**פרויקט - חלק ב'**



תוכן עניינים

[הכנת הנתונים לאימון ובחינה 3](#_Toc172404019)

[1. עצי החלטה-Decision Trees 3](#_Toc172404020)

[1.1. הכנת הנתונים למודל עץ החלטה 3](#_Toc172404021)

[1.2. בניית עץ החלטה מלא באמצעות סט האימון 3](#_Toc172404022)

[1.3. כוונון פרמטרים (Hyperparameter- tuning) 4](#_Toc172404023)

[1.4. אימון עץ ההחלטה 6](#_Toc172404024)

[2. רשתות נוירונים- Neural Networks 9](#_Toc172404025)

[2.1. הכנת הנתונים להכנסה לרשת נוירונים 9](#_Toc172404026)

[2.2. הרצת הרשת בערכי ברירת מחדל 9](#_Toc172404027)

[2.3. כוונון פרמטרים (Hyperparameter- tuning) 9](#_Toc172404028)

[2.4. אימון רשת הנוירונים 11](#_Toc172404029)

[3. למידה בלתי מונחת- Unsupervised Learning: Clustering 11](#_Toc172404030)

[3.1. הרצת מודל עם ערכי ברירת מחדל 11](#_Toc172404031)

[3.2. דיון בתוצאות החלוקה לאשכולות 11](#_Toc172404032)

[3.3. אימון מחדש עם ערכי k משתנים (סעיף רשות) 12](#_Toc172404033)

[4. אימון מודל נוסף- רגרסיה לוגיסטית 13](#_Toc172404034)

[5. השוואה בין מודלים 14](#_Toc172404035)

[5.1. יתרונות DT ו-MLP למול K-means 14](#_Toc172404036)

[5.2. השוואת ביצועי המודלים 14](#_Toc172404037)

[6. המודל נבחר 15](#_Toc172404038)

[6.1. הצגת המודל 15](#_Toc172404039)

[6.2. ניתוח תוצאות המודל 15](#_Toc172404040)

[נספחים 16](#_Toc172404041)

# הכנת הנתונים לאימון ובחינה

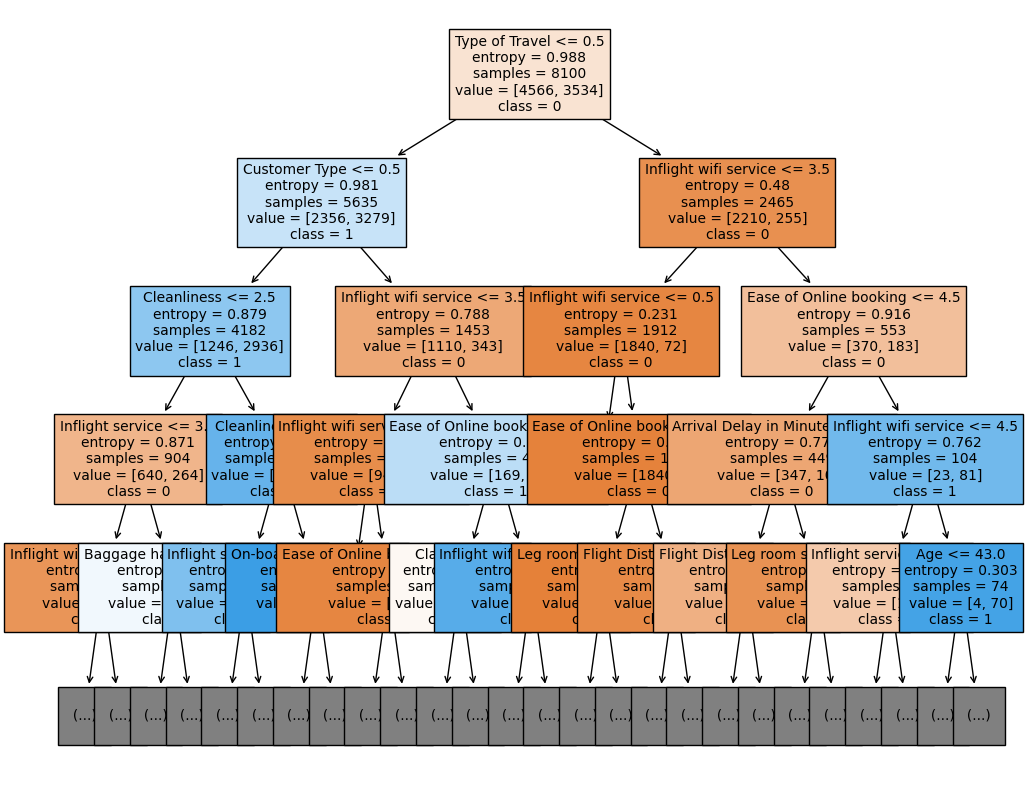
בחרנו להקצות 10% מסט הנתונים לטובת סט בחינה (כאשר השאר לסט האימון). גודל סט הבחינה הוא 900 רשומות. בחנו בסט בחינה גדול יחסית על מנת לצמצם ככל הניתן את הרעש אתו יתמודדו המודלים אל מול סט הבחינה, וזאת תוך כדי הימנעות מצמום משמעותי של סט האימון על מנת לוודא שיש מספיק נתונים לאימון המודל בצורה טובה ומקיפה.

# עצי החלטה-Decision Trees

# הכנת הנתונים למודל עץ החלטה

הכנת הנתונים שביצענו במיוחד עבור עץ ההחלטה (מעבר להכנה וניקוי המשתנים מחלק א' של הפרויקט) היא קידוד של משתנים קטגוריאליים כמתואר בקובץ הקוד של עץ החלטה). הקידוד בוצע על מנת כך שמודל עץ ההחלטה יוכל לקרוא וסווג את הרשומות לפיהן. בנוסף בוצע טיפול במשתנים חסרים וחריגים דרך החלפתם במשתנים מתוך התפלגות שנבנתה עבור כל מאפיין לפי הרשומות התקינות שלו.

