```python
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 5GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```python
import numpy as np
import pandas as pd
```

```python
dataset = pd.read_csv("../input/az-handwritten-alphabets-in-csv-format/A_Z Handwritten Data/A_Z Handwritten Data.csv").astype('float32')
dataset.rename(columns={'0':'label'}, inplace=True)

x= dataset.drop('label',axis = 1)
y = dataset['label']
```

```python
print(x.shape)
print(y.shape)

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from keras.utils import np_utils
x_train,x_test,y_train,y_test = train_test_split(x,y)

standard_scaler = MinMaxScaler()
standard_scaler.fit(x_train)

x_train = standard_scaler.transform(x_train)
x_test = standard_scaler.transform(x_test)

y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
```

```python
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
x_train.shape[1]

model = keras.Sequential()
model.add(layers.Dense(300, activation="relu" , input_dim = x_train.shape[1]))
model.add(layers.Dense(200, activation="relu"))
model.add(layers.Dense(200, activation="relu"))
model.add(layers.Dense(200, activation="relu"))
model.add(layers.Dense(300, activation="relu"))
model.add(layers.Dense(100, activation="relu"))
model.add(layers.Dense(len(y.unique()), activation="softmax"))
adam = keras.optimizers.Adam(lr=0.001,decay=1e-6)
model.compile(loss='categorical_crossentropy', optimizer=adam , metrics = ['Accuracy'])

model.fit(x_train,y_train,epochs=100)

model.summary()

model.evaluate(x_test,y_test)

import matplotlib.pyplot as plt

plt.imshow(x_test[100].reshape(28,28), cmap='Greys')
plt.show()

k=model.predict(x_test[250].reshape(1,784))
np.argmax(k)
```

DATA SET LINK: HANDWRITTEN CHARACTER RECOGNITION USING ANN | Kaggle