Final Report

Evaluating Personal Job Market Prospects in 2024

Yibei Yu, Fuhan Zhang, Jonathan Leon

October 10, 2025

Table of contents

# Final Report

### Evaluating Personal Job Market Prospects in 2024

Yibei Yu, Fuhan Zhang, Jonathan Leon  
October 10, 2025

## Executive Summary

This report uses Lightcast job posting data to conduct a comprehensive analysis of job market trends through 2024, assessing the impact of skills, industries, and job types on employability and compensation. The team, comprised of Yibei Yu, Fuhan Zhang, and Jonathan Leon, applied a structured analytical process, including data cleaning, exploratory data analysis (EDA), clustering, and regression modeling.

The EDA shows that hiring activity is concentrated in the information technology, computer systems design, and consulting industries, demonstrating the impact of digital transformation. Salary analysis reveals that positions in finance, healthcare, and professional services command higher average salaries, while remote work opportunities still only account for about one-fifth of total job postings.

K-means clustering categorizes job postings into two main groups: positions requiring machine learning (ML) and data science (DS) skills, and those requiring non-technical skills. The machine learning/data science (ML/DS) cluster exhibits a clear upward trend in salary and exhibits higher skill complexity. Multiple regression models confirm that ML/DS requirements generate a measurable salary premium, particularly in the information and retail trade industries, although this effect is not uniform across all industries.

Overall, the analysis indicates that technical specialization, particularly in analytics and machine learning, continues to provide job seekers with a competitive advantage. However, adoption of these capabilities is uneven, and future workforce growth will depend on how non-technical sectors integrate data-driven decision-making. For professionals entering the market, developing and mastering quantitative, analytical, and digital transformation skills remains the clearest path to long-term career resilience and higher earning potential.

title: “Business Analytics, Data Science, and Machine Learning Trends” author: “Group 12: Yibei Yu, Fuhan Zhang, Jonathan Leon” date: “2025-09-12” format: html: toc: true bibliography: references.bib csl: csl/econometrica.csl

## Introduction

Our project focuses on identifying job market trends for business analytics, data science, and machine learning professionals in 2024. As the job landscape shifts due to AI and automation, understanding what skills, tools, and roles are most in demand is critical for effective career planning.

## Research Rationale

The rapid evolution of AI and data-driven decision-making has reshaped hiring expectations for analytics professionals. Employers now seek individuals who not only analyze data but also deploy predictive models, build dashboards, and communicate insights. Our team aims to understand these changing demands by analyzing job postings, industry trends, and academic insights.

title: “About Us” format: html

## Yibei Yu

I am a graduate student in Applied Data Analytics at Boston University. My academic and professional interests include financial analytics, machine learning, and real estate modeling. I am particularly interested in careers in data analytics and fixed income research, where I can apply quantitative skills to support investment decisions. My goal is to begin my career as a financial/data analyst in the Boston area and eventually transition into asset management with a focus on fixed income research.

## Jonathan Leon

I am a graduate student in Insurance Management at Boston University with a career focused entirely on the corporate property segment of the insurance industry. I’m pursuing this degree to broaden my skill set, particularly in analytics, and to develop the capabilities needed to contribute to the digital transformation initiatives shaping the future of the industry.

## Fuhan Zhang

## I am a graduate student in Applied Business Analytics at Boston University. My academic and professional interests include data analytics, risk management, insurance, and market analysis. With an undergraduate background in actuarial science, I aim to integrate analytical and quantitative skills into the insurance and data analytics industries. My goal is to secure an internship or entry-level position next semester that bridges these two fields, ultimately pursuing a career where data-driven insights enhance actuarial and risk management practices.

title: “Business Analytics, Data Science, and Machine Learning Trends” author: “Group 12” date: “2025-09-27” toc: true bibliography: references.bib csl: csl/econometrica.csl format: html: default

## Background

Business analytics, data science, and machine learning have become central to modern organizations, shaping hiring practices and skill requirements. Recent studies highlight the growing demand for professionals who can bridge technical and business expertise.

## Literature Review

Yuan (2023) emphasizes the increasing role of business analytics in data-driven decision-making (Yuan and Bao).  
Badhon (2024) highlights the need to enhance machine learning pipelines to improve scalability and efficiency (Badhon et al. (2024)).  
Timpone (2025) reviews the application of AI in employment and underscores the importance of continuous skill development (Timpone and Yang (2025)).  
Li (2024) discusses strategies for preparing students for data science careers, stressing both technical and soft skills (Li, Milonas, and Zhang (2024)).  
Modak (2024) provides a comprehensive review of machine learning methods in real-world industries (Modak et al. (2024)).

Together, these studies suggest that technical proficiency in programming, machine learning, and data management must be paired with problem-solving, communication, and leadership skills.

## Research Questions

* What are the most in-demand skills for data science, business analytics, and ML roles?
* Have job descriptions evolved in 2024 to require AI/ML expertise?
* What industries are hiring the most data scientists and why?
* What is the career outlook for business analytics professionals?

## Transition to Data

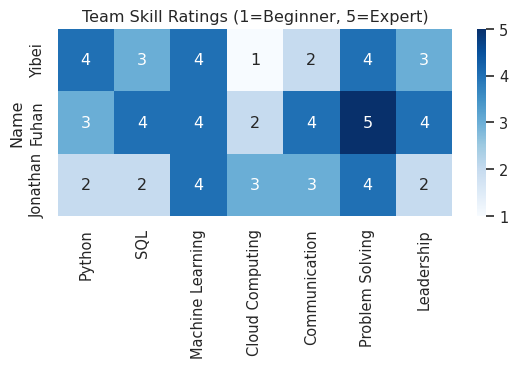
To address these questions, our team analyzes a real-world dataset of job postings (Lightcast, 2024). This dataset allows us to examine the skills most frequently requested by employers, identify industry-specific hiring patterns, and connect these findings to our team’s own skill gap analysis.

title: “Skill Gap Analysis” author: “Group 12” date: “2025-09-27” toc: true bibliography: references.bib csl: csl/econometrica.csl format: html: default

import pandas as pd  
import seaborn as sns  
sns.set\_theme(style="whitegrid")  
import matplotlib.pyplot as plt  
import os  
os.makedirs("figures", exist\_ok=True)

# Build Team Skill DataFrame

skills\_data = {  
 "Name": ["Yibei", "Fuhan", "Jonathan"],  
 "Python": [4, 3, 2],  
 "SQL": [3, 4, 2],  
 "Machine Learning": [4, 4, 4],  
 "Cloud Computing": [1, 2, 3],  
 "Communication": [2,4,3],  
 "Problem Solving": [4,5,4],  
 "Leadership": [3,4,2]  
}  
  
df\_skills = pd.DataFrame(skills\_data).set\_index("Name")  
df\_skills  
  
plt.figure(figsize=(6,4))  
sns.heatmap(df\_skills, annot=True, cmap="Blues", cbar=True)  
plt.title("Team Skill Ratings (1=Beginner, 5=Expert)")  
plt.tight\_layout()  
plt.savefig("figures/team\_skill\_heatmap.png", dpi=300, bbox\_inches="tight")  
plt.show()



# Extract Top Industry Skills

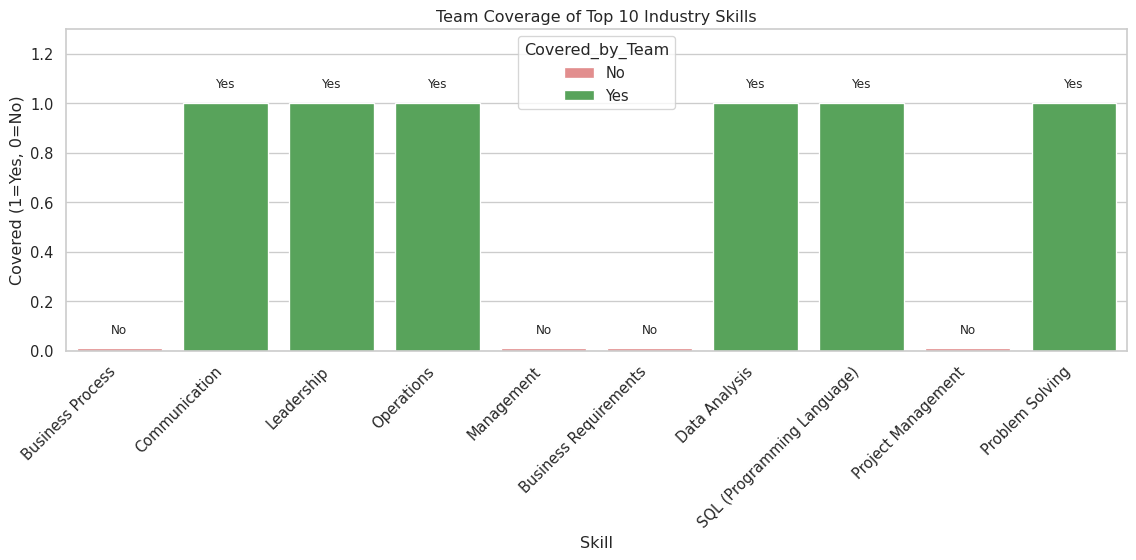
job\_posts = pd.read\_csv("data/lightcast\_job\_postings.csv", low\_memory=False)  
  
if "SKILLS\_NAME" in job\_posts.columns:  
 all\_skills = job\_posts["SKILLS\_NAME"].dropna().astype(str).str.split(",")  
 flat\_skills = [s.strip().strip('"') for sublist in all\_skills for s in sublist]  
 top\_job\_skills = pd.Series(flat\_skills).value\_counts().head(10)  
else:  
 top\_job\_skills = pd.Series([])  
  
print("Top 10 Industry Skills from Job Postings:")  
print(top\_job\_skills)

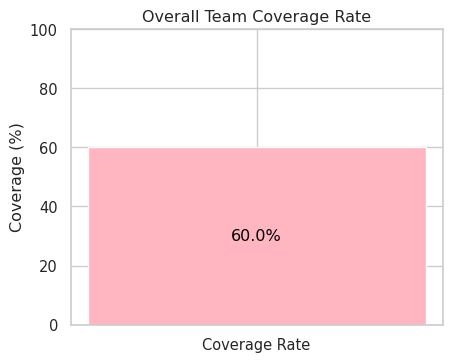
Top 10 Industry Skills from Job Postings:  
Communication 30768  
Data Analysis 26797  
Management 21274  
SQL (Programming Language) 20943  
Leadership 17535  
Problem Solving 16553  
Operations 14684  
Project Management 13609  
Business Process 13203  
Business Requirements 12977  
Name: count, dtype: int64

