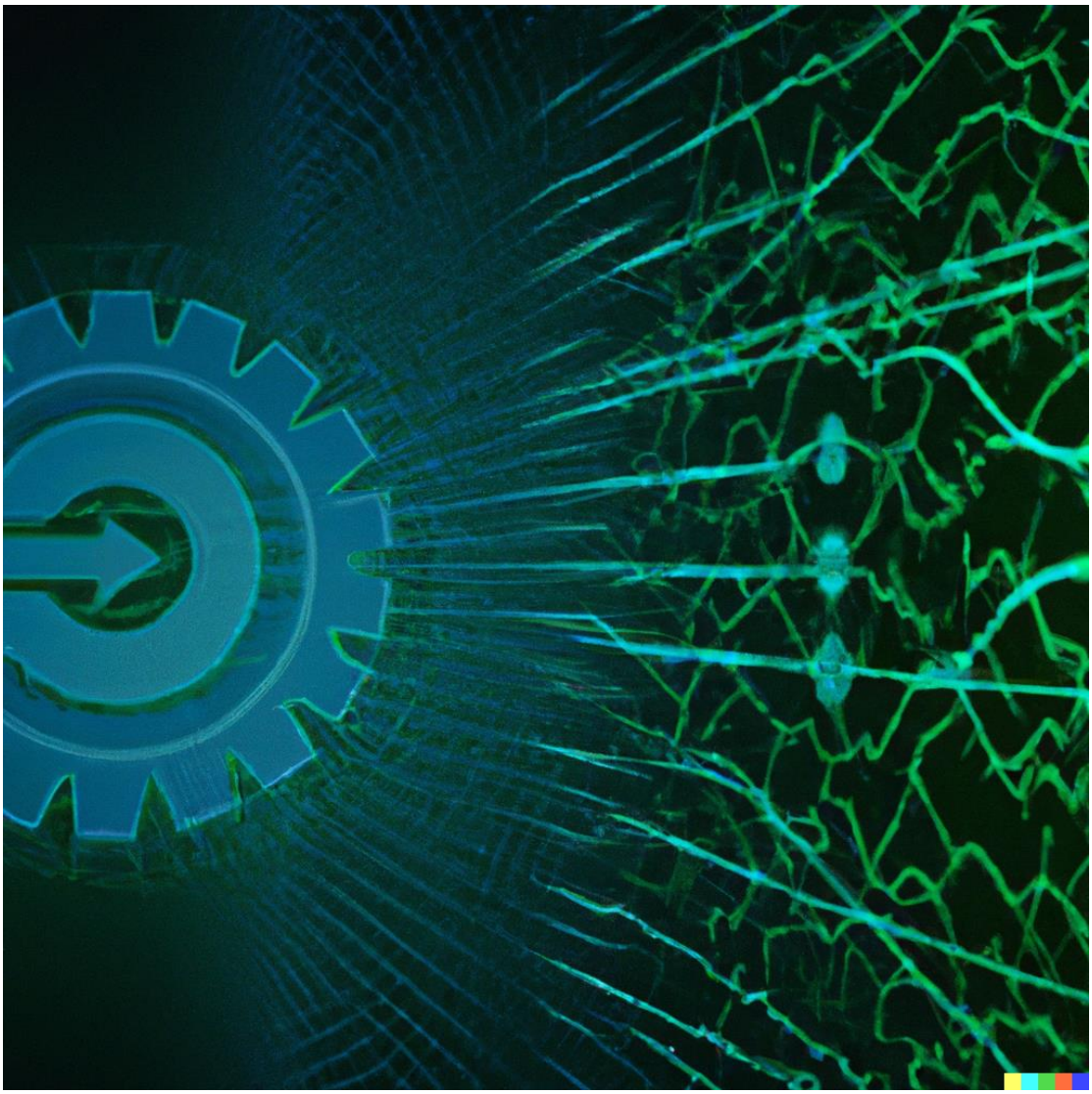
Machine learning project:

part 2



מגישים:

ענבר דובדבני: 206104028

אמיר יטיב: 207128513

תוכן עניינים

שינויים מחלק א'

לאחר בדיקה של הנתונים שלנו מחלק א', ביצענו קידוד למשתנים הקטגוריאליים שלנו. נוסף על כך בחרנו לבצע איזון של הנתונים על ידי הוספת רשומות נוספות מסוג משרה עוינת כך שכרגע ישנן 17678 רשומות סה"כ שמותכן 13599 הן משרות תמימות (77%) וכ-4079 משרות עויינות (23%). לבסוף עשינו נרמול ואדפטציות לכלל המשתנים בהתאם וכך שבשלב ה- Feature Selection השתמשנו בשיטת Fischer score ובחרנו את המאפיינים בעלי הניקוד הגבוה ביותר לעיינו ובחרנו ב-24 הפיצ'רים הבאים:

feature	Fischer Score
has_company_logo	209.01
Europe	1066.32
has_questions	954.1
Bachelor's Degree	813.82
description_ratio	768.3
precentilne_company_profile_num_of_words	741.02
Mid-Senior level	417.3
Associate	406.29

feature	Fischer Score
title_ratio	401.4
Full-time	335.49
company_profile_ratio	262.54
country_count_encoded_scaled	241.04
Contract	222.37
Some High School Coursework	195.35
Unspecified	182.21
industry_ratio	124.26

feature	Fischer Score
Asia	117.55
Not Applicable	117.35
requirements_ratio	115.28
benefits_ratio	113.78
North America	110.22
missing_data_required_education	75.57
Internship	75.37
Entry level	75.16

הכנת הנתונים לאימון ובחינת מערכת לומדת

עבור המודלים שלנו בחרנו להשתמש בשיטת holdout על סט הנתונים שלנו. חילקנו את סט הנתונים בקובץ XY_train לסט אימון (80%) וסט בחינה (20%). חלוקת 20-80 היא חלוקה שכיחה המאפשרת סט בחינה מספיק רחב לבדיקת המודלים, ומותירה מספיק תצפיות לאימון המודלים בצורה רחבה (במקרה שלנו 14142 תצפיות אימון ו- 3536 תצפיות מבחן).

