### **Building a Predictive Model for Fraudulent Job Listings: Lessons from Data Analysis and Machine Learning**

Fraudulent job postings are a persistent thorn in the side of today’s digital hiring landscape. These fake listings prey on job seekers, wasting time, eroding trust, and sometimes even leading to financial scams. Our team—Reid Dial, Jose Colchado, and Milo Dufresne-MacDonald—set out to tackle this problem head-on by building a machine learning model to detect fraudulent job listings. What started as a straightforward predictive task evolved into a journey of refining assumptions, overcoming obstacles, and rethinking our approach to achieve meaningful results.

Here’s how we went from raw data to a working model, the challenges we faced, and where we’re headed next. Whether you’re in fraud detection, machine learning, or hiring, we’d love your insights—let’s spark a conversation!

Keeping that in mind, we worked on this problem using the Fake/Real Job Posting Dataset from [Kaggle](https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction). The goal of this project is to gain insights and understand the characteristics of real and fake job postings by performing data analysis and data preparation to build an effective machine learning model.

#### **The Starting Point: A Deep Dive into Job Listings**

We began with a dataset of 17,880 job listings, where 4.84% (roughly 862 listings) were flagged as fraudulent. The dataset contained 18 columns, covering everything from job titles and descriptions to required education, experience, and listing features like salary ranges and company logos.

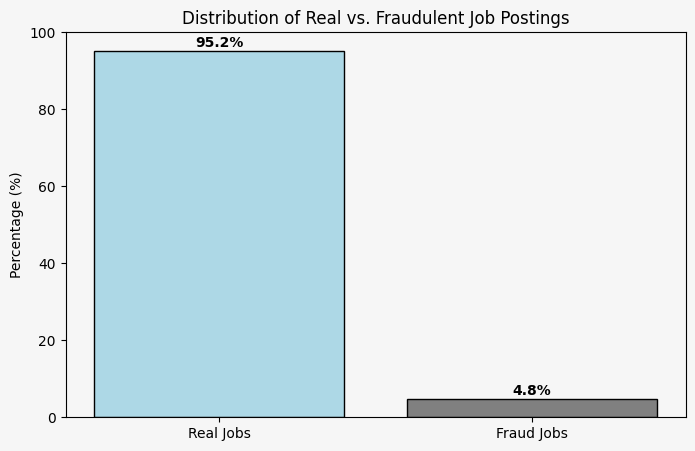
Our goal? To build a model that can distinguish real job postings from fraudulent ones.

### **Dataset Overview**

Dataset Shape: (17,880 rows, 18 columns) – meaning it contains 17,880 job postings with 18 features.

Class Distribution: The dataset is highly imbalanced –

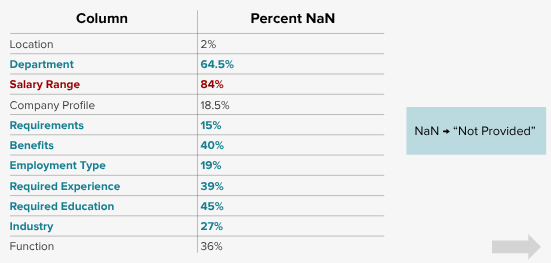
* 95.15% of postings are real jobs (Class 0)
* 4.84% are fraudulent jobs (Class 1)



But the data wasn’t perfect... missing values were a major hurdle:

A **majority of our columns** contained missing values, with some having as much as **64% to 84%** missing data. **Dropping all NaN values** would have left us with a nearly **useless dataset**, so we took a different approach.

Instead of removing missing data, we focused on **seven key columns** where missing information could be a red flag for fraudulent job postings. For these columns, **NaN meant that missing information was replaced with "Not Provided"**.



### **A Closer Look at Salary Ranges: A Different Approach**

While we handled missing values in most columns by replacing them with **"Not Provided"**, the **Salary Range** column required a different strategy. Salary transparency is a key indicator of **job legitimacy**, and fake job postings often manipulate or leave this field blank to lure applicants.

