

Microsoft Security Bulletin Analysis

דוח הכנות הנתונים מוצג על ידי בר כהן וסחר חיים יעקב.

דוח זה עוסק בשלבי הכנה הנתונים כחלק מהליך ניתוח הנתונים בפרויקט. מטרתו היא להעריך את איכות הנתונים שנאספו, לזהות בעיות אפואיות כגון ערכדים חסרים או שגויים, ולבצע שימושים מקדים לניתוח עמוק. התהליך כולל בחירת נתונים רלוונטיים, ניקוי ואינטגרציה, יצירה מאפיינים חדשים, התאמת פורמט הנתונים, וביצוע ניתוח נתונים ראשוני (EDA) כדי לזהות דפוסים ומגוון. דוח זה מספק תיעוד מפורט של שלבי ההכנה שבוצעו, במטרה להבטיח שימוש אופטימלי נתונים להמשך העבודה.



שם מרצה : מל אבי זכאי.

שם מנהה : מוטן לב.

מגייסים :

בר כהן 208110254

סחר יעקב 314741851



תוכן עניינים:

3.....	בחירה הנתונים
3	תכונות ה – Dataset
7.....	ניקוי נתונים
7	שלבי ניקוי הנתונים
12	בנייה נתונים חדשים
13.....	Orange Data Mining
17	שילוב נתונים
19.....	אתר ההורדות של מיקרוסופט
19	עיצוב נתונים
23.....	הסבר על המודלים השונים
23.....	הסביר מגדים
24.....	מסקנות מגדים
26	ניתוח נתונים (EDA)
26.....	טבלאות תדירות





בחירה הנתונים:

בסעיף זה אנו בוחנים את מידת ההשפעה של כל פיצ'ר על משתנה המטרה, Severity, אשר משמש לחיזוי מידת הפגיעה והנזק באירועי האבטחה.

כדי לבחור את הפיצ'רים הרלוונטיים ביותר, ננתח את מטריצת הקורלציה, ונזהה את התכונות בעלות הקשר ביותר למשתנה המטרה (Severity). פיצ'רים שערכם קרוב ל-0 מצביעים על חוסר קשר סטטיסטי ל- Severity, ולכן הם פחות רלוונטיים למודל החיזוי שלנו. בנוסף, פיצ'רים עם קורלציה חיובית או שלילית מובהקת כלומר, ערוכים גבוהים או נמוכים משמעותית מאפס עשויים להיות ממשמעותיים.

התכונות אתם אלו מתחזדים:

```
Index(['Date Posted', 'Bulletin Id', 'Bulletin KB', 'Severity', 'Impact',
       'Title', 'Affected Product', 'Component KB', 'Affected Component',
       'Impact.1', 'Severity.1', 'Supersedes', 'Reboot', 'CVEs'],
      dtype='object')
```

פירוט התכונות ב – Dataset

- Date Posted - תאריך פרסום של הפגיעה מופרד ב- [' , ' או ' /], נפריד את התאריך ליום, חדש ו嬗נה.
- התאריך הוא חסר משמעות, אך ניתן להוציאו מຕוך החודש מכיוון שאם בחודשים מסוימים דווחו על פגיאות, דבר זה יתרום למודל שלנו, מאשר יום ו嬗נה שגם הם פחות רלוונטיים.
- Bulletin Id - מספר מזווה, יכול להיות רלוונטי.
- Bulletin KB - המדריך שפורסם, השתמש בו כפיצ'ר אך סיכון ההשפעה שלו נמוכים.
- Severity - מידת הפגיעה הסופית, **משתנה המטרה**.
- Impact - השפעת הבעיה על המערכת, ניתן להשתמש בו כפיצ'ר והוא רלוונטי למודל.
- Title - כתורת המתארת את עדכון האבטחה, יוכל להשתמש בו כפיצ'ר, עם זאת מידת ההשפעה שלו נראה תהיה נמוכה.
- Affected Product - המוצר או הרכיב שהושפע בעקבות תיקור הבעיה, רלוונטי מאוד.
- Component KB - מדריך שסביר על הרכיב המשופע, אפשר להשתמש בו כפיצ'ר.
- Affected Component - רכיב המערכת המשופע, רלוונטי מאוד!
- Impact.1 - מידת השפעה משנהנית ניתן להשתמש.
- Severity.1 - מידת פגיעה ראשונית, ניתן להשתמש והיא חיובית להבנה ראשונית.
- Supersedes - מצין האם יש בעיה שהעדכון במערכת תיקון, פיצ'ר שיוביל להיות רלוונטי.
- Reboot - מצין האם צריך לבצע פעולה מחדש לאחר העדכון, רלוונטי למודל.
- CVEs - מספק זיהוי בעיית אבטחה, רלוונטי.

קובוצת הנתונים הנוכחית אינטואטיבית מאוד, וכוללת מספר רב של משתנים רלוונטיים. ישנו מספר משתנים אשר עשויים להשפיע ישירות על משתנה המטרה, כגון, כגון, Severity.1, Component, Supersedes, ועוד. בנוסף, קיימים פיצ'רים שיכולים להוסיף מידע מועיל, אך השפעתם עשוייה להיות נמוכה יותר, Component KB, ועוד.

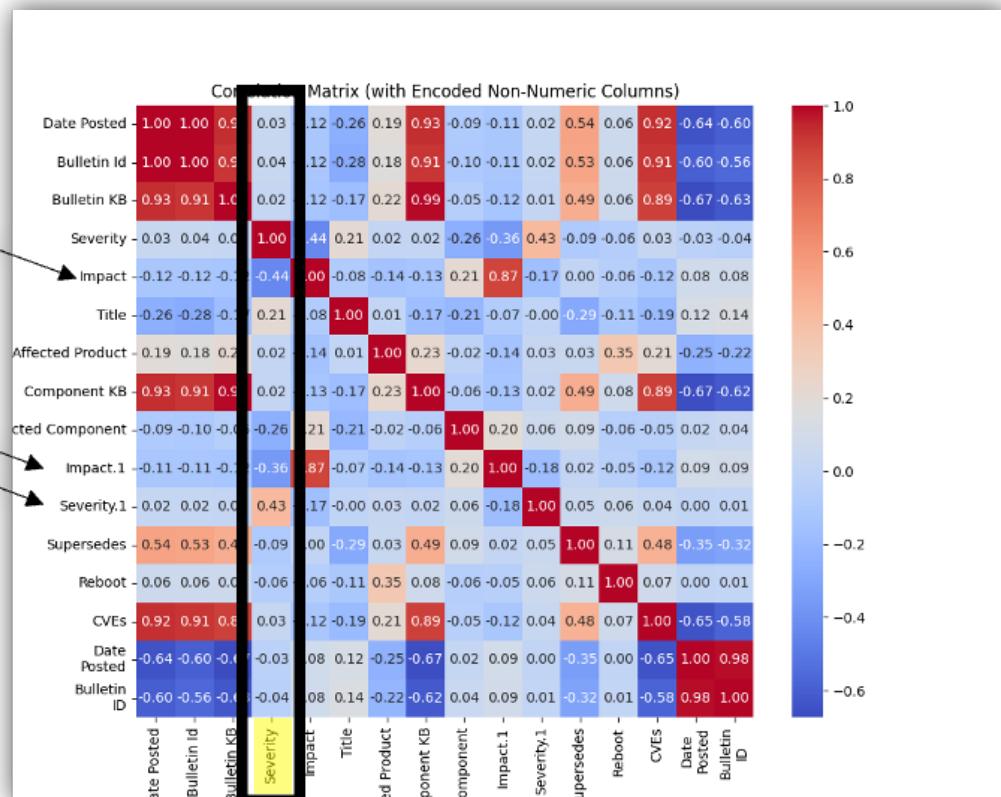


כמויות הנתונים הרחבה יכולה לשמש למודל להפיק תחזיות מדויקות, יחד עם הנדסת תכונות נכונה וניקוי נתונים יסודי.

הaicות הגבוהה של הנתונים והיקף הרחב מאפשרים יצירת מודל חייזי מדויק, במיוחד כאשר משלבים הנדסת תכונות ונקיון נתונים קפדי.

בשלב זה של המחקר, ביצעונו ניתוח מעמיק של התכונות הקיימות בתאונות, במטרה לזהות אילו פיצ'רים עשויים להשפיע בצורה המשמעותית ביותר על המודל. באנז'וות בחינה יסודית של הנתונים ושימוש במטריצת קוורלציה, הצלחנו לזהות קשרים בין משתני הקלט למשתנה המטריה, **severity, impact, impact.1, severity1** – (רמת פגיעות נוספת (ראשונית ומשנית ותיאור עדכון האבטחה) title (רמת פגיעות נוספת (ראשונית), מידת השפעה ראשונית ומשנית ותיאור עדכון האבטחה).

ניתוח זה אפשר לנו להבין אילו פיצ'רים רלוונטיים ביותר לחיזוי רמת הפגיעות, וכייזד הם תורמים למודל. כמו כן, גילינו כי ישנו פיצ'רים מסוימים שהשפעתם נמוכה יחסית, אך עדין עשויים להוסיף מידע מעיל. שלב זה ביחס לשלבים קודמים, מהווה התקדמות משמעותית בתחום פיתוח המודל, שכן הוא מספק תובנות קריטיות לגבי מבנה הנתונים ומאפשר לבצע בחירות מושכלות להמשך האופטימיזציה של המודל.



איור 1 : מטריצת קוורלציה המלאה

בנוסף, נוכל לבצע מבחן Chi-Brיבוע (Chi-Square Test) כדי לבדוק קשרים בין משתנים קטגוריאליים לבין משתנה המטריה. מבחון זה מאפשר להעריך האם קיימת תלות בין המשתנים ולזוזות אילו מהם משפיעים על המשתנה המוסבר. ערך Chi-Brיבוע גבוה מצביע על קשר חזק יותר בין המשתנים, בעוד שערך נמוך מעיד על כך שהקשר בין המשתנים איננו מקרי. ככל שערך ה- χ^2 קטן יותר, כך גוברת ההסתברות לקיומו של קשר מובהק בין המשתנים. השתמש במחון זה לכל משתנה קטgoriy מול משתנה המטריה, כדי להבין את הקשרים ולשפר את בחירת התכונות למודל.



chi Squered values		המשתנים		המטרה
p-value	Chi2	Feature_2	Feature_1	
0.0	38130.0	Bulletin Id	Severity	0
0.0	11842.162801727200	Impact	Severity	1
0.0	33013.17259777990	Title	Severity	2
4.82039883124206E-91	2296.0221257636500	Affected Product	Severity	3
0.0	7202.158837662320	Affected Component	Severity	4
0.0	8897.666566571950	Impact.1	Severity	5
0.0	14545.052266717300	Severity.1	Severity	6
0.0	19192.234879918600	Supersedes	Severity	7
6.65065642261957E-44	208.14738089916100	Reboot	Severity	8
0.0	37594.47824892500	CVEs	Severity	9

אייר 2 : מבחן χ^2 עבור ערכים קטגוריאליים

נבחן כי אלה הם עשרת המשתנים הקטגוריאליים שננתנו את הערכיהם הטובים ביותר. יתר על כן נשתמש בetest נוסף שבודח קשרים בין משתנה המטרה הקטגוריאלי לבין משתנים מספריים. נשתמש ב - **Anova test** :

anova_results		
p-value	F-Statistic	
2.22317917358793E-93	215.74697690438	Year
5.94836033207813E-59	135.01674679980500	Bulletin KB
1.34313183119562E-56	129.52189441005300	Component KB
9.29489246266934E-29	64.76454172160570	Month
0.00011432296753498600	9.080806032555660	Day

אייר 3 : מבחן F עבור ערכים נומריים



אם ערך ה - F גבוהה מאוד, ניתן להסיק כי יש הבדל מובהק בין הקבוצות, ככלומר שקיים קשר משמעותי בין המשתנה המסביר למשתנה המטרה. יחד עם זאת, יש להתחשב גם בערך ה - β , אשר מעיד על רמת המובהקות הסטטיסטית של הקשר – ערך קטן יותר מצביע על כך שההבדלים אינם מקריים.

