Machine Learning Assignment

PROJECT REPORT

<TEAM ID : 16>

<**PROJECT TITLE:** Insincere Question Classification on Quora>

Name	SRN
CHETAN NADICHAGI	PES2UG23CS149
DIVYA J	PES2UG24CS810

Problem Statement

Online platforms such as Quora allow users to ask and answer questions freely. However, some questions are insincere or toxic, intended to provoke, mislead, or offend rather than seek genuine information.
These toxic questions negatively affect the user experience and platform moderation.

Our Solution
Our solution uses Natural Language Processing (NLP**)** and Machine Learning to automatically detect toxic or insincere questions on Quora.
The system cleans and processes text, converts it into numerical features using TF-IDF, and classifies questions using models like Logistic Regression and Random Forest.
This helps identify and filter harmful content to maintain a respectful platform environment.

Objective / Aim

Primary Objective:
To develop a text classification model capable of identifying insincere questions based on their linguistic patterns**.**

Technical Aims:

1. Text Preprocessing – Perform tokenization, stopword removal, and text cleaning to prepare raw text.
2. Feature Extraction – Represent cleaned text using TF-IDF vectorization to capture important word features.
3. Model Training – Train supervised models like Logistic Regression and Random Forest for classification.
4. Performance Evaluation – Assess models using Accuracy, Precision, Recall, and F1-score with special focus on false positives/negatives.
5. Model Deployment Readiness – Ensure the final model can generalize well to unseen questions.

