```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "d5ac4036",
   "metadata": {},
   "source": [
    "### Binary Classification:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "169f39e0",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Epoch 1/10\n",
      "25/25 [==============================] - 1s 4ms/step - loss: 0.7030 - accuracy: 0.5288\n",
      "Epoch 2/10\n",
      "25/25 [==============================] - 0s 2ms/step - loss: 0.6976 - accuracy: 0.5238\n",
      "Epoch 3/10\n",
      "25/25 [==============================] - 0s 2ms/step - loss: 0.7150 - accuracy: 0.5300\n",
      "Epoch 4/10\n",
      "25/25 [==============================] - 0s 3ms/step - loss: 0.7101 - accuracy: 0.5050\n",
      "Epoch 5/10\n",
      "25/25 [==============================] - 0s 3ms/step - loss: 0.7066 - accuracy: 0.5337\n",
      "Epoch 6/10\n",
```

```
  "25/25 [==============================] - 0s 3ms/step - loss: 0.6977 - accuracy: 0.5188\n",
  "Epoch 7/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.7046 - accuracy: 0.5188\n",
  "Epoch 8/10\n",
  "25/25 [==============================] - 0s 3ms/step - loss: 0.6971 - accuracy: 0.5200\n",
  "Epoch 9/10\n",
  "25/25 [==============================] - 0s 3ms/step - loss: 0.6894 - accuracy: 0.5300\n",
  "Epoch 10/10\n",
  "25/25 [==============================] - 0s 3ms/step - loss: 0.6894 - accuracy: 0.5263\n",
  "7/7 [==============================] - 0s 2ms/step - loss: 0.6847 - accuracy: 0.5350\n",
  "Binary Classification Accuracy: 53.50%\n"
 ]
 }
],
"source": [
 "import numpy as np\n",
 "from tensorflow.keras.models import Sequential\n",
 "from tensorflow.keras.layers import Dense, Dropout\n",
 "from sklearn.model_selection import train_test_split\n",
 "from sklearn.metrics import accuracy_score\n",
 "\n",
 "# Generate random data for binary classification\n",
 "X = np.random.rand(1000, 10)\n",
 "y = np.random.randint(2, size=(1000,))\n",
 "\n",
 "# Split data into training and testing sets\n",
 "X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)\n",
 "\n",
 "# Build the binary classification model\n",
 "model_binary = Sequential([\n",
 "    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),\n",
```

```
  "    Dropout(0.2),\n",
  "    Dense(32, activation='sigmoid'),\n",
  "    Dropout(0.2),\n",
  "    Dense(1, activation='sigmoid')\n",
  "])\n",
  "\n",
  "# Compile the model with different optimizer and loss function\n",
  "model_binary.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])\n",
  "\n",
  "# Train the model\n",
  "model_binary.fit(X_train, y_train, epochs=10, batch_size=32)\n",
  "\n",
  "# Evaluate the model\n",
  "accuracy = model_binary.evaluate(X_test, y_test)[1]\n",
  "print(\"Binary Classification Accuracy: {:.2f}%\".format(accuracy * 100))"
 ]
},
{
 "cell_type": "markdown",
 "id": "13bc508c",
 "metadata": {},
 "source": [
  "### Regression:"
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "id": "dc6ec99e",
 "metadata": {},
 "outputs": [
```

```
{
 "name": "stdout",
 "output_type": "stream",
 "text": [
  "Epoch 1/10\n",
  "25/25 [==============================] - 1s 5ms/step - loss: 0.2748\n",
  "Epoch 2/10\n",
  "25/25 [==============================] - 0s 1ms/step - loss: 0.1255\n",
  "Epoch 3/10\n",
  "25/25 [==============================] - 0s 1ms/step - loss: 0.1135\n",
  "Epoch 4/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.1054\n",
  "Epoch 5/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.1013\n",
  "Epoch 6/10\n",
  "25/25 [==============================] - 0s 1ms/step - loss: 0.1033\n",
  "Epoch 7/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.0979\n",
  "Epoch 8/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.0976\n",
  "Epoch 9/10\n",
  "25/25 [==============================] - 0s 2ms/step - loss: 0.0930\n",
  "Epoch 10/10\n",
  "25/25 [==============================] - 0s 1ms/step - loss: 0.0985\n",
  "7/7 [==============================] - 0s 3ms/step - loss: 0.0920\n",
  "Mean Squared Error for Regression: 0.0920\n"
 ]
 }
],
"source": [
 "# Generate random data for regression\n",
```

```
   "X_reg = np.random.rand(1000, 10)\n",
   "y_reg = np.random.rand(1000, 1)\n",
   "\n",
   "# Split data into training and testing sets for regression\n",
   "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=42)\n",
   "\n",
   "# Build the regression model\n",
   "model_reg = Sequential([\n",
   "    Dense(64, activation='relu', input_shape=(X_train_reg.shape[1],)),\n",
   "    Dropout(0.2),\n",
   "    Dense(32, activation='relu'),\n",
   "    Dropout(0.2),\n",
   "    Dense(1, activation='linear')\n",
   "])\n",
   "\n",
   "# Compile the model with different optimizer and loss function for regression\n",
   "model_reg.compile(optimizer='adam', loss='mean_squared_error')\n",
   "\n",
   "# Train the regression model\n",
   "model_reg.fit(X_train_reg, y_train_reg, epochs=10, batch_size=32)\n",
   "\n",
   "# Evaluate the regression model\n",
   "mse = model_reg.evaluate(X_test_reg, y_test_reg)\n",
   "print(\"Mean Squared Error for Regression: {:.4f}\".format(mse))"
  ]
 }
],
"metadata": {
 "kernelspec": {
  "display_name": "Python 3 (ipykernel)",
```

```
  "language": "python",
  "name": "python3"
 },
 "language_info": {
  "codemirror_mode": {
   "name": "ipython",
   "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.10.9"
 }
},
"nbformat": 4,
"nbformat_minor": 5
}
```