דו"ח פרויקט - חלק ב' (תשפ"ד סמסטר ב')

קבוצה 6

שחר לוי אטיאס 319042685

עדי נעמתי 209353838

תמיר לבנטר 206557456

יובל עמית 208956789

תוכן עניינים

הכנת נתונים לאימון ובחינה

בחרנו לבצע את חלוקת הנתונים לסט הבחינה בצורה אקראית, המבטיחה שכל התצפיות מקבלות סיכוי שווה להיכלל בו, ובכך מבטיחה מדגם מייצג והוגן. גודל סט הבחינה נקבע מתוך התחשבות בכך שמספר התצפיות הכולל הוא 9000. החלטנו להקצות כ-10% מהנתונים לסט הבחינה, כלומר כ-900 תצפיות. כמות זו מספקת ייצוגיות וביצועי מודל מהימנים, ובאותה עת, שומרת על כמות מספקת של נתונים לאימון, מבלי לפגוע בתהליך האימון או לגרום ל-overfitting . בנוסף, לדעתנו חלוקה של כ-10% מהנתונים לסט הבחינה משמרת איזון אופטימלי בין הצורך בסט בחינה מייצג לבין שמירת רוב הנתונים לאימון המודל.

בנוסף, כחלק מהכנת הנתונים בחרנו לבטל את הדיסקריטיזציה שביצענו על משתנה גיל משום גרם לאיבוד מידע רב בהתפלגות הגילאים.

עץ החלטה (Training Tree)

1. **הכנת הנתונים:**

כדי להכין את הנתונים כך שיתאימו למודל עץ ההחלטה ביצענו מספר שלבים חשובים לניקוי, עיבוד והמרת הנתונים לפורמט שמתאים ללמידת מכונה. הנה תיאור קצר של הפעולות שביצענו ומשמעויותיהן:

שלב 1: ניקוי נתונים

- ניקוי ערכים בעמודות בדידות: השתמשנו בפונקציה clean_column_discrete לניקוי ערכים בעמודות בדידות כמו (Gender, Customer Type) כך שיישארו רק הערכים בטווח ההיגיון, והמרת ערכים לא בטווח ההיגיון כ-NaN. משמעות: פעולה זו מוודאת שכל הנתונים בעמודות הבדידות הם ערכים תקפים שתואמים לקטגוריות שנבחרו.
- ניקוי ערכים בעמודות רציפות: השתמשנו בפונקציה clean_column_continuous לניקוי ערכים בעמודות רציפות כמו (Age, Flight Distance) כך שיישארו רק הערכים בטווח ההיגיון בתחום שנבחר. משמעות: פעולה זו מוודאת שכל הנתונים בעמודות הרציפות הם ערכים תקפים ונמצאים בטווחים המתאימים.
- הסרת שורות עם ערכים חסרים בעמודת המטרה: הסרנו שורות שבהן יש ערכים חסרים בעמודת המטרה'satisfaction' .

משמעות: פעולה זו מוודאת שעמודת המטרה שלמה ואין בה ערכים חסרים, מה שמאפשר למודל ללמוד בצורה נכונה את הנתונים.

- הסרת שורות עם ערכים חסרים ביותר מ-4 עמודות: הסרנו שורות שבהן יש ערכים חסרים ביותר מ-4 עמודות. משמעות: פעולה זו מוודאת שכל הרשומות אכן תורמות ללמידת התנהגות הנתונים ולאימון המודל.

שלב 2: מילוי ערכים חסרים

- השלמת ערכים חסרים בעמודות רציפות עם הממוצע: השלמת ערכים חסרים בעמודות רציפות כמו Age, Flight Distance עם הממוצע של כל עמודה. משמעות: פעולה זו מסייעת להשלמת הנתונים ובנוסף שומרת ממוצע הנתונים המקוריים וכך מונעת פגיעה משמעותית בנתונים.

- השלמת ערכים חסרים בעמודות רציפות עם החציון: השלמנו ערכים חסרים בעמודות נוספות עם החציון של כל עמודה. משמעות: השלמת הנתונים תוך שימרה על המרכזיות של הנתונים.
- השלמת ערכים חסרים בעמודות קטגוריאליות עם הערך הנפוץ ביותר: השלמנו ערכים חסרים בעמודות קטגוריאליות עם הערך הנפוץ ביותר בכל עמודה. משמעות: השלמת הנתונים תוך שמירה על התפלגות הערכים הקטגוריאליים.

שלב 3: המרת נתונים

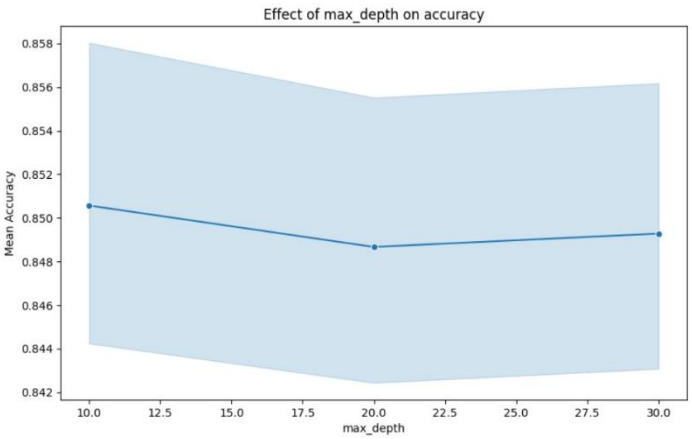
- שימוש ב-One-Hot Encoding: השתמשנו ב-One-Hot Encoding כדי להמיר עמודות קטגוריאליות לערכים בינאריים. משמעות: פעולה זו מוודאת שכל הנתונים קטגוריאליים יהיו במספריים ולא במחרוזות, מה שמאפשר למודל ללמוד מהם.
- נרמול הנתונים: נרמלנו את הנתונים בעזרת StandardScaler כך שכל העמודות יהיו בעלות ממוצע 0 וסטיית תקן 1. משמעות: פעולה זו מוודאת שהמודל יקבל נתונים בנורמה אחת וזאת על מנת שלכל הנתונים תהיה השפעה זהה על המודל ושלא יהיו תכונות שישלטו על תהליך הלמידה.

2. בניית עץ החלטה:

עץ החלטה מלא שאומן באמצעות סט האימון הראה תוצאות מושלמות על סט האימון עם Accuracy, Precision, Recall, F1 של 1.000. לעומת זאת, על סט הבדיקה התוצאות היו נמוכות יותר: Accuracy של 0.8710, Precision של 0.867580, Recall של 0.842572 ומדד F1 של 0.854893. ניתן להסיק מתוצאות אלה שעץ ההחלטה המלא מתאים יתר על המידה לסט האימון (Overfitting), ולכן ביצועיו על סט הבדיקה אינם מיטביים. עץ החלטה מלא לא תמיד יביא לתוצאה כזו על סט האימון, אך כאשר הוא מבצע התאמה מושלמת לנתוני האימון, זה לרוב מצביע על Overfitting שיפגע בביצועים על נתונים חדשים.

3. תהליך כוונון הפרמטרים:

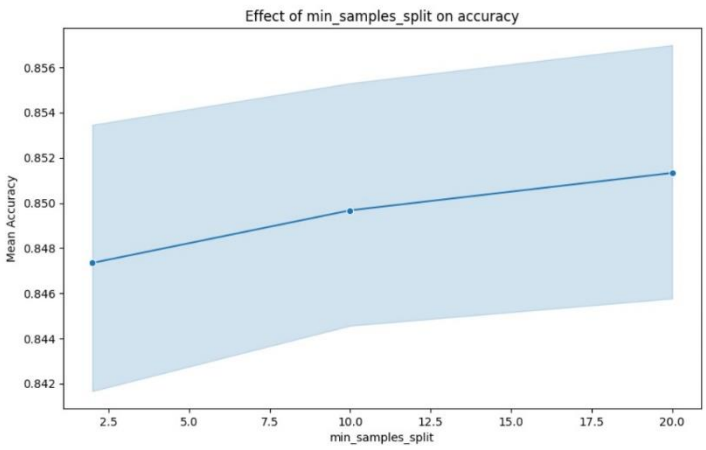
במהלך תהליך "כוונון פרמטרים" (Hyperparameter Tuning) למודל עץ ההחלטה, השתמשנו ב-GridSearchCV כדי למצוא את הקונפיגורציה המיטבית. להלן הפרמטרים שבחרנו לכוונן:



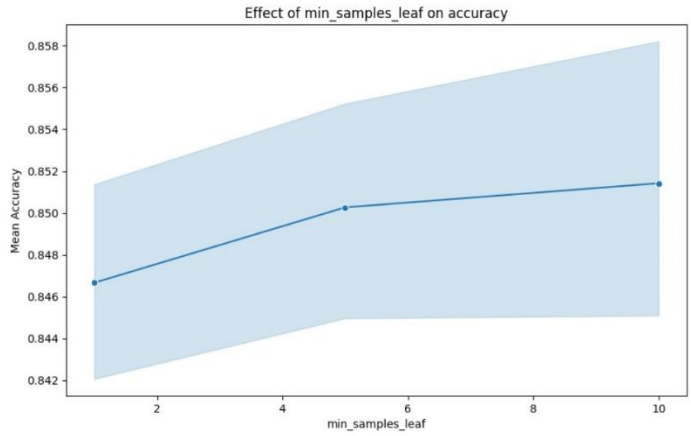
max_depth (עומק מרבי של העץ): המוטיבציה לכוונן פרמטר זה היא למנוע overfitting כאשר העץ עמוק מדי ולמנוע underfitting כאשר העץ רדוד מדי. הגדלת הערך תאפשר לעץ ללמוד יותר פרטים, אך עלולה להוביל ל-overfitting הקטנת הערך תצמצם את כמות הפרטים שהעץ יכול ללמוד, מה שעשוי להוביל ל-underfitting.

min_samples_split (מספר המינימום של דוגמאות הדרושות לפיצול צומת): המוטיבציה היא לשלוט באיזה קלות העץ יכול לפצל את הצמתים. הגדלת הערך תקטין את מספר הפיצולים ויכולה למנוע overfitting בכך

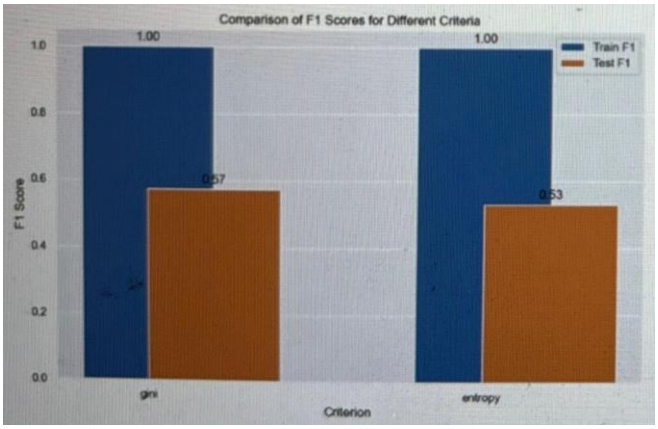
שהיא דורשת יותר דוגמאות לפיצול צומת, מה שגורם לעץ להיות פחות מורכב. הקטנת הערך תאפשר פיצולים רבים יותר, מה שיכול להוביל ל-overfitting אם הצמתים מפוצלים יותר מדי על סמך מעט דוגמאות.



<u>min_samples_leaf (מספר המינימום של דוגמאות הדרושות בכל עלה)</u>: המוטיבציה היא להבטיח שכל עלה יכיל מספיק דוגמאות כדי לספק תחזית אמינה. הגדלת הערך תקטין את מספר העלים ותמנע overfitting, הקטנת הערך תגדיל את מספר העלים, אך עלולה להוביל ל-overfitting.



<u>הקריטריון לפיצול 'gini' או 'entropy'</u>: המוטיבציה היא לבחור את הקריטריון הטוב ביותר עבור סט הנתונים.



כל אחד מהקריטריונים משתמש בגישה שונה למדידת טוהר הצומת ולבחירת הפיצול הבא. קריטריון gini מודד את טוהר הצומת על ידי מדידת השונות בין המחלקות בצומת, בעוד שקריטריון entropy מודד את טוהר הצומת על ידי מדידת הוודאות בין המחלקות בצומת. הבחירה בין הקריטריונים יכולה להשפיע על מבנה העץ והביצועים שלו.

4. אימון עץ החלטה:

לאחר כוונון הפרמטרים, המודל המותאם של עץ ההחלטות הראה את עם הקונפיגורציה הטובה ביותר הראה את מדדי הביצוע הבאים:

```
Tuned Decision Tree On the train set
Accuracy: 0.916604
Precision: 0.909356
Recall: 0.897583
F1 Score: 0.903431
```

<u>על סט האימון</u>: Accuracy דיוק של 0.919605 , Precision דיוק של 0.909356 , Recall שליפה של 0.897583 ו F1 מדד של 0.903431

```
Tuned Decision Tree On the test
Accuracy: 0.895000
Precision: 0.894977
Recall: 0.869180
F1 Score: 0.881890
```

<u>על סט הבחינה</u>: Accuracy דיוק של 0.895000 , Precision דיוק של 0.894977 , Recall שליפה של 0.869180 ו F1 מדד של 0.881890

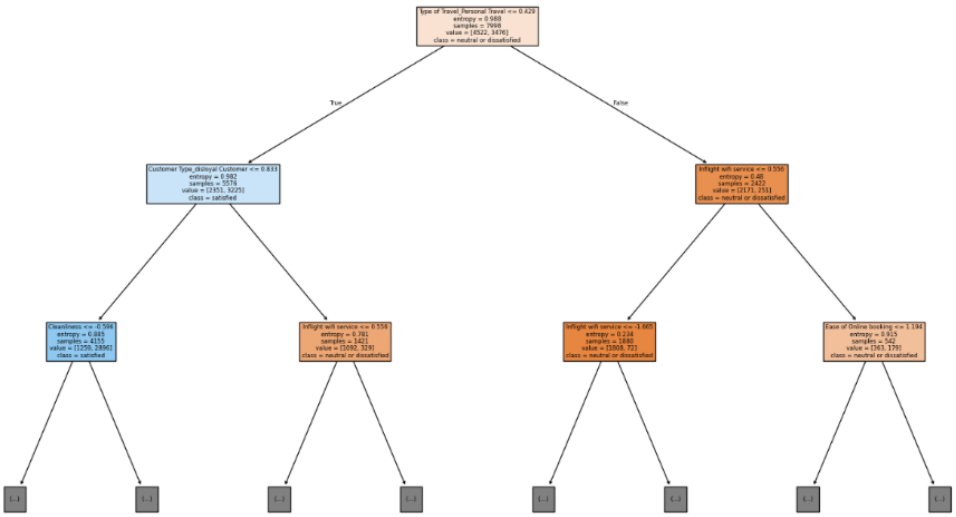
ההבדל בין התוצאות של העץ המלא לבין התוצאות של העץ המתואם מעיד על שיפור בביצועים הכלליים של המודל. בעץ המלא, היה Overfitting לסט האימון, דבר שהוביל לביצועים נמוכים יותר על סט הבחינה. בעץ המתואם, המדדים על סט האימון וסט הבדיקה מתקרבים יותר זה לזה, דבר המעיד על מודל כללי יותר שמתמודד טוב יותר עם נתונים חדשים.

<u>גרף העץ שהתקבל ותובנות</u>:

הקונפיגורציה המיטבית כוללת עומק מרבי של 10, min_samples_split של 25, min_samples_leaf של 6, ו-Criterion של Entropy.

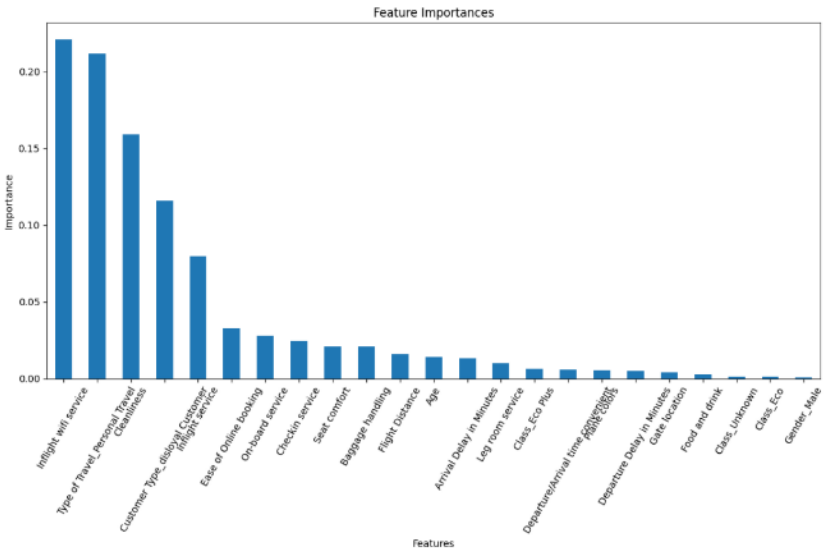
```
Tuned Decision Tree
Best Parameters: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 6, 'min_samples_split': 25}
```

שורש העץ הוא המאפיין המשמעותי ביותר, המאפיין הראשון שפוצל בעץ הוא " Type of Travel_Personal Travel", מה שמראה כי סוג הנסיעה (נסיעה אישית לעומת נסיעה עסקית) הוא גורם משמעותי בקביעת שביעות רצון הלקוחות. מאפיינים נוספים חשובים: "Inflight wifi service" מראה כי שירות ה-WiFi במהלך הטיסה תורם לשביעות הרצון. ניקיון המטוס "Cleanliness" מראה כי ניקיון המטוס משפיע על שביעות רצון הלקוחות.



חשיבות המשתנים- מעץ ההחלטה ניתן לראות כי המאפיינים שצוינו הם המשמעותיים ביותר בקביעת שביעות רצון הלקוחות, ממצאים אלו מתיישבים גם עם ניתוח החשיבות של המשתנים, ומדגישים את הצורך בשיפור תחומים אלו לשיפור שביעות רצון הלקוחות. בעוד שמאפיינים אחרים הם פחות משמעותיים. התובנות הללו עוזרות להבין אילו גורמים חשובים יותר עבור הלקוחות ויכולות לסייע בשיפור השירות ושביעות הרצון הכללית. הפונקציה feature_importances של המודל DecisionTreeClassifier מספקת את החשיבות של כל משתנה בתהליך קבלת ההחלטות של העץ. החשיבות מחושבת על פי הירידה באי-הטוהר (impurity) בכל צומת, ומחושבת כממוצע משוקלל על פני כל הצמתים בהם המשתנה השתמש.

מהתוצאות עולה כי המאפיינים החשובים ביותר הם:



1. Inflight wifi service (22%)
2. Type of Travel_Personal Travel (21%)
3. Cleanliness (16%)
4. Customer Type_disloyal Customer (12%)

תוצאות אלו מתיישבות עם המסקנות מהסעיף הקודם, המדגישות את החשיבות הרבה של שירות ה-WiFi בטיסה, סוג הנסיעה (אישית או עסקית), ניקיון המטוס ונאמנות הלקוח בקביעת שביעות הרצון של הלקוחות. מאפיינים אלו הם הבולטים ביותר בתרומתם להחלטות של המודל והם מהווים חלק משמעותי בניבוי שביעות רצון הלקוחות.

רשת נוירונים (Neural Networks)

1. הכנת הנתונים:

כחלק מהכנת הנתונים להכנסה לרשת הנוירונים ביצענו נרמול נתונים. על מנת לבצע את הנרמול הטוב ביותר עבור הנתונים שלנו, בדקנו נרמול בעזרת StandardScaler וגם בעזרת MinMaxScaler.עבור נרמול בעזרת StandardScaler קיבלנו דיוק של 85% על סט הבחינה-

```
Default neural network On the test set
Accuracy: 0.874000
Precision: 0.868481
Recall: 0.849224
F1 Score: 0.858744
```

בעזרת MinMaxScaler קיבלנו דיוק של 56% על סט הבחינה-

```
Tuned neural network On the test set
Precision: 0.652047
Recall: 0.494457
F1 Score: 0.562421
```

על כן, נרמול בעזרת StandardScaler הוא הנרמול הנכון ביותר עבור סט הנתונים.

2. אימון ובחינת רשת נוירונים על בסיס ערכי המחדל

הנתונים שהתקבלו מאימון ובחינה של הרשת בערכי המחדל:

```
Number of neurons in the input layer: 23
Number of hidden layers: 1
Number of neurons in each hidden layer: 100
Activation function: relu
```

```
Default neural network On the train set
Accuracy: 0.998625
Precision: 0.998561
Recall: 0.998274
F1 Score: 0.998417
```

```
Default neural network On the test set
Accuracy: 0.874000
Precision: 0.868481
Recall: 0.849224
F1 Score: 0.858744
```

<u>משמעות הקונפיגורציה</u>:

- <u>מספר נוירונים בשכבת הכניסה</u>:

המשמעות היא שבנתונים יש 23 תכונות (features). כל נוירון בשכבת הכניסה מייצג תכונה אחת מהקלט. מספר זה נקבע לפי מספר העמודות בנתונים המנורמלים לאחר כל פעולות ההכנה.

- <u>מספר שכבות חבויות</u>:

הפרמטר hidden_layer_sizes מציין שיש שכבה חבויה אחת. שכבות חבויות נמצאות בין שכבת הכניסה (input layer) לשכבת היציאה (output layer), ותפקידן ללמוד ייצוגים מורכבים יותר של הנתונים.

- <u>מספר נוירונים בכל שכבה חבויה</u>:

השכבה החבויה מכילה 100 נוירונים. מספר זה נקבע על פי הפרמטר hidden_layer_sizes שנקבע כ-(100). נוירונים בשכבות החבויות מסייעים למודל ללמוד דפוסים מורכבים יותר בנתונים.

- <u>פונקציית האקטיבציה היא relu</u>:

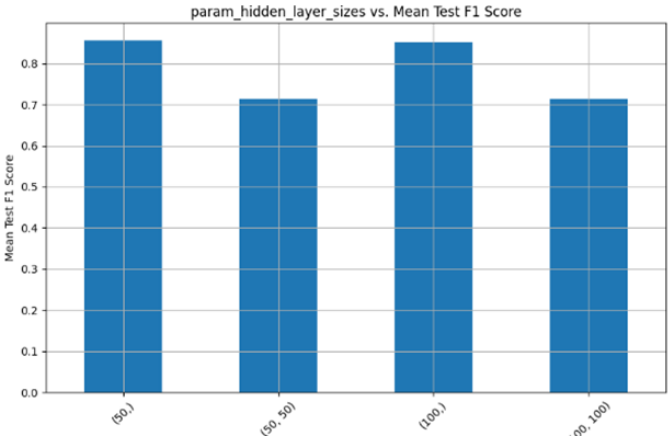
פונקציית relu מחזירה את הקלט אם הוא חיובי, ואפס אם הוא שלילי.

3. תהליך כיוונון הפרמטרים למציאת הקונפיגורציה המיטבית ביותר עבור סט הנתונים:

כיוונון הפרמטרים נועד על מנת למצוא את הפרמטרים המתאימים ביותר שיביאו את המודל לחיזוי הטוב ביותר. לכל סט נתונים יש התנהגות ומרכבות מסוימת, ולכן צריך להתאים את ההיפר פרמטרים על מנת שנוכל לבנות מודל שמתאים להתנהגות הנתונים. על מנת לבצע כיוונון פרמטרים למציאת הקונפיגורציה הטובה ביותר בחרנו להשתמש בכלי GridSearchCV. הכלי מבצע חיפוש על ידי הצלבת כל השילובים האפשריים של ההיפר-פרמטרים שהוגדרו מראש ומוצא את הסט האופטימלי של היפר-פרמטרים שמספקים את הביצועים הטובים ביותר.

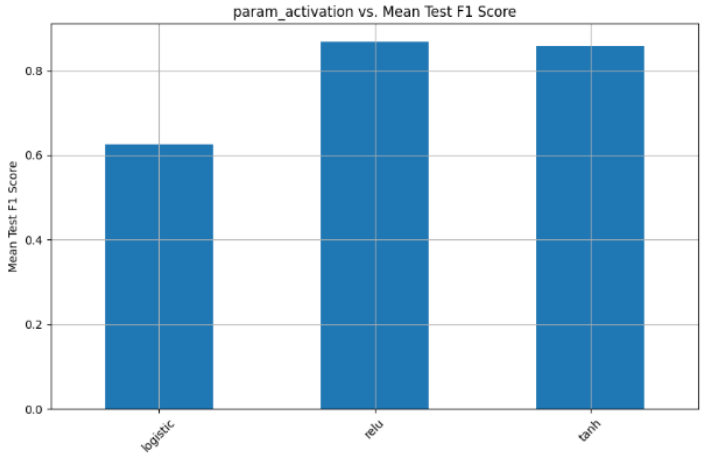
- <u>כיוונון hidden_layer_sizes:</u>

בחרנו לכוונן היפר-פרמטר זה מפני שמורכבות סט הנתונים היא חלק חשוב בבניית המודל. סט נתונים בעל התנהגות מורכבת מצריך מספר שכבות נוירונים גדול יותר על מנת ללמוד את המורכבות של הנתונים וכך לאפשר דיוק בחיזוי, ובכך, למנוע חוסר התאמת המודל לנתונים. לעומת זאת, סט בעל מורכבות נמוכה לא יצטרך מספר שכבות נוירונים גדול על מנת ללמוד את ההתנהגות של הנתונים, ואף מצב כזה יכול להביא את המודל להתאמת יתר לסט הנתונים. אנו שואפים למצוא את מספר השכבות האופטימלי למודל כך שלא ייווצר מצב של התאמר יתר או חוסר התאמה לסט הנתונים. בכוונון הפרמטרים אפשרנו ל-grid search לבחור בין מספר קומבינציות לשכבות [(50,), (100,), (50, 50), (100,)], (100). מספר השכבות שהתקבל הינו: (50,). משמעות התוצאה היא שיש שכבה אחת עם 50 נוירונים, אנו מסיקים כי התנהגות הנתונים פשוטה יחסית ולכן אין צורך במספר שכבות גדול יותר. מהגרף ניתן לראות שתוצאות הביצועים בין ש50 שכבות ל100 שכבות מאוד צמוד.



- <u>כיוונון activation:</u>

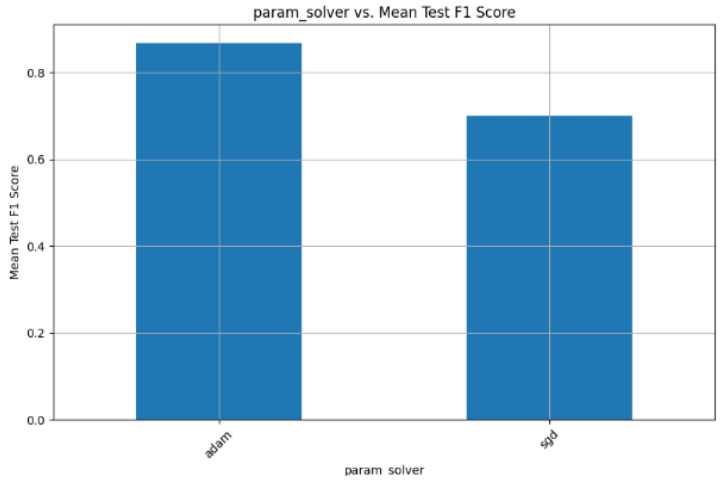
בחרנו לכוונן היפר-פרמטר זה מפני שאופן ביצוע למידת המודל מושפע מפונקציית האקטיבציה. לכל פונקציית אקטיבציה יש אופן לימוד דפוסים שונה. למשל, פונקציית ReLU ידועה כיעילה מאוד עבור רשתות עמוקות ומתאימה למגוון רחב של בעיות, בעוד ש-tanh עשויה להתאים יותר למקרים בהם הנתונים הם ביחס ניגודי או כשיש צורך בערכים בטווח רחב יותר. בחירת פונקציית אקטיבציה אשר מתאימה פחות לסט הנתונים



יכולה לגרום לכך שהמודל יתקשה לזהות דפוסים מסוימים בנתונים ויהיה פחות יעיל בחיזוי. מהגרף ניתן לראות כי פונקציית Logistic הכי פחות מתאימה ללמידת סט הנתונים, ואילו פונקציות ReLU ו-tanh מתאימות יותר. עם זאת, פונקציית האקטיבציה ReLU מספקת את הביצועים הטובים ביותר, עם F1 Score ממוצע גבוה יותר בהשוואה לפונקציות האחרות. השימוש ב-ReLU אפשר לרשת לזהות דפוסים מורכבים בצורה מדויקת יותר, מה שהוביל לביצועים משופרים הן על סט האימון והן על סט הבחינה.

