About

Group 2

2025-06-26

title: “Preliminary Research” author: “Group 2” date: June 26, 2025 format: html: bibliography: references.bib csl: csl/econometrica.csl toc: true —

# Problem

What does the job market look like, and how can we refine our skills to better match the career we want?

# Introduction

To answer this question, our group began by analyzing information on how the job market has evolved over recent years. A major change that has impacted the workforce is COVID-19, which has accelerated shifts toward remote work, automation, and digital evolution. During the pandemic, organizations were forced to adapt quickly, which in turn altered the way the workplace was structured and facilitated further technological implementation. Now that it’s been a few years since the pandemic, how are the working conditions distributed within the workforce? According to (Half (2024)), 26% of U.S. workers are fully remote while 29% are hybrid. This article also highlights that altough remote has decline since the pandemic, it still very much remains as an option in the workforce. A survey was conducted an 50% of workers said they would quit if they had to return to the office full-time. Similarly, workers also mentioned that they would accept a lower salary for remote options. This demonstrates how important working conditions are for employees, therefore it should be something management considers. This article also compared job titles to flexibility of work conditions. They discovered that 31% of newly created senior-level roles were hybrid, while 15% were fully remote. By analyzing the market, we can adapt effectively and refine our strategies; therefore, we need to assess whether this element of a job posting should be something to consider.

Artificial intelligence (AI) is a technology that allows computers and machines to mimic human abilities such as learning, understanding, solving problems, making decisions, being creative, and operating independently (Stryker (2024)). Over the next several years, generative AI will become the main focus of AI researchers and enthusiasts. Generative AI is a type of technology that can produce original text, images, and videos. With the rising popularity of generative AI, many companies are faced with the question of how to walk the line between technological advancements and ethical responsibility. Advocates of AI claim that when implemented properly, AI can boost productivity, accelerate product development, and improve business decision-making. In 2024, the International Monetary Fund estimated that 40% of jobs around the world will be affected by AI. Tech experts hypothesize that industries and sectors such as cybersecurity, financial services, and manufacturing are likely to be affected (Panel (2021)). On the other hand, some argue that while job displacement is inevitable, AI will also create new jobs. According to Lee, Samanta, and Lee Lee, Samanta, and Lee ((2024)), AI will not eliminate jobs but instead will reshape career structures. Even so, there are still job types that can be at risk of being replaced. An example of a high-risk job would be one that involves repetition and predictable tasks, such as telemarketing and a customer support specialist. On the other hand, the more resilient jobs are those that require emotional intelligence or creative problem-solving. Some examples of these careers are teachers, writers, and employees with skilled trades like plumbers. Another important aspect to consider is that there will be some jobs where AI is used in conjunction with their regular responsibilities; an example of this is teachers. Teachers may use AI to grade papers, but they still will need to perform their regular tasks, like lecturing and mentoring students. Overall, Lee, Samanta, and Lee (2024) conclude that the rise of AI has a twofold effect: potential job displacement and the creation of new opportunities.

The double-edged effect appears to be the sentiment for Ito Ito ((2025)), where she looks into the impact of job postings since OpenAI’s launch in 2022. Ito mentioned that she believed it would take a couple of years to truly see the impact of AI on the job market, but when evaluating in 2025, she found that AI’s revolution may have already begun. To confirm her hypothesis, Ito asked for Revelio Labs, an analytics provider, to see whether they can find jobs where AI has already replaced them. They first began by looking at current job descriptions to see which responsibilities AI could already replace or augment. Ultimately, they found that over the past three years, there was a 19% decline in jobs involving tasks that can be performed by AI. This drop was mainly due to companies hiring fewer roles that AI can handle. They also classified the jobs online into three categories: high exposure roles, low exposure roles, and those in between. Fundamentally, jobs with the highest exposure to AI started to disappear faster from online job postings. According to Revelio Labs, the jobs with the highest exposure are the ones that manage various technical functions like IT specialists, data engineers, and database administrators. On the other hand, the jobs with the lowest exposure are in-person roles like restaurant manager or mechanic. Essentially, there is data showing that AI has already disrupted the job market, but the article also emphasizes the point that its projection is still unknown. Experts cannot determine the sustainability of replacing humans with AI, especially when it comes to the quality of service that is produced.

From a geographic perspective, McElheran et al. McElheran et al. ((2024)) offers important insight into how AI adoption appears to be concentrated in select cities. This academic journal examines the 2018 Annual Business Survey to not only find out geographically where the highest AI implementation is, but also which firms are leading the charge. The data used for their analysis looks at 850,000 U.S. firms from early AI adoption to now. The journal classifies AI adoption by whether a firm has used at least one AI-related technology, including: machine learning, machine vision, automated guided vehicles, natural language processing, and voice recognition. Their findings showed that San Fransico was ranked as the top adopter of AI especially when it came to larger firms. Other notable cities with a high adoption rate were Nashville, TN; San Antonio, TX; and Tampa, FL. These findings suggest that early adopters of AI are large firms or high-growth startups in regions where there is more AI exposure.

## Purpose

With this in mind, our research will look at the rising trends within the job market, with a particular focus on roles that align with our own career interests. This includes analyst positions, some of which may be influenced by advances in AI. We will also look at what working conditions, more specifically remote versus not remote jobs.

## References

# Introduction

Our group has decided to evaluate the distribution of jobs related to techonology across the United States. In order to do this, we used a count of any jobs containing the word “analyst” and categorized them by state. The results of this are shown below.

Analyst Job Distribution Across the United States

According to the visual above, Texas and California are the two clear leaders in the total amount of jobs being offered that contain the word “analyst” in the title. Additionally, eastern states show a considerably greater amount of these jobs compared to western states.

The chart above displays the salary breakdown between states for job postings impacted and not impacted by AI. The size of the data points illustrates the number of job postings that each state has. Interestingly, both Texas and California have the largest amount of job postings not affected by AI. We can conclude that a possiblity for this is that in those tech hubs, the jobs AI can replace decreased in demand.

Analyst Job Distribution Across Cities in the United States

New York City is the clear leader in analyst jobs available. Another insight is that Texas has 3 cities in the top 10, which indicates a strong job market. Something that surprised us, is that San Francisco finishes lower than expected by rounding out the bottom of the top 10.

# Remote VS Onsite job

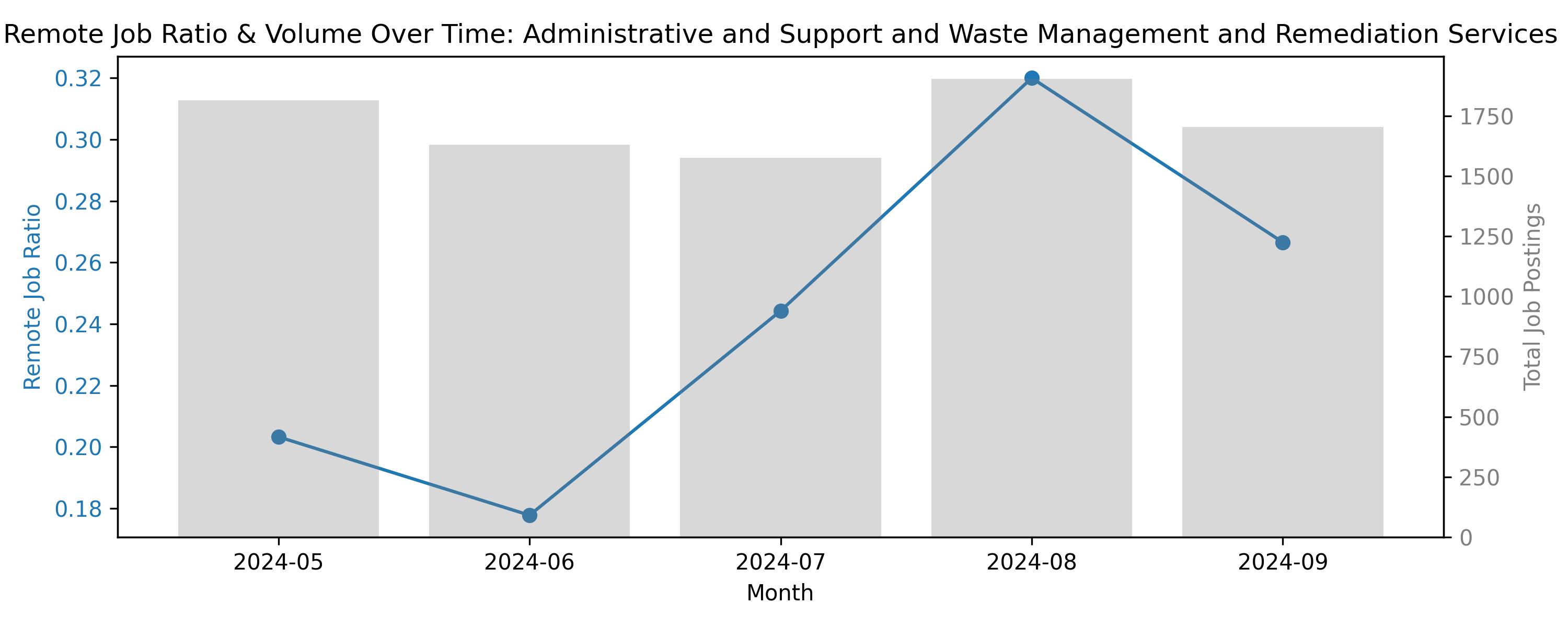
## Remote Job Distribution by General Industry