Here’s how we tackled it:

**Step 1: Identifying Fake Entry Points** – We analyzed salary ranges to spot inconsistencies and unrealistic figures that could indicate fraud.

**Step 2: Creating Max-Min Variables** – To filter out fake postings, we introduced **maximum and minimum salary variables**, ensuring unrealistic or missing values didn’t distort our model.

**Step 3: Recalibrating Job Postings** – Based on industry standards, we adjusted salary data to better reflect realistic job offers.

**Step 4: Aligning Salaries with Job Types** – We structured salary ranges to match typical pay for different job categories, making the dataset more reliable for prediction.

**Data highlights**

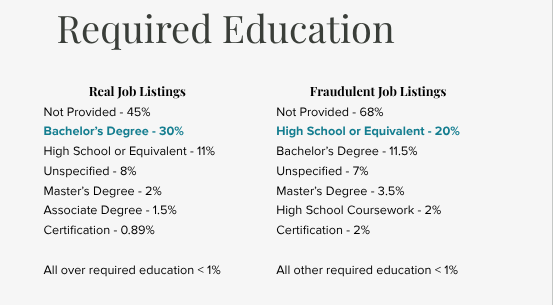
Top Industries:

****

**Key Observations:**

* "Not Provided" dominates both categories (53% for real jobs, 68% for fraudulent jobs), meaning many postings lack industry details, which can be a red flag.
* Finance and IT Services are found in both real and fraudulent job postings because they are two of the most dominant industries in the dataset making up over 16% of our dataset

Required Education:

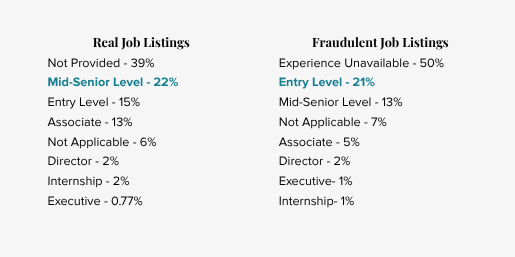


**Key Findings:**

* Missing Education Requirements:
  + Fake job postings are more likely to lack education requirements, with 68% unspecified compared to 45% in real listings.
* **Bachelor’s Degree vs. High School Requirement:**
  + **30% of real job postings** require a **Bachelor’s Degree**, making it the most common education requirement..
  + **20% of fraudulent job postings** list **"High School or Equivalent"**

