

## עבודה מסכמת ניתוח נתוניים

### מבוא:

מטרה זו עוסקת בניתוח נתונים EEG, רישום פעילות חשמלית של המוח) אשר נאספו במהלך ניסוי של דמיון תנועתי (Motor Imagery) של יד ימין ויד שמאל. נתונים אלו נאספו משני ערוצי EEG מרכזיים- C3 ו- C4, אשר תואימים לאזור השליטה בידיים בקורטקס המוחי. מטרתה של מטריה זו היא ביצוע של ניתוח של הנתונים, חילוץ תכונות משמעותיות מתוכם ובהבוסס על תכונות אלו לבניית אלגוריתם למידת מכונה אשר יבצע סיווג של כל ניסיון (trial) לפי תוויתו (ימין או שמאל). תהליך זה משלב שלבים של ויזואלייזציה, ניתוח ספקטורי, הפקת תכונות, הפחתת מדדים (PCA) ולבסוף סיווג. המטריה מתבססת על עקרונות וכלים אשר נלמדו בקורס יחד עם שיפור הביצועים לצורך השגת רמת דיוק גבוהה ככל הניתן על מערך האימון ומערך המבחן.

### שאלות המטריה:

**שאלה 1 -** טיענת הנתונים, ייצור גרפים של 20 ניסיונות אקראיים מכל אחת משתי הקטגוריות (יד ימין ויד שמאל) יחד עם הצגת שני הערוצים (C3 ו-C4) בכל ניסיון בצד לאבחן הבדלים ויזואליים ראשוניים בין הקבוצות.

**שאלה 2 -** חישוב ספקטרים כוח ממוצע באמצעות שיטת Welch לכל הקטגוריות והערוצים יחד עם בדיקת תחומי תדר וזמן אשר מסוימים להפרדה בין הקבוצות. בניית ספקטוגרמות במטרה להשוות בין הקטגוריות (ביחס ל- baseline) כך שייהי ניתן להזיהות את טווחי הזמן והתדרים האינפורטטיביים בכל אחד מהמצביים.

**שאלה 3 -** חילוץ תכונות הקשורות מאזורים בתדר ובזמן שנמצאו כאינפורטטיביים והוספת שתי תכונות נוספות לשיפור הסיווג. בניית היסטוגרמות של התפלגות התכונות וביצוע הפחתת מדדים (PCA) עם הצגה גרפית הכוללת בתוכה הפרדה לינארית בין הקטגוריות.

**שאלה 4 -** ביצוע אימון מסוג LDA על בסיס התכונות שנוצרו, ביצוע K – fold Cross – Validation, בדיקת ביצועי הסיווג (דיוק, סטיית תקן) ושיפור הביצועים באמצעות הפחתת מדדים (PCA) או בחירת תכונות אופטימליות. החלת תהליך זהה על סט המבחן וдиוקן תוצאות הסיווג.

**שיטות:**

הקוד הסופי חולק לארבעה קבצי `py` אשר מייצגים כל אחד בנפרד את השאלות (`visualization.py`, `power_spectre.py`, `features.py`, `classification.py`). בכל אחד מהקבצים הוגדרו הfonקציות הנדרשות והן מומשו בקובץ נפרד - `main.py`. כדי להריץ את הקוד יש לשמר את כל הקבצים בתיקייה משותפת ולהריץ את הקובץ `main.py`.

**קובץ הנתונים** כוללים הקלטות EEG משני ערוצים (C3 ו- C4) אשר נמדדו במהלך 160 초רות של דמיון תנועה (יד ימין או יד שמאל). קובץ האימון (`motor_imagery_train_data.npy`) מכיל 128 חזרות עם תווית (left/right) וקובץ המבחן (`motor_imagery_test_data.npy`) מכיל 32 חזרות ללא תווית עבור בדיקת המודל. כל חזרה נשכחת במשך 6 שניות ומכליה 768 דגימות.

**טעינת הקבצים (שורות 36-62 בקובץ `main.py`):**

ציירת הfonקציה (`_load_npy_sequence()`) בצד בטוען את כל המרכיבים (numpy arrays) שנשמרו בראף בתוך קובץ `py` אחד וה חוזרים ברשימה. שימוש בלולאת `while` אשר קוראת את המרכיבים עד לסיום הקובץ (EOF). טיענת קובץ האימון (`motor_imagery_train_data.npy`). באמצעות פונקציה זו ולאחר הטעינה בדיקה שהקובץ מכיל ארבעה רכיבים : אוטות EEG גולמיים, תוויות ביןירות לכל ניסוי, שמות התוויות ותדר הדגימה (Hz). רכיבים אלו מחולצים לששתנים מתאימים לצורך ניתוח וייזואלי וסטטיסטי בהמשך.

```

36  # — Utility —
37  def _load_npy_sequence(path: str | Path) -> list[np.ndarray]:
38      """Return *all* numpy arrays stored consecutively in a single .npy file."""
39      out: list[np.ndarray] = []
40      with open(path, "rb") as f:
41          while True:
42              try:
43                  out.append(np_load(f))
44              except EOFError:
45                  break
46      return out

```

```

49  # — Main Pipeline —
50  def main() -> None:
51      train_path = Path("motor_imagery_train_data.npy")
52      test_path = Path("motor_imagery_test_data.npy")
53
54      # === 0. Load TRAIN =====
55      trn = _load_npy_sequence(train_path)
56      if len(trn) != 4:
57          raise ValueError(
58              f"{train_path} must contain 4 arrays "
59              f"(data, labels, labels_name, fs) [ found {len(trn)} ]."
60          )
61      train_data, labels, labels_name, fs_arr = trn
62      fs = int(fs_arr)

```

קובץ `main.py` (שורות 41-81) שימש להרצת כלל פונקציות הסיווג וההערכה אשר יתוארו בהמשך הקוד, והוא על סט האימון והן על סט הבדיקה בהתאם למבנה הנתונים שהוזן.

[קובץ visualization.py לשאלת 1](#)בחירה הניסויים המתאימים (שורות 7-21) :

יצירת הפונקציה extract\_valid\_trials() אשר מקבלת את אותן ה- EEG והתוויות ומחזירה את הניסויים המתאימים בלבד אשר תוויגו ב- right או left. מתוך אותן הפונקציה שומרת את העורצים הרלוונטיים (C3 ו- C4) לפיעלות המוטורית. ניסוי יד שמאל בעל התווית '0' וניסוי יד ימין בעל התווית '1'. תוצר הפונקציה הוא מטריצה של אותן בגודל (n\_trails, n\_smaples, 2) וקטור תוויות בגודל (n\_trails).

```

7 def extract_valid_trials(data: np.ndarray, labels: np.ndarray) -> Tuple[np.ndarray, np.ndarray]:
8     """
9     No artifacts: each trial belongs exclusively to LEFT or RIGHT.
10    Returns:
11    | X | (n_trials, n_samples, 2) ← channels C3,C4
12    | y | (n_trials,)           ← 0=LEFT, 1=RIGHT
13    """
14    left_mask = labels[2] == 1
15    right_mask = labels[3] == 1
16    mask = left_mask | right_mask      # All relevant trials
17
18    X = data[mask, :, :2]            # First two channels
19    y = np.zeros(mask.sum(), dtype=int)
20    y[right_mask[mask]] = 1         # RIGHT → 1
21    return X, y

```

הציג ניסויים אקראיים (שורות 24-43) :

יצירת הפונקציה plot\_random\_trials() אשר מציגה בדקה גרפית 20 ניסויי EEG אקראיים מחלקה נבחרת (יד ימין או יד שמאל). כל גרפ מציג את אותן מושגי העורצים הרלוונטיים כך שניתן לראות בדקה את דפוסי אותן בכל אחת מן המחלקות.

```

24 def plot_random_trials(X: np.ndarray, y: np.ndarray, label_value: int, title: str) -> None:
25     """Displays 20 random trials from the requested class (Left/Right)."""
26     np.random.seed(5)
27     idx = np.where(y == label_value)[0]
28     chosen = np.random.choice(idx, 20, replace=False)
29
30     fig, axes = plt.subplots(4, 5, figsize=(15, 10), sharex=True, sharey=True)
31     for i, ax in enumerate(axes.flat):
32         trial = X[chosen[i]]
33         ax.plot(trial[:, 0], label="C3", color="blue", linewidth=0.7)
34         ax.plot(trial[:, 1], label="C4", color="red", linewidth=0.7)
35         ax.set_xticks([]); ax.set_yticks([])
36         ax.set_title(f"Trial {chosen[i]}")
37     fig.suptitle(title, fontsize=16)
38     fig.text(0.5, 0.04, "Time (samples)", ha='center')
39     fig.text(0.04, 0.5, "Amplitude (µV)", va='center', rotation='vertical')
40     handles, labels_ = axes[0, 0].get_legend_handles_labels()
41     fig.legend(handles, labels_, loc='upper right')
42     plt.tight_layout(rect=[0.04, 0.04, 0.98, 0.94])
43     plt.show()

```

[קובץ power\\_spectra.py לשאלה 2](#)[חילוץ אוטות EEG \(שורות 15-6\)](#)

