CONCLUSION

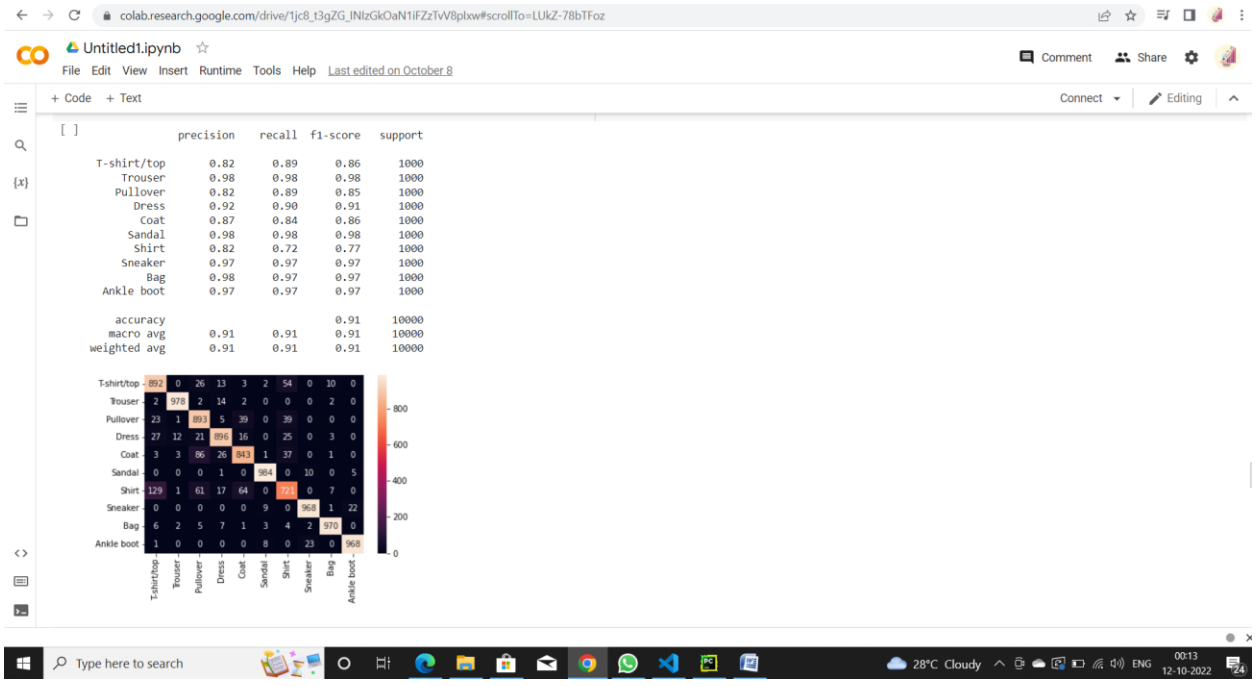
Finally we will get accuracy for our model, But accuracy may varies sometime, so in this case we can use confusion matrix

CONFUSION MATRIX:

```
from sklearn.metrics import confusion_matrix
plt.figure(figsize=(16,9))
y_pred_labels = [ np.argmax(label) for label in y_pred ]
cm = confusion_matrix(y_test, y_pred_labels)

sns.heatmap(cm, annot=True, fmt='d',xticklabels=class_labels, yticklab
els=class_labels)

from sklearn.metrics import classification_report
cr= classification_report(y_test, y_pred_labels, target_names=class_la
bels)
print(cr)
```



