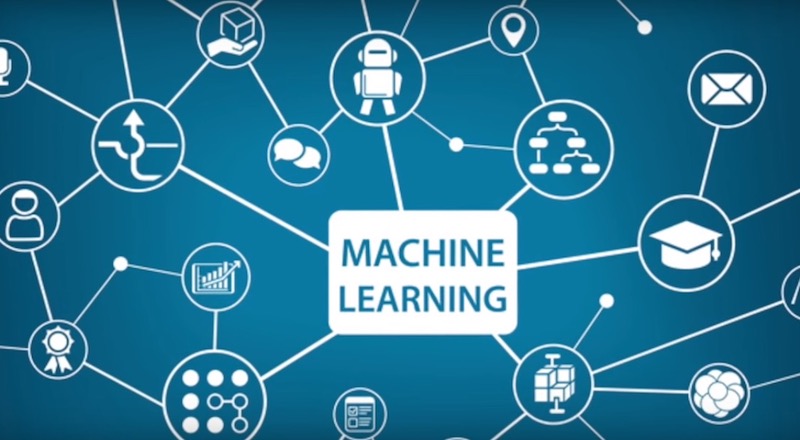


**תוכן עניינים**

**פרויקט קורס:**

חיזוי כמות מכירות באירופה באמצעות לימוד מכונה

**חלק ב'**

המחלקה להנדסת תעשיה וניהול

**קורס לימוד מכונה (ML)**

ד"ר ניר ניסים , דור זזון

שקד אברהמי – 203436670

יונתן שטרנברג - 307966192

[Model Training – המשך (5 נק') 2](#_Toc60940986)

[Decision Trees (25 נק') 2](#_Toc60940987)

[2.1 בניית עץ החלטה 2](#_Toc60940988)

[2.2 Hyperparameter Tuning 2](#_Toc60940989)

[2.3 אימון עץ ההחלטה 4](#_Toc60940990)

[Artificial Neural Networks (25 נק') 5](#_Toc60940991)

[3.1 בניית רשת נוירונים 5](#_Toc60940992)

[3.2 כוונון פרמטרים 6](#_Toc60940993)

[אמנו רשת באמצעות הקונפיגורציה הנבחרת מסעיף קודם. (7 נק') 7](#_Toc60940994)

[SVM (10 נק') 7](#_Toc60940995)

[Unsupervised Learning - Clustering (15 נק') 9](#_Toc60940996)

[השוואה בין המודלים - Evaluation (5 נק') 12](#_Toc60940997)

[שיפור המודל הנבחר – Improvements (10 נק') 12](#_Toc60940998)

[נספחים 14](#_Toc60940999)

[נספח 1 – חלוקה ראשונית לסט אימון וסט ואלידציה 14](#_Toc60941000)

[נספח 2 – מדגם מייצג של K-FOLD (חלוקה לסט אימון וסט ואלידציה) 14](#_Toc60941001)

[נספח 3 – ערך אלפא לקטימה 14](#_Toc60941002)

[נספח 4 : הצגת טבלה המציגה את ערכי ההיפר פרמטר כפונקציה של אחוז הדיוק על סט האימון וסט הואלידציה (Cross validation for DT Model): 15](#_Toc60941003)

[נספח 5 – הערכת פיצ'רים במודל SVM 15](#_Toc60941004)

[נספח 6 : Cross validation for ANN Model 15](#_Toc60941005)

[נספח 7 – הסבר אודות thresholds ומטריקת הAUC 16](#_Toc60941006)

[נספח 8-שלוש רמות ראשונות לעץ ההחלטה עם הקונפיגורציה המיטבית 16](#_Toc60941007)

[נספח 9 – פלט Adaboost 17](#_Toc60941008)

[נספח 10 - פלט GradientBoostingClassifier 17](#_Toc60941009)

[נספח 11 – פלט HistGradientBoosting 17](#_Toc60941010)

[נספח 12 – דוגמה למספר שורות מהדאטה: 17](#_Toc60941011)

[מקורות 18](#_Toc60941012)

[מקור 1- גישת הElbow method 18](#_Toc60941013)

[מקור 2- Davies Bouldin Index based hierarchical initialization K-means 18](#_Toc60941014)

[מקור 3- Boosting the accuracy of your Machine Learning models | by Prashant Gupta | Towards Data Science 18](#_Toc60941015)

[מקור 4- GradientBoostingClassifier (Friedman 2001) 18](#_Toc60941016)

# Model Training – המשך (5 נק')

משתנה המטרה שלנו הינו רציף ולכן סוג משימת הלימוד עבור ערכים רציפים הינה:

Supervised Learning – Prediction. בנוסף, ביצענו המרה של משתנה המטרה למשתנה בינארי ע"י שימוש ב KBinsDiscretizer. כמובן שלאחר המרת משתנה המטרה למשתנה בינארי המשימה כעת הינה משימת Supervised Learning – classification.

([ראה דוגמה למספר שורות מהדאטה בנספח 12](#_נספח_12_–)).

# Decision Trees (25 נק')

## 2.1 בניית עץ החלטה

תחילה, חילקנו את הדאטה באופן הבא:

* 80% מהדאטה ישמש אותנו לסט האימון (שאותו נחלק לסט אימון וסט ואלידציה).
* 20% מהדאטה ישמש אותנו לסט הבחינה (Test set). סט נתונים שהמודל לא התאמן עליו ויפסק לנו תוצאות חיזוי מהימנות ויכולת ניתוח (קטימה והתאמת מודל) טובים יותר. כלומר, הגדרנו סט נתונים בלתי תלוי לסט שבו אימנו את המודל ומצאנו את הקונפיגורציות הטובות ביותר ולבסוף להשוות בין המודלים השונים.

כעת, נבנה עץ החלטה מלא ולהלן אחוזי הדיוק המתקבלים:

|  |  |
| --- | --- |
| Set |  |
| 100% | Training Set |
| 89% | validation Set |

כפי שצפינו, כאשר לא הגבלנו את העץ כלל, הגענו לדיוק מירבי על סט האימון בגלל Overfitting, כלומר העץ "למד" את כל התצפיות בצורה מפורטת. בהתאמה לכך, ניתן לראות דיוק נמוך יותר על סט הבחינה, כלומר עבור תצפיות חדשות המודל לא חוזה בצורה פחות טובה. ([**ראה ממצאים בנספח 1**](#_נספח_1_–)).