יצירת הפקציה `extract_trails()` לחילוץ אוטות EEG של ניסויי יד שמאל ויד ימין בטוויה הזמן הרצוי לצורך איתור טוחוי זמן אינפורטטיביים להשוואה בין התנאים.

```

6  # -----
7  # 1. Power Spectrum (Welch) with log10 stability
8  #
9  def extract_trials(data: np.ndarray, labels: np.ndarray,
10     |   |   |   fs: int, start: float, end: float) -> Tuple[np.ndarray, np.ndarray]:
11     """Extract left/right trials using only samples between start and end (in seconds)."""
12     left = labels[2] == 1
13     right = labels[3] == 1
14     t0, t1 = int(start * fs), int(end * fs)
15     return data[left, t0:t1, :], data[right, t0:t1, :]
16

```

[חישוב ספקטרום העוצמה \(יחד עם הבונוס\) \(שורות 17-30\)](#)

יצירת הפקציה `compute_avg_psd_db()` אשר מקבלת מetriיצת ניסויים ומחשבת עבור כל ערוץ בכל ניסוי את ספקטרום העוצמה (Power Spectral Density) באמצעות שיטת Welch. הפקציה מרכזת את העוצמה סביב הממוצע, ממיראת את הערכים לסטולם לוגריתמי (דציבלים) ושומרת את התוצאה עבור כל ערוץ בנפרד. לאחר מכן, מחזירה את מערך התדרים יחד עם ממוצע וסטיית התקן (בונוס) של הספקטרום לאורך כל הניסויים.

```

17  def compute_avg_psd_db(trials: np.ndarray, fs: int):
18      """Compute Welch-PSD in dB scale (log10), with mean & std across trials."""
19      psd_all_db = []
20      for trial in trials:
21          per_trial_db = []
22          for ch in range(trial.shape[1]):
23              sig = trial[:, ch] - np.mean(trial[:, ch])
24              nseg = min(fs, len(sig))
25              f, pxx = welch(sig, fs=fs, nperseg=nseg, detrend='constant')
26              pxx_db = 10 * np.log10(pxx + 1e-12)
27              per_trial_db.append(pxx_db)
28          psd_all_db.append(np.stack(per_trial_db, axis=1))
29      psd_all_db = np.stack(psd_all_db)
30      return f, psd_all_db.mean(axis=0), psd_all_db.std(axis=0)

```

[חישוב ספקטוגרמת EEG \(שורות 59-68\)](#)

יצירת הפקציה `average_spectrogram()` אשר מחשבת את הספקטוגרמה של אות ה- EEG עבור כל אחד מהניסויים בערוץ שנבחר על ידי הסרת הרכיב הקבוע של האות (mean removal) וחישוב הספקטוגרמה על ידי `scipy` מספרית. שמירת התוצאה מכל ניסוי וחישוב הממוצע של כל הספקטוגרמות כך שמקבלת ספקטוגרמה ממוצעת המייצגת את הפעולות התדריתית לאורך זמן עבור כל הניסויים יחד.

212366702 אילונה בר שדה

315262683 שירה אלטרץ

```
59      # -----
60      # 2. Spectrograms - whole trial, +baseline correction
61      #
62      def average_spectrogram(data: np.ndarray, fs: int, chan: int):
63          specs = []
64          for trial in data:
65              f, t, Sxx = spectrogram(trial[:, chan] - trial[:, chan].mean(),
66                                      fs=fs, nperseg=fs // 2)
67              specs.append(Sxx)
68      return f, t, np.mean(specs, axis=0)
```

#### נرمול ספקטוגרמות (שורות 70-77):

יצירת הפונקציה `baseline_corrected()` אשר מחשבת ספקטוגרמות עבור כל ניסוי בנפרד תוך כדי נרמול ביחס לבסיס (`baseline`) המוגדר כעוצמת הספקטרום הממוצעת בשנייה הראשונה של הניסוי. המרת הספקטוגרמה לסלום דציבלים והחזרת ממוצע הספקטוגרמות המנורמלות.

```
70      def baseline_corrected(data: np.ndarray, fs: int, chan: int):
71          all_specs = []
72          for trial in data:
73              f, t, Sxx = spectrogram(trial[:, chan] - trial[:, chan].mean(),
74                                      fs=fs, nperseg=fs // 2)
75              baseline = Sxx[:, t < 1.0].mean(axis=1, keepdims=True)
76              all_specs.append(10 * np.log10(Sxx / (baseline + 1e-10)))
77      return np.mean(all_specs, axis=0), f, t
```

קובץ לשאלה 3 : features.pyчисוב עוצמת הספקטרום (שורות 20-23) :

יצירת הפונקציה () \_band\_power() לחישוב עוצמת הספקטרום בתחום תדרים המוגדר מראש באמצעות Welch על ידי סיכום הערכים בתחום הרלוונטיים והמרתם ליחידות דציבילים [dB].

```
20     def _band_power(sig: np.ndarray, fs: int, f_low: float, f_high: float) -> float:
21         f, Pxx = welch(sig, fs=fs, nperseg=fs)
22         band = Pxx[(f >= f_low) & (f <= f_high)].sum()
23         return 10 * np.log10(band + 1e-12)
```

чисוב עוצמת הספקטרום (שורות 29-32) :

יצירת הפונקציה () \_ensure\_trial\_first() לוודוא גודל תקין של המטריצות וביצוע transpose \_ensure\_trial\_first() במידת הצורך יחד עם החזרת שגיאה במידה והמבנה לא תקין.

```
29     def _ensure_trial_first(labels: np.ndarray) -> np.ndarray:
30         if labels.ndim != 2 or 4 not in labels.shape:
31             raise ValueError("labels must be 2-D and include 4 columns (ARTIFACT, REMOVE, LEFT, RIGHT)")
32         return labels.T if labels.shape[0] == 4 else labels
```

чисוב מאפיינים (שורות 38-53) :

יצירת הפונקציה () extract\_band\_power\_features() לחישוב מאפייני band – power לשולשות התחומיים שהוגדרו מראש עבור כל ניסוי רלוונטי ולכל אחד מהعروצים. שמירת הניסויים של יד ימין ויד שמאל ובעורם חישוב הספק בתחום תדרים מוגדר לאחר הסרת הממווצע מכל עroz. הפונקציה מחזירה מטריצת פיצרים, וקטור תוויות מתאים ורשימת שמות הפיצרים אשר ישמשו להמשך הסיווג.

```
38     def extract_band_power_features(data: np.ndarray, labels: np.ndarray, fs: int) -> Tuple[np.ndarray, np.ndarray, List[str]]:
39         labels = _ensure_trial_first(labels)
40         sel_idx = (labels[:, 2] == 1) | (labels[:, 3] == 1)
41         y = np.where(labels[sel_idx, 3] == 1, 1, 0)
42
43         X = []
44         for trial in data[sel_idx]:
45             feats = []
46             for t0, t1, ch, f_low, f_high, _ in BAND_DEFS:
47                 s0, s1 = int(t0 * fs), int(t1 * fs)
48                 sig = trial[s0:s1, ch] - trial[s0:s1, ch].mean()
49                 feats.append(_band_power(sig, fs, f_low, f_high))
50             X.append(feats)
51
52         feature_names = [b[-1] for b in BAND_DEFS]
53         return np.array(X), y, feature_names
```

ספרת חציות האפס (שורות 59-62) :

יצירת הפונקציה () zero\_cross\_rate() לחישוב שיעור חציות האפס באוט. ספירת כמהות שינוי הסימן בדגימה לעומת הדגימה הקודמת. חלוקה במספר הדגימות ובחזרת שיעור החציות ביחס לאורך האוט.

212366702 אילונה בר שדה

315262683 שירה אלטרץ

```
59     def zero_cross_rate(sig: np.ndarray) -> float:
60         """Zero-crossing rate per sample (fraction)."""
61         crossings = np.where(np.diff(np.signbit(sig)))[0]
62         return len(crossings) / (len(sig) - 1 + 1e-12)
```

### ZCR ויחס שוניות לוגריתמי (שורות 38-53)

ביצירת הפקנץיה () extract\_additional\_features לחישוב שני מאפייני זמן בכל ניסוי (קצב ח齐ות האפס ZCR ויחס השוניות הלוגריתמי Log Variance Ratio) בין העורכים C3 ו-C4. סינוון הניסויים לפיקו דמיון התנוועה, לקיחת חלון זמן בהתאם לניתוח נתוני הספקטוגרמה (6-שניות), הסרת הממוצע מכל ערוץ, חישוב קצב ח齊ות האפס ב- C3 ויחס השוניות בין C3 ל- C4 תוך הרמת הערכיםelogarithmic. הפקנץיה מחזירה את מטריצת הפיצ'רים ושמותיהם.

```
65     def extract_additional_features(data: np.ndarray, labels: np.ndarray, fs: int) -> Tuple[np.ndarray, List[str]]:
66         labels = _ensure_trial_first(labels)
67         sel_idx = (labels[:, 2] == 1) | (labels[:, 3] == 1)
68
69         feats_all = []
70         for trial in data[sel_idx]:
71             s0, s1 = int(4.0 * fs), int(6.0 * fs)
72             sig_c3 = trial[s0:s1, 0] - trial[s0:s1, 0].mean()
73             sig_c4 = trial[s0:s1, 1] - trial[s0:s1, 1].mean()
74
75             # Feature A: Zero-crossing rate in C3
76             zcr_c3 = zero_cross_rate(sig_c3)
77
78             # Feature B: log variance ratio C3/C4
79             var_ratio = np.var(sig_c3) / (np.var(sig_c4) + 1e-12)
80             feats_all.append([zcr_c3, np.log(var_ratio + 1e-12)])
81
82         return np.array(feats_all), ["ZCR_C3", "logVarRatio_C3/C4"]
```

