In [ ]:

```
from transformers import BertTokenizer, TFBertModel , TFBertForSequenceClassification
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import AUC
import tensorflow as tf
```

In [ ]:

```
def final_function_1(x):
    """This function will take the row input comment and predict
       all the probablities corresponding to each label.
    """
    #loding bert tokenzier
    tokenizer = BertTokenizer.from_pretrained('bert-base-cased')

    # encoding the comment for the input of bert
    encoding = tokenizer.encode_plus(x  , max_length = 128 , pad_to_max_length = True  ,do_lower_case = False)
    input_ids , attention_id = encoding["input_ids"] , encoding["attention_mask"]

    #loading the best weights for single model
    model = tf.keras.models.load_model('C:\\Users\\my pc\\project_2nd final model')
    model.load_weights("C:\\Users\\my pc\\Downloads\\final_model.h5")
    prediction = (model.predict([input_ids , attention_id])

    # final results
    return print("comment has probablity of being toxic == " , prediction[0]),
           print("comment has probablity of being severe_toxic == " , prediction[0]),
           print("comment has probablity of being obscene == " , prediction[0]),
           print("comment has probablity of being threat == " , prediction[0]),
           print("comment has probablity of being insult == " , prediction[0]),
           print("comment has probablity of being identity_hate == " , prediction[5])
```

In [ ]:

```
def final_function_2(x , y):
    """This function will take the row input comment and predict
       with the metric.
    """
    y = str(y).lower()
    lst = ["clean" , "toxic" , "sever" , "threat" ,"obscene" , "insult" , "identity_hate"]
    if y in lst:
        if y=="toxic":
            vec = [1, 0, 0, 0 ,0 , 0]
        elif y=="sever_toxic":
            vec = [0, 1 ,0 , 0 ,0 ,0]
        elif y=="obscene":
            vec = [0, 0 ,1 , 0 ,0 ,0]
        elif y=="threat":
            vec = [0, 0 ,0 , 1 ,0 ,0]
        elif y=="insult":
            vec = [0, 0 ,0 , 0 ,1 ,0]
        elif y=="identity_hate":
            vec = [0, 0 ,0 , 0 ,0 ,1]
        vec = np.array(vec)


        #loding bert tokenzier
        tokenizer = BertTokenizer.from_pretrained('bert-base-cased')

        # encoding the comment for the input of bert
        encoding = tokenizer.encode_plus(x  , max_length = 128 , pad_to_max_length = True  ,do_lower_case = False)
        input_ids , attention_id = encoding["input_ids"] , encoding["attention_mask"]

        #loading the best weights for single model
        model = tf.keras.models.load_model('C:\\Users\\my pc\\project_2nd final model')
        model.load_weights("C:\\Users\\my pc\\Downloads\\final_modle3.h5")
        prediction = (model.evaluate([input_ids , attention_id]) , vec)
        dt = dict(zip(model.metrics_names, prediction))

        # final results
        return print("The auc of being "+y+ " is", dt["auc"]*100,str("%"))
    else:
        print("Please enter the valid string as real or fake")
```