**Detecting Fake Job Postings Using NLP and Machine Learning**

**Team Members:**

Pranitha Anumolu (ZP32772)

Swetha Manchukonda (TA08134)

**Professor:**

Zeynep Kacar

**Research Question:**

**“Can natural language processing (NLP) combined with machine learning effectively detect fake job postings to enhance online job market security?”**  
The goal is to assess whether automated models can reduce fraud on freelancing platforms like Freelancer and Upwork by identifying suspicious job listings before they mislead users.

**Key Findings**

* Fake job posts tend to use vague or appealing language such as “easy”, “quick”, or “money”, while real jobs include skill-based terms like “developer”, “project”, or “design”.
* Repetitive titles (e.g., “Portuguese Recording Project” over 600 times) were strong indicators of fake or spammy behavior.
* Budget data played a critical role: fake postings often had missing or unusually low budgets.
* Among the models tested, **Random Forest** and **Logistic Regression** performed best, with Random Forest outperforming others due to its strength in handling nonlinear patterns in text data.
* The final system was deployed using a lightweight web application that accepts a job title and description and predicts whether the post is real or fake in real time.

**Significance of the Project:**Increasing fake job advertisements on freelance marketplaces lead to economic and psychological distress for job applicants, undermining platform trustworthiness.  
The solution is scalable, data-driven, and based solely on text and budget features, meaning no user profile data is required, which enables it to be quickly deployed. The method improves trust and security within the gig economy, enabling platforms to better protect users.

**Summary of Methodologies:**

 **Data Collection & Preprocessing**:

* Web-scraped 15,000 job postings (10,000 real, 5,000 fake) from Freelancer and Upwork.
* Cleaned text data, handled missing budgets, and applied TF-IDF vectorization to convert descriptions into numerical features.

 **Exploratory Data Analysis (EDA)**:

* Uncovered key language patterns and title repetitions.
* Visualized term frequencies and budget trends across real vs. fake jobs.

 **Modeling**:

* Four machine learning models were applied: **Logistic Regression**, **Random Forest**, **Naive Bayes**, and **XGBoost**.
* Models were trained on TF-IDF features; performance evaluated using accuracy and qualitative analysis.

 **Implementation**:

* Developed a working demo using **Flask/Streamlit**, capable of predicting job authenticity from user input.
* Designed to be lightweight and easily integrable into existing job platforms or browser extensions.

**Summary of methodologies:**

This project demonstrates that simple textual and budget-based features are sufficient to detect fake job postings with high accuracy. The models, particularly Random Forest, successfully identify patterns indicative of fraudulent content. This system, if integrated into freelance platforms, could significantly reduce the number of fake listings, increasing platform trust and user safety.

**Importance of Research question:**

**“Can natural language processing (NLP) combined with machine learning effectively detect fake job postings to enhance online job market security?”**  
The research also examines the impact of implementing such detection systems on job seekers' trust, application behaviors, and the overall integrity of freelancing platforms. As fraudulent job posts become more prevalent, especially on websites like Freelancer and Upwork, this question is vital for improving platform safety and user experience.

**Background and Context:**

Fake job postings on freelance platforms pose a serious threat by misleading job seekers, wasting their time, and sometimes leading to financial or emotional harm. This undermines trust in online job markets and harms the credibility of the platforms themselves.

Previous studies in spam detection and deceptive content identification have shown that machine learning and natural language processing techniques can be powerful tools for pattern recognition in text data. By analyzing linguistic patterns and structural traits of job descriptions, this project builds on these principles to target a real-world application—fake job detection.

The dataset used was web-scraped from freelancing platforms, comprising 15,000 job posts (10,000 real, 5,000 fake). The work emphasizes lightweight, scalable solutions based on textual content and budget fields—excluding user profile data—to allow for quick integration into existing systems.

**Objectives:**

* To identify distinguishing features in fake vs. real job postings using natural language and budget analysis.
* To develop and evaluate multiple machine learning models (including Logistic Regression, Random Forest, Naive Bayes, and XGBoost) for classifying job authenticity.
* To implement a real-time, user-friendly demo application that flags fake job posts based solely on the input job title and description.
* To offer a scalable solution that freelance platforms can use to enhance security, reduce fraud, and improve user trust.

**Data Overview:**

**Dataset Description**

The dataset used in this project was web-scraped from leading freelancing platforms such as Freelancer and Upwork. It includes 15,000 job postings, of which 10,000 are labeled as real and 5,000 as fake. The dataset is structured with five key features:

* **Keyword** – Common phrases extracted from job listings
* **Title** – The title of the job post
* **Description** – A detailed explanation of the job
* **Budget** – The offered payment for the job
* **Job Type** – Classification indicating whether a job is real or fake

This dataset forms the basis for identifying linguistic and structural patterns distinguishing fake job posts from genuine ones.

**2. Data Preparation**

Data preprocessing was critical to ensuring the quality of inputs fed into machine learning models. The steps involved were:

* **Text Cleaning**: Removed unnecessary characters, stop words, punctuation, and special symbols from the descriptions and titles.
* **TF-IDF Vectorization**: Applied Term Frequency-Inverse Document Frequency (TF-IDF) to convert cleaned text into numerical vectors, making the content suitable for model input.
* **Budget Handling**: Missing or zero-value budgets were flagged, as they were suspected indicators of fraudulent posts.
* **Data Splitting**: The dataset was divided into training (70%) and testing (30%) subsets to evaluate model performance effectively.

This preparation enabled the identification of significant textual signals and budget anomalies linked to fake postings.

**3. Challenges with Data**

Several challenges arose during data handling:

* **Missing Values**: Many fake job posts had either missing or non-numeric budget fields. These were either imputed with placeholders or used as negative signals during model training.
* **Textual Noise**: Spammy posts often reused boilerplate text, making it difficult to distinguish uniqueness. TF-IDF helped mitigate this by down-weighting common words.
* **Imbalanced Classes**: Although the dataset had a 2:1 ratio of real to fake jobs, this imbalance posed a challenge. To address this, class weighting and evaluation metrics (e.g., F1-score) were used to prevent bias toward the majority class.
* **Platform Variability**: Since the data was scraped from multiple sources, formatting inconsistencies had to be resolved during preprocessing.

These issues were systematically addressed to ensure the dataset was reliable and representative for modeling purposes.

**Methodology**

**1. Exploratory Data Analysis (EDA)**

Exploratory Data Analysis was a crucial early step in understanding the structure and behavior of the job postings in the dataset. Through various visualizations and frequency-based analysis, several insightful patterns emerged:

**Word Patterns in Descriptions**: Fake job postings frequently used vague, overly positive, or clickbait terms such as “easy”, “quick”, and “money”. In contrast, real job listings focused more on concrete skill-based language, featuring words like “developer”, “project”, and “design”.

Repetitive Titles: One of the most striking discoveries was the repetition of job titles. For instance, the title “Portuguese Recording Project” appeared over 600 times, suggesting automated or spam-like post generation. This was used as a strong feature to flag potential fake listings.

* **Budget Disparities**: Many fake job posts had either missing, vague, or unusually low budgets. Real listings tended to have clearly defined budget ranges, contributing another differentiating signal.
* **Class Imbalance Recognition**: The dataset maintained a 2:1 ratio of real to fake jobs (10,000 vs. 5,000), which mirrors real-world distribution. Acknowledging this imbalance early helped in designing appropriate evaluation strategies.

These EDA insights directly informed the feature engineering and model selection processes, particularly the use of TF-IDF for keyword strength and the importance placed on budget as a predictive attribute.

**2. Modeling / Analysis Techniques**

Four machine learning algorithms were applied and evaluated for their effectiveness in classifying job postings:

* **Logistic Regression**  
  A baseline linear classifier used for its interpretability and simplicity. TF-IDF vectorized text was passed into the model, which then calculated the probability of a post being real or fake. Despite its simplicity, it performed reasonably well and was retained as a reference model.
* **Random Forest**  
  An ensemble method using multiple decision trees, Random Forest excelled in capturing nonlinear patterns and word interactions that linear models missed. It showed superior performance in classifying complex text data and handled feature importance analysis well.
* **Naive Bayes**  
  A probabilistic classifier that works well with high-dimensional text data. It assumed word independence but provided rapid classification, making it a viable lightweight model for initial screening purposes.
* **XGBoost**  
  A powerful gradient-boosting framework used to push performance boundaries further. It iteratively optimized weak learners, combining them into a strong classifier. XGBoost performed well in terms of accuracy but had higher training complexity compared to Random Forest.

**Text Feature Engineering**

* **TF-IDF Vectorization**: Term Frequency-Inverse Document Frequency was used to convert job descriptions and titles into numerical vectors, giving more weight to rare but potentially meaningful terms.
* **Budget Normalization**: Budget values were cleaned and standardized for consistency, and missing values were treated as separate informative flags.

**Model Evaluation Metrics**:  
All models were evaluated using:

* **Accuracy**
* **Precision**
* **Recall**
* **F1-Score**

**3. Assumptions and Limitations**

**Assumptions:**

* **Textual Clues Are Sufficient**: The project assumed that job descriptions and budgets alone contain enough information to distinguish between real and fake listings.
* **Budget Clarity Reflects Authenticity**: It was assumed that fake jobs are more likely to leave budget fields empty or vague, while genuine postings are financially specific.
* **Real Jobs Are More Prevalent**: The 2:1 ratio of real to fake posts was considered reflective of actual platform dynamics and used as the class distribution for model training.
* **Word Patterns Vary Significantly**: The premise that real and fake posts use significantly different language underpinned the use of TF-IDF and frequency analysis in model design.

**Limitations:**

* **Limited Feature Scope**: Only text and budget data were used. Features like user history, post timing, or employer ratings were not included, which could further enhance detection accuracy.
* **Data Source Generalizability**: The data came only from Freelancer and Upwork. The models may not generalize well to other platforms with different formatting, audience, or content styles.
* **Static Snapshot**: Since the data was web-scraped at a single point in time, temporal patterns or evolving spam tactics were not considered.
* **False Positives Risk**: There’s a risk that creative or unusually phrased real job posts could be flagged as fake if they diverge too much from common patterns.

**Results:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

**Visualizations**

**A close-up of words

AI-generated content may be incorrect.**

This visualization highlights frequently occurring terms in fake job descriptions, such as "money," "click," or "investment." These keywords often point to scam-related language, supporting our hypothesis that linguistic patterns can indicate job posting authenticity.

**A graph with red and blue squares

AI-generated content may be incorrect.**

This bar chart reveals that certain job titles (e.g., "Data Entry Clerk") are disproportionately represented in fake postings. This pattern supports our model’s ability to learn title-based signals for prediction.

**A close-up of a pie chart

AI-generated content may be incorrect.**

Fake job descriptions tend to be either excessively long or unusually short, lacking structured formatting. This reinforces our assumption that description length and structure are key discriminators in model training.

**Web Interface Implementation:**

To enhance user accessibility and demonstrate the real-world applicability of our fake job detection model, we developed an interactive web interface using Flask. This interface allows users to input job titles and descriptions, which are then processed by our trained machine learning model to predict whether the job posting is **real** or **fake**. The front end was built using HTML and CSS, styled with a responsive and visually appealing design featuring a background warning image to reflect the nature of job scams. The results are presented with dynamic bar charts powered by Chart.js, which display the prediction outcome and provide a summary of the classification. This user-friendly tool aims to help individuals, especially job seekers, to easily evaluate the authenticity of job postings without requiring any technical background.

A hand holding a puzzle piece

AI-generated content may be incorrect.

**Interpretation of Results:**

The results of this project clearly show that machine learning model especially **Random Forest** can effectively detect fake job postings by analyzing just the text and budget information. This is significant because it means platforms like Freelancer and Upwork can use relatively simple, fast models to automatically screen listings and protect users from scams.

One of the most important findings was that text alone tells us a lot about whether a job post is real or fake. Words like “easy,” “quick,” and “money” showed up frequently in fake posts, while genuine listings used more job-specific terms like “developer,” “design,” or “project.” Also, budget values played a critical role fake job posts often had missing or extremely low budgets, which helped the models flag them.

By building a system that can predict job authenticity in real time through a lightweight web interface, this project has made an important step toward improving online job market security and restoring user trust.

**2. Comparison with Existing Literature**

Previous research on spam and scam detection often relied on deep user profiling or behavioral tracking. In contrast, this project proves that you don't need detailed user data—just a job’s title, description, and budget can be enough.

This aligns with findings in earlier natural language processing studies, which show that language patterns alone can be powerful indicators of intent or authenticity. However, while other works have often focused on emails, product reviews, or social media posts, our application to freelancing job listings is relatively novel.

Moreover, the success of models like Random Forest and Logistic Regression here mirrors their performance in past text classification tasks, reinforcing their reputation as strong choices for NLP problems involving structured text data.

**3. Unexpected Findings**

A few results surprised us during the analysis:

Title repetition was more telling than expected. The fact that a job title like “Portuguese Recording Project”appeared over 600 times strongly indicated automated or spammy posting behavior. We hadn’t initially planned to weigh title repetition so heavily, but it became one of the most useful features.

Basic models performed remarkably well. We expected complex models like XGBoost to clearly outperform simpler ones, but Random Forest—a relatively straightforward ensemble method—was both faster and more accurate in practice. This shows that high performance doesn’t always require high model complexity, especially when the features are well-selected.

Real posts can sometimes look fake. A few legitimate jobs had vague descriptions or missing budgets, which caused them to be flagged incorrectly. This highlights a trade-off between automation and false positives and reminds us that no system is perfect without some human review.

**Conclusion and Recommendations**

**1. Summary of Key Findings**

This project successfully demonstrated that machine learning models can detect fake job postings with high accuracy using just the job title, description, and budget information. The following are the major takeaways:

* Text-based patterns are highly predictive: Fake job listings often used vague, enticing language like "easy," "money," or "quick," while real job listings included skill-oriented terms such as "developer," "design," and "project."
* Repetitive job titles are strong indicators of fraud: Posts like "Portuguese Recording Project" appeared more than 600 times, suggesting mass-produced or bot-generated spam.
* Budget fields provide crucial clues: Fake listings often had missing, unrealistic, or unclear budgets.
* Random Forest outperformed other models: Among all tested algorithms—Logistic Regression, Naive Bayes, XGBoost, and Random Forest—the Random Forest model delivered the best performance in terms of detecting nuanced patterns and handling nonlinear data.
* A lightweight, web-based demo system was developed, allowing users to input a job title and description and instantly get a prediction on whether the job is real or fake.

These findings support the idea that natural language processing (NLP) and machine learning can offer a practical, scalable solution for increasing security and trust in online freelancing platforms.

**2. Recommendations**

Based on the findings, the following recommendations are proposed for real-world implementation:

* Integrate detection tools directly into freelance platforms to pre-screen job postings in real time. This can help reduce scam exposure for job seekers and boost overall platform credibility.
* Use budget consistency and text pattern analysis as red flags during job post submission. Automatic feedback or warnings can prompt suspicious employers to revise unclear or vague content.
* Combine this model with user feedback systems, such as job seeker reports or platform moderation tools, for continuous model refinement and training.
* Deploy browser plugins or extensions to help freelancers independently verify the legitimacy of a job post before applying.

**3. Future Work**

Although the project has laid a strong foundation, there are several areas for improvement and further exploration:

* Expand the feature set: Incorporate additional data such as user profiles, employer ratings, job post history, or geographic location to improve accuracy and reduce false positives.
* Adapt to evolving scam tactics: Periodically retrain the model on new data, especially as scammers update language patterns or techniques.
* Broaden data sources: Gather job data from more freelancing platforms (e.g., Fiverr, Guru, PeoplePerHour) to improve model generalizability.
* Implement live platform testing: Deploy the model in a real-world environment to observe how it performs on active data streams and user interactions.
* Develop multilingual support: Many platforms host posts in various languages. Training the model to handle different languages would increase its global applicability.

**References:**

T. A. Yaphet, M. A. W. Putra, V. C. Avianny, I. S. Edbert and D. Suhartono, "Fake Job Vacancy Detection Using Ensemble Voting Classifier," *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, Medan, Indonesia, 2024, pp. 1-7, doi: 10.1109/ICTIIA61827.2024.10761155.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10761155&isnumber=10761061>

T. Bhatia and J. Meena, "Detection of Fake Online Recruitment Using Machine Learning Techniques," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 300-304, doi: 10.1109/ICAC3N56670.2022.10074276.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10074276&isnumber=10073974

Freelancer.com. (n.d.). *Freelance jobs and projects*. <https://www.freelancer.com/?msclkid=1c0b1ed4545812c6bb7476f94a51d580&ft_prog=BNH&ft_prog_id=75179245799712>

Microsoft. (n.d.). *Visual Studio Code*. Retrieved March 10, 2025, from <https://code.visualstudio.com/>

**GitHub Repository:**

<https://github.com/pranianumolu/606project>