### הורדת מדדים עם PCA (שורות 105-107)

ביצירת הפקנץיה () pca\_visualization שימוש בחלוקת PCA כדי להוריד את הפיצ'רים לשולשה מדדים ובכך לבדוק את ביצירת הפיצ'רים.

```
105     def pca_visualization(X: np.ndarray, y: np.ndarray, scaled: bool = False):
106         X_proc = StandardScaler().fit_transform(X) if scaled else X
107         comps  = PCA(n_components=3).fit_transform(X_proc)
```

קובץ classification.py לשאלה 4 :הערכת פונקציה LDA (שורות 28-57)

יצירת הפונקציה evaluate\_lda\_crossval() אשר מבצעת הערכה מסוג LDA באמצעות שיטת K-fold cross-validation כך שבירית המודל היא חלוקה ל-5 קייפולים. בכל איטרציה, אחת מהקבוצות משמשת כט הולידציה והיתר כט האימון. דיק הולידציה מחושב באמצעות cross\_val\_score() בעוד שבדיקה האימון מחושב ידנית על ידי התאמת המודל על קבוצת האימון ובדיקתו עלייה. הפונקציה מחזירה את תוצאות הדיק בכל קייפול בשלב האימון ולשלב הולידציה.

```

28     def evaluate_lda_crossval(
29         X: np.ndarray,
30         y: np.ndarray,
31         k: int = 5,
32         random_state: int = 42,
33     ):
34         print("\n---- LDA Classification with k-fold Cross Validation ---")
35         print(f"Data shape: X = {X.shape}, y = {y.shape}")
36         print(f"Using k = {k} folds")
37
38         lda = LinearDiscriminantAnalysis()
39         kf = KFold(n_splits=k, shuffle=True, random_state=random_state)
40
41         val_scores = cross_val_score(lda, X, y, cv=kf, scoring="accuracy")
42         print(
43             f"Validation Accuracy: {val_scores.mean()*100:.2f}% ± {val_scores.std()*100:.2f}%""
44         )
45
46         train_scores = []
47         for train_idx, _ in kf.split(X):
48             X_train, y_train = X[train_idx], y[train_idx]
49             lda.fit(X_train, y_train)
50             y_train_pred = lda.predict(X_train)
51             train_scores.append(accuracy_score(y_train, y_train_pred))
52
53         train_scores = np.asarray(train_scores)
54         print(
55             f"Train Accuracy : {train_scores.mean()*100:.2f}% ± {train_scores.std()*100:.2f}%""
56         )
57         return val_scores, train_scores
58

```

הערכת נוספת (בונוס) (שורות 199-241)

יצירת הפונקציה evaluate\_classifier\_crossval() המבצעת הערכת ביצועים גנריית עבור כל מסוג מושס sklearn באמצעות שיטת K-fold cross-validation. הפונקציה מקבלת את המסוג כפרמטר (clf), יוצרת אובייקט Kfold שמחולק את הנתונים ל-k קבוצות לצורך אימון ובדיקה ומבצעת חישוב של דיק הולידציה באמצעות cross\_val\_score(). בonus, החישוב מתבצע באופן ידני עבור קבוצת האימון גם כן. הפונקציה מחזירה את ציוני הדיק לכל קייפול בשלב האימון ולשלב הולידציה ומדפיסה את שם המסוג הנבחן.

212366702 אילונה בר שדה

315262683 שירה אלטרץ

```
199     def evaluate_classifier_crossval(
200         X: np.ndarray,
201         y: np.ndarray,
202         clf,
203         k: int = 5,
204         random_state: int = 42,
205         name: str | None = None,
206     ):
207         """
208             Generic K-fold evaluation tool for *any* sklearn classifier.
209
210             Parameters
211             -----
212             clf : sklearn.base.BaseEstimator
213                 A classifier object (e.g., LogisticRegression()).
214             name : str
215                 Name to print; default is the class name.
216             """
217             name = name or clf.__class__.__name__
218             print(f"\n--- {name} with {k}-fold Cross Validation ---")
219             print(f"Data shape: X = {X.shape}, y = {y.shape}")
220
221             kf = KFold(n_splits=k, shuffle=True, random_state=random_state)
222
223             # Validation
224             val_scores = cross_val_score(clf, X, y, cv=kf, scoring="accuracy")
225             print(
226                 f"Validation Accuracy: {val_scores.mean():.2f}% ± "
227                 f"{val_scores.std():.2f}%"
228             )
229
230             # Training
231             train_scores = []
232             for train_idx, _ in kf.split(X):
233                 X_tr, y_tr = X[train_idx], y[train_idx]
234                 clf.fit(X_tr, y_tr)
235                 train_scores.append(accuracy_score(y_tr, clf.predict(X_tr)))
236             train_scores = np.asarray(train_scores)
237             print(
238                 f"Train Accuracy : {train_scores.mean():.2f}% ± "
239                 f"{train_scores.std():.2f}%"
240             )
241             return val_scores, train_scores
```

### הגדרת מסוווג (שורות 28-57) Logistic Regression

יצירת הפונקציה() אשר יוצרת את המסווג evaluate\_logreg\_crossval() עם מספר איטרציות מקסימלי של 1000 ומעבירה אותו יחד עם הדאטה לפונקציית ההערכה evaluate\_classifier\_crossval().

```
247     def evaluate_logreg_crossval(
248         X: np.ndarray,
249         y: np.ndarray,
250         k: int = 5,
251     ):
252         """K-fold CV for Logistic Regression (linear)."""
253         logreg = LogisticRegression(
254             penalty="l2",
255             solver="liblinear", # Suitable for small datasets
256             max_iter=1000,
257         )
258         return evaluate_classifier_crossval(X, y, logreg, k)
```

סיווג מבוסס PCA עם LDA (שורות 64-72) :

יצירת הפונקציה `evaluate_with_pca()` המבצעת סיווג באמצעות LDA לאחר מצום ממדים בשיטת PCA. נרמול הנתונים בעזרת StandardScaler ולאחר מכן ביצוע חישוב הרכיבים הראשיים באמצעות PCA תוך שמירה על מספר רכיבים מוגדר מראש עם ברירת מחדל של 3. התוצאה היא ייצוג מצומצם של הדאטה אשר משמר את רוב השונות. הערכה של הנתונים המוצמצמים באמצעות `evaluate_lda_crossval()` בשיטת `K-folds`.

```

64     X: np.ndarray,
65     y: np.ndarray,
66     n_components: int = 3,
67     k: int = 5,
68   ):
69     print(f"\n--- PCA-based Classification with top {n_components} components ---")
70     X_scaled = StandardScaler().fit_transform(X)
71     X_pca = PCA(n_components=n_components).fit_transform(X_scaled)
72     return evaluate_lda_crossval(X_pca, y, k=k)

```

בחירה תכונות אופטימליות (שורות 136-147) :

יצירת הפונקציה `evaluate_with_rfe()` כך שתתבצע בחירת פיצ'רים בשיטת RFE (Recursive Feature Elimination) עבור מסוג מסווג LDA. עבור כל מספר אפשרי של פיצ'רים (עד למינימום שנקבע), הפונקציה מחשבת את דיק הולידציה באמצעות התאמת מודל עם אותו מספר פיצ'רים נבחרים. לאחר סריקת כלל האפשרויות נבחרת תת-קבוצה אשר הניבה את הדיק הגבוה ביותר ומוצגים שמות הפיצ'רים שנבחרו. ביצוע הרצת סיווג נוספת עם שלושת הפיצ'רים הטובים ביותר והשוואה נפרדת עם ארבעה פיצ'רים, לצורך בחינה נוספת של השפעת מספר הפיצ'רים על הביצועים.

```

78     def evaluate_with_rfe(
79       X: np.ndarray,
80       y: np.ndarray,
81       feature_names: list[str],
82       max_features: int | None = None,
83       k: int = 5,
84     ):
85       print("\n--- RFE-based Feature Selection ---")
86       if max_features is None:
87         max_features = X.shape[1]
88
89       scores: list[float] = []
90       features = list(range(1, max_features + 1))
91       best_selector: RFE | None = None
92
93       for n_feat in features:
94         lda = LinearDiscriminantAnalysis()
95         selector = RFE(lda, n_features_to_select=n_feat)
96         X_rfe = selector.fit_transform(X, y)
97         val_scores = cross_val_score(
98           lda, X_rfe, y, cv=KFold(n_splits=k, shuffle=True, random_state=42)
99         )
100        scores.append(val_scores.mean())
101        if n_feat == np.argmax(scores) + 1:
102          best_selector = selector

