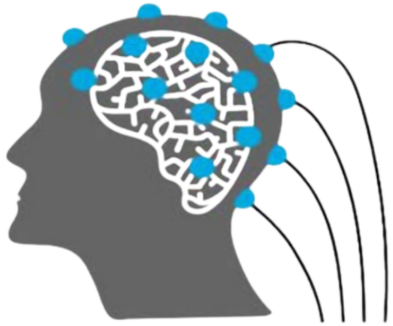
Prof. Michele Nappi
Dott.ssa Chiara Pero



Eeg for human recognition

Progetto Fondamenti di Visione Artificiale e Biometria
a.a. 2022/23

Mattia d'Argenio
Simone Masullo

Programma

- Introduzione
- Descrizione progetto
- Stato dell'arte
- Metodo proposto
- Sperimentazione
- Risulati ottenuti
- Conclusioni

Introduzione

Introduzione alla biometria cognitiva

TORNA AL PROGRAMMA

Introduzione

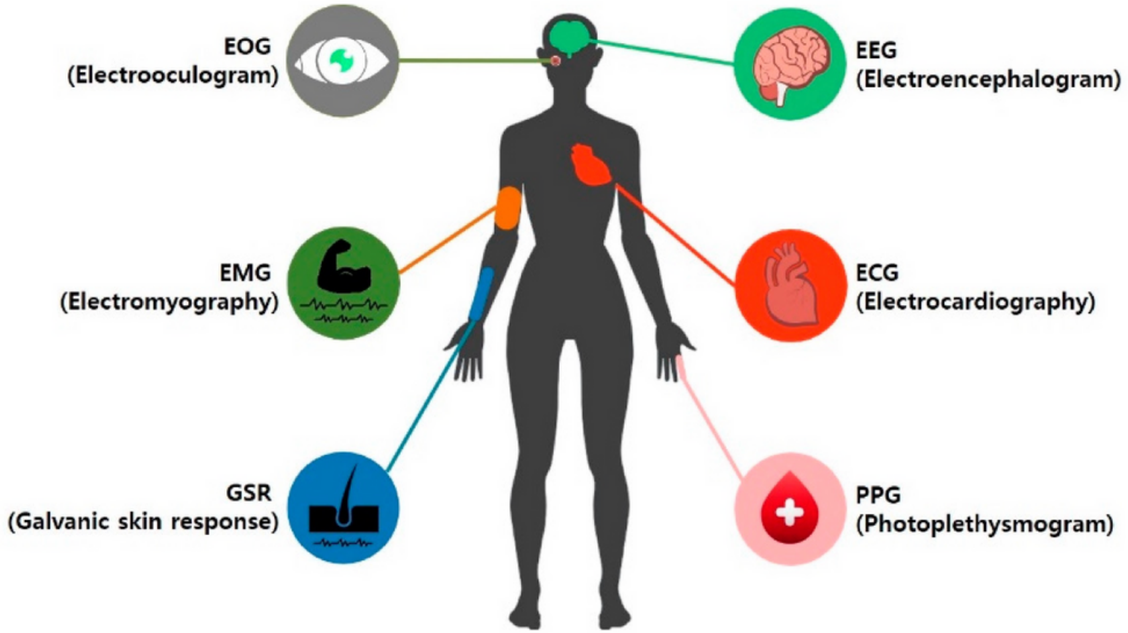
Cos'è la biometria cognitiva?

Integra le tradizionali modalità biometriche basate su caratteristiche fisiologiche e comportamentali con ulteriori elementi legati "al modo in cui pensiamo, sentiamo e reagiamo".

TORNA AL PROGRAMMA

Introduzione

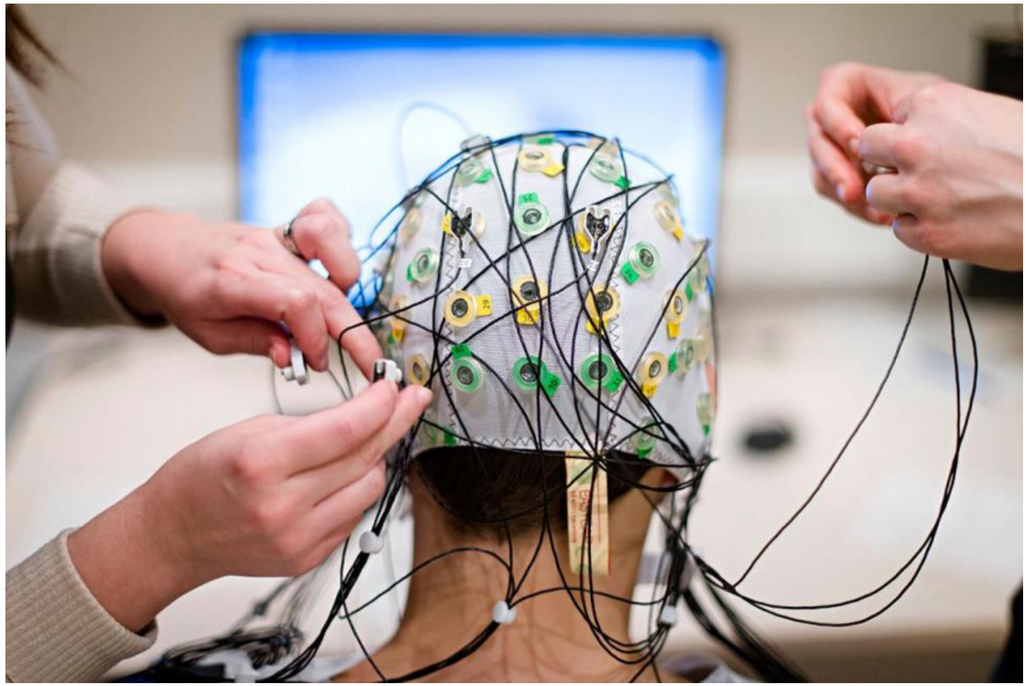
Essa si basa sui segnali provenienti dal cervello, dal cuore e dal sistema nervoso autonomo poiché questi trasmettono informazioni sul processo cognitivo ed emotivo di un individuo.



TORNA AL PROGRAMMA

Introduzione

La biometria cognitiva rivela un elevato grado di conformità alla **privacy**, robustezza e protezione degli utenti. La tecnologia di rilevamento attuale impedisce che tali biosegnali possano essere catturati senza il coinvolgimento consapevole dell'utente.