Decision Trees- עץ החלטה

כוונון פרמטרים

נבחר לכוונן את הפרמטרים בעזרת Grid Search הבוחן את כל הקונפיגורציות האפשריות בטווח ערכים נתון ובכך בהכרח נמצא את הקונפיגורציה הטובה ביותר מבין האפשרויות.

Criterion- הקריטריון הוא פונקציה המשמשת להערכת איכות הפיצול בעץ החלטות. בחרנו לכוונן פרמטר זה כיוון ששיטת פיצול העץ הינה חשובה. ישנם 2 קריטריונים: Gini הבודק את הסיכוי שדגימה רנדומלית תסווג לא נכון בתוך הענף ו- Entropy המודד את רמת האי וודאות. נרצה למזער את הקריטריון הנבחר.

Max depth- כוונון פרמטר זה נבחר היות ונרצה מצד להגביל את עומק העץ על מנת למנוע Overfitting מצב בו מודל מאומן טוב מדי על נתוני האימון וכתוצאה מכך, הוא אינו מסוגל להכליל היטב לנתונים חדשים, ומצד שני לא לקבל עומק עץ קטן עלול לסווג בצורה לא מספיק טובה את משתנה המטרה. טווח הערכים שעליהם נרוץ הינו בין 1-28 בקפיצות של 1, כיוון שעומק העץ המלא שלנו הוא 28. (נספח 1)

Max features - קריטריון המייצג את מספר המאפיינים אשר מבניהם נבחר את הצומת הבאה בעץ. ככל שמספר המאפיינים יהיה קטן יותר ייתכן ולא נבחר את המאפיינים המשפיעים (under fitting) בעוד שבמידה ונבחר מתוך מספר גדול של מאפיינים אנו בוחרים אותם על סמך סט האימון וישנו חשש ל- over fitting על-כן חשוב למצוא את האיזון הנכון. לכן, נרצה לבחון את הערכים ['sqrt', 'log2', None]. 'None' נבחר היות והוא משמש כאופציית הבסיס, נרצה שעץ ההחלטה יחקור את כל הפיצולים האפשריים בהתבסס על כלל התכונות. 'sqrt', 'log2' נבחרו בהתחשב בכך שהן יכולות לעזור להפחית את העלות החישובית ואת הפוטנציאל התאמת יתר על ידי הגבלת מספר התכונות החשובות.

עבור סט הוולידציה קיבלנו כי הקומבינציה האופטימלית עבור הפרמטרים שלנו היא:

DecisionTreeClassifier (criterion='entropy', max_depth=8, max_features=None)

כך שעבור מדד ה – AUC-ROC נקבל:

אחוזי הדיוק המתקבלים על סט האימון: 99.98%

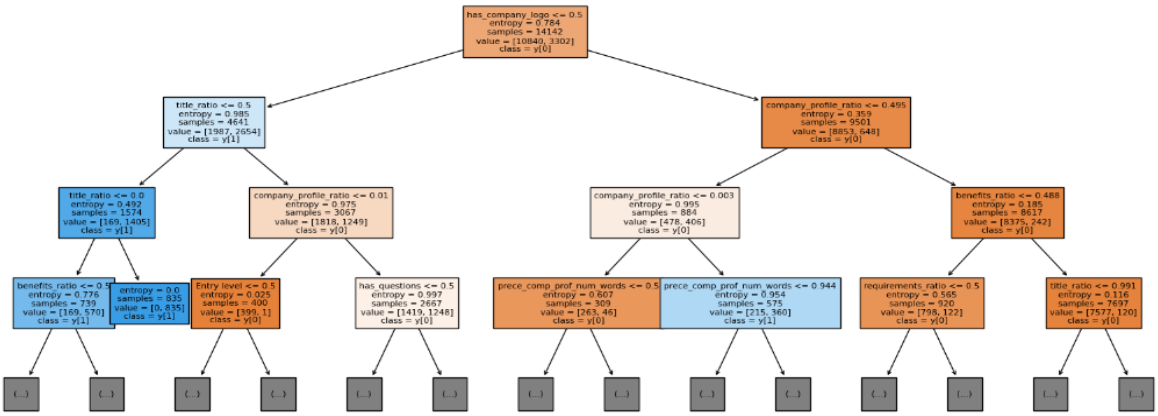
אחוזי הדיוק המתקבלים על סט הוולידציה: 97.13%

אחוזי הדיוק המתקבלים על סט הבחינה: 93.5%

ניתן להסיק כי המודל שלנו קרא בצורה כמעט מלאה את סט האימונים היות ותוצאות המדד כמעט שואפות ל-1. סט הוולידציה גם הוא קיבל ציון גבוהה מאוד אך נמוך מסט האימון ועל כן נפחת החשש שיהיה התאמת יתר. לבסוף נראה כי על סט המבחן שלנו שלא נראה על ידי המודל לפני גם כן קיבל ציון גבוה יחסית אך נמוך מהשניים האחרים ועל כן ניתן לומר שהמודל מסווג באופן טוב יחסית את הנתונים שהוא מקבל.

הכוונה במושג interpretability שנאמר על מודל עץ ההחלטה היא מודל עץ ההחלטה נותן לנו הבנה אילו פיצ'רים תורמים לנו יותר עבור המודל ואילו פחות ובנוסף מראה לנו את הדרך כולה שבה בוא פועל לקבל תוצאת הסיווג. ניתן לראות זאת למשל על ידי האופן הפשוט שבו נבנים כללי ההחלטה של המודל- שימוש בתנאי if-else בכל צומת, על ידי מבנה העץ המעיד על חשיבות הפיצ'רים בכל רמה (רמת חשיבות יורדת כל שיורדים בעץ) וכמובן הוא מודל שממחיש ממש בצורה גרפית את דרך ההחלטה.

