**למידת מכונה**

פרויקט הקורס

חלק ב'

קבוצה 19 – ספיר גבאי 209516574, רוני קרוצ'קוב 211899034



**תוכן עניינים**

Data Leakage...................................................................................................3

Decision Trees..................................................................................................3

6………………………….………….……………………Artificial Neural Networks

SVM..................................................................................................................8

Unsupervised Learning -Clustering ..................................................................9

Evaluation.........................................................................................................10

Improvements...................................................................................................12

Test...................................................................................................................12

**קישור לColab Notebook**

**[ML Project - colab](https://colab.research.google.com/drive/1t2INV4qfUoUoWMKuWg9q5Y5rA4oQsxgq?usp=sharing)**

**קישור לGIT**

[**ML\_Project - github**](https://github.com/SapirGabay/ML_Project)

**0. לפני שמתחילים**

בשלב זה חילקנו את הנתונים ל־80% סט אימון ו־20% סט ולידציה, תוך שימוש בפרמטר random\_state=42 לשם שמירת עקביות. תהליך כוונון ההיפר־פרמטרים בוצע באמצעות Cross Validation מסוג Stratified K-Fold (K=5) על סט האימון בלבד. בסיום התהליך נבחר המודל בעל ביצועי ה־AUC הגבוהים ביותר לפי ממוצע תוצאות הקרוס ולידציה, והוא הוערך גם על סט הוולידציה החיצוני (20%) שנשמר מראש. לבסוף, המודל הנבחר הופעל על סט הבדיקה (test) לצורך חיזוי.

**Data Leakage**

זליגת מידע (Data Leakage) מתרחשת כאשר מידע מתוך סט הוולידציה “דולף” לשלב האימון וגורם להערכת־יתר של ביצועי המודל. כדי למנוע זאת, יישמנו שיטת Hold-Out וחילקנו את הדאטה מראש ל־80% סט אימון ו־20% ולידציה חיצוני, שלא שימש באימון כלל. בתוך סט האימון ביצענו כוונון היפר־פרמטרים בשיטת Stratified K-Fold (K=5), שמפחיתה את ההטיה בתהליך האימון על ידי שימוש חוזר בחלקים שונים של הדאטה כסט אימון וסט בדיקה פנימי. הקפדנו שכל שלבי העיבוד, הנירמול והשלמות הערכים יתבצעו רק בתוך folds, ללא שימוש במידע מהיעד. גם על סט הבדיקה הוחלו שלבי העיבוד ללא גישה ל־y. כך הבטחנו הימנעות מזליגת מידע לאורך כל התהליך.

**Model Training .1**

* 1. **Decision Trees**

1. בדקנו מספר היפר-פרמטרים כדי למצוא אתהקונפיגורציה המיטבית לעץ ההחלטה**.**

Max Depth: פרמטר זה מייצג את העומק המרבי של עץ ההחלטה. אנו נרצה עומק מרבי מתון על מנת למנוע over-fitting ולשפר את יכולת ההכללה של העץ. בחנו ערכים שונים בין 3 ל-30 (בקפיצות של 2), כדי לחקור עומקים שונים של העץ.

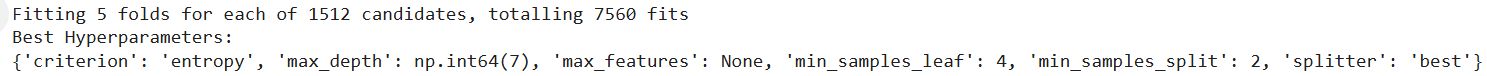
Minimum Samples Split: היפר-פרמטר זה קובע את מספר הסאמפלים המינימלי הנדרש לפיצול צומת פנימי במהלך בניית העץ ומשפיע על יכולת ההכללה של העץ. בחנו כמה אופציות [6, 4, 2] כערכים פוטנציאליים כדי להעריך את השפעתם.

Minimum Samples Leaf: היפר-פרמטר זה מגדיר את המספר המינימלי של הסאמפלים הנדרש להיות בצומת עלה. גם הוא משפיע על יכולת ההכללה של המודל ומונע מצב של פיצול עבור מספר מועט של סאמפלים, בחנו כמה אופציות [1, 2, 4] כערכים פוטנציאליים.

Criterion: היפר-פרמטר זה מגדיר את הקריטריון לפיו נבחר הפיצול האופטימלי בכל צומת בעץ ההחלטה. בחנו את האפשרויות 'gini' ו 'entropy', כאשר Gini מודד את מידת אי-הטהרה של הצומת בצורה חישובית יעילה יותר, בעוד ש Entropy מבוסס על האינפורמציה ומודד את כמות האי-ודאות או חוסר הסדר.

