Ben-Gurion University of the Negev | Department of Industrial Engineering & Management

ML Project

Part B

Car Insurance

Group 7

Shany Balshan - 209054964

Keren Reif - 208494237

Yoav Ayalon - 209491018

תוכן עניינים

הכנת הנתונים לאימון ובחינה 3
Decision Trees 3
Neural Networks 8
Unsupervised Learning – Clustering 12
אימון מודל נוסף 15
השוואה בין המודלים 15
המודל הנבחר 16
נספחים 17

הכנת הנתונים לאימון ובחינה

בחרנו להגדיר את סט הבדיקה כ-20% מסך הנתונים, כלומר 1,600 רשומות מתוך 8,000. חלוקה זו תואמת להמלצות[1] לפיהן סט הבדיקה צריך להוות בין 20% ל-30% מהנתונים הכוללים, כדי להבטיח הערכה מהימנה של ביצועי המודל.

בנוסף, חילקנו את הנתונים באופן שמבטיח כי אחוז המבוטחים שתובעים את הביטוח יישמר זהה בסט האימון ובסט הבדיקה (חלוקה שמורה). מכיוון שמספר הרשומות הכולל שלנו יחסית קטן, חשוב לשמור על פרופורציות מדויקות בין הקבוצות, כדי להבטיח שהחלוקה מייצגת נאמנה את הנתונים ומאפשרת הערכה מדויקת של המודל.

Decision Trees

1. **הכנת הנתונים -** המודל בו נשתמש ליצירת עץ ההחלטה לא מסוגל לעבד את המידע עם משתנים קטגוריאליים שאינם ניתנים לסידור בצורתם המקורית, מכיוון שמדובר במודל שמצפה לקבל נתונים מספריים. לכן, כחלק מהכנת הנתונים במודל, ביצענו קיטלוג של הנתונים המקוריים לערכים מספריים באופן ידני. בנוסף, את המשתנים הרציפים/ בדידים עם מרחב ערכים גדול חילקנו לקבוצות וגם עליהן ביצענו קיטלוג באותו אופן. (בנספח 1.1 יש את המילון המלא של המשתנים והערכים).
את ההחלטה כיצד לחלק עשינו לאחר הסתכלות בהתפלגות גרפית של הנתונים לטווח הערכים השונים ועל סמך ידע אישי.
לדוגמא - שינינו את משתנה Income לערכים הללו -
{poverty=1, working class=2, middle class=3, upper class=4}
שינינו את משתנה Speeding Violations כאשר אם יש 0 עבירות = 1, 0-5 עבירות = 2, יותר מ-5 עבירות = 3.
2. **עץ החלטה מלא -** העץ המלא גדול ומורכב מידי, ולכן מופיע בנספח 1.2. בדקנו את המדדים F1 ודיוק[2]. הערכים שיצאו לנו הם:

	F1	דיוק
סט האימון	0.9402	96.39%
סט הבחינה	0.6294	76.62%

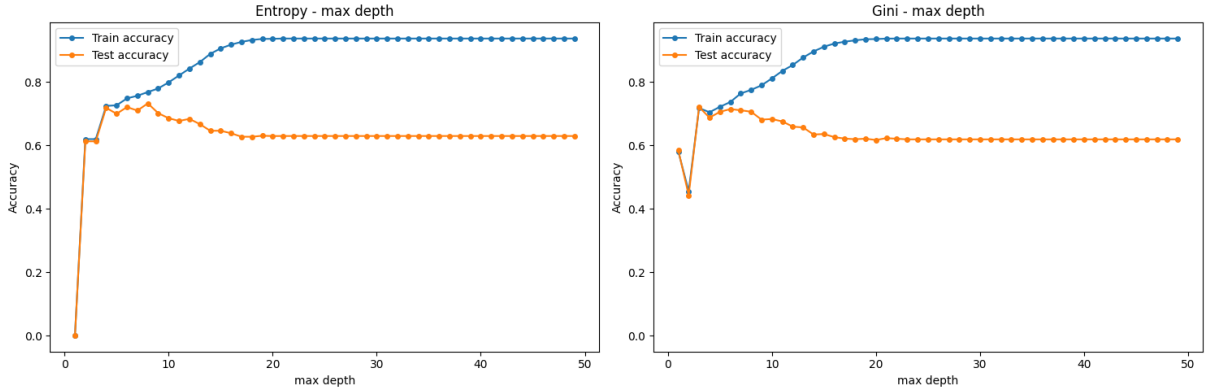
ציפינו למדדים גבוהים על סט האימון מכיוון שלא הגבלנו את המודל והוא לומד את הנתונים בצורה מלאה. ערכי המדדים יהיו נמוכים יותר בסט הבחינה כי המודל הותאם לנתונים בסט האימון, ולא על הנתונים הקיימים הסט הבחינה. על ההבדל הגדול יחסית בין השניים נוכל להגיד שיש במידה מסוימת

[1] Muraina, I. (2022, May). Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts. In *7th international Mardin Artuklu scientific research conference* (pp. 496-504).

[2] מדד F1 הוא מדד המשקלל את הדיוק (precision) ואת הרגישות (recall) של המודל. דיוק מתייחס לאחוז התחזיות החיוביות שנכונות, בעוד שרגישות מתייחסת לאחוז המקרים החיוביים שאותרו נכון על ידי המודל.
ערך המדד F1 נע בין 0 ל-1, כאשר ערך קרוב יותר ל-1 מצביע על מודל טוב יותר.

Overfitting - המודל התאים את עצמו לנתוני האימון. אנו מצפים שעץ מלא יביא תוצאות כאלו (הבדלים גדולים בין סט האימון לבחינה) משום שהוא לומד את נתוני האימון בצורה מלאה ומתאים את עצמו אליהם ולכן יביא ערך נמוך בהכרח בסט הבחינה.

3. **<u>כיוונון הפרמטרים -</u>** תחילה, ניסינו להבין איך שינוי הפרמטרים משפיע על ביצועי המודל. לכן עשינו גרף פשוט שבונה מודל כאשר אנו מגבילים רק את עומק העץ, ובכל פעם עומק עץ שונה מ-1 ל-50. ביצענו זאת על קריטריון אנטרופיה וג'יני וקיבלנו את הגרפים הבאים:



אפשר לראות שההתנהגות בין הקריטריונים דומה וכאשר עומק העץ גדול מעומק של 5-7, מדד הדיוק על נתוני המבחן יורדים. (בנספח 1.3 ביצענו גרפים תלת ממדיים שכוללים יותר מפרמטר אחד).

לאחר מכן, השתמשנו ב-Grid Search CV המגדיר רשת מלאה של כל הקומבינציות האפשריות של ערכי היפר-פרמטרים שאותם ניתן לבדוק עם קרוס-ולידציות. מטרת השימוש ב-Grid Search CV היא למצוא את השילוב האופטימלי של היפר-פרמטרים עבור המודל, כך שיביא לביצועים הטובים ביותר של המדדים. בחרנו לעשות קרוס ולידציה של 10, משמע הוא יחלק את הנתונים ל-10 קבוצות ויבדוק כל שילוב של הפרמטרים על כל קבוצה, כדי להבטיח הכללה טובה של המודל לנתונים חדשים. הבחירה בסט היפר-פרמטרים אלו נעשתה כדי לאפשר איזון בין גמישות המודל, שמאפשרת זיהוי קשרים מורכבים בנתונים, לבין הימנעות מבעיית התאמת יתר (Overfitting), שעלולה להתרחש כאשר המודל מתאים את עצמו יתר על המידה לנתוני האימון. בחרנו לכוונן את הפרמטרים הבאים:

- **Criterion -** קובע את הקריטריון לפיו יתפצלו הצמתים בעץ. השאיפה היא למצוא את הקריטריון שיביא את הביצועים הטובים ביותר בעץ ההחלטה. הקריטריונים שבחרנו הם שני מדדים נפוצים למידת האי-סדר בנתונים:

  Gini - מתעדף פשטות חישובית, מה שהופך אותו ליעיל מבחינת זמן עיבוד.

  Entropy - מתעדף פיצולים המספקים מקסימום מידע.

  השוואת הקריטריונים מאפשרת להבין האם מודל שמתעדף פשטות עדיף על מודל שמעדיף דיוק חישובי בהקשר הספציפים של הנתונים. בחירה בקריטריון שונה עשויה להשפיע על מבנה העץ, עומקו, ויכולתו לשפר את הביצועים על נתוני המבחן.
- **max_depth -** מגביל את העומק המקסימלי של העץ. העומק משפיע ישירות על יכולת המודל ללמוד דפוסים מורכבים. בחירת טווח ערכים רחב (3 עד 22) מאפשרת לבדוק אילו עומקים תורמים לדיוק

מבלי ליצור עץ עמוק מדי שמוביל להתאמת יתר. עומק גדול יותר עשוי לשפר את הביצועים על נתוני אימון אך להוביל לירידה בביצועים על נתוני מבחן. עומק קטן מדי עלול למנוע מהעץ לזהות קשרים חשובים בנתונים.

