1. Develop a program to build and train a **Feedforward Neural Network** from scratch using a deep learning framework like TensorFlow, keras etc.

Binary Classification on a simple dummy dataset, generated using `make_classification`.

In [ ]:

```
import tensorflow as tf
from tensorflow.keras import Sequential # type: ignore
from tensorflow.keras.layers import Dense, Input  # type: ignore
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

In [3]:

```
X, y = make_classification(
    n_samples=1000,
    n_features=20,
    n_informative=15,
    n_redundant=5,
    random_state=42
)
```

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [8]:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [11]:

```
model = Sequential([
    Input(shape=(X_train.shape[1],)),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

In [12]:

```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

In [13]:

```
history = model.fit(
    X_train, y_train,
    epochs=20,
    batch_size=32,
    validation_split=0.2,
    verbose=1
)
```

```
Epoch 1/20
18/18 [==============================] - 1s 9ms/step - loss: 0.6666 - accuracy: 0.5661 - val_l
oss: 0.6104 - val_accuracy: 0.7071
Epoch 2/20
18/18 [==============================] - 0s 2ms/step - loss: 0.5553 - accuracy: 0.8143 - val_l
oss: 0.5299 - val_accuracy: 0.8143
Epoch 3/20
18/18 [==============================] - 0s 2ms/step - loss: 0.4656 - accuracy: 0.8696 - val_l
oss: 0.4564 - val_accuracy: 0.8714
Epoch 4/20
18/18 [==============================] - 0s 2ms/step - loss: 0.3848 - accuracy: 0.8839 - val_l
oss: 0.3959 - val_accuracy: 0.8857
Epoch 5/20
18/18 [==============================] - 0s 2ms/step - loss: 0.3193 - accuracy: 0.9054 - val_l
oss: 0.3573 - val_accuracy: 0.8857
Epoch 6/20
18/18 [==============================] - 0s 2ms/step - loss: 0.2700 - accuracy: 0.9179 - val_l
oss: 0.3248 - val_accuracy: 0.9000
Epoch 7/20
18/18 [==============================] - 0s 2ms/step - loss: 0.2355 - accuracy: 0.9268 - val_l
oss: 0.2962 - val_accuracy: 0.8929
Epoch 8/20