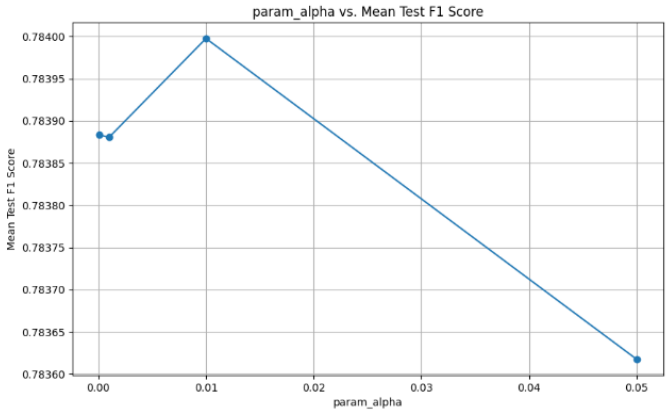
- <u>כיוונון solver:</u>

בחרנו לכוונן היפר-פרמטר זה מפני שהוא משפיע על אופן עדכון המשקלים של נוירונים כדי למזער את פונקציית ההפסד. בכוונון הנתונים נתנו ל- grid search לבחון את האלגוריתמים Adam ו- SGD. בחירת אלגוריתם שלא מתאים לסט הנתונים יכול להשמיע על זמן האימון בכך שהוא יגרום להתכנסות איטית יותר אל עבר הפתרון מפני שהמשקלים לא יצליחו להתייצב וכך מספר האיטרציות יהיה גדול. אנו רוצים לשאוף להתכנסות מהירה אל עבר הפתרון וזמן ריצה קצר ויעיל.



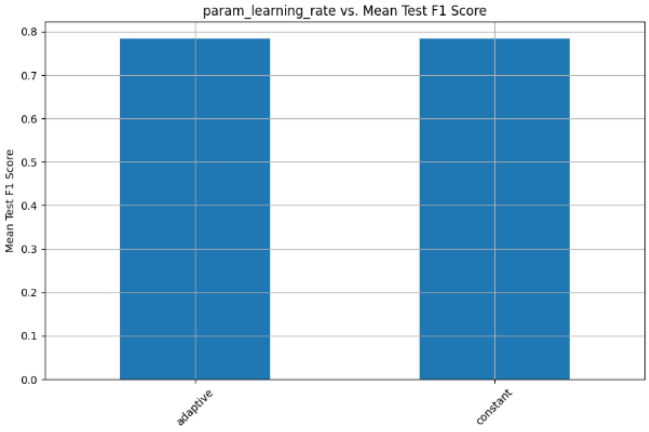
- <u>כיוונון alpha:</u>

בחרנו לכוונן היפר-פרמטר זה מפני שהוא עוזר למנוע התאמת יתר לסט הנתונים בכך שהוא מוסיף "קנס" על המשקלים של המודל ובכך מוודא שהמודל לא מפתח תלות בנתוני האימון ויבצע הכללה טובה יותר על נתונים חדשים. בחירת alpha קטנה תביא לכך שהרגולריזציה היא פחות חזקה, ולכן המודל יהיה חופשי יותר להתאים את עצמו לנתוני האימון ויש סיכוי להגיע להתאמת יתר. בערך גדול של alpha הרגולריזציה חזקה יותר, ולכן המודל יכוון למשקלים קטנים יותר. המשמעות היא שהמודל ינסה לשמור על פשטות ולהימנע מהתאמה מוגזמת לנתוני האימון. אם הalpha גדולה מידיי המודל עלול להפוך לפשוט מידיי ועשוי להביא לחוסר התאמה. השאיפה היא למצוא את הערך האופטימלי של alpha שמציע את האיזון הטוב ביותר בין התאמה יתרה לתת-התאמה. נראה כי הערך של alpha בסביבות 0.01 הוא הערך האופטימלי שמניב את F1 Score הגבוה ביותר. זה מצביע על כך שכמות מסוימת של רגולריזציה היא חשובה כדי להשיג ביצועים טובים יותר במודל.



- <u>כיוונון learning_rate:</u>

בחרנו לכוונן את ההיפר-פרמטר זה מפני שהוא קובע את גודל השינוי במשקלים של המודל בכל איטרציה של האימון. כאשר קצב הלמידה קבוע (constant), השינוי במשקלים נשאר עקבי לאורך כל תהליך האימון. קצב למידה נמוך מדי עלול להוביל להתכנסות איטית ולא מספקת, בעוד שקצב גבוה מדי עלול לגרום למודל לדלג מעל נקודת המינימום ולא להתכנס כראוי. מצד שני, קצב למידה מותאם (adaptive) מתחיל בערך קבוע אך מתעדכן במהלך האימון, כך שקצב הלמידה יורד כאשר המודל מפסיק להשתפר. זה מאפשר למודל להתחיל בלמידה מהירה ולהתכנס בצורה מדויקת יותר בשלב מאוחר יותר. מהתוצאות המוצגות בגרף ניתן לראות שאין הבדל משמעותי בין השימוש בקצב למידה קבוע לבין קצב למידה מותאם, כאשר שני הפרמטרים מניבים F1 Score דומה. הדבר מצביע על כך שהמודל שלך יציב ומתכנס בצורה טובה עם שני סוגי קצבי הלמידה.



4. הקונפיגורציה שנבחרה בתהליך כיוונון ההיפר-פרמטרים היא:

```
Best parameters found: {'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes': (50,), 'learning_rate': 'constant', 'solver': 'adam'}
Best cross-validation score: 0.8898961086117044
```

מספר השכבות החבויות: 1

מספר הנוירונים בכל שכבה חבויה: 50

פונקציית אקטיבציה: relu

Alpha: 0.001

קצב למידה- constant

Solver: adam

הערכת ביצועים על סט האימון וסט הבדיקה מציגה את המדדים הבאים:

```
Tuned neural network On the train set
Precision: 0.913478
Recall: 0.920311
F1 Score: 0.916882
```

```
Tuned neural network On the test set
Precision: 0.895556
Recall: 0.893570
F1 Score: 0.894562
```

התוצאות על סט האימון מצביעות על כך שהמודל מתפקד בצורה מצוינת, עם Precision, Recall, F1 Score גבוהים, מה שמעיד על כך שהמודל מצליח לזהות ולסווג למחלקות בצורה טובה מאוד.

הביצועים על סט הבחינה גם הם טובים מאוד, אך מעט נמוכים יותר מאלה שהושגו על סט האימון. ה-F1 Score של 0.894562 מעיד על איזון טוב בין Precision ו-Recall.

מהתוצאות אנו מסיקים כי המודל מתאים לחיזוי טוב של הנתונים. הירידה הקלה בביצועים בין סט האימון לסט מובעת מהסיבה העיקרית שהמודל אומן על סט האימון ונבחן לאחר מכן

על נתונים חדשים שהוא לא ראה במהלך האימון. המודל מציג ביצועים עקביים בשני סוגי הסטים (אימון ובחינה), דבר שמצביע על כך שהקונפיגורציה שנבחרה מוצלחת ומביאה לתוצאות טובות.

