

# Microsoft Security Bulletin Analysis

דוח המציג ממצג על ידי בר כהן וסוחר חיים יעקב.

דוח זה עוסק בבייצוע המודלים כחלק מתהליכי ניתוח הנתונים בפרויקט. מטרתו היא לתאר את שלב בניית המודלים, אימונים והערכותם, מתוך מטרה להזות את המודל המתאים ביותר לפתרון הבעייה שהוגדרה בפרויקט.

```
= import numpy as np
= import pandas as pd
= import matplotlib.pyplot as plt
= import seaborn as sns
...
apply_pmodel()
train_test_split(X, stratify=y)
filter(X)
attr(Y, test(X_prob))
train_test_split(X, stratify=y)
apply_log(in_csnse)
randomForestClassifier()
sns.heatmap(base())
filter(X_test, test(y_test))
etrest(y_test)
...
p= y
p= y
k= random_accuracy.sort_excrement()
roc_auc_score(y_test, y_prob)
return triple(train_test_split, X, nobj, merg = k)
compare(product_axis, y='trify=y', statify=t_reccal)
```

שם מרצה : מר אבי זכאי.  
שם מנהה : מר חנן לב.  
מגייסים :  
**בר כהן 208110254**  
**סוחר חיים יעקב 314741851**



### תוכן עניינים

- 3 - .....	<b>בחירה שיטות מידול - בחירה :</b>
- 7 - .....	<b>בחירה שיטות מידול נכונות - סינון + תיאור ראשוני :</b>
- 10 - .....	<b>הנחות המודלים – קבלת החלטות + תיעוד :</b>
- 13 - .....	<b>עיצוב המבחנים למודלים – הגדרות קרייטוריוניות :</b>
- 15 - .....	<b>תיאור המודלים – הרצות ראשוניות + מסקנות :</b>
- 23 - .....	<b>הגדרות הפרמטרים – שינויי הפרמטרים בשיטות המידול :</b>
- 25 - .....	<b>תיאור המודלים – הרצות סופיות + מסקנות :</b>
- 32 - .....	<b>הערכת המודלים – תוצאות + סיכום :</b>



## 1. בחרת שיטות מיזול - בחירה :

בשלב זה, נתאר את המודלים שבהם השתמש ואת דרכי הפעולה של הנתונים, תוך התמקדות בטכניקות שונות המותאמות למאפייני הנתונים. הנתונים שלנו מכילים מספר רב של משתנים קטגוריאליים, ולכן נצורך להתמודד עם האתגר של טיפול באוטם המשתנים. נוכל לבחור להמיר את המשתנים הקטגוריאליים באמצעות מושני דאמי, או להשתמש במודלים שמוסוגים להתמודד עם משתנים קטגוריאליים באופן ישיר, כמו מודל **CatBoost**. בנוסף למודל זה, נבחן גם שלושה מודלים נוספים : **Random Forest**, **רשת נוירוניים** ו- **AdaBoost** וגם מודלים מתקדמים שבסיסם על עצי החלטה כגון : **XGBoosts** ועוד. כל אחד מהמודלים מציע יתרונות מסוימים בהתאם למאפייני הנתונים והמשמעות שאליה נדרשו.

### - CatBoost

**יתרונות :** טיפול אוטומטי במשתנים קטגוריאליים, בעל מנגנון בוסטרואפ אוטומטי, שיפור ביצועים, מניעת Overfitting .  
 **חסרונות :** דרישות זיכרון גבוהות במקרים רבים, זמן אימון ארוך, לא מתאים לנtones קטנים.

#### **מהו המודל ? CatBoost**

המודל מיועד לטפל במשתנים קטגוריאליים בצורה ייילה ואוטומטית, מבוסס על טכניקת Gradient Boosting (SHIPOR ביצועים של מודל חישוב עייני סדרה של מודלים חלשים). המודל מאפשר להתמודד עם משתנים קטגוריאליים ללא צורך בקידוד משתנים קטגוריאליים במספרים באמצעות שיטות שונות. מבצע בוסטרואפ דיפולטיבי עם פרמטר Bayesian (קבالت משקלים רנדומליים לכל תצפית מתוך דריילוק – חילוק בין המשקלים כך שסכומם תמיד 1). חלוקת הנתונים לסטים של אימון ובדיקה מהוות שלב חשוב בתהליכי בניית המודל. חלוקה זו היא להערך את ביצועי המודל על נתונים שלא נראו במהלך האימון, ובכך להבטיח שהמודל לא סובל overfitting, בשלבי הרצאת המודל אנחנו נבחן את החלוקה ואת משתנה ה test כך שיפחית את מידת השגיאה.

```
X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size= test, random_state=42)
```

אור 1 : חלוקת סט אימון ובדיקה

### - Random Forest

**יתרונות :** מהירות גבוהה, עמידות לרעש, הבנה ויזואלית של התוצאות, מתאים לנtones קטגוריאליים ומספריים.  
 **חסרונות :** ירידה בביצועים עם משתנים לא רלוונטיים, בעיות חישוביות עם נתונים גדולים.

#### **מהו המודל ? Random Forest**

מודל שנועד לבצע סיווג ורגסיה באמצעות שילוב של מספר עצי החלטה. שבו הוא מאמן מספר עצים על תתי-קבוצות שונות על הנתונים ומשלב את התוצאות כדי לשפר את הדיקוק והעמידות. חלוקת הנתונים לסטים של אימון ובדיקה היא שלב קריטי. המטרה של חלוקה זו היא להערך את ביצועי המודל על נתונים שלא נראו במהלך האימון, ובכך להבטיח שהמודל לא סובל מ- overfitting.

### - Neural Network

**יתרונות :** למידת תכניות מורכבות, ביצועים מצוינים עם נתונים גדולים, גמישות גבוהה, שילוב סוגים נתונים שונים.  
 **חסרונות :** דורש הרבה נתונים, זמן חישוב ארוך, קשה לפרש את המודל.

#### **מהו המודל ? Neural Network**

מודל זה הוא מודל חישובי, משמש ללמידה תכניות מורכבות מנתונים.



הרשות מורכבת מ-**ת-סוגי שכבות** :  
שכבה הקלט, שכבות מוסתרות, שכבת הפלט.  
בכל נוירון, מתבצע חישוב לפי פונקציית ההפעלה הנבחרת, והעברת התוצאה בין כל גן השכבות.  
הרשות מתאמת באמצעות – **Back-propagation**, שיטה שמעדכנת את המשקלים כך שייפחיתו את השגיאה.

### - AdaBoost

**יתרונות:** עמיד ל- **Overfitting**, חיזוק מודלים פשוטים, יכולה להתמודד עם נתונים לא מאוזנים, עבדה עם משקלים שונים, קל לשימוש.  
**חסרונות:** רגish לרוש, תלוי במספר המודלים, מוגבל על ידי עצים פשוטים .

**מהו המודל ? AdaBoost**  
מודל שנועד לשפר את הביצועים על ידי שילוב של מספר מודלים פשוטים לייצרת מודל חזק יותר. המודל מותמך במרקירים שבهم המודלים הפשוטים טויעם, ומקצה משקל גבוה יותר למודלים אלו במהלך האימון, מה שמאפשר לו לשפר את הדיקום ומצביע התמודדות עם נתונים לא מאוזנים. לאחר אימונו הראשוני, המודל מעדכן את המשקלים של הנתונים השגויים. מודל שיטה מקבל משקל גבוה יותר, בעוד שמודל שזהה נכון מקבל משקל נמוך יותר.  
ההבדל בין מודל זה למודל **CatBoost** הוא שהמודל הזה מותמך בשיפור מודלים פשוטים **Boosting** באמצעות **CatBoost**, אך רגish לרוש ודורש הינה מוקדמות של הנתונים. לעומת **CatBoost** שמתפלט בצורה אוטומטית במשתנים קטגוריאליים, וביצועו טובים יותר עם נתונים גדולים יותר.

### - LightGBM

**יתרונות:** שימוש יעיל בזכרון, תמייה ישירה בנתונים קטגוריאליים, מתאים לביג דאטה.  
**חסרונות:** רגish ל- **overfitting**, פירוש לא שקוּף וקושי בהסביר המודל.

**מהו המודל ? LightGBM**  
המודל מבוסס עצי החלטה ומ吒אים במיוחד למיוחד לעיבוד נתונים גדולים.  
משתמש באלגוריתם מבוסס **Gradient Boosting** שמטרתו לשפר את מהירות האימון והביצועים על ביג דאטה. הוא פועל באמצעות בניית עצי החלטה בצורה עיליה, תוך שימוש בטכניקת **Leaf wise Growth** (האלגוריתם בוחר להרחיב את העלה עם הפיצול שմביא את השיפור הגדול ביותר בפונקציית המטרה).

### - XGBoost

**יתרונות:** דיקוק גבוה, תמייה בסוגי נתונים שונים, פונקציית הענשה **loss function**, בקרה **overfitting**.  
**חסרונות:** רגישות לפרטיטרים, קושי פירוש מודל, זיכרון גבוה.

**מהו המודל ? XGBoost**  
מודל המבוסס על עצי החלטה עם גרדיאנטים מוגברים המשתמש באסטרטגיות של בקרת התאמאה יתר והענשה כדי לשפר את הדיקוק של המודל. הוא מאפשר שימוש בסוגי נתונים שונים ומצביע פונקציות שגיאה מתקדמות להתאמאה אופטימלית והענשה מרבית של המודל. בנוסף, האלגוריתם מבצע חיפוש חכם אחר פיצולים בעז.

### - Quadratic Discriminant Analysis (QDA)

**יתרונות:** מ吒אים לגבולות לא לינאריות, ביצועים גבוהים עם הנחות שונות.  
**חסרונות:** רגish להנחה מסווגות, דורש ביג דאטה, **לא מ吒אים לפיצ'רים קטגוריאליים ובים**.

**מהו המודל ? QDA**  
המודל משתמש בשיטה סטטיסטית למידה מפוקחת, ומניח שככל מחלוקת בין נתונים מתפלגת נורמלית, אך מאפשרות שונות בין המחלקות – מה שmobiel להפרדה לא לינארית.



המודל מחשב את ההסתברות על ידי שימוש במטריצת הקוואריאנס על הנתונים ובממצאים על המחלקות השונות.

המודל מצרך חלוקה ליסטים של אימון ובדיקה, הוא מחשב את השגיאה על ידי בוחנת ההסתברות שהתחזיות שלו שגויות עבור סט הבדיקה. המודל משתמש בחוק ביס כדי להעריך את ההסתברות שדגם שיכת כל אחת מהקטגוריות, ומשיק אותה לקטgorיה עם ההסתברות הגבוהה ביותר. השגיאה נמדדת בדרך כלל באמצעות מדדים כמו דיווק, שיעור השגיאה, או מדי השתירות אחרים.

### - Extra Trees

**יתרונות:** מהיר, פשוט להבנה, עמיד לרעש, לא רגיש ל - overfitting.  
 **חסרונות:** קושי הסבר המודל, לא מתאים למשתנים קטגוריאליים.

### **?Extra tree**

המודל מבוסס עצי החלטה שמטרתו לשפר את הדיווק של תחזיותו באמצעות אקראיות מוגברת בתחילת בניית העצים, דומה ל Random Forest אך בניגוד לו, האלגוריתם מוחפש את הפיזול הטוב ביותר עבור כל צומת בעץ. בוורר פיצול אקראי לחוטין מתוך טווח הערכים האפשריים של הפיצ'ר הנבחר, בנוסף, המדגם עבור כל עץ נלקח מכל הנתונים ולא מגדגים אקראי עם החזרה. תהליך זה הופך את המודל למהיר יותר ופחות רגיש לרעש בתנאים.

### - Tabnet model

**יתרונות:** לומד חשיבות פיצ'רים, ניתן לראות את השפעת כל פיצ'ר, מתמודד טוב עם נתונים לא מאוזנים, מקטין את השגיאה ע"י קבלת פרמטר מס' פעמיים.

**חסרונות:** זמן אימון ארוך, רגיש, כורך בהמרה קטגוריאלית.

### **? Tabnet**

מודל ניירוני, המבוסס על תשומת לב לפיצ'רים ולומד אילו תכונות חשובות במהלך תהליכי הניובי. הוא מאפשר להבין את תרומותם כל פיצ'ר לניבוי. בכל שלב, הוא משתמש במידע שנלמד בשלבים הקודמים כדי להתמקד בפיצ'רים הרלוונטיים ביותר עבור הדוגמה הנוכחיית. תהליך זה מאפשר למודל להיות שקווי ולהסביר על פיצ'רים המשפיעים. דורש זמן אימון ארוך ודורש המרה של משתנים קטגוריאליים.

