**🧠 Patient Feedback Sentiment Analyzer**

*A Dual-Model NLP Project for Healthcare Feedback Analysis*

**📘 1. Project Overview**

The **Patient Feedback Sentiment Analyzer** is an NLP-based web application designed to analyze patient feedback and classify it as **Positive**, **Negative**, or **Neutral**.  
It uses two separate models:

1. A **Baseline Machine Learning model** (TF-IDF + Logistic Regression)
2. A **Transformer model** (DistilBERT from Hugging Face)

This project highlights how both traditional ML and modern deep learning approaches can be used and compared for sentiment classification in the **healthcare domain** — where understanding patient experience is critical.

**🎯 2. Objectives**

* Automate the process of analyzing patient reviews from hospitals, clinics, or healthcare apps.
* Compare a **classical NLP pipeline** vs. a **Transformer-based model** in real-world use.
* Provide a **simple interactive interface** via Streamlit where users can test models using:
  + a single review (manual input)
  + multiple reviews (via CSV upload)
* Enable **CSV download** of analyzed results for reporting.

**🧩 3. Project Architecture**

📁 Patient\_Feedback\_Sentiment\_Analyzer

│

├── app.py → Streamlit app script

├── requirements.txt → Dependencies list

├── sample\_reviews.csv → Example input file

│

├── models/

│ ├── baseline\_lr\_model.joblib → Logistic Regression model

│ ├── tfidf\_vectorizer.joblib → TF-IDF vectorizer

│ └── distilbert\_sst2/ → DistilBERT model folder (Hugging Face format)

│ ├── config.json

│ ├── pytorch\_model.bin

│ ├── tokenizer.json

│ └── tokenizer\_config.json

│

└── assets/ (optional)

├── screenshots/

└── app\_preview.png

**⚙️ 4. Technology Stack**

| **Component** | **Technology Used** |
| --- | --- |
| Programming Language | Python 3.10+ |
| Web Framework | Streamlit |
| NLP Libraries | Transformers, Scikit-learn |
| ML Model Persistence | Joblib |
| Visualization | Matplotlib |
| Data Handling | Pandas |
| Deep Learning Backend | PyTorch |

**🧠 5. Models Explained**

**A. Baseline Model (TF-IDF + Logistic Regression)**

**Workflow:**

1. **Preprocessing:** Tokenization, stopword removal (optional).
2. **Feature Extraction:** TF-IDF Vectorizer converts text into numerical form.
3. **Model Training:** Logistic Regression learns to classify sentiments.
4. **Inference:** Model predicts sentiment labels for new text inputs.

**Why TF-IDF?**

* Lightweight and interpretable
* Effective for short and structured text data (like reviews)
* Easy to train and deploy

**Advantages:**

✅ Fast inference  
✅ Low resource usage  
✅ Easy interpretability

**Limitations:**

❌ Cannot understand context or semantics deeply  
❌ Fails on sarcasm or complex phrases

**B. DistilBERT Model (Transformer)**

**Model Info:**

* **Architecture:** DistilBERT — a distilled (smaller, faster) version of BERT
* **Source:** Hugging Face Model Hub
* **Dataset:** Fine-tuned on SST-2 (Stanford Sentiment Treebank)

**Workflow:**

1. Tokenizes text using WordPiece tokenizer
2. Encodes context-aware embeddings
3. Passes through Transformer layers
4. Outputs probabilities for Positive/Negative/Neutral classes

**Advantages:**

✅ Understands sentence context and meaning  
✅ Robust for complex phrasing  
✅ Outperforms classical models in accuracy

**Limitations:**

❌ Larger model size  
❌ Slower inference  
❌ Requires GPU for heavy batch predictions

**🖥️ 6. Streamlit App Workflow**

**User Interface**

| **Section** | **Description** |
| --- | --- |
| **Single Review Input** | User types any review text and gets predictions instantly |
| **CSV Upload** | Upload .csv file with text column for batch predictions |
| **Results Table** | Displays Baseline and DistilBERT predictions side-by-side |
| **Visualization** | Bar chart comparing sentiment distribution of both models |
| **Download Button** | Exports analyzed data to sentiment\_predictions.csv |

**🧮 7. Sample Workflow (Example)**

**Input:**

*“The doctor was rude.”*

**Output:**

| **Model** | **Prediction** |
| --- | --- |
| Baseline (TF-IDF + LR) | 2 (Negative) |
| DistilBERT | Negative |

**Observation:**  
Both models agree that sentiment is negative, but DistilBERT provides a clearer, interpretable label.

**📊 8. Sample CSV File**

| **text** |
| --- |
| The doctor was rude. |
| The nurse was kind and helpful. |
| The waiting time was too long. |
| I’m very satisfied with my surgery results. |

**📂 9. How to Run the Project**

**Step 1: Clone or download the repository**

git clone https://github.com/yourusername/patient-feedback-sentiment-analyzer.git

cd patient-feedback-sentiment-analyzer

**Step 2: Install dependencies**

pip install -r requirements.txt

**Step 3: Run Streamlit app**

streamlit run app.py

**Step 4: Test the models**

* Enter a sample review OR
* Upload a CSV file (text column required)
* Download predictions

**📈 10. Sample Output**

**Sentiment Comparison Table**

| **Text** | **Baseline** | **DistilBERT** |
| --- | --- | --- |
| The doctor was rude. | Negative | Negative |
| The nurse was kind. | Positive | Positive |
| Long waiting time. | Negative | Negative |
| Staff was very friendly. | Positive | Positive |

**Visualization Example:**  
Bar chart comparing counts of Positive vs. Negative predictions across models.

**🧩 11. Key Learnings**

* Difference between traditional ML (TF-IDF + Logistic Regression) and deep learning (Transformers) in NLP.
* Building an interactive web app using **Streamlit**.
* Handling **text-based CSV inputs and outputs** for real-world applications.
* Structuring ML projects for **deployability and portfolio demonstration**.

**🧠 12. Future Enhancements**

* Add multilingual support (e.g., Hindi, Malayalam feedback).
* Fine-tune DistilBERT on domain-specific medical datasets.
* Include explainability tools like **LIME** or **SHAP** for transparency.
* Add **sentiment trend analytics dashboard** for hospitals.
* Store feedback analysis results in a database for longitudinal insights.

**👤 13. Author**

**Pranav P**  
NLP & Data Science Enthusiast  
📍 Kerala, India  
📧 pranavprabhash93@gmail.com