Individual Career Strategy based on Research and Quantitative analysis

Final Report

Hiten Ladkani, Xinyi Dong (Sheryl), Yao Ma

# Welcome everyone!

This is a structure analysis of how three of the most competitive and rapidly evolving fields in data and tech are demanding candidates to constantly reskill and upskill. The purpose of the analysis is to identify patterns and gaps in the labour market and provide a summarized recommendation for ourselves and anyone interested in these domains so that we can land our dream jobs either in the United states or one’s home country.

Please read through both the qualitative and quantitative analysis for detailed walkthrough of the approach and results.

***Good Luck***



Welcome Banner

title: “Qualitative Research on Business Analytics, Data Science and Machine Learning Trends” format: html: csl: apa.csl bibliography: references.bib —

# Selection Rationale:

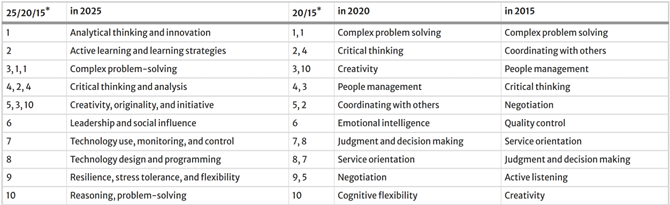
We are at the early stages of our careers in the data analytics field, and as we look to pursue opportunities in the current, rapidly evolving U.S. market—specifically for data or business analysts—our priority is to focus on the critical requirements of data roles that help us fill any existing gaps in our skill set. Hence, we will conduct a qualitative deep dive into trends and recent changes in the analytics, data science, and machine learning domain before proceeding to verify whether this qualitative analysis is supported by the quantitative assessment of the Lightcast job posting data we have at hand.

Also, it is important to note that most data analytics principles can be applied to business analytics, as the two roles have a high degree of overlap. The former involves cleaning and processing data, while the latter (business analytics) takes those insights further and translates them into business decisions or improvements in policies, processes, or operations. In essence, data analytics focuses on the “what” and “why” from data, while business analytics goes further into developing the “how” and “what next” by leveraging those insights. [@Liu2024embracingAI]

# Introduction:

This is a brief overview of how the convergence of Data analytics, Big Data and Machine Learning has become an influential force in every business that relies on data-driven decision-making. It also covers how Industry 4.0 is revolutionizing the manufacturing as well as service processes all over the world. In simple words, Industry 4.0 is an amalgamation of multiple disciplines like Internet of Things (IoT), Industrial Internet of Things (IIoT), artificial intelligence, machine learning, hyper convergent Infrastructure, deep learning and virtualization, all these create a strong and intelligent production system for businesses that enable them to carve insights from heaps of data that they are collecting but at the same time translate them into actionable items that are strongly backed. [@Li2022reskilling]. The review then shift focus on the importance of reskilling and/or upskilling is critical to be part of the talent that realizes Industry 4.0 to realize its full potential.

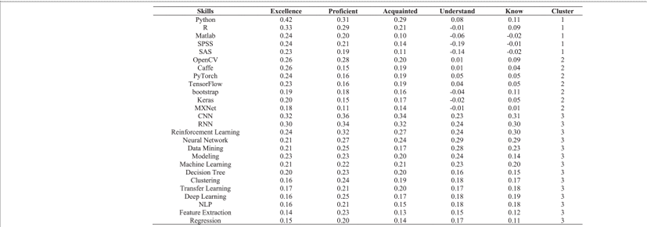
The World Economic Forum projected that half of all employees worldwide would need to reskilling by 2025 in its future jobs report 2020. [@wef2020futurejobs]. Life-long learning is not a choice but a norm in 2025 with the advent of AI, advanced applications of machine learning and increase uncertainty around ability of AI to replace candidates with jobs that have tasks that have potential to be automated.

The top 10 skills comparison from 2015, 2020 to 2025: 

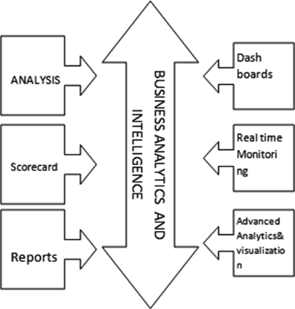
In context of Data science and machine learning, these are most sought soft skills in 2025, however industry reports reveal that there is shortage of big data client in the field and companies prefer candidates that possess one to three years of experience because they can quickly adapt to changing industry needs and help companies realize their full potential backed by big data processing and analysis. [@Yugal2022BusinessAnalytics] The literature review delves into details of each domain, explaining how they have evolved and what is required for success in this fast-evolving tech space.