# בניית עץ החלטה מלא באמצעות סט האימון

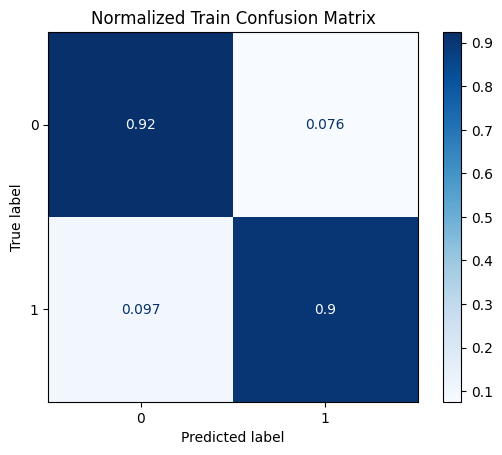
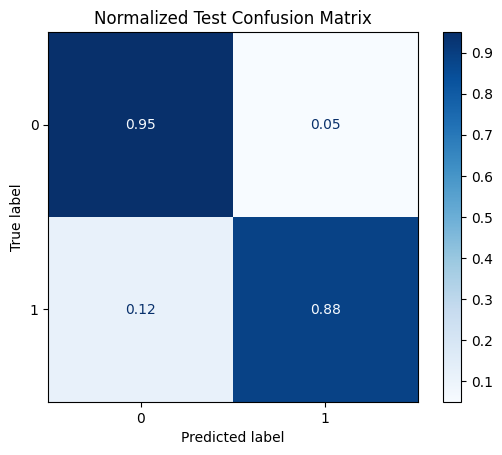
בחרנו במדד F1 לבחינת ביצועי העץ כך שנוכל שיקוף מהימן של ביצועי המודל גם עבור ערכים שאינם מאוזנים. בנוסף, נאמר כי המודל יבחן על פי מדד זה ולכן נרצה לבצע אופטימיזציה למודל על פי הקריטריונים לפיהם הוא נבחן.

לאחר הרצת עץ מלא ללא הגבלות או כוונון קיבלנו את הערכים הבאים עבור מדד F1:

test\_f1\_accuracy: 0.8489583, train\_f1\_accuracy: 1.0

ניתן לראות שביצועי המודל עבור סט האימון הם מושלמים, בעוד ביצועיו על סט הבחינה אינם מספקים (0.849). ניתן לחשוד כי מדובר במצב של התאמת יתר. נצפה לקבל מצב זה של התאמת יתר עבור עץ החלטה מלא כאשר לא מוגדרים פרמטרים או הגבלות כלשהן למודל. לא ניתן לקבוע כי תמיד יתקבלו תוצאות שכאלה, אך בהחלט ניתן לצפות להן ועל כן נדרש לבצע כיוונן פרמטרים.

בנוסף למדד F1, בחרנו בשלב זה בלבד לבחון את מטריצת הבלבול על מנת להבין האם המודל מוטה יותר עבור אחת המחלקות ("מרוצה" או "לא מרוצה"). ניתן לראות גם במטריצות הבלבול כי ביצועי המודל על סט האימון מושלמות בעוד שביצועיו על סט הבחינה לוקות בחסר, בעיקר ביכולתו של המודל לזהות מקרים של שביעות רצון ("מרוצה").



# כוונון פרמטרים (Hyperparameter- tuning)

לכל בעיה קיימת קונפיגורציה ייחודית לה של כוונן היפר-פרמטרים. נחפש את הקונפיגורציה שתתאים למודל שלנו על פי ההיפר-פרמטרים הבאים:

1. **max\_depth**: העומק המקסימלי של העץ. מגדיר את מספר הרמות המקסימלי שהעץ יכול לגדול אליו. עץ עמוק מדי יכול להוביל ל-overfitting, בעוד שעץ רדוד מדי עלול לא לתפוס מספיק מידע ולגרום ל-underfitting, ולכן נחפש את ערך מתאים ומאוזן עבורו.

2. **min\_samples\_split**: המספר המינימלי של דגימות הנדרש לפיצול צומת. נרצה למנוע מהעץ להתפצל תחת מקרים קטנים ולהתחקות אחרי רעשים בסט הנתונים. ערך גבוה יכול למנוע פיצולים עודפים (מה שמקטין את הסיכוי ל-overfitting), אך חשוב שלא להגדיר ערך גבוה מדי כך שהעץ יהיה רגיש מספיק לסט הנתונים. ערך נמוך מאפשר פיצולים רבים יותר שיכולים להוביל להעמסת יתר ללא הגבלה מדויקת מספיק, ולכן נחפש ערך מאוזן.

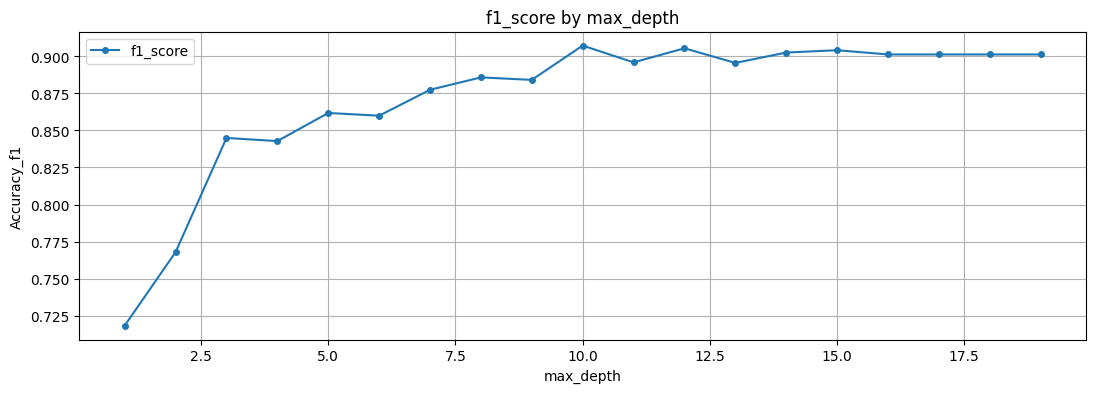
3. **min\_samples\_leaf**: המספר המינימלי של דגימות הנדרש בכל עלה. היפר-פרמטר זה מונע היווצרות עלים קטנים מדי. ערך גבוה מדי יכול לגרום לעץ להיות פשוט מדי, בעוד שערך נמוך מדי יכול לגרום ל-overfitting ומאותה סיבה נחפש גם לו ערך מאוזן.

4. **max\_features**: המספר המקסימלי של תכונות שנלקחות בחשבון בעת חישוב הפיצול הטוב ביותר. הגבלת מספר התכונות שנלקחות בחשבון בכל פיתול יכולה למנוע שילוב התחשבות בגורמי רעש ולהקטין את הסיכוי ל-overfitting.

5. **min\_impurity\_decrease**: הצמצום המינימלי ב-impurity הנדרש לפיצול צומת. צמצום זה מונע פיצולים קטנים וחסרי משמעות. ערך גבוה יותר יבנה מודלים פשוטים יותר, בעוד שערך נמוך מאפשר פיצולים מרובים שיכולים להוביל ל-overfitting ולכן נבדוק מה האיזון הדרש.

