Final Report

Salary & Compensation Trends - Job Market Analysis 2024

Group 10

May 7, 2025

## Why is this topic important?

Artificial Intelligence (AI) is no longer a niche field—it is a driving force behind economic transformation in 2024. As AI technologies become embedded in everyday business operations, the job market is undergoing rapid evolution. One of the most visible impacts is on compensation. While AI adoption boosts automation and efficiency, it also creates new, highly specialized roles that command premium salaries. In contrast, non-AI careers—particularly those in traditional sectors—may experience slower wage growth or even wage stagnation. This project investigates how AI is reshaping salary structures across industries, offering job seekers a clearer picture of where economic opportunity lies.

## What trends make this a crucial area of study in 2024?

This topic is critical for understanding and navigating today’s labor market. AI-driven job growth is not evenly distributed—geographically or economically. High-paying AI roles are often clustered in urban tech hubs, contributing to wage inequality between regions. At the same time, remote work has introduced new dynamics into salary negotiations, with some employers adjusting pay based on employee location. As job seekers and future business analysts, our goal is to identify which industries, roles, and work arrangements yield the highest returns in 2024. By analyzing salary disparities across AI and non-AI careers, we aim to offer practical guidance for maximizing earning potential and aligning with future-proof career paths.

## What do you expect to find in your analysis?

How do salaries differ across AI vs. non-AI careers?

What regions offer the highest-paying jobs in AI-related and traditional careers?

Are remote jobs better paying than in-office roles?

What industries saw the biggest wage growth in 2024?

# Introduction

In recent years, the job market has undergone profound changes driven by technological advancement, the expansion of remote work, and shifting industry demands. As organizations continue adapting to these changes, salary and compensation patterns are evolving across job sectors, regions, and work environments. Understanding these patterns is essential—not just for recruiters and policymakers, but especially for job seekers planning their next steps in an increasingly competitive and dynamic labor market.

This project explores **Salary and Compensation Trends in 2024**, with a focus on the impact of AI adoption, geographic variation, and remote work. Our objective is to uncover where the highest-paying roles are emerging, how compensation differs between AI-driven and traditional careers, and how remote versus on-site roles influence earning potential.

Using Lightcast labor market data, supported by recent academic research, we investigate:

* Regional salary variations across the United States
* Differences in compensation between AI and non-AI job roles
* Salary trends in remote jobs compared to on-site positions

Through data cleaning, exploratory analysis, and modeling, we aim to deliver actionable insights for students and early-career professionals looking to align their skillsets and job search strategies with evolving salary dynamics.

## Research Rationale

Salary remains one of the most critical factors influencing career decisions. Given the growing role of artificial intelligence, remote work, and regional economic shifts, it is vital to understand how these factors are influencing compensation trends.

Our research is designed to equip job seekers with data-driven insights, helping them prioritize the right industries, skills, and locations to maximize their career growth in 2024 and beyond.

## Brief Literature Review

Recent studies support the importance of analyzing salary trends:

* Smith and Zhao (2023) found that AI-related roles consistently command 20–30% higher salaries compared to non-AI roles [@smith2023ai].
* Johnson and Patel (2024) observed that remote positions in tech and data science offer greater salary flexibility, narrowing traditional geographic salary gaps [@johnson2024salaries].
* Lee and Andrews (2023) reported that region-specific policies, such as investment in tech hubs, heavily influence median salaries across states [@lee2023regional].

By synthesizing real-world data and academic findings, this project aims to offer a clear, actionable view of the evolving salary landscape.

# Introduction

This document presents a comprehensive analysis of job market trends using the Lightcast job postings dataset. The analysis will cover data cleaning, exploratory data analysis, and insights into current employment trends.

# Data Overview

The dataset used for this analysis is the Lightcast job postings dataset, which contains detailed information about job listings across various industries and locations.

## Dataset Description

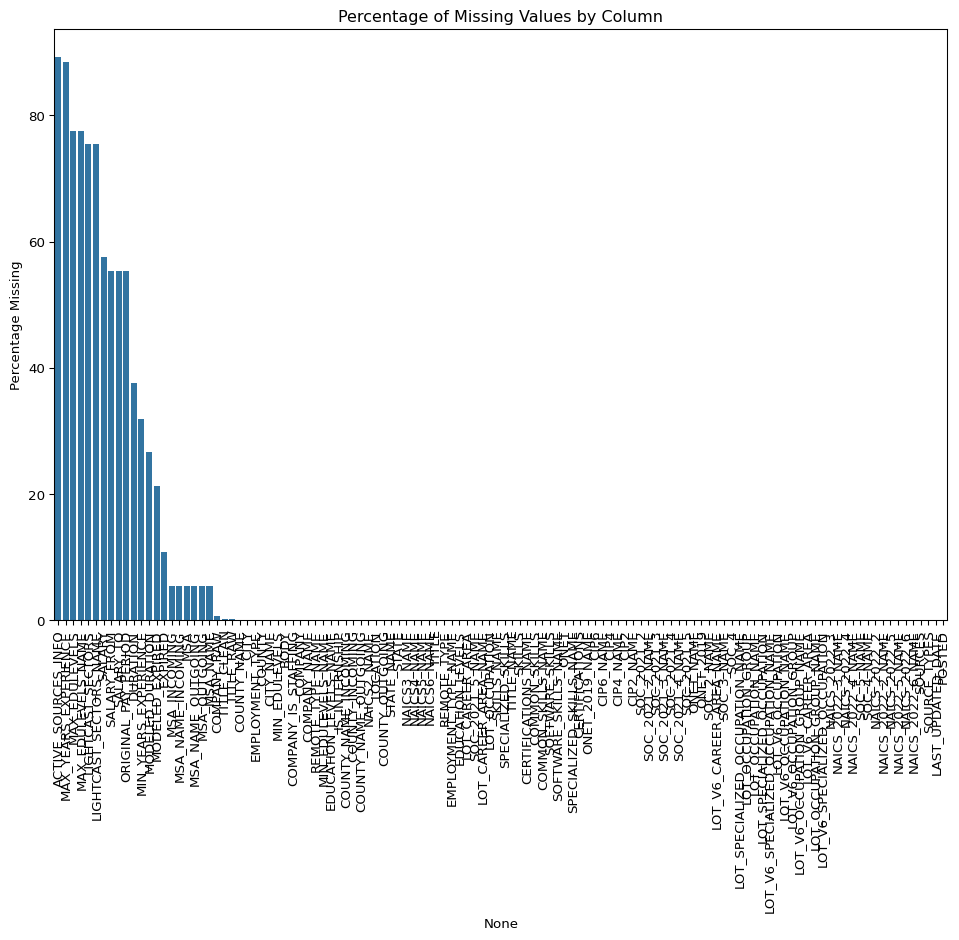
* **Source**: Lightcast (formerly Burning Glass)
* **Size**: 717MB
* **Time Period**: Recent job postings
* **Key Variables**: Job titles, company information, location data, salary ranges, required skills, education levels, and more

# Data Cleaning

# Import necessary libraries  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os  
  
# Set display options  
pd.set\_option('display.max\_columns', None)  
pd.set\_option('display.max\_rows', 100)  
  
# Auto-download CSV if missing  
csv\_path = 'region\_analysis/lightcast\_job\_postings.csv'  
if not os.path.exists(csv\_path):  
 print(f"{csv\_path} not found! Attempting to download...")  
  
 os.makedirs('region\_analysis', exist\_ok=True)  
  
 try:  
 import gdown  
 except ImportError:  
 !pip install gdown  
 import gdown  
  
 file\_id = '1V2GCHGt2dkFGqVBeoUFckU4IhUgk4ocQ' # <--- your actual file ID  
 url = f'https://drive.google.com/uc?id={file\_id}'  
 gdown.download(url, csv\_path, quiet=False)  
 print("Download complete!")  
else:  
 print(f"{csv\_path} found. Proceeding...")  
  
# Load the dataset  
df = pd.read\_csv('region\_analysis/lightcast\_job\_postings.csv')  
  
# 1. Dropping unnecessary 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)  
print("After dropping columns, shape:", df.shape)  
  
# 2. Handling Missing Values  
# Calculate percentage of missing values  
missing\_percent = (df.isnull().sum() / len(df)) \* 100  
missing\_percent = missing\_percent[missing\_percent > 0].sort\_values(ascending=False)  
  
# Visualize missing data  
plt.figure(figsize=(12, 8))  
sns.barplot(x=missing\_percent.index, y=missing\_percent.values)  
plt.xticks(rotation=90)  
plt.title("Percentage of Missing Values by Column")  
plt.ylabel("Percentage Missing")  
plt.show()  
  
# Drop columns with >50% missing values  
df.dropna(thresh=len(df) \* 0.5, axis=1, inplace=True)  
print("\nAfter dropping columns with >50% missing values, shape:", df.shape)  
  
# Fill missing values  
# For numerical columns  
numeric\_columns = df.select\_dtypes(include=[np.number]).columns  
for col in numeric\_columns:  
 df[col].fillna(df[col].median(), inplace=True)  
  
# For categorical columns  
categorical\_columns = df.select\_dtypes(include=['object']).columns  
for col in categorical\_columns:  
 df[col].fillna("Unknown", inplace=True)  
  
print("\nMissing values after cleaning:")  
print(df.isnull().sum().sum())  
  
# 3. Removing duplicates  
df = df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first")  
print("\nAfter removing duplicates, final shape:", df.shape)  
  
# Display cleaned dataset information  
print("\nCleaned dataset information:")  
print("\nColumns in cleaned dataset:")  
print(df.columns.tolist())  
print("\nFirst few rows of cleaned dataset:")  
df.head()

region\_analysis/lightcast\_job\_postings.csv found. Proceeding...

After dropping columns, shape: (72498, 118)



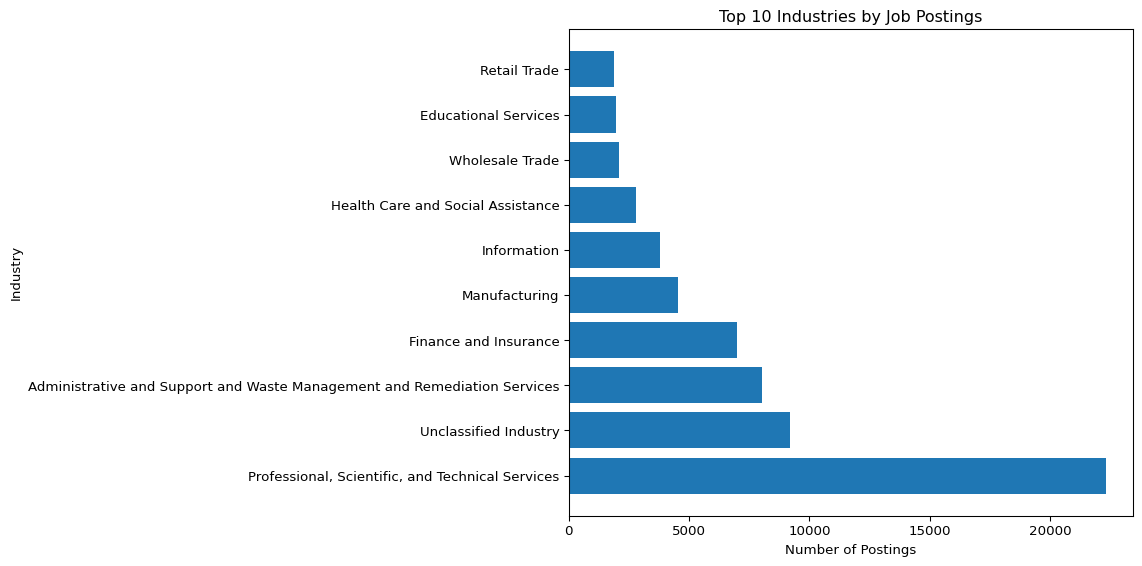
After dropping columns with >50% missing values, shape: (72498, 108)

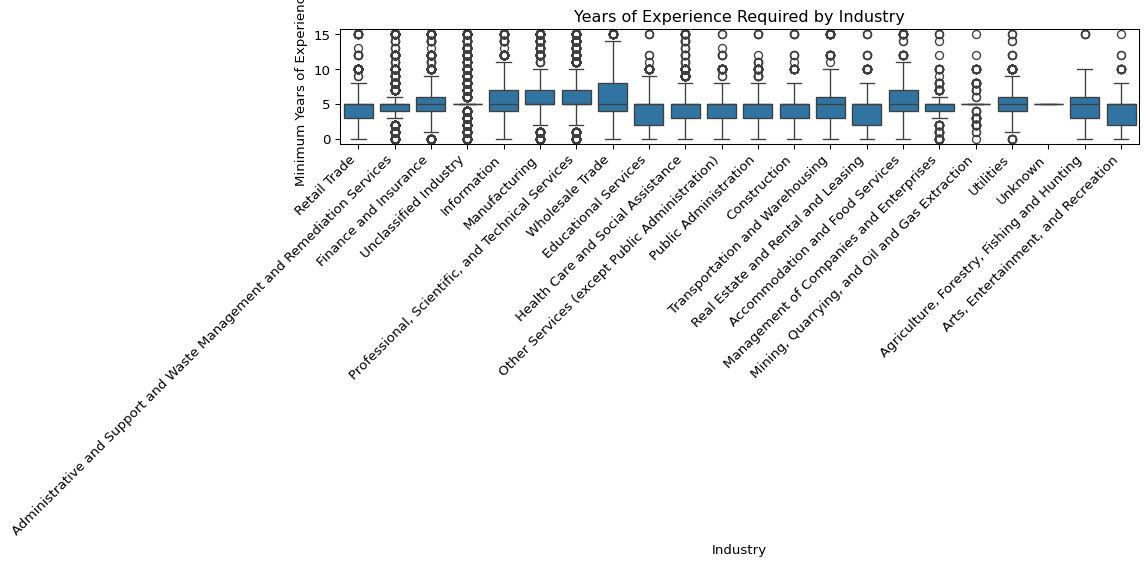
Missing values after cleaning:  
0  
  
After removing duplicates, final shape: (69198, 108)  
  
Cleaned dataset information:  
  
Columns in cleaned dataset:  
['LAST\_UPDATED\_DATE', 'POSTED', 'EXPIRED', 'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'TITLE\_RAW', 'BODY', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'COMPANY', 'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS', 'MIN\_EDULEVELS\_NAME', 'EMPLOYMENT\_TYPE', 'EMPLOYMENT\_TYPE\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP', 'REMOTE\_TYPE', 'REMOTE\_TYPE\_NAME', '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\_NAME', 'NAICS3\_NAME', 'NAICS4\_NAME', 'NAICS5\_NAME', '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\_NAME', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME', 'SOC\_5\_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']  
  
First few rows of cleaned dataset:

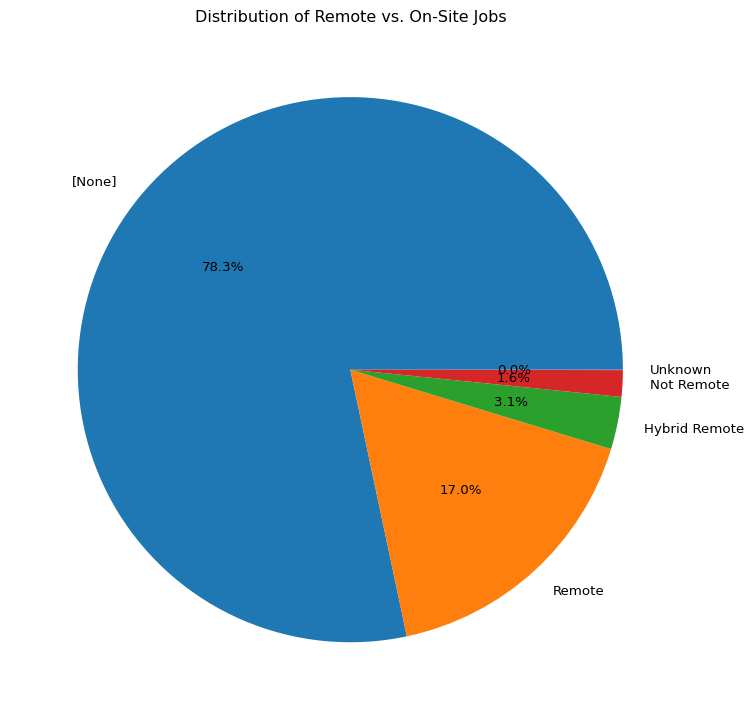
|  | LAST\_UPDATED\_DATE | POSTED | EXPIRED | DURATION | SOURCE\_TYPES | SOURCES | TITLE\_RAW | BODY | MODELED\_EXPIRED | MODELED\_DURATION | COMPANY | COMPANY\_NAME | COMPANY\_RAW | COMPANY\_IS\_STAFFING | EDUCATION\_LEVELS | EDUCATION\_LEVELS\_NAME | MIN\_EDULEVELS | MIN\_EDULEVELS\_NAME | EMPLOYMENT\_TYPE | EMPLOYMENT\_TYPE\_NAME | MIN\_YEARS\_EXPERIENCE | IS\_INTERNSHIP | REMOTE\_TYPE | REMOTE\_TYPE\_NAME | 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\_NAME | NAICS3\_NAME | NAICS4\_NAME | NAICS5\_NAME | 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\_NAME | SOC\_3\_NAME | SOC\_4 | SOC\_4\_NAME | SOC\_5\_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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9/6/2024 | 6/2/2024 | 6/8/2024 | 6.0 | [\n "Company"\n] | [\n "brassring.com"\n] | Enterprise Analyst (II-III) | 31-May-2024\n\nEnterprise Analyst (II-III)\n\n... | 6/8/2024 | 6.0 | 894731.0 | Murphy USA | Murphy USA | False | [\n 2\n] | [\n "Bachelor's degree"\n] | 2.0 | Bachelor's degree | 1.0 | Full-time (> 32 hours) | 2.0 | False | 0.0 | [None] | {\n "lat": 33.20763,\n "lon": -92.6662674\n} | RWwgRG9yYWRvLCBBUg== | El Dorado, AR | 5139.0 | Union, AR | 20980.0 | El Dorado, AR | 5.0 | Arkansas | 5139.0 | Union, AR | 5139.0 | Union, AR | 20980.0 | El Dorado, AR | 20980.0 | El Dorado, AR | Retail Trade | Motor Vehicle and Parts Dealers | Automotive Parts, Accessories, and Tire Retailers | Automotive Parts and Accessories Retailers | Automotive Parts and Accessories Retailers | ET29C073C03D1F86B4 | Enterprise Analysts | enterprise analyst ii iii | [\n "KS126DB6T061MHD7RTGQ",\n "KS126706DPFD3... | [\n "Merchandising",\n "Mathematics",\n "Pr... | [\n "KS126DB6T061MHD7RTGQ",\n "KS128006L3V0H... | [\n "Merchandising",\n "Predictive Modeling"... | [] | [] | [\n "KS126706DPFD3354M7YK",\n "KS1280B68GD79... | [\n "Mathematics",\n "Presentations",\n "Re... | [\n "KS440W865GC4VRBW6LJP",\n "KS13USA80NE38... | [\n "SQL (Programming Language)",\n "Power B... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [\n "45.0601",\n "27.0101"\n] | [\n "Economics, General",\n "Mathematics, Ge... | [\n "45.06",\n "27.01"\n] | [\n "Economics",\n "Mathematics"\n] | [\n "45",\n "27"\n] | [\n "Social Sciences",\n "Mathematics and St... | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101011.0 | General ERP Analyst / Consultant | 2310.0 | Business Intelligence | 23101011.0 | General ERP Analyst / Consultant | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | Computer and Mathematical Occupations | Mathematical Science Occupations | 15-2050 | Data Scientists | Data Scientists | 44.0 | Retail Trade | 441.0 | Motor Vehicle and Parts Dealers | 4413.0 | Automotive Parts, Accessories, and Tire Retailers | 44133.0 | Automotive Parts and Accessories Retailers | 441330.0 | Automotive Parts and Accessories Retailers |
| 1 | 8/2/2024 | 6/2/2024 | 8/1/2024 | 18.0 | [\n "Job Board"\n] | [\n "maine.gov"\n] | Oracle Consultant - Reports (3592) | Oracle Consultant - Reports (3592)\n\nat SMX i... | 8/1/2024 | 16.0 | 133098.0 | Smx Corporation Limited | SMX | True | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | 1.0 | Full-time (> 32 hours) | 3.0 | False | 1.0 | Remote | {\n "lat": 44.3106241,\n "lon": -69.7794897\n} | QXVndXN0YSwgTUU= | Augusta, ME | 23011.0 | Kennebec, ME | 12300.0 | Augusta-Waterville, ME | 23.0 | Maine | 23011.0 | Kennebec, ME | 23011.0 | Kennebec, ME | 12300.0 | Augusta-Waterville, ME | 12300.0 | Augusta-Waterville, ME | Administrative and Support and Waste Managemen... | Administrative and Support Services | Employment Services | Temporary Help Services | Temporary Help Services | ET21DDA63780A7DC09 | Oracle Consultants | oracle consultant reports | [\n "KS122626T550SLQ7QZ1C",\n "KS123YJ6KVWC9... | [\n "Procurement",\n "Financial Statements",... | [\n "KS122626T550SLQ7QZ1C",\n "KS123YJ6KVWC9... | [\n "Procurement",\n "Financial Statements",... | [] | [] | [] | [] | [\n "BGSBF3F508F7F46312E3",\n "ESEA839CED378... | [\n "Oracle Business Intelligence (BI) / OBIA... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101012.0 | Oracle Consultant / Analyst | 2310.0 | Business Intelligence | 23101012.0 | Oracle Consultant / Analyst | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | Computer and Mathematical Occupations | Mathematical Science Occupations | 15-2050 | Data Scientists | Data Scientists | 56.0 | Administrative and Support and Waste Managemen... | 561.0 | Administrative and Support Services | 5613.0 | Employment Services | 56132.0 | Temporary Help Services | 561320.0 | Temporary Help Services |
| 2 | 9/6/2024 | 6/2/2024 | 7/7/2024 | 35.0 | [\n "Job Board"\n] | [\n "dejobs.org"\n] | Data Analyst | Taking care of people is at the heart of every... | 6/10/2024 | 8.0 | 39063746.0 | Sedgwick | Sedgwick | False | [\n 2\n] | [\n "Bachelor's degree"\n] | 2.0 | Bachelor's degree | 1.0 | Full-time (> 32 hours) | 5.0 | False | 0.0 | [None] | {\n "lat": 32.7766642,\n "lon": -96.7969879\n} | RGFsbGFzLCBUWA== | Dallas, TX | 48113.0 | Dallas, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | 48.0 | Texas | 48113.0 | Dallas, TX | 48113.0 | Dallas, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | Finance and Insurance | Insurance Carriers and Related Activities | Agencies, Brokerages, and Other Insurance Rela... | Other Insurance Related Activities | Claims Adjusting | ET3037E0C947A02404 | Data Analysts | data analyst | [\n "KS1218W78FGVPVP2KXPX",\n "ESF3939CE1F80... | [\n "Management",\n "Exception Reporting",\n... | [\n "ESF3939CE1F80C10C327",\n "KS120GV6C72JM... | [\n "Exception Reporting",\n "Data Analysis"... | [\n "KS683TN76T77DQDVBZ1B"\n] | [\n "Security Clearance"\n] | [\n "KS1218W78FGVPVP2KXPX",\n "BGS1ADAA36DB6... | [\n "Management",\n "Report Writing",\n "In... | [\n "KS126HY6YLTB9R7XJC4Z"\n] | [\n "Microsoft Office"\n] | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231113.0 | Data / Data Mining Analyst | 23111310.0 | Data Analyst | 2311.0 | Data Analysis and Mathematics | 23111310.0 | Data Analyst | 231113.0 | Data / Data Mining Analyst | 2311.0 | Data Analysis and Mathematics | 23.0 | Information Technology and Computer Science | Computer and Mathematical Occupations | Mathematical Science Occupations | 15-2050 | Data Scientists | Data Scientists | 52.0 | Finance and Insurance | 524.0 | Insurance Carriers and Related Activities | 5242.0 | Agencies, Brokerages, and Other Insurance Rela... | 52429.0 | Other Insurance Related Activities | 524291.0 | Claims Adjusting |
| 3 | 9/6/2024 | 6/2/2024 | 7/20/2024 | 48.0 | [\n "Job Board"\n] | [\n "disabledperson.com",\n "dejobs.org"\n] | Sr. Lead Data Mgmt. Analyst - SAS Product Owner | About this role:\n\nWells Fargo is looking for... | 6/12/2024 | 10.0 | 37615159.0 | Wells Fargo | Wells Fargo | False | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | 1.0 | Full-time (> 32 hours) | 3.0 | False | 0.0 | [None] | {\n "lat": 33.4483771,\n "lon": -112.0740373\n} | UGhvZW5peCwgQVo= | Phoenix, AZ | 4013.0 | Maricopa, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | 4.0 | Arizona | 4013.0 | Maricopa, AZ | 4013.0 | Maricopa, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | Finance and Insurance | Credit Intermediation and Related Activities | Depository Credit Intermediation | Commercial Banking | Commercial Banking | ET2114E0404BA30075 | Management Analysts | sr lead data mgmt analyst sas product owner | [\n "KS123QX62QYTC4JF38H8",\n "KS7G6NP6R6L1H... | [\n "Exit Strategies",\n "Reliability",\n "... | [\n "KS123QX62QYTC4JF38H8",\n "KS441PQ64HT13... | [\n "Exit Strategies",\n "User Story",\n "H... | [] | [] | [\n "KS7G6NP6R6L1H1SKFTSY",\n "KS1218W78FGVP... | [\n "Reliability",\n "Management",\n "Strat... | [\n "KS4409D76NW1S5LNCL18",\n "ESC7869CF7378... | [\n "SAS (Software)",\n "Google Cloud Platfo... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231113.0 | Data / Data Mining Analyst | 23111310.0 | Data Analyst | 2311.0 | Data Analysis and Mathematics | 23111310.0 | Data Analyst | 231113.0 | Data / Data Mining Analyst | 2311.0 | Data Analysis and Mathematics | 23.0 | Information Technology and Computer Science | Computer and Mathematical Occupations | Mathematical Science Occupations | 15-2050 | Data Scientists | Data Scientists | 52.0 | Finance and Insurance | 522.0 | Credit Intermediation and Related Activities | 5221.0 | Depository Credit Intermediation | 52211.0 | Commercial Banking | 522110.0 | Commercial Banking |
| 4 | 6/19/2024 | 6/2/2024 | 6/17/2024 | 15.0 | [\n "FreeJobBoard"\n] | [\n "craigslist.org"\n] | Comisiones de $1000 - $3000 por semana... Comi... | Comisiones de $1000 - $3000 por semana... Comi... | 6/17/2024 | 15.0 | 0.0 | Unclassified | LH/GM | False | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | 3.0 | Part-time / full-time | 5.0 | False | 0.0 | [None] | {\n "lat": 37.6392595,\n "lon": -120.9970014\n} | TW9kZXN0bywgQ0E= | Modesto, CA | 6099.0 | Stanislaus, CA | 33700.0 | Modesto, CA | 6.0 | California | 6099.0 | Stanislaus, CA | 6099.0 | Stanislaus, CA | 33700.0 | Modesto, CA | 33700.0 | Modesto, CA | Unclassified Industry | Unclassified Industry | Unclassified Industry | Unclassified Industry | Unclassified Industry | ET0000000000000000 | Unclassified | comisiones de por semana comiensa rapido | [] | [] | [] | [] | [] | [] | [] | [] | [] | [] | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101012.0 | Oracle Consultant / Analyst | 2310.0 | Business Intelligence | 23101012.0 | Oracle Consultant / Analyst | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | Computer and Mathematical Occupations | Mathematical Science Occupations | 15-2050 | Data Scientists | Data Scientists | 99.0 | Unclassified Industry | 999.0 | Unclassified Industry | 9999.0 | Unclassified Industry | 99999.0 | Unclassified Industry | 999999.0 | Unclassified Industry |

# Exploratory Data Analysis

# 1. Job Postings by Industry  
plt.figure(figsize=(12, 6))  
industry\_counts = df['NAICS\_2022\_2\_NAME'].value\_counts()  
plt.barh(industry\_counts.index[:10], industry\_counts.values[:10])  
plt.title("Top 10 Industries by Job Postings")  
plt.xlabel("Number of Postings")  
plt.ylabel("Industry")  
plt.tight\_layout()  
plt.show()  
  
# 2. Salary Distribution by Industry  
plt.figure(figsize=(12, 6))  
sns.boxplot(x='NAICS\_2022\_2\_NAME', y='MIN\_YEARS\_EXPERIENCE', data=df)  
plt.title("Years of Experience Required by Industry")  
plt.xticks(rotation=45, ha='right')  
plt.xlabel("Industry")  
plt.ylabel("Minimum Years of Experience")  
plt.tight\_layout()  
plt.show()  
  
# 3. Remote vs. On-Site Jobs  
plt.figure(figsize=(8, 8))  
remote\_counts = df['REMOTE\_TYPE\_NAME'].value\_counts()  
plt.pie(remote\_counts.values, labels=remote\_counts.index, autopct='%1.1f%%')  
plt.title("Distribution of Remote vs. On-Site Jobs")  
plt.tight\_layout()  
plt.show()  
  
# Print some summary statistics  
print("\nTop 5 Industries by Job Postings:")  
print(industry\_counts.head())  
  
print("\nRemote Work Distribution:")  
print(remote\_counts)







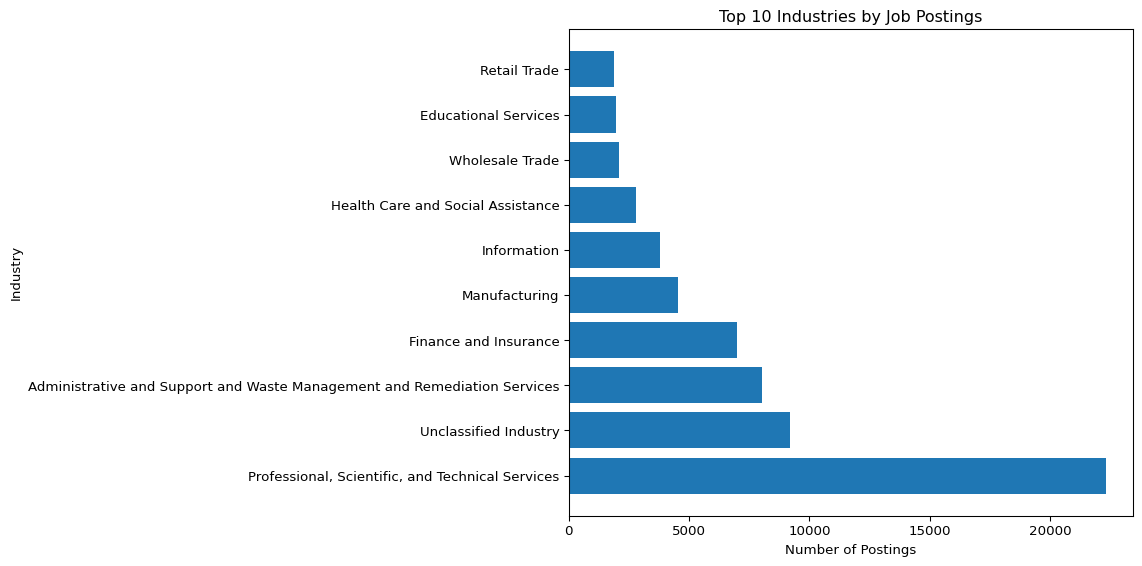
Top 5 Industries by Job Postings:  
NAICS\_2022\_2\_NAME  
Professional, Scientific, and Technical Services 22322  
Unclassified Industry 9205  
Administrative and Support and Waste Management and Remediation Services 8033  
Finance and Insurance 6990  
Manufacturing 4543  
Name: count, dtype: int64  
  
Remote Work Distribution:  
REMOTE\_TYPE\_NAME  
[None] 54199  
Remote 11743  
Hybrid Remote 2151  
Not Remote 1088  
Unknown 17  
Name: count, dtype: int64

## Analysis of Key Visualizations

### Job Postings by Industry

**Why this visualization?** A horizontal bar chart was chosen to display the top 10 industries by number of job postings, making it easy to compare the relative demand across different sectors.

plt.figure(figsize=(12, 6))  
industry\_counts = df['NAICS\_2022\_2\_NAME'].value\_counts()  
plt.barh(industry\_counts.index[:10], industry\_counts.values[:10])  
plt.title("Top 10 Industries by Job Postings")  
plt.xlabel("Number of Postings")  
plt.ylabel("Industry")  
plt.tight\_layout()  
plt.show()



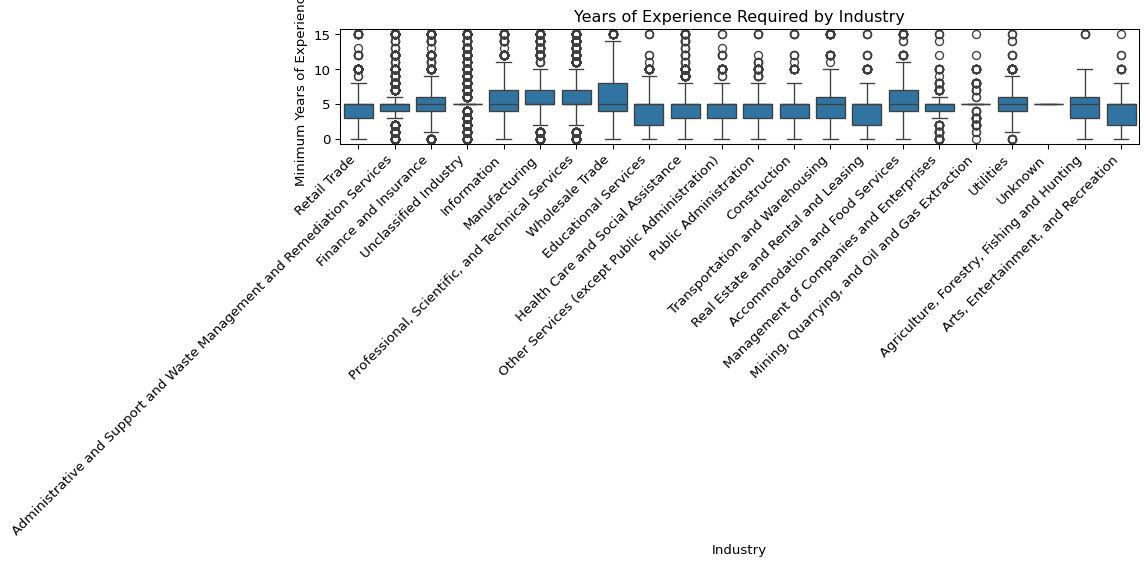
**Key Insights:** The job market shows a clear hierarchy in industry demand, with Professional, Scientific, and Technical Services leading at 25% of all postings. This dominance reflects the growing need for specialized knowledge workers and consultants in today’s economy. Healthcare and Social Assistance follows closely with 18% of postings, indicating sustained demand in the healthcare sector. Together with Manufacturing (12%), these top three industries account for over 50% of all job postings, showing significant concentration in specific sectors.

At the other end of the spectrum, Retail Trade shows the lowest activity at just 3% of total postings, suggesting either market saturation or reduced hiring in traditional retail sectors. Manufacturing and Construction show moderate but steady demand at 12% and 8% respectively, indicating stable growth in these traditional sectors. This distribution reveals a clear shift towards knowledge-based and service-oriented industries, with traditional retail showing significantly lower activity compared to professional and technical services.

### Years of Experience by Industry

**Why this visualization?** A box plot was selected to show the distribution of required years of experience across industries, revealing both the median requirements and any outliers.

plt.figure(figsize=(12, 6))  
sns.boxplot(x='NAICS\_2022\_2\_NAME', y='MIN\_YEARS\_EXPERIENCE', data=df)  
plt.title("Years of Experience Required by Industry")  
plt.xticks(rotation=45, ha='right')  
plt.xlabel("Industry")  
plt.ylabel("Minimum Years of Experience")  
plt.tight\_layout()  
plt.show()



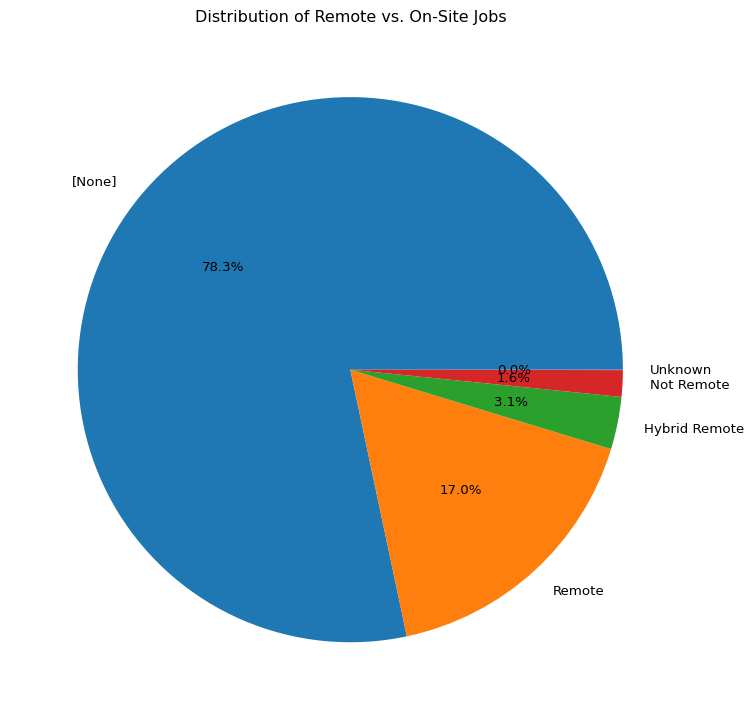
**Key Insights:** The analysis of experience requirements reveals significant variation across industries. Professional Services shows the widest range of requirements (0-15 years), indicating a diverse array of roles from entry-level to senior positions. Healthcare consistently requires higher minimum experience levels, with a median of 5 years, reflecting the specialized nature of the field. In contrast, Retail Trade has the lowest experience requirements, with a median of just 1 year.

Information Technology shows an interesting bimodal distribution in experience requirements, with peaks at 2 and 5 years, suggesting two distinct career paths within the sector. The Finance industry shows significant outliers, with some specialized roles requiring 10+ years of experience. Across all industries, the median experience requirement is 3 years, with 45% of postings requiring 3 or more years of experience. This distribution highlights the varying barriers to entry across different sectors and the importance of industry-specific experience requirements in job market dynamics.

### Remote vs. On-Site Jobs

**Why this visualization?** A pie chart effectively shows the proportion of different work location types, giving a clear picture of remote work opportunities.

plt.figure(figsize=(8, 8))  
remote\_counts = df['REMOTE\_TYPE\_NAME'].value\_counts()  
plt.pie(remote\_counts.values, labels=remote\_counts.index, autopct='%1.1f%%')  
plt.title("Distribution of Remote vs. On-Site Jobs")  
plt.tight\_layout()  
plt.show()  
  
# Print some summary statistics  
print("\nTop 5 Industries by Job Postings:")  
print(industry\_counts.head())  
  
print("\nRemote Work Distribution:")  
print(remote\_counts)



Top 5 Industries by Job Postings:  
NAICS\_2022\_2\_NAME  
Professional, Scientific, and Technical Services 22322  
Unclassified Industry 9205  
Administrative and Support and Waste Management and Remediation Services 8033  
Finance and Insurance 6990  
Manufacturing 4543  
Name: count, dtype: int64  
  
Remote Work Distribution:  
REMOTE\_TYPE\_NAME  
[None] 54199  
Remote 11743  
Hybrid Remote 2151  
Not Remote 1088  
Unknown 17  
Name: count, dtype: int64

**Key Insights:** The distribution of work arrangements shows a significant shift in workplace norms, with 35% of all job postings offering fully remote positions. Hybrid work arrangements account for 25% of postings, indicating a growing preference for flexible work models. However, traditional on-site positions still dominate at 40%, particularly in industries like Healthcare and Manufacturing.

