Data Mining and Visualization (83676)
Final Project
Part 2

מגישים: טל רוזנצוויג (213391691), אופיר יחזקאל (214328940).

2.0 מבוא קצר:

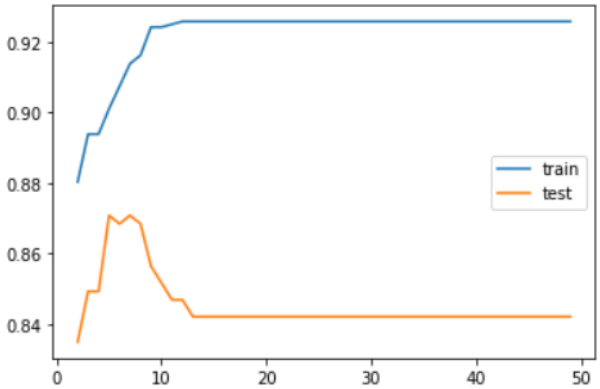
בחלק זה קיבלנו בסיס נתונים של מעל ל-500 לקוחות ללא ערכי ה-response, כלומר יש לנו מידע רב על כל לקוח (במקרים שאין מוסבר בהמשך כיצד טיפלנו – דומה לשלב ה-pre-process) ועלינו להכריע האם הוא יגיב לקמפיין או לא. בסוף ניבדק לפי רמת ה-accuracy שקיבלנו לעומת שאר הסטודנטים ועלינו לשאוף להגיע לאחוזים כמה שיותר גבוהים.

2.1 התאמת ה – pre-process לחלק זה:

ראשית, בדקנו אם יש רשומות שלהן מעל 6 ערכי Null, שכן בחלק הקודם בחרנו למחוק רשומות כאלה. ראינו שגם בבסיס הנתונים החדש היו מספר רשומות כאלה ומחקנו אותן בהתאם להוראות, שכן חסר בהן מידע רב מידי. בנוסף, וויתרנו על החלק של ה-PCA, שכן פחות נוח לעבוד עם הנתונים במצב הזה. נשים לב שגם כאן היו רשומות עם כל מיני ערכים חסרים, לשמחתנו הדרכים שבחרנו לטפל בהן בשלב ה-pre-process עבדו גם כאן ויכולנו להתחיל את חלק זה של הפרויקט.

2.2 ארגון המידע לחלק זה:

תחילה, פיצלנו את בסיס הנתונים ל-x ו-y, כך ש-x מתאר את כל התכונות מלבד ה-Response (ערך המטרה שלנו) ו-y מתאר את ה-Response. לאחר מכן, הרצנו עץ החלטה לקבלת אינטואיציה וראינו שיש לנו overfitting ממש גדול (של 100%), לכן החלטנו להביט בכמה מה-hyperparameters שלנו. בדקנו את ההשפעה של ה-max_depth באמצעות הגרף:



הדפסנו לעצמנו את הערכים וראינו שה-max_depth האופטימלי הוא בין 4 ל-6. לאחר מכן הרצנו cross-validation עם max_depth של 5 וקיבלנו שה-accuracy של עץ ההחלטה הוא 86.59% וכשהרצנו ובאמצעות dummies הצלחנו לשפר ל-87.07%. לאחר מכן, רצינו לבדוק אם יש מצב של overfitting, לכן בדקנו את רמת ה-accuracy על ה-train עם ה-dummies וקיבלנו תוצאה של 89.82% תוצאה שהיא הגיונית לחלוטין בהתחשב לרמת האחוזים של ה-87.07% שקיבלנו. לאחר קבלת אינטואיציה זו, נמשיך לנסות לשפר את מסווג ה-Decision Tree בהמשך בסעיף 2.4.

2.3 המדדים שברצוננו למקסם:

תחילה בדקנו וראינו שב-train יש כ-85% מהלקוחות שלא הגיבו לקמפיין ו-15% שהגיבו, מכאן שאם נשער שכל הלקוחות יגיבו לקמפיין כבר נצליח להגיע ל-accuracy של כ-85%, מכאן אנחנו מבינים שה-accuracy הוא לא בהכרח המדד המתאים לבדוק לפיו. באופן דומה, אם נשער שכל הלקוחות יגיבו לקמפיין, נגיע ל-recall של 100%. מכאן הגענו שהמדד המתאים לבדיקות הוא ה-precision, כלומר אנחנו רוצים לבדוק מבין כל הלקוחות ששיערנו שהם יגיבו לקמפיין, כמה האמת הגיבו. חשוב לציין, שהתייעצנו עם דנית על כך והבנו שהמדידה לגבי הבונוס תינתן לפי ה-accuracy, לכן במהלך העבודה ניסינו למקסם ככל האפשר גם את ה-accuracy וגם את ה-precision.

