PROJECT OVERVIEW -
Build a multi-class classifier prototype for the Go Emotions dataset of 58k carefully curated comments extracted from Reddit, with human annotations into 13 emotion categories or "neutral".

DATASET -
The details of the dataset, including how to download, are given below:
• https://huggingface.co/datasets/go_emotions/viewer/simplified/train
• Number of examples: 58,009.
• Total Number of labels: 27 + Neutral.
• Maximum sequence length in training and evaluation datasets: 30.
* LABELS TO BE MAPPED: admiration, anger, approval, disappointment, confusion, curiosity, sadness, pride, excitement, gratitude, surprise, desire, fear + neutral
* LABELS TO BE MERGED: amusement, annoyance, caring, disapproval, nervousness, realization, grief, joy, optimism, love, relief, disgust, embarrassment, remorse

NO	LABELS TO BE MAPPED	LABELS TO BE MERGED
1	Admiration	amusement
2	anger	annoyance
3	approval	caring
4	disappointment	disapproval
5	confusion	nervousness
6	curiosity	realization
7	sadness	grief, remorse
8	pride	joy
9	excitement	optimism
10	gratitude	love
11	surprise	relief
12	desire	disgust
13	fear	embarrassment
14	neutral	

Based on the comment in the text column the machine will classify into one of the labels listed under 'LABELS TO BE MAPPED'. The comments which have labels listed under 'LABELS TO BE MERGED' – these will be merged with the mapping labels. So a comment which looks like an amusement comment will be labelled as Admiration, a comment which looks like annoyance will be labelled as Anger etc.

KEY DELIVERABLES -

I) A multi-class classifier prototype for the GoEmotions dataset of 58k carefully curated comments extracted from Reddit, with human annotations into 13 emotion categories or "neutral".

The following must be covered in the implementation as separate jupytr notebooks -
1) Data Analysis - Analyse and visualise the dataset.
2) Data Pre-processing experiments - Tokenisation, Apply Stop - Words, Normalisation.
3) Text featurisation/transformation into numerical vectors evaluation techniques - Topic Modelling (using LDA) vs Latent Semantic Analysis (LSA)
4) NLP Algorithm technique evaluation - Logistic Regression VS (Multinomial) Naive Bayes VS Linear Support Vector Machine VS Random Forest VS Deep Learning (RNN,CNN)
5) Choices of loss functions and optimisers – explain choices with facts from the results.
6) Hyperparameter optimisation – what are the most appropriate values (e.g., learning rate, training cycles, etc., depending on the algorithm).
7) Finetuning vs full training – which one is more appropriate (it might depend on the dataset).

NOTE -
I will be putting together a report based on the above so should be able to extract the following information from the above -
Demarcated details of algorithms and hyperparameters used.
The results on test set obtained in terms of classification report.
Confusion matrices.
Conclusion on results explaining which classes were misclassified the most, and which were misclassified the least.

The following are also needed for the GoEmotions - multi-class classifier:
8) Build a simple web service to host the model implemented above as an endpoint. The web service must run locally i.e local host.
9) Build some functionality in a notebook to perform testing on the deployed endpoint (i.e., some client function to consume the service via HTTP) and document the process and findings (in the notebook). No need for any UI here, just command line interaction will suffice.
10) Build some basic monitoring capability to capture user inputs and the model predictions, and store the inputs, model predictions and time/date of the interaction in a text log file (in a way that it can be parsed programmatically).
11) Build a basic CI/CD pipeline that will build and deploy the model when data or code changes. There is no need to trigger this automatically, so a manual execution script will be sufficient.

NOTE – I SHOULD BE ABLE TO RUN AND EXECUTE THIS AT MY END FROM GOOGLE COLLAB / JUPTYR NOTEBOOK.