דוח מסכם מבוא לבינה מלאכותית

מגישים: אמילי בדרוב - 215025255, שירה לביא - 216006395, עומר שדמי – 327722575

חלק ראשון:

השתמשנו בדאטה סט של זיהוי הונאות בכרטיסי אשראי.

קישור לדאטה סט: https://www.kaggle.com/datasets/fraud-detection

הדאטה מורכב מ 3075 דגימות, 11 פיצ'רים ומשתנה מטרה אחד – הונאה/לא הונאה.

כ-15% מהדגימות שלנו מסווגות כהונאה.

פירוט המשתנים (הפיצ'רים) בדאטה סט שלנו:

1. **Merchant_id** – ת"ז הסוחר.
2. **Transaction date** – תאריך העסקה.
3. **Average Amount/transaction/day** – סכום ממוצע לעסקה פר יום.
4. **Transaction_amount** – מספר עסקאות.
5. **Is declined**– האם העסקה נדחתה.
6. **Total Number of declines/day** – מספר דחיות פר יום.
7. **isForeignTransaction** – האם העסקה ממקור זר.
8. **isHighRiskCountry** – האם העסקה ממדינה "מסוכנת".
9. **Daily_chargeback_avg_amt** – סכום ממוצע להחזר יומי.
10. **6_month_avg_chbk_amt** – סכום ממוצע להחזר חצי-שנתי.
11. **6-month_chbk_freq** – תדירות החזרות חצי-שנתית.

ומשתנה המטרה שלנו: **isFradulent**– האם העסקה אכן הונאה.

לאחר ניתוח ראשוני של הדאטה, מצאנו כי שני המשתנים הראשונים אינם רלוונטיים לצורך החיזוי שלנו. **Merchant_id** אינו רלוונטי היות ומייצג אך ורק ת"ז של הסוחר. ו-**Transaction date** הוסר מהדאטה סט שלנו היות ויש בו יותר מדי ערכים חסרים, ולכן לא נוכל להסיק ממנו מידע.

סך הכל, המשתנים שהשתמשנו בהם לצורך חיזוי הקלאסיפיקציה (הונאה/לא הונאה) הינם:

- **Average Amount/transaction/day**
- **Transaction_amount**
- **Is declined**
- **Total Number of declines/day**
- **isForeignTransaction**
- **isHighRiskCountry**,
- **Daily_chargeback_avg_amt**
- **6_month_avg_chbk_amt**
- **6-month_chbk_freq**

חלק שני:

לצורך זיהוי הונאות, שיטת הפעולה שלנו הייתה למצוא outliers, כלומר ערכים חריגים בדאטה סט, אלו עלולים להעיד לנו על הונאות בכרטיסי האשראי.

השתמשנו במספר שיטות של קלאסיפיקציה, מרביתן לא מפוקחות. היות ואנחנו מעוניינים לחזות הונאות בכרטיסי האשראי וברגע שנשתמש במודלים מפוקחים – המודל יזהה את סוגי ההונאה הספציפיים המצויים בדאטה סט, והנוכלים יוכלו לשנות את דרך הפעולה שלהם על מנת להתחמק מזיהוי ההונאה.

המודלים הלא-מפוקחים בהם השתמשנו היו:

- **Isolation Forest**

זהו אלגוריתם לזיהוי אנומליות (ערכים חריגים), מודל שמשתמש בעצי החלטה. יעילותו נובעת מיכולתו לזהות ערכים חריגים מהר תוך כדי שימוש במספר קטן של עצים וחלוקות.
האלגוריתם בוחר ברנדומליות פיצ׳ר מהדאטא, ברנדומליות בוחר ערך פיצול בין הערך המינימלי והמקסימלי בפיצ׳ר. דרך זו יוצרת מסלולים קצרים בעצים לנקודות האנומליה, וכך ״מפרידות״ אותם משאר הדאטא. כך בדרך זו האלגוריתם ממשיך באופן רקורסיבי ויוצר חלוקות של הדאטא לפי כך שלאנומליות המסלולים קצרים יותר(הדרך מהנקודה אל השורש נקראת מסלול) וכך מפרידה את

Isolation of a normal point

Isolation of an anomaly

האנומליות מהערכים הנורמלים הרגילים.