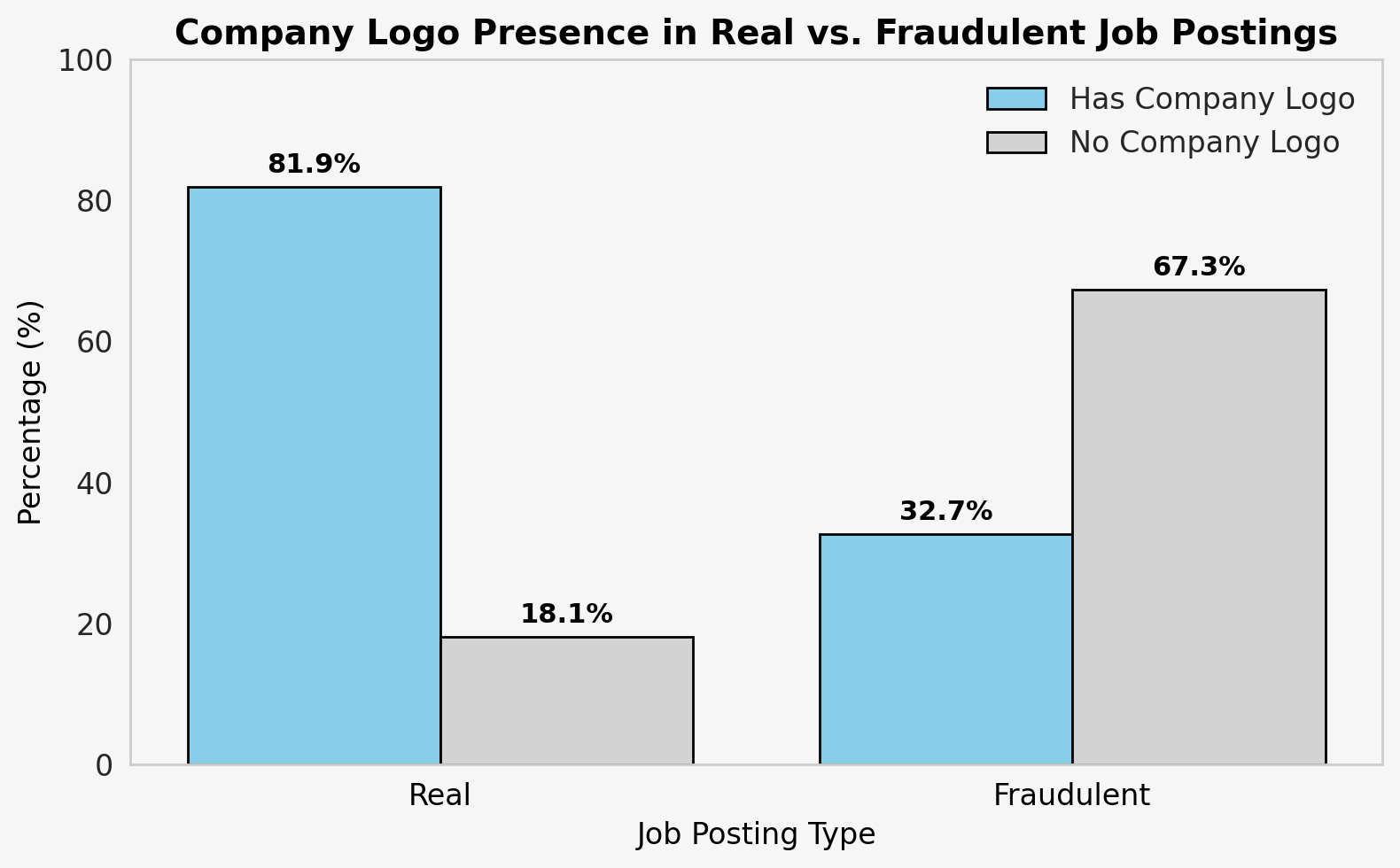
Our main emphasis on this graph is that real jobs emphasize going to college. That is why you should listen to your parents when they tell you to go to college because the numbers speak for themselves.

Required Experience:

****

**Key Findings:**

* One key takeaway? Fraudulent jobs are way more likely to target entry-level applicants—and it’s easy to see why.
* Think about it: Entry-level jobs are easier to apply for, often requiring just a resume upload or even a single click on LinkedIn. Scammers love this because the less effort required, the more people apply—giving them a bigger pool of potential victims.
* And the numbers back it up:  
  22% of real job postings require mid-senior level experience while 21% of fraudulent job postings target entry-level applicants.