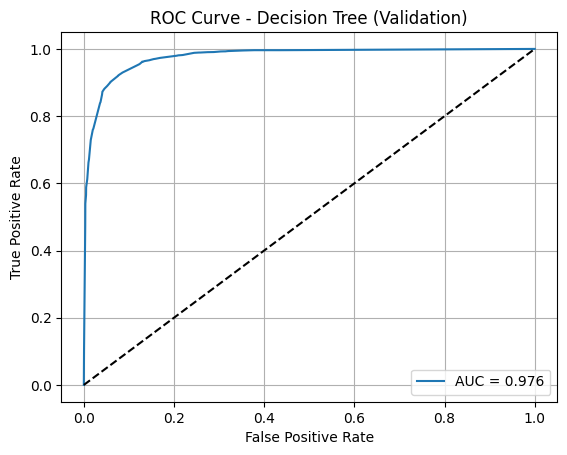
: Splitterפרמטר זה קובע את האסטרטגיה בה נבחרים הפיצולים לאורך בניית העץ. השתמשנו בשתי אפשרויות - 'best' אשר בוחרת את הפיצול הטוב ביותר בהתאם לקריטריון שנבחר Gini) או (Entropy ו 'random'-, אשר בוחרת פיצול רנדומלי מתוך קבוצה מוגבלת של אפשרויות.

max\_features: מגדיר את מספר הפיצ'רים שנבחנים בכל פיצול. בדקנו את הערכים: None (כל הפיצ'רים), 'sqrt' (שורש מספר הפיצ'רים) ו־'log2' (לוגריתם ב־2 של מספר הפיצ'רים), במטרה לבחון איזון בין דיוק לבין הכללה ולצמצם סיכון להתאמת יתר.

לאחר כיוונון ההיפר פרמטרים לפי K-fold כאשר k=5, קיבלנו את הקונפיגורציה המיטבית:

ביצועי המודל:

|  |  |
| --- | --- |
| **סט הנתונים** | **אחוז הACU-ROC** |
| אימון | 98.08 % |
| וולידציה | 97.63 % |

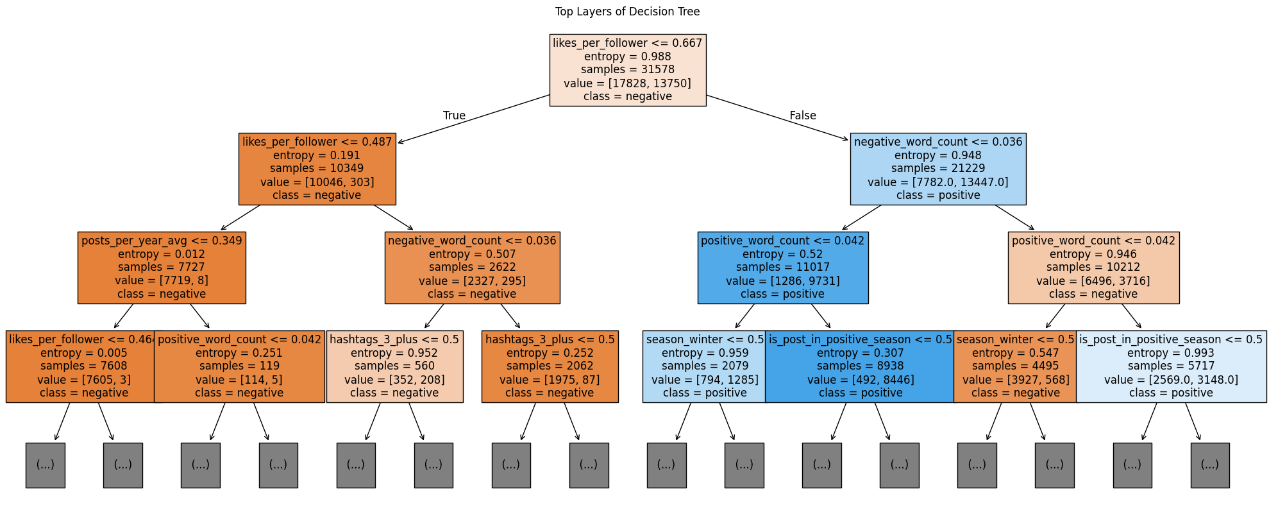


המודל הגיע ל־AUC גבוה הן על סט האימון (0.9808) והן על סט הוולידציה (0.9763), תוך שמירה על פער נמוך ביניהם - דבר המעיד על איזון מוצלח בין דיוק לבין יכולת הכללה. השימוש במגבלות על עומק העץ (max\_depth) ובפרמטרים כמו min\_samples\_leaf סייע לצמצם למידת יתר של דפוסים מקומיים ורעש, ולאפשר למודל ללמוד תבניות כלליות ורלוונטיות יותר.

1. **פירוש והבנת המודל**

אחד היתרונות הבולטים של עץ החלטה הוא יכולת פירוש המודל (interpretability), כלומר להסביר כל החלטת סיווג באמצעות כלל ברור, המבוסס על בדיקה פשוטה של ערכי תכונות. תכונה זו מאפשרת הבנה מעמיקה של האופן שבו המודל עושה שיפוט בין קטגוריות שונות ומאפשרת להסביר תחזיות באופן אינטואיטיבי וברור .זאת בניגוד למודלים כמו רשתות עצביות (ANN) שבהם קשה לעקוב אחרי הלוגיקה הפנימית.