חלוקת האימון במודול זה היא שונה מאשר המודלים, אופן החלוקת לאימון ובדיקה הוא רגיל, לאחר מכן האלגוריתם מחלק שוב את סט האימון לחת-סט אימון וחת-סט ולידציה. חת-סט הולידציה משמש לכובנוון פרמטרים ולמניעת התאמה יתר של המודל לנינוי האימון. המודל משתמש במנגוני קשב סלקטיביים כדי להתמקד בפיצ'רים הרלוונטיים ביותר לכל דוגמה. זה מאפשר למודל ללמידה אילו פיצ'רים חשובים יותר לחיזוי ולהתעלם מפיצ'רים לא רלוונטיים, מה שמקטין את השגיאה.

```
model = TabNetClassifier()
model.fit(X_train=X_train, y_train=y_train, eval_set=[(X_test, y_test)],
           max_epochs=max_epochs,
           patience=0)
```

אייר 2 : הרצת מודל TabNet עם הפרמטרים השונים

**השימוש ב - eval\_set -** הפרמטר של המודל בפונקציה, משמש להערכת המודל במהלך האימון על סט הבדיקה.

```
epoch 88 | loss: 0.07112 | val_0_accuracy: 0.97662 | 0:00:15s
epoch 89 | loss: 0.06991 | val_0_accuracy: 0.97895 | 0:00:16s
epoch 90 | loss: 0.06479 | val_0_accuracy: 0.98002 | 0:00:16s
epoch 91 | loss: 0.06661 | val_0_accuracy: 0.97747 | 0:00:16s
epoch 92 | loss: 0.06743 | val_0_accuracy: 0.97555 | 0:00:16s
epoch 93 | loss: 0.06948 | val_0_accuracy: 0.9781 | 0:00:16s
epoch 94 | loss: 0.06372 | val_0_accuracy: 0.98065 | 0:00:17s
epoch 95 | loss: 0.06208 | val_0_accuracy: 0.97853 | 0:00:17s
epoch 96 | loss: 0.08449 | val_0_accuracy: 0.97555 | 0:00:17s
```

אייר 3 : רשימת דינון Accuracy עבור מודל TabNet



### - Ft transformer model

**יתרונות:** למידה חזקה, גמיש, זיהוי תכונות חשובות, עבודה עם נתונים לא לינאריים.  
**חסרונות:** אימון ארוך וזמן ריצה איטי מאוד, רגיש, צורך המרה של הנתונים קטגוריאליים.

### ? FT Transformer

המודל מיועד לעיבוד נתונים טקסטואליים. הוא מחלק את הנתונים לסטים של אימון ובדיקה בדרכן כלל בשיטה אקראית, ובמצע אופטימיזציה של פונקציית השגיאה באמצעות מנגנון ושייטות שונות לשיפור הדיקוק. פונקציית השגיאה משתמשת בפונקציות Loss ו-Antrropic, שמודדות את השגיאה, ומטרתן היא למזער את הערך שלhn במהלך האימון. המודל מתמקד בפתרונות החשובים ביותר בכל שלב, מה שמאפשר לו לשפר את הדיקוק על ידי התמקדות במידע הרלוונטי. המודל מבצע אימון מספר פעמיים כדי למזער את השגיאה.

### לסיכום,

על מנת להתמודד עם מאפייני הנתונים הייחודיים, ובפרט עם הריבוי של משתנים קטגוריאליים, בחרנו במספר מודלים המתאימים הן לסוגי הנתונים השונים והן למשימות האנליטיות השונות:

**CatBoost, Random Forest, Neural Networks, AdaBoost, LightGBM, TabNet, XGBoost , QDA, Extra Trees, Ft transformer**

בחירה במודלים אלו תאפשר השוואת ביצועים בהתאם למאפייני הדאטה שלנו, תוך איזון בין דיקוק, מהירות, ויכולת פרשנות.



אייר 4 : סיכום המודלים



### 1.1. בחירת שיטות מידול נכונות - סינון + תיאור ראשוני :

בפרק זה נסקור את תהליכי בחינת המודלים שנבחרו ושיטות המידול. תהליך זה כולל פיצול של הנתונים לעrcות אימון ובדיקה, בחירת מודלים. המודלים שנבחרו נבדקים זה מזה ביכולת ההתחמיזות עם משתנים קטגוריאליים, רעש בתונונים ומורכבות. הנתונים שהתקבלו לאחר עיבוד ראשוני כוללים מספר של רשומות איכותיות ורבות, ותורם לאמיןנות וליציבות של תהליכי הלמידה.

המודלים שנבחרו מחייבים פיצול הנתונים לאימון ובדיקה, כדי להעריך את המודל ולבחו אותו על מספר מודלים שונים נערךapiro לפיה ממדד הדיווק של כל מודל על התכונות שנבחרו.

בשלב עיבוד הנתונים וניקוי הנתונים החסרים, הצלחנו לשמר על **מספר רב** של נתונים למעט כ-300 כפליות, עם הרצת מודל ראשוני מספר הרשות של הנתונים שלנו יותר גבוהה מאוד. ויתרומם לדיקוי המודלים השונים שנריצ'

**CatBoost** - מתמקד בשיפור ביצועים כאשר ישנו משתנים קטגוריאליים. המודל מטפל בקטגוריות באופן ישיר ומציע המרת חכמה של משתנים קטגוריאליים, מה שמשפט את תהליכי עיבוד הנתונים וmphight את הצורך בהכנה ידנית של הנתונים. יתרה מכך, המודל מבצע חיזוי בצורה מהירה ומדויקת, תוך שימוש במנגנון מניעת overfitting, מה שוביל לשיפור ביצועים בסביבה עם נתונים גדולים וקטגוריאליים. מודל זה יכול להשתלב בצורה טובה בתונונים שלנו. בנוסף המודל הניל מצריך איכות נתונים מסוימת, מארח שרגיש לרשות בתונונים שכולים ווצאות שגויות שעלולות להוריד את רמת הדיווק.

בחינת המודל של CatBoost, הנתונים שלנו טובים מאוד לעבודה עם המודל. בנוסף למודל זה נבחן מודלים שונים כגון : Ada Boost, Random Forest, רשות ניירונים ו-CatBoost. בשאר המודלים השונים אנו נצטרך לבצע בעצמנו המרת הנתונים הקטגוריאליים. עבור CatBoost יוכל לבדוק במשתנים המשפעים ביותר על המודל שלנו מתוך התוכנות הטובות ביותר שנבחרו. בנוסף מודל ה- AdaBoost מסוגל לעבודה עם משקלים הנתונים. מודל ה- LightGBM מזהה נתונים קטגוריאליים, ויכולת כווננו מתאימה של המודל התורמת להפחחת ה- overfitting.

נרי מודלים מתקדמים כמו tabnet ו- ft transformer שמצריכים המרת נתונים מותאמת.

```
Epoch 30 | Loss: 0.7333016395568848
Epoch 40 | Loss: 0.7070959210395813
Epoch 50 | Loss: 0.6808043718338013
Epoch 60 | Loss: 0.6659979224205017
Epoch 70 | Loss: 0.6830491423606873
Epoch 80 | Loss: 0.6500369310379028
```

איור 5 : מודל FT Transformer



## לסיכום,

לצורך הערכת ביצועי המודלים, הנתונים יפוצלו לאימון ובדיקה, תוך שמירה על מספר גבוהה של רשותות לאחר סינון כפיליות ועיבוד נתונים חסרים – מה שתורם לדיקת גובה יותר של המודלים. בין המודלים שנבחרו:

### - CatBoost

בולט בטיפול חכם במשתנים קטגוריאליים ומאפשר חיזוי מדויק תוך מניעת Overfitting. מתאים במיוחד למיפוי הדאטא שלנו, אם כי דרוש איקות נתונים גבוהה ורגישן לרעש.

### - Random Forest, AdaBoost

מודלים פשוטים, שבהם נדרש לבצע המרת ידנית של משתנים קטגוריאליים (למשל באמצעות Dummy Variables).

### - LightGBM

مزוהה המשתנים קטגוריאליים בצורה ישירה, מהיר ויעיל, אך דרוש כוונון מדויק כדי למנוע Overfitting. השילוב בין המודלים השונים מאפשר מكيف של השפעת התכונות שנבחרו והערכת ביצועים מדויקת. נוכל גם להפיק תובנות מהשפעת משתנים בולטים על המודלים השונים, בעיקר בעזרת CatBoost.

### - XGBoost

מודל עם ביצועים גבוהים ודיקן מרשימים. מתמודד עם בעיות של Overfitting XGBoost מהיר בזכות אופטימיזציה של תהליכי האימון. מותאים היטב לנוטונים בעלי קשרים לא לינאריים, אך דרוש כוונון מדויק של פרמטרים ורגישן לרעש. איןנו מתמודד טוב עם נתונים קטגוריאליים لكن מצריך המרת המשתנים הקטגוריאליים.

### - QDA

מודל המתאים לביעות עם גבולות לא לינאריים, ומספק ביצועים טובים אם ההנחות הסטטיסטיות מתקיימות. ה - QDA טוב מאוד כשייש צורך להבדיל בין קבוצות נתונים עם התפלגויות נורמליות שונות. הוא מאפשר יכולת ניבוי גבוהה כאשר יש מספיק נתונים, אך הוא רגיש להנחות מסוימות. הוא לא מותאים לפיצ'רים קטגוריאליים רבים ודרש המרת נתונים קטגוריאליים למספריים.

### - Extra Trees

מודל עם ביצועים טובים,iesel ומהיר בזכות יצירת עצים החלטה אקראיים. מותאים לביעות עם רעש גבוה ונתונים לא לינאריים. אין דרוש הרבה כוונון פרמטרים, אבל לעיתים פחות מדויק Random Forest – במקרים שבהם אין הרבה רעש. קשה לפרש את המודל בצורה ברורה, מאחר שהוא מבוסס על מספר גדול של עצים אקראיים. בנוסף מצריך המרת של נתונים קטגוריאליות.

### - TabNet

מודל למידת מכונה מתתקדם מבית PyTorch. המודל מצריך המרת של משתנים קטגוריאליים לקידוד מספרי. היתרונו המרכזי של המודל הוא יכולת לבחור באופן חכם את הפיצ'רים החשובים ביותר לכל דוגמה, תוך כדי ביצועים גבוהים במיוחד. המודל מבצע במידה של אילו פיצ'רים הם הכי רלוונטיים בזמן אימון המודל.



המודל מציע ביצועים גבוהים בדעתה לא לינארי, ומסוגל להתמודד עם בעיות רעש. הוא מתבצע היטב כאשר יש כמות גדולה של דוגמאות, וביצועים מצוינים גם כאשר יש הרבה משתנים קטגוריאליים. יחד עם זאת, המודל דורש מושגים חישוביים גבוהים בזמן האימון, והתהליך יכול לקחת זמן, אך אם תהליכי האימון מותבצע בצורה נכונה, המודל מתבצע טוב מאוד על מגוון בעיות הסיווג והרגולציה.

הרצה עם מספר גבוה מאוד (4000x) של חזרות יכולה לשפר מאוד את הדיקוק של המודל.

### - Ft transformer

מודל מתקדם מבית PyTorch. המודל מצריך המרה של משתנים קטגוריאליים לקידוד מספרי. ההיתרונו המרכזי של המודל הוא השימוש שמאפשר לו ללמידה פיצ'רים תלולים בצורה מאוד ייילה. המודל יכול לבצע באופן אוטומטי את הפיצ'רים החשובים ביותר לכל דוגמה בזמן האימון. המודל מציע ביצועים גבוהים במיוחד על נתונים לא לינאריים, ומثمود בזורה טובה עם בעיות רעש.

