Data Analysis

Comprehensive Data Cleaning & Exploratory Analysis of Geographic Data

Dakota Alder

Julio Garcia

# Load and Preview LightCast Data

from pyspark.sql import SparkSession  
  
  
# Start a Spark session  
spark = SparkSession.builder.appName("JobPostingsAnalysis").getOrCreate()  
  
# Load the CSV file into a Spark DataFrame  
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("../data/lightcast\_job\_postings.csv")

WARNING: Using incubator modules: jdk.incubator.vector  
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
25/10/16 06:03:16 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
[Stage 1:> (0 + 1) / 1]

# Drop unnecessary columns

Redundant or irrelevant columns are dropped here.

NAICS and SOC levels are removed because each job is already described by its most detailed (and most general) industry and occupation classification (NAICS\_2022\_6\_NAME and SOC\_6\_NAME).

Timestamps and system variables (like LAST\_UPDATED\_TIMESTAMP) are not meaningful.

Simplifying the dataset this way speeds up our processing and makes the whole dataframe look more clean and user-friendly.

import pandas as pd  
from pyspark.sql.functions import when, col  
  
#Clean Data to convert to Pandas  
columns\_to\_drop = ["ID", "BODY", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",  
 "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",  
 "SOC\_2", "SOC\_3", "SOC\_4", "SOC\_5", "LAST\_UPDATED\_DATE", "LAST\_UPDATED\_TIMESTAMP", "EXPIRED", "SOURCE\_TYPES", "SOURCES", "ACTIVE\_SOURCES\_INFO", "MODELED\_EXPIRED", "MODELED\_DURATION", "NAICS2\_NAME", "NAICS3\_NAME", "NAICS4\_NAME", "NAICS5\_NAME", "NAICS6\_NAME",  
 "SOC\_2\_NAME", "SOC\_3\_NAME", "SOC\_4\_NAME", "SOC\_5\_NAME", "EDUCATION\_LEVELS", "MIN\_EDULEVELS"  
   
]  
cleaned\_data = df.drop(\*columns\_to\_drop)  
  
cleaned\_data = cleaned\_data.withColumn(  
 "REMOTE\_TYPE\_NAME",  
 when(col("REMOTE\_TYPE\_NAME") == "Remote", "Remote")  
 .when(col("REMOTE\_TYPE\_NAME") == "Hybrid Remote", "Hybrid")  
 .when(col("REMOTE\_TYPE\_NAME") == "[None]", "On-site")  
 .when(col("REMOTE\_TYPE\_NAME").isNull(), "On-site")  
 .when(col("REMOTE\_TYPE\_NAME") == "Not Remote", "On-site")  
 .otherwise(col("REMOTE\_TYPE\_NAME"))  
)  
  
cleaned\_data = cleaned\_data.withColumn(  
 "EMPLOYMENT\_TYPE\_NAME",  
 when(col("EMPLOYMENT\_TYPE\_NAME") == "Part-time / full-time", "Flexible")  
 .when(col("EMPLOYMENT\_TYPE\_NAME").isNull(), "Full-Time")  
 .when(col("EMPLOYMENT\_TYPE\_NAME") == "Part-time (â‰¤ 32 hours)", "Part-Time")  
 .when(col("EMPLOYMENT\_TYPE\_NAME") == "Full-time (> 32 hours)", "Full-Time")  
 .otherwise(col("EMPLOYMENT\_TYPE\_NAME"))   
)  
  
cleaned\_data = cleaned\_data.filter(col("NAICS\_2022\_2\_NAME") != "Unclassified Industry")  
  
median\_salary = cleaned\_data.approxQuantile("SALARY", [0.5], 0.01)[0]  
cleaned\_data = cleaned\_data.withColumn(  
 "SALARY",  
 when(col("SALARY").isNull(), median\_salary).otherwise(col("SALARY"))  
)  
  
  
clean\_pdf = cleaned\_data.toPandas()