18/18 [==============================] - 0s 2ms/step - loss: 0.2058 - accuracy: 0.9357 - val_l
oss: 0.2765 - val_accuracy: 0.9071
Epoch 9/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1834 - accuracy: 0.9411 - val_l
oss: 0.2608 - val_accuracy: 0.9071
Epoch 10/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1663 - accuracy: 0.9464 - val_l
oss: 0.2525 - val_accuracy: 0.9071
Epoch 11/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1523 - accuracy: 0.9518 - val_l
oss: 0.2514 - val_accuracy: 0.9071
Epoch 12/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1386 - accuracy: 0.9571 - val_l
oss: 0.2410 - val_accuracy: 0.9143
Epoch 13/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1269 - accuracy: 0.9625 - val_l
oss: 0.2376 - val_accuracy: 0.9071
Epoch 14/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1177 - accuracy: 0.9571 - val_l
oss: 0.2364 - val_accuracy: 0.9214
Epoch 15/20
18/18 [==============================] - 0s 3ms/step - loss: 0.1097 - accuracy: 0.9643 - val_l
oss: 0.2284 - val_accuracy: 0.9143
Epoch 16/20
18/18 [==============================] - 0s 2ms/step - loss: 0.1006 - accuracy: 0.9750 - val_l
oss: 0.2421 - val_accuracy: 0.9143
Epoch 17/20
18/18 [==============================] - 0s 2ms/step - loss: 0.0945 - accuracy: 0.9732 - val_l
oss: 0.2337 - val_accuracy: 0.9143
Epoch 18/20
18/18 [==============================] - 0s 2ms/step - loss: 0.0874 - accuracy: 0.9821 - val_l
oss: 0.2221 - val_accuracy: 0.9000
Epoch 19/20
18/18 [==============================] - 0s 3ms/step - loss: 0.0813 - accuracy: 0.9821 - val_l
oss: 0.2360 - val_accuracy: 0.9000
Epoch 20/20
18/18 [==============================] - 0s 2ms/step - loss: 0.0734 - accuracy: 0.9839 - val_l
oss: 0.2285 - val_accuracy: 0.9071
```

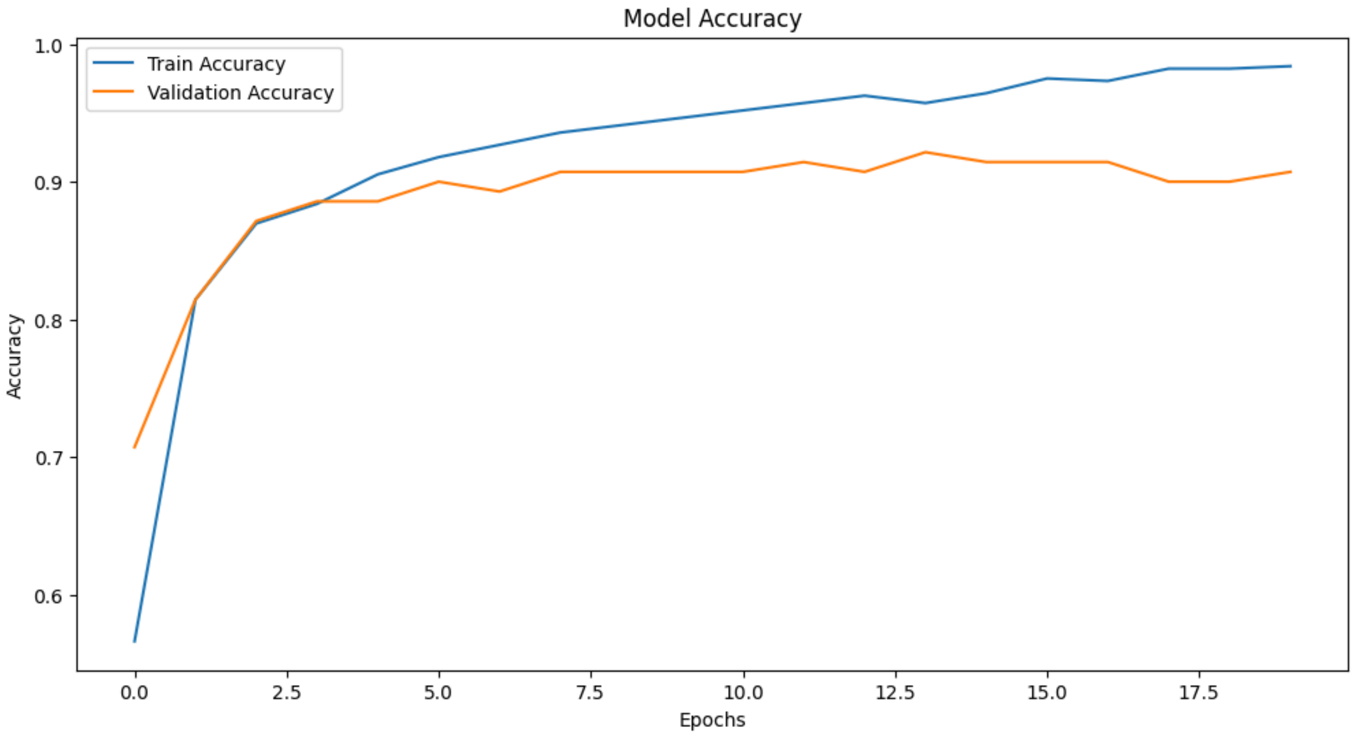
In [14]:

```
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f"Test Loss: {loss:.4f}")
print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

```
Test Loss: 0.1763
Test Accuracy: 93.33%
```

In [15]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



In [ ]: