```
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras import layers

df = pd.read_csv('mobile.csv')
df.head()
```

attery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_hei
842	0	2.2	0	1	0	7	0.6	188	2	...	
1021	1	0.5	1	0	1	53	0.7	136	3	...	9
563	1	0.5	1	2	1	41	0.9	145	5	...	12
615	1	2.5	0	0	0	10	0.8	131	6	...	12
1821	1	1.2	0	13	1	44	0.6	141	2	...	12

s × 21 columns

```
X = df.iloc[:,:20].values
Y = df.iloc[:,20:21].values
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
```

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder()
Y = ohe.fit_transform(Y).toarray()
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25)
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
model.add(Dense(40, input_dim=20, activation='relu'))
model.add(Dense(20,activation='relu'))
model.add(Dense(4,activation='softmax'))
```

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

```
history = model.fit(X_train, Y_train,
                    validation_data = (X_test,Y_test),
                    epochs=100, batch_size=64)
```

```
24/24 [==============================] - 0s 4ms/step - loss: 0.0248 - accuracy: 0.9993 - val_loss: 0
Epoch 73/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0239 - accuracy: 0.9993 - val_loss: 0
Epoch 74/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0233 - accuracy: 0.9993 - val_loss: 0
Epoch 75/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0224 - accuracy: 0.9993 - val_loss: 0
Epoch 76/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0215 - accuracy: 0.9993 - val_loss: 0
Epoch 77/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0204 - accuracy: 0.9993 - val_loss: 0
Epoch 78/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0200 - accuracy: 0.9993 - val_loss: 0
Epoch 79/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0196 - accuracy: 0.9993 - val_loss: 0
Epoch 80/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0188 - accuracy: 0.9993 - val_loss: 0
Epoch 81/100
24/24 [==============================] - 0s 3ms/step - loss: 0.0180 - accuracy: 0.9993 - val_loss: 0
Epoch 82/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0175 - accuracy: 0.9993 - val_loss: 0
Epoch 83/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0173 - accuracy: 0.9993 - val_loss: 0
Epoch 84/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0165 - accuracy: 1.0000 - val_loss: 0
Epoch 85/100
24/24 [==============================] - 0s 5ms/step - loss: 0.0159 - accuracy: 0.9993 - val_loss: 0
Epoch 86/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0153 - accuracy: 0.9993 - val_loss: 0
Epoch 87/100
24/24 [==============================] - 0s 3ms/step - loss: 0.0149 - accuracy: 0.9993 - val_loss: 0
Epoch 88/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0142 - accuracy: 1.0000 - val_loss: 0
Epoch 89/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0141 - accuracy: 1.0000 - val_loss: 0
Epoch 90/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0137 - accuracy: 1.0000 - val_loss: 0
Epoch 91/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0132 - accuracy: 1.0000 - val_loss: 0
Epoch 92/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0127 - accuracy: 1.0000 - val_loss: 0
Epoch 93/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0125 - accuracy: 1.0000 - val_loss: 0
Epoch 94/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0121 - accuracy: 1.0000 - val_loss: 0
Epoch 95/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0116 - accuracy: 1.0000 - val_loss: 0
Epoch 96/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0111 - accuracy: 1.0000 - val_loss: 0
Epoch 97/100
24/24 [==============================] - 0s 3ms/step - loss: 0.0109 - accuracy: 1.0000 - val_loss: 0
Epoch 98/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0106 - accuracy: 1.0000 - val_loss: 0
Epoch 99/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0103 - accuracy: 1.0000 - val_loss: 0
Epoch 100/100
24/24 [==============================] - 0s 4ms/step - loss: 0.0099 - accuracy: 1.0000 - val_loss: 0
```

```
Y_pred = model.predict(X_test)
#Converting predictions to label
pred = list()
for i in range(len(Y_pred)):
    pred.append(np.argmax(Y_pred[i]))
```

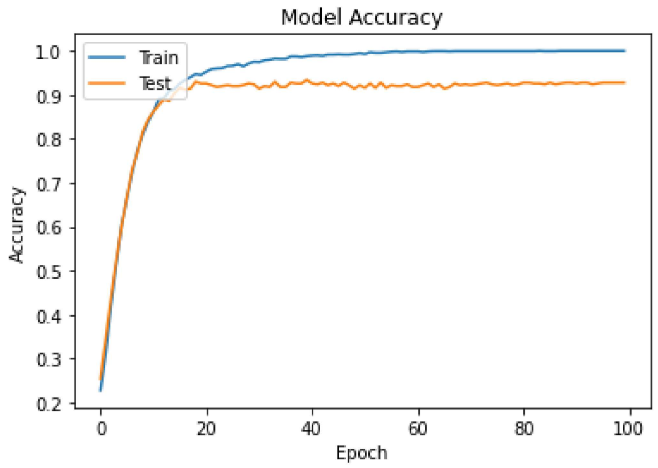
```
#converting One Hot Encoded test label to label
test = list()
for i in range(len(Y_test)):
    test.append(np.argmax(Y_test[i]))
```

```
16/16 [==============================] - 0s 2ms/step
```

```
from sklearn.metrics import accuracy_score
a = accuracy_score(pred,test)
print('Accuracy: ',a*100)
```

```
Accuracy:  92.80000000000001
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train','Test'], loc='upper left')
plt.show()
```



✓ 0s completed at 10:54 AM ● ×