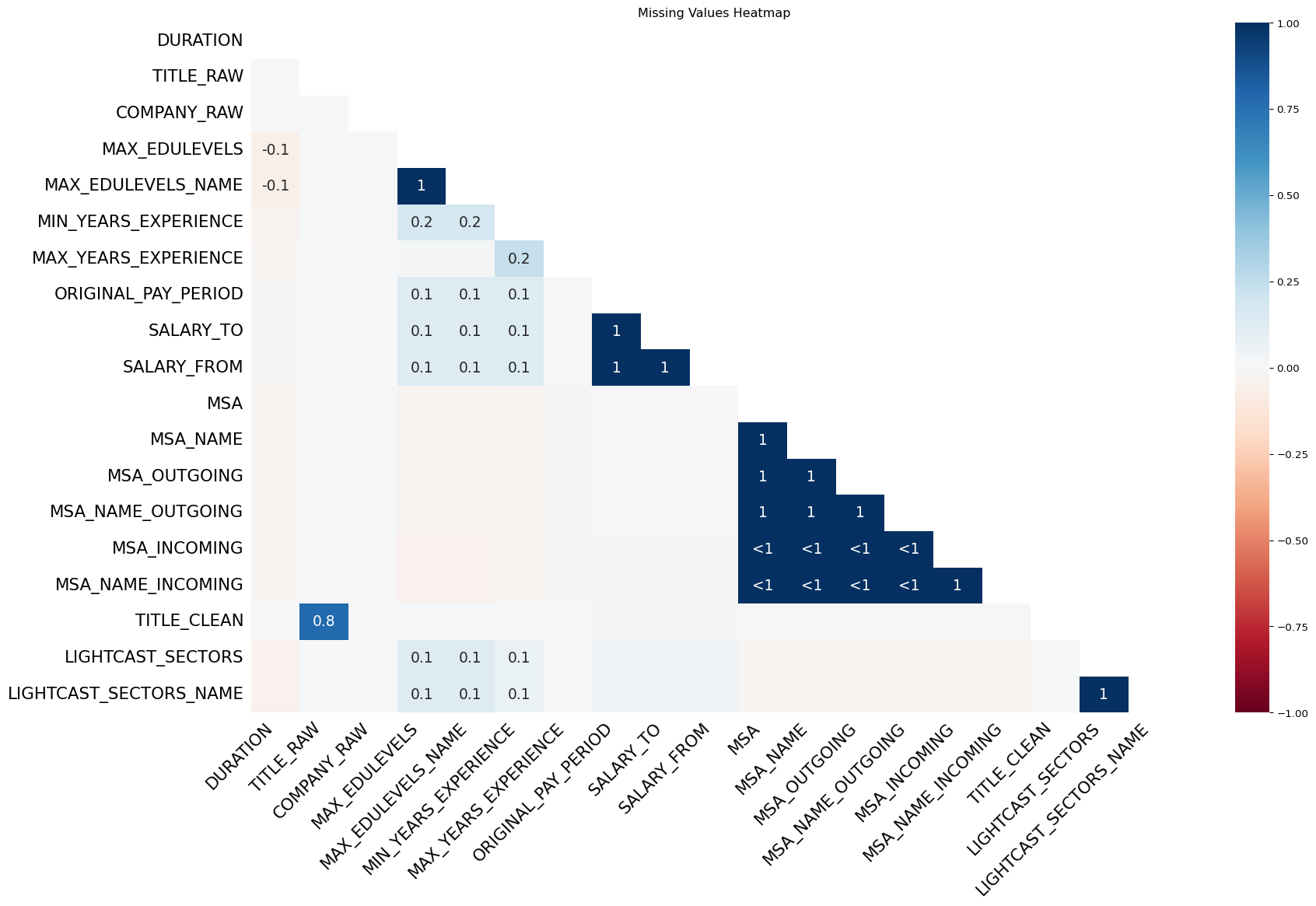
[Stage 2:> (0 + 1) / 1] 25/10/16 06:03:36 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.  
[Stage 3:> (0 + 1) / 1]

# Handle Missing Values

We also cleaned categorical values

Missing categorical data (like City, Company, State) were replaced with “Unknown” so that there is data in all rows. Duplicates were also dropped to not skew the analysis. Salary missing values were replaced with the median salary. Remote Type Name and Employment type were simplified into smaller groupings.