# Compare Team vs Industry Skills

rename\_map = {  
 "SQL": "SQL (Programming Language)",  
 "Python": "Data Analysis",   
 "Machine Learning": "Data Analysis",  
 "Cloud Computing": "Operations",   
 "Communication": "Communication",  
 "Problem Sloving": "Problem Solving",  
 "leadership": "Leadership"  
}  
  
team\_skills = set([rename\_map.get(c, c) for c in df\_skills.columns])  
industry\_skills = set(top\_job\_skills.index)  
covered\_skills = industry\_skills & team\_skills  
missing\_skills = industry\_skills - team\_skills  
  
print("✅ Covered Skills:", covered\_skills)  
print("⚠️ Missing Skills:", missing\_skills)  
  
skill\_coverage = pd.DataFrame({  
 "Skill": list(industry\_skills),  
 "Covered\_by\_Team": ["Yes" if s in covered\_skills else "No" for s in industry\_skills]  
})  
skill\_coverage["Covered\_Value"] = skill\_coverage["Covered\_by\_Team"].map({"Yes": 1, "No": 0}).astype(int)  
skill\_coverage["Plot\_Value"] = skill\_coverage["Covered\_Value"].replace(0, 0.01)  
  
plt.figure(figsize=(12,6))  
ax = sns.barplot(  
 data=skill\_coverage,   
 x="Skill",   
 y="Plot\_Value",   
 hue="Covered\_by\_Team",   
 dodge=False,   
 palette={"Yes":"#4CAF50", "No":"#F08080"}  
)  
  
for i, row in skill\_coverage.iterrows():  
 ax.text(i, row["Plot\_Value"]+0.05, row["Covered\_by\_Team"],   
 ha="center", va="bottom", fontsize=9)  
  
plt.xticks(rotation=45, ha="right")  
plt.title("Team Coverage of Top 10 Industry Skills")  
plt.ylabel("Covered (1=Yes, 0=No)")  
plt.ylim(0,1.3)  
plt.tight\_layout()  
plt.savefig("figures/skill\_gap\_analysis.png", dpi=300, bbox\_inches="tight")  
plt.show()  
  
coverage\_rate = len(covered\_skills) / len(industry\_skills) \* 100  
plt.figure(figsize=(5,4))  
plt.bar(["Coverage Rate"], [coverage\_rate], color="#FFB6C1")  
plt.ylabel("Coverage (%)")  
plt.ylim(0,100)  
plt.title("Overall Team Coverage Rate")  
plt.text(0, coverage\_rate/2, f"{coverage\_rate:.1f}%",   
 ha="center", va="center", fontsize=12, color="black")  
plt.savefig("figures/coverage\_rate.png", dpi=300, bbox\_inches="tight")  
plt.show()  
  
mapping\_table = pd.DataFrame({  
 "Team Skill": df\_skills.columns,  
 "Mapped Industry Skill": [rename\_map.get(c, c) for c in df\_skills.columns],  
 "In Top 10": ["Yes" if rename\_map.get(c, c) in industry\_skills else "No" for c in df\_skills.columns]  
})  
display(mapping\_table)

✅ Covered Skills: {'Communication', 'Leadership', 'SQL (Programming Language)', 'Data Analysis', 'Operations', 'Problem Solving'}  
⚠️ Missing Skills: {'Project Management', 'Management', 'Business Requirements', 'Business Process'}





|  | Team Skill | Mapped Industry Skill | In Top 10 |
| --- | --- | --- | --- |
| 0 | Python | Data Analysis | Yes |
| 1 | SQL | SQL (Programming Language) | Yes |
| 2 | Machine Learning | Data Analysis | Yes |
| 3 | Cloud Computing | Operations | Yes |
| 4 | Communication | Communication | Yes |
| 5 | Problem Solving | Problem Solving | Yes |
| 6 | Leadership | Leadership | Yes |

**Results** Our team analyzed and compared their self-assessed skills scores with the top ten most in-demand industry skills extracted from job postings. The results showed strong coverage across technical areas, particularly in data analytics, SQL, and operations, with Python, programming languages, and cloud computing as the foundation. These skills align closely with market demand and position the team well for data-centric and technical roles.

However, the analysis also revealed significant shortcomings. The team lacked formal expertise in project management, business needs, and leadership. While corporate HR departments consistently emphasize these areas as crucial for roles bridging technology and business, particularly in managing teams or translating technology solutions into business value, our overall coverage was 60%, highlighting significant room for improvement.

**Improvement Plan** While our team has certain strengths in technical skills like data analysis, SQL, and cloud computing, a comparison with industry requirements reveals significant shortcomings. Project management, business needs analysis, and communication are particularly important skills that recruiters value most, yet they are currently largely absent within our team. Therefore, if we aspire to take on more comprehensive roles in the future, relying solely on technical proficiency will not be enough.

To address our shortcomings, we need to proactively develop these soft skills. For example, in class group assignments, we can try creating a project coordination role to hone project management and communication skills through practical experience. Outside of class, we can leverage BU resources to participate in projects and courses related to Project Management and Business Process Management to enhance these skills. Leadership is also a weakness worth addressing. While each of us has demonstrated organizational skills to varying degrees, our overall level is still insufficient. This can be developed through rotating leadership roles within the group.

In the long run, we must not only maintain our edge in data analysis and technology but also gradually build a comprehensive capability structure that encompasses both software and hardware. This will allow us to secure the jobs we desire in a highly competitive job market. Furthermore, we will be able to not only produce technical achievements at work but also drive projects to fruition.

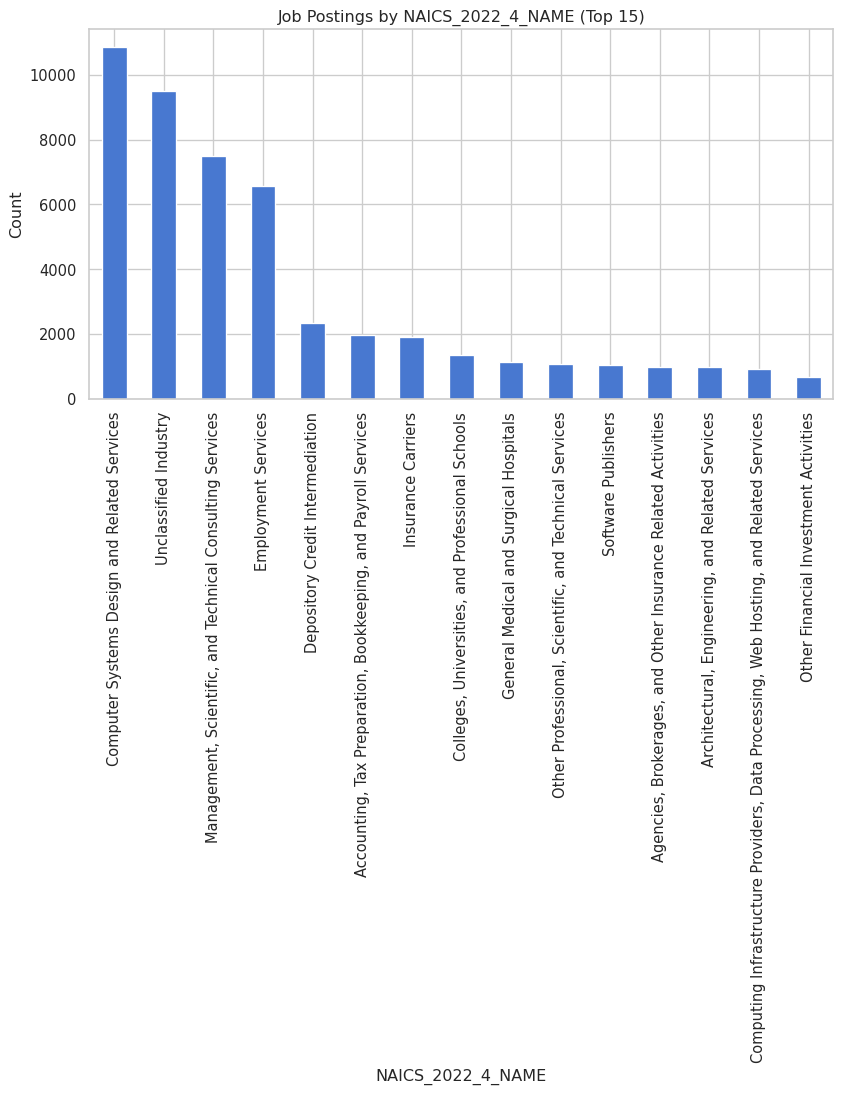
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
import numpy as np  
import os  
from IPython.display import display  
  
  
sns.set\_theme(style="whitegrid", palette="muted")  
df = pd.read\_csv("data/lightcast\_job\_postings.csv", low\_memory=False)

# Salary setup  
if "SALARY\_FROM" in df.columns and "SALARY\_TO" in df.columns:  
 df["AVG\_SALARY"] = (  
 pd.to\_numeric(df["SALARY\_FROM"], errors="coerce") +  
 pd.to\_numeric(df["SALARY\_TO"], errors="coerce")  
 ) / 2  
  
salary\_candidates = ["AVG\_SALARY", "SALARY", "Average\_Salary", "AVERAGE\_SALARY"]  
salary\_col = next((c for c in salary\_candidates if c in df.columns), None)  
  
# Industry setup  
industry\_candidates = ["Industry", "NAICS\_2022\_4\_NAME", "LIGHTCAST\_SECTORS\_NAME"]  
industry\_col = next((c for c in industry\_candidates if c in df.columns), None)  
  
print("Detected salary\_col:", salary\_col)  
print("Detected industry\_col:", industry\_col)

Detected salary\_col: AVG\_SALARY  
Detected industry\_col: NAICS\_2022\_4\_NAME

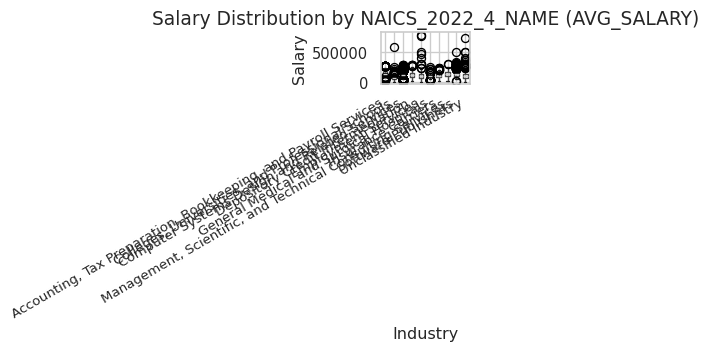
# Job postings by Industry  
if industry\_col:  
 plt.figure(figsize=(10,5))  
 df[industry\_col].value\_counts().head(15).plot(kind="bar")  
 plt.title(f"Job Postings by {industry\_col} (Top 15)")  
 plt.xlabel(industry\_col)  
 plt.ylabel("Count")  
 plt.tight\_layout()  
 plt.savefig("figures/job\_postings\_by\_industry.png", dpi=300, bbox\_inches="tight")  
 display(plt.gcf())  
 plt.close()

/tmp/ipykernel\_6941/121335137.py:8: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all Axes decorations.  
 plt.tight\_layout()



**Analysis** Computer Systems Design and Related Services accounts for the largest number of job postings, followed by Management, Scientific, and Technical Consulting Services. This indicates that the technology and consulting industries are currently the primary sectors for hiring. Job opportunities are primarily concentrated in areas related to digital transformation and information technology.