TORNA AL PROGRAMMA

Introduzione

Campi di applicazione

Sicurezza

Interazione uomo-macchina

Sanità

Task

Identificazione
Autenticazione

Emotion recognition

TORNA AL PROGRAMMA

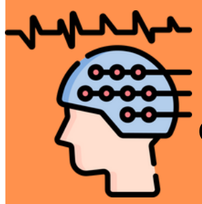
Descrizione del progetto

Descrizione dell'obiettivo preposto

TORNA AL PROGRAMMA

Descrizione progetto

Biosignal acquisition



Signal elicitation protocol

Signal pre-processing

Filtering

De-noising

Segmentation

Artifact removal

Features extraction

Feature selection/ extraction

Classification

ML

DL

TORNA AL PROGRAMMA

Descrizione progetto

Obiettivo

L'idea del progetto è stata quella di risolvere un'istanza del comune problema di **human identification** basandosi sui segnali EEG relativi ad ogni singolo soggetto. Per farlo vengono esaminate diverse tecniche di preprocessing note ed adoperati algoritmi di machine learning e deep learning.

TORNA AL PROGRAMMA

Descrizione progetto

BED (Biometric EEG Dataset)

Il dataset comprende le risposte EEG di 21 soggetti a 12 stimoli diversi, suddivisi in 3 sessioni cronologicamente distinte. Sono stati inoltre considerati stimoli mirati a suscitare diversi stati affettivi, al fine di facilitare future ricerche sull'influenza delle emozioni per task biometrici basati su EEG.

TORNA AL PROGRAMMA

Stato dell'arte

Descrizione dello stato dell'arte e dei lavori correlati

TORNA AL PROGRAMMA

Stato dell'arte

BED: A New Data Set for EEG-Based Biometrics, 2021

Representation Learning and Pattern Recognition in Cognitive Biometrics: A Survey, 2022

The PREP pipeline: standardized preprocessing for large-scale EEG analysis, 2015

State-of-the-art methods and future perspectives for personal recognition based on electroencephalogram signals, 2015

TORNA AL PROGRAMMA

Stato dell'arte

A novel deep learning approach for classification of eeg motor imagery signals, 2016	90%
Human identification from brain eeg signals using advanced machine learning method eeg-based biometrics, 2016	94.4%
Human identification with electroencephalogram (eeg) signal processing, 2012	95.1%
Support vector machine approach for human identification based on eeg signals, 2020	99.1%

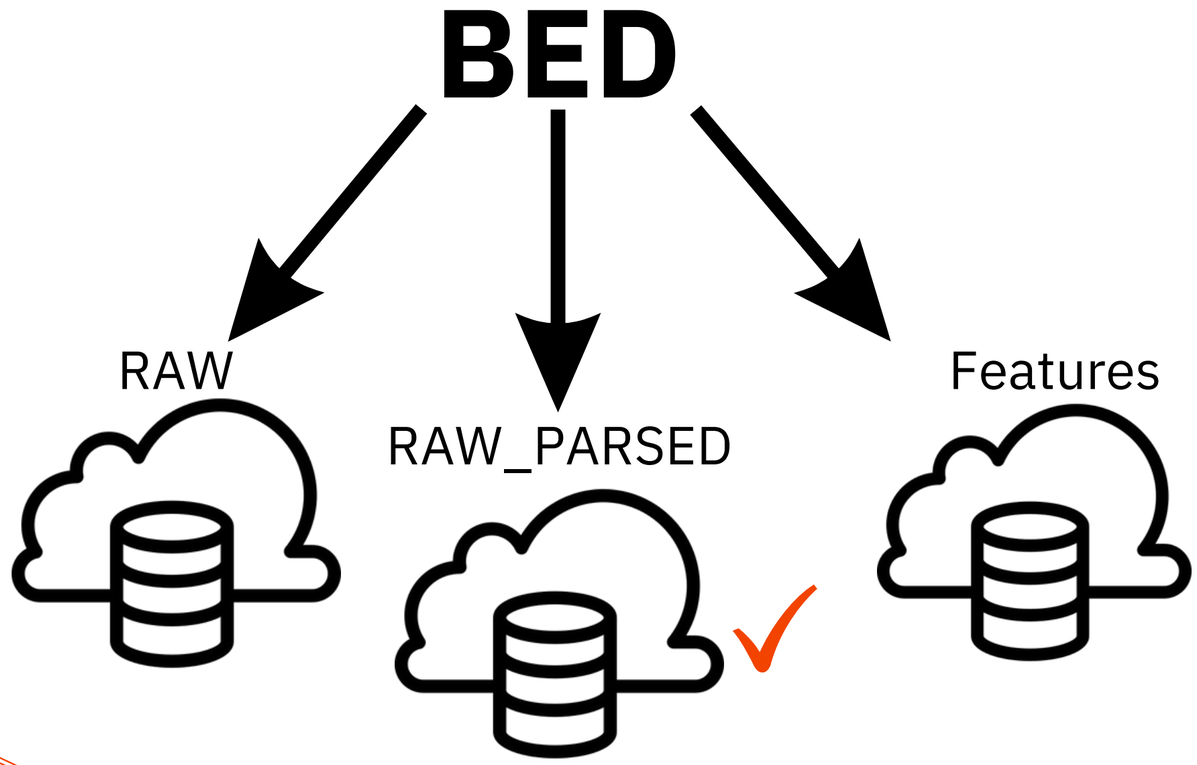
TORNA AL PROGRAMMA

Metodo proposto

Approfondimento del metodo e sulle tecniche utilizzate per lo svolgimento del progetto

TORNA AL PROGRAMMA

Metodo proposto



TORNA AL PROGRAMMA

Metodo proposto

Tecniche di pre-processing utilizzate

Filtro passa-banda → 1. eliminare la frequenza del segnale elettrico proveniente dai dispositivi stessi (50Hz);
2. eliminare le frequenze poco significative

Principal Component Analysis → ridurre la numerosità delle informazioni

TORNA AL PROGRAMMA

Metodo proposto

Altre tecniche di feature extraction

Power Spectral Density

Wavelet Transform

Ottenere valori statistici come ampiezza del segnale, entropia, centroidi...

TORNA AL PROGRAMMA

Metodo proposto

Scelta del modello

Random Forest → algoritmo di machine learning comunemente utilizzato che combina l'output di più strutture ad albero decisionali per raggiungere un unico risultato

Grid Search → effettua analisi su ogni possibile combinazione dei parametri assegnati in fase di inizializzazione e restituisce la combinazione che raggiunge le performance migliori

Xgboost → basato su un insieme di algoritmi di apprendimento ad albero decisionale, è un modello sequenziale, il che significa che ogni albero successivo dipende dal risultato dell'ultimo.