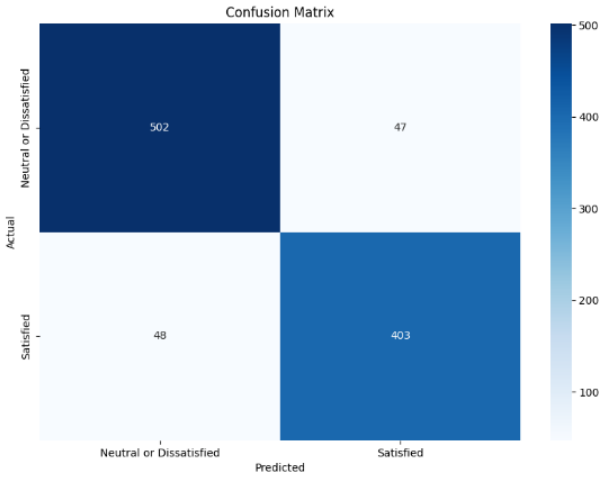
מטריצת סיווגים:

502 תצפיות סווגו נכון כ- "Neutral or Dissatisfied".

403 תצפיות סווגו נכון כ- "Satisfied"

47 תצפיות סווגו לא נכון כ- "Satisfied" למרות שהן היו "Neutral or Dissatisfied"

48 תצפיות סווגו לא נכון כ- "Neutral or Dissatisfied" למרות שהן היו "Satisfied".



<u>Unsupervised Learning - Clustering</u>

1. **הרצת מודל K-means עם ערכי ברירת המחדל על סט האימון:**

כדי להריץ את מודל K-means נצטרך למצוא את מספר האשכולות המתאים. בחרנו בערך K=2 מכיוון שיש לנו 2 אפשרויות לחלוקה (satisfied/ neutral or dissatisfied).

2. **מרכזי האשכולות (Cluster Centers):**

- האשכול הראשון (Cluster 0) מאופיין בערכים שליליים ברוב התכונות.
- האשכול השני (Cluster 1) מאופיין בערכים חיוביים ברוב התכונות.

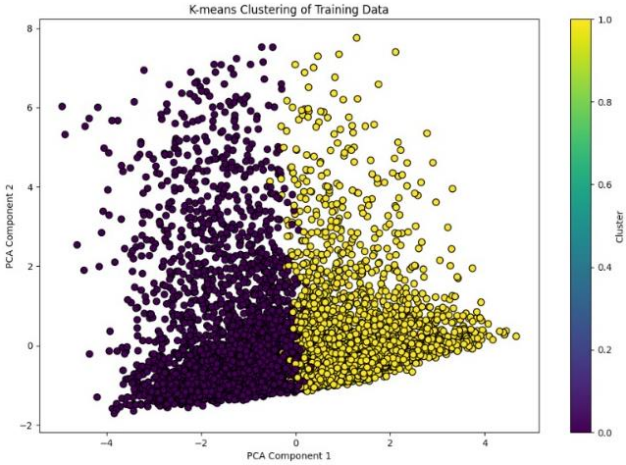
ערך ה-Inertia הוא 171544.55794366015. ערך זה מייצג את מידת הפיזור של הנקודות סביב מרכזי האשכולות. ככל שהערך נמוך יותר, כך האשכולות צפופים יותר סביב המרכזים שלהם.

<u>נבחן את טיב ההתאמה בין האשכולות למחלקות:</u>

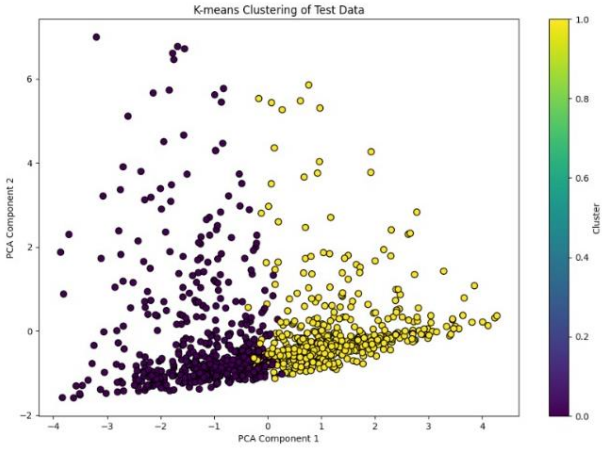
- מדדי ה-ARI (Adjusted Rand Index) וה-NMI (Normalized Mutual Information) עבור סט האימון הם 0.273 ו-0.209 בהתאמה, ועבור סט הבדיקה הם 0.287 ו-0.221 בהתאמה. מדדים אלה מצביעים על התאמה נמוכה יחסית בין האשכולות שנוצרו ע"י ה-K-means לבין המחלקות המקוריות.
- הומוגניות (Homogeneity) ו-V-Measure מצביעים גם הם על טיב התאמה נמוך, אם כי ההתאמה בסט הבדיקה מעט טובה יותר מזו של סט האימון.

נבחן את הגרפים הבאים:

גרף 1 - K-means של סט האימון:



גרף 2 - K-means של סט הבחינה:



בשני הגרפים כל נקודה מייצגת תצפית והצבעים השונים מייצגים את האשכולות השונים. הנתונים מצביעים על כך שהמדדים השונים לבדיקת טיב ההתאמה בין האשכולות למחלקות אינם גבוהים, אך בגרפים נראית הפרדה יחסית ברורה בין האשכולות. ייתכן שזה נובע מכמה סיבות:

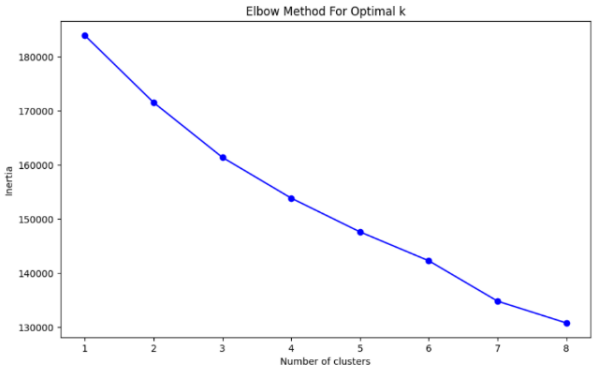
א. פשטות המדדים: מדדים כמו Adjusted Rand Index (ARI) ו- Normalized Mutual Information (NMI) בודקים התאמה על פי קריטריונים מחמירים. ייתכן שהאשכולות מופרדים היטב מבחינת התצפיות, אך המדדים לא מתייחסים לפרמטרים נוספים כמו מרחק ביניהם וגודל כל אשכול.

ב. סף גבול: המדדים כוללים ערכי סף שמגדירים כמה השוואות. במקרים מסוימים, כאשר התפלגות האשכולות אינה אחידה, המדדים לא מצליחים לשקף את ההפרדה בצורה מלאה.

ג. ממדיות נתונים: הפחתת הממד בעזרת PCA לשני ממדים משפיעה על כל המידע המקורי של הנתונים. ייתכן שישנן דקויות שלא מוצגות בגרפים הדו-ממדיים ולכן בגרפים ניתן לראות הפרדה שאינה נתפסת במדדים מחושבים.

לסיכום, על אף תוצאות הגרפים, התוצאות מראות שהשימוש במודל K-means על הסט הנתון אינו יוצר אשכולות שמתאימים במידה טובה למחלקות המקוריות של הנתונים.

