```
Start coding or ge_nerate with AI.
```

```python
import kagglehub

# Download selected version
path = kagglehub.dataset_download("oddrationale/mnist-in-csv/versions/1")

print("Path to dataset files:", path)
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/oddrationale/mnist-in-csv?dataset_version_number=1...
100%|██████████| 15.2M/15.2M [00:00<00:00, 121MB/s]Extracting files...

Path to dataset files: /root/.cache/kagglehub/datasets/oddrationale/mnist-in-csv/versions/1
```

```python
import pandas as pd
import os

# Path to the downloaded dataset (from your output)
path = "/root/.cache/kagglehub/datasets/oddrationale/mnist-in-csv/versions/1"

# List files in the directory
print("Files in the dataset directory:")
print(os.listdir(path))

# The MNIST dataset typically comes in two CSV files:
# mnist_train.csv and mnist_test.csv

# Load and view the training data
train_path = os.path.join(path, "mnist_train.csv")
if os.path.exists(train_path):
    train_data = pd.read_csv(train_path)
    print("\nTraining data preview:")
    print(train_data.head())
else:
    print(f"\nTraining file not found at: {train_path}")

# Load and view the test data
test_path = os.path.join(path, "mnist_test.csv")
if os.path.exists(test_path):
    test_data = pd.read_csv(test_path)
    print("\nTest data preview:")
    print(test_data.head())
else:
    print(f"\nTest file not found at: {test_path}")
```

```
Files in the dataset directory:
['mnist_test.csv', 'mnist_train.csv']

Training data preview:
   5  0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  ...  0.608  0.609  0.610  \
0  0  0    0    0    0    0    0    0    0    0  ...      0      0      0
1  4  0    0    0    0    0    0    0    0    0  ...      0      0      0
2  1  0    0    0    0    0    0    0    0    0  ...      0      0      0
3  9  0    0    0    0    0    0    0    0    0  ...      0      0      0
4  2  0    0    0    0    0    0    0    0    0  ...      0      0      0

   0.611  0.612  0.613  0.614  0.615  0.616  0.617
0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0
2      0      0      0      0      0      0      0
3      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0

[5 rows x 785 columns]

Test data preview:
   7  0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  ...  0.658  0.659  0.660  \
0  2  0    0    0    0    0    0    0    0    0  ...      0      0      0
1  1  0    0    0    0    0    0    0    0    0  ...      0      0      0
2  0  0    0    0    0    0    0    0    0    0  ...      0      0      0
3  4  0    0    0    0    0    0    0    0    0  ...      0      0      0
4  1  0    0    0    0    0    0    0    0    0  ...      0      0      0

   0.661  0.662  0.663  0.664  0.665  0.666  0.667
0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0
2      0      0      0      0      0      0      0
3      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0

[5 rows x 785 columns]
```

```
!pip install kagglehub

import kagglehub

# Download directly in Colab
path = kagglehub.dataset_download("oddrationale/mnist-in-csv/versions/1")
print("Dataset downloaded to:", path)

# Now you can work with the files
import pandas as pd
train_df = pd.read_csv(path + "/mnist_train.csv")
test_df = pd.read_csv(path + "/mnist_test.csv")

train_df.head()
```

```
Requirement already satisfied: kagglehub in /usr/local/lib/python3.11/dist-packages (0.3.12)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from kagglehub) (24.2)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from kagglehub) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kagglehub) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from kagglehub) (4.67.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (3.4.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (2025.4.26)
Dataset downloaded to: /kaggle/input/mnist-in-csv
```