### **השוויה מודלי בינה מלאכותית: מודלי מידת מכונה ומאפייניהם**

מודל	תיאור ואופן פעולה	Models
CatBoost	מודל Gradient Boosting שמתפלג במשתנים קטגוריאליים בצורה טבעית. משתמש בטכניקת Ordered Boosting למניעת דילפת מידע, ומתאים במיוחד לנתונים טבלאיים.	Gradient Boosting
Random Forest	מודל מבוסס עץ החלטה שמבצע תחזיות של מספר רב של עצים. מתאים לבעיות סיווג ורגסיה, עמיד בפני רעש ובעל יכולת למנוע overfitting.	Decision Tree
Neural Networks	מודלים ניירניים עמוקים המתבססים על שכבות מלאכותיות. מתאימים למשימות מורכבות כמו עיבוד תמונה ושפה, עם יכולת ללמידה קשורים לא לינאריים בין משתנים.	Neural Network
AdaBoost	מודל Boosting שמחזק תחזיות חלשות על ידי מרתן משקל גבוה לדוגמאות שגויות בכל איטרציה. מתאים לבעיות סיווג ורגסיה, אך רגיש לרעש נתונים.	Boosting
LightGBM	אלגוריתם מהיר ויעיל שמתמקד במבנה עצים באמצעות Leaf-wise Growth. מזהה משתנים קטגוריאליים באופן ישיר ומתאים לנתוני Big Data, אך דורש כוונון מדויק למניעת overfitting.	Gradient Boosting
TabNet	מודל ניירני מבוסיס Attention שמתמקד בלמידת חשיבות הפיצ'רים עבור כל דוגמה. מתאים לנתונים לא לינאריים ולא מאוזנים, אך מצריך המרה של משתנים קטגוריאליים ודורש זמן אימון ארוך.	Attention
XGBoost	אלגוריתם Gradient Boosting עם פונקציות ענישה מובנות למניעת overfitting. מתאים לנתונים בעלי קשרים לא לינאריים ודורש המרה של משתנים קטגוריאליים לפני השימוש.	Gradient Boosting
QDA	מודל סטטיסטי שמניח התפלגות נורמלית רבו-משתנית לכל קטgorיה ומחשב הסתברויות באמצעות חוק ביס. מתאים לגבולות החלטה לא לינאריים, אך דורש נתונים רבים והמרה של משתנים קטגוריאליים למספריים.	Quadratic Discriminant Analysis
Extra Trees	מודל מבוסיס עץ החלטה אקראיים שמנפש את תהליכי בניית העצים ומפחית רגשות לרעש. מתאים לנתונים לא לינאריים, אך מצריך המרה של משתנים קטגוריאליים וקשה לפרש את תוצאותיהם.	Random Forest
FT Transformer	מודל ניירני מתקדם מבוסיס Attention לעובודה עם נתונים טבלאיים ולא לינאריים. מזהה קשרים מורכבים בין משתנים ובוחר פיצ'רים חשובים באופן אוטומטי, אך דורש המרה של משתנים קטגוריאליים וזמן ריצה ארוך מאוד.	Ft Transformer

איור 6 : טבלת מאפיינים של כל מודל





## 2.2. הנחות המודלים – קבלת החלטות + תיעוד :

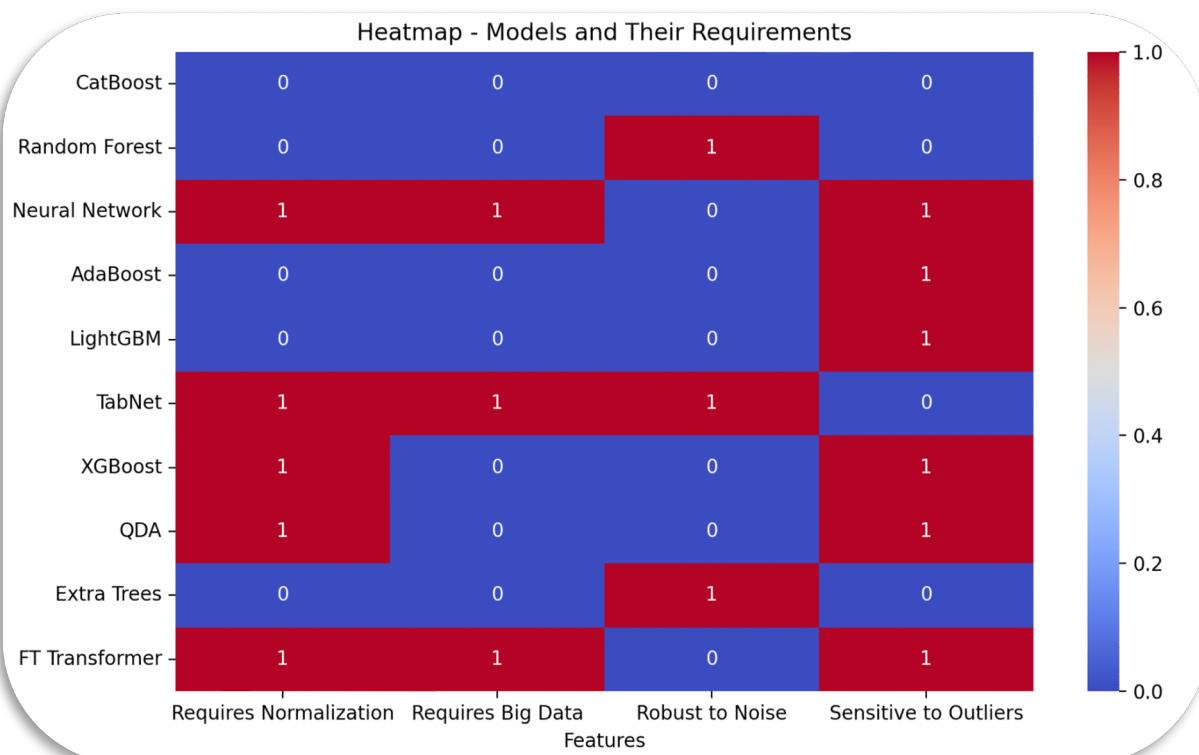
בסעיף זה נציג את ההנחות שעליהן אנו מtabססים בעת בניית המודל. הבנת ההנחות ועמידה בהן חיונית, שכן אי קיומן עלול לפגוע בביטחוני המודל ואף להביא לירידה ברמת הדיקוק שלו.

### **הנחות המודל עbow כל אחד מהמודלים הנבדקים -**

- **CatBoost** - המודל מניח כי לא קיים קשר תלotti ולינארי בין משתנים למשתנה המטרה, בנוסף המודל ידריש נתונים איקוניים כדי לא לפגוע בבדיקה המודל.
- **Random Forest** - כמו Catboost, המודל מניח כי לא קיים קשר לינארי בין משתנים, בנוסף, הגדרה נכונה של מספר העצים חשובה כדי להקטין את רמתה ה- overfitting של המודל.
- **Neural Network** - הנתונים צריים לעבר נרמול לפני אימון המודל, דרוש ביג דטה, דרוש טיפול מתקדים בנתונים – רעש ונתונים חסרים.
- **AdaBoost** - מניח כי כל התוצאות הן שוות משקל, מודל רגיש לרעש.
- **LightGBM** - מניח כי לא קיים קשר לינארי, דרוש ביצוע ניקוי נתונים כדי להימנע מ overfitting ותחזיות מתחזקות, והכנה של פיצירים קטגוריאליים.
- **TabNet** - עמיד לרעש, משתמש בטכניקה של רשת נוירונים لكن דרוש ביג דטה וטיפול מתקדים ונרמול.
- **XgBoost** - אינו מניח קשר לינארי בין משתנים, דרוש נרמול נתונים לשיפור התוצאות, רגיש לחריגות ודרוש טיפול מתאים, הנחה שכל תוכנה תורמת למודל.
- **QDA** - מניח התפלגות שונה, מניח שמטריצת השונות של כל תוכנה שונה, רגיש ודרוש נרמול.
- **Extra tree** - אינו מניח קשר לינארי, לא רגיש לרעש ולא דרוש נרמול.
- **FT transformer** - נדרש נרמול נתונים, דרוש ביג דטה של נתונים, מניח כי יש תלות בין הנתונים ולומד תוך כדי הרצה, רגישות גבוהה אם אין נרמול.



### מייפוי דרישות תפעוליות של האלגוריתמים למידת מכונה -



אייר 7 : דרישות והנחות מודלים

טבלת החום מציגה את הדרישות של מספר מודלים למידת מכונה. בעוד שהעמודות מייצגות דרישות כמו נרמול נתונים, דרישת לביג דאטה, עמידות לרעש ורגישות לחריגות. הצלבים בטבלה מסמנים האם כל דרישת מתקינה או לא, מה שמאפשר השוואת מהירות בין המודלים.



### **ניתן לסייע את המודלים כך -**

#### **մասսայի առաջնային գործությունները**

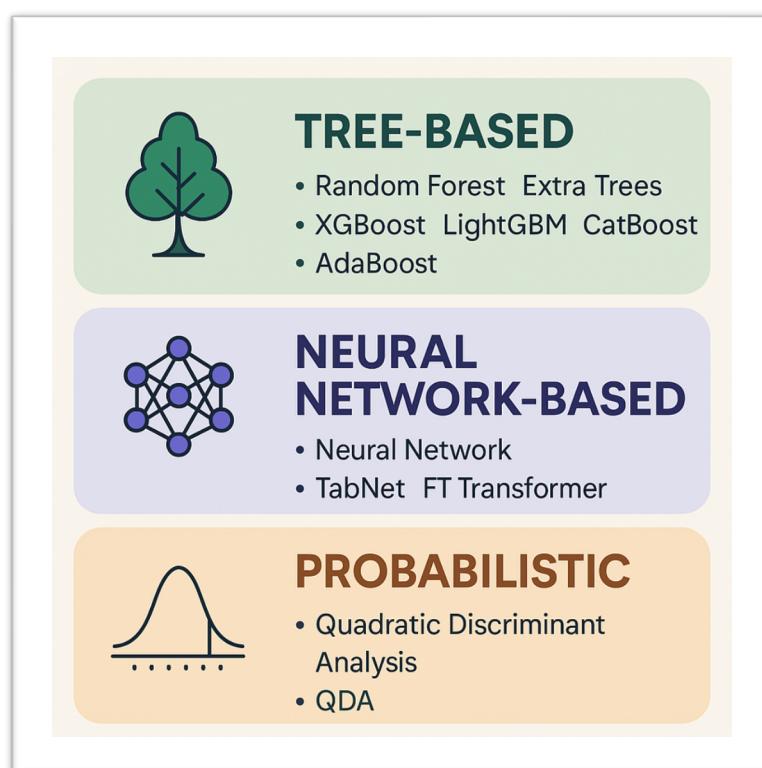
- Random Forest ,Extra Trees ,XGBoost ,LightGBM, CatBoost, AdaBoost
- הנחות עיקריות של מודלים מבוססי עצים החלטה : לא מניחים לינאריות ורגישים לרעש.

#### **մասսայի բազմապատճենային առաջնային գործությունները**

- FT transformer ,TabNet ,Neural Network
- הנחות עיקריות של מודלי רשת נוירוניים : מתמודדים עם קשרים לא לינאריים, לא מניחים שהנתונים מנוורמלים ולכן צריך לבצע נרמול והכנה, רג Ishim לחירגים, ומצריכים ביג דאטָה.

#### **մասսայի հստաբությունները**

- Quadratic Discriminant Analysis (QDA)
- מניח התפלגות נורמלית לכל קטגוריה ומחשב הסתברויות באמצעות חוק בייס, הנחות על שונה בין השונות של התכונות, אין קשר לינארי.



איור 8 : סיווג המודלים



## 2. עיצוב המבחנים למודלים – הגדרת קритריונים :

בסעיף זה, אנו נבחן את הקритריונים השונים שעליים נסתמך לצורך הערכת טיב המודלים ובחירה המודל המתאים ביותר לנוטונים שלנו. הגדרה ברורה של קритריונים אלו היא חשובה מאוד, שכן היא מאפשרת לנו להעריך את ביצועי המודלים בצורה אובייקטיבית, להשוות ביניהם ולבחר את המודל שספק את התוצאות הטובות ביותר.

### kritериונים עיקריים להערכת המודלים -

הקריטריון המרכזי שעליינו נסתמך להערכת המודלים הוא אחוז הדיוק (Accuracy). אחוז הדיוק מציין את כמות התוצאות שהמודל הצליח לנ剖 נכון מתוך כלל התוצאות. למרות שדיוק הוא ממד חשוב, יש לקחת בחשבון גם קритריונים נוספים שכולים לתת תמונה רחבה יותר על ביצועי המודל.

