<h1> DL Final Project Semester 1 2025</h1>

Presenter Ezra Ella 034199943

Collaborator Bezalel Itzhaky 058750191

1) בחר דאטהסט מתוך KAGGLE לפתרון בעיית קלאסיפיקציה / רגרסיהCLUSTERING /
2) בצע EDA לדאטה סט
3) בצע PREPROCESSING לפי הנדרש:
    a. ניקוי נתונים
    b. השלמת נתונים
    c. הורדת מימדים
    d. איזון הנתונים
    e. נירמול
    f. Feature engineering

4) חלק ל. training / validation / test sets
5) הפעל אלגוריתם למידת מכונה בסיסי לצורך השוואת התוצאות (עץ החלטה-K / Linear regression , SVM / means)
6) בחן את התוצאות שהתקבלו לפי 5-3 מדדים רלוונטיים
7) בחר רשת נוירונים רלוונטית והפעל אותה על הנתונים. יש לבחור אותה עם היפרפרמטרים המהווים ברירת מחדל .
8) השווה את התוצאות למדדים שהתקבלו בסעיף 6
9) בחר 3 היפרפרמטרים ובהתייחס לכל אחד מהם, שנה אותו ל 3 ערכים שונים ובחן את התוצאות.
10) שנה את הדאטהסט (הוסף רשומות / מחק רשומות) כך שהתוצאות ישתפרו באופן משמעותי הסבר.
11) שנה את הדאטהסט (הוסף רשומות / מחק רשומות) כך שהתוצאות יהיו פחות טובות באופן משמעותי הסבר.
12) הצע דרך לשיפור ארכיטקטורת הרשת, והראה כיצד זה משפיע על התוצאות ועל מהירות ההתכנסות
13) הצע מטריקה חדשה והראה מה הערכים שלה לאורך האימון.
14) שנה את האיזון (חוסר האיזון) של הנתונים ל 3 רמות שונות. השווה את התוצאות המתקבלות בכל אחד מהמקרים.
15) בחר שיטה להורדת מימדים, הפעל אותה על הדאטהסט והראה מהן התוצאות עבור הדאטהסט המצומצם.

1) בחר דאטהסט מתוך KAGGLE לפתרון בעיית קלאסיפיקציה / רגרסיהCLUSTERING /

a) Final_Augmented_dataset_Diseases_and_Symptoms.csv

b) print("Dataset shape:", Org_data.shape)

i) Dataset shape: (246945, 378)

ii) יש 246945 תצפיות ו 377 סימפטומים (עמודה אחת היא שמות המחלות)

c) print("Unique diseases count:", Org_data['diseasesd'].nunique())

i) Unique diseases count: 773

ii) יש 773 מחלות שונות

d) הצגנו 5 רשומות עליונות ו 5 רשומות תחתונות

2) בצע EDA לדאטה סט

a) <h1>2. Data preprocessing: . Exploratory Data Analysis (EDA)</h1>

בגלל המורכבות שרון הציעה לקחת את 20 המחלות עם מספר התצפיות הגדול ביותר , בפועל לקחנו 22

b) <h1>From Now on our project data is data_df the top 22 from Kaggle</h1>

i) <h2>Print the information of the original (raw) data and the processed project data.</h2>

(1) נתוני המקור מ kaggle

```
Kagel Raw dataset info
********************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246945 entries, 0 to 246944
Columns: 378 entries, diseases to neck weakness
dtypes: int64(377), object(1)
memory usage: 712.2+ MB
None
********************
Raw Dataset shape:     (246945, 378)
Raw Unique diseases count:  773
```

(2) נתוני המקוצץ ל 22 מחלות

```
procecesd to top 22 diseases dataset info
********************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26705 entries, 0 to 26704
Columns: 378 entries, diseases to neck weakness
dtypes: int64(377), object(1)
memory usage: 77.0+ MB
None
********************
processed shape: (26705, 378)
processed Unique diseases count: 22
```

c) <h2>describe the numeric statistics of the processed project data </h2>

i) למעשה כל 377 הסימפטומים שהם הפרדיקטורים. ערכי כולם כמותיים הם או 0 או 1

ii) כך נראית הטבלה בעמודים הראשונים

```
1 data_df.describe()
```
✓ 0.7s

	anxiety and nervousness	depression	shortness of breath	depre or psyc symp
count	26705.000000	26705.000000	26705.000000	26705.00
mean	0.022131	0.022580	0.065643	0.04
std	0.147111	0.148563	0.247662	0.20
min	0.000000	0.000000	0.000000	0.00
25%	0.000000	0.000000	0.000000	0.00
50%	0.000000	0.000000	0.000000	0.00
75%	0.000000	0.000000	0.000000	0.00
max	1.000000	1.000000	1.000000	1.00