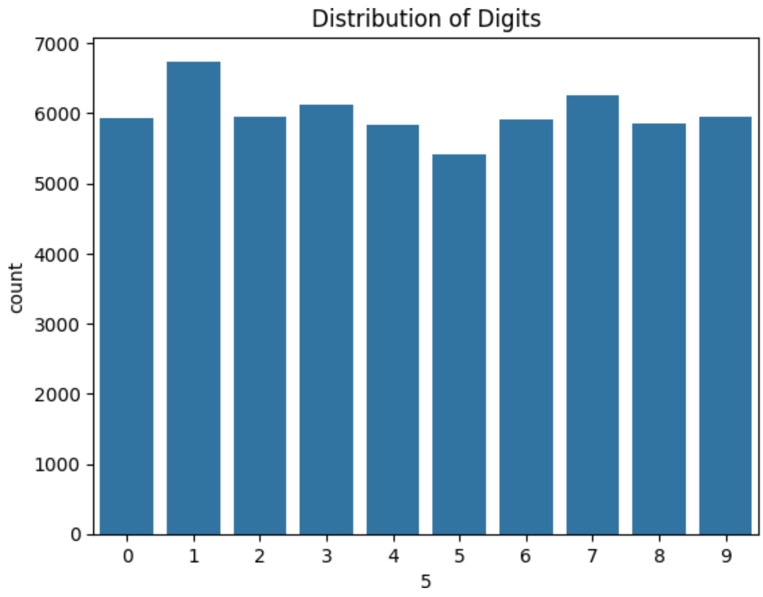
	5	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	...	0.608	0.609	0.610	0.611	0.612	0.613	0.614	0.615	0.616	0.617
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

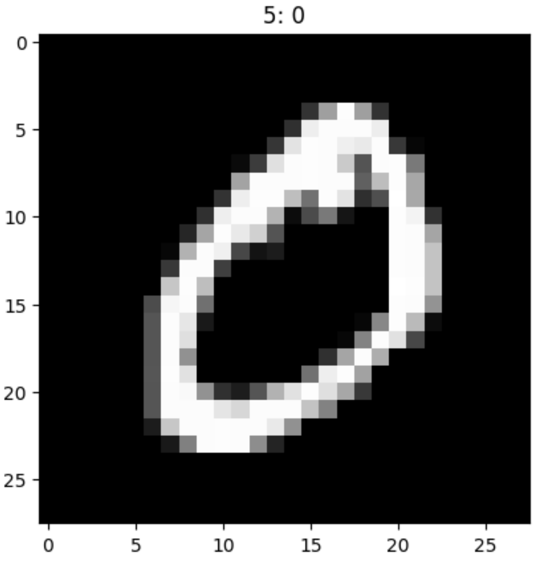
5 rows × 785 columns

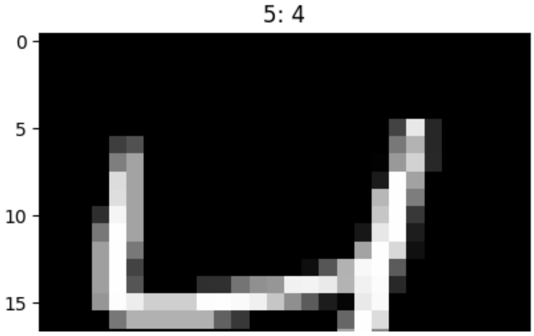
```
# Install and import necessary libraries
!pip install kagglehub
import kagglehub
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.utils import to_categorical

# Load dataset from KaggleHub
path = kagglehub.dataset_download("oddrationale/mnist-in-csv/versions/1")
print("Dataset downloaded to:", path)

# Read data
train_df = pd.read_csv(path + "/mnist_train.csv")
test_df = pd.read_csv(path + "/mnist_test.csv")

# 1. Data Cleaning
print("Checking for missing values:")
print(train_df.isnull().sum())

# 2. Exploratory Data Analysis
print("Label distribution:")
sns.countplot(x=train_df['5'])
plt.title('Distribution of Digits')
plt.show()

# Visualize sample digits
for i in range(5):
    digit = train_df.iloc[i, 1:].values.reshape(28, 28)
    plt.imshow(digit, cmap='gray')
    plt.title(f"5: {train_df.iloc[i, 0]}")
    plt.show()

# 3. Feature Engineering
X = train_df.drop('5', axis=1)
y = train_df['5']

# Normalize features
X = X / 255.0
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
```

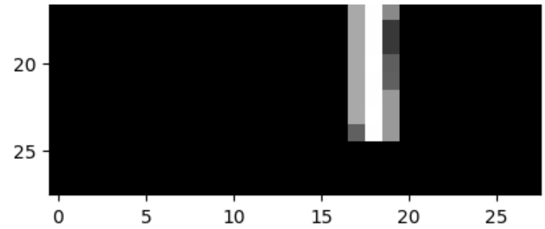
```
# One-hot encode labels
y_train = to_categorical(y_train, num_classes=10)
y_val = to_categorical(y_val, num_classes=10)

# 4. Model Engineering
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(10, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

# Train model
model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=10, batch_size=128)

# 5. Reporting (Model Accuracy)
loss, accuracy = model.evaluate(X_val, y_val)
print(f"Validation Accuracy: {accuracy:.4f}")
```

