פרויקט לימוד מכונה

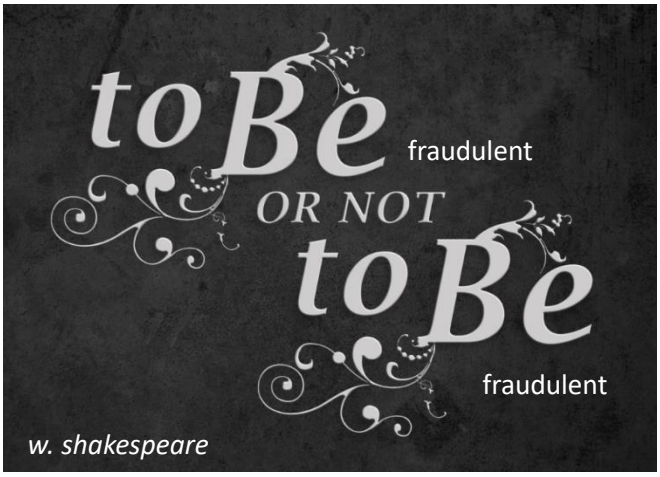
חלק ב'

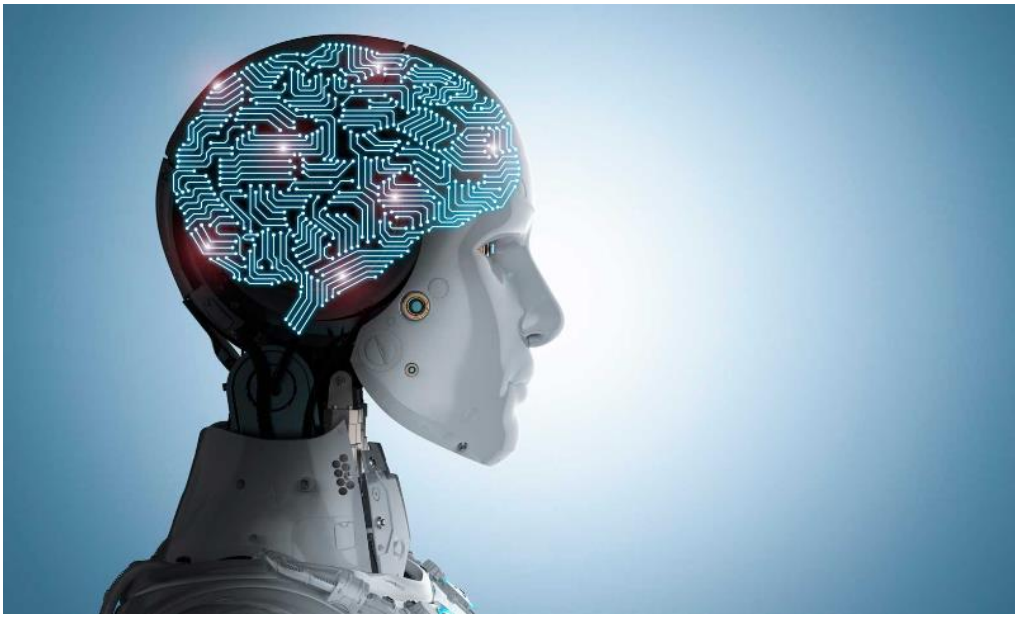
תאריך הגשה: 15/06/2023

מספר קבוצה: 30

מגישים: דור קורנט - 318871282

גיא ברון - 205984503





תוכן

Model training: 3
Decision Trees: 3
1. הפרמטרים שכיוונו בבניית העץ: 3
2. היתרונות הבולטים של עץ ההחלטה 3
3. העץ שהתקבל: 3
Artificial Neurons Network: 5
1.מודל דפולטיבי 5
2.Hyperparameter Tuning: 5
SVM: 7
Unsupervised Learning Clustering: 7
Evaluation - השוואה בין מודלים: 8
מסקנות עבור המודל: 9
הצעות לשיפור המודל – מורכבות המודל 9
הצעות לשיפור המודל – איזון הדאטה 9
נספחים: 11
נספח א- תוצאות אימון רשת נוירונים: 11
נספח ב תהליך אימון רשת נוירונים: 11
נספח ג – פרמטרי SVM שכוננו 13
נספח ד אחוזי דיוק עבור ממדים שונים בSVM: 13
נספח ה התפלגות נתונים בSVM 13
נספח ו – גרפים עבור בדיקות לקלסטרים 14
נספח ז טבלאות עבור חלופה שנייה : 15

Model training:

K fold - בחרנו לחלק את הנתונים בשיטה זו בשונה ממטריקת precision אותה כתבנו בחלק א' של העבודה. ביצענו 10 חלוקות שונות של הדאטה וחישבנו את התוצאות לפי מטריקת auc-roc מאחר והנתונים אינם מאוזנים. הערה: על מנת לאזן מעט יותר את הדאטה, מחקנו סמפלים בהם היו מעל 2 ערכים חסרים עבור המודעות המסווגות כמודעות אמת, לאחר ניקוי זה הבאנו את הדאטה ל-11% מודעות חשודות ול-4751 סמפלים סה"כ.

Decision Trees:

1. הפרמטרים שכיוונו בבניית העץ:

א. Criterion- פרמטר זה הוא הקריטריון לפיו העץ נבנה. האפשרויות אותן בחנו הן: gini ו- entropy, כאשר gini הוא מדד לתדירות שבה רכיב שנבחר באקראי מתוך קבוצה מסוימת יסומן באופן שגוי ו- entropy הוא מדד המעיד על אי הסדר של הפיצ'רים עם משתני המטרה. בנינו עץ עם כל אחד מהקריטריונים הללו וקיבלנו ב- entropy תוצאות טובות יותר מאשר ה-gini ולכן בחרנו בו.

ב. Max depth- בדקנו בלולאה את עומק העץ אשר חזה את סט הוולידציה בצורה הטובה ביותר. בדקנו ערכים החל מ1 ועד לעומק 20 והעומק עבורו קיבלנו את התוצאות הטובות ביותר הינו 4.

ג. n-splits- פרמטר זה מייצג את מספר הקיפולים שלתוכם מחלקים את סט הנתונים. לאחר שניסינו מספרים שונים קיבלנו את התוצאות הטובות ביותר עם 5 קיפולים.

- **אחוז הדיוק המקסימלי שקיבלנו על סט האימון הינו: 94.3%**
- **אחוז הדיוק המקסימלי שקיבלנו על סט הוולידציה הינו: 92.9%**

ניתן להסיק מתוצאות אלו שהעץ יודע לסווג נכונה את מרבית הדאטה, אך לא את כולה. כיוונון הפרמטרים לא הניבו תוצאות שונות בצורה משמעותית אחת מהשנייה, מה שיכול להעיד על כך שייתכן וישנם קשרים ודפוסים מסובכים בתוך הדאטה שהעץ לא מצליח להתמודד איתם.