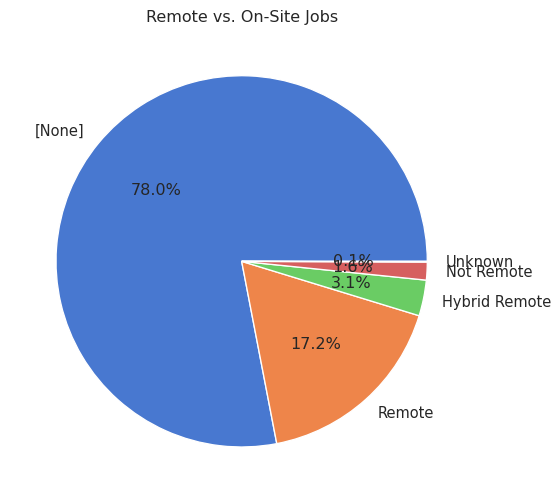
# Salary distribution by Industry  
if salary\_col and industry\_col:  
 df[salary\_col] = pd.to\_numeric(df[salary\_col], errors="coerce")  
 tmp = df[[industry\_col, salary\_col]].dropna()  
  
 if not tmp.empty:  
 top\_ind = tmp[industry\_col].value\_counts().head(10).index  
 tmp = tmp[tmp[industry\_col].isin(top\_ind)]  
  
 plt.figure(figsize=(14,8))   
 tmp.boxplot(column=salary\_col, by=industry\_col)  
  
 plt.title(f"Salary Distribution by {industry\_col} ({salary\_col})", fontsize=14)  
 plt.suptitle("")  
 plt.xlabel("Industry", fontsize=12)  
 plt.ylabel("Salary", fontsize=12)  
 plt.xticks(rotation=30, ha="right", fontsize=10)  
 plt.tight\_layout()  
 plt.savefig("figures/salary\_distribution\_by\_industry.png", dpi=300, bbox\_inches="tight")  
 display(plt.gcf())  
 plt.close()



<Figure size 1344x768 with 0 Axes>

**Analysis** Salary distribution shows that finance, healthcare, and consulting industries have relatively high salaries, while traditional industries like education and accounting have relatively concentrated salaries with a lower range. However, there is a large dispersion in salaries in financial services, indicating that there are many high-paying positions but overall competition is fierce.

# Remote vs On-site  
remote\_col = next((c for c in ["REMOTE\_TYPE\_NAME","REMOTE\_TYPE","Remote\_Type","REMOTE\_GROUP"] if c in df.columns), None)  
if remote\_col:  
 plt.figure(figsize=(6,6))  
 df[remote\_col].fillna("Unknown").value\_counts().plot(kind="pie", autopct="%1.1f%%")  
 plt.title("Remote vs. On-Site Jobs")  
 plt.ylabel("")  
 plt.tight\_layout()  
 plt.savefig("figures/remote\_vs\_onsite.png", dpi=300, bbox\_inches="tight")  
 display(plt.gcf())  
 plt.close()  
  
print("Final Columns:", df.columns.tolist())



Final Columns: ['ID', 'LAST\_UPDATED\_DATE', 'LAST\_UPDATED\_TIMESTAMP', 'DUPLICATES', 'POSTED', 'EXPIRED', 'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'URL', 'ACTIVE\_URLS', 'ACTIVE\_SOURCES\_INFO', 'TITLE\_RAW', 'BODY', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'COMPANY', 'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS', 'MIN\_EDULEVELS\_NAME', 'MAX\_EDULEVELS', 'MAX\_EDULEVELS\_NAME', 'EMPLOYMENT\_TYPE', 'EMPLOYMENT\_TYPE\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP', 'SALARY', 'REMOTE\_TYPE', 'REMOTE\_TYPE\_NAME', 'ORIGINAL\_PAY\_PERIOD', 'SALARY\_TO', 'SALARY\_FROM', 'LOCATION', 'CITY', 'CITY\_NAME', 'COUNTY', 'COUNTY\_NAME', 'MSA', 'MSA\_NAME', 'STATE', 'STATE\_NAME', 'COUNTY\_OUTGOING', 'COUNTY\_NAME\_OUTGOING', 'COUNTY\_INCOMING', 'COUNTY\_NAME\_INCOMING', 'MSA\_OUTGOING', 'MSA\_NAME\_OUTGOING', 'MSA\_INCOMING', 'MSA\_NAME\_INCOMING', 'NAICS2', 'NAICS2\_NAME', 'NAICS3', 'NAICS3\_NAME', 'NAICS4', 'NAICS4\_NAME', 'NAICS5', 'NAICS5\_NAME', 'NAICS6', 'NAICS6\_NAME', 'TITLE', 'TITLE\_NAME', 'TITLE\_CLEAN', 'SKILLS', 'SKILLS\_NAME', 'SPECIALIZED\_SKILLS', 'SPECIALIZED\_SKILLS\_NAME', 'CERTIFICATIONS', 'CERTIFICATIONS\_NAME', 'COMMON\_SKILLS', 'COMMON\_SKILLS\_NAME', 'SOFTWARE\_SKILLS', 'SOFTWARE\_SKILLS\_NAME', 'ONET', 'ONET\_NAME', 'ONET\_2019', 'ONET\_2019\_NAME', 'CIP6', 'CIP6\_NAME', 'CIP4', 'CIP4\_NAME', 'CIP2', 'CIP2\_NAME', 'SOC\_2021\_2', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4', 'SOC\_2021\_4\_NAME', 'SOC\_2021\_5', 'SOC\_2021\_5\_NAME', 'LOT\_CAREER\_AREA', 'LOT\_CAREER\_AREA\_NAME', 'LOT\_OCCUPATION', 'LOT\_OCCUPATION\_NAME', 'LOT\_SPECIALIZED\_OCCUPATION', 'LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_OCCUPATION\_GROUP', 'LOT\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_SPECIALIZED\_OCCUPATION', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION', 'LOT\_V6\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION\_GROUP', 'LOT\_V6\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_CAREER\_AREA', 'LOT\_V6\_CAREER\_AREA\_NAME', 'SOC\_2', 'SOC\_2\_NAME', 'SOC\_3', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME', 'SOC\_5', 'SOC\_5\_NAME', 'LIGHTCAST\_SECTORS', 'LIGHTCAST\_SECTORS\_NAME', 'NAICS\_2022\_2', 'NAICS\_2022\_2\_NAME', 'NAICS\_2022\_3', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_4', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_5', 'NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6', 'NAICS\_2022\_6\_NAME', 'AVG\_SALARY']

**Analysis** Remote work accounts for approximately 17%, while traditional on-site work still accounts for over 70%. This suggests that while remote positions exist, most companies still maintain an offline work model. Hybrid remote work accounts for a small percentage and is not currently supported by companies, but it may become a trend in the future.

**Summary** This EDA reveals the following:

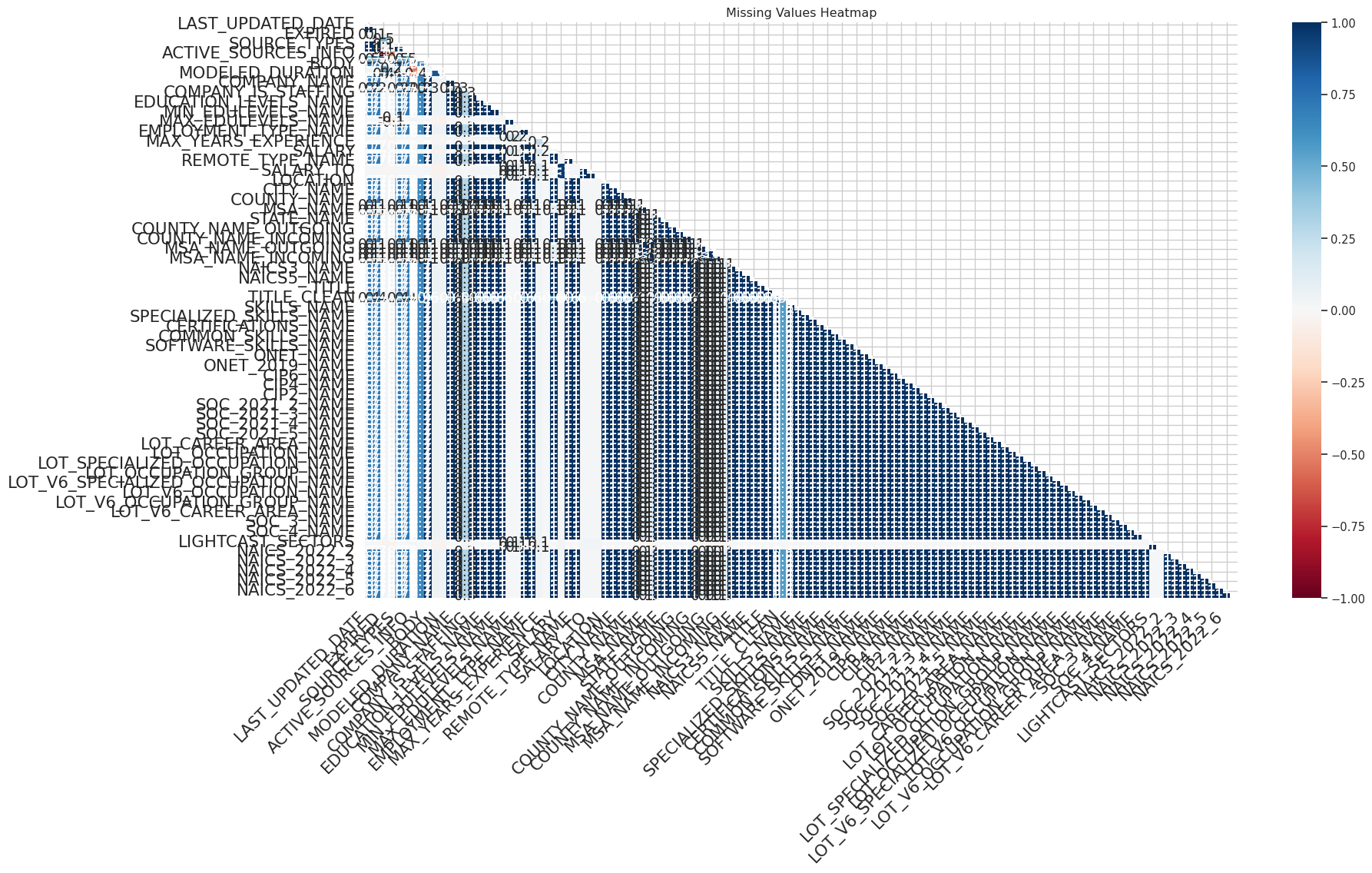
* Hiring is concentrated in the IT and consulting industries, demonstrating a clear demand for digital transformation.
* Salaries in finance and healthcare are high and fluctuate widely, indicating fierce competition for certain positions within these sectors.
* Remote positions remain a minority, but their growth potential warrants attention.