### בין הkritериונים הנוספים -

- מודד את אחוז החיזויים החיוביים הנכונים מתוך כלל החיזויים החיוביים שהמודל ביצע. **Precision**

- מודד את אחוז החיזויים החיוביים הנכונים מתוך כלל התוצאות החיוביות הקיימות. **Recall**

- השילוב בין precision ו-recall, הוא המדד המażן ביניהם ומספק תמונה כוללת של יכולת המודל לחזות את הקטגוריות בצורה מאוזנת. **F1-Score**

### השוואת מודלים שונים -

נבחן מודלים שונים של במידה מפוקחת, כולל מודלים כמו XGBoost, CatBoost, LightGBM, Random Forest, Extra Trees, AdaBoost,

### במהלך הבחירה נבצע את הפעולות הבאות -

- **הרצת המודלים מספר פעמים** – כל מודל יירוץ על מספר חלוקות שונות של הנתונים, תוך שימוש בטכניקות כמו K-Fold Cross Validation, מטרת הגישה זו היא להבטיח שהביצועים שהושגו הם עקביים ולא תלויים בחלוקת מקרים כלשהו של הנתונים.
- **השוואת ביצועים לפי מספר מדדים** – נבחן כל מודל לפי מדדים שונים.
- **הצבת רף סף לבחירה** – המודל שיבחר חייב לעמוד סף ביצועים גבוה במיוחד. במקרה זה, הוגדר סף דיוק של 0.9965 ומעלה, על מנת להבטיח שהתוצאות לא רק טובות – אלא גם יוצאות דופן. רף זה מבטיח שהמודל מספק תוצאות מדויקות ואמינות, תוך שמירה על איכות גבוהה של תוצאות.
- כל מודל יתמודד עם הבעה בצורה שונה ויכול להציג יתרונות שונים בהתאם לאופי הנתונים. המודלים הללו יופרדו לסטים של אימון ובדיקה. המודל ילמד על סט האימון (למשל, 70% מהנתונים) ויבדק על סט הבדיקה (30%). זאת כדי לוודא שהמודל מצליח להכליל על נתונים חדשים ולא רק ללמידה את הנתונים שהיו זמינים לו (כלומר המצלחת להסיק מסקנות חדשות וללמוד מתוכם למקומות הבאים).

### בחירה המודל הטוב ביותר -

לאחר הרצת המודלים השונים מספר פעמים והערכת הביצועים שלהם, נבחר את המודל שהשיג את התוצאות הטובות ביותר על פי המדדים שהוזכו. המודל ייבחן לאורך כמה ריצות כדי



להבטיח שההצלחה לא קرتה במקרה, אלא היא תוצאה של ביצועים עקובים. אנו נציג רף גבויו 매우ו – מעל ל - 0.9965 רמת דיק, בכך לוודא שהמודול שנבחר מספק ביצועים טובים מאוד ומתמודד בצורה טובה עם הנתונים.

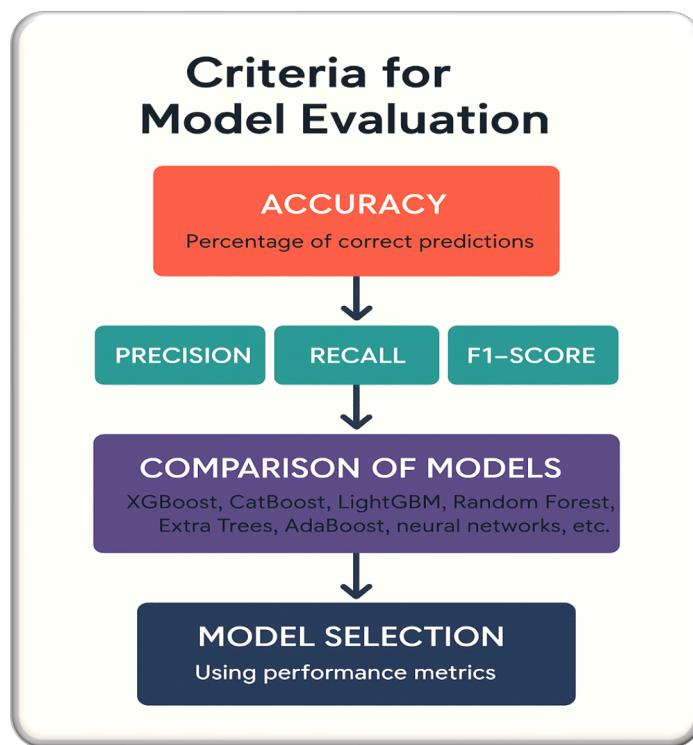
### מודלים ללא פיקוח -

למרות שתמוך במודלים עם פיקוח, ניתן גם לבחון מודלים ללא פיקוח, כגון קיבוץaverages באמצעות Davies-Bouldin Index ו-Silhouette Score, מכיוון שהמודול שלנו מתמקד בלמידה מופוקת, לא נבחן את המודדים הללו בפרויקט זה.

### סיכום,

בפרק זה בחנו את הקритריונים המרכזיים שעל פייהם נבצע את הערכת המודלים שבחרנו, במטרה לוחות את המודל המתאים והיעיל ביותר להתחדשות עם הבעה שניצבת בפנינו. הגדנו כי אמם אחו הדיק יהווה את הקритריון המרכזי, אך נשלב מודדים נוספים כמו Precision ו-Recall ו- F1 Score על מנת לקבל תמונה מקיפה, אמינה ואובייקטיבית יותר של ביצועי כל מודל.

בנוסף, תיארנו את תהליך החשואה בין המודלים, כולל חלוקה לسطוי אימון ובדיקה, הרצה חזורת של כל מודל על פני קבוצות נתונים שונות, ושימוש במידדי ביצוע שנבחרו מראש לצורך הערכה והסקת מסקנות. בחרנו ב轟ון מודלים, כאשר כל אחד מהם מבוסס על מגנון שונה שונה, מתוך כוונה להתאים את המודל לאופי הנתונים היהודי שלנו. למרות שמודלים בלתי מונחים אינם מהווים את מוקד הניותה, התיחסנו בקצרה גם לקריטריונים הרלוונטיים לגבייה. מטרת פרק זה הייתה להציג מетодולוגיה ברורה, שיטית ומדויקת לבחירת המודל הסופי, תוך שמירה על עקרונות של הכללה, עקבות והימנעות מההתאמות יתר (Overfitting).



איור 9 : מדריך המודל



### 3. תיאור המודלים – הרצות ראשוניות + מסקנות :

בסייף זה אנו נבחן את תהליכי בניית המודלים, תוך התמקדות בדגמים שנבנו, בבחירה הפרמטרים, ובשיקולים שעמדו מאחריו כל החלטה. נציג את שלבי הניסוי השונים, את ההתלבבות שנקחו בחשבון, ואת אופן השוואת הביצועים בין המודלים. נראה השוואת בין המודלים השונים ונבדוק את רמת הדיק של כל מודל.

הבחירה במודלים אלה נבעה מהיכולת שלהם להתמודד היטב עם בעיות סיווג מורכבות,עמיות לרעיש, והתאמאה לנתונים בעלי מבנה שונה. בנוסף, חלק מהמודלים מאפשרים ניתוח של חשיבות משתנים.

בסייף זה אנו נבחן את המודל הרלוונטי ביותר עבור הנתונים שלנו, המודל אשר ייתן את התוצאות הטובות ביותר.

- **אנו נבחן את המודדים הבאים להערכתה -**

#### • דיוק (Accuracy) -

מדד זה מייצג את היחס בין מספר התחזויות הנכונות לבין סך כל התחזויות שבוצעו. במלילים אחרות, זהו אחוז הדגימות שסווגו נכון מתוך כלל הדגימות.

- נוסחה :  $(TP + TN) / (TP + TN + FP + FN)$
- טווח ערכים : 0 עד 1, כאשר 1 מייצג דיוק מושלם
- מוגבלת : עלול להטעות במקרים של אי-איזון בין הקטגוריות

#### • Precision (דיוק) -

מדד זה מודד את יכולת המודל להימנע מתחזיות חיוביות שגויות. הוא מייצג את היחס בין מספר התחזויות החיוביות הנכונות לבין סך כל התחזויות החיוביות.

- נוסחה :  $TP / (TP + FP)$
- טווח ערכים : 0 עד 1, כאשר 1 מייצג דיוק מושלם
- שימוש : חשוב במיוחד כאשר העלות של זיהוי חיובי שגוי (false positive) היא גבוהה

#### • Recall (רגישות/הচזרה) -

מדד זה מודד את יכולת המודל לזהות את כל המקרים החיוביים האמיתיים. הוא מייצג את היחס בין מספר התחזויות החיוביות הנכונות לבין סך כל המקרים החיוביים האמיתיים.

- נוסחה :  $TP / (TP + FN)$
- טווח ערכים : 0 עד 1, כאשר 1 מייצג רגישות מושלמת
- שימוש : חשוב במיוחד כאשר העלות של פספוס מקרה חיובי אמיתי (false negative) היא גבוהה

#### • F1 Score

מדד זה הוא הממוצע ההרמוני בין Precision ו-Recall. הוא מאزن בין שני המודדים ונוטן ציון ייחיד שמשקף את ביצועי המודל.

- נוסחה :  $2 * (Precision * Recall) / (Precision + Recall)$
- טווח ערכים : 0 עד 1, כאשר 1 מייצג ביצועים מושלמים



- שימוש : שימושו במיוחד כאשר יש צורך באיזון בין Precision ו-Recall, או כאשר ישנו אי-איזון בין הקטגוריות

#### **כalışו -**

- TP : תחזית חיובית נכונה.
- TN (True Negative) : תחזית שלילית נכונה.
- FP (False Positive) : תחזית חיובית שגויה (סוג 1 שגיאה).
- FN (False Negative) : תחזית שלילית שגויה (סוג 2 שגיאה).

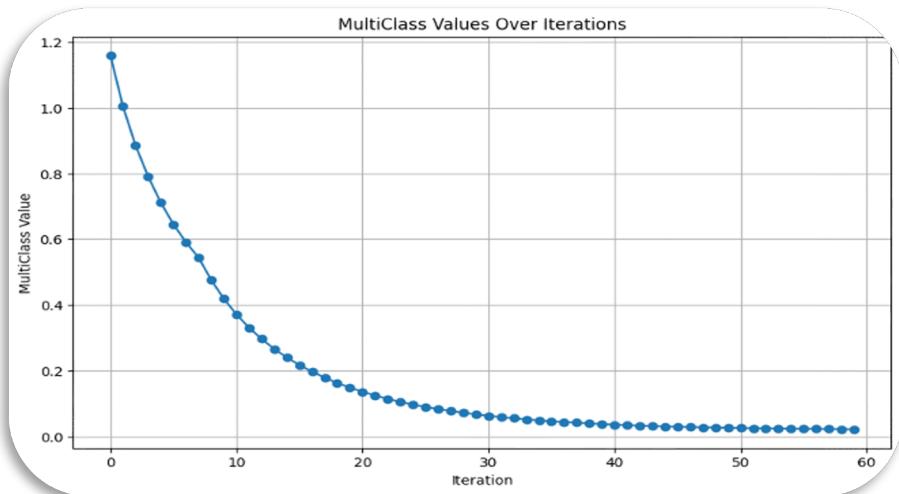
#### **בעת נדיעת המודלים השונים -**

כדי להעריך את המודל **CatBoost** לא הטרכנו להמיר את הנתונים הקטגוריאליים מכיוון שהוא יכול להתחמוד איטם באופן עצמאי, הרצנו את המודל עבור 30 עצים וקיבלו את המידדים הבאים :

```
Number of trees in model: 30
● Accuracy: 0.9991
F1 Score: 0.9992
Precision: 0.9992
Recall: 0.9991
```

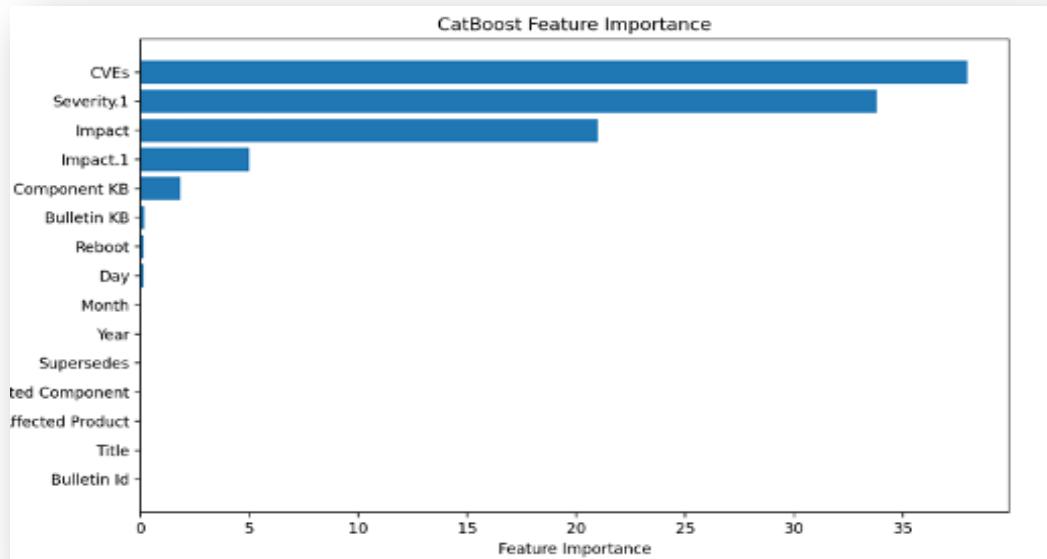
איור 10 : מודל *CatBoost* עבור 30 עצים

לאחר הרצת המודל עם מספר עצים גבוה יותר קיבלנו שהמדדים נעים כלפי מטה וחוזריםשוב עבור מספר עצים גבוה אף יותר. לכן נשתמש ב-30 עצים.  
עבור המודל הזה קבלנו כי **קצב למידת השגיאה** בכל אטרקציית עצ החלטה הוא :



איור 11 : קצב למידת השגיאה פר הרצת

בנוסף חישבנו את התכונות הći משפייעות עבור המודל CatBoost וקיבלו תוצאות טובות מאוד, ניתן להסיק מכך שהמודל יודע להבחן ולנתח את התכונות השונות ושילובם במודל בצורה פרופורציונלית.



