Data Analysis

Comprehensive Data Cleaning & Exploratory Analysis of Job Market Trends

Jiapei Li

2025-04-29

|  |
| --- |
| Note |
| This is the final integrated report combining all sections: data cleaning, skill gap analysis, machine learning methods, and final insights. |

title: “Salary and Compensation Trends in AI Careers” author: “Team 3” date: today format: html: toc: true toc-depth: 2 toc-exclude: [“Welcome to Our Research”] bibliography: references.bib csl: csl/econometrica.csl

# Welcome to Our Research

## Research Rationale

Our team chose **Salary Disparities Across Disciplines: Who Benefits Most from AI’s Rise?**  as our research topic. Due to the rapid development of AI chatbots and substantial infrastructure investments by companies such as OpenAI, Google, NVIDIA, and DeepSeek, a significant restructuring of the current workforce appears inevitable. Some existing occupations will diminish or potentially disappear, while new roles will emerge and command higher compensation. Understanding these changes, identifying potential trends, and analyzing reliable data regarding AI’s impact are of paramount importance.

## Why is this topic important?

The rapid advancement of AI is transforming the workforce by reshaping job opportunities and salary structures. While AI-driven technologies enhance productivity and facilitate the emergence of new roles, they also contribute to job displacement, particularly in routine-based occupations.

Empirical studies have demonstrated that automation has been a key driver of income inequality (Adhikari (2024)), disproportionately impacting middle-skilled and less-educated workers. Conversely, AI-driven professions are experiencing new work opportunities, particularly in technology-related fields, necessitating a shift in workforce skills and adaptability.

The adoption of AI technologies brings **wage volatility and job opportunities** across different income groups. While AI roles often require unique, high-demand skillsets within specific occupations, it is overall important to implement policy measures to reduce the variations in industry adoption and workforce accessibility. As AI reshapes job requirements, reskilling becomes imperative, particularly for non-technical workers who are at higher risk of displacement. Addressing the digital skills gap through targeted training programs is essential to ensuring workforce adaptability and mitigating the socioeconomic disruptions caused by AI-driven labor market transformations.

## Trend Analysis of AI Development and Salary Fluctuations

The rapid advancement of AI has led to an increasing market demand for AI-related skills, significantly influencing employment trends and salary structures in the United States. Studies suggest that various job sectors are highly susceptible to AI-driven transformation, including those in traditionally high-skilled industries (Colombo et al. (2024)).

Similarly, in China, the demand for AI professionals surged in 2024, particularly in specialized domains such as **healthcare and applied sciences**, as companies sought to attract top talent in these fields. Salaries for AI specialists have risen substantially. Moreover, the growing competition in the AI sector has reshaped existing salary distributions. AI-related professions offer **higher salary premiums and employment benefits** compared to traditional IT engineering roles, potentially exacerbating the wage disparity between AI professionals and non-AI professionals, thereby influencing labor market equilibrium (Stone, Lukaszewski, and Johnson (2024)).

AI is fundamentally transforming labor markets across various industries, exerting a significant influence on **wage structures depending on job characteristics**. Repetitive rule-based tasks, such as **basic data entry or customer service automation**, are highly susceptible to AI-driven automation, reducing the demand for such roles in the labor market, thereby cutting down employment opportunities and lowering compensation obtained. For instance, AI has been widely adopted in **human resources management candidate screening**, prompting some employers to lessen workforce hiring needs (Sezgin (2023)).

Conversely, occupations requiring **advanced cognitive skills, creativity, or interpersonal communication**—such as those in **professional services**—are less likely to be fully replaced by AI and may instead benefit from hybrid automation models.

* **Medical field**: AI-powered diagnostic tools have enhanced precision and efficiency in medical decision-making, but final assessments still rely heavily on human expertise (Ansari and Ansari (2024)).
* **Legal sector**: AI-driven automation has accelerated case analysis and legal procedures, but **complex legal judgments continue to rely on human practitioners**. Harvard Law Professor **David Wilkins** has found that while generative AI has the potential to transform legal practices, the primary role remains that of a **supportive tool rather than a replacement**.
* **Finance and Governance**: Industries with a focus on **financial and governance decision-making** may experience **wage polarization** due to AI-induced advancements, where professionals may experience **wage growth** due to AI-driven productivity enhancements.