בחרנו לבצע בחינה של היפר-פרמטרים בשני שלבים, תחילה איטרטיבי באינטרוולים גדולים ("גסים") בין ערכי ההיפר פרמטרים שנבדקו. לאחר מציאת טווח ערכים שמציג ביצועי טובים יותר עבור העץ עברנו לבחינת ערכים איטרטיבית "עדינה", כלומר באינטרוולים קטנים ככל הניתן בטווח ערכים מצומצם סביב הערכים שמצאנו בשלב הקודם (מעיין חיפוש גס ולאחריו חיפוש מקומי). את ערכי האיטרציות שביצענו ותוצאותיהם ניתן לראות בנספח (נספח 2: הרצות היפר-פרמטרים עבור עץ בינארי:

). בחרנו לבחון מספר ערכי K גדולים יחסית באופן איטרטיבי גם כן. בחרנו בערכי K גדולים יחסית כיוון שסט הנתונים אינו גדול ונרצה שסט הבחינה יהיה מייצג ככל הניתן (כלומר יציב ומנוקה מרעשים עד כמה שניתן). המודל שלנו אינו מסובך באופן יחסי ולכן אין מניעה להשתמש בערכי K גבוהים מבחינת סיבוכיות חישוב, נוסף על כך ששימוש ב-Google Collab מקצה לנו כח חישוב נוסף, מה שמאפשר בדיקה איטרטיבית מדוקדקת באופן יחסי. ערך ה-K שנבחר הוא 11. ערכי הפרמטרים max\_depth ו- min\_samples\_leafהציגו תנודתיות מעניינת למול מדד F1, והיוו עבורנו אתגר מרכזי בתהליך כוונון הפרמטרים:

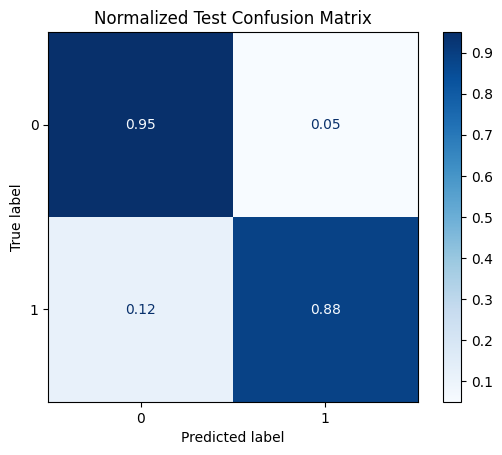
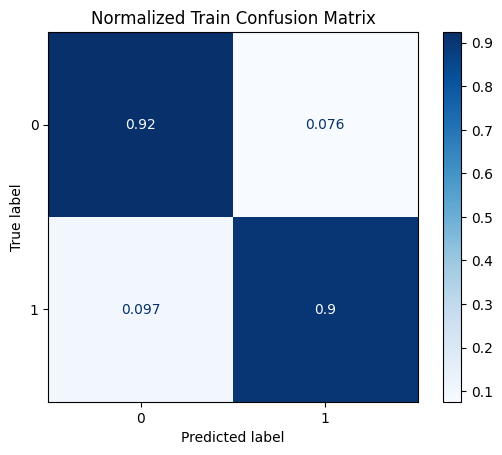


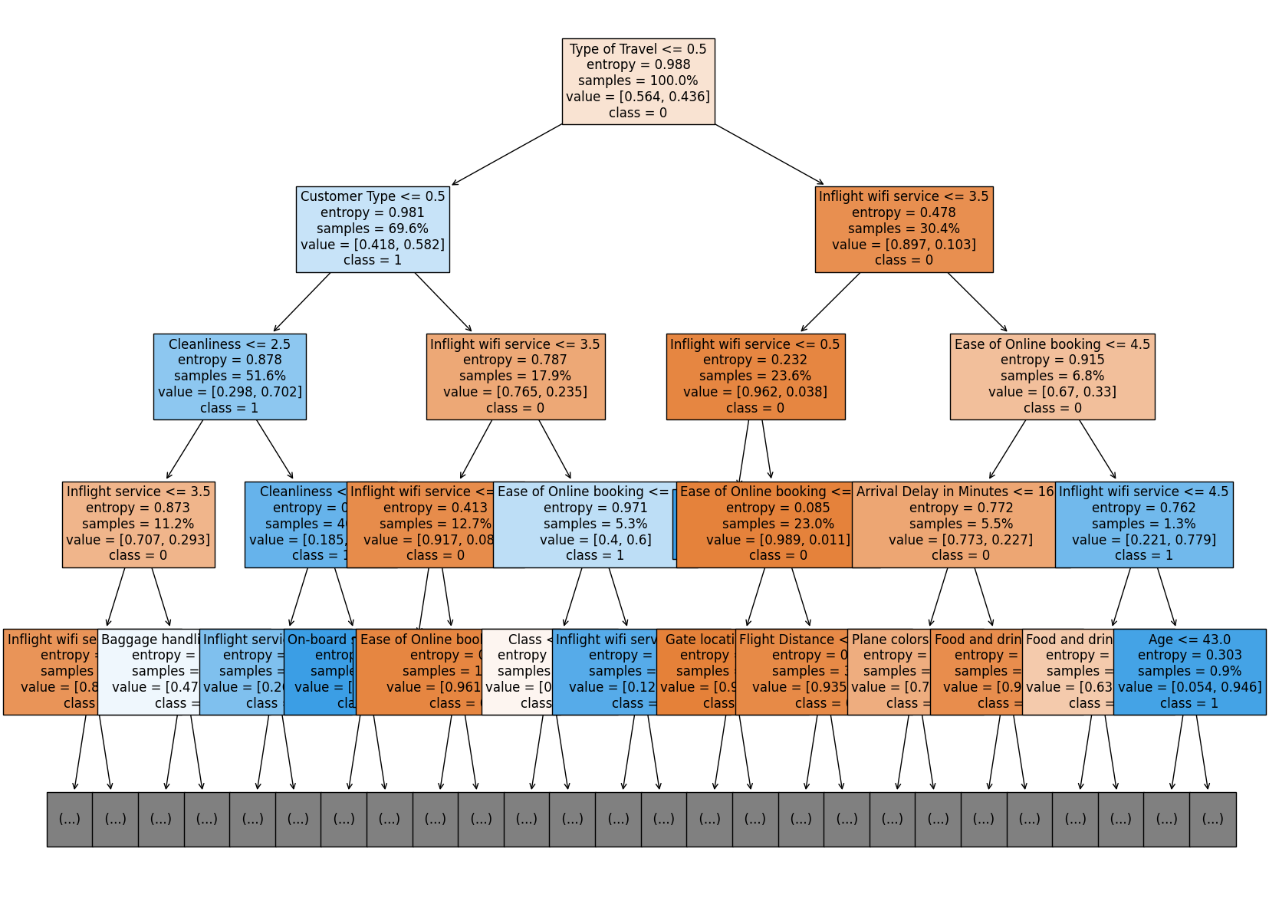
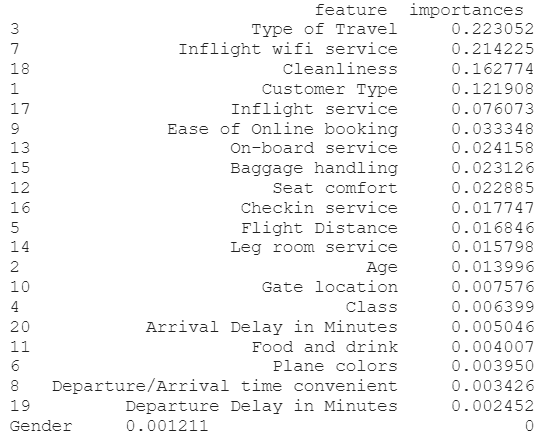
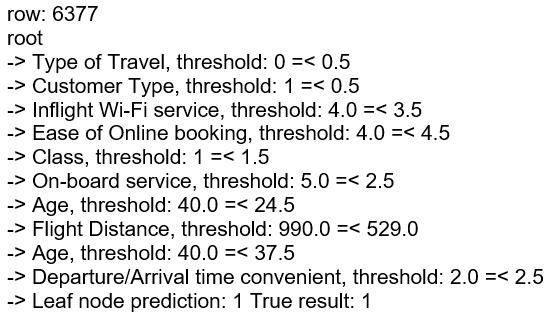
תמונה שמכילה קו, עלילה, תרשים, טקסט