# Machine Learning Trends:

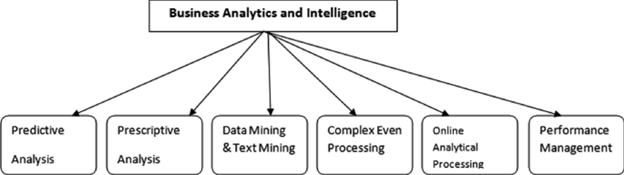
As the digital era continues to push the envelope of transformation, the trifecta or the combination of Data Analytics, Big Data and Machine learning becomes a vital source for businesses to search practical insights and long-term advancement. This intersection has necessitated the need for new skill sets to effectively analyze and interpret complex, high-volume data. To understand the required skills in this domain, it is important to understand the full range of applications of ML. It ranges from decision-making processes, modeling layout form, identification of results from data analysis, and forecast for big data analytics-oriented programs.

In terms of industry specific applications, in life sciences ML is progressively used as a analytical and computational tool for study of vast unstructured data of genome libraries to set up automated workflows. Virtual assistants have been powered by ML-enabled speech recognition technology. In combination with ML, Deep learning has become a potent instrument in variety of sectors like audio and picture identification, natural language processing and even medicine [@Badhan2024ConvergenceofBigData]. For these a sustainable talent training system is required, and it becomes the mission and responsibility of educational institutions as well as individual candidates to realize the full-scale potential of this domain. For instance, deep learning libraries like PyTorch, TensorFlow, Keras and MXNet have recently been developed and if these are introduced as conceptual content to students in advance, it will place a foundation of big data talents at the forefront and address the shortage of experienced and skilled talents that the business are looking for. [@Han2024BigDataprofessionals] A cluster analysis of Big Data expertise needs provided scores for various variables for each tool, algorithms and techniques. 

# Business Analytics Trends:

With the advent of AI and digital governance evolving rapidly, the Business analytics field has expanded beyond just managerial outlook and expertise in the business process with some level of analysis. The emergence of Big Data has led to the involvement of Business analysts in statistical analysis as well as processing of vast amounts of data to generate insights. BA & intelligence teams can take those insights further to translate them into business process improvements, development of new practices by generating dashboards or written reports for stakeholders. [@Yugal2022BusinessAnalytics] 

BA & I involves a lot more techniques post 2023 that are implemented to structure a process of business decision-making. AI/ML integration, Cloud based analytics, and predictive analytics are at the forefront of this domain as volume of data is expanding rapidly and so as the cloud infrastructure adoption to execute analytics and forecast trends, customer behavior and understand market dynamics. [@asmibmr\_evolution]



BA techniques

# Data Science Trends

To establish an understanding of this domain in simpler terms, one can say it is a step or few advanced to data analytics. It has high overlap with ML engineers and MLOps engineers and that makes sense in the real world as more and more companies expect their employees to have an understanding of a bit of everything because of the need of strong collaboration between these teams for each project.

About 20,800 openings for data scientists are projected each year since 2024, on average over the decade. The growth of this space is expected to be 36% from 2023 to 2033, which is much faster than average of all the other occupations. [@bls2024datascientists]

Data science is more of an umbrella term that has been derived from industrial demand. The T-shaped configuration of skills is generally used to analyze the breadth of knowledge and skills required for a domain and to identify a functional area that one would like to gain expertise within that domain. The T-shape or π-shaped diagram for analytics is shown below: [@shirani2016competencies] 

Exploring the skillsets that have been dominant in data science roles across the board in recent years are below: ##***Deep Learning:*** It is at the forefront of many in-demand frameworks right now, such as generative adversarial networks (GANs) and reinforcement learning which dive into artificial intelligence but as described in this entire review, each of these fields touch AI discipline in some form with data science and ML being closest to it. Sophisticated models like image classification, NLPs and anomaly detection use deep learning techniques and candidates who are looking to enter this space must have strong foundation of mathematical concepts (linear algebra, calculus, probability and statistics) and then should have understanding of core principles of deep learning – neural networks and backpropagation. [@Nobrega2023DataScienceSkills]

##***Machine Learning Deployment:*** It is understood that experimenting various ML models on the data to capture patterns is a pre-requisite now, deploying the validated models in a production environment is crucial to derive business value. For this, candidates are required to have a strong grasp of fundamentals of ML, they should be familiar with one of the cloud platforms and should have understanding of deployment tools like Docker or Kubernetes. In addition to this, understanding of MLOps best practices and lifecycle management of ML models is also expected. Hence, it is clear that the role involves lifelong learning and hands on practice on the tools and to have all basics covered so that foundation for creativity can be laid when it comes to execution. [@Nobrega2023DataScienceSkills]

# Conclusion:

It is evident that in the past 2 years, each role deals with vast amount of data and hence understanding of cloud infrastructure and big data techniques is essential for success in these roles. Although, the expertise or level of skills in these disciplines may vary for each role but it is essential to have curiosity and willingness to upskill and reskill as per the evolving needs in the analytics domain and overall business requirements to stay relevant in the market.