## 2.2 Hyperparameter Tuning

הפרמטרים אותם נרצה לכוון הינם:

|  |  |  |  |
| --- | --- | --- | --- |
| שם הפרמטר | משמעות ומוטיבציה | ערכים לבדיקה | מס' ערכים |
| max\_depth | עומק עץ המקסימלי. ככל שעומק העץ גדול יותר, העץ יהיה יותר מסועף וידע לסווג תצפיות יותר בפירוט. אך, כפי שציינו, עומק רב מדי עלול להוביל ל-Overfit. **פרמטר זה חשוב לאיזון בין Generalization Capabilityו-Overfitting של המודל.** ([ראה ממצאים בנספח 2](#_נספח_2_–)) | (1-25) | 25 |
| criterion | סוג המדד בכדי למדוד את איכות הפיצול. **נרצה לבחון האם קריטריונים שונים תוצאות ועצים שונים מסט הנתונים.** | ['gini', 'entropy'] | 2 |
| max\_features | מספר המאפיינים המקסימלי בבחירת הפיצול האופטימלי. נרצה לבדוק האם נקבל תוצאות דיוק טובות יותר לאחר כיוונון של פרמטר זה. | ['auto', 'sqrt',  'log2', None] | 4 |

Gini- המטרה היא למדוד באיזו תדירות אלמנט שנבחר באופן אקראי מהסט יתויג בצורה שגויה (ויקיפדיה)

Entropy- מדד האנטרופיה הוא מדד יחסי, לכן אנחנו נרצה לבחון איזה פיצ'ר מוביל להפחתה הגבוה ביותר באנטרופיה. כלומר כמה אינפורמציה התווספה לי לנתונים לאחר הוספה של פיצ'ר מסוים לעץ ההחלטה.

Auto- max\_features=sqrt(n\_features)

Sqrt- max\_features=sqrt(n\_features)

Log2- max\_features=log2(n\_features)

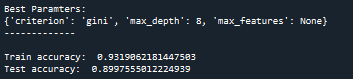
None- max\_features=n\_features

כדי להציג את השינוי בדיוק על סט הואלידציה של כל אחד מהפרמטרים, קיבענו שניים מהפרמטרים הנבחרים ולהציג את הדיוק כפונקציה של הפרמטר השלישי. לאחר מכן, יש לשנות את אחד מאותם שני פרמטרים, ונבצע שוב קיבוע של שני פרמטרים ונציג את הדיוק כפונקציה של הפרמטר השלישי. ([ראה ממצאים נספח 4).](#_נספח_4_:)

בכדי להציג את ערכי הפרמטרים שבחנו, הרצנו פקודת GridSearchCV אשר הניבה את הקונפיגורציה האופטימלית עבור סט האימון וסט הואלידציה אותם היא סרקה. הגדרנו פונקציה ל-Cross-Validation : הפונקציה עושה סריקה עבור כל קומבינציה אפשרית של הפרמטרים הנבדקים, חלוקת סט האימון בעזרת K-Fold ( k=10 ), בדיקת הדיוק הממוצע וסטיית התקן הממוצעת על סט האימון וסט האימות. כשבסוף סידור בסדר יורד לפי הדיוק על סט הואלידציה. העץ שנבחר ע"י אלגוריתם GridSearchCV (צבוע אדום בנספח 4) מקיים:

* דיוק ממוצע גבוה על סט האימון ועל סט האימות
* סטיית תקן נמוכה על סט האימון וסט האימות

סה"כ נבצע סה"כ 200 קומבינציות

נבצע זאת ע"י פקודת GridSearchCV , פקודה אשר מחלקת את סט הנתונים לסט אימון וסט ואלידציה, זאת ע"י (בעזרת K-Fold Cross Validation, k=10). הפקודה תסרוק את כלל הקומבינציות האפשריות בפרמטרים הנתונים, ותציג את הקומבצינציה שהניבה את הדיוק הממוצע הגבוה ביותר. להלן התוצאות

|  |  |
| --- | --- |
|  | Set |
| criterion | gini |
| max\_depth | 8 |
| max\_features | none |
| Train Accuracy | 93% |
| Test Accuracy | 90% |

ממצאי אלגוריתם GridSearchCV הניב את הקונפיגורציה האופטימלית מבחינת דיוק ממוצע וסטיית תקן ממוצעת על סט האימון וסט האימות (ראה נספח 2). יצוין כי כדי להציג את השינוי בסט הואלידציה קיבענו כל פעם שני פרטמטרים תוך הצגת הדיוק כפונקציה של הפרמטר השלישי.

## 2.3 אימון עץ ההחלטה

נשתמש בקונפיגורציות האופטימליות שמצאנו בסעיף הקודם:

|  |  |
| --- | --- |
| Set |  |
| 93% | Training Set |
| 90% | Test Set |

כפי שצפינו ניתן לראות עלייה בדיוק של סט הבחינה (Test set). כיוונון הפרמטרים הפחית את התאמת העץ לסט האימון והגביר את יכולת ההסתגלות (Generalization) של המודל לדוגמאות חדשות. כעת המודל יודע לחזות טוב יותר עבור נתונים חדשים. בנוסף ציון הדיוק עבור סט האימון אינו 100 אחוז מאחר והפחתנו את עומק העץ והפכו אותו לבעל יכולת הסתגלות טובה יותר לנתונים חדשים.

ביצענו Post Pruning , ביצענו זאת ע"י פונקציית cost\_complexity\_pruning\_path([ראה נספח 3](#_נספח_3_–)).

ניתן לראות כי ערכי האלפא אשר מעלים את אחוז הדיוק בסט הואלידציה מפחיתים משמעותית את אחוז הדיוק בסט האימון, וקוטמים את העץ יותר מדי, דבר אשר יוביל לפישוט העץ. לכן, נשאר עם ערך אלפא=0.

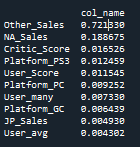
([ראה גרף את הגרף שך העץ שהתקבל לאחר הכוונן בנספח 8](#_נספח_8-שלוש_רמות)).

אחד היתרונות של מודל עץ החלטה הוא יכולת ההסברה של המודל, ניתן להסביר זאת על ידי כך שבסופו של דבר אנחנו יודעים להגיד מהם כללי ההחלטה בכל שלב בעץ, לדוגמה (הכנסה גדולה ממיליון אז פיצול ימינה או שמאלה), וכך קל יותר להבין איך המודל מסווג כל דגימה ואפשר לקבל החלטות בצורה מושכלת יותר על כיוונון הפרמטרים והבנת המודל בכלל. לעומת זאת ברשתות נוירונים למשל, אנו מקבלים משקולות סופיות אבל אין להן משמעות שניתן לפרש אותה.

מהסתכלות על מבנה העץ שקיבלנו, ניתן לראות כי המאפיינים המשמעותיים ביותר הינם Other\_Sales (כמות המכירות בשאר העולם), NA\_Sales (כמות המכירות בצפון אמריקה), Critic\_Score (דירוג המשחק ע"י מבקרים).

כלומר, כאשר משחק עם כמות מכירות גבוה בשאר העולם ובצפון אמריקה, יחד עם ציון גבוה ע"י מבקרי המשחק נוכל לסווג אותו כמשחק עם כמות מכירות גבוהה באירופיה (NA\_Sales).