****

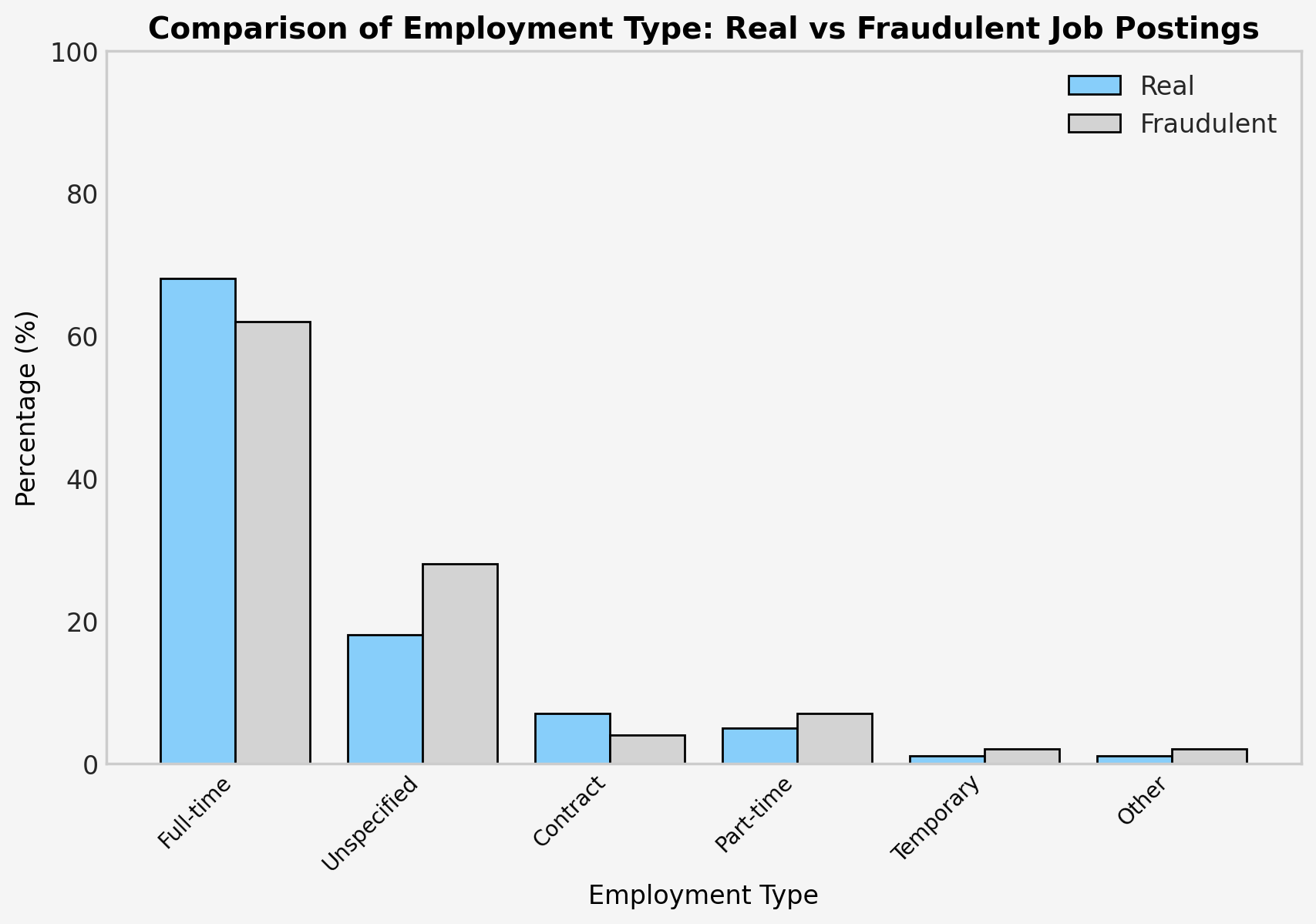
### **The Importance of Company Logos in Detecting Fraudulent Jobs**

This graph highlights a **key factor** in identifying whether a job posting is real or fraudulent: **the presence of a company logo**.

**Real Job Postings:** Over **81.9%** of legitimate job listings include a company logo, while only **18.1%** do not.

**Fraudulent Job Postings:** Only **32.7%** of fake job listings have a company logo, whereas a staggering **67.3%** are missing one.

This **significant difference** shows that **company logos are a strong indicator of job legitimacy**. Since fraudulent job postings tend to **lack a company logo**, this feature will be an **important variable** in our machine learning model for fraud detection.