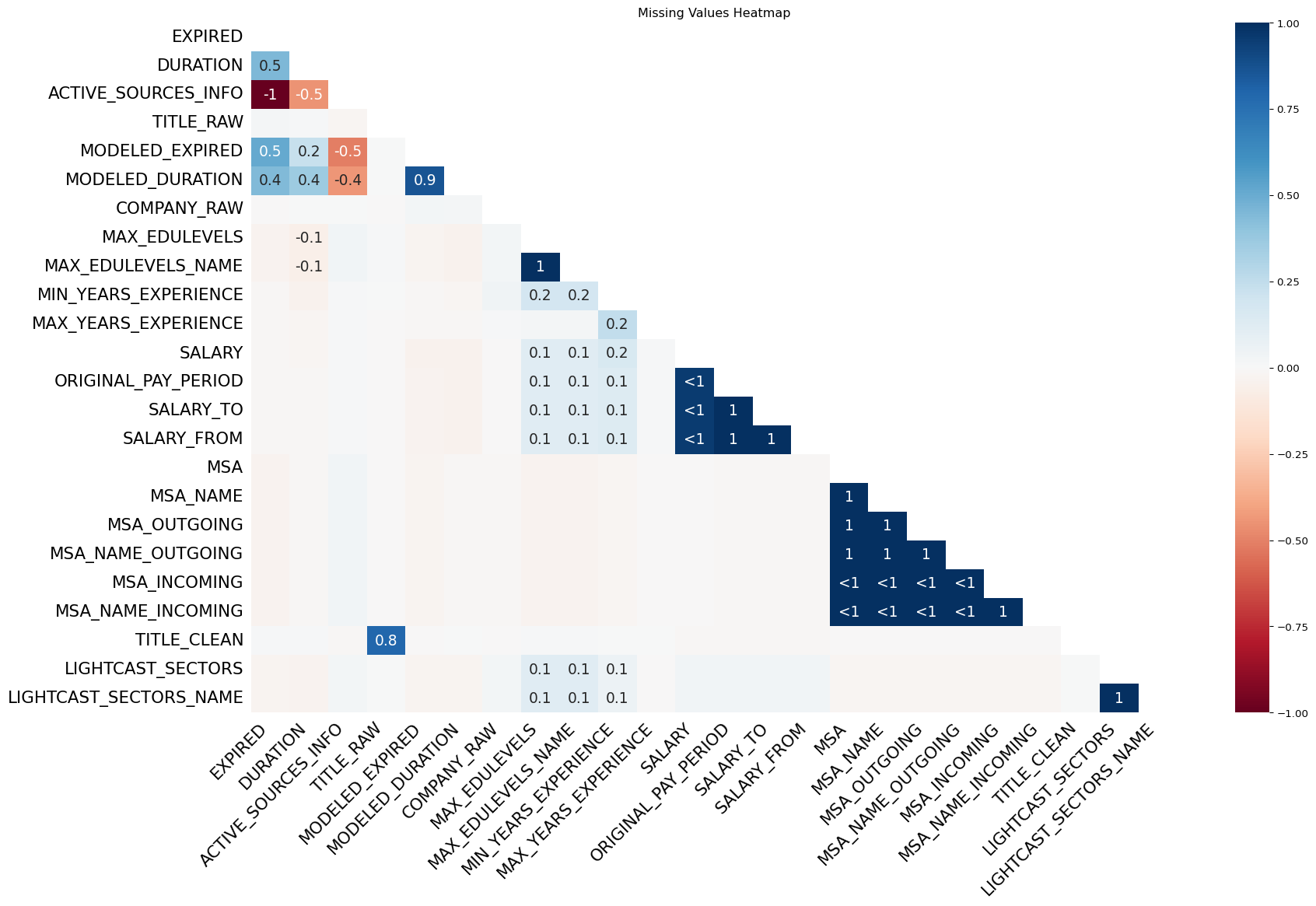
## What do you expect to find in your analysis?

Our research encompasses various aspects of **salary disparities**. Specifically, we aim to investigate: - **Income and job distribution differences**, including an analysis of which **geographical regions** see the highest demand and wage disparities for both AI-related and traditional professional roles. - **Comprehensive analysis of recent datasets**, identifying the most sought-after required skills among job listings. - **The differential impact of AI-driven changes on high-skill versus middle-skill occupations**.

## References

## Data Cleaning & Preprocessing

import pandas as pd  
import missingno as msno  
import matplotlib.pyplot as plt  
import plotly.express as px  
  
# 1. Read the CSV file  
df = pd.read\_csv("lightcast\_job\_postings.csv")  
  
# 2. Columns to drop (if they exist)  
columns\_to\_drop = [  
 "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",  
 "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",  
 "SOC\_2", "SOC\_3", "SOC\_5"  
]  
# Filter out columns that actually exist in the DataFrame to avoid KeyError  
valid\_cols\_to\_drop = [col for col in columns\_to\_drop if col in df.columns]  
df.drop(columns=valid\_cols\_to\_drop, inplace=True)  
  
# 3. Visualize missing values  
msno.heatmap(df)  
plt.title("Missing Values Heatmap")  
plt.show()  
  
# 4. Basic missing value processing  
df.dropna(thresh=len(df) \* 0.5, axis=1, inplace=True) # If a column has more than 50% missing values, delete it.  
  
# If there is a 'SALARY' column, fill missing values with the median.  
if "SALARY" in df.columns:  
 df["SALARY"].fillna(df["SALARY"].median(), inplace=True)  
# If there is an 'Industry' column, fill missing values with 'Unknown'.  
if "Industry" in df.columns:  
 df["Industry"].fillna("Unknown", inplace=True)  
  
# 5. Remove duplicates  
df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first", inplace=True)



### Job Postings by Industry

if "Industry" in df.columns:  
 fig = px.bar(df["Industry"].value\_counts(), title="Job Postings by Industry")  
 fig.show()

**Explanation:**  
This bar chart shows the number of job postings across different industries. It highlights which sectors are most active in recruiting, allowing us to identify high-demand areas such as technology, healthcare, or finance.

### Salary Distribution by Industry

if "SALARY" in df.columns and "Industry" in df.columns:  
 fig = px.box(df, x="Industry", y="SALARY", title="Salary Distribution by Industry")  
 fig.show()

**Explanation:**  
This box plot illustrates how salary ranges differ across industries. It helps reveal not only the median salary but also the variability and presence of outliers, giving insight into income inequality or competitive compensation within certain sectors.

### Remote vs. On-Site Jobs

if "REMOTE\_TYPE\_NAME" in df.columns:  
 fig = px.pie(df, names="REMOTE\_TYPE\_NAME", title="Remote vs. On-Site Jobs")  
 fig.show()

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

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

**Explanation:**  
This pie chart breaks down the proportion of jobs by remote type. It shows how many roles are fully remote, hybrid, or entirely on-site, offering insights into post-pandemic work trends and flexibility offered by employers.