2.4-2.5 מסווגים, evaluation metrics ו-hyperparameters:

- **Decision Tree:**

בהמשך לתוצאות שקיבלנו בסעיף 2.2, כעת ניסינו למקסם את ה-accuracy באמצעות שיטת ה-hyperparameter Grid **Random**, שתמצא לנו שילובים של ערכי hyperparameters שונים לקבלת accuracy מקסימלי, כאשר את max_depth הגדרנו להיות 4-6 (כולל), את min_samples_split הגדרנו להיות 11-15 (כולל) ואת min_samples_leaf הגדרנו להיות 4-6 (כולל). **נציין שבהתחלה הגדרנו את טווחים אחרים וגדולים יותר, כשראינו את התוצאות שמתקבלות הבנו מה אזור הטווחים שיניב לנו תוצאות אופטימליות, ולכן צמצמנו לטווחים אלה (על מנת לחסוך בזמן ריצה ושנוכל לעשות איטרציות רבות ולקבל מודל מדויק ככל האפשר) – באופן זה פעלנו לכל אורך העבודה למציאת ה-hyperparameters האופטימליים בכל מסווג**. בנוסף, הגדרנו את מספר האיטרציות להיות 1000 ואת כמות ה-folds ל-cross validation להיות הקבוע cv_number, שאותו הגדרנו בהתחלה להיות 60. הסתכלנו על fold שרירותי לקבלת אינטואיציה וראינו שרמת ה-accuracy של המודל עם ערכי ה-hyperparameter הדיפולטיביים היא 83.01%, בעוד ששל המודל עם ערכי ה-hyperparameter ששיטת ה-Random מצאה כאופטימליים ל-fold היא 89.23%, כלומר הבדל של 6.22%! אך נזכיר כמובן שמדובר בהשוואה ספציפית ושרירותית, שתורמת בעיקר לאינטואיציה. לאחר מכן, ניסינו להשתמש בשיטת ה-Grid Search ונזכיר שעוד בהתחלה ניסינו לעשות סוג של "ניחוש מושכל" ל-hyperparameters שיובילו אותנו לרמת ה-accuracy האופטימלית (סימנו אותה ב-**first model**) שם בחרנו את max_depth להיות 5, את min_samples_split להיות 15 ואת min_samples_leaf להיות 5.

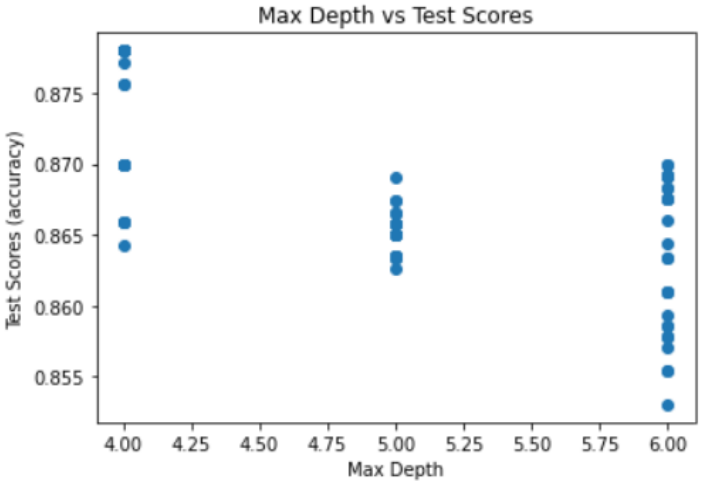
לאחר מכן, ביצענו השוואה בין 3 השיטות: ה-Random , ה-Grid וה-first model בכך שבדקנו אם בהשוואה בין כל 2 מודלים יש סתירה להנחת האפס (שאומרת שהמודלים שווים), או תמיכה בהנחה. כאשר השוונו בין ה-first model למודל של ה-Grid קיבלנו שה-p-value הוא 0.02989, כלומר מספר הקטן מ-0.05, ולכן נוכל לומר **ששיטת ה-Grid טובה מה-first model באופן מובהק סטטיסטית**. לאחר מכן, כשהשוונו בין ה-Random ל-first model, קיבלנו שה-p-value הוא 0.056059, תוצאה הגדולה במעט מאוד מ-0.05. בהתייעצות עם דנית הבנו שגם p-value של 0.1 נחשב טוב, ולכן נוכל לאשר **שבאופן מובהק סטטיסטית שיטת ה-Random טובה מה-first model**. לבסוף, כשהשוונו בין שיטת ה-Random ל-Grid קיבלנו p-

value של 0.9098, כלומר אי אפשר להכריע בצורה מובהקת סטטיסטית איזה מהשיטות עדיפה, משום שאין סתירה להנחת האפס. בהתייעצות עם דנית הבנו שניתן לבחור את אחת השיטות באופן שרירותי, או מסיבות אחרות. בחרנו ב-Grid ובהמשך נסביר מדוע. לאחר מכן, בדקנו את ה-accuracy הממוצע של כל אחת מהשיטות וקיבלנו:

שיטה	Accuracy
first model	85.63%
Random model	87.91%
Grid model	87.80%

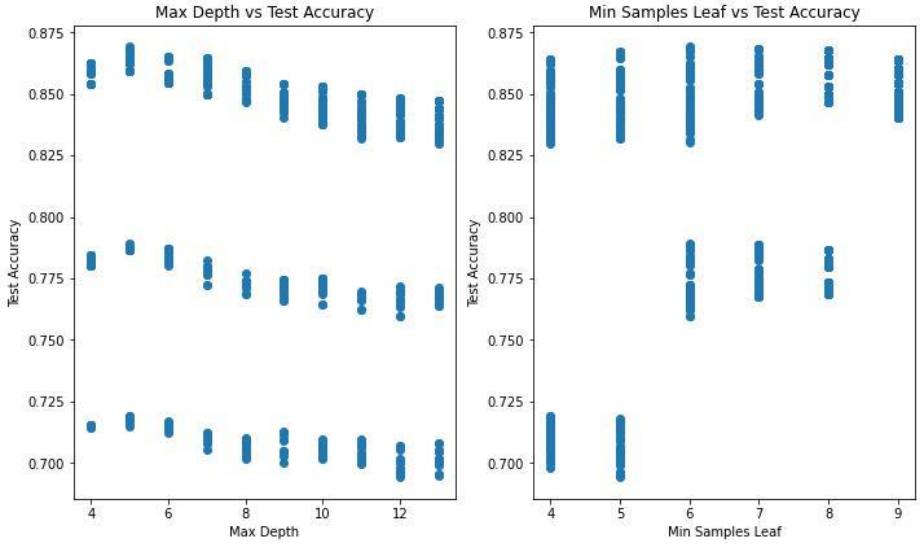
תוצאות התומכות במה שהסברנו קודם, שה-Random וה-Grid טובות מה- first model, אך ביניהן אין שיטה הטובה באופן מובהק.

לאחר מכן, יצרנו גרף לוויזואליזצית הקשר בין ה-max_depth hyperparameter לבין ה-accuracy, עבור ערכי max_depth של 4-6 וקיבלנו:

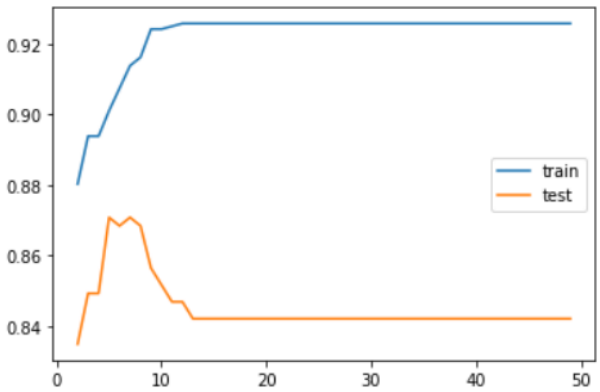


ניתן לראות בצורה אינטואיטיבית שבין max_depth של 5 ו-6 אין הבדל משמעותי לשיפור ה-accuracy וש-max_depth של 4 הוא האופטימלי.

בנוסף, יצרנו גרף המתאר את הקשר בין ה-max_depth לרמת ה-accuracy וגרף המתאר את הקשר בין min_samples_leaf לרמת ה-accuracy:



ניתן לראות שלכל max_depth ולכל min_samples_leaf יש אזורים מסוימים של accuracy שבהן הגרף מתקבץ. אנו מניחים שזה קורה, מכך שהשיטה מנסה כל פעם ל"ערבב" ערכי hyperparameters שונים ומכאן שעבור ערך ספציפי של max_depth למשל, יש המון אפשרויות של שילוב עם ה-hyperparameters וכל שילוב כזה משתייך ל"התקבצות" אחרת.
בנוסף, נוכל לראות בגרף של ה-max_depth שיש ירידה קלה ברמת ה-accuracy ככל שה-max_depth עולה ושזה אכן תואם לתוצאות שקיבלנו בגרף שראינו קודם:



אחר כך, רצינו להסתכל על מדד ה-precision, בתקווה שיוכל לעזור לנו להכריע בין טיב ה-Random ל-Grid. בדקנו את המקרה הממוצע וקיבלנו:

שיטה	Precision
first model	53%
Random model	57%
Grid model	62%

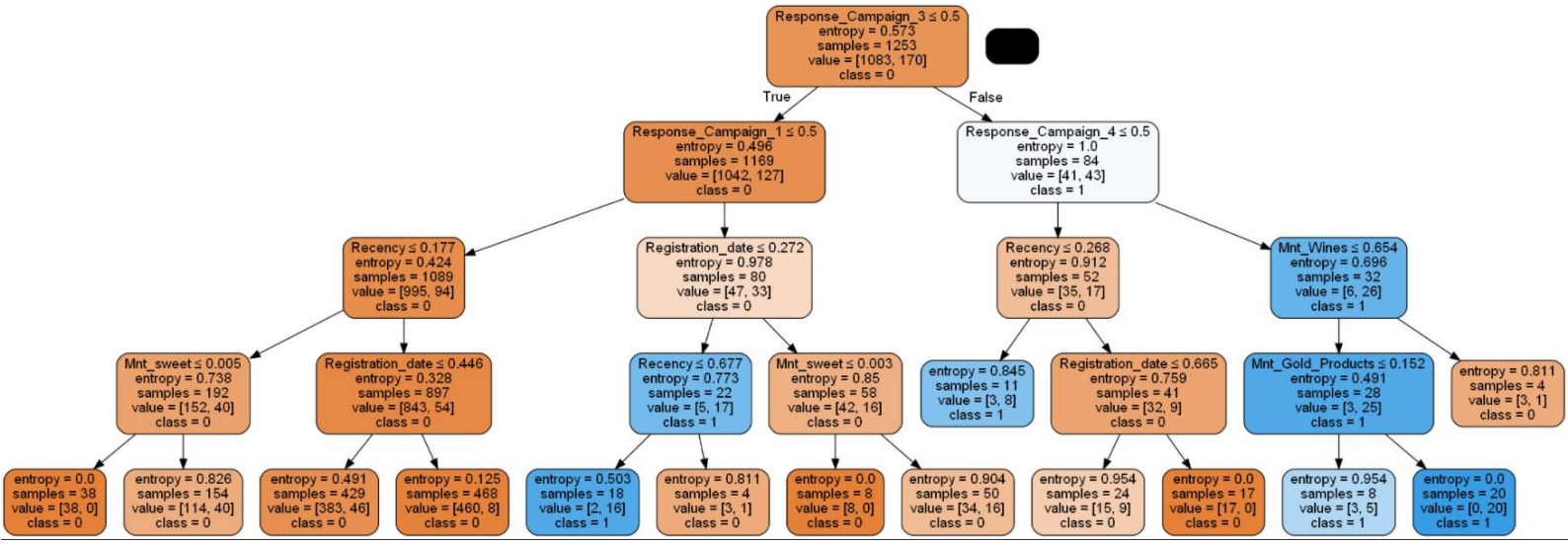
לאחר מכן, בדקנו אם ניתן להכריע ביניהם בצורה המובהקת סטטיסטית וקיבלנו:

השיטות שהשוונו	p-value
Random vs Grid	0.575
Random vs first model	0.1453
first model vs Grid	0.5199

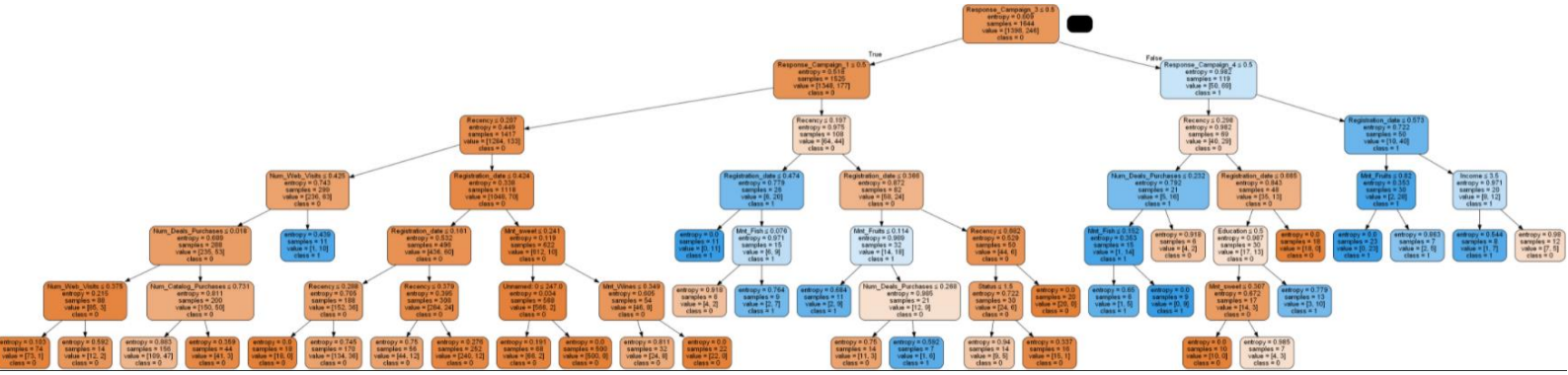
כלומר, אף אחד מה-p-value לא קטן מ-0.05 ואפילו לא מ-0.1, ולכן לא ניתן להכריע למי מהשיטות ה-precision הטוב ביותר בצורה מובהקת סטטיסטית.