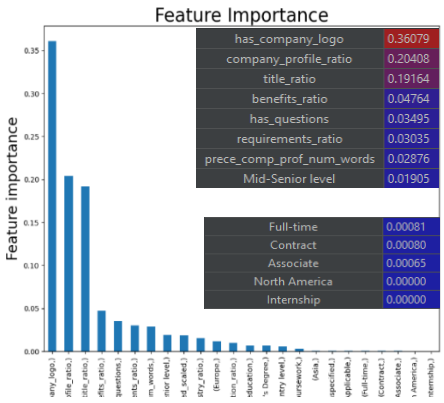
גרף העץ שהתקבל:



תובנות:

- המאפיין company_profile_ratio חוזר בשני ענפי העץ המרכזיים כבר ברמה השנייה, ולכן ניתן להסיק מכך שהוא משתנה משמעותי בעת הסיווג בין המחלקות.
- ניתן לראות כי כבר עבור שני פיצ'רים בלבד- משרות המקיימות has_company_logo=0 (בינארי בנתונים שלנו) וגם שה- title_ratio הוא בין 0 ל-0.5 יהיה ניתן לסווג אותן כמשרות עויינות (class=1)
- ניתן לראות כי לאורך הענף הימני ביותר יש ירידה הדרגתית במדד ה-entropy עם סיווג של משרות תמימות. על כן ענף זה עשוי להעיד כי המשך הסיווגים עבור משרות שלא עומדות בתנאי הפיצ'רים (false- ענף ימני) בענף זה יעידו כי הן משרות תמימות.

features importance :



כפי שניתן לראות בערכי ה- features importance שלושת המאפיינים שקיבלו את הציון הגבוה ביותר הם: has_company_logo, company_profile_ratio, title_ratio כלומר, הם המאפיינים בעלי החשיבות הגבוה ביותר. זאת, בהתאמה לכך שאותם מאפיינים נבחרו כצמתים בראשית העץ עובדה המעידה על היותם משמעותיים בתהליך הסיווג. נציין בנוסף כי company_profile_ratio קיבל את הניקוד הכי גבוה במבחן פישר ועל כן נמצא גם בראש הטבלה הנוכחית. דבר המעיד כי לפיצ'ר יש הרבה השפעה בסיווג המשרות.

Artificial Neural Network

ברשת ANN אנו יוצרים נוירונים שידמו את מבנה המוח. המוח שלנו מבצע פעולות חישוביות מסובכות בעזרת נוירונים שכל אחד מהם מבצע פעולת חישוב פשוטה ומעביר את המידע לנוירון הבא. שכבת הכניסה היא שכבה שבנויה מנוירונים ככמות הfeatures שיש לנו במודל. מספר השכבות ומספר התאים מגדירים את גודל הרשת. יש לבחור רשת גדולה מספיק, אך לא גדולה מדי. רשת קטנה מדי לא תוכל לקרב בדיוק מספיק את המיפוי הנדרש, ואילו רשת גדולה מדי תמנע לימוד יעיל ועשויה לבצע overfitting.

במודל ברירת המחדל של MLPClassifier:

hidden_layer_sizes: tow layers with 100 neurons in each layer, **activation** : 'rely'**, solver**: 'adam' **alpha**: 0.0001, **batch_size**: 'auto' –(the minimum between 200 and the samples size), **learning rate init** : 0.001

כך שעבור מדד ה – AUC-ROC נקבל:

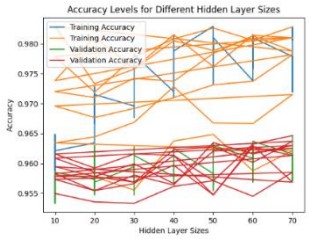
אחוזי הדיוק המתקבלים על סט האימון: 96.35%

אחוזי הדיוק המתקבלים על סט הוולידציה: 94.03%

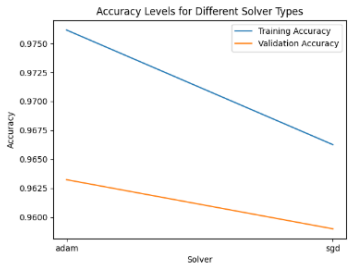
ניתן ללמוד מאחוזים אלו שאנו מקבלים אחוזי התאמה גבוהים מאוד על סט האימון מה

שמצביע על מודל טוב אך אולי גם על התאמת יתר, אך בעזרת סט הוולידציה אנו יכולים לראות שגם עליו אחוזי ההתאמה שקיבלנו הינם גבוהים ובהפרש של כ 2% וזה מאשש את זה שאנחנו לא נמצאים במצב של התאמת יתר.

את טווחי היפר-הפרמטרים בחרנו לפי בחירת המחדל של המודל, ולפי ניסיוניות שעשינו ועל פיהם הסקנו מה יכולים להיות הטווחים הטובים ביותר.



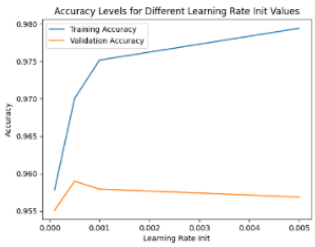
שכבות חבויות- כיוונון השכבות החבויות בעצם אחראי על השכבות החבויות בלבד (שכבת inputו שכבת outputו לא מושפעות מכוונון פרמטר זה) והוא קובע את מספר השכבות החבויות וגם את כמות הנוירונים בכל שכבה. ניסינו בעזרת ניסוי וטעיה לראות מה כמות השכבות הטובה ביותר (מצאנו שזה 2 שכבות) ולאחר מכן כוונו את ההיפר פרמטר הזה בעזרת לולאת for בכדי למצוא את כמות הנוירונים הטובה ביותר לכל שכבה. טווח הערכים שהכנסנו היה בין 10 ל-80. (נספח 2 -דוגמאות לגרפים של שכבה בודדת ו-3 שכבות)



סולבר- הסולבר אחראי על התאמת משקלי הרשת במהלך האימון על מנת למזער ככל הניתן את הטעויות. ישנן שתי שיטות אותן בדקנו על המודל ANN שבנינו:

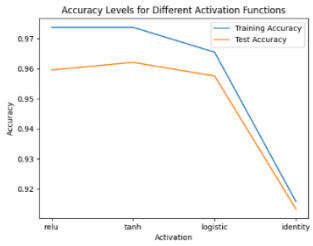
Adam (Adaptive Moment Estimation): הוא אלגוריתם אופטימיזציה אדפטיבית המשלב את ירידה בשיפוע עם מומנטום ו-RMSprop. הוא מחשב קצבי למידה אדפטיביים עבור כל פרמטר בהתבסס על הרגע הראשון והשני של ההדרגות. אלגוריתם זה הוא גם אלגוריתם בחירת המחדל של הרשת.