8 rows × 377 columns

d) <h2>describe the none numeric statistics of the processed project data </h2>

i) רק שמות המחלות הם נתונים קטגוריאלים

ii) במקוצץ שלנו יש 22 מחלות

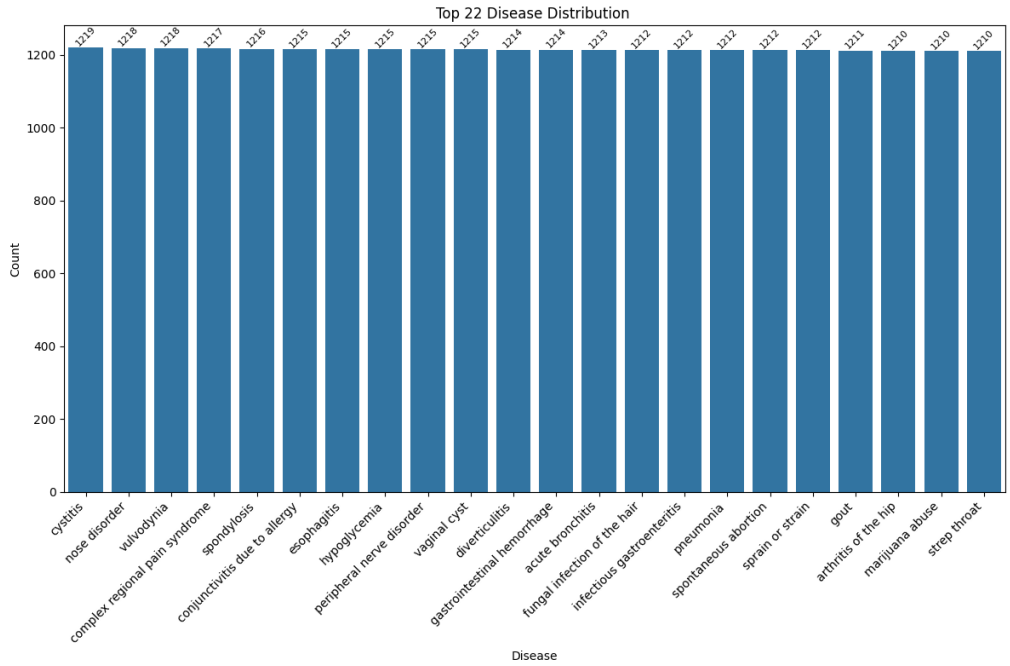
max frequency is: cystitis with trequecy of: 1219

min frequency is: arthritis of the hip with trequecy of: 1210

	diseases
count	26705
unique	22
top	cystitis
freq	1219

(iii

e) <h2> Plot the distribution of the top 22 diseases in processed project data.</h2>



f) <h2> Table displaying the distribution of the top 22 diseases in the processed project dataset.

	Disease	Count
0	cystitis	1219
1	nose disorder	1218
2	vulvodynia	1218
3	complex regional pain syndrome	1217
4	spondylosis	1216
5	hypoglycemia	1215
6	conjunctivitis due to allergy	1215
7	peripheral nerve disorder	1215
8	vaginal cyst	1215
9	esophagitis	1215
10	gastrointestinal hemorrhage	1214
11	diverticulitis	1214
12	acute bronchitis	1213
13	spontaneous abortion	1212
14	sprain or strain	1212
15	infectious gastroenteritis	1212
16	fungal infection of the hair	1212
17	pneumonia	1212
18	gout	1211
19	arthritis of the hip	1210
20	marijuana abuse	1210
21	strep throat	1210

בצע PREPROCESSING לפי הנדרש:

a) ניקוי נתונים
   i) הנתונים נמצאו נקיים לכן אין קוד
b) השלמת נתונים
   i) הנתונים נמצאו שלמים לכן אין קוד
c) נירמול
   i) מכיוון שכל הנתונים הם או 0 או 1 בחרנו אחרי התייעצות עם שרון לא לנרמל
d) איזון הנתונים