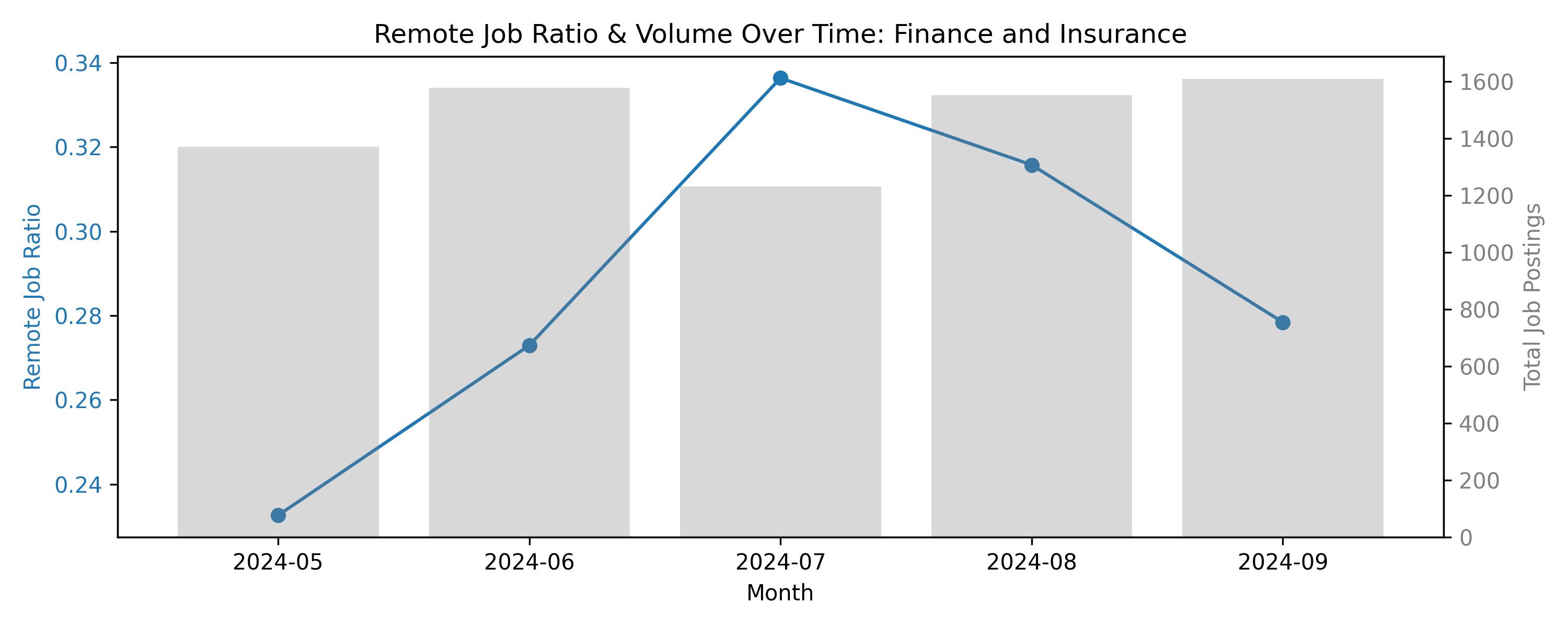
In terms of the Remote Job Distribution by general industry, the top 3 industries are Real Estate and Rental and Leasing, Information, Finance and Insurance. As we can see even the traditional industries such as manufacturing, educational services starte to have higher remote job proportion, which means the remote work nowadays has already became a widespread jobs or requirements across multiple industries.

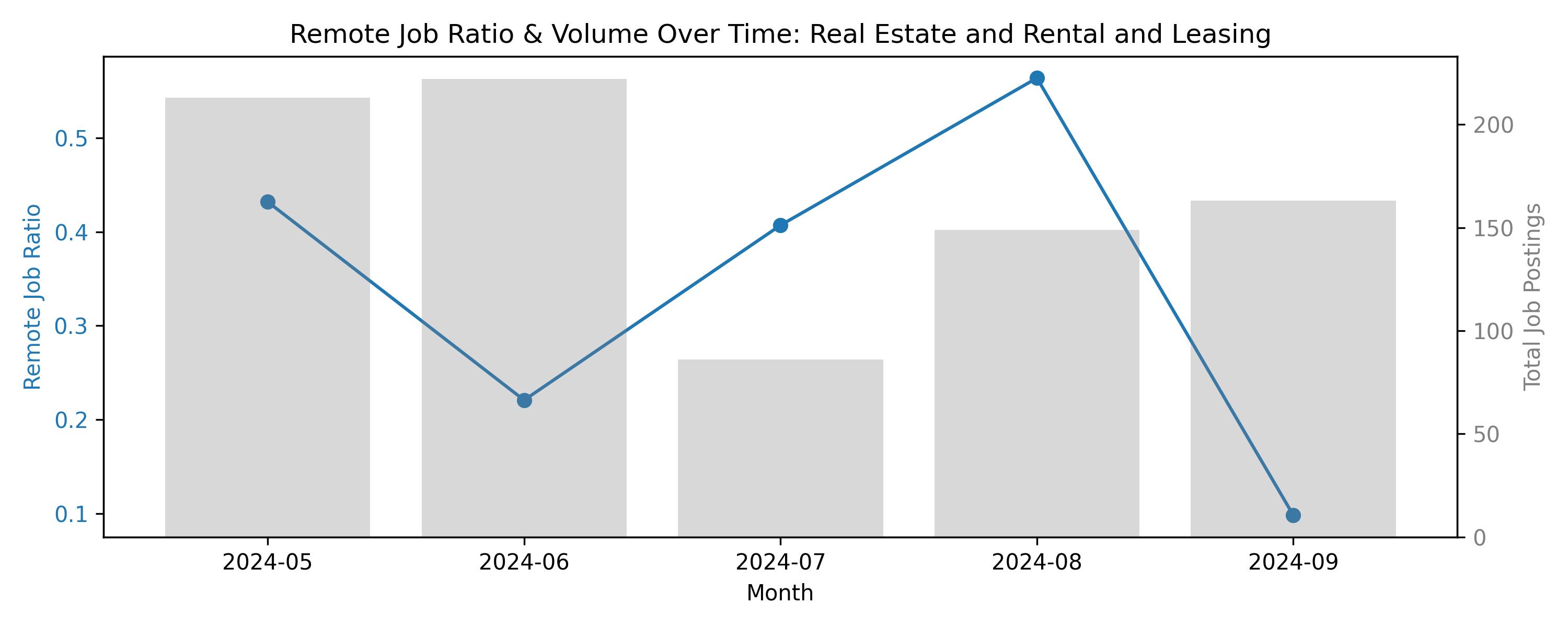
###Top Industries: Remote job Trends Over time

Take a dive into the Top industries over time with specific industries, in general, this plot displays monthly trends in job postings for from May 2024 to September 2024. As we can see in here, the remote job ratio was represented by the y-axis, which is the proportion of job postings marked as remote out of total postings in each industry per month. Higher remote ratio, represents their have more remote job opportunities open in each industry.

Indeed, we can see the general trend in this plot as the remote job among different industries, which reaches the peak in Auguest 2024 and then drop siginificantly in September, which suggests that the many companies end their summer recruitings. As we can see the real estate and rental industry goes from high and crash in September. In contrast, the Finance and Insurance industry has higher stability， which has consistent job postings from May to September.

 From the above plot result, we can see that the remote job ratio trend started low in May, then continue drop low in June, then reached the peak in August (31%), after that drop low in September. In compare with the trend alighed with gray area: job postings are also have more postings in Auguest and less in September.

 From the above plot result, we can see that the remote job ratio trend started low in May, gradually increase in June, then reached the peak in July (33%), after that gradually drop low August. The job posting area remained high and stable through july, which is friendly for people who are seek for remote jobs.

 From the above plot result, we can see that the remote job ratio trend started high (43%) in May, drop in June, then rebound in July then reached the peak in August (53%), after that has a significantly drop in September(near 0). The job posting area remained comparable stable, which has high volumn in May and June, then has a siginificant drop in July and modest increase in Auguest and September. The differences between Job posting

From above choreleograph, we can see that the green color depth represents the remote jobs proportions to their job postings, and with the average salary and average job experiences. To be more clearly, we can see that the Maine, Vermont, Alaska and Colorado have higher remote ratio, which represents their high demands on remote jobs souther state on the othe hand has lighter remote ratio, which represents that there are fewer remote job demands in these regions’ industries.

import findspark  
findspark.init()  
  
from pyspark.sql import SparkSession  
import pandas as pd  
import plotly.express as px  
import plotly.io as pio  
import numpy as np  
  
np.random.seed(423548)  
  
pio.renderers.default = "notebook"  
  
# Initialize Spark Session  
spark = SparkSession.builder.appName("LightcastData").getOrCreate()  
  
# Load Data  
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("./data/lightcast\_job\_postings.csv")  
  
  
df.printSchema() # comment this line when rendering the submission  
df.show(5)

# clean the data  
  
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler  
from pyspark.ml import Pipeline  
from pyspark.sql.functions import when, col  
  
df= df.drop("ID" ,"LAST\_UPDATED\_DATE","LAST\_UPDATED\_TIMESTAMP","DUPLICATES", "EXPIRED","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")  
  
df = df.withColumn("REMOTE\_TYPE\_NAME",   
 when(col("REMOTE\_TYPE\_NAME") == "[None]", "Not Remote")  
 .otherwise(col("REMOTE\_TYPE\_NAME")))  
  
df\_clean = df.na.drop(subset=[  
 "salary", "MIN\_YEARS\_EXPERIENCE", "MAX\_YEARS\_EXPERIENCE",  
 "EDUCATION\_LEVELS\_NAME", "EMPLOYMENT\_TYPE\_NAME", "REMOTE\_TYPE\_NAME",  
 "DURATION",  
])  
  
df\_clean.show(5)

titles = df\_clean.select("TITLE\_NAME").distinct().rdd.flatMap(lambda x: x).collect()  
for title in titles:  
 print(title)