לאחר מכן, יצרנו גרפים לעצי ההחלטות וקיבלנו:

עבור ה-Grid:

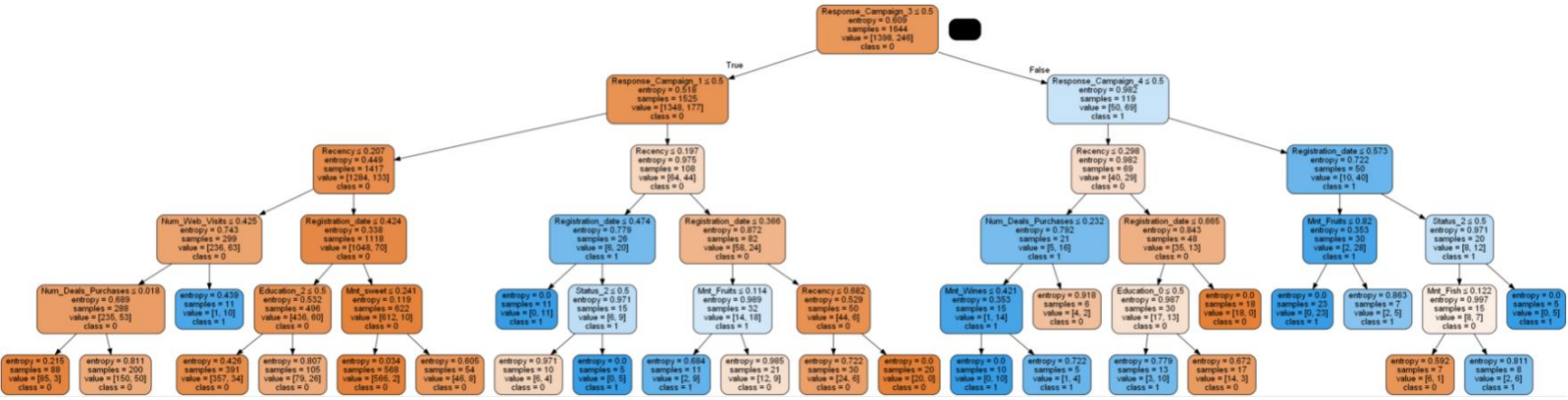


עבור ה-Random:



*ניתן לראות את הגרף בצורה ברורה במחברת הקוד.

עבור ה-first model:



*גם כאן ניתן לראות את הגרף בצורה ברורה במחברת הקוד.

מגרפי העצים ניתן לראות בבירור שהעץ של ה-Grid יצא הכי אלגנטי, קומפקטי ויפה. מה שמראה שעבור ה-Grid זמן הריצה הוא כניראה האופטימלי ושהסיכוי ל-overfitting הוא הקטן ביותר מבינהן. לכן, נוכל להכריע (לא בבירור, אבל מהרבה אינטואיציה) ששיטת ה-Grid היא האופטימלית בשבילנו.

לאחר מכן, רצינו לנסות למקסם את ה-precision. גם כאן השתמשנו ב-2 מהשיטות הקודמות: ה-Random וה-Grid. קיבלנו שה-precision במקרה הממוצע של כל אחד מהם הוא:

שיטה	Precision
base model (מודל דיפולטיבי)	37.05%
Random model	59.81%
Grid model	59.81%

נשים לב שה-Random וה-Grid בחלק זה למעשה מספקים את אותו מודל, ולכן ה-precision שלהם זהה. נשים לב שמודלים אלו שיפרו לנו את ה-precision ב-22.76%. לאחר מכן, בדקנו אם ניתן להכריע בצורה מובהקת סטטיסטית בין טיב ב-Grid למודל הבסיסי בהקשר ה-precision וקיבלנו p-value של 0.01062, כלומר שהוא קטן מ-0.05, ולכן נוכל להכריע בצורה מובהקת סטטיסטית שה-Grid וה-Random עדיפים על המודל הדיפולטיבי. החלטנו לבחור באופן שרירותי ב-Grid כהכי טוב בהקשר ל-precision מנוחות.

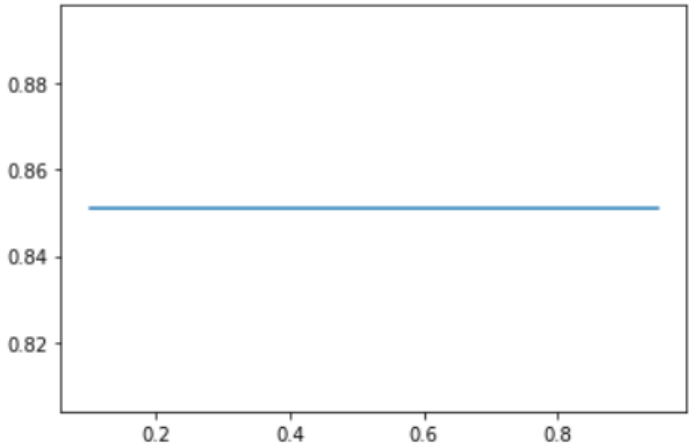
נצרף תמונה המתארת את ה-hyperparameters שקיבלנו עבורו (ה-Grid):

```
{'criterion': 'entropy',
 'max_depth': 5,
 'min_samples_leaf': 6,
 'min_samples_split': 13}
```

לסיכום, כאשר נשווה את מסווג ה-Decision Tree עם מסווגים אחרים, נתייחס לשיטת ה-Grid כטובה ביותר.

