דוח ניתוח הונאות באשראי – פרויקט סוף קורס למידת מכונה

מבוא

הפרויקט שבחרנו לעשות עוסק בפיתוח מודלים מבוססי למידת מכונה לאיתור מקרי הונאה פיננסית.
במסגרת הפרויקט, נעשה שימוש במגוון אלגוריתמים לניבוי וקלסיפיקציה, במטרה לשפר את דיוק הגילוי של עסקאות הונאה מתוך מאגר נתונים אמיתי הכולל מעל 140,000 עסקאות. ניתוח הנתונים כלל שלבים של הכנה, ניקוי ועיבוד הנתונים, פיתוח מודלים שונים והשוואה ביניהם. נוסף על כך, פותח מודל כלכלי המעריך את החיסכון האפשרי כתוצאה משימוש במערכת זיהוי הונאות מבוססת למידת מכונה. הדו"ח מציג את תהליך העבודה, תוצאות המודלים המרכזיים והמסקנות הנגזרות מהן, תוך בחינת החזר ההשקעה הפוטנציאלי.

הדו"ח שכתבנו מציג את עיקרי הדברים. התמקדנו ברעיון שעמד מאחורי ההחלטות שלנו בהכנת הנתונים, בבחירת המודלים ובניתוח והבנה של התוצאות. מתוך כך למדנו על התנהגות הנתונים והמשמעות הנובעת מכך, וגם הבנה מעמיקה על סוגי המודלים שהרצנו וכיצד להתאים אותם בצורה הטובה ביותר.

<u>הכנת הנתונים לקראת הרצאת המודל</u>

1. המרת עמודת תאריך ושעה לפורמט POSIXct - עמודת transDateTransTime הומרה מפורמט טקסט לפורמט POSIXct כדי שנוכל לחלץ רכיבי זמן כמו שעה, יום, חודש ושנה לצורך ניתוחי זמן מדויקים.
2. המרת עמודות קטגוריאליות לפקטורים.
3. טיפול בערכי NA - זיהינו והחלפנו ערכי NA בעמודות מפתח. לדוגמה, ערכים חסרים בעמודת תאריך הלידה dob הוחלפו בתאריך הלידה החציוני (15 ביוני 1971), .תהליך זה כלל גם תיקון ערכים לא תקינים בעמודת המטרה - **is_fraud**.
4. מחיקת עמודות לא רלוונטיות – עמודת trans_num, שכללה מזהי עסקאות, הוסרה משום שלא תרמה לערך הניבוי.
5. החלפת תאריכי לידה חסרים בערך ממוצע - עמודת age חושבה מתוך עמודת dob, וערכים חסרים הוחלפו בגיל החציוני (52 שנים), זאת על מנת להימנע מהטיה בניתוח הנתונים.

EDA

1. קיבוץ קטגוריות נדירות - בעמודות כמו merchant, job, city - קטגוריות המייצגות פחות מ-0.5% מהנתונים אוחדו לקטגוריה בשם "Other" כדי למנוע דילול בנתונים במהלך האימון. לדוגמה, בעמודת merchant הקטגוריות הנדירות צומצמו מ-3,000+ לכ-50 ערכים בלבד.
2. הצגת התפלגות העסקאות באופן חזות - בחנו את התפלגות העסקאות לפי ערים, חודשים, שעות וימי השבוע. לדוגמה, השעות הנפוצות ביותר לעסקאות היו בין 10:00 ל-16:00. הוספת עמודות רלוונטיות:
6. הוספת משתנים - חישוב מרחק ויחסים : חישבנו את המרחק בין הלקוח לסוחר (distance_to_merchant) על בסיס קואורדינטות, ויצרנו משתנה חדש amt_to_distance_ratio , שמשווה בין סכום העסקה למרחק. – דבר זה משמש ככלי נוסף לניתוח עסקאות לא סבירות.
7. קיבוץ קבוצות גיל: חילקנו את הגילאים לקבוצות כמו "18-30", "31-45", "46-60" ו-"61+" כדי לנתח את שיעורי ההונאה בכל קבוצה.

