UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING SEMESTER 2,

SESION 2023/2024

ASSIGNMENT 2 (HYPERPARAMETER)

SECB4313 : BIOINFORMATIK MODEL DAN SIMULASI

SECTION 1

NAME	METRIC NO
MUHAMMAD DANISH BIN TAZURIN	B21EC0035
AMZAR RAZIQ KHAN BIN AZAM KHAN	B21EC0011

SUBMISSION DATE : 8 JUNE 2024

LECTURER'S NAME : DR AZURAH BTE A SAMAH

1. Identify 4 hyperparameters and propose two values for each hyperparameter. Justify the selection of the hyperparameters and its corresponding values.

1. **Number of Neurons in Hidden Layers**:

   Proposed Values: 64 and 128

   Justification:

- The learning of the model is influenced by the quantity of neurons in hidden layers. Less neurons (64) reduce costs and prevent overfitting, but they may overlook intricate patterns. In order to learn more information, more neurons (128) run risk of overfitting and increased costs. These are only common beginning points that can be changed according with the data.

2. **Learning Rate for the Optimizer**:

   Proposed Values: 0.01 and 0.1

   Justification:

- The amount that the model changes with each update is determined by the learning rate. Precise adjustments are made at a lower rate (0.01), which increases accuracy but requires more training time. While learning is accelerated at a higher rate (0.1), the optimal answer may be overshot. These numbers are typical beginning points and strike a balance between too slow (0.001) and too fast (1.0) modifications.

3. **Number of Epochs**:

   Proposed Values: 50 and 100

   Justification:

- The number of training sessions on the full dataset is determined by the number of epochs. Shorter training sessions with fewer epochs (50) assist prevent overfitting, while longer training sessions (100) can enhance performance in cases when overfitting is not an issue. These parameters are based on real-world experience and strike a balance between computational expense and training adequacy.

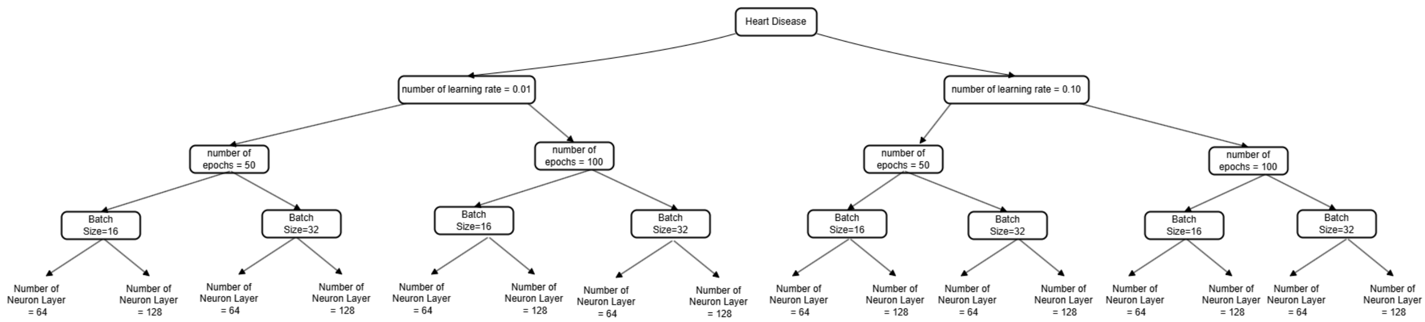
4. **Batch Size:**

Proposed Values: 16 and 32

Justification:

- The batch size is the number of samples processed before updating the model. Smaller batches (16) offer frequent, noisy updates that help avoid local minima, while larger batches (32) provide stable updates but need more memory. These sizes balance efficiency and stability.

2. <u>Construct a tree diagram that display the proposed hyperparameters and its corresponding values.</u>



3. <u>Tabulate the proposed experimental design based on your tree diagram</u>

Below is the table which consists of the experiments for which have different combinations of hyperparameters.