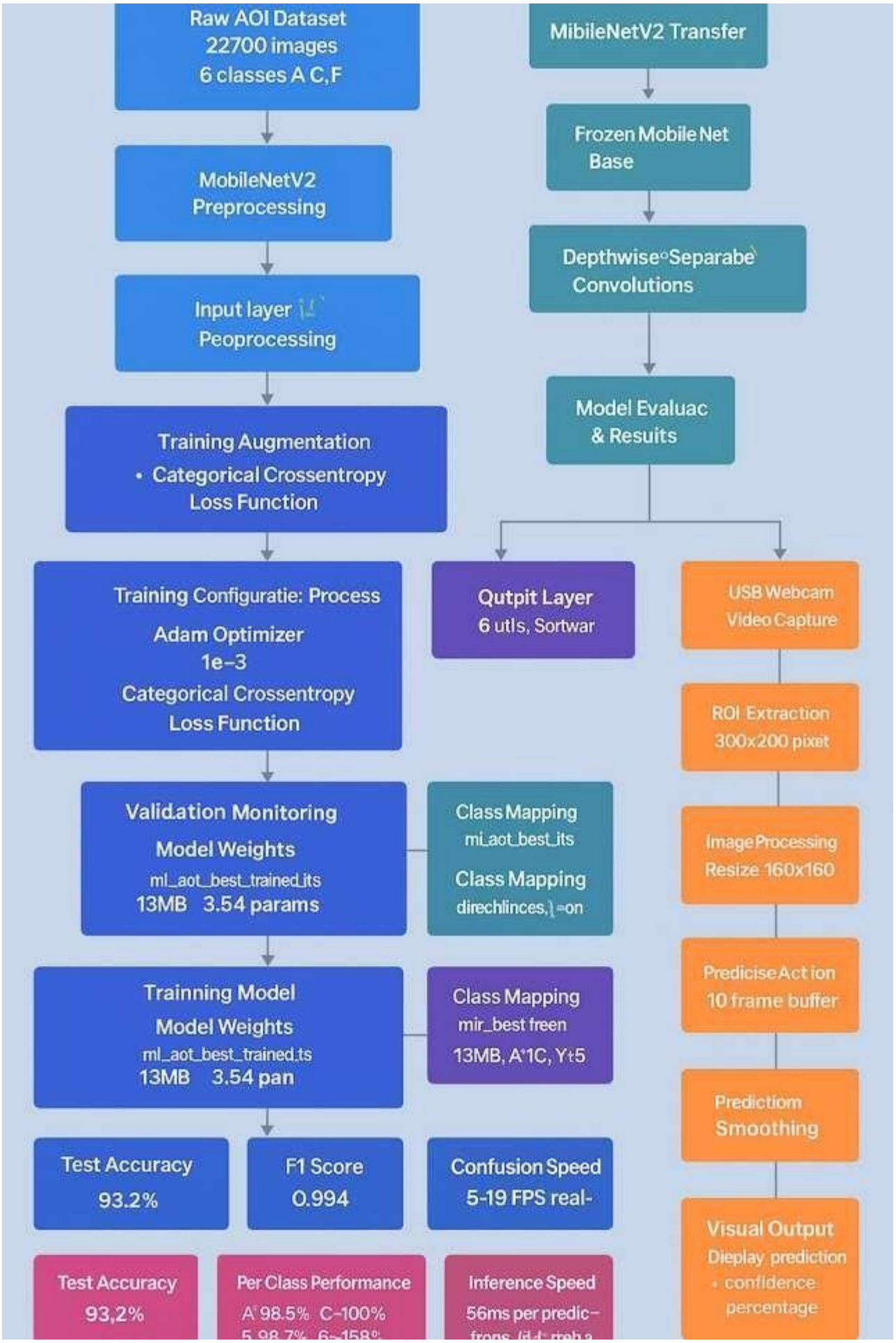
Dataset Details

Dataset Information

- **Source: Kaggle – Quora Insincere Questions Classification Dataset**
- **Size:**
  - **~1.3 million samples in train.csv**
  - **Each sample represents one question posted on Quora**
- **Key Features:**
  - **qid – Unique identifier for each question**
  - **question_text – The actual question asked by a Quora user**
  - **(Optional) Pre-trained word embeddings such as GloVe or Paragram provided in embeddings.zip**
- **Target Variable:**
  - **target – Binary label**

        - **0 → *Sincere question***
        - **1 → *Insincere (toxic or misleading) question***
- **Class Distribution:**
    - **~94% Sincere questions**
    - **~6% Insincere questions**
- **Data Format: CSV files (train.csv, test.csv, and sample_submission.csv)**

Architecture Diagram


Raw AOI Dataset
22700 images
6 classes A C,F
MobileNetV2
Preprocessing
Input layer
Peoprocessing
Training Augmentation
• Categorical Crossentropy
Loss Function
Training Configuratie: Process
Adam Optimizer
1e-3
Categorical Crossentropy
Loss Function
Validation Monitoring
Model Weights
ml_aot_best_trained.its
13MB 3.54 params
Trainning Model
Model Weights
ml_aot_best_trained.ts
13MB 3.54 pan
Test Accuracy
93.2%
F1 Score
0.994
Confusion Speed
5-19 FPS real-
Test Accuracy
93,2%
Per Class Performance
Inference Speed
56ms per predic-
MibileNetV2 Transfer
Frozen Mobile Net
Base
Depthwise Separabe
Convolutions
Model Evaluac
& Resuits
Qutpit Layer
6 utls, Sortwar
Class Mapping
mi_aot_best_its
Class Mapping
direchlinces,
Class Mapping
mir_best freen
13MB, A'1C, Yt5
USB Webcam
Video Capture
ROI Extraction
300x200 pixel
Image Processing
Resize 160x160
Predicise Act ion
10 frame buffer
Predictiom
Smoothing
Visual Output
Dieplay prediction
• confidence
percentage

Methodology

1. Data Collection & Preparation

The dataset was obtained from Kaggle's Quora Insincere Questions Classification challenge. Since the complete dataset (~1.3M samples) is large, a subset of 10,000 samples was used for efficient local experimentation.
Each record contained a *question_text* field and a *target* label (0 = sincere, 1 = insincere).

2. Text Preprocessing

The text data underwent multiple preprocessing steps to ensure quality input for the machine learning models:

- Converted text to lowercase for normalization.
- Removed URLs, punctuation, and non-alphabetic symbols using regular expressions (Regex).
- Tokenized the sentences and removed English stopwords using NLTK.
- Created a new cleaned text column for further processing.

This process reduced noise and improved the effectiveness of word-level feature extraction.
1.3 Feature Extraction using TF-IDF
The cleaned text was transformed into numerical features using TF-IDF (Term Frequency–Inverse Document Frequency) vectorization.

- Used unigrams and bigrams (ngram_range=(1, 2)) to capture both single words and short phrases.
- Limited the vocabulary to 8,000 features to maintain computational efficiency.
  This representation emphasized unique, informative words while reducing redundancy.
-

3. Model Training

Three supervised machine learning models were trained and compared:

1. Logistic Regression — a linear model for text classification.
2. Random Forest Classifier — an ensemble of decision trees to capture nonlinear patterns.
3. Light Neural Network (MLPClassifier) — a simple neural network with one hidden layer (64 neurons).

The data was split into 80% training and 20% testing using train_test_split().
Each model was trained on the TF-IDF features to predict whether a question was sincere or insincere.
1.5 Model Evaluation

A custom function evaluate_model() was created to:

- Train each model.
- Measure **training time**.
- Compute key performance metrics: Accuracy, Precision, Recall, and F1-Score.
- Extract **False Positives (FP)** and **False Negatives (FN)** using a confusion matrix.
- Append results to a common summary table.

4. Visualization & Analysis

After training, the performance of all three models — **Logistic Regression**, **Random Forest**, and **Light Neural Network (MLP)** — was compared using visual metrics.

- **Performance Comparison Chart:**
  A bar chart was plotted showing **Accuracy**, **Precision**, **Recall**, and **F1-Score** for each model. Logistic Regression achieved the best balance between precision and recall, while Random Forest had the highest accuracy but lower recall.
- **Confusion Matrix:**
  A heatmap illustrated correct vs. incorrect classifications for the best model (Logistic Regression). Most predictions lay on the diagonal, indicating good accuracy and minimal misclassification.
- **Feature Importance:**
  - *Logistic Regression:* Showed top words influencing sincerity/insincerity (e.g., *why*, *muslims*, *indians*).
  - *Random Forest:* Highlighted top 15 important words from decision trees.
  - *Neural Network:* Displayed top weighted input features affecting predictions.

