Binary Sentiment Classification on IMDB (25k reviews)

1) Data Description

Dataset. We use the **IMDB Large Movie Review** dataset for binary sentiment analysis (positive vs. negative). The official training split contains **25,000 labeled reviews**; we reserve **20% (~5,000)** as a validation set with a fixed random seed for reproducibility.

Features & labels. Each instance is a free-form English review (text) with a binary label (target ∈ {0,1}), where 1 denotes *positive* sentiment and 0 denotes *negative*. Reviews are relatively long (often >200 tokens), mixing colloquialisms and formal prose.

Pre-processing. We unescape HTML entities; remove URLs and @mentions; keep hashtags as tokens; lowercase; and normalize whitespace. For Keras models we build a 30k token vocabulary and cap sequences at **128 tokens**; for DistilBERT we use its WordPiece tokenizer with **max_len=128**. No lemmatization/stemming was applied to preserve sentiment-bearing word forms.

Splitting & reproducibility. Stratified **80/20** train/validation split (seed=42). All random seeds for NumPy/TensorFlow/PyTorch are fixed.

---

2) Main Objective

Goal. Develop and compare several NLP models that **classify IMDB reviews as positive or negative**, and **select the model that best balances precision and recall** (F1) for general deployment.

Business/operational rationale. A high-F1 sentiment classifier supports product analytics and content moderation by reducing manual review while maintaining accuracy.

---

3) Deep Learning Variations & Best Model

Models evaluated

1. **Baseline:** TF-IDF (1–2 grams) + Logistic Regression

2. **TextCNN:** parallel 1D convolutions (kernel sizes 3/4/5) + max-pool + dense
3. **BiLSTM:** embedding → bidirectional LSTM → global max-pool → dense
4. **DistilBERT:** distilbert-base-uncased fine-tuned (linear head on [CLS])

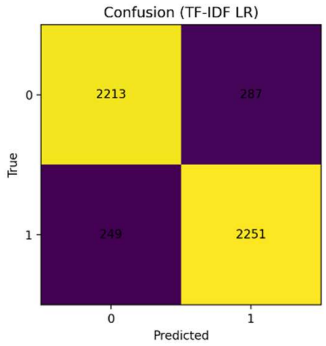
Training details. Binary cross-entropy (or logits variant); early stopping; metrics recorded on the hold-out validation set: **F1 (primary), Precision, Recall, ROC-AUC, PR-AUC**.

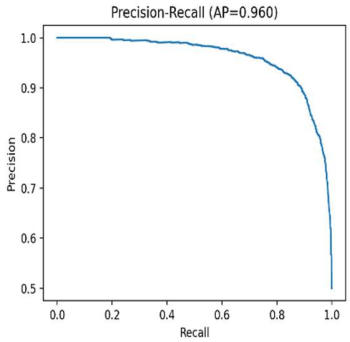
Model selection criterion. Primary: **highest validation F1** (balanced precision/recall). Secondary: ROC-AUC / PR-AUC and stability.

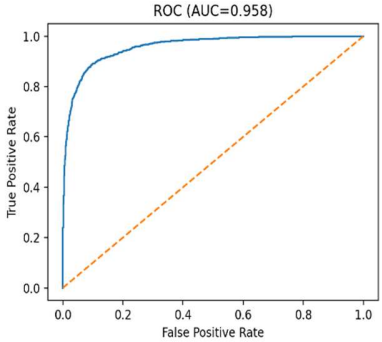
Result: best overall model — TF-IDF + Logistic Regression.
It achieved the top F1 with excellent AUCs (table below). DistilBERT was a close second. If recall is prioritized over precision (catching positives at the cost of more false alarms), the **BiLSTM** is preferable due to its very high recall.

Validation metrics (n≈5,000)