#gave chatgpt the list and told it to split into ai impacted and not  
ai\_impacted\_jobs = [  
 'Trust Officers',  
 'Cloud Migration Engineers',  
 'EDI Developers',  
 'Permit Specialists',  
 'Data Integration Leads',  
 'Blockchain Developers',  
 'Lead Intelligence Analysts',  
 'Sales Architects',  
 'Business Intelligence Leads',  
 'Data Analyst Managers',  
 'Project Support Analysts',  
 'Sales Planning Analysts',  
 'Hourly Managers',  
 'Industrial Equipment Mechanics',  
 'Value Engineers',  
 'Enterprise Applications Consultants',  
 'SAP FICO Consultants',  
 'SAP ABAP Consultants',  
 'Oracle Erp Consultants',  
 'Member Liaisons',  
 'Data Solutions Analysts',  
 'Transformation Analysts',  
 'GIS Data Analysts',  
 'Japanese Bilingual Administrative Assistants',  
 'Project Leads',  
 'People Analytics Analysts',  
 'Human Resources Reporting Analysts',  
 'Implementation Consultants',  
 'Lead Enterprise Architects',  
 'Branch Bankers',  
 'Growth Marketing Analysts',  
 'Chargeback Analysts',  
 'Strategy Leads',  
 'Innovation Analysts',  
 'Business Insights Managers',  
 'CRM Business Analysts',  
 'Localization Producers',  
 'EDI Analysts',  
 'Scientific Data Analysts',  
 'Bilingual Japanese Customer Service Representatives',  
 'SQL/ETL Developers',  
 'Data Quality Leads',  
 'Data Visualization Analysts',  
 'Data Analytics Engineers',  
 'Foundation Administrators',  
 'SQL Reporting Analysts',  
 'Procurement Analysts',  
 'Manual Testers',  
 'Analytics Associates',  
 'Supply Chain Architects',  
 'SAP SD Analysts',  
 'Oracle Cloud Financials Consultants',  
 'Data Quality Assurance Analysts',  
 'Client Finance Directors',  
 'Population Health Analysts',  
 'Enterprise Solutions Consultants',  
 'Digital Product Analysts',  
 'Line Pilots',  
 'Processing Clerks',  
 'Client Solutions Strategists',  
 'Business Intelligence Specialists',  
 'Accounting Consultants',  
 'Business Intelligence Analysts',  
 'SAP CRM Consultants',  
 'Human Capital Management Consultants',  
 'IAM Architects',  
 'SAP Ariba Consultants',  
 'Territory Assistants',  
 'Immigration Analysts',  
 'Customer Experience Associates',  
 'Global Analysts',  
 'Analysts',  
 'Document Management Clerks',  
 'Data Reporting Analysts',  
 'Quality Analysts',  
 'SAP HANA Consultants',  
 'Site Analysts',  
 'Business Architects',  
 'Data Analytics Developers',  
 'Digital Solution Architect Managers',  
 'Information Governance Analysts',  
 'Business Coaches',  
 'Configuration Management Analysts',  
 'Commercial Analysts',  
 'Analytics Consultants',  
 'Buyers',  
 'Enterprise Solutions Architects',  
 'Remediation Analysts',  
 'Search Planners',  
 'Insurance Associates',  
 'People Operations Generalists',  
 'Appeals Specialists',  
 'IT Project Assistants',  
 'Data Analytics Architects',  
 'Methods Analysts',  
 'Liquidity Analysts',  
 'Functional Consultants',  
 'SAP HR Consultants',  
 'Performance Directors',  
 'ERP Implementation Specialists',  
 'Tribal Administrators',  
 'Study Design Leads',  
 'Equipment Analysts',  
 'Quality Assurance Monitors',  
 'Integration Team Leads',  
 'Invoice Analysts',  
 'SAP FICO Functional Consultants',  
 'Enterprise Relationship Managers',  
 'Enterprise Data Architects',  
 'SAP PP/QM Consultants',  
 'Management Analysts',  
 'Speech Assistants',  
 'SAP Business Analysts',  
 'Data Migration Analysts',  
 'SAP Techno Functional Consultants',  
 'SAP Supply Chain Consultants',  
 'Enterprise Directors',  
 'Data Processing Analysts',  
 'SAS Data Analysts',  
 'Manufacturing Consultants',  
 'Demand Analysts',  
 'Netsuite Principal Consultants',  
 'OTM Consultants',  
 'Modernization Superintendents',  
 'Mapping Analysts',  
 'Enterprise Systems Architects',  
 'Business Intelligence Associates',  
 'Oracle Business Analysts',  
 'SAP Support Analysts',  
 'Automation Engineers',  
 'Excel VBA Developers',  
 'Financial Data Analysts',  
 'Power Analysts',  
 'Sales Analysts',  
 'Lead IT Analysts',  
 'Development Analysts',  
 'Analytics Managers',  
 'Financial Applications Specialists',  
 'Sales Solution Architects',  
 'Survey Analysts',  
 'Data Analysts/Data Scientists',  
 'Quality Control Reviewers',  
 'Oracle EBS Consultants',  
 'Data Services Analysts',  
 'ERP Implementation Managers',  
 'Knowledge Analysts',  
 'Enterprise Business Analysts',  
 'Test Data Analysts',  
 'Techno Functional Analysts',  
 'Netsuite Consultants',  
 'Cryptologists',  
 'PMO Analysts',  
 'Reference Data Analysts',  
 'Clinical Data Analysts',  
 'Enterprise Business Architects',  
 'Enterprise Cloud Architects',  
 'SAP Security Consultants',  
 'Data Leads',  
 'Tools Developers',  
 'Marketing Analysts',  
 'Financial Planning and Analysis Analysts',  
 'Finance Systems Analysts',  
 'Production Operators',  
 'Oracle Functional Analysts',  
 'IT Buyers',  
 'Process Engineers',  
 'Privacy Analysts',  
 'Enterprise Resources Planning Managers',  
 'Real Estate Analysts',  
 'Systems Integration Architects',  
 'End User Computing Analysts',  
 'Data Analysts/Developers',  
 'Publishing Specialists',  
 'SQL Analysts',  
 'Account Analysts',  
 'Engineering Data Analysts',  
 'Oracle EBS Business Analysts',  
 'Directors of Business Intelligence',  
 'Reporting Associates',  
 'SAP HCM Consultants',  
 'Feasibility Managers',  
 'Data Management Administrators',  
 'Walkers',  
 'Production Analysts',  
 'HRIS Associates',  
 'Data Analytics Leads',  
 'Data Analytics Specialists',  
 'Data Security Analysts',  
 'Principal Data Scientists',  
 'Researchers',  
 'Procurement Business Analysts',  
 'Oracle Applications Analysts',  
 'Forecast Analysts',  
 'Supply Chain Data Analysts',  
 'Analytics and Insights Managers',  
 'MDM Developers',  
 'Business Support Analysts',  
 'Food and Beverage Analysts',  
 'Intelligence Research Analysts',  
 'Validation Leads',  
 'Associate Business Managers',  
 'Enterprise Data Analysts',  
 'IT Governance Analysts',  
 'Domain Architects',  
 'Compliance Business Analysts',  
 'Implementation Specialists',  
 'Placement Managers',  
 'Corporate Architects',  
 'Splunk Developers',  
 'Work Force Management Analysts',  
 'Banking Consultants',  
 'Data Stewards',  
 'SAP MDM Consultants',  
 'SAP Specialists',  
 'Data Validation Analysts',  
 'Business Intelligence Data Warehouse Architects',  
 'Data Science Associates',  
 'Solution Leads',  
 'SAP Data Analysts',  
 'SAP Finance Consultants',  
 'Doctors',  
 'SQL Data Analysts',  
 'Patient Revenue Cycle Specialists',  
 'People Analytics Managers',  
 'Data Scientists',  
 'Digital Data Analysts',  
 'Data Control Clerks',  
 'Storeroom Clerks',  
 'Finance Business Analysts',  
 'SAP HR Analysts',  
 'Business Intelligence and Analytics Managers',  
 'Brand Activation Managers',  
 'Enterprise Project Managers',  
 'Data Analytics Consultants',  
 'Programmer Analysts',  
 'Sales Data Analysts',  
 'Data Reviewers',  
 'Contract Analysts',  
 'Decision Support Analysts',  
 'Data Associates',  
 'E-Commerce Architects',  
 'Risk Control Managers',  
 'Debt Specialists',  
 'Risk and Controls Managers',  
 'Data Entry Analysts',  
 'Platform Analysts',  
 'Financial Systems Analysts',  
 'Claims Resolution Analysts',  
 'Lead Business Intelligence Analysts',  
 'Inside Auto Claims Representatives',  
 'Customer Contact Center Managers',  
 'Data Governance Analysts',  
 'Business Operations Specialists',  
 'SAP Technical Consultants',  
 'Data Engineering Managers',  
 'SAP EWM Consultants',  
 'Tax Controllers',  
 'Transmission Analysts',  
 'Business Analysts',  
 'Credit Analysts',  
 'CSV Consultants',  
 'Patient Services Associates',  
 'Research Data Analysts',  
 'Medical Economics Analysts',  
 'SQL Administrators',  
 'SAP Master Data Analysts',  
 'Customer Care Analysts',  
 'Title Processors',  
 'Power Business Intelligence Developers',  
 'Data Quality Analysts',  
 'Quality Assurance Analysts',  
 'Consulting Engagement Managers',  
 'Vice Presidents of Operational Excellence',  
 'Insights Analysts',  
 'Inventory Analysts',  
 'Data Support Analysts',  
 'SAP Business One Consultants',  
 'Data Management Leads',  
 'Platform Administrators',  
 'Oracle Financial Functional Consultants',  
 'Architecture Managers',  
 'ERP Analysts',  
 'Growth Associates',  
 'Directors of School Nutrition',  
 'Operations and Policy Analysts',  
 'Lead Data Analysts',  
 'SAP SD Functional Consultants',  
 'Assistant Federal Security Directors',  
 'Oracle Cloud Architects',  
 'Quality Assurance Leads',  
 'SAP FICO Analysts',  
 'Repair Analysts',  
 'SAP SD Consultants',  
 'SAP SRM Consultants',  
 'Hyperion Consultants',  
 'SAP Systems Analysts',  
 'Adobe Experience Manager Architects',  
 'Quality Assurance Analysts/Engineers',  
 'Oracle Utilities Professional Services CC&B Implementation Consultants',  
 'SAP OTC Consultants',  
 'Data Warehouse Business Analysts',  
 'Client Data Analysts',  
 'Documentum Administrators',  
 'Project Analysts',  
 'Korean Linguists',  
 'Data Governance Managers',  
 'Oracle Consultants',  
 'Navigators',  
 'Customer Data Analysts',  
 'Enterprise Technical Architects',  
 'Cash Management Associates',  
 'Equity Traders',  
 'Financial Analysts',  
 'ERP Architects',  
 'Business Intelligence and Data Analysts',  
 'Automation Consultants',  
 'Governance Analysts',  
 'ERP Systems Analysts',  
 'Commercial Banking Credit Analysts',  
 'Certified Public Accountants',  
 'Data Operations Analysts',  
 'Category Analysts',  
 'Regional Business Managers',  
 'GCP Auditors',  
 'Architectural Project Coordinators',  
 'Delivery Analysts',  
 'SAP MM Functional Consultants',  
 'Techno Functional Leads',  
 'Clinical Data Managers',  
 'Administrators',  
 'Enterprise Network Managers',  
 'Healthcare Business Analysts',  
 'Solutions Architects',  
 'Ward Clerks',  
 'Account Coordinators',  
 'Data Abstractors',  
 'Data Scientists/Analysts',  
 'SAP Functional Consultants',  
 'Product Lifecycle Managers',  
 'Data Intelligence Analysts',  
 'Data Warehouse Analysts',  
 'CNC Administrators',  
 'Life Sciences Consultants',  
 'Data Processing Clerks',  
 'Production Planning Analysts',  
 'Business Growth Consultants',  
 'Chief Architects',  
 'Logistics Administrators',  
 'Vendor Analysts',  
 'Quantitative Analytics Managers',  
 'Treasury Consultants',  
 'Crime Analysts',  
 'SAP Business Process Analysts',  
 'Call Center Analysts',  
 'GRC Analysts',  
 'Competitive Intelligence Analysts',  
 'Oracle Technical Consultants',  
 'Headend Engineers',  
 'Claims Business Analysts',  
 'General Managers',  
 'Systems Architecture Engineers',  
 'SAP Ariba Managers',  
 'Customer Analytics Managers',  
 'Customer Experience Analysts',  
 'LIS Analysts',  
 'SAP Consultants',  
 'SAP Hybris Consultants',  
 'SAP PI Consultants',  
 'Enterprise Application Analysts',  
 'Data and Analytics Consultants',  
 'Agile Product Owners',  
 'Business Intelligence Reporting Analysts',  
 'Management Associates',  
 'Health Data Analysts',  
 'Enterprise Systems Engineers',  
 'Lead Analysts',  
 'Data Migration Consultants',  
 'Professors of Biological Sciences',  
 'Oracle Business Systems Analysts',  
 'Regional Fleet Managers',  
 'Enterprise Systems Analysts',  
 'Inflight Supervisors',  
 'Operations Process Engineers',  
 'SAP Developers',  
 'Customer Care Experts',  
 'Reporting Analysts',  
 'Commercial Excellence Managers',  
 'Lead Quality Analysts',  
 'Oracle EBS Technical Consultants',  
 'Records Assistants',  
 'Directors of Cloud Security',  
 '.NET Technical Architects',  
 'Associates',  
 'Trend Analysts',  
 'Digital Analysts',  
 'Innovation Specialists',  
 'Data Governance Leads',  
 'Data Analysts',  
 'Directors of Sponsored Programs',  
 'Enterprise Resource Planning Managers',  
 'Wealth Management Analysts',  
 'Quantitative Data Analysts',  
 'Origination Associates',  
 'Enterprise Business Consultants',  
 'SAP Application Consultants',  
 'Information Analysts',  
 'Process Improvement Analysts',  
 'Consumer Analysts',  
 'Communications Analysts',  
 'GIS Consultants',  
 'Data Analyst Specialists',  
 'Plant Maintenance Managers',  
 'Netsuite Functional Consultants',  
 'Plant Chemists',  
 'Integration Consultants',  
 'Data Management Analysts',  
 'Oracle Financial Consultants',  
 'SAP Plant Maintenance Consultants',  
 'Analytics Leads',  
 'Lead Level Designers',  
 'Solutions Analysts',  
 'Data and Reporting Analysts',  
 'Data Analytics Product Managers',  
 'SAP Analysts',  
 'Implementation Analysts',  
 'Revenue Cycle Analysts',  
 'Case Management Associates',  
 'SAP Successfactors Consultants',  
 'IT Governance Managers',  
 'Sales Professionals',  
 'Principal Architects',  
 'Oracle HCM Consultants',  
 'Data Science Analysts',  
 'HANA Consultants',  
 'OSP Managers',  
 'Interface Analysts',  
 'Transportation Systems Analysts',  
 'Forms Analysts',  
 'Master Data Coordinators',  
 'Enterprise Application Architects',  
 'Epic Security Analysts',  
 'Intercompany Accountants',  
 'Client Insights Analysts',  
 'Data Analytics Associates',  
 'SAP SD/MM Consultants',  
 'SAP Administrators',  
 'Supply Chain Business Managers',  
 'Assistant Service Experience Managers',  
 'Data Clerks',  
 'Data Visualization Specialists',  
 'Growth Specialists',  
 'SAP Basis Leads',  
 'Functional Analysts',  
 'SAP Functional Analysts',  
 'Knowledge Experts',  
 'Artificial Intelligence Engineers',  
 'Financial Planning and Analysis Managers',  
 'Business Intelligence Data Analysts',  
 'Peoplesoft Business Analysts',  
 'Platform Architects',  
 'Systems Architects',  
 'Assessment Analysts',  
 'Solutions Consultants',  
 'Surface Designers',  
 'SAP GTS Consultants',  
 'Data Insights Analysts',  
 'Oracle Cloud Technical Consultants',  
 'Data Modelers',  
 'Distribution Consultants',  
 'ERP Administrators',  
 'SAP MM Consultants',  
 'ERP Consultants',  
 'Financial Planning Analysts',  
 'Intelligence Analysts',  
 'SAP Basis Consultants',  
 'Consulting Technical Managers',  
 'Insights and Reporting Analysts',  
 'Advanced Analytics Analysts',  
 'Medical Data Analysts',  
 'Solution Architects/Principal Consultants',  
 'Technical Architects',  
 'Data and Analytics Managers',  
 'Travel and Expense Analysts',  
 'OFSAA Business Analysts',  
 'Facilities HVAC Technicians',  
 'Data Analytics Analysts',  
 'Web Data Analysts',  
 'Staff Data Engineers',  
 'Program Data Analysts',  
 'Associate Consultants',  
 'Safety Data Analysts',  
 'Controllers',  
 'CAD Operators',  
 'Security Analysts',  
 'Oracle Finance Functional Consultants',  
 'Enterprise Architects',  
 'Finance Data Analysts',  
 'ERP Business Systems Analysts',  
 'IT Data Analysts',  
 'Business Analysis Interns',  
 'Information Services Analysts',  
 'Data Solutions Consultants',  
 'Enterprise Program Managers',  
 'SAP MM Analysts',  
 'IT Applications Specialists',  
 'Oracle EBS Functional Consultants',  
 'Research and Evaluation Analysts',  
 'Customer Support Leads',  
 'Oracle SCM Functional Consultants',  
 'IT Data Analytics Analysts',  
 'Finance Assistants',  
 'Business Operations Associates',  
 'Quality Assurance Automation Testers',  
 'Technical Project Managers',  
 'Data Science and Analytics Managers',  
 'Oracle Cloud HCM Consultants',  
 'Business Intelligence Managers',  
 'Customer Support Administrators',  
 'Procurement Operations Associates',  
 'Analytics and Reporting Analysts',  
 'Research Specialists',  
 'BSA Consultants',  
 'Oracle Systems Analysts',  
 'Bilingual Office Managers',  
 'Institutional Sales Analysts',  
 'Consulting Solutions Architects',  
 'Corporate Finance Associates',  
 'Immunology Specialists',  
 'Global Trade Analysts',  
 'Search Coordinators',  
 'Higher Education Consultants',  
 'Enterprise Risk Analysts',  
 'ERP Support Specialists',  
 'Spanish and English Teachers',  
 'Assistant Vice Presidents',  
 'Data Protection Specialists',  
 'Manufacturing Services Managers',  
 'Cash Managers',  
 'Pricing Data Analysts',  
 'Directors of Toxicology',  
 'Data Acquisition Analysts',  
 'Process Analysts',  
 'Data Technicians',  
 'Clinical Quality Analysts',  
 'ERP Specialists',  
 'IAM Analysts',  
 'Data Infrastructure Engineers',  
 'Industry Analysts',  
 'Oracle Database Developers',  
 'Platform Support Specialists',  
 'School Education Managers',  
 'Clinical Data Associates',  
 'Oracle Functional Consultants',  
 'Enterprise Analysts',  
 'Manufacturing Analysts',  
 'Targeting Analysts',  
 'Master Data Analysts',  
 'SAP Basis Administrators',  
 'Data Center Analysts',  
 'Business Assistants',  
 'Philanthropy Assistants',  
 'IT Analysts',  
 'Middle Office Analysts',  
 'Investment Data Analysts',  
 'Salesforce Consultants'  
]