Stochastic Gradient Descent (SGD): הוא אלגוריתם אופטימיזציה שמעדכן את המשקולות בהתבסס על שיפועים של פונקציית ההפסד המחושבת על ידי אצווה אקראית של נתוני האימון, המתאימה במיוחד עם מערכי נתונים גדולים.



קצב למידה- קצב למידה הוא "גודל הצעד" אותו המודל עושה כאשר הוא לומד על הנתונים שהזנו לו. קצב למידה גבוה מידי כלול להוביל להתכנסות מהירה מידי של המודל, להפוך את התהליך לבלתי יציב ולחריגות. מצד שני קצב למידה נמוך מידי עלול להוביל להתכנסות איטית מידי של המודל ולהיתקע בפתרונות שאינם אופטימליים.

אקטיבציה- פונקציות האקטיבציה מיושמות על הפלטים של נוירונים קובעות אם יש להפעיל נוירון או לא על סמך הקלט שהוא מקבל. בדקנו על המודל שלנו ארבע פונקציות אקטיבציה שונות:

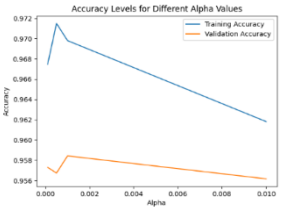


Sigmoid (Logistic): פונקציית הופכת את כל הערכים שבין לטווח ערכים שבין 0 ל-1 (מה שכבר עשינו על הנתונים שלנו בשלב הנרמול).

ReLU (Rectified Linear Unit): פונקציית הפעלה פופולרית (בחירת המחדל

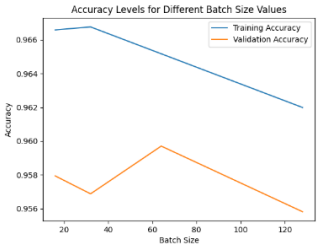
של המודל) שמגדירה את כל ערכי הקלט השליליים לאפס ושומרת על ערכים חיוביים ללא שינוי. שיטה זו יעילה מבחינה חישובית ועוזרת בבעיית קביעת השיפוע, ועוזרת למודל ללמוד מהר יותר.

Tanh (Hyperbolic Tangent): פונקציה היא שימושית עבור מודלים בהם יש צורך בכניסות שליליות להיות מוגדל לתפוקות שליליות.



Identity: משמש בדרך כלל במשימות רגרסיה שבהן ערכי הפלט צריכים להיות באותו טווח כמו ערכי היעד.

אלפא- היפר-פרמטר זה אחראי על ענישת המודל על הטעויות שלו. עם אלפא גדולה אנחנו עלולים להגיע למודל שלא לומד מספיק טוב אך עם אלפא קטנה מידי אנו עלולים להגיד למודל עם התאמת יתר.



גודל אצווה- גודל האצווה מתייחס לכמות הסמפלים שנלקחים ללמידה בכל פעם עד שהמודל "מתאים" את המודל הקיים למודל עם האצווה החדשה. כמות האצווה גם משפיע על כמות האיטרציות (מהירות התכנסות המודל) ועל כמות הזיכרון הנדרש.

כך שעבור מדד ה – AUC-ROC נקבל:

אחוזי הדיוק המתקבלים על סט האימון עם המודל הטוב ביותר: 96.25%

אחוזי הדיוק המתקבלים על סט הוולידציה עם המודל הטוב ביותר:94.14%

אחוזי הדיוק המתקבלים על סט הבחינה עם המודל הטוב ביותר: 94.81%

מנתונים אלה ניתן להסיק שבנינו מודל עם תוצאות חיזוי טובות על סט האימון והוולידציה וההפרש הנמוך (כ2%) מעיד על כך שאנחנו לא נמצאים במצב של התאמת יתר.

כפי שציפינו, המודל שלנו נותן תוצאות טובות ממודל בחירת המחדל יותר אך בהפרש ממש קטן (כ-0.1% על סט הוולידציה). יש לציין שבכל הרצה אנו נקבל ציונים שונים וההפרש הנמוך יכול שלא לייצג את ההבדל האמיתי בין מודל בחירת המחדל למודל הטוב ביותר- נוכל לקבל מצב שבו ההפרש גבוהה יותר וגם מצב שמודל ברירת המחדל יהיה אף טוב יותר.

היפר הפרמטרים של המודל הטוב ביותר:

```
model = MLPClassifier(hidden_layer_sizes=(30, 70), solver='adam', alpha=0.005,
activation='relu', batch_size=64, learning_rate_init=0.0005, max_iter=1000)
```

מטריצת המבוכה של המודל הטוב ביותר:

```
[2636  123]
[  46  731]
```

מהמטריצה הזו ניתן להסיק שהטעות בסיווג של סמפלים שהם תמימים וסווגו כעוינים (כ4.6%) ועוינים שסווגו כתמימים אינו (6.2%) לא גדול מאוד. יכול להיות שההפרש הזה נובע מהעובדה שניסינו לאזן את הנתונים שלנו וה-up-sampling שעשינו גרם למעט הטיה או מכך שלכל הרצה יש את ההטיות שלה וכך יצא בהרצה המקרית הזו .

