AI 361

Assignment 3

Group 2

Datasets used:

1- Estimation of Obesity Levels Based On Eating Habits and Physical Condition

2- SPECT Heart

3- ILPD (Indian Liver Patient Dataset)

4- AIDS Clinical Trials Group Study 175

Algorithms used:

Classification (Logistic Regression, Random Forest (Ensemble), Gradient Boosting (Ensemble), Support Vector Machine, Decision Tree, K-Neighbors Classifier, Ada Boost Classifier (Ensemble), Passive Aggressive Classifier, Stochastic Gradient Descent Classifier, Extra Trees Classifier)

Regression (Decision Tree Regression, Random Forest Regression, ElasticNet)

Clustering (Kmean, agglomerativeClustering, DBSCAN)

Sample run on some dataset for each algorithm: (the numbers next to each picture refers to the dataset used)

1- Estimation of Obesity Levels Based On Eating Habits and Physical Condition

2- SPECT Heart

3- ILPD (Indian Liver Patient Dataset)

4- AIDS Clinical Trials Group Study 175

Classification:

1.Logistic Regression:

1-

```
Results for Logistic Regression:
Best Parameters: {'classifier__C': 10, 'classifier__solver': 'lbfgs'}
Training Time: 3.1021292209625244
Accuracy: 0.9456264775413712
                    precision    recall  f1-score   support

Insufficient_Weight       0.90      1.00      0.95        56
      Normal_Weight       0.96      0.82      0.89        62
     Obesity_Type_I       0.97      0.97      0.97        78
    Obesity_Type_II       0.98      0.98      0.98        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.88      0.89      0.88        56
Overweight_Level_II       0.90      0.94      0.92        50

           accuracy                           0.95       423
          macro avg       0.94      0.94      0.94       423
       weighted avg       0.95      0.95      0.95       423



Fitting 5 folds for each of 24 candidates, totalling 120 fits
```

2-

```
Results for Logistic Regression:
Best Parameters: {'classifier__C': 0.1, 'classifier__solver': 'liblinear'}
Training Time: 3.6484482288360596
Accuracy: 0.7222222222222222
              precision    recall  f1-score   support

           0       0.20      0.22      0.21         9
           1       0.84      0.82      0.83        45

    accuracy                           0.72        54
   macro avg       0.52      0.52      0.52        54
weighted avg       0.73      0.72      0.73        54
```

2.Random Forest (Ensemble):

1-

```
Results for Random Forest:
Best Parameters: {'classifier__max_depth': 30, 'classifier__max_features': 'sqrt', 'classifier__n_estimators': 100}
Training Time: 3.4674222469329834
Accuracy: 0.9385342789598109
                     precision    recall  f1-score   support

Insufficient_Weight       1.00      0.96      0.98        56
      Normal_Weight       0.84      0.90      0.87        62
     Obesity_Type_I       0.99      0.94      0.96        78
    Obesity_Type_II       0.98      0.98      0.98        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.84      0.84      0.84        56
Overweight_Level_II       0.92      0.94      0.93        50

           accuracy                           0.94       423
          macro avg       0.94      0.94      0.94       423
       weighted avg       0.94      0.94      0.94       423



Fitting 5 folds for each of 18 candidates, totalling 90 fits
```

2-

```
Results for Random Forest:
Best Parameters: {'classifier__max_depth': None, 'classifier__max_features': 'sqrt', 'classifier__n_estimators': 100}
Training Time: 1.8468258380889893
Accuracy: 0.7777777777777778
              precision    recall  f1-score   support

           0       0.33      0.33      0.33         9
           1       0.87      0.87      0.87        45

    accuracy                           0.78        54
   macro avg       0.60      0.60      0.60        54
weighted avg       0.78      0.78      0.78        54
```

3.Gradient Boosting (Ensemble):