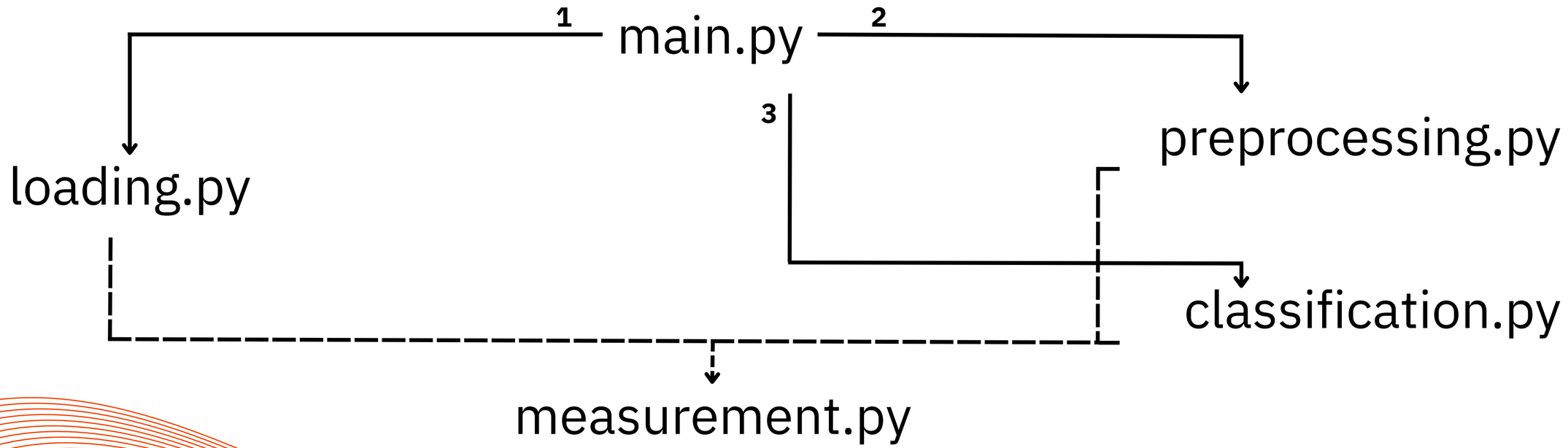
TORNA AL PROGRAMMA

Sperimentazione

Implementazione delle tecniche di preprocessing e dei modelli utilizzati

TORNA AL PROGRAMMA

Sperimentazione



TORNA AL PROGRAMMA

Eeg for human recognition

Sperimentazione

main.py

```python
import loading
import preprocessing
import classification

if __name__ == '__main__':
    # Loading the measurements for different users
    users_measurements = loading.load(loading.PATH_TO_DS_2)

    print("## INFO: starting preprocessing...")

    print("## INFO: applying butterworth bandpass filter...")

    # Apply a butterworth bandpass filter
    new_users_measurements = []
    for user_measurements in users_measurements:
        new_user_measurements = preprocessing.bandpass_filter(user_measurements)
        new_users_measurements.append(new_user_measurements)

    users_measurements = new_users_measurements

    print("## INFO: butterworth bandpass filter applied successfully!")

    # Apply principal component analysis to reduce the complexity of the signal

    print("## INFO: applying Principal Component Analysis...")

    pca_values = []
    for user_measurements in users_measurements:
        current_user_pca_values = preprocessing.pca_processing(user_measurements)
        pca_values.append(current_user_pca_values)

    # Reduce the number of measurements
    users_measurements = pca_values

    print("## INFO: Principal Component Analysis applied successfully!")

    # Starting Classification
    print("## INFO: starting classification...")

    """ TEST TEST TEST"""
    # x_train, y_train, x_test, y_test = classification.train_test_split_session(users_measurements, 2)

    # Split data in train and test
    x_train, y_train, x_test, y_test = classification.train_test_split(users_measurements, perc_train=70)

    # Compute a model for machine learning
    model = classification.classification_by_lstm(x_train, y_train)

    # Make predictions using the computed model
    predictions = classification.prediction_by_lstm(model, x_test)

    # Compute the confusion matrix based on the predictions
    confusion_matrix = classification.compute_confusion_matrix(predictions, y_test)

    # Compute the metrics
    metrics = classification.compute_metrics(confusion_matrix)

    # Print output
    for k, v in metrics.items():
        print("=====\n\tUser ID: {0}\n=====".format(k))
        print("Accuracy: {0}%\nPrecision: {1}%\nRecall: {2}%\nF-Score: {3}".format(*[metric for metric in v]))
```

TORNA AL PROGRAMMA

Sperimentazione

Librerie utilizzate

Tensorflow

Keras

Scikit-learn

TORNA AL PROGRAMMA

Sperimentazione

Tecniche di divisione del dataset → Split randomico e split per sessioni

---

Matrice di confusione → Computazione della matrice di confusione e delle metriche di valutazione del modello

TORNA AL PROGRAMMA

Eeg for human recognition

Sperimentazione

```python
def compute_metrics(confusion_matrix):
    """
    Calculates accuracy, precision, recall and fscore for each class c
    :param confusion_matrix: the confusion matrix previously computed
    :return: a dictionary of key, values where keys are the classes ar
    """

    # Identify all the classes
    classes = set()
    for k in confusion_matrix.keys():
        _, c = k
        classes.add(c)

    # Create a dictionary in which keys are the classes and values are
    metrics = dict()

    for cl in classes:
        # For each class we are computing TP, TN, FP, FN
        tp = tn = fp = fn = 0
        for k, v in confusion_matrix.items():
            p, c = k
            if p == c:
                tp += v
            elif p == cl and c != cl:
                fp += v
            elif p != cl and c == cl:
                fn += v
            elif p != cl and c != cl:
                tn += v

        if tp == tp == fp == fn == 0:
            acc = pr = recall = fscore = 0
        else:
            acc = ((tp + tn) / (tp + tn + fp + fn)) * 100
            pr = (tp / (tp + fp)) * 100
            recall = (tp / (tp + fn)) * 100
            fscore = (2 * recall * pr) / (recall + pr) / 100

        metrics[cl] = (acc, pr, recall, fscore)

    return metrics
```

```python
def compute_confusion_matrix(predictions, correct_labels):
    """
    Computes the confusion matrix given the predictions and
    :param predictions: the predictions of a model
    :param correct_labels: the expected predictions
    :return: a dictionary containing keys as couples (p, c)
    """

    # Create a dictionary for which the key is the tuple (p,
    # is the number of times the model predicted p and the e
    # If p == c then the model predicted the correct value.
    confusion_matrix = dict()

    for i in range(len(predictions)):
        p = predictions[i].item()
        c = correct_labels[i].item()

        key = (p, c)

        if not key in confusion_matrix.keys():
            confusion_matrix[key] = 0
        else:
            confusion_matrix[key] += 1

    return confusion_matrix
```