- **min_samples_leaf -** קובע את המספר המינימלי של דוגמאות שנדרשות בכל עלה של העץ. היפר-פרמטר זה נועד למנוע מהעץ ליצור עלים קטנים מאוד שמבוססים על מעט מדי דוגמאות, מה שעלול לגרום להתאמת יתר. עלייה בערך זה מגדילה את הכללה של המודל אך עשויה להפחית את הדיוק במקרים מסוימים. ערך נמוך מדי עלול לגרום לעץ להסתמך יתר על המידה על דוגמאות בודדות. הטווח שנבדק: 5-25.
- **max_features -** קובע את המספר המקסימאלי של המאפיינים שיילקחו בחשבון בעל פיצול של העץ בצורה רנדומלית. על ידי צמצום מספר המאפיינים האפשריים, ניתן להפחית את ההסתברות ליצירת מודל שמתאים יתר על המידה לנתונים. ערכים נמוכים יותר מגבילים את החיפוש במשתנים ומפחיתים את זמן החישוב, אך עלולים להוריד את דיוק המודל. הערכים שנבדקו הם: 10,11,12,13,14 מפני שאנו רוצים לשמור על דיוק המודל (ערך None מאפשר בדיקה ללא הגבלה).
- **ccp_alpha -** משמש לגיזום (Pruning) של העץ על ידי הוספת עלות גיזום על צמתים מיותרים בעץ. גיזום מבוקר של העץ מסייע למנוע התאמת יתר על ידי הסרת צמתים שאינם תורמים משמעותית לביצועי המודל. ערך גבוה יותר מוביל לעצים קטנים יותר ופחות מורכבים, בעוד שערך נמוך שומר על עץ מפורט יותר. הטווח שנבחר מאפשר איזון בין הפשטת המודל לבין שמירה על המידע הדרוש. הטווח שנבדק: 0-0.001 בקפיצות של 0.0005.

לאחר כיוונון הפרמטרים, הערכים שהתקבלו הם: (נספח 1.4)

Criterion: gini, Max depth: 7, Max features: 14, Min samples leaf: 9, ccp alpha: 0

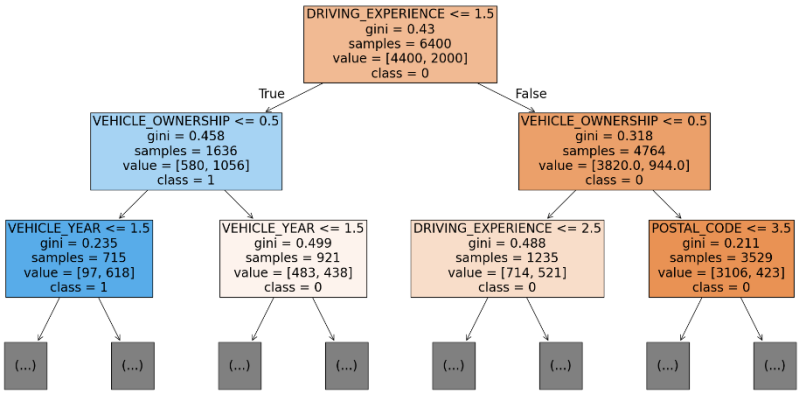
הוצאנו מפות חום מהתוצאות של Grid Search CV על ערך המדד הנבדק - F1 (נספח 1.5).
בגרף הראשון, ניתן לראות כי עבור מספר המאפיינים והעומק העץ, קיבלנו מדד F1 נמוך כאשר עומק העץ היה קטן מ-5, וערך מדד זהה כאשר העומק גדול מ-11, ללא תלות במספר המאפיינים בעץ. מאוד מובחן בגרף כי עומק עץ 7 מקבל ערך מדד גדול יותר כאשר מספר המאפיינים בעץ גדל.
בגרף השני, ניתן לראות כי קריטריון ג'יני מקבל ערכי מדד טובים יותר מאשר אנטרופיה, ללא תלות במספר המינימאלי של דוגמאות בעלה. כמו כן, ערך המדד גבוה כאשר מספר הדוגמאות בעלה הוא 9, ומעל 22.

4. **<u>אימון עץ החלטה עם הקונפיגורציה הטובה ביותר -</u>** לאחר אימון הנתונים על המודל עם היפר-הפרמטרים שקיבלנו ב-Grid Search CV, קיבלנו את ערכי המדדים הבאים:

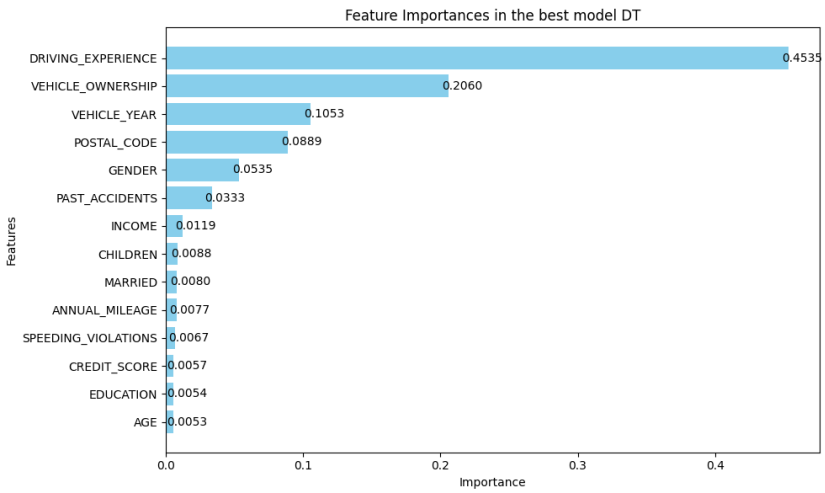
	F1	דיוק
סט האימון	0.7608	85.14%
סט הבחינה	0.7135	81.62%

ניתן לראות שערכי המדדים על סט המבחן גבוהים יותר מאשר על העץ המלא. התוצאה תואמת לציפיות שלנו משום שבנינו מודל על סמך חיפוש פרמטרים אשר יביא לערך מדד גבוה, והפער בין סט האימון לבחינה קטן משמעותית, מה שמצביע על התמודדות טובה יותר עם נתונים שלא נראו וסבילות להתאמת יתר (Overfitting).

העץ המתקבל (<u>בנספח 1.6</u> מופיע העץ המלא) -



ניתן לראות כי המאפיין הראשון שנבחר הוא Driving Experience מה שמעיד על כך שזהו המאפיין החשוב במודל. כמו כן, בפיצול השני נבחר המאפיין Vehicle Ownership בשתי הפיצולים, לכן אפשר להגיד שגם הוא מאפיין חשוב במידה ניכרת ואף דומה למאפיין הראשון.



על מנת לראות באופן ברור את ההשפעה שיש לכל מאפיין על מדד F1, ביצענו חשיבות משתנים על המודל שאימנו.
ניתן לראות, כמצופה, שהמדדים בעל החשיבות הגבוהה ביותר הם - Driving Experience (0.4535) ו- Vehicle Ownership (0.206). נראה כי רוב המשתנים בעלי חשיבות נמוכה ואף זניחה.
הופתענו לגלות כי המאפיינים Past Accidents ו-Speeding Violations יחסית לא חשובים. דבר זה מפתיע בגלל האינטואיציה לחשוב שמידע על עבירות ותאונות יכול לחזות טוב יותר על תביעות ביטוח עתידיות. כמו כן המאפיינים שחשיבותם גבוהה יותר עולים בקנה אחד עם המאפיינים שהעץ בחר כחשובים יותר.

סעיף בונוס - בחרנו רשומה לדוגמא מסט הבחינה -

ID=1, GENDER=female, EDUCATION=high school, INCOME=middle class, CREDIT_SCORE=0.541595591, VEHICLE_OWNERSHIP=0, VEHICLE_YEAR=before 2015, MARRIED=1, CHILDREN=1, POSTAL_CODE=32765, ANNUAL_MILEAGE=11000, VEHICLE_TYPE=sedan, SPEEDING_VIOLATIONS=1, PAST_ACCIDENTS=0, AGE=39, DRIVING_EXPERIENCE=14

לאחר הורדת המשתנים שהסרנו והתאמת הערכים לקטגוריזציה שביצענו בסעיף 1 (לפי הטבלה בנספח 1.1), קיבלנו תחזית שהלקוחה תתבע את חברת הביטוח. כמו כן ישנה התאמה של 54% לחיזוי זה.

```
Decisions made by the tree:
If DRIVING_EXPERIENCE <= 1.50?
 -> No: DRIVING_EXPERIENCE=2 > 1.50)
If VEHICLE_OWNERSHIP <= 0.50?
 -> Yes: VEHICLE_OWNERSHIP=0 <= 0.50)
If DRIVING_EXPERIENCE <= 2.50?
 -> Yes: DRIVING_EXPERIENCE=2 <= 2.50)
If VEHICLE_YEAR <= 1.50?
 -> Yes: VEHICLE_YEAR=1 <= 1.50)
If GENDER <= 0.50?
 -> No: GENDER=1 > 0.50)
If PAST_ACCIDENTS <= 1.50?
 -> Yes: PAST_ACCIDENTS=1 <= 1.50)
If MARRIED <= 0.50?
 -> No: MARRIED=1 > 0.50)

Leaf node reached. Predicted class: 1
Class probabilities: [0.45652174 0.54347826]
```

Neural Networks

1. **הכנת הנתונים -** בעת הכנת הנתונים עבור רשת נוירונים, ביצענו נרמול למניעת השפעה עודפת של תכונות עם ערכים גבוהים על תהליך האימון ולהבטחת תרומה שווה של כל התכונות. בחנו שלושה סוגי נרמול, התוצאות להלן:

סוג הנרמול	נוסחה	F1 Score (train set)	F1 Score (test set)	תיאור
StandardScaler	$z = \frac{(X - mean)}{(std)}$	0.8776	0.8250	קנה מידה המביא כל תכונה לממוצע 0 וסטיית תקן 1. נרמול זה מבטיח שכל תכונה תתרום באופן יחסי שווה, גם אם יחידות המידה שלה שונות. הוא מתאים בעיקר לנתונים שמתפלגים בצורה נורמלית.
MinMaxScaler	$z = \frac{(X - \min)}{(max - \min)}$	0.8627	0.8249	קנה מידה המביא כל תכונה לטווח מוגדר מראש, בדרך כלל בין 0 ל-1. שיטה זו שומרת על היחסים בין הערכים, אך עלולה להיות מושפעת מקיומם של ערכים קיצוניים.
RobusterScaler	$z = \frac{(X - median)}{(IQR)}$	0.8711	0.8240	קנה מידה המבוסס על חציון (median) והטווח הבין-רבעוני(IQR) , מה שהופך אותו לעמיד בפני ערכים חריגים. הוא מתאים במיוחד לנתונים שבהם קיימים חריגים משמעותיים שעלולים לעוות את הניתוח.