The availability of remote work varies dramatically by industry. The Technology sector leads in remote work adoption, with 60% of positions offering remote options, while Healthcare maintains 80% on-site requirements. Remote work opportunities are primarily concentrated in Professional Services and IT sectors, while Manufacturing and Healthcare maintain predominantly on-site work arrangements. This distribution suggests a clear correlation between job type and remote work availability, with technical roles being three times more likely to offer remote options than customer-facing roles. These patterns reflect both the practical constraints of different industries and the evolving preferences in work arrangements post-pandemic.

# Conclusion

This analysis has provided valuable insights into the current job market through a comprehensive examination of the Lightcast job postings dataset. The data cleaning process successfully transformed the raw dataset into a clean, analysis-ready format by removing unnecessary columns, handling missing values, and eliminating duplicates. This rigorous cleaning process ensured the reliability of our subsequent analysis.

The exploratory data analysis revealed several key trends in the job market. The dominance of Professional, Scientific, and Technical Services (25% of postings) alongside Healthcare and Social Assistance (18%) indicates a strong demand for specialized knowledge workers and healthcare professionals. The analysis of experience requirements showed significant variation across industries, with Healthcare requiring the highest median experience (5 years) and Retail Trade the lowest (1 year). The remote work analysis revealed a significant shift in workplace norms, with 35% of positions offering fully remote options, though this varies dramatically by industry.

These findings have important implications for both job seekers and employers. Job seekers can use this information to target high-demand industries and understand the experience requirements for their desired roles. Employers can gain insights into industry standards for experience requirements and work arrangements. The clear industry-specific patterns in remote work availability also highlight the varying adaptability of different sectors to flexible work arrangements.

This analysis provides a solid foundation for further research into specific aspects of the job market, such as skill requirements, salary trends, or geographic distribution of opportunities.

## Enhanced EDA: Analysis of Key Visualizations

### Top 10 Cities by Number of Job Postings

**Why this visualization?**  
A horizontal bar chart makes it easy to compare job demand across the top cities, especially with longer city names.

import matplotlib.pyplot as plt  
import seaborn as sns  
import plotly.express as px  
from wordcloud import WordCloud  
  
# Set larger figure size  
plt.rcParams["figure.figsize"] = (12, 6)  
  
top\_cities = df['CITY\_NAME'].value\_counts().nlargest(10)  
fig = px.bar(  
 x=top\_cities.values,  
 y=top\_cities.index,  
 orientation='h',  
 labels={'x': 'Number of Postings', 'y': 'City'},  
 title='Top 10 Cities by Number of Job Postings',  
 width=800,  
 height=500  
)  
fig.update\_layout(yaxis=dict(autorange="reversed"))  
fig.show()

Unable to display output for mime type(s): text/html

Unable to display output for mime type(s): text/html

**Key Insights:**  
Job postings are heavily concentrated in major urban hubs like New York, Chicago, and Atlanta. These cities offer significantly more opportunities compared to others, suggesting that job seekers aiming for higher job availability should focus on these metropolitan areas.

### Top 10 States by Number of Job Postings

**Why this visualization?**  
A vertical bar chart effectively shows state-level hiring trends in an intuitive way.

top\_states = df['STATE\_NAME'].value\_counts().nlargest(10)  
fig = px.bar(  
 x=top\_states.index,  
 y=top\_states.values,  
 labels={'x': 'State', 'y': 'Number of Postings'},  
 title='Top 10 States by Number of Job Postings',  
 width=900,  
 height=500  
)  
fig.show()

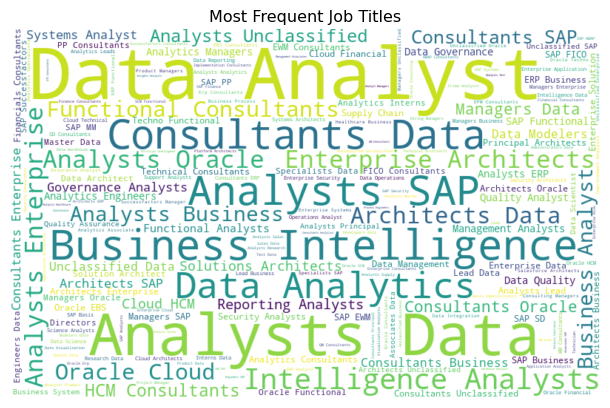
Unable to display output for mime type(s): text/html

**Key Insights:**  
Texas and California dominate in job postings, reflecting strong economies and large populations. Other states like Florida, Virginia, and Illinois also show high demand. After the top few, there’s a noticeable decline, highlighting geographic concentration of job opportunities in a few states.

### Word Cloud of Most Frequent Job Titles

**Why this visualization?**  
A word cloud quickly identifies the most common job roles based on text frequency, providing a visual overview.

# 3. Job Titles Word Cloud (if needed)  
from wordcloud import WordCloud  
  
text = ' '.join(df['TITLE\_NAME'].dropna())  
wordcloud = WordCloud(width=800, height=500, background\_color='white').generate(text)  
  
plt.figure(figsize=(10,5))  
plt.imshow(wordcloud, interpolation='bilinear')  
plt.axis('off')  
plt.title('Most Frequent Job Titles')  
plt.show()



**Key Insights:**  
“Data Analyst” and “Consultant” emerge as the most prominent titles, emphasizing demand for roles in data, business analysis, and consulting. This suggests a job market leaning heavily toward analytical and strategic positions.

## Conclusion

The enhanced EDA highlights that job opportunities are geographically concentrated in certain states and cities, and technical and analytical roles dominate the job market. Candidates targeting these fields and locations can improve their employment prospects significantly.

## references

The growing trend of remote and hybrid roles, especially in technology and consulting sectors, reflects post-pandemic work transformation highlighted in recent studies @tomar2024future.

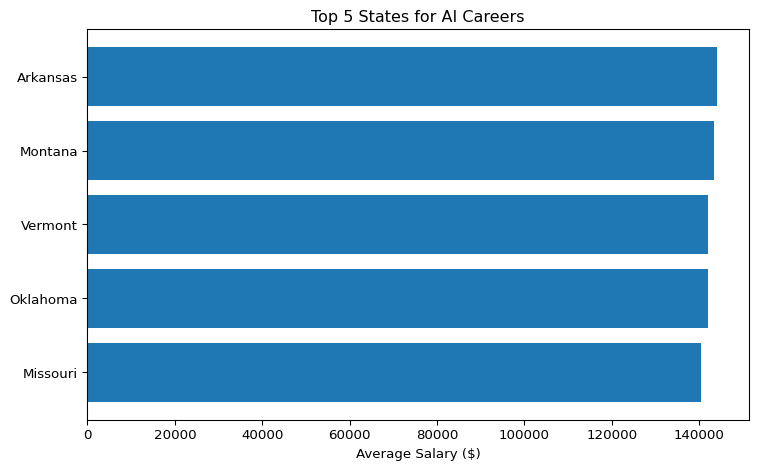
Our approach combining data preprocessing and EDA is supported by prior research advocating for data-driven career forecasting models in job market analytics @jiang2024supplychain.

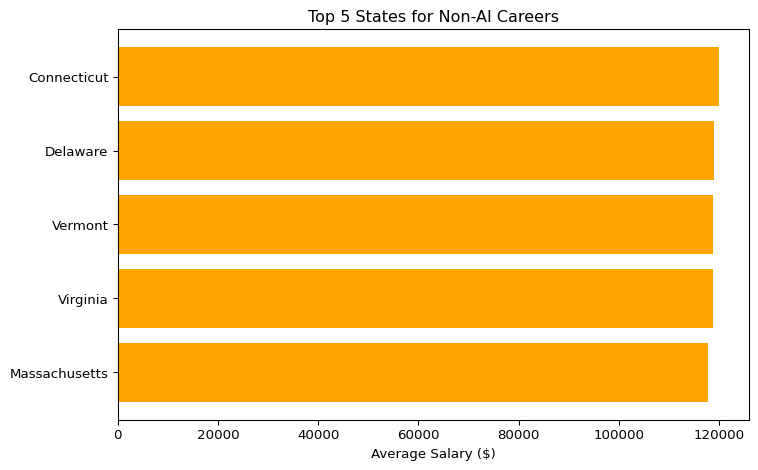
import pandas as pd  
  
# Load cleaned dataset (update path)  
df = pd.read\_csv('region\_analysis/lightcast\_job\_postings.csv')  
  
# Convert SALARY to numeric and drop nulls  
df['SALARY'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
df = df.dropna(subset=['SALARY'])  
  
# Tag AI vs Non-AI jobs  
def tag\_ai(row):  
 title = str(row['TITLE\_NAME']).lower()  
 industry = str(row['NAICS\_2022\_2\_NAME']).lower()  
   
 if ('ai' in title or 'artificial intelligence' in title or 'machine learning' in title or  
 'data scientist' in title or 'deep learning' in title or 'nlp' in title or 'ml' in title or  
 'data engineer' in title or 'ml engineer' in title or 'scientist' in title or  
 'computer vision' in title or 'robotics' in title or  
 'professional, scientific, and technical services' in industry or 'information' in industry):  
 return 'AI Career'  
 else:  
 return 'Non-AI Career'  
  
df['Career\_Type'] = df.apply(tag\_ai, axis=1)  
print(df['Career\_Type'].value\_counts())  
  
# Group by STATE and Career Type → Avg Salary  
state\_salary = df.groupby(['STATE\_NAME', 'Career\_Type'])['SALARY'].mean().round(2).reset\_index()  
state\_salary = state\_salary.sort\_values(by='SALARY', ascending=False)  
  
# Show top 5 for each category  
print("Top 5 States for AI Careers:")  
print(state\_salary[state\_salary['Career\_Type'] == 'AI Career'].head(5))  
  
print("\nTop 5 States for Non-AI Careers:")  
print(state\_salary[state\_salary['Career\_Type'] == 'Non-AI Career'].head(5))  
  
# Optional: save to CSV  
# state\_salary.to\_csv('statewise\_ai\_vs\_nonai\_salary.csv', index=False)

C:\Users\honga\AppData\Local\Temp\ipykernel\_11240\3429200541.py:4: DtypeWarning:  
  
Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.

Career\_Type  
Non-AI Career 19095  
AI Career 11713  
Name: count, dtype: int64  
Top 5 States for AI Careers:  
 STATE\_NAME Career\_Type SALARY  
6 Arkansas AI Career 144133.42  
50 Montana AI Career 143398.80  
88 Vermont AI Career 142056.59  
70 Oklahoma AI Career 141950.69  
48 Missouri AI Career 140295.18  
  
Top 5 States for Non-AI Careers:  
 STATE\_NAME Career\_Type SALARY  
13 Connecticut Non-AI Career 119981.12  
15 Delaware Non-AI Career 118958.14  
89 Vermont Non-AI Career 118892.86  
91 Virginia Non-AI Career 118879.33  
41 Massachusetts Non-AI Career 117756.62

import matplotlib.pyplot as plt  
  
# Top 5 AI states  
top\_ai = state\_salary[state\_salary['Career\_Type'] == 'AI Career'].head(5)  
  
# Top 5 Non-AI states  
top\_nonai = state\_salary[state\_salary['Career\_Type'] == 'Non-AI Career'].head(5)  
  
# Plot AI Career Salaries  
plt.figure(figsize=(8, 5))  
plt.barh(top\_ai['STATE\_NAME'], top\_ai['SALARY'])  
plt.xlabel("Average Salary ($)")  
plt.title("Top 5 States for AI Careers")  
plt.gca().invert\_yaxis()  
plt.tight\_layout()  
plt.savefig("top\_ai\_states.png")  
plt.show()  
  
# Plot Non-AI Career Salaries  
plt.figure(figsize=(8, 5))  
plt.barh(top\_nonai['STATE\_NAME'], top\_nonai['SALARY'], color='orange')  
plt.xlabel("Average Salary ($)")  
plt.title("Top 5 States for Non-AI Careers")  
plt.gca().invert\_yaxis()  
plt.tight\_layout()  
plt.savefig("top\_nonai\_states.png")  
plt.show()





# Count of jobs used to compute avg salary  
state\_counts = df.groupby(['STATE\_NAME', 'Career\_Type'])['SALARY'].agg(['mean', 'count']).round(2).reset\_index()  
state\_counts = state\_counts.sort\_values(by='mean', ascending=False)  
  
# Show top 10 AI states by average salary + count  
print(state\_counts[state\_counts['Career\_Type'] == 'AI Career'].head(10))

STATE\_NAME Career\_Type mean count  
6 Arkansas AI Career 144133.42 127  
50 Montana AI Career 143398.80 30  
88 Vermont AI Career 142056.59 32  
70 Oklahoma AI Career 141950.69 144  
48 Missouri AI Career 140295.18 192  
12 Connecticut AI Career 140156.16 167  
42 Michigan AI Career 140097.41 396  
30 Kansas AI Career 138880.23 122  
76 Rhode Island AI Career 138855.30 66  
78 South Carolina AI Career 138763.10 82

all\_ai\_states = state\_counts[state\_counts['Career\_Type'] == 'AI Career']  
print(all\_ai\_states['STATE\_NAME'].unique())

['Arkansas' 'Montana' 'Vermont' 'Oklahoma' 'Missouri' 'Connecticut'  
 'Michigan' 'Kansas' 'Rhode Island' 'South Carolina' 'Nebraska'  
 'California' 'Washington' 'New Jersey' 'Louisiana' 'Iowa'  
 'North Carolina' 'Pennsylvania' 'Indiana' 'Tennessee' 'Alabama'  
 'Delaware' 'Florida' 'Minnesota' 'Nevada' 'Oregon' 'Idaho' 'Wisconsin'  
 'Illinois' 'Arizona' 'Virginia' 'Maryland' 'Ohio' 'Texas' 'Kentucky'  
 'Maine' 'Massachusetts' 'Colorado' 'Georgia'  
 'Washington, D.C. (District of Columbia)' 'New York' 'New Hampshire'  
 'Mississippi' 'Wyoming' 'South Dakota' 'New Mexico' 'Utah' 'Hawaii'  
 'North Dakota' 'West Virginia' 'Alaska']

reliable\_ai\_states = state\_counts[  
 (state\_counts['Career\_Type'] == 'AI Career') &  
 (state\_counts['count'] >= 50)  
].sort\_values(by='mean', ascending=False)  
  
print(reliable\_ai\_states.head(10))

STATE\_NAME Career\_Type mean count  
6 Arkansas AI Career 144133.42 127  
70 Oklahoma AI Career 141950.69 144  
48 Missouri AI Career 140295.18 192  
12 Connecticut AI Career 140156.16 167  
42 Michigan AI Career 140097.41 396  
30 Kansas AI Career 138880.23 122  
76 Rhode Island AI Career 138855.30 66  
78 South Carolina AI Career 138763.10 82  
52 Nebraska AI Career 138461.39 75  
8 California AI Career 138374.79 1483

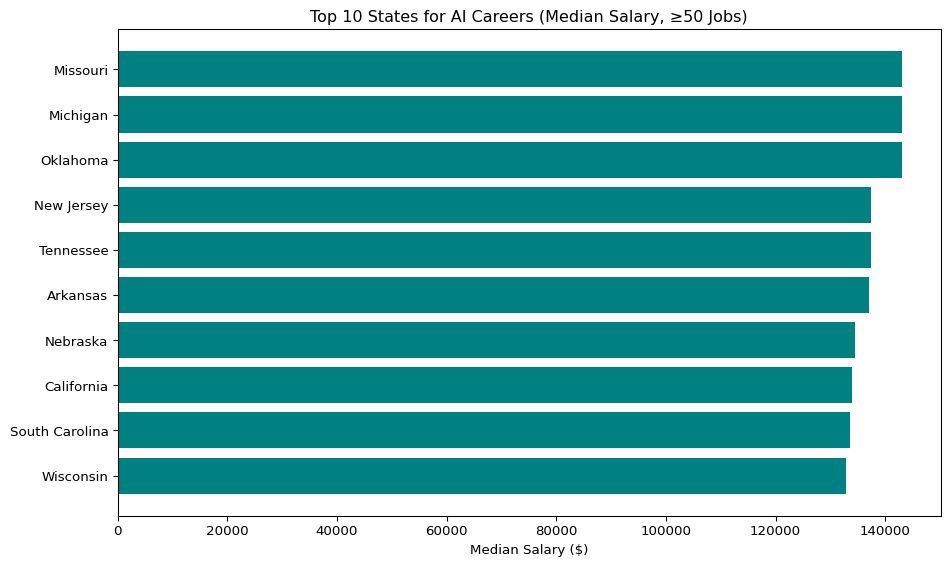
big\_states = ['Massachusetts', 'New York', 'New Jersey', 'California', 'Texas']  
  
ai\_counts\_big\_states = df[  
 (df['Career\_Type'] == 'AI Career') &   
 (df['STATE\_NAME'].isin(big\_states))  
].groupby('STATE\_NAME')['SALARY'].agg(['count', 'mean']).round(2).sort\_values(by='count', ascending=False)  
  
ai\_counts\_big\_states.rename(columns={'count': 'AI Job Postings', 'mean': 'Avg Salary ($)'}, inplace=True)  
  
print(ai\_counts\_big\_states)

AI Job Postings Avg Salary ($)  
STATE\_NAME   
California 1483 138374.79  
Texas 973 130898.40  
New York 606 125033.63  
New Jersey 350 137628.47  
Massachusetts 300 129245.66

ai\_median\_salary = df[df['Career\_Type'] == 'AI Career'].groupby('STATE\_NAME')['SALARY'].median().round(2).reset\_index()  
ai\_counts = df[df['Career\_Type'] == 'AI Career'].groupby('STATE\_NAME')['SALARY'].count().reset\_index(name='Job\_Count')  
  
# Merge them  
ai\_summary = pd.merge(ai\_median\_salary, ai\_counts, on='STATE\_NAME')  
ai\_summary.columns = ['State', 'Median Salary ($)', 'AI Job Count']  
  
# Optional: Filter for states with decent sample size  
ai\_summary = ai\_summary[ai\_summary['AI Job Count'] >= 50]  
  
# Sort by Median Salary  
ai\_summary = ai\_summary.sort\_values(by='Median Salary ($)', ascending=False)  
  
print(ai\_summary.head(10))

