Detecting Fraudulent Job Postings with DistilBERT and Text Classification

Real (0) vs Fake (1) classification using job text + metadata-as-text
Result badge: Acc 0.9424 | F1 0.9429

J. Hastings 1/27/2026

Executive summary:

- **Goal:** Classify job postings as Real (0) vs Fake (1)
- **Best model:** DistilBERT (Hugging Face)
- **Top results (hold-out test):** Acc 0.9424 | F1 0.9429 | Recall 0.9538
- **Key takeaway:** Transformer model outperformed TF-IDF baselines (NB/SVC/Ridge/SGD)

Problem & Purpose

- **Goal:** Classify job posting text as **Real (0)** vs **Fake (1)** using a labeled dataset
- **Why it matters:** Fake postings reduce trust, waste job seekers' time, and can lead to scams; faster flagging helps platforms review/remove bad posts sooner
- **Who benefits:**
  - **Job seekers:** fewer scams, more time on real opportunities
  - **Employers/platforms:** cleaner listings and a more reliable applicant pool
- **Why it's challenging:** Fake posts can look legitimate, so **context** (phrasing/tone) matters—not just keywords
- **What I did (contribution):** Built and benchmarked **DistilBERT vs TF-IDF baselines** (NB/SVC/Ridge/SGD)
- **Success criteria:** High **Accuracy / Precision / Recall / F1** (plus error review via confusion matrix)
  - **Best model: DistilBERT — Acc 0.9424, F1 0.9429**

Related Work

What others have done

1) Problem framing

- Fraud job posting detection = supervised text classification (real vs. fraud)

2) Baselines(standard practice)

- Represent text using **TF-IDF / word importance features**
- Train classic classifiers: *Logistic Regression, SVM, SGD Linear*
- Why baselines matter: fast + clear starting point for comparison

3) Modern approach + what I did

- Transformers: pre-trained language models fine-tuned on labeled job posts
- My approach: *DistilBERT-base-uncased* (captures more context than word counts)
- Tuning matters: results can change based on training settings (hyperparameters)

4) Evaluation focus

- Not just accuracy: false positives vs false negatives matter
- Report Accuracy/precision/recall/F1 + error breakdown (confusion matrix)