מצאנו כי השימוש ב-**StandardScaler** הוא המתאים ביותר עבור סט נתונים זה, כיוון שהוא מוביל לערך הגבוה ביותר של מדד F1, המבטא איזון בין דיוק (precision) לרגישות (recall).

2. **רשת ברירת מחדל -** ערכי ברירת המחדל של רשת הנוירונים הם:

מספר נוירונים בשכבת הכניסה - 14. יש לנו 17 מאפיינים בבסיס הנתונים , אחד מהם הוא משתנה המטרה ולכן לא נכלל בשכבת הכניסה. כמו כן, הורדנו 2 מאפיינים (ID, VEHICLE_TYPE).

מספר שכבות חבויות - 1. מספר השכבות בין שכבת הקלט לשכבת הפלט ברשת, שבהן מתבצע חישוב ביניים. שכבה חבויה אחת מאפשרת לשמור על מבנה פשוט ויעיל של הרשת, שעדיין מאפשר לה ללמוד קשרים לא ליניאריים בין הקלט לפלט.

מספר נוירונים חבויים בכל שכבה - 100. מספר היחידות בכל שכבה חבויה שמבצעות חישובים על המידע הנכנס אליה. מספר זה של נוירונים מספק לרשת מספיק יכולת חישובית ללכוד תבניות מורכבות בנתונים, תוך שמירה על איזון בין ביצועים לבין סיכון של מורכבות יתר.

פונקציית אקטיבציה - ReLU. פונקציה שמחליטה אם נוירון יופעל או לא, על ידי יצירת קשר לא ליניארי בין הקלט לפלט. פונקציית האקטיבציה ReLU מחזירה את הערך עצמו אם הוא חיובי, ואת הערך 0 אם הוא שלילי. הפונקציה נפוצה ברשתות נוירונים בזכות פשטותה וחישוביה היעילים, והיא עוזרת למנוע בעיית דעיכת גרדיאנטים.

מדד הביצוע שבחרנו המתקבל על סט האימון וסט הבחינה הינו F1 ודיוק.

ערכי המדד שהתקבלו הם (נספח 2.1) :

	F1	דיוק
סט האימון	0.8793	88.00%
סט הבחינה	0.8287	82.88%

תוצאה זו מצביעה על כך שהמודל מצליח לזהות באופן מדויק ומאוזן את הקשר בין דיוק(Precision) לשליפה (Recall) בסט האימון, עם ביצועים גבוהים יחסית. עם זאת, הירידה בציון בסט הבחינה מעידה על כך שהמודל עשוי להיות מעט מותאם מדי לנתוני האימון (Overfitting) , מה שעלול לפגוע ביכולת ההכללה שלו לנתונים חדשים .

חשוב לציין שמדד ה-F1 בסט האימון אינו 1 מכיוון שערכי ברירת המחדל של המודל מגבילים את מספר האיטרציות ל-200. הגבלה זו עשויה לגרום לכך שהמודל לא יגיע לקונפיגורציה מלאה (מצב בו המודל לא משתפר יותר באופן משמעותי), ולכן ביצועיו אינם מושלמים גם על נתוני האימון. ניתן לשפר את הציון על ידי הגדלת מספר האיטרציות או כיוונון נוסף של הפרמטרים.

3. **כיוונון היפר פרמטרים -** הפרמטרים אותם בחרנו לכוונן הם:

- **מספר הנוירונים בשכבות החבויות ומספר השכבות החבויות -** כמות השכבות החבויות ומספר הנוירונים בכל שכבה משפיעים על יכולת המודל ללמוד תבניות מורכבות בנתונים. רשת עם שכבות מועטות או מעט נוירונים עשויה להיות פשוטה מדי ,(Underfitting) בעוד שרשת גדולה מדי עלולה להיות מורכבת מדי ולהוביל להתאמת יתר .(Overfitting) לכן חשוב לכוונן פרמטרים אלה בהתאם למורכבות הנתונים ולגודל הdataset.
הערכים שבדקנו הם: [(200,150,100) ,(250,100),(100,50),(100,),(50,)].
- **פונקציית אקטיבציה -** בחירת פונקציית האקטיבציה משפיעה על אופן הפעלת הנוירונים ועל היכולת של הרשת ללמוד קשרים לא ליניאריים. ReLU מתאימה לרוב הבעיות בגלל חישוביה הפשוטים והיעילות במניעת דעיכת גרדיאנטים, בעוד Tanh עשויה להיות עדיפה במקרים מסוימים בזכות הערכים שהיא מייצרת, שנעים בין (-1) ל-1, מה שעוזר לרשת להתכנס מהר יותר. ו-logistic, מתאימה למקרים בהם בעיית הסיווג היא בינארית. הערכים שבדקנו הם: [ReLU, tanh, logistic].
- **Solver** - בחירת האלגוריתם לאופטימיזציה משפיעה על האופן שבו המודל מעדכן את המשקלים. Adam משלב את היתרון של SGD (יעילות על סט נתונים גדול) ולכן יעיל ברוב המקרים, כמו כן הוא טוב לבעיות תזמון. כוונון ה-solver חשוב כדי לשפר את מהירות הלמידה והדיוק. הערכים שבדקנו הם: [Adam, SGD].
- **Alpha** - הפרמטר Alpha שולט בעוצמת הרגולריזציה (L2 Regularization) , המונעת מהמודל להתאים את עצמו יתר על המידה לנתוני האימון. כוונון הערך של Alpha עוזר לאזן בין ביצועים על סט האימון ליכולת ההכללה לנתונים חדשים. הערכים שבדקנו הם: [0.001, 0.01].

- **Learning Rate Init** - קצב הלמידה ההתחלתי מכתיב את גודל הצעדים שהמודל לוקח בעת עדכון המשקלים. ערך גבוה מדי עלול לגרום למודל לדלג על הפתרון האופטימלי, בעוד שערך נמוך מדי עלול לגרום ללמידה איטית מאוד. כוונון קצב הלמידה הוא קריטי להשגת קונברגנציה יעילה. הערכים שבדקנו הם: [0.001].
- **Max Iterations** - קובע את מספר האיטרציות המקסימליות שמודל הנוירונים יבצע במהלך הלמידה. חשוב לכוונן פרמטר זה מכיוון שמספר איטרציות נמוך מדי עלול להוביל לכך שהמודל לא יספיק להתכנס ולמצוא את הפתרון האופטימלי, מה שיביא לביצועים גרועים. לעומת זאת, מספר איטרציות גבוה מדי יכול להוביל לבזבוז זמן ומשאבים אם המודל כבר הגיע לקונברגנציה. כוונון היפר פרמטר זה עוזר לאזן בין ביצועים יעילים לבין זמן הלמידה. הערכים שבדקנו הם: [500].
- **Batch Size** - גודל ה-batch משפיע על אופן עדכון המשקלים בכל איטרציה. באצ'ים קטנים מספקים עדכונים מהירים אך רעשניים, בעוד שבאצ'ים גדולים יותר מספקים עדכונים מדויקים יותר אך דורשים יותר משאבי חישוב. כוונון נכון של גודל הbatch עוזר לשפר את מהירות הלמידה והדיוק הכולל. הערכים שבדקנו הם: [32, 64, 128].

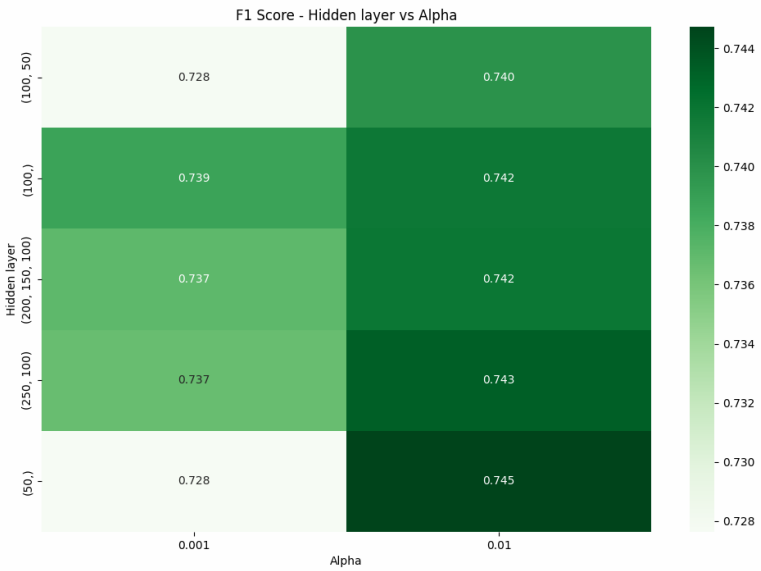
ביצענו כמה בדיקות והרצות של חיפוש ברשת, ומצאנו כי באופציות המובילות מבחינת המדדים שלנו (F1 ודיוק) ישנם פרמטרים מסוימים שהערך שלהם לא השתנה בין הקומבינציות השונות ולכן בחרנו בחיפוש הסופי לכוון אותם לערך קבוע.
הערכים שהתקבלו לאחר כוונון הפרמטרים:

Activation: logistic , alpha: 0.01, batch size: 32 , hidden layer size: (50,), learning rate init : 0.001, max iterations: 500, solver: adam

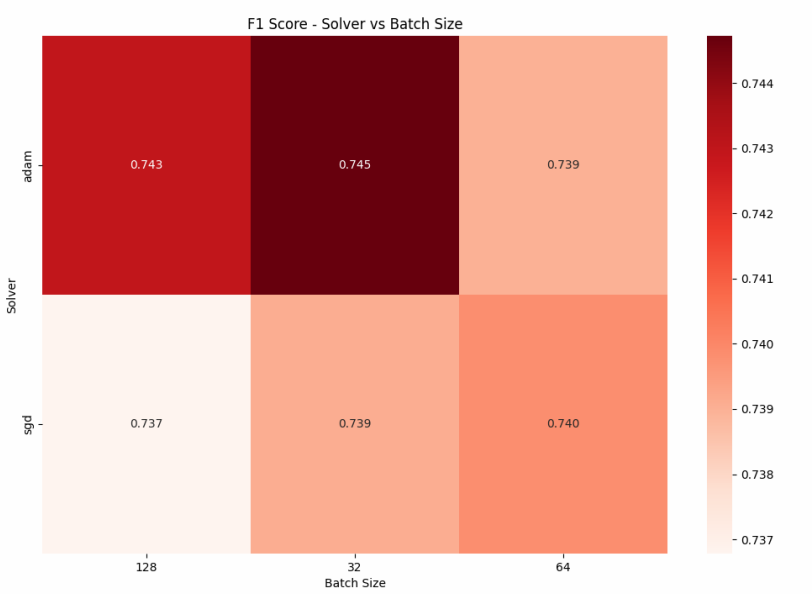
גרפים עבור ערכי המדד שנבחרו:

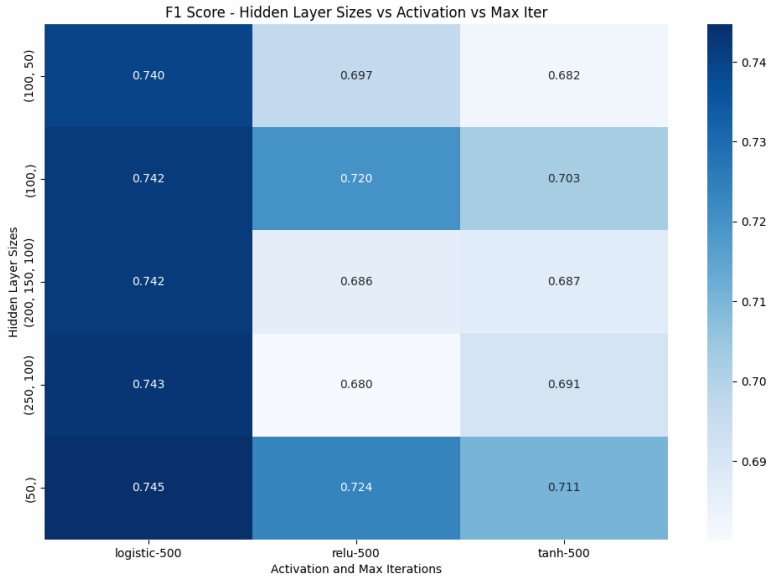
<u>Hidden layer vs. Alpha</u>

ערך האלפא (alpha) שולט על הרגולריזציה (מונע התאמת יתר), בכך ש"מעניש" על משקלים גדולים יותר שכבות גדולות ומורכבות יותר. מספר השכבות והנוירונים בשכבות מנסים להסביר את המודל, וככול שיש יותר שכבות ונוירונים בכל שכבה, כך המודל יכול להיות מדויק יותר ועם זאת להיות שחוף להתאמת יתר.



ניתן לראות שאלפא גבוהה מביאה ביצועים טובים יותר בכל סוגי השכבות שבחרנו לבדוק. הערכים של מדד F1 גבוהים יותר ככול שיש פחות שכבות ושכבות מורכבות פחות.





<u>Solver vs. Batch Size</u>

Adam עובד טוב עבור סט נתונים בגודל בינוני-גדול. אפשר להגיד כי סט הנתונים שלנו הוא במידות אלו, ולכן ציפינו כי הוא יעבוד טוב יותר מ-SGD.

הגרף מראה כי אלגוריתם Adam השיג תוצאות גבוהות יותר במדד F1 עבור כל ערך של batch size, כצפוי.

כמו כן, עבור חלוקת סט הנתונים להרבה קבוצות קטנות יותר (גודל באצ'), ניתן לראות שהוא משיג תוצאות טובות יותר למדד F1.

<u>פונקציית אקטיבציה ומספר השכבות החבויות</u>

פונקציית האקטיבציה יכולה להגביל את יכולת הלמידה לפי השכבות השונות של הרשת - רשתות קטנות ופשוטות יותר יעבדו טוב יותר עם logistic, tanh ובעוד שרשתות גדולות ומורכבות יעבדו טוב עם relu.

ניתן לראות לפי הגרף שדווקא relu עובדת יותר טוב עם רשתות קטנות ופשוטות, אך בכל מקרה, logistic עובד הרבה יותר טוב בכל סוגי הרשתות שבחנו.

4. **אימון ובחינה של הקונפיגורציה האופטימאלית** - הקונפיגורציה האופטימלית שנבחרה למודל היא - פונקציית האקטיבציה שנבחרה היא logisticבשל התאמתה לבעיות סיווג בינאריות והיכולת שלה לייצג הסתברויות. קצב הלמידה (alpha) הוגדר כ-0.01 כדי לשמור על יציבות בלמידה. גודל ה-batch size הוגדר ל-32 כתחום ביניים בין מהירות חישוב להתכנסות יעילה. גודל השכבה החבויה נקבע ל-50 נוירונים כדי לאפשר למודל גמישות מספקת ללמידת דפוסים. קצב הלמידה ההתחלתי נקבע ל-0.001, לערכים קטנים המקדמים למידה מדויקת ומניעת קפיצות גדולות מדי. מספר האיטרציות המקסימלי נקבע

ל-500 כדי להבטיח התכנסות מלאה (ועדיין בחלק מהקומבינציות לא קיבלנו התכנסות מלאה). ה-solver שנבחר הוא Adam בזכות יעילותו בעבודה עם בעיות מורכבות ואופטימיזציה יציבה.
הקונפיגורציה כולה נבחרה לאחר חיפוש ברשת שנעשה כדי להגיע לביצועים הטובים ביותר.

ערכי המדדים שבחרנו על נתונים האימון והבחינה הם (נספח 2.1):

	F1	דיוק
סט האימון	0.8448	84.58%
סט הבחינה	0.8349	83.44%

ערכי המדדים על סט הבחינה קרובים לערכי המדדים על סט האימון, מה שמעיד על כך שהמודל שומר על ביצועים טובים גם על נתונים שלא נראו במהלך האימון. הפער הקטן בין המדדים מצביע על כך שהמודל לא סובל מהתאמת יתר, כלומר, הוא מצליח להעביר את הידע שנלמד מנתוני האימון לנתוני הבחינה בצורה יעילה. תוצאה זו עשויה להעיד על כך שהפרמטרים שנבחרו עבור האימון תואמים היטב לנתונים.
מטריצות מבוכה (נספח 2.2) - ניתן לראות שהשגיאות מסוג ראשון וסוג שני קטנות יחסית גם בסט האימון וגם בסט הבחינה (בסט האימון יש 15.31% של טעויות, ובסט הבחינה יש 16.56% של טעויות).
ניתן לראות שההבדלים בין הטעויות לא גדולים, לכן נוכל להגיד כי יכולת ההכללה של המודל מספקת בין נתונים בהם הוא ראה (האימון) ועל כאלו אשר הוא לא ראה (הבחינה).

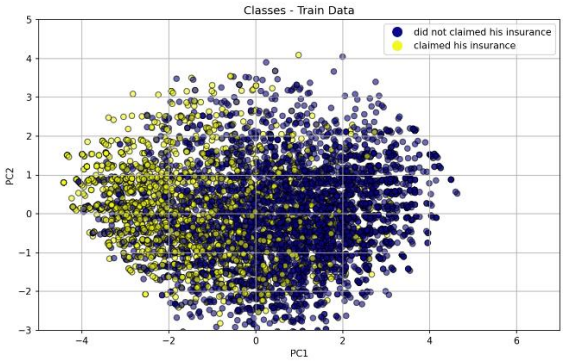
Unsupervised Learning – Clustering

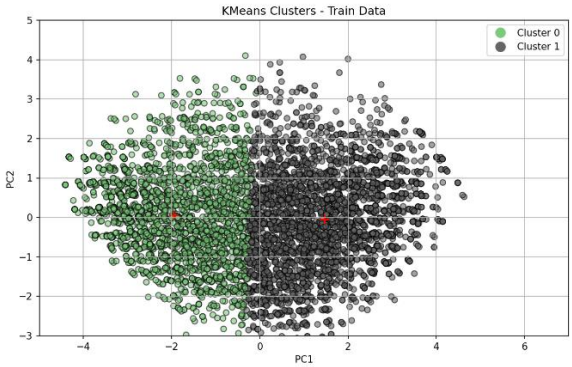
1. **מודל ברירת מחדל -** הרצנו את מודל K-Means על סט האימון עם ערכי ברירת המחדל, פרט למספר האשכולות שאותו קבענו לפי מספר המחלקות של משתנה המטרה שלנו (OUTCOME), שהוא 2 (0 או 1).
K-Means הוא אלגוריתם קלאסי ללמידת clustering, שבו נתונים ממוינים לקבוצות (מחלקות) על פי דמיון בין אובייקטים. השאיפה היא למזער את המרחק הפנימי בתוך כל קבוצה, כלומר, להקטין את הסכום של ריבועי המרחקים בין האובייקטים למרכזי הקבוצות.
2. אלגוריתם K-Means הצליח לחלק את הנתונים ל-2 אשכולות ברורים אך ניתן לראות לפי הגרף כי לא הצליח להבחין בצורה מדויקת בין שני הקבוצות. אופן שיוך התצפיות לאשכולות נעשה באמצעות סיווג כל קבוצה על ידי פונקציה שמטרתה למצוא את התווית הנפוצה ביותר בכל אשכול, נשתמש בתווית זו כתווית הקבוצה החדשה.

