



Department of Computer Science & Engineering

Machine Learning

UE22CS352A

Orange Program

*Name: Mohammed Hassan*

*Srn: PES2UG22CS317*

*Date: 30/09/2024*

### **1. Fine-Tuned Model Parameters:**

'hidden\_layer\_sizes': [(128,64,64,32),(64,64,32,32)],

'activation': ['tanh', 'relu'],

'solver': ['sgd', 'adam'],

'alpha': [0.0001, 0.05],

'learning\_rate': ['constant','adaptive'],

'learning\_rate\_init': [0.001, 0.01, 0.1]

This dictionary defines the hyperparameters to be tuned and their possible values:

- hidden\_layer\_sizes: Different configurations of neurons in the hidden layers.
- activation: Activation functions to be used.
- solver: Optimization algorithms.
- alpha: Regularization parameter.
- learning\_rate: Learning rate schedule.
- learning\_rate\_init: Initial learning rate.

## 2. Analysis of Metrics:

### Original Model Performance:

	precision	recall	f1-score	support
0	0.65	0.65	0.65	34
1	0.79	0.69	0.74	45
2	0.64	0.66	0.65	38
3	0.71	0.71	0.71	38
4	0.58	0.50	0.54	42
5	0.80	0.84	0.82	43
6	0.45	0.61	0.52	28
7	0.67	0.65	0.66	40
accuracy			0.67	308
macro avg	0.66	0.66	0.66	308
weighted avg	0.67	0.67	0.67	308

### Confusion Matrix:

```
[[22  0  6  1  1  1  0  3]
 [ 0 31  2  0  0  4  8  0]
 [ 2  1 25  2  4  1  1  2]
 [ 0  1  1 27  5  0  4  0]
 [ 7  0  3  3 21  0  5  3]
 [ 0  4  0  0  0 36  2  1]
 [ 0  2  0  1  1  3 17  4]
 [ 3  0  2  4  4  0  1 26]]
```

### Best Model Performance:

	precision	recall	f1-score	support
0	0.68	0.79	0.73	34
1	0.80	0.71	0.75	45
2	0.61	0.61	0.61	38
3	0.65	0.74	0.69	38
4	0.63	0.45	0.53	42
5	0.67	0.81	0.74	43
6	0.56	0.64	0.60	28
7	0.61	0.50	0.55	40
accuracy			0.66	308
macro avg	0.65	0.66	0.65	308
weighted avg	0.66	0.66	0.65	308

```
Confusion Matrix:
[[27  0  5  0  0  0  0  2]
 [ 0 32  1  1  0  7  4  0]
 [ 1  0 23  3  3  0  3  5]
 [ 2  1  1 28  3  0  3  0]
 [ 5  1  2  6 19  4  2  3]
 [ 0  5  2  0  0 35  1  0]
 [ 0  1  0  2  1  3 18  3]
 [ 5  0  4  3  4  3  1 20]]
```

### 3. K-Fold Cross Validation:

```
from sklearn.model_selection import cross_val_score

# Evaluate the original model using 5-fold cross-validation
cv_scores_original = cross_val_score(model_mlp, X_train, y_train, cv=5,
scoring='accuracy')
print(f"Original Model Cross-Validation Scores: {cv_scores_original}")
print(f"Original Model Average Cross-Validation Score:
{np.mean(cv_scores_original):.2f}")

# Evaluate the best model using 5-fold cross-validation
cv_scores_best = cross_val_score(best_mlp_model, X_train, y_train, cv=5,
scoring='accuracy')
print(f"\nBest Model Cross-Validation Scores: {cv_scores_best}")
print(f"Best Model Average Cross-Validation Score: {np.mean(cv_scores_best):.2f}")
```

```
Original Model Cross-Validation Scores: [0.60162602 0.63821138 0.58536585 0.5877551  0.54285714]
Original Model Average Cross-Validation Score: 0.59

Best Model Cross-Validation Scores: [0.58943089 0.63821138 0.60569106 0.58367347 0.57959184]
Best Model Average Cross-Validation Score: 0.60
```

- i. Cross-Validation: A technique to evaluate the performance of a model by splitting the data into multiple folds and training/testing the model on these folds.
- ii. 5-Fold Cross-Validation: The data is split into 5 parts, and the model is trained on 4 parts and tested on the remaining part. This process is repeated 5 times, each time with a different part as the test set.
- iii. Accuracy: The metric used to evaluate the model's performance.
- iv. Original Model: The initial MLP model before hyperparameter tuning.

- v. Best Model: The MLP model with the best hyperparameters found using Grid Search.
4. **Comparative Study:** comparative study of model performance with respect to SVM and KNN.

#### Classification Reports

- SVM Model:
  - Precision, recall, F1-score, and support for each class.
  - Overall accuracy, macro average, and weighted average.
- KNN Model:
  - Precision, recall, F1-score, and support for each class.
  - Overall accuracy, macro average, and weighted average.

#### Confusion Matrices

- SVM Model:
  - Shows the number of correct and incorrect predictions for each class.
  - Helps identify which classes are being misclassified.
- KNN Model:
  - Shows the number of correct and incorrect predictions for each class.
  - Helps identify which classes are being misclassified.

#### Cross-Validation Scores

- SVM Model:
  - Cross-validation scores for each fold.
  - Average cross-validation score.
- KNN Model:
  - Cross-validation scores for each fold.
  - Average cross-validation score.

SVM Model Performance:				
	precision	recall	f1-score	support
0	0.42	0.29	0.34	34
1	0.78	0.87	0.82	45
2	0.60	0.68	0.64	38
3	0.43	0.63	0.51	38
4	0.60	0.43	0.50	42
5	0.79	0.86	0.82	43
6	0.56	0.50	0.53	28
7	0.64	0.53	0.58	40
accuracy			0.61	308
macro avg	0.60	0.60	0.59	308
weighted avg	0.61	0.61	0.61	308

Confusion Matrix:

```
[[10  1  5 16  1  0  0  1]
 [ 0 39  1  0  0  3  2  0]
 [ 2  0 26  3  4  0  0  3]
 [ 2  2  2 24  3  1  4  0]
 [ 6  1  3  5 18  3  2  4]
 [ 0  3  0  0  0 37  2  1]
 [ 0  4  0  3  2  2 14  3]
 [ 4  0  6  5  2  1  1 21]]
```

KNN Model Performance:

	precision	recall	f1-score	support
0	0.67	0.65	0.66	34
1	0.79	0.67	0.72	45
2	0.63	0.68	0.66	38
3	0.60	0.55	0.58	38
4	0.64	0.43	0.51	42
5	0.53	0.86	0.65	43
6	0.48	0.43	0.45	28
7	0.58	0.55	0.56	40
accuracy			0.61	308
macro avg	0.62	0.60	0.60	308
weighted avg	0.62	0.61	0.61	308

Confusion Matrix:

```
[[22  0  2  2  2  2  1  3]
 [ 0 30  1  0  1 10  3  0]
 [ 2  1 26  3  0  4  0  2]
 [ 2  1  1 21  4  2  3  4]
 [ 1  0  6  3 18  7  3  4]
 [ 0  3  1  1  0 37  1  0]
 [ 1  3  1  2  1  5 12  3]
 [ 5  0  3  3  2  3  2 22]]
```

SVM Model Cross-Validation Scores: [0.57723577 0.62601626 0.57317073 0.5755102 0.5877551 ]  
SVM Model Average Cross-Validation Score: 0.59

KNN Model Cross-Validation Scores: [0.50813008 0.55691057 0.54065041 0.59183673 0.53469388]  
KNN Model Average Cross-Validation Score: 0.55