SVM

בחרנו עבור מודל זה בהיפר פרמטרים הבאים:

param_grid = {'C': np.arange(1, 120, 6), 'dual': [True, False], 'tol': [1e-4, 1e-3, 1e-2]}

עבור סט הוולידציה קיבלנו כי הקומבינציה האופטימלית עבור הפרמטרים שלנו היא:

LinearSVC(C=13, dual=False, tol=0.01)

<u>כך שעבור מדד ה – AUC-ROC נקבל:</u>

אחוזי הדיוק המתקבלים על סט האימון: 96.14%

אחוזי הדיוק המתקבלים על סט הוולידציה: 96.06%

אחוזי הדיוק המתקבלים על סט הבחינה: 87%

טבלה מפורטת ב- (נספח 3)

rank_test_score	std_test_score	mean_test_score	std_train_score	mean_train_score	params	param_tol	param_dual	param_C
1	0.005882308	0.96061267	0.00057607	0.961350039	{'C': 13, 'dual': False, 'tol': 0.01}	0.01	FALSE	13
2	0.005882421	0.960610715	0.000574878	0.961350537	{'C': 19, 'dual': False, 'tol': 0.01}	0.01	FALSE	19
3	0.00588086	0.960610432	0.000575422	0.961350923	{'C': 7, 'dual': False, 'tol': 0.01}	0.01	FALSE	7

<u>משוואת הישר :</u>

$$y = 2.6801 - 0.5727 \cdot X_1 - 0.7525 \cdot X_2 - 0.2864 \cdot X_3 - 0.3401 \cdot X_4$$
$$-0.2521 \cdot X_5 - 0.3952 \cdot X_6 - 0.4014 \cdot X_7 - 0.3728 \cdot X_8 - 0.1448 \cdot X_9$$
$$-0.5489 \cdot X_{10} - 1.1585 \cdot X_{11} - 1.22035 \cdot X_{12} - 0.4670 \cdot X_{13} - 0.2878 \cdot X_{14}$$
$$-0.87033 \cdot X_{15} - 0.7369 \cdot X_{16} - 0.9262 \cdot X_{17} - 0.3278 \cdot X_{18} - 0.7430 \cdot X_{19}$$
$$-0.6953 \cdot X_{20} + 0.2385 \cdot X_{21} - 0.7165 \cdot X_{22} - 1.1391 \cdot X_{23} + 0.9691 \cdot X_{24}$$

X1 = country_count_encoded_scaled
X2 = missing_data_required_education
X3 = Unspecified
X4 = Some High School Coursework
X5 = Bachelor's Degree
X6 = Not Applicable
X7 = Mid-Senior level
X8 = Internship
X9 = Entry level
X10 = Associate
X11 = Full-time
X12 = Contract
X13 = North America
X14 = Europe
X15 = Asia
X16 = has_questions
X17 = has_company_logo
X18 = prece_comp_prof_num_words
X19 = company_profile_ratio
X20= description_ratio
X21= requirements_ratio
X22= benefits_ratio
X23= industry_ratio
X24= title_ratio

עבור משוואת הישר שמתקבלת עבור מודל הSVM המקדמים שהתקבלו מייצגים את התרומה של כל תכונה בפונקציית ההחלטה של ה-SVM. נראה כי מרבית הפיצ'רים שקיבלו אצלנו ציון גבוהה ב-Score Ficher עליו ביצענו את ההשוואה אינם מקבלים תרומה גבוה יחסית בהשוואה לשאר. אנו מניחים שהיות ומודל הSVM עושה התאמות בהתאם לוקטורי התמיכה ושואף למצוא את המישור המקסם את המרווח, היה עליו לעשות התאמות שונות שהביאו לתוצאות הנ"ל. כתוצאה מכך נעריך שלא נבחר במודל זה הן בגלל הנאמר לעייל והן בגלל תוצאות ה AUC-ROC הנמוכות יחסית שיצאו במודל זה בהשוואה לשאר המודלים.

Unsupervised Learning – Clustering

הרצנו את המודל על סט הנתונים כולו ולא על סט האימון והבחינה היות ומדובר בלמידה לא מונחית. לפיכך גם לא השתמשנו כלל בעמודה הקלאס שלנו היות ומכורך האלגוריתם הוא ייתן לנו את האשכולות שלפיו יש לחלק את הנתונים. בחרנו במודל זה לבצע תהליך PCA עם שתי קומפוננטות (PC1,PC2) על כלל הפיצ'רים שלנו היות והליך יפחית את הממדיות של מערך הנתונים שהינו חיוני לפני ביצוע אלגוריתם ה- K-medoids ובנוסף ייתן לנו את

מכלול התכונות האינפורמטיביות ביותר בנתונים. לפיכך, בסופו של דבר, תהליך זה יביא לפוטנציאל שיפור תוצאות האשכולות על ידי שימוש בהיבטים המבדילים ביותר של המקרים.

<u>ההיפר פרמטרים שבחרנו עבור מודל זה הם:</u>

```
k_medoids = KMedoids(n_clusters=3, metric='euclidean', max_iter=100, method="pam",
init="heuristic", random_state=10)
```

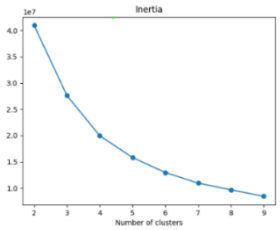
שיטת הלמידה שבחרנו היא PAM אשר מומלצת לשימוש באלגוריתם זה, עם מספר איטרציות מקסימאלי של 100 איטרציות. מטריקת הלמידה שבחרנו היא מדד המרחק האוקלידי המתאים לנתונים רציפים ובעל חשיבות לגודל ולקרבתם של הערכים. היות ונרמלנו את הערכים ועשינו את השלבים הבסיסיים ההכרחיים להביא בשלב ה- Pre-processing אנו נקווה לראות מספר לא גדול (קרוב למספר הקלאסים לבעיה האמיתית).