ערכי המדד שבחרנו על נתונים האימון והבחינה הם (נספח 3.1):

	F1	דיוק
סט האימון	0.7638	75.59%
סט הבחינה	0.7637	75.56%

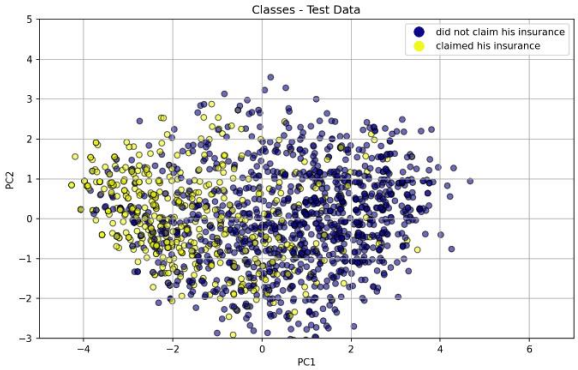
בגרף השמאלי נראה כי המחלקות האמיתיות מציגות פיזור יחסית מעורב עם חפיפה בין הלקוחות שתבעו את הביטוח לבין אלה שלא. לעומת זאת, בגרף ימין, ניתן לראות שהאשכולות שנוצרים ע"י האלגוריתם מחלקים את הנתונים באופן הרבה יותר סימטרי סביב ציר PC1 ביחס לנתונים המקוריים. אין הבדל בין המדדים על סט האימון והבחינה, מכיוון שכי שציינו, אין לסט הבחינה משמעות, אלא רק להשוות בין המודלים האחרים.

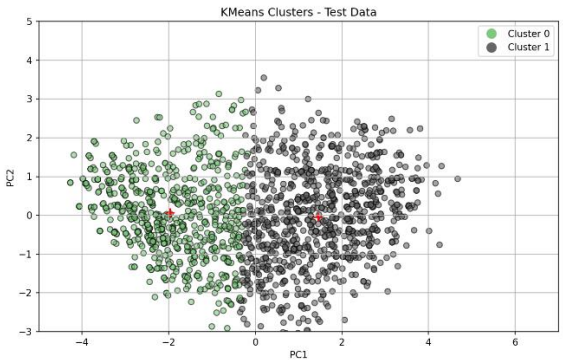




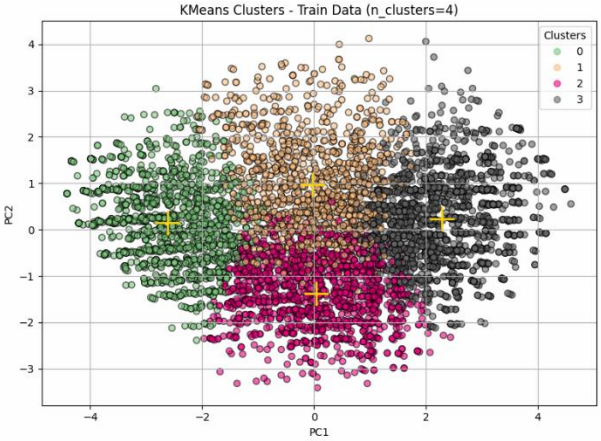
על פי גרף שמאל ניתן לראות שגם בנתוני הבחינה ישנה חפיפה בין המחלקות אך למרות החפיפה ניתן לראות שהמודל מצליח ליצור אשכולות עם רמת דיוק סבירה (לפי גרף ימין). החלוקה נראית דומה לסט האימון הן של הנתונים האמיתיים והן של האלגוריתם.

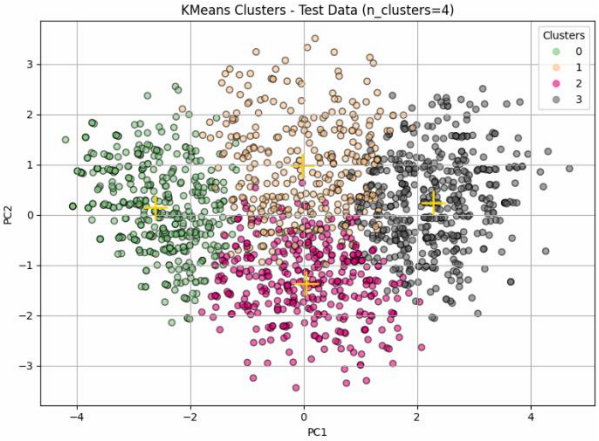
לפי הגרפים ניתן לומר שהמודל יחסית יציב ומסווג את הנתונים בצורה סבירה.





<u>סעיף בונוס -</u> ביצענו 8 מודלים נוספים של K-means עם מספר מחלקות שונה (בהתעלמות מאופי סט הנתונים). מצאנו כי כאשר נחלק ל-4 מחלקות, אנו נקבל את מדד F1 הגבוה ביותר. בנספח 3.2 ניתן לראות את התוצאות של כל שמונת המודלים.





הרוב המוחלט של התוצאות קיבל ערך מדד F1 יותר גבוה מחילוק ל-2 מחלקות, אך בהבדלים קטנים יחסית. כפי שניתן לראות בגרפים למעלה, הנתונים דיי חופפים בין המחלקות. אין סיבה מסוימת בה הוא בחר לחלק דווקא ל-4 מחלקות, אך אנו מבינים בגלל החפיפה הגדולה בין שתי המחלקות המקוריות, שאין בהכרח קו לינארי יחיד בו ניתן לחלק את הנתונים לשניים וצריך חלוקה מורכבת יותר מזו, ולכן נשער שזו הסיבה שמדד F1 הוא גבוה יותר בכל השאר החלוקות שאינן שתי מחלקות.

אימון מודל נוסף

בחרנו לאמן את המודל XGBoost - מודל מיטבי לסיווג שמתמודד עם נתונים מורכבים וחוסר איזון בין המחלקות.XGBoost מבוסס על Gradient Boosting כלומר, הוא בונה עצים באופן איטרטיבי, כך שכל עץ חדש מתקן את השגיאות של העצים הקודמים.

היפר-פרמטרים שנבחרו לבחינת המודל והקונפיגורציה המיטבית:

- max_depth=8 - מגביל את עומק העצים כדי למנוע למידת יתר ולשפר הכללה.
- learning_rate=0.3 - קובע את קצב הלמידה ומאזן בין מהירות האימון לביצועים.
- n_estimators=100 - מספר העצים במודל, המספק יציבות מבלי להאריך יתר על המידה את זמן החישוב.
- subsample=0.6 - משתמש רק ב-60% מהנתונים לכל עץ, כדי להפחית למידת יתר.
- min_child_weight=10 - מגביל פיצול של צמתים.

למרות שהמודל אמור להתמודד טוב עם חוסר איזון, במדדים שלנו הוא קיבל ציון F1 נמוך יותר ממודל ה-NN, שכנראה הצליח ללמוד דפוסים חבויים שה-XGBoost פחות רגיש אליהם, או שהוא מותאם יותר לנתונים הספציפיים שלנו.

תוצאות המודל במודל ברירת המחדל ולאחר כוונון הפרמטרים (נספח 4.1):

לפני כוונון פרמטרים	F1	דיוק
סט האימון	0.8590	91.20%
סט הבחינה	0.7087	81.25%

אחרי כוונון פרמטרים	F1	דיוק
סט האימון	0.8253	87.91%
סט הבחינה	0.7254	80.50%

השוואה בין המודלים

1. בסט הנתונים עליו עשינו את העבודה, הרשומות מתויגות למחלקות (האם תבעו את הביטוח או לא).

המודלים DT ו-NN הם מודלים המיועדים ללמידה מונחית - אשר מזהים דפוסים בנתונים ומנסים לשייך את הרשומה למחלקה ידועה. הבחירה בין DT ל-NN תלויה במורכבות הנתונים - עבור נתונים פשוטים עדיף להשתמש בעץ החלטה, בעוד שנתונים מורכבים ידרשו שימוש ברשת נוירונים, היודעת לזהות דפוסים וקשרים מורכבים יותר.

לעומת זאת, K-means הוא מודל שמתאים ללמידה לא מונחית - הוא מנסה לחלק את הנתונים ל-K אשכולות על סמך קרבה בין הנתונים מבלי להסתמך על התוויות. המטרה שלו היא למצוא את מרכזי האשכולות ולשייך כל תצפית לאשכול הקרוב אליה ביותר.