התיאור נוצר באופן אוטומטי

# אימון עץ ההחלטה

* אחוזי הדיוק של מדד ההצלחה F1 על סט הבחינה הגיעה ל-0.9072 (ועבור סט האימון 0.9028) כאשר בבדיקה נוספת של מטריצת הבלבול ניתן לראות שיכולת הזיהוי כעת מאוזנת יותר עבור שתי הקטגוריות בהשוואה לעץ המקורי. ההסבר לשיפור הין העצים נובע מהפרמטרים שהוגדרו על מנת לקטום את העץ כך שימנע מ-Overfit.



* ניתן להסיק כי הקריטריונים 'max\_features ו- min\_impurity\_decrease לא משפיעים מהותית על יכולת העץ להימנע מ-Overfit ולכן ערכיהם נקבעו כ-None. במקביל, ערכי מדד ההצלחה F1 רגישים לשינויים בערך הפרמטר min\_samples\_leaf. ההבדל והשיפור שקיבלנו בהשוואה לעץ המלא נובע מכוונון הפרמטרים על מנת להימנע ממצב של Overfit תוך רגישות מספקת שמועת מהעץ הליקטם מוקדם מדי. כך הצלחנו מצד אחד לשפר את העץ כך שימנע מלהתחקות אחר רעשים ומצד שני למנוע מצב של אדישות לנתונים מהותיים בסט האימון.
* תובנות: העץ מגדיר את Type of Travel כמאפיין חשוב בחיזוי מידת שביעות הרצון של הטסים מהטיסה, כפי שהראו המדדים הסטטיסטים בחלק א' של הפרויקט. נוסף על כך ניתן לראות שהמודל מצא חשיבות גבוה גם למאפיינים Customer Type ו-Inflight Wi-Fi Service היות וגם הם ממוקמים גבוה בתהליך הסיווג בעץ.
* פונקציית feature\_importances במודל DecisionTreeClassifier מחזירה מדדים של חשיבות המאפיינים במודל. החשיבות של מאפיין מציינת כמה חשוב המאפיין בתהליך הלמידה של המודל. ככל שהערך של החשיבות גבוה יותר, כך המאפיין הוא יותר חשוב לקביעת החלטות במודל. התוצאות של פונקציית feature\_importances עזרו לנו להבין כיצד המודל קיבל את ההחלטות. בהקשר של הסעיף הקודם, ניתן לראות שהתוצאות של feature\_importances מתיישבות בדיוק עם ההבנה שלנו את חשיבות המאפיינים בעץ- מאפיין שנמצא מוקדם יותר בשלבי ההיררכיה הוא ככל הנראה בעל מדד feature\_importances גבוה יותר (לכל הפחות ברמות הראשונות, לפי שהעץ מסתעף והרמות בודקות מאפיינים שונים ברמות הבאות).
* סעיף רשות: בחרנו את רשומה 6377 אקראית מסט הבחינה והעברנו אותה דרך העץ באופן פרטני. אלו הנתונים שהתקבלו:

ניתן לראות שהרשומה עברה בעץ דרך ההסתעפות השמאלית בשלב הראשון, הימנית בשלב השני, הימנית בשלב השלישי וכן הלאה. החיזוי תאם את תוצאות האמת וקבע שמודבר בלקוח מרוצה.

* סעיף רשות: בהשוואת מדדי האימפורטנס למול השערותינו מחלק א', ניתן לראות שאכן צדקנו לגבי המאפיינים Type of Travel ו-Cleanliness בהם חשדנו כבעלי השפעה (עקב ערכים סטטיסטים גבוהים בt-value p-value וקורלציה) ואכן מדדי האימפורטנס שלהם גבוהים ביחס לרוב המאפיינים. צדקנו גם לגבי המאפיין Plane Color אותו חשדנו שניתן להסיר עקב קורלציה נמוכה עם משתנה המטרה והעדר ספרות שתהווה סימוכין לחשיבות מאפיין זה, ואכן גם מדד האימפורטנס שלו נמוך מאוד.

תמונה שמכילה טקסט, תרשים, צילום מסך, מקביל

התיאור נוצר באופן אוטומטילמרות כל האמור, לא ניתן לקבוע כי מאפיין בעל קורלציה גבוה עם מאפיין המטרה הוא אכן בעל חשיבות דרסטית ובעל ערך אימפורטנס גבוה. כך למשל מדד Class הוא בעל קורלציה גבוה מאוד למול משתנה המטרה (השני הכי גבוה לאחר Type of Travel) ובכל זאת מדד האימפורטנס שלו די נמוך (שישי מהסוף). לכן, לא ניתן לקבוע שקורלציה מהווה אינדיקציה ישירה לחשיבות מאפיין במודל וניתן להניח כי החישובים הנוספים (מבוססי אנטרופיה למשל) שעושה המודל בודקים נדבכים נוספים של סט הנתונים בבחינת המאפיינים שבו.

# רשתות נוירונים- Neural Networks

# הכנת הנתונים להכנסה לרשת נוירונים

בנוסף לשינויים שביצענו בחלק א', ביצענו תהליך קידוד עבור המשתנים קטגוריאליים השונים בסט הנתונים שלנו(ובפרט עבור Age, Arrival Delay, Departure Delay). משמעות הקידוד היא הפיכת המשתנים לקלט מספרי עבור רשת הנוירונים. בנוסף בוצע טיפול במשתנים חסרים וחריגים דרך החלפתם במשתנים מתוך התפלגות שנבנתה עבור כל מאפיין לפי הרשומות התקינות שלו.

