COHORT REPORT

Prepared by: **Team 9 (COHORT 3)**
Pranav Senthilkumaran (ps1471)
Sanjith Ganesh (sg2151)
Master's in Data Science (MSDS)

PART 1: COHORT 3 ACTIVITY: Cohort 3 recording link

TEAM 7

PROJECT TITLE: MULTI-HOP CLAIM VERIFICATION

TEAM MEMBERS: GIREESHEE PENDELA, VEERA JEESHITHA KOLLA

GRADE: A

DESCRIPTION:

- The project focuses on building an automated system that verifies whether a given claim is true by analyzing textual evidence. Using machine learning and NLP techniques, the system classifies each claim as supported, refuted or lacking enough information. The team trained classical models and DistilBERT on the **FEVER** dataset and their overall goal is to enhance automated claim verification by selecting the best model.

STRENGTHS:

- The team showed **strong understanding of NLP foundations**, including text representation (TF-IDF), classical ML models (Logistic Regression, SVM, Random Forest) and transformer-based models (DistilBERT).
- They used **good experimental design**, following practices such as a 70/15/15 train/validation/test split, mixed-precision fp16 training and using uniform evaluation metrics across all models.
- They provided a **good comparison between traditional ML models and DistilBERT**.
- Their plan of action includes choosing an advanced model, such as **RoBERTa**, which could improve contextual reasoning and reduce NEI misclassifications.

WEAKNESS:

- There is **missing multi-hop evidence handling**, as the current version focuses mainly on a single input which is claim + one evidence sentence. Full multi-hop pipelines require evidence retrieval, multi-sentence reasoning, and combining evidence across multiple hops. **(Potential Implementation)**
- There is **limited error analysis** in the presentation, with missing elements such as confusion matrix and analysis of failure cases. Including these would demonstrate deeper understanding of model behavior.

SUGGESTIONS FOR IMPROVEMENT:

- The team could add a real multi-hop pipeline to strengthen the system's ability to handle multi-sentence reasoning.

- They could incorporate an error analysis section, which would help identify where the model fails, suggest improvements for NEI classification, and justify next steps such as RoBERTa or BERT-base.
- They can also try **parameter-efficient finetuning, such as LoRA or layer freezing**, as these methods require very few trainable parameters, improves accuracy, and significantly reduce training time.

TEAM 8

PROJECT TITLE: QUANTIFYING THE ENVIRONMENTAL COST OF AI: CARBON EMISSIONS IN LANGUAGE MODEL FINE-TUNING FOR QUESTION ANSWERING

MEMBERS - DIKSHA PHULORIA, SHRUTI ELANGOVAN, SANJANA UMESH SAWANT

GRADE: A

DESCRIPTION:

This project studies the environmental cost of training language models, focusing on how much carbon is emitted when you fine tune them for Q&A's. They used the SQuAD 2.0 dataset, where the model has to pick the correct span of text from a Wikipedia passage and tested transformer models like DistillBERT, BERT, RoBERTa and GPT2 with different training setups such as full fine tuning, LoRA and few shot training. Using a tool called **CodeCarbon**, they measured how much energy the hardware used and turned that into estimated carbon emissions. With that they compared how emissions and F1 scores changed as they varied the dataset size and training method.

STRENGTHS:

- One major strength is that the topic is very timely and **clearly connected to real world concerns**.
- Their setup was also clear using a **well known datase**t and familiar models and a structured pipeline.
- The **visualizations were strong** with the pipeline diagram showed the overall flow, and the plots made the trade off between accuracy and carbon cost easy to see.

WEAKNESS:

- **Limited scope of models** and data so it is hard to know if their conclusions will hold for larger language models.
- Even though they showed F1 scores, they could add **more comparison metrics**, like exact match scores, training time, or carbon emissions per unit of F1.
- There was **no qualitative error analysis**.
- For example, they could show a few Q/A cases where full fine tuning does better than LoRA or vice versa.

IMPROVEMENTS:

- They could add a few concrete question and answer examples where full fine tuning works better than LoRA or also cases where the lighter methods are good enough. It would also help if they gave a short list of practical takeaways on when to use which training method and when it is worth using for full fine tuning.
- Personally we thought, they could talk a bit more about the **limits of their setup**, like using only smaller models and a single dataset, and how the results might change for larger models or different tasks.

PROJECT PROGRESS AFTER INTERIM PRESENTATION

PROJECT TITLE: NEWS HEADLINE CLASSIFICATION: COMPARISON OF TRADITIONAL ML AND TRANSFORMER MODELS

MEMBERS - SANJITH GANESH, PRANAV SENTHILKUMARAN

1) IDENTIFYING BOTTLENECKS IN OUR DISTILBERT MODEL:

- After the interim we focused on finding bottlenecks and saw that DistilBERT still struggles with some category boundaries even though it does better than TF IDF. The main issue is that different categories **share very similar words**, especially in short headlines.
- By reading these misclassified headlines, we saw clear patterns like World being predicted as Business when the headline talks about trade or financial deals, and Business being predicted as World when it mentions countries or government actions.
- Overall, we found that DistilBERT struggles when headlines mix economic, political and tech language with very little context, and this error analysis is what motivated us to try LoRA fine tuning and compare it more carefully against our TF IDF baselines.

2) IMPROVING DISTILBERT WITH LORA (PEFT):

- LoRA trains only a tiny part of the model instead of updating all 66M+ DistilBERT weights. It adds small low-rank matrices to a few attention layers while keeping the main model frozen, which makes training much faster and lighter.
- **trainable params: 1,183,492 || all params: 68,140,040 || trainable%: 1.7369**
- We did not include Key as it increases parameter count and can sometimes reduce stability on smaller datasets like AG News and increases memory usage.
- For classification, **(Query +Value)** is ideal.

Evaluation Results:

- LoRA improves model performance, achieving **94% accuracy** and **94% macro F1**; **2% increase over full fine-tuning**. It performs best on categories like **Sports (F1 = 0.98)**, Business and Sci/ Tech did much better.

For **final deliverables**, we will include **RoBERTa model** and further optimize hyper parameters and regularize dropout.