```

ailona bar shuda 2

315262683 שירה אלטרץ

```
104     plt.figure(figsize=(6, 4))
105     plt.plot(features, scores, marker="o")
106     plt.title("Validation Accuracy vs. Number of Selected Features (RFE)")
107     plt.xlabel("Number of Features")
108     plt.ylabel("Validation Accuracy")
109     plt.grid(True)
110     plt.tight_layout()
111     plt.show()
112
113     best_n = np.argmax(scores) + 1
114     print(
115         f"Best number of features: {best_n} "
116         f"with accuracy: {scores[best_n - 1] * 100:.2f}%"
117     )
118
119     if best_selector is not None:
120         mask = best_selector.support_
121         selected = [n for n, keep in zip(feature_names, mask) if keep]
122         print("Selected features:", selected)
123
124         selected_idx = [i for i, n in enumerate(feature_names) if n in selected]
125         X_best = X[:, selected_idx]
126         print("\n[4.2c] LDA with only top 3 selected features:")
127         evaluate_lda_crossval(X_best, y, k=k)
128
129         if len(selected_idx) >= 3:
130             lda_full = LinearDiscriminantAnalysis()
131             rfe4 = RFE(lda_full, n_features_to_select=4)
132             X_rfe4 = rfe4.fit_transform(X, y)
133             print("\n[4.2d] LDA with top 4 features (for comparison):")
134             evaluate_lda_crossval(X_rfe4, y, k=k)
135
136     return best_n, scores
```

#### סיווג עם LDA לאחר נרמול (שורות 142-149)

יצירת הפקציה `evaluate_on_normalized_features()` המבצעת הערכת סיווג באמצעות LDA לאחר נרמול הפיצירים בשיטת Z-score. לאחר הנרמול, הערתת הנתונים לפונקציית ההערכתה (`RFE`) בפונקציה זו יש שימוש בנרמול בלבד, ללא הפקחת ממד או סינון תכונות.

```
142     def evaluate_on_normalized_features(
143         X: np.ndarray,
144         y: np.ndarray,
145         k: int = 5,
146     ):
147         print("\n[4.2e] LDA on Z-score Normalized Features (no PCA or RFE):")
148         X_scaled = StandardScaler().fit_transform(X)
149         evaluate_lda_crossval(X_scaled, y, k=k)
```

#### סיווג סופי על סט הבדיקה (שורות 142-149)

יצירת הפקציה `evaluate_final_classifier_on_test()` המבצעת אימון סופי של מסובג LDA על כל נתוני האימון והערכתה ביצועים על סט הבדיקה. חילוץ פיצירים מ-`train` באמצעות `build_all_features()`, בחרית שלושה פיצירים על ידי RFE, נרמול באמצעות Z-score ואימון המסובג על נתוני האימון. ביצוע RFE גם לסט הבדיקה ונרמול באמצעות האובייקטים שאומנו על סט האימון, ללא התאמת מחדש. ביצוע על סט הבדיקה, חישוב דיווק ושמירת קובץ תוצאות.

212366702 אילונה בר שדה

315262683 שירה אלטרץ

```
155     def evaluate_final_classifier_on_test(
156         train_data,
157         train_labels,
158         test_data,
159         test_labels,
160         fs,
161         feature_names,
162     ):
163         # ❶ Full feature extraction – Train
164         X_train, y_train, _ = build_all_features(train_data, train_labels, fs)
165
166         # ❷ RFE for selecting 3 features
167         lda = LinearDiscriminantAnalysis()
168         rfe = RFE(lda, n_features_to_select=3).fit(X_train, y_train)
169         X_train_sel = rfe.transform(X_train)
170
171         # ❸ Z-score normalization
172         scaler = StandardScaler().fit(X_train_sel)
173         X_train_scaled = scaler.transform(X_train_sel)
174
175         # ❹ Final LDA training
176         clf = LinearDiscriminantAnalysis().fit(X_train_scaled, y_train)
177
178         # ❺ Test feature extraction
179         X_test, y_test, _ = build_all_features(test_data, test_labels, fs)
180         X_test_sel = rfe.transform(X_test)
181         X_test_scaled = scaler.transform(X_test_sel)
182
183         # ❻ Prediction + accuracy
184         y_pred = clf.predict(X_test_scaled)
185         acc = accuracy_score(y_test, y_pred)
186         print(f"Test Accuracy: {acc * 100:.2f}%")
187
188         np.save("test_predictions.npy", y_pred)
189         print("Saved predictions to test_predictions.npy")
```

212366702 אילונה בר שדה

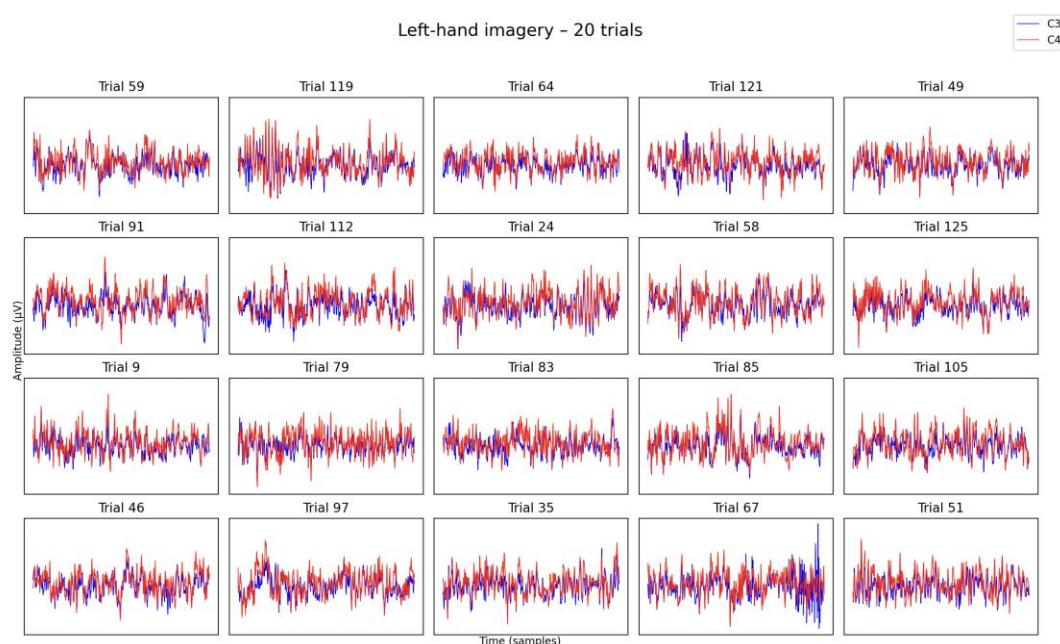
315262683 שירה אלטרץ

### תוצאות:

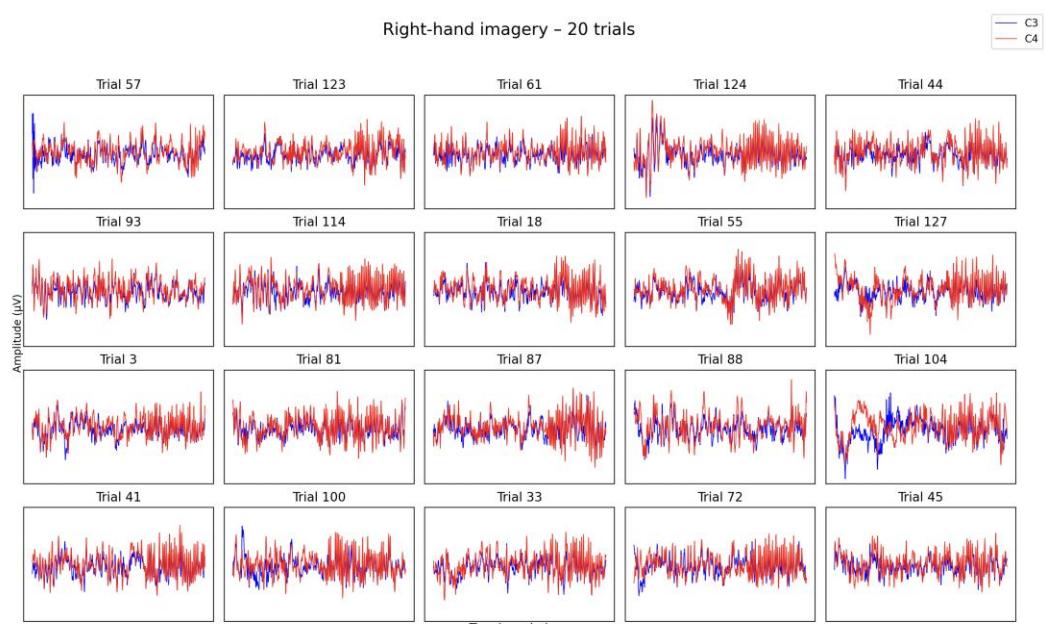
#### מוצגים 1,2

20 חזרות של תנאי דמיון תנועה של יד שמאל (מוצג 1) ויד ימין (מוצג 2). הצגתאות ה-EEG מהערוצים C3 (כחול) ו- C4 (אדום) בכל אחד מהגרפים.