איזון הנתונים

9. ניתוח התפלגות הונאות לפי קבוצות גיל ומדינות ,לפני ואחרי איזון נתונים. (ניתוח חזותי)
10. ביצוע Oversampling על עמודת המטרה:
איזון הנתונים כדי לשפר את ביצועי המודלים בניתוח תופעות נדירות.

ניתוחים נוספים

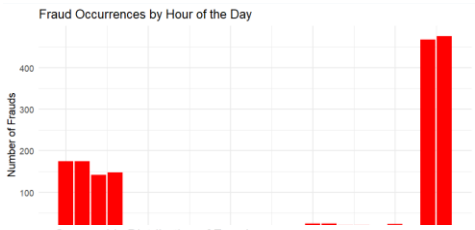
11. ביצוע Oversampling לאיזון המחלקות : מאחר שמקרי ההונאה היו במיעוט, ביצענו איזון על עמודת המטרה is_fraud. האיזון שיפר את הייצוג של המחלקה המועטה מ-2% לכ-50%.

12. ניתוח לאחר איזון -לאחר האיזון, ניתחנו מחדש את מגמות ההונאה לפי קטגוריות, וגילינו קשרים משמעותיים יותר במדינות ובסכומי העסקאות.

13. הגבלת ערכי סכום העסקאות לעסקאות עד 1,000$: עסקאות חריגות מעל סכום זה סוננו מתוך הנתונים כדי למנוע עיוותים בניתוח. לדוגמה, העסקאות בסכומים גבוהים במיוחד היוו פחות מ-1% מכלל העסקאות ונטו להיות בעלות שונות משמעותית מההתפלגות הכללית.

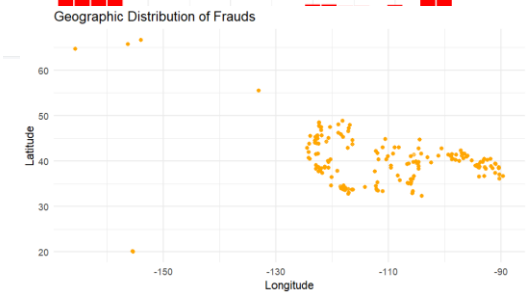
14. בחינת מתאם בין משתנים מספריים : ניתוח קשרים אפשריים בין פרמטרים מספריים שונים, כמו המתאם בין סכום העסקה למרחק מהסוחר, זוהה קשר חיובי קל (מתאם של כ-0.3), מה שעשוי לרמוז על עסקאות הונאה בסכומים גבוהים יותר כאשר המרחק גדול.

ניתוחים כלליים

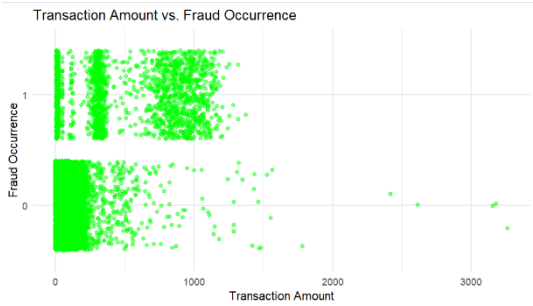


פילוח הונאות לפי שעות ביום:

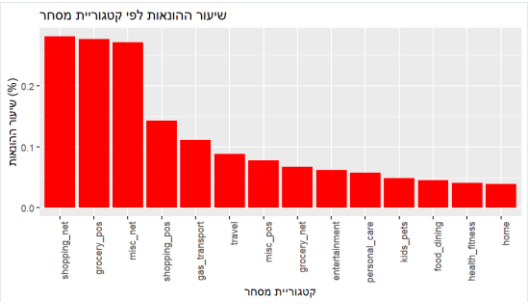
ישנם שני פיקים מובהקים במספר מקרי ההונאות - אחד קטן יותר מוקדם יותר בערב והשני, המרכזי והגדול ביותר, באותו הערב בשעה 23:00. שעה זו מציגה את המספר הגבוה ביותר של הונאות, שמגיע לכמעט 500 מקרים.



ניתוח הונאות לפי אזור גיאוגרפי:
רוב ההונאות מתרחשות באזורים מסוימים, כאשר ניתן לראות ריכוז גבוה יותר באזורים מסוימים כמו החלק המערבי והמרכז של הארצות הברית. זה יכול להצביע על כך שאולי ישנם מרכזים עירוניים גדולים באותם אזורים שבהם מתרחשת פעילות כלכלית גדולה יותר וכן פוטנציאל גבוה יותר להונאות.