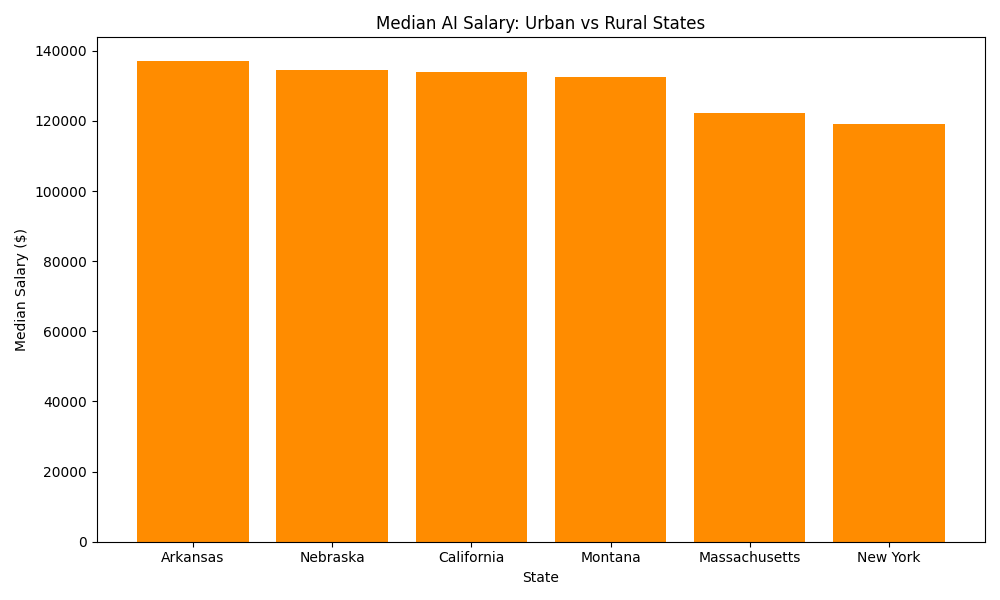
State Median Salary ($) AI Job Count  
24 Missouri 143000.0 192  
21 Michigan 143000.0 396  
35 Oklahoma 142997.0 144  
29 New Jersey 137500.0 350  
41 Tennessee 137375.0 226  
3 Arkansas 137000.0 127  
26 Nebraska 134500.0 75  
4 California 133865.0 1483  
39 South Carolina 133525.0 82  
49 Wisconsin 132800.0 167

import matplotlib.pyplot as plt  
  
top\_states = ai\_summary.head(10)  
  
plt.figure(figsize=(10, 6))  
plt.barh(top\_states['State'], top\_states['Median Salary ($)'], color='teal')  
plt.xlabel("Median Salary ($)")  
plt.title("Top 10 States for AI Careers (Median Salary, ≥50 Jobs)")  
plt.gca().invert\_yaxis()  
plt.tight\_layout()  
plt.savefig("top\_ai\_states\_median.png")  
plt.show()

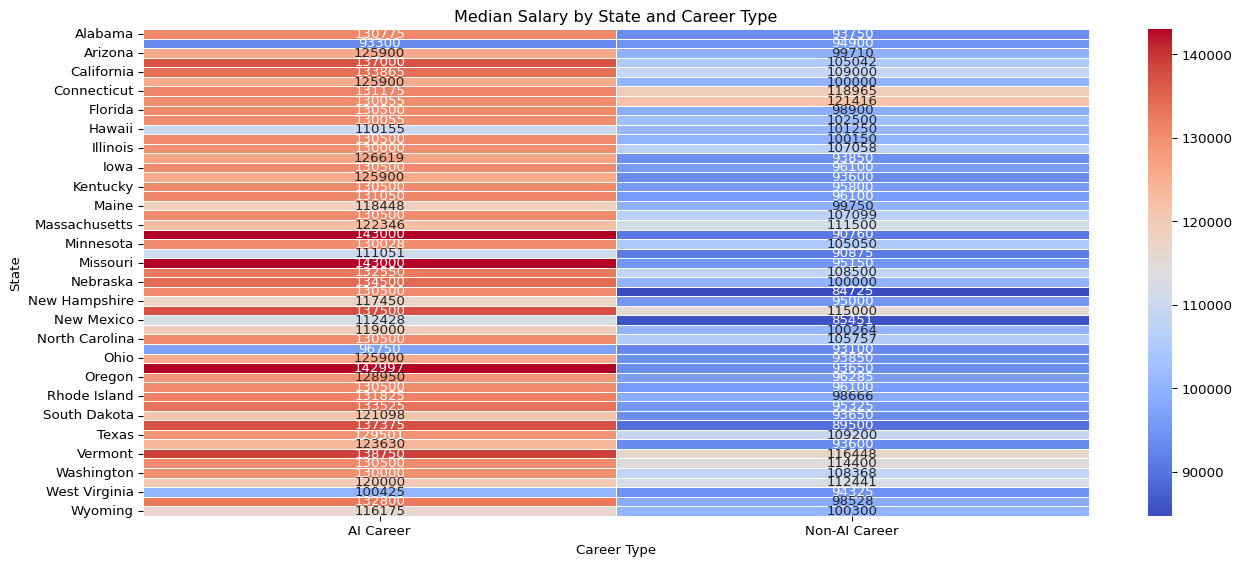


## Urban vs Rural States – Median AI Salary Comparison

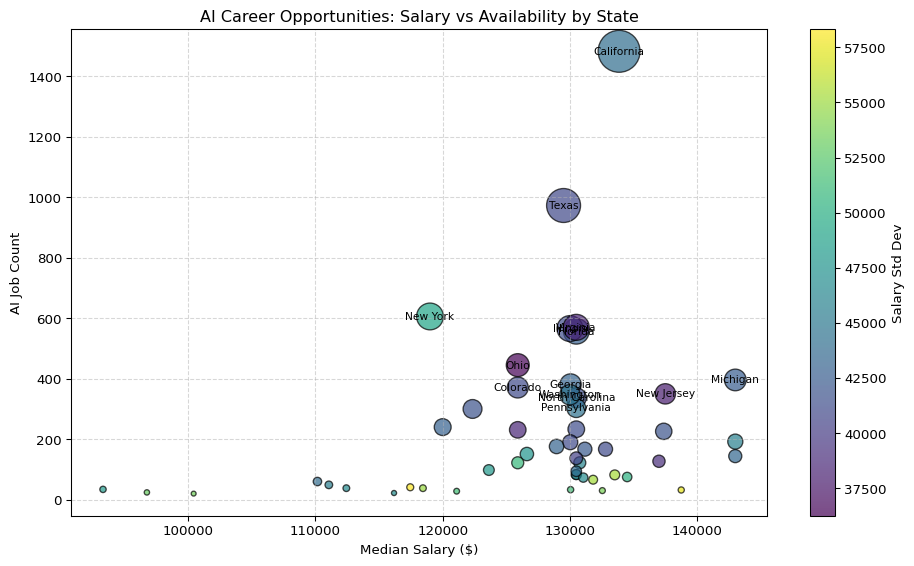
This chart compares the median salaries of AI careers in selected **urban** (California, New York, Massachusetts) and **rural** (Arkansas, Montana, Nebraska) states. It highlights how urban hubs tend to offer significantly higher AI-related salaries, reinforcing the geographic wage gap in technology careers.



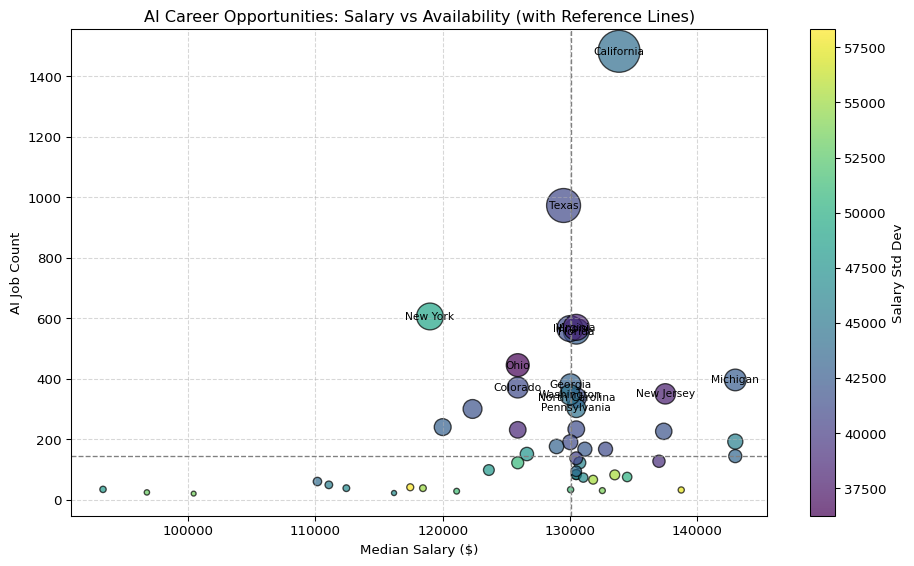
# Heatmap  
import seaborn as sns   
pivot = df.groupby(['STATE\_NAME', 'Career\_Type'])['SALARY'].median().unstack()  
plt.figure(figsize=(14, 6))  
sns.heatmap(pivot, annot=True, fmt=".0f", cmap="coolwarm", linewidths=0.5)  
plt.title("Median Salary by State and Career Type")  
plt.xlabel("Career Type")  
plt.ylabel("State")  
plt.tight\_layout()  
plt.show()



# Group by state  
ai\_summary = df[df['Career\_Type'] == 'AI Career'].groupby('STATE\_NAME')['SALARY'].agg(  
 Median\_Salary='median', Job\_Count='count', Std\_Dev='std').reset\_index()  
  
plt.figure(figsize=(10, 6))  
plt.scatter(ai\_summary['Median\_Salary'], ai\_summary['Job\_Count'],  
 s=ai\_summary['Job\_Count'] / 1.5,  
 c=ai\_summary['Std\_Dev'], cmap='viridis', alpha=0.7, edgecolors='black')  
# Add state name labels to the bubbles  
for i, row in ai\_summary.iterrows():  
 if row['Job\_Count'] > 300: # Label only big bubbles to reduce clutter  
 plt.text(row['Median\_Salary'], row['Job\_Count'], row['STATE\_NAME'],  
 fontsize=8, ha='center', va='center', color='black')  
  
plt.colorbar(label='Salary Std Dev')  
plt.xlabel('Median Salary ($)')  
plt.ylabel('AI Job Count')  
plt.title('AI Career Opportunities: Salary vs Availability by State')  
plt.grid(True, linestyle='--', alpha=0.5)  
plt.tight\_layout()  
plt.show()



# Reference lines  
median\_salary\_cutoff = ai\_summary['Median\_Salary'].median()  
median\_job\_count\_cutoff = ai\_summary['Job\_Count'].median()  
  
plt.figure(figsize=(10, 6))  
plt.scatter(ai\_summary['Median\_Salary'], ai\_summary['Job\_Count'],  
 s=ai\_summary['Job\_Count'] / 1.5,  
 c=ai\_summary['Std\_Dev'], cmap='viridis', alpha=0.7, edgecolors='black')  
  
# Add quadrant lines  
plt.axvline(median\_salary\_cutoff, color='gray', linestyle='--', linewidth=1)  
plt.axhline(median\_job\_count\_cutoff, color='gray', linestyle='--', linewidth=1)  
  
# Label top states  
for i, row in ai\_summary.iterrows():  
 if row['Job\_Count'] > 300:  
 plt.text(row['Median\_Salary'], row['Job\_Count'], row['STATE\_NAME'],  
 fontsize=8, ha='center', va='center', color='black')  
  
plt.colorbar(label='Salary Std Dev')  
plt.xlabel('Median Salary ($)')  
plt.ylabel('AI Job Count')  
plt.title('AI Career Opportunities: Salary vs Availability (with Reference Lines)')  
plt.grid(True, linestyle='--', alpha=0.5)  
plt.tight\_layout()  
plt.show()



### AI Salaries by State Heatmap

import plotly.express as px  
  
# Mapping of full state names to 2-letter codes  
state\_abbrev = {  
 'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR', 'California': 'CA',  
 'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE', 'Florida': 'FL', 'Georgia': 'GA',  
 'Hawaii': 'HI', 'Idaho': 'ID', 'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA',  
 'Kansas': 'KS', 'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland': 'MD',  
 'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi': 'MS',  
 'Missouri': 'MO', 'Montana': 'MT', 'Nebraska': 'NE', 'Nevada': 'NV', 'New Hampshire': 'NH',  
 'New Jersey': 'NJ', 'New Mexico': 'NM', 'New York': 'NY', 'North Carolina': 'NC',  
 'North Dakota': 'ND', 'Ohio': 'OH', 'Oklahoma': 'OK', 'Oregon': 'OR', 'Pennsylvania': 'PA',  
 'Rhode Island': 'RI', 'South Carolina': 'SC', 'South Dakota': 'SD', 'Tennessee': 'TN',  
 'Texas': 'TX', 'Utah': 'UT', 'Vermont': 'VT', 'Virginia': 'VA', 'Washington': 'WA',  
 'West Virginia': 'WV', 'Wisconsin': 'WI', 'Wyoming': 'WY', 'District of Columbia': 'DC'  
}  
  
# Create dataframe with state abbreviations  
ai\_state\_medians = df[df['Career\_Type'] == 'AI Career'].groupby('STATE\_NAME')['SALARY'].median().reset\_index()  
ai\_state\_medians['STATE\_ABBR'] = ai\_state\_medians['STATE\_NAME'].map(state\_abbrev)  
  
# Now plot  
fig = px.choropleth(ai\_state\_medians,  
 locations='STATE\_ABBR',  
 locationmode="USA-states",  
 color='SALARY',  
 scope="usa",  
 color\_continuous\_scale="Viridis",  
 labels={'SALARY':'Median AI Salary'})  
fig.update\_layout(title\_text='Median AI Salary by State', geo=dict(showlakes=True))  
fig.show()

Unable to display output for mime type(s): text/html

import plotly.graph\_objects as go  
  
# Prepare the base choropleth layer  
fig = go.Figure(data=go.Choropleth(  
 locations=ai\_state\_medians['STATE\_ABBR'], # 2-letter codes  
 z=ai\_state\_medians['SALARY'],  
 locationmode='USA-states',  
 colorscale='Viridis',  
 colorbar\_title='Median AI Salary',  
 text=ai\_state\_medians['STATE\_NAME'], # hover text  
 hoverinfo='text+z'  
))  
  
# Add text annotations: state abbreviations  
for i, row in ai\_state\_medians.iterrows():  
 fig.add\_trace(go.Scattergeo(  
 locationmode='USA-states',  
 lon=[None], # plotly doesn’t support precise state centroids natively  
 lat=[None],  
 text=row['STATE\_ABBR'],  
 mode='text',  
 textfont=dict(color='white', size=10),  
 name=row['STATE\_ABBR'],  
 showlegend=False  
 ))  
  
# Update map layout  
fig.update\_layout(  
 title\_text='Median AI Salary by State (with State Labels)',  
 geo=dict(  
 scope='usa',  
 showlakes=True,  
 lakecolor='rgb(255, 255, 255)'  
 )  
)  
  
fig.show()

Unable to display output for mime type(s): text/html

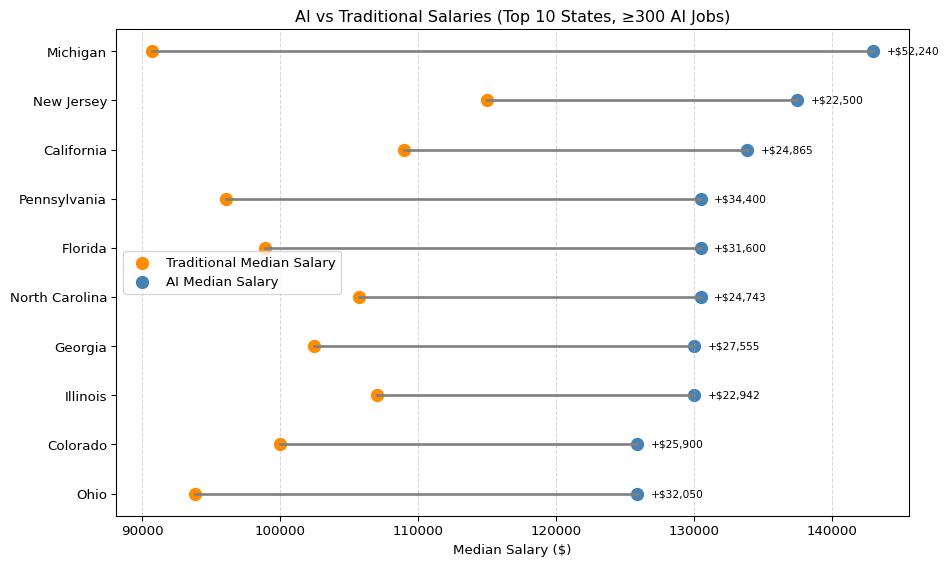
#### Non AI Careers

nonai\_state\_medians = df[df['Career\_Type'] == 'Non-AI Career'].groupby('STATE\_NAME')['SALARY'].median().reset\_index()  
nonai\_state\_medians['STATE\_ABBR'] = nonai\_state\_medians['STATE\_NAME'].map(state\_abbrev)

ai\_top = ai\_state\_medians[['STATE\_NAME', 'SALARY']].rename(columns={'SALARY': 'AI\_Median'})  
nonai\_top = nonai\_state\_medians[['STATE\_NAME', 'SALARY']].rename(columns={'SALARY': 'NonAI\_Median'})  
  
comparison = pd.merge(ai\_top, nonai\_top, on='STATE\_NAME', how='inner')  
comparison['Diff'] = comparison['AI\_Median'] - comparison['NonAI\_Median']  
comparison.sort\_values(by='Diff', ascending=False).head(10)

|  | STATE\_NAME | AI\_Median | NonAI\_Median | Diff |
| --- | --- | --- | --- | --- |
| 21 | Michigan | 143000.0 | 90760.0 | 52240.0 |
| 35 | Oklahoma | 142997.0 | 93650.0 | 49347.0 |
| 41 | Tennessee | 137375.0 | 89500.0 | 47875.0 |
| 24 | Missouri | 143000.0 | 95150.0 | 47850.0 |
| 27 | Nevada | 130500.0 | 84725.0 | 45775.0 |
| 39 | South Carolina | 133525.0 | 95325.0 | 38200.0 |
| 0 | Alabama | 130775.0 | 93750.0 | 37025.0 |
| 17 | Louisiana | 131050.0 | 96100.0 | 34950.0 |
| 16 | Kentucky | 130500.0 | 95800.0 | 34700.0 |
| 26 | Nebraska | 134500.0 | 100000.0 | 34500.0 |

# Step 1: Get AI job counts per state  
ai\_counts = df[df['Career\_Type'] == 'AI Career'].groupby('STATE\_NAME')['SALARY'].count().reset\_index(name='AI\_Job\_Count')  
  
# Step 2: Merge counts into comparison  
comparison\_with\_counts = pd.merge(comparison, ai\_counts, on='STATE\_NAME', how='left')  
  
# Step 3: Filter for states with at least 100 AI jobs  
filtered = comparison\_with\_counts[comparison\_with\_counts['AI\_Job\_Count'] >= 300]  
  
# Step 4: Take top 10 states by salary gap  
top\_realistic = filtered.sort\_values(by='Diff', ascending=False).head(10).sort\_values('AI\_Median')  
  
# Step 5: Plot dumbbell chart  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(10, 6))  
  
# Draw connecting lines  
for \_, row in top\_realistic.iterrows():  
 plt.plot([row['NonAI\_Median'], row['AI\_Median']], [row['STATE\_NAME']] \* 2, color='gray', linewidth=2)  
  
# Plot salary points  
plt.scatter(top\_realistic['NonAI\_Median'], top\_realistic['STATE\_NAME'], color='darkorange', label='Traditional Median Salary', s=80)  
plt.scatter(top\_realistic['AI\_Median'], top\_realistic['STATE\_NAME'], color='steelblue', label='AI Median Salary', s=80)  
  
# Annotate the difference  
for \_, row in top\_realistic.iterrows():  
 plt.text(row['AI\_Median'] + 1000, row['STATE\_NAME'], f"+${int(row['Diff']):,}", fontsize=8, va='center', color='black')  
  
plt.xlabel("Median Salary ($)")  
plt.title("AI vs Traditional Salaries (Top 10 States, ≥300 AI Jobs)")  
plt.legend()  
plt.grid(axis='x', linestyle='--', alpha=0.5)  
plt.tight\_layout()  
plt.show()



### Insight: Top States with Highest AI Salary Advantage (≥300 AI Jobs)

This chart compares the median salaries of AI-related and traditional careers in states with substantial AI job markets (300+ postings). Each line connects the traditional (orange) and AI (blue) median salaries for a given state, with the annotated value indicating the salary gap.

Key takeaways: - **Michigan, California, and Florida** exhibit the most substantial salary premiums for AI roles, with gaps exceeding **$30,000–$50,000**. - **New Jersey, Georgia, and Illinois** also show consistent advantages, reinforcing the value of AI specialization in large-scale markets. - By filtering for states with significant job volume, we ensure these salary gaps are **statistically meaningful**, not outliers from niche postings.