# References

# Introduction to Quantitative Analysis

Now that we had a deep dive in qualtitative analysis of Machine Learning, Data Analyst and Data Scientist positions, we want to leverage the Lightcast job postings data from 2024 to verify whether the findings from the research align with the current data from the market. One of the consistent themes from the research was an overlap between Machine Learning and Data Science positions because of skillsets required for these experienced roles. We will also explore if Business analyst/Data analyst positions have few most in-demand skillsets and what are they as these are foundational roles in data.

# Methodology

The first step would be to clean the dataset and identify the most relevant columns. Subset the data for Machine Learning, BA/DA and Data scientist positions data so these can be processed in silos Peform EDA for each role and highlight patterns or trends Perform Random Forest algorithms to predict salaries for these positions Summarise the findings

# Data Cleaning and Preprocessing

import pandas as pd  
import numpy as np  
import plotly.express as px  
import seaborn as sns  
import matplotlib.pyplot as plt  
import polars as pl  
from pyspark.sql import SparkSession  
import plotly.io as pio  
np.random.seed(42)  
  
pio.renderers.default = "notebook+notebook\_connected+vscode"  
# 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")  
  
# Show schema  
#df.show(5, truncate = False)

# Cleaning the dataset to remove redundant variables  
cols\_to\_drop = ["ID", "LAST\_UPDATED\_DATE", "LAST\_UPDATED\_TIMESTAMP", "DUPLICATES", "SOURCE\_TYPES","SOURCES", "BODY", "COMPANY", "COMPANY\_RAW", "TITLE\_RAW",  
 "URL", "ACTIVE\_URLs", "POSTED", "EXPIRED", "DURATION", "ACTIVE\_SOURCES\_INFO", "MODELED\_EXPIRED", "MODELED\_DURATION", "EDUCATION\_LEVELS",   
 "MIN\_EDULEVELS", "MIN\_EDULEVELS\_NAME", "MAX\_EDULEVELS", "COMPANY\_IS\_STAFFING", "EMPLOYMENT\_TYPE", "IS\_INTERNSHIP", "ORIGINAL\_PAY\_PERIOD", "REMOTE\_TYPE", "CITY", "COUNTY", "STATE", "COUNTY\_OUTGOING",  
 "COUNTY\_NAME\_OUTGOING", "COUNTY\_INCOMING", "COUNTY\_NAME\_INCOMING", "MSA\_OUTGOING", "MSA\_NAME\_OUTGOING", "MSA\_INCOMING", "MSA\_NAME\_INCOMING", "NAICS2",  
 "NAICS2\_NAME", "NAICS3", "NAICS3\_NAME", "NAICS4", "NAICS4\_NAME", "NAICS5", "NAICS5\_NAME", "NAICS6", "NAICS6\_NAME", "TITLE", "TITLE\_NAME", "SKILLS",  
 "SPECIALIZED\_SKILLS", "CERTIFICATIONS", "CERTIFICATIONS\_NAME", "COMMON\_SKILLS", "SOFTWARE\_SKILLS", "ONET", "ONET\_2019", "ONET\_2019\_NAME", "CIP6", "CIP6\_NAME", "CIP4",  
 "CIP4\_NAME", "CIP2", "CIP2\_NAME", "SOC\_2021\_2", "SOC\_2021\_3", "SOC\_2021\_4", "SOC\_2021\_5", "LOT\_CAREER\_AREA", "LOT\_OCCUPATION", "LOT\_SPECIALIZED\_OCCUPATION", "LOT\_OCCUPATION\_GROUP",   
 "LOT\_OCCUPATION\_GROUP\_NAME", "LOT\_V6\_SPECIALIZED\_OCCUPATION", "LOT\_V6\_OCCUPATION", "LOT\_V6\_OCCUPATION\_GROUP", "LOT\_V6\_OCCUPATION\_GROUP\_NAME", "LOT\_V6\_CAREER\_AREA", "LOT\_V6\_CAREER\_AREA\_NAME", "SOC\_2",  
 "SOC\_2\_NAME", "SOC\_3", "SOC\_3\_NAME", "SOC\_4", "SOC\_4\_NAME", "SOC\_5", "SOC\_5\_NAME", "LIGHTCAST\_SECTORS", "LIGHTCAST\_SECTORS\_NAME", "NAICS\_2022\_2", "NAICS\_2022\_3", "NAICS\_2022\_4", "NAICS\_2022\_5", "NAICS\_2022\_6",  
 "MSA", "MSA\_NAME"]  
df\_cleaned = df.drop(\*cols\_to\_drop)  
#df\_cleaned.show(5, truncate = False)