כחלק מתהליכי בחירת התכונות (Feature Selection), נוכל להתמקד במשתנים בעלי השפעה משמעותית על המודל, תוך סינון תכונות פחות רלוונטיות שעשויה להווסף רעש למודל ולפגוע ביביצועים.

לסיכום,

בשלב זה של המחקר, אנו מתמקדים בבחירה הפיצ'רים הרלוונטיים ביותר לחיזוי משתנה המטרה - Severity, המותאר את מידת ההגעה של עדכוני האבטחה. תהליכי זה הינו קרייטרי להצלחת המודל, שכן הוא מאפשר לנו להתמקד במשתנים התורמים ביותר ולסמן מידע לא רלוונטי עשוי להווסף רעש.

בכדי לבחור את הפיצ'רים המשמעותיים ביותר, ניתחנו את מטריצת הקורלציה וזיהינו את התכונות בעלות הקשרים חזקים ביותר עם משתנה המטרה.

מטריצת הקורלציה מאפשרת לנו להבין את הפיצ'רים, כאשר:
פיצ'רים עם ערכיהם הקרובים ל-0 מצביעים על קשר חלש או היעדר קשר סטטיסטי ל-Severity, וכן הם פחות רלוונטיים למודל.
פיצ'רים עם קורלציה גבוהה (חויבית או שלילית) נחוצים לבני פוטנציאלי גבוה להשפעה על מידת הפגיעה ויכולם להיות מדדים חשובים בחיזוי.

בנוסף, כדי להעריך את השפעתם של המשתנים קטגוריאליים על משתנה המטרה, ביצענו מבחן Chi בריבוע (Chi-Square Test). מבחן זה מאפשר לקבוע האם קיימת תלות בין משתנים Katgorialים שונים לבין Severity. מבחן זה סייע לנו לזהות עשרה משתנים קטגוריאליים שנגנו את הערכים הטובים ביותר, ובכך צמצמנו את מספר התכונות שניתן לכלול במודל החיזוי.

ניתוח משתנים מסוורים – מבחן ANOVA (F-Test)
כדי להעריך את הקשר בין משתנה המטרה הקטגוריאלי למשתנים מסוימים, ביצענו מבחן ANOVA. מטרת מבחן זה היא לבדוק האם קיים הבדל מובהק בין קבוצות המשתנים המספריים ביחס למשתנה המטרה.

תהליכי זה מהווה התקדמות משמעותית בפיתוח המודל, התהליך מספק לנו תובנות חשובות על מבנה הנתונים ומאפשר לבצע בחירות חיויניות להמשך האופטימיזציה של המודל. בעזרת הנדסת תכונות מדויקת, נוכל לשפר את ביצועי המודל ולספק תחזיות מדויקות יותר לגבי רמת הפגיעה של עדכוני האבטחה.



ניקוי נתונים:

בסעיף זה אנו נחקור את הנתונים עצם ונבדוק האם ואיפה ישנו נתונים חסרים - בתוכנות ובמשתנה המטרה. הנתונים שלנו מכילים נתונים חסרים, דבר שיפגע באמינוות ודיקוק המודל, עם זאת ננסה למזער ולמלא את הנתונים החסרים בצורה מועילה ועם כמה שפחות נזק למודל. בנתונים שלנו יש נתונים חסרים הן במטרה והן בתוכנות. נטפל בערכים חסרים, כאשר שלבי העבודה הם: מילוי, הפרדה, הרצת מודל ראשוןי, חיזוי, איחוד.

כહקדמה אנו נבחן את הנתונים שלנו ונבדוק איפה קיימים ערכים חסרים והיכן :

Number of NaN values per column in Data Set:	
Date Posted	4450
Bulletin Id	4450
Bulletin KB	45
Severity	16
Affected Product	2
Component KB	21
Affected Component	12153
Severity.1	426
Supersedes	10485
Reboot	390
CVEs	185
Date\nPosted	19065
Bulletin\nId	19065
dtype:	int64

Data columns (total 16 columns):			
#	Column	Non-Null Count	Dtype
0	Date Posted	19065 non-null	datetime64[ns]
1	Bulletin Id	19065 non-null	object
2	Bulletin KB	23470 non-null	float64
3	Severity	23499 non-null	object
4	Impact	23515 non-null	object
5	Title	23515 non-null	object
6	Affected Product	23513 non-null	object
7	Component KB	23494 non-null	float64
8	Affected Component	11362 non-null	object
9	Impact.1	23515 non-null	object
10	Severity.1	23089 non-null	object
11	Supersedes	13030 non-null	object
12	Reboot	23125 non-null	object
13	CVEs	23330 non-null	object
14	Date		
	Posted	4450 non-null	datetime64[ns]
15	Bulletin		
	Id	4450 non-null	object
	dtypes: datetime64[ns](2), float64(2), object(12)		
	memory usage: 2.9+ MB		

איור 5 : הצגת כמות NaNs בהן נתונים בנתונים

איור 4 : הצגת כמות nulls בהן נתונים

בנוסף קיימים 219 רשומות כפולות אותן נרצה להסיר מתוך הדטה, כדי למנוע הטוית ושיפור אינטראקטיביות הנתונים.

שלבי ניקוי הנתונים:

שלב 1 :

מילוי - בשלב עיבוד הנתונים, חשוב להבטיח שאין ערכים חסרים שעלולים לפגוע בניתוח או בвиיזוע המודל. לכן, נבצע תחילה **_fillna** Impute על כל הערכים החסרים בעוזרת שיטות שונות. ראשית, נחליף את כל הערכים מסוג **None** בערך **NAN**.��' כדי להבטיח שכל הספריות שמבצעות חישובים על הנתונים יזהו אותם כערכים חסרים. לאחר מכן, נתמודד עם הערכים החסרים באמצעות **simple imputer**.

עבור משתנים מסווגים כקטגוריאליים וnominal, השתמש באסטרטגיית **הערך השכיח**, כך שככל ערך חסר יוחלף בערך שモופיע הכי הרבה פעמים באותה עמודה. שיטות אלו נבחרו מכיוון שמלילו עם הערך השכיח עבור משתנים קטגוריאליים מונע הכנסת ערכים חדשים שעולמים לשבש את ההתפלגות הקיימת.



```
def fill_missing_categorical_values(df, strategy='most_frequent'):
    df = df.replace({None: np.nan})
    imputer = SimpleImputer(strategy=strategy, fill_value='missing')
    filled_array = imputer.fit_transform(df)
    df = pd.DataFrame(filled_array, columns=df.columns)
    return df
```

אייר 6 : פונקציית מילוי ערכי null

שלב 2 :

הפרדה - בשלב עיבוד הנתונים, נבצע הפרדה של הנתונים בהתאם להימצאות ערכים חסריים במשתנה המטרה. הפרדה זו מדגישה את שיפור אינכיות הנתונים ומאפשרת גישה מוקדמת לטיפול בערכים החסרים, תוך שמירה על דיקוק ואמינות הנתונים התקינים. לשם כך, נחלק את הנתונים לשני Data Frames נפרדים :

- **מערך נתונים חסר** – יכול אץ ורק רשומות שבוחן משתנה המטרה חסר. קובוצה זו תעבור תהליכי עיבוד מותאים, כגון השלמת ערכים חסרים באמצעות חיזוי מבוסס מודלים, שימוש במומצאים או בשיטות סטטיסטיות אחרות, או לחלופין הסרת רשומות אם החסרים ממשוערים מדי.
- **מערך נתונים מלא** – יכול רשומות שבוחן המשתנה המטרה וכל שאר התכונות מלאים. נתונים אלה ישמשו ישירות לאימון המודל, ללא צורך בהתרבות נוספת.

למה להפריד?

כדי להשלים את אופן הטיפול המדויק והמלא של הנתונים נctrיך לבצע הפרדה ראשונית לנитוח מיידי שיועיל לרמת הדיקוק של המודל הסופי.

טיפול נפרד בנתונים החסרים מאפשר לבנות מודלים ספציפיים להשלמת ערכים, מה שעשויל לשפר את דיקוק התוצאות. לדוגמה, המודל יכול להתמקד בהבנת הדפוסים שבהם ערכים ולבמש שיטות להשלמה באופן יותר מדויק. ניתוח נפרד של הנתונים החסרים עשוי לחשוף דפוסים או מגמות ייחודיות שלא היו ניכרות בניתוח הכלול. ניתן שנמצא קשר בין החסרים לבין תכונות אחרות, מה שיכול לעזור במנון הטברים לתופעות נתונים. בנוסף, הפרדה מאפשרת לנו לנסות גישות שונות על כל קבוצת נתונים בנפרד ולבחר את השיטה המיטבית לכל אחת. לדוגמה, אם אנחנו מטפלים בנתונים חסרים לצורך אחרית מלאה המלאים, נוכל לבדוק שיטות שונות ולהשווות את הביצועים.

הפרדה זו מאפשרת לנו למקד את תהליך הטיפול בערכים החסרים ללא פגיעה בנתונים התקינים, ובכך לשמר את אמינות המודל. בנוסף,

```
def split_by_target_null(df):
    df_null_target = df[df[target].isnull()]
    df_not_null_target = df[df[target].notnull()]
    return df_null_target, df_not_null_target
```

אייר 7 : חלוקת הנתונים לשני dataframes שונים

divide dataframe null in severity



אייר 8 : שם המסמך



שלב 3 :

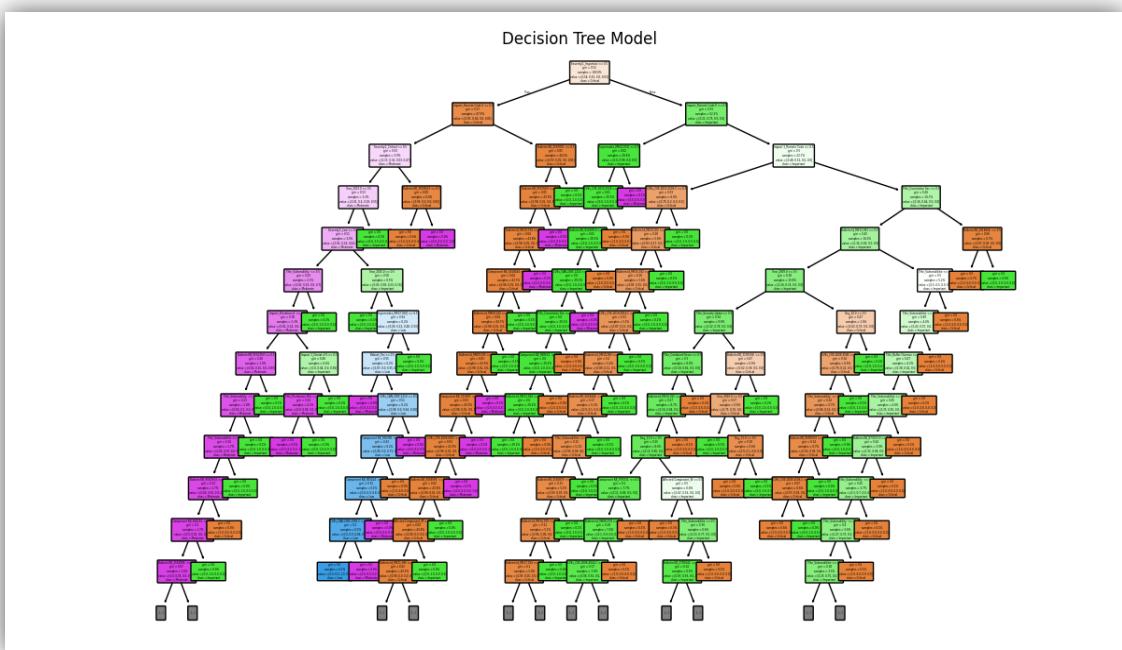
הרצת מודל ראשוני - לאחר עיבוד הנתונים והשלמת הערכים החסרים בתוכנות, נבצע הרצת מודל ראשוני. שלב זה כולל שני תהליכי עיקריים : עיבוד הנתונים המלאים והשלמת הערכים החסרים עבור הנתונים שבthem המשנה המטרה חסר. נבחר במודל עץ החלטה (Decision Tree). עץ החלטה הוא מודל פשוט ויעיל לחיזוי, ויתרונו בכך שהוא מסוגל ללמידה דפוסים לא ניתנים בצורה ברורה וניתנת להבנה. המודל ייבנה על בסיס הנתונים המלאים ויכלול את התכונות הידועות לחיזוי הערכים החסרים של המשנה המטרה, בנוסף הוא יכול לשמש בערכים קטגוריאליים לאחר קידוד בצורה מהירה.