Eeg for human recognition

Sperimentazione

Random split

```
# Create lists of tuples (m, i) where m is a numpy array and i is the id of the su
train_data = []
test_data = []

# For each subject split into train and test
for user_measurement in users_measurements:
    # Calculate the number of measurements to use for training
    training_dataset_length = math.ceil(len(user_measurement) * perc_train / 100)

    # Get the measurement values
    data_set = user_measurement.values

    # Shuffle the measurement values
    np.random.shuffle(data_set)

    # Get the training portion
    train_data_0 = data_set[0:training_dataset_length]
    train_data.append((train_data_0, user_measurement.subject_id))

    # Get the testing portion
    test_data_0 = data_set[training_dataset_length:]
    test_data.append((test_data_0, user_measurement.subject_id))
```

Session split

```
# Create lists of tuples (m, i) where m is a numpy array and i is the id of the subject (source of
train_data = []
test_data = []

# For each user extract the rows of each session
for user_measurement in users_measurements:
    first_session_rows = user_measurement.sessions[1]
    second_session_rows = user_measurement.sessions[2]

    # Split data into three list of values containing each session
    first_session_measurements = user_measurement.values[:first_session_rows]
    second_session_measurements = user_measurement.values[first_session_rows:second_session_rows]
    third_session_measurements = user_measurement.values[second_session_rows:]

    user_session_measurements = [first_session_measurements, second_session_measurements,
                                 third_session_measurements]

    # Gather train data
    for i in range(n_session_train):
        train_data_0 = (user_session_measurements[i], user_measurement.subject_id)
        train_data.append(train_data_0)

    # Gather test data
    for i in range(n_session_train, len(user_session_measurements)):
        test_data_0 = (user_session_measurements[i], user_measurement.subject_id)
        test_data.append(test_data_0)
```

Eeg for human recognition

Sperimentazione

gridsearch

```python
def classification_by_gridsearch(x_train, y_train):
    """
    Trains a model using the random forest algorithm and the gridsearch tecnique to compute the best params.
    :param x_train: the measurements for the train
    :param y_train: the labels for the training
    :return: the trained model
    """
    param_grid = {
        'n_estimators': [10, 50, 100],
        'max_depth': [None, 5, 10],
        'criterion': ['gini', 'entropy']
    }

    rfc = RandomForestClassifier()
    grid_search = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=5)
    grid_search.fit(x_train, y_train)
    print("#### GridSearch output ####")
    print("Migliori parametri:", grid_search.best_params_)
    print("Miglior punteggio di validazione incrociata:", grid_search.best_score_)
    print("Miglior modello trovato:", grid_search.best_estimator_)

    return grid_search
```

```
/Users/mattiadargenio/PycharmProjects/provaEEG/venv/bin/python /Users/mattiadargenio/PycharmProjects/provaEEG/
## INFO: loading 937363 measurements for subject 1
## INFO: loading 828630 measurements for subject 2
## INFO: loading 873457 measurements for subject 3
## INFO: loading 3 subjects
## INFO: starting preprocessing...
## INFO: applying butterworth bandpass filter...
## INFO: butterworth bandpass filter applied successfully!
## INFO: applying Principal Component Analysis...
## INFO: Principal Component Analysis applied successfully!
## INFO: starting classification...
Migliori parametri: {'criterion': 'gini', 'max_depth': None, 'n_estimators': 100}
Miglior punteggio di validazione incrociata: 0.9998695616548678
Miglior modello trovato: RandomForestClassifier()

Process finished with exit code 0
```

Risultati ottenuti

Comparazioni dei risultati ottenuti nei vari modelli

TORNA AL PROGRAMMA

Eeg for human recognition

Risultati ottenuti

tensorflow_decision_forests

```
/Users/mattiadargenio/PycharmProjects/provaEEG/venv/bin/python /Users/ma                    /python-project/main.py
## INFO: loading 937363 measurements for subject 1
## INFO: loading 828630 measurements for subject 2
## INFO: loading 873457 measurements for subject 3
## INFO: loading 881930 measurements for subject 4
## INFO: loading 970655 measurements for subject 5
## INFO: loading 933960 measurements for subject 6
## INFO: loading 1080314 measurements for subject 7
## INFO: loading 884098 measurements for subject 8
## INFO: loading 1071813 measurements for subject 9
## INFO: loading 1090372 measurements for subject 10
## INFO: loading 1331210 measurements for subject 11
## INFO: loading 867922 measurements for subject 12
## INFO: loading 877334 measurements for subject 13
## INFO: loading 890254 measurements for subject 14
## INFO: loading 848005 measurements for subject 15
## INFO: loading 941757 measurements for subject 16
## INFO: loading 945612 measurements for subject 17
## INFO: loading 877143 measurements for subject 18
## INFO: loading 761855 measurements for subject 19
## INFO: loading 940829 measurements for subject 20
## INFO: loading 849249 measurements for subject 21
## INFO: loading 21 subjects
## INFO: starting preprocessing...
## INFO: applying butterworth bandpass filter...
## INFO: butterworth bandpass filter applied successfully!
## INFO: applying Principal Component Analysis...
## INFO: Principal Component Analysis applied successfully!
## INFO: starting classification...
Use /var/folders/c5/lnjb3v793d91yp_vjy4yk0jh0000gn/T/tmprmw1wmd6 as tem
Reading training dataset...
2023-06-12 15:53:47.920759: W tensorflow/tsl/platform/profile_utils/cpu_utils.cc:128] Failed to get CPU frequency: 0 Hz
Training dataset read in 0:00:17.976269. Found 13778642 examples.
Training model...
[INFO 23-06-12 17:29:29.1491 CEST kernel.cc:1242] Loading model from path /var/folders/c5/lnjb3v793d91yp_vjy4yk0jh0000gn/T/tmprmw1wmd6/model/ with prefix d0d0899d8c1b47e7
[INFO 23-06-12 17:29:31.9894 CEST decision_forest.cc:660] Model loaded with 300 root(s), 2069978 node(s), and 13 input feature(s).
[INFO 23-06-12 17:29:31.9894 CEST abstract_model.cc:1312] Engine "RandomForestGeneric" built
[INFO 23-06-12 17:29:31.9894 CEST kernel.cc:1074] Use fast generic engine
Model trained in 1:35:26.738192
```