# Listing data types of the cleaned DataFrame  
#df\_cleaned.dtypes

# Converting spark dataframe to SQL table to have subsets for Machine Learning positions.  
df\_cleaned.createOrReplaceTempView("job\_postings\_data")  
  
ml\_data\_df = spark.sql("""  
 SELECT \*  
 FROM job\_postings\_data  
 WHERE SALARY IS NOT NULL  
 AND SALARY > 0  
 AND TITLE\_CLEAN LIKE '%machine learning%';  
 """)  
ml\_data\_pd = ml\_data\_df.toPandas()  
#ml\_data\_pd.info()

25/06/29 23:31:35 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 2:> (0 + 1) / 1]

ml\_data\_df.select("TITLE\_CLEAN").distinct().show(10, truncate=False)

[Stage 3:> (0 + 1) / 1]

+---------------------------------------------------------------------------------------------------------------------------------+  
|TITLE\_CLEAN |  
+---------------------------------------------------------------------------------------------------------------------------------+  
|enterprise artificial intelligence machine learning ai ml platform product manager senior lead data product management consultant|  
+---------------------------------------------------------------------------------------------------------------------------------+

**ML specific positions are limited in title clean** - Since we cannot seem to distinguish the data between Data Science and Machine Learning, from the dataset using job titles, we will focus on Data Scientist and Business Analyst positions. Qualitative data also suggested a strong overlap between Data Science and Machine Learning positions because the current job market expects candidates for Data scientist positions to have skills in not just ML models but production deployment and Artificial intelligence as well. [@Nobrega2023DataScienceSkills]

# Subsetting the main dataframe for Business Analyst, Data Analyst and Business Intelligence positions.  
  
ba\_da\_data\_df = spark.sql("""  
 SELECT \*  
 FROM job\_postings\_data  
 WHERE SALARY IS NOT NULL  
 AND SALARY > 0  
 AND LOT\_SPECIALIZED\_OCCUPATION\_NAME IN ('Business Intelligence Analyst',  
 'Business Analyst (General)', 'Data Analyst');  
""")  
  
ba\_da\_data\_pd = ba\_da\_data\_df.toPandas()

# Subsetting the main dataframe for Data Science positions.  
ds\_data\_df = spark.sql("""  
 SELECT \*  
 FROM job\_postings\_data  
 WHERE SALARY IS NOT NULL  
 AND SALARY > 0  
 AND SOC\_2021\_5\_NAME = 'Data Scientists';  
""")  
ds\_data\_pd = ds\_data\_df.toPandas()

# Condensing both dataframes to remove additional unncessary columns.  
ba\_da\_cols\_to\_drop = ['ONET\_NAME', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4\_NAME',  
 'SOC\_2021\_5\_NAME', 'LOT\_CAREER\_AREA\_NAME', 'LOT\_OCCUPATION\_NAME','LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME',  
 'LOT\_V6\_OCCUPATION\_NAME', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6\_NAME', 'MAX\_YEARS\_EXPERIENCE', 'MAX\_EDULEVELS\_NAME']  
ba\_da\_data\_pd = ba\_da\_data\_pd.drop(columns=ba\_da\_cols\_to\_drop, axis=1)   
#ba\_da\_data\_pd.info()

# Removing job postings with no titles and min years of experience  
ba\_da\_data\_pd = ba\_da\_data\_pd.dropna(subset = ['TITLE\_CLEAN'])  
ba\_da\_data\_pd = ba\_da\_data\_pd.dropna(subset = ['MIN\_YEARS\_EXPERIENCE'])  
#ba\_da\_data\_pd.info()

ds\_data\_cols\_to\_drop = ['ONET\_NAME', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4\_NAME','LOT\_CAREER\_AREA\_NAME',   
 'LOT\_OCCUPATION\_NAME','LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME',  
 'LOT\_V6\_OCCUPATION\_NAME', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_3\_NAME','NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6\_NAME',  
 'MAX\_YEARS\_EXPERIENCE', 'MAX\_EDULEVELS\_NAME']  
ds\_data\_pd = ds\_data\_pd.drop(columns=ds\_data\_cols\_to\_drop, axis=1)   
#ds\_data\_pd.info()