3. בחירת K-Optimal



לטובת בחירת מספר האשכולות האופטימלי (K). בחרנו בשיטת המרפק (Elbow Method) משום שאנו מחפשים את הנקודה שבה הוספת אשכולות נוספים מפחיתה פחות ופחות מהאינרציה. הגרף המוצג מראה את שיטת המרפק (Elbow Method) מהגרף ניתן לראות את השינוי באינרציה (Inertia) כתלות במספר האשכולות (K). בחרנו ב-K שבו אנו רואים את נקודת המרפק (Elbow Point) (המקום בו הירידה באינרציה מתחילה להתמתן). בגרף המוצג, ניתן ב-K=3, ניתן לראות שהירידה באינרציה אינה גדולה כמו הירידות שבין K=1 ל-K=2 ו-K=2 ל-K=3, ולכן נקודה זו נחשבת לנקודת המרפק.

הקשר לסיפור שלנו הוא שבהתחשב בכך שאנחנו עוסקים בנתונים על שביעות רצון לקוחות (מרוצה ולא מרוצה), ניתן היה לצפות ל-K=2, מכיוון שישנן שתי קטגוריות עיקריות. עם זאת, K=3 מציע שיש קבוצה שלישית פוטנציאלית בנתונים, שיכולה לייצג לקוחות שהם ניטרליים או שישנם גורמים אחרים שיכולים להשפיע על השונות בקבוצות.

הסיבה האפשרית לכך ש-K=3 מתאים יותר מהציפיות המקוריות של K=2 היא שהנתונים מכילים מורכבות נוספת שאינה נראית לעין בשלב ראשון. ייתכן שגורמים נוספים או תתי קבוצות של לקוחות משפיעים על התוצאות, ולכן כדאי לחקור יותר את המשמעויות האפשריות של האשכול השלישי.

באופן כללי, הבחירה ב-K=3 נראית מתאימה לפי שיטת המרפק והנתונים שמתקבלים.

4. בחינת שיטת אשכול נוספת והשוואה ל-K-means

נבחן את שיטת אשכולות היררכיים (Hierarchical Clustering). בשיטה זו, כל נקודת נתונים מתחילה באשכול משלה, ולאט לאט מאחדים אשכולות סמוכים לפי מדדי דמיון מסוימים עד שמתקבלים מספר אשכולות מצומצם.

השוואה בין סכמות האשכולות:

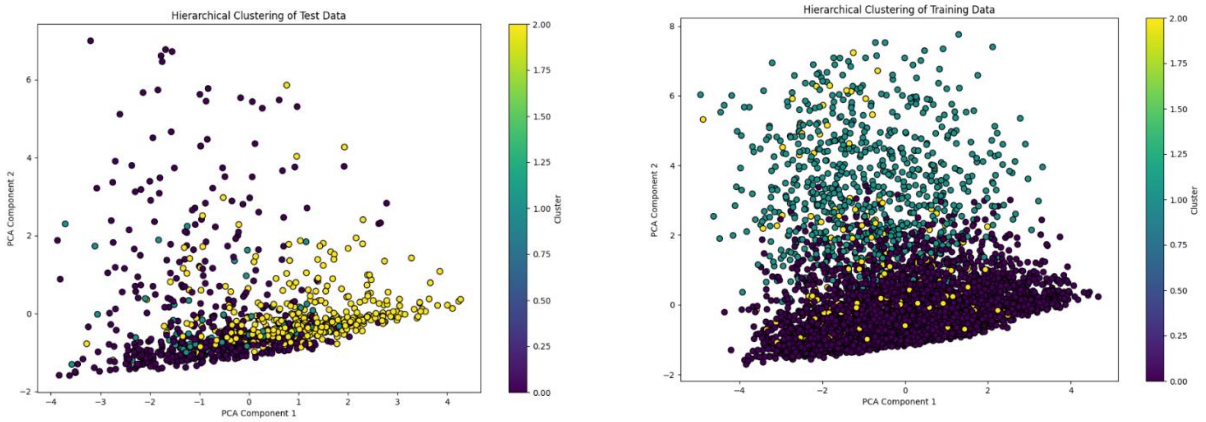
```
Hierarchical Clustering on the training set
Train Adjusted Rand Index (ARI): -0.01057228535708909
Train Normalized Mutual Information (NMI): 0.009191829060054559
Train Homogeneity: 0.00813741992759477
Train Completeness: 0.010560166450079123
Train V-Measure: 0.009191829060054557
Test Adjusted Rand Index (ARI): 0.2370112881611077
Test Normalized Mutual Information (NMI): 0.18232739479137886
Test Homogeneity: 0.2083123501524235
Test Completeness: 0.1621062095568043
Test V-Measure: 0.18232739479137886
```

```
K-means clustering on the training set
Train Adjusted Rand Index (ARI): 0.2730408079372288
Train Normalized Mutual Information (NMI): 0.20912678335605484
Train Homogeneity: 0.21035854306080168
Train Completeness: 0.20790936487536607
Train V-Measure: 0.20912678335605486
Test Adjusted Rand Index (ARI): 0.2865881180241796
Test Normalized Mutual Information (NMI): 0.22107968581960036
Test Homogeneity: 0.22184404662089136
Test Completeness: 0.22032057412315065
Test V-Measure: 0.22107968581960039
```

K-means הוא אלגוריתם פשוט ומהיר לקלאסטרינג, קל לשימוש בגלל מעט הפרמטרים לכוונון, אך הוא רגיש לנקודות קצה ודורש בחירה מראש של מספר האשכולות (K), מה שעלול להיות מאתגר. Hierarchical Clustering, לעומת זאת, אינו דורש בחירת K מראש ומתאים לנתונים מורכבים יותר, אך סובל מזמן ריצה ארוך וצריכת זיכרון גבוהה. התוצאות מצביעות על כך ש-K-means מתאים יותר לנתונים כאשר מהירות ופשטות חשובים, בעוד ש-Hierarchical Clustering עדיף במקרים של נתונים מורכבים יותר.

התוצאות של K-means מצביעות על התאמה טובה יותר של האשכולות למחלקות בהשוואה ל- Hierarchical Clustering, כפי שמשתקף במדדים כמו Completeness, Homogeneity, ARI, NMI, ו-V-Measure. Hierarchical Clustering הציג ביצועים נמוכים משמעותית על סט האימון, אך עדיין הצליח לספק תוצאות קבילות על סט הבדיקה. הבחירה בין שתי השיטות תלויה במטרות ובתנאים של הבעיה הנתונה: אם מהירות ופשטות חשובים יותר, אז K-means הוא הבחירה המתאימה; אם הנתונים מורכבים יותר ויש צורך בגמישות, Hierarchical Clustering עשוי להיות עדיף.

הגרפים המוצגים לעיל מדגימים את תוצאות שני האלגוריתמים בשני ממדים באמצעות PCA:



<u>אימון מודל נוסף: SVM</u>

מנגנון הסיווג של SVM (Support Vector Machine) הוא אלגוריתם למידה המשמש לסיווג ורגרסיה. בסיווג בינארי, במקרה שלנו, האלגוריתם מנסה למצוא וקטור אופטימלי שמפריד בין שתי הקבוצות במרחב התכונות.