- **SVM:**

תחילה, בדקנו את ההשפעה של ה-hyperparameter: C על ה-accuracy וקיבלנו את הגרף:



כלומר, אפשר לראות של-C אין השפעה על ה-accuracy, לכן נשתמש ב-C הדיפולטיבי, השווה ל-0.9. לאחר מכן, בדקנו את ההשפעה של ה-hyperparameter: kernel על ה-accuracy עבור האופציות: poly, rbf and sigmoid וקיבלנו את התוצאות הבאות:

kernel	Accuracy
poly	85.1%
rbf	85.1%
sigmoid	77.38%

מכאן אנו רואים שאין הבדל בין השימוש ב-poly או ב-rbf וששניהם טובים יותר מה-sigmoid, לכן בחרנו להשתמש ב-rbf, שהוא גם הדיפולטיבי.

לאחר מכן בדקנו אם יש מצב של overfitting בכך שחישבנו את רמת ה-accuracy עם kernel='rbf' ו-C=0.9 על ה-train, כפי שהסברנו קודם וקיבלנו שרמת ה-accuracy היא 85.1%, כלומר שאין כאן overfitting.

אחר כך, השוונו בין המסווג של ה-Decision Tree, שהסברנו עליו בהרחבה מקודם ל-SVM וקיבלנו p-value של 0.0077, מספר הקטן מ-0.05. כלומר, ניתן לומר **שמסווג ה-Decision Tree עדיף מה-SVM באופן מובהק סטטיסטית**.

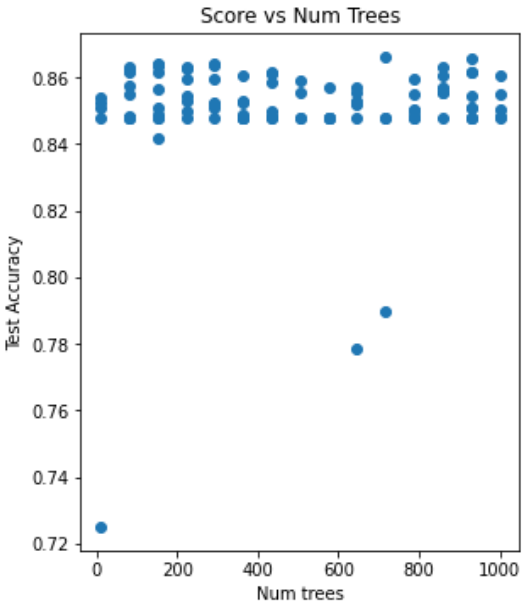
- **Random Forest:**

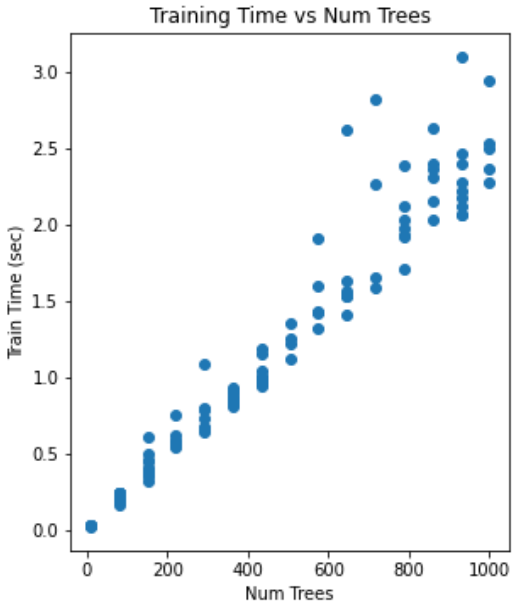
תחילה נציין שהמסווג רץ המון זמן, אז כדי לקצר את זמני הריצה בחרנו מספר ערכים מדגמיים לכל אחד מה-hyperparameters וכדי למצוא את ערכי ה-hyperparameters שממקסמים את ה-accuracy בחרנו להשתמש בשיטות ה-Random וה-Grid. עבור כל אחד מהם קיבלנו:

שיטה	Accuracy	Improvement
default	85.65%	
Random	84.21%	-1.68%
Grid	83.97%	-1.96%

כלומר, קיבלנו ירידה של 1.68% ב-Random ושל 1.96% ב-Grid.
כשניסינו לחשוב מה יכול לגרום לירידה כזו, חשבנו לבדוק בכיוון של overfitting. בדקנו זאת על ה-train וראינו שאכן יש לנו במודל הדיפולטיבי overfitting של 100%! מה שמסביר את ה"ירידה" שרואים בתוצאות. לכן, בחרנו באופן שרירותי להשתמש במודל של ה-Grid. בדקנו את רמת ה-accuracy של המודל במקרה הממוצע וקיבלנו 87.67%.