# Removing job postings with no titles and min years of experience  
ds\_data\_pd = ds\_data\_pd.dropna(subset = ['TITLE\_CLEAN'])  
ds\_data\_pd = ds\_data\_pd.dropna(subset = ['MIN\_YEARS\_EXPERIENCE'])  
#ds\_data\_pd.info()

# Sorting categories under remote type name for both dataframes  
ba\_da\_data\_pd['REMOTE\_TYPE\_NAME'].unique()  
ba\_da\_data\_pd['REMOTE\_TYPE'] = ba\_da\_data\_pd['REMOTE\_TYPE\_NAME'].replace({  
 '[None]': 'Onsite',  
 'Not Remote': 'Onsite',  
 'Hybrid Remote': 'Hybrid',  
 'Remote': 'Remote'  
}).astype('category')  
  
ds\_data\_pd['REMOTE\_TYPE\_NAME'] = ds\_data\_pd['REMOTE\_TYPE\_NAME'].replace({  
 '[None]': 'Onsite',  
 'Not Remote': 'Onsite',  
 'Hybrid Remote': 'Hybrid',  
 'Remote': 'Remote'  
}).astype('category')  
  
# Sorting categories under employment type name for both dataframes  
ba\_da\_data\_pd['EMPLOYMENT\_TYPE\_NAME'].astype('category')  
ds\_data\_pd['EMPLOYMENT\_TYPE\_NAME'].astype('category')

#ba\_da\_data\_pd['EDUCATION\_LEVELS\_NAME'].unique()

# Converting education levels into 2 categories Bachelor's or lower and Master's or PhD  
import ast # Using ast.literal\_eval to first convert stringified lists into python lists   
import re # Need to remove those extra spaces in the Masters degree for proper classification  
  
def clean\_text(text):  
 text = text.replace("\n", " ")   
 text = re.sub(r"0\s+", " ", text) # Replacing those multiple spaces with single space for Masters degree  
 return text.strip()  
  
  
def classify\_education\_level(edu\_levels):  
 try:  
 levels = ast.literal\_eval(edu\_levels)  
  
 levels = [clean\_text(level) for level in levels]   
  
 # defining categories now   
 Bachelors\_or\_lower = {"Bachelor's degree", "Associate degree", "No Education Listed", "High school or GED"}  
 Masters\_or\_PhD = {"Master's degree", "Ph.D. or professional degree"}  
  
 # Now applying IF logic  
 if any(level in Bachelors\_or\_lower for level in levels):  
 return "Bachelors\_or\_lower"  
 elif any(level in Masters\_or\_PhD for level in levels):  
 return "Masters\_or\_PhD"  
 else:  
 return "Other"  
 except:  
 return "Unknown"  
  
 # Now we apply it to both dataframes   
ba\_da\_data\_pd['EDUCATION\_LEVELS'] = ba\_da\_data\_pd['EDUCATION\_LEVELS\_NAME'].apply(classify\_education\_level).astype('category')  
ds\_data\_pd['EDUCATION\_LEVELS'] = ds\_data\_pd['EDUCATION\_LEVELS\_NAME'].apply(classify\_education\_level).astype('category')

#ba\_da\_data\_pd['EDUCATION\_LEVELS'].unique()  
#ds\_data\_pd['EDUCATION\_LEVELS'].unique()

# Separating Longitude and Latitude from the location column for maps visualization  
def extract\_lat\_long(location):  
 try:  
 loc\_dict = ast.literal\_eval(location.replace("\n", " ").strip())  
 return pd.Series([loc\_dict.get("lat"), loc\_dict.get("lon")])  
 except:  
 return pd.Series([None, None])  
   
ba\_da\_data\_pd[['LATITUDE', 'LONGITUDE']] = ba\_da\_data\_pd['LOCATION'].apply(extract\_lat\_long)  
ds\_data\_pd[['LATITUDE', 'LONGITUDE']] = ds\_data\_pd['LOCATION'].apply(extract\_lat\_long)

#ba\_da\_data\_pd[["LONGITUDE", "LATITUDE"]].head(5)

ba\_da\_data\_pd.drop(columns=['REMOTE\_TYPE\_NAME', 'EDUCATION\_LEVELS\_NAME',   
 'LOCATION'], axis = 1, inplace = True)  
ds\_data\_pd.drop(columns=['REMOTE\_TYPE\_NAME', 'EDUCATION\_LEVELS\_NAME',  
 'LOCATION'], axis = 1, inplace = True)

**Summarizing the cleaning of data for both positions:** The most important data cleaning steps performed here were slicing the data for each position to process them individually. Location column was separated into Latitude and Logitude to enable geographic analysis of job postings. Remote type, education levels and employment type columns were converted into categories to visualized summarized data. Skillsets columns will be cleaned at a later stage when we analyze top skills for each position and then progress into Machine Learning.

