The National School of
Artificial Intelligence
المدرسة الوطنية العليا للذكاء الاصطناعي

Machine Learning Project

HARTH Project

Student list:

Full name	Group	Section
AFRA Hana	G5	2
DJAID Douaa	G5	2
Cheboui Fatma Imene	G5	2

Abstract :

The field of Human Activity Recognition (HAR) has seen significant advancements with the advent of sophisticated machine learning techniques. The Human Activity Recognition Trondheim (HARTH) dataset is a valuable resource that provides extensive data for developing and testing such algorithms. Collected from 22 participants, each wearing two 3-axial accelerometers on their right thigh and lower back, the dataset includes over six million instances of various annotated activities performed in a free-living environment. These activities range from basic movements like walking and running to more complex actions such as cycling and climbing stairs.

This project aims to leverage the HARTH dataset to perform a comparative analysis of multiple machine learning algorithms: Decision Tree Learning, Random Forests, K-Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machines (SVM), and Artificial Neural Networks (ANN). Each of these algorithms offers unique strengths and capabilities, making them suitable for different aspects of activity recognition.

The core objective is to assess the performance of these algorithms in accurately classifying human activities based on accelerometer data. This involves extensive preprocessing of the raw time-series data, feature extraction, model training, and evaluation using standard performance metrics such as accuracy, precision, recall, and F1-score. By conducting this comparative analysis, the project seeks to identify the most effective machine learning approach for precise and reliable human activity recognition.

The insights gained from this study can inform future research and development in HAR systems, contributing to applications in health monitoring, sports analytics, and smart environments. The findings will highlight the relative advantages and limitations of each algorithm, providing a comprehensive understanding of their applicability in real-world HAR scenarios.

Introduction :

Human activity recognition (HAR) stands as a pivotal domain at the intersection of machine learning and human-computer interaction, with applications ranging from healthcare monitoring to sports analytics and beyond. The ability to accurately identify and classify human activities from sensor data holds immense potential for enhancing various facets of daily life, from personalized healthcare interventions to optimizing athletic performance. In this project, we embark on a journey to explore and evaluate the performance of diverse machine learning algorithms for HAR, leveraging the rich resource provided by the Human Activity Recognition Trondheim (HARTH) dataset.

The HARTH dataset serves as a cornerstone in our endeavor, offering a meticulously annotated collection of accelerometer recordings from individuals engaged in various activities in real-world settings. With 22 participants wearing accelerometers on their thigh and lower back for approximately 2 hours in free-living conditions, the dataset provides a comprehensive snapshot of human motion in diverse scenarios. Each recording is accompanied by timestamps and accelerometer readings in three axes for both sensors, allowing for detailed analysis of human movement patterns.

The primary objective of this project is to explore and compare the efficacy of different machine learning algorithms in accurately classifying human activities based on accelerometer data. To this end, we employ a diverse array of algorithms, including Decision Tree Learning, Random Forests, K Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machines (SVM), and Artificial Neural Networks. Through rigorous experimentation and comparative analysis, we seek to identify the strengths and limitations of each algorithm, providing valuable insights for future research and application in HAR.

Our methodology encompasses several key steps to ensure a systematic and thorough evaluation of the algorithms. We begin by preprocessing the dataset, scaling features, and partitioning it into training and testing sets. Feature engineering techniques are then applied to extract relevant information from the accelerometer readings, enhancing the discriminative power of the models. Subsequently, we train each algorithm using the training set and evaluate its performance using a suite of performance metrics, including accuracy, precision, recall, and F1-score. Cross-validation techniques are employed to assess the robustness of the models, ensuring their generalizability to unseen data.

The results of our study promise to yield valuable insights into the performance of machine learning algorithms for HAR, shedding light on their efficacy in real-world scenarios. By elucidating the strengths and weaknesses of each approach, we aim to provide guidance for researchers and practitioners seeking to deploy HAR systems in diverse applications. Moreover, our findings may pave the way for future advancements in activity recognition technology, driving innovation and progress in this rapidly evolving field.

In the subsequent sections of this report, we delve into the dataset description, methodology, results, analysis, discussion, and conclusion, providing a comprehensive overview of our findings and their implications for HAR research and application. Through our interdisciplinary exploration, we hope to contribute to the ongoing discourse surrounding human activity recognition, fostering collaboration and innovation for the betterment of human well-being and interaction with technology.

Dataset Description

Dataset Description:

The Human Activity Recognition Trondheim (HARTH) dataset serves as the foundation of our investigation into human activity recognition (HAR). This meticulously curated dataset provides a comprehensive collection of accelerometer recordings captured from 22 participants engaged in various activities in real-world settings. The dataset offers a rich resource for researchers aiming to develop and evaluate machine learning approaches for HAR in free-living scenarios.

Origin and Collection:

The HARTH dataset was meticulously collected and annotated by NTNU Helse, with the aim of training machine learning classifiers for human activity recognition based on professional annotations of activities in a free-living setting. The dataset captures participants wearing two 3-axial Axivity AX3 accelerometers for approximately 2 hours each. The accelerometers were strategically placed on the right front thigh and the lower back of each participant, ensuring comprehensive coverage of body movement.