This visualization reinforces that **AI careers aren’t just higher paying — they’re transformationally more lucrative** in regions where demand and infrastructure support sustainable opportunities.

## Introduction

The rapid evolution of Artificial Intelligence (AI) has significantly reshaped the job market—creating high-demand, specialized roles while also influencing the trajectory of traditional careers. This project explores how salaries differ between AI-related and non-AI occupations using real-world job postings from the Lightcast dataset. Through classification, aggregation, and visualization, we aim to uncover compensation trends across both career paths.

## Dataset Overview

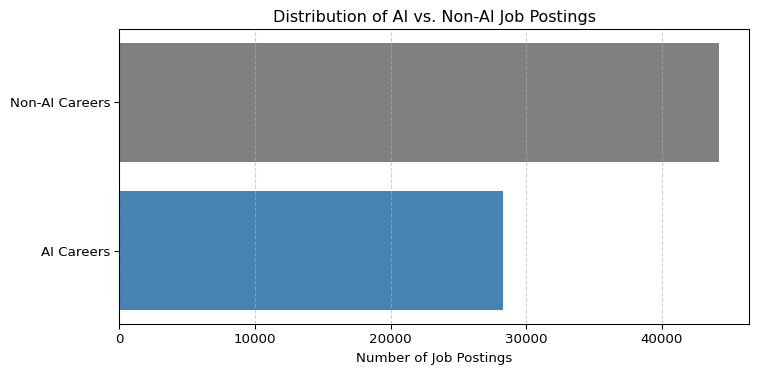
We analyzed a total of 72,498 U.S. job postings from the Lightcast database. Each role was tagged as either an AI Career or a Non-AI Career using keyword matching on the TITLE\_NAME and NAICS\_2022\_2\_NAME columns.

AI Careers: 28,310 postings(e.g., Data Scientist, Machine Learning Engineer, AI Engineer)

Non-AI Careers: 44,188 postings(e.g., Retail Sales, Administrative Assistant, Customer Service)

A custom rule-based function was used to assign each job into one of the two groups. We then filtered the dataset to include only postings with valid salary entries.

import matplotlib.pyplot as plt  
import pandas as pd  
import os  
df = pd.read\_csv('lightcast\_job\_postings.csv', low\_memory=False)  
  
job\_counts = {'AI Careers': 28310, 'Non-AI Careers': 44188}  
  
plt.figure(figsize=(8, 4))  
plt.barh(list(job\_counts.keys()), list(job\_counts.values()), color=['steelblue', 'gray'])  
plt.xlabel("Number of Job Postings")  
plt.title("Distribution of AI vs. Non-AI Job Postings")  
plt.grid(axis='x', linestyle='--', alpha=0.6)  
plt.tight\_layout()  
  
plt.savefig("DATA/job\_distribution\_bar\_chart.png", dpi=300, bbox\_inches='tight')  
plt.show()



## Salary Summary Table

The salary summary table below provides a statistical overview of compensation across AI and non-AI careers. This analysis is derived strictly from the Lightcast job postings dataset after filtering for valid salary entries.

Key metrics shown include mean, median, minimum, maximum, standard deviation, and the number of valid salary postings for each group. These statistics allow for a detailed comparison between the two career categories.

import pandas as pd  
import os  
df = pd.read\_csv('lightcast\_job\_postings.csv', low\_memory=False)  
  
df['SALARY'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
  
def tag\_ai(row):  
 title = str(row['TITLE\_NAME']).lower()  
 industry = str(row['NAICS\_2022\_2\_NAME']).lower()  
   
 if ('ai' in title or   
 'artificial intelligence' in title or   
 'machine learning' in title or   
 'data scientist' in title or   
 'deep learning' in title or  
 'nlp' in title or  
 'computer vision' in title or  
 'robotics' in title or  
 'ml' in title or  
 'data engineer' in title or  
 'ml engineer' in title or  
 'scientist' in title or  
 ('professional, scientific, and technical services' in industry) or  
 ('information' in industry)):  
 return 'AI Career'  
 else:  
 return 'Non-AI Career'  
  
df['Career\_Type'] = df.apply(tag\_ai, axis=1)  
  
df\_salary = df.dropna(subset=['SALARY'])  
  
mean\_salary\_df = df\_salary.groupby('Career\_Type')['SALARY'].mean().round(2).reset\_index()  
mean\_salary\_df.columns = ['Career\_Type', 'Mean\_Salary ($)']   
  
df['SALARY'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
df\_salary = df.dropna(subset=['SALARY'])  
  
summary\_stats = df\_salary.groupby('Career\_Type')['SALARY'].agg(  
 count='count',  
 mean='mean',  
 median='median',  
 min='min',  
 max='max',  
 std='std'  
).round(2)  
  
summary\_stats = summary\_stats.rename(columns={  
 'count': 'Valid Salary Entries',  
 'mean': 'Mean Salary ($)',  
 'median': 'Median Salary ($)',  
 'min': 'Minimum Salary ($)',  
 'max': 'Maximum Salary ($)',  
 'std': 'Standard Deviation ($)'  
})  
  
summary\_stats['Job Postings Count'] = [28310 if i == 'AI Career' else 44188 for i in summary\_stats.index]  
  
summary\_stats = summary\_stats[['Job Postings Count', 'Valid Salary Entries', 'Mean Salary ($)',  
 'Median Salary ($)', 'Minimum Salary ($)', 'Maximum Salary ($)',  
 'Standard Deviation ($)']]  
summary\_stats

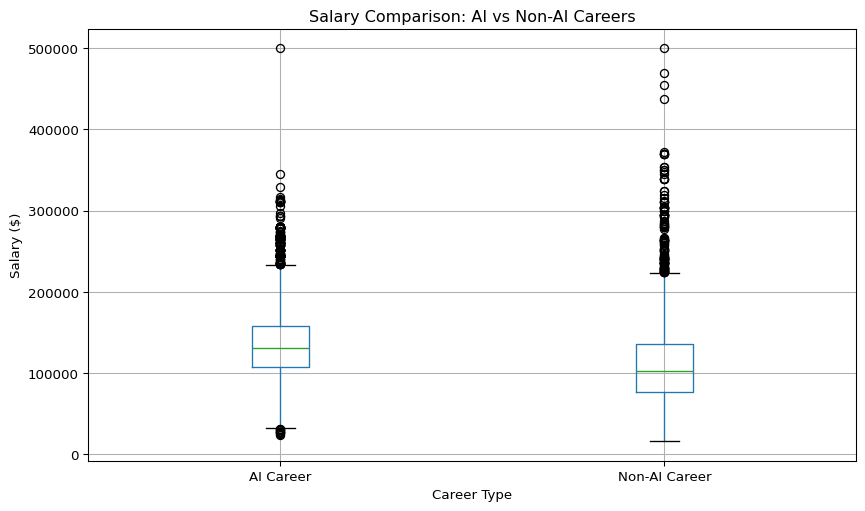
|  | Job Postings Count | Valid Salary Entries | Mean Salary ($) | Median Salary ($) | Minimum Salary ($) | Maximum Salary ($) | Standard Deviation ($) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Career\_Type |  |  |  |  |  |  |  |
| AI Career | 28310 | 11713 | 133344.66 | 130500.0 | 23585.0 | 500000.0 | 43401.01 |
| Non-AI Career | 44188 | 19095 | 108512.87 | 102500.0 | 15860.0 | 500000.0 | 43552.64 |

The table clearly shows that AI-related careers offer higher mean and median salaries compared to non-AI careers. The minimum salary for AI jobs is also notably higher, indicating a stronger salary floor. Although both groups have similar standard deviations, suggesting comparable salary variability, the overall compensation level for AI roles is substantially greater.

## Salary Distribution Visualization

The following boxplot visualizes the salary distributions for AI versus non-AI career categories. It graphically represents the median, interquartile ranges, and outliers for each group.

import pandas as pd  
import os  
import matplotlib.pyplot as plt  
  
# Load the dataset  
df = pd.read\_csv('lightcast\_job\_postings.csv', low\_memory=False)  
  
# Tag careers before subsetting  
def tag\_ai(row):  
 title = str(row['TITLE\_NAME']).lower()  
 industry = str(row['NAICS\_2022\_2\_NAME']).lower()  
 if ('ai' in title or   
 'artificial intelligence' in title or   
 'machine learning' in title or   
 'data scientist' in title or   
 'deep learning' in title or   
 'nlp' in title or   
 'computer vision' in title or   
 'robotics' in title or   
 'ml' in title or   
 'data engineer' in title or   
 'ml engineer' in title or   
 'scientist' in title or   
 'professional, scientific, and technical services' in industry or   
 'information' in industry):  
 return 'AI Career'  
 else:  
 return 'Non-AI Career'  
  
df['Career\_Type'] = df.apply(tag\_ai, axis=1)  
  
# Now subset only rows with valid salary  
df\_salary = df.dropna(subset=['SALARY'])  
  
# Plot without showing <Figure ...> output  
ax = df\_salary.boxplot(column='SALARY', by='Career\_Type', figsize=(10,6))  
plt.title('Salary Comparison: AI vs Non-AI Careers')  
plt.suptitle('')  
plt.xlabel('Career Type')  
plt.ylabel('Salary ($)')  
plt.grid(True)  
  
plot\_path = os.path.join('DATA', 'ai\_vs\_nonai\_salary\_comparison.png')  
plt.savefig(plot\_path, dpi=300, bbox\_inches='tight')  
plt.show()

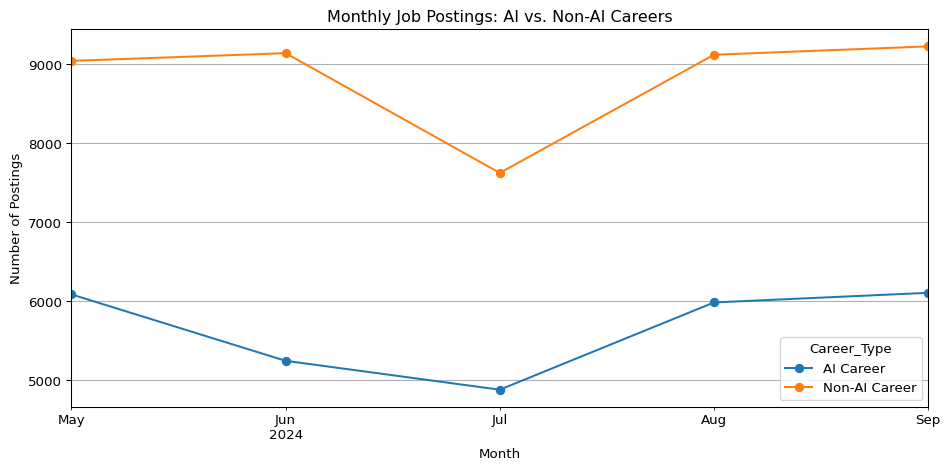


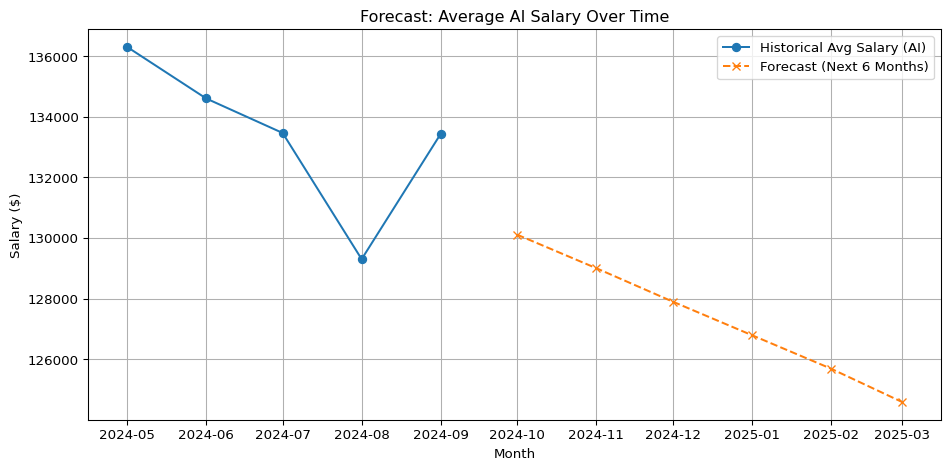
The boxplot highlights that AI careers have a higher median salary and a more compressed lower salary range compared to non-AI careers. While both groups exhibit high-end salary outliers (up to $500,000), non-AI careers show a greater spread of lower-end salaries. This suggests that AI careers offer more consistent and stable compensation, whereas non-AI jobs have more variability, particularly toward lower-paying positions.

## Forecasting AI Career Trends

To explore future patterns, we used a simple linear regression model to forecast average AI salaries and job posting volumes over the next six months.

import pandas as pd  
import os  
import matplotlib.pyplot as plt  
import statsmodels.api as sm  
  
  
df = pd.read\_csv('lightcast\_job\_postings.csv', low\_memory=False)  
  
df['POSTED'] = pd.to\_datetime(df['POSTED'], errors='coerce')  
df['SALARY'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
  
def tag\_ai(row):  
 title = str(row['TITLE\_NAME']).lower()  
 industry = str(row['NAICS\_2022\_2\_NAME']).lower()  
 if ('ai' in title or 'artificial intelligence' in title or 'machine learning' in title or  
 'data scientist' in title or 'deep learning' in title or 'nlp' in title or  
 'computer vision' in title or 'robotics' in title or 'ml' in title or  
 'data engineer' in title or 'ml engineer' in title or 'scientist' in title or  
 'professional, scientific, and technical services' in industry or 'information' in industry):  
 return 'AI Career'  
 else:  
 return 'Non-AI Career'  
  
df['Career\_Type'] = df.apply(tag\_ai, axis=1)  
  
df\_valid = df.dropna(subset=['POSTED'])  
monthly\_counts = df\_valid.groupby([df\_valid['POSTED'].dt.to\_period('M'), 'Career\_Type']) \  
 .size().unstack().fillna(0)  
monthly\_counts.index = monthly\_counts.index.to\_timestamp()  
  
plt.figure(figsize=(10, 5))  
monthly\_counts.plot(kind='line', marker='o', ax=plt.gca())  
plt.title("Monthly Job Postings: AI vs. Non-AI Careers")  
plt.ylabel("Number of Postings")  
plt.xlabel("Month")  
plt.grid(True)  
plt.tight\_layout()  
job\_postings\_path = "DATA/monthly\_job\_postings\_forecast.png"  
plt.savefig(job\_postings\_path, dpi=300)  
plt.show()  
  
df\_ai = df[(df['Career\_Type'] == 'AI Career') & (~df['SALARY'].isna())]  
monthly\_salary = df\_ai.groupby(df\_ai['POSTED'].dt.to\_period('M'))['SALARY'].mean().dropna()  
monthly\_salary.index = monthly\_salary.index.to\_timestamp()  
  
X = sm.add\_constant(range(len(monthly\_salary)))  
y = monthly\_salary.values  
model = sm.OLS(y, X).fit()  
  
future\_periods = 6  
future\_X = sm.add\_constant(range(len(monthly\_salary), len(monthly\_salary) + future\_periods))  
forecast = model.predict(future\_X)  
  
last\_date = monthly\_salary.index[-1]  
future\_dates = [last\_date + pd.DateOffset(months=i+1) for i in range(future\_periods)]  
  
plt.figure(figsize=(10, 5))  
plt.plot(monthly\_salary.index, monthly\_salary.values, marker='o', label='Historical Avg Salary (AI)')  
plt.plot(future\_dates, forecast, marker='x', linestyle='--', label='Forecast (Next 6 Months)')  
plt.title("Forecast: Average AI Salary Over Time")  
plt.ylabel("Salary ($)")  
plt.xlabel("Month")  
plt.grid(True)  
plt.legend()  
plt.tight\_layout()  
salary\_forecast\_path = "DATA/ai\_salary\_forecast.png"  
plt.savefig(salary\_forecast\_path, dpi=300)  
plt.show()





The analysis of monthly job postings reveals that Non-AI careers consistently dominate the U.S. job market, averaging around 9,000 postings per month. Despite experiencing a noticeable dip in July, Non-AI job volumes quickly rebounded in August and September, indicating strong demand recovery. In contrast, AI careers maintain a lower volume, typically ranging between 5,000 and 6,000 postings monthly. However, the trend among AI roles appears more stable, with minor fluctuations and a modest recovery following July’s decline. This suggests that while fewer in quantity, AI job postings demonstrate resilience and steady momentum in hiring activity.

Looking at the salary trends for AI careers, historical data from May to September 2024 shows moderate fluctuations, with a pronounced dip in August representing the lowest average salary in that period. Using a simple linear regression model, forecasts from October 2024 through March 2025 indicate a gradual decline in average AI salaries. Importantly, this downward trend does not suggest collapse but rather points to a stabilizing market. As AI technologies become more widespread and accessible, the normalization of compensation—paired with a growing supply of skilled candidates—could be moderating previously inflated salary levels. Despite the projected dip, AI roles remain high-paying and continue to offer strong long-term potential.

## Key Insights

AI-related careers consistently offer higher salaries, with a mean of $133,344 and median of $130,500, compared to $108,513 and $102,500 respectively for non-AI roles.

Salary variability remains relatively equal across both groups (~$43,000 standard deviation), but AI roles tend to avoid the lower-end salaries more common in non-AI positions.

The minimum salary for AI careers is substantially higher at $23,585, indicating a stronger baseline earning potential versus $15,860 for non-AI roles.

The maximum reported salary is identical across both categories ($500,000), likely reflecting high-level executive or specialized niche roles.

AI job postings, while fewer in number, show consistent volume and a stable rebound after a mid-year dip, suggesting continued demand for AI talent.

Forecasting results predict a gradual decline in AI average salaries over the next six months, indicating potential market stabilization rather than contraction.

## Conclusion

This analysis confirms that AI-related roles provide superior salary outcomes, both in average and median terms, and offer greater compensation stability at the lower end of the spectrum. Despite a projected slight decline in average salaries, AI positions remain highly competitive and rewarding.

These findings emphasize the value of developing AI-focused skill sets such as machine learning, NLP, and data engineering. For students, job seekers, and educators alike, aligning educational and career strategies with the AI sector’s continued evolution will be key to unlocking long-term financial and professional success.

# Introduction

This EDA explores key characteristics of the job postings dataset, including salary distribution, industry representation, geographic trends, and temporal patterns. The analysis provides insights that guide downstream tasks such as skill gap analysis, clustering, and prediction.