import missingno as msno  
import matplotlib.pyplot as plt  
  
# Visualize missing data  
msno.heatmap(clean\_pdf)  
plt.title("Missing Values Heatmap")  
plt.show()  
  
fill\_cols = ["CITY\_NAME", "CITY", "LOCATION", "STATE", "STATE\_NAME", "COMPANY", "COMPANY\_NAME"]  
clean\_pdf[fill\_cols] = clean\_pdf[fill\_cols].fillna("Unknown")  
  
clean\_pdf = clean\_pdf.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first")  
  
clean\_pdf.dropna(thresh=len(clean\_pdf)\*0.5, axis=1, inplace=True)



# Helper Columns for classifying AI and Posted Dates

We created columns to classify what job titles may be affected by AI vs Non-AI Job Titles We created a column for the month that the job was posted in order to create growth data

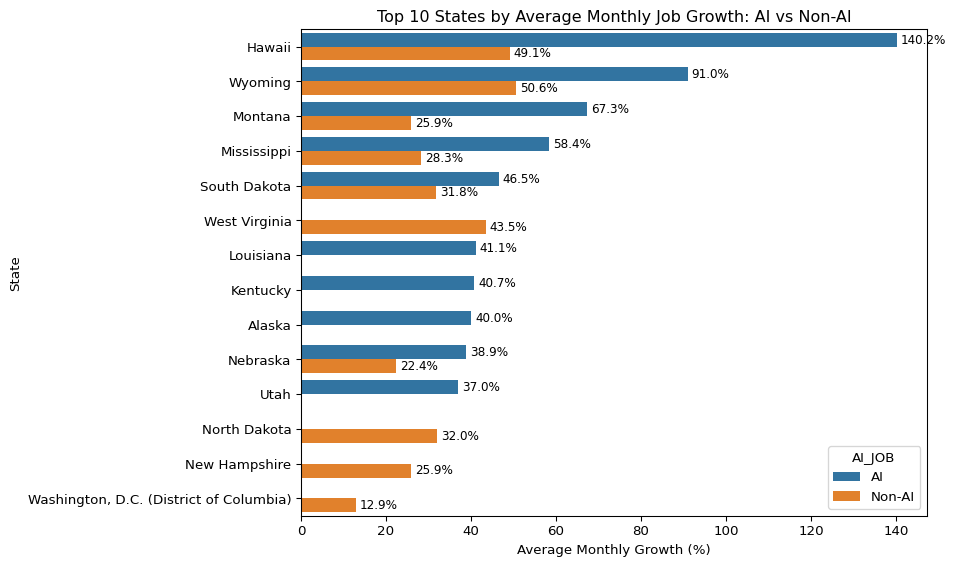
#New Column to Classify AI Jobs and Add Month of Posting Date  
  
  
ai\_keywords = [  
 "AI", "Machine Learning", "Data Scientist", "Data Analyst", "ML",   
 "Artificial Intelligence", "Deep Learning", "NLP", "Predictive Analytics"  
]  
  
#Function to classify AI vs Non-AI Jobs  
def classify\_ai(title):  
 title\_lower = str(title).lower()  
 for keyword in ai\_keywords:  
 if keyword.lower() in title\_lower:  
 return "AI"  
 return "Non-AI"  
  
clean\_pdf["AI\_JOB"] = clean\_pdf["TITLE\_RAW"].apply(classify\_ai)  
  
clean\_pdf["POSTED"] = pd.to\_datetime(clean\_pdf["POSTED"], errors="coerce")  
clean\_pdf["POSTED\_MONTH"] = clean\_pdf["POSTED"].dt.month  
  
#clean\_pdf.head(25)

