

```

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

y_pred = ViT.predict(df_te)

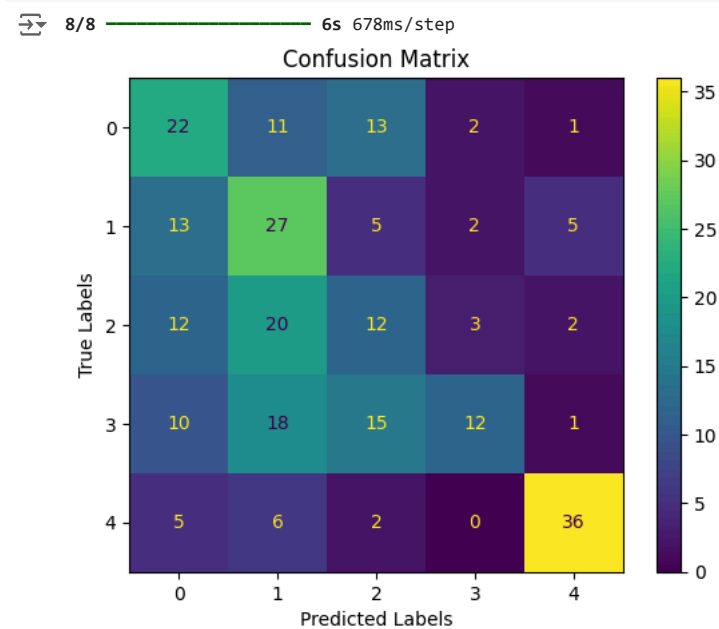
y_pred_labels = np.argmax(y_pred, axis=1)

y_true_labels = np.argmax(Y_te2, axis=1)

cm = confusion_matrix(y_true_labels, y_pred_labels)

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=range(5))
disp.plot()
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()

```



- Визуализируйте кривые обучения построенных моделей для показателей потерь на обучающей и валидационной выборках на одном рисунке в зависимости от эпохи обучения, подписывая оси и рисунок и создавая легенду. Используйте для визуализации относительные потери (потери, деленные на начальные потери на первой эпохе).

Transformer

```

losses = history.history['loss']
loss0 = history.history['loss'][0]

relative_losses = [loss / loss0 for loss in losses]

```

```

val_losses = history.history['val_loss']
val_loss0 = history.history['val_loss'][0]
relative_val_losses = [val_loss / val_loss0 for val_loss in val_losses]

```

```

plt.plot(relative_losses, label='relative loss')
plt.plot(relative_val_losses, label="relative val_loss")
plt.xlabel("Epochs")
plt.ylabel(relative_losses)
plt.title("Train and Validation losses Over Epochs")
plt.legend()
plt.grid()
plt.show()

```



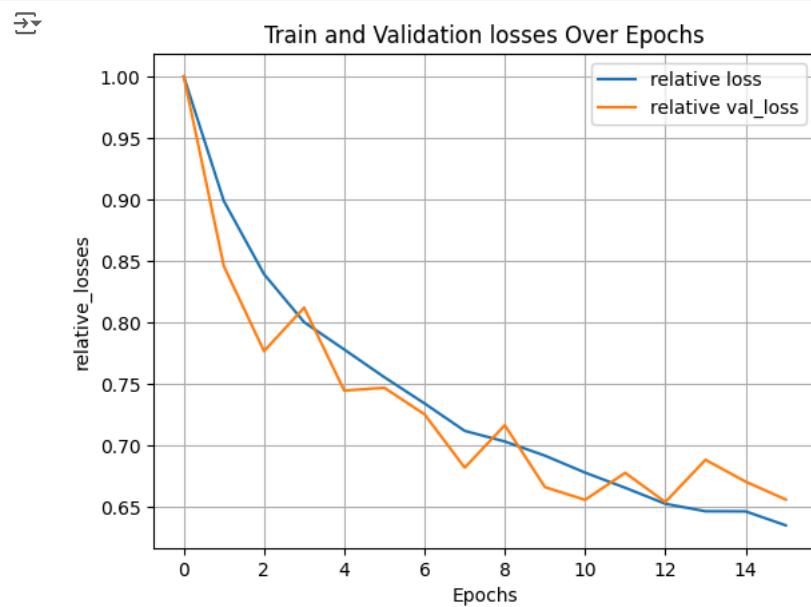
CNN

```
losses = history1.history['loss']
loss0 = history1.history['loss'][0]

relative_losses = [loss / loss0 for loss in losses]
```

```
val_losses = history1.history['val_loss']
val_loss0 = history1.history['val_loss'][0]
relative_val_losses = [val_loss / val_loss0 for val_loss in val_losses]
```

```
plt.plot(relative_losses, label='relative loss')
plt.plot(relative_val_losses, label="relative val_loss")
plt.xlabel("Epochs")
plt.ylabel("relative_losses")
plt.title("Train and Validation losses Over Epochs")
plt.legend()
plt.grid()
plt.show()
```



6. Оцените качество многоклассовой классификации нейронными сетями на тестовой выборке при помощи показателя качества, указанного в индивидуальном задании, и выведите название нейронной сети с лучшим качеством.

Показатель качества многоклассовой классификации: максимальная полнота классов, где полнота (recall) класса равна доле правильных предсказаний для всех точек, принадлежащих этому классу.

```
X1 = cnn.predict(df_te)
X2 = ViT.predict(df_te)
```

8/8 1s 110ms/step
8/8 11s 1s/step

```
for i in range(len(X1)):
    for j in range(len(X1[i])):
        if X1[i][j] == min(X1[i]):
            X1[i][j] = 1
        else:
            X1[i][j] = 0
for i in range(len(X2)):
    for j in range(len(X2[i])):
        if X2[i][j] == min(X2[i]):
            X2[i][j] = 1
        else:
            X2[i][j] = 0
```


```
X11 = np.array(X1, dtype=np.int32)
X22 = np.array(X2, dtype=np.int32)
```

```
X22_re = np.resize(X22, (255, 5))
```

```
m1 = tf.keras.metrics.Recall()
m2 = tf.keras.metrics.Recall()
```

```
m1.update_state(X11, Y_te2)
m2.update_state(X22_re, Y_te2)
```

```
m1.result().numpy(), m2.result().numpy()
```

```
 (0.2, 0.221843)
```