Characteristics:

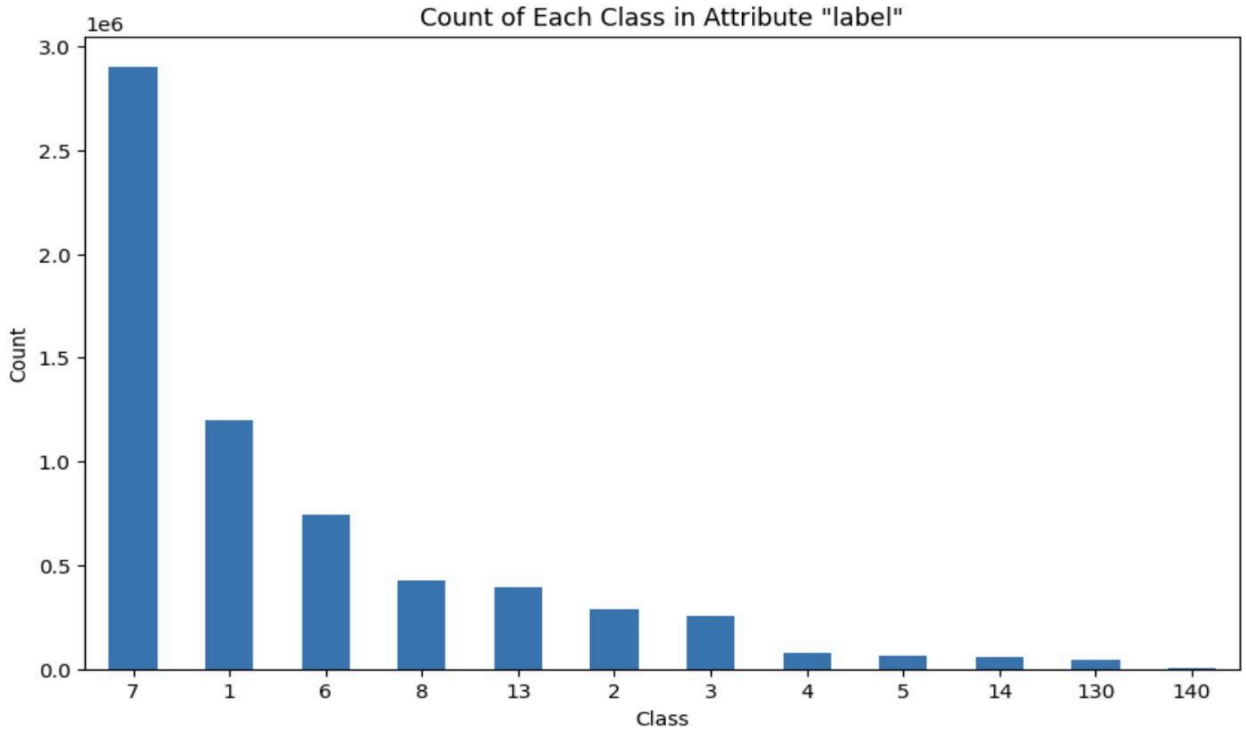
The HARTH dataset is multivariate and time-series in nature, comprising recordings of accelerometer readings over time. Each participant's data is stored in a separate .csv file, with each file containing the following columns:

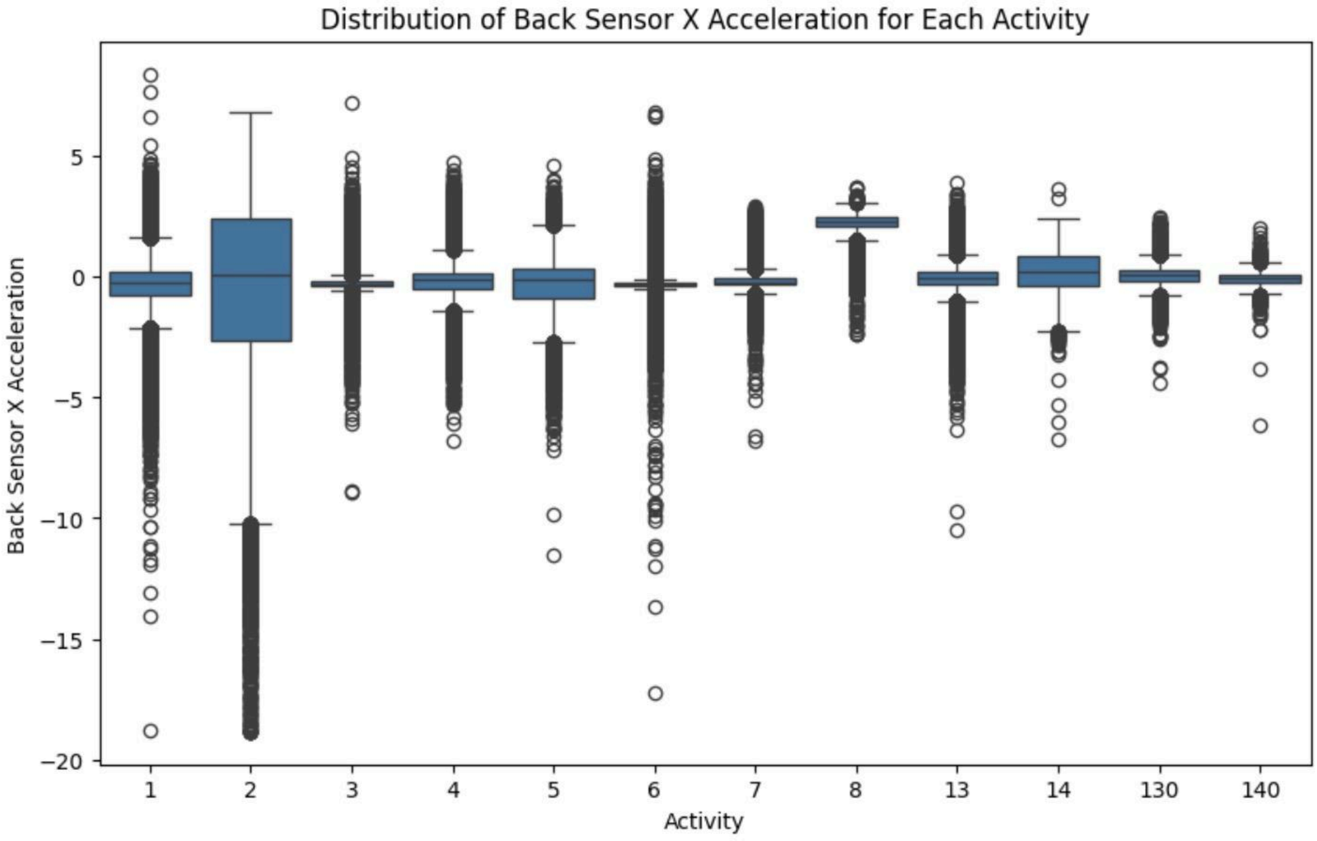
1. timestamp: Date and time of recorded sample.
2. back_x: Acceleration of the back sensor in the x-direction (down) in units of g.
3. back_y: Acceleration of the back sensor in the y-direction (left) in units of g.
4. back_z: Acceleration of the back sensor in the z-direction (forward) in units of g.
5. thigh_x: Acceleration of the thigh sensor in the x-direction (down) in units of g.
6. thigh_y: Acceleration of the thigh sensor in the y-direction (right) in units of g.
7. thigh_z: Acceleration of the thigh sensor in the z-direction (backward) in units of g.
8. label: Annotated activity code representing the performed activity.