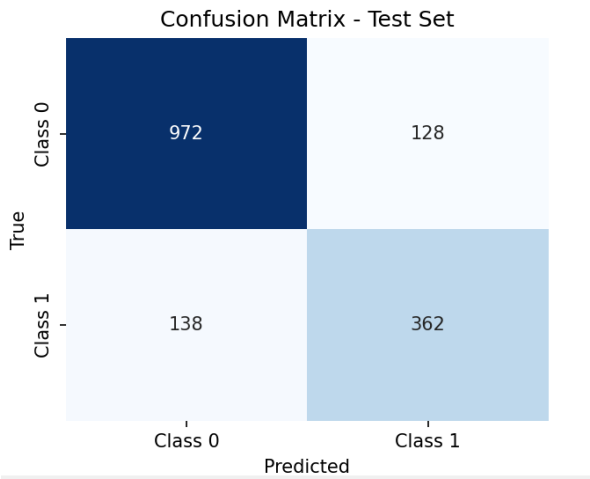
2. לאורך העבודה, השוונו את המודלים לפי מדד הדיוק ו-F1:

מודל	F1 אימון	F1 בחינה	דיוק אימון	דיוק בחינה
DT	0.7608	0.7135	0.8514	0.8162
NN	**0.8448**	**0.8349**	**0.8458**	**0.8344**
K-means	0.7638	0.7637	0.7559	0.7556
XGboost	0.8253	0.7254	0.8791	0.8050

המודל בעל הביצועים הטובים ביותר הינו NN על סט הבחינה. ניתן לראות שההפרשים בין המדדים בין האימון לבחינה הם הקטנים ביותר במודל זה, מה שמרמז על סבירות נמוכה להתאמת יתר ועל כך שמודל זה מתמודד עם נתונים חדשים בצורה טובה ובעל יכולת הכללה טובה יותר. ציפינו לראות מדדים נמוכים יותר ב-K-means לעומת ה-DT משום שהוא מתאים פחות למשימת סיווג, כפי שהסברנו לעיל, אך מדד F1 של K-means גבוה יותר גם על סט האימון וגם על סט הבחינה.
בנוסף, ניתן לראות הבדל גדול במדדים של המודל DT - מדד הדיוק מודד את אחוז הדגימות שסווגו נכון מכלל הדגימות, בעוד שמדד F1 הוא ממוצע של דיוק ושליפה (כפי שהסברנו בתחילת העבודה). כאשר סט הנתונים אינו מאוזן (68% שלא תבעו את הביטוח, ו-32% שכן), מדד הדיוק יכול להשיג תוצאה טובה יותר ע"י ניחוש של המחלקה הנפוצה מבלי ללמוד את הדפוסים של אותה מחלקה, בעוד שמדד F1 הינו מדד רגיש יותר ומושפע ממקרים בהם הוא לא מצליח לזהות את המחלקה הנפוצה.

<u>המודל הנבחר</u>

1. בחרנו לקחת את המודל של NN משום שהוא משיג את המדדים הטובים ביותר על סט הבחינה. ערכי ההיפר-פרמטרים שבחרנו הם - Activation: logistic , alpha: 0.01, batch size: 32 , hidden layer size: (50,), learning rate init : 0.001, max iterations: 500, solver: adam.
2. ניתן לראות כי כפי שציינו לעיל, שהמודל מזהה בצורה טובה יותר את מחלקה 0 בסט הבחינה, אשר מהווה נקודת חוזקה שלו.



עם זאת, משום שיש חוסר איזון בין המחלקות המודל נוטה לסווג דוגמאות רבות למחלקה 0 (138 תצפיות סווג למחלקה 0 למרות שהן ממחלקה 1 בפועל - 27.6%), מה שעלול להוות חולשה בביצועיו.
מבחינה עסקית, ניתן להגיד שמצד אחד טוב לזהות בצורה טובה יותר את מחלקה 0, משום שאז נוכל להציע תנאים טובים יותר עבור לקוחות שלא יתבעו את הביטוח ולהרוויח מהם ביותר וודאות, אך מצד שני, נרצה לזהות בצורה טובה את מחלקה 1 (הלקוחות שיתבעו את הביטוח) ולגבות מהם תשלום גבוה יותר עבורו.

נספחים

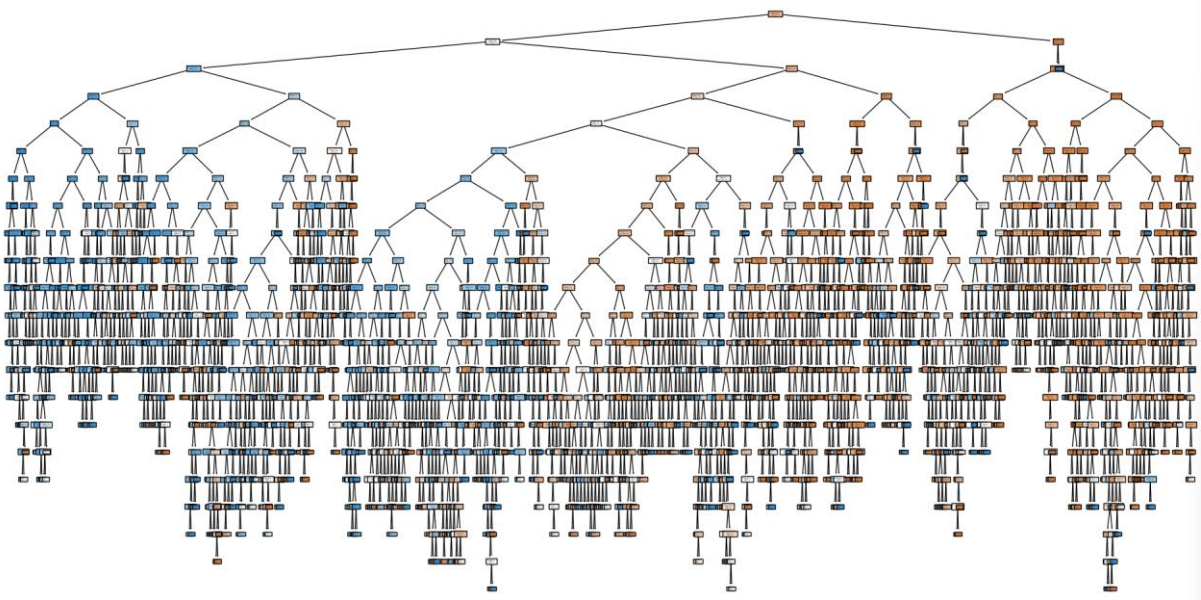
נספח 1 - עץ החלטה

נספח 1.1 - טבלת המשתנים לאחר עיבוד לעץ החלטה

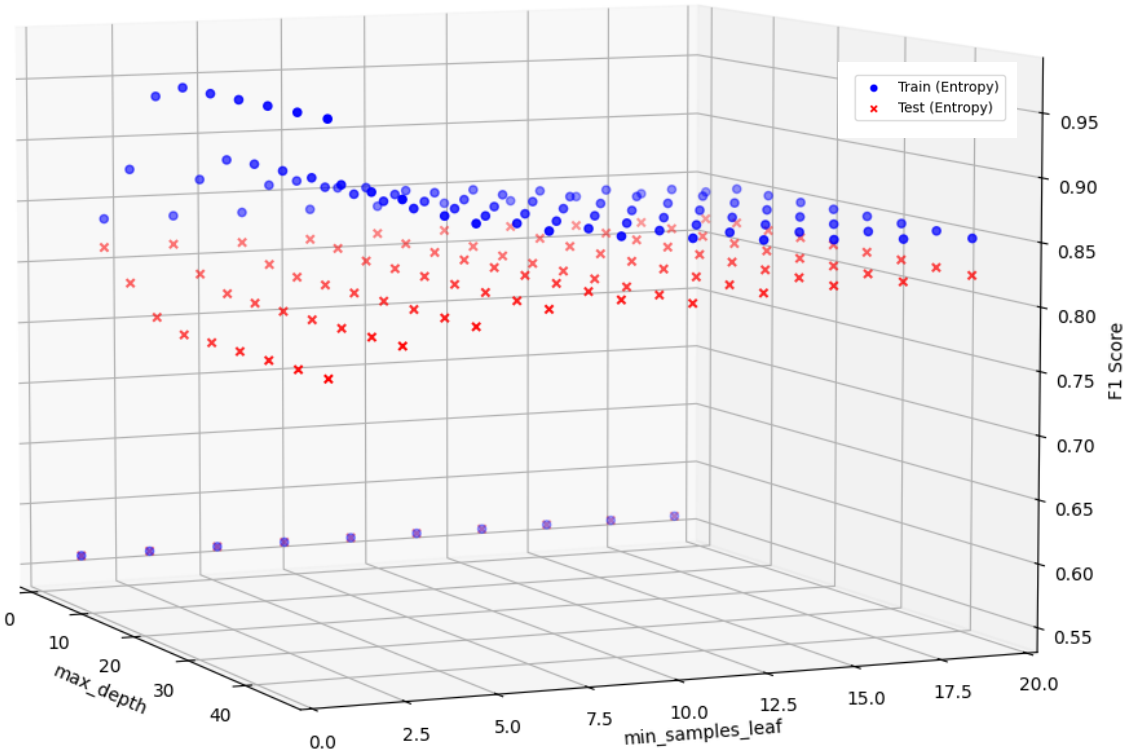
שם המאפיין	סוג מקורי	צורת עיבוד	מילון
GENDER	קטגוריאלי	הפיכה לבינארי	• 0=זכר • 1=נקבה
EDUCATION	קטגוריאלי	הפיכה לקטגוריאלי שניתן לסידור	• 1=תיכון, • 2=אוניברסיטה, • 3=ללא
INCOME	קטגוריאלי	הפיכה לקטגוריאלי שניתן לסידור	• 1=מעמד נמוך, • 2=מעמד עובדים, • 3=מעמד ביניים, • 4=מעמד גבוה
CREDIT_SCORE	רציף	הפיכה לקטגוריאלי שניתן לסידור	• 1= נמוך (נמוך מהממוצע פחות סטיית תקן אחת) • 2=בינוני (בין נמוך לגבוה) • 3=גבוה (גבוה מהממוצע ועוד סטיית תקן אחת)
VEHICLE_OWNERSHIP	בינארי	-	• 0= הלקוח לא בעל הרכב • 1= הלקוח בעל הרכב
VEHICLE_YEAR	קטגוריאלי	הפיכה לקטגוריאלי שניתן לסידור	• 1= לפני 2015 • 2= אחרי 2015
MARRIED	בינארי	-	• 0= לא נשוי • 1= נשוי
CHILDREN	בינארי	-	• 0= אין ללקוח ילדים • 1= יש ללקוח ילדים
POSTAL_CODE	בדיד	הפיכה לקטגוריאלי שניתן לסידור	• 1= 10238 • 2= 32765 • 3=92101 • 4=21217
ANNUAL_MILEAGE	בדיד-רציף	הפיכה לקטגוריאלי שניתן לסידור	• 1= נמוך (קטן מ-10,000) • 2=בינוני (בין 10,000 ל-15,000) • 3=הרבה (גבוה מ-15,000)
SPEEDING_VIOLATIONS	בדיד	הפיכה לקטגוריאלי	• 1= ללא (0 עבירות)