איור 12 : מודל CatBoost והמודלים המשפיעים ביותר

נחשב שוב, עם מספר עצים קצר גובה יותר , 100 :

- Accuracy: 0.9994
- F1 Score: 0.9994
- Precision: 0.9994
- Recall: 0.9994
- Number of trees in model: 100

איור 13 : מודל CatBoost עבור 100 עצים

לכן נעדיף 30-100 עצים כדי להימנע מ - overfitting, התוצאות הצפויות שלו מספיקות טובות.

עבור הרצת רשת נוירונית נשמש בפונקציית הפעלה Relu, עם 100 שכבות נסתרות ו-  $a=0.008$  (קצב הלמידה).

- עבור המודלים AdaBoost, Random Forest, Neural Network – קיבלו לאחר הרצת :

Model	AUC	CA	F1	Prec	Recall	MCC
AdaBoost	0.995	0.995	0.995	0.995	0.995	0.990
Random Forest	0.999	0.993	0.993	0.993	0.993	0.986
Neural Network	0.995	0.958	0.958	0.958	0.958	0.920

איור 14 : מדדים של המודלים השונים



## הסבר הדוח -

### מציג ביצועים מצוינים עם AdaBoost

AUC: 0.995 -  
CA (Classification Accuracy): 0.995 -  
 F1 Score: 0.995 -  
 Precision: 0.995 -  
 Recall: 0.995 -  
 MCC (Matthews Correlation Coefficient): 0.990 -

### מציג את ה AUC הגבוה ביותר: Random Forest

AUC: 0.999 - (הגבוה ביותר מבין שלושת המודלים)  
CA: 0.993 -  
 F1 Score: 0.993 -  
 Precision: 0.993 -  
 Recall: 0.993 -  
 MCC: 0.986 -

### מציג ביצועים טובים אך נמוכים יחסית לשני המודלים האחרים: Neural Network

AUC: 0.995 -  
CA: 0.958 -  
 F1 Score: 0.958 -  
 Precision: 0.958 -  
 Recall: 0.958 -  
 MCC: 0.920 -

לפי נתוני הדוח, Random Forest מציג את הביצוע הטוב ביותר ביחס למודול AUC, בעוד Sh- AdaBoost מציג את הביצועים הטובים ביותר בחלוקת המודלים האחרים. הרשות העצפית משגינה תוצאות טובות, אך פחות טובות בהשוואה לשני המודלים האחרים (Neural Network).

## מפורט המודדים -

**AUC (Area Under the Curve)**: מייצג את השטח מתחת לעקומה ROC ומודד את יכולת המודל להבדין בין הקטגוריות. ערך 1 מציין מודל מושלם.

**CA (Classification Accuracy)**: אוחז בתוצאות הנכונות מתוך כלל התוצאות. ממד בסיסי המציג את יכולת המודל לחזות נכון.

**F1 Score**: ממוצע הרמוני בין Precision ו- Recall, מספק ממד מאוזן לביצועי המודל במיוחד כשייש אי-אייזון בין הקטגוריות.

**Precision (איזיק)**: מתחזק כל התוצאות החיוביות שהמודל ביצע, כמה היו נכונות באמת. מודד את יכולת המודל להימנע מתחזיות חיוביות שגויות.

**Recall (רמיישות)**: מתחזק כל המקרים החיוביים האמיתיים, כמה המודל הצליח לזהות נכון. מודד את יכולת המודל לא לפספס מקרים חיוביים.

**MCC (Matthews Correlation Coefficient)**: ממד איזoctה המתחשב בכל ארבעת ערבי מטרייצת הבלבול (TP, TN, FP, FN) ונותן הערכת מאוזנת גם במקרים של נתונים לא מאוזנים.



- עברו המודל **LightGBM**, נחשב את אחוז הדיווק ומדדים נוספים של הדוח לאחר הרצת המודל:

	● Accuracy: 0.996811			
classification_report:				
	precision	recall	f1-score	support
Critical	1.00	1.00	1.00	2591
Important	1.00	1.00	1.00	1993
Low	0.75	1.00	0.86	6
Moderate	0.99	0.97	0.98	114
accuracy			1.00	4704
macro avg	0.93	0.99	0.96	4704
weighted avg	1.00	1.00	1.00	4704

איור 15 : מדדים עברו model lightGBM

#### הסבר חווית -

דיווק כללי - Accuracy - זהו אחוז הדגימות שהמודל סיוג נכון מתוך כלל הדגימות. ככלומר, כמעט 99.7% מהתחזיות היו נכונות.

דווח המפרט את ביצועי המודל לפי קטגוריות classification report.

דווח דיווק precision - מהתוך כל הפעמים שהמודל ניבא קטgorיה מסוימת, כמה פעמים הוא צדק? למשל, אם דיווק = 1.00 לקטgorיה "Critical", זה אומר שככל פעם שהמודל ניבא "Critical", התחזית הייתה נכונה.

דווח ריגישות/recall - מהתוך כל הדגימות שבאמת שייכות לקטgorיה מסוימת, כמה אחוז המודל הצליח להזיהות נכון? recall = 1.00 לקטgorיה "Important" פירושו שהמודל זיהה 100% מהדגימות שייכות באמת לקטgorיה זו.

דווח f1-score - ממוצע הרמוני בין precision ו-recall נתן ממד מאוזן לביצועי המודל. נע בין 0 ל-1, כאשר 1 הוא הטוב ביותר.

דווח support - מספר הדגימות בפועל בכל קטgorיה. למשל, יש 2591 דגימות בקטgorיה "Critical".

דווח macro avg (ממוצע מאקרו) - ממוצע פשוט של המדדים בכל הקטgorיות, ללא התחשבות במספר הדגימות בכל קטgorיה. נתן משקל שווה לכל קטgorיה.

דווח weighted avg (ממוצע משוקל) - ממוצע של המדדים משוקל לפי מספר הדגימות בכל קטgorיה. נתן משקל גדול יותר לקטgorיות עם יותר דגימות.

ניתן לראות שהמודל מתפרקמצוין ברוב הקטgorיות, עם ירידה קלה בדיקון עבור קטgorיות "Low" שיש לה רק 6 דגימות - דבר שהגינוי כי כמות קטנה של דגימות יכולה להשפיע על המודל ללמידה את המאפיינים שלחן כראוי.



• עברו המודל **XGBoost**, המדדים שהתקבלו בדוח הם :

#### Report\_XGBoost

support	f1-score	recall	precision	
2591.0	0.9972983404091090	0.9972983404091090	0.9972983404091090	Critical
1993.0	0.9967377666248430	0.9964877069744110	0.9969879518072290	Important
6.0	0.9230769230769230		1.0	0.8571428571428570
114.0	0.9912280701754390	0.9912280701754390	0.9912280701754390	Moderate
0.9968112244897960	0.9968112244897960	0.9968112244897960	0.9968112244897960	accuracy ●
4704.0	0.9770852750715780	0.9962535293897390	0.9606643048836580	macro avg
4704.0	0.9968190540862620	0.9968112244897960	0.9968409534640020	weighted avg

איור 16 : מדדי מודל XGBoost

#### qda\_classification\_report

accuracy	support	f1-score	recall	precision	
0.9349489795918370	2591.0	0.9604182225541450	0.99266692396758	0.9301989150090420	Critical
0.9349489795918370	1993.0	0.9287894201424210	0.9162067235323630	0.9417225373904080	Important
0.9349489795918370	6.0	0.0	0.0	0.0	Low
0.9349489795918370	114.0	0.0	0.0	0.0	Moderate
0.9349489795918370	0.9349489795918370	0.9349489795918370	0.9349489795918370	0.9349489795918370	accuracy ●
0.9349489795918370	4704.0	0.4723019106741420	0.4772184118749860	0.46798036309986200	macro avg
0.9349489795918370	4704.0	0.9225172042903140	0.9349489795918370	0.9113517019148620	weighted avg

איור 17 : מדדי המודל סטטיסטי QDA

#### extra\_trees\_report

accuracy	support	f1-score	recall	precision	
0.997874149659864	2591.0	0.9984567901234570	0.998842145889618	0.9980717315850370	Critical
0.997874149659864	1993.0	0.9979929754139490	0.9979929754139490	0.9979929754139490	Important
0.997874149659864	6.0	0.9230769230769230		1.0	0.8571428571428570
0.997874149659864	114.0	0.98666666666666670	0.9736842105263160		Low
0.997874149659864	0.997874149659864	0.997874149659864	0.997874149659864	0.997874149659864	Moderate
0.997874149659864	4704.0	0.9765483388202490	0.9926298329574710	0.9633018910354610	accuracy ●
0.997874149659864	4704.0	0.9978784023699700	0.997874149659864	0.9979053387924510	macro avg
0.997874149659864	4704.0	0.9978784023699700	0.997874149659864	0.9979053387924510	weighted avg

איור 18 : מדדי מודל ExtraTree

• נציג את דיקט המודול ומדדים נוספים באחזois עברו המודל TabNet בערך 4000 הרצות עם  
חולקת test של 0.2 ●

#### tabnet\_model

Recall	Precision	F1 Score	Accuracy	
99.1921768707483	99.2038885003237	99.1944765751421	99.1921768707483	0

איור 19 : הרצת TabNet עם פרמטרים שונים



ועבור כ-20,000 תוצאות עם  $test=0.2$  שערך ריצה של כ-**20:25:01** הראה מדדים די דומים :

```
epoch 19931| loss: 0.0119 | oval_0_accuracy: 0.99256 | 1:25:45s
epoch 19932| loss: 0.0138 | oval_0_accuracy: 0.99447 | 1:25:45s
epoch 19933| loss: 0.01144 | oval_0_accuracy: 0.99277 | 1:25:46s
epoch 19934| loss: 0.00978 | oval_0_accuracy: 0.99362 | 1:25:46s
```

איור 20 : מודל TabNet עברו מספר הרצות גבוה

#### ft\_transformer\_model

Loss	Recall	Precision	F1 Score	Accuracy	max_apoches
0.4130721092224120	0.79421768707483	0.7979386997897020	0.7893247284836950	0.79421768707483	500

איור 21 : מודל FT transformer

המודל עלול להיות מוטה עבור חוסר איזון המשתנים במשתנה המטרה, המודל עלול ללמידה להעדיף את התחזית של הקטגוריות הדומיננטיות בלבד – דבר המוביל לירידה חדה בדיק, ב - F1 ובמדדי הרגשות עבור הקבוצות הפחות מייצגות, המודל דורש הכנה מעמיקה של הנתונים והתעמקות מעמיקה. הדיק שמצוג הוא נמוך יחסית לשאר המודלים שנבדקו.