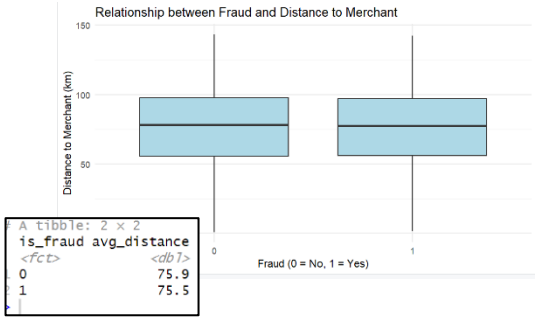
הקשר בין סכומי העסקאות לבין הונאות ההונאות אינן בהכרח מתמקדות בסכומי עסקאות גבוהים במיוחד, מה שמעיד שאסטרטגיות זיהוי הונאה צריכות להיות רגישות לטווח רחב של סכומי עסקאות.



אחוזי הונאות לפי קטגוריות מסחר:

קטגוריות כמו "shopping_net" ו "grocery_pos"מציגות את השיעורים הגבוהים ביותר של הונאות. זה יכול להצביע על כך שפעילות רבה ברשתות המקוונות ובקניות במקומות פיזיים בקטגוריה זו מושכת פעילות הונאתית.

הקשר בין המרחק מהסוחר לקיומה של הונאה :



גרף הקופסה מראה שהטווחים של המרחקים לעסקאות הונאה ולא הונאה כמעט זהים, והמידע המרוכז (הקו שבתוך התיבה) נמצא קרוב מאוד זה לזה בשני המקרים. אין פערים משמעותיים בטווחים או בתבנית המפוזרת שמצביעים על קשר מובהק בין המרחק לבין הונאה. ולכן המשתנה הזה כנראה לא יהיה אחד מהפיצ'רים המשפיעים ביותר על חיזוי.

הרצת המודלים

המודלים שבחרנו להריץ על הנתונים שלנו הם : Logistic Regression, XGBoost, SVM, AdaBoost ו-Decision Tree. המודלים שהתאימו בצורה המדויקת ביותר למאפיינים הספציפיים ולהתנהגות של הנתונים שלנו, שכוללים נתונים קטגוריאליים רבים ולא מאוזנים.

- ✓ Logistic Regression - מודל פשוט ופרשני, שמתאים במיוחד כאשר יש קשר ליניארי בין המשתנים. המודל יכול להתמודד עם תכונות קטגוריאליות לאחר המרה מתאימה.
- ✓ XGBoost ו-AdaBoost - נבחרו כיוון שיש להם יכולת להתמודד עם נתונים מורכבים ולא מאוזנים, על ידי בניית עצים רבים המשקללים תכונות קטגוריאליות שונות, וכך הם מתאימים במיוחד במקרים בהם ישנם דפוסי הונאה נסתרים בתוך מספר רב של קטגוריות.
- ✓ SVM - יעיל בזיהוי גבולות החלטה מורכבים ולא ליניאריים, מה שמועיל במיוחד כאשר הקשר בין התכונות הקטגוריאליות לבין משתנה המטרה הוא מסובך ולא פשוט להגדרה בצורה ליניארית.
- ✓ Decision Tree - התאים לנו כיוון שהוא פשוט וישלו את היכולת לייצר מודל שקל להבין ולהציג, למרות הפשטות היחסית שלו.

מודלים כמו Naive Bayes ו-KNN לא נבחרו כיוון שהם מתקשים להתמודד עם הנתונים הקטגוריאליים הרבים בנתונים שלנו. Naive Bayes מסתמך על הנחת אי-תלות בין התכונות, שלא מתקיימת כאן, במיוחד כאשר מדובר בתכונות קטגוריאליות מרובות הקשורות זו לזו. KNN רגיש מאוד לערכים חריגים ולריבוי משתנים קטגוריאליים, מה שמוביל לירידה בביצועים עם הנתונים הלא מאוזנים. Random Forest נחשב כאופציה, אם כי יש לנו באמת יותר מידי קטגוריות לכל משתנה, אך בסוף באמת העדפנו את XGBoost ו - AdaBoost, שיש להם יתרון עם היכולת שלהם להתמודד עם נתונים קטגוריאליים ועם חוסר איזון קיצוני בין מקרי ההונאה למקרים הרגילים. כמו כן, רגרסיה לינארית לא נבחרה מכיוון שהקשר בין התכונות הקטגוריאליות הרבות לבין משתנה המטרה אינו ליניארי, ולמודל זה יש מגבלות בהסקת תובנות מדויקות במצבים כאלה.