איך יבוצע התהליך?

אימון המודל : המודל מאומן על נתונים מלאים, בהם כל המשתנים (כולל משנה המטרה) מכילים ערכים מלאים.

חיזוי הערכים החסרים : לאחר שהמודל מאומן, אנו מיישמים אותו על Data Frame שבו המשנה המטרה חסר. המודל מנצל את הדפוסים שנלמדו מהנתונים המלאים כדי לחזות את הערכים החסרים של המשנה המטרה. תהליך זה מאפשר לנו לבצע הלמה מובוסת-מודל בצורה שימושתית הקיימת נתונים.

לסיום, מטרת הרצת המודל הראשוני עם עץ החלטה לשפר את איות המודל הסופי.



איור 9 : הרצת מודל ראשוני על הנתונים בעלי משנה מטרה מלא

```
the accuracy of full df with filled nulls in target :
Confusion Matrix for full DataFrame :
[[5113  18   0   2]
 [ 15 4019   0   2]
 [   0    0  12   0]
 [ 15     1   2 208]]
accuracy : 0.9941532901031147
```

איור 10 : דיקן המודל ומטריצה בלבול



שלב 4 :

חיזוי - לאחר הרצת המודל הראשוני, נשתמש בו כדי לבצע חיזוי לערכים החסרים במשתנה המטרית. המודל, שאומן על ה- **Data Frame** ייזה את ערכי המטריה עבור הרשומות שבוחן הם חסרים. כתוצאה לכך, נקבל **Data Frame**, שבו לכל רשותה יש ערך מטריה מסווג, מה שיאפשר לנו להמשיך בניתוחים ובשלבי עיבוד המידע הבאים.

```
if (null_target > 0).all():
    # פיזול הנתונים לשורות עם ובין ארכיים טריים בשום פונק
    null_target_row, not_null_target_row = split_by_target_null(full_df_concat)
    # אוסף מודל טרנינג על נתונים טריים בלבד ומאפשר חזרה
    model, accuracy, null_target_row = tree_model(not_null_target_row, null_target_row, target, name: 'Severity')
    print('the accuracy of first model with part of df not null : (' : null_target_row.shape[0] / full_df_all.shape[0])
    print(f'accuracy : {accuracy} %')
    full_df_all = pd.concat( objs: [not_null_target_row, null_target_row], ignore_index=True)
else:
    print("No null values in target row, training model on full data.") # מודעה אם אין ארכיים טריים בשום הינען
    full_df_all = full_df_concat # שימוש בכל הנתונים ללא צורך בפיזול
```

איור 11 : חיזוי הנתונים החסרים וחיבור הנתונים

23515	MS13-081	293818	Spoofing	Erroneous Microsoft	293818	Windows	I	Spoofing
23516	MS13-081	293818	Spoofing	Erroneous Microsoft	293818	Windows	I	Spoofing
23517	MS13-081	282132	Information	Web Client Microsoft	282132	Windows	I	Information
Important	MS15-023	Yes	CVE-2013	2015	10	8	I	Important
Important	MS15-023	Yes	CVE-2013	2015	10	8	I	Important
Low	MS15-023	Yes	CAN-2001	2015	10	8	L	Low

איור 12 : הצגה מתוך הנתונים המלאים עם 23,517 נתונים

ניתן לראות שלאחר החיזוי וחיבור הנתונים שמרנו על כמות נתונים מלאה להרצת המודל הסופי, תהליך זה יוביל להעלאת הדיקוק של המודל ותרומה למטרת הפרויקט.

שלב 5 :

איחוד - לאחר חיזוי הערכים החסרים במשתנה המטריה, נבצע איחוד של שני ה- **Data Frames**, האחד עם הנתונים המקוריים והמלאים, והשני עם הנתונים שהשלמו באמצעות המודל. פעולה זו מאפשרת לנו לשמור על **הגודל המקורי של הדאטא**, ובמקביל להגדיל את כמות הנתונים הזמינים למודל. כתוצאה לכך, נוכל לשפר את **רמת הדיקוק** של המודל בשל ניצול מרבי של כל הנתונים הקיימים.

```
full_df_all.to_excel(os.path.join('output data' , r'original all non null with target.xlsx'), index=False)
```

איור 13 : איחוד ושמירה של הנתונים המלאים



הסבר על אופן השימוש –

יצרנו תיקייה בשם output data באמצעות ספרייה os בפייתון, שמשיעת ביצירת תיקיות ו שינויי מערכת הפעלה.

לאחר מכן, שמרנו את הקובץ בפורמט excel שמאפשר עיצוב ברור של הטבלה. זאת כדי לשמר על סדר וארגון הקבצים.

בנוסף בעמודות התאריך אנו נתקלנו בחוסר עקביות בקידוד, כאשר מספר רב של תצפיות תאריך מופרדות באמצעות '/' ואחרים באמצעות '.', דבר שהקשה על הפרדה התאריך. כדי להתגבר על חוסר עקביות בתאריך נזחה את המפריד בתאריך ונפעיל:

```
def convert_to_datetime(df, column_name, separator='-' ):  # usage
    df[column_name] = df[column_name].astype(str)
    if df[column_name].str.contains(r'\.', regex=True).any():
        separator = '.'
    elif df[column_name].str.contains(r'\\', regex=True).any():
        separator = '\\'
    else:
        separator = '-'

    try:
        # מפרידים את התאריך לשלבים (יום, חודש, שנה)
        df[['Year', 'Month', 'Day']] = df[column_name].str.split(separator, expand=True)

        # מופכים את המזהה לנתונים ממספריים (אם הם לא כבר כאלה)
        df['Day'] = pd.to_numeric(df['Day'], errors='coerce')
        df['Month'] = pd.to_numeric(df['Month'], errors='coerce')
        df['Year'] = pd.to_numeric(df['Year'], errors='coerce')

        # מוחקים את عمودם המקורי המיותר
        df = df.drop(columns=[column_name])

    except Exception as e:
        print(f"Error occurred while converting date: {e}")

    return df
```

אייר 14 : המרה של תאריך ל 3 עמודות נתונים

לבסוף, נשמרו את ה - **Data Frame**, המעודכן עם הערכים החזוים, כך שנוכל להשתמש בו עבור **מגוון המודלים הטופיים** שלנו. שבירת הפלט תאפשר לנו לבצע השוואה בין **ביצועי המודלים** השונים ולחזור את ההשפעה של שלבי העיבוד על התוצאות. תהליך זה קרייטי לשיפור הדיק ולביצירת המודל האופטימלי, שכן הוא מבטיח שנוכל לנתח, להשוות ולכונן את האלגוריתמים בצורה מבוססת נתונים.



לסיכום,

בפרק זה ביצענו חקר נתונים מקיים וויהינו כי קיימים ערכים חסרים הוו בתוכנות והוו במשתנה המטרה, מצב שעלול להשפיע לרעה על ביצוע המודל. כדי לטפל בכך, השתמשנו בשיטות מלאי ערכים חסרים. לאחר מכן, חילקנו את הנתונים לשתי קבוצות: רשותות עם ערכי מטרה חסרים ורשותות עם ערכים מלאים, כדי שנוכל לבצע השלמות מבוססות-מודל. בשלב הבא, אימנו מודל עצ' החלטה על הנתונים המלאים וניצלנו אותו כדי לחזות את הערכים החסרים במטרה. לאחר שהתקבלו התוצאות החזיות, ביצענו איחוד של הנתונים ושחזרנו את המסגרת המקורית, אך עם מידע מלא יותר. נוסף לכך, זיהינו חוסר עקביות בקידוד התאריכים – חלקים הופרדו על ידי '/' ואחריהם על '-' וכאן ביצענו המרה תקנית כדי לאחד את הפורמתם. בסוף התהליך, שמרנו את הנתונים המקוריים לשימוש עתידי במודלים נוספים ולביצוע השוואות בין שיטות ניבוי שונות. השלבים שננקטו בתהליך זה הבטיחו שימוש יעל במידע הזמן, שיפור הדיקוק של המודל וניהול נתונים אינטואיטיבי יותר.

בנייה נתונים חדשים:

יצירת נתונים חדשים והכנותם לניתוח

בשלב זה, אנו נתאר את הנתונים החדשניים ואת תהליך ייצור הרשותות והעמודות החדשנות.

גירסת נתונים - בשלב הראשון, נבצע גזירת נתונים, הכולמת הפקת עמודות חדשות מתוך עמודות קיימות. לדוגמה :

תאריכים : נתונים הכוללים תאריך ישובצו כך שנפריד אותם ליום, חודש ו שנה. פועלה זו מאפשרת ניתוח עמוק יותר של הנתונים על בסיס זמן, כגון זיהוי מגמות עונתיות או שבועיות.

יצירת רשותות חדשות - בנוסף, נבצע ייצור רשותות חדשות, הכולמת הוספת עמודות חדשות לנתונים שלנו, בהתאם לצורך. זה יכול לכלול הוספה מידע חדש שנאוסף או נוצר במהלך הניתוח.

Year	Month	Day	Severity
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical
2017	3	14	Critical

איור 15 : ייצור עמודות יום, חודש ו שנה מתוך עמודת התאריך



קידוז משתנים "דמיים":

כדי להכין את הנתונים לשלב המודל הסופי, علينا להתחשב במשתנים הקטגוריאליים ולהמיר אותם לייצוג מספרי מותאים. לשם כך, נבצע קידוז משתני דמי (Dummy Variables), שיצור עמודה ביןארית לכל פרמטר בעמודות התוכנה. כך נקבל N עמודות חדשות בהתאם למספר הערכים הייחודיים בכל משתנה קטגוריאלי. פעולה זו חשובה כדי לאפשר למודל להבין טוב יותר את ההשפעה של כל תוכנה ולשפר את ביצועי החיזוי.

ההערכה מדגישה את החשיבות של כל שלב בתהליך ומספקת הסבר מעמיק יותר על הפעולות הנדרשות להכנות הנתונים לניטוח ולחיזוי.

```
df_not_target = df.drop(columns=[target])
df_null_not_target = df_null.drop(columns=[target]) # this df has null in target

df_combined = pd.concat([df_not_target, df_null_not_target])
df_combined_dummies = pd.get_dummies(df_combined)

df1_dummies = df_combined_dummies.iloc[:len(df)]
df2_dummies_oh = df_combined_dummies.iloc[len(df):]
```

איור 16 : המראה של משתנים קטגוריאליים באמצעות דמי



איור 17 : לוגו התוכנה
Mining

:Orange Data Mining

הסבר על תוכנה – Orange – ממשק גրפי

רקע על התוכנה –

הנתונים וההבנה של אלגוריתמים ללמידה מוכנונהaranage Data Mining נועדה לשפר את ניתוח בצורה נגישה, אינטואטיבית ויעילה. התוכנה הושקה כחלק מחקר בתחום אינטלקט צבאי מלאכותית וניתוח נתונים, על ידי צוות חוקרים. היא נועדה לספק כל ניתוח נתונים מתקדמים לגורמים מקצועיים כמו מדעני נתונים, חוקרים, אנשי אקדמיה וסטודנטים, וגם מאפשר למשתמשים פשוטים עם מעט ניסיון בתוכנות להפיק תובנות משמעותיות מנתונים.