#### סיכום,

בסעיף זה, בחרנו את המודל המתאים ביותר לנתחים שלנו על ידי הרצת מודלים שונים והשוואת ביצועיהם.

- המודל עם דיק גבוה, כאשר השתמשנו ב-30 עצי החלטה, מה שנחשב למספר נמוך CatBoost מאוד עבור מודל זה (בדרך כלל, מספר העצים נע בין 500 ל-1500). היתרון העיקרי של CatBoost הוא טיפול במשתנים קטגוריאליים באופן אוטומטי, שפיישת את תהליכי הכהנה.

רשתות נירוניות - הציגו ביצועים טובים עם פונקציית הפעלה ReLU, עם 100 שכבות נסתרות, וקצב למידה של 0.008. עם זאת, הדיק שהושג לא הגיע לרמות של CatBoost.

- LightGBM - הציג ביצועים טובים במדדים כמו דיק ו- F1-Score עם זאת, גם LightGBM לא השיג את הדיק של CatBoost בשקלול כל המדדים.

- XGBoost - הציג ביצועים טובים, אך לא הצליח להראות מדדים טובים יותר מאשר מודל CatBoost, המודל דרש כוונון מדויק של הפרמטרים (כמו מספר העצים ושיעור הלמידה) כדי למנע overfitting. XGBoost דרש המרה של משתנים קטגוריאליים, בניגוד ל- CatBoost שידע לטפל בהם באופן אוטומטי.

גם המודלים האחרים שבחנו לא הגיעו לרמת הדיק של המודל הקטגוריאלי CatBoost – היה המודל שהציג את הביצועים הטובים ביותר עבור הנתונים, עם דיק גבוה ומדדים מרשימים, לצד פשוטות הכהנה ויכולת להתמודד עם משתנים קטגוריאליים בצורה אוטומטית.



הנתונים והדעתה שנבחרו מציגים תוצאות מצוינות בכל המודלים, דבר המעיד על נתונים טובים וaicוטיים מאוד.

#### סיכום דיקן מודלים לפי רמת הדיקן -

Catboost [100] - 0.9994	.1
Catboost [30] - 0.9991	.2
Extra Tree – 0.997	.3
LightGBM – 0.996	.4
XGBoost – 0.996	.5
AdaBoost – 0.995	.6
Tabnet [19932] - 0.9944	.7
Random Forest – 0.993	.8
Tabnet [4000] - 0.991	.9
Natural Network – 0.958	.10
QDA – 0.9349	.11
Ft transformer – 0.794	.12

המודל עם המודדים הטובים ביותר הוא **CatBoost** עם דיקן גבוהה במיוחד עם מספר לא גבוה במיוחד של עצי החלטה (30).



### 3. הגדרות הפרמטרים – שינוי הפרמטרים בשיטות המידול :

בסייף זה אנו נריץ את המודלים השונים תוך שינוי פרמטרים של כל מודל במטרה להקטין את גודל השגיאה ולהגדיל את דיקט כל מודל שנבחן. הנסה לשפר את המודלים ע"י שינוי מספר שכבות העץ, מספר הרצות ועוד. כדי לשפר את הדיקט של מודלים שונים כמו עץ החלטה, מודלים הסטברותיים ורשתות נוירוניות, יש לשנות פרמטרים שניתן לשנות לכל סוג של מודל.

#### • מודלים שבוסטיים על עצי החלטה:

פרמטרים חשובים שיכולים לשפר את רמת הדיקט -

- הגבלת עומק העץ - הגבלת עומק העץ תסייע לשפר את הכללת המודל ולמנוע overfitting.
- מספר הדוגימות המינימלי - הגדלת הערך יכול למנוע יצירת עץ שיתאים לרעש.
- מספר הדוגימות המינימלי בקצב הצומת - הגדלת ערך תקטין את ה overfitting.
- הקרייטריוון - לחישוב איכות החלוקת. שינוי הקרייטריוון עשוי להשפיע על הביצועים.

#### • מודלים מבוסטיים הסטברות:

פרמטרים חשובים שיכולים לשפר את רמת הדיקט -

- פרמטר רגוליזציה (Regularization Parameter) - מהויה כלי מרכזי במניעת התאמת יתר (overfitting) ובהפחתת השפעת הקולינאריות (מתאים גבוח בין משתנים מסוימים).
- ערך גבוח של פרמטר זה מעודד פשוטות במודל, משפר את עמידתו בפני נתונים חריגים ומפחית רעש, בעוד ערך נמוך מאפשר למודל לזהות קשרים מדויקים ועדינים יותר בין המשתנים ובטיוטו טוב יותר של המורכבות הנתונים. כיוונו נכון של פרמטר זה קרייטי להשגת איזון אופטימלי בין דיקט וכושר הכללה.

#### • מודלים מבוסטיים רשתות נוירוניות:

פרמטרים חשובים שיכולים לשפר את רמת הדיקט -

- גבוח מיידי learning\_rate קצב למידה - קצב נמוך מדי יכול להוביל לאימון איטי מאוד, בעוד שקצב גבוח מיידי עשוי להחמיר את ההתאמה המיטבית ולהוביל ל overfitting או underfitting.
- גודל הקבוצות batch\_size - קבוצות קטנות יותר עשויות להוביל להתאמה אופטימלית, אך ידרשו יותר זמן לחישוב.
- Epochs - מספר ההרצות על כל הנתונים. הגדלת מספר ההרצות יכולה לשפר את הדיקט עד שלב מסוים, אך יש להיזהר מ overfitting - אם אין פיקוח מתאים.
- hidden\_layers מספר השכבות הנסתרות - ככל שמספר השכבות והרווח בין שכבות גדלים, כך המודל יוכל למדוד דפוסים יותר מורכבים, אך עלולה להתעורר בעיות overfitting.

עבור מודלים שונים ניתן לשנות על מספר ההרצות וקצב למידת הדיקט של המודל.

#### לסיכום,

כדי לשפר את רמת הדיקט של המודלים, ביצעו סקירה מעמיקה של פרמטרים מרכזים שיכולים להשפיע באופן מהותי על ביצועי המודלים. פרמטרים אלו נבחנו בהקשר למודלים שהרכזו בשלבים הקודמים של הפרויקט, מודלים מבוסטי עצי החלטה, מודלים הסטברותיים, ומודלים לרשתות נוירוניות מתקדמות.



לצורך כך, חילקנו את המודלים לקטגוריות על פי הסוג שהן מייצגות, ובחנו את הפרמטרים המרכזיים המשפיעים על יעילותם. מטרת החלוקה הייתה לייצר הבחנה בין אופי הפעולה של כל סוג מודול ולהתאים את הפרמטרים הקritisטים לשיפור הביצועים.

המטרה המרכזית הייתה להזות אילו פרמטרים יכולים להוביל לשיפור משמעותית במדדים כמו דיוק (Accuracy), מzd הרגשות (Recall) ו-mdד הדיקוק החובי (Precision).  
באמצעות ניסוי וטעיה, גישה זו מאפשרת להגיע למודל הסופי כשהוא אופטימלי ככל האפשר – מדויק, מואزن, ומתאים בצורה טובה יותר לבעה הشخصית שאוותה אנו מנסים לפתור.



### 2.3. תיאור המודלים – הרצות סופיות + מסקנות :

בסייף זה אנו נריץ שוב את המודלים עם שינוי הפרמטרים השונים, במטרה להגעה לרמת דיקוק גבוהה יותר. כדי לקבל רמת דיקוק גבוהה יותר אנו נבדוק את הרף של הדיקוק וננסה לקבל תוצאות טובות יותר ונציב את התוצאה הטובה ביותר לביצוע לכל מודל.

עבור המודל **CatBoost**, ביצעו שינוי בעומק העצים. כדי להימנע מ- overfitting הגבלנו את עומק העץ מ 3 ועד 9, ואת אחוז החלוקה בקיימות ל - 0.1 לאימון ובדיקה (באיטרציות). הרצינו את המודל עבור 7 עומקים שונים, ו - 9 אחוזי בדיקה ל - 9 test split שונים בסה"כ 63 הרצות. וקיבלנו כי המודל עם הדיקוק הטוב ביותר הוא : **Test Split של 0.4 עם עומק של 6**.

CatBoost best model		
Test Split	Depth	Accuracy
4	6	<b>0.999574784734772</b>

אייר 22 : נסען שיפור מודל עם CatBoost

עבור המודל **Extra tree**, הרצינו עם אחוז חלוקה שונה בקיימות של 0.1 של אימון ובדיקה (באיטרציות), אין שינוי בדיקת הקודם, עבור פרמטרים שונים מתקבלים דיקוקים קטנים יותר.

עבור המודל **FT Transformer**, נשנה את מספר הרצות מ100 ל - 300 כדי שילמד את המודל טוב יותר, ואת חלוקת סט האימון והבדיקה ל - 0.4, קיבלנו דיקוק נמוך יותר מהקודם.

עבור המודל **lightGBM**, נריץ עם המשתנים הבאים תוך שינוי העומק :

```
model = lgb.LGBMClassifier(
    n_estimators=300,
    learning_rate=0.01,
    max_depth=T,
    num_leaves=20,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=0.05,
    reg_lambda=0.1,
    random_state=42
)
```

אייר 23 : מודל lightGBM ושינוי פרמטרים שלו

ככל שהעומק גדול כך מטיב הדיקוק עוד יותר, אך לא מגיע לדיקוק של המודל .CatBoost

```
מטריצה בלבול :
[[3882  11   1   0]
 [ 5 2970   0   0]
 [ 0   0   11   0]
 [ 2   4   1 168]]
depth = 15 ⚪ accuracy = 0.9965981573352233
```

אייר 24 : הרצת מודל lightGBM עם עומק 15



### מפורט לפורמטרים -

- **$n\_estimators=300$**  - מספר העצים (iterations) שירכיבו את המודל הסופי. מספר גבוה יותר של עצים עשוי לשפר את הדיוק, אך גם להגדיל את זמן האימון ואת הסיכון ל-overfitting.
- **$learning\_rate=0.01$**  - קצב הלמידה של המודל. ערך נמוך (0.01) מייצג למידה איטית וזהירה, המאפשרת התכנסות יציבה יותר אך דורשת בכך כל יותר עצים ( $n\_estimators$  מוגברת).
- **$max\_depth=T$**  - עומק מקסימלי של העצים במודל. ערך זה מוגדר כמשתנה T (כנראה הוגדר במקומות אחרים בקוד). עומק גבוה מאפשר למודל ללמידה דפוסים מורכבים יותר, אך מגביר את הסיכון ל-overfitting.
- **$num\_leaves=20$**  - מספר העלים המקוריים בכל עץ. פרמטר זה מגביל את מורכבות העץ ומסייע במניעת overfitting. LightGBM בונה עצים באופן אסימטרי (שלא כמו עצים החלטה וריגלים), כך שניתן להגדיר מספר עליים באופן ישיר. עוזר להכניס אקרואיות למודל ולמנוע overfitting.
- **$subsample=0.8$**  - אחוז הדגימות שייבחרו באופן אקראי (80%) לבניית כל עץ. עוזר לבניית כל עץ. גם פרמטר זה מסייע במניעת overfitting ומספר את יכולת הכללה של המודל.
- **$reg\_alpha=0.05$**  - מקדם רגולרייזציה מסוג Lasso (.). ערך זה עוזר לצמצם את מספר המאפיינים על ידי דחיקת מאפיינים פחות חשובים לאפס, ובכך מעודד פשוטות במודל.
- **$reg\_lambda=0.1$**  - מקדם רגולרייזציה מסוג Ridge (.). ערך זה מסייע במניעת התאמת-יתר על ידי הענשת משקלים גדולים, ובכך מעודד מודל מאוזן יותר.
- **$random\_state=42$**  - משמש לקיבוע תהליכי האקרואיות במודל, על מנת להבטיח שתוצאות הריצה יהיו עקביות וניתנות לשחזור. ערך זה מאפשר ביצוע ניסויים אמינים והשוואה מדויקת בין מודלים שונים, גם כאשר קיימים תהליכי אקרים (כגון חלוקת נתונים או דוגמיה).