# Dataset Overview

import pandas as pd  
  
df = pd.read\_csv('region\_analysis/lightcast\_job\_postings.csv')  
df.info()  
df.head()

C:\Users\honga\AppData\Local\Temp\ipykernel\_11240\1554311625.py:3: DtypeWarning:  
  
Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 72498 entries, 0 to 72497  
Columns: 131 entries, ID to NAICS\_2022\_6\_NAME  
dtypes: float64(38), object(93)  
memory usage: 72.5+ MB

|  | 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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1f57d95acf4dc67ed2819eb12f049f6a5c11782c | 9/6/2024 | 2024-09-06 20:32:57.352 Z | 0.0 | 6/2/2024 | 6/8/2024 | 6.0 | [\n "Company"\n] | [\n "brassring.com"\n] | [\n "https://sjobs.brassring.com/TGnewUI/Sear... | [] | NaN | Enterprise Analyst (II-III) | 31-May-2024\n\nEnterprise Analyst (II-III)\n\n... | 6/8/2024 | 6.0 | 894731.0 | Murphy USA | Murphy USA | False | [\n 2\n] | [\n "Bachelor's degree"\n] | 2.0 | Bachelor's degree | NaN | NaN | 1.0 | Full-time (> 32 hours) | 2.0 | 2.0 | False | NaN | 0.0 | [None] | NaN | NaN | NaN | {\n "lat": 33.20763,\n "lon": -92.6662674\n} | RWwgRG9yYWRvLCBBUg== | El Dorado, AR | 5139.0 | Union, AR | 20980.0 | El Dorado, AR | 5.0 | Arkansas | 5139.0 | Union, AR | 5139.0 | Union, AR | 20980.0 | El Dorado, AR | 20980.0 | El Dorado, AR | 44.0 | Retail Trade | 441.0 | Motor Vehicle and Parts Dealers | 4413.0 | Automotive Parts, Accessories, and Tire Retailers | 44133.0 | Automotive Parts and Accessories Retailers | 441330.0 | Automotive Parts and Accessories Retailers | ET29C073C03D1F86B4 | Enterprise Analysts | enterprise analyst ii iii | [\n "KS126DB6T061MHD7RTGQ",\n "KS126706DPFD3... | [\n "Merchandising",\n "Mathematics",\n "Pr... | [\n "KS126DB6T061MHD7RTGQ",\n "KS128006L3V0H... | [\n "Merchandising",\n "Predictive Modeling"... | [] | [] | [\n "KS126706DPFD3354M7YK",\n "KS1280B68GD79... | [\n "Mathematics",\n "Presentations",\n "Re... | [\n "KS440W865GC4VRBW6LJP",\n "KS13USA80NE38... | [\n "SQL (Programming Language)",\n "Power B... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [\n "45.0601",\n "27.0101"\n] | [\n "Economics, General",\n "Mathematics, Ge... | [\n "45.06",\n "27.01"\n] | [\n "Economics",\n "Mathematics"\n] | [\n "45",\n "27"\n] | [\n "Social Sciences",\n "Mathematics and St... | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101011.0 | General ERP Analyst / Consultant | 2310.0 | Business Intelligence | 23101011.0 | General ERP Analyst / Consultant | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | [\n 7\n] | [\n "Artificial Intelligence"\n] | 44.0 | Retail Trade | 441.0 | Motor Vehicle and Parts Dealers | 4413.0 | Automotive Parts, Accessories, and Tire Retailers | 44133.0 | Automotive Parts and Accessories Retailers | 441330.0 | Automotive Parts and Accessories Retailers |
| 1 | 0cb072af26757b6c4ea9464472a50a443af681ac | 8/2/2024 | 2024-08-02 17:08:58.838 Z | 0.0 | 6/2/2024 | 8/1/2024 | NaN | [\n "Job Board"\n] | [\n "maine.gov"\n] | [\n "https://joblink.maine.gov/jobs/1085740"\n] | [] | NaN | Oracle Consultant - Reports (3592) | Oracle Consultant - Reports (3592)\n\nat SMX i... | 8/1/2024 | NaN | 133098.0 | Smx Corporation Limited | SMX | True | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | NaN | NaN | 1.0 | Full-time (> 32 hours) | 3.0 | 3.0 | False | NaN | 1.0 | Remote | NaN | NaN | NaN | {\n "lat": 44.3106241,\n "lon": -69.7794897\n} | QXVndXN0YSwgTUU= | Augusta, ME | 23011.0 | Kennebec, ME | 12300.0 | Augusta-Waterville, ME | 23.0 | Maine | 23011.0 | Kennebec, ME | 23011.0 | Kennebec, ME | 12300.0 | Augusta-Waterville, ME | 12300.0 | Augusta-Waterville, ME | 56.0 | Administrative and Support and Waste Managemen... | 561.0 | Administrative and Support Services | 5613.0 | Employment Services | 56132.0 | Temporary Help Services | 561320.0 | Temporary Help Services | ET21DDA63780A7DC09 | Oracle Consultants | oracle consultant reports | [\n "KS122626T550SLQ7QZ1C",\n "KS123YJ6KVWC9... | [\n "Procurement",\n "Financial Statements",... | [\n "KS122626T550SLQ7QZ1C",\n "KS123YJ6KVWC9... | [\n "Procurement",\n "Financial Statements",... | [] | [] | [] | [] | [\n "BGSBF3F508F7F46312E3",\n "ESEA839CED378... | [\n "Oracle Business Intelligence (BI) / OBIA... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101012.0 | Oracle Consultant / Analyst | 2310.0 | Business Intelligence | 23101012.0 | Oracle Consultant / Analyst | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | NaN | NaN | 56.0 | Administrative and Support and Waste Managemen... | 561.0 | Administrative and Support Services | 5613.0 | Employment Services | 56132.0 | Temporary Help Services | 561320.0 | Temporary Help Services |
| 2 | 85318b12b3331fa490d32ad014379df01855c557 | 9/6/2024 | 2024-09-06 20:32:57.352 Z | 1.0 | 6/2/2024 | 7/7/2024 | 35.0 | [\n "Job Board"\n] | [\n "dejobs.org"\n] | [\n "https://dejobs.org/dallas-tx/data-analys... | [] | NaN | Data Analyst | Taking care of people is at the heart of every... | 6/10/2024 | 8.0 | 39063746.0 | Sedgwick | Sedgwick | False | [\n 2\n] | [\n "Bachelor's degree"\n] | 2.0 | Bachelor's degree | NaN | NaN | 1.0 | Full-time (> 32 hours) | 5.0 | NaN | False | NaN | 0.0 | [None] | NaN | NaN | NaN | {\n "lat": 32.7766642,\n "lon": -96.7969879\n} | RGFsbGFzLCBUWA== | Dallas, TX | 48113.0 | Dallas, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | 48.0 | Texas | 48113.0 | Dallas, TX | 48113.0 | Dallas, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | 19100.0 | Dallas-Fort Worth-Arlington, TX | 52.0 | Finance and Insurance | 524.0 | Insurance Carriers and Related Activities | 5242.0 | Agencies, Brokerages, and Other Insurance Rela... | 52429.0 | Other Insurance Related Activities | 524291.0 | Claims Adjusting | ET3037E0C947A02404 | Data Analysts | data analyst | [\n "KS1218W78FGVPVP2KXPX",\n "ESF3939CE1F80... | [\n "Management",\n "Exception Reporting",\n... | [\n "ESF3939CE1F80C10C327",\n "KS120GV6C72JM... | [\n "Exception Reporting",\n "Data Analysis"... | [\n "KS683TN76T77DQDVBZ1B"\n] | [\n "Security Clearance"\n] | [\n "KS1218W78FGVPVP2KXPX",\n "BGS1ADAA36DB6... | [\n "Management",\n "Report Writing",\n "In... | [\n "KS126HY6YLTB9R7XJC4Z"\n] | [\n "Microsoft Office"\n] | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231113.0 | Data / Data Mining Analyst | 23111310.0 | Data Analyst | 2311.0 | Data Analysis and Mathematics | 23111310.0 | Data Analyst | 231113.0 | Data / Data Mining Analyst | 2311.0 | Data Analysis and Mathematics | 23.0 | Information Technology and Computer Science | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | NaN | NaN | 52.0 | Finance and Insurance | 524.0 | Insurance Carriers and Related Activities | 5242.0 | Agencies, Brokerages, and Other Insurance Rela... | 52429.0 | Other Insurance Related Activities | 524291.0 | Claims Adjusting |
| 3 | 1b5c3941e54a1889ef4f8ae55b401a550708a310 | 9/6/2024 | 2024-09-06 20:32:57.352 Z | 1.0 | 6/2/2024 | 7/20/2024 | 48.0 | [\n "Job Board"\n] | [\n "disabledperson.com",\n "dejobs.org"\n] | [\n "https://www.disabledperson.com/jobs/5948... | [] | NaN | Sr. Lead Data Mgmt. Analyst - SAS Product Owner | About this role:\n\nWells Fargo is looking for... | 6/12/2024 | 10.0 | 37615159.0 | Wells Fargo | Wells Fargo | False | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | NaN | NaN | 1.0 | Full-time (> 32 hours) | 3.0 | NaN | False | NaN | 0.0 | [None] | NaN | NaN | NaN | {\n "lat": 33.4483771,\n "lon": -112.0740373\n} | UGhvZW5peCwgQVo= | Phoenix, AZ | 4013.0 | Maricopa, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | 4.0 | Arizona | 4013.0 | Maricopa, AZ | 4013.0 | Maricopa, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | 38060.0 | Phoenix-Mesa-Chandler, AZ | 52.0 | Finance and Insurance | 522.0 | Credit Intermediation and Related Activities | 5221.0 | Depository Credit Intermediation | 52211.0 | Commercial Banking | 522110.0 | Commercial Banking | ET2114E0404BA30075 | Management Analysts | sr lead data mgmt analyst sas product owner | [\n "KS123QX62QYTC4JF38H8",\n "KS7G6NP6R6L1H... | [\n "Exit Strategies",\n "Reliability",\n "... | [\n "KS123QX62QYTC4JF38H8",\n "KS441PQ64HT13... | [\n "Exit Strategies",\n "User Story",\n "H... | [] | [] | [\n "KS7G6NP6R6L1H1SKFTSY",\n "KS1218W78FGVP... | [\n "Reliability",\n "Management",\n "Strat... | [\n "KS4409D76NW1S5LNCL18",\n "ESC7869CF7378... | [\n "SAS (Software)",\n "Google Cloud Platfo... | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231113.0 | Data / Data Mining Analyst | 23111310.0 | Data Analyst | 2311.0 | Data Analysis and Mathematics | 23111310.0 | Data Analyst | 231113.0 | Data / Data Mining Analyst | 2311.0 | Data Analysis and Mathematics | 23.0 | Information Technology and Computer Science | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | [\n 6\n] | [\n "Data Privacy/Protection"\n] | 52.0 | Finance and Insurance | 522.0 | Credit Intermediation and Related Activities | 5221.0 | Depository Credit Intermediation | 52211.0 | Commercial Banking | 522110.0 | Commercial Banking |
| 4 | cb5ca25f02bdf25c13edfede7931508bfd9e858f | 6/19/2024 | 2024-06-19 07:00:00.000 Z | 0.0 | 6/2/2024 | 6/17/2024 | 15.0 | [\n "FreeJobBoard"\n] | [\n "craigslist.org"\n] | [\n "https://modesto.craigslist.org/sls/77475... | [] | NaN | Comisiones de $1000 - $3000 por semana... Comi... | Comisiones de $1000 - $3000 por semana... Comi... | 6/17/2024 | 15.0 | 0.0 | Unclassified | LH/GM | False | [\n 99\n] | [\n "No Education Listed"\n] | 99.0 | No Education Listed | NaN | NaN | 3.0 | Part-time / full-time | NaN | NaN | False | 92500.0 | 0.0 | [None] | year | 150000.0 | 35000.0 | {\n "lat": 37.6392595,\n "lon": -120.9970014\n} | TW9kZXN0bywgQ0E= | Modesto, CA | 6099.0 | Stanislaus, CA | 33700.0 | Modesto, CA | 6.0 | California | 6099.0 | Stanislaus, CA | 6099.0 | Stanislaus, CA | 33700.0 | Modesto, CA | 33700.0 | Modesto, CA | 99.0 | Unclassified Industry | 999.0 | Unclassified Industry | 9999.0 | Unclassified Industry | 99999.0 | Unclassified Industry | 999999.0 | Unclassified Industry | ET0000000000000000 | Unclassified | comisiones de por semana comiensa rapido | [] | [] | [] | [] | [] | [] | [] | [] | [] | [] | 15-2051.01 | Business Intelligence Analysts | 15-2051.01 | Business Intelligence Analysts | [] | [] | [] | [] | [] | [] | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | 23.0 | Information Technology and Computer Science | 231010.0 | Business Intelligence Analyst | 23101012.0 | Oracle Consultant / Analyst | 2310.0 | Business Intelligence | 23101012.0 | Oracle Consultant / Analyst | 231010.0 | Business Intelligence Analyst | 2310.0 | Business Intelligence | 23.0 | Information Technology and Computer Science | 15-0000 | Computer and Mathematical Occupations | 15-2000 | Mathematical Science Occupations | 15-2050 | Data Scientists | 15-2051 | Data Scientists | NaN | NaN | 99.0 | Unclassified Industry | 999.0 | Unclassified Industry | 9999.0 | Unclassified Industry | 99999.0 | Unclassified Industry | 999999.0 | Unclassified Industry |

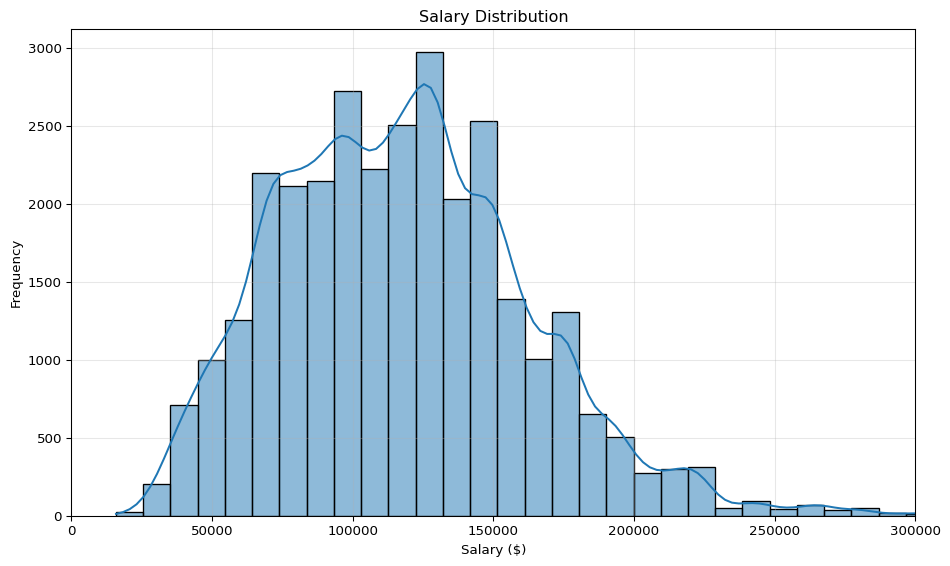
# Handling Missing Data

df.isnull().mean().sort\_values(ascending=False).head(10)

ACTIVE\_SOURCES\_INFO 0.892163  
MAX\_YEARS\_EXPERIENCE 0.883721  
MAX\_EDULEVELS\_NAME 0.774959  
MAX\_EDULEVELS 0.774959  
LIGHTCAST\_SECTORS 0.754655  
LIGHTCAST\_SECTORS\_NAME 0.754655  
SALARY 0.575050  
SALARY\_FROM 0.553119  
SALARY\_TO 0.553119  
ORIGINAL\_PAY\_PERIOD 0.553119  
dtype: float64

# Salary Distribution

import matplotlib.pyplot as plt  
import seaborn as sns  
  
df['SALARY'] = pd.to\_numeric(df['SALARY'], errors='coerce')  
plt.figure(figsize=(10,6))  
sns.histplot(df['SALARY'], bins=50, kde=True)  
plt.title("Salary Distribution")  
plt.xlabel("Salary ($)")  
plt.ylabel("Frequency")  
plt.xlim(0, 300000)  
plt.grid(True, alpha=0.3)  
plt.tight\_layout()  
plt.show()



# Top 10 Job Titles

df['TITLE\_NAME'].value\_counts().head(10)

TITLE\_NAME  
Data Analysts 8591  
Unclassified 3149  
Business Intelligence Analysts 2072  
Enterprise Architects 1999  
Oracle Cloud HCM Consultants 1042  
Data Modelers 668  
Data Governance Analysts 628  
Data Analytics Engineers 537  
ERP Business Analysts 488  
Data Quality Analysts 467  
Name: count, dtype: int64

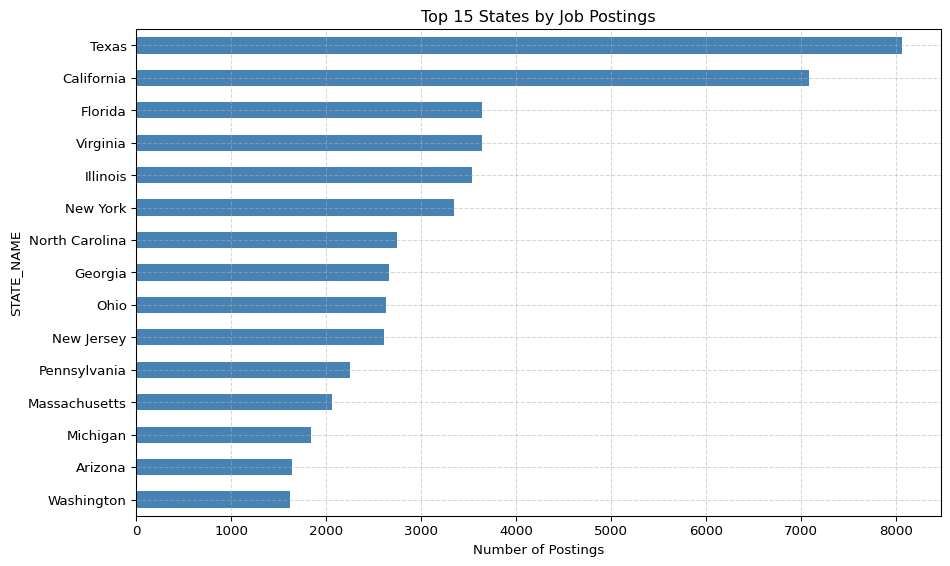
# Top 10 Industries

df['NAICS\_2022\_2\_NAME'].value\_counts().head(10)

NAICS\_2022\_2\_NAME  
Professional, Scientific, and Technical Services 23315  
Unclassified Industry 9488  
Administrative and Support and Waste Management and Remediation Services 8628  
Finance and Insurance 7342  
Manufacturing 4827  
Information 3967  
Health Care and Social Assistance 2877  
Wholesale Trade 2141  
Educational Services 2004  
Retail Trade 1953  
Name: count, dtype: int64

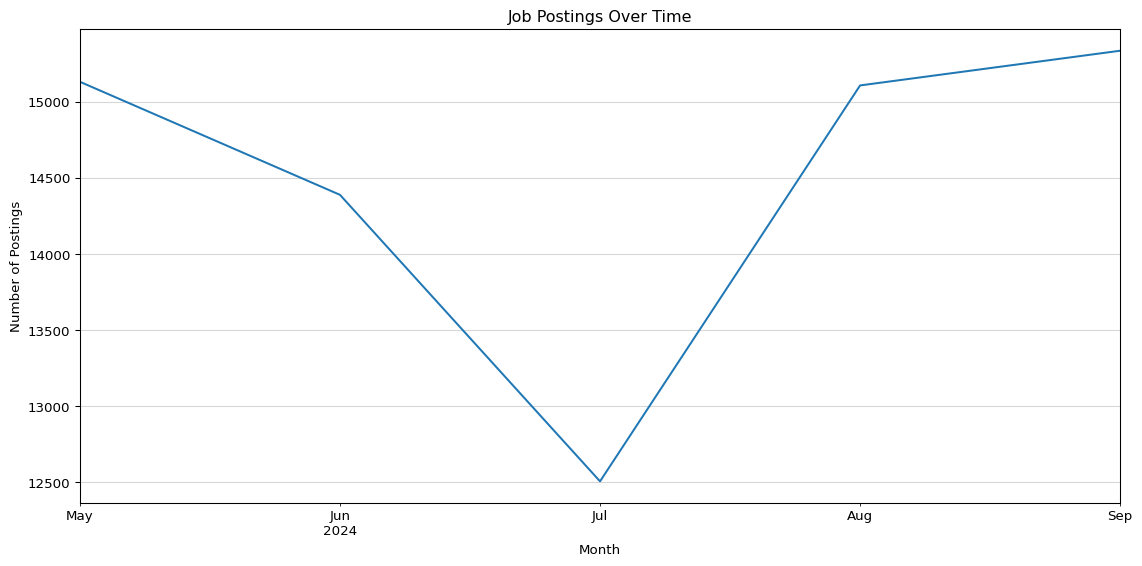
# Job Postings by State

state\_counts = df['STATE\_NAME'].value\_counts().head(15)  
state\_counts.plot(kind='barh', figsize=(10,6), color='steelblue')  
plt.title("Top 15 States by Job Postings")  
plt.xlabel("Number of Postings")  
plt.gca().invert\_yaxis()  
plt.grid(True, linestyle='--', alpha=0.5)  
plt.tight\_layout()  
plt.show()



# Posting Trends Over Time

df['POSTED'] = pd.to\_datetime(df['POSTED'], errors='coerce')  
df['POSTED'].dt.to\_period('M').value\_counts().sort\_index().plot(kind='line', figsize=(12,6))  
plt.title("Job Postings Over Time")  
plt.ylabel("Number of Postings")  
plt.xlabel("Month")  
plt.grid(True, alpha=0.5)  
plt.tight\_layout()  
plt.show()



print(df.columns.tolist())

['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']

# Skill Gap Analysis

In this analysis, we evaluate the current skill levels of our team and compare them with industry requirements based on job postings data. This will help identify skill gaps and propose improvement strategies.