במסגרת משימת ניתוח הסנטימנט שלנו, יכולת פירוש המודל מסייעת בזיהוי הפיצ’רים המרכזיים וחשיפת סדר העדיפויות שלהם: אנו רואים באיזה שלב ובאילו תנאים כל פיצ’ר משפיע על הסיווג, כיצד נוצרים כללי ההחלטה, ואילו דפוסים עיקריים מופיעים בנתונים.

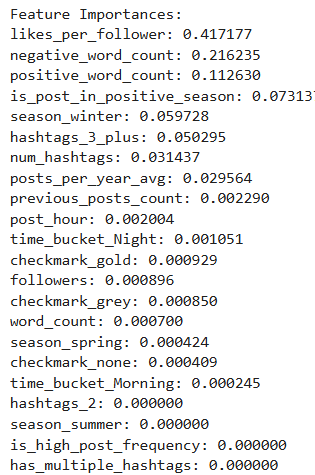
הגרף הבא מציג את השכבות העליונות של העץ שהתקבל לאחר כוונון ההיפר־פרמטרים:

\* גרף מלא של העץ ניתן לראות במחברת colab.

כשמתבוננים על מבנה העץ שנוצר ניתן להפיק מספר תובנות מרכזיות על דפוסי הסיווג:

1. הפיצולים בשורש העץ משקפים את הפיצ’רים המרכזיים שנבחרו על ידי המודל. בצומת השורש מופיע הפיצ’ר likes\_per\_follower (גבול ≈ 0.667), מה שמעיד כי יחס הלייקים לעוקבים הוא הגורם המשפיע ביותר בתחילת תהליך הסיווג. עבור דוגמאות שבהן הערך נמוך מהסף, המודל נוטה לסווג את הפוסט כשלילי כלומר אינטראקציה נמוכה יחסית עשויה להעיד על סנטימנט שלילי.

בהמשך, פיצ'רים כמו negative\_word\_count, positive\_word\_count, ו־is\_post\_in\_positive\_season מופיעים בצמתים מרכזיים נוספים, ומצביעים על כך שגם תוכן הפוסט (מספר מילים חיוביות או שליליות) וגם עונת הפרסום משפיעים על הסיווג.

2. מבנה העץ מצביע על כך שרוב הסיווגים מתקבלים לאחר מספר פיצולים מועט יחסית (כ־3-4 שלבים), דבר המעיד על יעילות גבוהה של המודל בסיווג של דוגמאות ברורות. יחד עם זאת, ההחלטות אינן מבוססות על פיצ'ר בודד, אלא על שילוב של מספר תנאים,   
לדוגמה: יחס לייקים נמוך, מספר מילים שליליות גבוה ועונת פרסום. תבנית זו מלמדת שהמודל מתבסס על הקשרים מורכבים בין מאפיינים שונים, ולא לפי אינדיקטור יחיד.

3. לאחר חישוב ה־Feature Importances ניתן לראות כי יש התאמה בין ההיררכיה בעץ לבין דירוג הפיצ'רים לתרומת למשימת הסיווג. הפיצ’רים likes\_per\_follower, negative\_word\_count ו־positive\_word\_count, שמופיעים בצמתים הראשיים של העץ, מדורגים בראש טבלת ה־Feature Importances, מה שמצביע על תרומתם הגבוהה להחלטות המודל. במיוחד, הפיצ’ר likes\_per\_follower בולט כבעל החשיבות הגבוהה ביותר, עם ערך של כ־41%, ונראה שהמודל מתבסס עליו כבר מהשלב הראשוני של הסיווג.

A graph with blue bars

AI-generated content may be incorrect.לעומת זאת, ישנם פיצ’רים שהשפעתם שולית, כגון hashtags\_2 ו־season\_summer, שקיבלו חשיבות אפסית. פיצ’רים כמו hashtags\_3\_plus, למרות שאינם בראש הטבלה, תרמו באופן עקבי להתפצלויות בעץ והופיעו במספר ענפים, מעיד על תרומה בינונית.

תובנות אלו מאפשרות להעריך אילו פיצ’רים תורמים משמעותית למשימת הסיווג, ויכולות לשמש בסיס להמשך אופטימיזציה, למשל, בגיזום תכונות לא אפקטיביות או פישוט המודל תוך שמירה על ביצועים גבוהים.