מכיוון שבחרנו להתמקד ב-22 המחלות השכיחות, כאשר השכיחות הגבוהה ביותר היא 1,219 והנמוכה ביותר היא 1,210, הנתונים מאוזנים ואין צורך להפעיל שיטות לאיזון. בתחילת העבודה, כאשר השתמשנו בכל בסיס הנתונים (לפני הצעתה של שרון להתמקד במחלות השכיחות), ביצענו איזון באמצעות שיטות של **resampling**.

e) בשלב זה לא בוצע (כי נתוני העץ וה NN היו גבוהים מלכתחילה ולכן יבוצע בהמשך)
   i) הורדת מימדים יבוצע במסגרת PCA
   ii) Feature engineering

3) חלק ל. training / validation / test sets

a) <h1>preapre X and Y and split it for modeles to train</h1>
b) <h2>separate to x and y</h2>

בחרנו לבצע חלוקה בשני אופנים **Simple** :(ברירת מחדל) ו **Stratify**,-כדי להשוות את ביצועי עץ ההחלטות ולבחור את השיטה העדיפה לאמידת המודל וללמידת מכונה.

בחלוקה לפי ברירת המחדל, **(Simple** ללא **Stratify**) בסט הנתונים **Testing**, נמצאו רק 4 מחלקות עם **Support**, ולכן הערכנו את ביצועי המודל באמצעות average='weighted'. התוצאות שהתקבלו היו נמוכות, מה שהוביל אותנו להבין שכדאי להשתמש ב **Stratify Split**-כבסיס לכל אימון.

באמצעות **Stratify**, כל מחלקה קיבלה כ-180 דוגמאות (Support), מה שאפשר לנו להעריך את המודל לפי average='macro'.

מכך הסקנו שחלוקה ללא **Stratify** יוצרת **חוסר איזון בין המחלקות**, מה שמוביל להבדלים משמעותיים בין נתוני האימון לנתוני המבחן וגורם לירידה בביצועי המודל.

<h3> y to One-Hot Encoding</h3>

<h3> step 1 :create a series of the target call it y_org</h3>

<h3> step 2 :convert y_org to df call it y_org_df</h3>

<h3> step 3 :Determine the number of unique classes in `y` </h3>

22

<h3> step 4 :Convert disease labels (strings) to integer labels</h3>

<h3>Step 5: Convert the integer labels to one-hot encoded format</h3>

	y_org	y_org_df	y_encoded	y_cat
shape	(26705,)	(26705, 2)	(26705,)	(26705, 22)
type	Series	DataFrame	ndarray	ndarray

Out	1 y_org ✓ 0.0s 0 infectious gastroenteritis 1 infectious gastroenteritis 2 infectious gastroenteritis 3 infectious gastroenteritis 4 infectious gastroenteritis ... 26700 strep throat 26701 strep throat 26702 strep throat 26703 strep throat 26704 strep throat Name: diseases, Length: 26705, dtype: object	1 y_org_df ✓ 0.0s index diseases 0 0 infectious gastroenteritis 1 1 infectious gastroenteritis 2 2 infectious gastroenteritis 3 3 infectious gastroenteritis 4 4 infectious gastroenteritis ... ... 26700 26700 strep throat 26701 26701 strep throat 26702 26702 strep throat 26703 26703 strep throat 26704 26704 strep throat 26705 rows × 2 columns	array([11, 11, 11, ..., 19, 19, 19])	array([[0., 0., 0., ..., 0., 0., 0.], [0., 0., 0., ..., 0., 0., 0.], [0., 0., 0., ..., 0., 0., 0.], ..., [0., 0., 0., ..., 1., 0., 0.], [0., 0., 0., ..., 1., 0., 0.], [0., 0., 0., ..., 1., 0., 0.]])

c) <h2>simple split to train test and validation</h2>

=== Evaluation Metrics for the Tree Model Using Simple Split ===
Accuracy: 0.00%
Precision: 100.00%
Recall: 0.00%
F1 Score (macro): 0.0000

d) <h4> Overview of simple(default) Model Evaluation</h4>

=== Evaluation Metrics for the Tree Model Using Simple Split ===
Accuracy: 0.00%
Precision: 100.00%
Recall: 0.00%
F1 Score (macro): 0.0000

These metrics indicate a completely dysfunctional model that is likely making extreme misclassifications.

Accuracy: 0.00% → The model is always wrong; it never makes a correct prediction.

Precision: 100.00% → When the model does make a positive prediction, it is always correct (but this is misleading).