#### מוצג 1



#### מוצג 2

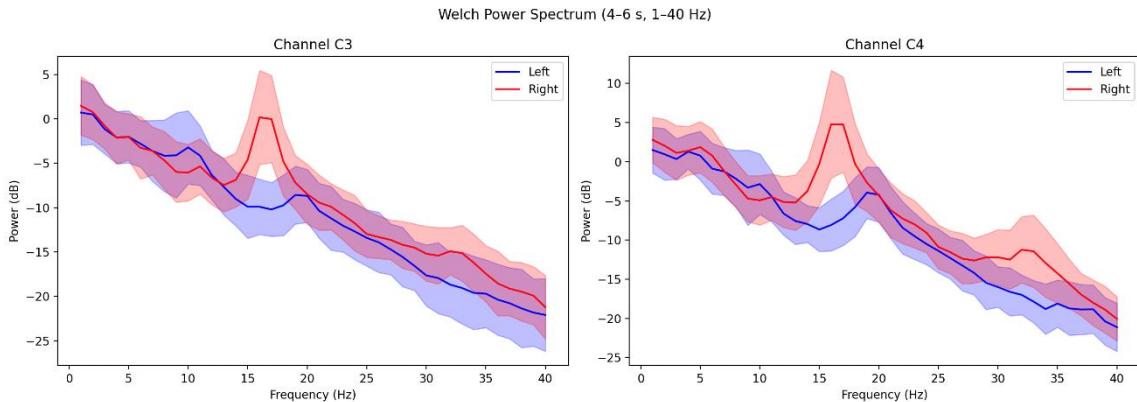


212366702 אילונה בר שדה

315262683 שירה אלטרץ

### מוצג 3

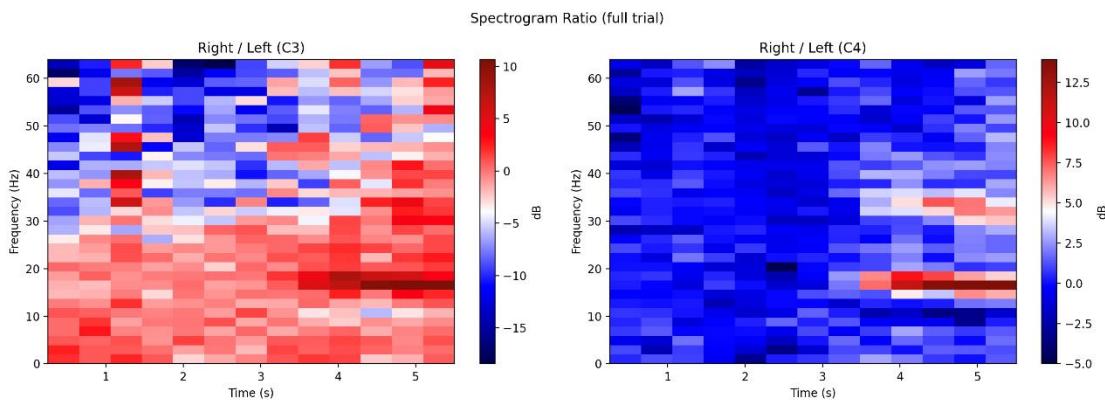
בגרף זה מוצגת עוצמת ההספק (Power) כפונקציה של תדר, כפי שוחושה בשיטת Welch עבור העורוצים C3 ו- C4. הקווים מייצגים את הממוצע עבור תנאי הדמיון של יד שמאל (כחול) ויד ימין (אדום) כשהשיטה הצבוע מייצג את סטיית התקן.



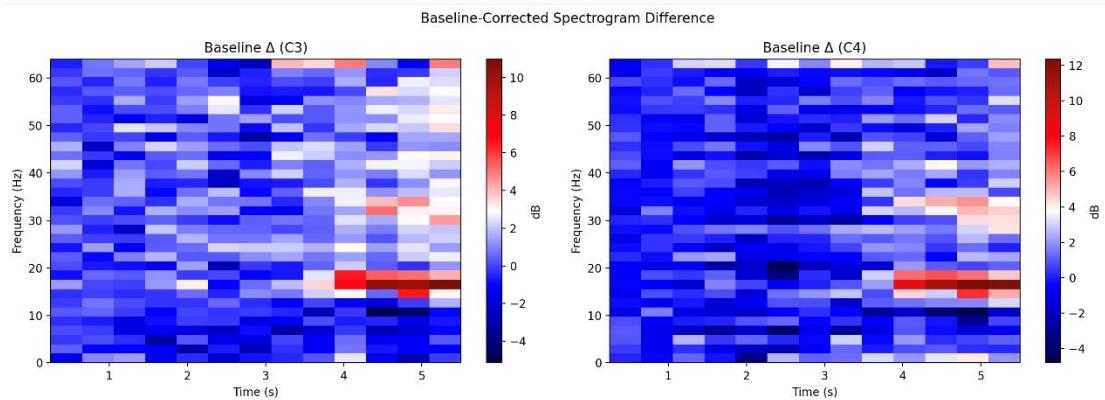
### מוצג 4,5

בגרפים אלו מוצג היחס בין ספקטוגרמות של תנאי יד ימין לתנאי יד שמאל (Right/Left) עבור העורוצים C3 ו- C4 לאורך זמן ותדר. הערכים מוצגים בדציבלים כך שצבעים חמים (אדומים) מציינים עוצמה גבוהה יותר ביד ימין לעומת יד שמאל וצבעים קרים (כחול) להפך. מוצג 4 מציג את היחס הכללי, בעוד מוצג 5 מציג יחס לאחר ניקוי קו בסיס (baseline).

### מוצג 4



### מוצג 5



מוצגים 8-6

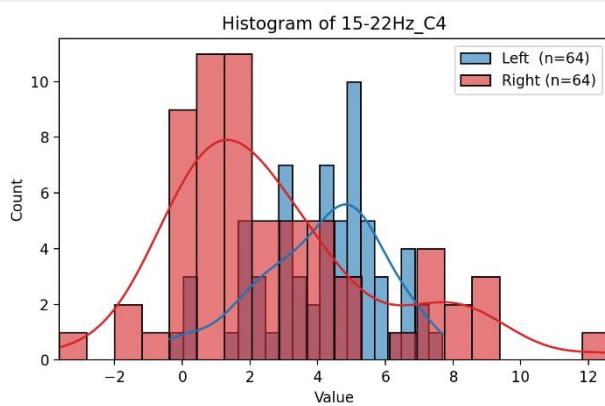
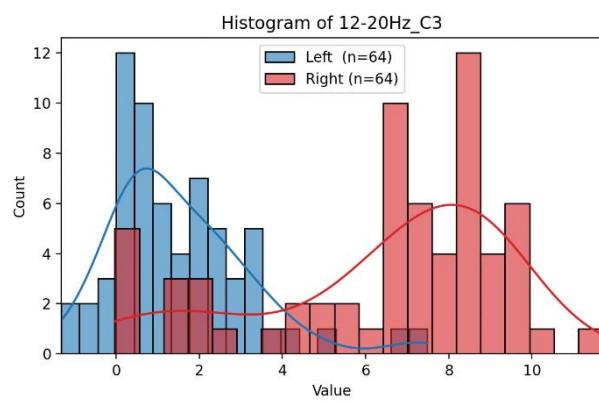
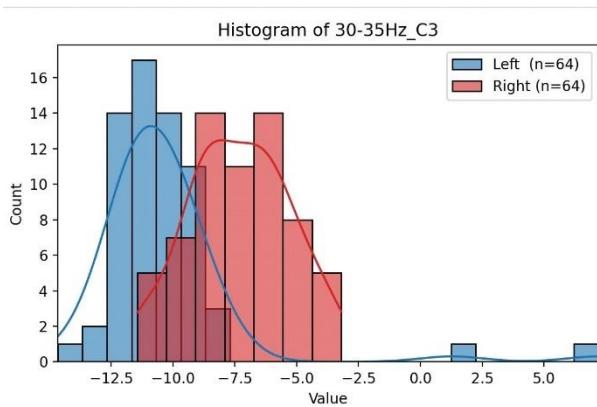
היסטוגרמות אלו מציגות את התפלגות ערכי ה-band-power (Power Spectrum) בעורצים שונים עבור תדרים שנבחרו לשמש כפיצ'רים בתהיליך הסיווג על סמך ניתוח ספקטורי מוקדם (スペクトルモード) וספקטוגרמות (スペクトログラム) שבו נצפו הבדלים מובהקים בין התנאים. ישנה הפרדה בין התפלגות עבור יד שמאל (כחול) ויד ימין (אדום).