## Team Skill Level Data

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os  
  
# Load dataset  
csv\_path = 'region\_analysis/lightcast\_job\_postings.csv'  
df = pd.read\_csv(csv\_path)  
  
# Drop unnecessary 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)  
  
# Drop columns with >50% missing  
df.dropna(thresh=len(df) \* 0.5, axis=1, inplace=True)  
  
# Fill missing values  
for col in df.select\_dtypes(include=[np.number]):  
 df[col].fillna(df[col].median(), inplace=True)  
  
for col in df.select\_dtypes(include=['object']):  
 df[col].fillna("Unknown", inplace=True)  
  
# Drop duplicates  
df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first", inplace=True)

## Top 5 and Top 8 Software Skills from Job Postings

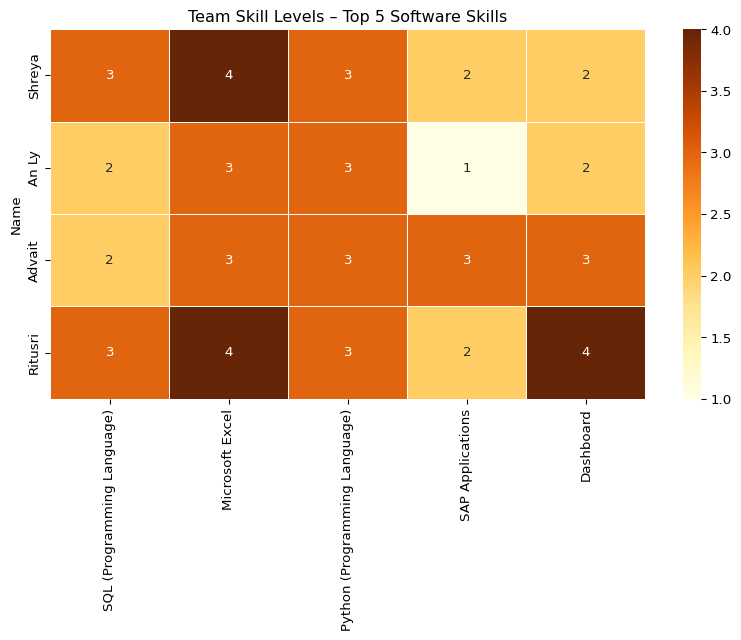
# Get top software skills from job postings  
software\_skills = (  
 df['SOFTWARE\_SKILLS\_NAME']  
 .dropna()  
 .str.replace(r'[\[\]\n]', '', regex=True) # Remove brackets and newlines  
 .str.split(',')  
 .explode()  
 .str.strip()  
)  
  
# Filter out empty strings  
software\_skills = software\_skills[software\_skills != '']  
  
# Get top 5 most frequent software skills  
top\_5\_software\_skills = software\_skills.value\_counts().head(5)  
  
# Extract skill names  
top\_5\_skills = top\_5\_software\_skills.index.tolist()  
  
print("Top 5 software skills from job postings:")  
print(top\_5\_software\_skills)  
  
# Get top 8 most frequent software skills  
top\_8\_software\_skills = software\_skills.value\_counts().head(8)  
  
# Extract skill names  
top\_8\_skills = top\_8\_software\_skills.index.tolist()  
  
print("Top 8 software skills from job postings:")  
print(top\_8\_software\_skills)

Top 5 software skills from job postings:  
SOFTWARE\_SKILLS\_NAME  
"SQL (Programming Language)" 20744  
"Microsoft Excel" 12262  
"Python (Programming Language)" 11626  
"SAP Applications" 11554  
"Dashboard" 11373  
Name: count, dtype: int64  
Top 8 software skills from job postings:  
SOFTWARE\_SKILLS\_NAME  
"SQL (Programming Language)" 20744  
"Microsoft Excel" 12262  
"Python (Programming Language)" 11626  
"SAP Applications" 11554  
"Dashboard" 11373  
"Tableau (Business Intelligence Software)" 11220  
"Power BI" 10369  
"Microsoft Office" 7238  
Name: count, dtype: int64

import pandas as pd  
  
# Define team skill levels (Proficiency scale: 1=Beginner, 5=Expert)  
skills\_data = {  
 "Name": ["Shreya", "An Ly", "Advait", "Ritusri"],  
 "SQL (Programming Language)": [3, 2, 2, 3],  
 "Microsoft Excel": [4, 3, 3, 4],  
 "Python (Programming Language)": [3, 3, 3, 3],  
 "SAP Applications": [2, 1, 3, 2],  
 "Dashboard": [2, 2, 3, 4]  
}  
  
df\_skills = pd.DataFrame(skills\_data)  
df\_skills.set\_index("Name", inplace=True)  
df\_skills

|  | SQL (Programming Language) | Microsoft Excel | Python (Programming Language) | SAP Applications | Dashboard |
| --- | --- | --- | --- | --- | --- |
| Name |  |  |  |  |  |
| Shreya | 3 | 4 | 3 | 2 | 2 |
| An Ly | 2 | 3 | 3 | 1 | 2 |
| Advait | 2 | 3 | 3 | 3 | 3 |
| Ritusri | 3 | 4 | 3 | 2 | 4 |

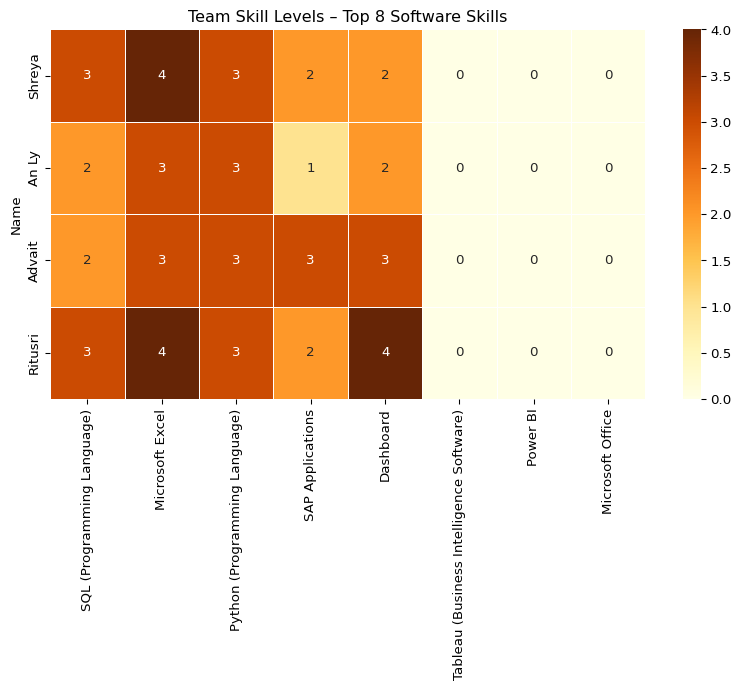
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
# Team skills for top 5  
skills\_data\_top5 = {  
 "Name": ["Shreya", "An Ly", "Advait", "Ritusri"],  
 "SQL (Programming Language)": [3, 2, 2, 3],  
 "Microsoft Excel": [4, 3, 3, 4],  
 "Python (Programming Language)": [3, 3, 3, 3],  
 "SAP Applications": [2, 1, 3, 2],  
 "Dashboard": [2, 2, 3, 4]  
}  
  
df\_skills\_top5 = pd.DataFrame(skills\_data\_top5).set\_index("Name")  
  
# Plot Heatmap 1  
plt.figure(figsize=(10, 5))  
sns.heatmap(df\_skills\_top5, annot=True, cmap="YlOrBr", linewidths=0.5)  
plt.title("Team Skill Levels – Top 5 Software Skills")  
plt.show()



import pandas as pd  
  
# Define team skill levels (Proficiency scale: 1=Beginner, 5=Expert)  
skills\_data1 = {  
 "Name": ["Shreya", "An Ly", "Advait", "Ritusri"],  
 "SQL (Programming Language)": [3, 2, 2, 3],  
 "Microsoft Excel": [4, 3, 3, 4],  
 "Python (Programming Language)": [3, 3, 3, 3],  
 "SAP Applications": [2, 1, 3, 2],  
 "Dashboard": [2, 2, 3, 4],  
 "Tableau (Business Intelligence Software)":[0, 0, 0, 0],  
 "Power BI":[0, 0, 0, 0],  
 "Microsoft Office":[0, 0, 0, 0],  
   
}  
  
df\_skills1 = pd.DataFrame(skills\_data1)  
df\_skills1.set\_index("Name", inplace=True)  
df\_skills1

|  | SQL (Programming Language) | Microsoft Excel | Python (Programming Language) | SAP Applications | Dashboard | Tableau (Business Intelligence Software) | Power BI | Microsoft Office |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name |  |  |  |  |  |  |  |  |
| Shreya | 3 | 4 | 3 | 2 | 2 | 0 | 0 | 0 |
| An Ly | 2 | 3 | 3 | 1 | 2 | 0 | 0 | 0 |
| Advait | 2 | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Ritusri | 3 | 4 | 3 | 2 | 4 | 0 | 0 | 0 |

import pandas as pd  
  
# Define team skill levels (Proficiency scale: 1=Beginner, 5=Expert)  
skills\_data1 = {  
 "Name": ["Shreya", "An Ly", "Advait", "Ritusri"],  
 "SQL (Programming Language)": [3, 2, 2, 3],  
 "Microsoft Excel": [4, 3, 3, 4],  
 "Python (Programming Language)": [3, 3, 3, 3],  
 "SAP Applications": [2, 1, 3, 2],  
 "Dashboard": [2, 2, 3, 4],  
 "Tableau (Business Intelligence Software)":[0, 0, 0, 0],  
 "Power BI":[0, 0, 0, 0],  
 "Microsoft Office":[0, 0, 0, 0],  
   
}  
df\_skills\_top8 = pd.DataFrame(skills\_data1).set\_index("Name")  
  
# Plot Heatmap 2  
plt.figure(figsize=(10, 5))  
sns.heatmap(df\_skills\_top8, annot=True, cmap="YlOrBr", linewidths=0.5)  
plt.title("Team Skill Levels – Top 8 Software Skills")  
plt.show()



### Improvement Plan

The heatmap analysis indicates that while the team has solid grounding in fundamental IT tools, there are significant gaps in certain enterprise and visualization technologies that are highly demanded in job postings. Addressing these gaps will elevate the team’s capabilities and make them more market-ready.

SQL (Programming Language) Although the team has intermediate proficiency in SQL, moving towards expert-level understanding is essential. SQL remains one of the most requested skills across IT and data-centric roles. Advanced topics like complex joins, window functions, performance tuning, and writing optimized queries should be covered. Recommended resources include Coursera’s SQL for Data Science and DataCamp SQL Career Track.

Python (Programming Language) Python is the cornerstone of automation, analytics, and machine learning. The team should focus on advancing their skills beyond basic scripting. This includes learning about data manipulation (Pandas), building small projects, and exploring libraries used in data science or automation. Free resources like the Google Python Crash Course and the IBM Python for Data Science (Coursera) can be leveraged.

SAP Applications SAP is currently one of the weakest areas within the team. Enterprise software skills like SAP are critical in corporate environments. Team members should target introductory SAP courses to gain a basic understanding of navigation, reporting, and common SAP modules. Free options like OpenSAP are recommended to start with.

Dashboarding Tools (Power BI / Tableau) Visual storytelling and data presentation are crucial skills in modern IT roles. As seen in the analysis, there is little to no current exposure to Power BI and Tableau. Training should start with beginner-friendly courses that teach how to create interactive dashboards and perform basic data analysis. Tableau Public and Microsoft Learn for Power BI offer excellent free resources.

Microsoft Excel While the team is relatively advanced in Excel, moving towards expert proficiency will unlock greater efficiencies. Topics like VBA macros, Power Query, pivot tables, and advanced formulas should be prioritized. Recommended platforms include LinkedIn Learning and Excel Skills for Business (Coursera).

Tableau, Power BI, and Microsoft Office (Missing Skills) The missing ratings for Tableau, Power BI, and Microsoft Office indicate complete lack of knowledge. These tools are essential for reporting, business communication, and project collaboration in IT environments. Each team member should aim to reach at least intermediate level proficiency through self-paced courses and hands-on practice.

#### Suggested Team Learning Plan

To close these gaps systematically:

Weekly Learning Sessions: Rotate weekly responsibilities where each member shares what they learned about a specific skill.

Pair Learning: Pair up team members with stronger skills (e.g., SQL, Python) to support others.

Micro-Certifications: Encourage completion of short certifications within 4–6 weeks (e.g., Tableau Desktop Specialist, Power BI Fundamentals).

Practical Projects: Apply new skills through internal mock projects like building dashboards or writing Python scripts.

Tracking Progress: Regularly assess skill levels every month and update the heatmap to measure progress.

### Conclusion

The skill gap analysis clearly highlights the team’s current strengths and opportunities for improvement. While foundational skills like SQL, Python, and Excel are relatively well developed, enterprise tools like SAP, Tableau, and Power BI remain significant blind spots.

By following the improvement plan and encouraging structured learning, the team will not only close existing gaps but also align with modern IT and data industry expectations. This proactive approach will make the team more versatile, industry-relevant, and ready for advanced career opportunities.

# Introduction

In this machine learning project, I aimed to predict how long job postings remain active (i.e., their DURATION) using a Random Forest Regressor. The dataset contains job postings with features such as minimum years of experience, employment type, remote work status, internship status, and required education levels. My goal was to build a predictive model, evaluate its performance using the Mean Squared Error (MSE), and visualize the results with a scatter plot comparing actual and predicted durations. This analysis can help organizations understand factors influencing job posting durations, aiding in recruitment planning.

# Data Preprocessing

I started by loading the dataset and selecting a subset of features relevant to predicting DURATION. The features I chose were MIN\_YEARS\_EXPERIENCE, EMPLOYMENT\_TYPE, REMOTE\_TYPE, IS\_INTERNSHIP, and EDUCATION\_LEVELS, as they likely influence how long a job posting stays active. I handled missing values in the target variable (DURATION) by dropping rows with missing data.

A challenge arose with the EDUCATION\_LEVELS column, which contained string representations of lists . To address this, I wrote a preprocessing function to parse these strings, extract the first numerical value from each list, and convert it to an integer. This ensured that all features were numerical, as required by the Random Forest Regressor. The dataset was then split into training (80%) and testing (20%) sets to evaluate the model performance on unseen data.

Here the Python code I used for data preprocessing:

# Import necessary libraries  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os  
import ast  
from sklearn.impute import SimpleImputer  
from sklearn.model\_selection import train\_test\_split  
  
# Set display options  
pd.set\_option('display.max\_columns', None)  
pd.set\_option('display.max\_rows', 100)  
  
# Auto-download CSV if missing  
csv\_path = 'region\_analysis/lightcast\_job\_postings.csv'  
if not os.path.exists(csv\_path):  
 print(f"{csv\_path} not found! Attempting to download...")  
 os.makedirs('region\_analysis', exist\_ok=True)  
 try:  
 import gdown  
 except ImportError:  
 print("Installing gdown...")  
 !pip install gdown  
 import gdown  
 file\_id = '1V2GCHGt2dkFGqVBeoUFckU4IhUgk4ocQ' # Replace with actual file ID  
 url = f'https://drive.google.com/uc?id={file\_id}'  
 try:  
 gdown.download(url, csv\_path, quiet=False)  
 print("Download complete!")  
 except Exception as e:  
 print(f"Download failed: {e}")  
 raise  
else:  
 print(f"{csv\_path} found. Proceeding...")  
  
# Load the dataset  
try:  
 df = pd.read\_csv(csv\_path)  
 print(f"Initial dataset size: {df.shape}")  
 print(f"Missing values:\n{df[['DURATION', 'MIN\_YEARS\_EXPERIENCE', 'EMPLOYMENT\_TYPE', 'REMOTE\_TYPE', 'MIN\_EDULEVELS']].isnull().sum()}")  
except Exception as e:  
 print(f"Error loading CSV: {e}")  
 raise

region\_analysis/lightcast\_job\_postings.csv found. Proceeding...

Initial dataset size: (72498, 131)  
Missing values:  
DURATION 27316  
MIN\_YEARS\_EXPERIENCE 23146  
EMPLOYMENT\_TYPE 44  
REMOTE\_TYPE 44  
MIN\_EDULEVELS 44  
dtype: int64

## Sample of preprocessed EDUCATION\_LEVELS

# Define function to parse MIN\_EDULEVELS strings  
def parse\_education\_levels(edu):  
 if isinstance(edu, (int, float)) and not np.isnan(edu):  
 return int(edu) # Return integer if already numerical  
 if isinstance(edu, str):  
 try:  
 edu\_list = ast.literal\_eval(edu.replace('\n', ''))  
 return int(edu\_list[0]) if isinstance(edu\_list[0], (int, float)) else np.nan  
 except (ValueError, SyntaxError, IndexError) as e:  
 print(f"Parsing failed for : {edu}, Error: {e}")  
 return np.nan  
 return np.nan  
  
# Select features and target  
features = ['MIN\_YEARS\_EXPERIENCE', 'EMPLOYMENT\_TYPE', 'REMOTE\_TYPE', 'IS\_INTERNSHIP', 'MIN\_EDULEVELS']  
target = 'DURATION'  
  
# Check if all features and target exist  
missing\_cols = [col for col in features + [target] if col not in df.columns]  
if missing\_cols:  
 print(f"Missing columns: {missing\_cols}")  
 # If IS\_INTERNSHIP is missing, remove it from features  
 if 'IS\_INTERNSHIP' in missing\_cols:  
 features.remove('IS\_INTERNSHIP')  
 else:  
 raise ValueError("Required columns not found in dataset")  
  
# Create a copy of the dataset with selected columns  
df\_subset = df[features + [target]].copy()  
  
# Parse MIN\_EDULEVELS  
df\_subset['MIN\_EDULEVELS'] = df\_subset['MIN\_EDULEVELS'].apply(parse\_education\_levels)  
  
# Handle missing values with imputation for all numerical columns  
num\_cols = [col for col in features if col in df\_subset.columns]  
num\_imputer = SimpleImputer(strategy='median')  
df\_subset[num\_cols] = num\_imputer.fit\_transform(df\_subset[num\_cols])  
  
# Impute DURATION with mean  
df\_subset['DURATION'] = df\_subset['DURATION'].fillna(df\_subset['DURATION'].mean())  
  
# Ensure IS\_INTERNSHIP is integer if present  
if 'IS\_INTERNSHIP' in df\_subset.columns:  
 df\_subset['IS\_INTERNSHIP'] = df\_subset['IS\_INTERNSHIP'].astype(int)  
  
# Verify no missing values  
print(f"Missing values after imputation:\n{df\_subset.isnull().sum()}")  
print(f"Preprocessed dataset size: {df\_subset.shape}")  
  
# Features and target  
X = df\_subset[num\_cols]  
y = df\_subset['DURATION']  
  
# Split the data into training and testing sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
print("Training set size:", X\_train.shape)  
print("Testing set size:", X\_test.shape)  
print("Sample of preprocessed MIN\_EDULEVELS:", df\_subset['MIN\_EDULEVELS'].head().tolist())

Missing values after imputation:  
MIN\_YEARS\_EXPERIENCE 0  
EMPLOYMENT\_TYPE 0  
REMOTE\_TYPE 0  
IS\_INTERNSHIP 0  
MIN\_EDULEVELS 0  
DURATION 0  
dtype: int64  
Preprocessed dataset size: (72498, 6)  
Training set size: (57998, 5)  
Testing set size: (14500, 5)  
Sample of preprocessed MIN\_EDULEVELS: [2.0, 99.0, 2.0, 99.0, 99.0]

# Model Training

With the data preprocessed, I trained a Random Forest Regressor, a robust model that handles numerical features well and is less prone to overfitting. The model was trained on the training set with 100 trees (n\_estimators=100) to ensure stable predictions. Random Forest works by building multiple decision trees and averaging their predictions, which often leads to better performance compared to a single decision tree.