This graph reveals an interesting trend—whether a job posting is **real or fraudulent**, the **majority of job listings are for full-time positions**.

**Full-Time Jobs:** The most common employment type in both **legitimate and fraudulent** postings, showing that scammers often try to mimic real job trends to appear credible.

This insight reinforces the **importance of looking beyond just employment type** when identifying fake job listings—other features like **company logos, salary transparency, and job descriptions** play a crucial role in fraud detection. Now let's take a deep dive into our model!

#### **Model Evolution: Iterating Toward Accuracy**

Our journey unfolded in four key versions, each teaching us something new about our data and approach.

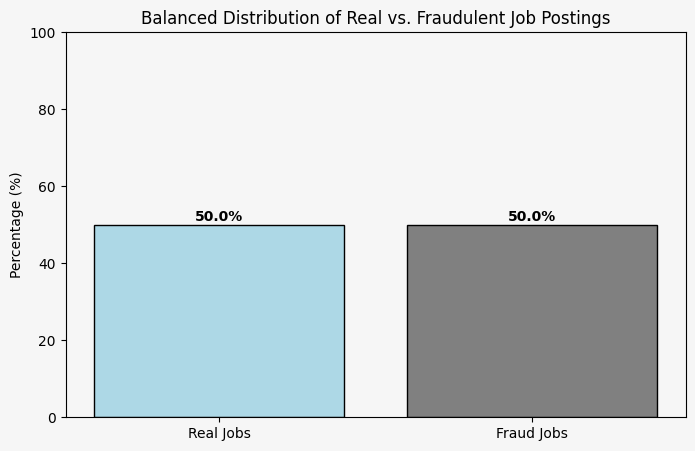
##### **Version 1: The Overly Optimistic Start**

We kicked off with a Logistic Regression model, hitting an impressive 95% accuracy. Cue the celebration—until we realized this was a red flag. With only 4.84% of listings being fraudulent, our model was likely just predicting “real” for everything, exploiting the imbalance. Time for a reset.

##### **Version 2: Balancing the Scales**

To address this, we balanced the dataset, ensuring an equal split of real and fraudulent listings. Retraining our Logistic Regression model dropped accuracy to a more realistic 76.5%. Feature analysis revealed key indicators:

* **Fraudulent Signals**: Real Estate Industry (1.66), Engineering Department (1.42), Finance Industry (1.09)
* **Legitimate Signals**: Education Industry (-1.49), Electronics Industry (-1.85), Company Logo (-2.26)



But we noticed something odd: many features had suspiciously similar accuracy scores, hinting at multicollinearity—where variables are too closely related, muddying the model’s weights.

##### **Version 3: Tackling Multicollinearity**

Digging deeper, we calculated Variance Inflation Factors (VIF) and found perfect collinearity, especially in our one-hot encoded Industry and Department columns. The culprit? A rookie mistake: we hadn’t dropped a reference column during encoding, creating a dummy variable trap. After fixing this, accuracy plummeted to 50%—a hit we expected, as we lost some predictive power. But by version’s end, it stabilized at 78%, with updated top features:

* **Fraudulent Signals**: Engineering Department (1.06), Administrative Department (0.91)
* **Legitimate Signals**: Electronics Industry (-0.81), Company Logo (-2.15)

##### **Version 4: Pruning the Feature Tree**

Even after fixing encoding, collinearity lingered. So, we experimented with dropping highly correlated columns one by one:

* **Dropping Department**: Accuracy fell to 49%, spotlighting Real Estate (1.31) and Finance (1.05) as fraud signals, with Company Logo (-2.24) and Education (-1.47) as legitimate markers.
* **Dropping Industry**: Accuracy dipped to 46%, leaving Engineering Department (1.44) and Company Logo (-2.12) as key predictors.

These drops told us those columns held valuable info—we just needed a smarter way to use them.