1-

```
Results for Gradient Boosting:
Best Parameters: {'classifier__learning_rate': 0.2, 'classifier__max_depth': 5, 'classifier__n_estimators': 200}
Training Time: 52.96963119506836
Accuracy: 0.9527186761229315
                    precision    recall  f1-score   support

Insufficient_Weight       0.93      0.96      0.95        56
      Normal_Weight       0.92      0.87      0.89        62
     Obesity_Type_I       0.99      0.94      0.96        78
    Obesity_Type_II       0.93      0.98      0.96        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.90      0.95      0.92        56
Overweight_Level_II       1.00      0.98      0.99        50

          accuracy                           0.95       423
         macro avg       0.95      0.95      0.95       423
      weighted avg       0.95      0.95      0.95       423



Fitting 5 folds for each of 6 candidates, totalling 30 fits
```

2-

```
Results for Gradient Boosting:
Best Parameters: {'classifier__learning_rate': 0.01, 'classifier__max_depth': 3, 'classifier__n_estimators': 200}
Training Time: 2.646685838699341
Accuracy: 0.7407407407407407
              precision    recall  f1-score   support

           0       0.22      0.22      0.22         9
           1       0.84      0.84      0.84        45

    accuracy                           0.74        54
   macro avg       0.53      0.53      0.53        54
weighted avg       0.74      0.74      0.74        54
```

4.Support Vector Machine:

1-

```
Results for Support Vector Machine:
Best Parameters: {'classifier__C': 10, 'classifier__kernel': 'linear'}
Training Time: 0.4617767333984375
Accuracy: 0.9787234042553191
                    precision    recall  f1-score   support

Insufficient_Weight       0.98      1.00      0.99        56
      Normal_Weight       0.98      0.95      0.97        62
     Obesity_Type_I       0.96      1.00      0.98        78
    Obesity_Type_II       1.00      1.00      1.00        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.95      0.96      0.96        56
Overweight_Level_II       0.98      0.92      0.95        50

           accuracy                           0.98       423
          macro avg       0.98      0.98      0.98       423
       weighted avg       0.98      0.98      0.98       423



Fitting 5 folds for each of 12 candidates, totalling 60 fits
```

2-

```
  Results for Support Vector Machine:
  Best Parameters: {'classifier__C': 1, 'classifier__kernel': 'rbf'}
  Training Time: 0.06931304931640625
  Accuracy: 0.7592592592592593
                precision    recall  f1-score   support

             0       0.25      0.22      0.24         9
             1       0.85      0.87      0.86        45

      accuracy                           0.76        54
     macro avg       0.55      0.54      0.55        54
  weighted avg       0.75      0.76      0.75        54
```

5.Decision Tree:

1-

```
Results for Decision Tree:
Best Parameters: {'classifier__max_depth': None, 'classifier__min_samples_split': 2}
Training Time: 0.22823452949523926
Accuracy: 0.9456264775413712
                    precision    recall  f1-score   support

Insufficient_Weight       0.90      0.98      0.94        56
      Normal_Weight       0.90      0.85      0.88        62
     Obesity_Type_I       0.96      0.96      0.96        78
    Obesity_Type_II       0.96      0.95      0.96        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.91      0.91      0.91        56
Overweight_Level_II       0.98      0.96      0.97        50

           accuracy                           0.95       423
          macro avg       0.95      0.95      0.94       423
       weighted avg       0.95      0.95      0.95       423
```

2-


```
Results for Decision Tree:
Best Parameters: {'classifier__max_depth': None, 'classifier__min_samples_split': 2}
Training Time: 0.22823452949523926
Accuracy: 0.9456264775413712
                    precision    recall  f1-score   support

Insufficient_Weight       0.90      0.98      0.94        56
      Normal_Weight       0.90      0.85      0.88        62
     Obesity_Type_I       0.96      0.96      0.96        78
    Obesity_Type_II       0.96      0.95      0.96        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.91      0.91      0.91        56
Overweight_Level_II       0.98      0.96      0.97        50

           accuracy                           0.95       423
          macro avg       0.95      0.95      0.94       423
       weighted avg       0.95      0.95      0.95       423
```


6. K-Neighbors Classifier:

1-

```
Results for K-Nearest Neighbors:
Best Parameters: {'classifier__n_neighbors': 3, 'classifier__weights': 'distance'}
Training Time: 0.10709786415100098
Accuracy: 0.8534278959810875
                    precision    recall  f1-score   support

Insufficient_Weight       0.84      0.95      0.89        56
      Normal_Weight       0.86      0.40      0.55        62
     Obesity_Type_I       0.86      0.94      0.90        78
    Obesity_Type_II       0.90      0.98      0.94        58
   Obesity_Type_III       0.98      1.00      0.99        63
 Overweight_Level_I       0.78      0.89      0.83        56
Overweight_Level_II       0.73      0.80      0.76        50

           accuracy                           0.85       423
          macro avg       0.85      0.85      0.84       423
       weighted avg       0.86      0.85      0.84       423
```

2-

```
Results for K-Nearest Neighbors:
Best Parameters: {'classifier__n_neighbors': 7, 'classifier__weights': 'uniform'}
Training Time: 0.06806182861328125
Accuracy: 0.7222222222222222
              precision    recall  f1-score   support

           0       0.20      0.22      0.21         9
           1       0.84      0.82      0.83        45

    accuracy                           0.72        54
   macro avg       0.52      0.52      0.52        54
weighted avg       0.73      0.72      0.73        54
```

7. Ada Boost Classifier (Ensemble):

1-

```
Results for AdaBoost:
Best Parameters: {'classifier__learning_rate': 1, 'classifier__n_estimators': 50}
Training Time: 2.1931943893432617
Accuracy: 0.2695035460992908
                    precision    recall  f1-score   support

Insufficient_Weight       1.00      0.05      0.10        56
      Normal_Weight       0.35      0.60      0.44        62
     Obesity_Type_I       0.16      0.27      0.20        78
    Obesity_Type_II       0.56      0.09      0.15        58
   Obesity_Type_III       0.00      0.00      0.00        63
 Overweight_Level_I       0.29      0.75      0.42        56
Overweight_Level_II       0.22      0.12      0.16        50

           accuracy                           0.27       423
          macro avg       0.37      0.27      0.21       423
       weighted avg       0.35      0.27      0.21       423
```

2-

```
Results for AdaBoost:
Best Parameters: {'classifier__learning_rate': 0.1, 'classifier__n_estimators': 100}
Training Time: 0.7686986923217773
Accuracy: 0.7962962962962963
              precision    recall  f1-score   support

           0       0.38      0.33      0.35         9
           1       0.87      0.89      0.88        45

    accuracy                           0.80        54
   macro avg       0.62      0.61      0.62        54
weighted avg       0.79      0.80      0.79        54
```

8. Passive Aggressive Classifier:

1-

```
Results for PAC:
Best Parameters: {'classifier__C': 0.01, 'classifier__loss': 'squared_hinge'}
Training Time: 0.17616009712219238
Accuracy: 0.6879432624113475
                    precision    recall  f1-score   support

Insufficient_Weight       0.85      1.00      0.92        56
      Normal_Weight       0.71      0.40      0.52        62
     Obesity_Type_I       0.70      0.40      0.51        78
    Obesity_Type_II       0.76      0.97      0.85        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.41      0.79      0.54        56
Overweight_Level_II       0.48      0.32      0.39        50

           accuracy                           0.69       423
          macro avg       0.70      0.70      0.67       423
       weighted avg       0.71      0.69      0.67       423
```

2-

```
Results for PAC:
Best Parameters: {'classifier__C': 0.01, 'classifier__loss': 'squared_hinge'}
Training Time: 0.059052467346191406
Accuracy: 0.7592592592592593
              precision    recall  f1-score   support

           0       0.33      0.44      0.38         9
           1       0.88      0.82      0.85        45

    accuracy                           0.76        54
   macro avg       0.61      0.63      0.62        54
weighted avg       0.79      0.76      0.77        54
```

9. Stochastic Gradient Descent Classifier:

1-

```
Results for SGD:
Best Parameters: {'classifier__alpha': 0.01, 'classifier__loss': 'modified_huber', 'classifier__penalty': 'elasticnet'}
Training Time: 0.44444704055786133
Accuracy: 0.7186761229314421
                    precision    recall  f1-score   support

Insufficient_Weight       0.80      0.98      0.88        56
      Normal_Weight       0.68      0.42      0.52        62
     Obesity_Type_I       0.64      0.69      0.67        78
    Obesity_Type_II       0.80      0.97      0.88        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.65      0.39      0.49        56
Overweight_Level_II       0.43      0.56      0.49        50

          accuracy                           0.72       423
         macro avg       0.71      0.72      0.70       423
      weighted avg       0.72      0.72      0.71       423
```

2-

```
Results for SGD:
Best Parameters: {'classifier__alpha': 0.01, 'classifier__loss': 'hinge', 'classifier__penalty': 'elasticnet'}
Training Time: 0.14285707473754883
Accuracy: 0.7407407407407407
              precision    recall  f1-score   support

           0       0.27      0.33      0.30         9
           1       0.86      0.82      0.84        45

    accuracy                           0.74        54
   macro avg       0.57      0.58      0.57        54
weighted avg       0.76      0.74      0.75        54
```

10. Extra Trees Classifier:

1-

```
Results for ETC:
Best Parameters: {'classifier__max_depth': None, 'classifier__max_features': 'sqrt', 'classifier__n_estimators': 100}
Training Time: 2.195096254348755
Accuracy: 0.9148936170212766
                    precision    recall  f1-score   support

Insufficient_Weight       0.96      0.96      0.96        56
      Normal_Weight       0.79      0.84      0.81        62
     Obesity_Type_I       0.91      0.92      0.92        78
    Obesity_Type_II       1.00      0.98      0.99        58
   Obesity_Type_III       1.00      1.00      1.00        63
 Overweight_Level_I       0.83      0.80      0.82        56
Overweight_Level_II       0.92      0.88      0.90        50

          accuracy                           0.91       423
         macro avg       0.92      0.91      0.91       423
      weighted avg       0.92      0.91      0.92       423
```

2-

```
Results for ETC:
Best Parameters: {'classifier__max_depth': 10, 'classifier__max_features': 'log2', 'classifier__n_estimators': 100}
Training Time: 1.0753378868103027
Accuracy: 0.7592592592592593
              precision    recall  f1-score   support

           0       0.30      0.33      0.32         9
           1       0.86      0.84      0.85        45

    accuracy                           0.76        54
   macro avg       0.58      0.59      0.58        54
weighted avg       0.77      0.76      0.76        54
```

Clustering:

Kmean:

2-

```
Results for KMeans:
Best Parameters: {'algorithm': 'lloyd', 'copy_x': True, 'init': 'k-means++', 'max_iter': 300, 'n_clusters': 8, 'n_init': 'aut
o', 'random_state': None, 'tol': 0.0001, 'verbose': 0}
Silhouette Score: 0.1531924404367866
Davies-Bouldin Score: 2.0908216852641432
```

agglomerativeClustering:

2-

```
Results for Agglomerative Clustering:
Best Parameters: {'compute_distances': False, 'compute_full_tree': 'auto', 'connectivity': None, 'distance
_threshold': None, 'linkage': 'ward', 'memory': None, 'metric': 'euclidean', 'n_clusters': 2}
Silhouette Score: 0.17298782262114518
Davies-Bouldin Score: 2.5308420244226433
```

DBSCAN :

2-

```
Results for DBSCAN:
Best Parameters: {'algorithm': 'auto', 'eps': 0.5, 'leaf_size': 30, 'metric': 'euclidean', 'metric_param
s': None, 'min_samples': 5, 'n_jobs': None, 'p': None}
Silhouette Score: -0.03692305277621896
Davies-Bouldin Score: 1.2878481751594684
```

Regression:

Decision Tree Regression

2-

```
You have chosen Decision Tree Regression
Results for Decision Tree Regression:
Training Time: 0.005003690719604492
Mean Squared Error (MSE): 0.27195473251028807
Mean Absolute Error (MAE): 0.31296296296296294
R² Score: -0.9580740740740741
```

Random Forest Regression

2-

```
You have chosen Random Forest Regression
Results for Random Forest Regression:
Training Time: 0.09308457374572754
Mean Squared Error (MSE): 0.19360769344340814
Mean Absolute Error (MAE): 0.29403935555604704
R² Score: -0.39397539279253846
```