כעת נדרג את חשיבות הFeatures ע"י שימוש בפונקציית feature\_importances. דירוג הFeatuers בפונקציה זו מתבסס על כמות המידע שהם תורמים לעץ ההחלטה. כלומר, ככל שהתרומה יותר גבוהה מדד האנטרופיה:  *יורד ולמעשה* הFeature *חשוב ותורם יותר למודל.*

*התוצאות אכן מתיישבות עם המסקנות מהסעיף הקודם:*

יצוין כי, את מאפיינים Other\_Sales , NA\_Sales , Critic\_Score ו User\_Score סימנו עוד בשלב א' כמאפיינים בעלי השפעה גבוהה על משתנה המטרה. צפינו זאת עפ"י תרשימי הפיזור, מתאם פירסון ולבסוף בהרצת הרגרסיה. עוד יצוין כי, בשלב א בפרויקט משתנים User\_many ו- User\_avg הם משתני אינטרקציה שהוספו על ידינו, לאחר הבנה כי יהיה נכון יותר להסתכל על כמות הדירוגים וציון הדירוג במשתנה אחד, ולכן משמח לראות כי אכן משתנים אלו מקבלים ציון די גבוה בהשפעה על המודל.

|  |  |
| --- | --- |
|  | Feature importance |
| Other\_Sales | 0.721 |
| NA\_Sales | 0.1886 |
| Critic\_Score | 0.0165 |
| Platform\_PS3 | 0.012 |
| User\_Score | 0.011 |
| Platform\_PC | 0.009 |
| User\_many | 0.007 |
| Platform\_GC | 0.006 |
| JP\_Sales | 0.0049 |
| User\_avg | 0.0043 |

# Artificial Neural Networks (25 נק')

## 3.1 בניית רשת נוירונים

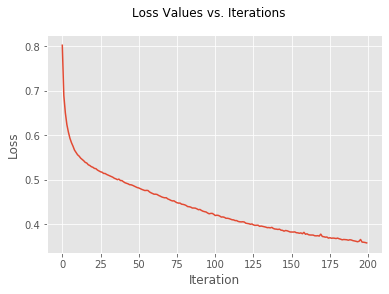
תחילה ביצענו נרמול לעמודות המאפיינים הרציפים

['NA\_Sales', 'JP\_Sales', 'Other\_Sales','Critic\_Score','User\_Score','User\_many','User\_avg']

בעזרת חבילת StandardScalar. בנוסף, לכל חלוקה שמרנו את הממוצע וסטיית התקן של מאפיינים אלו.

נבנה רשת נוירונים מלאה עם ערכי ברירת מחדל באמצעות שלושת הסטים לאימון. אחוזי הדיוק המתקבלים:

|  |  |
| --- | --- |
| Set |  |
| 87.42% | Training Set |
| 82.1% | Test Set |



ניתן לראות כי אחוזי הדיוק יחסית גבוהים גם על סט האימון וגם על סט הבחינה. ניתן להניח שככל הנראה אין Overfitting (אחרת היינו רואים אחוזי דיוק נמוכים יותר על סט הבחינה וגבוהים יותר על סט הבדיקה).האלגוריתם ציין כי הגיע למספר האיטרציות המקסימלי (ברירת המחדל – 200) אך עדין לא הגיע להתכנסות, ולכן נצטרך להגדיל את פרמטר זה בהמשך. משמאל ניתן לראות את עקומות הלמידה כפונקציה של מספר האיטרציות. ניתן לראות קצב למידה מעריכי, כאשר עד 150 איטרציות לערך מתבצעת למידה רבה, לאחר מכן הלמידה קטנה, ולבסוף נעצרת ב-200 איטרציות לפי ערך ברירת המחדל.

הקונפיגורציה שנלמדה ע"י המודל:

מספר נוירונים בשכבת הכניסה – מספר המאפיינים (עמודות) בסט הנתונים (28)

מספר שכבות חבויות – לפי ערך ברירת המחדל (1)

מספר נוירונים חבויים בשכבה – לפי ערך ברירת המחדל (100)

## 3.2 כוונון פרמטרים

|  |  |  |  |
| --- | --- | --- | --- |
| שם הפרמטר | משמעות ומטויבציה | ערכים לבדיקה | מס' ערכים |
| hidden\_layer\_sizes | גדלי השכבות החבויות – מספר הנוירונים בכל שכבה חבויה. ככל שנגדיל את מספר הנוירונים, זמן הלמידה של המודל יהיה ארוך יותר, שכן כל נוירון מהווה "יחידת עיבוד".מצד שני ככל הנראה ישנו סיכוי שהדיוק יהיה גבוה יותר אבחר לבדוק:  שכבה אחת: עם מספר נוירונים 1-28  שתי שכבות נוירונים: ראשונה עם 28 ושכבה שניה בין 1 ל-22 נוירונים (כדי ליצור צורת "משפך"). | (1-28),  (28, 1-28) | 56 |
| activation | פונקציית אקטיבציה. נרצה לבדוק האם פונקציות אקטיבציה שונות מניבות תוצאות שונות עבור סט הנתונים ומגדילות את הדיוק של המודל על סט הבחינה. | ['identity, 'logistic', 'tanh’, ‘relu’] | 4 |
| learning\_rate\_init | קצב הלמידה ההתחלתי. מגדיר באיזו מהירות הרשת מעדכנת את המשקולות שלה בכל איטרציה. ככל שקצב הלמידה ברשת גדול יותר, הצעדים בכיוון המינימום יהיו גדולים יותר ולהיפך. הסכנה היא בפספוס ערכי ביניים אופטימליים. מצד שני בדיקה של ערכים קטנים מידי יכולה להוות "בזבוז זמן/לטחון מים" – אם עוד איטרציה כבר כמעט ולא משפרת את דיוק המודל- נעצור ולא נבדוק יותר. | [0.1, 0.01,  0.001] | 3 |
| alpha | מרסן / גורם ענישה לגודל המשקולות. מונע מהמשקולות לקבל ערך גבוה מידי, מה שגורם לפונקציה להתאים את עצמה יותר מידי לנתונים (Overfit). | [0.0001] | 1 |
| max\_iter | מספר איטרציות מקסימלי. ככל שנגדיל את מספר האיטרציות, זמן הריצה ללמידה של כל מודל יגדל, ובמספר גדול מידי יש סכנה להתאמת יתר (Overfit). מכיוון שבהרצות עם ערכי ברירת המחדל 200 הרצות לא הספיקו, אבחר לבדוק גם את פרמטר early\_stopping כדי למנוע Overfit, פרמטר זה מייתר מאיתנו את ההחלטה על מספר האיטרציות. כאשר arly\_stoppingיהיה False נבדוק את הערכים הנבחרים | [300,350] | 2 |
| early\_stopping | מנגנון עצירה במידה והדיוק על סט האימות לא עולה למשך מספר איטרציות. אגדיר כ-True כדי לחסוך זמן ריצה מיותר, לטובת בדיקת אפשרויות רבות. (בדקתי גם ערך של False: זמני הריצה גדול משמעותית, אך השיפור בדיוקים היה זניח).  נבחר סט קטן עליו נבצע את הבדיקה. המדד עליו מתבצעת הבדיקה הוא activation. | [True,False] | 2 |