#### **Key Insights: What Makes a Listing Suspicious?**

Across iterations, patterns emerged. Listings missing department or industry details, lacking a company logo, or tied to Real Estate and Finance were more likely fraudulent. Conversely, Electronics and Education industries, plus a visible logo, leaned legitimate. Job description length also mattered—shorter blurbs often signaled fakes.

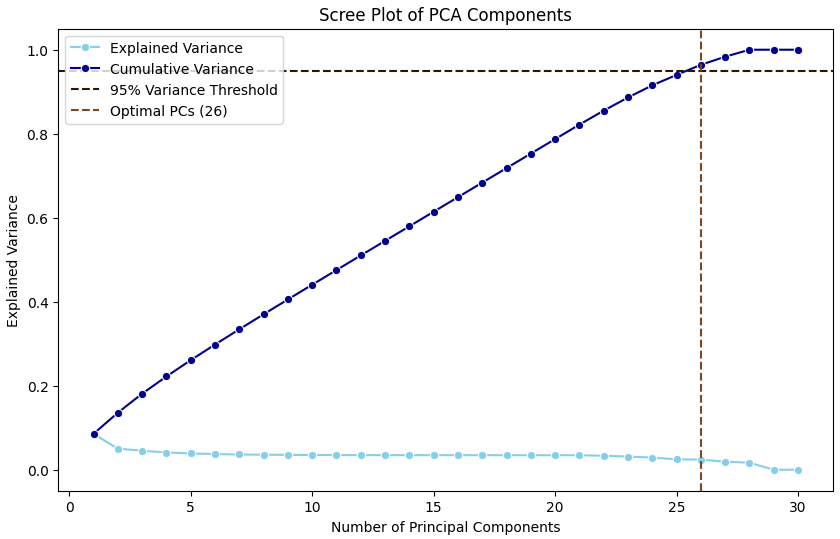
**Version 5: Utilizing PCA**

**What is PCA?**

**Principal Component Analysis (PCA)** is a technique used to **simplify large datasets** by reducing the number of variables while keeping as much important information as possible. It does this by transforming the original features into a smaller set of new variables called **principal components**, which capture the most significant patterns in the data.

In this case, **PCA has identified 26 principal components** as the optimal number to retain about **95% of the important information** from the original dataset. This means instead of dealing with a large number of individual variables, we can focus on just **26 key components**, making the data easier to analyze and interpret.

