```
import tensorflow as tf
print("TensorFlow version:", tf.__version__)
```

```
TensorFlow version: 2.15.0
```

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import pandas as pd
```

#importing Tensor flow and checking version of it

#Load a dataset

```
mnist = tf.keras.datasets.mnist
```

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

Build a machine learning model

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

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

```
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

```
Epoch 1/5
1875/1875 [==============================] - 8s 4ms/step - loss: 0.3011 - accuracy: 0.9125
Epoch 2/5
1875/1875 [==============================] - 7s 3ms/step - loss: 0.1433 - accuracy: 0.9568
Epoch 3/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.1068 - accuracy: 0.9678
Epoch 4/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0856 - accuracy: 0.9739
Epoch 5/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0744 - accuracy: 0.9773
313/313 [==============================] - 1s 2ms/step - loss: 0.0687 - accuracy: 0.9791
[0.06870933622121811, 0.9790999889373779]
```

```
predictions = model(x_train[:1]).numpy()
predictions
```

```
array([[3.1200800e-09, 9.5119679e-10, 1.9356932e-07, 4.0199850e-03,
        2.5051228e-17, 9.9597979e-01, 9.0535725e-11, 6.2725247e-09,
        4.7381151e-11, 3.3913022e-08]], dtype=float32)
```

```
tf.nn.softmax(predictions).numpy()
```

```
array([[0.08538686, 0.08538686, 0.08538687, 0.08573081, 0.08538686,
        0.2311743 , 0.08538686, 0.08538686, 0.08538686, 0.08538686]],
      dtype=float32)
```

```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
probability_model = tf.keras.Sequential([
  model,
  tf.keras.layers.Softmax()
])
```

```
probability_model(x_test[:5])
```

```
<tf.Tensor: shape=(5, 10), dtype=float32, numpy=
array([[0.08534048, 0.08534048, 0.08534049, 0.08536554, 0.08534048,
        0.08534056, 0.08534048, 0.23191004, 0.08534048, 0.0853409 ],
       [0.08533723, 0.08534041, 0.2319616 , 0.08533724, 0.08533723,
        0.08533736, 0.08533724, 0.08533723, 0.08533724, 0.08533723],
       [0.08534106, 0.23190129, 0.08534184, 0.08534121, 0.0853414 ,
        0.08534131, 0.0853424 , 0.0853596 , 0.08534881, 0.08534107],
       [0.23192433, 0.08533962, 0.08535109, 0.08533959, 0.08533987,
        0.0853396 , 0.08534381, 0.08534228, 0.0853396 , 0.08534021],
       [0.08542545, 0.08542314, 0.08542412, 0.08542314, 0.23059645,
        0.08542317, 0.08542336, 0.08545218, 0.08542477, 0.08598425]],
      dtype=float32)>
```