Eeg for human recognition

Risultati ottenuti

```
Model trained in 1:35:26.738192
Compiling model...
WARNING:tensorflow:AutoGraph could not transform <function simple_ml_inf
Please report this to the TensorFlow team. When filing the bug, set the
Cause: could not get source code
To silence this warning, decorate the function with @tf.autograph.experi
WARNING:tensorflow:AutoGraph could not transform <function simple_ml_inf
Please report this to the TensorFlow team. When filing the bug, set the
Cause: could not get source code
To silence this warning, decorate the function with @tf.autograph.experi
Model compiled.

## INFO: starting predictions
184535/184535 [==============================] - 329s 2ms/step
 =====
 Class 1
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 2
=====
Accuracy: 99.89
Precision: 99.99
Recall: 99.89
F-Score: 99.94
 =====
 Class 3
=====
Accuracy: 99.75
Precision: 100.0
Recall: 99.75
F-Score: 99.87
 =====
 Class 4
=====
Accuracy: 99.94
Precision: 100.0
Recall: 99.94
F-Score: 99.97
 =====
 Class 5
=====
Accuracy: 99.98
Precision: 100.0
Recall: 99.98
F-Score: 99.99
 =====
 Class 6
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 7
=====
Accuracy: 99.95
Precision: 99.96
Recall: 99.98
F-Score: 99.97
 =====
 Class 8
=====
Accuracy: 99.99
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 9
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 10
=====
Accuracy: 99.97
Precision: 100.0
Recall: 99.97
F-Score: 99.98
 =====
 Class 11
=====
Accuracy: 99.98
Precision: 100.0
Recall: 99.98
F-Score: 99.99
 =====
 Class 12
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 13
=====
Accuracy: 99.99
Precision: 100.0
Recall: 99.99
F-Score: 99.99
 =====
 Class 14
=====
Accuracy: 99.99
Precision: 100.0
Recall: 99.99
F-Score: 99.99
 =====
 Class 15
=====
Accuracy: 99.98
Precision: 99.99
Recall: 99.99
F-Score: 99.99
 =====
 Class 16
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 17
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 18
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 18
=====
Accuracy: 100.0
Precision: 100.0
Recall: 100.0
F-Score: 100.0
 =====
 Class 19
=====
Accuracy: 99.91
Precision: 100.0
Recall: 99.92
F-Score: 99.96
 =====
 Class 20
=====
Accuracy: 99.98
Precision: 100.0
Recall: 99.98
F-Score: 99.99
 =====
 Class 21
=====
Accuracy: 99.95
Precision: 99.95
Recall: 100.0
F-Score: 99.97

Process finished with exit code 0
```

Eeg for human recognition

Risultati ottenuti

tensorflow_decision_forests con session split

```
=====
    User ID: 1
=====
Accuracy: 99.9974353779769%
Precision: 99.95806204403787%
Recall: 99.99396359443806%
F-Score: 0.9997600959616154
=====
    User ID: 2
=====
Accuracy: 99.88891476809366%
Precision: 99.53626814485689%
Recall: 97.54303493523622%
F-Score: 0.9852957189554346
=====
    User ID: 3
=====
Accuracy: 99.76518385275263%
Precision: 99.93455920024455%
Recall: 94.15637359294189%
F-Score: 0.9695945663706464
=====
    User ID: 4
=====
Accuracy: 99.94252988510493%
Precision: 99.96532920298621%
Recall: 98.70837860768349%
F-Score: 0.9933287772545569
=====
    User ID: 5
=====
Accuracy: 99.99306422974884%
Precision: 99.98975590801513%
Recall: 99.85853483006541%
F-Score: 0.9992410228893377
=====
=====
    User ID: 6
=====
Accuracy: 99.99549981418588%
Precision: 99.96503310199678%
Recall: 99.94206816645746%
F-Score: 0.9995354931514374
=====
    User ID: 7
=====
Accuracy: 99.9072058459188%
Precision: 99.31041939475487%
Recall: 98.96533250328477%
F-Score: 0.9913757564764276
=====
    User ID: 8
=====
Accuracy: 99.98556392005864%
Precision: 99.88638837487754%
Recall: 99.80932052166642%
F-Score: 0.9984783957701101
=====
    User ID: 9
=====
Accuracy: 99.99777410164032%
Precision: 100.0%
Recall: 99.95777156251339%
F-Score: 0.9997888132221288
=====
    User ID: 10
=====
Accuracy: 99.96340171465144%
Precision: 99.9775418991656%
Recall: 99.46524064171123%
F-Score: 0.9972073330584565
=====
=====
    User ID: 11
=====
Accuracy: 99.98728979777229%
Precision: 99.96193537472509%
Recall: 99.81614852271255%
F-Score: 0.9988898875518076
=====
    User ID: 12
=====
Accuracy: 99.99975805452613%
Precision: 100.0%
Recall: 99.99466186471741%
F-Score: 0.999973308611176
=====
    User ID: 13
=====
Accuracy: 99.98880598940859%
Precision: 99.96893123446561%
Recall: 99.7836346526748%
F-Score: 0.9987619700018553
=====
    User ID: 14
=====
Accuracy: 99.98651557225588%
Precision: 99.9996487740458%
Recall: 99.70758288361799%
F-Score: 0.9985340225999003
=====
    User ID: 15
=====
Accuracy: 99.97245047537446%
Precision: 99.73936879861775%
Recall: 99.63839890463636%
F-Score: 0.9968885828478576
=====
=====
    User ID: 16
=====
Accuracy: 99.99959675754354%
Precision: 99.99666859887931%
Recall: 99.99500298155434%
F-Score: 0.9999583578328083
=====
    User ID: 17
=====
Accuracy: 99.99491914504857%
Precision: 99.9826011891418%
Recall: 99.91206424993815%
F-Score: 0.9994732027433594
=====
    User ID: 18
=====
Accuracy: 99.9989515696132%
Precision: 99.98807333812809%
Recall: 99.98843471032563%
F-Score: 0.9998825402390036
=====
    User ID: 19
=====
Accuracy: 99.89754415666195%
Precision: 99.92551178902907%
Recall: 97.45087275413883%
F-Score: 0.9867267917368429
=====
    User ID: 20
=====
Accuracy: 99.98053145420198%
Precision: 99.95318449470464%
Recall: 99.62945424001826%
F-Score: 0.9979105681627369
=====
=====
    User ID: 21
=====
Accuracy: 99.91199636630158%
Precision: 99.24374881506604%
Recall: 98.8453495772343%
F-Score: 0.9904414856341976

Process finished with exit code 0
```

Risultati ottenuti

Grid Search

```
/Users/mattiadargenio/PycharmProjects/provaEEG/venv/bin/python /Users/mattiadargenio/
## INFO: loading 937363 measurements for subject 1
## INFO: loading 828630 measurements for subject 2
## INFO: loading 873457 measurements for subject 3
## INFO: loading 3 subjects
## INFO: starting preprocessing...
## INFO: applying butterworth bandpass filter...
## INFO: butterworth bandpass filter applied successfully!
## INFO: applying Principal Component Analysis...
## INFO: Principal Component Analysis applied successfully!
## INFO: starting classification...
Migliori parametri: {'criterion': 'gini', 'max_depth': None, 'n_estimators': 100}
Miglior punteggio di validazione incrociata: 0.9998695616548678
Miglior modello trovato: RandomForestClassifier()


Process finished with exit code 0
```