- **Local Outlier Factor**

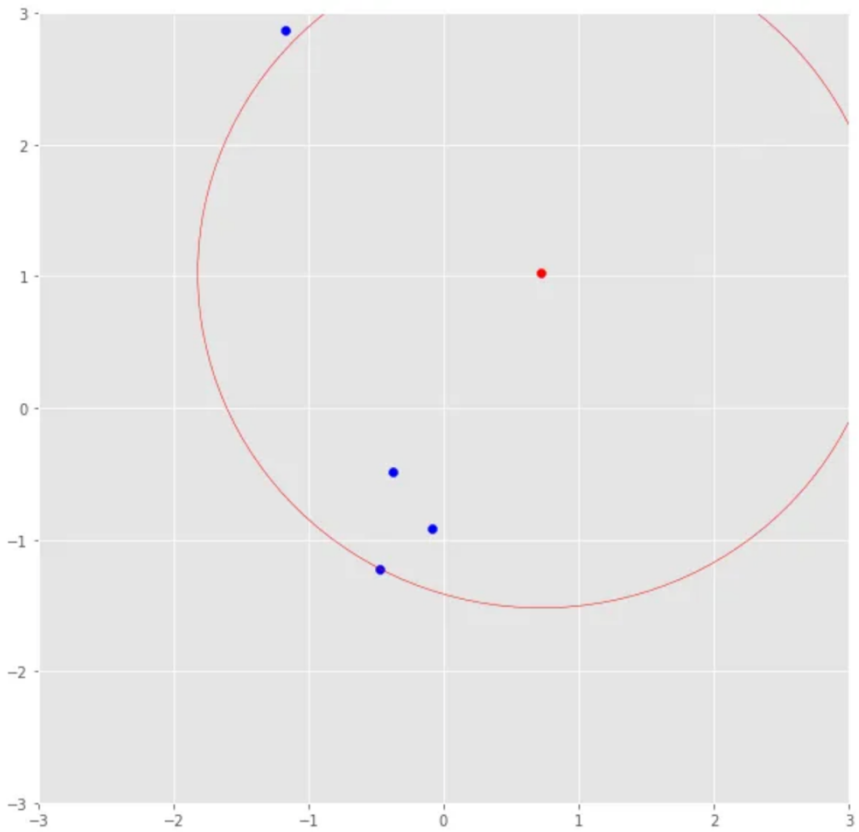
אלגוריתם זה משתמש בערך LOF שהוא מייצג עבורנו "מה הסיכוי שנקודה מסוימת היא אנומליה"
שני פרמטרים חשובים באלגוריתם זה הם:

1. כמות השכנים
2. Contamination(הידבקות)

האלגוריתם מחשב את המרחק מערך השכנים של נקודה(אם מספר השכנים הוא שלושה, למשל, הוא מחשב את המרחק בין הנקודה הנוכחית לנקודה השלישית הקרובה ביותר).

ואז מחשב את צפיפות הנקודות במרחק שמצא קודם לכן. צפיפות כל נקודה תושווה עם צפיפות k הנקודות השכנות. בעצם בעבור נקודת דאטא α היחס הממוצע בין הצפיפות של השכנים של α לצפיפות α הוא ה LOF. אם היחס הנ"ל גדול מ-1 הצפיפות של הנקודה α בממוצע קטנה מצפיפות שכניה, ולכן מנקודה זו עלינו לעבור מרחקים ארוכים יותר על מנת להגיע לנקודה הבאה. חשוב לזכור שהשכנים של נקודה α עשויים שלא להתייחס לשכן מכיוון שיש להם נקודות יותר קרובות.

Novelty Detection - The model is fitted with clear data (no outliers) and then, when the predict function is called with another dataset, the model can predict if each point is an outlier or not



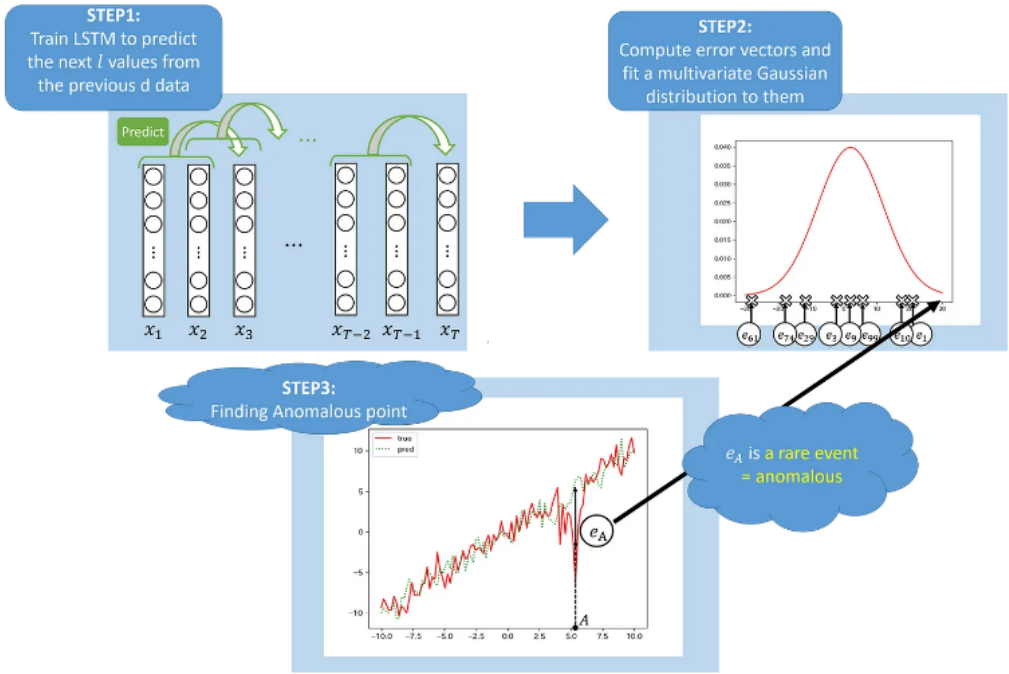
- **Gaussian Mixture**

מודל הסתברותי שמניח כי ההתפלגויות של כל הנקודות בדאטאסט הוא סכום ממושקל של התלפגויות נורמליות עם פרמטרים לא ידועים.
המודל מנסה למצוא תתי אוכלוסיות שונות בדאטא מבלי לדעת לאן כל נקודה שייכת כאשר המטרה היא לנבא לאן כל נקודה בדאטא שייכת.
הטכניקה הנפוצה ביותר לשימוש במודל זה הוא אלגוריתם EM אשר למדנו.