title: “Data Analysis” subtitle: “Comprehensive Data Cleaning & Exploratory Analysis of Job Market Trends” author: - name: Yibei Yu, Fuhan Zhang, Jonathan Leon affiliations: - id: bu name: Boston University city: Boston state: MA bibliography: references.bib csl: csl/econometrica.csl format: html: toc: true number-sections: true df-print: paged

import pandas as pd  
import missingno as msno  
import matplotlib.pyplot as plt  
import numpy as np  
import os  
from IPython.display import display  
import seaborn as sns  
sns.set\_theme(style="whitegrid")  
  
os.makedirs("figures", exist\_ok=True)  
  
df = pd.read\_csv("data/lightcast\_job\_postings.csv")  
print("Columns in dataset:", df.columns.tolist())  
  
# drop irrelevant columns  
columns\_to\_drop = [  
 "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",  
 "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",  
 "SOC\_2", "SOC\_3", "SOC\_5"  
]  
df.drop(columns=columns\_to\_drop, inplace=True, errors="ignore")  
  
# Missing values heatmap  
plt.figure(figsize=(10,6))  
msno.heatmap(df)  
plt.title("Missing Values Heatmap")  
plt.tight\_layout()  
plt.savefig("figures/missing\_values\_heatmap.png", dpi=300, bbox\_inches="tight")  
display(plt.gcf())  
plt.close()  
  
salary\_related = ["SALARY", "Salary", "Average\_Salary", "AVERAGE\_SALARY", "SALARY\_FROM", "SALARY\_TO"]  
thresh = len(df) \* 0.5  
df = df.loc[:, (df.notna().sum() >= thresh) | (df.columns.isin(salary\_related))]  
  
# numeric vs categorical cleaning  
num\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()  
cat\_cols = [c for c in df.columns if c not in num\_cols]  
for c in num\_cols:  
 df[c] = pd.to\_numeric(df[c], errors="coerce")  
df[num\_cols] = df[num\_cols].fillna(df[num\_cols].median())  
for c in cat\_cols:  
 df[c] = df[c].fillna("Unknown")  
  
if "SALARY\_FROM" in df.columns and "SALARY\_TO" in df.columns:  
 df["AVG\_SALARY"] = (  
 pd.to\_numeric(df["SALARY\_FROM"], errors="coerce") +  
 pd.to\_numeric(df["SALARY\_TO"], errors="coerce")  
 ) / 2  
  
salary\_candidates = ["AVG\_SALARY", "SALARY", "Salary", "Average\_Salary", "AVERAGE\_SALARY"]  
salary\_col = next((c for c in salary\_candidates if c in df.columns), None)  
  
if salary\_col:  
 df[salary\_col] = pd.to\_numeric(df[salary\_col], errors="coerce")  
 q1, q3 = df[salary\_col].quantile([0.25, 0.75])  
 iqr = q3 - q1  
 lo, hi = q1 - 1.5\*iqr, q3 + 1.5\*iqr  
 df[salary\_col] = df[salary\_col].clip(lower=lo, upper=hi)  
  
print("Detected salary\_col:", salary\_col)  
  
dup\_keys = [c for c in ["TITLE", "COMPANY", "LOCATION", "POSTED"] if c in df.columns]  
if dup\_keys:  
 before = len(df)  
 df = df.drop\_duplicates(subset=dup\_keys, keep="first")  
 after = len(df)  
 print(f"Removed duplicates: {before - after}")  
else:  
 df = df.drop\_duplicates()

/tmp/ipykernel\_6941/3896394006.py:12: DtypeWarning: Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.  
 df = pd.read\_csv("data/lightcast\_job\_postings.csv")

Columns in dataset: ['ID', 'LAST\_UPDATED\_DATE', 'LAST\_UPDATED\_TIMESTAMP', 'DUPLICATES', 'POSTED', 'EXPIRED', 'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'URL', 'ACTIVE\_URLS', 'ACTIVE\_SOURCES\_INFO', 'TITLE\_RAW', 'BODY', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'COMPANY', 'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS', 'MIN\_EDULEVELS\_NAME', 'MAX\_EDULEVELS', 'MAX\_EDULEVELS\_NAME', 'EMPLOYMENT\_TYPE', 'EMPLOYMENT\_TYPE\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP', 'SALARY', 'REMOTE\_TYPE', 'REMOTE\_TYPE\_NAME', 'ORIGINAL\_PAY\_PERIOD', 'SALARY\_TO', 'SALARY\_FROM', 'LOCATION', 'CITY', 'CITY\_NAME', 'COUNTY', 'COUNTY\_NAME', 'MSA', 'MSA\_NAME', 'STATE', 'STATE\_NAME', 'COUNTY\_OUTGOING', 'COUNTY\_NAME\_OUTGOING', 'COUNTY\_INCOMING', 'COUNTY\_NAME\_INCOMING', 'MSA\_OUTGOING', 'MSA\_NAME\_OUTGOING', 'MSA\_INCOMING', 'MSA\_NAME\_INCOMING', 'NAICS2', 'NAICS2\_NAME', 'NAICS3', 'NAICS3\_NAME', 'NAICS4', 'NAICS4\_NAME', 'NAICS5', 'NAICS5\_NAME', 'NAICS6', 'NAICS6\_NAME', 'TITLE', 'TITLE\_NAME', 'TITLE\_CLEAN', 'SKILLS', 'SKILLS\_NAME', 'SPECIALIZED\_SKILLS', 'SPECIALIZED\_SKILLS\_NAME', 'CERTIFICATIONS', 'CERTIFICATIONS\_NAME', 'COMMON\_SKILLS', 'COMMON\_SKILLS\_NAME', 'SOFTWARE\_SKILLS', 'SOFTWARE\_SKILLS\_NAME', 'ONET', 'ONET\_NAME', 'ONET\_2019', 'ONET\_2019\_NAME', 'CIP6', 'CIP6\_NAME', 'CIP4', 'CIP4\_NAME', 'CIP2', 'CIP2\_NAME', 'SOC\_2021\_2', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4', 'SOC\_2021\_4\_NAME', 'SOC\_2021\_5', 'SOC\_2021\_5\_NAME', 'LOT\_CAREER\_AREA', 'LOT\_CAREER\_AREA\_NAME', 'LOT\_OCCUPATION', 'LOT\_OCCUPATION\_NAME', 'LOT\_SPECIALIZED\_OCCUPATION', 'LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_OCCUPATION\_GROUP', 'LOT\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_SPECIALIZED\_OCCUPATION', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION', 'LOT\_V6\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION\_GROUP', 'LOT\_V6\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_CAREER\_AREA', 'LOT\_V6\_CAREER\_AREA\_NAME', 'SOC\_2', 'SOC\_2\_NAME', 'SOC\_3', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME', 'SOC\_5', 'SOC\_5\_NAME', 'LIGHTCAST\_SECTORS', 'LIGHTCAST\_SECTORS\_NAME', 'NAICS\_2022\_2', 'NAICS\_2022\_2\_NAME', 'NAICS\_2022\_3', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_4', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_5', 'NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6', 'NAICS\_2022\_6\_NAME']



Detected salary\_col: AVG\_SALARY  
Removed duplicates: 3300

/tmp/ipykernel\_6941/3896394006.py:46: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`  
 df["AVG\_SALARY"] = (

<Figure size 960x576 with 0 Axes>

title: “Regression, Classification, and Topic Insights” subtitle: “Analyzing Salary and Industry Trends for ML/Data Science Roles” author: - name: “Yibei Yu, Fuhan Zhang, Jonathan Leon” affiliations: - id: bu name: Boston University city: Boston state: MA date: “2025-09-27” bibliography: references.bib csl: csl/econometrica.csl toc: true format: html: default

## Introduction

This module extends previous analyses by applying **machine-learning clustering and regression models** to identify salary patterns across industries, contrasting jobs that require machine learning (ML) / data-science (DS) skills with those that do not.

import pandas as pd  
import missingno as msno  
import matplotlib.pyplot as plt  
import numpy as np  
import os  
from IPython.display import display  
import seaborn as sns  
sns.set\_theme(style="whitegrid")  
  
os.makedirs("figures", exist\_ok=True)  
  
csv\_path = "data/lightcast\_job\_postings.csv"  
if not os.path.exists(csv\_path):  
 print(f"Data file not found: {csv\_path}")  
 print("Place the dataset in the `data/` folder or update the path. Skipping data load.")  
 df = None  
else:  
 df = pd.read\_csv(csv\_path)  
 print("Columns in dataset:", df.columns.tolist())  
# close previous python block and start a new one for analysis