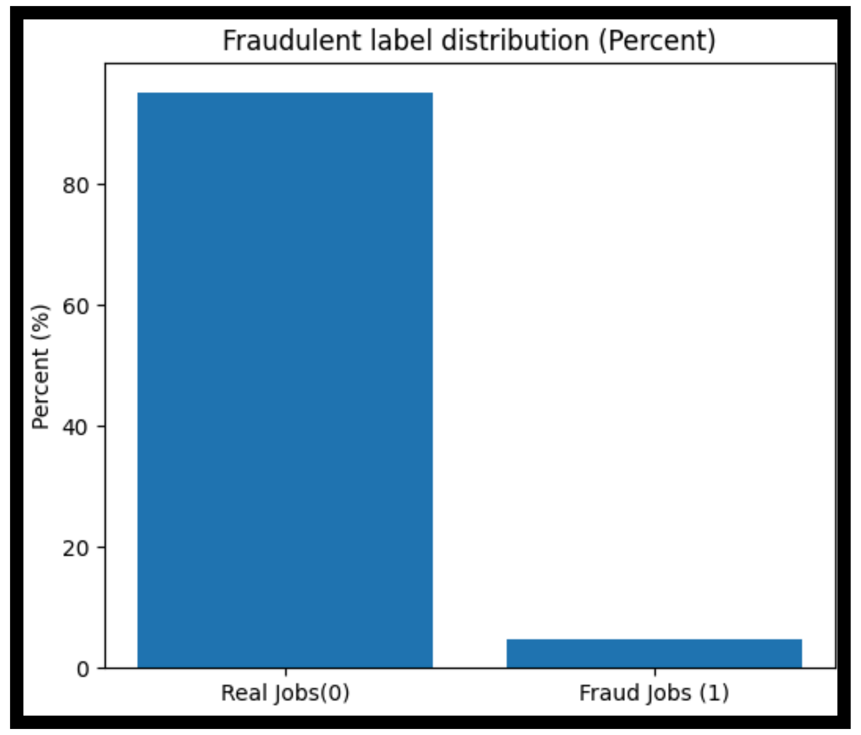
Proposed Work

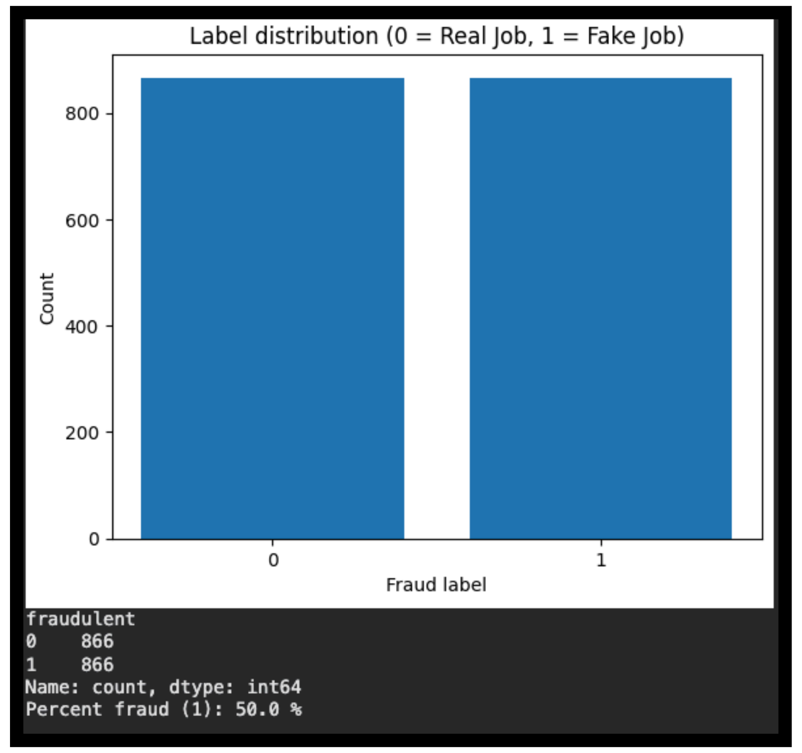
1. **Data:** Loaded the Kaggle job postings dataset with job text + metadata and a fraudulent label (Real=0, Fake=1); checked label encoding and class distribution; fixed target variable from unbalanced to balanced.
2. **EDA**: Inspected columns and ran EDA to understand distributions and which features were most useful for predicting the target.
3. **Preprocessing / Feature construction**: Cleaned text (handled missing/messy entries, removed extra whitespace) and created a single model input column **`final_text`** by combining the selected text fields.
4. **Split:** Created train/validation sets using a 90/10 split with random_state=42 for reproducibility.
5. **Modeling:** Trained DistilBERT-base-uncased (Hugging Face + PyTorch); tokenized **`final_text`**, used padding for batching, and tuned learning rate, epochs, and max token length (512).
6. **Baselines (benchmarking)**: Built classic text baselines using **TF-IDF** (e.g., Logistic Regression / SVM / Naive Bayes) for comparison.
7. **Evaluation + iteration**: Evaluated with **accuracy, precision, recall, F1**, and a **confusion matrix**, then used results to guide model and preprocessing adjustments.

Data

Kaggle: "Real/Fake Job Postings"
Size overview: 17881 rows & 18 columns

Troubleshoot : Data include's **missing values**, **class imbalances** & **whitespaces**.





These features have the most impact on the target variable(fraudulent) and were combined into a variable named `final_text`.

Chi-square test (Top P-value & Cramer's V for categorical features):

- Industry(4.420e-58/0.545)
- state(3.007e-29/0.483)
- has_company_logo(3.560e-89/0.481)

Logistic regression(features with highest f1 score for text data):