- **Autoencoder**

(Neural Networks with LSTM autoencoders)

סוגים אלה של רשתות מצטיינים במציאת קשרים מורכבים בנתוני סדרות זמן, רבות משתנים.
רשתות זיכרון לטווח קצר (LSTM) הן תת-סוג של הרשתות העצביות החוזרות הכלליות יותר (RNN).
תכונה מרכזית של רשתות עצביות חוזרות היא היכולת שלהן להתמיד במידע, או מצב התא, לשימוש בו מאוחר יותר. זה הופך אותם למתאימים במיוחד לניתוח נתונים זמניים המתפתחים עם הזמן. רשתות LSTM משמשות במשימות כמו זיהוי דיבור, תרגום טקסט ובמקרה שלנו, בניתוח של זרמי נתונים עוקבים לזיהוי אנומליות.
גישה זו טובה יותר כאשר האנומליות הם מקרים חריגים ולא רבים לעומת בעיות סיווג שבהן נרצה כמה שיותר מקרים שתויגו גם כחריגים וגם כנורמלים.
המקודד האוטומטי מאומן לשחזר את סדרת הזמן הרגילה וההנחה היא שמודל כזה ישחזר לא טוב את סדרות הזמן החריגות לאחר שלא ראה אותן במהלך האימון. המערכת בוחנת את הערכים הקודמים לאורך שעות או ימים וחוזה את ההתנהגות לדקה הבאה. אם הערך בפועל דקה לאחר מכן נמצא בסטיית תקן, אז אין בעיה. אם זה יותר, נחשיב אז נקודה זו כנקודת אנומליה.

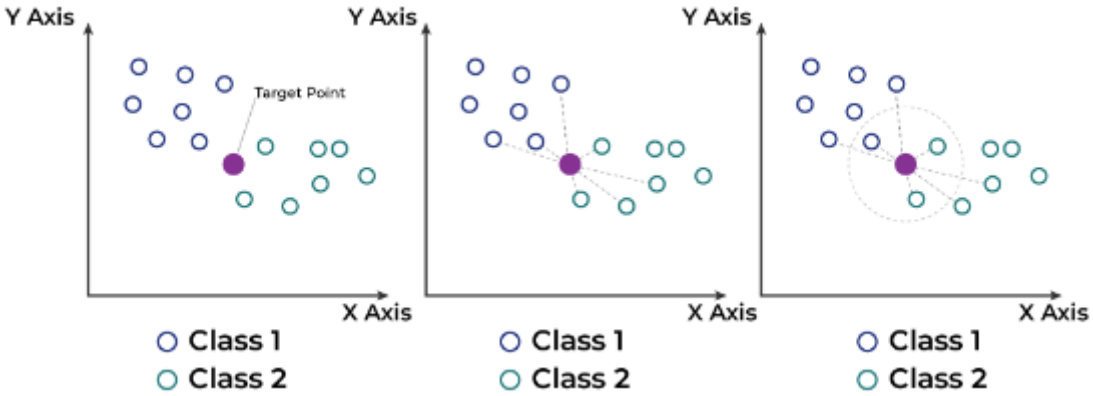


בנוסף, לצורך השוואה הבאנו יישום של מודל קלאסיפיקציה מפוקח, על מנת להמחיש את בעיית ה-over fitting, בניגוד למודלים הלא-מפוקחים שהשתמשנו בהם קודם.

- **KNN**

(k-nearest neighbor algorithm)

האלגוריתם פועל על עיקרון הדמיון, שבו הוא מנבא את התווית או הערך של נקודה חדשה על ידי התחשבות בתוויות או הערכים של K שכנותיה הקרובות ביותר בדאטה של האימון.



חלק שלישי:

חילקנו את הדאטה סט ל-80% training ו-20% test.

לאחר מכן, חילקנו את הtraining set בעצמו ל80% training ו 20% validation .

לכל מודל נבדוק את המטריקות הבאות:

- Accuracy
- Precision
- Recall
- F1-Score
- Support

לאחר החלוקה ל Train ו- Test, נחלק את ה- Train לפי:

X = כל הדאטא מלבד עמודת הניבוי(האם בוצעה הונאה)

y = האם בוצעה הונאה

במקרה זה, חשוב לנו מאוד שלא תהיה הונאה ואנו מבינים שזה יכול לבוא על חשבון זה שאנו נזהה גם דברים שהם לא הונאה כחשודים להונאה (שהרי בתור בנק אם אכן תהיה הונאה הבנק יהיה אחראי לה והנזק של להתריע לבן אדם שיש לו פעילות חשודה בחשבון הוא לא מאוד גדול, אך הכל במידה כמובן).