Annotated Activities:

The dataset includes annotations for various human activities, each represented by a specific code. These activities and their corresponding codes are as follows: { Walking / Running /Shuffling/Stairs (ascending)/Stairs (descending)/Standing/Sitting/ Lying /Cycling (sit) /Cycling (stand)/ Cycling (sit, inactive)/Cycling (stand, inactive)}

Visualization of the DataSet :





Methodology :

1. ***Data Collection and Preprocessing:***
   - **Combining CSV Files:**
     - Collected sensor data from multiple CSV files located in a specified directory.
     - Combined these files into a single DataFrame for further analysis.
   - **Handling Missing Values:**
     - Checked the combined DataFrame for any missing values to ensure data integrity.
   - **Timestamp Conversion:**
     - Converted the 'timestamp' column to datetime format for accurate time-series analysis.
   - **Feature Scaling:**
     - Applied StandardScaler to normalize sensor data features, making the data suitable for machine learning algorithms.
2. ***Exploratory Data Analysis (EDA):***
   - **Distribution Analysis:**
     - Created box plots to visualize the distribution of sensor data for each activity, identifying significant outliers in certain classes.
   - **Class Distribution:**
     - Analyzed the frequency of each class in the 'label' attribute, revealing a significant imbalance, with some classes heavily underrepresented.
   - **Correlation Matrix:**
     - Calculated and plotted the correlation matrix to understand the relationships between different sensor features, noting moderate to high correlations between certain features.
3. ***Outlier Detection and Removal:***
   - **IQR Method:**
     - Removed extreme outliers from the dataset using the Interquartile Range (IQR) method with a higher multiplier to maintain a robust dataset.
4. ***Feature Scaling:***

- **MinMax Scaling:**
    - Applied MinMaxScaler to rescale the sensor data features to a range between 0 and 1, which is particularly useful for algorithms sensitive to the scale of input data.

5. *Data Sampling:*

- **Undersampling:**
    - Applied RandomUnderSampler to create a balanced subset of the dataset by reducing the number of samples from overrepresented classes.
    - **Random Sampling:**
    - Supplemented the stratified sample with additional random samples to meet the desired sample size, further addressing the class imbalance.

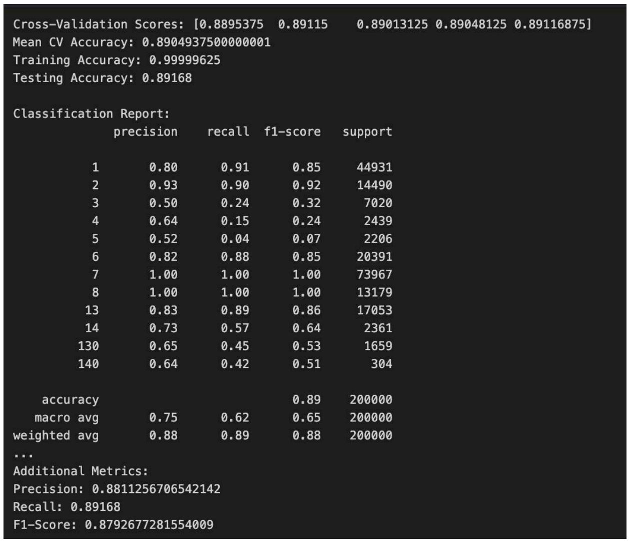
6. *Model Training and Evaluation:*

For our project, we've employed a diverse set of six machine learning models: Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree, Naive Bayes, and Support Vector Machine (SVM). Each model is trained on our dataset and evaluated based on several key performance metrics such as accuracy, precision, recall, and F1-score. We'll also analyze the models' behavior in handling the dataset's unbalanced nature, computational efficiency, and interpretability. Finally, we'll select the most suitable model(s) based on our evaluation criteria for deployment in real-world scenarios