# Geographical analysis of Business Analytics, Data Science and Machine Learning

# Since the data is ready we can now create comparison bar plots   
# Top 10 states with highest number of job postings for Business Analyst, Data Analyst and Data Scientist positions  
import hvplot.pandas  
  
ba\_da\_state\_counts = (ba\_da\_data\_pd.groupby(['STATE\_NAME', 'EDUCATION\_LEVELS']).size().reset\_index(name='COUNT'))  
ds\_state\_counts = (ds\_data\_pd.groupby(['STATE\_NAME', 'EDUCATION\_LEVELS']).size().reset\_index(name='COUNT'))  
  
top\_states\_da = (ba\_da\_state\_counts.groupby('STATE\_NAME')['COUNT'].sum().nlargest(10).index)  
ba\_da\_top10 = (ba\_da\_state\_counts[ba\_da\_state\_counts['STATE\_NAME'].isin(top\_states\_da)]).sort\_values(by='COUNT', ascending=False)   
  
top\_states\_ds = (ds\_state\_counts.groupby('STATE\_NAME')['COUNT'].sum().nlargest(10).index)  
ds\_top10 = (ds\_state\_counts[ds\_state\_counts['STATE\_NAME'].isin(top\_states\_ds)]).sort\_values(by='COUNT', ascending=False)  
  
# Plotting stacked bar plots   
ba\_da\_plot = ba\_da\_top10.hvplot.bar(x='STATE\_NAME', y='COUNT', by='EDUCATION\_LEVELS', stacked=True,   
 title='Top 10 States for BA and DA Positions',rot = 45,  
 xlabel='States', ylabel='Number of Job Postings', width=500, height=400)   
ds\_plot = ds\_top10.hvplot.bar(x='STATE\_NAME', y='COUNT', by='EDUCATION\_LEVELS', stacked=True,  
 title='Top 10 States for Data Scientist Positions',rot = 45,  
 xlabel='States', ylabel='Number of Job Postings', width=500, height=400)  
combined\_plot = (ba\_da\_plot + ds\_plot).opts(shared\_axes=False)  
combined\_plot  
  
# Saving the plot for HTML rendering  
from holoviews import save  
hvplot.save(combined\_plot, "./visualizations/top\_states\_job\_postings.html")

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

Unable to display output for mime type(s): application/javascript, application/vnd.holoviews\_load.v0+json

Unable to display output for mime type(s): application/javascript, application/vnd.holoviews\_load.v0+json

Unable to display output for mime type(s): application/vnd.holoviews\_exec.v0+json, text/html

#combined\_plot

**Analysis** - Our first step in laying our career path is to narrow down the geographical locations where the positions are frequently available, so we have better chance of landing a job. Higher the number of positions available in a location, higher the chances of landing a job even though the competition is high, the odds of receiving an offer remain optimistic. The above plot is an hvplot providing a side by side comparison of top 10 states in the US for Business Analyst/Data Analyst and Data Scientist positions. The data is aggregated by the number of positions available in each state. The plot shows that California, Texas, New York, and Florida are the top states for both positions, with California having the highest number of positions available. This is expected as these states have high concentrations of technology companies and startups, which are the primary employers of data professionals. To add a layer of granularity, we can see that very few positions focus on education levels for both positions. However, Data scientist positions generally demand for a Masters or PhD degree alongwith a strong experience in analytics and expertise in ML models. Surprisingly, Massachusetts does not fall in the top 10 states which is our preferred state but with current labor market these initial signs suggest that relocation should be considered to keep enough options open.