Elastic Net

2-

```
You have chosen ElasticNet
Results for ElasticNet:
Training Time: 0.005004405975341797
Mean Squared Error (MSE): 0.14131896228702415
Mean Absolute Error (MAE): 0.31064162754303587
R² Score: -0.01749652846657379
```

Validation Process:

Hold-out method was used for Validation process

```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Train 80% & Test 20%

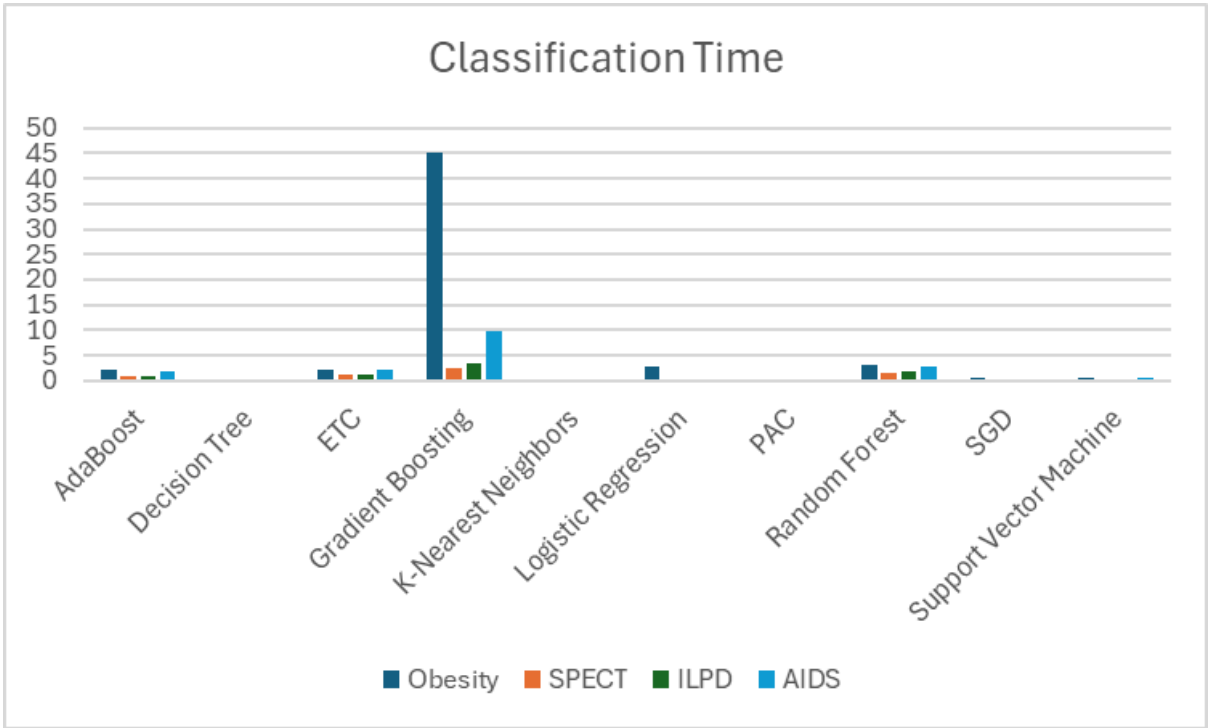
Hyperparameter Tuning:

We used GridSearch for choosing the best hyperparameter out of some defined choices. The best hyperparameter is listed under each sample run.

Best model performance & Training times For Classification

Performance:

Time:



Preprocessing: we split the data into 2 columns, one with numerical data and the other with objects (like strings). Then we filled the missing values (if any) with appropriate values. If it was a missing numerical then it will be filled with the mean of that attribute from other examples, if it was an object then it will be filled with the one with higher frequency.

In conclusion, some algorithms take a huge amount of time compared to others, like gradient boosting which takes about 10 times the amount of time of other algorithms without substantial improvements. Also, the efficiency of an algorithm mostly depends on the dataset. For example, the AdaBoost algorithm is the worst one in terms of performance for the Obesity dataset, but the best algorithm for the AIDS dataset. We also noticed that adjusting hyperparameters may also improve the algorithm's performance by a decent amount of time (which our grid search does automatically).

As a last note, all algorithms we used were imported from scikit-learn libraries and we used jupyter as a programming environment.