מוצג 7

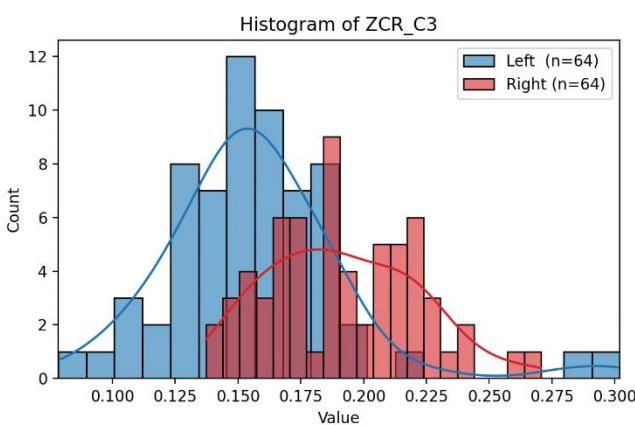
ערוץ C3 בתחום התדרים 30 – 35 Hz וזמן [6 – 4] וזמן [Hz]

מוצג 6

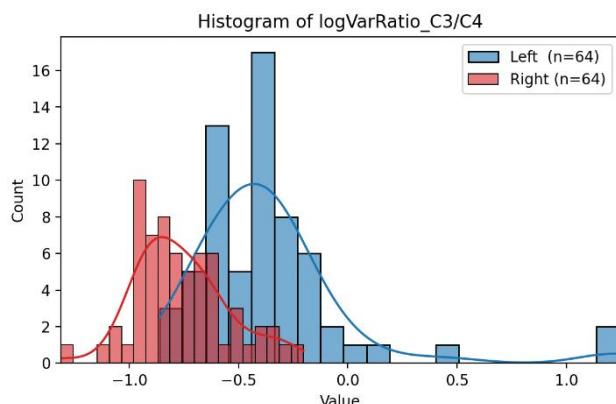
ערוץ C3 בתחום התדרים 12 – 20 Hz וזמן [6 – 4] וזמן [Hz]

מוצג 8

ערוץ C4 בתחום התדרים 15 – 22 Hz וזמן .1.8 – 4.1[s]

מוצג 9

היסטוגרמה המציגת את התפלגות ערכי הפיצ'ר ZCR (Zero Crossing Rate) שחושב עבור ערוץ C3 תוך השוואת בין תנאי הדמיון של יד שמאל (כחול) ויד ימין (אדום). ציר ה-X מתאר את השיעור בו האות חזה את קו האפס וציר ה-Y מראה כמה ניסיונות (trails) קיבל כל ערך.

**מוצג 10-**

ההיסטוגרמה המציגה את התפלגות ערכי הפיצ'ר log-variance ratio בין העורכים C3 ו- C4 עברו תנאי הדמיון של יד שמאל (כחול) ויד ימין (אדום). ציר ה- X מייצג את ערך הפיצ'ר וציר Y מייצג את מספר הניסיונות (trails) עברו כל ערך.

**מוצגים 11-14-**

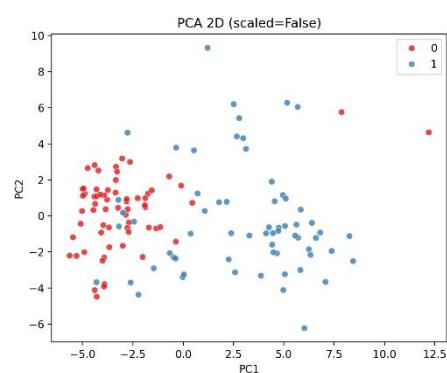
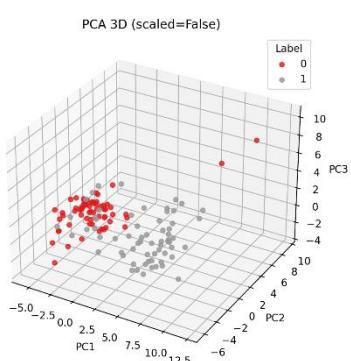
הגרפים הבאים מציגים את הדאטה לאחר ביצוע PCA עם הורדת הפיצ'רים לשולש ממדים – PC3 ו- PC1, PC2 (כך שגם מציגים את מרבית השונות). גרפים המציגים את המידע בצורה זו – מדידת (הקרנה על PC1 ו- PC2 או בצורה תלת – מדידת PC1, PC2 ו- PC3) עם צבע התווית ('0' = שמאל, '1' = ימין).

**מוצג 12-**

פייזור תלת – מדדי על הצירים PC1, PC2 ו- PC3, ללא נרמול.

**מוצג 11-**

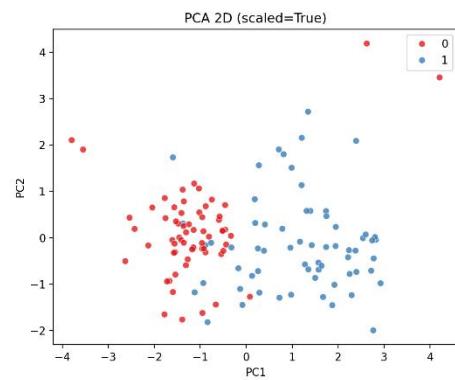
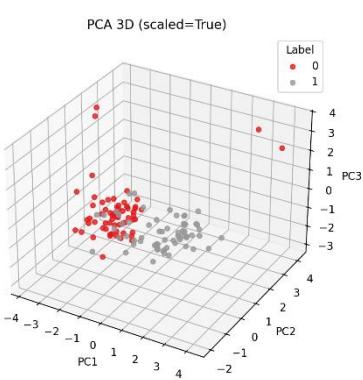
פייזור הנתונים על ציריו PC1 ו- PC2, ללא נרמול מקדים של הפיצ'רים.

**מוצג 14-**

הקרנת הדאטה בתלת – מדד על PC1, PC2 ו- PC3 לאחר נרמול.

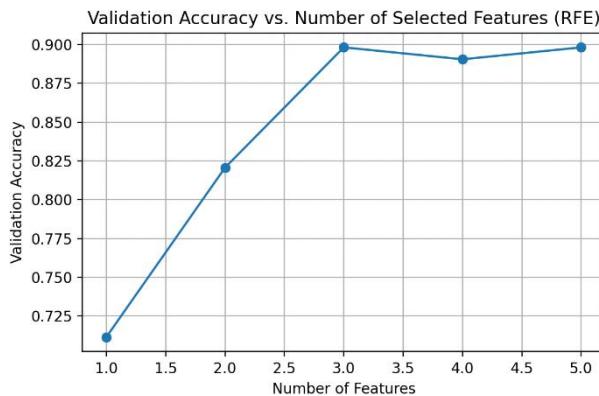
**מוצג 13-**

הקרנת הדאטה על ציריו PC1 ו- PC2 לאחר נרמול Z-score על PC2.



**מוצג-15**

הגרף מציג את הקשר בין מספר הפיצ'רים שנבחרו לבין דיק הווילידציה של המודל כפי שנמדד בשיטת (RFE) (Recursive Feature Elimination). ציר-X מייצג את מספר הפיצ'רים שנבחרו, ציר-Y מייצג את הווילידציה וכל נקודה מייצגת את הדיק שהתקבל עבור מספר מסוים של פיצ'רים.

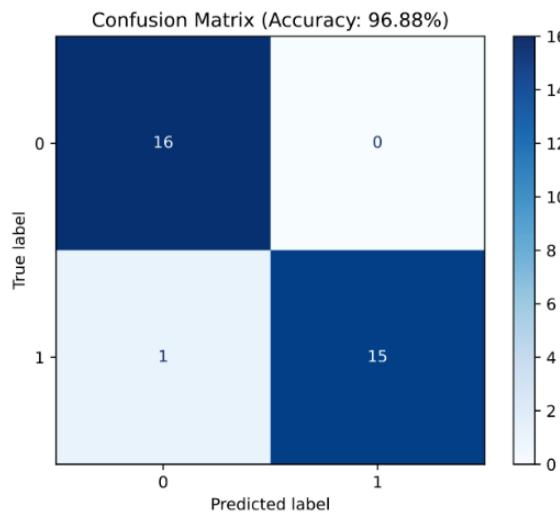
**מוצג-16**

במוצג זה ניתן לראות את תוצאות הסיווג שהתקבלו ממספר מודלים וגישות שונות, תוך השוואת ביצועים לפי דיק ממוצע וסטיית תקן. עבור כל שיטה מוצגים דיקי אמן ווילידציה שנמדד באמצעות קבוצות פיצ'רים שונות. על בסיס K-fold cross-validation.