# הרצת הרשת בערכי ברירת מחדל

בקונפיגורציה הדיפולטיבית של מודל הMLP- של Sklearn ישנה שכבת כניסת בעלת 81 נוירונים שמייצגים את כל הקומבינציות של המאפיינים הקטגוריאליים ומשתני הדמה שמיצגים את המאפיינים הרציפים. ישנה שכבה אחת חבויה של 10 נוירונים, תהליך האימון של הרשת ייקח לכל היותר 1000 הרצות כדי לייעל את פרמטרי המודל בהתבסס על נתוני האימון. אם קריטריוני ההתכנסות לא יתקיימו בתוך 1000 איטרציות אלו, האימון ייפסק גם אם המודל לא התכנס במלואו. פונקציית האקטיבציה של המודל היא פונקציית relu, שמשמעותה לקית הערך הגבוה בין x ו-0 (max(0,x)) על מנת להבטיח ערכים חיוביים בלבד לערור הנוירונים.

כדי להעריך את דיוק מודל ברירת המחדל נשתמש גם כאן במדד F1 שהוא מדד נפוץ במשימות סיווג בינארי, על מנת להעריך את הביצועים של המודל תוך התחשבות גם בדיוק וגם בזיכרון. קיבלנו שממוצע הדיוק עבור סט בחינה ואימון לרשת הלא מכווננת הוא 0.8861 עבור סט האימון ו-0.8755 עבור סט המבחן. המשמעות היא שעל פי ערכי המדד ניתן לשערך כ-87.5% הצלחה בסיווג הרשומות ל"מרוצה" או "לא מרוצה" בשימוש ברשת הנוירונים.

# כוונון פרמטרים (Hyperparameter- tuning)

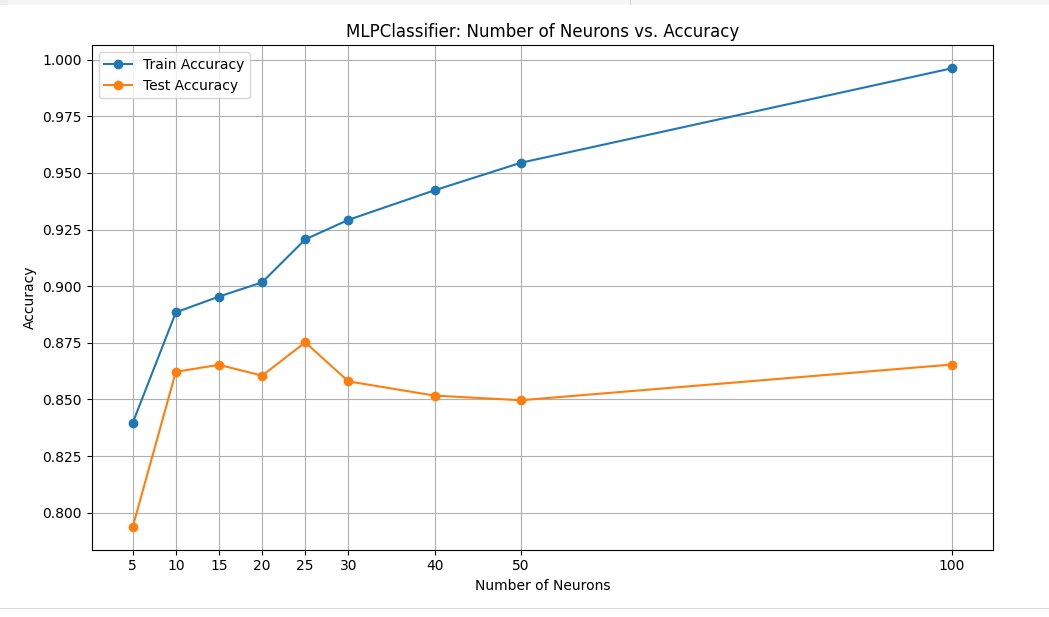
**learning rate:** היפר-פרמטר זה מציין את שיעור הלמידה של הרשת מעדכוני המשקלים. ככל שנגדיל ערך זה, תהיה פחות למידה של הרשת ולהפך. ערך גדול של פרמטר זה יכול להוביל לפספוס של נק' מסוימות בתהליך הלמידה. לכן, ע"י כוונון נרצה למצוא את הערך המתאים ביותר לרשת כך שתתבצע למידה במידה המתאימה, תוך הפחת הסיכוי לדילוג על נקודות מינימום מקומיות.

**neurons:** ערכו של היפר-פרמטר זה עולה כשערך המאפיינים והנתונים עולה ולהפך. שימוש במספר עודף של נוירונים מגדיל את המשקל בכל שכבה ומוביל לסיבוכיות מיותרת ברשת. הקטנת מספר הנוירונים עלול להוביל לכך שלא תתבצע למידה של מאפיינים ונתונים בעלי משמעות. על מנת להגיע לתוצאת דיוק טובה יותר, נצטרך מספר נוירונים המתאים לכמות הנתונים והמאפיינים בסט הנתונים שלנו. נרצה למצוא את מספר הנוירונים המתאים ולכן נבצע כוונון להיפר-פרמטר זה.

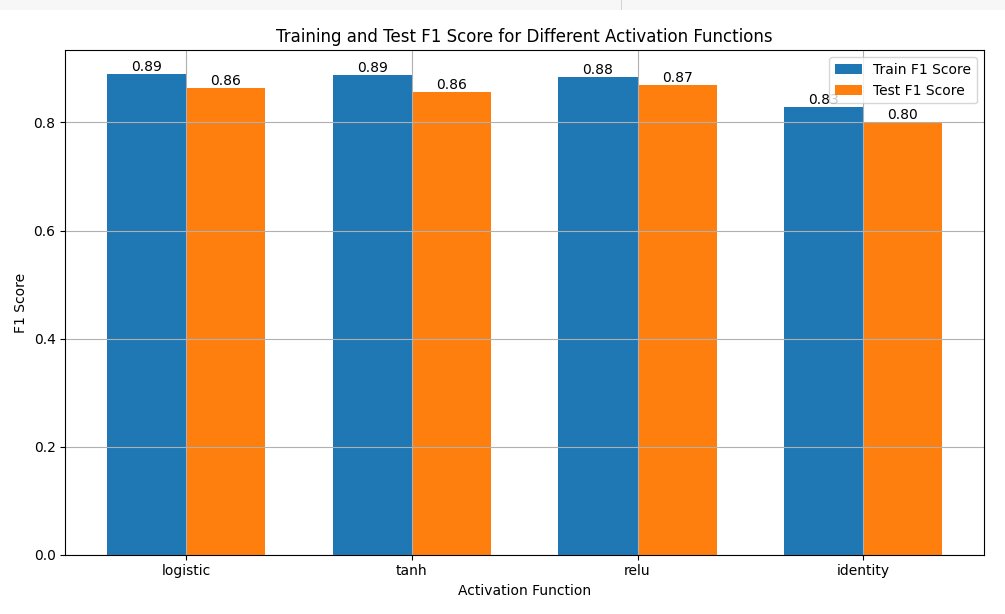
**activation:** פונקציות אקטיבציה הן פונקציות לא לינאריות המוחלות על הפלט של כל נוירון ברשת. הן משמשות להוסיף מורכבות לרשת ולשפר את יכולתה ללמוד דפוסים לא ליניאריים בנתונים. קיימות פונקציות אקטיבציה וביניהן:logistic tanh, relu, identity אותן אנו נבדוק. בחירת פונקציית האקטיבציה הנכונה ברשת נוירונים משפיעה על יכולת הלמידה והביצועים של הרשת.