2. היתרונות הבולטים של עץ ההחלטה הוא יכולת הפרשנות שלו. יתרון זה בא לידי ביטוי בכך שהעץ מציג תחזיות המבוססות על סדרה של תנאים מסוג if-else ומראה את הסיווג לייבלים השונים בצורה די ברורה לפי תנאים אלו. יכולת זו של העץ תורמת למשימת הלמידה בכך שהיא מספקת תובנות לגבי חשיבות הפיצ'רים במשימת החיזוי. בנוסף, ניתן ללמוד מהתפצלויות העץ על קשרים שונים בין פיצ'רים ואיך שילוב של ערכים מסוימים בפיצ'רים אלו באים לידי ביטוי בסיווג לייבל מסוים.

3. העץ שהתקבל:

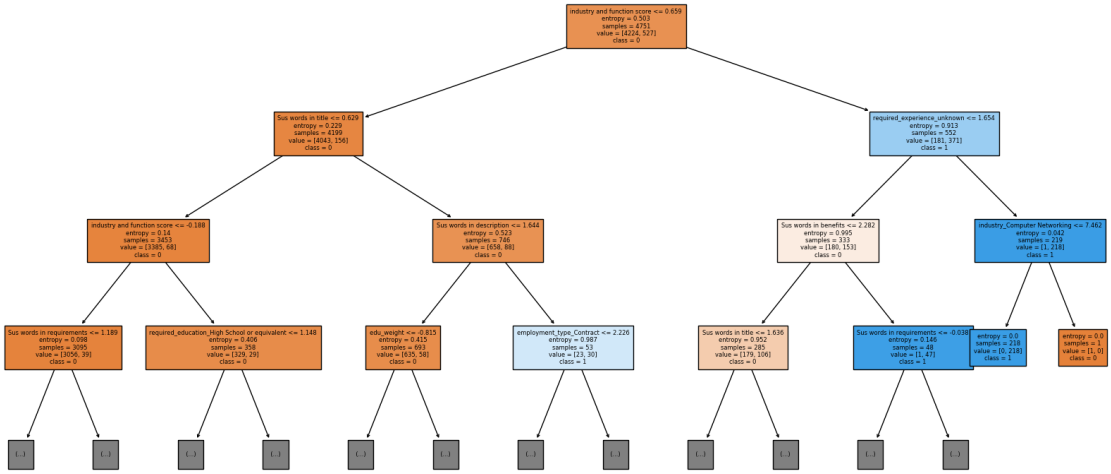
א. עצי החלטה מתפצלים על סמך תכונות שהן אינפורמטיביות ביותר למשימת הקלסיפיקציה. לכן, ככל שהתכונה מופיעה למעלה יותר בעץ היא חשובה יותר לתהליך קבלת ההחלטות של העץ. ניתן לראות כי industry and function score הוא הפיצ'ר החשוב ביותר למשימה זו, כלומר לה יש את ההשפעה הגבוהה ביותר על תהליך הקלסיפיקציה.

ב. ככל שפיצ'ר יותר נמוך בעץ וקרוב יותר לרמה הנמוכה ביותר בעץ הוא מבחין בהבדלים קצת יותר עדינים בדאטה. כך ניתן לראות כי הפיצ'ר industry computer networking מצליח לקבל מסקנות על 219 סמפלים ולסווג אותם ללייבלים, לאחר שהפיצ'רים industry and function score ו- required experience unknown סיווגו אותם בהתאמה.

ג. פיצ'רים הנמצאים על אותו ענף יכולים להעיד על אינטראקציה גבוהה בניהם. ניתן לראות כי לדוגמא הפיצ'ר sus words in benefits מפצל את הדאטה ולאחר מכן sus words in title מפצל אותו שוב, מה שיכול להצביע על כך שישנו קשר בין השניים.

ד. העץ מציג בצורה ברורה את הכללים לפיהם הוא מסווג סמפל לקלאס מסוים. כך לדוגמא, רשימת התנאים הבאים תגרום לעץ לסווג את המודעה כמודעת שקר:

- Industry and function score > 0.659
- Required experience unknown > 1.654
- Industry computer networking <= 7.462




```
industry and function score: 0.6233750335518053
required_experience_unknown: 0.10550414830819416
Sus words in title: 0.08307600298029626
Sus words in benefits: 0.0492373022189404 04
Sus words in description: 0.0326717107169154 64
Sus words in requirements: 0.0242059984126525 57
n_words_in_company_profile: 0.01676387288198224
edu_weight: 0.016465261792917546
required_experience_Entry level: 0.011119448066210465
required_education_High School or equivalent: 0.008668339227668872
Sus words in department: 0.0068723116095459605
edu_exp_mismatch: 0.006369576160814884
industry_Computer Networking: 0.005956234296043836
industry_Telecommunications: 0.004329341688316744
employment_type_Contract: 0.0040925776625361345
city_unknown: 0.0012928404251590854
```

השתמשנו בפונקציית חשיבות הפיצ'רים של המודל. להלן הציונים של חשיבות הפיצ'רים, כאשר ככל שהציון גבוה יותר כך הפיצ'ר יותר חשוב בקבלת ההחלטות בעץ. הציונים מסודרים מהציון הגבוה לנמוך:

- הפלט מראה כי הפיצ'ר החשוב ביותר הוא industry and function score. פיצ'ר זה הוא הפיצ'ר הראשון המופיע בעץ ולכן הוא הכי חשוב למשימת הלמידה, והעובדה שהוא קיבל את ציון החשיבות הגבוה ביותר לא מפתיעה אותנו מאחר וגם לפי העץ ניתן להבין זאת.
- ניתן לראות שהפיצ'רים הבאים בעלי הניקוד הגבוה גם הם נמצאים ברמות הראשונות של העץ, מה שמלמד אותנו על כך שבתהליך קבלת ההחלטות של העץ הוא משתמש בפיצ'רים בעלי רמת החשיבות הגבוהה ביותר ואלה שעושים את ההבדלה הגבוהה ביותר בין הלייבלים. כך לדוגמא

required experience unknown הוא מקום השני בחשיבותו והוא אכן נמצא ברמה השנייה בעץ, אך ניתן לראות כי רמת האנטרופיה שאיתה הוא מתמודד (entropy בעץ) גבוהה יותר מהפיצ'ר השני שנמצא באותה רמה בעץ (sus words in title) ולכן הגיוני כי הוא קיבל ציון גבוה ממנו.

