Binary Classification:

In [1]:
```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Generate random data for binary classification
X = np.random.rand(1000, 10)
y = np.random.randint(2, size=(1000,))

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build the binary classification model
model_binary = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.2),
    Dense(32, activation='sigmoid'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])

# Compile the model with different optimizer and loss function
model_binary.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model_binary.fit(X_train, y_train, epochs=10, batch_size=32)

# Evaluate the model
accuracy = model_binary.evaluate(X_test, y_test)[1]
print("Binary Classification Accuracy: {:.2f}%".format(accuracy * 100))
```

```
Epoch 1/10
25/25 [==============================] - 1s 4ms/step - loss: 0.7030 - accuracy: 0.5288
Epoch 2/10
25/25 [==============================] - 0s 2ms/step - loss: 0.6976 - accuracy: 0.5238
Epoch 3/10
25/25 [==============================] - 0s 2ms/step - loss: 0.7150 - accuracy: 0.5300
Epoch 4/10
25/25 [==============================] - 0s 3ms/step - loss: 0.7101 - accuracy: 0.5050
Epoch 5/10
25/25 [==============================] - 0s 3ms/step - loss: 0.7066 - accuracy: 0.5337
Epoch 6/10
25/25 [==============================] - 0s 3ms/step - loss: 0.6977 - accuracy: 0.5188
Epoch 7/10
25/25 [==============================] - 0s 2ms/step - loss: 0.7046 - accuracy: 0.5188
Epoch 8/10
25/25 [==============================] - 0s 3ms/step - loss: 0.6971 - accuracy: 0.5200
Epoch 9/10
25/25 [==============================] - 0s 3ms/step - loss: 0.6894 - accuracy: 0.5300
Epoch 10/10
25/25 [==============================] - 0s 3ms/step - loss: 0.6894 - accuracy: 0.5263
7/7 [==============================] - 0s 2ms/step - loss: 0.6847 - accuracy: 0.5350
Binary Classification Accuracy: 53.50%
```

Regression:

In [2]:
```
# Generate random data for regression
X_reg = np.random.rand(1000, 10)
y_reg = np.random.rand(1000, 1)

# Split data into training and testing sets for regression
X_train_reg, X_test_reg, y_train_reg, y_test_reg = train_test_split(X_reg, y_reg, test_size=0.2, random_state=4

# Build the regression model
model_reg = Sequential([
    Dense(64, activation='relu', input_shape=(X_train_reg.shape[1],)),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='linear')
])

# Compile the model with different optimizer and loss function for regression
model_reg.compile(optimizer='adam', loss='mean_squared_error')

# Train the regression model
model_reg.fit(X_train_reg, y_train_reg, epochs=10, batch_size=32)

# Evaluate the regression model
mse = model_reg.evaluate(X_test_reg, y_test_reg)
print("Mean Squared Error for Regression: {:.4f}".format(mse))
```

```
Epoch 1/10
25/25 [==============================] - 1s 5ms/step - loss: 0.2748
Epoch 2/10
25/25 [==============================] - 0s 1ms/step - loss: 0.1255
Epoch 3/10
25/25 [==============================] - 0s 1ms/step - loss: 0.1135
Epoch 4/10
25/25 [==============================] - 0s 2ms/step - loss: 0.1054
Epoch 5/10
25/25 [==============================] - 0s 2ms/step - loss: 0.1013
Epoch 6/10
25/25 [==============================] - 0s 1ms/step - loss: 0.1033
Epoch 7/10
25/25 [==============================] - 0s 2ms/step - loss: 0.0979
Epoch 8/10
25/25 [==============================] - 0s 2ms/step - loss: 0.0976
Epoch 9/10
25/25 [==============================] - 0s 2ms/step - loss: 0.0930
Epoch 10/10
25/25 [==============================] - 0s 1ms/step - loss: 0.0985
7/7 [==============================] - 0s 3ms/step - loss: 0.0920
Mean Squared Error for Regression: 0.0920
```