Recall: 0.00% → The model never predicts the actual positive class. It likely predicts only negatives.

F1 Score: 0.0000 → Since recall is 0, the F1 score is also 0, meaning the model is completely ineffective.

e) <h2> stratify split to train test and validation</h2>

sets shapes per stratify split
Training set: (18693, 377) (18693, 1)
Validation set: (4006, 377) (4006, 1)
Test set: (4006, 377) (4006, 1)

f) <h2> create a tree for both splits and evaluate</h2>

g) <h3> create Tree Model Using Simple Split(default parameters) </h3>

h) <h3>Evaluation Metrics for the Tree Model Using Simple Split</h3>

```
Evaluation Metrics for the Tree Model Using Simple Split
Accuracy: 0.00%
Precision: 100.00%
Recall: 0.00%
F1 Score (macro): 0.0000
```

i) <h3> interpretation Metrics for the Tree Model Using Simple Split<</h3>

These metrics indicate a **completely dysfunctional model** that is likely making extreme misclassifications.

Accuracy: 0.00% → The model is always wrong; it never makes a correct prediction.
Precision: 100.00% → When the model does make a positive prediction, it is always correct (but this is misleading).
Recall: 0.00% → The model **never** predicts the actual positive class. It likely predicts only negatives.
F1 Score: 0.0000 → Since recall is 0, the F1 score is also 0, meaning the model is completely ineffective.

Next Steps: verify benefit of stratification split

j) <h3> create a tree for stratify splits and evaluate</h3>

=== Evaluation Metrics for the Tree Model Using Stratified Split ===
Accuracy: 95.56%
Precision: 95.57%
Recall: 95.56%
F1 Score (weighted): 0.9555

i) These metrics indicate a **well-performing model** with strong predictive power.
Accuracy (95.56%) → The model correctly classifies 95.56% of all instances, meaning it's highly reliable overall.
Precision (95.57%) → When the model predicts a positive class, it's correct 95.57% of the time. This suggests **low false positives**.
Recall (95.56%) → The model successfully identifies 95.56% of all actual positive cases, meaning **low false negatives**.
F1 Score (0.9555) → A high F1 score confirms a strong balance between precision and recall.

ii) Interpretation:
This model is **highly effective**, with **consistent performance across precision, recall, and accuracy**. There's no major bias towards false positives or false negatives, which is ideal for most applications.

k) **Next Steps: compare and choose the better split**

l) <h3> Compare the Performance of Simple vs. Stratified Split</h3>
accuracy_strat is better it is 95.56%
Precision_simple is better it is 100.00%
Recall_strat is better it is 95.56%
F1_strat is better it is 0.96
Stratification performs better across all metrics except precision, so we will proceed with stratified splitting.

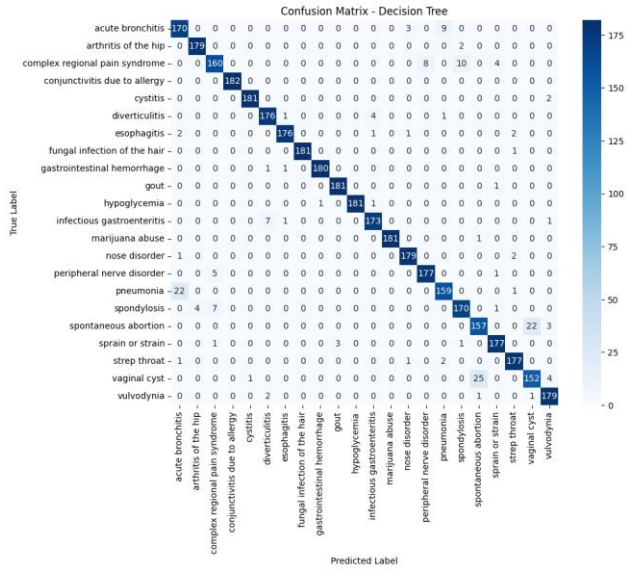
i) חלוקת ברירת המחדל ללא Stratify נתנה ביצועים גרועים. ולכן המשכנו עם החלוקה של ה Stratify שנתנה ביצועים טובים מאוד.