- הפיצ'רים בעלי הציונים הנמוכים ביותר בפונקציית החשיבות מתקשים יותר לסווג את הדאטה וצריכים סיווג קצת יותר מדויק בשביל לקחת חלק במשימת הקלסיפיקציה. כך לדוגמא employment type contract קיבל ציון יחסית נמוך בפונקציית החשיבות, כאשר ניתן לראות בעץ כי הוא נמצא ברמה יחסית נמוכה בעץ והוא מצליח לחלק את הדאטה רק לאחר שרוב הדאטה כבר מחולק, ולסווג לרמה הבאה 53 סמפלים בלבד.

Artificial Neurons Network:

1.מודל דפולטיבי

<u>מספר נוירונים בשכבת הכניסה</u> – מספר הנוירונים בשכבת הקלט שווה למספר הפיצ'רים שהמודל יכול ללמוד מהם, כל פיצ'ר מהווה נוירון ברשת הכניסה. לאחר ביצוע קטגוריזציה למודל, שכבת הקלט שלנו כללה 924 פיצ'רים שונים.

<u>מספר שכבות חבויות</u> – שכבה נסתרת אחראית ללמידת דפוסים מורכבים מתוך נתונים הקלט, מספר השכבות הנסתרות קובע את עומק ואת יכולתה לייצג מערכות יחסים מורכבות.

<u>מספר נוירונים חבויים בכל שכבה</u> – מספר הנוירונים החבויים בכל שכבה נסתרת מגדיר את יכולת הרשת ללמוד ולייצג מידע, כאשר קיום של יותר נוירונים בשכבה נסתרת מגדיל את יכולת הרשת ללמוד דפוסים מורכבים בנתונים. יחד עם זאת, שימוש במספר רב של נוירונים עלול לגרום להתאמת יתר עבור המודל הנלמד.

לאחר הרצת מודל רשת הנוירונים בערכי ברירת המחדל אלו הם האחוזים שקיבלנו על סט האימון וסט הוולידציה:

```
Accuracy for train: 1.000
Accuracy for test: 0.918
```

<u>ערכי ברירת המחדל:</u> 2 שכבות של 100 נוירונים, פונקציית אקטיבציה: 'relu', קצב לימוד: 0.001 וכמות איטרציות מקסימלית: 500. ניתן להסיק מתוצאות אלה כי רשת הנוירונים למדה בצורה מושלמת את המודל של סט האימון ועל כן קיבלה 100%. בנוסף, דיוק של 91.8% על סט הוולידציה מעידים על התחלה טובה למודל, שכן התוצאות הושגו לפני כיוונון הפרמטרים.

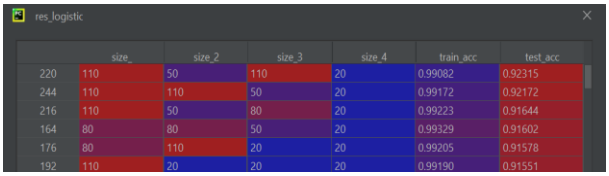
2.Hyperparameter Tuning:

מספר השכבות החבויות: הגדלת מספר השכבות החבויות במודל עשויה לגרום למודל ללמוד דפוסים מורכבים יותר, אך יותר מידי שכבות עלולות לגרום להתאמת יתר. מאחר וערך ברירת המחדל של המודל (העומד על 2 שכבות) נתן תוצאה טובה, ביצענו בדיקה עבור 1 עד 4 שכבות נסתרות של רשת הנוירונים, כאשר המטרה הייתה לבדוק מהו המודל שילמד בצורה הטובה ביותר את התנהגות הנתונים ולבדוק התנהגויות שונות בהתאם למספר השכבות.

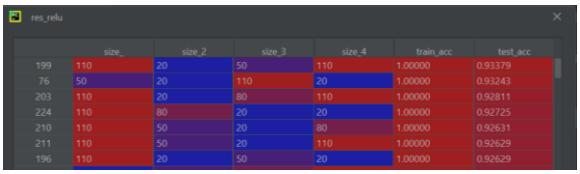
מספר הנוירונים בכל שכבה חבויה: הגדלת מספר הנוירונים בכל שכבה מספקת לרשת יכולת גדולה יותר ללמוד ולייצג יחסים מורכבים בתוך הנתונים. עם זאת, מספר נוירונים גבוה מידי יכול לגרום להתאמת יתר ולכן ניסינו לבחון מספר קומבינציות שונות של מספר נוירונים ומספר שכבות חבויות, כאשר נבדקו גם מספר לא שווה של נוירונים בכל שכבה. ביצענו כיוונון החל מ1 ועד ל120. בחרנו בטווח ערכים זה כי רצינו לראות איך ישפיע כמות גבוהה של קומבינציות של מספר נוירונים בכל שכבה ומספר שכבות.

Max_Iter: פרמטר המעיד כמה פעמים תבצע הרשת עדכון למשקולות המודל (הלוך וחזור על כל הרשת) במטרה להביא ל min את "טעות המודל". הפרמטר נועד למנוע הרצה אין סופית של הרשת במידה והוא אינו מגיע להתכנסות כנדרש בפרמטרים אחרים. בחנו את מספר האיטרציות הנדרש עד להתכנסות המודל, הפרמטרים שנבדקו היו {100,300,500} כי רצינו לראות כמה באמת השינוי של מספר האיטרציות ישפיע על ביצועי המודל.

Activation: פונקציות activation ממלאות תפקיד מכריע בהחדרת טרנספורמציות לא ליניאריות לרשת. ללא אי-לינאריות, הרשת תהיה מסוגלת לייצג רק יחסים ליניאריים, מה שמגביל מאוד את כוח הביטוי שלה. פונקציות הפעלה מציגות אי-לינאריות על ידי יישום פונקציה מתמטית על הסכום המשוקלל של התשומות המועברות לנוירון. בדקנו אילו ערכים פונקציה זו יכולה לקבל וניסינו לבדוק איזו פונקציה תיתן לנו את ערכי הדיוק הגבוהים ביותר, כאשר ניסינו את פונקציית relu, שזו פונקציית ברירת המחדל של המודל ואת פונקציית logistic.



	size_	size_2	size_3	size_4	train_acc	test_acc
220	110	50	110	20	0.99082	0.92315
244	110	110	50	20	0.99172	0.92172
216	110	50	80	20	0.99223	0.91644
164	80	80	50	20	0.99329	0.91602
176	80	110	20	20	0.99205	0.91578
192	110	20	20	20	0.99190	0.91551



	size_	size_2	size_3	size_4	train_acc	test_acc
199	110	20	50	110	1.00000	0.93379
76	50	20	110	20	1.00000	0.93243
203	110	20	80	110	1.00000	0.92811
224	110	80	20	20	1.00000	0.92725
210	110	50	20	80	1.00000	0.92631
211	110	50	20	110	1.00000	0.92629
196	110	20	50	20	1.00000	0.92629

טבלת ערכים מלאה – נספח א, תהליך כיוונון הפרמטרים – נספח ב.

