HR Visionary

1. **Introduction**

In a competitive job market, HR teams and recruiters need data-driven insights to track industry trends, benchmark salaries, standardize job roles, and anticipate workforce demands. Our project tackles these challenges by integrating multi-source data-including LinkedIn, and web-scraped job listings from Indeed. Leveraging advanced machine learning and NLP techniques, such as Prophet for time-series forecasting and XGBoost for predictive modeling, we provide actionable insights that help organizations optimize recruitment strategies and make informed, evidence-based HR decisions.

2. **Data Collection and Integration**

2.1 Datasets Overview

Linkedin:

From LinkedIn **profiles**, we extracted detailed candidate information—including job titles, work history, and skill sets—to support our analysis of job seeker trends. From **company** data, we utilized key firm attributes such as industry classification, company size, and geographic location to provide context for the job postings. Kaggle-Sourced Data (124K records) - reinforces and expands the LinkedIn postings with similar structure and content.

Indeed:

Scraped data (5K records) - Job Listings were obtained using Selenium and BeautifulSoup with techniques such as rotating user agents, proxy management, time delays, and cookie handling. The data fields captured include job location, salary range, job type, post date, and full job descriptions from Indeed.

Scraping Methodology – The implementation uses Selenium and BeautifulSoup with rotating user agents, proxy handling, randomized delays, and cookie management to bypass anti-scraping measures. Error recovery strategies ensure resilience, retrying failed requests with backoff handling. Selenium dynamically loads JavaScript-rendered content before parsing it with BeautifulSoup, while job postings are opened in separate tabs to preserve pagination. The scraper automates dynamic URL generation across job titles, locations, and pages, storing results in real-time in a CSV file to prevent data loss. This ensures a scalable and adaptive approach to extracting structured job data.

2.2 Integration Process

Our data integration involved several steps. In terms of Deduplication & Schema Standardization, we unified disparate datasets by aligning column names and eliminating duplicate entries. For Salary Normalization, all salary figures were converted to USD using predetermined exchange rates, with annualized hourly wages (assuming a 40-hour week and 52 weeks per year) and monthly salaries (multiplied by 12) computed for consistent comparisons. Date Conversion was achieved by transforming UNIX timestamps to a uniform dd/MM/yyyy format to support accurate time-series analyses. The Text Cleaning & NLP Enrichment process involved removing stop and noise words (e.g., “certified,” “enthusiast”) and trimming lengthy job descriptions, as well as employing semantic mapping techniques (fuzzy matching, Word2Vec embeddings) to align job postings with the official O*NET-SOC categories. Each unique job posting (or user profile) is considered one record, and the final integrated dataset comprises tens of thousands of rows.

3. **Data Analysis**

Analysis techniques:

Exploratory Data Analysis (EDA) and Descriptive Statistics -

For Descriptive Insights, summary statistics (means, medians, standard deviations) were computed for key metrics such as salaries, views, and application counts, and histograms and boxplots were employed to visualize salary distributions and detect outliers (e.g., salaries exceeding $1M). In the Trend Analysis, window functions were utilized to calculate year-over-year growth and decline in job postings.

Enhanced EDA and NLP on Job Titles -

The process began with Schema and Sample Inspection in order to understand the data structure and content. Record and Quality Checks were then performed by computing the total record count and identifying duplicate and null entries in the updated_title column, ensuring that downstream analyses use clean data.

Frequency Analysis of job titles was conducted by grouping the data by updated title and ordering by count.

Data cleaning for NLP involved creating a new column, clean_title, by converting job titles to lowercase, removing punctuation (retaining only word characters and whitespace), this standardization minimizes noise and improves the reliability of subsequent text analyses.

Tokenization and StopWords Removal were executed using Spark ML's RegexTokenizer and StopWordsRemover, which split the cleaned titles into individual tokens and removed common stopwords (e.g., "the", "and").

Finally, by exploding the token list and grouping by individual tokens, word frequencies across all titles were computed, leaving the most informative terms that characterize the job roles.

Time-Series Forecasting with Prophet

Prophet is an open-source forecasting tool that fits models with trend, seasonality (yearly, weekly, daily), and is particularly effective at capturing non-linear trends, and abrupt change points.

Predictive Modeling with XGBoost

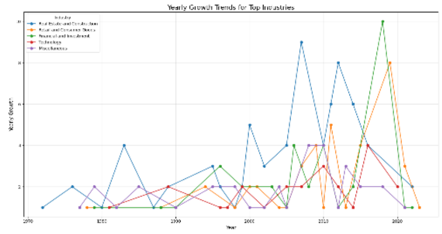
XGBoost is a high-performance gradient-boosted decision tree library that builds an ensemble of trees sequentially, where each new tree corrects the errors of the previous ones. This process captures complex feature interactions effectively while remaining scalable for large datasets. In our application, XGBoost was used to build both regression models—for example, to predict salaries—and classification models to forecast job posting success. We incorporated regularization techniques (L1 and L2) to prevent overfitting.

Feature Selection:

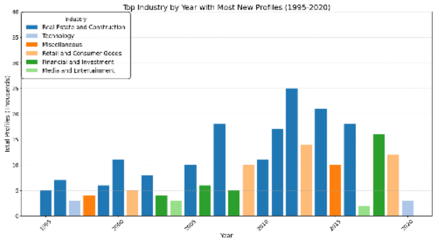
We selected key features, including a **cleaned job title** that was lowercased, stripped of punctuation, and tokenized, **normalized salary** as meantioned before, **views** - counts job post impressions and "popularity".

Visualizations:

The chart depicts yearly growth trends for top industries over a range of years, The lines show variations in growth for each industry across the years.



This bar chart shows the yearly distribution of the industry with the most new profiles created between 1995 and 2020, This visualization captures which industries led in creating new profiles year over year.



4. AI Methodologies

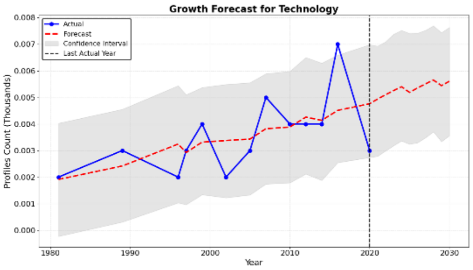
4.1 NLP and Title Standardization

In addition to the previously described techniques (fuzzy matching and Word2Vec embedding), the updated pipeline now includes Enhanced Text Preprocessing—the additional steps for cleaning and tokenizing job titles (as detailed in Section 3.2) ensure that minor variations (e.g., "**S**enior Engineer" vs. "**s**enior engineer") are minimized before matching. Token Frequency Analysis is also performed, examining the frequency of individual words in job titles to inform the design of stopword lists and support further refinement of semantic clustering approaches (including Generative AI clustering).

4.2 Time-Series Forecasting (Prophet)

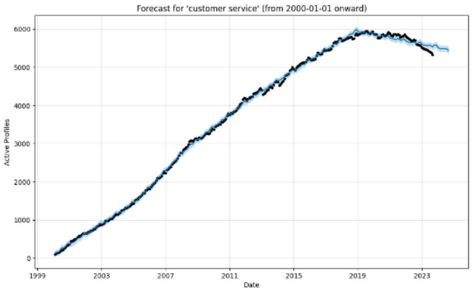
Case 1: Forecasting Emerging Industries

The code aggregates historical active profiles for top growing industries, converts these counts into a time-series format, and applies Prophet to forecast future industry growth over an extended period (e.g., 10 years). The resulting forecast and component plots provide insights into trends and seasonality, helping to identify which industries are likely to experience continued growth.



Case 2: Forecasting for a Chosen Job Title

The code filters data for a specific job title (e.g., "customer service"), computes active profiles over time from joined and left counts, and uses Prophet to predict future trends over a shorter period (e.g., 12 months). This targeted forecast aids in understanding the future trajectory of that particular job role. We allow the user to select the number of top industries to observe, based on those with the highest number of records, and to select the prediction period ahead. The second option enables the user to view the prediction for a specific job and specify the number of months to forecast.



4.3 Predictive Modeling (XGBoost)

The code uses XGBoost in two distinct pipelines on different datasets.