LDA Model after normalizing features	LDA Model with four features selected	LDA Model with three features selected	LDA Model with two features selected	LDA Model after PCA	Logistic Regression Model	LDA Model	
X = (128,5) Y = (128)	X = (128,4) Y = (128)	X = (128,3) Y = (128)	X = (128,2) Y = (128)	X = (128,5) Y = (128)	X = (128,5) Y = (128)	X = (128,5) Y = (128)	Data Shape
5	5	5	5	5	5	5	Using K
89.82% ± 6.34%	89.05% ± 5.72%	89.82% ± 4.73%	89.82% ± 4.73%	89.78% ± 4.13%	89.82% ± 6.34%	89.82% ± 6.34%	Validation Accuracy
90.42% ± 1.92%	90.43% ± 1.46%	90.82% ± 1.19%	89.26% ± 1.24%	90.04% ± 1.16%	90.42 ± 1.92%	90.42 ± 1.92%	Train Accuracy

מוצג-17

במוצג זה ניתן לראות את מטריצת Confusion Matrix המתקבלת לאחר הפעלת המודל על סט הבדיקה שכלל 32 דוגימות. המספרים מייצגים את כמות הסיווגים הנכונים והשגוויים לפי תוויות אמת ותחזית כך שהדיק ה כולל עמד על 96.88% (16 מתוך 32 הדוגימות סוגגו נכונה עם טעות אחת בלבד מסוג False Negative).



**דיוון:**

מטרתו של תרגיל זה הינה סיוג תנועות דמיוניות של יד ימין ויד שמאל מתוך אוטות EEG אשר נמדדו משני עروצים (C3 ו-C4) באמצעות אלגוריתם של למידת מכונה, ככלומר, לזהות האם נבדק דמיין תנועת יד ימין או יד שמאל רק מתוך אחת המוח החשמלי ובאופן אוטומטי (ממוחשב) בהתקבש על ניסיון בודד. במהלך הדיוון יובא פירוט על שלבי תהליכי העבודה ועל התוצאות שהוצעו בחלק הקודם.

תחילה הzbתעה הכרה ויזואלית עם האותות עצם באמצעות אמצעות ויזואליות של ניסוי EEG גולמיים. תוצאות אלו (موظגים 1 ו-2) מציגים 20 ניסיונות אקראיים לכל תנאי (דמיון תנועת יד שמאל או ימין) בשני הערווצים C3 ו- C4 לאורך זמן המדידה. ניתן לראות שהנתונים רועשים ומכך קשה להסיק תובנות ברורות מהתבוננות חזותית בהם, אך בשליש האחרון של מספר ניסויים ביד ימין ניכרת פעילות מוגברת בערוץ C4, דבר אשר עשוי להעיד על הבדל רלוונטי לצורך הסיוג.

מהפקת גדר ספקטורים העוצמה, **Power Spectral Density**, (موظג 3) שחוسب על ידי שיטת Welch, בו נראים ממוצעי תנאי הדמיון של יד שמאל וימין בערווצים השונים, ניתן לראות כי קיימת הפרדה ברורה בין יד ימין ליד שמאל בעיקר בטוחה התדרים של [Hz] 20 – 15 בזמינים 4 – 6 שנלקחו מנתוח נתוני ההכרה הוויזואלית. מכאן שתדרים אלו עשויים להיות אינפורטטיביים להמשך הסיוג. הבדלים בין התנאים בטוחה התדרים המוזכר לעיל בעליים בקנה אחד עם ממצאים מהספרות, המעידים על ירידה בעוצמת גלי בטא β ([Hz] 20 – 15) באזוריים מוטוריים במהלך דמיון תנועה, תופעה הידועה כ- **Event-Related Desynchronization (ERD)**. תופעה זו משלקפת ירידה בסביבון העצמי ונחשבת למדד נוירופיזיולוגי מהימן לפעולה מוטורית ממשית או מודומינית ומכאן כי היא תואמת את הנתונים ממטה זו.

בהמשך נבנו ספקטוגרמות הממחישות את הבדלים בין תנאי דמיון יד ימין לבין דמיון יד שמאל כפונקציה של זמן ותדר. שלא כמו ב- PSD, בספקטוגרמות אלו נעשו שימוש באות המלה מאחר ששיטה זו מאפשרת זיהוי שינויים גם לאורך הזמן (כולל תחילת הפעולות המוטוריות והסתגלות). הספקטוגרמות (موظג 4) חושבו באמצעות לוגריתם שליחס ההספק בין תנאי יד ימין ליד שמאל, כך שערכים חיוביים מצביעים על עוצמה גבוהה יותר ביד ימין ולהפך, דבר אשר אפשר זיהוי ברור של אזורים מבדילים בין התנאים. הממצאים מצביעים על הבדל מובהק בטוחה התדרים [Hz] 20 – 12 ובטוחה זמינים של 6 – 4 שניות ומכאן שפערים אלו יכולים לשמש לאפיון תכונות אפקטיביות לצורך סיוג. כדי לאשש ממצאים אלו בוצע נרמול של הספקטוגרמות (موظג 5) לפי השניה הראשונה בניסוי המוגדרת כ- baseline. הנרמול סייע בהבהת אזורים מבדילים ביו התנאים, לפיו ניתן לראות כי קיים שינוי נוסף באוט של C3 גם בין [Hz] 35 – 30 בתווים הזמינים (6 – 4 שניות) ומכאן שיש לנרמול משמעות חשובה בבחירה התכונות לשיווג. ההבדלים הנראים בטוחה התדרים מתאימים גם במקרה זה לתופעת ה- ERD, שכן מדובר על אותן טווחי התדרים של גלי הבטא β וזהו עוד אינדיקציה לכך שינוי חשיבות לבחירת התכונות בהקשר לתופעה זו.

לשם בניית מערכת תכונות אפקטיבי לשיווג תנאי דמיון יד ימין מול יד שמאל, נבחרו חמישה פיצ'רים מרכזיים (موظגים 6-10). שלושת הפיצ'רים הראשונים מtabססים על **ממוצע עוצמה**

**ספקטROLית (PSD)** בטוחי תדרים וזמן אשר נמצא אינפורטטיבים מתוך ניתוח גרעין הספקטロום והספקטוגרמות שכן בטוחים אלו קיימים הבדלים משמעותיים בין האותות: הראשון מעורץ C3 בטוחי תדר [Hz] 20 – 12 – 6 – 4 (מושג 6), השני מעורץ העורץ בטוחי תדר [Hz] 35 – 30 – 6 – 4 (מושג 7) והשלישי מעורץ C4 בטוחי תדר [Hz] 22 – 15 – 6 – 4 (מושג 8). **הפייצ'ר הריבי עלי, ZCR (Zero Crossing Rate)**, אשר מוגדר כתכונת אחת בולטת המשקפת את קצב המעברים של האות דרכן חציית האפס<sup>iii</sup> והוא משקף את רמת התנודתיות וקצב השינוי של האות בעורץ C3 לאורץ הזומו. הוא נבחר מתוך הנחה כי הוא מוסיף מידע נוסף למערכת הפעילה, כיון שכן ניתן לבצע מעקב על רמת התנודתיות של האות המוחי אשר עשוי להשנות בהתאם לרמת הפעולות העצביות באזורי המוטורי המשופעת מסוג הדמיון המוטורי, בכך היא יכולה לספק אינדיקציה שונה בין התאים (מושג 9). לבסוף, נבחר הפיצ'ר החמייש (מושג 10) המתאר את היחס הלוגריטמי בין **שונות האות ב- C3 לעומת C4** (שיטת בה ייחסו שונות ערוצים לשמש להבנה בין מצביו מוח<sup>iv</sup>) ומשווה בין שני צידי המוח, מכך ניתן לקבל מידע על פעילות או שינוי בהם. היחס הלוגריטמי משקף את האסימטריה בפעולות החשמלית בין שני חצאי המוח אשר עלולה להופיע במצב דמיון תנועה. מכיוון ש- C3 ו- C4 ממוקמים מעל ההmisפרות המוטוריות השמאלית והימנית בהתאם, דמיון תנועה יד ימין או שמאל עשוי להפעיל את אחד העורוצים יותר מהשני ובכך להוות אינדיקציה טובה לסייע התנאים.

לכל אחד מהפייצ'רים שתואר לעיל מצורפת היסטוגרמה המתאימה לו (מושגים 10 – 6) כך שנitinן לראות בהן הפרוזות בין תנועת יד ימין ותנועת יד שמאל ומכאן להסיק כי הם מהווים פיצ'רים המתארים בצורה טובה את הסיווג, בעיקר בפייצ'ר הראשון, הריבי ווחמייש.

**PCA (Principal Component Analysis)** לאחר מכן בוצעה הפחתת ממדים באמצעות שיטת (PCA) אשר נועדה לצמצם את מספר הפיצ'רים תוך שימירה על מרבית השונות במידע. מתוך חמישת הפיצ'רים שנבחרו קודם لكن בוצעה הורדה לשולשה ממדים. בצורה זו הרכיבים החדשניים שנוצרו (PC1, PC2, PC3) מהווים צירופים ליניאריים של הפיצ'רים המקוריים ובכך מתאפשרת הדמייה נוחה יותר של מבנה הנתונים. התוצאות הוצגו בגרף דו – ממדי (מושג 11), אשר מtabסס על שני הרכיבים הראשיים עם מרבית השונות ובגרף שלישי – ממדדי (מושג 12) אשר מראה את הייצוג לפי שלושת הרכיבים. ניתן לראות שבשני המקרים קיימת הפרדה כמעט מלאה בין הקבוצות, דבר המצביע על כך **שהפייצ'רים שנבחרו אכן נושאים מידע רלוונטי לסיווג בין התנאים** (כלומר, בין יד ימין ליד שמאל), וכן כי רוב השונות מיוצגת על ידי שני הרכיבים הראשונים (PC1 ו- PC2, דבר אשר יהיה רלוונטי להמשך הסיווג). בשלב לאחר מכן בוצע **נרטול פיצ'רים** שהיו בעלי אותן ייחודות מידה, דבר אשר אפשר להשוות בינהם בצורה הוגנת כיון שהוא ביטל הבדלים בגודל וביחידות בין הפיצ'רים השונים. מושגים אלו (מושגים 14-13) ממחישים כי גם במצב בו כל הפיצ'רים בעלי משקל שווה, **קיימת מגמת הפרדה כמעט מלאה בין התנאים** ובכך ישנו חיזוק לתרומה המשולבת של כלל הפיצ'רים לבניה הנתונים ועל כך PCA עם הורדה ל-3 ממדים (ואף שניים, כיון שרוב השונות נשמרת בתצוגה דו – ממדית) מהוות בסיס טוב להמשך מהלך הסיווג.