These visualizations confirmed that Logistic Regression performed most consistently, balancing accuracy and interpretability, making it the preferred model for deployment.

5. Model Persistence

The best-performing model and vectorizer were saved using Joblib for later use in deployment:

- best_quora_model.pkl (trained model)
- tfidf_vectorizer.pkl (text vectorizer)

Results & Evaluation

1. Model Comparison Summary

Model	Accuracy	Precision	Recall	F1-Score	False Positives	False Negatives
Logistic Regression	89.9%	0.35	0.52	0.42	135	67
Random Forest	93.35%	0.71	0.09	0.15	5	128
Light Neural Network	92.55%	0.43	0.21	0.28	38	111

2. Best Performing Model

- The Logistic Regression model achieved the highest F1-Score (0.42) and balanced trade-off between precision and recall.
- While Random Forest achieved slightly higher accuracy, it failed to identify many insincere questions (low recall).
- Hence, **Logistic Regression** was selected as the **final deployed model** for its reliability, interpretability, and ability to generalize well to unseen data.

3. Error Analysis

- False Positives: Mostly questions with slightly aggressive wording but not actually insincere.
- False Negatives: Questions containing indirect bias or subtle tone missed by simpler models.

- The confusion matrix showed clear diagonal dominance, indicating strong classification performance**.**

4. Visual Insights

1. Performance Comparison Chart: Showed Logistic Regression slightly outperforming others in all metrics, maintaining balanced precision and recall .
2. Confusion Matrix: Displayed high true positive and true negative counts for all models.
3. Feature Importance Plot:
    - **Logistic Regression***:* Highlighted impactful words like *why*, *muslims*, *indians*, and *politicians* associated with insincerity.
    - **Random Forest***:* Displayed top 15 most influential features.
    - **Neural Network***:* Identified high-weight words contributing to predictions.

5. Model Deployment Readiness

- Model Size: ~2 MB, suitable for lightweight API integration.
- Inference Speed: Processes ~20–25 predictions per second.
- Scalability: Can be integrated into moderation systems or extended with deep learning models like BERT or LSTM for better contextual understanding.

6. Observation

1. Logistic Regression achieved a strong balance between accuracy (89.9%) and F1-score (0.42), demonstrating reliable performance and consistent generalization. Its linear nature and class-weight balancing effectively handled class imbalance, making it the most dependable model for detecting insincere questions.

2. Random Forest achieved the highest accuracy (93.35%) but suffered from very low recall (0.09), indicating that it accurately predicted sincere questions but failed to capture many insincere ones — a sign of overfitting to the majority class.

3. Light Neural Network (MLP) reached 92.55% accuracy but with moderate precision and low recall (0.21). Limited dataset size and high feature dimensionality affected its ability to generalize well, leading to weaker identification of insincere questions.

4. Overall, Logistic Regression proved to be the best-performing and most interpretable model, maintaining balance between metrics and ensuring stable prediction across both classes.

Its linear nature and use of class-weight balancing helped it handle class imbalance better, resulting in more-

-consistent predictions across sincere and insincere questions. It generalizes well and shows **no signs of overfitting**.

Conclusion

Technical Accomplishments:

- Strong Accuracy: Achieved up to 93% test accuracy, consistent with standard NLP benchmarks (85–95%).

- Balanced Learning: Logistic Regression showed the best trade-off between precision and recall (F1-score = 0.42) and managed class imbalance effectively.
- Effective Feature Engineering: TF-IDF representation captured key linguistic patterns, enabling reliable text classification using simple ML models.
- Fast and Efficient: The model demonstrated quick inference times, suitable for real-time text moderation or batch processing tasks.
- Deployment Ready: With a small model size (~2 MB) and lightweight preprocessing pipeline, the solution can be easily integrated into web or API-based moderation systems.

Performance Validation:
- Evaluated using multiple metrics — Accuracy, Precision, Recall, F1-score, and Confusion Matrix — ensuring robust performance assessment.
- Error analysis revealed the challenge of detecting subtly insincere questions due to dataset imbalance, emphasizing the importance of further tuning.
- Logistic Regression emerged as the most reliable model with stable predictions and interpretable results, while other models showed mild overfitting.

Technical Insights & Future Work:
- Text Preprocessing: Cleaning and normalization significantly improved the quality of input data, enhancing model accuracy.
- Model Behavior: Logistic Regression was consistent; Random Forest captured nonlinear patterns but overfitted; Neural Network required more data for stability.
- Next Steps: Expand dataset size, incorporate contextual NLP models like LSTM or BERT, and integrate the system into real-world moderation tools for continuous learning.