### Job Postings Over Time

if "POSTED" in df.columns:  
 df['POSTED'] = pd.to\_datetime(df['POSTED'], errors='coerce')  
 postings\_over\_time = df['POSTED'].value\_counts().sort\_index()  
 fig = px.line(x=postings\_over\_time.index, y=postings\_over\_time.values, labels={'x': 'Date Posted', 'y': 'Number of Job Postings'}, title="Job Postings Over Time")  
 fig.show()

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

**Explanation:**

This line graph tracks how job postings have changed over time. Peaks and troughs may indicate hiring cycles, economic events, or seasonal effects in the job market.

title: “Skill Gap Analysis”  
format: html kernel: python3  
execute: echo: true  
eval: true

## 3.1 Skill Gap Analysis

### 3.1.1 Team Skill Self-Assessment

import pandas as pd  
  
# Team members and their self-rated proficiency (1–5)  
# Team members and their self-rated proficiency (1–5)  
skills\_data = {  
 "Name": ["Deyang", "Yani", "Jiapei", "Junhao"],  
 "Python": [4, 3, 4, 3],  
 "SQL": [3, 3, 4, 3],  
 "Machine Learning": [2, 5, 5, 2],  
 "Cloud Computing": [1, 4, 2, 3],  
 "R": [3, 5, 4, 2], #   
 "AWS": [4, 4, 2, 3], #   
 "Git": [4, 3, 2, 1], #   
 "Excel": [3, 4, 5, 2], #   
}  
  
df\_skills = pd.DataFrame(skills\_data).set\_index("Name")  
df\_skills

|  | Python | SQL | Machine Learning | Cloud Computing | R | AWS | Git | Excel |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name |  |  |  |  |  |  |  |  |
| Deyang | 4 | 3 | 2 | 1 | 3 | 4 | 4 | 3 |
| Yani | 3 | 3 | 5 | 4 | 5 | 4 | 3 | 4 |
| Jiapei | 4 | 4 | 5 | 2 | 4 | 2 | 2 | 5 |
| Junhao | 3 | 3 | 2 | 3 | 2 | 3 | 1 | 2 |

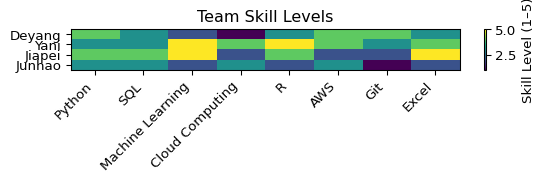
3.1.2 Compare to Market Demand

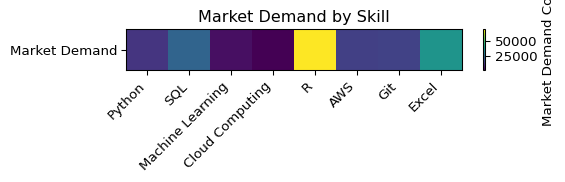
# Load raw job postings data  
raw\_df = pd.read\_csv("lightcast\_job\_postings.csv")  
  
# Drop irrelevant columns  
columns\_to\_drop = [  
 "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",  
 "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",  
 "SOC\_2", "SOC\_3", "SOC\_5"  
]  
valid\_cols\_to\_drop = [col for col in columns\_to\_drop if col in raw\_df.columns]  
raw\_df.drop(columns=valid\_cols\_to\_drop, inplace=True)  
  
# Drop columns with >50% missing  
raw\_df.dropna(thresh=len(raw\_df) \* 0.5, axis=1, inplace=True)  
  
# Fill missing values  
if "SALARY" in raw\_df.columns:  
 raw\_df["SALARY"].fillna(raw\_df["SALARY"].median(), inplace=True)  
if "Industry" in raw\_df.columns:  
 raw\_df["Industry"].fillna("Unknown", inplace=True)  
  
# Remove duplicates  
raw\_df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first", inplace=True)  
  
# Count keyword occurrences  
top\_skills = df\_skills.columns.tolist()  
job\_text = raw\_df["BODY"].fillna("")  
skill\_counts = {s: job\_text.str.contains(s, case=False).sum() for s in top\_skills}  
  
# Append demand row  
df\_skills.loc["Market Demand"] = [skill\_counts[s] for s in top\_skills]  
df\_skills

|  | Python | SQL | Machine Learning | Cloud Computing | R | AWS | Git | Excel |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name |  |  |  |  |  |  |  |  |
| Deyang | 4 | 3 | 2 | 1 | 3 | 4 | 4 | 3 |
| Yani | 3 | 3 | 5 | 4 | 5 | 4 | 3 | 4 |
| Jiapei | 4 | 4 | 5 | 2 | 4 | 2 | 2 | 5 |
| Junhao | 3 | 3 | 2 | 3 | 2 | 3 | 1 | 2 |
| Market Demand | 11784 | 23207 | 3974 | 1303 | 69181 | 14245 | 14696 | 36384 |

3.1.3 Visualize Skill Gaps

import matplotlib.pyplot as plt  
  
# Team skill levels heatmap  
plt.figure(figsize=(6,2))  
plt.imshow(df\_skills.iloc[:-1], aspect="auto")  
plt.colorbar(label="Skill Level (1–5)")  
plt.yticks(range(len(df\_skills.index)-1), df\_skills.index[:-1])  
plt.xticks(range(len(df\_skills.columns)), df\_skills.columns, rotation=45, ha="right")  
plt.title("Team Skill Levels")  
plt.tight\_layout()  
plt.show()  
  