עבור המודל **Tabnet** נרץ שוב עם חלוקה של 0.4 ל- test, ועם כ- 12000 חזרות (epochs) ונתקבל דיקן של 0.9929 :

```
epoch 12069 | loss: 0.08775 | val_0_accuracy: 0.77245 | 0:28.21s
epoch 12070 | loss: 0.01067 |oval_0_accuracy: 0.9916 | 0:28:21s
epoch 12071 | loss: 0.01228 |oval_0_accuracy: 0.99192 | 0:28:21s
epoch 12072 | loss: 0.01182 |oval_0_accuracy: 0.99118 | 0:28:21s
epoch 12073 | loss: 0.0114 |oval_0_accuracy: 0.99181 | 0:28:21s
epoch 12074 | loss: 0.00949 |oval_0_accuracy: 0.99256 | 0:28:22s
epoch 12075 | loss: 0.0119 |oval_0_accuracy: 0.99224 | 0:28:22s
epoch 12076 | loss: 0.01078 |oval_0_accuracy: 0.99298 | 0:28:22s
```

אир 25 : מול TabNet עם מספר חזרות גבוהות



עבור המודל **XGBoost** נרץ עם הפרמטרים הבאים :

```
model = xgb.XGBClassifier(
    objective='multi:softprob',
    eval_metric='mlogloss',
    n_estimators=300,
    max_depth=D,
    learning_rate=LR,
    subsample=0.8,
    colsample_bytree=0.8,
    gamma=1,
    reg_alpha=1,
    reg_lambda=1,
    random_state=42,
    use_label_encoder=False
)
```

אייר 26 : הרצת מודל XGBoost ו שינוי פרמטרים של depth & learning rate

#### מפורט לפרמטרים -

- **objective='multi:softprob'** - משמש למטרת של **סיווג רב-מחלקי**, ומצביע הסתברויות לכל מחלקה (ולא רק את התחזית הסופית).
- **Multi-class eval\_metric='mlogloss'** - מدد להערכת ביצועים של המודל לפי **Logarithmic Loss**, נפוץ בעיות סיווג רב-מחלקי.
- **n\_estimators=300** - מספר העצים (iterations) שמרכיבים את המודל. יותר עצים יכולים לשפר דיוק אך גם להעלות סיוכן על account of overfitting.
- **max\_depth=D** - עומק מקסימלי לעץ. ערכי גובהים מאפשרים למודל ללמידה דפוסים מורכבים, אך עלולים להוביל ל account of overfitting.
- **learning\_rate=LR** - קצב הלמידה של המודל. ערך קטן = למידה איטית וזהירה יותר (דורש יותר עצים).
- **subsample=0.8** - אחוז הדגימה של הנתונים בכל איטרציה (80%), מסייע במניעת overfitting על ידי הנסחת אקרים.
- **colsample\_bytree=0.8** - אחוז הפיצרים שנבחרים באקראי לכל עץ. גם פרמטר זה עוזר למניעת overfitting.
- **gamma=1** - ערך סף לרוח המינימלי הדורש לפיצול צומת. ערך גבוה גורם לפחות פיצולים, ככלمر עצים פשוטים יותר.
- **reg\_alpha=1** - מקדם רגולרייזציה מסווג (Lasso). עוזר להוריד משקל של פיצרים חשובים.
- **reg\_lambda=1** - מקדם רגולרייזציה מסווג (Ridge). עוזר למניע משקלים גדולים מדי.
- **random\_state=42** - מאפשר שייחזור תוצאות קבועות ע"י קיבוע האקרים.
- **use\_label\_encoder=False** - מדגל על השימוש ב LabelEncoder הפנימי של XGBoost (כיום כבר לא חובה).



```
T = 0.3 , LR = 1e-05 , depth = 12 , Accuracy = 0.9800141743444366
T = 0.3 , LR = 2e-05 , depth = 12 , Accuracy = 0.9800141743444366
T = 0.3 , LR = 3e-05 , depth = 12 , Accuracy = 0.9800141743444366
T = 0.3 , LR = 4e-05 , depth = 12 , Accuracy = 0.9800141743444366
```

איור 27 : תוצאות XGBoost

ונסה לשפר את דיקט המודל בזכות המודל הטוב ביותר שייצא עד כה CatBoost, עבור המודל הזה נתאים פרמטרים מסוימים שונים ונרייך אותו במספר לוලאות כדי לבדוק את המודל בצורה הטובה ביותר.

```
for D in range (5,7):
    for number in range (130,200,1):
        start_time = time.time() # התחילה
        rand = random.uniform( a: 0.3, b: 0.5)
```

איור 28 : הרצת לוולה ובחירה מספר רנדומלי ל-

- מייצג את עומק בעצים כדי להימנע מ – overfitting •
- מייצג את מספר החזרות לקבלת העץ המושלם עבור כל הריצה. •
- מייצג את המספר הרנדומלי לכל הריצה עבור חלוקה לבדיקה/ מבחן. •

לבסוף אנחנו נציב את הרף הגבוה ביותר שנראה עד כה בדיק, וננסה לשפר את המודל אף יותר.

```
if metrix["Accuracy"] > 0.99955:
    print(f'🎉🎉🎉 Best model found with D={D} and number={number} has an accuracy of {metrix["Accuracy"]:.6f} , '
          f' F1 = {metrix["F1 Score"]:.6f} , Precision = {metrix["Precision"]:.6f} , Recall = {metrix["Recall"]:.6f} <<<<<<<<<
```

איור 29 : תנאי הרף הגבוה ביותר שנמצא

לאחר שנסיים את כל 140 ההצלחות, נשמר באמצעות התנאי את כל ההצלחות הטובות ביותר ביותר וננסה למצוא 5 ההצלחות הטובות ביותר ביותר.

ב ששמנו נמצאים העמודות הבאות :

```
best_models[mon] = {
    'Accuracy': round(metrix["Accuracy"], 6),
    'F1 Score': round(metrix["F1 Score"], 6),
    'Depth': D,
    'Test Split': rand,
    'Iteration': number,
    'start time': start_time,
    'end time': end_time,
    'elapsed_time': elapsed_time
}
```

איור 30 : מילון עם מידע לרלונטי



### בתוך מיליון *best\_models* שמרו את הפורמטוריים הבאים -

- מייצג את הדיוק עד 6 ספרות עשרוניות. *Accuracy*
- מייצג את מzd איזון המודול עד 6 ספרות עשרוניות. *F1 Score*
- מייצג את עומק העצים. *Depth*
- מייצג בצורה רנדומלית את חלוקת קבוצת הבדיקה של המודול. *Test Split*
- מייצג את מספר העצים שהמודול יבנה עד למודל המשלים. *Iteration*
- מייצג את הזמן הראשוני שבו התחיל אימון המודול. *Start time*
- מייצג את הזמן הסופי שבו הסטיים אימנו המודול. *End time*
- מייצג את הזמן שהמודול רץ עד שהגיע לתוכנית (distance timer) *Elapsed\_time*

לבסוף נמיין את ה *Dataframe* מהדיוק הגבוה ביותר לנמוך.

```
df_best_matrices = df_best_matrices.sort_values(by='Accuracy', ascending=False)
df_best_matrices.to_csv( path_or_buf: 'CatBoost best model.csv', index=False)
```

אייר 31 : מין מודלים לפי דיוק ושמירה בקובץ

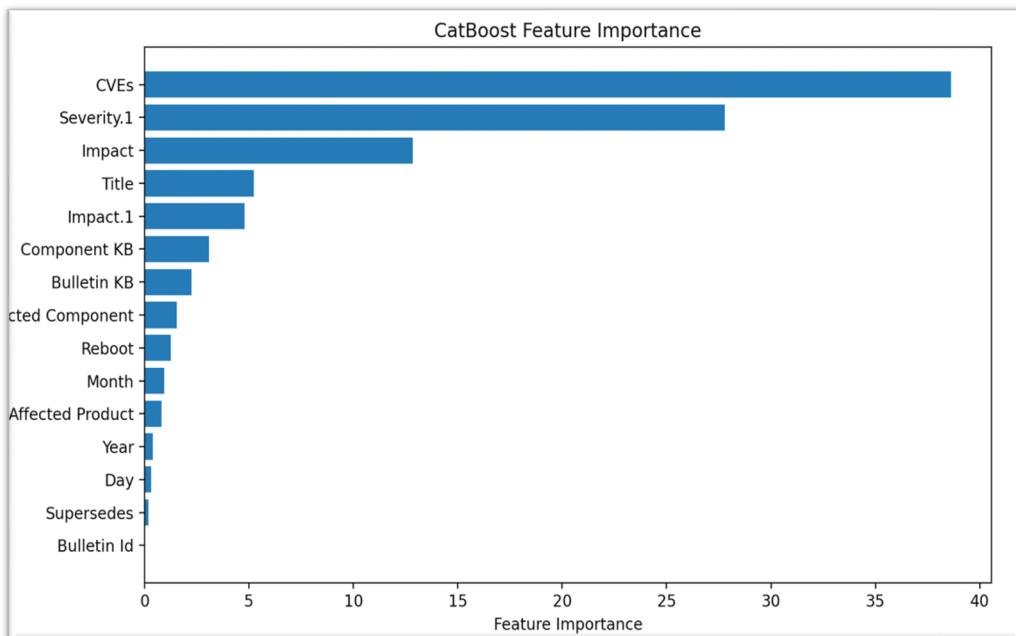
### הרכנו וקיבלונו 34 מודלים בעלי דיוק גבוהה יותר עם עומק של 5,6 -

CatBoost best model								
elapsed_time	end time	start time	iteration	Test Split	Depth	F1 Score	Accuracy	●
1.6187028884887700	1744577097.012460	1744577095.393750	183	0.48115493662725	5	0.999648	0.999646	
1.529327154159550	1744577056.222000	1744577054.6926700	163	0.4766454517904900	5	0.999645	0.999643	
1.2436130046844500	1744576980.811990	1744576979.568370	134	0.4751605636915480	5	0.999643	0.999642	
2.035146951675420	1744577234.965060	1744577232.929920	185	0.47212483147997800	6	0.999641	0.99964	
1.5386080741882300	1744577029.674340	1744577028.135730	154	0.4645018013357010	5	0.999635	0.999634	
1.9362788200378400	1744577197.949450	1744577196.013170	168	0.4633596283534340	6	0.999634	0.999633	
6.420832872390750	1744577026.361230	1744577019.9404000	152	0.45556970375659700	5	0.999628	0.999627	
1.5930700302124000	1744577054.692650	1744577053.099580	162	0.44717528605553900	5	0.999621	0.99962	
1.4924380779266400	1744577129.167440	1744577127.675010	130	0.4462040949708250	6	0.99962	0.999619	
1.7852427959442100	1744577174.928330	1744577173.143080	156	0.4432509613933060	6	0.999618	0.999616	
1.7285339832305900	1744577091.708470	1744577089.979930	180	0.43637574307095700	5	0.999612	0.99961	
2.0547969341278100	1744577223.654360	1744577221.599560	180	0.43461976517565700	6	0.99961	0.999609	
1.8602423667907700	1744577176.7885800	1744577174.9283400	157	0.43555165966278400	6	0.999611	0.999609	
1.902238130569460	1744577127.6749900	1744577125.772750	199	0.4339422491789050	5	0.99961	0.999608	
2.2635021209716800	1744577253.300230	1744577251.036730	193	0.4283009822524310	6	0.999605	0.999603	
1.748016119003300	1744577161.5105500	1744577159.762540	149	0.42603987184114300	6	0.999603	0.999601	
1.7744438648223900	1744577028.1357200	1744577026.361280	153	0.4221999276757360	5	0.999599	0.999597	

אייר 32 : המודלים שהתקבלו לאחר ההרצה



### **曩יג תובנות והקשרים עבור המודל הטוב ביותר שהתקבל כאשר -**



אייר 33 : תוכנות משפייעות על המודל הטוב ביותר ביחסו ברשימה

### **תוצאות :**

- Accuracy : 0.9996464869642068
- F1 Score : 0.9996478384052286
- Precision : 0.9996521739327688
- Recall : 0.9996464869642068

במהלך הניסויים עם המודלים השונים, המטרה המרכזית הייתה לשפר את דיוק המודל ולזוזות דפוסים משמעותיים בנתונים. לדוגמה, מודלים כמו **CatBoost** הציגו יכולת לזהות מאפיינים חשובים בנתונים. בנוסף, המודלים הצליחו להסיק מסקנות משמעותיות בהתבסס על הדיוק ומדדים נוספים, כאשר **CatBoost** השיג תוצאות גבוהות, התוצאות הגבוהות שלו נקבעו מכך שהמודל מותאם לעובדה עם משתנים קטגוריאליים ובנתונים רוב העמודות הם קטגוריאליות.