* 1. **Artificial Neural Networks**

**א.** לשם אימון מודל ה־MLP בגרסתו הבסיסית השתמשנו במחלקה MLPClassifier מתוך ספריית scikit-learn (גרסת sklearn.neural\_network). המודל אומן עם כל פרמטרי ברירת המחדל של הספרייה, למעט קביעת random\_state לצורך שיחזור תוצאות. [[1]](#footnote-1)

מבנה מודל ה־MLP **- מספר נוירונים בשכבת הקלט**: בשלב ה Feature Selection - בחרנו 30 פיצ’רים, לכן שכבת ה Input- מכילה 30 נוירונים (כל אחד מקבל פיצ’ר אחד). מודל ברירת המחדל כלל שכבה חבויה אחת עם 100 נוירונים, פונקציית ההפעלה היא ReLU, והיא מופעלת על כל נוירון בשכבה החבויה. שכבת הפלט כללה נוירון יחיד, המתאים למשימת סיווג בינארי, עם פונקציית הפעלה מסוג Sigmoid שנבחרת אוטומטית למשימה זו. מטריקת ההערכה שנבחרה לביצועי המודל היא AUC-ROC, בהתאם להנחיות.

ביצועי המודל:

|  |  |
| --- | --- |
| **סט הנתונים** | **אחוז הACU-ROC** |
| אימון | 99.01 % |
| וולידציה | 98.50 % |

ב. בשלב טיוב ההיפר־פרמטרים לרשת ה־MLP, בחרנו מספר פרמטרים מרכזיים, אשר הוגדרו בטווחים שונים ונבחנו באמצעות GridSearchCV על־פי מדד AUC.

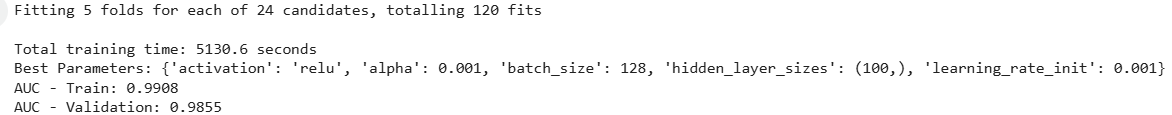
ניסינו לבדוק יותר קומבינציות אך בשל זמן ריצה ארוך (מעל 3 שעות) ולאור העובדה שקיבלנו תוצאות יפות גם כאשר בדקנו מספר מועט יותר של היפר - פרמטרים (לראייה אחוז דיוק גבוה במודל הדיפולטיבי).

**: hidden\_layer\_sizes** מגדיר את מבנה הרשת - מספר השכבות החבויות וכמות הנוירונים בכל אחת. נבדקו 2 תצורות: שכבה אחת עם 50 או 100 נוירונים, רשת גדולה יותר עשויה לשפר ביצועים אך מעלה סיכון ל־overfitting.

**batch\_size:** שולט בכמות הדוגמאות בכל עדכון משקל. נבחנו ערכים 32, 64 ו־128. ערכים קטנים עדכניים ומהירים אך רועשים, בעוד שגדולים מספקים יציבות אך דורשים יותר זמן ומשאבים.

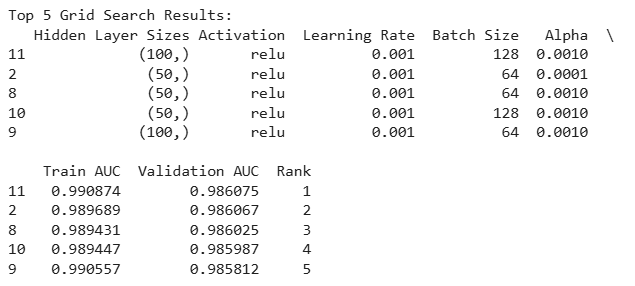
**alpha**: מייצג רגולריזציה מסוג L2. נבחנו ערכים 0.0001 ו־0.001. ערך גבוה מצמצם מורכבות ומפחית overfitting, בעוד שערך נמוך שומר על גמישות אך עלול להוביל ללמידת יתר.

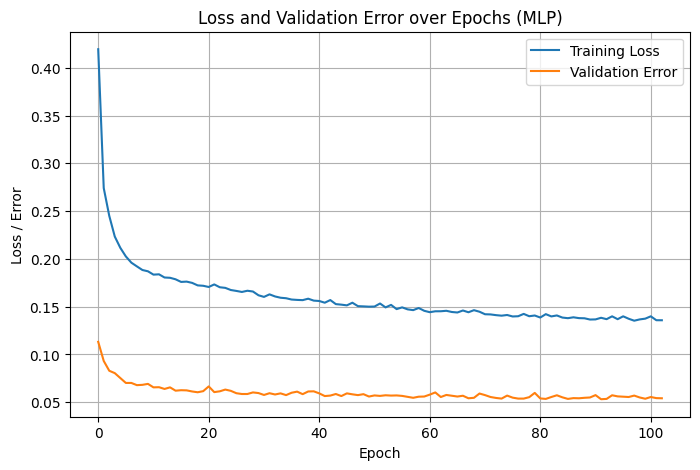
**: learning\_rate\_init** קובע את קצב הלמידה ההתחלתי. נבחנו 0.001 ו־0.01. ערך נמוך מוביל ללמידה יציבה אך איטית, גבוה יותר מאיץ למידה אך עלול לגרום לחוסר יציבות.