ניתן לראות כי התוצאה האופטימלית התקבלה עבור 4 שכבות חבויות של נוירונים, כאשר כמות הנוירונים הינה: השכבה הראשונה מכילה 110, השנייה 20, השלישית מכילה 50 והרביעית מכילה 110. פונקציית ה- activation שיישמנו הינה פונקציית relu.

אחוזי הדיוק שקיבלנו הינם 100% על סט האימון ו- 93.379% על סט הוולידציה. לאחר כיוונון פרמטרים הצלחנו לשפר את אחוזי האימון של סט הוולידציה וכעת המודל יכול לעשות קלסיפיקציה בצורה מדויקת יותר. ניתן להבחין כי התוצאות של מודל רשת הנוירונים טובות יותר מהתוצאות של עץ ההחלטות, ועל כן נסיק כי רשת נוירונים מתאימה יותר למשימת הלמידה שלנו.

רשת הנוירונים הינה מודל מורכב יותר וביכולתו לתפוס דפוסים ויחסים מורכבים יותר בדאטה על ידי כיוונון מספר השכבות החבויות ומספר הנוירונים בכל שכבה (כפי שהזכרנו קודם לכן). ייתכן ובסט הנתונים שלנו ישנם יחסים מורכבים, מה שגורם לרשת הנוירונים ללמוד יחסים אלו טוב יותר מהעץ. בהתאם לכך, כיוונון פונקציית האקטיבציה ב- ANN גורמת לרשת הנוירונים להתמודד עם קשרים שאינם ליניאריים, בניגוד לעץ, שמוגבלת לקשרים ליניאריים או לקשרים פשוטים שאינם ליניאריים, מה

שגורם לרשת הנוירונים להבין קשרים מורכבים יותר ולתת תוצאות טובות יותר.

```
[[603  39]
 [ 16  55]]
```

הרצנו את המודל עם הערכים שהניבו לנו את התוצאות האופטימליות על סט הוולידציה, המכיל 713 סמפלים. זו מטריצת הבלבול שהתקבלה:

ניתן לראות כי המודל זיהה יותר דברים כנכונים (תמימים) אשר אינם כאלה יותר מאשר הכריז על חשודים עבור סמפלים תמימים. דבר זה יכול להיגרם בעקבות חוסר איזון בנתונים מה שגרם לאלגוריתם נטייה להציג סמפלים כתמימים.

SVM:

בחרנו ראשית להציג את תוצאות האימון על כל סט הפיצ'רים שלנו היות והמודל שלנו מורכב ומכיל פיצ'רים רבים, לאחר מכן בחנו את המודל גם עבור הורדת מימד כדי להציג אותו בצורה ויזואלית.

התוצאה הטובה ביותר שקיבלנו: הינה בעזרת C של 0.05 עבור כל הפיצ'רים שלנו (924) לאחר נרמול וקידוד הפיצ'רים הקטגוריאליים וביצוע קרוס-ולידציה ל10 מקבצים שונים (נספח ג):

פרמטר	C	סט האימון	סט הוולידציה
תוצאה	0.05	0.9878	0.9304

היות ולא ניתן להציג ויזואליזציה לסט של924 פיצ'רים וגם לא להציג משוואת ישר באופן נוח בחרנו לבצע הורדת ממד בעזרת פונקציית PCA ולבדוק את תוצאות המודל (נספח ד). בחרנו להוריד ל-2 ממדים כדי להצליח להציג את הנתונים בצורה נוחה (נספח ה) למרות שאחוזי הדיוק שלו פחתו מהמודל המקורי.

משוואת קו ההפרדה: y=-1.8335 +0.926x1+0.7896x2

התוצאה על סט האימון: 0.926, תוצאה על סט הבדיקה: 0.7893

ניתן לראות כי המשתנה הראשון מסביר את הנתונים בצורה טובה יותר מהמשתנה השני, אך אחוז השונות המוסברת במקרה זה עדיין נמוך (רק 0.0375 בסה"כ, נספח ה) כאשר הערכים זהים.

המשמעות היא שרוב השונות במודל במקרה הזה היא בלתי מוסברת. תוצאה זו הגיונית היות ועבור מודלים בהם בחנו את כל הפיצ'רים כמו ברשת נוירונים קיבלנו תוצאות טובות יותר על סט הבחינה מאשר בעץ ההחלטות.

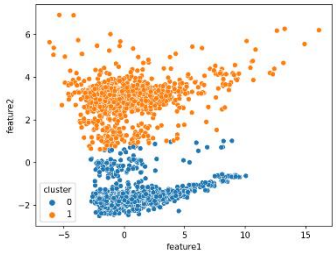
Unsupervised Learning Clustering:

במשימות הלימוד המתויגות שעשינו חילקנו את סט הנתונים לסט אימון וסט וולידציה. בעזרת סט האימון המודלים למדו את הקשרים בין הפיצ'רים השונים ואת הקשר שלהם לקלאס אליו הם מסווגים, ולאחר מכן ניסינו ליישם את מסקנות אלו על סט הוולידציה ולנסות לסווג אותו לקלאס שלו. במשימת לימוד שאינה מתויגת אנו נצטרך את כל סט הנתונים, ללא חלוקה לסט אימון וסט וולידציה. הסיבה היא שבמשימה שאינה מתויגת משימת הלמידה הינה זיהוי אשכולות דומים, זיהוי דפוסים וניסיון להסיק מסקנות על התנהגויות ותכונות מסוימות שייגרמו לסמפלים להשתייך לקלאס מסוים.

בחרנו להריץ את אלגוריתם K-medoids בעזרת אלגוריתם PCA אשר יוריד לנו את כמות הפיצ'רים

ל2, הסיבה לכך היא שלאחר תהליך ה- Feature Selection, הנורמליזציה וקידוד הפיצ'רים הקטגוריאליים נשארנו עם 924 פיצ'רים שרובם קטגוריאליים ולכן תרומתו של כל פיצ'ר למודל היא קטנה אך המורכבות היא שהופכת את המודל לטוב.