4) הפעל אלגוריתם למידת מכונה בסיסי לצורך השוואת התוצאות (עץ החלטה / K-means) / SVM , Linear regression

5) בחן את התוצאות שהתקבלו לפי 3-5 מדדים רלוונטיים

	Model	Accuracy	Precision	Recall	F1 Score
0	Simple split	0.00%	100.00%	0.00%	0.0
1	Stratify split	95.56%	95.57%	95.56%	0.96%

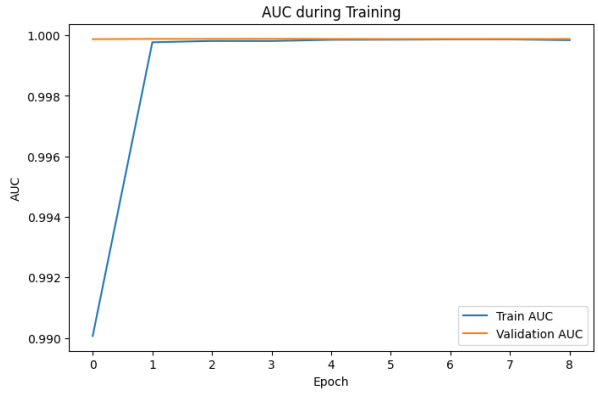
פה מוצגת ה Confusion Matrix.



6) בחר רשת נוירונים רלוונטית והפעל אותה על הנתונים. יש לבחור אותה עם היפרפרמטרים המהווים ברירת מחדל .

7) השווה את התוצאות למדדים שהתקבלו בסעיף 6

	Model	Accuracy	Precision	Recall	F1 Score
0	Simple split	0.00%	100.00%	0.00%	0.0
1	Stratify split	95.56%	95.57%	95.56%	0.96%
2	Defult NN Recall	96.16%	96.35%	96.15%	0.96%
3	Defult NN Loss	96.01%	96.10%	96.00%	0.96%
4	Defult NN AUC	96.16%	96.35%	96.15%	0.96%



8) בחר 3 היפרפרמטרים ובהתייחס לכל אחד מהם, שנה אותו ל 3 ערכים שונים ובחן את התוצאות.

```
learning_rates = [0.001, 0.01, 0.1]
batch_sizes = [16, 32, 64]
epoch_list = [30, 50, 70]
Best hyperparameters (learning rate, batch size, epochs): (0.01, 32, 30)
with test accuracy: 96.33%
Learning Rate: 0.01
Batch Size: 32
Epochs: 30
```

	Model	Accuracy	Precision	Recall	F1 Score
0	Simple split	0.00%	100.00%	0.00%	0.0
1	Stratify split	95.56%	95.57%	95.56%	0.96%
2	Defult NN Recall	96.16%	96.35%	96.15%	0.96%
3	Defult NN Loss	96.01%	96.10%	96.00%	0.96%
4	Defult NN AUC	96.16%	96.35%	96.15%	0.96%
5	Best NN Recall	96.16%	96.35%	96.15%	0.96%

9) שנה את הדאטהסט (הוסף רשומות / מחק רשומות) כך שהתוצאות ישתפרו באופן משמעותי הסבר

10)עצם העובדה שקיצצנו משמעותיתאת הנתוניסוחילקנו Stratify נתן תוצאות גבוהות מאוד.

11)שנה את הדאטהסט (הוסף רשומות / מחק רשומות) כך שהתוצאות יהיו פחות טובות באופן משמעותי הסבר.