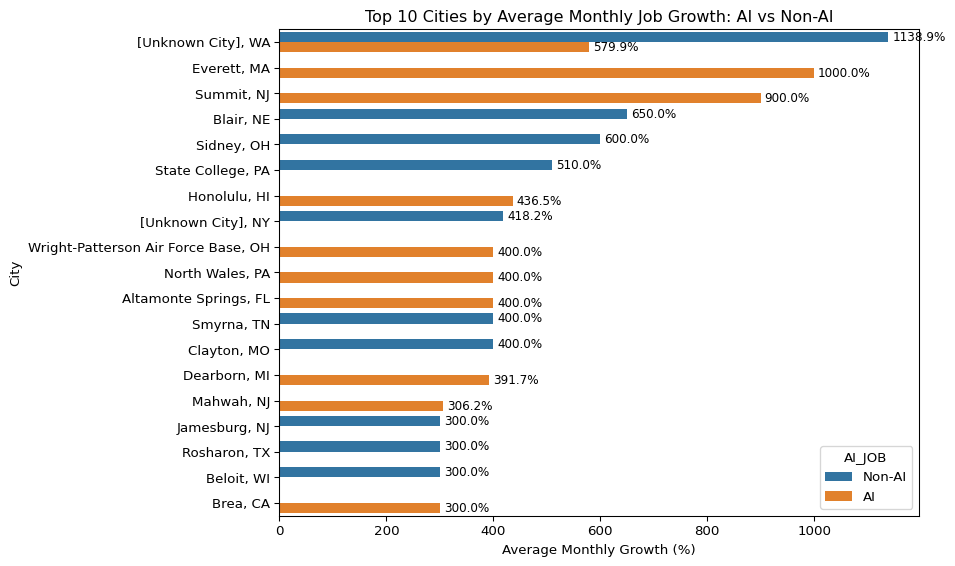
# City and State Analysis by AI vs Non-AI Jobs

Key Insights:

#Question 1 Visualization: Which Cities or States have the highest job growth for AI vs Non-AI  
  
count\_by\_month\_state = (  
 clean\_pdf.groupby(["STATE\_NAME", "POSTED\_MONTH", "AI\_JOB"])  
 .size()  
 .reset\_index(name="count")  
)  
  
count\_by\_month\_city = (  
 clean\_pdf.groupby(["CITY\_NAME", "POSTED\_MONTH", "AI\_JOB"])  
 .size()  
 .reset\_index(name="count")  
)  
  
#Measure job growth by State and then by city  
count\_by\_month\_state = count\_by\_month\_state.sort\_values(["STATE\_NAME", "AI\_JOB", "POSTED\_MONTH"])  
count\_by\_month\_state["GROWTH"] = (  
 count\_by\_month\_state  
 .groupby(["STATE\_NAME", "AI\_JOB"])["count"]  
 .pct\_change() \* 100  
)  
  
count\_by\_month\_city = count\_by\_month\_city.sort\_values(["CITY\_NAME", "AI\_JOB", "POSTED\_MONTH"])  
count\_by\_month\_city["GROWTH"] = (  
 count\_by\_month\_city  
 .groupby(["CITY\_NAME", "AI\_JOB"])["count"]  
 .pct\_change() \* 100  
)  
  
avg\_growth\_state = (  
 count\_by\_month\_state.groupby(["STATE\_NAME", "AI\_JOB"])["GROWTH"]  
 .mean()  
 .reset\_index()  
 .dropna()  
 .sort\_values("GROWTH", ascending=False)  
)  
  
avg\_growth\_city = (  
 count\_by\_month\_city.groupby(["CITY\_NAME", "AI\_JOB"])["GROWTH"]  
 .mean()  
 .reset\_index()  
 .dropna()  
 .sort\_values("GROWTH", ascending=False)  
)  
  
print("Top 10 States by AI Job Growth:")  
print(avg\_growth\_state[avg\_growth\_state["AI\_JOB"] == "AI"].head(10))  
  
print("\nTop 10 States by Non-AI Job Growth:")  
print(avg\_growth\_state[avg\_growth\_state["AI\_JOB"] == "Non-AI"].head(10))  
  
print("Top 10 Cities by AI Job Growth:")  
print(avg\_growth\_city[avg\_growth\_city["AI\_JOB"] == "AI"].head(10))  
  
print("\nTop 10 Cities by Non-AI Job Growth:")  
print(avg\_growth\_city[avg\_growth\_city["AI\_JOB"] == "Non-AI"].head(10))  
  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
state\_visual = avg\_growth\_state.groupby("AI\_JOB").head(10)  
  
plt.figure(figsize=(10,6))  
ax\_state = sns.barplot(data=state\_visual, y="STATE\_NAME", x="GROWTH", hue="AI\_JOB")  
plt.title("Top 10 States by Average Monthly Job Growth: AI vs Non-AI")  
plt.xlabel("Average Monthly Growth (%)")  
plt.ylabel("State")  
  