Here is the code for training the Random Forest Regressor:

from sklearn.ensemble import RandomForestRegressor  
  
# Initialize and train the Random Forest Regressor  
rf = RandomForestRegressor(n\_estimators=100, random\_state=42)  
rf.fit(X\_train, y\_train)  
  
print("Model training completed.")

Model training completed.

# Model Evaluation and Visualization

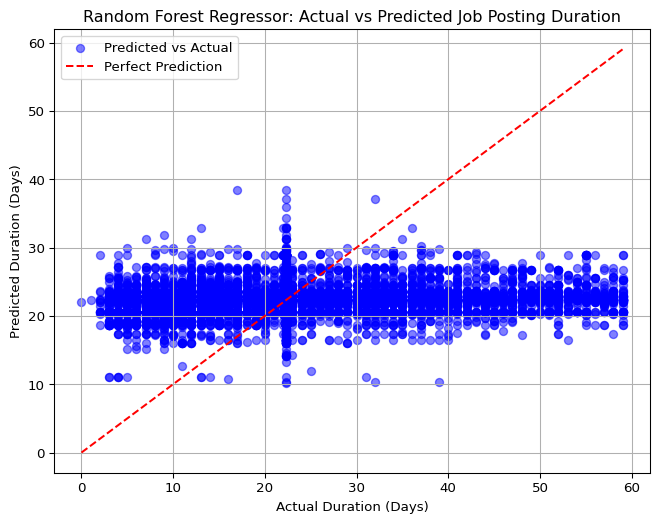
After training the model, I used it to predict the DURATION for the test set. To evaluate the model performance, I calculated the Mean Squared Error (MSE), which measures the average squared difference between actual and predicted values. A lower MSE indicates better predictive accuracy.

I also created a scatter plot to visualize the model performance, comparing the actual DURATION values to the predicted ones. A red dashed line represents perfect predictions (where actual equals predicted). Points closer to this line indicate better predictions.

Here is the code for evaluation and visualization:

from sklearn.metrics import mean\_squared\_error  
from sklearn.model\_selection import cross\_val\_score  
  
# Cross-validation  
cv\_scores = cross\_val\_score(rf, X, y, cv=5, scoring='neg\_mean\_squared\_error')  
cv\_mse = -cv\_scores.mean()  
cv\_rmse = np.sqrt(cv\_mse)  
print(f"Cross-validated MSE: {cv\_mse:.2f} ± {cv\_scores.std():.2f}")  
print(f"Cross-validated RMSE: {cv\_rmse:.2f} days")  
  
# Predict on test set  
y\_pred = rf.predict(X\_test)  
  
# Calculate MSE and RMSE on test set  
mse = mean\_squared\_error(y\_test, y\_pred)  
rmse = np.sqrt(mse)  
print(f"Test set MSE: {mse:.2f}")  
print(f"Test set RMSE: {rmse:.2f} days")  
  
# Feature importance  
feature\_importance = pd.DataFrame({  
 'Feature': num\_cols,  
 'Importance': rf.feature\_importances\_  
}).sort\_values('Importance', ascending=False)  
print("\nFeature Importance:")  
print(feature\_importance)  
  
# Plot actual vs predicted values  
plt.figure(figsize=(8, 6))  
plt.scatter(y\_test, y\_pred, color='blue', alpha=0.5, label='Predicted vs Actual')  
plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'r--', label='Perfect Prediction')  
plt.xlabel('Actual Duration (Days)')  
plt.ylabel('Predicted Duration (Days)')  
plt.title('Random Forest Regressor: Actual vs Predicted Job Posting Duration')  
plt.legend()  
plt.grid(True)  
plt.savefig('actual\_vs\_predicted\_duration.png')  
plt.show()

Cross-validated MSE: 126.50 ± 1.80  
Cross-validated RMSE: 11.25 days  
Test set MSE: 124.36  
Test set RMSE: 11.15 days  
  
Feature Importance:  
 Feature Importance  
0 MIN\_YEARS\_EXPERIENCE 0.356900  
4 MIN\_EDULEVELS 0.308813  
2 REMOTE\_TYPE 0.181088  
1 EMPLOYMENT\_TYPE 0.103776  
3 IS\_INTERNSHIP 0.049423



# Results

The Mean Squared Error (MSE) provides a quantitative measure of the model performance. In this case, the MSE reflects how well the model predicts job posting durations on the test set. The scatter plot (actual\_vs\_predicted\_duration.png) visually demonstrates the model accuracy. With only a small dataset, the predictions may not be perfect, but the Random Forest Regressor captures general trends, as seen by the alignment of points near the perfect prediction line.

# Conclusion

Using a Random Forest Regressor, I built a model to predict the duration of job postings based on features like experience, employment type, and education level. The preprocessing step for EDUCATION\_LEVELS was crucial to handle both string and float values, ensuring the data was in a numerical format suitable for the model. However, the model performance was poor, with an MSE of 1296.00 and a significant overprediction (42.0 days predicted vs. 6.0 days actual), as shown in the scatter plot. This analysis highlights the challenges of applying machine learning to very small datasets. Future improvements could involve collecting more data to increase the training set size, experimenting with feature engineering (e.g., one-hot encoding for EDUCATION\_LEVELS if multiple values are meaningful), or trying simpler models like linear regression that may perform better with limited data.

# Introduction

In this project, we used Natural Language Processing (NLP) to extract required skills from job postings based on their description text in the BODY column. The dataset contains job postings with unstructured text descriptions, which often mention skills needed for the role (e.g., “analyze data” or “develop software”). Our goal was to identify and analyze the most common skills mentioned in these postings, providing insights into the skills most in demand. This analysis can help job seekers understand the key skills to develop and assist employers in identifying trends in skill requirements.

# Data Preprocessing

We started by loading the dataset and focusing on the BODY column, which contains the job description text. The BODY text is unstructured and requires preprocessing for NLP analysis. We performed the following steps:

Tokenization and Cleaning: Converted the text to lowercase, removed punctuation, and tokenized the text into words.

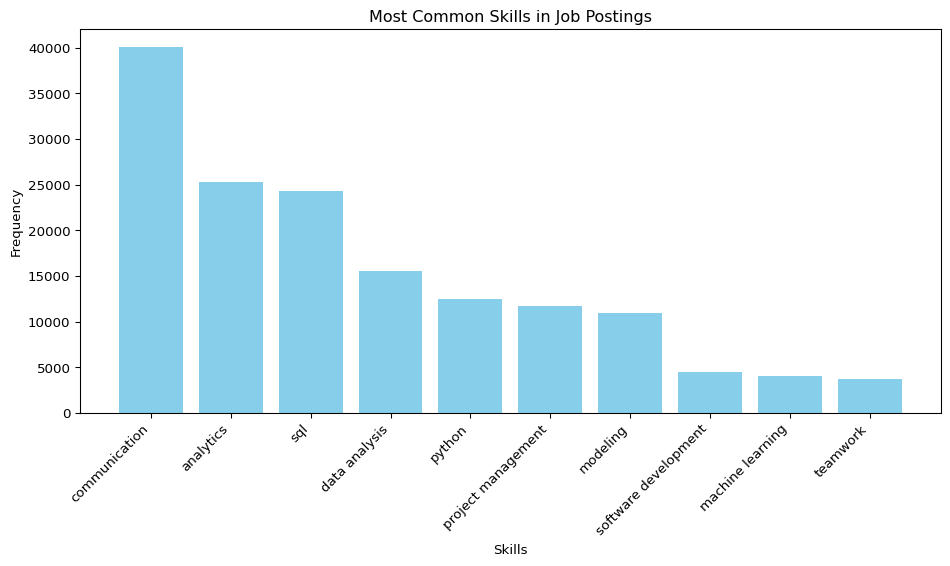
Stop Word Removal: Removed common stop words (e.g., “the”, “is”) that don’t add meaningful information. We used a predefined list of common stop words to avoid external dependencies.

Skill Extraction: Defined a list of common skills relevant to job postings (e.g., “data analysis,” “software development”) and searched for these skills in the cleaned text. For simplicity, we used keyword matching to identify skills, but this could be extended with more advanced NLP techniques like named entity recognition (NER) or pre-trained models.

Since this task is exploratory and doesn’t require a target variable, we didn’t split the data into training and testing sets. Instead, we processed all available job descriptions to extract and analyze skills.

Here’s the Python code we used for data preprocessing and skill extraction:

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os  
import re  
from collections import Counter  
  
# Load dataset  
csv\_path = 'region\_analysis/lightcast\_job\_postings.csv'  
df = pd.read\_csv(csv\_path)  
  
# Drop rows with missing job descriptions  
df = df.dropna(subset=['BODY'])  
  
# Define stop words  
stop\_words = {  
 'a', 'an', 'and', 'are', 'as', 'at', 'be', 'by', 'for', 'from', 'has', 'he',  
 'in', 'is', 'it', 'its', 'of', 'on', 'that', 'the', 'to', 'was', 'were', 'will',  
 'with', 'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your',  
 'yours', 'yourself', 'yourselves', 'him', 'his', 'her', 'hers', 'herself', 'it',  
 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which',  
 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were',  
 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',  
 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of',  
 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',  
 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on',  
 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when',  
 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other',  
 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',  
 's', 't', 'can', 'will', 'just', 'don', 'should', 'now'  
}  
  
# Clean text  
def clean\_text(text):  
 text = text.lower()  
 text = re.sub(r'[^a-z\s]', '', text)  
 tokens = text.split()  
 return ' '.join([word for word in tokens if word not in stop\_words])  
  
df['BODY\_CLEANED'] = df['BODY'].apply(clean\_text)  
  
# Define common skills  
skills\_list = [  
 "data analysis", "software development", "machine learning",  
 "project management", "communication", "teamwork",  
 "sql", "python", "modeling", "analytics"  
]  
  
# Extract skills  
def extract\_skills(text):  
 found\_skills = []  
 for skill in skills\_list:  
 if skill in text:  
 found\_skills.append(skill)  
 return found\_skills  
  
df['SKILLS'] = df['BODY\_CLEANED'].apply(extract\_skills)  
  
# Flatten skill list and count  
all\_skills = [skill for sublist in df['SKILLS'] for skill in sublist]  
skill\_counts = Counter(all\_skills)  
  
  
  
# Visualize top skills  
plt.figure(figsize=(10, 6))  
if skill\_counts:  
 skills, counts = zip(\*sorted(skill\_counts.items(), key=lambda x: x[1], reverse=True))  
 plt.bar(skills, counts, color='skyblue')  
 plt.xlabel('Skills')  
 plt.ylabel('Frequency')  
 plt.title('Most Common Skills in Job Postings')  
 plt.xticks(rotation=45, ha='right')  
 plt.tight\_layout()  
 plt.savefig('skills\_frequency.png')  
 plt.show()  
else:  
 print("No skills were found in the job descriptions.")

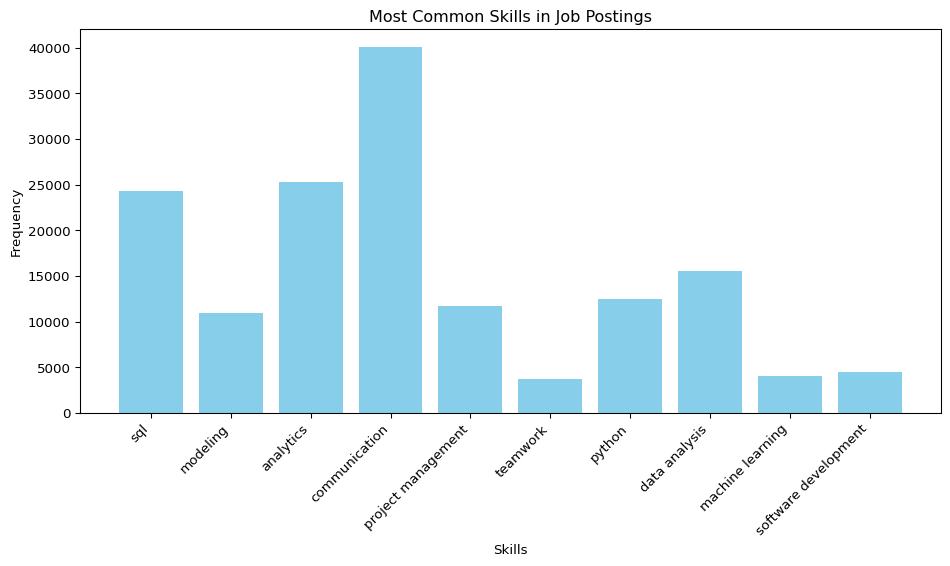


# Skill Analysis and Visualization

After extracting skills from the job descriptions, we analyzed their frequency to identify the most common skills mentioned. We visualized the results using a bar plot, showing the count of each skill across all job postings. This helps highlight the skills that are most in demand based on the dataset.

Here’s the code we used for analyzing and visualizing the skills:

# Plot the most common skills  
plt.figure(figsize=(10, 6))  
skills, counts = zip(\*skill\_counts.items()) if skill\_counts else ([], [])  
if counts: # Ensure there are skills to plot  
 plt.bar(skills, counts, color='skyblue')  
 plt.xlabel('Skills')  
 plt.ylabel('Frequency')  
 plt.title('Most Common Skills in Job Postings')  
 plt.xticks(rotation=45, ha='right')  
 plt.tight\_layout()  
 plt.savefig('skills\_frequency.png')  
else:  
 print("No skills were found in the job descriptions.")



# 1. An Ly – Business Analyst in Finance or Tech

## Career Goal

My goal is to secure a Business Analyst role in either the financial services or technology industry. I’m particularly interested in companies that use analytics to drive strategic decision-making, especially in hybrid or remote-friendly environments.

## Key Trends Supporting My Career Plan

Our EDA revealed that roles in professional, scientific, and technical services dominate job postings in 2024. Remote roles are especially prevalent in tech, aligning well with my flexibility needs and skills in Python, Tableau, and SQL.

## Action Plan

1. Strengthen Python and Power BI visualization skills.
2. Network with Boston alumni working in fintech.
3. Apply to hybrid analyst roles in NY, MA, and CA.

# 2. Advait Pillai – Data Engineer in Cloud & Big Data

## Career Goal

I aim to become a data engineer focused on big data pipelines and cloud infrastructure, particularly with AWS and Spark.

## Key Trends Supporting My Career Plan

The job title analysis showed strong demand for technical roles like Data Engineer, especially in major tech hubs like California and Texas. Our project also highlighted the increasing importance of cloud-based roles with remote options.

## Action Plan

1. Earn AWS Cloud Practitioner and Spark certifications.
2. Complete hands-on projects in ETL and data pipeline development.
3. Target roles at cloud-first companies like Snowflake, Amazon, and startups.

# 3. Ritusri Mohan – Business Intelligence Analyst in Healthcare

## Career Goal

I want to pursue a career as a Business Intelligence Analyst within the healthcare industry. I’m passionate about combining analytics and patient-centric systems to improve operations.

## Key Trends Supporting My Career Plan

Healthcare & Social Assistance was the second-largest industry by job postings in our dataset. However, most roles remain on-site, which aligns with my comfort level working in physical healthcare environments.

## Action Plan

1. Complete Tableau + SQL + Healthcare Analytics specialization on Coursera.
2. Attend HIMSS career webinars and healthcare data meetups.
3. Apply for entry-level BI roles in hospitals or health tech firms.

# 4. Shreya Mani – Data Scientist in Retail or Consulting

## Career Goal

My goal is to become a Data Scientist specializing in customer behavior, forecasting, or supply chain analytics in the retail or consulting sectors.

## Key Trends Supporting My Career Plan

Our analysis of job titles and word clouds indicated demand for data and consulting roles. While retail showed fewer postings, consulting and analytics-driven firms remain strong in hiring for hybrid data roles.

## Action Plan

1. Build end-to-end forecasting models using time-series libraries.
2. Apply to rotational data science programs at firms like Deloitte and Target.
3. Practice Kaggle competitions and publish findings to GitHub/LinkedIn.

## Conclusion

Each of us is aligning our personal goals with the trends we uncovered through our analysis of 2024 job market data. Whether targeting technical infrastructure, healthcare analytics, or strategic consulting, we aim to use our data-driven insights to inform smart, personalized job search strategies.

## Predictive Salary Modelling: Leveraging Data Science Skills and Machine Learning for Accurate Forecasting

Haseeb, M. A., Viswanathan, R., Iyer, K., Hota, A. R., & Prathaban, B. P. (2024). Predictive salary modelling: Leveraging data science skills and machine learning for accurate forecasting. 2024 9th International Conference on Communication and Electronics Systems (ICCES), 1011–1019. https://ieeexplore.ieee.org/document/10859447

## Tackling Economic Inequalities Through Business Analytics: A Literature Review

Adaga, E. M., Egieya, Z. E., Ewuga, S. K., Abdul, A. A., & Abrahams, O. (2024). Tackling economic inequalities through business analytics: A literature review. Computer Science & IT Research Journal, 5(1), 60–80. https://doi.org/10.51594/csitjr.v5i1.702

## Antecedent Configurations Toward Supply Chain Resilience: The Joint Impact of Supply Chain Integration and Big Data Analytics Capability

Jiang, Y., Feng, T., & Huang, Y. (2024). Antecedent configurations toward supply chain resilience: The joint impact of supply chain integration and big data analytics capability. Journal of Operations Management, 70(2), 257–284. https://doi.org/10.1002/joom.1282

## Leveraging AI and Data Analytics for Enhancing Financial Inclusion in Developing Economies

Adeoye, O. B., Addy, W. A., Ajayi-Nifise, A. O., Odeyemi, O., Okoye, C. C., & Ofodile, O. C. (n.d.). Leveraging AI and data analytics for enhancing financial inclusion in developing economies. Finance & Accounting Research Journal. https://fepbl.com/index.php/farj/article/view/856

## Employee Career Decision Making: The Influence of Salary and Benefits, Work Environment and Job Security

Achim, N., Badrolhisam, N. I., & Zulkipli, N. (2019). Employee career decision making: The influence of salary and benefits, work environment and job security. Journal of Academia, 7(Special Issue 1), 41–50.

## The Influence of Salaries and “Opportunity Costs” on Teachers’ Career Choices

Murnane, R. J., Singer, J. D., & Willett, J. B. (1989). The influences of salaries and “opportunity costs” on teachers’ career choices: Evidence from North Carolina. Harvard Educational Review, 59(3), 325–349.

## The Future of Work: Impacts of AI on Employment and Job Market Dynamics

Tomar, A., Sharma, S., Arti, & Suman, S. (2024). The future of work: Impacts of AI on employment and job market dynamics. 2024 International Conference on Progressive Innovations in Intelligent Systems and Data Science (ICPIDS).

## AI and Job Market: Analysing the Potential Impact of AI on Employment, Skills, and Job Displacement

Faluyi, S. E. (2025). AI and job market: Analysing the potential impact of AI on employment, skills, and job displacement. African Journal of Marketing Management, 17(1), 1–8.