לאחר כיוונון ההיפר פרמטרים לפי K-fold כאשר k=5, קיבלנו את הקונפיגורציה המיטבית :

ביצועי המודל:

|  |  |
| --- | --- |
| **סט הנתונים** | **אחוז הACU-ROC** |
| אימון | 99.08 % |
| וולידציה | 98.55 % |

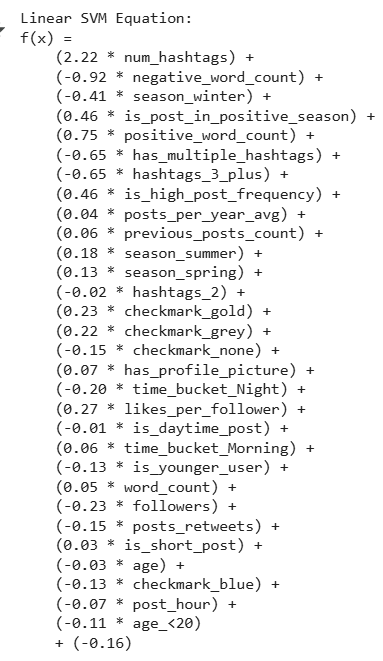
   
הטבלה מציגה את חמש הקונפיגורציות המובילות שנבחנו בתהליך כוונון ה־MLP, כולן עם ביצועי AUC גבוהים על סט האימון והוולידציה. הקונפיגורציה המדויקת ביותר כללה שכבה חבויה עם 100 נוירונים, alpha=0.001 ו־batch\_size=128, והובילה ל־AUC של 0.9909 על סט האימון ו־0.9861 על סט הוולידציה. מדובר בשיפור קל ביחס למודל הדיפולטיבי (AUC של 0.9901 ו־0.9850 בהתאמה), שניתן לייחס לבחירה מדויקת של ערכי ההיפר־פרמטרים. הפער הנמוך בין הסטים מעיד על איזון מוצלח בין למידת דפוסים לבין הכללה, תוך הימנעות מ־overfitting.

ג. בהתבסס על הגרף של פונקציית ה־Loss לאורך האפוקים, ניתן לראות כי המודל מתכנס בהצלחה: ערכי ה־Training Loss יורדים באופן חד בתחילת האימון ומתייצבים סביב אפוק 60, בעוד ש־Validation Error שומר על רמה יציבה ונמוכה לאורך רוב תהליך האימון. נקודת ההתכנסות הסופית התרחשה לאחר 103 אפוקים, בזכות שימוש ב־early stopping, שמעיד כי לא נצפה שיפור נוסף משמעותי בסט הוולידציה לאחר נקודה זו. התנהגות זו מעידה על יציבות ויכולת הכללה טובה של המודל.

* 1. ***SVM***

לשם אימון מודל SVM השתמשנו ב־SVC מספריית scikit-learn, עם גרעין לינארי (kernel='linear') על פי הדרישות. לצורך טיוב המודל נבחנו מספר היפר־פרמטרים:  
 C - קבוע רגולריזציה, שנבדק בטווחים שונים (0.1, 1, 10) כדי לאזן בין התאמה לדאטה לבין שליטה במורכבות המודל, class\_weight- שנקבע כ־'balanced' או 'None' במטרה להתמודד עם חוסר האיזון בין המחלקות, shrinking- שבוחן שימוש באסטרטגיית חישוב מהירה בקירוב לעומת פתרון מלא, ו־tol - סף העצירה של תהליך האופטימיזציה, שנקבע לערך 0.001 לפי ברירת המחדל. בחינת שילובים אלו אפשרה לבחור את הקונפיגורציה היעילה ביותר תוך שמירה על ביצועים מדויקים וזמן ריצה סביר.

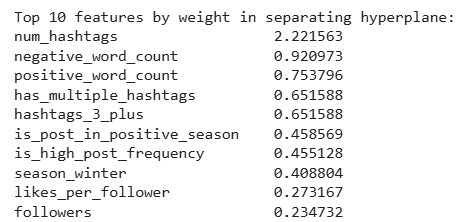
המודל הנבחר:

******ביצועי המודל:

|  |  |
| --- | --- |
| **סט הנתונים** | **אחוז הACU-ROC** |
| אימון | 96.48 % |
| וולידציה | 96.34 % |

משוואת קו ההפרדה שמתקבלת ממודל SVM עם kernel ליניארי מתארת היפר־מישור שמפריד בין דוגמאות עם סנטימנט חיובי לשלילי במרחב הפיצ’רים. קו זה מוגדר על־ידי משקל (coefficient) לכל פיצ’ר ו־bias (מונח חופשי), וכל נקודה מסווגת לפי הצד שבו היא נופלת ביחס למישור. ככל שהמקדם של פיצ’ר מסוים גבוה יותר בערכו המוחלט, כך השפעתו על ההחלטה גדולה יותר.