The first pipeline focuses on predicting job views from LinkedIn title data. The features used in this pipeline include the raw job title (updated_title) and the normalized salary, while the target variable is the number of job views. The pipeline consists of a text processing step that applies TF-IDF vectorization on updated_title, and a numeric pipeline that scales the normalized salary. These processed features are combined using a ColumnTransformer and fed into an XGBoost regressor. After training, with an optional hyperparameter tuning step, the model is used to predict future demand (views) for a new job title example, such as "validation engineers."

The second pipeline predicts salary from scraped job data. It uses a combined text field (job name, title, company, and description) along with optional categorical features like city or job type. The target is the parsed salary (salary_parsed). The pipeline applies TF-IDF vectorization to text data and OneHotEncoder to categorical features, merging them via a ColumnTransformer before feeding into an XGBoost regressor. After training and hyperparameter tuning with RandomizedSearchCV, the model predicts salaries for new job postings like "Data Scientist."
In summary, one XGBoost pipeline predicts job views using structured title and salary data, while the other forecasts salary based on a richer text description and categorical inputs.

5. Evaluation and Results

5.1 Key Findings

Industry Growth Trends indicate that the Tech sector remains dominant in both volume and growth rate, with Healthcare and Finance also showing strong expansion, and stable but slower growth in industries such as Manufacturing. Regarding Job Market Dynamics, the views-to-applies ratio reveals roles with high attention yet low conversion, thereby informing potential optimizations in job descriptions or compensation. In terms of Forecasting Accuracy, Prophet yielded interpretable forecasts with error margins (MAPE) typically in the 5-10% range, and the forecasts indicated predictable cyclical hiring spikes-such as a surge in Tech postings. Predictive Modeling Performance with XGBoost demonstrating high accuracy in classification tasks, such as predicting job-post success, while regression models for salary predictions consistently outperformed simpler baselines.

NLP-Based Title Standardization - Using fuzzy logic and Word2Vec, reduced inconsistent job titles to standardized references, and Generative AI clustering further grouped job roles into intuitive categories (e.g., “Management,” “Software Development”). Improved Job Title Standardization through enhanced cleaning and tokenization has led to more consistent job title representations, with frequency analysis revealing the most dominant tokens across the dataset, which in turn improves the accuracy of semantic matching and clustering routines. Additionally, the extended XGBoost pipeline, integrating both textual and numeric features, has demonstrated promising results in predicting job view counts, providing actionable estimates of job posting demand and offering HR professionals another layer of insight into market trends.

6. Limitations and Reflection

6.1 Data Quality and Variation

Missing Fields & Inconsistencies were observed, as varying data quality across sources (e.g., incomplete salary or description fields) required extensive cleaning and standardization, and Source Heterogeneity was evident due to differences in format between the original LinkedIn data, the Linkedin data from Kaggle and the scraped data from Indeed, necessitating robust schema mapping.

6.2 Scraping Challenges

Dynamic Content and Anti-Bot Measures occasionally disrupted the Selenium-based scraping pipeline due to changing website structures and captchas, although mitigated through random delays and proxy rotations, these factors still pose a risk of data loss.

6.3 Forecasting and Modeling Assumptions

Assumption Limitations exist in Prophet models assume historical stability in trends, sudden market shifts can reduce forecast accuracy, and engagement metrics (views, applies) may be platform-specific and not entirely comparable across sources.

7. Conclusions

Our project demonstrates a scalable HR analytics tool that integrates multi-platform data and leverages advanced ML and NLP techniques to support data-driven decision-making in HR. Key achievements include Unified Data Aggregation through the integration of LinkedIn, and scraped data from Indeed into a standardized schema using the O*NET-SOC 2019 taxonomy. Forecasting & Prediction via the effective use of Prophet for seasonal trend forecasting and XGBoost for predictive modeling, which enables accurate insights into job posting success and salary trends. Actionable Insights derived from detailed analyses of industry growth, job market dynamics, and standardized job roles that provide clear guidance for recruitment strategies and HR budgeting. Enhanced Data Insights Through NLP where the newly integrated EDA and NLP processing on job titles, coupled with an extended predictive pipeline using XGBoost, provide deeper insights into role popularity and demand dynamics. This advancement further empowers HR teams to refine their recruitment strategies with precision. By combining advanced analytics with scalable data integration, our solution empowers HR professionals to navigate complex labor markets and make strategic, evidence-based decisions.