הפלטפורמה כוללת יותר מ-100 גדגטים, שהם רכיבים גרפיים SMB צבעים פעולות כמו טעינת נתונים, ניתוחם, והפעלת אלגוריתמים מיידת מכונה שונים. Orange תומכת במגוון רחב של פורמטים של קבצים, כולל CSV, Excel, JSON, SQL, ומסדי נתונים אחרים, מה שמאפשר לה להתמודד עם סוגים נתונים מגוונים.



היא תומכת גם בהפקת ויזואלייזציות אינטראקטיביות ומתקדמות, שמאפשרות להציג את התוצאות בצורה ברורה ונגישה. היא כוללת גרפים, דשborדים, תרשימים ועוד, אשר מסייעים להבין את התוצאות ולהפיק תובנות מהנתונים.

הפלטפורמה מציעה גם יכולות עיבוד נתונים מתקדמות, כולל טיפול נתונים חסרי, סינוו נתונים, ניתוח נתונים קטגוריאליים, הפחחת ממדים (Dimensionality Reduction), ניתוח סדרות זמן, ועוד. בנוסף, היא מציעה כלים לבחירת מודל, חישוף אחר המודל המתאים ביותר לנ נתונים, והערכת המודל באמצעות שיטות כמו cross-validation, confusion matrix, precision, recall.

ממשק הנורפי -

ממשק גרפי הוא אחד הייתרונות, הוא מאפשר יצירת של זרימות עבודה תוך גירירה וחרור של רכיבים. ממשק זה מאפשר ליצור ולבצע תהליכי מורכבים של ניתוח נתונים בצורה ויזואלית, כך שאין צורך בכתיבה קוד. התוכנה גם תומכת בקוד Python, כך שניתן לשלב אלגוריתמים מותאמים אישית בתוך הממשק, או להריץ קוד מתקדם כשייך צורך בכך.

התוכנה פותחה בשפת Python ובסיסת עלייה, כך שהקוד שלו כתוב בעיקר ב-**Python**. כל רכיב בתוכנה הוא מודול Python שנימן להרחב ולחטאים אישית. יתרה מכך, התוכנה מציעה API של Python שמאפשר לעבוד עם הנתונים שבה תכונתיות ומתקדמות, ומספקת את הפעולות של עבודה עם קוד לצד היתרונו של עבודה ויזואלית עם ממשק גרפי.

מה Orange יכולה לעשות? -

התוכנה מספקת כלים מגוונים בתחוםים של ניתוח נתונים, חיזוי, ולמידת מכונה:
טעינת נתונים: ניתן לטעון נתונים מגוון מקורות, כמו קבצי CSV, Excel, MySQL נתונים SQL,
ועוד.

עריכת נתונים: כולל אפשרות לניקוי נתונים, המרת סוגים משתנים (כגון המרת בין משתנים קטגוריאליים למספריים), טיפול בערכים חסרים ע"י impute ומילוי ערכים ע"י ממוצע או השכיח, ושינוי טיפוסים של עמודות.

המרת משתנים קטגוריאליים למשתני דמי: התוכנה תומכת בהמרת משתנים קטגוריאליים למשתנים דמי (One-Hot Encoding).

עיבוד נתונים: מתבצע עיבוד נתונים מתקדם כמו הפעלת מודלים לשיפור נתונים, טיפול בערכים חסרים, וביצוע הפחחת ממדים (PCA).

למידת מכונה: התוכנה תומכת במודלים מתקדמים של למידת מכונה, כולל חיזוי, סיוג, קיבוץ ובמגוון רחב של סוגים מרחקים כמו אוקלידי וקושינוס.

ויזואלייזציות: יש בתוכנה מגוון רחב של כלים להפקת ויזואלייזציות של נתונים ותחזיות, כולל גרפים, תרשימים, וdashboards אינטראקטיביים.
בנוסף המערכת תומכת במגוון רכיבים שנייתנים להתקנה ונitinן לבצע בהם ויזואלייזציות של מודלים שונים כמו מודל ארימה ועוד.

הערכת ותחזית: לאחר בניית המודל, ניתן להריץ את ביצועי המודל על ידי מדדים שונים כמו דיוק, F1 Score, AUC (Area Under Curve), ועוד.



אלגוריתמים במערכת - Orange

התוכנה מציעה מגוון רחב של מודלים של למידת מכונה, בהם ניתן להשתמש לביצוע חיזוי, סיווג, קיבוץ ועוד :

(Classification):

- Decision Trees •
- Random Forest •
- Naive Bayes •
- k-Nearest Neighbors (k-NN) •
- Logistic Regression •
- Neural Network •

(Clustering):

- K-Means •
- Hierarchical Clustering •

(Regression):

- Linear Regression •
- Ridge/Lasso Regression •

(Time Series Forecasting):

- Time Series Forecasting •

(Dimensionality Reduction):

- PCA (Principal Component Analysis) •
- t-SNE •

המאפשרים ניתוח ויצירת מודלים חיזוי מדדיים. PCA, Decision Trees, K-Means, Linear Regression, K-Means, t-SNE, PCA, Time Series Forecasting, Classification, Clustering, Regression, Dimensionality Reduction.

הערכת מודלים ב-Orange

לאחר אימון המודל, ניתן להעריך את ביצועי המודל על ידי מדדים שונים. Orange מאפשרת הערכה של המודל במספר דרכים :



Cross-validation: מאפשרת לבדוק את ביצועי המודל בצורה לא מוטה, על ידי חלוקה אוטומטית של הנתונים ל-k קבוצות.

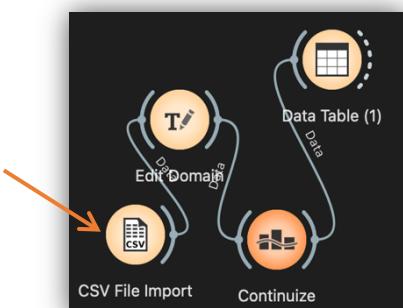
Test & Train Split: חלוקה של הנתונים לסטים נפרדים של אימון ובדיקה.

Confusion Matrix: להערכת ביצועים של מודלים לשיווג, כולל מדדים כמו דיקוק, זיכרון, F1 Score, Precision, Recall.

ROC Curve & AUC: להציג ביצועי המודל על ידי גרף המתאר את יחסי FPR ו-TPR (True Positive Rate) המתקבלים בכל סף של המודל.

באמצעות אלו, ניתן להעריך את איכות המודל ולבצע אופטימיזציה אם יש צורך בכך. ניתן לחושף קוד מותאם אישית של פיתון.

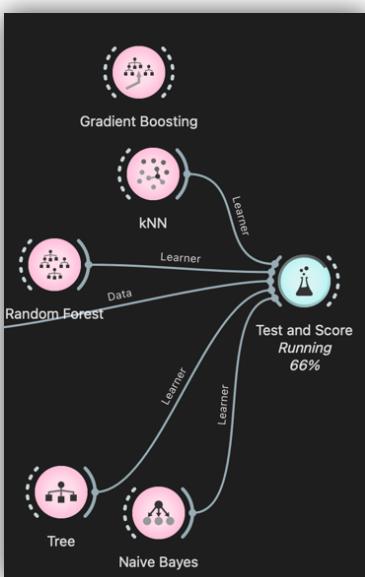
תוכנת Orange Data Mining היא פלטפורמה גמישה וחזקת שמאפשרת ביצוע ניתוחי נתונים מתקדמים, יצירה מודלים של למידת מכונה והפקת ויזואלייזציות של נתונים בצורה אינטואיטיבית. עם התממותה בלמידת מכונה, הערצת ביצועים, וטיפול נתונים, Orange מציעה את הכלים הנדרשים לחוקרם, מדעני נתונים ומשתמשים מתחילה כאחד.



איור 18 : תהליך :
orange data mining

איור 19 : תמונה זו מראה את תהליך encoder של התוכנה בנוסף לשינוי סוג הנתונים והציגתם בטבלה.

טעינת הנתונים ← שינוי סוג הנתונים ←
ցגה בטבלה את טבלת ← encoder



איור 19 : תהליך הערצת מודלים
orange data mining

איור 20 מציג את תהליך יצירת המודלים השונים לצד שלב הערכתם. התהליך מתחיל מחיבור הנתונים אשר עברו עיבוד מוקדם בשלבים הקודמים, ולאחר מכן הם מזומנים למודלים שונים לצורך למידת מכונה. כל מודל מקבל את הנתונים שנוקו, הומרו והתאימו לפורט אופטימלי ללמידה, ובונה מהם דפוסים על פי האלגוריתם שבו נעשה שימוש.

בשלב הבא, מתבצעת הערצת המודלים באמצעות רכיב Test and Score Running, אשר אחראי על בדיקת ביצועי המודלים שהוכשרו. רכיב זה מחלק את הנתונים לסטים אימון ובדיקה, מריץ את המודלים על הנתונים ומחשב מדדי ביצועים שונים. המדדים כוללים בין היתר:

F1-Score – מדד מאוזן הבוחן את איכות המודל תוך שקלול Precision ו-Recall.

Precision – אחוז התוצאות החיוביות שנמצאו נכון מתוך כלל התוצאות החיוביות.

Recall – אחוז הדגימות החיוביות שהמודל הצליח לזהות מתוך כלל הדגימות החיוביות.



לאחר חישוב, אנו נוכל לייצר את מטריצת הבלבול עבור כל אחד מהמודלים ולראות את מידות הנכונות של כל מודל.

בתמונה זו חישבנו מודלים שונים ובנייהם, עץ החלטה, נאייב בייס ועוד.

לסיכום,

בשלב זה של הפרויקט, אנו יוצרים נתונים חדשים ומכינים אותם לניטוח. תחילת, אנו גוזרים נתונים מעמודות קיימות, כמו פירוק תאריכים או חישוב כמות מכירות. לאחר מכן, אנו מחדים את כל מסמכיו האקסל למסד נתונים אחיד. כדי להיערך למודל, אנו ממירמים מושנים קטגוריאליים לערכים מספריים באמצעות קידוד משתני דמי. לבסוף, אנו בוחרים את המודל בעל הביצועים הטובים ביותר לחיזוי מדויק ואמון.

שילוב נתונים:

בשלב זה אנו נתאר שלב שלילוב הנתונים, תהליך מיזוג וצירוף הנתונים שלנו מתוך אתר ההורדות של מיקרוסופט. הנתונים מכילים מגוון מסמכים שמתווארים באתר ההורדות :

CVRF Information.docx – מידע על פורמט דיווח CVRF
BulletinSearch2001-2008.xlsx – קובץ Excel עם מידע על תיקוני אבטחה בין השנים 2001-2008.

MSRC-CVRF.zip – קובץ ZIP הכולל תיקוני אבטחה בפורמט CVRF החל מינואר 2012.
BulletinSearch.xlsx – קובץ Excel עם מידע על תיקוני אבטחה מחודש נובמבר 2008 ועד היום.

מה הם קובצי cvrf

קובץ (Common Vulnerability Reporting Framework - CVRF) הוא פורמט מבנה נתונים סטנדרטי שנועד לשימוש במתן דיווחים אודוט פגיעות אבטחה במערכות מידע – לוגים במערכת.

בין הרכיבים העיקריים בקובץ CVRF :

Document Title : כותרת המסמך.

