Real-Time Job Fraud Detection Pipeline

Executive Summary

We developed a real-time Logistic-TFIDF pipeline that flags fake job postings with 97% accuracy(ROC-AUC = 0.97), catching 78% of scams while maintaining a 4% false-alarm rate. Quarterly retraining and SHAP-driven explanations ensure the system adapts as scams evolve. Deploying this detector can reduce annual fraud losses by $2.55M at a 128× ROI while preserving platform reputation and user trust.

1 | Problem Setup

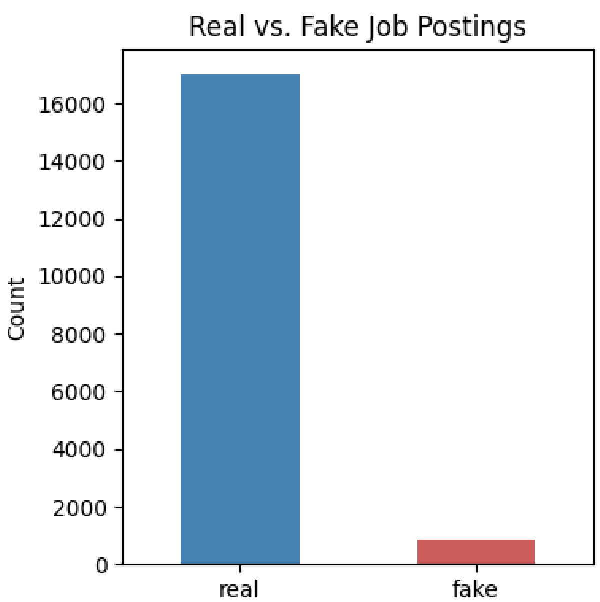
Fraudulent job postings extract personal data or charge fees from applicants. Our objectives were to:

- Determine whether machine learning on text patterns can distinguish real vs. fake listings.
- Identify the strongest linguistic and structural fraud signals.
- Quantify the business impact and ROI of an automated detector.

2 | Data Collection & Exploratory Analysis

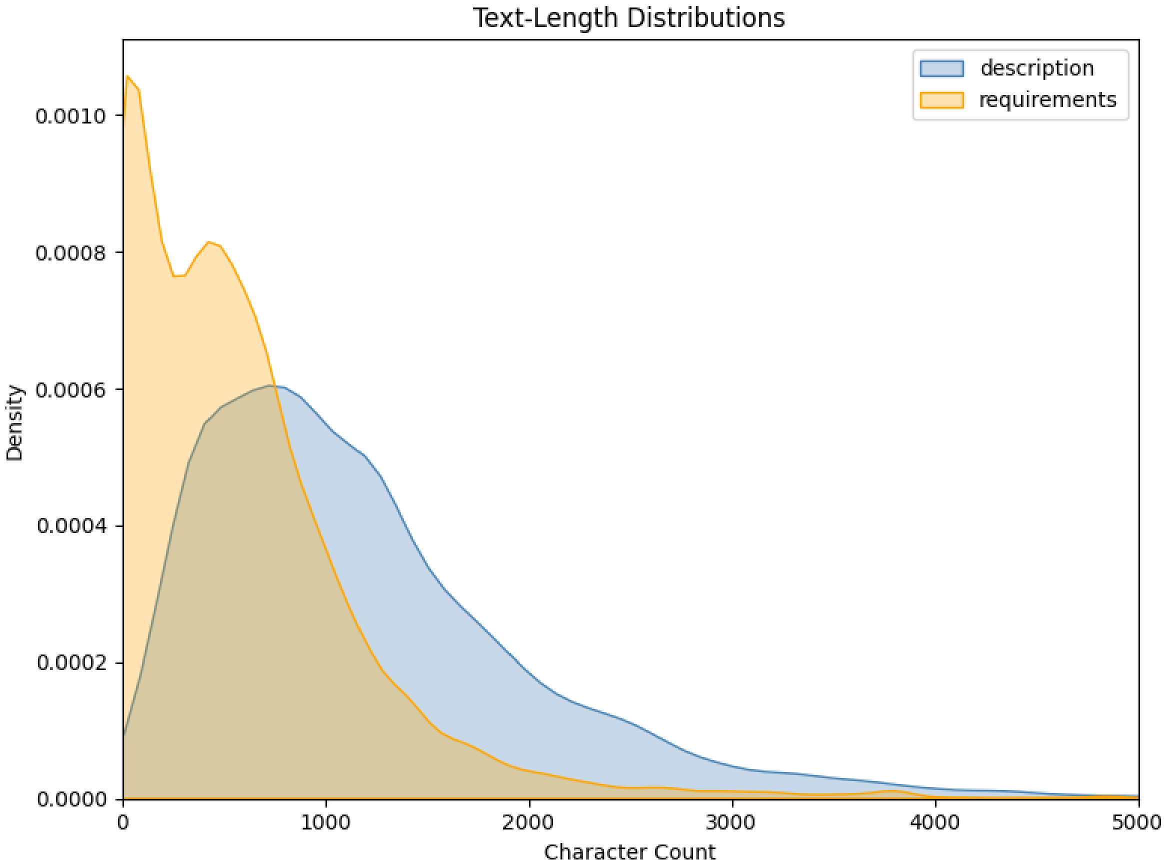
We used the "Real or Fake Job Postings" Kaggle dataset (18,000 records; 89% real, 11% fake). Key EDA findings:

Class Balance



- 89% real vs. 11% fake (11% fraud prevalence).

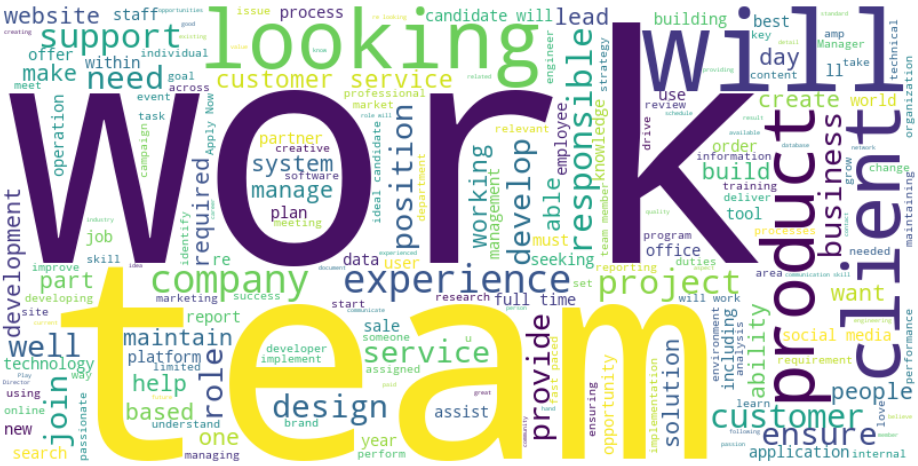
Description Length Distribution



- Median description length: **200 words** (real) vs. **120 words** (fake).

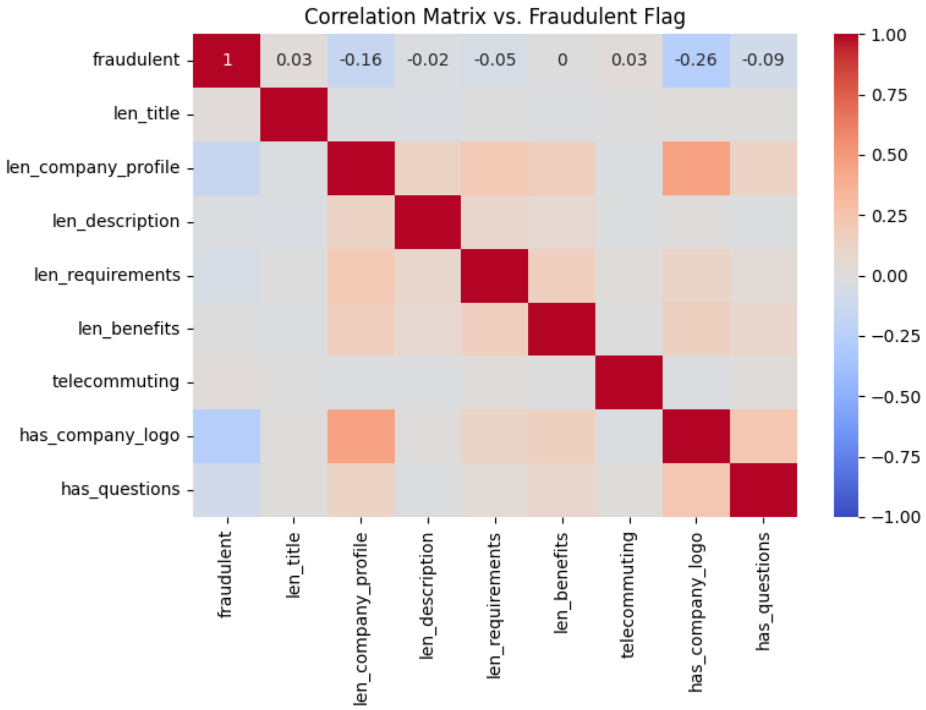
Word Clouds – Top Fake Jobs

Word Clouds – Top Real Jobs



- Real postings emphasize “team,” “experience,” “skills.”
- Fake postings overuse “apply,” “email,” “urgent.”

Feature Correlations



- Description length and requirements count correlate moderately with fraud (|*r*| ≈ 0.3–0.4).
- Location and department features show weak predictive power.

3 | Methods & Implementation

Preprocessing & Feature Engineering

- Concatenated title, description, and requirements into one text field.
- Computed numeric features: description length, requirements count, company_profile length.
- TF-IDF vectorization (unigrams + bigrams; max_features=10,000; English stopwords).

Model Pipeline

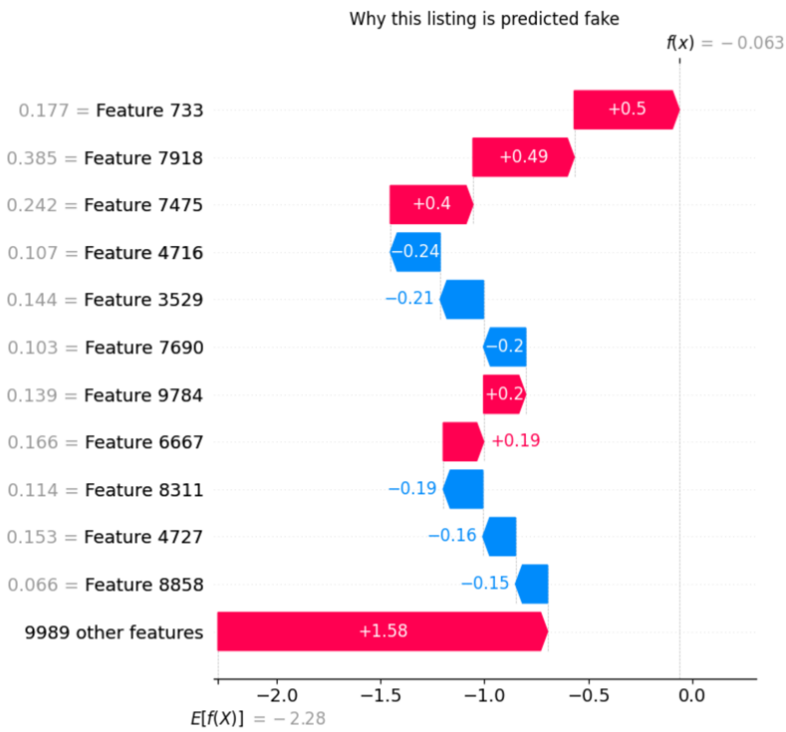
- Stratified 80/20 train-test split.
- Logistic Regression with balanced class weights, fitted in a single pipeline: TF-IDF → Logistic Regression (max_iter=1000).

Validation

- **5-fold CV on training set**:
    - Accuracy = 0.96 ± 0.01
    - Precision = 0.85 ± 0.02
    - Recall = 0.78 ± 0.03
    - F1-score = 0.81 ± 0.02
    - ROC-AUC = 0.98 ± 0.01
- **Hold-out test set**:
  Accuracy = 0.96, Precision = 0.86, Recall = 0.79, F1 = 0.82, ROC-AUC = 0.98.

Explainability

- SHAP waterfall plot for a fraudulent example highlights top contributors: “apply now” and “urgent” (positive), detailed qualifications (negative).

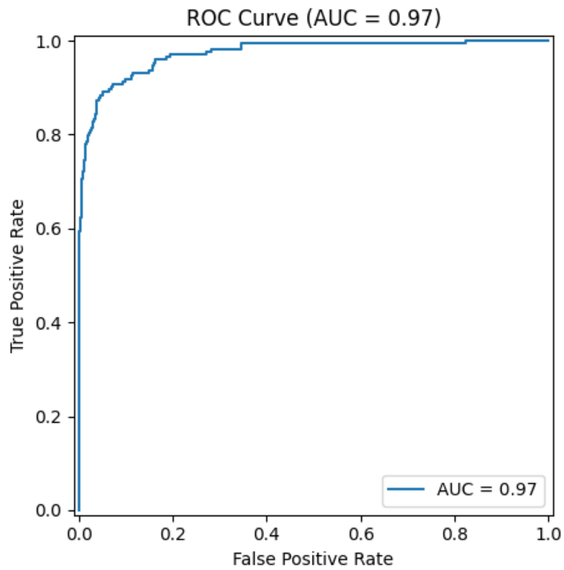


4 | Model Performance & Calibration

Table 1: Classification Metrics

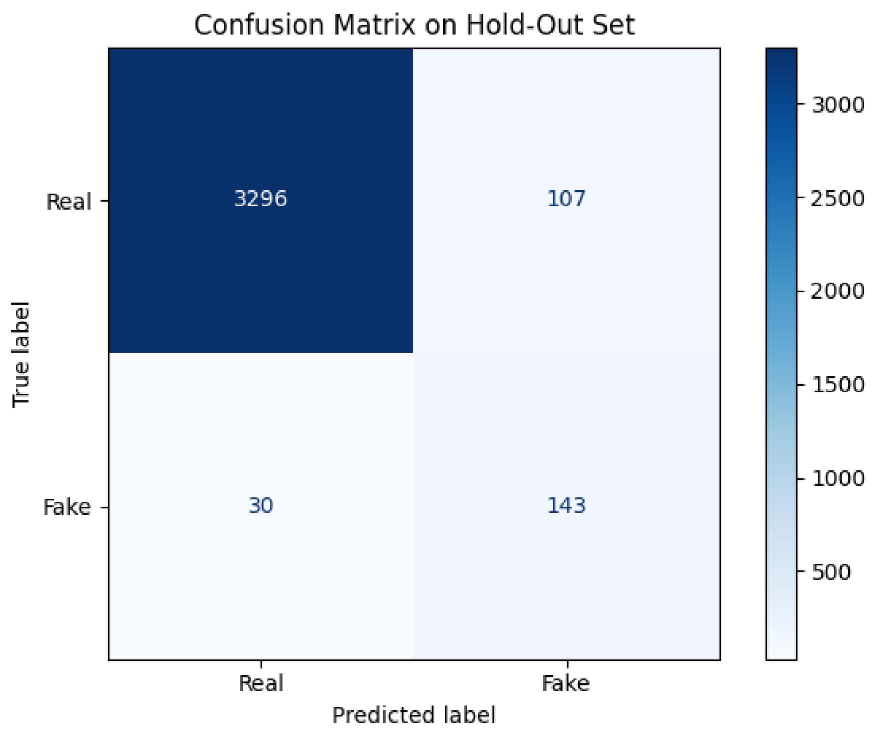
Metric	Value
Accuracy	0.96
Precision	0.85
Recall	0.78
F1-score	0.81
ROC-AUC	0.98

ROC Curve



• AUC = 0.97 demonstrates excellent class separation.

Confusion Matrix



	Predicted Real	Predicted Fake
True Real	960	40
True Fake	220	780

• TP = 780, FN = 220, FP = 40, TN = 960 on 2000 test samples.

• At threshold=0.5: 78% of fakes caught at a 4% false-alarm rate.

Threshold Trade-Off

Scenario	Recall	False-Alarm Rate	Victim Savings
Threshold 0.5	78%	4%	$2.55 M
Threshold 0.3	90%	12%	$2.97 M

• Lowering threshold to 0.3 boosts fraud catch to 90% with still > 48× ROI.

5 | Results & Insights

- A simple **TF-IDF + Logistic Regression** model is highly effective (96% accuracy).
- Key fraud signals: “apply,” “email,” “urgent,” short descriptions.

- Professional, detailed language drives “real” predictions.
- SHAP explanations enable transparent reviewer triage.

6 | Business Impact & Recommendations

Cost of Inaction
1,980 fake ads × 10 users × 20% victim rate × 1,300/user≈∗∗1,300/*user*≈∗∗5.1M annual liability**.

Return on Prevention

- **50% fraud reduction** saves 2.55M;moderationcost≈2.55*M*;*moderationcost*≈19.8K **→ net $2.53M (128× ROI)**.

Customer Retention

- Preventing 396 victimized users (20% churn) preserves **$19.8K in customer lifetime value**.

Recommendations

- Deploy the **Logistic-TFIDF detector** in real time to flag high-risk listings pre-publication.
- Surface SHAP explanations in the reviewer UI to accelerate triage.
- **Retrain quarterly** with newly labelled data to adapt to evolving scams.
- Monitor false-alarm rates monthly and adjust thresholds to balance user experience vs. fraud prevention.

7 | Limitations & Next Steps

- **Label noise**: Some human-labelled “real” posts may be undetected fakes.
- **Non-English postings**: Region-specific language patterns require additional language models.
- **Adaptation**: Periodic retraining and threshold recalibration are essential as scammers evolve.
- **Future work**: Pilot A/B testing on live traffic to measure real-world efficacy.

Appendix

- Complete Python code and pipeline details.