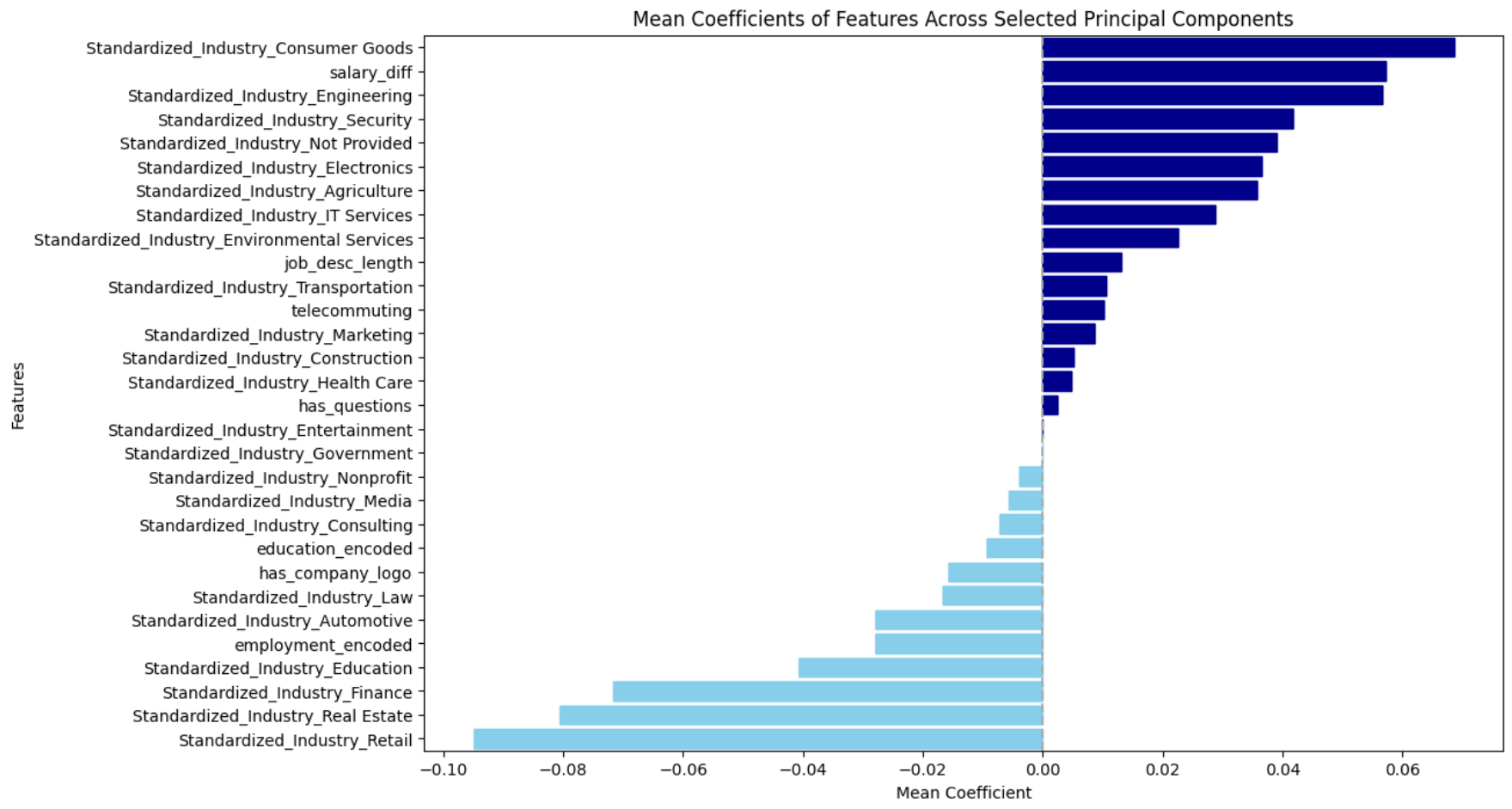
The graph below illustrates this process, showing how the number of components impacts the amount of information retained. By using PCA, we can work with cleaner, more manageable data without losing valuable insights!



After applying **Principal Component Analysis (PCA)**, we can explore which features contribute the most to the newly created principal components. The bar chart below shows the **mean coefficients of various features**, helping us understand which variables had the greatest influence.

**Positive values** (dark blue) indicate features that had a **stronger positive impact** on the principal components.  
**Negative values** (light blue) represent features that contributed in the **opposite direction** but were still important.

From the chart, we can see that **industries like Consumer Goods, Engineering, and Security** played a significant role, while **industries such as Retail, Real Estate, and Finance** had the opposite effect.