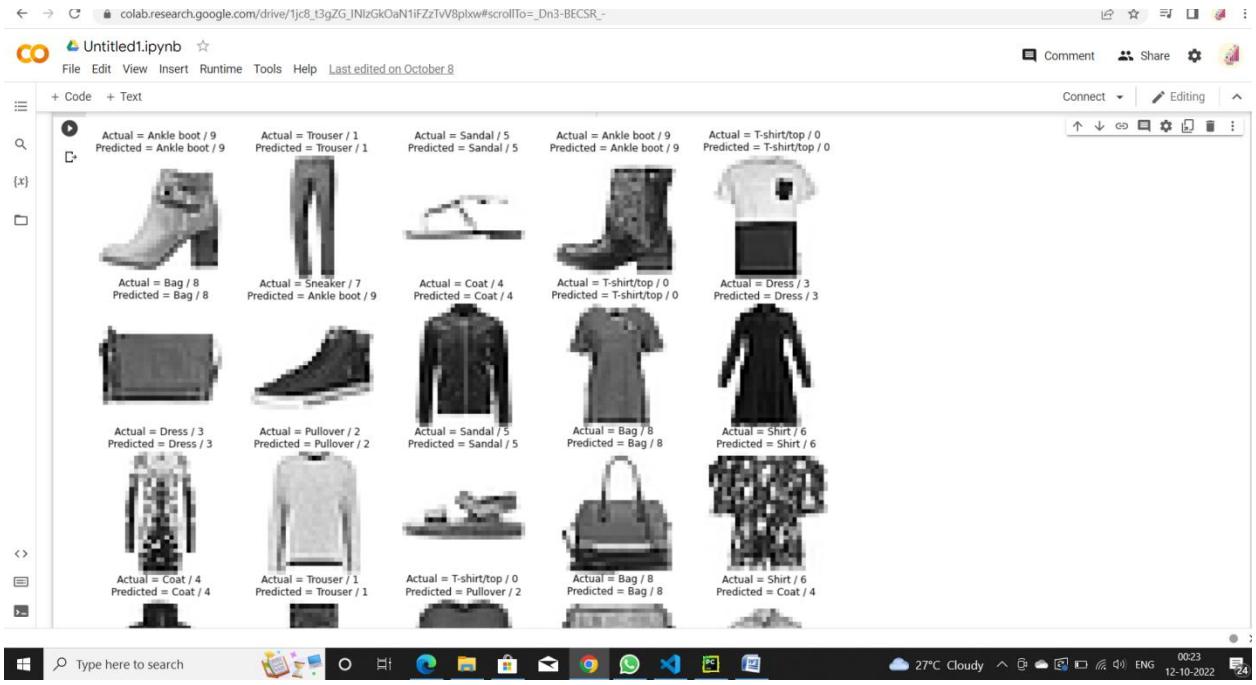
The Classification Report:

	precision	recall	f1-score	support
T-shirt/top	0.82	0.89	0.86	1000
Trouser	0.98	0.98	0.98	1000
Pullover	0.82	0.89	0.85	1000
Dress	0.92	0.90	0.91	1000
Coat	0.87	0.84	0.86	1000
Sandal	0.98	0.98	0.98	1000
Shirt	0.82	0.72	0.77	1000
Sneaker	0.97	0.97	0.97	1000
Bag	0.98	0.97	0.97	1000
Ankle boot	0.97	0.97	0.97	1000
accuracy			0.91	10000
macro avg	0.91	0.91	0.91	10000
weighted avg	0.91	0.91	0.91	10000

This deep learning algorithm now identifies and predicted the images.

```
plt.figure(figsize=(16,16))

j=1
for i in np.random.randint(0, 1000,25):
  plt.subplot(5,5, j); j+=1
  plt.imshow(X_test[i].reshape(28,28), cmap = 'Greys')
  plt.title('Actual = {} / {} \nPredicted = {} / {}'.format(class_labe
ls[y_test[i]], y_test[i], class_labels[np.argmax(y_pred[i])],np.argmax
(y_pred[i])))
  plt.axis('off')
```



So we finally completed, and images are predicted correctly with the label name given. The confusion matrix represents how many products or images are perfectly predicted. And the classification report says the precision, recall, f1score of the model an also accuracy.

THANK YOU.....