# Market demand heatmap  
plt.figure(figsize=(6,2))  
plt.imshow([df\_skills.loc["Market Demand"]], aspect="auto")  
plt.colorbar(label="Market Demand Count")  
plt.yticks([0], ["Market Demand"])  
plt.xticks(range(len(df\_skills.columns)), df\_skills.columns, rotation=45, ha="right")  
plt.title("Market Demand by Skill")  
plt.tight\_layout()  
plt.show()





hj

Based on the heatmap comparison of each person’s self-ratings (1–5) against the normalized market-demand scores, here’s a concise, paragraph-style improvement plan:

Deyang shows the largest gaps in Cloud Computing and R. To close these, start with an AWS Fundamentals micro-course (Coursera or AWS’s own training) and follow up by building a small R Shiny dashboard using the free “R for Data Science” online book and the swirl R package for hands-on exercises. Aim to spend 2–3 hours per week on labs, then peer-review each other’s code in GitHub.

Yani needs to boost Machine Learning and AWS skills. I recommend Andrew Ng’s Machine Learning specialization on Coursera, paired with the AWS Certified Cloud Practitioner path on AWS Training. After completing each module, apply what you’ve learned by deploying a simple classification model on AWS Sagemaker and sharing the workflow in a team repo, so everyone can give feedback.

Jiapei would benefit most from deeper Cloud Computing practice and reinforcing R. Enroll in Google Cloud’s “Data Engineering” Qwiklabs quests and run through interactive R exercises via swirl. Host a 30-minute “teach-back” session after completing each mini-project—this both cements your own understanding and helps teammates pick up new tricks.

Junhao has room to grow in Excel and SQL. Take an “Excel Essentials” short course on LinkedIn Learning, then tackle Mode’s SQL tutorial problems. Organize weekly problem-solving sessions where one member presents a real-world dataset and the rest write SQL queries together. This combination of structured learning and peer collaboration will efficiently close the remaining gaps.