Eeg for human recognition

RandomForestClassifier

Risultati ottenuti

```
/Users/mattiadargenio/PycharmProjects/provaEEG/venv/bin/python
## INFO: loading 937363 measurements for subject 1
## INFO: loading 828630 measurements for subject 2
## INFO: loading 873457 measurements for subject 3
## INFO: loading 881930 measurements for subject 4
## INFO: loading 970655 measurements for subject 5
## INFO: loading 933960 measurements for subject 6
## INFO: loading 1080314 measurements for subject 7
## INFO: loading 884098 measurements for subject 8
## INFO: loading 1071813 measurements for subject 9
## INFO: loading 1090372 measurements for subject 10
## INFO: loading 1331210 measurements for subject 11
## INFO: loading 867922 measurements for subject 12
## INFO: loading 877334 measurements for subject 13
## INFO: loading 890254 measurements for subject 14
## INFO: loading 848005 measurements for subject 15
## INFO: loading 941757 measurements for subject 16
## INFO: loading 945612 measurements for subject 17
## INFO: loading 877143 measurements for subject 18
## INFO: loading 761855 measurements for subject 19
## INFO: loading 940829 measurements for subject 20
## INFO: loading 849249 measurements for subject 21
## INFO: loading 21 subjects
## INFO: starting preprocessing...
## INFO: applying butterworth bandpass filter...
## INFO: butterworth bandpass filter applied successfully!
## INFO: applying Principal Component Analysis...
## INFO: Principal Component Analysis applied successfully!
## INFO: starting classification...

## INFO: starting predictions
=====
    User ID: 1
=====
Accuracy: 99.99981371718883%
Precision: 100.0%
Recall: 99.99981370510568%
F-Score: 0.9999990685246607
=====
=====
    User ID: 2
=====
Accuracy: 99.99744284504656%
Precision: 99.9987975633553%
Recall: 99.99864514387386%
F-Score: 0.9999872135355651
=====
    User ID: 3
=====
Accuracy: 99.99815410668924%
Precision: 99.99850966028045%
Recall: 99.99964434671318%
F-Score: 0.99999077000278
=====
    User ID: 4
=====
Accuracy: 99.9996782387807%
Precision: 99.99989838451705%
Recall: 99.99977983338128%
F-Score: 0.9999983910891403
=====
    User ID: 5
=====
Accuracy: 99.99989839119391%
Precision: 99.9999491922327%
Recall: 99.9999491922327%
F-Score: 0.9999994919223271
=====
    User ID: 6
=====
Accuracy: 99.99996613039797%
Precision: 99.999996612814941%
Recall: 100.0%
F-Score: 0.9999998306407183
=====
=====
    User ID: 7
=====
Accuracy: 99.997900084674%
Precision: 99.99839111243736%
Recall: 99.9995088604122%
F-Score: 0.9999894998330134
=====
    User ID: 8
=====
Accuracy: 99.999593564477561%
Precision: 99.99989838451705%
Recall: 99.9996951541707%
F-Score: 0.9999979676924062
=====
    User ID: 9
=====
Accuracy: 100.0%
Precision: 100.0%
Recall: 100.0%
F-Score: 1.0
=====
    User ID: 10
=====
Accuracy: 99.99974597798476%
Precision: 99.99979676924062%
Recall: 99.9999491922327%
F-Score: 0.9999987298067857
=====
    User ID: 11
=====
Accuracy: 99.99984758679086%
Precision: 99.99991532041653%
Recall: 99.99993225632176%
F-Score: 0.9999992378836844
=====
=====
    User ID: 12
=====
Accuracy: 100.0%
Precision: 100.0%
Recall: 100.0%
F-Score: 1.0
=====
    User ID: 13
=====
Accuracy: 99.99989839119391%
Precision: 99.99989838451705%
Recall: 100.0%
F-Score: 0.9999994919223272
=====
    User ID: 14
=====
Accuracy: 99.99993226079593%
Precision: 100.0%
Recall: 99.99993225632176%
F-Score: 0.999999661281494
=====
    User ID: 15
=====
Accuracy: 99.99939034716341%
Precision: 99.9994072459274%
Recall: 99.99998306407183%
F-Score: 0.9999969515417071
=====
    User ID: 16
=====
Accuracy: 99.99998306519899%
Precision: 99.99998306407183%
Recall: 100.0%
F-Score: 0.9999999153203518
=====
=====
    User ID: 17
=====
Accuracy: 99.99996613039797%
Precision: 99.99996612814941%
Recall: 100.0%
F-Score: 0.9999998306407183
=====
    User ID: 18
=====
Accuracy: 99.99998306519899%
Precision: 99.99998306407183%
Recall: 100.0%
F-Score: 0.9999999153203518
=====
    User ID: 19
=====
Accuracy: 99.99817104149025%
Precision: 99.99944111739939%
Recall: 99.99872982130617%
F-Score: 0.9999908546808791
=====
    User ID: 20
=====
Accuracy: 99.99932260795936%
Precision: 99.99989838451705%
Recall: 99.99942418166053%
F-Score: 0.9999966128252663
=====
    User ID: 21
=====
Accuracy: 99.99796782387807%
Precision: 99.99962741090548%
Recall: 99.99834030630494%
F-Score: 0.9999898385446356

Process finished with exit code 0
```