בגלל שאנו לא רוצים לפספס הונאות, הריקול (recall) של החיזוי 1(fraud) יהיה מרכיב משמעותי בהחלטה שלנו האם מודל הוא טוב לנו או לא. שכן הריקול מוגדר כך:

$$\text{Recall} = \frac{\text{Relevant retrieved instances}}{\text{All relevant instances}}$$

הprecision גם הוא משמעותי כמובן, שכן, ניתן לחזות שכל הנתונים שלנו הם הונאות ולקבל ריקול 1 לחיזוי של הונאה, אך ל precision ניתן משקל פחות כבד. שהרי הprecision מוגדר כך:

$$\text{Precision} = \frac{\text{Relevant retrieved instances}}{\text{All retrieved instances}}$$

ה f-1 score שלנו הוא בעצם שכלול של ה recall וה precision, ואליו נתייחס פחות כי הוא נותן להם משקל שווה ואצלינו לריקול יש משקל כבד יותר.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}$$

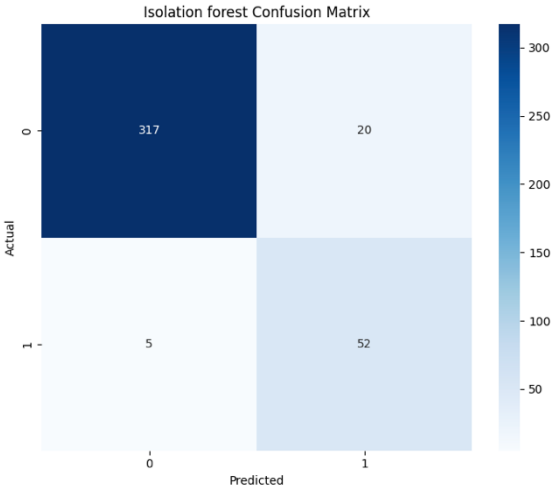
ה accuracy הוא מדד לכמה הקלאסטרים שלנו טובים, וזהו מדד חשוב במקרה כמו שלנו.

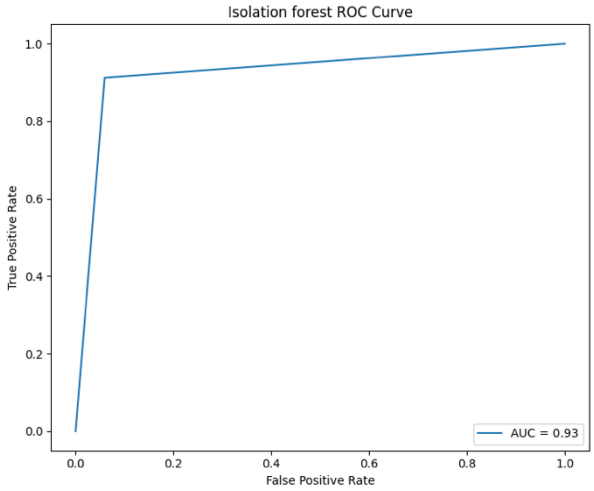
ה support הוא מספר הדגימות שלנו שזהה בכל המודלים.

נציג את התוצאות בעבור כל מודל:

- Isolation Forest:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.94      0.96       337
           1       0.72      0.91      0.81        57

    accuracy                           0.94       394
   macro avg       0.85      0.93      0.88       394
weighted avg       0.95      0.94      0.94       394
```



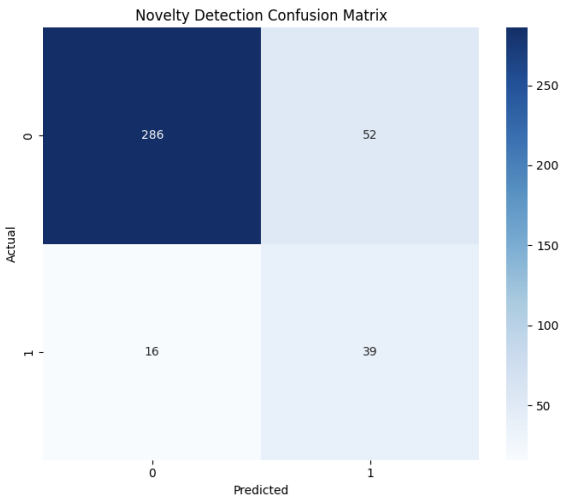


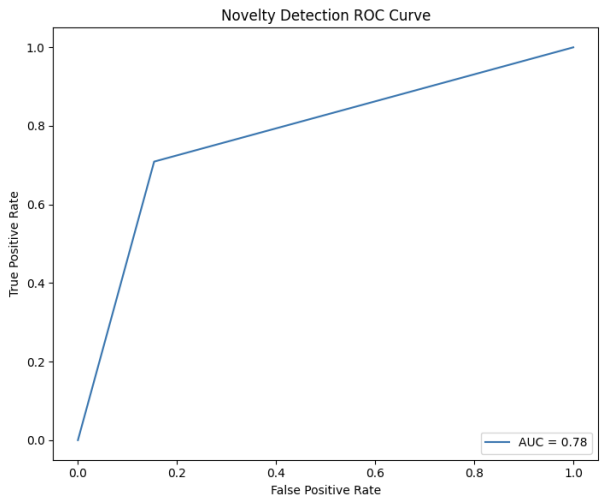
- Accuracy –
0.94: המדד לקלאסטרים שלנו הוא טוב מאוד
- Precision –
עבור 0: 0.98, תוצאה מדהימה
עבור 1: 0.72, תוצאה לא מלהיבה אבל ה Recall שלנו טוב ולכן אנחנו לא מודאגים
- Recall –
עבור 0: 0.94, תוצאה טובה כצפוי
עבור 1: 0.91, תוצאה מצוינת שביחד עם ה Accuracy נותנת לנו תחושה שהמודל טוב מאוד
- F1 – Score –
כצפוי-
עבור 0: 0.96
עבור 1: 0.81
- Support –

עבור 0: 337
עבור 1: 57
סה"כ: 394

- Local Outlier Factor:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.85      0.89       338
           1       0.43      0.71      0.53        55

    accuracy                           0.83       393
   macro avg       0.69      0.78      0.71       393
weighted avg       0.87      0.83      0.84       393
```