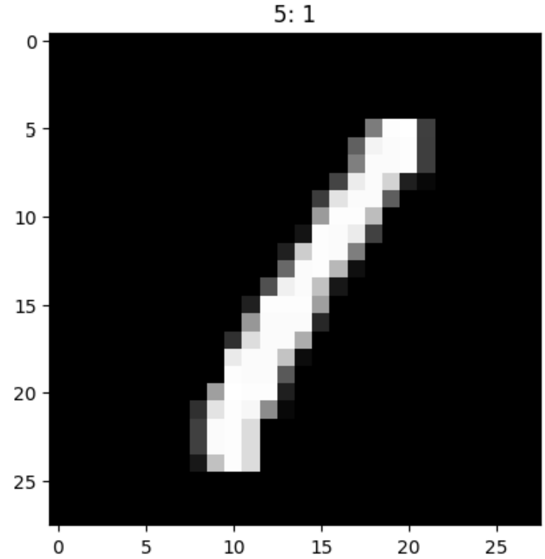
```
Requirement already satisfied: kagglehub in /usr/local/lib/python3.11/dist-packages (0.3.12)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from kagglehub) (24.2)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from kagglehub) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kagglehub) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from kagglehub) (4.67.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (3.4.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->kagglehub) (2025.4.26)
Dataset downloaded to: /kaggle/input/mnist-in-csv
Checking for missing values:
5        0
0        0
0.1      0
0.2      0
0.3      0
        ..
0.613    0
0.614    0
0.615    0
0.616    0
0.617    0
Length: 785, dtype: int64
Label distribution:
```

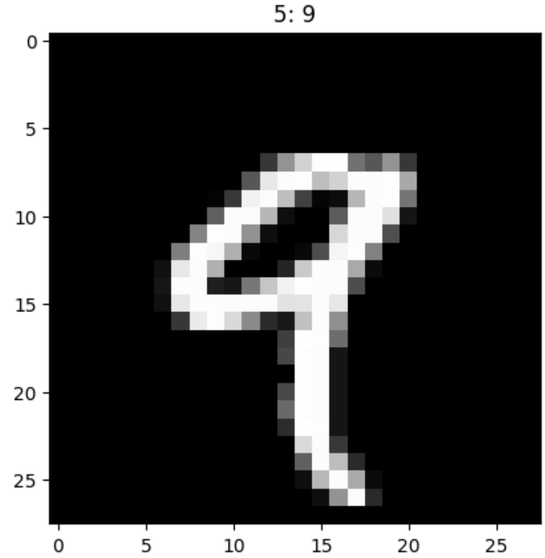


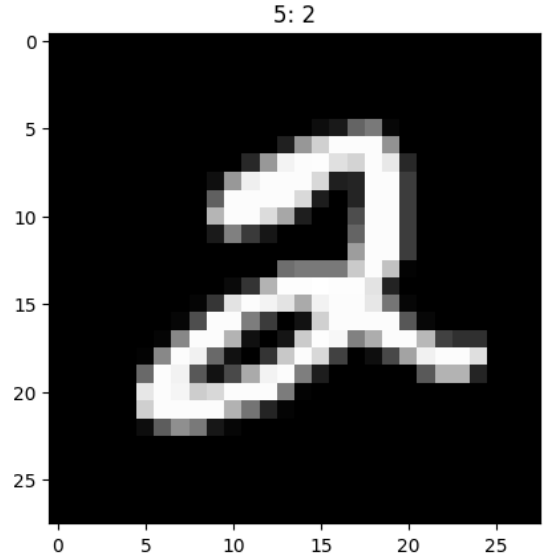












```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` ar
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential_2"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ dense_6 (Dense)                 │ (None, 512)            │       401,920 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_4 (Dropout)             │ (None, 512)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_7 (Dense)                 │ (None, 256)            │       131,328 │
```