Sub-questions for Problem 2

1. **Why did you choose the particular algorithm?**

I chose a neural network for its ability to handle complex and non-linear relationships within the data. Neural networks, especially deep learning models implemented using Keras, excel at capturing intricate patterns that might be challenging for traditional machine learning algorithms. Their adaptability and capacity to learn hierarchical representations make them well-suited for tasks where the underlying data structure is intricate and not easily modeled by simpler algorithms.

2. **What are the different tuning methods used for the algorithm?**

The model is trained with 20 epochs and uses the Adam optimizer with a binary cross-entropy loss function. Tuning methods could include adjusting the learning rate, exploring different architectures (changing the number of layers, units per layer), experimenting with activation functions, or employing regularization techniques like dropout.

3. **Did you consider any other choice of algorithm?Why or why not?**

In selecting an algorithm for our classification task, I considered multiple options. I opted for a neural network using Keras due to its capability to handle intricate patterns and non-linear relationships within the data. Neural networks excel in capturing complex structures that might be challenging for other algorithms.

Other considered algorithms include:
Random Forest, Support Vector Machines (SVM), Gradient Boosting (e.g., XGBoost)

Ultimately, the neural network was chosen for its ability to adapt to complex data patterns and capture non-linear relationships effectively, aligning with the nature of the classification task.

4. **What is the accuracy?**

The Performance function provided in the code prints various metrics, including accuracy, which can be used to assess model performance.

Accuracy: 98%

```
F1 Score: 0.9767441860465117
Kappa Score: 0.965362438968315
Accuracy Score: 0.9815478777343184
Confusion Matrix:
[[1520   14]
 [  21  735]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1534
           1       0.98      0.97      0.98       756

    accuracy                           0.98      2290
   macro avg       0.98      0.98      0.98      2290
weighted avg       0.98      0.98      0.98      2290
```

5. **What are the different types of metrics that can be used to evaluate the model?**

The code uses several metrics to evaluate the model, including:

- **F1 Score:** A balance between precision and recall, especially useful when dealing with imbalanced datasets.
- **Kappa Score**: Measures agreement between predicted and actual classifications, accounting for chance agreement.
- **Accuracy:** The ratio of correctly predicted instances to the total number of instances.
- **Balanced Accuracy Score:** Similar to accuracy but considers imbalanced datasets.
- **Confusion Matrix:** A table showing the number of true positive, true negative, false positive, and false negative predictions.
- **Classification Report:** Provides precision, recall, F1 score, and support for each class.

The choice of metrics depends on the specific goals of the analysis and the nature of the classification problem.