for container in ax\_state.containers:  
 ax\_state.bar\_label(container, fmt="%.1f%%", label\_type="edge", padding=3, fontsize=9)  
  
plt.tight\_layout()  
plt.savefig("../ad688\_group6\_geographic\_analysis/images/top10state.png", dpi=300)  
plt.show()  
  
city\_visual = avg\_growth\_city.groupby("AI\_JOB").head(10)  
  
plt.figure(figsize=(10,6))  
ax\_city = sns.barplot(data=city\_visual, y="CITY\_NAME", x="GROWTH", hue="AI\_JOB")  
plt.title("Top 10 Cities by Average Monthly Job Growth: AI vs Non-AI")  
plt.xlabel("Average Monthly Growth (%)")  
plt.ylabel("City")  
  
for container in ax\_city.containers:  
 ax\_city.bar\_label(container, fmt="%.1f%%", label\_type="edge", padding=3, fontsize=9)  
  
plt.tight\_layout()  
plt.savefig("../ad688\_group6\_geographic\_analysis/images/top10city.png", dpi=300)  
plt.show()

Top 10 States by AI Job Growth:  
 STATE\_NAME AI\_JOB GROWTH  
20 Hawaii AI 140.217803  
100 Wyoming AI 90.990260  
50 Montana AI 67.291667  
46 Mississippi AI 58.393822  
80 South Dakota AI 46.527778  
34 Louisiana AI 41.077075  
32 Kentucky AI 40.737045  
2 Alaska AI 40.043290  
52 Nebraska AI 38.870132  
86 Utah AI 36.988636  
  
Top 10 States by Non-AI Job Growth:  
 STATE\_NAME AI\_JOB GROWTH  
101 Wyoming Non-AI 50.555556  
21 Hawaii Non-AI 49.105634  
97 West Virginia Non-AI 43.492753  
67 North Dakota Non-AI 32.006313  
81 South Dakota Non-AI 31.818182  
47 Mississippi Non-AI 28.331625  
57 New Hampshire Non-AI 25.913029  
51 Montana Non-AI 25.904481  
53 Nebraska Non-AI 22.365222  
95 Washington, D.C. (District of Columbia) Non-AI 12.850958  
Top 10 Cities by AI Job Growth:  
 CITY\_NAME AI\_JOB GROWTH  
1366 Everett, MA AI 1000.000000  
4155 Summit, NJ AI 900.000000  
4876 [Unknown City], WA AI 579.875000  
1964 Honolulu, HI AI 436.507937  
4743 Wright-Patterson Air Force Base, OH AI 400.000000  
3133 North Wales, PA AI 400.000000  
82 Altamonte Springs, FL AI 400.000000  
1072 Dearborn, MI AI 391.666667  
2532 Mahwah, NJ AI 306.250000  
474 Brea, CA AI 300.000000  
  
Top 10 Cities by Non-AI Job Growth:  
 CITY\_NAME AI\_JOB GROWTH  
4877 [Unknown City], WA Non-AI 1138.925972  
396 Blair, NE Non-AI 650.000000  
3974 Sidney, OH Non-AI 600.000000  
4106 State College, PA Non-AI 510.000000  
4851 [Unknown City], NY Non-AI 418.161765  
4003 Smyrna, TN Non-AI 400.000000  
822 Clayton, MO Non-AI 400.000000  
2101 Jamesburg, NJ Non-AI 300.000000  
3720 Rosharon, TX Non-AI 300.000000  
323 Beloit, WI Non-AI 300.000000