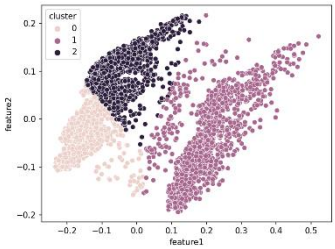
<u>ראשית הצגנו את הדאטה בעזרת הורדת ממד:</u>



על בסיס הנראה בצורה ויזואלית ניתן לחלק את הדאטה ל-3 או 4 קלסטרים שונים. אנחנו בחרנו לחלק לשלושה קלסטרים היות והחלק הכתום בתמונה לדעתנו אינו צפוף ומופרד מספיק כמו 2 החלקים הכחולים.

<u>הערה:</u> היות ואנחנו יודעים כי המודל הינו בעל 2 קלסטרים בחרנו להציג ב-2 צבעים את הפלט למעלה, בפועל לצבעים אין שום חשיבות והם נועדו לעזור בוויזואליזציה בלבד.

<u>מדד המרחק בו נבחר להשתמש הוא מדד Gower</u> כיוון שהוא יודע להתמודד גם עם פיצ'רים קטגוריאליים מנורמלים וגם עם פיצ'רים אורדינליים (בעלי חשיבות לגודל) שגם הם מנורמלים. לאחר חישוב מטריצת המרחק בין הפיצ'רים הורדנו את הממד ל2 בעזרת פונקציית PCA ובחנו את התנהגות הנתונים על 3 קלסטרים כפי שראינו בהצגה הקודמת.



<u>הקריטריונים שבחרנו להשוואת ערכי ה-K</u> הינם cluster validation index. מדדים אלו נועדו למדוד את איכות הקלסטרים ועבור כל מדד הם מתקבלים כתוצאה מערך שונה בהתאם למוסבר מטה. מדד intertia ה"מרפק" מתקבל בערך 3, במדד silhouette הירידה החדה ביותר התקבלה בערך 3, במדד Davies-bouldin העלייה החדה ביותר התקבלה בערך 6 (הגרפים מוצגים ב- <u>נספח ו</u>). **אנו נצפה לראות 3 מחלקות** בעקבות השימוש במדדים intertia, silhouette ו-Davies-bouldin. לא נוכל לייחס משמעות לכל קלסטר מאחר ובחרנו פיצ'רים בעזרת PCA ולכן הקומפוננטות שנבחרו אינן ברות הסברה.

Evaluation - השוואה בין מודלים:

שיטת בדיקה/מודל	SVM	MLP	DT
תוצאות סט אימון	0.9878	1	0.9413
תוצאות סט בדיקה	0.9304	0.93379	0.929

עבור משימת למידה זו נמליץ על מודל MLP ממספר סיבות: ראשית, היות וסט הפיצ'רים איתם בחרנו לעבוד הכיל כמות גדולה של פיצ'רים ביחס לכמות הסמפלים (924 פיצ'רים ו-4751 סמפלים), המודל מורכב באופן יחסי לבעיה ולכן דרוש אלגוריתם למידה שמסוגל להתמודד עם מודלים מורכבים. מודל DT מטבעו מסוגל להתמודד עם כמות סופית (קטנה לרוב) של פיצ'רים על מנת להכריע על חיזוי ובנוסף נקטם בשלב מוגדר על מנת למנועה התאמת יתר. אכן קיבלנו כי עומק העץ האופטימלי הינו 4 ומכאן שעבור כל סמפל מס' הפיצ'רים המקסימליים עבורו הינו 4. מודל ה-SVM גם הוא עובד בדרך כלל עם כמות נמוכה של פיצ'רים ולכן ניסינו לאמן אותו לאחר הורדת ממד עם PCA. למרות זאת, אחוז ההצלחה הגבוה ביותר במודל זה היה כאשר בחנו את המודל על כל

הפיצ'רים ומכאן שישנה חשיבות רבה לשימוש ברובם במודל. למרות שיכולת ההסברה על MLP מורכבת ואינה פשוטה כמו במודלים SVM ו- DT מתוצאות המודל עולה כי ישנם קשרים בין משתני המודל שלא ניתנים להסברה ובהכרח השילוב ביניהם הוא שיוצר את יכולת החיזוי. **לכן נחליט להשתמש באלגוריתם MLP אשר עליו קיבלנו אחוזי דיוק ולידציה של 0.93379**

מסקנות עבור המודל:

1. מורכבות המודל- קיבלנו מודל המורכב מ924 פיצ'רים שונים, מסיבה זו רשת הנוירונים עבדה טוב יותר משיטות אחרות אך עדיין אנחנו חושבים שאינה אופטימלית. מספר הנוירונים החווים בכל שכבה ומספר שכבות חבויות נבחרו בעזרת פונקציית כיוונון Hyper Parameters ללא חשיבה כיצד מורכבות המודל ביחס למספר הסמפלים ישפיע עליהם. על מנת לפתור את הבעיה נערוך חיפוש באינטרנט אודות הגורמים המשפיעים על בחירת טווח הערכים למספר הנוירונים הרצויים בשכבות החבויות ומספר השכבות החבויות. נצפה להבין כיצד ניתן לאמן את המודל בצורה טובה יותר.
2. חוסר איזון בדאטה- לאחר עבודת איזון הדאטה, הגענו למצב של 11% מודעות שקר מול 89% מודעות אמת, שזה מעט טוב יותר מהמצב ההתחלתי (95% אמת למול 5% שקר) אך עדיין מאוד לא מאוזן לבעיית לימוד מכונה. איזון זה הניב אחוזי דיוק גבוהים יותר מאחוז הדיוק ההתחלתי, והבנו כי צריך ייצוג מספיק לשני הלייבלים כדי שהמודל יוכל ללמוד בצורה המיטבית ביותר. בפתרון לבעיה זו נרצה להביא את הדאטה לאיזון גבוה יותר ולבחון האם ישפר את המודל.

הצעות לשיפור המודל – מורכבות המודל: על מנת לשפר את התמודדות המודל עם כמות הפיצ'רים בדקנו מה משפיע על תהליך הלימוד של המודל. עבור מודלים מורכבים מספר הנוירונים הוא זה שהכי רלוונטי לכוונון ולאו דווקא מס' הרשתות, גילינו כי ישנם מס' כללי אצבע נהוגים בספרות על מנת לקבוע את כמות הנוירונים:

א. כמות הנוירונים תהייה בין הכמות בשכבת הקלט (943) לשכבת הנוירונים בשכבת הפלט (2)

ב. כמות הנוירונים המוסתרים צריך להיות קטן מפעמיים שכבת הקלט+ שכבת הפלט

ג. מספר הנוירונים בשכבות הנסתרות יורדות ככל שמתקרבים לOUTPUT על מנת להתקרב לחילוץ הדפוסים ולזהות את המחלקות בשכבת הפלט.