# Geospatial analysis of top 10 industries that have BA/DA positions  
ba\_da\_ind\_counts = (ba\_da\_data\_pd.groupby("NAICS\_2022\_2\_NAME").size().reset\_index(name = "COUNT").sort\_values(by = "COUNT",   
 ascending = False).head(10))  
  
ba\_da\_filtered = ba\_da\_data\_pd[ba\_da\_data\_pd["NAICS\_2022\_2\_NAME"].isin(ba\_da\_ind\_counts["NAICS\_2022\_2\_NAME"])]  
ba\_da\_filtered = ba\_da\_filtered[(ba\_da\_filtered["LATITUDE"] != 0) & (ba\_da\_filtered["LONGITUDE"] != 0)]  
  
  
ba\_da\_map = ba\_da\_filtered.hvplot.points("LONGITUDE", y = "LATITUDE", geo = True,   
 tiles = True, c = "NAICS\_2022\_2\_NAME", colorbar = False,  
 hover\_cols = ["NAICS\_2022\_2\_NAME", "CITY\_NAME", "STATE\_NAME", "SALARY"],  
 title = 'Top 10 Industries hiring for BA/DA positions',  
 width = 800, height = 500, xlim = (-130, -65), ylim = (24,50)).opts(  
 legend\_position = "bottom\_left", legend\_opts = {'label\_text\_font\_size': '5pt',  
 'title\_text\_font\_size': '4pt'})  
hvplot.save(ba\_da\_map, "./visualizations/ba\_da\_indgeomap.html")   
#ba\_da\_map

**Analysis** - We first identified our preferred industry and those were Marketing (Unclassified industry), Healthcare and Finance. By plotting the job postings for both BA/DA and Data Scientist positions, we looked at which states have high postings for these industries. The above plot is specifically for BA/DA jobs, we can see that along the east cost these industries are more prominent, specially in NYC, Boston, Philadelphia and Washington. When we move to the west, the tight cluster as expected is around LA and San Jose in California. Finance industry is more prominent on the east likely because of the presence of Wall street in NYC.

# Geospatial analysis of top 10 industries that have Data scientist positions  
ds\_ind\_counts = (ds\_data\_pd.groupby("NAICS\_2022\_2\_NAME").size().reset\_index(name = "COUNT").sort\_values(by = "COUNT",   
 ascending = False).head(10))  
  
ds\_filtered = ds\_data\_pd[ds\_data\_pd["NAICS\_2022\_2\_NAME"].isin(ds\_ind\_counts["NAICS\_2022\_2\_NAME"])]  
ds\_filtered = ds\_filtered[(ds\_filtered["LATITUDE"] != 0) & (ds\_filtered["LONGITUDE"] != 0)]  
  
  
ds\_map = ds\_filtered.hvplot.points("LONGITUDE", y = "LATITUDE", geo = True,   
 tiles = True, c = "NAICS\_2022\_2\_NAME", colorbar = False,  
 hover\_cols = ["NAICS\_2022\_2\_NAME", "CITY\_NAME", "STATE\_NAME", "SALARY"],  
 title = 'Top 10 Industries hiring for BA/DA positions',  
 width = 800, height = 500, xlim = (-130, -65), ylim = (24,50)).opts(  
 legend\_position = "bottom\_left", legend\_opts = {'label\_text\_font\_size': '5pt',  
 'title\_text\_font\_size': '4pt'})  
hvplot.save(ds\_map, "./visualizations/ds\_indgeomap.html")   
   
#ds\_map

**Analysis**- This plot focuses on the Data Science positions and it is evident that the number of jobs are considerably higher for this position regardless of the industry. The trend is similar to BA/DA positions in terms of clustering around east cost states and on the west coast in Sacramento, CA and San Jose. However, surprisingly for us Seattle has dense cluster for Healtcare and North Carolina and Chicago seem to have a big cluster for Marketing (Unclassified industry)

# Salary Comparison & Trends across both positions

# Combining both dataframes to create overlayed histograms   
ba\_da\_data\_pd["ROLE"] = "BA/DA"  
ds\_data\_pd["ROLE"] = "Data Scientist"  
combined\_df = pd.concat([ba\_da\_data\_pd, ds\_data\_pd])  
  
combined\_hist = px.histogram(combined\_df, x = "SALARY", color = "ROLE",   
 nbins = 50, barmode = "overlay", histnorm = "probability density",  
 title = "Salary distributions for BA/DA and Data Scientists",  
 marginal = "violin")  
combined\_hist.update\_traces(opacity = 0.75)  
  
import kaleido  
combined\_hist.write\_html("./visualizations/salary\_distribution.html")  
combined\_hist.write\_image("./visualizations/salary\_distribution.png")  
  
#combined\_hist.show()

**Salary distribution analysis:** We have now moved to the monetary value offered by both positions and we already know that data science being the senior position commands higher salary but the purpose of this plot is to identify what is the sweet spot of the salary range where both positions lie in 2024. The peak for BA/DA positoions is between $80K-$100K. We can also see from the overlay plot that the violin plot starts to shrink immediately after $100K and the right tail is light compared to Data science positions. For Data Scientist the median salary is $120K and the upper fence is almost $240K with much heavier right tails which is likely influenced by the information sector where salaries for this position can be as good as senior software developer position.