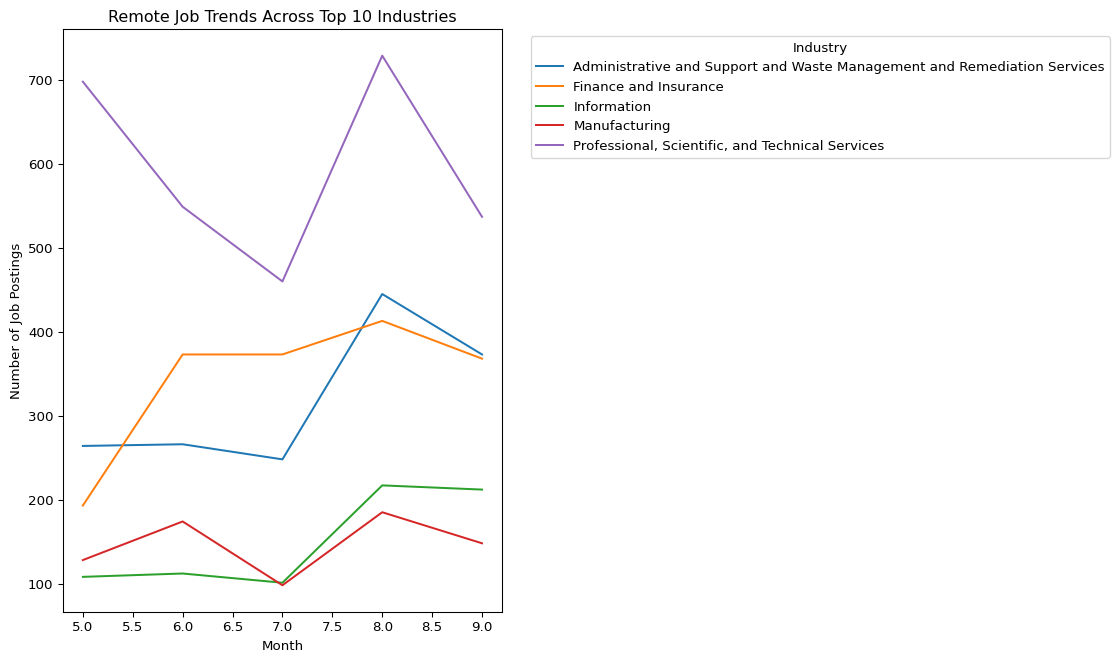
## Key Insights

The analysis shows that AI jobs postings are increasing at a higher rate than Non-AI jobs, and that is especially true in Hawaii, Wyoming, Montana, and Mississippi. However, there seems to be no AI Job growth in multiple states, and an increase in Non-AI jobs in those same states, suggesting that AI isn’t quite yet displacing jobs. In fact, there are only a select few states where AI Jobs are increasing, most of the 50 states have an increase in Non-AI jobs.

# Remote Job Growth by Industry

Key Insights:

#Question 2: Are remote jobs increasing or decreasing across industries?  
  
remote\_only = clean\_pdf[clean\_pdf["REMOTE\_TYPE\_NAME"] == "Remote"]  
  
remote\_growth = (  
 remote\_only.groupby(["NAICS\_2022\_2\_NAME", "POSTED\_MONTH"])  
 .size()  
 .reset\_index(name="count")  
)  
  
  
top\_5\_industries = (  
 remote\_only["NAICS\_2022\_2\_NAME"]  
 .value\_counts()  
 .head(5)  
 .index  
)  
  
top\_remote\_growth = remote\_growth[remote\_growth["NAICS\_2022\_2\_NAME"].isin(top\_5\_industries)]  
  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(12, 7))  
sns.lineplot(  
 data=top\_remote\_growth,  
 x="POSTED\_MONTH",  
 y="count",  
 hue="NAICS\_2022\_2\_NAME"  
)  
plt.title("Remote Job Trends Across Top 10 Industries")  
plt.xlabel("Month")  
plt.ylabel("Number of Job Postings")  
plt.legend(title="Industry", bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.savefig("../ad688\_group6\_geographic\_analysis/images/remoteindustries.png", dpi=300)  
plt.show()



## Key Insights

When looking at the growth of Remote jobs by industry, we can see that there is an increase in remote jobs in the Finance and Insurance Industries, the Information Industries, and the Administrative and Support and Waste Management and Remediation Services Industries. This seems to suggest that more tech heavy industries can support and are seeing growth in remote positions, but industries like Professional, Scientific, and Technical Services are actually seeing a decline in remote work. These industries might have job titles that require in person work and assistance, which would lead to on-site.