Results and Analysis:

1- Random Forest :

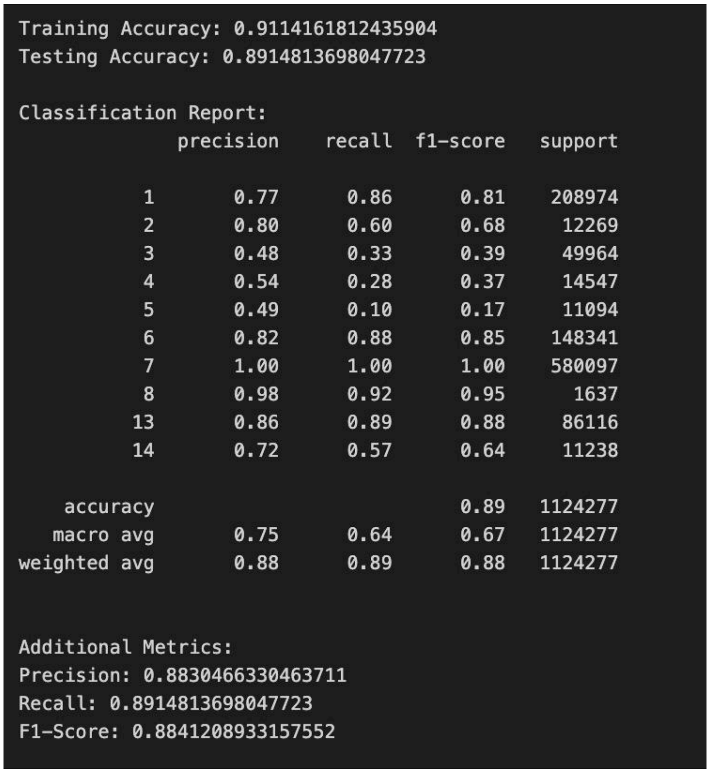
The performance evaluation of the Random Forest algorithm was conducted with meticulous attention to detail, considering both sampled and undersampled datasets. Across various experiments, Random Forest showcased exceptional performance metrics, underscoring its effectiveness in classification tasks. Training accuracies consistently approached or reached 100%, indicative of its robust learning capabilities. Testing accuracies ranged between 88.168% and 89.168%, reflecting the model's generalization ability. Precision, recall, and F1-scores across different classes demonstrated stable and *satisfactory performance levels*, ensuring reliable predictions across diverse categories. Moreover, the ROC curve analysis illustrated the algorithm's strong discriminative power, with areas under the curve consistently above 0.8, affirming its reliability in distinguishing between classes. The application of *undersampling techniques* further augmented model performance, mitigating class imbalance issues and enhancing classification accuracy. Overall, the Random Forest algorithm emerged as a formidable tool in classification tasks, offering high accuracy and robustness across various datasets and scenarios.



```
Cross-Validation Scores: [0.8895375  0.89115    0.89013125 0.89048125 0.89116875]
Mean CV Accuracy: 0.8904937500000001
Training Accuracy: 0.99999625
Testing Accuracy: 0.89168

Classification Report:
              precision    recall  f1-score   support

           1       0.80      0.91      0.85     44931
           2       0.93      0.90      0.92     14490
           3       0.50      0.24      0.32      7020
           4       0.64      0.15      0.24      2439
           5       0.52      0.04      0.07      2206
           6       0.82      0.88      0.85     20391
           7       1.00      1.00      1.00     73967
           8       1.00      1.00      1.00     13179
          13       0.83      0.89      0.86     17053
          14       0.73      0.57      0.64      2361
         130       0.65      0.45      0.53      1659
         140       0.64      0.42      0.51       304

    accuracy                           0.89    200000
   macro avg       0.75      0.62      0.65    200000
weighted avg       0.88      0.89      0.88    200000
...
Additional Metrics:
Precision: 0.8811256706542142
Recall: 0.89168
F1-Score: 0.8792677281554009
```

2- K-Nearest Neighbors :