בחרנו לבחון את המודל עם 2 שכבות המכילות 1500 פרמטרים בשכבה הראשונה ו1300 פרמטרים בשכבה השנייה. כפי שניתן לראות המודל הצליח ללמוד את דפוסי הדאטה בצורה טובה יותר מאשר המודל הראשוני שניסינו ובנוסף הציג תוצאות חיזוי טובות יותר על סט הוולידציה. להבנתנו, כאשר ישנם מספר רב של שכבות המודל לעיתים לומד התנהגות של סמפלים ספציפיים כלומר יוצר מצב של over fitting לנתונים.

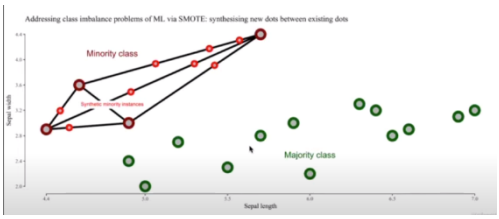
```
Accuracy for train: 1.000
Accuracy for test: 0.944
[[624  18]
 [ 15  56]]
```

הצעות לשיפור המודל – איזון הדאטה

הדאטה איתו עבדנו מתחלק ל-89% ו- 11%, ורצינו לבדוק האם העלאת מספר הסמפלים של ה- minority class (מודעות שקריות) ואיזון גדול יותר ייתן תוצאות טובות יותר של המודל הנבחר.

לשם כך נעשה over-sampling, העלאת מספר הסמפלים של ה- minority class בשביל ליצור איזון

בדאטה. אחת הטכניקות של over-sampling שבה נשתמש היא אלגוריתם SMOTE (Synthetic Minority Oversampling Technique), אשר יוצר סמפלים סינטטיים של ה- minority class בשביל לאזן את הדאטה. SMOTE עובד בצורה כזו: עבור כל שני סמפלים מה- minority class ומגדיר קשרים בניהם, ועל הקשרים האלו יוצר סמפלים חדשים בצורה רנדומלית.



בגלל שבהצעת השיפור הראשונה הצלחנו לשפר את המודל, נשתמש במודל הכי טוב שלנו נשתמש במודל זה בשביל לבדוק אם איזון הדאטה באמת מניב תוצאות גבוהות יותר. חשוב לציין כי האלגוריתם SMOTE הוכל רק על סט האימון כדי לבחון את המודל בצורה דומה לכל המקרים וכי מדדי ההצלחה לא שונו (שימוש במטריקת auc roc score).

תחילה, בדקנו איזון של 50-50. שנית, בדקנו איזון של 25-75:

```
proportion before over-sampling:  {0.0: 3582, 1.0: 456}
proportion after over-sampling:  {0.0: 3582, 1.0: 1791}
Accuracy for train: 1.000
Accuracy for test: 0.881
[[610  32]
 [ 18  53]]
```

```
proportion before over-smapling:  {0.0: 3582, 1.0: 456}
proportion after over-smapling:  {0.0: 3582, 1.0: 3582}
Accuracy for train: 1.000
Accuracy for test: 0.896
[[631  11]
 [ 19  52]]
```

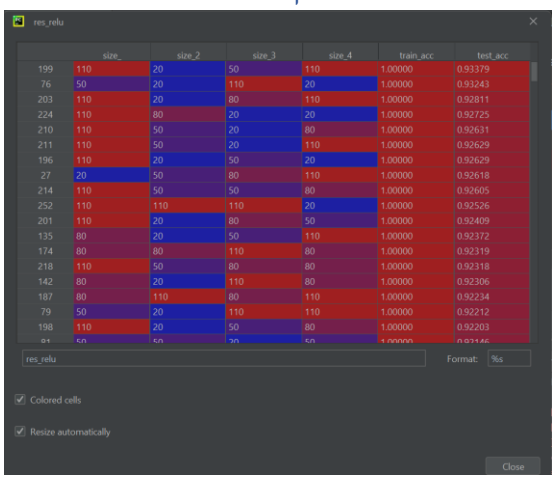
רעיון זה הביא לנו דיוק נמוך יותר מאשר המודל המקורי (94.4%). ייתכן כי ייצור סמפלים סינטטיים לא מייצג את העולם האמיתי מה שהכניס יותר רעש לסט האימון ובמבחן התוצאה המודל הצליח בצורה פחות טובה. סיבה נוספת שאנו מעריכים הינה שצם העובדה שמספר הסמפלים גדל משמעותית כעת, המודל המתאים לבעיה זו יהיה פחות מורכב מהמודל המתאים לבעיה המקורית. לאחר בדיקה במקורות חיצוניים, ראינו שסט אימון גדול יותר יכול להביא מודלים מורכבים למצב של over fitting, מאחר ומודלים מורכבים לומדים דפוסים מורכבים ולא לינאריים ייתכן והמודל שלנו למד את סט האימון יתר על המידה ולא הצליח להכיל את מסקנותיו על סט הוולידציה. בעקבות כך, ייתכן כי ברשת נוירונים עבור מספר גבוה יותר של סמפלים ואיזון גבוה יותר בין הקלאסים דרוש מודל פחות מורכב, המכיל פחות שכבות ופחות נוירונים כדי שהמודל יוכל לבצע הכללה בצורה גבוהה יותר. בשביל לחזק את טענה זו, חילקנו את סט האימון שוב ל-50-50 ול- 25-75 והרצנו מודל פשוט יותר של רשת נוירונים עם מספר שכבות משתנות (1-4) ומספר נוירונים משתנים (1-100), (נספח ז).

התוצאות שקיבלנו שיפרו את הדיוק היה 93.3% על סט הוולידציה, ומחזקות את טענתינו. קיבלנו כי מודל של רשת נוירונים בעל שכבה אחת עם 38 נוירונים קיבל אחוזי דיוק של 93% עבור איזון של 50-50 ו-93.3% עבור איזון של 25-75, כאשר המודל הוא שכבה אחת של 75 נוירונים. אחוזים אלו גבוהים יותר מהמודל המורכב שבדקנו לפני שיפור המודל אך עם זאת, תוצאות אלו פחות טובות מהמודל המקורי לפני ה- oversampling שקיבל אחוזי דיוק של 94.4%. מתהליך זה למדנו שאיזון הסמפלים על ידי oversampling ושימוש באלגוריתם SMOTE פחות מתאים למודל שלנו ולא מיטיב עם תוצאות המודל.

