**✅ Project Report: Sentiment Classification using Transformers**

**Project Title:**  
**Fine-Tuning Transformer Models for 3-Class Sentiment Analysis on App Reviews**

**🧾 Problem Statement**

The goal of this project was to classify app review texts into **three sentiment classes** — **Negative**, **Neutral**, and **Positive** — using both traditional machine learning and transformer-based NLP models. This task is crucial for analyzing customer feedback, monitoring brand perception, and identifying satisfaction trends.

**📂 Dataset**

* **Source:** Internal dataset containing user-generated app reviews.
* **Labels:** Original labels mapped to:
  + 0 → Negative
  + 1 → Neutral
  + 2 → Positive
* **Total samples:** ~10k+ for training, separate test set for evaluation.

**⚙️ Models Tried & Results**

| **Model** | **Description** | **Accuracy** |
| --- | --- | --- |
| **Logistic Regression** | Baseline using TF-IDF features | 68% |
| **cardiffnlp/twitter-roberta-base-sentiment** | Pretrained on tweets (zero-shot) | 26% |
| **Fine-tuned RoBERTa (cardiffnlp)** | 3-class fine-tuned on full training set | **72%** |
| microsoft/deberta-v3-small | General-purpose model (zero-shot) | 20% |
| distilbert-base-uncased | Lightweight transformer (zero-shot) | 21% |

**📈 Evaluation Metrics (on test set)**

**Accuracy:** 72%  
**Model:** Fine-tuned twitter-roberta-base-sentiment

**Confusion Matrix Insight:**

* **Positive (class 2):** Excellent recall (1625/1982 correct)
* **Negative (class 0):** Strong performance (1541/1981 correct)
* **Neutral (class 1):** Most confused class (only 435/1037 correct)

**Next Steps:**

* Apply class weights or oversampling to improve Neutral class performance.
* Explore fine-tuning deberta-v3-small, which may outperform RoBERTa when trained.

**🛠️ Tools & Libraries**

* Python, Pandas, Scikit-learn
* Hugging Face Transformers (Trainer, AutoModel, pipeline)
* Tokenizers, PyTorch, datasets
* Matplotlib (for confusion matrix)