{ “cells”: [ { “cell\_type”: “markdown”, “id”: “8a0ee288”, “metadata”: {}, “source”: [ “—”, “title: "Project 4"”, “format:”, ” html:“,” self-contained: true“,” theme: flatly“,” toc: true“,”—” ] }, { “cell\_type”: “code”, “execution\_count”: null, “id”: “cc516944”, “metadata”: {}, “outputs”: [ { “name”: “stderr”, “output\_type”: “stream”, “text”: [ “/var/folders/z6/j54j6tys1bbdr2gqg5vfm6tw0000gn/T/ipykernel\_18012/3166840042.py:3: DtypeWarning: Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.”, ” df = pd.read\_csv("lightcast\_job\_postings.csv")” ] } ], “source”: [ “”, “import pandas as pd”, “”, “df = pd.read\_csv("lightcast\_job\_postings.csv")”, “”, “columns\_to\_drop = [”, ” "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",”, ” "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",”, ” "SOC\_2", "SOC\_3", "SOC\_5"”, ”]”, “valid\_cols\_to\_drop = [col for col in columns\_to\_drop if col in df.columns]”, “df.drop(columns=valid\_cols\_to\_drop, inplace=True)”, “”, “df.dropna(thresh=len(df) \* 0.5, axis=1, inplace=True) ”, “”, “if "SALARY" in df.columns:”, ” df["SALARY"].fillna(df["SALARY"].median(), inplace=True)“,”if "Industry" in df.columns:“,” df["Industry"].fillna("Unknown", inplace=True)“,”“,”df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first", inplace=True)“,”“,”if "POSTED" in df.columns:“,” df[‘POSTED’] = pd.to\_datetime(df[‘POSTED’], errors=‘coerce’)” ] }, { “cell\_type”: “code”, “execution\_count”: 3, “id”: “5df86f59”, “metadata”: {}, “outputs”: [], “source”: [ “import pandas as pd”, “import numpy as np”, “from sklearn.preprocessing import LabelEncoder”, “from sklearn.cluster import KMeans”, “import matplotlib.pyplot as plt”, “from sklearn.decomposition import PCA” ] }, { “cell\_type”: “code”, “execution\_count”: 4, “id”: “c90afd3a”, “metadata”: {}, “outputs”: [], “source”: [ “cols\_to\_use = [‘NAICS\_2022\_5\_NAME’, ‘EMPLOYMENT\_TYPE\_NAME’, ‘REMOTE\_TYPE\_NAME’, ‘EDUCATION\_LEVELS\_NAME’]”, “”, “df\_kmeans = df[cols\_to\_use].dropna()” ] }, { “cell\_type”: “code”, “execution\_count”: 5, “id”: “73f0ec7d”, “metadata”: {}, “outputs”: [], “source”: [ “X = pd.get\_dummies(df\_kmeans)” ] }, { “cell\_type”: “code”, “execution\_count”: 6, “id”: “08ba9c52”, “metadata”: {}, “outputs”: [ { “data”: { “image/png”: “”, “text/plain”: [ “<Figure size 800x500 with 1 Axes>” ] }, “metadata”: {}, “output\_type”: “display\_data” } ], “source”: [ “from sklearn.cluster import KMeans”, “import matplotlib.pyplot as plt”, “”, “inertias = []”, “K\_range = range(1, 15) ”, “”, “for k in K\_range:”, ” kmeans = KMeans(n\_clusters=k, random\_state=42, n\_init=10)“,” kmeans.fit(X)“,” inertias.append(kmeans.inertia\_)“,”“,”plt.figure(figsize=(8,5))“,”plt.plot(K\_range, inertias, marker=‘o’)“,”plt.xlabel(‘Number of Clusters (k)’)“,”plt.ylabel(‘Inertia’)“,”plt.title(‘Elbow Method for Optimal k (Multi-Feature)’)“,”plt.grid(True)“,”plt.show()” ] }, { “cell\_type”: “code”, “execution\_count”: 7, “id”: “9a066823”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ “cluster”, “0 23580”, “2 18064”, “3 15794”, “1 11743”, “Name: count, dtype: int64” ] } ], “source”: [ “k\_optimal = 4”, “”, “kmeans\_final = KMeans(n\_clusters=k\_optimal, random\_state=42, n\_init=10)”, “df\_kmeans[‘cluster’] = kmeans\_final.fit\_predict(X)”, “”, “print(df\_kmeans[‘cluster’].value\_counts())” ] }, { “cell\_type”: “code”, “execution\_count”: 8, “id”: “e5c81a4e”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ ” NAICS\_2022\_5\_NAME EMPLOYMENT\_TYPE\_NAME \“,”0 Automotive Parts and Accessories Retailers Full-time (> 32 hours) “,”1 Temporary Help Services Full-time (> 32 hours) “,”2 Other Insurance Related Activities Full-time (> 32 hours) “,”3 Commercial Banking Full-time (> 32 hours) “,”4 Unclassified Industry Part-time / full-time “,”“,” REMOTE\_TYPE\_NAME EDUCATION\_LEVELS\_NAME cluster “,”0 [None] [\n "Bachelor’s degree"\n] 0 “,”1 Remote [\n "No Education Listed"\n] 1 “,”2 [None] [\n "Bachelor’s degree"\n] 0 “,”3 [None] [\n "No Education Listed"\n] 2 “,”4 [None] [\n "No Education Listed"\n] 2 ” ] } ], “source”: [ “print(df\_kmeans.head())” ] }, { “cell\_type”: “code”, “execution\_count”: 9, “id”: “b006c816”, “metadata”: {}, “outputs”: [ { “data”: { “image/png”: “”, “text/plain”: [ “<Figure size 800x600 with 1 Axes>” ] }, “metadata”: {}, “output\_type”: “display\_data” } ], “source”: [ “from sklearn.decomposition import PCA”, “import seaborn as sns”, “”, “pca = PCA(n\_components=2)”, “X\_pca = pca.fit\_transform(X)”, “”, “pca\_df = pd.DataFrame(data=X\_pca, columns=[‘PCA1’, ‘PCA2’])”, “pca\_df[‘cluster’] = df\_kmeans[‘cluster’].values”, “”, “plt.figure(figsize=(8,6))”, “sns.scatterplot(”, ” x=‘PCA1’, y=‘PCA2’,“,” hue=‘cluster’,“,” palette=‘tab10’,“,” data=pca\_df,“,” legend=‘full’,“,” alpha=0.8“,”)“,”plt.title(‘KMeans Clustering with Multiple Features (PCA Reduced)’)“,”plt.xlabel(‘Principal Component 1’)“,”plt.ylabel(‘Principal Component 2’)“,”plt.legend(title=‘Cluster’)“,”plt.grid(True)“,”plt.show()” ] }, { “cell\_type”: “code”, “execution\_count”: 10, “id”: “ac6fa23c”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ ” Cluster Top NAICS\_2022\_5\_NAME \“,”0 0 Unclassified Industry “,”1 1 Unclassified Industry “,”2 2 Computer Systems Design and Related Services “,”3 3 Computer Systems Design and Related Services “,”“,” Top Employment Type Top Remote Type Top