סה"כ מספר הקומבינציות לבדיקה הוא :1344+672=2016

במקרה של רשת נוירונים, זמן הריצה הדרוש ללמידה של כל מודל בנפרד הינו ארוך, ובדיקת כל הקומבינציות האפשרויות לוקח זמן רב. לכן, כדי למצוא את הקונפיגורציה האופטימלית, במקרה של שכבה אחת אשתמש בפקודת GridSearchCV, שכן מספר הקומבינציות קטן, ובמקרה של שתי שכבות, אשתמש בפקודת RandomSearchCV אשר מחלקת את סט הנתונים לסט אימון וסט ולידציה (בעזרת K-Fold Cross Validation, k=10), תסרוק קומבינציות באופן אקראי (אגדיר 1,000 קומבינציות), ותבחר את המודל עם הפרמטרים שהניבו את הדיוק הממוצע הגבוה ביותר על סט האימון. התוצאות שהתקבלו:

|  |  |  |
| --- | --- | --- |
|  | 1 Layer | 2 Layers |
| hidden\_layer\_sizes | (6) | (9,9) |
| activation | tanh | identity |
| learning\_rate\_init | 0.1 | 0.1 |
| Train Accuracy | **89.30%** | **90.53%** |
| Test Accuracy | **82.41%** | **81.39%** |

לאחר מספר הרצות על סט נתונים שונה התקבלו קונפיגורציות ייחודיות, כאשר פרמטר learning\_rate\_init הוא היחיד שכמעט אחיד בין כולן – 0.1. הדיוקים על סט האימון אינם גבוהים, במקרים מסויימים הדיוק על סט האימון קרוב מאוד לדיוק על סט הבחינה, למרות שקיווינו לראות דיוק גבוה יותר על סט האימון.

כדי להציג את השינוי בדיוק על סט האימות של כל אחד מהפרמטרים, יש לקבע שניים מהפרמטרים הנבחרים ולהציג את הדיוק כפונקציה של הפרמטר השלישי. לאחר מכן, יש לשנות את אחד מאותם שני פרמטרים, ולבצע זאת שוב. מהממצאים ([נספח 6](#_נספח_6_:)).

לכן החלטנו להסתמך על הבדיקה הידנית ולבחור את המודל המשוקלל עבור דיוק וסטיית תקן גם עבור סט הבחינה וגם עבור הוולידציה הטוב ביותר

הקונפיגורציה הנבחרת היא:

מספר נוירונים בשכבת הכניסה – 28

מספר שכבות חבויות – 1

מספר נוירונים חבויים בשכבה – 12

קצב למידה של : 0.01

early\_stopping=True ולכן לא נקבע את מספר האיטרציות.

פונקציית activation : tanh

אחוזי הדיוק המתקבלים בקונפיגורציה הטובה ביותר הם:

סט אימון:

דיוק: 89.4% סטיית תקן: 0.9%

סט וולידציה:

דיוק:87.2% וסטיית תקן: 1.1%

ניתן לראות כי מבחינת סט האימון לא חל שינוי משמעותי, המודל לא נמצא בoverfitting ותוצאותיו הם יחסית טובות.

לעומת זאת בסט הוולידציה חל שיפור ניכר כאשר תהליך ה Hyperparameter Tuning שיפר את דיוקו של המודל במעל 5%.

ניתן להסיק כי פונקציית האקטיבציה ומספר הנוירונים ברשת החבויה במודל הדיפולטיבי שאומן בסעיף 1 לא התאימו לסט הנתונים ברלוונטי.

## אמנו רשת באמצעות הקונפיגורציה הנבחרת מסעיף קודם. (7 נק')

לאחר הפעלת פונקציית **predict\_proba** על הרשת הנבחרת:

ה sampels שהמודל היה הכי פחות בטוח לגביהם אך דורגו כ 1 הם: 2153 = 50.07% , 3888 = 50.003%

ה sampels שהמודל היה הכי פחות בטוח לגביהם אך דורגו כ 0 הם: 1515=49.995%, 49.994%=1158

# SVM (10 נק')

תחילה , גם כאן חילקנו את סט הנתונים לסט אימון (יתחלק לאימון וואלידציה) וסט מבחן (סט בלתי תלוי).

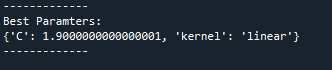
בחנו את שני סטים אלה למול פונקציית קרנל לינארית עם ערך C=1. להלן התוצאות:

|  |  |
| --- | --- |
| Set |  |
| 76% | Training Set |
| 73% | Validation Set |

לאחר מכן, ביצענו Hyperparameter Tuning והרצנו פקדות grid\_search למציאת הקונפיגורציה הטובה ביותר. הרצנו מספר פונקציות קרנל – נסביר מהי פונקציית קרנל:

זוהי פונקציה אשר לוקחת כקלט וקטורים (sampels) מהמימד המקורי ומחזירה את הוקטורים במימד נמוך יותר, פונקציית קרנל בנוסף מאפשרת לחשב את תוצאת מכפלת ההמרה מבלי להמיר וזה חוסך זמן חישוב יקר.

פונקציות קרנל שהרצנו –( 'linear', 'poly','rbf','sigmoid'). בנוסף בדקנו מספר ערכי C (ערך C מציין כמה ארצה להימנע מסיווג שגוי של sample).



|  |  |
| --- | --- |
| Set |  |
| 77% | Training Set |
| 75% | Validation Set |

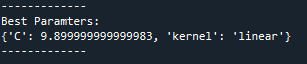
הקונפיגורציה הטובה ביותר שהתקבלה הינה:

**ערך C-1.9**

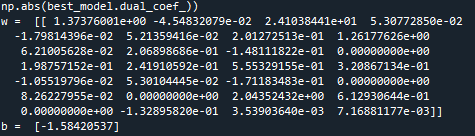
**פונקציית קרנל- Linear**

לאחר מציאת הקונפיגורציה הטובה ביותר, חיפשנו עבור פונקציית הקרנל הלינארית טווח ערכים רווח יותר של הפרמטר C.

הממצאים שהתקבלו הינם:



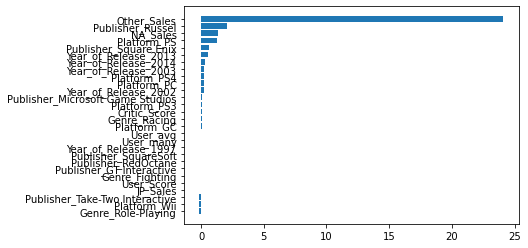
|  |  |
| --- | --- |
| Set |  |
| 80% | Training Set |
| 78% | Validation Set |

להלן משוואת הקו הישר המפריד :

ניתן לראות את המקדמים אשר קיבל כל פיצ'ר ואת ערך החותך (b). בכדי לקבוע את הפיצ'רים התורמים ביותר למודל הSVM הגדרנו פונקצייה ([ראה נספח 5](#_נספח_5_–)) אשר מדרגת את תרומת הפיצ'רים בסדר יורד.

ניתוח הממצאים:

מבחינת המסקנות מסעיפים קודמים ניתן לראות כי בהתאם למסקנות מעץ ההחלטה משתנה Other\_Sales הוא המשתנה אשר תרומתו היא הגבוה ביותר. בנוסף ובדומה למודלים קודמים גם משתנה NA\_sales מספק תרומה גבוה לסיווג המודל.