Eeg for human recognition

Risultati ottenuti

RandomForestClassifier con session split

```
=====
    User ID: 1
=====
Accuracy: 99.98949956643372%
Precision: 99.80450040666325%
Recall: 99.99939635944381%
F-Score: 0.9990185332877024
=====
    User ID: 2
=====
Accuracy: 99.86128459497682%
Precision: 97.92411258651282%
Recall: 98.45150326355304%
F-Score: 0.9818709973902667
=====
    User ID: 3
=====
Accuracy: 99.79844329056168%
Precision: 99.806756704989%
Recall: 95.11530270763615%
F-Score: 0.9740457190806334
=====
    User ID: 4
=====
Accuracy: 99.9517560725088%
Precision: 99.80508357041919%
Recall: 99.08049655418782%
F-Score: 0.9944147014171457
=====
    User ID: 5
=====
Accuracy: 99.9925319497063%
Precision: 99.94352655487278%
Recall: 99.89310736536115%
F-Score: 0.9991831059968489
=====
=====
    User ID: 6
=====
Accuracy: 99.987128500789 71%
Precision: 99.76343864522109%
Recall: 99.97136702480081%
F-Score: 0.9986729460597472
=====
    User ID: 7
=====
Accuracy: 99.86130072467509%
Precision: 97.990835805228 43%
Recall: 99.46575600762607%
F-Score: 0.9872278738026595
=====
    User ID: 8
=====
Accuracy: 99.95656272258984%
Precision: 99.23658449253136%
Recall: 99.85282671280136%
F-Score: 0.9954375187420901
=====
    User ID: 9
=====
Accuracy: 99.99975805452613%
Precision: 99.99571612603158%
Recall: 99.9996939968298%
F-Score: 0.9999770502187114
=====
    User ID: 10
=====
Accuracy: 99.981531495494%
Precision: 99.86078202688601%
Recall: 99.85808498205192%
F-Score: 0.9985943348625824
=====
=====
    User ID: 11
=====
Accuracy: 99.9821121646313%
Precision: 99.75884312154813%
Recall: 99.92933120857711%
F-Score: 0.9984401438612136
=====
    User ID: 12
=====
Accuracy: 99.99988709211219%
Precision: 99.99928824103006%
Recall: 99.99822062157247%
F-Score: 0.9999875442845171
=====
    User ID: 13
=====
Accuracy: 99.98428967389621%
Precision: 99.78984861976848%
Recall: 99.86312307516825%
F-Score: 0.9982647240127277
=====
    User ID: 14
=====
Accuracy: 99.99056412651878%
Precision: 99.99719280515963%
Recall: 99.79793451957794%
F-Score: 0.9989746430092562
=====
    User ID: 15
=====
Accuracy: 99.96125646478306%
Precision: 99.21284042753214%
Recall: 99.91806621706262%
F-Score: 0.9956420453720771
=====
=====
    User ID: 16
=====
Accuracy: 99.99846767866543%
Precision: 99.96969353839596%
Recall: 99.99866746174783%
F-Score: 0.9998417840101926
=====
    User ID: 17
=====
Accuracy: 99.99406427104087%
Precision: 99.90310530782442%
Recall: 99.97392019579915%
F-Score: 0.9993850020722757
=====
    User ID: 18
=====
Accuracy: 99.99608048332318%
Precision: 99.9183817926392%
Recall: 99.9938559398605%
F-Score: 0.999561046191308
=====
    User ID: 19
=====
Accuracy: 99.91412548647169%
Precision: 99.54543363623817%
Recall: 98.25118974405541%
F-Score: 0.9889407737303909
=====
    User ID: 20
=====
Accuracy: 99.97938624562562%
Precision: 99.70318728537163%
Recall: 99.85551480627576%
F-Score: 0.9977929290835994
=====
=====
    User ID: 21
=====
Accuracy: 99.8582199523077%
Precision: 97.7478656919017%
Recall: 99.21216614097786%
F-Score: 0.9847457274153466

Process finished with exit code 0
```

Eeg for human recognition

Risultati ottenuti

xgboost

```
/Users/mattiadargenio/PycharmProjects/provaEEG/venv/bin/python /l
## INFO: loading 937363 measurements for subject 1
## INFO: loading 828630 measurements for subject 2
## INFO: loading 873457 measurements for subject 3
## INFO: loading 881930 measurements for subject 4
## INFO: loading 970655 measurements for subject 5
## INFO: loading 933960 measurements for subject 6
## INFO: loading 1080314 measurements for subject 7
## INFO: loading 884098 measurements for subject 8
## INFO: loading 1071813 measurements for subject 9
## INFO: loading 1090372 measurements for subject 10
## INFO: loading 1331210 measurements for subject 11
## INFO: loading 867922 measurements for subject 12
## INFO: loading 877334 measurements for subject 13
## INFO: loading 890254 measurements for subject 14
## INFO: loading 848005 measurements for subject 15
## INFO: loading 941757 measurements for subject 16
## INFO: loading 945612 measurements for subject 17
## INFO: loading 877143 measurements for subject 18
## INFO: loading 761855 measurements for subject 19
## INFO: loading 940829 measurements for subject 20
## INFO: loading 849249 measurements for subject 21
## INFO: loading 21 subjects
## INFO: starting preprocessing...
## INFO: applying butterworth bandpass filter...
## INFO: butterworth bandpass filter applied successfully!
## INFO: applying Principal Component Analysis...
## INFO: Principal Component Analysis applied successfully!
## INFO: starting classification...

[10. 17. 20. ... 17. 16. 17.]
=====
    User ID: 1
=====
Accuracy: 99.83279546073463%
Precision: 99.98304768753148%
Recall: 99.82079970886161%
F-Score: 0.9990185782257172
=====
    User ID: 12
=====
Accuracy: 99.76181863261134%
Precision: 99.74039765069699%
Recall: 99.98040189447028%
F-Score: 0.9986025556617859
=====
    User ID: 13
=====
Accuracy: 99.71997092507313%
Precision: 99.78811207461314%
Recall: 99.88341990540017%
F-Score: 0.9983574324369296
=====
    User ID: 14
=====
Accuracy: 99.39545219911312%
Precision: 99.41331435153184%
Recall: 99.87990585668646%
F-Score: 0.9964606390481433
=====
    User ID: 15
=====
Accuracy: 99.29846238888595%
Precision: 99.72905510846826%
Recall: 99.450362305288%
F-Score: 0.9958951373271977
=====
    User ID: 16
=====
Accuracy: 99.2978527096179%
Precision: 99.32047157454326%
Recall: 99.85930302134838%
F-Score: 0.9958915846057101
=====
    User ID: 7
=====
Accuracy: 98.280789141747 72%
Precision: 98.66954473480826%
Recall: 99.33268350462335%
F-Score: 0.9900000364476828
=====
    User ID: 8
=====
Accuracy: 98.70858076150296%
Precision: 99.68351297816986%
Recall: 98.8142123826424%
F-Score: 0.9924695917364299
=====
    User ID: 9
=====
Accuracy: 99.75443473925372%
Precision: 99.75255465236957%
Recall: 99.95951896998754%
F-Score: 0.9985592957122057
=====
    User ID: 10
=====
Accuracy: 99.55791478851582%
Precision: 99.71973358914448%
Recall: 99.76213978501455%
F-Score: 0.9974093217968891
=====
    User ID: 11
=====
Accuracy: 99.62543676745342%
Precision: 99.81387979972116%
Recall: 99.74697077549179%
F-Score: 0.9978041407093369
=====
    User ID: 2
=====
Accuracy: 99.59554554778327%
Precision: 99.89154088477272%
Recall: 99.63465390764274%
F-Score: 0.9976293202714658
=====
    User ID: 3
=====
Accuracy: 99.58690842481 91%
Precision: 99.72761025781068%
Recall: 99.78817152327849%
F-Score: 0.9975788169912413
=====
    User ID: 4
=====
Accuracy: 99.583233413675 52%
Precision: 99.75302990480108%
Recall: 99.75843621995566%
F-Score: 0.9975573298912883
=====
    User ID: 5
=====
Accuracy: 99.00474940149819%
Precision: 99.77788795791746%
Recall: 99.06199828411195%
F-Score: 0.9941865440064086
=====
    User ID: 6
=====
Accuracy: 98.91897091558923%
Precision: 98.86038327840774%
Recall: 99.88260589471064%
F-Score: 0.9936886571627394
=====
    User ID: 17
=====
Accuracy: 94.96796135446345%
Precision: 94.91656332904242%
Recall: 99.44577654216809%
F-Score: 0.9712839795224792
=====
    User ID: 18
=====
Accuracy: 95.2251274568381%
Precision: 99.60013359272256%
Recall: 95.04859384188602%
F-Score: 0.972711485887584
=====
    User ID: 19
=====
Accuracy: 99.12958123163341%
Precision: 99.48288808981593%
Recall: 99.49951359199983%
F-Score: 0.9949120014635571
=====
    User ID: 20
=====
Accuracy: 95.2909389467181%
Precision: 95.43355702403686%
Recall: 99.25702575245056%
F-Score: 0.9730774742256064
=====
    User ID: 21
=====
Accuracy: 95.67966028671184%
Precision: 99.9925975791298%
Recall: 95.17527743769395%
F-Score: 0.9752448465700616

Process finished with exit code 0
```