<u>בחירת הK האופטימלי:</u>

<u>נתבונן ב-3 פרמטרים</u>:

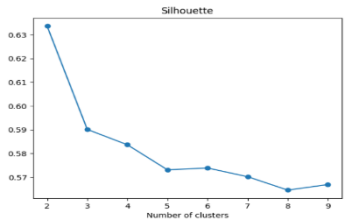
<u>Inertia</u> :

מדד זה מחושב על ידי סכום המרחקים בריבוע בין הנקודות בתוך אשכול למרכזו של האשכול ועל כן נרצה למזער את המדד. המדד אמנם פחות מתאים להחלטה על מספר האשכולות כיוון שהוא צפוי לרדת עם כל הוספת אשכול. עם זאת, נראה כי השינוי המשמעותי ביותר במדד הוא במעבר בין 2 ל-3 אשכולות (הירידה החדה ביותר). לכן נבחר k=3.

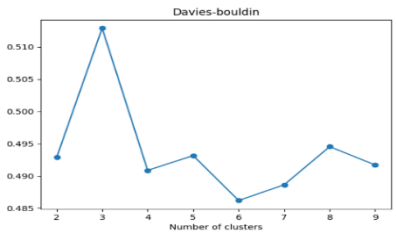


<u>Silhouette</u> :

מדד זה מראה עד כמה אובייקט דומה לאשכול שלו בהשוואה לאשכולות אחרים. המדד נע בין 1- ל 1. ציון של 1 מצביע על כך שהאובייקט מותאם בצורה מושלמת לאשכול שלו ולא מתאים לאשכולות שכנות. מצד שני, ציון של 1- מצביע על כך שהאובייקט לא מותאם לאשכול שלו ומותאם בצורה מושלמת לאשכול שכן. נרצה למקסם מדד זה על מנת להגיע למצב שהאשכולות במערך הנתונים מוגדרים היטב ומובחנים. לכן נבחר לפי הגרף k=2.

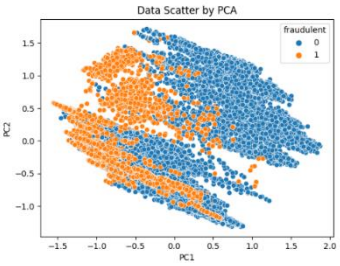


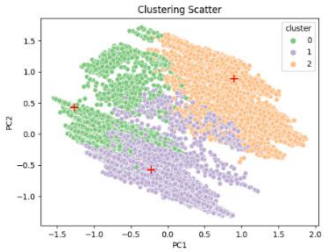
<u>Davies-Bouldin:</u> מדד זה מראה את הקיבוץ וההפרדה של אשכולות במערך נתונים. את המדד מחשבים על ידי חישוב המרחק הממוצע של כל הנקודות באשכול למרכז האשכולות. עבור מדד זה לראות את העלייה החדה ביותר. נראה כי גם פה העלייה המשמעותית ביותר במדד נמצאת במעבר בין 2 ל-3 אשכולות ולכן נבחר k=3.



על כן לאחר שכלול שלושת המדדים נבחר לקחת K=3.

על מנת לשער את שיוך שלושת האשכולות שקיבלנו, ביצענו ניתוח של הנתונים והצגנו גרף





לפני בחירת ה-K שלנו, בו אנו מראים את הפיזור בנתונים בהתאם ל PCA ולפי סיווג הנתונים, ובנוסף גרף של ביצוע המודל לאחר בחירת ה-K. אנו משערים כי החלוקה ל-3 אשכולות נובעת מהעובדה שהתכונות שבחרנו עשויות שלא ללכוד כראוי את ההפרדה הבסיסית של המחלקות היות ובשלב ה- Preprocessing ניתן היה למדל את התכונות המקוריות כראות עיינו. על כן יתכן שחלק מהתכונות עשויות שלא ללכוד כראוי את ההפרדה הבסיסית של המחלקות ועל כן גדל מספר האשכולות הסיווג. עם זאת, ניתן לראות שבאיור הימני נקודות רבות שמסווגות כעויינות מקובצות בחלק השמאלי של הגרף ובאיור השמאלי ניתן לראות שהמקבץ עבור אשכול 0 (ירוק) מתמקד באותו האזור. לפיכך אנו מניחים כי מודל האשכול יתכן וכן זיהה את מודעות המזויפות וסיווג אתם באשכול 0. כך גם עבור המשרות התמימות עבורם יש מקבץ באיור הימני למעלה ועל כן נשער כי יתכן והוא סיווג אותם כתמימות בגרף השמאלי באשכול 2 (כתום). עבור אשכול 1 יתכן כי המודל לא הצליח בדרכו למדל את אותן הנקודות בהתאם ויצר להם אשכול חדש שעשוי להכיל הן נתונים השייכים למשרות עויינות עם תכונות מסיביות המצביעות על משרות תמימות או להיפך.

השוואה בין מודלים Evaluation

על מנת להשוות בין המודלים בחרנו להשוות לפי שני מדדי השווה במודלי ה-ML, מדד ה -AUC ROC ומספר הטעיות מסוג ראשון והשני לפי מטריצת המבוכה.

להלן מטריצות המבוכה של שלושת המודלים שלנו: (נספח 3)

		SVM		ANN		DT	
		בפועל		בפועל		בפועל	
		תמים	עוין	תמים	עוין	תמים	עוין
חיזוי	תמים	2632	127	2636	123	2606	153
	עוין	166	611	46	731	56	721

להלן סיכום תוצאות ההשוואה שלנו

פרמטר/מודל	DT	ANN	SVM
טעות מסוג 1	5.87% (153טעויות)	4.66% (123 טעויות)	4.82% (127 טעויות)
טעות מסוג 2	7.76% (56 טעויות)	6.29% (46 טעויות)	27.16 (166 טעויות)
אחוז דיוק על המבחן	93.5%	94.81%	87%

לאחר שביצענו שיפורים להיפר-פרמטרים של כל המודלים, ניתן לראות באופן מובהק שגם על פי טעות מסוג 1 וגם על פי טעות מסוג 2 ה-ANN נותן את התוצאות הטובות ביותר לאחריו הDT ולבסוף ה-SVM. בחרנו במודל ANN כיוון שהוא בעל אחוז הדיוק הגבוהה ביותר על סט המבחן - 94.81% ואחוז דיוקו על סט הוולידציה הינו 94.14%.