**משוואת מישור ההפרדה שלנו מוצגת משמאל 🡨**

******מניתוח המשוואה ניתן לראות כי הפיצ’רים המשפיעים ביותר על הסיווג כוללים את,num\_hashtags negative\_word\_count, positive\_word\_count, ו־likes\_per\_follower. סימן המשקל (חיובי או שלילי) מייצג את כיוון ההשפעה, כאשר פיצ'רים כמו positive\_word\_count דוחפים לסנטימנט חיובי, לעומת negative\_word\_count שמקושר לסנטימנט שלילי. ישנה הלימה בין הפיצ'רים המשמעותיים שנבחרו על ידי מודל ה־SVM לבין אלו שהופיעו בראש עץ ההחלטה.  
הטבלה הבאה מציגה באופן נוח לקריאה את 10 הפיצ'רים בעלי ההשפעה הגדולה ביותר על קו ההפרדה במודל ה־SVM הליניארי, כלומר את הפיצ'רים עם המשקלים (weights) הגבוהים ביותר מה שמעיד על חשיבותם לסיווג.

* 1. ***Clustering - Unsupervised Learning***

**הערה**: עקב בעיות תאימות גרסאות בין scikit-learn-extra לספריות אחרות בסביבת Google Colab, ובאישור שניתן בפורום הקורס, השתמשנו באלגוריתם K-Means כתחליף ל־K-Medoids לצורך ניתוח האשכולות.

א. בניגוד למודלים הקודמים שהתבססו על למידה מונחית (Supervised Learning), שלב זה כלל למידה בלתי־מונחית (Unsupervised Learning), שבה לא נעשה שימוש במשתנה המטרה sentiment. המטרה לזהות דפוסים חבויים במבנה הנתונים ולהבחין בין קבוצות (clusters) באופן טבעי, על סמך מאפייני הפוסטים בלבד, ללא תוויות (labels).

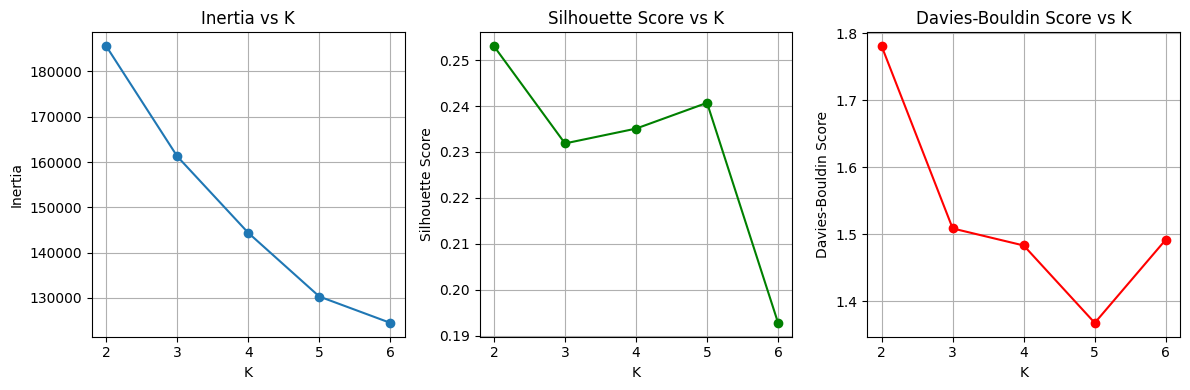
לצורך כך, בחרנו 6 פיצ’רים שנמצאו משמעותיים בשלבים הקודמים: likes\_per\_follower, positive\_word\_count, negative\_word\_count, word\_count, posts\_per\_year\_avg, ו־num\_hashtags. לפני ביצוע הקלאסטרינג, נרמלו הפיצ’רים בעזרת StandardScaler כדי להביא את כולם לאותו סדר גודל, ולאזן בין טווחי ערכים שונים. לאחר מכן, חישבנו מרחקים בין הדוגמאות לפי מדד Euclidean distance - מדד נפוץ עבור דאטה נומרי – והרצנו את אלגוריתם KMeans על מערך זה.

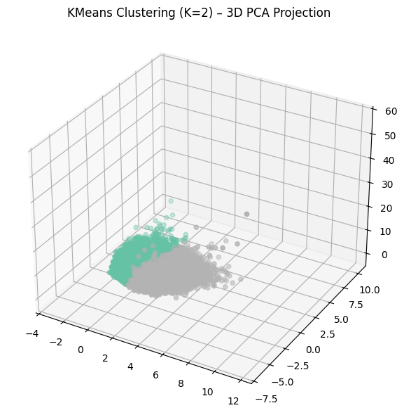
ב. כדי להעריך את איכות החלוקה לקלאסטרים, השתמשנו בשלושה מדדים מקובלים: Silhouette Score, Davies-Bouldin Index ו־Inertia.

**Silhouette Score**: מודד את מידת ההפרדה בין האשכולות, ערכים גבוהים מצביעים על חלוקה ברורה, והמדד הגיע לשיאו ב־K=2, מה שמעיד על הפרדה מיטבית.