from pyspark.sql.functions import when, col, lit, avg, count  
  
df\_final = df\_clean.withColumn(  
 'ai\_impacted',  
 when(col('title\_name').isin(ai\_impacted\_jobs), lit('yes')).otherwise(lit('no'))  
)  
  
avg\_salary = df\_final.groupBy("state\_name").agg(avg("salary").alias("avg\_salary"))  
  
  
df\_final = df\_final.join(avg\_salary, on="state\_name", how="left")  
  
count = df\_final.groupBy("state\_name").agg(count("\*").alias("count"))  
  
  
df\_final = df\_final.join(count, on="state\_name", how="left")  
  
df\_final.select("count").distinct().show()

df\_final\_pd = df\_final.toPandas()  
  
fig = px.scatter(  
 df\_final\_pd,   
 x="STATE\_NAME",   
 y="avg\_salary",   
 size="count",   
 color="ai\_impacted",   
 title="Average Salary by State and AI Impact",  
 size\_max=60,  
 color\_discrete\_map={  
 "yes": "#2dbf78",   
 "no": "#8adfbd"   
 }  
  
)  
  
fig.update\_layout(  
 legend\_title\_text="AI Impacted",  
 xaxis\_title="State",  
 yaxis\_title="Average Salary",  
 xaxis=dict(tickangle=45),  
 yaxis=dict(categoryorder='category descending')  
)  
  
fig.show()

|  |
| --- |
| import findspark findspark.init()  from pyspark.sql import SparkSession import pandas as pd import plotly.express as px import plotly.io as pio import numpy as np  np.random.seed(42)  pio.renderers.default = "notebook"  # Initialize Spark Session spark = SparkSession.builder.appName("LightcastData").getOrCreate()  # Load Data df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("./data/lightcast\_job\_postings.csv")  # Show Schema and Sample Data print("---This is Diagnostic check, No need to print it in the final doc---")  df.printSchema() # comment this line when rendering the submission df.show(5)  Figure 1 |

import pandas as pd  
df = pd.read\_csv("./data/lightcast\_job\_postings.csv")

import pandas as pd  
  
df = pd.read\_csv("./data/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"  
]  
  
df.drop(columns=columns\_to\_drop, inplace=True)

print(df.columns.tolist())