		שניתן לסידור	• 2=מעט (פחות מ-5 עבירות) • 3=הרבה (יותר מ-5 עבירות)
PAST_ACCIDENTS	בדיד	הפיכה לקטגוריאלי שניתן לסידור	• 1= ללא (0 תאונות) • 2=מעט (פחות מ-3 תאונות) • 3=הרבה (יותר מ-3 תאונות)
AGE	בדיד-רציף	הפיכה לקטגוריאלי שניתן לסידור	• 1= צעיר (גיל 16-32) • 2=מבוגר (גיל 32-67) • 3=זקן (מבוגר מ-67)
DRIVING_EXPERIENCE	בדיד-רציף	הפיכה לקטגוריאלי שניתן לסידור	• 1= מעט (פחות מ-5 שנים) • 2=בינוני (בין 5-15 שנים) • 3=הרבה (יותר מ-15 שנים)
OUTCOME	בינארי	-	• 0= לא תבע את הביטוח בעבר • 1= תבע את הביטוח בעבר

נספח 1.2 - עץ ההחלטה המלא



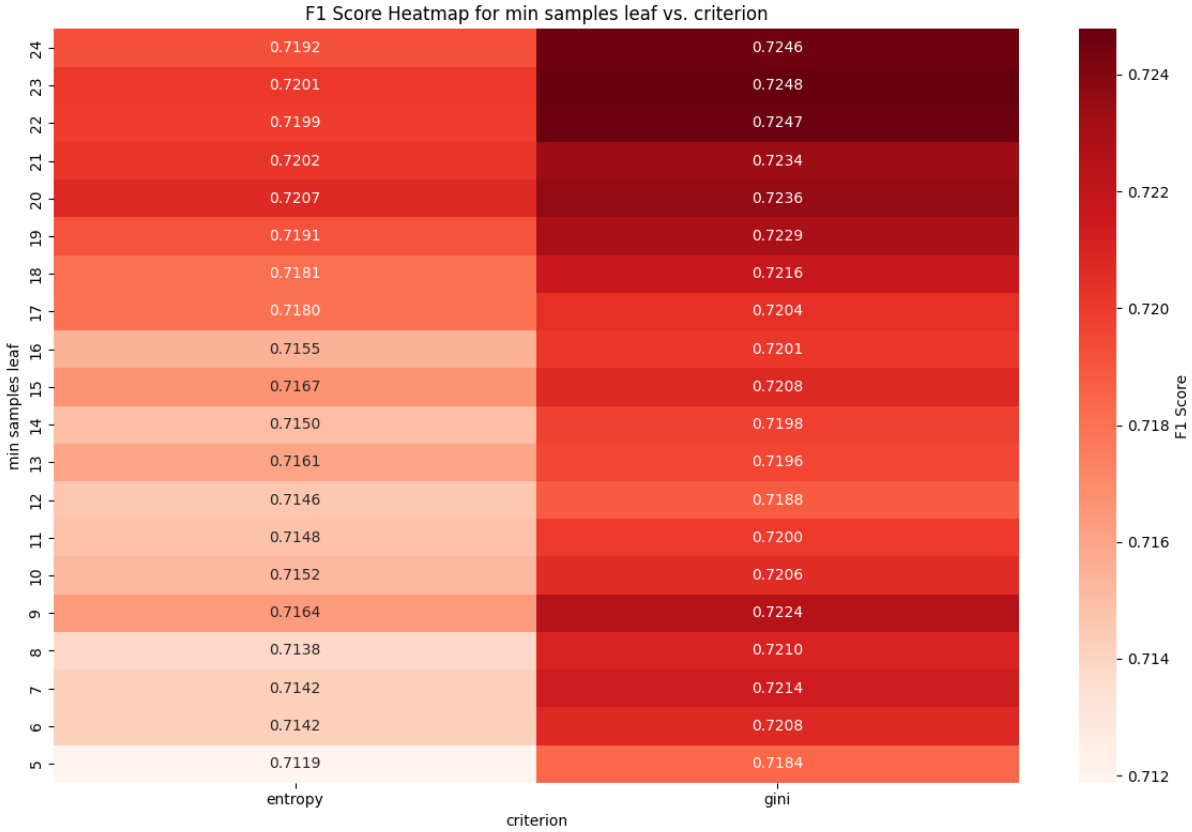
נספח 1.3 - מודלים עם שינוי במספר פרמטרים