באמצעות שיטת<sup>v</sup> **RFE (Recursive Feature Elimination)** מודגם הקשר בין מספר הפיצ'רים הנבחרים לבין דיקט המודל. RFE מדרגת את הפיצ'רים לפי תרומתם לסיווג ומסירה בכל

איטרציה את הפיצ'ר בעל ההשפעה הפלטתית ביותר. מהגרף (МОЦГ 15) ניתן לראות כי הדוק משתפר עם הוספת הפיצ'רים ומתייצב לאחר הוספה שלושה פיצ'רים בערך של כ- 90%. מכאן ניתן להסיק כי עיקר התרומה לשינוי נובעת משלשות הפיצ'רים המרכזיים והפיצ'רים האחרונים מוסףים מיעט וAINS תורמים רבו לשינוי.

כדי לבחור את המודל הטוב ביותר נבחנו שישה סוגים מודלים: מודל LDA (עם חמשת הפיצ'רים), מודל רגסיה לוגיסטיבית (לשאלת הבונוס), מודל LDA לאחר PCA ל-2 ממדים (המייצגים את מרבית השונות כפי שניתן לראות בМОצגים 11-14), מודל LDA עם שלושה פיצ'רים נבחרים, מודל LDA עם 4 פיצ'רים נבחרים ומודל LDA לאחר נרמול כלל הפיצ'רים. כל אחד מהמודלים נבחן בשיטת **K-fold cross-validation**<sup>vii</sup>, שיטה נפוצה להערכת ביצועי מודל למידת מכונה בה באמצעות חלוקה למשתני קבוצות מאנימים את המודל כך שבכל פעע קבוצה אחרת משמשת כקבוצת הבדיקה והשאר כקבוצות האימון ולבסוף מודדים את ממוצע הביצועים. ערך ה- K הנבחר לקיפול הצלוב הוא 5, מאחר וזה מהו איזון אידיאלי בין חלוקה מואזנת של הנתונים המאפשרת יציבות סטטיסטית, זמן ריצה סביר והערכת ביצועים מהימנה. ערך קטן עלול להוביל להערכתה שאינה מייצגת עקב מספר ריצות נמוך להסתאמות המודל, בעוד שערך גדול עשוי לגרום להסתאמות יתר לדגימות ולפגוע ביכולת ההכללה של המודל ולזמן ריצה ארוך. מכאן שהשיטה מספקת הערכה יציבה ומהימנה ליכולת ההכללה של המודל תוך שימוש מיטבי בכלל המידע. תוצאות הבדיקה מוצגות בטבלה (МОצג 16).

האלגוריתם של **LDA** (Liner discriminant Analysis)<sup>viii</sup> נועד למצוא גבול ביןארי המפרד – בצורה מיטבית בין שתי קבוצות על ידי חישוב וחס בין שונות בין – קבוצתית לבין שונות Tonc – קבוצתית. הוא פשוט ומתאים למבנה נתונים ליניארי יחסית בדומה למבנה במתלה זו. מודל הרוגסיה הלוגיסטיבית (logistic regression)<sup>viii</sup> הוא אלגוריתם המחשב הסטברות להשתיכות לדגמים מסוימים באמצעות פונקציית סיגמוואיד ומתחאים אף הוא לביעית הסיווג הבינארי, אך הוא נבחר כמודל מתאים. כיוון שהמודלים מtabססים על הפרדה בצורה שונה ניתן לבדוק בצורה זו את הצלחת הפיצ'רים וההפרדה הליניארית בצורה המיטבית. במודלים בהם יתכונו שליבוי פיצ'רים שונים (קבוצות של שלושה או ארבעה פיצ'רים) נעשה שימוש בשיטת **RFE** כך שיבחרו הפיצ'רים בעלי הדוק המקסימלי. תוצאותיה של חקירה זו מראות כי המודל הטוב ביותר לשינוי דמיון בתחומי שבין [Hz] 20 – 1 – [s] 6 – 4 (לפי RFE), מודל זה מניב תוצאה של  $89.82\% \pm 4.73\%$  של Validation Accuracy (אחוז דיוק המודל על קבוצת הנתונים ששימשה לאימוט) ותוצאה של Train Accuracy (אחוז הדגימות שהמודל סיוג נכון במהלך האימון), אשר יחד מהווים את התוצאה הטובה ביותר ביותר בהשוואה למודלים האחרים. הבחירה במודל זה מותיישבת עם הנסיבות אשר נראו בМОצגים קודמים במתלה כפי שנאמר קודם לכן (МОצגים 6, 9 ו- 10 אשר מהווים סיוג טוב של תנומות הידיים).

ביצוע ההערכתה הסופית על סט הבדיקה (32=t) הניב מטריצת Confusion Matrix (МОצג 17) עם דיוק גבוהה של 96.88%. מתוך 32 דגימות, 31 סווגו נכון ודגימה אחת בלבד סוגה כשגיאה מסווג False Negative. מהדוק הגובה ניתן להסיק כי ביצועי הכלליים של המודל יכולת ההכללה שלו טובים.

ailona br shda 212366702

shirah altrz 315262683

לשיקום, מהלך העבודה כלל ניתוח תדרים, זמינים גזירת פיצ'רים רלוונטיים, הפקחת ממדים ובחינות מודלים שונים אשר הובילו בסופו של דבר לייצור מודל סיווג מדויק ואפקטיבי. ממצאי העבודה מצבעים על כך ששילוב בין עיבוד אותות EEG, בחירת תכונות מושכלת ומימוש אלגוריתמים הניב ביצועים מרשימים ב מבחני ההקללה ונוצר מודל בעל יכולות סיווג מוצלחות.

**ביבליוגרפיה:**

<sup>i</sup> Luo, Z., Jin, R., Shi, H., & Lu, X. (2021). Research on recognition of motor imagination based on connectivity features of brain functional network. *Neural Plasticity*, 2021, 6655430.

<https://PMC7895585/>

<sup>ii</sup> Putzolu, M., Samogin, J., Cosentino, C., Mezzarobba, S., Bonassi, G., Lagravinese, G., Pelosin, E., & Avanzino, L. (2022). Neural oscillations during motor imagery of complex gait: An HdEEG study. *Scientific Reports*, 12, Article 4314.

<https://www.nature.com/articles/s41598-022-07511-x>

<sup>iii</sup> Stancin, I., Cifrek, M., & Jovic, A. (2021). A review of EEG signal features and their application in driver drowsiness detection systems. *Sensors*, 21(11), 3786.  
[https://www.researchgate.net/publication/351995829\\_A\\_Review\\_of\\_EEG\\_Signal\\_Features\\_and\\_Their\\_Application\\_in\\_Driver\\_Drowsiness\\_Detection\\_Systems](https://www.researchgate.net/publication/351995829_A_Review_of_EEG_Signal_Features_and_Their_Application_in_Driver_Drowsiness_Detection_Systems)

<sup>iv</sup> Mouri, F. I., Valderrama, C. E., & Camorlinga, S. G. (2023). Identifying relevant asymmetry features of EEG for emotion processing. *Frontiers in Psychology*, 14, 1217178.

<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1217178/full>

<sup>v</sup> Yin, Z., Liu, L., Chen, J., Zhao, B., & Wang, Y. (2020). Locally robust EEG feature selection for individual-independent emotion recognition. *Expert Systems with Applications*, 162, 113768.

<https://www.sciencedirect.com/science/article/abs/pii/S0957417420305923>

<sup>vi</sup> Wilimitis, D., & Walash, C. G. (2023). Practical considerations and applies examples of cross-validation for model development and evaluation in health care: Tutorial. *JMIR AI*, 2, e49023.

<https://PMC11041453/>

<sup>vii</sup> Prasan, N. H. (2024, January 6). Linear Discriminant Analysis (LDA) in Classification. *Medium*. <https://medium.com/@prasanNH/linear-discriminant-analysis-lda-in-classification-354aca40ac57>

212366702 אילונה בר שדה

315262683 שירות אלטרץ

---

viii Sperandei, S. (2013). Understanding logistic regression analysis. *Biochimia Medica*, 23(1), 12–18.

<https://PMC3936971/>