!pip install missingno

import missingno as msno  
import matplotlib.pyplot as plt  
# Visualize missing values  
msno.heatmap(df)  
plt.title("Missing Values Heatmap")  
plt.show()  
  
# Drop columns with >50% missing values  
df.dropna(thresh=len(df) \* 0.5, axis=1, inplace=True)  
  
# Fill only the columns you actually have  
if 'Industry' in df.columns:  
 df["Industry"].fillna("Unknown", inplace=True)  
 df["Salary"].fillna(df["Salary"].median(), inplace=True)

df = df.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first")

df = df[df['NAICS\_2022\_2\_NAME'] != 'Unclassified Industry']

df['REMOTE\_TYPE\_NAME'] = df['REMOTE\_TYPE\_NAME'].replace('[None]', 'Not Remote')

import pandas as pd  
import plotly.express as px  
  
# Step 1: Prepare data  
data = {  
 'Industry': [  
 'Wholesale Trade', 'Retail Trade', 'Real Estate and Rental and Leasing',  
 'Professional, Scientific, and Technical Services', 'Manufacturing',  
 'Information', 'Health Care and Social Assistance',  
 'Finance and Insurance', 'Educational Services',  
 'Administrative and Support and Waste Management and Remediation Services'  
 ],  
 'Flexible Remote': [87.8, 94.4, 97.6, 92.2, 89.7, 95.8, 92.1, 94.8, 89.0, 94.8],  
 'Onsite': [12.2, 5.6, 2.4, 7.8, 10.3, 4.2, 7.9, 5.2, 11.0, 5.2]  
}  
  
df = pd.DataFrame(data)  
  
# Step 2: Sort in ascending order of Flexible Remote  
df\_sorted = df.sort\_values(by='Flexible Remote', ascending=True)  
df\_sorted['Industry'] = pd.Categorical(df\_sorted['Industry'], categories=df\_sorted['Industry'], ordered=True)  
  
# Step 3: Melt data for stacked bar format  
df\_melted = df\_sorted.melt(  
 id\_vars='Industry',  
 value\_vars=['Flexible Remote', 'Onsite'],  
 var\_name='Remote Type',  
 value\_name='Percentage'  
)  
  
# Step 4: Plot  
fig = px.bar(  
 df\_melted,  
 x='Percentage',  
 y='Industry',  
 color='Remote Type',  
 orientation='h',  
 text='Percentage',  
 color\_discrete\_map={  
 'Flexible Remote': '#1aab89',  
 'Onsite': '#88d4c3'  
 },  
 title="Remote Job Distribution by Industry (Top 10 Industries)"  
)  
  
# Step 5: Layout adjustments  
fig.update\_layout(  
 xaxis\_title="Percentage of Jobs",  
 yaxis\_title="",  
 xaxis=dict(tickformat=".0f"),  
 legend\_title="Remote Type",  
 barmode='stack',  
 margin=dict(l=10, r=10, t=60, b=40),  
 height=500  
)  
  
# Step 6: Label formatting  
fig.update\_traces(texttemplate='%{text:.1f}%', textposition='inside')  
  
# Save plot  
fig.write\_html("./figures/top\_industries.html")  
  
  
# Show plot  
fig.show()

import pandas as pd  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import classification\_report, confusion\_matrix, f1\_score, accuracy\_score  
import matplotlib.pyplot as plt  
import seaborn as sns

#read files  
file\_path = "./data/lightcast\_job\_postings.csv"  
df = pd.read\_csv(file\_path)

print(df['REMOTE\_TYPE\_NAME'].value\_counts(dropna=False).head(10))

# Step 1: Standardize formatting  
df['REMOTE\_TYPE\_NAME'] = (  
 df['REMOTE\_TYPE\_NAME']  
 .astype(str)  
 .str.strip()  
 .str.title()  
 .replace({'None': pd.NA, 'Nan': pd.NA})  
)

# Step 2: Fill missing or ambiguous entries with 'Not Remote'  
df['REMOTE\_TYPE\_NAME'] = df['REMOTE\_TYPE\_NAME'].fillna('Not Remote')  
df.loc[df['REMOTE\_TYPE\_NAME'] == "[None]", 'REMOTE\_TYPE\_NAME'] = "Not Remote"  
print(df['REMOTE\_TYPE\_NAME'].value\_counts(dropna=False).head(10))

# Convert all values to strings and strip whitespace  
df['REMOTE\_TYPE\_NAME'] = df['REMOTE\_TYPE\_NAME'].astype(str).str.strip()

# Apply new classification logic  
df['REMOTE\_BINARY'] = df['REMOTE\_TYPE\_NAME'].apply(  
 lambda x: 1 if x in ['Remote', 'Hybrid Remote'] else 0  
)

print(df['REMOTE\_TYPE\_NAME'].value\_counts())  
print("\nBinary classification:")  
print(df['REMOTE\_BINARY'].value\_counts())

import pandas as pd  
import numpy as np  
  
# Ensure salary columns are numeric and handle missing values  
df['SALARY\_FROM'] = pd.to\_numeric(df['SALARY\_FROM'], errors='coerce').replace(0, np.nan)  
df['SALARY\_TO'] = pd.to\_numeric(df['SALARY\_TO'], errors='coerce').replace(0, np.nan)  
  
# Calculate average salary (mean of SALARY\_FROM and SALARY\_TO)  
df['AVERAGE\_SALARY'] = df[['SALARY\_FROM', 'SALARY\_TO']].mean(axis=1)  
  
# Drop rows with missing values in AVERAGE\_SALARY, REMOTE\_TYPE\_NAME, or STATE\_NAME  
df\_salary = df.dropna(subset=['AVERAGE\_SALARY', 'REMOTE\_TYPE\_NAME', 'STATE\_NAME'])  
  
# Group by state and remote type, then calculate average salary  
avg\_salary\_by\_state\_remote = df\_salary.groupby(['STATE\_NAME', 'REMOTE\_TYPE\_NAME'])['AVERAGE\_SALARY'].mean().reset\_index()  
  
# Round the results for easier reading  
avg\_salary\_by\_state\_remote['AVERAGE\_SALARY'] = avg\_salary\_by\_state\_remote['AVERAGE\_SALARY'].round(2)  
  
# Show results  
print(avg\_salary\_by\_state\_remote)

df = df.merge(avg\_salary\_by\_state\_remote,  
 on=['STATE\_NAME', 'REMOTE\_TYPE\_NAME'],  
 how='left')

df = df.merge(  
 avg\_salary\_by\_state\_remote,  
 on=['STATE\_NAME', 'REMOTE\_TYPE\_NAME'],  
 how='left',  
 suffixes=('', '\_STATE\_REMOTE\_AVG')  
)

[col for col in df.columns if 'AVG' in col or 'SALARY' in col]

df = df.rename(columns={'AVERAGE\_SALARY\_y': 'AVERAGE\_SALARY\_STATE\_REMOTE\_AVG'})

import pandas as pd  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import OneHotEncoder, StandardScaler  
from sklearn.compose import ColumnTransformer  
from sklearn.pipeline import Pipeline  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.metrics import (accuracy\_score, f1\_score, confusion\_matrix,  
 classification\_report, precision\_score,  
 recall\_score, balanced\_accuracy\_score)  
from sklearn.inspection import permutation\_importance  
import seaborn as sns  
import matplotlib.pyplot as plt

# Remove duplicate column names across full dataframe  
df = df.loc[:, ~df.columns.duplicated()]

df['AVG\_YEARS\_EXPERIENCE'] = (df['MIN\_YEARS\_EXPERIENCE'] + df['MAX\_YEARS\_EXPERIENCE']) / 2  
df['EXP\_SPREAD'] = df['MAX\_YEARS\_EXPERIENCE'] - df['MIN\_YEARS\_EXPERIENCE']

df = df.drop(columns=['MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE'])

num\_feats = [  
 'AVG\_YEARS\_EXPERIENCE',  
 'AVERAGE\_SALARY\_STATE\_REMOTE\_AVG',  
 'IS\_INTERNSHIP'  
]  
  
cat\_feats = [  
 'STATE\_NAME',  
 'NAICS\_2022\_2\_NAME',  
 'EDUCATION\_LEVELS\_NAME',  
 'COMMON\_SKILLS\_NAME',  
 'SOFTWARE\_SKILLS\_NAME',  
 'TITLE\_CLEAN'  
   
]  
  
X = df[num\_feats + cat\_feats]  
y = df['REMOTE\_BINARY']

from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import StandardScaler, OneHotEncoder  
from sklearn.compose import ColumnTransformer  
from sklearn.pipeline import Pipeline  
from sklearn.ensemble import RandomForestClassifier

preprocess = ColumnTransformer(transformers=[  
 ("num", StandardScaler(), num\_feats),  
 ("cat", OneHotEncoder(handle\_unknown='ignore', sparse\_output=False), cat\_feats)  
])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 X, y, test\_size=0.2, random\_state=42, stratify=y  
)

from sklearn.ensemble import RandomForestClassifier  
# Preprocessing step  
preprocess = ColumnTransformer(transformers=[  
 ("num", StandardScaler(), num\_feats),  
 ("cat", OneHotEncoder(handle\_unknown='ignore', max\_categories=500, sparse\_output=False), cat\_feats)  
])  
clf = RandomForestClassifier(random\_state=42, class\_weight='balanced')

from sklearn.ensemble import RandomForestClassifier  
from sklearn.pipeline import Pipeline  
  
rf = RandomForestClassifier(  
 n\_estimators=200,  
 max\_depth=None,  
 class\_weight='balanced',  
 random\_state=42,  
 n\_jobs=-1  
)  
  
pipe = Pipeline(steps=[  
 ('prep', preprocess),  
 ('model', rf)  
])

pipe.fit(X\_train, y\_train)

y\_pred = pipe.predict(X\_test)  
  