Education Level “,”0 Full-time (> 32 hours) [None] Bachelor’s degree “,”1 Full-time (> 32 hours) Remote Bachelor’s degree “,”2 Full-time (> 32 hours) [None] No Education Listed “,”3 Full-time (> 32 hours) [None] Bachelor’s degree ” ] } ], “source”: [ “features = [‘NAICS\_2022\_5\_NAME’, ‘EMPLOYMENT\_TYPE\_NAME’, ‘REMOTE\_TYPE\_NAME’, ‘EDUCATION\_LEVELS\_NAME’]”, “”, “profile\_summary = []”, “”, “for cluster\_label in sorted(df\_kmeans[‘cluster’].unique()):”, ” cluster\_data = df\_kmeans[df\_kmeans[‘cluster’] == cluster\_label]“,” “,” top\_features = {}“,” for feature in features:“,” if cluster\_data[feature].nunique() > 0:“,” if feature == ‘EDUCATION\_LEVELS\_NAME’:“,” clean\_edu = cluster\_data[feature].astype(str).str.replace(r’[\[\]\n\"]‘,’‘, regex=True) # 去掉[], \n, "这些符号“,” clean\_edu = clean\_edu.str.split(’,‘).str[0] “,” top\_features[feature] = clean\_edu.value\_counts().idxmax()“,” else:“,” top\_features[feature] = cluster\_data[feature].value\_counts().idxmax()“,” else:“,” top\_features[feature] = ’N/A’“,” “,” profile\_summary.append({“,” ‘Cluster’: cluster\_label,“,” ‘Top NAICS\_2022\_5\_NAME’: top\_features[‘NAICS\_2022\_5\_NAME’],“,” ‘Top Employment Type’: top\_features[‘EMPLOYMENT\_TYPE\_NAME’],“,” ‘Top Remote Type’: top\_features[‘REMOTE\_TYPE\_NAME’],“,” ‘Top Education Level’: top\_features[‘EDUCATION\_LEVELS\_NAME’]“,” })“,”“,”profile\_df = pd.DataFrame(profile\_summary)“,”“,”print(profile\_df)” ] }, { “cell\_type”: “markdown”, “id”: “b8bec517”, “metadata”: {}, “source”: [ “In this project, we used the KMeans clustering model to group job postings based on several key features: industry classification (NAICS codes), employment type (full-time or part-time), remote work type (remote, on-site, or hybrid), and minimum education requirements (such as bachelor’s or master’s degrees).”, “The purpose of clustering was to identify distinct groups within the job market and provide more targeted career insights and recommendations for job seekers.”, “”, “We selected four features for clustering: NAICS\_2022\_5\_NAME, EMPLOYMENT\_TYPE\_NAME, REMOTE\_TYPE\_NAME, and EDUCATION\_LEVELS\_NAME.”, “After applying one-hot encoding to these categorical variables, we used the KMeans algorithm to build the model.”, “To determine the optimal number of clusters, we used the Elbow Method and selected k=4, which divided the data into four distinct job groups.”, “We then reduced the dimensionality of the clustered data to two components using PCA and visualized the results.”, “The clusters were clearly separated and showed good internal consistency.”, “”, “Through analysis of the clustering results, we found the following:”, “Cluster 0 mainly consists of jobs classified under “Unclassified Industry,” with most being full-time positions, low levels of remote work, and a majority requiring a bachelor’s degree.“,”Cluster 1 also contains jobs from unclassified industries, but with a higher proportion of remote work opportunities, making it suitable for those who prefer remote jobs.“,”Clusters 2 and 3 are primarily composed of jobs in “Computer Systems Design and Related Services,” with higher education requirements, often needing a master’s degree or higher.“,”These clusters are more appropriate for technical job seekers.“,”“,”Based on the cluster characteristics, we recommend that job seekers who prefer remote work focus on opportunities within Cluster 1.“,”Those with advanced degrees and technical backgrounds should prioritize jobs in Clusters 2 and 3.“,”Entry-level candidates or those with lower education requirements may find suitable opportunities in Cluster 0.” ] }, { “cell\_type”: “code”, “execution\_count”: 11, “id”: “9861ea2d”, “metadata”: {}, “outputs”: [ { “name”: “stderr”, “output\_type”: “stream”, “text”: [ “/var/folders/z6/j54j6tys1bbdr2gqg5vfm6tw0000gn/T/ipykernel\_18012/2645817170.py:3: DtypeWarning: Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.”, ” df = pd.read\_csv("lightcast\_job\_postings.csv")“,”/var/folders/z6/j54j6tys1bbdr2gqg5vfm6tw0000gn/T/ipykernel\_18012/2645817170.py:14: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.“,”The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.“,”“,”For example, when doing ‘df[col].method(value, inplace=True)’, try using ‘df.method({col: value}, inplace=True)’ or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.“,”“,”“,” df["SALARY"].fillna(df["SALARY"].median(), inplace=True)” ] } ], “source”: [ “import pandas as pd”, “”, “df = pd.read\_csv("lightcast\_job\_postings.csv")”, “”, “columns\_to\_drop = [”, ” "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",”, ” "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",”, ” "SOC\_2", "SOC\_3", "SOC\_5"”, ”]”, “valid\_cols\_to\_drop = [col for col in columns\_to\_drop if col in df.columns]”, “df.drop(columns=valid\_cols\_to\_drop, inplace=True)”, “”, “if "SALARY" in df.columns:”, ” df["SALARY"].fillna(df["SALARY"].median(), inplace=True)“,”if "Industry" in df.columns:“,” df["Industry"].fillna("Unknown", inplace=True)“,”df["MIN\_YEARS\_EXPERIENCE"] = df["MIN\_YEARS\_EXPERIENCE"].fillna(df["MIN\_YEARS\_EXPERIENCE"].median())“,”“,”df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first", inplace=True)“,”“,”if "POSTED" in df.columns:“,” df[‘POSTED’] = pd.to\_datetime(df[‘POSTED’], errors=‘coerce’)” ] }, { “cell\_type”: “code”, “execution\_count”: 12, “id”: “16ab9a72”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ “EMPLOYMENT\_TYPE\_NAME 17”, “REMOTE\_TYPE\_NAME 17”, “EDUCATION\_LEVELS\_NAME 17”, “NAICS\_2022\_5\_NAME 17”, “TITLE\_NAME 17”, “SALARY 0”, “MIN\_YEARS\_EXPERIENCE 0”, “dtype: int64” ] } ], “source”: [ “cols\_to\_check = [”, ” ‘EMPLOYMENT\_TYPE\_NAME’, ”, ” ‘REMOTE\_TYPE\_NAME’, ”, ” ‘EDUCATION\_LEVELS\_NAME’, ”, ” ‘NAICS\_2022\_5\_NAME’, ”, ” ‘TITLE\_NAME’, ”, ” ‘SALARY’,”, ” ‘MIN\_YEARS\_EXPERIENCE’”, ”]”, “”, “missing\_counts = df[cols\_to\_check].isnull().sum()”, “”, “print(missing\_counts)” ] }, { “cell\_type”: “code”, “execution\_count”: 13, “id”: “2f6dafbb”, “metadata”: {}, “outputs”: [], “source”: [ “cols\_to\_use = [”, ” ‘EMPLOYMENT\_TYPE\_NAME’, ”, ” ‘REMOTE\_TYPE\_NAME’, ”, ” ‘EDUCATION\_LEVELS\_NAME’, ”, ” ‘TITLE\_NAME’, ”, ” ‘NAICS\_2022\_5\_NAME’,”, ” ‘SALARY’”, ”]”, “df\_regression = df[cols\_to\_use].dropna()”, “”, “y = df\_regression[‘SALARY’]”, “”, “X = pd.get\_dummies(df\_regression.drop(‘SALARY’, axis=1), drop\_first=True)” ] }, { “cell\_type”: “code”, “execution\_count”: 14, “id”: “a2a01640”, “metadata”: {}, “outputs”: [], “source”: [ “from sklearn.model\_selection import train\_test\_split”, “”, “X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=688)” ] }, { “cell\_type”: “code”, “execution\_count”: 15, “id”: “cd54aee2”, “metadata”: {}, “outputs”: [ { “data”: { “text/html”: [ “], “text/plain”: [ “LinearRegression()” ] }, “execution\_count”: 15, “metadata”: {}, “output\_type”: “execute\_result” } ], “source”: [ “from sklearn.linear\_model import LinearRegression”, “”, “lr\_model = LinearRegression()”, “lr\_model.fit(X\_train, y\_train)” ] }, { “cell\_type”: “code”, “execution\_count”: 16, “id”: “0af59309”, “metadata”: {}, “outputs”: [], “source”: [ “y\_pred = lr\_model.predict(X\_test)” ] }, { “cell\_type”: “code”, “execution\_count”: 17, “id”: “abe0dbd4”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ “RMSE: 24855.89”, “R²: 0.2912” ] } ], “source”: [ “from sklearn.metrics import mean\_squared\_error, r2\_score”, “import numpy as np”, “”, “rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))”, “r2 = r2\_score(y\_test, y\_pred)”, “”, “print(f"RMSE: {rmse:.2f}")”, “print(f"R²: {r2:.4f}")” ] }, { “cell\_type”: “code”, “execution\_count”: 18, “id”: “2184e8f5”, “metadata”: {}, “outputs”: [ { “name”: “stdout”, “output\_type”: “stream”, “text”: [ ” Feature Coefficient \“,”5571 TITLE\_NAME\_Urgent Care Physicians 251386.121082 “,”3976 TITLE\_NAME\_Portfolio Strategists 218543.202752 “,”677 TITLE\_NAME\_Business Performance Analysts 200638.126095 “,”4260 TITLE\_NAME\_Psychiatry Physicians 198843.552601 “,”3135 TITLE\_NAME\_Lead Security Architects 189207.776080 “,”4020 TITLE\_NAME\_Primary Care Physicians 157627.519893 “,”1699 TITLE\_NAME\_Digital Verification Engineers 153248.368916 “,”866 TITLE\_NAME\_Certified Tumor Registrars 144003.199611 “,”1759 TITLE\_NAME\_Directors of Global Logistics 138363.896627 “,”5113 TITLE\_NAME\_Staff IT Engineers 133667.627850 “,”2085 TITLE\_NAME\_Enterprise Services Managers 133393.234468 “,”2169 TITLE\_NAME\_Executive Business Directors 131837.191508 “,”5668 TITLE\_NAME\_Water/Wastewater Engineers 131641.345181 “,”893 TITLE\_NAME\_Chief Systems Architects 130059.716545 “,”3209 TITLE\_NAME\_Loan Specialists 128697.981039 “,”2927 TITLE\_NAME\_Insurance Executives 127916.188998 “,”4443 TITLE\_NAME\_Regulatory Associates 127037.822853 “,”1278 TITLE\_NAME\_Counterintelligence Agents 116219.281719 “,”5097 TITLE\_NAME\_Sponsorship Managers 114539.904505 “,”121 TITLE\_NAME\_Actuarial Services Managers 110330.218256 “,”“,” Coefficient\_Abs “,”5571 251386.121082 “,”3976 218543.202752 “,”677 200638.126095 “,”4260 198843.552601 “,”3135 189207.776080 “,”4020 157627.519893 “,”1699 153248.368916 “,”866 144003.199611 “,”1759 138363.896627 “,”5113 133667.627850 “,”2085 133393.234468 “,”2169 131837.191508 “,”5668 131641.345181 “,”893 130059.716545 “,”3209 128697.981039 “,”2927 127916.188998 “,”4443 127037.822853 “,”1278 116219.281719 “,”5097 114539.904505 “,”121 110330.218256 ” ] } ], “source”: [ “import pandas as pd”, “”, “coef\_df = pd.DataFrame({”, ” ‘Feature’: X\_train.columns,“,” ‘Coefficient’: lr\_model.coef\_“,”})“,”“,”coef\_df[‘Coefficient\_Abs’] = coef\_df[‘Coefficient’].abs()“,”coef\_df = coef\_df.sort\_values(by=‘Coefficient\_Abs’, ascending=False)“,”“,”print(coef\_df.head(20))” ] }, { “cell\_type”: “markdown”, “id”: “8949aa7d”, “metadata”: {}, “source”: [ “In this project, we used a linear regression model to predict salary levels based mainly on the job title feature (TITLE\_NAME). The model results show that roles in healthcare, such as Urgent Care Physicians and Psychiatry Physicians, as well as positions in finance like Portfolio Strategists and IT security roles like Lead Security Architects, have a strong positive impact on salary levels. This means that jobs requiring high specialization and in-demand skills tend to offer higher salaries.”, “For job seekers, these findings provide useful guidance for career planning and salary expectations. People who aim for higher-paying jobs might consider focusing on fields like healthcare, finance, or IT security. However, it is also important to understand that these roles often come with higher professional requirements and work pressure. When