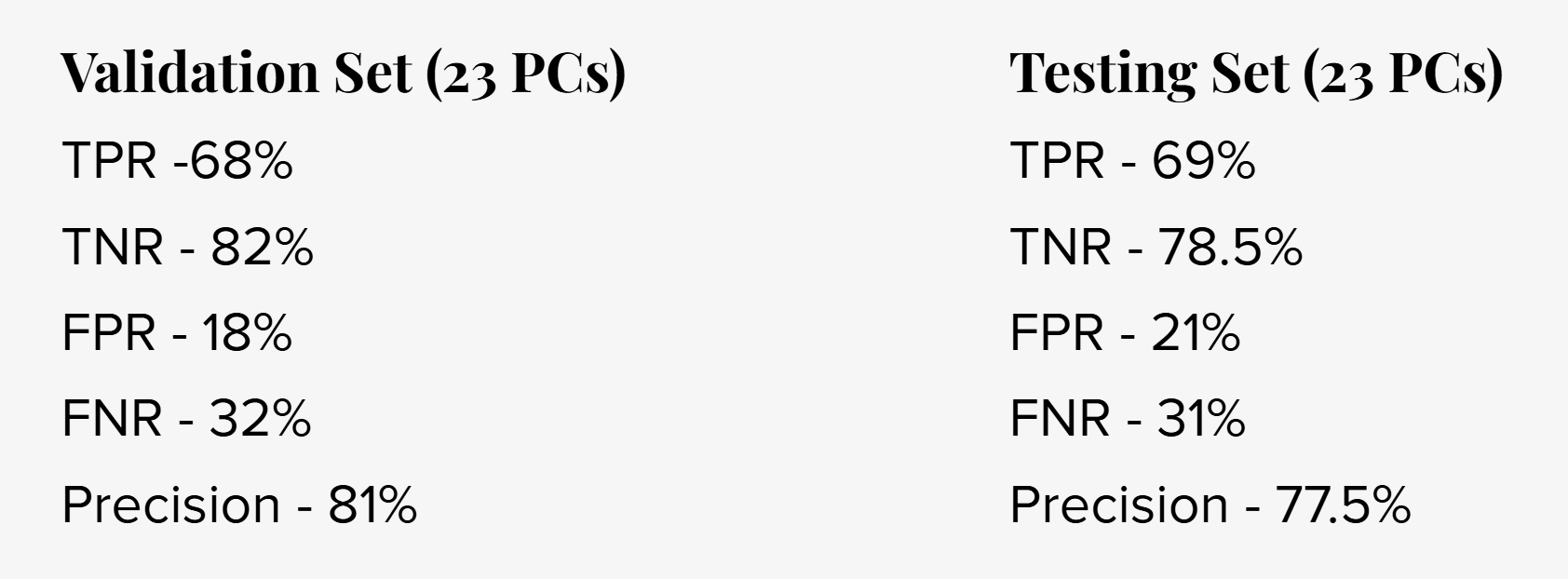
After applying **Principal Component Analysis (PCA)** with **23 principal components**, we evaluated the model on both the **Validation Set** and the **Testing Set** to see how well it generalizes.

**Key Performance Metrics**

* **True Positive Rate (TPR)** – How often the model correctly identifies positive cases.
* **True Negative Rate (TNR)** – How well it correctly classifies negative cases.
* **False Positive Rate (FPR)** – The percentage of negative cases mistakenly labeled as positive.
* **False Negative Rate (FNR)** – The proportion of actual positives the model fails to detect.
* **Precision** – The accuracy of positive predictions.

### **What This Means**

The model is **consistent across validation and testing**, meaning it generalizes well to new data.



#### **Complete Model: Versions 1-5**

Our project was a rollercoaster of lessons—machine learning isn’t just about crunching numbers or picking the flashiest algorithm. It’s about digging into the *why* behind the results, wrestling with messy data, and iterating until something clicks. We started with a shiny 95% accuracy that turned out to be fool’s gold, and after peeling back the layers, landed at a gritty, honest 76%. It’s not perfect, but it’s a solid step toward spotting job fraud in a way that actually matters.

#### **Final Thoughts: It’s About the Journey**

This project taught us that machine learning isn’t just about algorithms—it’s about questioning results, understanding data quirks, and iterating relentlessly. From a deceptive 95% accuracy to a hard-earned 77%, we’ve built a foundation to detect job fraud meaningfully.

To the LinkedIn community: How do you tackle fraud in hiring? What’s worked—or hasn’t—in your models or processes? Drop your thoughts below—let’s learn from each other!

To further explore our analysis and model creation steps, please see our [github repository](https://github.com/MiloDufMac/Fraudulent-Job-Postings)