# Classification report and confusion matrix  
print(classification\_report(y\_test, y\_pred))  
print("Confusion Matrix:")  
print(confusion\_matrix(y\_test, y\_pred))  
  
# Additional custom metrics  
print("\nCustom Metrics:")  
print("Accuracy:", round(accuracy\_score(y\_test, y\_pred), 3))  
print("F1 Score:", round(f1\_score(y\_test, y\_pred), 3))  
print("Precision:", round(precision\_score(y\_test, y\_pred), 3))  
print("Sensitivity (Recall 1):", round(recall\_score(y\_test, y\_pred), 3))  
print("Specificity (Recall 0):", round(  
 recall\_score(y\_test, y\_pred, pos\_label=0), 3))  
print("Balanced Accuracy:", round(balanced\_accuracy\_score(y\_test, y\_pred), 3))

cm = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')  
plt.xlabel("Predicted")  
plt.ylabel("True")  
plt.title("Confusion Matrix")  
plt.show()

rf\_model = pipe.named\_steps["model"] # RandomForestClassifier  
feature\_names = pipe.named\_steps["prep"].get\_feature\_names\_out()  
  
importances = rf\_model.feature\_importances\_  
  
feat\_imp = (  
 pd.DataFrame({"Feature": feature\_names, "Importance": importances})  
 .sort\_values(by="Importance", ascending=False)  
 .reset\_index(drop=True)  
)  
  
print("\nTop 9 – Tree-based Importances")  
print(feat\_imp.head(9).to\_string(index=False))

import seaborn as sns  
import matplotlib.pyplot as plt  
  
top\_n = 9 # change to show more/less  
plt.figure(figsize=(8, 6))  
sns.barplot(  
 data=feat\_imp.head(top\_n),  
 x="Importance", y="Feature",  
 palette="crest"  
)  
plt.title(f"Top {top\_n} Feature Importances (Random Forest)")  
plt.xlabel("Mean Decrease in Impurity")  
plt.ylabel("")  
plt.tight\_layout()  
plt.show()

import pandas as pd  
import plotly.express as px  
  
# Step 1: Create state abbreviation mapping  
us\_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'  
}  
  
# Step 2: Map state names to abbreviations  
df['STATE\_ABBR'] = df['STATE\_NAME'].map(us\_state\_abbrev)  
  
# Step 3: Group by state and compute metrics  
choropleth\_data = df.groupby('STATE\_ABBR').agg(  
 remote\_ratio=('REMOTE\_BINARY', 'mean'),  
 avg\_salary=('AVERAGE\_SALARY\_STATE\_REMOTE\_AVG', 'mean'),  
 avg\_experience=('AVG\_YEARS\_EXPERIENCE', 'mean'),  
 job\_count=('STATE\_NAME', 'count')  
).reset\_index()  
  
# Step 4: Define custom green scale (start from light, move to #1aab89)  
custom\_green\_scale = [  
 [0, "#e0f7f1"], # light mint  
 [0.5, "#70d8b5"], # mid-green  
 [1, "#1aab89"] # deep teal green  
]  
  
# Step 5: Create the choropleth with custom green  
fig = px.choropleth(  
 data\_frame=choropleth\_data,  
 locations='STATE\_ABBR',  
 locationmode="USA-states",  
 color='remote\_ratio',  
 color\_continuous\_scale=custom\_green\_scale,  
 scope="usa",  
 labels={'remote\_ratio': 'Remote Job Ratio'},  
 hover\_data={  
 'remote\_ratio': ':.2f',  
 'avg\_salary': ':.0f',  
 'avg\_experience': ':.1f',  
 'job\_count': True  
 },  
 title='Remote Job Ratio by State (Custom Green), Avg Salary & Experience in Hover'  
)  
  
fig.update\_layout(margin={"r":0,"t":40,"l":0,"b":0})  
fig.write\_html("./figures/state\_remote\_job\_ratio.html")  
  
fig.show()

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Step 1: Ensure 'POSTED' is in datetime format and create Year-Month  
df['POSTED'] = pd.to\_datetime(df['POSTED'])  
df['POSTED\_YM'] = df['POSTED'].dt.to\_period('M').astype(str)

industry\_trends = (  
 df.groupby(['NAICS\_2022\_2\_NAME', 'POSTED\_YM'])['REMOTE\_BINARY']  
 .mean()  
 .reset\_index(name='REMOTE\_RATIO')  
)

# Step 3: Select top 5 industries with highest overall average remote ratio  
top\_industries = (  
 industry\_trends.groupby('NAICS\_2022\_2\_NAME')['REMOTE\_RATIO']  
 .mean()  
 .sort\_values(ascending=False)  
 .head(5)  
 .index.tolist()  
)  
filtered\_trends = industry\_trends[industry\_trends['NAICS\_2022\_2\_NAME'].isin(top\_industries)]

import plotly.express as px  
  
fig = px.line(  
 filtered\_trends,  
 x='POSTED\_YM',  
 y='REMOTE\_RATIO',  
 color='NAICS\_2022\_2\_NAME',  
 markers=True,  
 title="Top Industries: Remote Job Trends Over Time"  
)  
  
fig.update\_layout(  
 xaxis\_title="Posted Month",  
 yaxis\_title="Remote Job Ratio",  
 legend\_title="Industry",  
 legend=dict(x=1.02, y=1, bordercolor="Black"),  
 margin=dict(l=40, r=40, t=60, b=40),  
 width=1000,  
 height=500  
)  
  
fig.update\_xaxes(tickangle=45)  
fig.write\_html("./figures/remote\_job\_over\_time.html")  
fig.show()

#Groupby industry + month and calculate both:  
industry\_month\_stats = df.groupby(['NAICS\_2022\_2\_NAME', 'POSTED\_YM']).agg(  
 TOTAL\_JOBS=('REMOTE\_BINARY', 'count'),  
 REMOTE\_RATIO=('REMOTE\_BINARY', 'mean')  
).reset\_index()

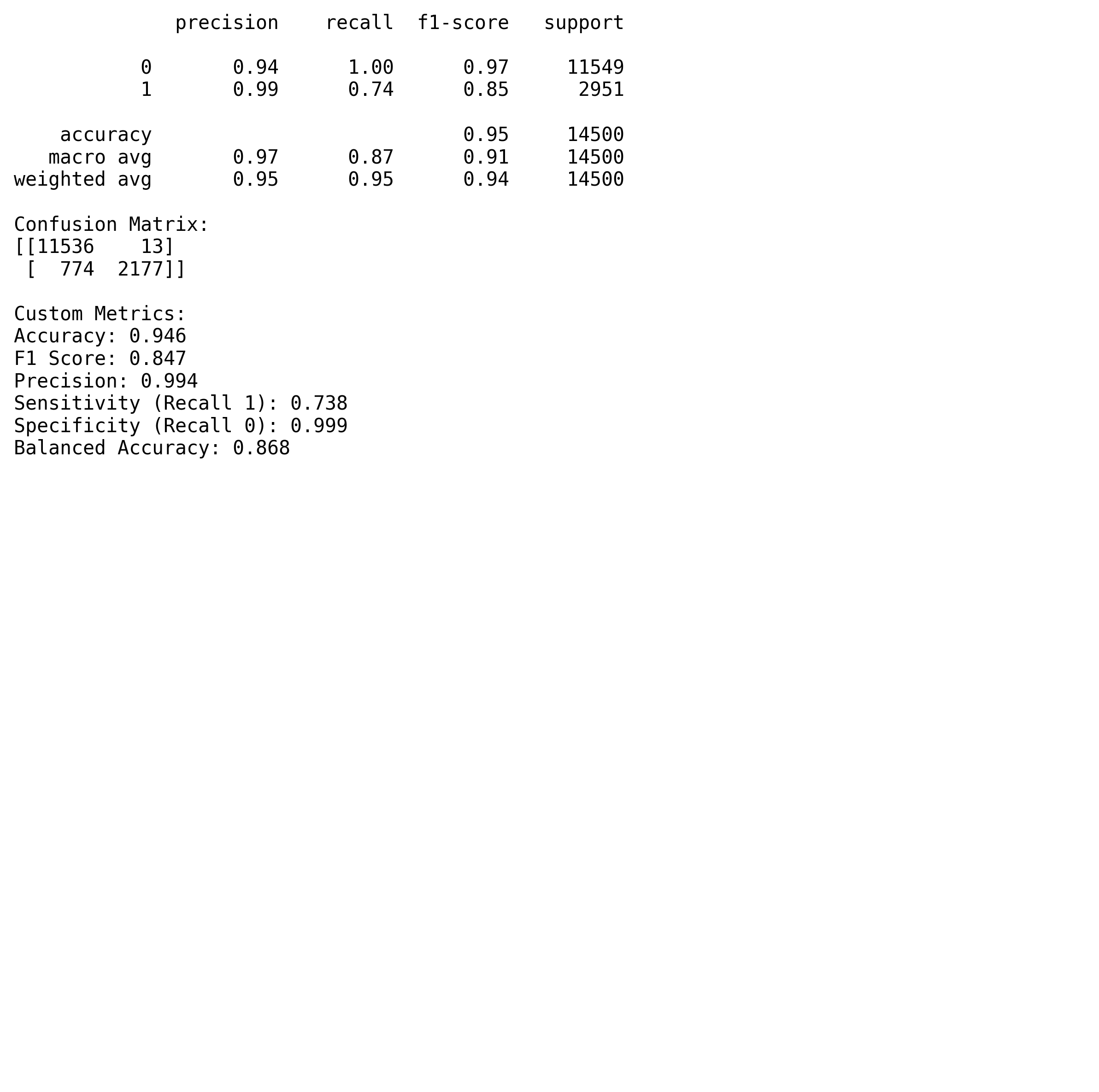
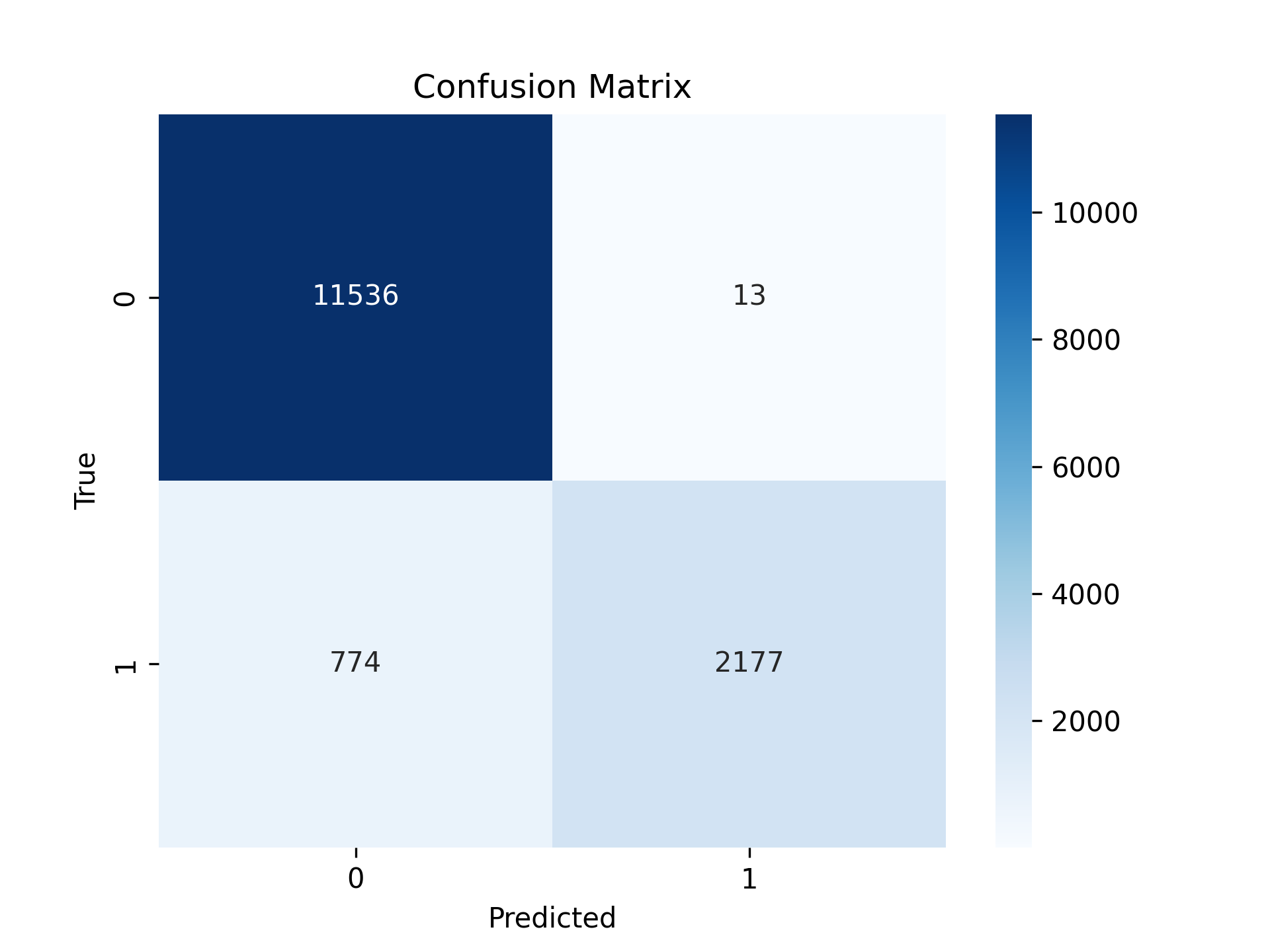
job\_count = df.groupby(['NAICS\_2022\_2\_NAME', 'POSTED\_YM']).size().reset\_index(name='JOB\_COUNT')