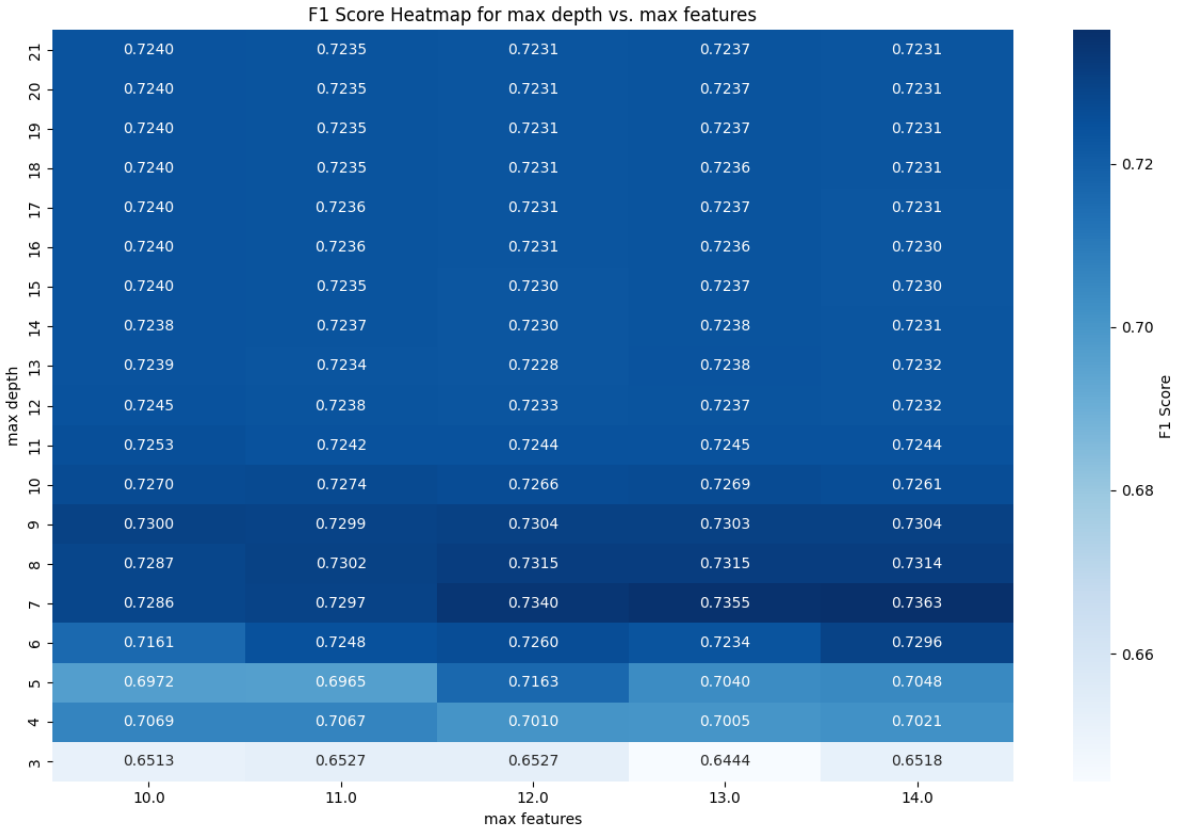


נספח 1.4 - ערכי הפרמטרים לאחר כיוונון

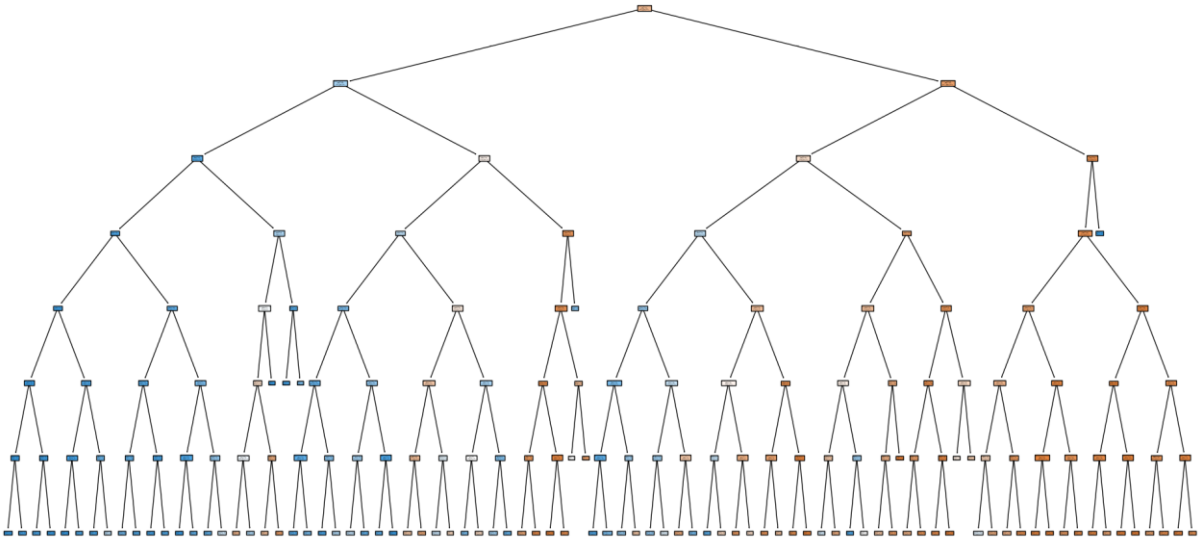
```
Best parameters: {'ccp_alpha': '0.00000', 'criterion': 'gini', 'max_depth': np.int64(7), 'max_features': 14, 'min_samples_leaf': np.int64(9)}
average F1 score obtained from the cross-validation: 0.7460
F1 Score of the best model found on the entire train set: 0.8506
F1 Score test set: 0.8171
```

נספח 1.5 - מפות חום לערכי הפרמטרים


F1 Score Heatmap for min samples leaf vs. criterion
min samples leaf
criterion
entropy
gini
24 0.7192 0.7246
23 0.7201 0.7248
22 0.7199 0.7247
21 0.7202 0.7234
20 0.7207 0.7236
19 0.7191 0.7229
18 0.7181 0.7216
17 0.7180 0.7204
16 0.7155 0.7201
15 0.7167 0.7208
14 0.7150 0.7198
13 0.7161 0.7196
12 0.7146 0.7188
11 0.7148 0.7200
10 0.7152 0.7206
9 0.7164 0.7224
8 0.7138 0.7210
7 0.7142 0.7214
6 0.7142 0.7208
5 0.7119 0.7184
F1 Score
0.724
0.722
0.720
0.718
0.716
0.714
0.712


F1 Score Heatmap for max depth vs. max features
max depth
max features
10.0 11.0 12.0 13.0 14.0
21 0.7240 0.7235 0.7231 0.7237 0.7231
20 0.7240 0.7235 0.7231 0.7237 0.7231
19 0.7240 0.7235 0.7231 0.7237 0.7231
18 0.7240 0.7235 0.7231 0.7236 0.7231
17 0.7240 0.7236 0.7231 0.7237 0.7231
16 0.7240 0.7236 0.7231 0.7236 0.7230
15 0.7240 0.7235 0.7230 0.7237 0.7230
14 0.7238 0.7237 0.7230 0.7238 0.7231
13 0.7239 0.7234 0.7228 0.7238 0.7232
12 0.7245 0.7238 0.7233 0.7237 0.7232
11 0.7253 0.7242 0.7244 0.7245 0.7244
10 0.7270 0.7274 0.7266 0.7269 0.7261
9 0.7300 0.7299 0.7304 0.7303 0.7304
8 0.7287 0.7302 0.7315 0.7315 0.7314
7 0.7286 0.7297 0.7340 0.7355 0.7363
6 0.7161 0.7248 0.7260 0.7234 0.7296
5 0.6972 0.6965 0.7163 0.7040 0.7048
4 0.7069 0.7067 0.7010 0.7005 0.7021
3 0.6513 0.6527 0.6527 0.6444 0.6518
F1 Score
0.72
0.70
0.68
0.66

נספח 1.6 - העץ המלא לאחר כוונון הפרמטרים



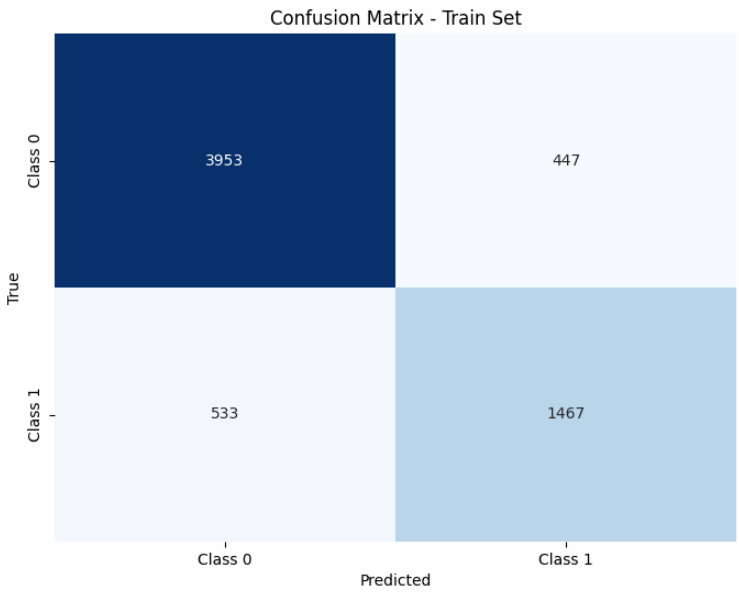
נספח 2 - רשת נוירונים

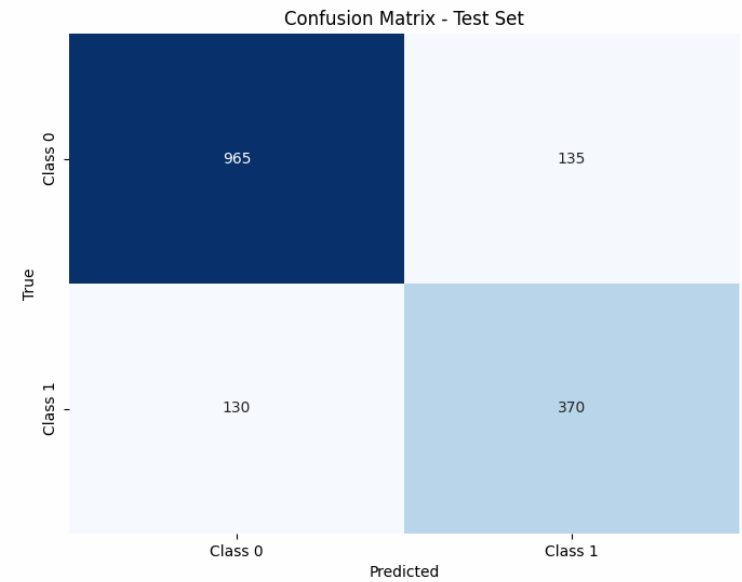
נספח 2.1 - ערכי המדדים לסט האימון והבחינה ברשת ברירת המחדל וברשת עם ערכי הפרמטרים בקונפיגורציה הנבחרת

```
F1 Score train set: 0.8793
Train Accuracy: 88.00%
F1 Score test set: 0.8287
Test Accuracy: 82.88%

Best NN F1 score and Accuracy after Hyperparameter-tuning:
F1 Score train set: 0.8448
Train Accuracy: 84.58%
F1 Score test set: 0.8349
Test Accuracy: 83.44%
```

נספח 2.2 - מטריצות מבוכה





נספח 3 - K-Means

נספח 3.1 - ערכי המדדים של k-means

```
K-means F1 score and Accuracy:
F1 Score train set: 0.7638
Train Accuracy: 75.59%
F1 Score test set: 0.7637
Test Accuracy 75.56%
```

נספח 3.2 - ערכי המדדים ל-k-means עם מספר מחלקות

n_clusters	F1 Score (Train)	Accuracy (Train)	F1 Score (Test)	Accuracy (Test)
4	0.7858	79.09%	0.7832	78.75%
20	0.7917	79.83%	0.7810	78.62%
15	0.7818	78.92%	0.7777	79.31%
8	0.7740	78.19%	0.7733	78.06%
12	0.7709	77.88%	0.7714	77.31%
6	0.7690	78.48%	0.7668	78.06%
2	0.7616	75.36%	0.7613	75.31%
10	0.7593	76.50%	0.7559	76.69%

נספח 4 - XGB

נספח 4.1 - ביצועי המודל

```
defult XGB F1 score and Accuracy:
F1 Score train set: 0.8590
Train Accuracy: 91.20%
F1 Score test set: 0.7087
Test Accuracy: 81.25%

Best XGB F1 score and Accuracy after Hyperparameter-tuning:
F1 Score train set: 0.8253
Train Accuracy: 87.91%
F1 Score test set: 0.7254
Test Accuracy: 80.50%
```