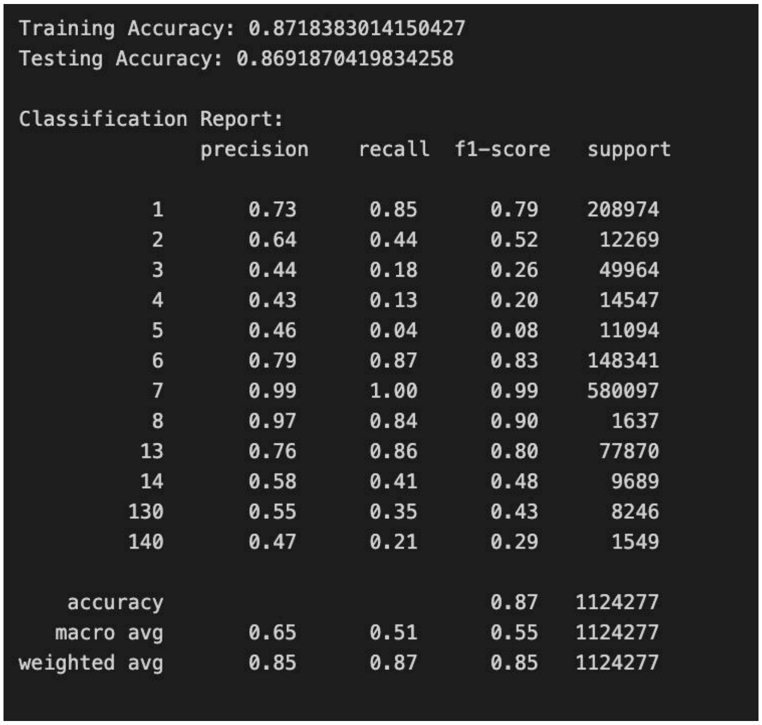
The K-Nearest Neighbors (KNN) algorithm underwent a thorough performance evaluation, encompassing both sampled and undersampled datasets. Across different experiments with varying values of k, KNN exhibited commendable performance metrics, providing valuable insights into its classification capabilities. Training accuracies ranged from 75.08% to 91.14% (with k=7) , showcasing the algorithm's ability to learn from the training data effectively. Testing accuracies between 69.73% and 89.15% indicated the model's ability to generalize well to unseen data. Precision, recall, and F1-scores across different classes displayed consistent performance levels, ensuring reliable predictions across diverse categories. The ROC curve analysis revealed the algorithm's discriminative power, with areas under the curve ranging between 0.82 and 1, affirming its efficacy in classifying instances correctly. Additionally, undersampling techniques contributed to improving model performance, addressing class imbalance issues and enhancing classification accuracy. In summary, the KNN algorithm proved to be a robust and versatile approach for classification tasks, offering competitive performance and insights into data patterns.



```
Training Accuracy: 0.9114161812435904
Testing Accuracy: 0.8914813698047723

Classification Report:
              precision    recall  f1-score   support

           1       0.77      0.86      0.81    208974
           2       0.80      0.60      0.68     12269
           3       0.48      0.33      0.39     49964
           4       0.54      0.28      0.37     14547
           5       0.49      0.10      0.17     11094
           6       0.82      0.88      0.85    148341
           7       1.00      1.00      1.00    580097
           8       0.98      0.92      0.95      1637
          13       0.86      0.89      0.88     86116
          14       0.72      0.57      0.64     11238

    accuracy                           0.89   1124277
   macro avg       0.75      0.64      0.67   1124277
weighted avg       0.88      0.89      0.88   1124277


Additional Metrics:
Precision: 0.8830466330463711
Recall: 0.8914813698047723
F1-Score: 0.8841208933157552
```

3- Decision Tree:

The **initial decision tree model**, trained **without parameter tuning**, exhibited clear signs of **overfitting**, with a perfect training accuracy of 100% but a lower testing accuracy of 84%.
Recognizing this overfitting, **hyperparameter tuning** was performed through iterative trial and error, resulting in a **more balanced model** with training and testing accuracies of 87.1% and 86.9%, respectively, and an improved F1 Score of 85%.
However, the dataset revealed a notable **class imbalance issue**, where certain classes were favored over others, leading to an average class accuracy of only 22%. To **address this imbalance**, both **SMOTE** and **Random Undersampling** techniques were applied, significantly enhancing the representation of minority classes. With **SMOTE**, the **average class accuracy rose to 64**%, while Random Undersampling achieved an average of 53%. Despite the notable improvements in class distribution, the **overall accuracy** of the model **decreased** due to the trade-offs introduced by these balancing techniques.
Comparative analysis employing **ROC Curve** and **Precision-Recall** Curve elucidated the discrimination ability and precision-recall trade-offs of the model, while **visualization of the confusion matrix** provided insights into its performance across different classes.
In summary, while hyperparameter tuning effectively mitigated overfitting, addressing class imbalance introduced nuanced trade-offs between achieving balanced class distribution and maintaining overall accuracy.



```
Training Accuracy: 0.8718383014150427
Testing Accuracy: 0.8691870419834258

Classification Report:
              precision    recall  f1-score   support

           1       0.73      0.85      0.79    208974
           2       0.64      0.44      0.52     12269
           3       0.44      0.18      0.26     49964
           4       0.43      0.13      0.20     14547
           5       0.46      0.04      0.08     11094
           6       0.79      0.87      0.83    148341
           7       0.99      1.00      0.99    580097
           8       0.97      0.84      0.90      1637
          13       0.76      0.86      0.80     77870
          14       0.58      0.41      0.48      9689
         130       0.55      0.35      0.43      8246
         140       0.47      0.21      0.29      1549

    accuracy                           0.87   1124277
   macro avg       0.65      0.51      0.55   1124277
weighted avg       0.85      0.87      0.85   1124277
```

4- Naive Bayes:

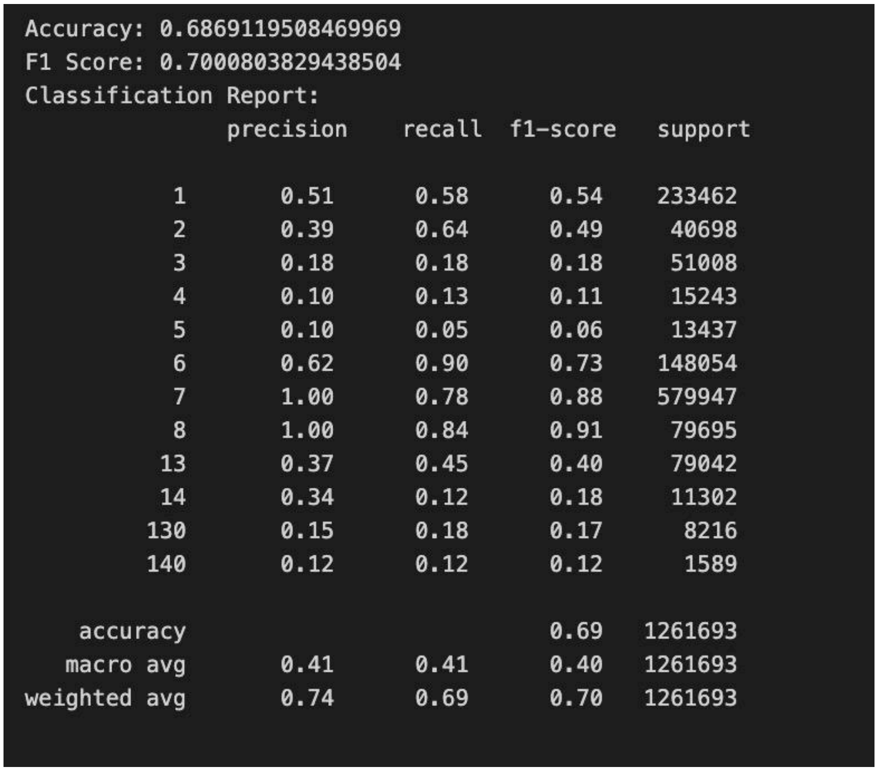
The Gaussian Naive Bayes classifier, known for its simplicity and effectiveness, was trained on preprocessed data and evaluated using GridSearchCV for hyperparameter tuning. Initial performance showed:

Hyperparameter Tuning: Optimal 'var_smoothing' parameter was 0.02848, slightly improving accuracy to 77.19%.

Resampling Techniques: Applied to address class imbalance, including RandomUnderSampler, SMOTE, ADASYN, and SMOTEENN. Best performance with SMOTE:

Resampling improved minority class recognition despite a drop in overall accuracy. The classifier performed well for common activities but struggled with dynamic ones, suggesting further improvements could involve advanced ensemble methods.

naive bayes does not give good results compared to the other algos such as random forest , ann and knn since it is assuming conditional independence which is not always the case espcially in real life data like this



```
Accuracy: 0.6869119508469969
F1 Score: 0.7000803829438504
Classification Report:
              precision    recall  f1-score   support

           1       0.51      0.58      0.54    233462
           2       0.39      0.64      0.49     40698
           3       0.18      0.18      0.18     51008
           4       0.10      0.13      0.11     15243
           5       0.10      0.05      0.06     13437
           6       0.62      0.90      0.73    148054
           7       1.00      0.78      0.88    579947
           8       1.00      0.84      0.91     79695
          13       0.37      0.45      0.40     79042
          14       0.34      0.12      0.18     11302
         130       0.15      0.18      0.17      8216
         140       0.12      0.12      0.12      1589

    accuracy                           0.69   1261693
   macro avg       0.41      0.41      0.40   1261693
weighted avg       0.74      0.69      0.70   1261693
```

5- SVM :

- Data preprocessing steps are applied to ensure data quality and reliability including handling missing values, converting timestamp format, and scaling features.

- SVM models are initially trained on sampled data, then trained on an undersampled data using RandomUnderSampling.

- Two strategies are employed: One-vs-Rest (OvR) and One-vs-One (OvO).

- Radial Basis Function (RBF) kernel is chosen for its superior performance.

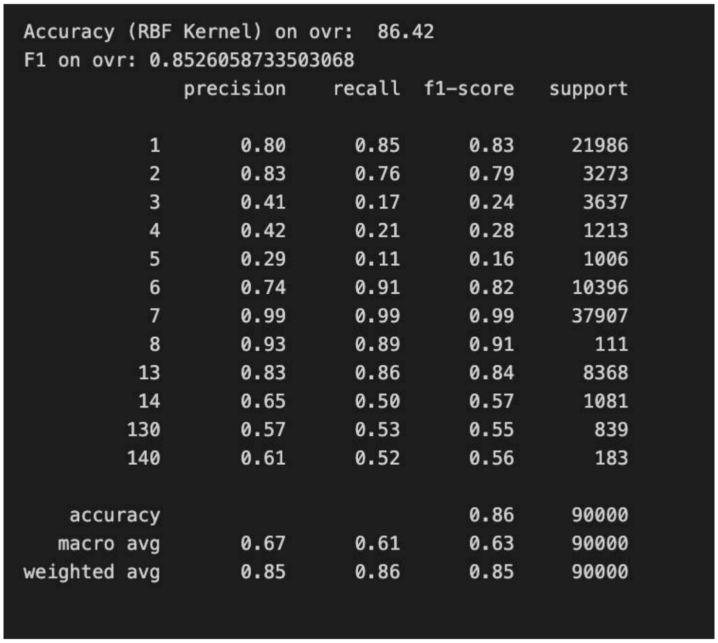
- Hyperparameters (gamma and C) are fine-tuned using grid search.

- Accuracy and F1 score metrics are computed for evaluation.