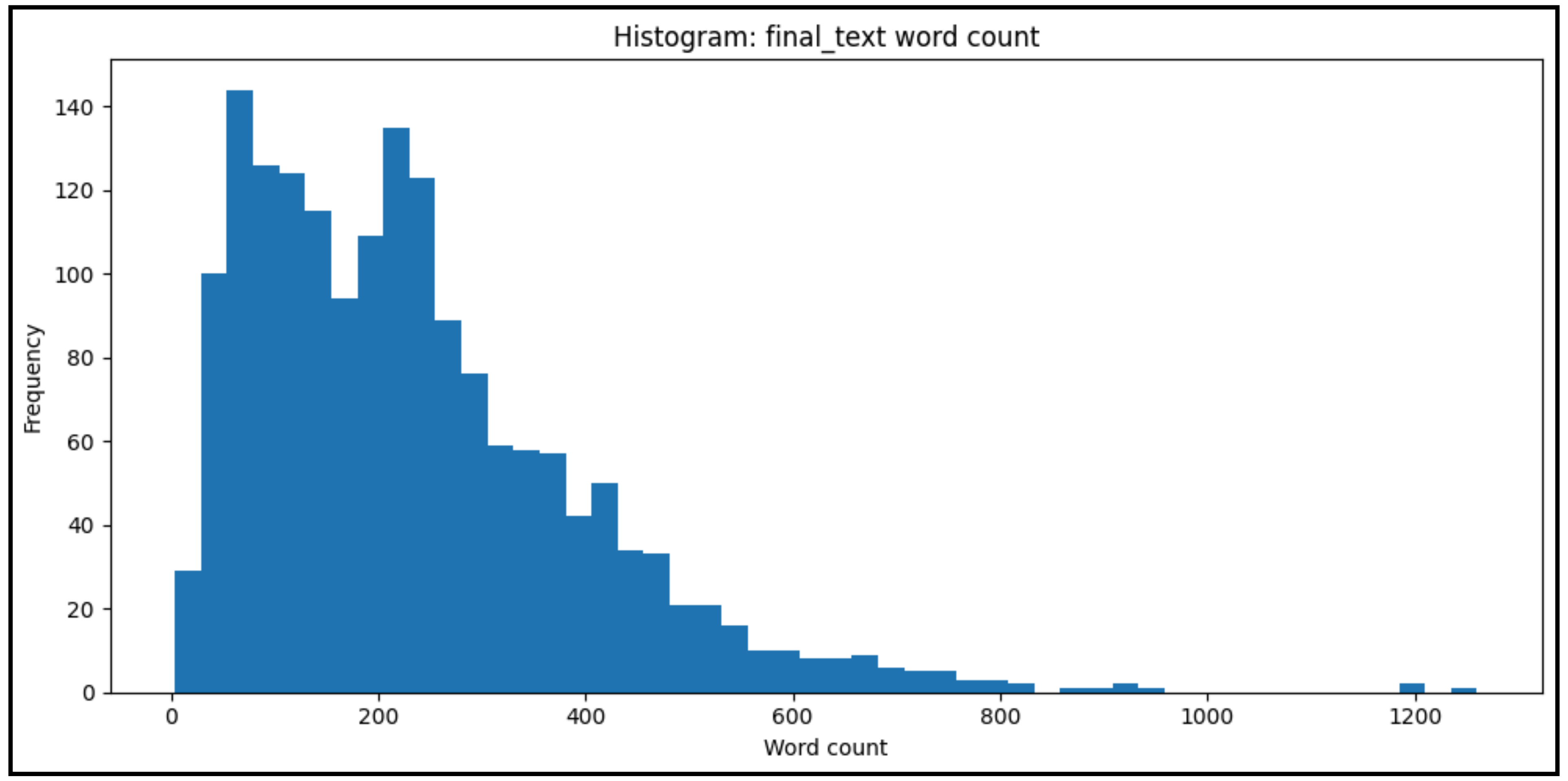
- company profile(0.905)
- description(0.869)

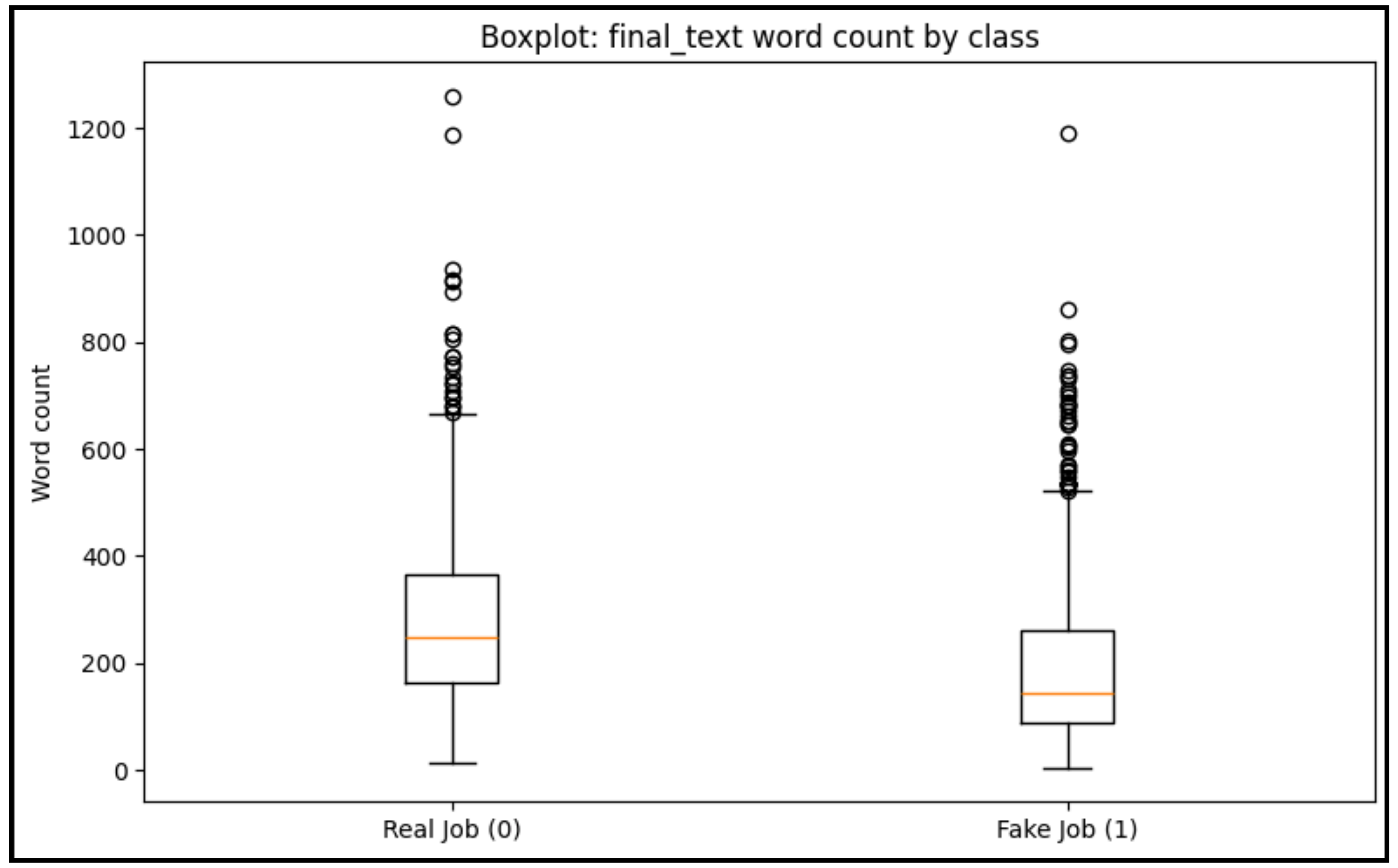
Data continued…

What these plots show: How long the combined job-post text is overall (histogram) and how lengths differ between **Real (0)** vs **Fake (1)** (box-plot).

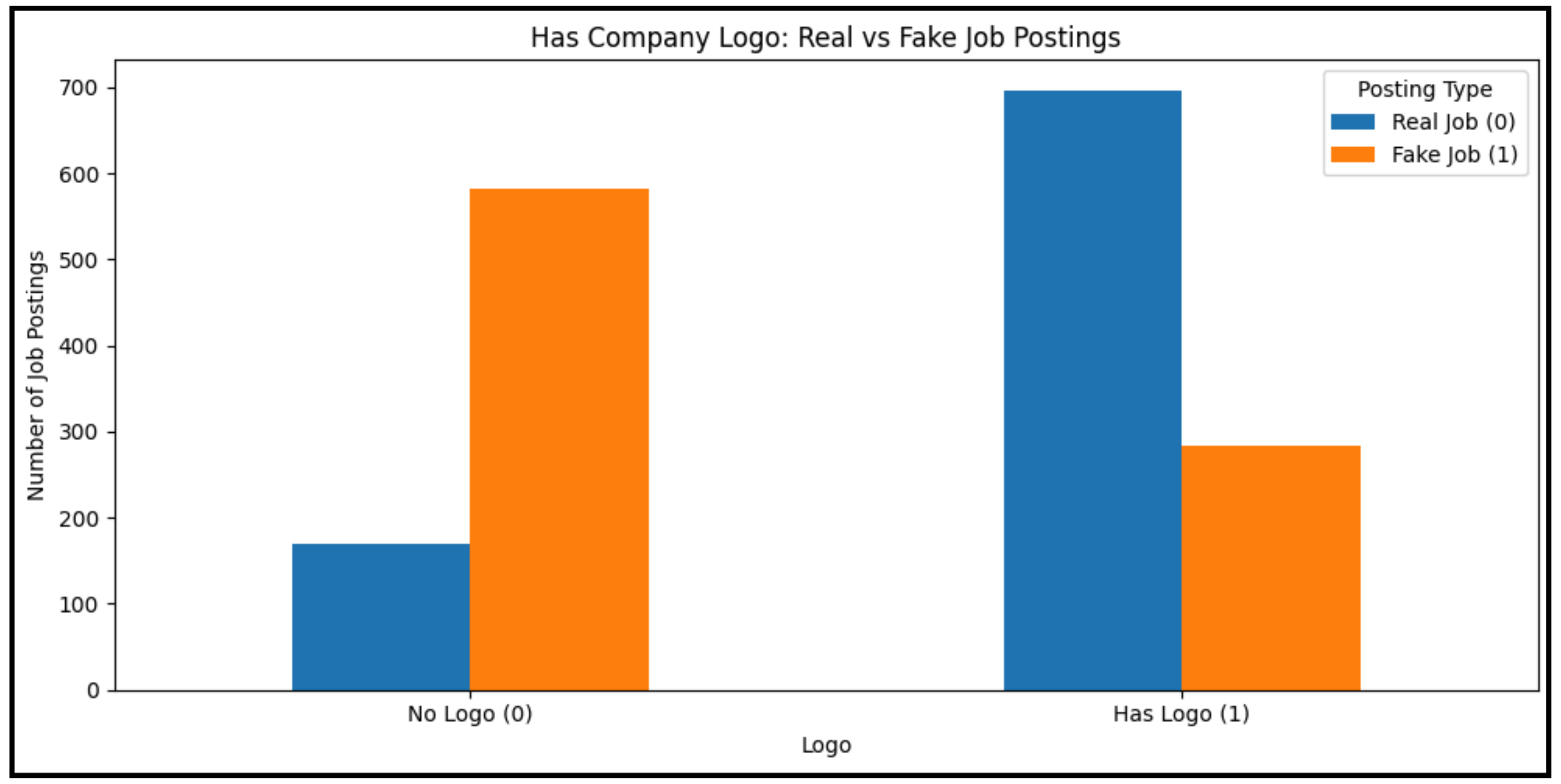
Key takeaway: Most posts are a few hundred words, but some are **very long** (over 1,000 words). Real Job posts tend to be **longer on average**, though both classes have long outliers.

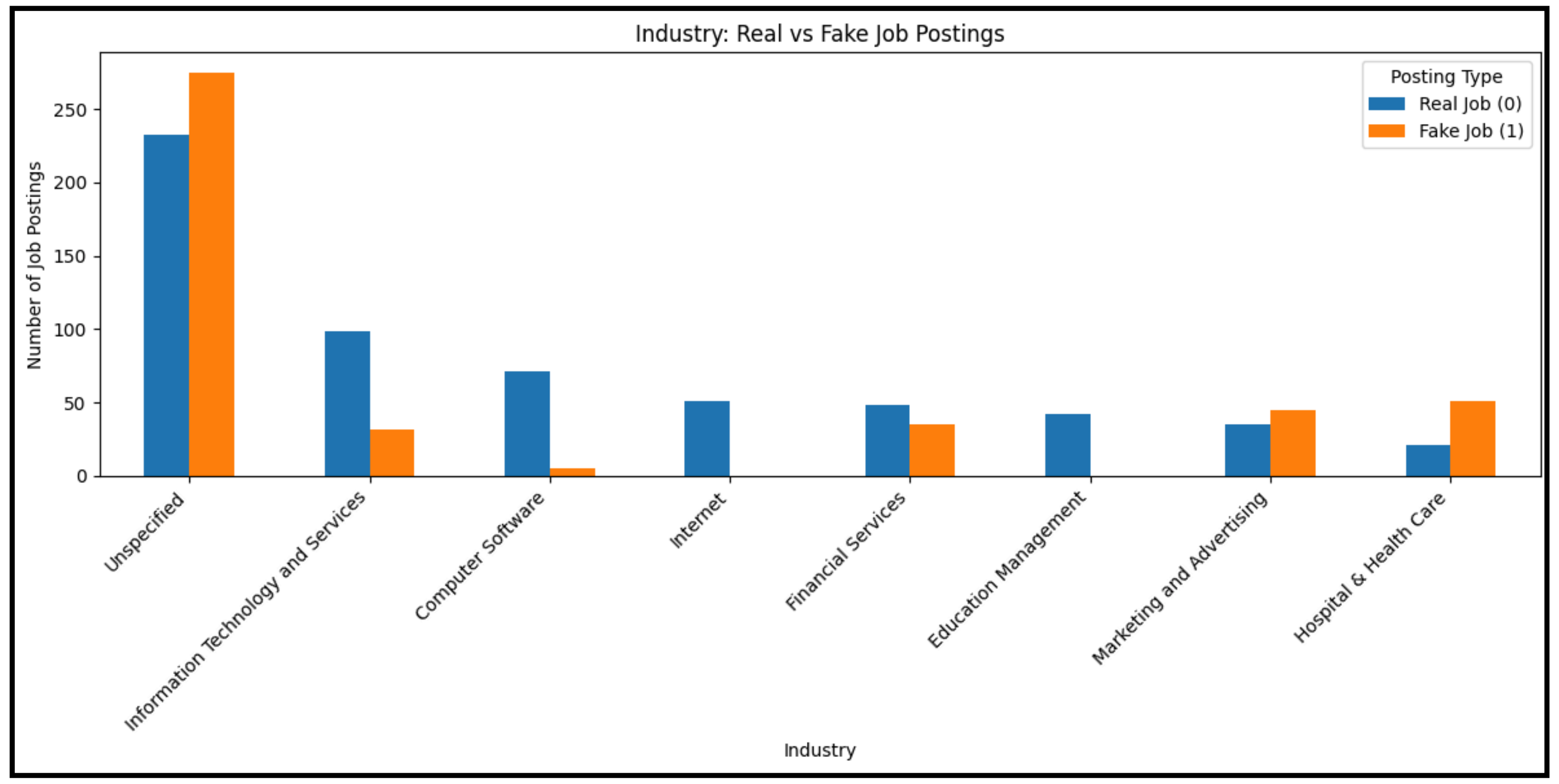
Why I made these: To understand input size before modeling so I can choose a reasonable **max_length**, estimate **truncation risk(important info that gets cut off)**, and see whether text length might relate to the label.

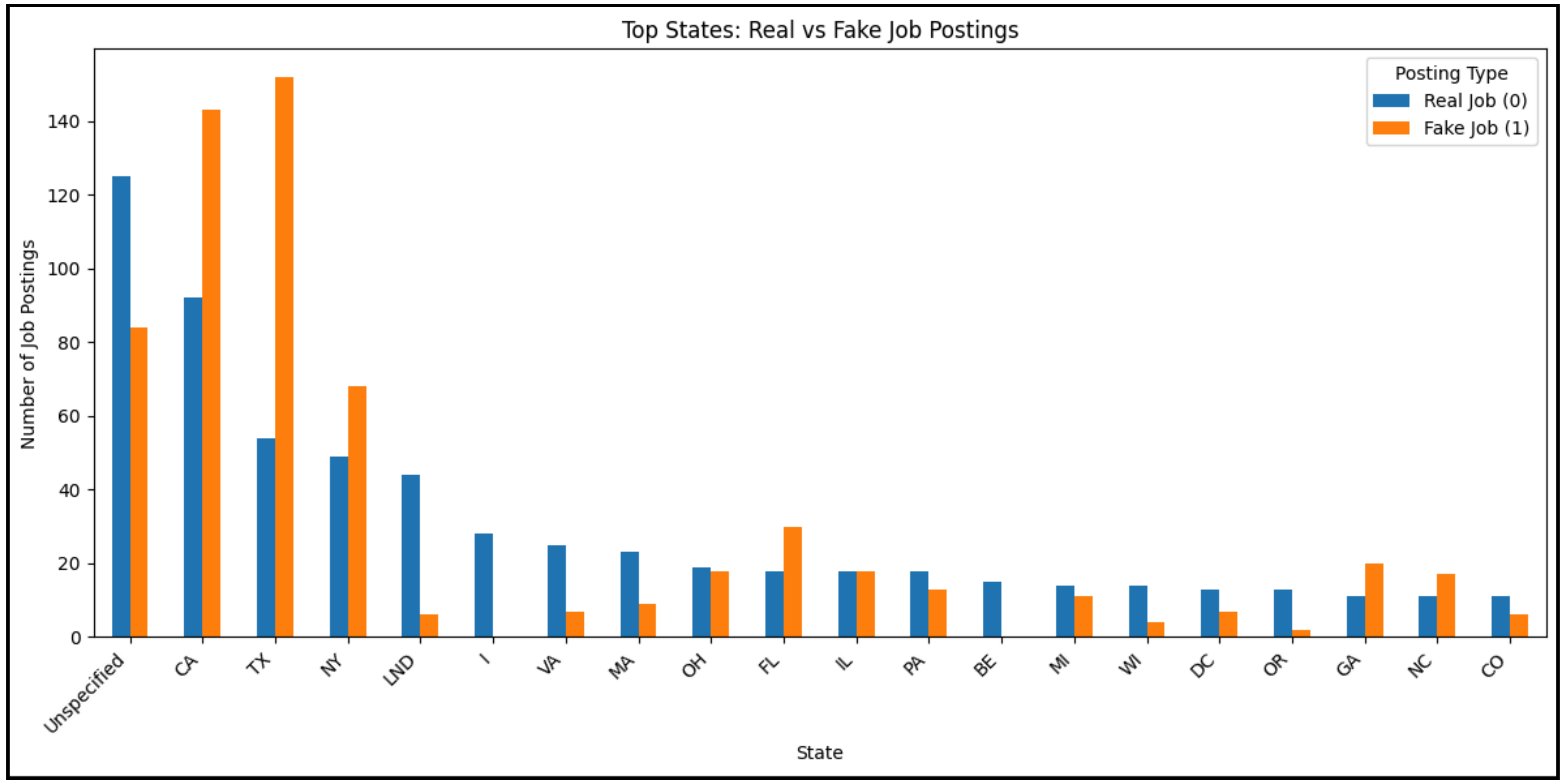




Exploratory Data Analysis:







Modeling details (DistilBERT) & Training Setup

Model used

- **DistilBERT-base-uncased** (Hugging Face Transformers + PyTorch)
- Task: binary text classification (real vs fake).

Token limit + truncation

- Combined text feature: **final_text** (job description + metadata-as-text)
- Token limit: model reads up to **512 tokens** → longer posts are **truncated**.
- Tested max_length = 512 (baseline/benchmark) vs 385 (tuned) to study context vs speed tradeoff.

Training + tuning (what changed)

- Tuned: **learning rate**, **epochs**, **max_length**
- Final settings used:
  - **Baseline:** LR **0.0001**, Epochs **10**, Max length **513**
  - **Tuned:** LR **0.0001**, Epochs **4**, Max length **385**, Weight decay **0.01**, Warmup ratio **0.06**
  - Early stopping (patience=2) used to reduce overfitting

Why these choices:

- Longer max_length keeps more context; shorter max_length speeds training but can increase truncation risk.

Evaluation & Key results

Evaluation set up:

- **Hyperparameter tuning:** Train/Validation split (pick best settings using validation performance)
- **Final benchmarking:** Train/Test split (hold-out test set for final model comparison)
- **Fair comparison:** Same train/test split used across all benchmarked methods
- **Metrics**: Precision/Recall/F1 & confusion matrix
- **Latency**: Runtime/training time (how fast it predicts)
- **Error review**: review common misclassifications.

Current test results (DistilBERT)

- **Accuracy**: 0.9306 (173 test samples)

Per-class performance

- **REAL job (0)**: Precision 0.9390 | Recall 0.9167 | F1 0.9297 (n=84)
- **FAKE job (1)**: Precision 0.9231 | Recall 0.9438 | F1 0.9333 (n=89)

What this means:

- **Catches most fraud:** High FAKE recall (0.9438) → few fake jobs are missed
- **Main tradeoff:** Some false alarms → 5 real jobs flagged as fake

```
Baseline Transformer
LR = 0.0001 | Epochs = 10 | Max length = 513

               precision    recall  f1-score   support

REAL JOB (0)     0.9390    0.9167    0.9277        84
FAKE JOB (1)     0.9231    0.9438    0.9333        89

    accuracy                         0.9306       173
   macro avg     0.9311    0.9302    0.9305       173
weighted avg     0.9308    0.9306    0.9306       173

Confusion matrix (rows=true, cols=pred):
 [[77  7]
 [ 5 84]]


Final Tuned Transformer
LR = 0.0001 | Epochs = 4 | Max length = 385 | Weight
decay = 0.01 | Warmup ratio = 0.06

               precision    recall  f1-score   support

 REAL JOB(0)     0.9286    0.9286    0.9286        84
 FAKE JOB(1)     0.9326    0.9326    0.9326        89

    accuracy                         0.9306       173
   macro avg     0.9306    0.9306    0.9306       173
weighted avg     0.9306    0.9306    0.9306       173

Confusion Matrix:
 [[78  6]
 [ 6 83]]
```

Evaluation continued..

Evaluation: Model Benchmark Results (Hold-out Test)

	Classifier	Accuracy	Precision	Recall	F1
0	DistilBERT (Hugging Face)	0.942363	0.932203	0.953757	0.942857
1	Multinomial NB	0.922190	0.929412	0.913295	0.921283
2	Linear SVC	0.922190	0.929412	0.913295	0.921283
3	Ridge Classifier	0.916427	0.928571	0.901734	0.914956
4	SGD (linear)	0.910663	0.927711	0.890173	0.908555

Interpretation:

- **Best overall:** DistilBERT achieved the top **Accuracy (0.9424)** and **F1 (0.9429)**.
- **Key strength:** Highest **Recall (0.9538)** → catches most fake jobs, with fewer missed fraud cases.

Challenges / changes

Current obstacles

- **Token limit:** combined text sometimes too long → **truncation risk**.
- **Truncation risk:** If important text gets cut off, it can hurt accuracy and cause wrong predictions.
- **Training time:** tuning transformers is slow → limited runs to key hyperparameters
- **Label noise:** borderline posts → labels may be unclear
- **Fixes:** tested max lengths (385 vs 512), focused tuning, and did error review to diagnose failures

Token Length & Truncation Check

- final_text token lengths: **p50=307, p75=468, p90=653, p95=808, p99=1104**
- **Meaning:** Many posts are **longer than 512 tokens**, so shorter limits would cut off important text
- **Decision:** Used **max_length=512** for benchmark; tested **385** during tuning (faster, more truncation)

Error review

Baseline vs Tuned DistilBERT (same accuracy, different errors)

Baseline Transformer (max_len=513)

- **Confusion matrix (true → prediction): [[77, 7], [5, 84]]**
- **False Positives (Real→Fake):** 7
- **False Negatives (Fake→Real):** 5
- **Takeaway:** Slightly more **false alarms** than missed fraud

Final Tuned Transformer (max_len=385)

- **Confusion matrix (true → prediction):** [[78, 6], [6, 83]]
- **False Positives (Real→Fake):** 6
- **False Negatives (Fake→Real):** 6
- **Takeaway:** Fewer false alarms, but missed fraud increased by 1

What this means

- Both models have the same overall accuracy (**0.9306**), but the **error tradeoff shifts**.
- Baseline (513) is slightly better at **not missing fake jobs** (FN=5), while tuned (385) is slightly better at **not flagging real jobs** (FP=6).

What I’m looking for in error review

- Real posts flagged as fake: **vague company info**, **unrealistic benefits**, “scammy” phrasing.
- Fake posts missed: **professional tone** / normal business language.
- Possible truncation effect: shorter max length (**385**) may remove late details that help classification.

Timeline

Week 1: Cleaned data, converted metadata → text, built combined input, ran dataset checks

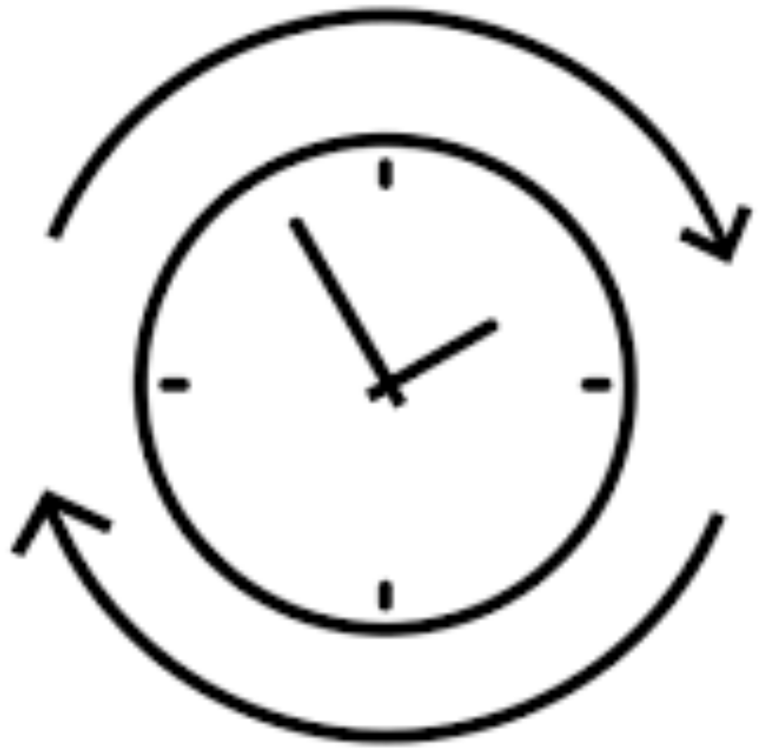
Week 2: Trained DistilBERT and tuned hyperparameters using train/validation splits

Week 3: Benchmarked models using a single train/test split, ran full evaluation + error review, finalized report/ slides

Delivered: Final benchmark table + key visuals + written report + presentation/video

Future Work

- Expand baseline comparisons and additional splits/datasets (generalization)
- Try other transformer models (e.g., RoBERTa)
- Add more error analysis/interpretability visuals



Conclusion

- Built an end-to-end pipeline to classify job postings as **Real (0)** vs **Fake (1)** using **DistilBERT** + tuned training setup
- **Benchmark result (hold-out test):** DistilBERT achieved Accuracy = 0.9424 and F1 = 0.9429 (best among compared models)
- **Key takeaway:** Transformer model captured context better than TF-IDF baselines, leading to stronger overall performance
- **Error insight:** Remaining mistakes include both **real flagged as fake** and **fake missed**, shown via confusion matrix + brief error review
- **Future work:** Try a larger transformer (e.g., **RoBERTa**) and test generalization on additional datasets

Appendix: Learning Rate / Epoch / Max Length Sweeps

LR sweep: “Used to select LR with lowest validation loss / highest validation accuracy.”
Epoch curves: “Used to pick stopping point (best epoch ~4) and avoid overfitting.”
Max_length sweep: “Used to compare context vs truncation/compute tradeoff.”