נספחים:

נספח א- תוצאות אימון רשת נוירונים:

	size_	size_2	size_3	size_4	train_acc	test_acc
220	110	50	110	20	0.99082	0.92315
244	110	110	50	20	0.99172	0.92172
216	110	50	80	20	0.99223	0.91644
164	80	80	50	20	0.99329	0.91602
176	80	110	20	20	0.99205	0.91578
192	110	20	20	20	0.99190	0.91551
212	110	50	50	20	0.99051	0.91398
204	110	20	110	20	0.98895	0.91369
228	110	80	50	20	0.99125	0.91350
188	80	110	110	20	0.99281	0.91306
252	110	110	110	20	0.99285	0.91143
232	110	80	80	20	0.99079	0.91095
180	80	110	50	20	0.99306	0.91069
156	80	50	110	20	0.99205	0.91056
172	80	80	110	20	0.99260	0.91016
152	80	50	80	20	0.99271	0.90928
96	50	80	20	20	0.99479	0.90814
72	50	20	80	20	0.99358	0.90810



	size_	size_2	size_3	size_4	train_acc	test_acc
199	110	20	50	110	1.00000	0.93379
76	50	20	110	20	1.00000	0.93243
203	110	20	80	110	1.00000	0.92811
224	110	80	20	20	1.00000	0.92725
210	110	50	20	80	1.00000	0.92631
211	110	50	20	110	1.00000	0.92629
196	110	20	50	20	1.00000	0.92629
27	20	50	80	110	1.00000	0.92618
214	110	50	50	80	1.00000	0.92605
252	110	110	110	20	1.00000	0.92526
201	110	20	80	50	1.00000	0.92409
135	80	20	50	110	1.00000	0.92372
174	80	80	110	80	1.00000	0.92319
218	110	50	80	80	1.00000	0.92318
142	80	20	110	80	1.00000	0.92306
187	80	110	80	110	1.00000	0.92234
79	50	20	110	110	1.00000	0.92212
198	110	20	50	80	1.00000	0.92203

נספח ב תהליך אימון רשת נוירונים:

בעקבות זליגת מידע בין סמפלים בסט האימון וסט הוולידציה שהתגלתה מס' ימים לפני מועד ההגשה נאלצנו לאמן את המודל מחדש ולהציג את התוצאות המעודכנות. כמובן שהתוצאות המעודכנות היו פחות טובות מאשר לפני הזליגה אך אמינות יותר.

את תהליך כוונון הפרמטרים נציג במודל הישן על מנת להדגים את התהליך (במודל החדש התהליך היה קצר יותר מפעת קוצר הזמן):

<u>הניסיונות שביצענו:</u>

שכבות חבויות: 2

נוירונים בכל שכבה: בין 20 ל-80

פונקציית אקטיבציה: relu

מקסימום איטרציות: 100

size_	size_2	train_acc	test_acc
81	21	1.00000	0.98494
41	21	1.00000	0.98303
41	41	1.00000	0.98244
81	41	1.00000	0.98122
61	61	1.00000	0.98117
61	21	1.00000	0.98102
21	81	1.00000	0.98051
41	61	1.00000	0.98041
81	81	1.00000	0.97947
41	81	1.00000	0.97861
21	41	1.00000	0.97844
81	61	1.00000	0.97840
21	61	1.00000	0.97553
61	41	1.00000	0.97459

תוצאה אופטימלית: 81 נוירונים בשכבה הראשונה, 21 נוירונים בשכבה השנייה, כאשר התוצאה הינה 100% על סט האימון ו- 98.4% על סט הוולידציה.

שכבות חבויות: 2

נוירונים בכל שכבה: בין 1 ל- 140

פונקציית אקטיבציה: relu

size_	size_2	max_ites_size	train_acc	test_acc
100	40	100	1.00000	0.98451
100	40	500	1.00000	0.98451
100	40	300	1.00000	0.98451
40	40	300	1.00000	0.98265
40	40	100	1.00000	0.98265
40	40	500	1.00000	0.98265
100	70	100	1.00000	0.98224
100	70	300	1.00000	0.98224
100	70	500	1.00000	0.98224
40	100	100	1.00000	0.97994
40	100	300	1.00000	0.97994
40	100	500	1.00000	0.97994
70	40	100	1.00000	0.97987
70	40	300	1.00000	0.97987

מקסימום איטרציות: 100,300,500

תוצאה אופטימלית: 100 נוירונים בשכבה הראשונה, 40 נוירונים בשכבה השנייה ומקסימום של 100 איטרציות. התוצאות הינן 100% על האימון ו- 98.4% על הוולידיציה.

שכבות חבויות: 3

נוירונים בכל שכבה: בין 1 ל- 140

פונקציית אקטיבציה: בטבלה העליונה – relu

בטבלה התחתונה – logistic

מקסימום איטרציות: 100

תוצאה אופטימלית: ראינו שה- logistic הניב אחוזים מעט גבוהים יותר על הוולידיציה.

size_	size_2	size_3	train_acc	test_acc
80	20	20	1.00000	0.98504
80	50	50	1.00000	0.98291
20	20	50	1.00000	0.98086
50	20	50	1.00000	0.98000
80	20	80	1.00000	0.97949
20	50	20	1.00000	0.97940
80	20	50	1.00000	0.97889
80	50	20	1.00000	0.97855
50	50	50	1.00000	0.97808
50	20	20	1.00000	0.97799
80	80	80	1.00000	0.97786
80	80	20	1.00000	0.97756
20	20	20	1.00000	0.97671
50	80	80	1.00000	0.97667

size_	size_2	size_3	train_acc	test_acc
80	80	50	0.99881	0.98521
80	80	20	0.99787	0.97748
80	50	20	0.99865	0.97588
80	50	50	0.99844	0.97560
50	50	50	0.99862	0.96846
50	80	20	0.99865	0.95930
80	20	50	0.99848	0.95252
50	50	20	0.99846	0.95150
50	20	20	0.99839	0.95051
80	20	20	0.99841	0.93789
50	80	50	0.99872	0.93152
80	20	80	0.99846	0.93109
80	50	80	0.99856	0.92199

שכבות חבויות: 4

נוירונים בכל שכבה: בין 1 ל- 140

פונקציית אקטיבציה: בטבלה העליונה – relu

בטבלה התחתונה – logistic

מקסימום איטרציות: 100

תוצאה אופטימלית: בדומה לניסיון הקודם, ראינו שה- logistic הניב אחוזים גבוהים יותר על הוולידיציה, רק שהפעם הפער קצת יותר משמעותי. קיבלנו 98.506% על הוולידיציה עם פונקציית relu ו- 99.154% עם פונקציית logistic.