- Accuracy –

0.83: המדד לקלאסטרים שלנו הוא לא מלהיב אבל גם ממש לא גרוע

- Precision –

עבור 0: 0.95, תוצאה מדהימה

עבור 1: 0.43, תוצאה גרועה, נרצה לבדוק מה מצב ה Recall אך היא מדאיגה

- Recall –

עבור 0: 0.85, תוצאה לא מלהיבה אך לא גרועה(וגם לא מעניינת כמו התוצאה עבור 1)

עבור 1: 0.71, תוצאה לא טובה שביחד עם הPrecision נותנת תחושה שהמודל לא מתאים לצורכינו

- F1 – Score –

כצפוי-

עבור 0: 0.89

עבור 1: 0.53

- Support –

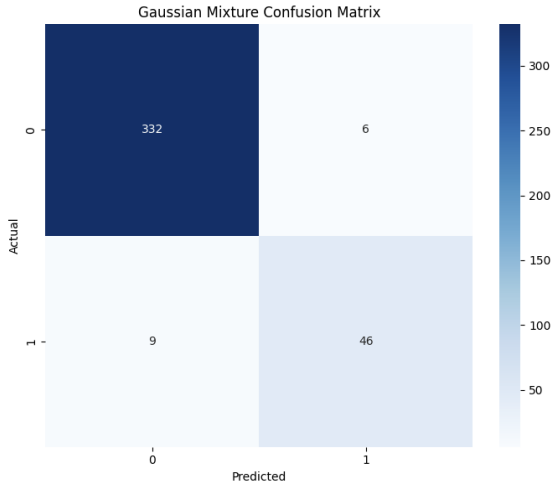
עבור 0: 338

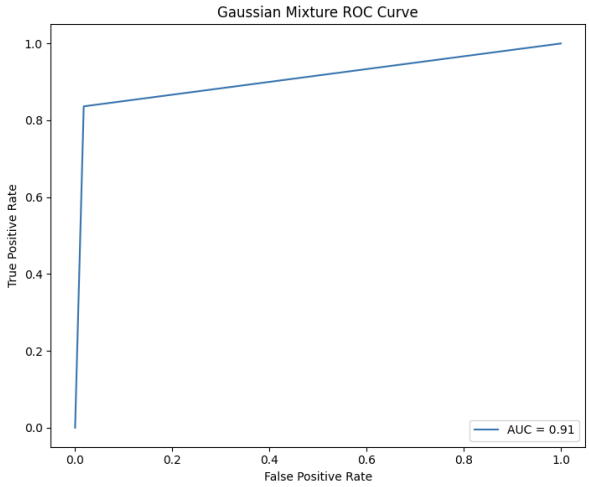
עבור 1: 55

סה''כ: 393

- Gaussian Mixture

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       338
           1       0.88      0.84      0.86        55

    accuracy                           0.96       393
   macro avg       0.93      0.91      0.92       393
weighted avg       0.96      0.96      0.96       393
```





- Accuracy –

0.96: המדד לקלאסטרים מצוין

- Precision –

עבור 0: 0.97, תוצאה מדהימה

עבור 1: 0.88, תוצאה די טובה

- Recall –

עבור 0: 0.98, תוצאה מצוינת

עבור 1: 0.84, תוצאה סבירה, ביחד עם ה Precision אנחנו מבינים שמודל זה כמובן טוב יותר מהמודל הקודם אך פחות טוב מהמודל הראשון(Isolation Forest).

- F1 – Score –

כצפוי-

עבור 0: 0.98

עבור 1: 0.86

- Support –

עבור 0: 338

עבור 1: 55

סה"כ: 393

- Autoencoder