# Tech-Hubs vs emerging Markets

Key Insights:

#Question #3: Do Tech hubs (Silicon Valley, Austin, Boston) still dominate hiring, or are other locations emerging?  
  
tech\_hubs = ["Austin-Round Rock-Georgetown, TX", "Boston-Cambridge-Newton, MA-NH","Los Angeles-Long Beach-Anaheim, CA","San Diego-Chula Vista-Carlsbad, CA","San Francisco-Oakland-Berkeley, CA","San Jose-Sunnyvale-Santa Clara, CA","Seattle-Tacoma-Bellevue, WA"]  
  
# Create a column classifying if the city is a tech hub  
clean\_pdf["TECH\_HUB"] = clean\_pdf["MSA\_NAME"].apply(  
 lambda x: "Tech Hub" if x in tech\_hubs else "Other"  
)  
  
#Count number of postings by Tech Hub  
tech\_hub\_counts = (  
 clean\_pdf.groupby("MSA\_NAME")  
 .size()  
 .reset\_index(name="count")  
 .sort\_values("count", ascending=False)  
)  
  
# Merge to add TECH\_HUB classification to each MSA  
tech\_hub\_counts = tech\_hub\_counts.merge(  
 clean\_pdf[["MSA\_NAME", "TECH\_HUB"]].drop\_duplicates(),  
 on="MSA\_NAME",  
 how="left"  
)  
  
  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
plt.figure(figsize=(14, 8))  
sns.barplot(  
 data=tech\_hub\_counts.head(15),  
 x="MSA\_NAME",  
 y="count",  
 hue="TECH\_HUB",  
 palette="viridis"  
)  
plt.title("Top 15 Metro Areas by Count of Job Postings")  
plt.xlabel("Tech Hub")  
plt.ylabel("Number of Job Postings")  
plt.xticks(rotation=60, ha="right")  
  
# Add labels on top of bars  
for i, row in tech\_hub\_counts.head(15).iterrows():  
 plt.text(row["count"] + 100, i, f"{row['count']:,}", va="center", fontsize=10)  
  
#plt.tight\_layout()  
plt.savefig("../ad688\_group6\_geographic\_analysis/images/top15techhubs.png", dpi=300)  
plt.show()