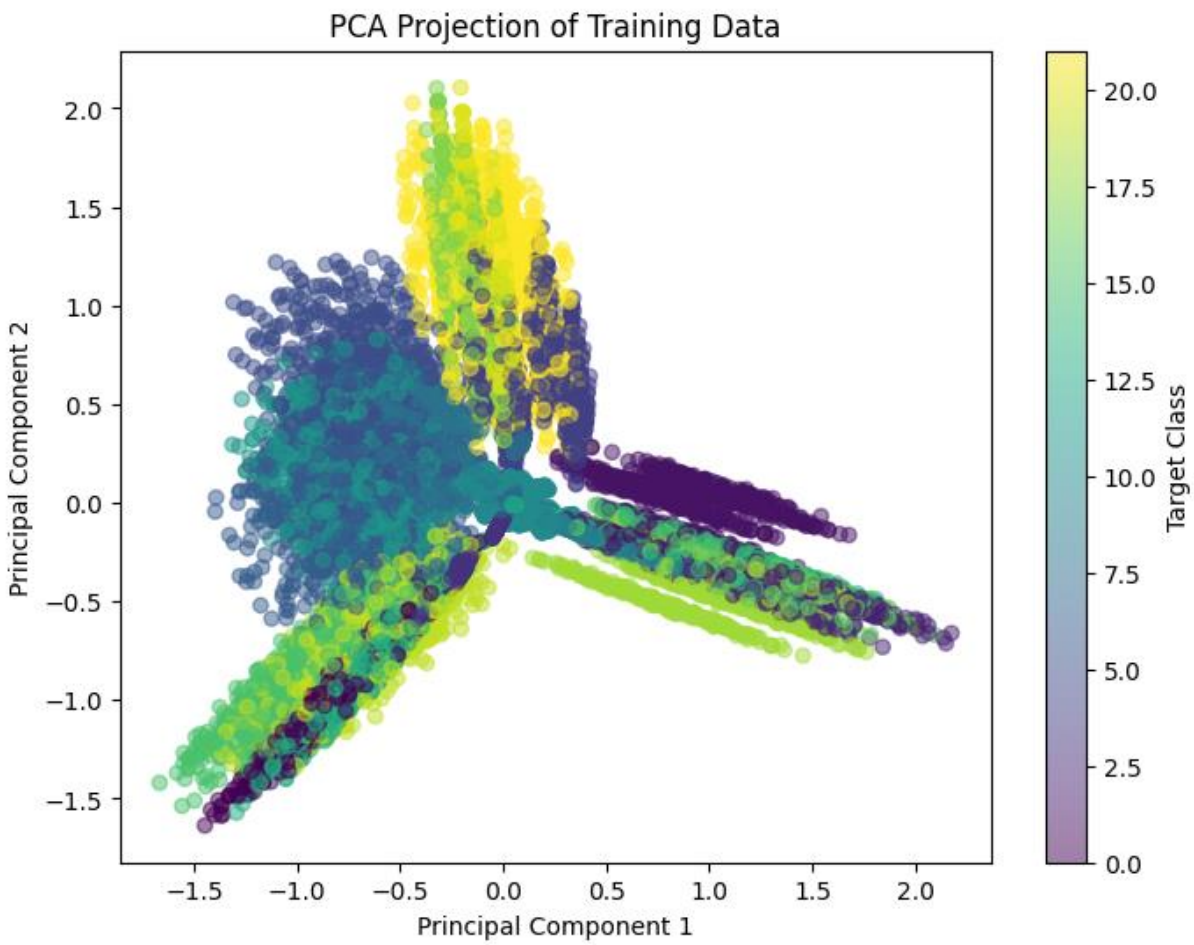
```
a) <h1> 10. Dataset Modification for Performance Degradation</h1>
   i) <h2> Randomly remove 50% portion of the data</h2>
source of data shape is: (26705, 378)
data_degraded shape is: (13352, 378)
samples reduced 13353
data degrade number of categories 22
   ii)  <h2> Preprocessing for the degraded dataset<h2>
      (1)     <h3> convert categorial to dumies</h3>
      (2)     Explanation:
      - pd.get_dummies(): This is a function from the `pandas` library
      that is used to convert categorical (non-numeric) columns into
      multiple binary (0 or 1) columns. This process is called **one-hot
      encoding**. Each category in the original column gets its own column
      with binary values: `1` if the record belongs to that category, and
      `0` otherwise.
      - data_degraded_processed: This DataFrame transformation includes
      the source data, where non-numeric columns are converted into one-
      hot encoded columns.
```

```
- `columns=non_numeric_columns`: This specifies which columns to
apply one-hot encoding to. The `non_numeric_columns` is expected to
be a list of column names in `data_degraded_processed` that contain
categorical data. You only apply one-hot encoding to these columns.
- `drop_first=True`: This argument drops the first category in each
specified column during the encoding process. This is done to avoid
the **dummy variable trap**, which is a problem that can occur when
you have multicollinearity in a regression model. By dropping the
first category, you remove redundancy, as the first category can be
inferred from the other columns.
```