מבחינת זמן הרצה לכל מודל, הרצנו את המודלים ללא תלות בזמן אך עם הכוונה של פרמטרים נכונים בצורה נכונה, מצאנו חסמים עליונים ותחותונים של המודל **CatBoost** וככה ידענו איפה לתחום את הפרמטרים. זמן הרצה של המודלים לא היה ארוך, אך יש מודלים כמו **TabNet** כמו **CatBoost** שהריצות היו איטיות ומסובכות עבור פרמטרים גבוהים. קראנו את המודלים והגענו לתוצאות דיוק טובות מאוד.



aicoot ha-ntotnim notraha gorim kriti li-hatzlatah ha-modlim. la-ziono be-uyot spatziyot ba-icoot ha-ntotnim, um-nikoi kpdni, ysoodi u-shmirat godl ha-ntotnim um-uybod mo-kdm apshro le-mnuu fgi'ah b-diok ha-model.

#### nbutz diilug :

- |           |  |          |   |
|-----------|--|----------|---|
| .0.999646 | - סט בדיקה : 5, עומק : 48.11%              | CatBoost | 1 |
| .0.999642 | - סט בדיקה : 6, עומק : 47.51%              | CatBoost | 2 |
| .0.99964  | - סט בדיקה : 6, עומק : 47.21%              | CatBoost | 3 |
| .0.999597 | - סט בדיקה : 5, עומק : 42.219%             | CatBoost | 4 |
| .0.999574 | - סט בדיקה : 6, עומק : 40, chzorot : 12000 | CatBoost | 5 |
| .0.992980 | - סט בדיקה : 40, דיווק : 12000             | TabNet   | 6 |
| .0.980020 | - סט בדיקה : 40, דיווק : 40                | XGBoost  | 7 |

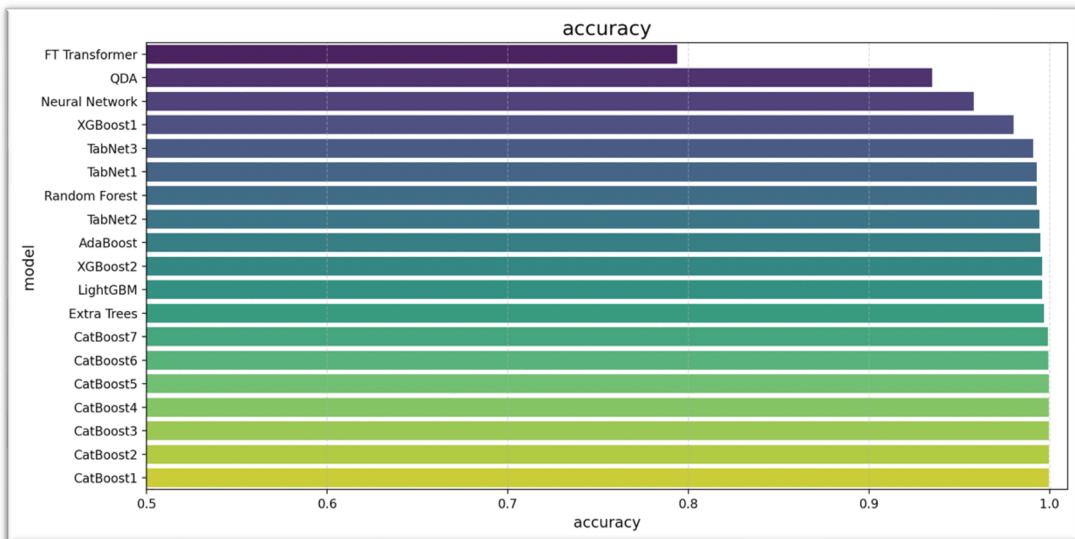
#### l'sic'om,

matrat ha-hatzot shonot hia l-sfar at diok ha-model u-lzotot dafusim meshmuotim ha-ntotnim. basimot ha-hatzot, nshmero ha-modlim ha-tovim bi-yoter, casher ha-model hzign at ha-bitzuim ha-tovim bi-yoter um diok gibba. ha-modlim horatzu um permatrim shonim, u-hatzchnu le-hbivn illo permatrim ooptimaliyim ubor ha-model CatBoost, shmotams b'miyachd le-uboda um mesh'tanim katgorialim. modlim nosfim, como XGBoost, TabNet, u CatBoost, hzign tozotot tovot ak la-bashwoah. -CatBoost, aiicot ha-ntotnim, kol nikoi u-uybod mo-kdm, hyytah chosuba li-hatzlatah ha-modlim u-hpika tozotot tovot maoz.



#### 4. הערכת המודלים – תוצאות + סיכום :

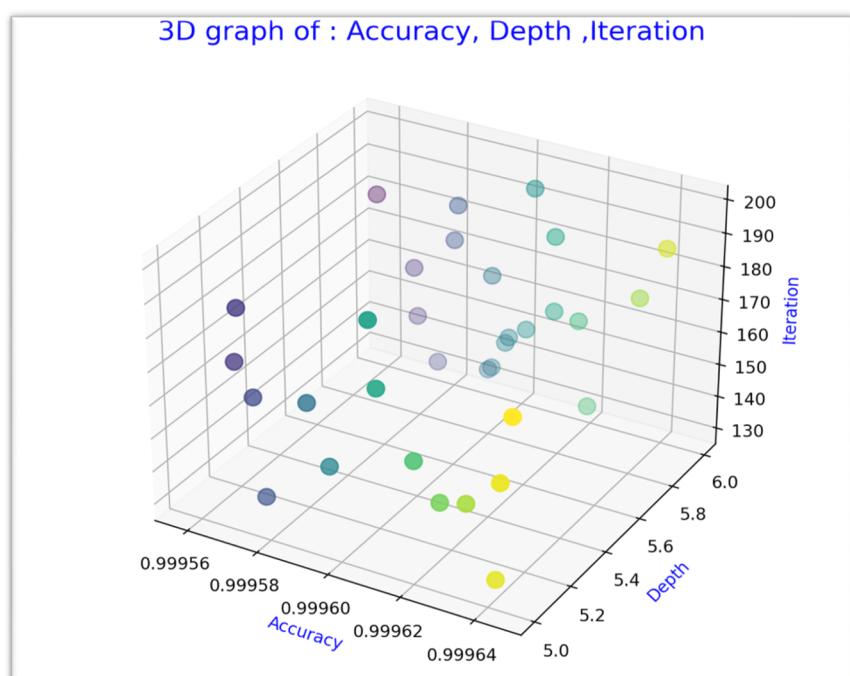
לאחר שהרכנו מספר רב של מודלים, אנו רוצים את המודלים היעילים והטובים ביותר.  
נזכיר את המודלים עם הדיקוק הטוב ביותר :



איור 34 : דירוג מודלים לפי דיוק

המודל שמציע תוצאות טובות במיוחד הוא .CatBoost

הגורף התלת ממדדי הבא מייצג את המודלים השונים של



איור 35 : גраф תלת ממד עבור עומק חזרות ודיוק המודל



כל נקודת פיזור על הגרף מייצגת הרצת מודל עם פרמטרים שונים, היצירום מייצגים את הטווה האפשרי של כל קטגוריה כאשר ציר  $x$  הוא הדיווק.  
**הערכת מודלים : הקרייטריוונים שנבחרו**

בחרנו להתמקד במודל **CatBoost**, מכיוון שביצועיו יכולתו להפיק גם חשיבות תכונות בرمמה גבוהה. בנוסף לעבודת המודל עם משתנים קטגוריאליים תורמת לדיקוק המודל.

#### **הקרייטריוונים להערכת מודלים סופיים הללו :**

- **עומק (Depth)** : 6, עומק מתון ומאוזן.
- **מספר חזרות (Iterations)** : בין 140 ל- 170, כדי למנוע התאמת יתר ולאפשר יציבות.
- **גודל סט בדיקה (Test Size)** : עד 45%, כדי לשמור מספק מידע לאימון מבליפגיעה באיכות הבדיקה.

#### **בין פרמטרי המודלים המוביילים שעוניים על הקרייטריוונים :**

- .Depth=6, Iteration=156, Test=44.32509%
- .Depth=6, Iteration=157, Test=43.555316%

למודלים אלו יש דיקוק גבוה ונמוך הסתכלות מעמיקה על התורמה של כל תכונה לניבוי, דבר שיכל לשיער ביצוע הערכה ולהביא להחלטות טובות יותר.

#### **דירוג המודלים הטוביים -**

כדי למצוא את כל המודלים הטוביים ביותר, נסנן ע"פ הקרייטריוונים :

**תנאי עבור :**  
 עומק 6, חזרות בין 140 ל- 170, גודל קטע 45%

```
df_D = data[data['Depth']==6]
df_I = df_D[df_D['Iteration']>140]
df_I = df_I[df_I['Iteration']<170]
df_finally = df_I[df_I['Test Split']<0.45]
print(df_finally)
```

איור 36 : סינון המודלים על פי קרטריונים שנקבעו

#### **דירוג המודלים הסופיים :**

	Accuracy	Depth	Iteration	Test Split
9	0.999616	6	156	0.443251
12	0.999609	6	157	0.435552
15	0.999601	6	149	0.426040
17	0.999596	6	145	0.421036
18	0.999595	6	143	0.420136
20	0.999591	6	163	0.416123
30	0.999569	6	144	0.394695

איור 37 : תוצאות סינון מודלים



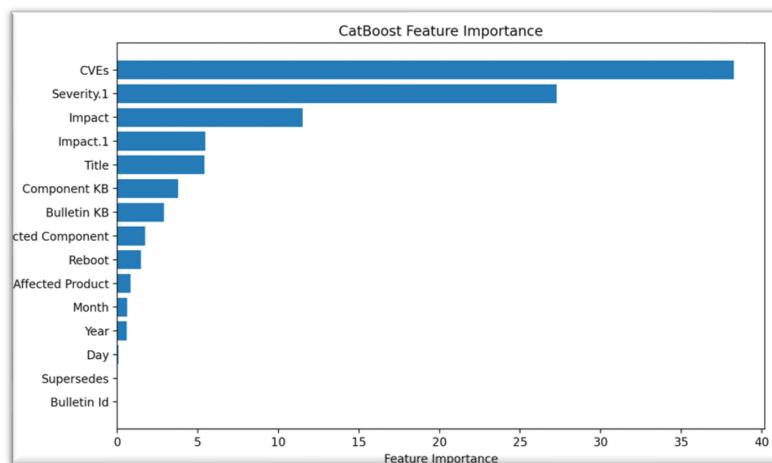
- נבחר את פרמטרי המודל הטוב ביותר עם חzikת הטוב ביותר -

- דיזוק = 0.999616
- עומק = 6
- חוזרות = 156
- מבנן = 0.44325

- המגדדים שהתקבלו -

- Accuracy: 0.9996162701458173
- F1 Score: 0.9996177316220862
- Precision: 0.9996224589047112
- Recall: 0.9996162701458173

- נריצ' את המודל שנבחר -

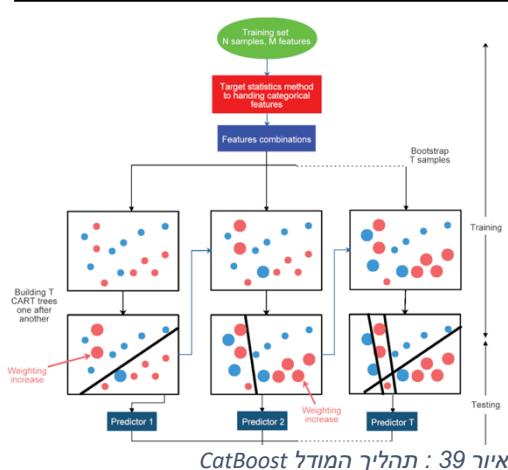


איור 38 : תכונות משמעותיות למודל החזק ביותר עד כה

## לסיכום,

הערכת המודלים שבוצעו למודל **CatBoost**, דיזוק גבוהה ומותאים לנוטונים קטגוריאליים. נבחרו מודלים עם עומק 6, מספר חוזרות בין 140 ל-170, וסט בדיקה עד 45%.

המודל הטוב ביותר שהתקבל הציג דיזוק גבוהה, עם איזון מצויין בין המגדדים השונים. התוצאות מראות שהמודול CatBoost בעל יכולת גבוהה של זיהוי דפוסים וסיפוק ניבויים מדויקים.



איור 39 : תחילן המודל