/tmp/ipykernel\_6941/951617275.py:18: DtypeWarning: Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.  
 df = pd.read\_csv(csv\_path)

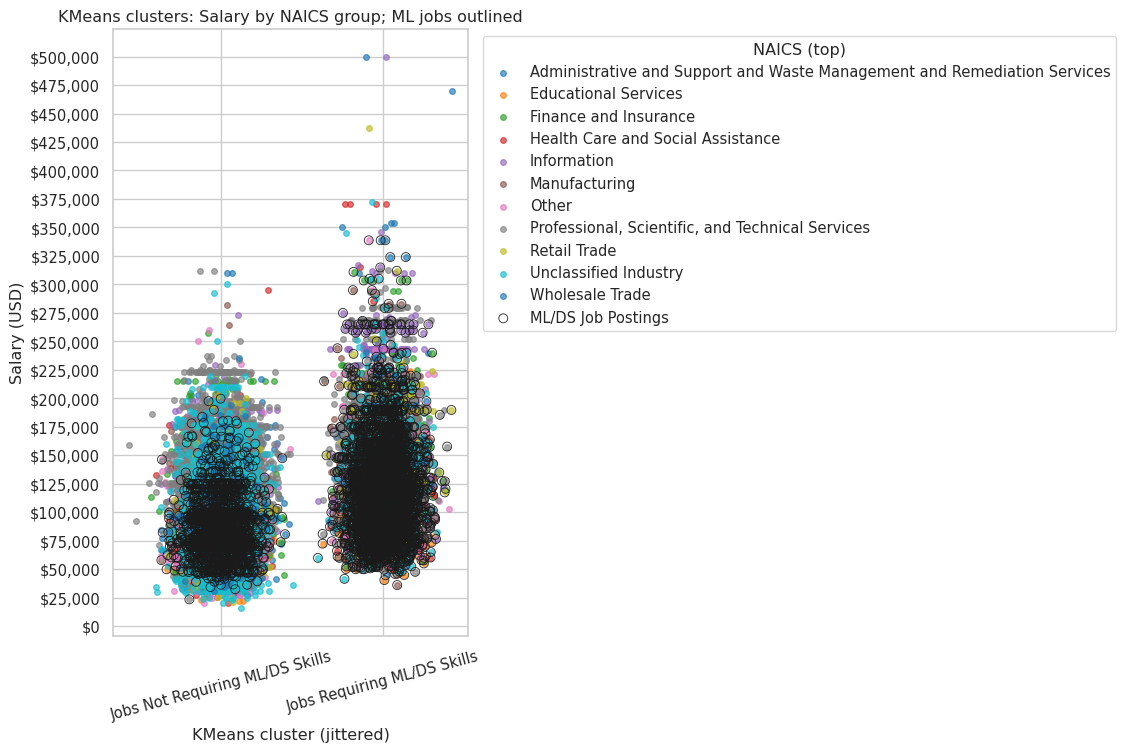
Columns in dataset: ['ID', 'LAST\_UPDATED\_DATE', 'LAST\_UPDATED\_TIMESTAMP', 'DUPLICATES', 'POSTED', 'EXPIRED', 'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'URL', 'ACTIVE\_URLS', 'ACTIVE\_SOURCES\_INFO', 'TITLE\_RAW', 'BODY', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'COMPANY', 'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS', 'MIN\_EDULEVELS\_NAME', 'MAX\_EDULEVELS', 'MAX\_EDULEVELS\_NAME', 'EMPLOYMENT\_TYPE', 'EMPLOYMENT\_TYPE\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP', 'SALARY', 'REMOTE\_TYPE', 'REMOTE\_TYPE\_NAME', 'ORIGINAL\_PAY\_PERIOD', 'SALARY\_TO', 'SALARY\_FROM', 'LOCATION', 'CITY', 'CITY\_NAME', 'COUNTY', 'COUNTY\_NAME', 'MSA', 'MSA\_NAME', 'STATE', 'STATE\_NAME', 'COUNTY\_OUTGOING', 'COUNTY\_NAME\_OUTGOING', 'COUNTY\_INCOMING', 'COUNTY\_NAME\_INCOMING', 'MSA\_OUTGOING', 'MSA\_NAME\_OUTGOING', 'MSA\_INCOMING', 'MSA\_NAME\_INCOMING', 'NAICS2', 'NAICS2\_NAME', 'NAICS3', 'NAICS3\_NAME', 'NAICS4', 'NAICS4\_NAME', 'NAICS5', 'NAICS5\_NAME', 'NAICS6', 'NAICS6\_NAME', 'TITLE', 'TITLE\_NAME', 'TITLE\_CLEAN', 'SKILLS', 'SKILLS\_NAME', 'SPECIALIZED\_SKILLS', 'SPECIALIZED\_SKILLS\_NAME', 'CERTIFICATIONS', 'CERTIFICATIONS\_NAME', 'COMMON\_SKILLS', 'COMMON\_SKILLS\_NAME', 'SOFTWARE\_SKILLS', 'SOFTWARE\_SKILLS\_NAME', 'ONET', 'ONET\_NAME', 'ONET\_2019', 'ONET\_2019\_NAME', 'CIP6', 'CIP6\_NAME', 'CIP4', 'CIP4\_NAME', 'CIP2', 'CIP2\_NAME', 'SOC\_2021\_2', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4', 'SOC\_2021\_4\_NAME', 'SOC\_2021\_5', 'SOC\_2021\_5\_NAME', 'LOT\_CAREER\_AREA', 'LOT\_CAREER\_AREA\_NAME', 'LOT\_OCCUPATION', 'LOT\_OCCUPATION\_NAME', 'LOT\_SPECIALIZED\_OCCUPATION', 'LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_OCCUPATION\_GROUP', 'LOT\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_SPECIALIZED\_OCCUPATION', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION', 'LOT\_V6\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION\_GROUP', 'LOT\_V6\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_CAREER\_AREA', 'LOT\_V6\_CAREER\_AREA\_NAME', 'SOC\_2', 'SOC\_2\_NAME', 'SOC\_3', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME', 'SOC\_5', 'SOC\_5\_NAME', 'LIGHTCAST\_SECTORS', 'LIGHTCAST\_SECTORS\_NAME', 'NAICS\_2022\_2', 'NAICS\_2022\_2\_NAME', 'NAICS\_2022\_3', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_4', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_5', 'NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6', 'NAICS\_2022\_6\_NAME']

# --- classify ML/Data-Science skills and cluster by salary ---  
import re  
from sklearn.cluster import KMeans  
from sklearn.preprocessing import StandardScaler  
  
def safe\_tokens(val):  
 if pd.isna(val):  
 return []  
 s = str(val)  
 # handle Python lists  
 try:  
 import ast  
 parsed = ast.literal\_eval(s)  
 if isinstance(parsed, (list, tuple, set)):  
 return [str(x).strip() for x in parsed if x not in (None, '')]  
 except Exception:  
 pass  
 # split on common separators  
 parts = re.split(r"\||;|\\n|\\r|\\t|,", s)  
 return [p.strip().strip('\"\'') for p in parts if p.strip()]  
  
# keywords to mark ML/DS-related skills  
ml\_keywords = [  
 'machine learning','data science','deep learning','neural','tensorflow','pytorch',  
 'scikit','sklearn','xgboost','lightgbm','pandas','numpy','nlp','natural language',  
 'computer vision','keras','statistics','statistical','rnn','cnn'  
]  
  
def count\_ml\_tokens(tokens):  
 cnt = 0  
 for t in tokens:  
 low = t.lower()  
 for k in ml\_keywords:  
 if k in low:  
 cnt += 1  
 break  
 return cnt  
  
# combine several skill/name columns to form a token set per posting  
skill\_cols = ['SKILLS\_NAME','SPECIALIZED\_SKILLS\_NAME','COMMON\_SKILLS\_NAME','SOFTWARE\_SKILLS\_NAME']  
for c in skill\_cols:  
 if c not in df.columns:  
 df[c] = None  
  
def build\_tokens\_row(row):  
 toks = []  
 for c in skill\_cols:  
 toks.extend(safe\_tokens(row.get(c, '')))  
 return toks  
  
df['all\_skill\_tokens'] = df.apply(build\_tokens\_row, axis=1)  
df['ml\_skill\_count'] = df['all\_skill\_tokens'].apply(count\_ml\_tokens)  
df['n\_skills'] = df['all\_skill\_tokens'].apply(len)  
  
# build a salary numeric column: prefer SALARY (if present) else average of SALARY\_FROM/SALARY\_TO  
if 'SALARY' in df.columns:  
 df['salary\_num'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
else:  
 sf = pd.to\_numeric(df.get('SALARY\_FROM', pd.Series([None]\*len(df))), errors='coerce')  
 st = pd.to\_numeric(df.get('SALARY\_TO', pd.Series([None]\*len(df))), errors='coerce')  
 df['salary\_num'] = sf  
 # if both available, take mean  
 df.loc[~sf.isna() & ~st.isna(), 'salary\_num'] = (sf + st)/2  
  
# keep rows with a positive salary  
df\_model = df[df['salary\_num'].notna() & (df['salary\_num']>0)].copy()  
if df\_model.shape[0] == 0:  
 print('No salary data available for clustering.')  
else:  
 X = df\_model[['salary\_num','ml\_skill\_count','n\_skills']].fillna(0).to\_numpy()  
 scaler = StandardScaler()  
 Xs = scaler.fit\_transform(X)  
  
 # cluster into 2 groups (we'll map one group to ML jobs and the other to non-ML)  
 kmeans = KMeans(n\_clusters=2, random\_state=42, n\_init=10)  
 labels = kmeans.fit\_predict(Xs)  
 df\_model['cluster'] = labels  
  
 # mark ML job by whether it has any ML tokens  
 df\_model['is\_ml\_job'] = df\_model['ml\_skill\_count']>0  
  
 # determine which cluster has the higher proportion of ML jobs and label clusters accordingly  
 try:  
 cluster\_ml\_pct = df\_model.groupby('cluster')['is\_ml\_job'].mean()  
 if not cluster\_ml\_pct.empty:  
 ml\_cluster = int(cluster\_ml\_pct.idxmax())  
 else:  
 ml\_cluster = 1  
 except Exception:  
 ml\_cluster = 1  
 df\_model['cluster\_is\_ml'] = df\_model['cluster'].apply(lambda c: c == ml\_cluster)  
 df\_model['cluster\_label'] = df\_model['cluster\_is\_ml'].map({True: 'ML cluster', False: 'Non-ML cluster'})  
  