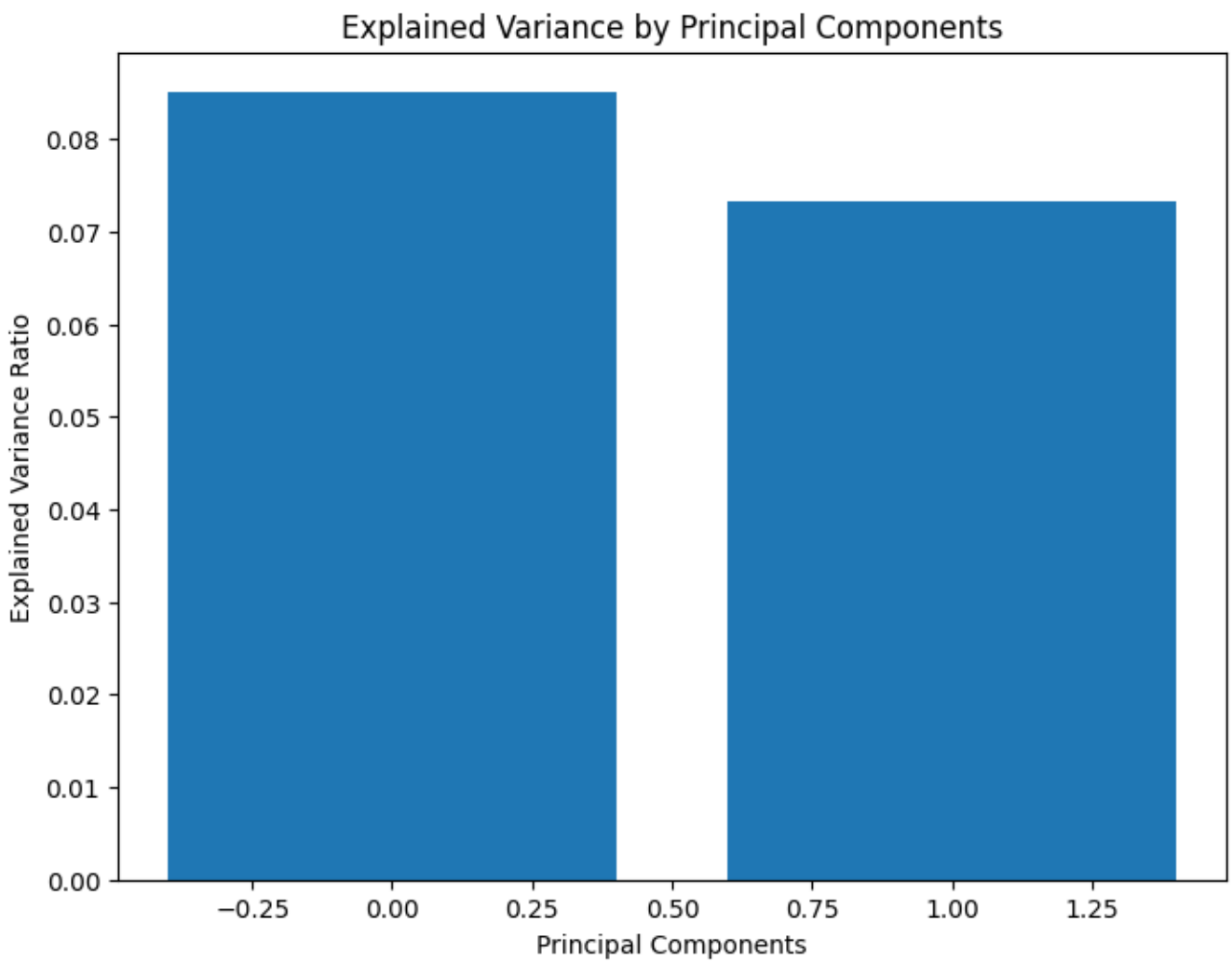
12)חזרנו למה שקודם לא בוצע (כי נתוני העץ וה NN היו גבוהים מלכתחילה ולכן יבוצע בהמשך)

a) הורדת מימדים

1) <h2>PCA</h2>
   a) <h3>Apply PCA to reduce dimensions to 2 for visualization</h3>
   b) <h3>>\ Visualize the 2D projection of the training data</h3>



   c) <h3> Visualize explained variance ratio</h3>



d) <h3> Apply PCA to retain 95% variance and Visualization</h3>

Original feature shape: (18693, 377)

Reduced feature shape: (18693, 100)