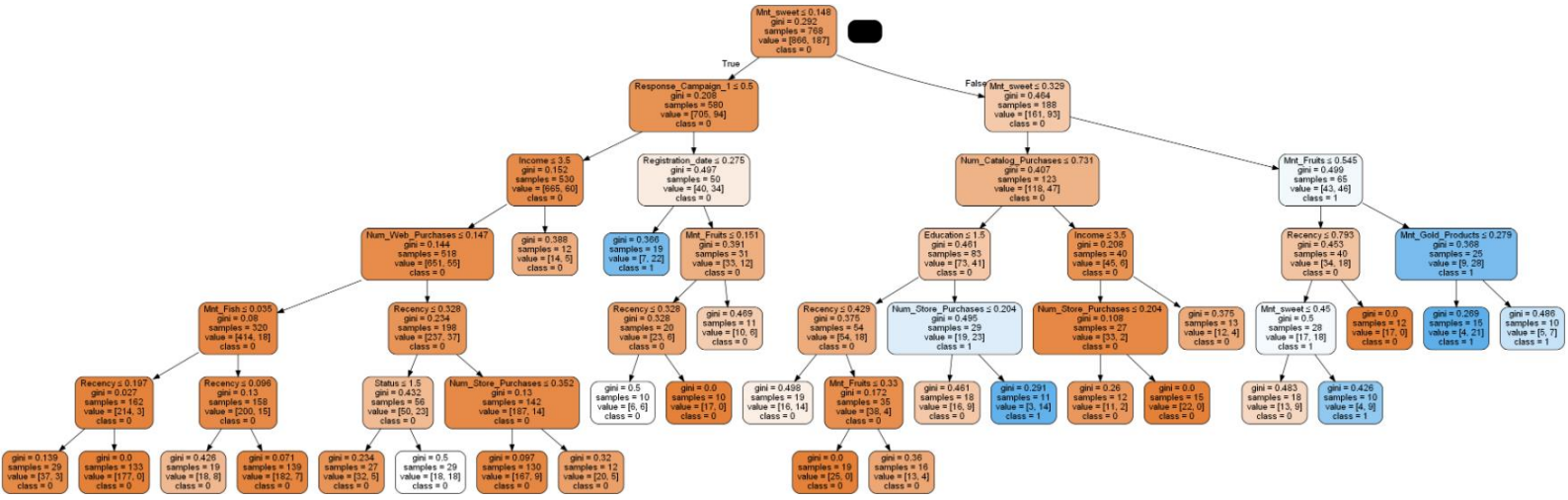
יצרנו גרפים המתארים את הקשר בין מס' העצים לרמת ה-accuracy ולזמן הריצה וקיבלנו:





ניתן לראות עפ"י הגרף הימני שככל שיש יותר עצים, זמן הריצה של המודל יותר ארוך, שכן זה מאוד הגיוני. בנוסף, עפ"י הגרף השמאלי קיבלנו שכבר אחרי כמות נמוכה של עצים מצליחים להגיע לרמת accuracy גבוהה ואז לסוג של "רוויה". ניתן לחשוב על שבאופן תיאורטי, אחרי שמגיעים לרוויה, מקבלים על כל הוספות של עצים שיפור מאוד נמוך ה-accuracy אך זמן הריצה עולה משמעותית, זהו שיקול שכדאי לקחת בחשבון (והאמת שחווינו זאת באופן אישי במהלך העבודה על חלק זה בפרויקט).

לאחר מכן, יצרנו גרף לתיאור העץ שמתקבל בסיווג זה:



*ניתן לראות את הגרף בצורה ברורה במחברת הקוד.

לבסוף, השוונו בין מסווג ה-Decision Tree ל- Random וקיבלנו p-value של 0.434, כלומר תוצאה הגדולה מ-0.05, ולכן לא ניתן לומר בצורה מובהקת סטטיסטית מי מהמסווגים יותר טוב. משיקולי זמן ריצה, **החלטנו להמשיך ולהתייחס ל- Decision Tree כמסווג הטוב שלנו**, שכן זמן הריצה שלו היה נמוך באופן מאוד משמעותי.

- <u>Naive Base:</u>

למסווג זה אין hyperparameters, ולכן לא נזדקק לשיטות שביצענו קודם כדי למצוא את ה-hyperparameters שיובילו אותנו ל-accuracy האופטימלי. בדקנו את רמת ה-accuracy ו-precision של המסווג וקיבלנו:

Accuracy	81.39%
Precision	39.35%

בדקנו מצב של overfitting, באמצעות בדיקת המסווג על ה-train וקיבלנו רמת accuracy של 82.19%, מכאן ניתן לומר שאין overfitting.