**Inertia**: מודד את סך המרחקים בין כל דוגמה למרכז האשכול שלה - ככל שקטן יותר, כך האשכולות צפופים. נראית ירידה מתמשכת ככל ש־K גדל, אך אין נקודת "מרפק" חדה ומובהקת. הירידה בין K=2 ל־K=3 עדיין משמעותית יחסית, כך שאפשר להעדיף K=2 או K=3, אבל לא בהכרח יש "שבירה" ברורה.  
**Davies-Bouldin Score**: בוחן את הקרבה בין אשכולות ואת מידת הפיזור הפנימית שלהם. ערכים נמוכים מעידים על אשכולות טובים יותר. הערך הנמוך ביותר התקבל ב K=5. מס' האשכולות נבחר לפי ערך ה־Silhouette Score הגבוה ביותר, מאחר שמדד זה מאזן בין צפיפות פנימית להפרדה בין אשכולות. לכן בחרנו בK=2 כמספר האופטימלי.

A screenshot of a computer screen

AI-generated content may be incorrect.  
  
  
  
  
  
מעבר למדדים הטכניים, כאשר השווינו בין הקלאסטרים לתוויות הסנטימנט, נמצא קשר ברור, אשכול 0 כלל כ־81% פוסטים שליליים, בעוד ש־אשכול 1 כלל כ־83% פוסטים חיוביים. נתון זה מצביע על כך שהמודל הצליח להבחין בין הסנטימנטים גם ללא שימוש בתוויות בזמן האימון. החלק התחתון של הטבלה מציג את הממוצע של כל פיצ'ר בכל אשכול (cluster).   
כלומר, הוא עוזר להבין מה מאפיין כל אשכול מבחינת הפיצ'רים שנבחרו. ניתן לראות שהמאפיינים של אשכול 1 תואמים לסנטימנט חיובי: הרבה לייקים, מילים חיוביות, יותר האשטגים. ואילו אשכול 0 נוטה להיות שלילי. זה מראה כי החלוקה לאשכולות לפי הפיצ’רים הצליחה לתפוס דפוסים בעלי משמעות סנטימנטלית, גם בלי לדעת את ה־sentiment.

A blue and orange pie chart

AI-generated content may be incorrect.A graph with a diagram of a number of dots

AI-generated content may be incorrect.במסגרת הניתוח החזותי, הופקו תרשימי פיזור דו־ממדיים ותלת־ממדיים (PCA) של האשכולות וכן תרשים עוגה המשקף את התפלגות הסנטימנט לפי אשכול. התרשימים ממחישים הפרדה טובה בין הקבוצות והדגמה ויזואלית של הקשר המשמעותי שנמצא בין האשכול לבין משתנה המטרה, גם ללא שימוש בו באימון.

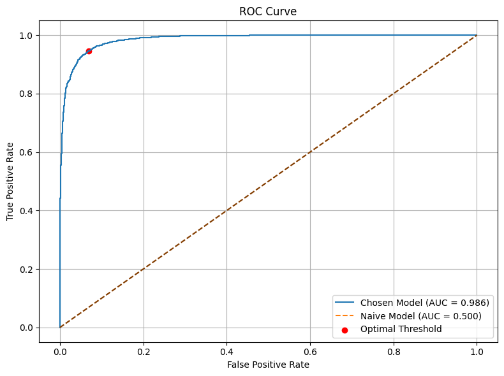
**Evaluation .2**

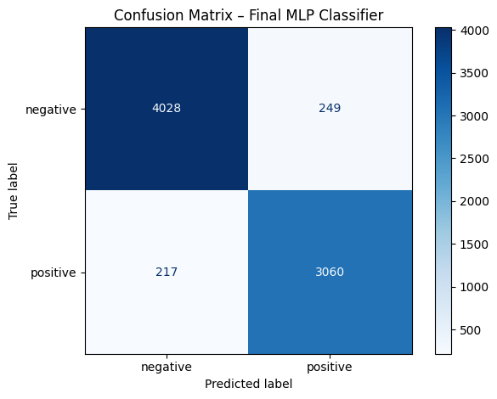
**א. השוואה בין המודלים:**

|  |  |  |  |
| --- | --- | --- | --- |
| **מודל** | SVM | **MLP** | DT |
| **אחוז הACU-ROC** | 96.34 % | **98.55 %** | 97.63 % |

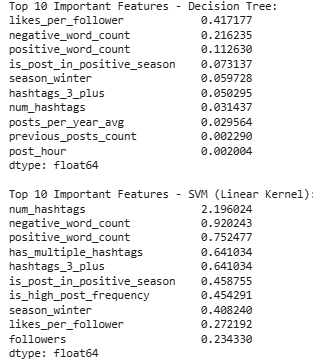


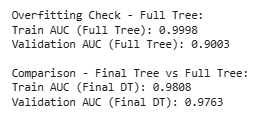
ה־MLP השיג את הביצועים הגבוהים ביותר, עם איזון טוב בין אימון לוולידציה, ולכן נבחר כמנוע החיזוי הסופי. יתרונו העיקרי הוא ביכולתו ללמוד ייצוגים מורכבים ובלתי־ליניאריים מתוך הנתונים. מגבלה אפשרית של MLP היא היעדר יכולת הסבר (interpretability) - בשונה מעץ החלטה שמאפשר מעקב אחרי תהליך קבלת ההחלטות.