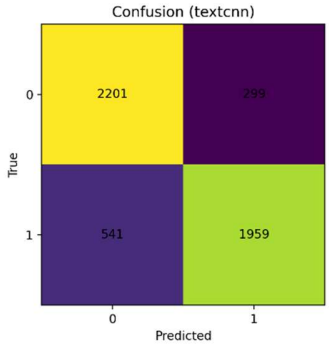
Baseline Figures

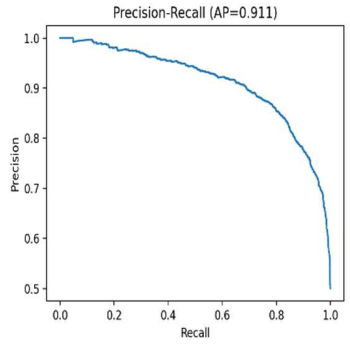


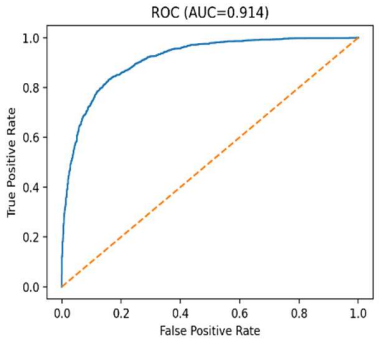




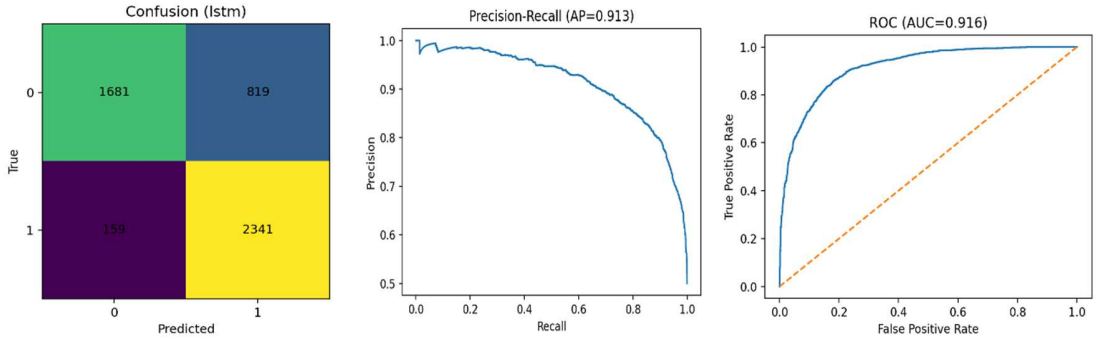
CNN Figures



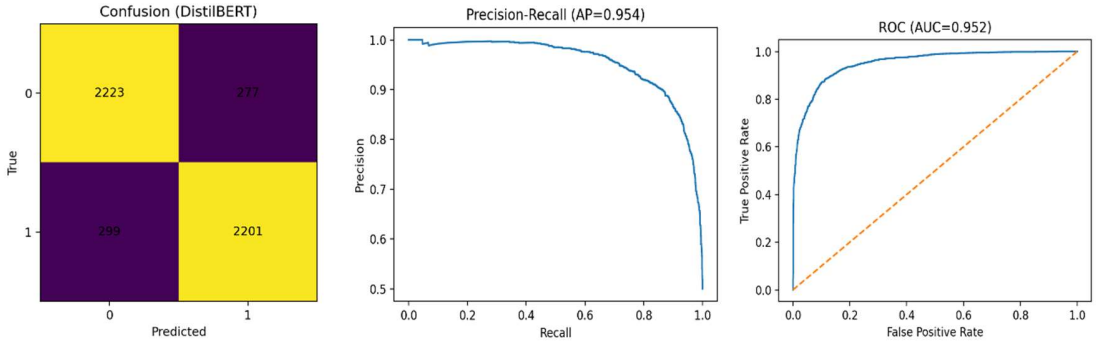




LSTM Figures



BERT figures



4) Key Findings (linked to the objective)

1. **Classic bag-of-words remains a strong baseline.** On long IMDB reviews, **TF-IDF + Logistic Regression** slightly **outperforms DistilBERT** on F1 and both AUCs (0.894 F1; ROC-AUC 0.958; PR-AUC 0.960).
2. **Transformers benefit from longer context.** With max_len=128 and 3 epochs, DistilBERT underutilizes full reviews; nevertheless it reaches F1=0.884, AUCs ≈0.95.
3. **Recall vs. precision trade-off.** The **BiLSTM** attains **very high recall (0.936)**—useful when missing positives is costly—while the **baseline is more precise**.
4. **TextCNN** is competitive but trails on long, nuanced text, reflecting its local-n-gram bias and the 128-token cap.

Why this matters. For general deployment where balanced accuracy is desired, the baseline is the simplest and strongest. If the application demands "catch every positive review," the BiLSTM's recall is attractive; DistilBERT is the best candidate to improve with modest tuning.

---

5) Limitations & Plan to Revisit

Limitations / possible flaws

- **Sequence truncation.** Transformers and Keras models were capped at **128 tokens**; IMDB reviews are longer, likely suppressing transformer gains.
- **Limited hyperparameter search.** DistilBERT trained for only ~3 epochs with default scheduling; deeper sweeps could narrow the gap.
- **Single split.** Results reported on one 80/20 split; cross-validation would better quantify variance.
- **Class-threshold not optimized.** All models used a 0.5 threshold; picking the PR-optimal threshold can improve F1 or tailor precision/recall.

Actionable plan

1. **Increase context for transformers** (max_len 256–320; dynamic padding) and train **3–4 epochs** with early stopping on PR-AUC.
2. **Try stronger encoders** (RoBERTa-base, DeBERTa-v3-base); they typically add **+1–3 F1** on IMDB.
3. **Threshold calibration** from the PR curve to match the operational target (precision-first vs recall-first); report cost-sensitive metrics if applicable.
4. **Ensemble** TF-IDF LR + DistilBERT (soft voting) for a small but consistent lift.
5. **Robustness checks:** 5-fold stratified CV; multiple seeds; calibration plots (reliability curves).
6. **Error analysis:** Audit top FPs/FNs to guide data cleaning (e.g., sarcasm, negation, domain slang) and targeted augmentation.