דירוג הפיצ'רים במודל SVM

יצוין כי, בשונה מעץ ההחלטה פיצ'ר publisher מספק תרומה גבוה לסיווג המודל בSVM בשונה מעץ ההחלטה.

במידה והיינו רוצים לבצע משימת קלסיפיקציה עם 3 מחלקות, נשתמש במתודולוגיה אשר נלמדה בהרצאה "one against all" – פטנט שניר הזכיר בהרצאה. כלומר אנחנו נייצר n SVM שונים כמספר הCLASS שלנו, במקרה זה 3. כל SVM כזה ילמד Class אחד , במקרה שלנו :חיזוי נמוך מול כל שאר המחלקות שהם לא חיזוי נמוך של מכירות. כך למעשה הפכנו את בעיית הסיווג לבעיה סיווג בינארית. לאחר מכן נעשה באופן דומה SVM שיפריד בין חיזוי בינוני לכל שאר המחלקות וכן הלאה. כך ירצנו "one against all": n SVM מהסוג הזה.

כאשר נקבל sample חדש, כל מה שנצטרך לעשות הוא לחשב עבור כל אחד מהSVM שיצרנו את המרחק שלו לצד החיובי מאחר והצד החיובי הוגדר כ או חיזוי נמוך או חיזוי בינוני או חיזוי גבוה. אנחנו נחשב עבור כל SVM את המרחק ונקח את המרחק החיובי הגבוה ביותר. למעשה לקחנו את הclassifier עם הביטחון הגבוה ביותר.

# Unsupervised Learning - Clustering (15 נק')

הסט הנתונים עליו הרצנו את המודל הינו הdataset המוגמר לאחר חלק א' ללא עמודת משתנה המטרה (ערכי Y). במשימות Unsupervised Learning באופן כללי ובאלגוריתם K-means בפרט, איננו מבצעים משימות prediction/classification אלא משימות אשכול, אנו נחתור למציאת הדימיון בין הsamples לפי הפיצ'רים שלהם ולא לפי הLabel שלהם.

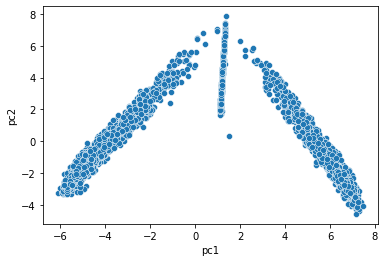
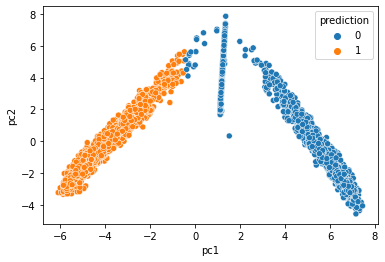
בנוסף, בניגוד supervised Learning שיש לנו את האמת האבסולוטית להערכת ביצועי המודל, לניתוח אשכולות אין מדד הערכה מוצק בו נוכל להשתמש כדי להעריך את התוצאות של אלגוריתמי אשכולות שונים. יתר על כן, מכיוון ש K-means דורש k כקלט ולא לומד אותו מנתונים, אין תשובה נכונה מבחינת מספר האשכולות שאנו צריכים לקבל בכל בעיה. כפי שנלמד בהרצאה ערכי K שונים כערך התחלתי יובילו לפתרון סופי שונה, ולכן בחירת ערך K באופן לא מושכל עלולה להוביל לפתרון בעייתי.

בכדי לנתח את הנתונים באופן ויזואלי ולקבל מסקנות על אופן החלקה ביצענו הורדת מימד (PCA) על הפיצ'רים.

לאחר מכן אימנו את המודל על סמך הדאטה המקורי (טרם שלב הPCA).

מאחר ואנו במשימת Unsupervised Learning נבחן את הדאטה בשני אופנים: האחד ללא אשכול של המודל תוך שאיפה לאפיין מגמה מסוימת מהסתכלות על הנתונים בלבד, והשני שיוך הפרדיקציות של המודל במטרה לקבל אינדקיציה איזה ערך K יתאים לנתונים שלנו .

בהמשך, ביצענו אשכול לפי k=2 ובכדי לבחון את איכות אשכול המודל הוספנו לדאטה את עמודת השיוכים (אשכולים).

להלן התוצאות:

ניתן לראות כי, מהסתכלות ראשונה עבור K=2 אנו מקבלים הפרדה טובה של הנתונים.

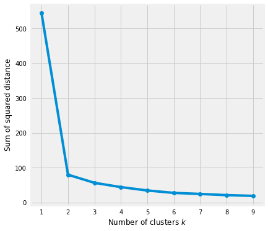
נבחן את שני הגרפים:

נזכיר כי האלגוריתם מחלק את התצפיות לקבוצות בעלות מספור משלו, ללא קשר למספרי המחלקות בסט הנתונים המקורי.

לכן, מהגרף השמאלי ניתן לזהות שני מגמות ברורות ערכי PC1 גבוהים מ1 וערכי PC1 נמוכים מ0, כישנם מקבץ נתונים סביב ערך PC1=0 שככה"נ נשייך בהמשך לאשכול נוסף.

מכיוון ואיננו רוצים שהיוריסטיקה שלנו לערכו של K תהיה על סמך התבוננות בלבד, החלטנו לחקור וליישם את

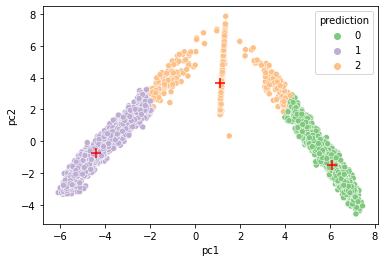
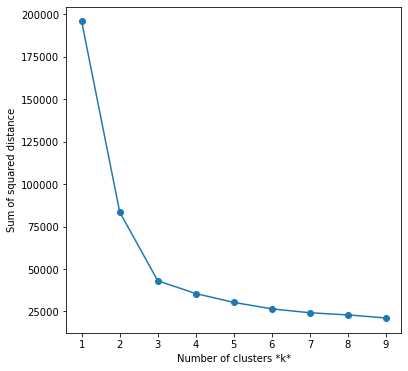
היוריסטיקה Elbow Method ([ראה מקור 1](#_מקור_1-_גישת)).

**הסבר היוריסטיקה:** שיטת המרפק נותנת לנו מושג מה יהיה מספר k טוב של אשכולות בהתבסס על סכום המרחק בריבוע (SSE) בין נקודות נתונים ומרכזי האשכולות שהוקצו להם. אנו בוחרים ב- k במקום בו SSE מתחיל להתייצב ויוצר מרפק. נשתמש במערך geyser ונעריך SSE לערכים שונים של k ונראה היכן העקומה עשויה ליצור מרפק ולהשתטח.

***EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN***

דוגמת המאמר: הגרף הנ"ל מראה כי k = 2 אינה בחירה רעה. לפעמים עדיין קשה להבין מספר רב של אשכולות לשימוש מכיוון שהעקומה פוחתת בצורה מונוטונית ואולי לא מראה שום מרפק או שיש לה נקודה ברורה שבה העקומה מתחילה להשתטח.

ניישם את היוריסטיקה על הנתונים שלנו, ועל מנת לבחון את איכות האשכול עלינו לחזור לשלב ההמרת עמודת המטרה ולסווגה ל 3 קטגוריות (הרבה, ממוצע , קצת) לפי אחוזונים.

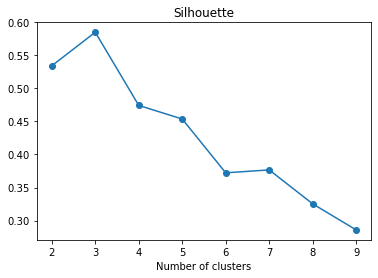


מסתמן כי הK הנבחר הינו 3.

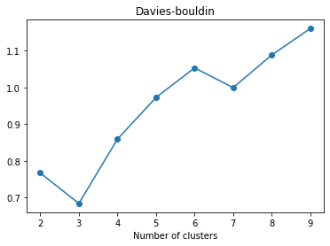
ניתן לראות זאת, הן בהיוריסטיקה והן בהפרדת הנתונים לאחר האשכול של המודל עבור K=3.

בנוסף לשיטת המרפק ובהמשך לנלמד בהרצאה נסתכל על דרכים נוספות למציאת ערך K שיוביל לאיכות אשכול מיטבית.

Silhouette- שיטה בה נותנת למעשה פרשנות על העקביות בתוך האשכולות ומספקת ייצוג גרפי על מידת הסיווג.

ערך השיטה הוא מדד כמה אובייקט מסוים דומה לאשכול בו הוא שויך (לכידות) לעומת אשכולת אחרים (הפרדה). ערך המדד נע בין 1- ל1 כאשר ערך גבוה מצביע על התאמה גבוה בין האובייקט לאשכול בו הוא נמצא ומתואם בצורה נמוכה מאוד לאשכולות אחרים (הפרדה).

**נשים לב שבדומה שיטת המרפק, גם כאן מספר האשכולות המיטבי עבורנו הינו 3 אשכולות.**

Davies–Bouldin index: הוא מדד להערכת ביצועי אשכולות, הרעיון הבסיסי של גישה זו הינו להעריך את ההפרדה בתוך האשכול ובין האשכולות, שבשאיפה יהיה גדול ככל האפשר בין האשכולות וקטן ככל האפשר בתוך האשכולות . מדד זה הוא מדד יחס, כאשר ככל שהערך נמוך יותר, כך הפרדת האשכולות ו"הידוק "בתוך האשכולות טובים יותר ([Junwei Xiao, Jianfeng Lu∗ Xiangyu Li 2017).](#_מקור_2-_Davies)

**נשים לב שבדומה לגישות אחרות גם כאן חלוקה ל-3 אשכולות היא הטובה ביותר.**

|  |  |
| --- | --- |
| Set |  |
| 77.2% | Training Set |
| 74.6% | Validation Set |

עבור 3 אשכולות, בחרנו להתאים את מודל הANN. ביצענו התאמה לשלושה Labels. תחילה ביצענו התאמה ולהלן התוצאות:

מאחר ותוצאות אלו נמוכות משמעותית ממודל הANN עבור שני Labels, החלטתנו לבצע את פקודת GridSearchCV לטובת מציאת הקונפיגורציה האופטימלית.

|  |  |
| --- | --- |
| Set |  |
| 82% | Training Set |
| 79% | Validation Set |

להלן התוצאות לאחר טיוב המודל ע"י Hyperparameter Tuning:

ניתן לראות כי עבור שלושה labels תוצאות המודל נמוכות יותר מאשר שני labels.

# השוואה בין המודלים - Evaluation (5 נק')

ראשית, המטריקות אשר בחרנו הינן: AUC, TPR,FPR. בנוסף ערכי המטריקות נבדקו על סט הבחינה (סט בלתי תלוי לסט האימון, כלומר בתחילת העבודה הגדרנו 20% מהדאטה כסט בחינה, בכדי לקבל תובנות אמינות וטובות יותר לאורך העבודה ובמיוחד בשלבים אלה.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מודל | AUC | TPR | FPR | Best configuration |
| DT | **89.8%** | **91.18%** | **11%** | Max depth : 8 , criterion : 'gini', max features= None |
| ANN | **81%** | **72%** | **7.5%** | activation='tanh', verbose=False , hidden\_layer\_sizes=(12,),  early\_stopping=True,  max\_iter=1000, learning\_rate\_init=0.01 |
| SVM | **78.1%** | **71.8%** | **14.5%** | C=9.89, kernel= 'linear' |

המטרקיה המרכזית שעל פיה נקבל את ההחלטה הינה מטריקת הAUC.

הסיבה לכך הינה שמדדים TPR וFPR מייצגים threshold (סף ההפרדה) קבוע. לעומת זאת מדד הAUC הוא מדד המתכלל את כלל הקומבינציות של ה thresholds ([ראה נספח 7 להסבר על threshold וAUC](#_נספח_7_–) ).

לכן, להבנתנו מטריקת הAUC הינה מטריקה מייצגת יותר להשוואה בין המודלים.

מסקנות: ניתן לראות כי עץ ההחלטה הניב את הציון הגבוה ביותר במטריקת הAUC ויתר על כן גם במדד הTPR. בסוף לעובדה שנמליץ על המודל בעל מדד הAUC הגבוה ביותר נזכיר בנוסף שעץ ההחלטה הניב גם את אחוזי הדיוק (accuracy) הגבוהים ביותר על סט הואלידציה וסט הבחינה (93% על סט הואלידציה ו90% על סט הבחינה). יחד עם זאת בעזרת מודל זה הצלחנו למפות את המאפיינים בעלי החשיבות הגבוה ביותר בסט הנתונים.

# שיפור המודל הנבחר – Improvements (10 נק')

בסטטיסטיקה ולמידת מכונה, **שיטות אנסמבל** משתמשות באלגוריתמי למידה מרובים כדי להשיג ביצועי חיזוי טובים יותר ממה שאפשר היה להשיג מכל אחד מאלגוריתמי הלמידה המרכיבים בלבד. אנסמבל למידת מכונה מורכב ממערכת סופית של מודלים אלטרנטיביים, אך בדרך כלל מאפשר ליצור מבנה גמיש יותר של המודל הסופי.