**solver**: ה-solver הוא הפרמטר שקובע באיזה אלגוריתם אופטימיזציה במהלך אימון רשת הנוירונים שלנו. אלגוריתם האופטימיזציה קובע כיצד מתעדכנות המשקולות המודל. בחירת ה-solver המתאים ברשת נוירונים משפיעה על קצב האימון ועל יעילותו ודיוק התוצאות. הsolver-ים שבדקנו הם sgd, adam ו-lbfgs.

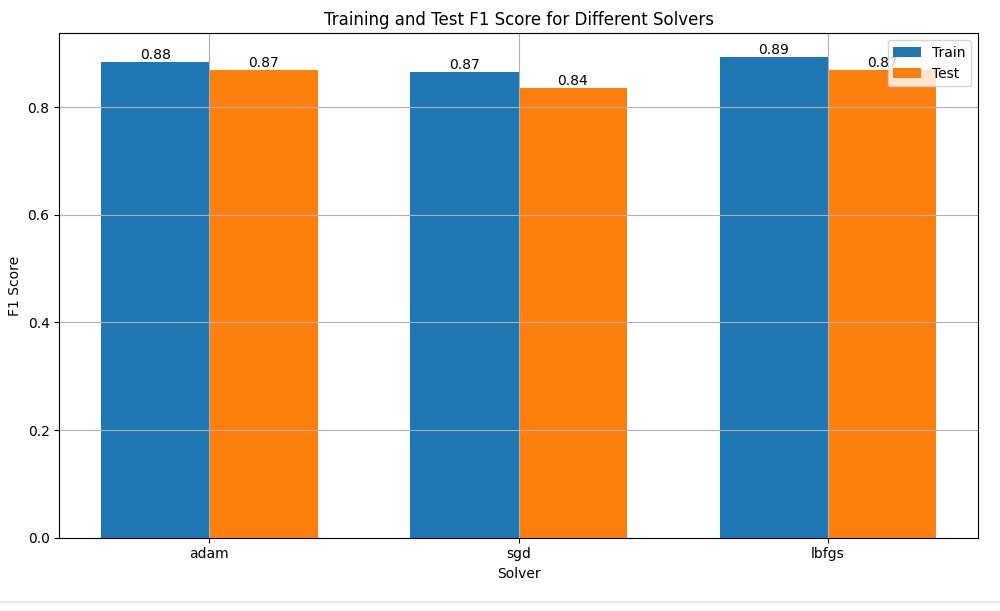
* תהליך כוונון פרמטרים בוצע באמצעות בשיטה איטרטיבית על מנת לחקור את ההשפעה של מספרי נוירונים משתנים על יכולות הלמידה והסיווג של הרשת. טווח מספרי הנוירונים שנבחנו נעים בין 5 ל-100 נוירונים. הגרף המתאר הקשר בין מספר נוירונים לדיוק המודל. ניתן לראות בגרף שקיבלנו, ש-25 נוירונים לכל שכבה נסתרת היא הבחירה העדיפה.



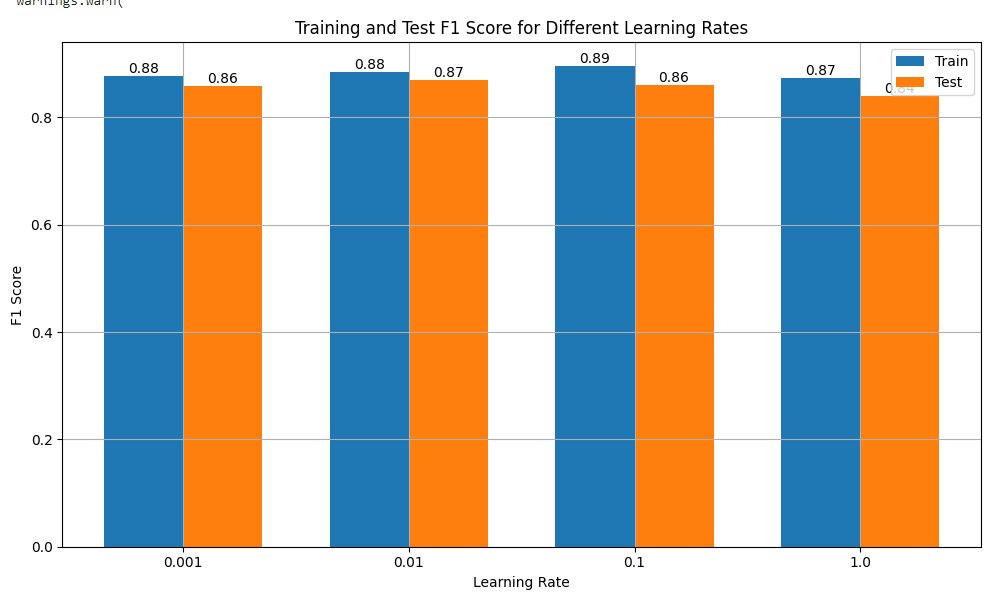
* בתרשים מטה מתוארת השוואה בין ארבעת סוגי פונקציות האקטיבציה. מהתרשים ניתן להסיק שפונקציית האקטיבציה הטובה ביותר היא relu לפי מדד F1.



* כדי לבחור את הSOLVER הטוב ביותר בחנו את ביצועי F1 של כל אחד מהאלגוריתמים. התבוננות בתרשים מעלה כי האלגו' המתאים ביותר הוא lbfgs.



* עבור ההיפר-פרמטר learning rate – השתמשנו שוב בתרשים דומה וגילינו ששיעור הלמידה הטוב ביותר של הרשת לפי מדד F1 הינו 0.1 .



# אימון רשת הנוירונים

* תמונה שמכילה טקסט, גופן, צילום מסך

  התיאור נוצר באופן אוטומטילאחר כוונון הגענו למודל בעל הפרמטרים הבאים: נוירונים בשכבה החבויה- 25 (שכבה בודדת), קצב למידה-0.01, פונקציית אקטיבציה-relu, אלגוריתם נבחר ל-solver: lbfgs.
* תמונה שמכילה טקסט, צילום מסך, גופן

  התיאור נוצר באופן אוטומטילאחר ביצוע 40 הרצות, קיבלנו ציון F1 עם התצורה הטובה ביותר של 97.7% עבור סט האימון  עם הבדל מובהק בתוצאות ו-84.48% על סט בחינה . זהו שיפור קטן יחסית, אך באחוזים גבוהים של דיוק כל שיפור הוא משמעותי, בייחוד כשמדובר בממוצע של הרבה הרצות.