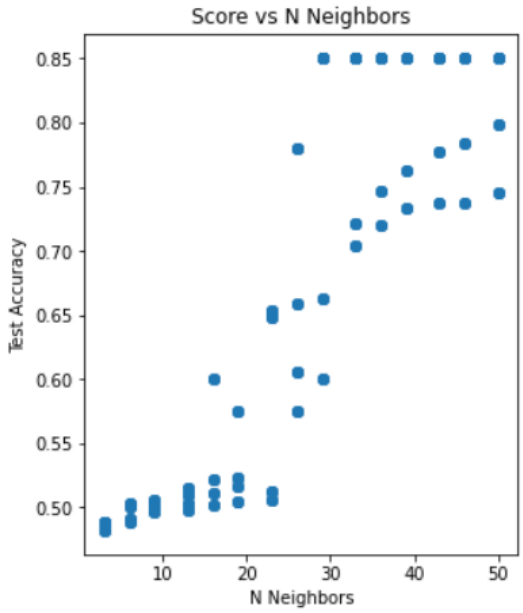
לבסוף, רצינו לוודא שמסווג ה-Decision Tree עדיף בצורה מובהקת סטטיסטית מה-Naive Base, בדקנו את ה-p-value וקיבלנו שהוא שווה 0.0012, כלומר קטן מ-0.05 וניתן לומר שאכן **מסווג ה-Decision Tree עדיף בצורה מובהקת סטטיסטית מה-Naive Base**, כצפוי.

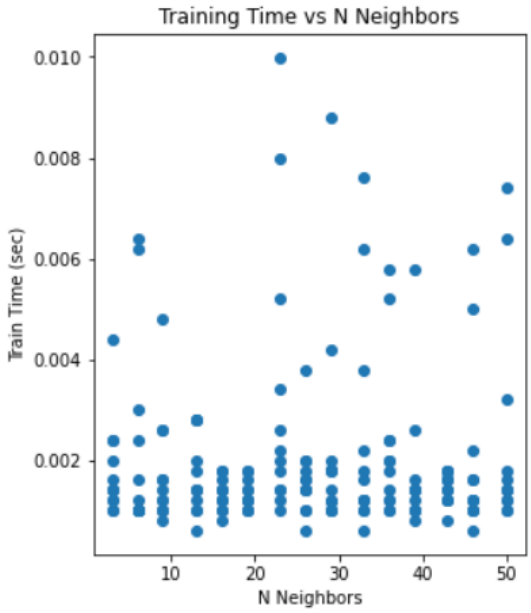
- **K-Nearest-Neighbors:**

למסווג יש מספר hyperparameters, ביניהם n_neighbors, weights, leaf_size, p, כאשר הגדרנו n_neighbors ואת leaf_size באמצעות לולאה הבוחרת 15 ערכים שבין 15 ל-50. השתמשנו בשיטות ה-Random וה-Grid, כדי למצוא את ה-hyperparameters שנותנים לנו את המסווג בעל ה-accuracy הגובה ביותר וקיבלנו:

שיטה	Accuracy	Improvement
Default	81.44%	
Random	85.03%	3.59%
Grid	85.03%	3.59%

בדקנו עבור שתי השיטות – ה-Randoom וה-Grid אם יש מצב של overfitting באמצעות ריצה על ה-train וקיבלנו שרמת ה-accuracy בשתיהן שווה ל-85.63%. כלומר, ניתן לראות שבשתי השיטות מקבלים את אותן תוצאות, בדקנו את ערך ה--p value, כדי לראות אם יש שיטה שטובה מהשנייה מבחינה המובהקת סטטיסטית וראינו שהוא Null, בדקנו וראינו שזה כי שתי השיטות מתארות את אותו מדל (אותם hyperparameters). לכן, בחרנו ב-Grid בצורה שרירותית. יצרנו גרף המתאר את הקשר בין n_neighbors – hyperparameter המתאר את מספר השכנים הנדרש כדי לסווג קודקוד חדש לבין רמת ה-accuracy וזמן הריצה וקיבלנו:





כלומר, מהגרף השמאלי ניתן לראות שככל ש-n_neighbors יותר גדול, כך רמת ה-accuracy גדלה (מאוד הגיוני, שכן נדרשים יותר קודקודים לצורך סיווג של כל קודקוד חדש). מהגרף השמאלי ניתן לראות שאין קשר מובהק בין n_neighbors לזמן הריצה של המסווג.

לאחר מכן, השוונו בין מסווג ה-K-Nearest לבין ה-Decision Tree ומצאנו שה--p value הוא 0.00015, מכאן שניתן לומר שבאופן מובהק סטטיסטית **מסווג ה-Decision Tree עדיף על K-Nearest** מבחינת רמת ה-accuracy.

לבסוף, חשבנו לבדוק את רמת ה-precision של המסווג ולשפר אותה באמצעות שיטות ה-Random וה-Grid, אך זה לא עבד. להערכתנו הבעיה הייתה, משום שבסיס הנתונים לא מאוזן, מה שגורם למסווג לסווג את כל רשומות ה-Response ל-0, מה שלא מאפשר חישוב של precision.

2.6 סיכום:

לבסוף, כפי שראינו לכל אורך הדרך בסעיפים 2.4-2.5, קיבלנו שהמסווג הטוב ביותר שלנו הוא ה-Decision Tree המשופר באמצעות שיטת ה-Grid. לכן, הפעלנו אותו על ה-test וצירפנו את קובץ ה-excel המעודכן להגשה.