Document Publisher : פרטי המפרסם, כולל פרטי יצירת קשר.

Product Tree : רשימה של המוצרים הפגיעים.

Vulnerability Information : פרטים על הפגיעות, כולל תיאור, CVE (מספר מזהה של פגיעה אבטחה), ורמת סיכון.

Threats : תיאור של האיום שנגרמים מהפגיעה.

Fixes or Mitigations : פתרונות או אמצעי מניעה להפחחת הסיכון מהפגיעה.

Document Tracking : מידע על גרסאות ושחרורים של המידע.



המסמכים כוללים שני מסמכים אקסל מופרדים שאיתם נעבד, אותם אנו צריכים לצורף באמצעות append וליצור את המסמך שאיתו נעבד.

```
# הוראת קובץ תרגולינו
Files = ['Bulletin Search (2008 - 2017).xlsx', 'Bulletin Search (2001 - 2008).xlsx']
Data_Frame = pd.concat([pd.read_excel(file) for file in Files], ignore_index=True)

# אכיפת סדר על ידי פידיש אינדקסים
Data_Frame = Data_Frame.reset_index(drop=True)

# שמירת הדאטה למגילה
with pd.ExcelWriter('Merged_Bulletin_Data.xlsx') as writer:
    Data_Frame.to_excel(writer, index=False, sheet_name='All Data')
```

אייר 21 : שילוב קבצי האקסל לקובץ אחד



אייר 22 : שם הקובץ המאוחד

הסבר על אופן השמירה
השמירה כאן היא שונה מהדרך הרגילה, ההבדל הוא יצרת הקובץ וביצוע שינוי כתיבה בפורמט Excel, לעומת שמירה רגילה כמו ב-csv . זה מאפשר להוסיף נתונים רבים לאוטו הקובץ במספר גילוונות (sheets), במקרה זה גילוון בשם 'All Data'.
ההבדל הוא שהקובץ נשמר באופן מובנה ונתמך ביכולות נוספות כמו גילוונות רבים.

```
df1 = pd.DataFrame({'name': ['oz', 'berri'], 'salary': [2300, 4020]})

df2 = pd.DataFrame({'X': [7, 8, 9], 'Y': [10, 11, 12]})

with pd.ExcelWriter('example.xlsx') as writer:
    df1.to_excel(writer, sheet_name='Organization', index=False)
    df2.to_excel(writer, sheet_name='Point', index=False)
```

אייר 20 : שמירה של 2 גילוונות שונות לקובץ אחד

בדרך זו ניתן לשמר מספר נתונים בגילוונות נפרדים בקלות , לדוגמה :



אייר 21 : שמות הגלגולות

כאן יצרנו קובץ אקסל בשם example שמכיל 2 גילוונות בשם organization, point :



אתר ההורדות של מיקרוסופט:

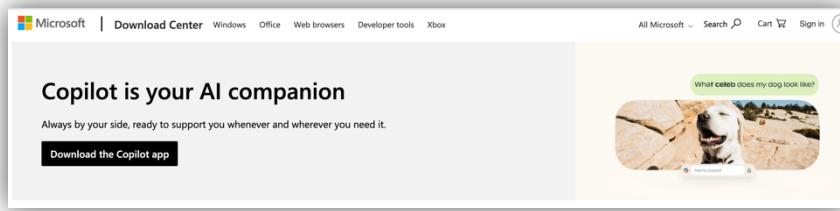
מרכז ההורדות של Microsoft הוא מקור رسمي להורדות תוכנות, עדכוןים וכליים שונים של Microsoft, כגון Windows, Office, Visual Studio ועוד. במאצ'וטו, ניתן להוריד גרסאות עדכניות של תוכנות, חבילות שירותים ועדכוני אבטחה.

האתר מתעדכן באופן קבוע, עם הוספהغرסאות חדשות, עדכוןים ותיקונים בהתאם לצורך ולמורים החדשניים של Microsoft. תדריות העדכוןים משתנה בהתאם למוצר ולגרסה, אך Microsoft מקפידה לספק עדכוני שוטפים כדי להבטיח את אבטחת וביצועי המוצרים שלה.

חלק ממטרותיו של אתר ההורדות הם : הפצת תוכנות , שיפור אבטחה ועוד .
הפצת תוכנה רשמית : להציג למשתמשים גישה ישירה להורדה של מערכות הפעלה, חבילות תוכנה, כלים לפיתוח, אפליקציות, ועוד, כולל באישור ובתמכה של Microsoft.

עדכוני ושדרוגים : מספק עדכוני שוטפים לכל התוכנות המיוצרות על ידי Microsoft, כולל תיקוני אבטחה, שדרוגים לגרסאות חדשות ותיקוני ביצועים.
שיפור אבטחה : מהבטיח שימושימי מיקרוסופט מקבלים את הגרסאות העדכניות ביותר של תוכנות, דבר שמשמעותו הגנה על המידע ובשמירה על ביצועי המערכת.
הדרגות וכליים לפיתוח : להעניק גישה למדריכים, כלים ו-API.

האתר מציע קטגוריות שונות להורדות, כולל ביניהם :
Windows : הורדות עבור מערכות הפעלה, כולל עדכוניים וכלי עזר.
Office : הורדות עבור חבילת Microsoft Office, כולל עדכוניים וערכות שפה.
Developer Tools : כלים למפתחים, כגון Visual Studio .
עדכוני אבטחה למורים שונים של Microsoft : Security Updates



אيو 22 : אתר ההורדות של מיקרוסופט

אתר ההורדות של מיקרוסופט

עיצוב נתונים:

השלב של בחירת המודל הסופי בתהליך ניתוח נתונים הוא קריטי להצלחת המודל וליכולת להפיק חיזויים מדויקים ואמינים מהנתונים. כדי להשג תוצאות מיטביות. בשלב זה נבוד בשיטתיות על מנת להבטיח מודל חזק ואמין , עם זאת חשוב לציין כי לכל מודל יש חסרונות מסוים וננסה למצמצם את הבעיות של המודל הסופי שנבחר.

בחירת המודל הסופי

בסוף התהליך, נבחר את המודל בעל הביצועים הטובים ביותר, שיספק את החיזוי המדויק והאמין ביותר עבור הנתונים שלנו. התהליך כולל השוואת מודלים שונים ובחירה המודל שמספק את התוצאות הטובות ביותר בהתאם למטרות הנิตוח.

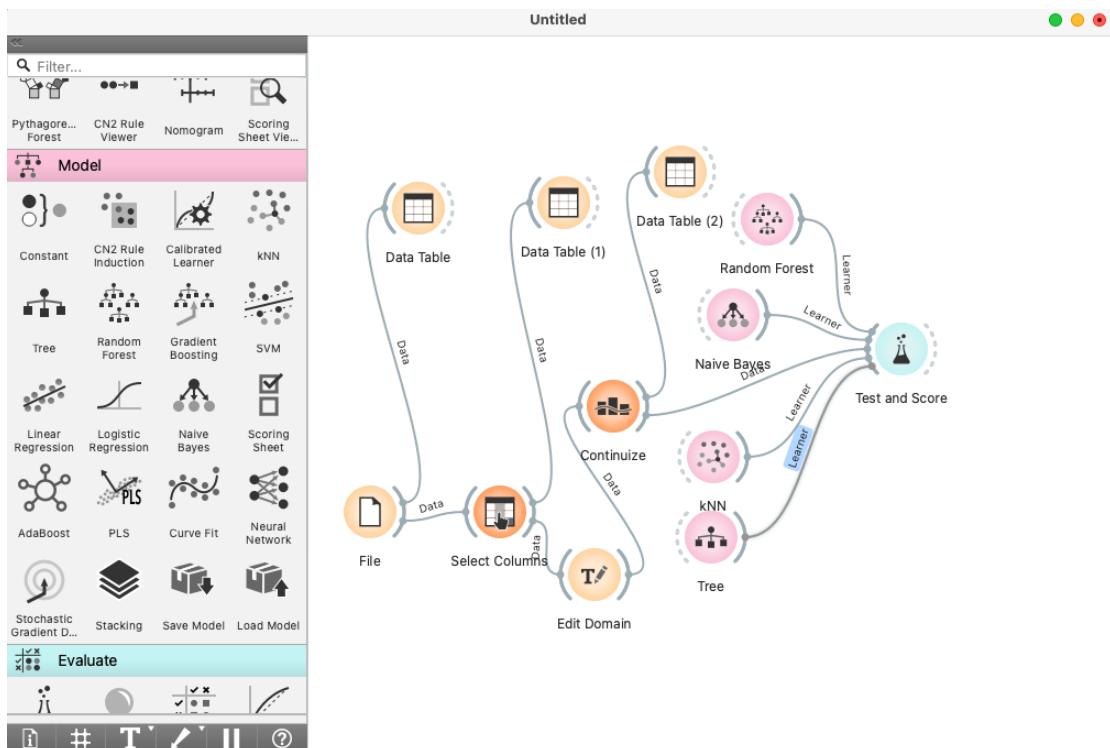


איך נבחר את המודל הסופי?

נבחר את המודל הסופי על ידי הרצת מגוון מודלים באמצעות תוכנת **Data Mining** לאחר שביצעונו הכנה מקיפה של הנתונים, כולל יצירתיות דמי ובחירה השוררות הרלוונטיות לניטות. לאחר הפעלת המודלים, נבחן את ביצועיהם בהתאם לקריטריונים שנקבעו, ונבחר את המודל שמציג את הביצועים הטובים ביותר והחיזוקים המדוקים ביותר עבור הנתונים שלנו.

שיטת העבודה לבחירת המודל הטוב ביותר היא הרצת מגוון מודלים שונים והערכת ביצועים שלהם.

הבחירה במודל הסופי היא שלב מרכזי בתהליך, נבצע אותה בצורה מסודרת על מנת להבטיח שהמודל יפיק את התוצאות המדוקינות והאמינות ביותר. הרצת מגוון מודלים, בדיקת ביצועיהם ושיפורם במידת הצורך, ולאחר מכן בחירת המודל שספק את הביצועים הטובים ביותר, מאפשרת להפיק את התוצאה המקסימלית מהנתונים.



איור 23 : תהליך מלא ב-*orange data mining* להערכת מודלים

לאחר מילוי הנתונים המקיים שהצינו בשלבים הקודמים,icut נרצה למצוא את המודל הייעיל והטוב ביותר מתוך המודלים הקיימים, ביניהם רשת נוירונים, עץ החלטה, נאיב בייס ועוד, התמונה מתארת איך אנו יכולים ליצור באמצעות תוכנת *Data mining* את התהליך החל מהעלאת הנתונים ועד מדידת ביצועי המודל. התמונה היא המשך המליא של *Orange data mining*, נפרט את שלבי התהליך של התוכנה:

- .1 – העלאתקובץ האקסל לאחר עיבוד ומילוי הערכים החסרים.
- .2 – בחירת העמודות הרצויות ובחירה משתנה המטרה, Severity – Select columns



- הפייצרים הם : Affected Product , Affected component , Title , Supersedes , CVEs , Reboot , Impact , Impact.1 , Component KB , Severity.1 , Year , Month Bulletin KB , .
- .3 – שינוי סוג הפייצרים, בתנאים שלנו ישים ערכים שאומנים מיוצגים כמספרים לצורך התייחס אליהם כמשתנים קטגוריאליים.
 - .4 – קידוד הערכים לפי משתנה דמי.
 - .5 – הערכת המודלים לפי הנתוני דמי.
 - .6 – חישוב מגוון מודדים על מודלים שונים.

שלבי בינויים:

– הצגה של המשתנים כטבלה של data table , data frame למשל (2) מציג את משתני הדמי של התוכנות.

– הרצאת מודלים שונים לחישוב המודדים שלהם לבחירת מיטבית של המודל הטוב ביותר.