remote\_ratio = df.groupby(['NAICS\_2022\_2\_NAME', 'POSTED\_YM'])['REMOTE\_BINARY'].mean().reset\_index(name='REMOTE\_RATIO')

industry\_month\_stats = pd.merge(remote\_ratio, job\_count, on=['NAICS\_2022\_2\_NAME', 'POSTED\_YM'])

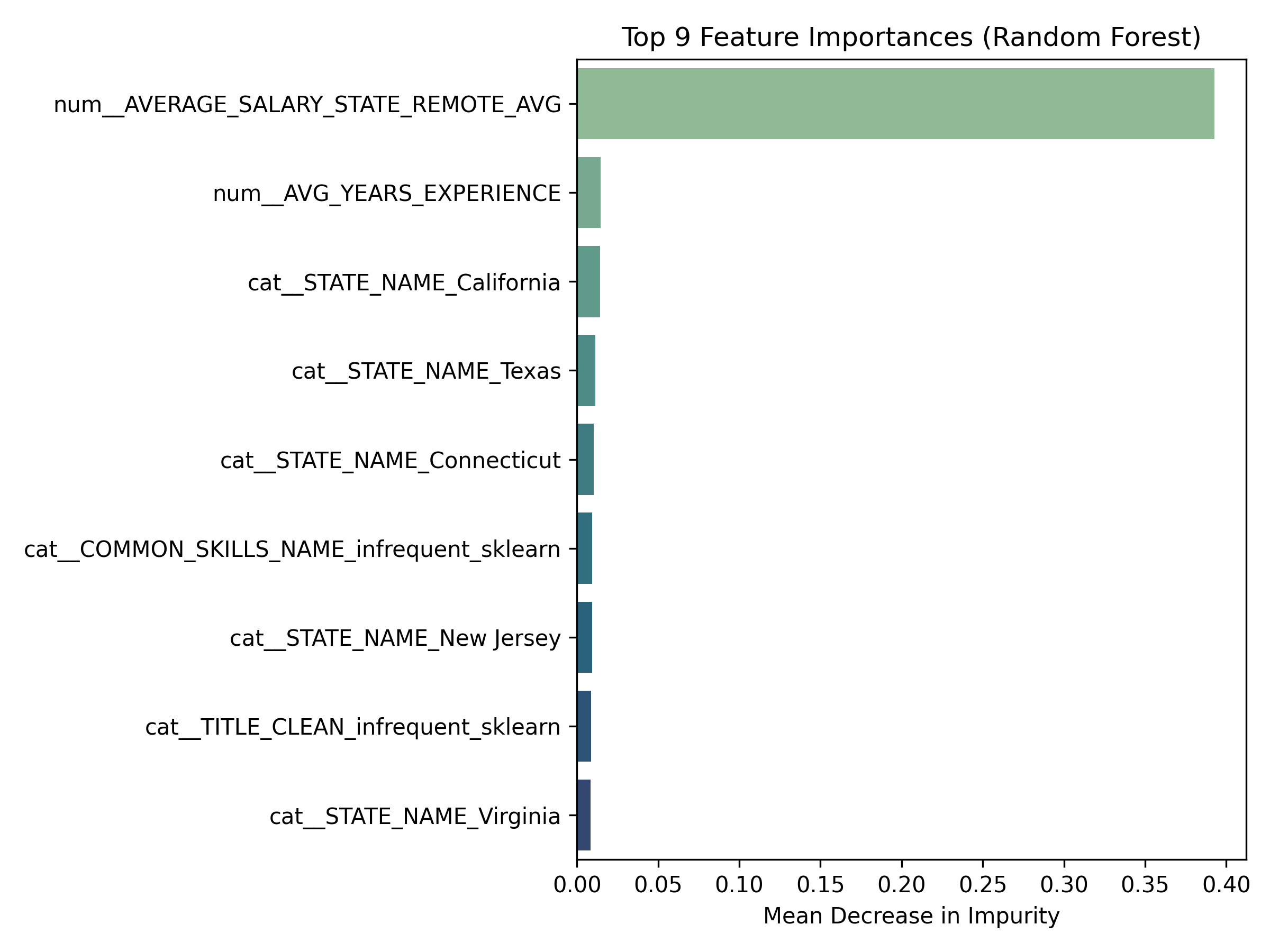
import matplotlib.pyplot as plt  
  
# Choose 2–3 industries to plot (or loop one at a time)  
selected\_industries = [  
 'Administrative and Support and Waste Management and Remediation Services',  
 'Arts, Entertainment, and Recreation',  
 'Finance and Insurance',  
 'Real Estate and Rental and Leasing',  
 'Utilities'  
]  
  
  
  
for industry in selected\_industries:  
 data = industry\_month\_stats[industry\_month\_stats['NAICS\_2022\_2\_NAME'] == industry]   
  
 fig, ax1 = plt.subplots(figsize=(10, 4))  
  
 # Plot remote ratio  
 ax1.plot(data['POSTED\_YM'], data['REMOTE\_RATIO'], color='tab:blue', marker='o')  
 ax1.set\_xlabel('Month')  
 ax1.set\_ylabel('Remote Job Ratio', color='tab:blue')  
 ax1.tick\_params(axis='y', labelcolor='tab:blue')  
 ax1.set\_title(f"Remote Job Ratio & Volume Over Time: {industry}")  
  
 # Plot job count on secondary y-axis  
 ax2 = ax1.twinx()  
 ax2.bar(data['POSTED\_YM'], data['JOB\_COUNT'], color='tab:gray', alpha=0.3)  
 ax2.set\_ylabel('Total Job Postings', color='gray')  
 ax2.tick\_params(axis='y', labelcolor='gray')  
  
 plt.xticks(rotation=45)  
 plt.tight\_layout()  
 plt.savefig("./figures/Remote\_jobs\_"+str(industry)+".jpg", dpi=300)  
 plt.show()

### Random Forest results

  As the two plots displayed a clear results for random forest results, our group has set our target variable for predict people’s preferences on Remote versus Onsite job, where 1 represents a remote job and 0 represents onsite. We also had a set of independent variables, including as: ‘AVG\_YEARS\_EXPERIENCE’, ‘AVERAGE\_SALARY\_STATE\_REMOTE\_AVG’, ‘IS\_INTERNSHIP’, ‘STATE\_NAME’, ‘NAICS\_2022\_2\_NAME’(industry) ‘EDUCATION\_LEVELS\_NAME’, ‘COMMON\_SKILLS\_NAME’, ‘SOFTWARE\_SKILLS\_NAME’, ‘TITLE\_CLEAN’ (occupation). Meanwhile, we split the data into training and testing sets using an 80/20 ratio to ensure generalizability, This means 20% of the data will go into the test set, and 80% will go into the training set. Then we conduct the randam forest model analysis. According to plots, we can conclude the accuracy reached to 94.6%;F1 score as 84.7%, which reflects the robust balance between precision and recall; the precision as 99.4%, which means it has highly accurate rate on predict the results;the sensitivity for class1 as 73.8%, which means it correctly identified the 74% people who pick remote; the sensitivity for class 0 even reached to 99.9%, which means almost all the people who choose non-remote job has correctly classified; balanced accuracy as 86.8%, which represents there is a balance performance between both cases. From the confusion matrix, there has a detailed display, which represents the model correctly predicts 11536 people who choose onsite jobs with only 13 false positives, and it also orrectly predicts 2177 people who choose remote jobs with 774 missed results.