Logistic Regression

תוצאות המודל –

```
> cat("Accuracy: ", conf_matrix$overall['Accuracy'], "\n")
Accuracy:  0.9457651
> cat("Sensitivity (Recall): ", conf_matrix$byClass['Sensitivity'], "\n")
Sensitivity (Recall):  0.6455696
> cat("Specificity: ", conf_matrix$byClass['Specificity'], "\n")
Specificity:  0.9896825
> cat("Precision: ", conf_matrix$byClass['Pos Pred Value'], "\n")
Precision:  0.9015152
> cat("Negative Predictive Value: ", conf_matrix$byClass['Neg Pred Value'], "\n")
Negative Predictive Value:  0.9502159
> cat("F1 Score: ", conf_matrix$byClass['F1'], "\n")
F1 Score:  0.7523709
```

מסקנות -

המודל שהרצנו מציג דיוק גבוה של כ-94.6% ומצליח לזהות כ-99% מהמקרים שבהם לא מדובר בהונאה (ספציפיות). עם זאת, הרגישות הנמוכה (64.6%) מראה שהוא מפספס חלק משמעותי ממקרי ההונאה. מצב זה תואם את הציפיות, שכן המודל מתמודד עם נתונים לא מאוזנים הכוללים משתנים קטגוריאליים רבים. מדד ה-F1 המשולב, שעומד על 75.2%, משקף את האיזון בין דיוק בזיהוי מקרים חיוביים (Precision) לבין היכולת לזהות את כל המקרים החיוביים (Recall). כלומר, המודל אמין בזיהוי מקרים רגילים, אך יש מקום לשיפור בזיהוי כל מקרי ההונאה.

XGBoost

תוצאות המודל –

```
> print(confusion_matrix)
          Actual
Predicted    0    1
        0 3779    5
        1    1  548
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.998615278098315"
> print(paste("Precision:", precision))
[1] "Precision: 0.998178506375228"
> print(paste("Recall:", recall))
[1] "Recall: 0.990958408679928"
> print(paste("F1 Score:", f1_score))
[1] "F1 Score: 0.994555353901996"
>
```

מסקנות –

מודל ה-XGBoost מציג תוצאות מצוינות עם דיוק גבוה מאוד של 99.86%, כלומר כמעט ואינו טועה בזיהוי הונאות או מקרים שאינם הונאה. המודל מזהה מקרים כהונאה בדיוק של 99.88%, מה שאומר שכמעט כל התחזיות שהוא מסמן כהונאה הן נכונות. בנוסף, הוא מזהה כ-99.09% ממקרי ההונאה בפועל, מה שמראה שהוא תופס את רוב המקרים המסוכנים. ואכן, גם מדד ה-F1, שעומד על 99.45%, משקף את האיזון הטוב בין היכולת של המודל לזהות מקרי הונאה לבין המניעה של זיהוי שגוי של מקרים כאלה.

הצלחת מודל ה-XGBoost נובעת מהיכולת שלו להתמודד היטב עם נתונים מורכבים כמו בפרויקט שלנו, הכוללים הרבה משתנים קטגוריאליים ונתונים לא מאוזנים. המודל מצליח לזהות קשרים מורכבים בין תכונות ולהימנע משגיאות בזיהוי הונאות בזכות שיפורים שמתרחשים במהלך האימון. בנוסף, הוא מתאים במיוחד בגלל היכולת שלו להפחית טעויות ולהימנע מאוברטפיטינג, וזו למעשה הסיבה שאנחנו משערים שהמודל הביא לדיוק גבוה בתוצאות.