making career choices, it is important to match personal interests, skills, and strengths with realistic goals. Companies can also use this insight to better design competitive salary strategies and attract top talent.” ] }, { “cell\_type”: “code”, “execution\_count”: null, “id”: “39fd46a6”, “metadata”: {}, “outputs”: [ { “data”: { “image/png”: “”, “text/plain”: [ “<Figure size 800x600 with 1 Axes>” ] }, “metadata”: {}, “output\_type”: “display\_data” } ], “source”: [ “results\_df = pd.DataFrame({”, ” ‘Actual Salary’: y\_test,“,” ‘Predicted Salary’: y\_pred“,”})“,”“,”plt.figure(figsize=(8, 6))“,”plt.scatter(results\_df[‘Actual Salary’], results\_df[‘Predicted Salary’], alpha=0.6, color=‘cornflowerblue’)“,”plt.plot(“,” [results\_df[‘Actual Salary’].min(), results\_df[‘Actual Salary’].max()],“,” [results\_df[‘Actual Salary’].min(), results\_df[‘Actual Salary’].max()],“,” color=‘red’, linestyle=‘–’“,”)“,”plt.title(‘Actual vs Predicted Salaries’, fontsize=14)“,”plt.xlabel(‘Actual Salary’, fontsize=12)“,”plt.ylabel(‘Predicted Salary’, fontsize=12)“,”plt.grid(True, linestyle=‘–’, alpha=0.7)“,”plt.tight\_layout()“,”plt.show()” ] }, { “cell\_type”: “markdown”, “id”: “4fd3431d”, “metadata”: {}, “source”: [ “The scatter plot shows the model’s prediction performance. The x-axis represents the actual salary, and the y-axis shows the predicted salary. The red dashed line represents a perfect prediction line. Overall, the model captures the general trend of salary changes fairly well. However, in the higher salary range, especially for jobs over 200,000 dollars per year, the model tends to underestimate salaries and shows larger prediction errors. In the lower salary range, especially for jobs below 100,000 dollars, the model predicts more accurately.”, “”, “This suggests that while the model provides helpful salary estimates, job seekers should use it carefully, especially when applying for high-paying positions. It is important to also consider personal background, industry trends, and specific job requirements. For those targeting high salaries, additional preparation, such as improving professional skills and gaining more experience, can help address the model’s potential underestimation.” ] }, { “cell\_type”: “code”, “execution\_count”: null, “id”: “883f8582”, “metadata”: {}, “outputs”: [ { “data”: { “image/png”: “”, “text/plain”: [ “<Figure size 1000x600 with 1 Axes>” ] }, “metadata”: {}, “output\_type”: “display\_data” } ], “source”: [ “import matplotlib.pyplot as plt”, “import pandas as pd”, “”, “feature\_importance = pd.Series(lr\_model.coef\_, index=X\_train.columns)”, “feature\_importance = feature\_importance.abs().sort\_values(ascending=False).head(20)”, “”, “plt.figure(figsize=(10, 6))”, “plt.barh(feature\_importance.index[::-1], feature\_importance.values[::-1], color=‘skyblue’)”, “plt.title("Top 20 Feature Importances (Linear Regression Coefficients)", fontsize=14)”, “plt.xlabel("Coefficient Magnitude", fontsize=12)”, “plt.ylabel("Features", fontsize=12)”, “plt.grid(axis=‘x’, linestyle=‘–’, alpha=0.7)”, “plt.tight\_layout()”, “plt.show()” ] }, { “cell\_type”: “markdown”, “id”: “c05a97ea”, “metadata”: {}, “source”: [ “From the feature importance chart, we can see that the model’s explanation of salary differences is highly concentrated around a small group of high-impact job titles. This suggests that selecting roles in key industries like healthcare, finance, and IT can significantly influence salary outcomes.”, “”, “For job seekers, aiming for these impactful roles can greatly improve earning potential. On the other hand, for roles with lower feature importance, it might be a good idea to adjust salary expectations based on the broader market. People working in these positions can also consider developing new skills or transitioning into fields with stronger market demand to boost their career development and salary prospects.” ] } ], “metadata”: { “kernelspec”: { “display\_name”: “venv”, “language”: “python”, “name”: “python3” }, “language\_info”: { “codemirror\_mode”: { “name”: “ipython”, “version”: 3 }, “file\_extension”: “.py”, “mimetype”: “text/x-python”, “name”: “python”, “nbconvert\_exporter”: “python”, “pygments\_lexer”: “ipython3”, “version”: “3.13.2” } }, “nbformat”: 4, “nbformat\_minor”: 5 }

Adhikari, P. (2024): “[Exploring the nexus between artificial intelligence and job displacement: A literature review](https://jndmeerut.org/wp-content/uploads/2024/07/JND-1.pdf),” *Journal of National Development*, 37, 1–13.

Ansari, A., and A. Ansari. (2024): “[Consequences of AI induced job displacement](https://ijbat.com/index.php/IJBAT/article/view/18/31),” *International Journal of Business, Analytics, and Technology*, 2, 4–19.

Colombo, E., F. Mercorio, M. Mezzanzanica, and A. Serino. (2024): “[Towards the Terminator Economy: Assessing Job Exposure to AI Through LLMs](https://doi.org/10.48550/arXiv.2407.19204),”

Sezgin, E. (2023): “[Artificial intelligence in healthcare: Complementing, not replacing, doctors and healthcare providers](https://doi.org/10.1177/20552076231186520),” *Digital Health*, 9.

Stone, D. L., K. M. Lukaszewski, and R. D. Johnson. (2024): “[Will artificial intelligence radically change human resource management processes?](https://doi.org/10.1016/j.orgdyn.2024.101034)” *Organizational Dynamics*, 53, 101034.