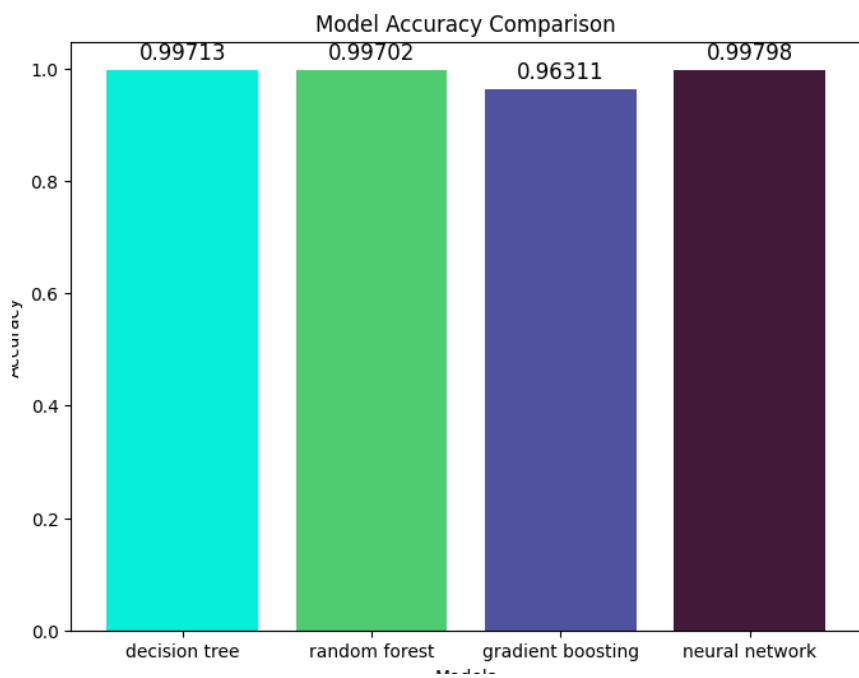
לאחר חישוב המודדים של המודלים שהרצנו קיבלו שהמודל המועדף ביותר עליו נקבע :

Model	AUC	CA	F1	Prec	Recall	MCC
Naive Bayes	0.993	0.951	0.954	0.960	0.951	0.908
Tree	0.982	0.974	0.974	0.974	0.974	0.951
Random Forest	1.000	0.992	0.992	0.992	0.992	0.985
kNN	0.992	0.975	0.975	0.976	0.975	0.953
AdaBoost	0.997	0.992	0.992	0.992	0.992	0.985

איור 24 : מגוון המודדים של המודלים

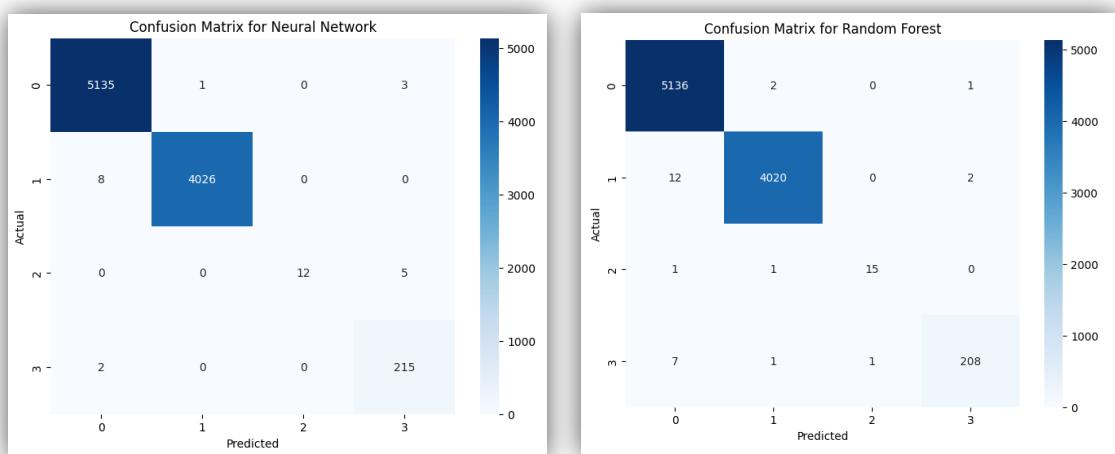


בנוסף מדי דיק של מודלים שונים :



איור 25 : השוואת דיק המודלים

מודל רשת הנוירונים עם כ-50 שכבות מושתרות ופונקציית הפעלה של `relu` מציג יכולות טובות .



איור 27 : מטריצת בלבול לרשת נוירונים

Random forest

מטריצות הבלבול הללו מציגות את יכולות שני המודלים , מש麻将 עצם החלטה ומימין מודל Random Forest .

מודל רשת הנוירונים מציג יכולות טובה עם חיזוי של 9389 תחזיות נכונות , לעומת זאת מודל Random forest שמציג 9379 תחזיות נכונות .



הסבר על המודלים השונים:

Random Forest – מבוסס על שילוב של מספר עצים החלטה, כאשר כל עצם מאמין על דגימה אקראית מהנתונים. מפחית overfitting ומשפר דיוק על ידי הצבעה משותפת של כל העצים.

Naïve Baye – מודל הסתברותי שבבסיסו על משפט בייס ומניח כי כל המשתנים בתכונות הם בלתי תלויים. מהיר ופשוט אך מניח הנחות שאuvwולות לא תמיד להתקיים.

AdaBoost – מודל המחזק מודלים חלשים (לרוב עצים החלטה קטנים) על ידי שינוי משקלים בדוגמאות הקשות לחיזוי. משפר דיוק אך רגיש לרעש ועריכים חריגים. לאחר בנייתו של מודלים "מוחלשים", כל המודלים שנבנו משוקלים בהתאם לביצועים שלהם, כך שלמודלים מדויקים יותר יש השפעה רבה יותר על התוצאות הסופיות. בדומה זו, משפר AdaBoost את הדיוק על ידי התקדמות בדוגמאות הקשות.

KNN – מסוג דגימה לפי הקטגוריה של ה-K השכנים הקרובים ביותר על בסיס מרחק. קל לישום אך דורש חישובים רבים שכן עלול להיות איטי.

Decision Tree – מודל היררכי שמחק את הנתונים לפי תנאים על התכונות עד להגעה לתוצאה סופית. קל להבנה ופרשנות, אך עלול לסבול מ-*overfitting* ללא הגבלת עומק.

GBM – מודל Boosting מתקדם שבונה סדרת של עצים כך שכל עצם חדש מתקן את הטיעות של העצים הקודמים. נותן ביצועים טובים אך דורש זמן חישוב ארוך ורגיש לפרטטים.

Neural Network – מודל מבוסס על שכבות של נוירונים המוחברים ביניהם. מתאים למשימות מורכבות כמו עיבוד. יכול להשיג תוצאות מעולות אך דורש חישובים גדולים ומשך אימון ארוך ביחס למודלים קודמים.

Gradient Boosting ו-**Random Forest** הם לרוב הבחירה הטובות ביותר, שכן הם יודעים להתמודד היטב עם נתונים כאלה ולספק חיזוי יציב ומדויק.

Naïve Bayes מתאים בעיקר לבחון המשתנים אכן בלתי תלויים, אך פחות מומלץ אם יש תלות ביניהם.

AdaBoost יכול לשפר את דיוק החיזוי על ידי מתן דגש לדוגמאות שקשوت לסייע, אך רגיש לרעש.

KNN פשוט לישום אך פחות יעיל כאשר יש משתנים רבים. **Decision Tree** קל לפרשנות אך עלול לסבול מ-*overfitting*. **Neural Networks** מתאימות יותר למשימות מורכבות כמו תמונה וטקסט.

הסבר ממדדים:

1 (AUC) Area Under the Curve

AUC מודד את השטח מתחת לעקומה ROC, ומיצג את יכולת המודל להבדיל בין קטגוריות שונות. ערך AUC הקרוב ל-1 מציין יכולת הבחנה מצוינת, בעוד שערך AUC הקרוב ל-0.5 מציין יכולת הבחנה מקרית, ורובה המודלים נחוצים לאי-מועילים במקרה זה.

2 (CA) Classification Accuracy

CA הוא אחוז התוצאות הנכונות מתוך כלל התוצאות שהמודל ביצע. זהו ממד פשוט, אך לא תמיד משקף את הביצועים האמתניים, במיוחד כשייחסו איזון בין הקטגוריות. לדוגמה, במקרים של



חומר איזון חריף, מdad זה יכול להטעות ולהציג דיווק גבוה למרות ביצועים גרועים במצבים מסוימים.

F1-Score .3

F1-Score הוא ממוצע הרמוני בין הדיווק (Precision) לריגושים (Recall). הוא משלב את שני המדרדים הללו במספר אחד. F1-Score שימושי במיוחד כמדד חשש איזון בין הקטגוריות, שכן הוא מודד האם המודל לא רק מדויק, אלא גם מזוהה היטב את כל הקטגוריות. ערך F1-Score גבוה מצין שילוב טוב של Recall ו-Precision.

Precision .4

Precision מודד את אחוז התחזיות החיוביות שהמודל סימן כחיוביות, שמאਮאות אמיטיות. אם המודל עושה הרבה טעויות מסווג חיוביים כזבאים (Precision), False Positives), Precision מdad זה חשוב במיוחד במקרים שבהם חיוביים כזבאים יכולים להיות מזיקים או לא רצויים, כמו במערכת רפואיות שבה המודל מסמן הרבה חולמים שלא חולמים.

Recall .5 (ריגישות)

Recall מודד את אחוז המקרים האמיטיים החיוביים ששומנו בצורה נכונה כחיוביים. אם המודל מפספס הרבה חיוביים אמיטיים (False Negatives), זה מdad קריטי במצבים שבהם חשוב לוזהות את כל המקרים החיוביים, אפילו במקרה של טעויות חיוביות כזבאים, כמו במערכות לזיהוי מחלות.

(MCC) Matthews Correlation Coefficient .6

MCC הוא מדד שספק תמונה מואצת של ביצועי המודל, גם במקרים של חומר איזון בין הקטגוריות. הוא לוקח בחשבון את כל ארבעת הערכים של מטrixת הבלבול: True Positives (TP), False Positives (FP), True Negatives (TN) ו-Fals Negatives (FN). ערכים קרובים ל-1 מעידים על מודל מצוין, ערכים קרובים ל-0 מעידים על מודל גרוע, וערכים שליליים מצביעים על ביצועים גרועים במיוחד.

מסקנות מדרדים:

AUC - ערך גבוה מציבע על כך שהמודל מסוגל להפריד היטב בין הקטגוריות, כלומר יש לו יכולת טוביה לוזהות מקרים חיוביים ושליליים ללא תלות בערך הסף שנקבע.

CA - ערך גבוה מציבע שהמודל מסוג אחוז גדול יותר מהדוגמאות נכון, כלומר מוצע כלל הדוגמאות שסוגו, רובן קיבלו את התוויות המתאימה.

F1-Score - מדד איזון בין דיווק (Precision) ושיליפה (Recall) כך שערך גבוה מעיד על כך שהמודל מצליח לוזהות נכון חיוביים אמיטיים תוך צמצום השגיאות בזיהוי חיובי כזב ושלילי כזב.

Precision - ערך גבוה מציבע על כך שכאשר המודל מסוג דוגמה חיובית, יש סבירות גבוהה יותר שהיא באמת חיובית, לעומת המודל מייצר פחות חיוביים כזבאים (False Positives).



- ערך גבוה מעיד על כך שהמודל מזזה את רוב המקרים החיוביים בפועל, כלומר הוא מחמץ פחות חיוביים אמיתיים - **Recall (False Negatives)**.