Support Vector Machine (SVM)

תוצאות המודל –

```
> print(confusion_matrix)
          Actual
Predicted    0    1
        0 3761   66
        1   19  487
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.980383106392799"
> print(paste("Precision:", precision))
[1] "Precision: 0.962450592885375"
> print(paste("Recall:", recall))
[1] "Recall: 0.880650994575045"
> print(paste("F1 Score:", f1_score))
[1] "F1 Score: 0.919735599622285"
```

מסקנות –

תוצאות מודל ה-SVM מציגות דיוק כללי של כ-98%, עם יכולת זיהוי של כ-96% מהמקרים שבהם ניבוי המודל היה שמדובר בהונאה (Precision). יחד עם זאת, המודל מצליח לזהות כ-88% ממקרי ההונאה בפועל (Recall), מה שמצביע על כך שישנם מקרים מסוימים של הונאה שהמודל מפספס. המדד המשולב (F1) עומד על כ-91.9%, ומראה על איזון טוב בין היכולת לזהות הונאות לבין הדיוק בניבוי שלהן.

ההצלחה של המודל קשורה ליכולת שלו להתמודד עם גבולות החלטה מורכבים ולא ליניאריים, מה שמותאם היטב לנתונים הקטגוריאליים והלא מאוזנים שהשתמשנו בהם. עם זאת, למרות הביצועים הטובים, ייתכן שיש עדיין מקום לשיפור באזורים שבהם הקשרים בין המאפיינים לא פשוטים, מה שיכול להוביל להחמצת חלק ממקרי ההונאה.

AdaBoost

תוצאות המודל –

```
> print(confusion_matrix)
          Actual
Predicted    0    1
        0 3778    6
        1    2  547
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.998153704131087"
> print(paste("Precision:", precision))
[1] "Precision: 0.996357012750455"
> print(paste("Recall:", recall))
[1] "Recall: 0.989150090415913"
> print(paste("F1 Score:", f1_score))
[1] "F1 Score: 0.992740471869329"
```

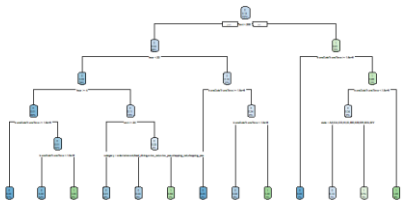
מסקנות –

מודל ה-AdaBoost מציג תוצאות מרשימות עם דיוק כללי של כ-99.8%. הוא מצליח לזהות כ-99.6% מהמקרים שבהם הוא מנבא הונאה (Precision) וכ-98.9% ממקרי ההונאה בפועל (Recall). המדד המשולב (F1) עומד על כ-99.3%, מה שמעיד על ביצועים מאוזנים במיוחד. להשערתנו, ההצלחה של המודל נובעת מהיכולת שלו לחזק את הביצועים שלו על ידי התמקדות במקרים שהמודלים הקודמים פספסו, מה שמתאים במיוחד לנתונים לא מאוזנים כמו אלו בפרויקט שלנו.

Decision Tree

תוצאות המודל

–



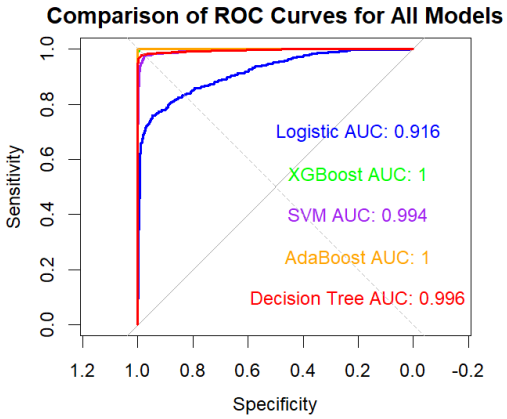
```
> print(confusion_matrix)
         Actual
Predicted    0    1
        0 3772   20
        1    8  533
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.993537964458804"
> print(paste("Precision:", precision))
[1] "Precision: 0.985212569316081"
> print(paste("Recall:", recall))
[1] "Recall: 0.963833634719711"
> print(paste("F1 Score:", f1_score))
[1] "F1 Score: 0.974405850091408"
>
```

מסקנות –

מודל ה Decision Tree-שהרצנו השיג דיוק גבוה של כ-99.3%. המודל הצליח לזהות כ-98.5% מהמקרים שבהם הוא מנבא הונאה ,(Precision) ורגישות של כ-96.3% בזיהוי מקרי ההונאה בפועל .(Recall) המדד המשולב (F1) עומד על כ-97.4 . הביצועים הטובים של המודל הם כיוון שיש למודל הזה יכולת לפצל את הנתונים ולהתאים קריטריונים שונים, מה שעוזר במיוחד בהתמודדות עם תכונות קטגוריאליות מורכבות כמו שלנו.