- Classification reports provide detailed performance metrics for each class.

- The results show that both OvR and OvO SVM models exhibit comparable accuracy and F1 scores, showcasing robust performance.

- SVM trained on undersampled data displays enhanced F1 scores for minority classes, albeit with a slight decrease in overall accuracy due to dataset reduction.



```
Accuracy (RBF Kernel) on ovr:  86.42
F1 on ovr: 0.8526058733503068
              precision    recall  f1-score   support

           1       0.80      0.85      0.83     21986
           2       0.83      0.76      0.79      3273
           3       0.41      0.17      0.24      3637
           4       0.42      0.21      0.28      1213
           5       0.29      0.11      0.16      1006
           6       0.74      0.91      0.82     10396
           7       0.99      0.99      0.99     37907
           8       0.93      0.89      0.91       111
          13       0.83      0.86      0.84      8368
          14       0.65      0.50      0.57      1081
         130       0.57      0.53      0.55       839
         140       0.61      0.52      0.56       183

    accuracy                           0.86     90000
   macro avg       0.67      0.61      0.63     90000
weighted avg       0.85      0.86      0.85     90000
```

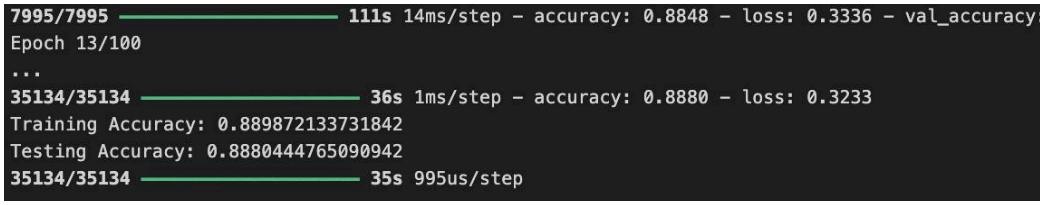
6- ANN:

The initial training of the Artificial Neural Network (ANN) on an extensive dataset exceeding 5 million rows necessitated a deep architecture with 4 hidden layers and numerous neurons. This configuration, coupled with 100 epochs and a batch size of 450, demanded substantial computational resources and over 2 hours of training time.

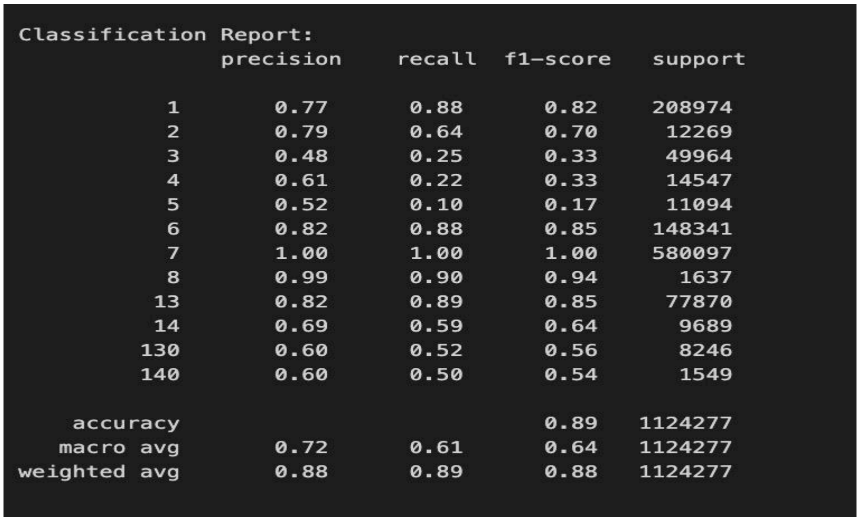
Despite the considerable computational burden, the model showcased promising results, boasting a training accuracy of 88.9% and a testing accuracy of 88.8%, with an average **F1 score of 88**, , and the loss function remained relatively low, around 0.3.

To combat overfitting, **early stopping** techniques were employed, ensuring the model's generalization ability and maintaining high accuracy levels while preventing excessive complexity. However, the model exhibited **imbalanced predictions**, with certain classes being favored over others, as indicated by an average class accuracy of 39%. **Despite this imbalance**, the **ROC curve** displayed **high ROC values** underscored its effectiveness in distinguishing between positive and negative instances across various threshold settings.

Attempts to **address the class imbalance using Random Undersampling improved** the **average class accuracy** but resulted in a trade-off, reducing overall accuracy to 78.2% for training and 77.7% for testing, with an F1 score of 75.



```
7995/7995 ━━━━━━━━━━━━━━━━━━━━ 111s 14ms/step - accuracy: 0.8848 - loss: 0.3336 - val_accuracy
Epoch 13/100
...
35134/35134 ━━━━━━━━━━━━━━━━━━━━ 36s 1ms/step - accuracy: 0.8880 - loss: 0.3233
Training Accuracy: 0.889872133731842
Testing Accuracy: 0.8880444765090942
35134/35134 ━━━━━━━━━━━━━━━━━━━━ 35s 995us/step
```