```
CONFUSION MATRIX:
 [[335   3]
 [ 38  17]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.99      0.94       338
           1       0.85      0.31      0.45        55

    accuracy                           0.90       393
   macro avg       0.87      0.65      0.70       393
weighted avg       0.89      0.90      0.87       393
```

- Accuracy –

0.90 המדד לקלאסטרים מצוין

- Precision –

עבור 0: 0.90, תוצאה מדהימה

עבור 1: 0.85, גם תוצאה טובה

- Recall –

עבור 0: 0.99, תוצאה מצוינת, וכמעט מושלמת.

עבור 1: 0.31 תוצאה מאוד נמוכה, גורמת לנו להבין כי המודל אולי אינו מתאים לבעיה על אף שאר המדדים הגבוהים .

- F1 – Score –

כצפוי-

עבור 0: 0.94

עבור 1: 0.45

- Support –

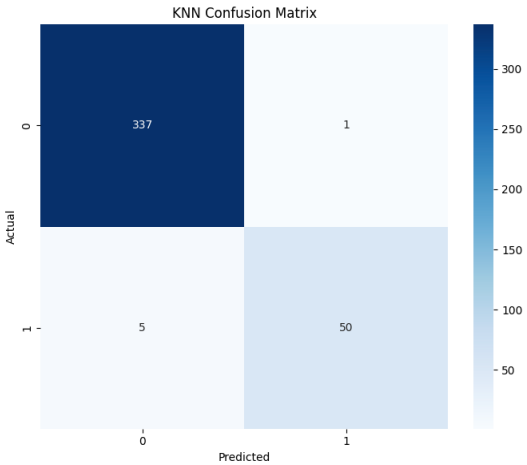
עבור 0: 338

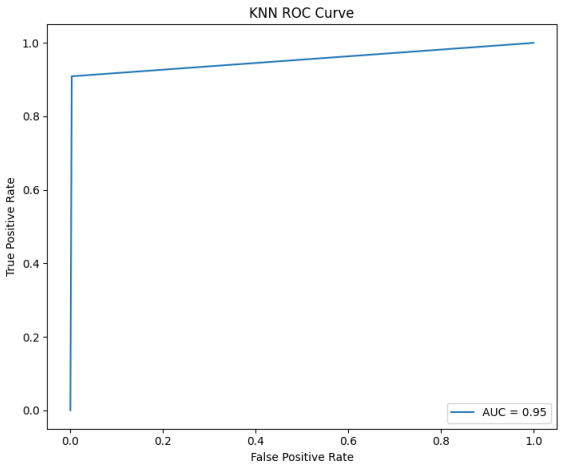
עבור 1: 55

סה"כ: 393

- KNN

	precision	recall	f1-score	support
0	0.99	1.00	0.99	338
1	0.98	0.91	0.94	55
accuracy			0.98	393
macro avg	0.98	0.95	0.97	393
weighted avg	0.98	0.98	0.98	393





- Accuracy –
0.98: המדד לקלאסטרים מצוין
- Precision –
עבור 0: 0.99, תוצאה מדהימה
עבור 1: 0.98, תוצאה מדהימה והכי טובה שראינו עד כה
- Recall –
עבור 0: 1, תוצאה מצוינת
עבור 1: 0.91 מצוינת שביחד עם הPrecision הייתה יכולה להביא אותנו למסקנה שזהו המודל הטוב ביותר
- F1 – Score –
כצפוי-
עבור 0: 0.99
עבור 1: 0.94
- Support –

עבור 0: 338

עבור 1: 55
סה''כ: 393

ניתן היה להבין ש KNN הוא המודל הטוב ביותר, אך נזכר כי זהו מודל מפוקח, ועל מנת לחזות הונאות נרצה להשתמש במודל לא מפוקח כפי שציינו בתחילת מסמך זה, מהסיבות שאנו רוצים ללמוד את המבנה החבוי של הבעיה ולא לקבל סיטואציה של שינון נקודתי של ערכי קיצון, שזה אכן מה ש- KNN עושה.

בסך הכל, נקבל כי **Isolation Forest** הוא המודל הטוב ביותר עבור מטרתינו.

חלק רביעי:

בשלב זה ביצענו Cross Validation וגם Hyperparameter tunning על המודל הטוב ביותר.

Training Results

results

	Metrics	Isolation Forest	LOF Novelty Detection	Gaussian Mixture	Autoencoder	KNN
0	True Negatives	319	286	332	335	337
1	False Negatives	18	52	6	3	1
2	False Positives	6	16	9	38	5
3	True Positives	51	39	46	17	50
4	Accuracy	0.939086	0.826972	0.961832	0.895674	0.984733
5	Precision	0.73913	0.428571	0.884615	0.85	0.980392
6	Recall	0.894737	0.709091	0.836364	0.309091	0.909091
7	F1-Score	0.809524	0.534247	0.859813	0.453333	0.943396
8	Support	None	None	None	None	None

התוצאות שהתקבלו ב - ו Cross Validation הן:

Cross Validation Results

resultsCrossVal

	Metrics	Isolation Forest	LOF (Novelty Detection)	Gaussian Mixture Model	Autoencoders	Knn
0	True Negatives	326	319	332	335	337
1	False Negatives	12	19	6	3	1
2	False Positives	2	32	9	38	5
3	True Positives	53	23	46	17	50
4	Accuracy	0.964377	0.870229	0.961832	0.895674	0.984733
5	Precision	0.815385	0.547619	0.884615	0.85	0.980392
6	Recall	0.963636	0.418182	0.836364	0.309091	0.909091
7	F1-Score	0.883333	0.474227	0.859813	0.453333	0.943396
8	Support	None	None	None	None	None

וב - Hyperparameter tunning

כאן, הערכים שקיבלנו מביאים פרמטרים פחות טובים מהפרמטרים הראשוניים שנבחרו ע״י המודל.

- על מנת לא להעמיס על המחברת הן מבחינת מלל וקוד, והן מבחינת זמן ריצה וכובד מחקנו את החלק הבא, אך בכל זאת נפרט בדו״ח. ניסינו להגיד את הטווחים של הפרמטרים מספר פעמים ובכל פעם ביצענו מחדש - Hyperparameter tunning. עם זאת בכל הרצה עדיין הפרמטרים הביאו תוצאות פחות טובות אחרי שביצענו עימם Cross Validation.

ניסינו לעשות חיפוש של הפרמטרים שהמודל מג׳נרט לנו כדי למצוא את הפרמטרים אשר מביאים את התוצאות ביותר, ולאחר זמן גדול של הרצה, מצאנו אכן פרמטרים טובים יותר (גם עליהם ביצענו Cross Validation, אך עדיין לא הצלחנו למצוא את הפרמטרים אותם המודל מג׳נרט לבד. לכן, לא החלפנו אחרי ביצוע של Hyperparameter tunning את הפרמטרים שלנו.

חלק חמישי:

בדיקות זמן ריצה וקצב התכנסות:

- בעבור המודל GMM בדקנו הן זמן ריצה והן קצב התכנסות, כיוון שבמודל הטוב ביותר (isolation forest) זו אינה שיטה איטרטיבית ולכן אין לה קצב התכנסות.