* Boosting – האנסמבל נבנה באופן הדרגתי על ידי אימון כל תצפית חדשה בכדי להדגיש את מקרי האימון שעברו מודל שגוי בעבר. דוגמה טיפוסית היא AdaBoost. אלה יכולים לשמש לבעיות מסוג רגרסיה וסוג סיווג. עצי החלטה שמשתמשים בשיטה זו נקראים **Boosted Trees**. נציג את מספר דוגמאות שמימשנו בפרויקט:

|  |  |
| --- | --- |
| Set |  |
| 85% | Training Set |
| 83% | Validation Set |

תחילה מימשנו את אלגוריתם AdaBoost ([ראה נספח 9](#_נספח_9_–) ). לאחר מימוש אלגוריתם זה תוצאות החיזוי היו:

ניתן לראות כי לא ניכר שיפור באיכות התחזית ולכן בחרנו לממש מספר גישות שונות.

|  |  |
| --- | --- |
| Set |  |
| 92% | Training Set |
| 91% | Validation Set |

לאחר מכן, בחנו את GradientBoostingClassifier ([ראה נספח 10](#_נספח_10_-) )שהניב שיפור בAcurracy של סט המבחן אך ירידה בסט האימון

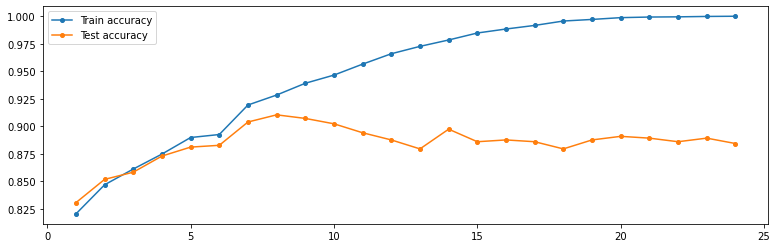
לבסוף בחנו את HistGradientBoosting ([ראה נספח 11](#_נספח_11_–)) שהניב את אחוזי הדיוק הגבוהים ביותר, הן בסט האימון והן בסט המבחן.

|  |  |
| --- | --- |
| Set |  |
| 100% | Training Set |
| 92% | Validation Set |

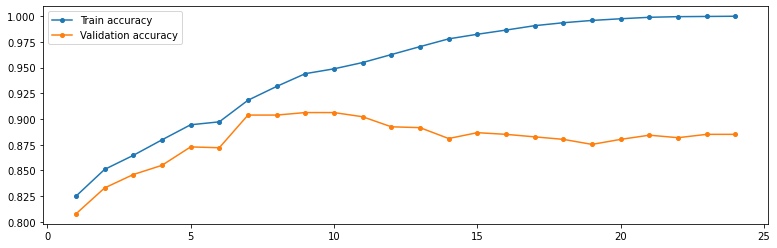
**לסיכום בכדי לטייב את תוצאות המודל הסתמכנו על מספר מאמרים ומקורות שהובילו אותו למימוש הגישות הנ"ל. לאחר מימושן ניתן לראות כי האלגוריתם אשר הניב את תוצאות הניבוי הטובות ביותר הינו HistGradientBoosting . (ראה מקורות 3 , 4).**

# נספחים

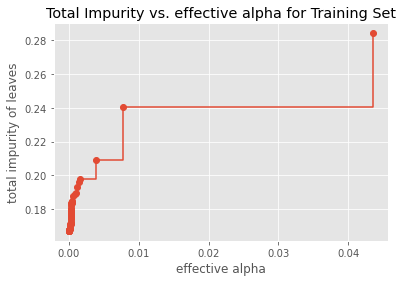
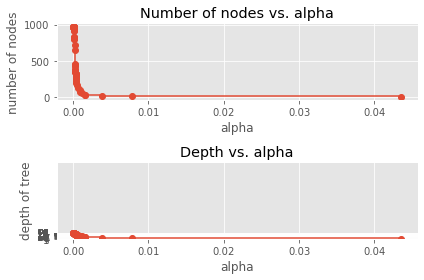
## נספח 1 – חלוקה ראשונית לסט אימון וסט ואלידציה

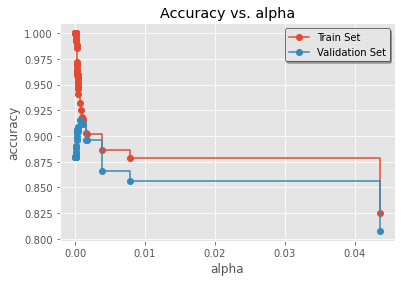


## נספח 2 – מדגם מייצג של K-FOLD (חלוקה לסט אימון וסט ואלידציה)



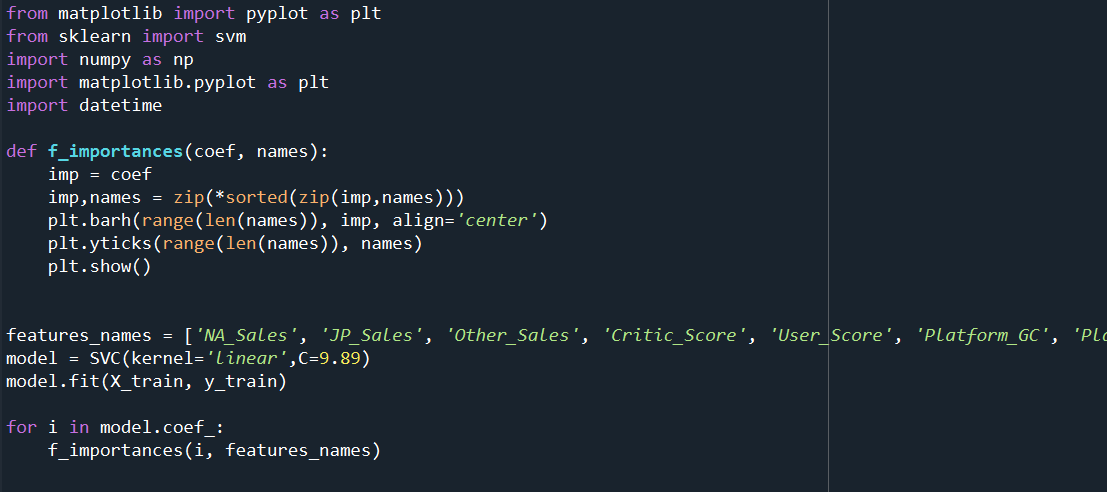
## נספח 3 – ערך אלפא לקטימה





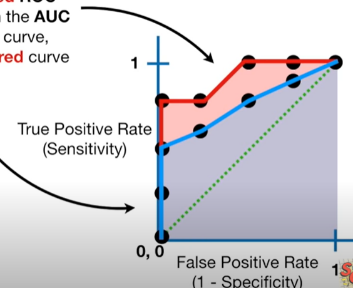
## נספח 4 : הצגת טבלה המציגה את ערכי ההיפר פרמטר כפונקציה של אחוז הדיוק על סט האימון וסט הואלידציה (Cross validation for DT Model):