 # summarize clusters  
 summary = df\_model.groupby('cluster').agg(  
 postings=('cluster','size'),  
 avg\_salary=('salary\_num','mean'),  
 median\_salary=('salary\_num','median'),  
 avg\_ml\_skill\_count=('ml\_skill\_count','mean'),  
 pct\_ml\_jobs=('is\_ml\_job', lambda x: 100\*x.mean())  
 ).sort\_index()  
  
 print('\nCluster summary (salary and ML presence):')  
 print(summary)  
  
 # show cluster centers in original feature space  
 centers = scaler.inverse\_transform(kmeans.cluster\_centers\_)  
 print('\nCluster centers (salary\_num, ml\_skill\_count, n\_skills):')  
 for i,c in enumerate(centers):  
 print(f'Cluster {i}:', c)  
  
 # show top 10 sample postings per cluster that are ML jobs  
 for i in sorted(df\_model['cluster'].unique()):  
 print('\n' + '='\*40)  
 print(f'Cluster {i} — top ML postings (sample)')  
 sample = df\_model[(df\_model['cluster']==i) & (df\_model['is\_ml\_job'])].head(10)  
 if sample.empty:  
 print(' (no ML postings in this cluster)')  
 else:  
 for idx,row in sample.iterrows():  
 title = row.get('TITLE') or row.get('TITLE\_NAME') or ''  
 print(f" id={row.get('ID')} | salary={row.get('salary\_num'):.0f} | ml\_count={row.get('ml\_skill\_count')} | title={title}")

Cluster summary (salary and ML presence):  
 postings avg\_salary median\_salary avg\_ml\_skill\_count \  
cluster   
0 20496 107614.271858 106080.0 0.305328   
1 10312 138504.380237 133700.0 2.484678   
  
 pct\_ml\_jobs   
cluster   
0 12.987900   
1 52.269201   
  
Cluster centers (salary\_num, ml\_skill\_count, n\_skills):  
Cluster 0: [1.07642300e+05 3.05321173e-01 3.73175633e+01]  
Cluster 1: [1.38469598e+05 2.48617176e+00 8.37555556e+01]  
  
========================================  
Cluster 0 — top ML postings (sample)  
 id=2725b337958d2ca49d99a8768741e6090bc6a74d | salary=84678 | ml\_count=2 | title=ETCD1DCE2110BB27EB  
 id=9608eb18fbcc70dae3f59025cca9993e89cb94a9 | salary=101798 | ml\_count=2 | title=ET3037E0C947A02404  
 id=a0db3935cdbf3d42e7a79b83f6e8074b329e085b | salary=171600 | ml\_count=2 | title=ETA167677FC704C4AD  
 id=b1de4586ebd2d2e67680b9c18d543dddef07e47d | salary=70880 | ml\_count=2 | title=ET3037E0C947A02404  
 id=d1be2dfb08255f7fdad6b302a4190cdf5e422720 | salary=69680 | ml\_count=2 | title=ET6DA4FDEAD969847C  
 id=8b445797f947f39c25b55c2c572137b71045a8ac | salary=69680 | ml\_count=2 | title=ET8E0C473DFBEDF8C8  
 id=80b1483d1b8816ec99650ea778bab64adf8c50d6 | salary=69680 | ml\_count=2 | title=ET6DA4FDEAD969847C  
 id=23f8f04e4a10c70f6d0a28cb74b35ba173d9a7e8 | salary=69680 | ml\_count=2 | title=ETA945BF69C78F1F2B  
 id=4c6903192f71f03f304f7484b73fd4d9610b1e69 | salary=165500 | ml\_count=2 | title=ET13E3A2844323866E  
 id=530c8ed958399be5655568d64c3ce2da86f20ff7 | salary=68576 | ml\_count=4 | title=ET6B57FB37DA4AA8A8  
  
========================================  
Cluster 1 — top ML postings (sample)  
 id=229620073766234e814e8add21db7dfaef69b3bd | salary=92962 | ml\_count=8 | title=ET1CE3CFA5447376E9  
 id=f361bb10174a44d316c48ac7ce669390abbf7c7b | salary=136950 | ml\_count=15 | title=ET5AA72D0E18D0EFE5  
 id=146621e071735303b16f75333b8593fb3f245ea0 | salary=118560 | ml\_count=6 | title=ET0000000000000000  
 id=9ec8abde1f4b88863ec96b3523e03f0a39b2e5bf | salary=140756 | ml\_count=13 | title=ET9B37BAEB716CCCDA  
 id=e68a72d999879ef6969e57be84023c6243716bdc | salary=156038 | ml\_count=19 | title=ET00335BE0181594E1  
 id=2e9acb4dfcf3979abcb892fdb02e8792cfc74a04 | salary=161840 | ml\_count=4 | title=ETC1360A6DCAF5E713  
 id=2b9cd1c2f0413e5eee667a0e785f131f3ab50817 | salary=103573 | ml\_count=4 | title=ET808060B2DD7A4902  
 id=ea8bc28ca5f1e012159fa50e4a17947270431a9a | salary=122500 | ml\_count=2 | title=ET3037E0C947A02404  
 id=5575c78b966843a96790769924eb8c5335367e23 | salary=71000 | ml\_count=8 | title=ET3037E0C947A02404  
 id=0625296f2e6627c2261bf27e634bad063442d1de | salary=76460 | ml\_count=4 | title=ET29293A7C0D786B75

# --- Static Matplotlib KMeans scatter: salary (y, log) vs cluster (x jitter), color by NAICS group, outline for ML jobs ---  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.ticker import FuncFormatter  
import math  
  
plot\_df = df\_model[df\_model['salary\_num'].notna() & (df\_model['salary\_num']>0)].copy()  
if plot\_df.empty:  
 print('No salary data available for static plotting.')  
else:  
 if 'cluster' not in plot\_df.columns:  
 from sklearn.cluster import KMeans  
 from sklearn.preprocessing import StandardScaler  
 X = plot\_df[['salary\_num','ml\_skill\_count','n\_skills']].fillna(0).to\_numpy()  
 sc = StandardScaler(); Xs = sc.fit\_transform(X)  
 kmeans\_tmp = KMeans(n\_clusters=2, random\_state=42, n\_init=10).fit(Xs)  
 plot\_df['cluster'] = kmeans\_tmp.labels\_  
 # map which temporary cluster is ML-heavy  
 try:  
 tmp\_ml\_pct = plot\_df.groupby('cluster')['ml\_skill\_count'].apply(lambda s: (s>0).mean())  
 tmp\_ml\_cluster = int(tmp\_ml\_pct.idxmax())  
 except Exception:  
 tmp\_ml\_cluster = 1  
 plot\_df['cluster\_is\_ml'] = plot\_df['cluster'].apply(lambda c: c == tmp\_ml\_cluster)  
 plot\_df['cluster'] = plot\_df['cluster\_is\_ml'].astype(int)  
  
 # Robust NAICS/industry column detection (case-insensitive). Prefer name/title/description columns.  
 import re as \_re  
 cols = list(plot\_df.columns)  
 naics\_like = [c for c in cols if \_re.search(r'naics|industry', c, \_re.I)]  
 naics\_name\_cols = [c for c in naics\_like if \_re.search(r'name|title|desc|sector', c, \_re.I)]  
 naics\_code\_cols = [c for c in naics\_like if \_re.search(r'code|id|num|^naics$', c, \_re.I)]  
  
 if naics\_name\_cols:  
 naics\_col = naics\_name\_cols[0]  
 plot\_df['naics\_group'] = plot\_df[naics\_col].fillna('Unknown').astype(str)  
 elif naics\_like and naics\_code\_cols:  
 # prefer code column if only codes are present  
 code\_col = naics\_code\_cols[0]  
 plot\_df['naics\_group'] = plot\_df[code\_col].fillna('Unknown').astype(str).apply(lambda v: f"NAICS {v}" if str(v).strip()!='' else 'Unknown')  
 elif naics\_like:  
 # fallback: use first matching column  
 naics\_col = naics\_like[0]  
 plot\_df['naics\_group'] = plot\_df[naics\_col].fillna('Unknown').astype(str)  
 else:  
 plot\_df['naics\_group'] = 'Unknown'  
  
 top\_naics = plot\_df['naics\_group'].value\_counts().nlargest(10).index.tolist()  
 plot\_df['naics\_top'] = plot\_df['naics\_group'].where(plot\_df['naics\_group'].isin(top\_naics), 'Other')  
  
 plot\_df['ml\_flag'] = plot\_df['is\_ml\_job']  
 rng = np.random.default\_rng(6)  
 plot\_df['x\_jitter'] = plot\_df['cluster'].astype(int) + rng.normal(0, 0.12, size=len(plot\_df))  
  
 # sample for plotting to keep figure readable  
 sample\_df = plot\_df.sample(n=min(20000, len(plot\_df)), random\_state=7)  
  
 plt.figure(figsize=(12,8))  
 groups = sample\_df.groupby('naics\_top')  
 cmap = plt.get\_cmap('tab10')  
 colors = {g: cmap(i % 10) for i,g in enumerate(groups.groups.keys())}  
  
 for g, sub in groups:  
 plt.scatter(sub['x\_jitter'], sub['salary\_num'], s=18, alpha=0.65, label=g, color=colors[g])  
  
 # outline ML jobs (do not add a legend entry for the outline)  
 ml\_sub = sample\_df[sample\_df['ml\_flag']]  
 plt.scatter(ml\_sub['x\_jitter'], ml\_sub['salary\_num'], facecolors='none', edgecolors='k', s=45, linewidths=0.6,label='ML/DS Job Postings')  
  
 # Use linear scale so we can show dollar ticks every $25,000 as requested  
 plt.yscale('linear')  
 plt.xlabel('KMeans cluster (jittered)')  
 plt.ylabel('Salary (USD)')  
 plt.title('KMeans clusters: Salary by NAICS group; ML jobs outlined')  
 # Label x-axis clusters explicitly: 1 = ML jobs, 0 = Non-ML jobs  
 cluster\_order = sorted(plot\_df['cluster'].unique())  
 # map 1->ML cluster label  
 xtick\_labels = []  
 for c in cluster\_order:  
 if 'cluster\_label' in plot\_df.columns:  
 # use cluster\_label if present  
 lab = 'Jobs Requiring ML/DS Skills' if (plot\_df['cluster\_label'].iloc[0] == 'ML cluster' and c==1) or (plot\_df['cluster\_label'].unique().size==2 and plot\_df.loc[plot\_df['cluster']==c,'cluster\_label'].iloc[0]=='ML cluster') else 'Jobs Not Requiring ML/DS Skills'  
 else:  
 lab = 'Jobs Requiring ML/DS Skills' if c==1 else 'Jobs Not Requiring ML/DS Skills'  
 xtick\_labels.append(lab)  
  
 plt.xticks(cluster\_order, xtick\_labels, rotation=15)  
  