size_	size_2	size_3	size_4	train_acc	test_acc
110	20	110	110	1.00000	0.98506
80	20	110	110	1.00000	0.98397
80	50	20	110	1.00000	0.98350
110	20	50	80	1.00000	0.98320
110	20	80	20	1.00000	0.98320
110	20	50	20	1.00000	0.98310
80	110	110	20	1.00000	0.98301
80	20	50	110	1.00000	0.98269
110	20	20	20	1.00000	0.98177
110	20	80	110	1.00000	0.98171
80	20	50	50	1.00000	0.98158
110	50	80	20	1.00000	0.98100
80	20	110	80	1.00000	0.98096

size_	size_2	size_3	size_4	train_acc	test_acc
110	80	80	20	0.99670	0.99154
80	80	20	20	0.99702	0.99096
80	110	20	50	0.99772	0.99032
110	110	20	80	0.99750	0.98996
110	110	50	20	0.99578	0.98876
110	50	50	20	0.99675	0.98865
110	110	20	50	0.99739	0.98855
110	110	20	20	0.99725	0.98820
80	50	20	20	0.99745	0.98784
50	80	20	20	0.99788	0.98692
110	50	20	20	0.99737	0.98641
110	80	20	50	0.99743	0.98607
80	110	20	20	0.99698	0.98586

נספח ג – פרמטרי SVM שכוננו

פרמטרים שכוונו:

```
best parameters: {'C': 0.05}
```

התוצאה על סט האימון: 0.9878

התוצאה על סט הבדיקה: 0.9304

```
training roc auc score : 0.9878492119466729
test roc auc score : 0.9304769426528016
```

נספח ד אחוזי דיוק עבור ממדים שונים בSVM:

עבור 4 ממדים:

```
the coef is: [[0.38438499 0.12902232 0.17048028 0.0429662 ]]
trainig ruc auc score : 0.9592478184971907
test ruc auc score : 0.7941635338345865
```

עבור 10 ממדים:

```
the coef is: [[ 0.38965785  0.11961542  0.17963322 -0.00127716 -0.00795598 -0.10017486
   0.00853785 -0.02218608 -0.01639931 -0.05021966]]
trainig ruc auc score : 0.9612822786397499
test ruc auc score : 0.7956766917293233
```

נספח ה התפלגות נתונים בSVM

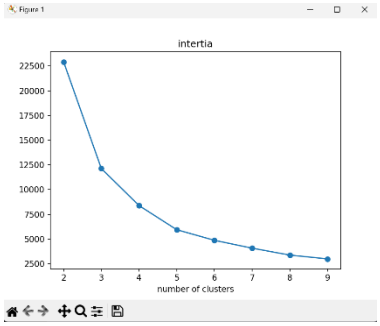
התפלגות הנתונים על סט האימון:


```
intercept: [-1.83352599]
the coef is: [[ 0.51129052 -0.01997494]]
trainig ruc auc score : 0.9260802060987197
test ruc auc score : 0.7893686104163924
SSR: [0.01885439 0.01865955]
SSR_SUM: 0.03751393340538367
```

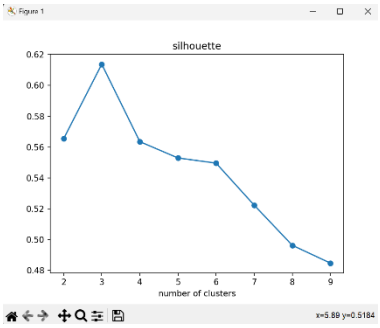
נספח ו – גרפים עבור בדיקות לקלסטרים

<u>נספח בדיקה עבור קלסטרים:</u>

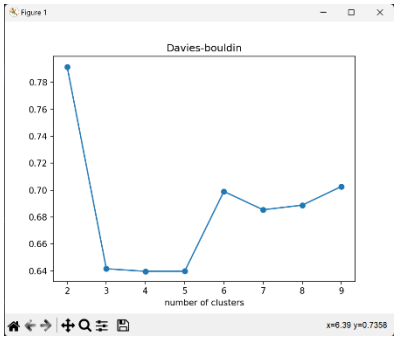
במדד intertia ה"מרפק" מתקבל בערך 3,



במדד silhouette הירידה החדה ביותר התקבלה בערך 3:



במדד Davies-bouldin העליה החדה ביותר התקבלה בערך 6:



נספח ז טבלאות עבור חלופה שנייה :

בשביל לחזק את טענה זו, חילקנו את סט האימון שוב ל-50-50 (הטבלה משמאל) ול- 75-25 (הטבלה מימין) והרצנו מודל פשוט יותר של רשת נוירונים עם מספר שכבות משתנות (1-4) ומספר נוירונים משתנים (1-100):

size_	train_acc	test_acc
(38,)	0.999999	0.930389
(53,)	0.999998	0.928327
(46,)	0.999998	0.927691
(74,)	0.999997	0.927362
(44,)	0.999998	0.924663
(22,)	0.999987	0.924005
(37, 67)	1.000000	0.919705
(31,)	0.999993	0.918630
(76, 29)	1.000000	0.917972
(70, 53)	1.000000	0.917007
(26,)	0.999996	0.916875
(55, 78)	1.000000	0.912246
(80, 58, 37, 47)	0.999753	0.910535
(80, 71, 34, 38)	1.000000	0.910250
(34, 34)	0.999999	0.910140
(28, 26)	0.999998	0.907705
(33, 34, 72)	1.000000	0.906981
(80, 59)	1.000000	0.906235
(27, 80, 31)	1.000000	0.905160
(70, 22, 61, 54)	1.000000	0.902505
(33, 26, 53, 57)	0.999999	0.901321
(34, 69)	1.000000	0.900926
(57, 21, 41)	0.999999	0.899785
(44, 48)	1.000000	0.899324
(40, 51, 22)	0.999999	0.898798

size_	train_acc	test_acc
(75,)	0.999998	0.933877
(68, 59, 56)	0.999999	0.930696
(62,)	0.999995	0.927779
(52,)	0.999998	0.927581
(61,)	0.999998	0.926901
(28,)	0.999983	0.926879
(49, 23)	1.000000	0.924992
(63,)	0.999995	0.922777
(48,)	0.999996	0.921065
(79,)	0.999993	0.920583
(75, 32)	0.999999	0.916392
(50,)	0.999992	0.915285
(29, 44, 66)	0.999999	0.915274
(57, 39, 43)	1.000000	0.914440
(79, 74, 51, 22)	1.000000	0.911654
(53, 57, 21, 62)	0.999937	0.911039
(32, 69)	1.000000	0.910688
(67, 36, 20)	0.999999	0.910469
(80,)	0.999995	0.910096