## נספח 5 – הערכת פיצ'רים במודל SVM

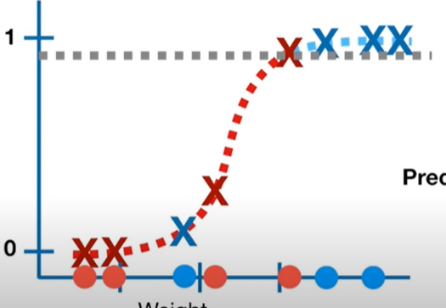


## נספח 6 : Cross validation for ANN Model

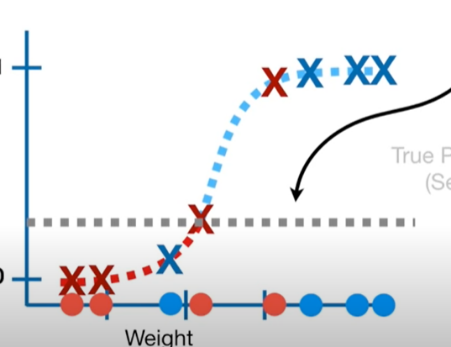
## נספח 7 – הסבר אודות thresholds ומטריקת הAUC

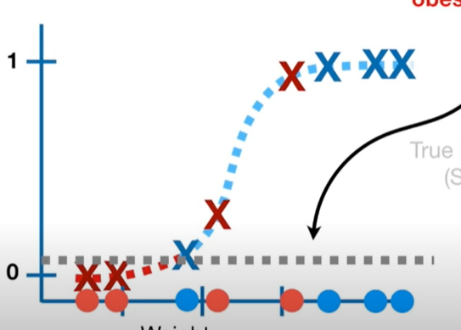


כל נקודה בגרף הAUC מתקבלת ע"י threshold שונה.



Threshold ספציפי



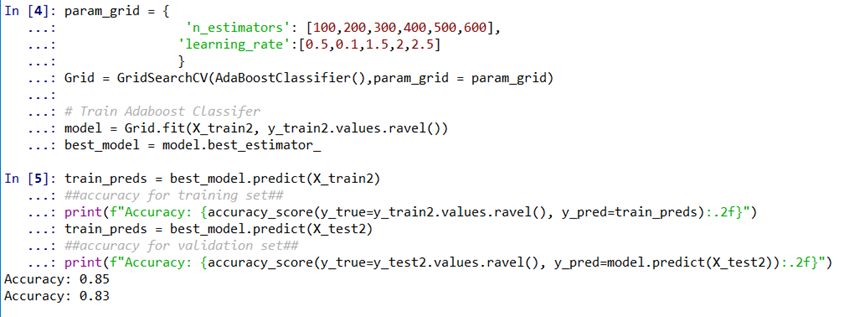


Threshold ספציפי

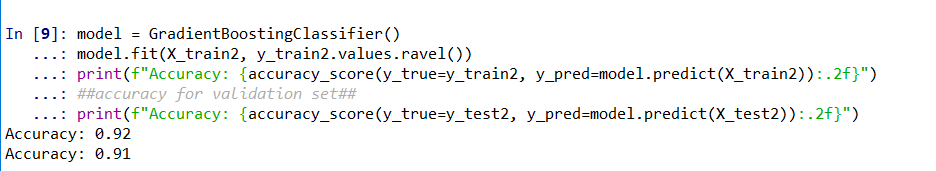
## נספח 8-שלוש רמות ראשונות לעץ ההחלטה עם הקונפיגורציה המיטבית

כל נקודה בגרף הAUC מתקבלת ע"י threshold שונה.

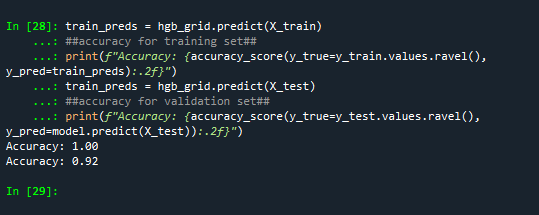
## נספח 9 – פלט Adaboost



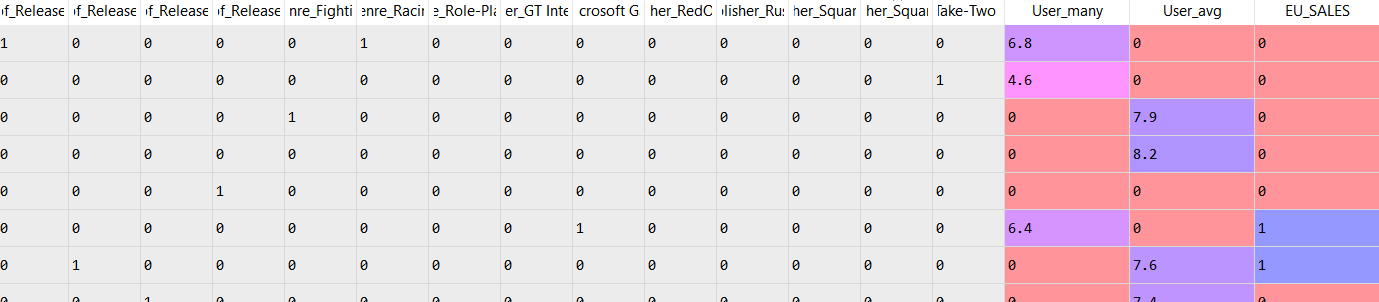
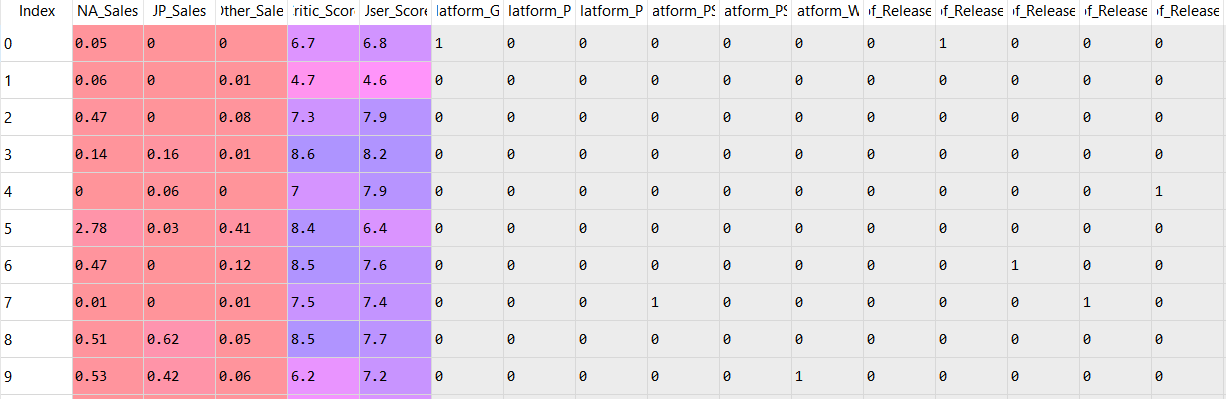
## נספח 10 - פלט GradientBoostingClassifier



## נספח 11 – פלט HistGradientBoosting



## נספח 12 – דוגמה למספר שורות מהדאטה:



# מקורות

## מקור 1- גישת הElbow method

## מקור 2- Davies Bouldin Index based hierarchical initialization K-means

## מקור 3- [Boosting the accuracy of your Machine Learning models | by Prashant Gupta | Towards Data Science](https://towardsdatascience.com/boosting-the-accuracy-of-your-machine-learning-models-f878d6a2d185)

## מקור 4- GradientBoostingClassifier (Friedman 2001)