Experiment	Learning Rate	Number of Epochs	Batch Size	Neurons in Hidden Layer
0	0.01	50	16	64
1	0.01	50	16	128
2	0.01	50	32	64
3	0.01	50	32	128
4	0.01	100	16	64
5	0.01	100	16	128
6	0.01	100	32	64
7	0.01	100	32	128
8	0.10	50	16	64
9	0.10	50	16	128

10	0.10	50	32	64
11	0.10	50	32	128
12	0.10	100	16	64
13	0.10	100	16	128
14	0.10	100	32	64
15	0.10	100	32	128

4. Perform hyperparameter tuning to improve current result. (Simulate the model and collect the results)

The code for optimising the hyperparameters, which prints the model's final test accuracy, best score, and optimal hyperparameter, is provided below.

```python
for lr in learning_rates:
    for epoch in epochs:
        for batch_size in batch_sizes:
            for neuron in neurons:
                # Define the Random Forest model with mapped hyperparameters
                model = RandomForestClassifier(
                    n_estimators=100,
                    max_depth=epoch,
                    min_samples_split=batch_size,
                    min_samples_leaf=neuron,
                    random_state=42
                )

                # Train the model
                model.fit(X_train, y_train)

                # Evaluate the model
                y_pred = model.predict(X_test)
                accuracy = accuracy_score(y_test, y_pred)

                # Record the results
                results.append({
                    'learning_rate': lr,
                    'epochs': epoch,
                    'batch_size': batch_size,
                    'neurons': neuron,
                    'accuracy': accuracy
                })


results_df = pd.DataFrame(results)
print(results_df)
```

5. <u>Tabulate the results based on your proposed experimental design</u>

The results of the experiments carried out using the code above are displayed in the table below. The performance metric for every model was its accuracy score. The ideal hyperparameters are found in the model with the highest accuracy.

	learning_rate	epochs	batch_size	neurons	accuracy
0	0.01	50	16	64	0.836066
1	0.01	50	16	128	0.819672
2	0.01	50	32	64	0.819672
3	0.01	50	32	128	0.819672
4	0.01	100	16	64	0.885246
5	0.01	100	16	128	0.868852
6	0.01	100	32	64	0.836066
7	0.01	100	32	128	0.885246
8	0.10	50	16	64	0.836066
9	0.10	50	16	128	0.524590
10	0.10	50	32	64	0.524590
11	0.10	50	32	128	0.852459
12	0.10	100	16	64	0.524590
13	0.10	100	16	128	0.524590
14	0.10	100	32	64	0.885246
15	0.10	100	32	128	0.754098

Index	Learning Rate	Epochs	Batch Size	Neurons	Accuracy
0	0.01	50	16	64	0.836066
1	0.01	50	16	128	0.819672
2	0.01	50	32	64	0.819672
3	0.01	50	32	128	0.819672
4	0.01	100	16	64	0.885246
5	0.01	100	16	128	0.868852
6	0.01	100	32	64	0.836066
7	0.01	100	32	128	0.885246
8	0.10	50	16	64	0.836066
9	0.10	50	16	128	0.524590
10	0.10	50	32	64	0.524590
11	0.10	50	32	128	0.852459
12	0.10	100	16	64	0.524590

13	0.10	100	16	128	0.524590
14	0.10	100	32	64	0.885246
15	0.10	100	32	128	0.754098

```
Hyperparameter Analysis Combination Result:
learning_rate      0.010000
epochs           100.000000
batch_size        16.000000
neurons           64.000000
accuracy           0.885246
```

6. Analyze the results. Which combination of hyperparameters generate most improved result?

The hyperparameter study indicates that 100 epochs and a learning rate of 0.01 yield the best results. Combinations of a batch size of 16 and 64 neurons, a batch size of 32 and 128 neurons, and, most significantly, a higher learning rate of 0.10 with 100 epochs and a batch size of 32 and 64 neurons yield the best accuracy (0.885246). This suggests that batch size and the number of neurons can be efficiently changed within these ideal settings to maximise accuracy, even though a lower learning rate and a greater number of epochs are essential for performance improvement. These results emphasise how crucial it is to perform a thorough hyperparameter tuning in order to determine the ideal model configuration.