MCC - ערך גבוה מצין שהמודל שומר על איזון טוב בין ארבעת סוגי הסיווגים (True Positives, True Negatives, False Positives, False Negatives), ולכן הוא מדוודם להערכת איכות המודל, במיוחד אם הנתונים לא מאוזנים.

לסיכום,

בשלב זה של הפיזיוקט התמקדנו בבחירה המודל השופי, זאת לאחר שביצעו הינה מקיפה של הנתונים והרצת מספר מודלים שונים על הנתונים המעובדים. ההשוואה בין המודלים בוצעה על בסיס מדדים שונים, שנבחרו כדי להעריך את איכות החיזוי והביצועים של כל מודל. לאחר חישוב המדדים והשוואת תוצאות המודלים, נמצא כי **Random Forest** הוא המודל היעיל ביותר, שכן הוא השיג את התוצאות הטובות ביותר במדדים המרכזיים. מודול זה הציג דיוק גבוה, רגישות גבוהה, איזון טוב בין חיוביים ושליליים, והפחית את מספר השגיאות בזיהוי.

בסיכום, לאחר תהליך בחירה מדוקן ועמוק, החלפנו להתקדם במודל **Random Forest** או **רשת נוירוניים ו- Ada Boost**. כמודדים הסופיים לחיזוי הנתונים, בשל הביצועים האופטימליים שלו בכל המדדים שנבחנו ומהירות ההרצה של האלגוריתם שלו - אותן נבחנו לעומק בפרק הבא.

מסקנות -

המודלים המעודפים במקרה זה הם Random Forest ו-Gradient Boosting, שכן הם מציגים חיזוי יציב ומדויק במיוחד במיוחד עבור נתונים עם משתני דמי. בנוסף רשת נוירוניים עם relu וכ-50 שכבות מוסתרות מציג ממדדים די טובים.

מודלים אחרים כמו Naïve Bayes ו-KNN פחות מתאימים, בשל ההנחה הטטיטיסטיות המגבילות או הוצרך בעיבוד חישובי כבד.



ניתוח נתוניים (EDA):

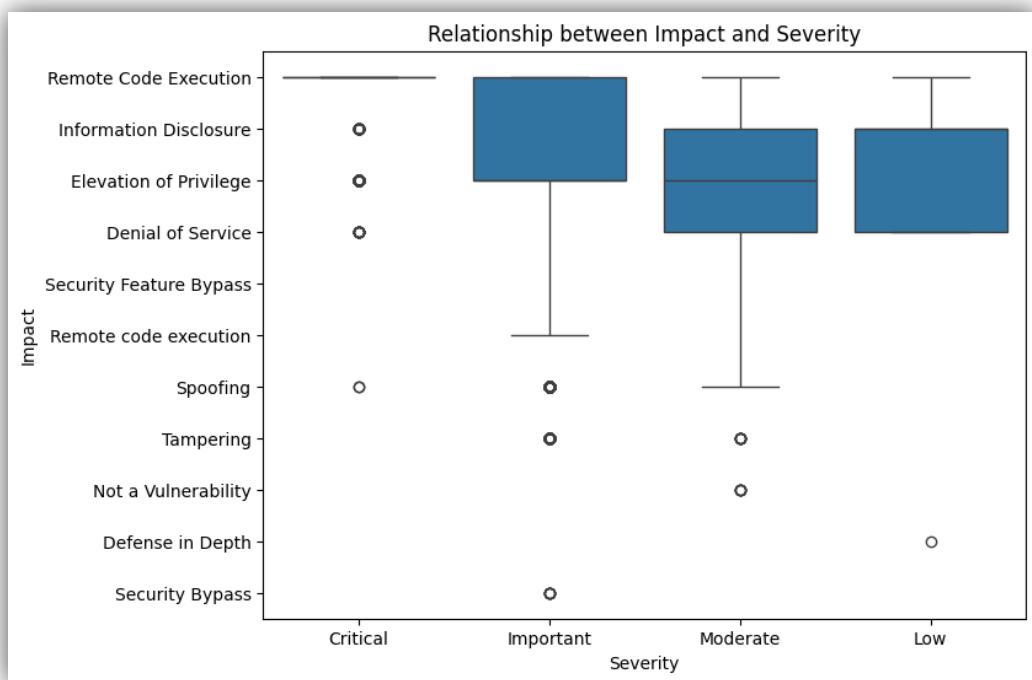
בסעיף זה, נבחן את קבוצת הנתונים הסופית באמצעות ניתוח נתונים. הניתוח ישיע לנו להבין את המאפיינים העיקריים של הנתונים ו寥חות קשרים פוטנציאליים בין התכונות למטרה. הנתונים הסופיים שנבחרו עבור ייעול המודל הם : Date Posted, Impact, Title, Severity.1, .Month , Component KB ,Affected component ,Supersedes, Reboot, CVEs משתנה המטרה של המודל הוא : Severity .
Severity :
. Ada Boost ,Random Forest - רשות נוירוניים ו -
אנו נתמקד במודלים של

	count	unique	top	freq
Impact	23516	11	Remote Code Execution	15696
Title	23516	869	Cumulative Security Update for Internet Explorer	3044
Component KB	23516.0	3577.0	954430.0	83.0
Severity.1	23516	4	Important	12268
Supersedes	23516	2096	MS15-023[3034344]	10561
Reboot	23516	3	Yes	12981
CVEs	23516	1349	CVE-2013-3128,CVE-2013-3200,CVE-2013-3879,CVE-2013-3880,CVE-2013-3881,CVE-2013-3888,CVE-2013-3894	350
Year	23516.0	10.0	2015.0	7843.0
Month	23516.0	12.0	10.0	6363.0
Day	23516.0	17.0	8.0	7612.0
Severity	23516	4	Critical	12783

איור 28 : טבלת מדדיות וערך מוסף

טבלאות תדירות:

לאחר בירהה קפנדית של הנתונים נציג ערכים על המשתנים שבחרנו למודל :
ניתן להבחין כי הערך השכיח ביותר במידה הפגיעה הראשונית והסופית שונות זו מזו.
גבוהה לטיפול מהיר בבעיות אלו .
Severity - רוב העדכנים מתייגים את הבעיה כ"חמורות" (Critical) , מה שמצויב על חשיבות צורך באתחול מחדש (Reboot) - רוב העדכנים דורשים אתחול מחדש , מה שמצויב על כך שהעדכנים משפיעים על המערכת באופן משמעותי .
Supersedes - ישנם 2,096 ערכים ייחודיים בעמודה, מה שמצויב על כך שישנם עדכנים רבים שמחליפים עדכנים קודמים .



איור 29 : דרכ Box Box מהתאר את ההשפעה ומידת הנזק

הסבר הגרף :

הגרף מציג את היחס בין הפגיעה (impact – ציר y) לחומרת הפגיעה (Severity – ציר x). שטח הקופסה של התרשים מבטא את התפלגות 0.5 הנזונים המרכזים ושטח קופסה קטן מעיד על שונות נמוכה, בעוד שhetto האנכי בתחום התפלגות 0.5 מהנתונים הוא חציון שמחולק את הנתונים באופן שווה. זוויות הקופסה מייצגות את מרבית הנתונים ללא השפעה על ערכיהם חריגים ומספקים ערך נוסף על התפלגות הנתונים. הערכים הקיצוניים מובאים לידי ביטוי על ידי הנקודות המופרדות מתחתי או מעל הדיאגרמה.

תרשים הקופסאות מספק כלי חזותי חזק להבנת התפלגות הנתונים, והזרועות והערכים הקיצוניים מספקים מידע חשוב על הוויריאציה והיווצראים מן הכלל בתנאים.

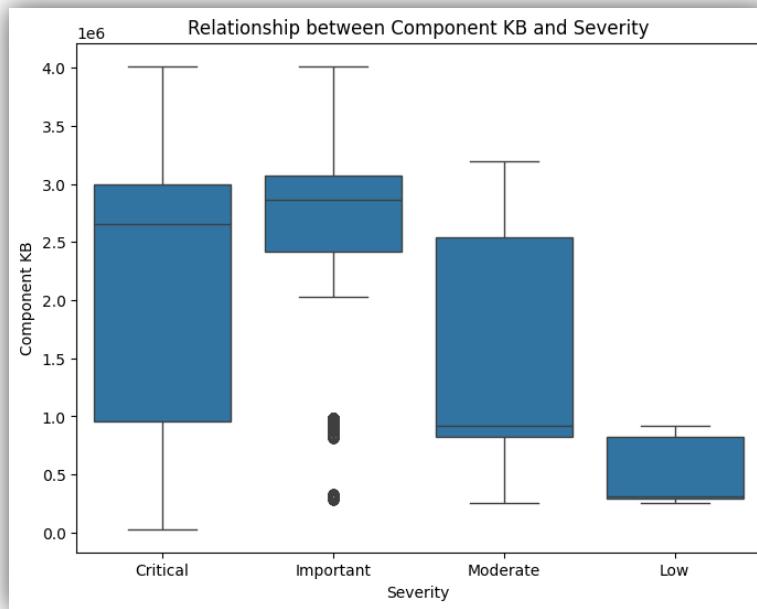
הקשר בין המשתנים :

הצגת הקשר בין השפעה לחומרה ע"י הגרף מראה את הקשרים בין סוגי הפגיעה השונות (כמו הרצת קוד מרוחקת, חשיפת מידע, הלאמת הרשות) לבין חומרת הפגיעה. כל סוג פגעה מוצג בשורה אחת, והkopfsאות מראות את התפלגות החומרה של כל סוג פגעה.

מסקנות עיקריות :

הגרף מציג את הקשר בין סוגי פגיאות אבטחה לחומרתן. פגיאות כמו הרצת קוד מרוחק מסווגות כחומרות ביוטר באופן עקבי, בעוד שפגיאות כמו Spoofing ו-Tampering מדורגות לרוב כחומרה נמוכה. שיטה הקופסאות הקטן מצביע על שונות נמוכה בתחום כל קטגוריה, כולל רוב הפגיעה מסווגות באופן יציב ללא חריגות משמעותיות.

תרשים זה מאפשר להבין איזה סוג פגיאות נחביבים יותר וכי怎ן מתפלגות בהקשר של חומרתן.



איור 30 : גרף Box מהתאר את הרכיב שהושפע למידת הנזק

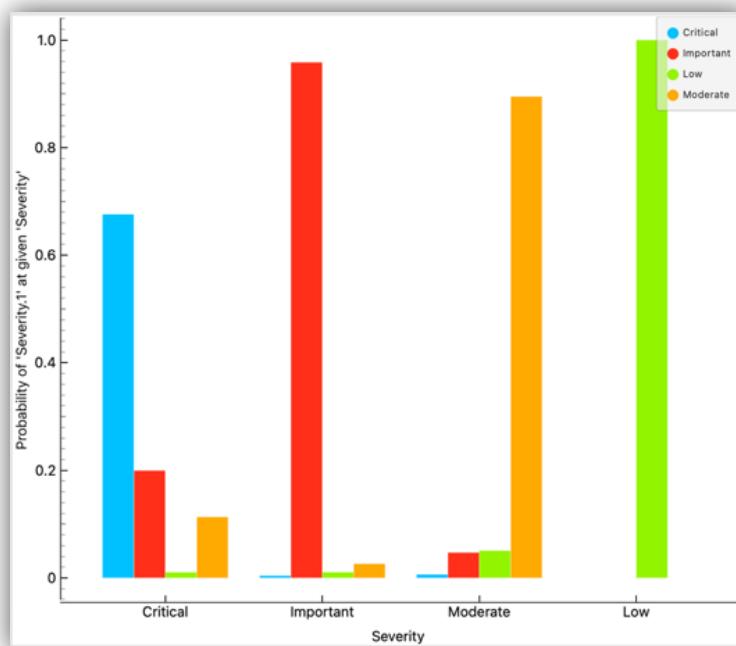
ניתן להסיק מספר מסקנות מתוך הגרף שמציג יחס בין מידע על רכיב המשפע ורמת הפגיעה :

חוمرة קרייטית (Critical) - התפלגות רחבה מאוד, עם טווח גדול של ערכים מהמיינימום למקסימום. החציון נמצא באמצע הטווח, מה שמצוין על התפלגות סימטרית יחסית. ישנו מספר ערכים מופרדים מתחת לקופסה, ניתן להסיק על עדכונים קטנים יותר מאוד בחומרה זו.