הדיאגרמה מציגה את עץ ההחלטה שנבנה, אשר מזהה הונאות ומציג אותם בצורה וויזואלית. כל צומת מייצגת החלטה על סמך תכונה מהנתונים, והעלים מסמלים את התחזיות הסופיות (הונאה או לא).

השוואה בין המודלים



בהשוואת עקומות ה-ROC לכל המודלים שבחנו, ניתן לראות שכל המודלים מציגים ביצועים מרשימים עם ערכי AUC גבוהים מאוד. מודלים כמו XGBoost ו-AdaBoost מגיעים לערך AUC מושלם של 1.0, מה שמצביע על יכולתם להבחין בצורה מדויקת בין מקרי הונאה למקרים רגילים. גם מודל ה-Decision Tree מציג ביצועים מצוינים עם AUC של 0.996, ואחריו ה-SVM עם AUC של 0.994. לעומת זאת, מודל הרגרסיה הלוגיסטית מציג את הערך הנמוך ביותר מבין המודלים שנבחנו, אך עדיין מכובד, עם AUC של 0.916. השוואה זו מדגישה את היתרון של מודלים מתקדמים כמו XGBoost ו-AdaBoost כאשר מדובר בזיהוי מקרים מורכבים כמו הונאות פיננסיות.

```
> print(results)
                      Model  Accuracy Precision    Recall  F1_Score       AUC
Accuracy Logistic Regression 0.9457651 0.9015152 0.6455696 0.7523709 0.9162677
1                   XGBoost 0.9935380 0.9852126 0.9638336 0.9744059 0.9999550
11                      SVM 0.9935380 0.9852126 0.9638336 0.9744059 0.9942569
12                 AdaBoost 0.9935380 0.9852126 0.9638336 0.9744059 0.9999914
13            Decision Tree 0.9935380 0.9852126 0.9638336 0.9744059 0.9957045
```

בטבלה מוצגות כלל התוצאות של חמשת המודלים שנבחנו. ניתן לראות בבירור שמודלים כמו XGBoost, SVM, AdaBoost, ו-Decision Tree מציגים תוצאות כמעט זהות עם דיוק (Accuracy) גבוה של כ-99.35%, ו-F1 Score של כ-97.44%, המצביעים על איזון טוב בין זיהוי מקרי הונאה לזיהוי מקרים רגילים. כל המודלים האלו בעצם מציגים גם Precision ורמת Recall כמעט זהות, מה שמעיד על ביצועים מצוינים.

לעומתם, בהתאמה למה שכבר ראינו, מודל הרגרסיה הלוגיסטית מציג דיוק נמוך יותר, עם ערך Recall של 0.645 ו-F1 Score של 0.752, מה שמראה שהוא פחות מתאים לזיהוי מקרים מורכבים בהשוואה למודלים האחרים.

מודל כלכלי

המודל הכלכלי שבחרנו בודק את הערך המוסף הכלכלי של מערכת לזיהוי הונאות פיננסיות, ומתמקד בהערכת ההשפעה הכלכלית של זיהוי מוקדם של הונאות פיננסיות על חיסכון כספי לעסק.

המודל נועד להעריך את התרומה הכלכלית הפוטנציאלית של המערכת בכך שהוא מחשב את החיסכון הכספי שנוצר בעקבות מניעת הונאות. בנוסף, הוא משקלל את עלות התפעול של המערכת, כמו עלויות שרתים, תוכנות וכוח אדם.

המודל חשוב לעסק כיוון שהוא מספק תמונה ברורה של החזר ההשקעה (ROI) בשימוש במערכת זיהוי הונאות. העסק יכול להחליט אם ההשקעה במערכת מצדיקה את העלות בהתבסס על חישוב החיסכון הפוטנציאלי והעלות השנתית של התפעול.