ב. **ניתוח ויזואלי של ביצועי המודל**

עקומת ה־ROC של מודל ה־MLP מדגימה ביצועים גבוהים מאוד   
(AUC = 0.986), עם יתרון ברור לעומת המסווג הנאיבי  
(AUC = 0.5). נקודת הסף ((threshold האופטימלית שסומנה בגרף ממקסמת את ההפרש בין True Positive Rate ל־False Positive Rate.

ממטריצת הבלבול על סט הוולידציה עולה כי המודל מצליח לזהות נכון הן פוסטים חיוביים והן שליליים ברמת דיוק גבוהה, עם איזון טוב בין שגיאות מסוג false positive ל־false negative . הדבר תומך במסקנה שהמודל יציב ואינו מוטה לקבוצה מסוימת.

ג. בהתבוננות על חשיבות הפיצ'רים בכל מודל, בלטו פיצ'רים משותפים כמו likes\_per\_follower, negative\_word\_count, ו־positive\_word\_count, מה שמעיד על עקביות בזיהוי התכונות המשפיעות ביותר על הסנטימנט. גם ב־DT וב־SVM הם הופיעו בראש הדירוג.

ביצענו בדיקת Overfitting לעץ ההחלטה, הרצנו עץ מלא (ללא הגבלות), וקיבלנו פער חד בין אימון (AUC = 0.9998) לוולידציה (AUC = 0.9003) - אינדיקציה ברורה ל־Overfitting. לעומתו, העץ המאוזן שבחרנו עם עומק מקסימלי של 7, הפחית את הפער והציג הכללה טובה יותר.

**Improvements .3**

א. בחרנו לשפר את מודל ה־MLP, שהשיג את הביצועים הגבוהים ביותר בשלב הקודם. השיפור התבסס על הוספת פיצ’רים טקסטואליים באמצעות TF-IDF ושיפור מבנה המודל בעזרת קצב למידה אדפטיבי (adaptive learning rate) ו־Early Stopping. מטרת השיפורים הייתה להעשיר את המידע שנכנס למודל, ולאפשר למודל להתכנס בצורה מדויקת יותר, תוך הפחתת הסיכון ל־Overfitting.

ב. הצעות לשיפור:

**שיפור מתחום עיבוד הנתונים (חלק א'):** הפעלנו את טכניקת TF-IDF על עמודת text, תוך בחירת 100 המילים המובהקות ביותר, והוספנו את הציונים שהתקבלו כ־features חדשים למודל. בניגוד לשיטת הספירה הישירה של מילים חיוביות ושליליות (מחלק א'), TF-IDF בוחנת את החשיבות היחסית של כל מילה בהקשר לכלל הטקסטים - כך שמילים נדירות אך ייחודיות מקבלות משקל גבוה, בעוד מילים שכיחות וחסרות הבחנה זוכות למשקל נמוך יותר.

**שיפור מתחום יצירת המודל (חלק ב')**: אימנו רשת MLP עם קצב למידה אדפטיבי, המאפשר התאמה דינמית של קצב הלמידה לפי קצב השיפור בביצועים, במטרה לשלב למידה מהירה ויציבה לאורך האימון. בנוסף, הפעלנו Early Stopping כדי לעצור את האימון עם עצירת השיפור, ולמנוע למידת יתר.

ג. **השוואת ביצועים והערכת השיפורים**

המודל המשופר, שהתבסס על שילוב פיצ’רים מטכניקת TF-IDF יחד עם קצב למידה אדפטיבי (Adaptive Learning Rate), השיג AUC של 0.984 על סט הוולידציה, לעומת 0.986 במודל ה־MLP המקורי. למרות שיפור מתודולוגי, לא נצפתה עלייה בביצועים, ואף חלה ירידה קלה. ייתכן שהוספת פיצ’רים טקסטואליים גרמה לרעש או לממדיות עודפת, או שהפרמטרים של המודל לא היו מותאמים מספיק לאופי הנתונים החדש.

ד. לא נצפה שיפור בביצועים לאחר השיפור (AUC 0.984), אך לאור התוצאה הגבוהה שהתקבלה במודל המקורי (AUC 0.986), אנו סבורות כי המודל חוזה את הסנטימנט באופן מדויק ויציב.

**.4 Test**

בשלב זה ביצענו חיזוי על נתוני test לאחר שהעברנו אותם את אותם שלבי עיבוד כמו סט האימון (באמצעות פונקציית מעטפת), תוך שימוש במודל ה-MLP שהראה את ביצועי הוולידציה הגבוהים ביותר. התוצאה נשמרה בקובץ group19.csv בהתאם להנחיות.

1. <http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html> [↑](#footnote-ref-1)