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

           1       0.77      0.88      0.82    208974
           2       0.79      0.64      0.70     12269
           3       0.48      0.25      0.33     49964
           4       0.61      0.22      0.33     14547
           5       0.52      0.10      0.17     11094
           6       0.82      0.88      0.85    148341
           7       1.00      1.00      1.00    580097
           8       0.99      0.90      0.94      1637
          13       0.82      0.89      0.85     77870
          14       0.69      0.59      0.64      9689
         130       0.60      0.52      0.56      8246
         140       0.60      0.50      0.54      1549

    accuracy                           0.89   1124277
   macro avg       0.72      0.61      0.64   1124277
weighted avg       0.88      0.89      0.88   1124277
```

COMPARATIVE ANALYSIS:

We choose the F1-score for model comparison in unbalanced datasets because it provides a balanced assessment by considering both precision and recall. Unlike traditional accuracy metrics, which can be misleading in unbalanced datasets, the F1-score evaluates how well the model identifies both positive and negative instances, especially for minority classes, making it a reliable metric for such scenarios.

Model	***F1-Score***
Decision-Tree	*85%*
Random Forest	*89%*
KNN	*89%*
SVM	*86%*
Naïve Bayes	*70%*
ANN	*88%*

In comparing the machine learning models, Random Forest and K-Nearest Neighbors (KNN) emerged as the top performers with F1-scores of 89%, demonstrating their robustness in handling diverse human activity data. Support Vector Machines (SVM) and Artificial Neural Networks (ANN) also showed strong performance, with F1-scores of 86% and 88%, respectively, highlighting their capability to capture complex patterns despite higher computational requirements. Decision Trees, after tuning, achieved an 85% F1-score but faced challenges with class imbalance. Naive Bayes, with a 70% F1-score, was the least effective, reflecting its limitation due to the assumption of feature independence. Overall, Random Forest and KNN provided the best balance of accuracy and efficiency, making them suitable for practical applications in human activity recognition.

Discussion:

The comparative analysis of machine learning algorithms on the Human Activity Recognition Trondheim (HARTH) dataset provided valuable insights into their performance and applicability. Random Forest and K-Nearest Neighbors (KNN) emerged as top performers with F1-scores of 89%, demonstrating robust classification capabilities. These models effectively handled the variability of human activity data, making them suitable for practical applications.

Support Vector Machines (SVM) and Artificial Neural Networks (ANN) also showed strong performance, with F1-scores of 86% and 88%, respectively. The RBF kernel in SVM provided a flexible decision boundary, improving accuracy but requiring extensive computational resources for tuning. ANN's deep architecture captured complex data patterns well, although it was computationally intensive and time-consuming to train, posing scalability challenges.

Decision Trees, with an F1-score of 85%, were hindered by overfitting and sensitivity to class imbalance, despite their interpretability and ease of use. Hyperparameter tuning and resampling techniques like SMOTE helped mitigate these issues but reduced overall accuracy. Naive Bayes performed the worst with an F1-score of 70%, due to its assumption of feature independence, which is not valid for correlated accelerometer data, since naive bayes does not give good results compared to the other algos such as random forest , ANN and KNN since it is assuming conditional independence which is not always the case especially in real life data like this.

Limitations and Potential Improvements:

Several limitations were noted. The high computational demands of SVM and ANN and Random Forest necessitate efficient resource management and optimization techniques. Future work could explore parallel processing or cloud-based solutions to address these challenges. Class imbalance remains a critical issue, particularly for Decision Trees and Naive Bayes. Advanced resampling techniques and ensemble methods should be further investigated to improve minority class representation.

The HARTH dataset's scope is limited to specific activities and sensor placements. Expanding it to include more diverse activities and additional sensors could enhance model generalizability. Incorporating temporal and contextual information could also improve feature richness and model performance.

Future Work:

Future research should focus on hybrid models that combine the strengths of multiple algorithms, such as integrating the interpretability of Decision Trees with the high accuracy of Random Forests or the deep learning capabilities of ANNs. Developing models capable of real-time processing is crucial for practical deployment in wearable devices or smartphones.Improving the transparency and interpretability of machine learning models is essential to foster user trust and facilitate adoption, especially in critical areas like healthcare and safety monitoring. By addressing these challenges and exploring innovative solutions, future research can advance the development of more accurate, efficient, and interpretable human activity recognition systems.

Conclusion:

This project conducted a comparative analysis of machine learning algorithms for human activity recognition using the HARTH dataset. Key findings highlight that Random Forest and K-Nearest Neighbors (KNN) were the most effective models, achieving F1-scores of 89%. These models demonstrated robust performance in classifying various human activities, making them suitable for real-world applications such as health monitoring and sports analytics.

Support Vector Machines (SVM) and Artificial Neural Networks (ANN) also performed well, with F1-scores of 86% and 88%, respectively. SVM's RBF kernel and ANN's deep architecture effectively captured complex patterns in the accelerometer data, despite their computational demands. Decision Trees showed good performance after hyperparameter tuning, achieving an F1-score of 85%, but struggled with class imbalance. Naive Bayes was the least effective, with a 70% F1-score, due to its assumption of feature independence.

The project's contributions lie in its detailed evaluation of these algorithms, providing insights into their strengths and limitations. This study guides the development of more accurate, efficient, and interpretable HAR systems, benefiting applications in healthcare, smart environments, and beyond. The work underscores the importance of selecting appropriate machine learning models based on the specific requirements of HAR tasks and lays the foundation for further advancements in this field.

Who did what in the project?

Student	*Work*
AFRA Hana	Naive Bayes SVM Report
DJAID Douaa	Pre-Processing Random forest KNN Report
CHEBOUI Fatma Imene	Decision Tree ANN Report