# למידה בלתי מונחת- Unsupervised Learning: Clustering

# הרצת מודל עם ערכי ברירת מחדל

הרצנו את מודל K-means עם ערכי ברירת המחדל והגדרנו את מספר האשכולות לשניים עבור קלסיפיקציה של שתי מחלקות ("מרוצה" ו"לא מרוצה") כפי שנדרש בבעיה שלנו. בחנו את סט הנתונים ויזואלית לפי מאפיינים על מנת לגלות האם ניתן לזהות בשלב מוקדם מקבצים כלשהם בנתונים. הגרפים מתוארים בנספחים (נספח 3 בחינה ויזואלית לזיהוי קבוצות בסט הנתונים (אלגוריתם k-means) לפני ואחרי שיטוח

).

# דיון בתוצאות החלוקה לאשכולות

המדד Adjusted Rand Index (ARI) שהתקבל הוא 0.096. מדד ה-ARI נע בין 1- ל-1, כאשר ערך של 1 מצביע על התאמה מושלמת בין שתי החלוקות וערך של 0 מצביע על כך שהחלוקה היא אקראית לגמרי. ערך שלילי מצביע על כך שהחלוקה גרועה מאקראית. ערך ה-ARI של 0.096 הוא ערך נמוך מאוד, דבר המצביע על כך שההתאמה בין האשכולות שנוצרו על ידי מודל ה-K-Means לבין המחלקות האמיתיות של שביעות הרצון היא נמוכה.

תמונה שמכילה צילום מסך, טקסט

התיאור נוצר באופן אוטומטי

# אימון מחדש עם ערכי k משתנים (סעיף רשות)

נשתמש במדדים שונים של הומוגניות והפרדה (silhouette , davies bouldin). כאשר :

ציון דייויס-באולדן ומדד סילואט מערבבים הומוגניות והפרדה.

תמונה שמכילה צילום מסך, טקסט

התיאור נוצר באופן אוטומטיתמונה שמכילה קו, טקסט, תרשים, עלילה

התיאור נוצר באופן אוטומטיתמונה שמכילה קו, תרשים, עלילה, טקסט

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, קו, תרשים, עלילה

התיאור נוצר באופן אוטומטיאיכות קיבוץ טובה יותר מיוצגת על ידי אינרציה (סכום ריבועי המרחקים של הדגימות למרכז האשכול הקרוב ביותר) נמוכה יותר וציון דייויס-באולדן נמוך יותר עם ציון סילואט גבוה יותר. לאחר שיכלול שלושת המדדים נבחר את מספר האשכולות כך שהוא ייתן תוצאה סבירה עבור המדדים, ולכן נבחר ב-5 אשכולות.

# אימון מודל נוסף- רגרסיה לוגיסטית

בחרנו לסווג באמצעות מודל נוסף-רגרסיה לוגיסטית. מנגנון הסיווג של רגרסיה לוגיסטית משתמש בפונקציה שמחזירה ערך הסתברותי עבור כל מחלקה, כאשר ההחלטה הסופית מתקבלת לפי ערך סף (threshold) שנקבע מראש. הפונקציה הלוגיסטית מתוארת על ידי הנוסחה: *. כאשר z* הוא צירוף לינארי של המאפיינים*.*

בדומה למודלים האחרים שכיווננו, גם ברגרסיה לוגיסטית יש מספר פרמטרים שניתן לכוונן על מנת לשפר את ביצועי המודל. הפרמטרים שבחנו בעת אימון מודל הרגרסיה הלוגיסטית הם:

* פרמטר **C**, הקובע את עוצמת הרגולריזציה של המודל. רגולריזציה היא תהליך שמסייע למנוע overfitting על ידי הוספת קנסות עבור משקלים גדולים במודל. ערכים קטנים מתאימים לעונש רגולריזציה חזק יותר, כלומר מניעת overfitting. ערכים גדולים מתאימים לעונש רגולריזציה חלש יותר, כלומר חתירה להתאמה מדויקת של המודל לנתונים הקיימים.
* פרמטר **penalty**, הקובע את סוג הרגולריזציה שתתבצע:
  + רגולריזציה מסוג **'l1'**, המבוססת על הערך המוחלט של המשקלים, מה שמוביל לקנס על מספר מצומצם יחסית של משקלים גדולים. רגולריזציה זו יכולה לגרום להפוך חלק מהמשקלים ל-אפס.
  + רגולריזציה מסוג **'l2'**,המבוססת על ריבוע המשקלים, מה שמוביל לקנס על מספר רב של משקלים קטנים.
  + **'elasticnet'** : שילוב של רגולריזציה מסוג 'l1' ו-'l2'.
  + **'none'** : ללא רגולריזציה.
* פרמטר **solver**  קובע את האלגוריתם שמשמש לאופטימיזציה. לכל אלגוריתם יש יתרונות וחסרונות, בהתאם לבעיה בה עוסקים. האלגוריתמים שבחנו ב-cv הם: newton-cg, lbfgs, liblinear, sag, ו-saga.
* פרמטר **max\_iter**  קובע את מספר האיטרציות המקסימלי שהאלגוריתם יריץ עד מציאת פתרון. ערכים גבוהים יותר מאפשרים לאלגוריתם לרוץ זמן רב יותר ולשפר את ההתכנסות, אך גם יקרים מבחינת זמן וסיבוכיות חישוב.

הכנת הנתונים עבור מודל הרגרסיה הלוגיסטית כללה:

* קידוד משתנים קטגוריאליים (בוצע בדומה למודלים הקודמים)- רגרסיה לוגיסטית, כמו רוב המודלים, לא יודעת להתמודד עם ערכים קטגוריאליים חופשיים.
* טיפול בערכים חסרים (בוצע בדומה למודלים הקודמים)- רגרסיה לוגיסטית לא יודעת להתמודד עם ערכים חסרים בכלל.
* סטנדרטיזציה (נרמול) של ערכים- רגרסיה לוגיסטית דורשת תהליך כיוון שהמודל רגיש לפערים בגדלי הערכים השונים בין המאפיינים.