חוمرة חשובה (Important) - החציון נמצא באזור העליון של הקופסה, מצובע על נטייה לערכים גבוהים יותר. ישנו מספר ערכים מופרדים מתחת לקופסה, אך פחות מאשר בחומרה קרייטית.

חוمرة מתונה (Moderate) - החציון נמצא באמצע הטווח, מה שמצוין על התפלגות סימטרית יחסית. ישנו מספר ערכים מופרדים מתחת לקופסה.

חוمرة נמוכה (Low) - התפלגות צרה מאוד, והקופסה קטנה יחסית, מה שמצוין על עדכונים קטנים נמוך). החציון נמצא באמצע הטווח, והקופסה קטנה יחסית, מה שמצוין על עדכונים קטנים יותר באופן כללי.

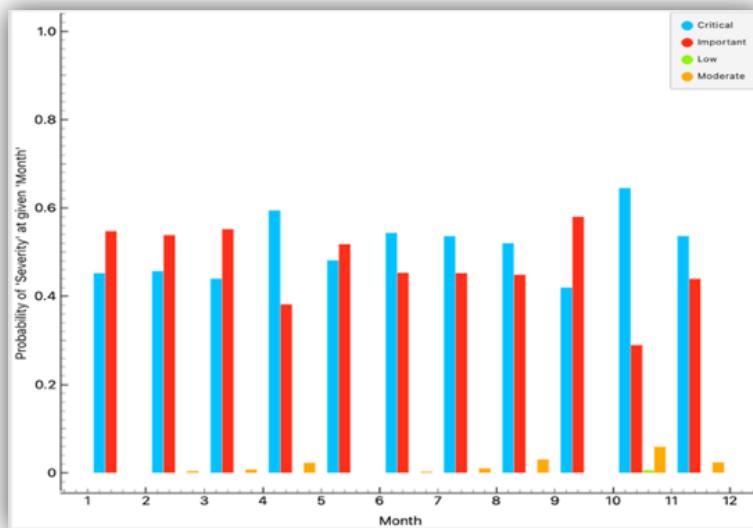


איור 31 : גראף המציג את יחס בין סוג החומרה הראשוני והסופי

גרף העמודות מציג ההסתברות של סוגי החומרה הראשוניים והתפלגותם על סוגי החומרה הסופיים הגרף מאפשר להבין את התפלגות החומרה של הפגיעה ואת ההסתברות של כל רמת חומרה. ציר י מיציג את מיקום הסתבות ראשונית של כל חומרה. ציר X מחולק לפגיעה ראשונית וסופית.

מסקנות :

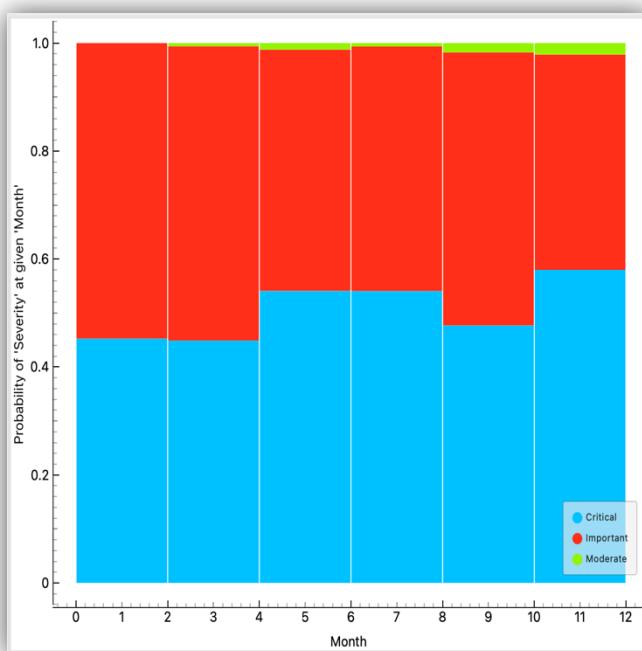
הגרף מראה שרמות החומרה important ו Moderate הן הנפוצות ביותר בין הפגיעה, כאשר לשתייהן יש הסתברות גבוהה יחסית. רמת החומרה Critical מופיעה פחות, אך עדין בהסתברות משמעותית. רמת החומרה low תהיה הנפוצה ביותר בקרב הפגיעה המדוחשת, כפי שניתן לראות מהעמודה הירוקהגובהה. הגרף מספק תמונה חזותית של התפלגות החומרה של הפגיעה, ויכול לשיער בקבלת החלטות בנוגע לאבטחה.



איור 32 : גרפּ המציג את מידת הנזק לפי חודשים

גרף עמודות מציג את ההסתברות של רמות חומרה שונות של פגיעות או אירועים לאורך תקופה של 12 חודשים. הגרף מאפשר לראות איך התפלגות החומרה של הפגיעות משתנה לאורך הזמן.

גרף העמודות המוערם נותן ערך מסוֹם ומציג את החלק מהשלום של חומרת הפגיעה לפי החודשים.



איור 33 : גרפּ מוערם של מידת הנזק על פני שנה

מסקנות :

ניתן להזות עצומות ותקיפה שונות לאורך זמן, דרך הגרף אפשר לבדוק ולהבין אם האסטרטגיות הנקויות יעילות או זכות לשינוי. בנוסף ניתן לארות לאורך כל החודשים ינסמ תקיפות ברמות גבוהות, חודשים קרייטיות או חשובות עשויים להצביע על נזודות ביקורת שבוחן יש צורך בהתרבות או בחיזוק אמצעי ההגנה.

מירב התקיפות הקרייטיות מתרכשות בחודשים 4, 6, 7, 10, 11, 12 ניתן להסיק לכך למחדל מירב ההגנה מפני התקיפות קשות בחודשים הללו יכול לסייע בתכנון משאבים.

יתר על כן דרך הגרף המוערם ניתן להסיק כי החודשים שצינו מתרחשים בהם מרבית התקיפות הקרייטיות. החברה מיקרוסופט מתמודדות עם מתפקיד בחומרות גבוהות לאורך כל השנה.

		#	ANOVA	χ^2
1	N Severity.1=Important		15837.631	5388.225
2	N Impact=Elevation of Privilege		3862.346	4297.248
3	N Impact=Remote Code Execution		13571.754	4044.518
4	N Severity.1=Moderate		2012.035	3038.741
5	N Component KB=3002885.0		1069.207	1920.510
6	N Component KB=2716513.0		974.808	1766.869
7	N Component KB=978338.0		836.357	1536.408
8	N Component KB=2539636.0		791.024	1459.588
9	N Component KB=2387149.0		791.024	1459.588
10	N Component KB=959426.0		791.024	1459.588
11	N Impact=Information Disclosure		807.387	1404.880

איור 34 : תצוגה של מבחנים סטטיסטיים על משתנה המטרה

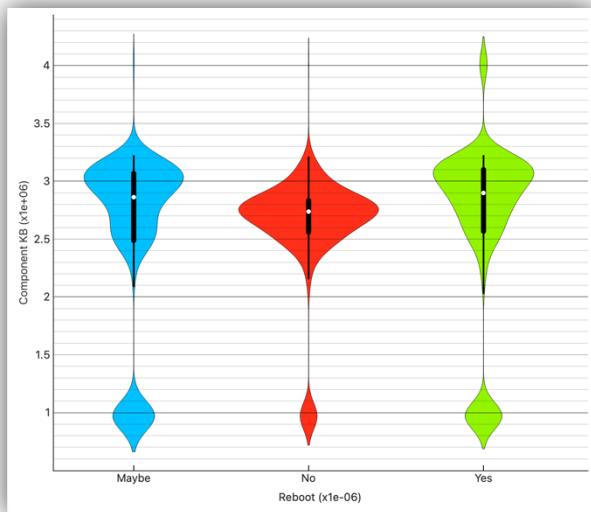
ANOVA: מייצגת ניתוח שונות.

זהו מבחן סטטיסטי המשמש להשואת הממצאים של שתי קבוצות או יותר. בדרך כלל, ערך גובה יותר של ANOVA מצביע על הבדל משמעותי יותר בין הקבוצות שנבדקו.

Chi-squared: מייצג סטטיסטיקה של Chi-squared מבחן משמש בדרך כלל כדי לבדוק אם יש קשר משמעותי בין שני משתנים קטגוריים. כמו ANOVA, ערך גובה יותר עשוי לرمז על קשר משמעותי יותר.

שורות: כל שורה מייצגת סוג מסוים של פגיעות או בעיה, יחד עם הערכים הסטטיסטיים שלה עבור ANOVA ו- χ^2 .

לסיום, הטבלה מספקת השוואת כמותית של קטgorיות באמצעות מבחנים סטטיסטיים (Chi-squared ANOVA). הערכים הללו יכולים לעזור בתייעוד ולהבין את המשמעות של בעיות שונות



גרף הכינור מציג את התפלגות רכיב KB בקטגוריות שונות של אתחול חדש: 'אולי', 'לא' ו'כן'.

הגרף מספק יותר פרטים על התפלגות הנתונים, במיוחד צפיפות ההסתברות של הנתונים בנקודות שונות.

ציר ה-א: ציר ה-א מייצג את הקטגוריה 'אתחול חדש', המהולכת לשולש קבוצות: 'אולי', 'לא' ו'כן'.

ציר ה-ע: ציר ה-ע מייצג את הערך של רכיב KB (בסיסם של 1e+06).

אוצרות הבינור: לכל קטגוריה של אתחול חדש, יש צורת כינור המייצגת את התפלגות רכיב KB עבור קטgorיה זו. הרוחב של צורת הcinor מציין את צפיפות הנתונים בערך זה, כאשר חלקים רחבים יותר מצביעים על סבירות גבוהה יותר של מזיאת נקודת נתונים בערך זה.

איור 35 : גרף כינור המציג את התפלגות הרכיב על פני טగוריות אתחול חדש

הcinor מציין את צפיפות הנתונים בערך זה, כאשר חלקים רחבים יותר מצביעים על סבירות גבוהה יותר של מזיאת נקודת נתונים בערך זה.

נקודות על צורת הcinor מייצגות מדדים סטטיסטיים שונים, הנקודה הלבנה בתוך כל צורת כינור מייצגת את החציון, סרגל האמצעי השחור מייצג את הטווח הבין-רביעוני, והוא מראה את התפשטות האמצעית של 50% מהנתונים עבור כל קטgorיה של Reboot.



מסקנות:

עבור Reboot הקטגוריה 'Maybe' יש התפלגות רחבה יותר לרכיב KB, מה שמצוין על מוגון גודל יותר של ערci KB בקרב דוגמאות אלה.

הចיוון של רכיב KB, שמיוצג על ידי הנקודות הלבנות, נראה דומה באופן ייחסי בכל שלוש הקטגוריות.

התפלגות שוונות מעידות על כך שההחלטה לאותל מחדש מחדש עשויה להיות קשורה בערכי רכיבי KB שונים, אך יש צורך בניתוח סטטיסטי נוסף כדי לקבוע אם הבדלים אלה משמעותיים.



הגרף הבא מייצא את המילים הנפוצות ביותר של הרכיבים שהושפטו במהלך התקיפות. בין הרכיבים נמצא את windows, מערכת הפעלה ואתרי אינטרנט שונים.

איור 36 : גרף ענן שマーה את השכיחות של הרכיבים שהושפעו