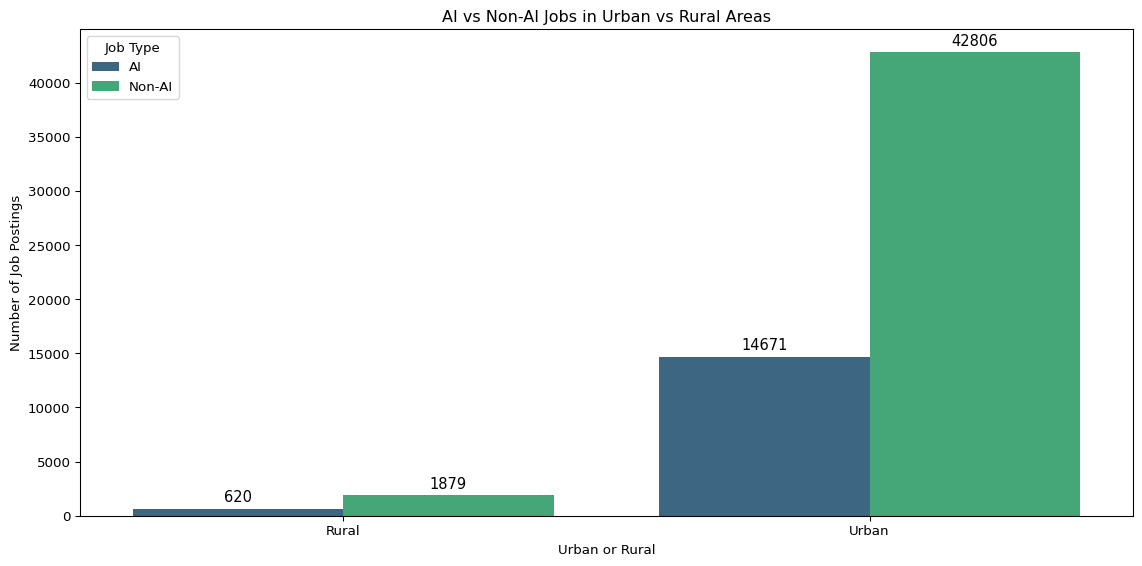
## Key Insights

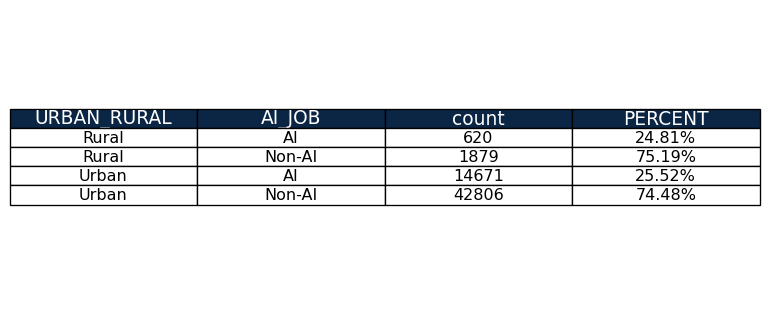
This plot is showing the concentration of job postings in each of the top 15 metro areas, which is interesting because the top 5 are actually not part of the tech hubs list that we created. Number 6, 7, 8, 11, and 15 are tech hubs, but this goes to show that a job seeker does not need to be in a tech hub for job searching. There are plenty of high population, metro areas that will have plenty of jobs. We would like to go further and look at industry and AI jobs as well, which is found in the next plot.

# A Comparison of the Urban and Rural Job Market in relation to AI Careers

Key Insights:

# Question 4: How do Urban vs. Rural Job markets differ for AI and non-AI careers?  
  
# Classify as 'Urban' if MSA\_NAME is present, else 'Rural'  
clean\_pdf["URBAN\_RURAL"] = clean\_pdf["MSA\_NAME"].apply(lambda x: "Urban" if pd.notnull(x) else "Rural")  
  
# Group data by month, urban/rural, and AI vs Non-AI  
urban\_rural\_jobs = (  
 clean\_pdf.groupby(["URBAN\_RURAL", "AI\_JOB"])  
 .size()  
 .reset\_index(name="count")  
)  
  
#Calculate percentages  
urban\_rural\_jobs["PERCENT"] = (  
 urban\_rural\_jobs.groupby("URBAN\_RURAL")["count"]  
 .apply(lambda x: 100 \* x / x.sum())  
 .values  
)  
  
#Convert Percentage into 2 decimal places  
  
urban\_rural\_jobs["PERCENT"] = urban\_rural\_jobs["PERCENT"].apply(lambda x:f"{x:.2f}%")  
  
# Visualization  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(12, 6))  
ax\_urban = sns.barplot(  
 data=urban\_rural\_jobs,  
 x="URBAN\_RURAL",  
 y="count",  
 hue="AI\_JOB",  
 palette="viridis"  
)  
  
plt.title("AI vs Non-AI Jobs in Urban vs Rural Areas")  
plt.xlabel("Urban or Rural")  
plt.ylabel("Number of Job Postings")  
plt.legend(title="Job Type")  
  
for container in ax\_urban.containers:  
 ax\_urban.bar\_label(container, fmt="%d", label\_type="edge", padding=3, fontsize=11)  
plt.tight\_layout()  
plt.savefig("../ad688\_group6\_geographic\_analysis/images/urbanai.png", dpi=300)  
plt.show()  
  
fig, ax = plt.subplots(figsize=(7,4))   
ax.axis('off')   
  
table = ax.table(  
 cellText=urban\_rural\_jobs.values,  
 colLabels=urban\_rural\_jobs.columns,  
 cellLoc='center',  
 loc='center',  
 colColours=["#0b2545"]\*len(urban\_rural\_jobs.columns), # Dark blue header  
 colWidths=[0.3]\*len(urban\_rural\_jobs.columns)  
)  
  
table.auto\_set\_font\_size(False)  
table.set\_fontsize(12)  
table.scale(1.2, 1.2)   
  
for key, cell in table.get\_celld().items():  
 if key[0] == 0:  
 cell.set\_fontsize(14)  
 cell.set\_text\_props(color='white')  
 cell.set\_facecolor('#0b2545')  
  
plt.show()





## Key Insights

This plot shows that the percentage of AI Jobs in Urban areas is actually almost exactly equal to the percentage of AI Jobs in rural areas. Even though the count of these types of jobs are much different, there are opportunities for job seekers to find AI related jobs at the same rate in rural areas compared to urban areas.