לאחר הכנת הנתונים בחנו את המאפיינים באמצעות cv בשיטת K-fold ומצאנו שהמודל הטוב ביותר מוחשב באמצעות אלגוריתם saga, על אף שציפינו שהטוב ביותר יהיה lbfgs כיוון שהוא נחשב מתאים יותר לסט נתונים בגודל שלנו ולסיווג בינארי כמו שאנחנו נדרשים אליו. המודל הטוב ביותר לא דורש רגולריזציה (Penalty: None). ומספר האיטרציות נקבע על ידינו מראש לאלף חזרות (מספר סטנדרטי שלפי הפורומים השונים נחשב עמוק מספיק לסט נתונים בגודל בינוני, והיה לנו נח עם זמן הריצה שלו). ציון F1 של המודל לפי הפרמטרים הללו הוא: 0.8593.

# השוואה בין מודלים

# יתרונות DT ו-MLP למול K-means

עצי החלטה ורשתות נוירונים לרוב יתאימו יותר למשימות סיווג בהשוואה לאלגוריתם K-means מכיוון שהם מתוכננים ללמידה מנוחת בה המודל מאומן עם נתונים מסומנים כדי ללמוד את הקשר בין המאפיינים שנתונים לנו לבין המחלקות אליהן נרצה לסווג את סט הנתונים. לעומת זאת, K-means הוא אלגוריתם לא מונחה המקבץ נתונים על סמך דמיון של תכונות, מה שהופך אותו לפחות יעיל עבור משימות הדורשות קלסיפיקציה מדויקת ככל הניתן.

# השוואת ביצועי המודלים

בהשוואת ביצועי המודלים השונים שבנינו על סט הבחינה שהקצנו מתוך סט האימון שלנו, ניתן לראות שעץ ההחלטה שבנינו הוא בעל הביצועים הטובים ביותר מבין המודלים עם ציון F1 של 0.9072. רשת הנוירונים היא המודל שהציג את הביצועים השניים בטיבם עם מדד F1 של 0.87. מודל הרגרסיה הלוגיסטית שבנינו הציג ציון של 0.58 עבור מדד F1. המודל שהיינו בוחרים הוא עץ ההחלטה שבנינו. מעבר לכך שהוא הציג את מדד F1 הטוב ביותר, הוא גם הציג נתונים טובים במטריצת המבוכה, מה שמצביע על יכולת מאוזנת למדי לזיהוי וקלסיפיקציה בשתי הקטגוריות הקיימות ("מרוצה" ו"לא מרוצה"). לאחר כל הבדיקות שעשינו אנחנו מעריכי שהמודל טוב ויציב ולכן נבחר בו.

# המודל נבחר

# הצגת המודל

המודל שבחרנו להעמיד לתחרות הוא מודל עץ ההחלטה שבנינו, כיוון שהוא כמובן בעל הביצועים הטובים ביותר מבין המודלים שבנינו. הפרמטרים שכיוונו במודל הם:

criterion='entropy', random\_state=42, ccp\_alpha=0.0, class\_weight=None, max\_depth = 10, max\_features=None, max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, min\_samples\_leaf=11, min\_samples\_split=2, min\_weight\_fraction\_leaf = 0.0, splitter = 'best'

# ניתוח תוצאות המודל

כפי שהצגנו בסעיף 1.4 (אימון עץ החלטה), ניתן לראות שהמודל שבנינו מאוזן יחסית בזיהוי וקלסיפיקציה בשתי המחלקות על סט האימון עם שיעור גילויי אמת של 0.92 למחלקה "לא מרוצה" ו-0.9 למחלקה "מרוצה". עבור סט הבחינה המודל הציג שיעור גילויי אמת של 0.95 למחלקה "לא מרוצה" ו-0.88 למחלקה "מרוצה". ניתן לראות שהמודל בעל הטיה קלה למול סט הבחינה ביכולת הקלסיפיקציה בין המחלקות עם עדיפות בביצועי הזיהוי של המחלקה "לא מרוצה", ובכל זאת הביצועים טובים ועץ ההחלטה מציג יכול סיווג גבוהה בכל מקרה.

# נספחים

נספח 1: טבלת הסבר לקידוד ממשתנה קטגוריאלי למשתנה מספרי:

|  |  |
| --- | --- |
| **המשתנה** | **הקידוד שבוצע** |
| Gender | Male: 0, Female: 1 |
| Customer Type | Loyal Customer: 0, disloyal Customer: 1 |
| Type of Travel | Personal Travel: 0, Business Travel: 1 |
| Class | Eco: 1, Eco Plus: 2, Business: 3 |
| Age | לכל bin תיוג במספר עם לקיחת חשיבות לסדר |
| Arrival Delay in Minutes | לכל bin תיוג במספר עם לקיחת חשיבות לסדר |
| Departure Delay in Minutes | לכל bin תיוג במספר עם לקיחת חשיבות לסדר |

נספח 2: הרצות היפר-פרמטרים עבור עץ בינארי:

הרצה מוקדמת באינטרוולים גדולים:

'max\_depth': np.arange(4, 40, 10),  
'min\_samples\_split': np.arange(2, 10, 2),  
'min\_samples\_leaf': np.arange(2, 10, 2),  
'max\_features': ['auto', 'sqrt', 'log2', None],  
'min\_impurity\_decrease': [0.0, 0.01, 0.05]

k : 10 {'max\_depth': 14, 'max\_features': None, 'min\_impurity\_decrease': 0.0, 'min\_samples\_leaf': 8, 'min\_samples\_split': 2}

k : 10 f1 \_score : 0. 0.8130824656473091{'ccp\_alpha': 0.0, 'class\_weight': None, 'criterion': 'entropy', 'max\_depth': 10, 'max\_features': None, 'max\_leaf\_nodes': None, 'min\_impurity\_decrease': 0.0, 'min\_samples\_leaf': 8, 'min\_samples\_split': 2, 'min\_weight\_fraction\_leaf': 0.0, 'random\_state': 42, 'splitter': 'best'}

הרצה באינטרוולים קטנים סביב הטווח שזוהה בהרצה המוקדמת:

'max\_depth': np.arange(8, 18, 1),

'min\_samples\_split': np.arange(0, 4, 1),

'min\_samples\_leaf': np.arange(6, 10, 1),

max\_features'= None,

'min\_impurity\_decrease'= 0.0

k : 11 {'max\_depth': 10, 'min\_samples\_leaf': 8, 'min\_samples\_split': 2}

k : 11 f1 \_score : 0.9072753209700427{'ccp\_alpha': 0.0, 'class\_weight': None, 'criterion': 'entropy', 'max\_depth': 10, 'max\_features': None, 'max\_leaf\_nodes': None, 'min\_impurity\_decrease': 0.0, 'min\_samples\_leaf': 8, 'min\_samples\_split': 2, 'min\_weight\_fraction\_leaf': 0.0, 'random\_state': 42, 'splitter': 'best'}

נספח 3 בחינה ויזואלית לזיהוי קבוצות בסט הנתונים (אלגוריתם k-means) לפני ואחרי שיטוח

תמונה שמכילה טקסט, תרשים, צילום מסך, מקביל

התיאור נוצר באופן אוטומטי