```
# Start the timer
start_time = time.time()

# Train the model
gmm = GaussianMixture(n_components=3, random_state=42)
gmm.fit(X_train)

# End the timer and calculate elapsed time
end_time = time.time()
training_time = end_time - start_time

print(f"Gaussian Mixture Model Training Time: {training_time} seconds")
print(f"Gaussian Mixture Model Converged in {gmm.n_iter_} iterations")
```

```
Gaussian Mixture Model Training Time: 0.07995200157165527 seconds
Gaussian Mixture Model Converged in 7 iterations
```

- בעבור המודל Isolation forest

```
# Measure the training time
start_time = time.time()
if_model.fit(X_train)
training_time = time.time() - start_time

# Measure the prediction time
start_time = time.time()
predictions = if_model.predict(X_test)
prediction_time = time.time() - start_time

# Print the results
print(f"Training Time: {training_time:.4f} seconds")
print(f"Prediction Time: {prediction_time:.4f} seconds")
```

```
Training Time: 1.7310 seconds
Prediction Time: 0.0310 seconds
```

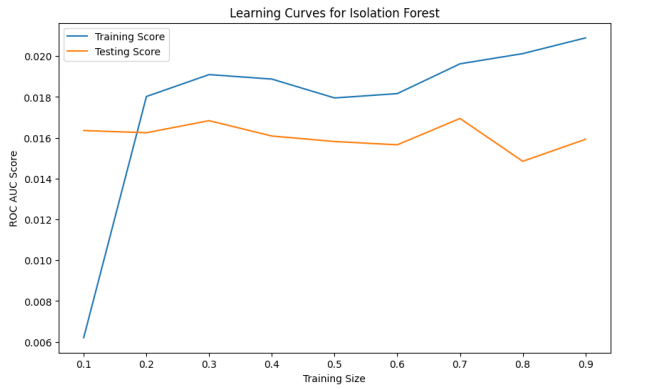
- בעבור המודל KNN

```
# Start the timer
start_time = time.time()

# Train the model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# End the timer and calculate elapsed time
end_time = time.time()
training_time = end_time - start_time