Eeg for human recognition

Risultati ottenuti

Xgboost con session split

```
=====
    User ID: 1
=====
Accuracy: 99.96033707198234%
Precision: 99.80946437983533%
Recall: 99.44766889108212%
F-Score: 0.9962823817550968
=====
    User ID: 2
=====
Accuracy: 99.33011750162586%
Precision: 93.51379573476846%
Recall: 88.5877101017958%
F-Score: 0.9098412436528938
=====
    User ID: 3
=====
Accuracy: 99.18498247669582%
Precision: 94.96723748692253%
Recall: 83.95254031031335%
F-Score: 0.8912084434081088
=====
    User ID: 4
=====
Accuracy: 99.46238102734564%
Precision: 95.40371329046295%
Recall: 92.0307220576634%
F-Score: 0.9368686820856766
=====
    User ID: 5
=====
Accuracy: 98.98523229346245%
Precision: 92.94328851089017%
Recall: 84.19823468401408%
F-Score: 0.8835490038926206
=====
```

```
=====
    User ID: 6
=====
Accuracy: 98.82132229975947%
Precision: 81.31620718923268%
Recall: 98.24340026568737%
F-Score: 0.8898193534963036
=====
    User ID: 7
=====
Accuracy: 98.28371945680338%
Precision: 80.92014143487967%
Recall: 89.18103538580796%
F-Score: 0.8484999494548962
=====
    User ID: 8
=====
Accuracy: 98.76591678624149%
Precision: 92.62812856130238%
Recall: 80.39298326711102%
F-Score: 0.8607795327170829
=====
    User ID: 9
=====
Accuracy: 99.7944269956953%
Precision: 96.93589318220054%
Recall: 99.23682809353906%
F-Score: 0.9807286665809827
=====
    User ID: 10
=====
Accuracy: 99.39928164775836%
Precision: 95.89794475521873%
Recall: 94.91585765285328%
F-Score: 0.9540437388404013
=====
```

```
=====
    User ID: 11
=====
Accuracy: 99.50538280290283%
Precision: 94.68509629286912%
Recall: 96.79991440911317%
F-Score: 0.9573082700695682
=====
    User ID: 12
=====
Accuracy: 99.74158610420044%
Precision: 94.86951103043275%
Recall: 99.68967640223917%
F-Score: 0.9721988439466849
=====
    User ID: 13
=====
Accuracy: 99.21635473980861%
Precision: 86.67862089903417%
Recall: 97.69697444964069%
F-Score: 0.918585672392124
=====
    User ID: 14
=====
Accuracy: 98.936004454377747%
Precision: 82.313537822199446%
Recall: 97.94397498170204%
F-Score: 0.8945108543557351
=====
    User ID: 15
=====
Accuracy: 98.34709304126106%
Precision: 92.6844146440651%
Recall: 68.05492840808121%
F-Score: 0.7848272965879265
=====
```

```
=====
    User ID: 16
=====
Accuracy: 99.37653877321387%
Precision: 89.5817315550825%
Recall: 98.58917512550845%
F-Score: 0.9386986730442587
=====
    User ID: 17
=====
Accuracy: 94.89490211208721%
Precision: 48.46528050630503%
Recall: 91.97176694017026%
F-Score: 0.6347951006467469
=====
    User ID: 18
=====
Accuracy: 91.88906832282106%
Precision: 0.04505877519646951%
Recall: 0.036864360837037784%
F-Score: 0.0004055174223404319
=====
    User ID: 19
=====
Accuracy: 95.56726858399315%
Precision: 12.164833375968001%
Recall: 2.1590810594397367%
F-Score: 0.036672742568704425
=====
    User ID: 20
=====
Accuracy: 94.98947053297685%
Precision: 47.87819483972848%
Recall: 83.21137354261795%
F-Score: 0.6078302651412816
=====
```

```
=====
    User ID: 21
=====
Accuracy: 95.3835997099235%
Precision: 5.058365758754864%
Recall: 0.004545867806164196 5%
F-Score: 9.083572359388047e-05

Process finished with exit code 0
```

Conclusioni

Descrivi in breve di cosa parlano le diapositive successive.

TORNA AL PROGRAMMA

Conclusioni

	Accuracy	Precision	Recall	FScore
Random forest con random split	99.9%	99.9%	99.9%	0.99
Random forest con session split	99.9%	99.9%	94.1%	0.96
Grid search con random split	99.9%	99.9%	99.9%	0.99
Grid search con session split	98.8%	97.7%	95.1%	0.97
Xgboost con random split	94.9%	94.9%	94.9%	0.97
Xgboost con session split	91.8%	80.9%	83.2%	0.60
LSTM con random split	~20,15%	- N/A -	- N/A -	- N/A -
LSTM con session split	~19,47%	- N/A -	- N/A -	- N/A -

Conclusioni

Sviluppi futuri

L'applicazione risulta performante dal punto di vista dell'accuracy ma non dal punto di vista dell'esecuzione. Al fine di risolvere questo problema si è pensato ad una parallelizzazione del carico di lavoro.

TORNA AL PROGRAMMA


Thank
You!