Decision Tree Accuracy on PCA-reduced data: 0.93%

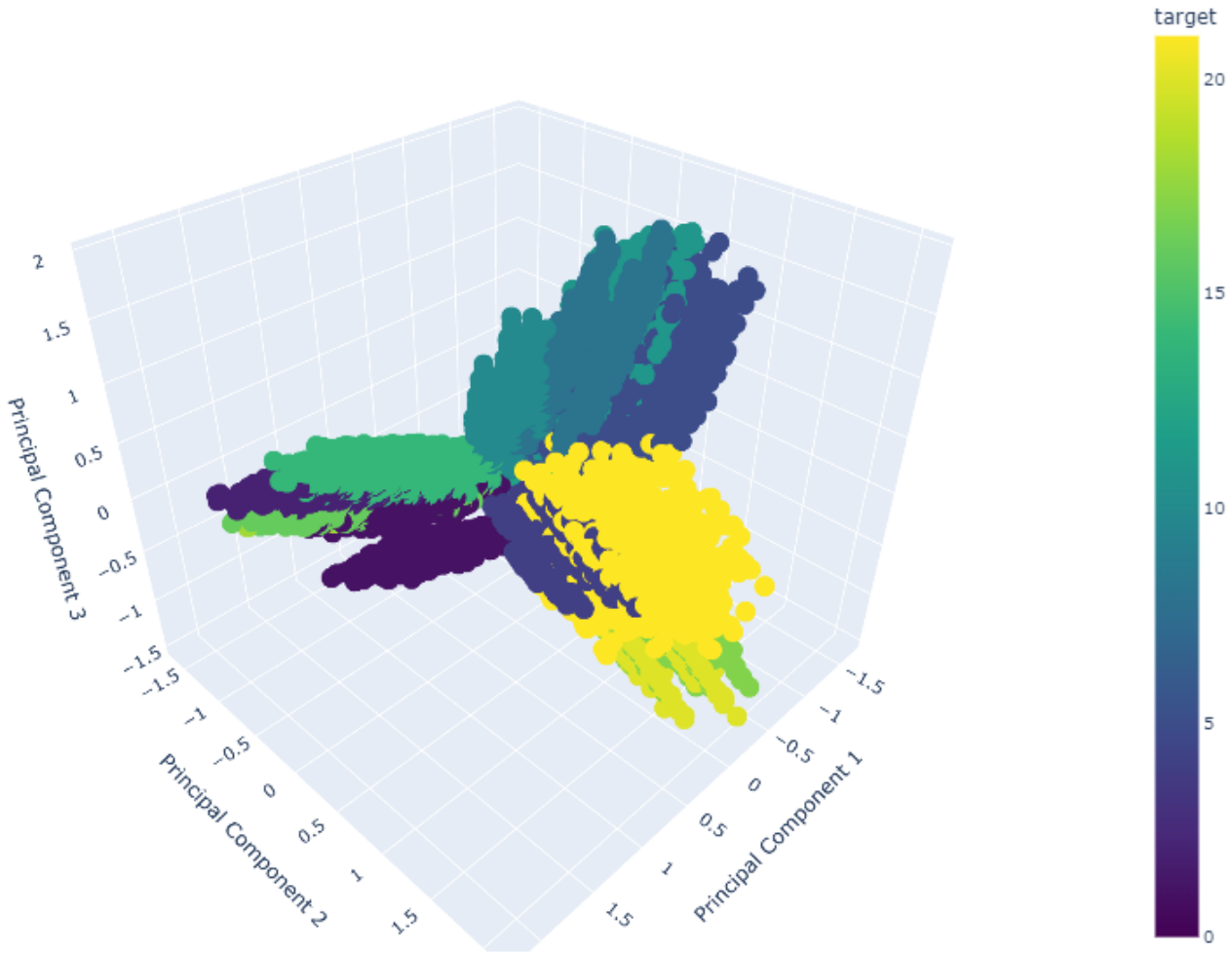
e) <h3> show how shape was reduced</h3>

Original feature shape: (18693, 377)

Reduced feature shape: (18693, 100)

number of reduced predictors 277

f) <h3> 3D Visualization of PCA-reduced data<h3>



2)

a)
b) איזון הנתונים
c) Feature engineering

סעיפים אלו מתוארים במאמר.

13) הצע דרך לשיפור ארכיטקטורת הרשת, והראה כיצד זה משפיע על התוצאות ועל מהירות ההתכנסות
14) הצע מטריקה חדשה והראה מה הערכים שלה לאורך האימון.
15) שנה את האיזון (חוסר האיזון) של הנתונים ל 3 רמות שונות. השווה את התוצאות המתקבלות בכל אחד מהמקרים.
16) בחר שיטה להורדת מימדים, הפעל אותה על הדאטהסט והראה מהן התוצאות עבור הדאטהסט המצומצם.