print(f"KNN Training Time: {training_time} seconds")
```

```
KNN Training Time: 0.015988826751708984 seconds
```

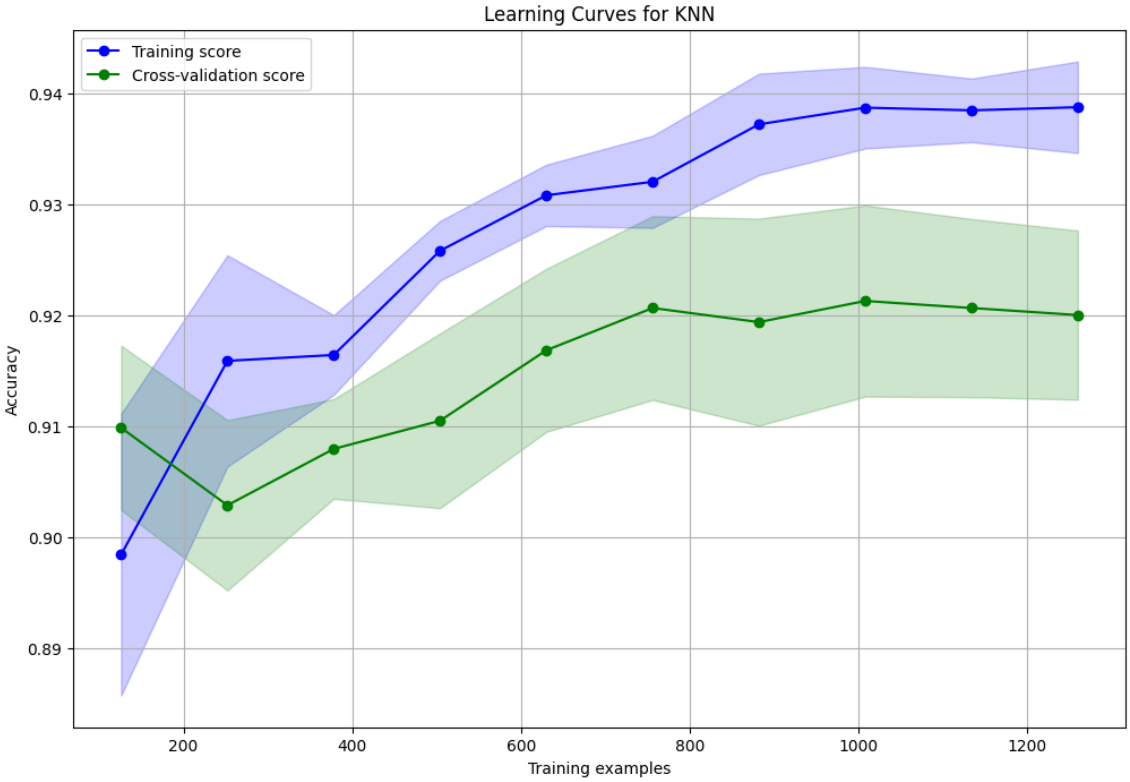
בדיקות Overfiiting, וטיפול ע״י רגולריזציה:

נרצה לחפש האם Isolation forest הוא Overfitting:



- העקומה הכחולה נשארת נמוכה בעבור כמויות שונות של גודל $Training\ size$, זה מביע לנו כי המודל לא עושה Overfitting.
- העקומה הכתומה גם נמוכה בעבור כמויות שונות של גודל $Training\ size$, אך עם זאת, היא לא ״רגישה״ לגודל כמות הדאטא, והיא יחסית קרובה לעקומה הכחולה.
- סה״כ נראה כי אין Overfitting.

נרצה לחפש האם KNN הוא Overfitting:



- ה Training Accuracy שלנו גבוהה, כפי שציינו קודם, אכן המודל מצליח "לשנן" את הדאטא, מה שמחזק הבנה זו הוא כי ה Cross-Validation Score יותר נמוך ברוב הזמן ונראה גמיש יותר, וכי הדאטא שנלקח למען למידה אכן מביא תוצאות טובות יותר, ז"א כי המודל מצליח לנבא בצורה משמעותית טובה יותר על דאטא שראה מאשר דאטא שלא ראה, וזהו סימן ל – Overfitting.
- כעת נרצה לעשות "רגולריזציה", כיוון שזו רגולריזציה ל- KNN המטרה שלנו היא למצוא את מספר השכנים האופטימלי.
- השיטות בהן השתמשנו:
  I. Feature Scaling
  II. PCA
  III. קומבינציה של השניים

בכל השיטות קיבלנו כי האופציה הטובה ביותר היא כאשר יש שכן 1 לכל פיצ׳ר.

עם זאת, בדקנו מהי השיטה הטובה ביותר:

```
Baseline Model Accuracy: 0.9294712364776059
Scaled Model Accuracy: 0.982208067940552
PCA Model Accuracy: 0.9294712364776059
Scaled + PCA Model Accuracy: 0.9650551005965019
```

קצת הסברים על הממצאים:

כאשר יש שכן אחד לכל פיצ׳ר זה עשוי להצביע על overfitting, במיוחד בתרחישים של זיהוי הונאות. בנוסף עשוי להצביע על כך שהמודל לא מצליח לזהות את התבניות החבויות בנתונים.

כעת כל הממצאים הללו מאששים לנו את ההבנה כי על אף שהמודל בעל Accuracy גבוהה, אינו מתאים לבעיה.

חלק שישי:

אימון המודל על ה- Test

```
CONFUSION MATRIX:
 [[99  8]
 [ 1 15]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.93      0.96       107
           1       0.65      0.94      0.77        16

    accuracy                           0.93       123
   macro avg       0.82      0.93      0.86       123
weighted avg       0.95      0.93      0.93       123
```

כל המדדים גבוהים, Accuracy, Recall גבוהים כנדרש.

המודל – Isolation forest אכן מצליח ללמוד את הדאטא ולזהות את פעולות ההונאה.