<u>הפרמטרים שנבחנו ונבחרו</u>:

1. <u>C</u>: פרמטר זה מאזן בין שגיאת הסיווג לבין מורכבות המודל. ערכים גבוהים יותר של C מובילים למודל מורכב יותר שמנסה להתאים יותר לנתוני האימון.
2. <u>Gamma</u>: פרמטר זה קובע את ההשפעה של כל נקודת אימון. ערך גבוה יותר אומר שכל נקודה משפיעה על תחום קטן יותר.
3. <u>Kernel</u>: פונקציית הקרנל מאפשרת ל- SVM לעבוד במרחבים לא ליניאריים. נבחנו שני סוגי קרנלים 'rbf' (Radial Basis Function), 'linear'.

בצענו תהליך כיוונון הפרמטרים על מנת למצוא את הפרמטרים האופטימליים ביותר.

קיבלנו שהפרמטרים הטובים ביותר במודל הם:

```
Best parameters found:  {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
```

תוצאות המודל SVM:

עבור סט האימון:

```
Tuned SVM On the train set
Accuracy: 0.915979
Precision: 0.925622
Recall: 0.877158
F1 Score: 0.900739
```

עבור סט הבחינה:

```
Tuned SVM On the test set
Accuracy: 0.896000
Precision: 0.910165
Recall: 0.853659
F1 Score: 0.881007
```

<u>השוואה בין מודלים</u>

1. נעדיף להשתמש ב DT או בNN על פני K-means מכיוון ש K-Means-הוא אלגוריתם המשמש בעיקר לסיווג נתונים על בסיס דמיון כללי, ואינו מתאים למקרים שבהם יש צורך לסווג נתונים על סמך תכונות מסוימות. לעומת זאת, עצי החלטה ורשתות נוירונים מתאימים למקרים שבהם המטרה היא לסווג נתונים על סמך תכונות ספציפיות.

2. השוואת ביצועי המודלים:

א. MLP- ה- MLP מציג את ה- F1 Score הגבוה ביותר מבין שלושת המודלים, עם ערך של 0.894562. זה מעיד על כך שהמודל מצליח לאזן בצורה מיטבית בין זיהוי נכון של דוגמאות חיוביות ושליליות, תוך שמירה על דיוק גבוה בסיווגים.

ב. DT- עץ ההחלטה מציג F1 Score של 0.881890. כלומר, המודל מצליח לסווג תצפיות בצורה טובה אך מעט נמוכה מזו של מודל MLP.

ג. ה- SVM מציג F1 Score של 0.881007, קרוב מאוד לזה של עץ ההחלטה, אך נמוך מזה של הMLP-.

מטרת בחינת המודלים היא לבנות מודל שיסווג ויחזה בצורה המדויקת ביותר. על כן, בהתבסס על התוצאות, נבחר במודל MNP המניב תוצאות סיווג הטובות ביותר בין המודלים.

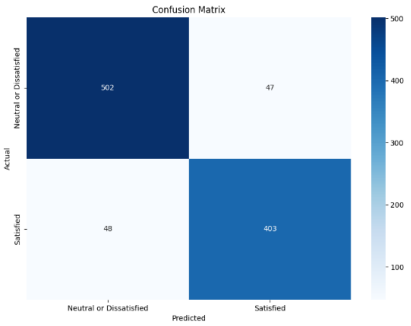
<u>המודל הנבחר</u>

בפרויקט זה, בחרנו ברשת נוירונים כמודל לטובת ההגשה לתחרות. רשת נוירונים מצטיינת ביכולותיה לזהות דפוסים מורכבים ולהתאים את עצמה לסוגים שונים של נתונים, דבר שהופך אותה לאופציה מתאימה למשימה זו. לצורך אימון הרשת, בחרנו בהיפר-פרמטרים הבאים:

- **מספר השכבות החבויות-1.** שכבה חבויה אחת נבחרה כדי לשמור על פשטות המודל תוך כדי שמירה על יכולת למידה מספקת של דפוסי הנתונים.
- **מספר הנוירונים בכל שכבה חבויה- 50.** מספר זה של נוירונים מאפשר לרשת גמישות מסוימת בלמידה של מאפיינים שונים, מבלי להעמיס על המודל יתר על המידה.

- **פונקציית אקטיבציה-**ReLU. פונקציית האקטיבציה הנפוצה ReLU נבחרה בשל יכולתה להתמודד עם בעיות של דילול שיפוע ולהאיץ את תהליך הלמידה.
- **ערך האלפא- 0.0001.** ערך זה של אלפא מייצג את הפרמטר לקבועה המוסיפה לפונקציית העלות לצורך מניעת overfitting על ידי הפחתת משקליהם של הפרמטרים.
- **קצב למידה- קבוע (constant).** קצב למידה קבוע מאפשר התקדמות יציבה בתהליך הלמידה, ומונע את השפעות ההפחתה הדרגתית של קצב הלמידה שיכולה להאט את תהליך הלמידה.
- **Adam – Solver.** האלגוריתם Adam נבחר לפתרון בעיית האופטימיזציה, בזכות יעילותו הרבה והיכולת שלו להתאים את קצב הלמידה במהלך האימון.

מטריצת מבוכה:



502 תצפיות סווגו נכון כ- "Neutral or Dissatisfied".

403 תצפיות סווגו נכון כ- "Satisfied"

47 תצפיות סווגו לא נכון כ- "Satisfied" למרות שהן היו " Neutral or Dissatisfied"

48 תצפיות סווגו לא נכון כ- "Neutral or Dissatisfied" למרות שהן היו "Satisfied".

1. **דיוק המודל**

המודל מצליח באופן כללי לסווג נכון את התצפיות, עם רוב התצפיות מסווגות נכון בשתי הקטגוריות. הדיוק הגבוה במטריצה מעיד על ביצועים טובים של המודל.

2. **הטיית המודל**

המודל מציג חלוקה יחסית מאוזנת בין שתי הקטגוריות, ללא העדפה ברורה לקטגוריה אחת על חשבון השנייה. עם זאת, ישנם מעט מקרים של סיווג שגוי (False Positives) ו (False Negatives-עבור שתי הקטגוריות, אבל אין הטיה מובהקת לאחת מהן.

3. **חוזקות המודל**

המודל מצליח לאתר במידה רבה את התצפיות בשתי הקטגוריות, במיוחד את התצפיות בקטגוריה "נייטרלי או לא מרוצה", עם שיעור טעויות נמוך יחסית. בנוסף הדגם אינו נוטה להעדיף קטגוריה אחת על פני האחרת, מה שמעיד על איזון טוב בין הקטגוריות.

4. **חולשות המודל**

המודל נכשל לסווג כראוי חלק קטן של התצפיות בקטגוריות השונות. בפרט, ישנו מספר דומה של טעויות סיווג בשני הכיוונים. בהסתכלות קדימה, כדי לשפר את המודל, ניתן לבצע ניסיון בהשלמת נתונים שונה ויצירת משתנים חדשים, שיפור הבחירה של משתנים, הגדלת נתוני האימון או כיוונון נוסף של ההיפר-פרמטרים.

סיכום: המודל מציג ביצועים טובים יחסית עם רמות דיוק גבוהות בשתי הקטגוריות. אין הטיה משמעותית לכיוון אחת הקטגוריות, אך יש מקום לשיפור בכדי להקטין את כמות הטעויות בסיווג התצפיות בין שתי הקטגוריות.