באמצעות מודל זה ניתן להבין לא רק את היעילות של המערכת אלא גם את התרומה הישירה שלה לשורת הרווח של העסק.

בבניית המודל התבססנו על כמה הנחות מפתח:

- ✓ ממוצע הנזק הפיננסי לכל מקרה הונאה – הגדרנו שהנזק הממוצע עומד על 1,000 דולר.
- ✓ שיעור זיהוי ההונאות על ידי המערכת (רגישות) – הנחנו שהמערכת מזהה כ-85% ממקרי ההונאה.
- ✓ מספר העסקאות השנתי – לפי הנתונים, בעסק מבוצעות כ-144,447 עסקאות בשנה.
- ✓ שיעור ההונאות המשוער מסך כל העסקאות – הנחנו ש-2% מהעסקאות הן הונאות.

לאחר חישוב מספר המקרים הפוטנציאליים של הונאות בשנה ושיעור המקרים שהמערכת מזהה בפועל, חישבנו את הסכום הכספי שניתן לחסוך על ידי גילוי ההונאות. לאחר מכן, הפחתנו את עלות ההפעלה של המערכת מהחיסכון הכספי הזה. התוצאה הסופית היא החיסכון הכספי הנקי לעסק.

תוצאות -

התוצאות מראות כי בעסקה יש כ-2,889 מקרים פוטנציאליים של הונאה בשנה, ומתוכם המודל מזהה כ-2,456 מקרים. בהתבסס על הנחת נזק ממוצע של 1,000 דולר למקרה הונאה, ניתן להעריך חיסכון פוטנציאלי של כ-2.45 מיליון דולר בשנה בזכות המערכת. כאשר מביאים בחשבון את עלות התפעול השנתית של המערכת, שנאמדת בכ-20,000 דולר, החיסכון הנקי לעסק הוא כ-2.43 מיליון דולר. נתונים אלו מראים שהמודל תורם בצורה משמעותית להקטנת ההפסדים הכספיים הקשורים להונאות פיננסיות.

סיכום

בפרויקט זה פיתחנו מערכת לזיהוי הונאות פיננסיות באמצעות מספר מודלים של למידת מכונה, תוך התמקדות בנתונים לא מאוזנים וקטגוריאליים. לאחר תהליך מקיף של הכנת הנתונים, שכלל המרת עמודות, טיפול בנתונים חסרים, יצירת משתנים חדשים וביצוע איזון נתונים, בחנו מספר מודלים, בהם: Logistic Regression, XGBoost, SVM, AdaBoost ו-Decision Tree.

מבחינת ביצועים, המודלים המתקדמים כמו XGBoost ו-AdaBoost הובילו עם מדדי דיוק ורגישות גבוהים מאוד, והצליחו לזהות כמעט את כל מקרי ההונאה עם דיוק כולל של מעל 99%. מנגד, מודל הרגרסיה הלוגיסטית, למרות הדיוק הכללי שלו, התקשה בזיהוי מקרי הונאה (רגישות נמוכה), בעיקר בשל הנתונים הלא מאוזנים והמורכבות הקטגוריאלית.

בנוסף, יצרנו מודל כלכלי להערכת ההשפעה של יישום המערכת. באמצעות הנחות בסיסיות לגבי שיעור ההונאות וממוצע הנזק הכספי, חישבנו את החיסכון הפוטנציאלי לעסק – המסתכם ביותר מ-2.4 מיליון דולר בשנה, גם לאחר שקלול העלויות התפעוליות של המערכת.

לסיכום, המודלים המתקדמים הראו יכולת גבוהה בזיהוי הונאות, והמודל הכלכלי מצביע על ערך כלכלי משמעותי בשימוש במערכת לזיהוי מוקדם של הונאות. הממצאים מהפרויקט הזה מדגישים את החשיבות בשילוב כלים טכנולוגיים מתקדמים עם ניתוח כלכלי לקבלת החלטות מושכלות בעסק.

נוסיף גם, כי היו לנו לא מעט אתגרים עם הכתיבה של הקוד, אך ההתמודדות איתם לימדה אותנו וחידדה לנו את החומר שלמדנו בקורס וסיפקה להו הבנה מעמיקה על העולם של למידת מכונה.