שיפור המודל- הנבחר Improvement

שיפור על הנתונים:

לאחר תיקון ניכר בחלק א' של הפיצ'רים (קידוד, נורמליזציה ובחירת פיצ'רים) ולפי הנאמר על התוצאות במודל האשכול, אנו מערכים כי יש לבצע שיפור בנתוני המודל היות ומידול ה -Pre processing שבוצע לראות עיינו ועשוי להביא לפיצ'רים בעלי השפעה נמוכה על פעולת הסיווג ועל כן נתייחס לטיפול בהם. אחת השיטות לטיפול בנתונים והבאת לשיפור המודל היא הורדת ממד. לאחר קריאה באינטרנט על שיפור המודל עבור נושאים שלא נלמדו בקורס, עבור שיפור הנתונים בחרנו לעשות הורדת ממד בשיטת UMAP. ה- UMAP היא טכניקת הפחתת ממד לא לינארית שמטרתה לשמר את המבנים המקומיים והגלובליים של הארגון, ומאפשרת ללכוד קשרים מורכבים יותר, וזאת לעומת שיטת ה-PCA שבה השתמשנו במהלך הפרויקט. על אף שלפי אופי הנתונים שלנו נראה כי שיטת ה-PCA היא המתאימה יותר עבור סט הנתונים (הן מבחינת התמודדות עם מספר הרשומות הן מבחינת חשיבות המאפיינים העיקריים והן מבחינת המבנה הגלובלי) רצינו לראות ניסיון גם של שיטת ה-UMAP על הנתונים שלנו, אך הצפי ששיטה זו לא תביא בהכרח לשיפור בתוצאות המדד (זאת ללא להתחשב בשיפור הנוסף שעשינו על המודל) עקב הנאמר לעיל. מימשנו את הפונקציה הבאה לאחר מספר ניסיונות של משחק על ההיפר פרמטרים להביא לרמת השיפור המיטבית (או במקרה שלנו להפחתה המיטבית):

```
umap_obj = umap.UMAP(n_components=3, n_neighbors=15, min_dist=0.9)
```

ואכן קיבלנו תוצאת דיוק נמוכה יותר על המודל שלנו עבור הורדת הממד בשיטת ה-UMAP ללא שיפור המודל שעשינו לאר מכן וקיבלנו תוצאת AUC-ROC של 94.576% על סט המבחן שלנו

שיפור על המודל:

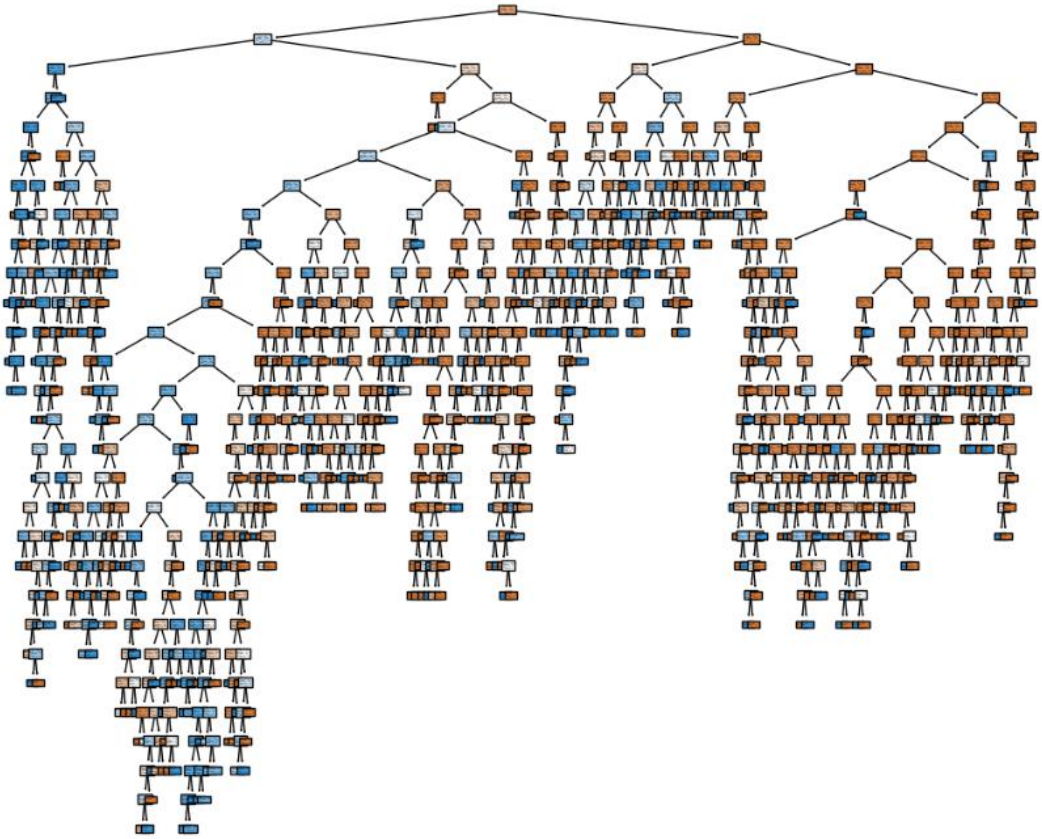
ברשת ANN אנו מגדירים learning rate קבוע שבקצב הזה המודל לומד. קצת גבוה מביא להתכנסות מהירה ודיוק נמוך וקצב נמוך שמביאים לדיוק גבוה והתכנסות איטית. הפתרון שמצאנו לבעיה הנ"ל הוא שיצרנו learning rate משתנה בזמן כתלות באצוות ובכך בהתחלה להביא את המודל "קרוב" להתכנסות באופן מהיר ועדיין שומר על רמת דיוק גבוהה. בחרנו בשיטת Learning rate scheduling ליישום מטרה זו. בשיטה הזו אנו שולטים בקצב הלמידה. במודל שלנו בחרנו להכניס decay factor=0.8 ו decay epochs=64 . הdecay factor שולט על אחוז ההורדה מהLearning rate, ואילו ה decay epochs שולט על כמות האצוות שעוברות בין כל factor לfactor. על כן ציפינו לשיפור בדיוק המודל ואכן, ללא הרצת שיפור הנתונים קיבלנו עליה ב 0.21% עם דיוק של95.25% .