 # set major y-ticks every $25,000 from min to max salary (rounded)  
 ymin = math.floor(sample\_df['salary\_num'].min() / 25000) \* 25000  
 ymax = math.ceil(sample\_df['salary\_num'].max() / 25000) \* 25000  
 y\_ticks = list(range(int(ymin), int(ymax)+1, 25000))  
 plt.gca().set\_yticks(y\_ticks)  
  
 def usd(x, pos):  
 return f"${x:,.0f}"  
  
 plt.gca().yaxis.set\_major\_formatter(FuncFormatter(usd))  
 # keep legend for NAICS groups only, not the ML outline  
 plt.legend(title='NAICS (top)', bbox\_to\_anchor=(1.02,1), loc='upper left')  
 plt.tight\_layout()  
 out\_png = 'figures/kmeans\_salary\_naics\_scatter.png'  
 plt.savefig(out\_png, dpi=220)  
 print('Saved static KMeans NAICS scatter to', out\_png)

Saved static KMeans NAICS scatter to figures/kmeans\_salary\_naics\_scatter.png

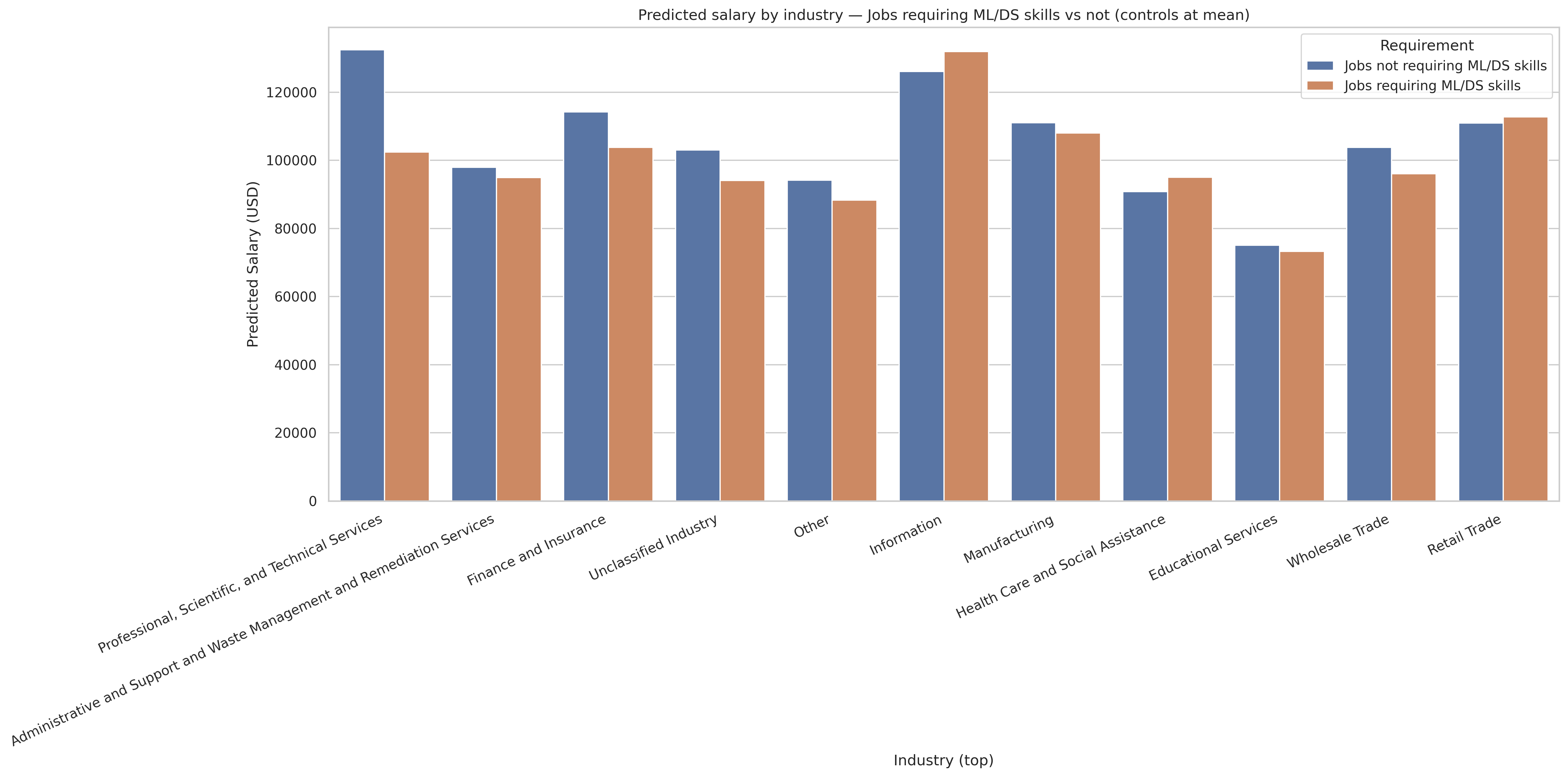


# Analysis

The KMeans clustering was run comparing positions that specifically required machine learning or data science skills against those that did not, with identifiers for the top ten industry segments and all others grouped as “Other.” If all positions had been plotted together, it appears there would have been a natural split around the $225,000 salary mark, as postings above that threshold become increasingly sparse. Looking at the two clusters side by side, it’s clear that while both include outliers, roles requiring machine learning or data science skills have not only a higher concentration of positions above the $225,000 mark but also significantly more extreme salary outliers.

# --- Multiple linear regression: log(salary) on ML-job indicator, industry fixed effects, and controls ---  
import statsmodels.formula.api as smf  
import statsmodels.api as sm  
  
# prefer df\_model if available (created during clustering); otherwise fall back to plot\_df  
source\_df = None  
if 'df\_model' in globals():  
 source\_df = df\_model.copy()  
elif 'plot\_df' in globals():  
 source\_df = plot\_df.copy()  
else:  
 source\_df = df.copy()  
  
# ensure salary numeric present  
if source\_df is None or source\_df.empty or 'salary\_num' not in source\_df.columns:  
 print('No usable salary data available for regression.')  
else:  
 df\_reg = source\_df[source\_df['salary\_num'].notna() & (source\_df['salary\_num']>0)].copy()  
 if df\_reg.empty:  
 print('No salary rows after filtering for regression.')  
 else:  
 # detect NAICS/industry column robustly (same approach as plotting)  
 import re as \_re  
 cols = list(df\_reg.columns)  
 naics\_like = [c for c in cols if \_re.search(r'naics|industry', c, \_re.I)]  
 naics\_name\_cols = [c for c in naics\_like if \_re.search(r'name|title|desc|sector', c, \_re.I)]  
 naics\_code\_cols = [c for c in naics\_like if \_re.search(r'code|id|num|^naics$', c, \_re.I)]  
  
 if naics\_name\_cols:  
 naics\_col = naics\_name\_cols[0]  
 df\_reg['naics\_group'] = df\_reg[naics\_col].fillna('Unknown').astype(str)  
 elif naics\_like and naics\_code\_cols:  
 code\_col = naics\_code\_cols[0]  
 df\_reg['naics\_group'] = df\_reg[code\_col].fillna('Unknown').astype(str).apply(lambda v: f"NAICS {v}" if str(v).strip()!='' else 'Unknown')  
 elif naics\_like:  
 naics\_col = naics\_like[0]  
 df\_reg['naics\_group'] = df\_reg[naics\_col].fillna('Unknown').astype(str)  
 else:  
 df\_reg['naics\_group'] = 'Unknown'  
  
 # build top-industry grouping similar to plotting  
 top\_naics = df\_reg['naics\_group'].value\_counts().nlargest(10).index.tolist()  
 df\_reg['naics\_top'] = df\_reg['naics\_group'].where(df\_reg['naics\_group'].isin(top\_naics), 'Other')  
  
 # create binary indicator: whether the posting requires ML/DS skills  
 # keep `is\_ml\_job` for backwards compatibility with the formula  
 df\_reg['requires\_ml\_skill'] = df\_reg.get('ml\_skill\_count', 0) > 0  
 df\_reg['is\_ml\_job'] = df\_reg['requires\_ml\_skill']  
  
 df\_reg['log\_salary'] = np.log(df\_reg['salary\_num'].astype(float))  
  
 # add simple controls if missing  
 if 'n\_skills' not in df\_reg.columns:  
 df\_reg['n\_skills'] = df\_reg.get('all\_skill\_tokens', []).apply(lambda x: len(x) if isinstance(x, (list, tuple)) else 0)  
 if 'ml\_skill\_count' not in df\_reg.columns:  
 df\_reg['ml\_skill\_count'] = 0  
  
 # Fit OLS with industry fixed effects and interaction: log\_salary ~ is\_ml\_job \* C(naics\_top) + n\_skills + ml\_skill\_count  
 formula = 'log\_salary ~ is\_ml\_job \* C(naics\_top) + n\_skills + ml\_skill\_count'  
 try:  
 model = smf.ols(formula=formula, data=df\_reg).fit(cov\_type='HC3')  
 print(model.summary())  
 # save summary to file  
 with open('figures/ols\_summary.txt', 'w') as f:  
 f.write(model.summary().as\_text())  
 except Exception as e:  
 print('Regression failed:', e)  
 model = None  
  
 # If model fit, compute predicted salaries for ML vs non-ML by industry  
 if model is not None:  
 preds = []  
 # use mean of numeric controls for prediction  
 mean\_n\_skills = df\_reg['n\_skills'].mean() if 'n\_skills' in df\_reg.columns else 0  
 mean\_ml\_skill\_count = df\_reg['ml\_skill\_count'].mean() if 'ml\_skill\_count' in df\_reg.columns else 0  
  
 for ind in df\_reg['naics\_top'].unique():  
 for ml\_flag in [0,1]:  
 # model expects `is\_ml\_job` variable (we set it above), so pass that for prediction  
 row = { 'is\_ml\_job': ml\_flag, 'n\_skills': mean\_n\_skills, 'ml\_skill\_count': mean\_ml\_skill\_count, 'naics\_top': ind }  
 try:  
 pred\_res = model.get\_prediction(pd.DataFrame([row]))  
 pred\_mean\_log = pred\_res.predicted\_mean[0]  
 pred\_mean = float(np.exp(pred\_mean\_log))  
 except Exception:  
 pred\_mean = np.nan  
 preds.append({'naics\_top': ind, 'is\_ml\_job': bool(ml\_flag), 'predicted\_salary': pred\_mean})  
  
 pred\_df = pd.DataFrame(preds).dropna()  
 # sort industries by total postings to keep consistent order  
 order = df\_reg['naics\_top'].value\_counts().loc[lambda x: x.index.isin(pred\_df['naics\_top'])].index.tolist()  
 pred\_df['naics\_top'] = pd.Categorical(pred\_df['naics\_top'], categories=order, ordered=True)  
  
 # Map boolean to readable labels and plot side-by-side bars for predicted salary by industry  
 pred\_df['requires\_label'] = pred\_df['is\_ml\_job'].map({True: 'Jobs requiring ML/DS skills', False: 'Jobs not requiring ML/DS skills'})  
 plt.figure(figsize=(18,9),dpi =300)  
 sns.barplot(data=pred\_df, x='naics\_top', y='predicted\_salary', hue='requires\_label')  
 plt.xlabel('Industry (top)')  
 plt.ylabel('Predicted Salary (USD)')  
 plt.title('Predicted salary by industry — Jobs requiring ML/DS skills vs not (controls at mean)')  
 plt.xticks(rotation=25, ha='right')  
 plt.legend(title='Requirement')  
 plt.tight\_layout()  
 out\_png2 = 'figures/regression\_predicted\_salary\_by\_industry.png'  
 plt.savefig(out\_png2, dpi=300, bbox\_inches="tight")  
 print('Saved regression predicted-salary figure to', out\_png2)

OLS Regression Results   
==============================================================================  
Dep. Variable: log\_salary R-squared: 0.191  
Model: OLS Adj. R-squared: 0.191  
Method: Least Squares F-statistic: 309.8  
Date: Fri, 10 Oct 2025 Prob (F-statistic): 0.00  
Time: 02:41:27 Log-Likelihood: -12854.  
No. Observations: 30808 AIC: 2.576e+04  
Df Residuals: 30784 BIC: 2.596e+04  
Df Model: 23   
Covariance Type: HC3   
======================================================================================================================================================  
 coef std err z P>|z| [0.025 0.975]  
------------------------------------------------------------------------------------------------------------------------------------------------------  
Intercept 11.3067 0.009 1294.039 0.000 11.290 11.324  
is\_ml\_job[T.True] -0.0311 0.015 -2.017 0.044 -0.061 -0.001  
C(naics\_top)[T.Educational Services] -0.2659 0.018 -14.947 0.000 -0.301 -0.231  
C(naics\_top)[T.Finance and Insurance] 0.1542 0.010 14.728 0.000 0.134 0.175  
C(naics\_top)[T.Health Care and Social Assistance] -0.0751 0.019 -3.963 0.000 -0.112 -0.038  
C(naics\_top)[T.Information] 0.2528 0.012 20.459 0.000 0.229 0.277  
C(naics\_top)[T.Manufacturing] 0.1256 0.013 9.645 0.000 0.100 0.151  
C(naics\_top)[T.Other] -0.0396 0.013 -3.073 0.002 -0.065 -0.014  
C(naics\_top)[T.Professional, Scientific, and Technical Services] 0.3019 0.009 35.075 0.000 0.285 0.319  
C(naics\_top)[T.Retail Trade] 0.1250 0.022 5.597 0.000 0.081 0.169  
C(naics\_top)[T.Unclassified Industry] 0.0511 0.012 4.428 0.000 0.028 0.074  
C(naics\_top)[T.Wholesale Trade] 0.0582 0.014 4.186 0.000 0.031 0.085  
is\_ml\_job[T.True]:C(naics\_top)[T.Educational Services] 0.0075 0.026 0.290 0.772 -0.043 0.058  
is\_ml\_job[T.True]:C(naics\_top)[T.Finance and Insurance] -0.0645 0.019 -3.468 0.001 -0.101 -0.028  
is\_ml\_job[T.True]:C(naics\_top)[T.Health Care and Social Assistance] 0.0758 0.025 3.013 0.003 0.026 0.125  
is\_ml\_job[T.True]:C(naics\_top)[T.Information] 0.0766 0.023 3.380 0.001 0.032 0.121  
is\_ml\_job[T.True]:C(naics\_top)[T.Manufacturing] 0.0032 0.024 0.135 0.893 -0.043 0.050  
is\_ml\_job[T.True]:C(naics\_top)[T.Other] -0.0328 0.020 -1.618 0.106 -0.072 0.007  
is\_ml\_job[T.True]:C(naics\_top)[T.Professional, Scientific, and Technical Services] -0.2256 0.018 -12.484 0.000 -0.261 -0.190  
is\_ml\_job[T.True]:C(naics\_top)[T.Retail Trade] 0.0471 0.033 1.421 0.155 -0.018 0.112  
is\_ml\_job[T.True]:C(naics\_top)[T.Unclassified Industry] -0.0603 0.021 -2.822 0.005 -0.102 -0.018  
is\_ml\_job[T.True]:C(naics\_top)[T.Wholesale Trade] -0.0459 0.034 -1.343 0.179 -0.113 0.021  
n\_skills 0.0036 7.65e-05 46.942 0.000 0.003 0.004  
ml\_skill\_count -0.0044 0.001 -3.682 0.000 -0.007 -0.002  
==============================================================================  
Omnibus: 561.884 Durbin-Watson: 1.929  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 619.562  
Skew: -0.304 Prob(JB): 2.91e-135  
Kurtosis: 3.336 Cond. No. 1.62e+03  
==============================================================================  
  
Notes:  
[1] Standard Errors are heteroscedasticity robust (HC3)  
[2] The condition number is large, 1.62e+03. This might indicate that there are  
strong multicollinearity or other numerical problems.  
Saved regression predicted-salary figure to figures/regression\_predicted\_salary\_by\_industry.png



## Conclusion

Both models highlight that while ML/DS skills correlate with higher earning potential in specific industries—particularly Information and Professional & Technical Services—the overall job-market premium is not universal. This suggests that data-driven expertise remains concentrated in certain sectors, offering a clear signal for students to prioritize industry context when developing advanced analytical skills.

# Analysis

Based on the KMeans clustering, we expected to see higher salaries across industries for roles requiring machine learning and data science skills. However, the regression results showed that only Information and Retail Trade—along with “Other”—had higher salaries for jobs requiring those skills. This suggests that while job postings requiring ML/DS skills tend to include higher salary outliers, the overall demand for these skills may still be relatively niche. In many industries, organizations may not yet know how to fully incorporate machine learning and data science into their operations, meaning that the highest-paying roles still emphasize more traditional skill sets. It’s likely that performing this same analysis five or ten years from now would yield very different results as these skills become more widely adopted and in demand across currently slower-to-adapt industries.

title: “Career Plan” subtitle: “Analyzing Salary and Industry Trends for ML/Data Science Roles” author: - name: “Group 12” date: “2025-09-27” toc: true bibliography: references.bib csl: csl/econometrica.csl format: html: default

## Career Plan

As students preparing to enter the analytics and data science job market, we want to align our coursework with employer demands.

### Yibei’s Plan

I am particularly interested in a career in data analysis or fixed income research. Through this project, I have enhanced my skills in data source usage, reference management, and Quarto—skills highly sought after by employers. My three goals are: to obtain a data analyst internship in spring 2026, to learn Python, SQL, R, and Excel modeling, and to understand industry trends in AI/ML applications.

### Fuhan’s Plan

I hope to develop a career path in the field of risk management or actuarial analysis. My undergraduate background in actuarial science has equipped me with a solid mathematical foundation and financial mathematical literacy. I have a certain theoretical foundation in probability models, life and property insurance pricing, and financial risk control. My short-term goal is to obtain a risk management or actuarial internship opportunity in the spring of 2026, preferably related to insurance companies, reinsurance companies, or financial institutions.

### Jonathan’s Plan

## I have spent my 13+ year career in property insurance, where I have seen how outdated and manual many data processes remain. While the industry is now embracing digital transformation and AI, gaps still exist between development teams and business users. My goal is to build the technical and analytical skills needed to bridge this divide, ensuring tools deliver real business value. To do this, I plan to strengthen my programming (Python, SQL), cloud analytics, and data visualization skills while deepening my business analytics expertise.

## Overall Insights and Conclusion

This project has given us a clearer understanding of how data skills impact job opportunities and salaries in today’s market. Our analysis revealed some useful insights, including strong demand for data-related positions in industries such as information technology, consulting, and finance. Jobs requiring machine learning or data science skills generally command higher salaries, which explains the growing value of technical and analytical skills.

At the same time, we’ve learned that salary advantages aren’t equal across industries. Some sectors are still in the early stages of digital transformation, which means opportunities are likely to increase in the future. For student job seekers like us, developing data, programming, and analytical thinking skills can provide a clear advantage, especially when combined with strong communication and teamwork skills. Overall, this project helped us connect data analysis with practical career plans and better understand which skills will be most critical for employability in 2024 and beyond.

Badhon, M. B., H. M. Hasan, M. N. U. Islam, N. Jaly, S. A. Sumon, and R. Ullah. (2024): “Enhancing productivity through business analytics and human capital,” *International Journal for Multidisciplinary Research*, 6, 1–11.

Li, D., E. Milonas, and Q. Zhang. (2024): “Preparing Undergraduate Data Scientists for Success in the Workplace: Aligning Competencies with Job Requirements,” *2024 ASEE annual conference & exposition*,.

Modak, S., P. Shinde, A. Tiwari, and S. Nalamwar. (2024): “A review of resume analysis and job description matching using machine learning,” *International Journal on Recent and Innovation Trends in Computing and Communication*, 12, 247–50.

Timpone, R., and Y. Yang. (2025): “AI, humans, and data science: Optimizing roles across workflows and the workforce,” *arXiv preprint arXiv:2507.11597*,.

Yuan, W., and L. Bao. “Use big data analysis methods to conduct in-depth research on future job market skills requirements,”