One major limitation is class imbalance. Remote jobs (class 1) are the minority, which leads the model to perform less effectively on them.To counter this, we used class\_weight=‘balanced’ in our Random Forest to give more weight to underrepresented classes. We also built a preprocessing pipeline using ColumnTransformer for encoding categorical and scaling numerical features.Still, further optimization is needed. we recommended three ways to overcome this issue in the future study: Adjusting sample sizes;Tuning model hyperparameters, like n\_estimators and max\_categories; experimenting with resampling techniques; add crossvalidation steps to void overfitting and ensure the final results’ accuracy.

#### Featured Importance

 Based on the above plot, we can see the top 9 features could be essential in predicting the peoples’prefrences on remote or onsite jobs, we can see the top three essential features are average remote salary by state, Average years of experience and location state-California. These three features can be easily interpreted as the job’s salary, job’s requirement for year experiences and location are vital elements that impact people’s decision on onsite or remote job types, peple prioritize jobs with high salary and better geographic location

import findspark  
findspark.init()  
  
from pyspark.sql import SparkSession  
import pandas as pd  
import plotly.express as px  
import plotly.io as pio  
import numpy as np  
  
np.random.seed(42)  
  
pio.renderers.default = "notebook"  
  
# Initialize Spark Session  
spark = SparkSession.builder.appName("LightcastData").getOrCreate()  
  
# Load Data  
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("./data/lightcast\_job\_postings.csv")

df.createOrReplaceTempView("jobs")

software\_skill\_counts\_by\_type = spark.sql("""  
 SELECT software\_skills\_name, COUNT(\*) AS count  
 FROM jobs  
 WHERE LOWER(title\_name) LIKE '%analyst%'  
 OR LOWER(title\_name) LIKE '%analysis%'  
 OR LOWER(title\_name) LIKE '%analytics%'  
 GROUP BY software\_skills\_name  
 ORDER BY count DESC  
 LIMIT 10  
""")  
software\_skill\_counts\_by\_type.show(truncate=False)

skill\_counts\_by\_type = spark.sql("""  
 SELECT skills\_name, COUNT(\*) AS count  
 FROM jobs  
 WHERE LOWER(title\_name) LIKE '%analyst%'  
 OR LOWER(title\_name) LIKE '%analysis%'  
 OR LOWER(title\_name) LIKE '%analytics%'  
 GROUP BY skills\_name  
 ORDER BY count DESC  
 LIMIT 10  
""")  
skill\_counts\_by\_type.show(truncate=False)

import pandas as pd  
  
skills\_data = {  
 "Name": ["Alyssa", "Adam", "Yihan"],  
 "Microsoft Office": [4, 5, 3],  
 "Dashboard": [3, 3, 1],  
 "SQL": [2, 2, 2],  
 "OneStream": [1, 1, 1],  
 "Cloud Computing": [2, 2, 2]  
}  
  
df\_skills = pd.DataFrame(skills\_data)  
df\_skills.set\_index("Name", inplace=True)  
df\_skills

import pandas as pd  
import plotly.graph\_objects as go  
  
# Your data  
skills\_data = {  
 "Name": ["Alyssa", "Adam", "Yihan"],  
 "Microsoft Office": [4, 5, 3],  
 "Dashboard": [3, 3, 1],  
 "SQL": [2, 2, 2],  
 "OneStream": [0, 0, 0],  
 "Cloud Computing": [2, 2, 2]  
}  
  
# Create DataFrame  
df\_skills = pd.DataFrame(skills\_data)  
df\_skills.set\_index("Name", inplace=True)  
  
# Get skill categories  
categories = df\_skills.columns.tolist()  
  
# Create Plotly radar chart  
fig = go.Figure()  
  
for name in df\_skills.index:  
 values = df\_skills.loc[name].tolist()  
 values += values[:1] # close the radar loop  
  
 fig.add\_trace(go.Scatterpolar(  
 r=values,  
 theta=categories + [categories[0]], # close the loop  
 fill='toself',  
 name=name  
 ))  
  
# Customize layout  
fig.update\_layout(  
 polar=dict(  
 radialaxis=dict(  
 visible=True,  
 range=[0, 6]  
 )),  
 showlegend=True,  
 title="Team Skillset Levels"  
)  
  
  
  
fig.write\_html("./figures/skill\_gap.html", include\_plotlyjs='cdn')  
  
#https://plotly.com/python/radar-chart/

The radar chart above displays each individual evaluation of our skills for the top five skills on demand for analyst roles.

# Recomendations

Given our analysis above, we have decided to focus on some of the key actions and learning goals that each of us can take in order to further our chances of landing a high quality position in our chosen industry.

### SQL

#### Beginner

We recommend using tools such as SQLBolt to begin developing a foundational understanding of basic syntax, queries, and selecting columns from datasets. This will build familiarity with the program itself and develop a confidence in import and simple manipulation of data.

#### Intermediate

Next, we will incorporate real-world data sets (ex. Kaggle) to begin creating analysis. As an example, you could utilize sales data, new customers, inventory levels, certain trends over time, etc. Utilizng applications such as LinkedIn Learning or Coursera can assist with this.

#### Advanced

At this stage, we will aim for constructing pipelines that are sufficient from beginning to end and that integrate a prouction quality result. As a final step, DataLemur provides candidates with interview questions that correspond to SQL and have been confirmed by various companies such as Amazon, Google, etc.

### OneStream

It is important to note that this is a private software application so receiving quality training will be difficult without being sponsored by a company.

#### Beginner

Youtube is the best resource for beginning to familiarize yourself with the main functionality and goals of the software. The company does have their own channel, so it would be advised to watch their videos and learn more about what the application does and how it works.

#### Intermediate

Consider purchasing an online course through Udemy or Keyteach. While this does require personal spending, it would be the easiest way to gain an understanding without requiring an official license to operate the software. This would help to practice working with the application and exploring key concepts.

#### Advanced

This stage would be difficult, because you would need access to the software in order to achieve an advanced level. If you are employed (especially in the financial services industry) consider asking IT for access. There you can work more on complicated structures such as macros and visualization.

### AWS

#### Beginner

AWS offers free tier accounts where users can begin learning the basic structure of the applications that are offered. Additionally, there are various free courses that are offered by AWS Academy.

#### Intermediate

Incorporate real world data within AWS to create a storage structure, run queries, and visualize data. Additionally, you can reference Youtube or LinkedIn learning to learn more about EC2 capabilities and Quicksight (for visualization).

#### Advanced

Consider learning more about best practices and cost structuring, which will be crucial components of AWS design in a real world company. The goal of this is to begin optimizing your pipelines to make sure that they are production quality. To achieve this, consider completing the AWS Certified Data Analytics course.

### Power BI

#### Beginner

At the beginner level it is important to understand the basic UI interface of the application, such as importing data and generating visuals (bar charts, line charts, cards, etc.). Microsoft Learn can help to achieve this with their beginner course.

#### Intermediate

At this point, you should have the ability to use more complicated processes such as data cleaning, DAX functions, and establishing relationships between multiple datasets. Again, Microsoft Learn has courses called “Design Power BI reports” and “Configure Power BI report filters” that will help achieve this competency.

#### Advanced

By now, you should be able to connect your dashboard to other applications like SQL or APIs. Your dashboards should have multiple pages, which include dynamic formatting and the ability to automatically refresh.

### Office 365

This will be broken down into 3 of the most common applications with an emphasis on Excel. Microsoft 365 is the recommended training tool, as it has learning courses available for all levels.

#### Beginner

Excel: Standard formulas are used such as SUMIF and VLOOKUP. There is familarity with pivot charts and conditional formatting. Outlook: User has the ability to schedule meetings and establish designated email folders. PowerPoint: There is an understanding of presentation structure, as well as formatting and placement of text and visuals.

#### Intermediate

Excel: Power Query, Data Validation tabs, Index formulas, KPI dashboards with slicers. Outlook: Creating shared calendars and group inboxes. PowerPoint: Linking visuals from other applications, such as Excel.

#### Advanced

Excel: VBA and macros combined with dynamic visuals. PowerPoint: Creating custom templates to align with a company’s brand.

Hello

About this site

## References

Half, R. (2024): “Remote Work Statistics and Trends,”Robert Half<https://www.roberthalf.com/us/en/insights/research/remote-work-statistics-and-trends>.

Ito, A. (2025): “New Data Confirms It: AI Is Taking Human Jobs,”Business Insider<https://www.businessinsider.com/ai-hiring-white-collar-recession-jobs-tech-new-data-2025-6>.

Lee, J. J., A. Samanta, and M. Lee. (2024): “[Redefining careers in the age of artificial intelligence](https://doi.org/10.48009/3_iis_2024_132),” *Issues in Information Systems*, 25, 428–38.

McElheran, K., J. F. Li, E. Brynjolfsson, Z. Kroff, E. Dinlersoz, L. Foster, and N. Zolas. (2024): “[AI adoption in america: Who, what, and where](https://doi.org/10.1111/jems.12576),” *Journal of Economics & Management Strategy*, 33, 375–415.

Panel, E. (2021): “14 Tech Experts Predict Which Industries and Sectors Will Benefit Most from AI,”Forbes<https://www.forbes.com/councils/forbestechcouncil/2021/01/20/14-tech-experts-predict-which-industries-and-sectors-will-benefit-most-from-ai/>.

Stryker, C. (2024): “What Is Artificial Intelligence (AI)?”IBM<https://www.ibm.com/think/topics/artificial-intelligence>.