לאחר שילוב של שני השיפורים קיבלנו אחוז דיוק של 95.02% על סט האימון שלנו. עקב כך נבחר לבחון את סט המבחן החיצוני רק על שיפור המודל על מנת לקבל תוצאה מיטבית.

נספחים

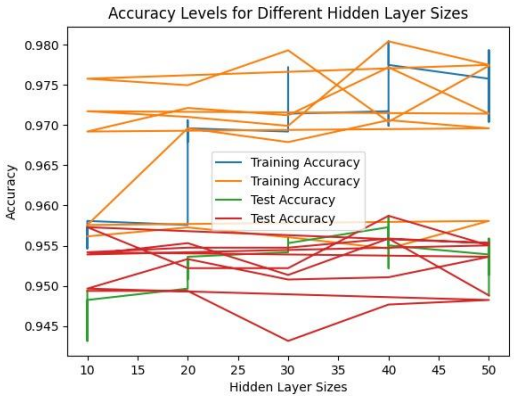
נספח 1

עץ ההחלטה מלא:

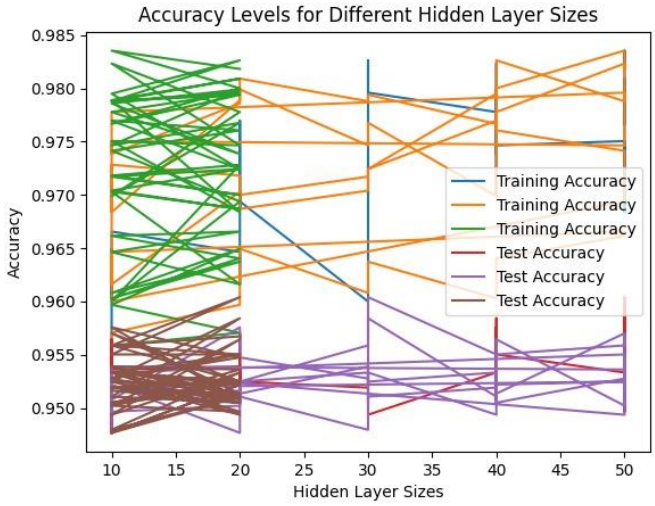


נספח 2

שכבה בודדת חבויה ברשת הנוירונים:



שלוש שכבות חבויות ברשת הנוירונים:



נספח 3

rank_test_score	std_test_score	mean_test_score	std_train_score	mean_train_score	params	param_tol	param_dual	param_C
1	0.005882308	0.96061267	0.00057607	0.961350039	{'C': 13, 'dual': False, 'tol': 0.01}	0.01	FALSE	13
2	0.005882421	0.960610715	0.000574878	0.961350537	{'C': 19, 'dual': False, 'tol': 0.01}	0.01	FALSE	19
3	0.00588086	0.960610432	0.000575422	0.961350923	{'C': 7, 'dual': False, 'tol': 0.01}	0.01	FALSE	7
4	0.005885438	0.960609041	0.000573961	0.961350197	{'C': 25, 'dual': False, 'tol': 0.01}	0.01	FALSE	25
5	0.005876578	0.960608756	0.000574993	0.961349518	{'C': 43, 'dual': False, 'tol': 0.01}	0.01	FALSE	43
6	0.00587674	0.960608197	0.000575102	0.961349011	{'C': 37, 'dual': False, 'tol': 0.01}	0.01	FALSE	37
6	0.005876361	0.960608197	0.00057488	0.96134935	{'C': 49, 'dual': False, 'tol': 0.01}	0.01	FALSE	49
8	0.005876891	0.960606799	0.00057523	0.961348999	{'C': 91, 'dual': False, 'tol': 0.01}	0.01	FALSE	91
9	0.005876192	0.960606799	0.000574901	0.961349221	{'C': 61, 'dual': False, 'tol': 0.01}	0.01	FALSE	61
10	0.00587631	0.96060652	0.000574908	0.961349119	{'C': 55, 'dual': False, 'tol': 0.01}	0.01	FALSE	55
10	0.005876481	0.96060652	0.000575543	0.96134865	{'C': 115, 'dual': False, 'tol': 0.01}	0.01	FALSE	115
10	0.00587603	0.96060652	0.000574897	0.961349123	{'C': 67, 'dual': False, 'tol': 0.01}	0.01	FALSE	67
13	0.005879415	0.960606244	0.000575463	0.961349159	{'C': 31, 'dual': False, 'tol': 0.01}	0.01	FALSE	31
14	0.005876248	0.96060624	0.000574889	0.96134913	{'C': 79, 'dual': False, 'tol': 0.01}	0.01	FALSE	79
14	0.005876248	0.96060624	0.000574779	0.961349295	{'C': 85, 'dual': False, 'tol': 0.01}	0.01	FALSE	85
14	0.005875621	0.96060624	0.000575539	0.961348759	{'C': 97, 'dual': False, 'tol': 0.01}	0.01	FALSE	97
14	0.005876528	0.96060624	0.000575786	0.961348807	{'C': 103, 'dual': False, 'tol': 0.01}	0.01	FALSE	103
18	0.005876367	0.960605961	0.000574773	0.961349087	{'C': 73, 'dual': False, 'tol': 0.01}	0.01	FALSE	73

נספח 4

ANN:

```
[2636  123]
[  46  731]
```

DT:

```
DT Confusion Matrix:
[[2606  153]
 [  56  721]]
```

SVM:

```
SVM Confusion Matrix:
[[2632  127]
 [ 166  611]]
```