# Salary distribution and employment type (Faceted)  
top10\_ba\_da\_inds = ba\_da\_data\_pd['NAICS\_2022\_2\_NAME'].value\_counts().nlargest(10).index  
top10\_ds\_inds = ds\_data\_pd['NAICS\_2022\_2\_NAME'].value\_counts().nlargest(10).index  
  
# Filtering datasets to include only top 10 industries to keep it consistent with other plots  
ba\_da\_filt1 = ba\_da\_data\_pd[ba\_da\_data\_pd['NAICS\_2022\_2\_NAME'].isin(top10\_ba\_da\_inds)]  
ds\_filt1 = ds\_data\_pd[ds\_data\_pd['NAICS\_2022\_2\_NAME'].isin(top10\_ds\_inds)]  
  
import plotly.graph\_objects as go  
  
# Create BA/DA box plot with industry in legend and employment type as facets  
ba\_da\_emp\_ind = px.box(  
 ba\_da\_filt1,   
 y='SALARY',   
 x='EMPLOYMENT\_TYPE\_NAME',   
 color='NAICS\_2022\_2\_NAME',   
 height=700,  
 width=1000)  
  
# Data Scientist plot  
ds\_emp\_ind = px.box(  
 ds\_filt1,   
 y='SALARY',  
 x='EMPLOYMENT\_TYPE\_NAME',  
 color='NAICS\_2022\_2\_NAME',  
 height=700,  
 width=1000)  
import plotly.graph\_objects as go  
dist\_fig = go.Figure()  
  
for trace in ba\_da\_emp\_ind.data:  
 trace.visible = True  
 dist\_fig.add\_trace(trace)  
for trace in ds\_emp\_ind.data:  
 trace.visible = False   
 dist\_fig.add\_trace(trace)  
  
n\_ba\_da = len(ba\_da\_emp\_ind.data)  
n\_ds = len(ds\_emp\_ind.data)  
  
dropdown\_buttons = [  
 dict(label="BA/DA",  
 method="update",  
 args=[{"visible": [i < n\_ba\_da for i in range(n\_ba\_da + n\_ds)]},  
 {"title": "BA/DA Salary Distribution by Employment Type and Industry (Top 10)"}]),  
  
 dict(label="Data Scientist",  
 method="update",  
 args=[{"visible": [i >= n\_ba\_da for i in range(n\_ba\_da + n\_ds)]},  
 {"title": "Data Scientist Salary Distribution by Employment Type and Industry (Top 10)"}])  
]  
  
# Update layout with dropdown  
dist\_fig.update\_layout(  
 updatemenus=[dict(  
 buttons=dropdown\_buttons,  
 direction="down",  
 showactive=True,  
 x=0.5,  
 xanchor="center",  
 y=1.15,  
 yanchor="top"  
 )],  
 xaxis\_title="Employment Type",  
 yaxis\_title="Salary",  
 legend\_title="Industry",  
 height=700,  
 width=1000,  
 boxmode="group"  
)  
dist\_fig.write\_html("./visualizations/salary\_dist\_emp\_ind.html")  
dist\_fig.write\_image("./visualizations/salary\_dist\_emp\_ind.png")  
  
#dist\_fig.show()

**Salary distribution by Industry** - In the previous plot, we compared salaries at the highest level, in this case we are being more specific to industries we are interested in (Unclassified for Marketing, Finance and Health care). For BA/DA positions we can see that Finance sector appears to be more lucrative compared to unclassified and Healthcare. Median salary for full time BA/DA role is $112K whereas for Marketing and Healthcare it is close to $95K which seems to be okay for full time positions but the range does fall as low as $40K which could be for junior or entry level positions which is worrisome specially for us as international students as there is benchmark for H1b approval in terms of salaries. In terms of Data Science, all the 3 industries are very lucrative as the average salary is well above $120K and can go as high as $220K. Finance and Insurance in this case display more rapid growth.

# Skills Trends Analysis across Business Analyst/Data Analyst and Data Scientist positions

# Cleaning skills column to get them as list for each row and continue with analysis   
def clean\_skills\_column(df, col):  
 df[col] = df[col].apply(lambda x: ast.literal\_eval(x) if pd.notnull(x) else [])  
 df[col] = df[col].apply(lambda lst: [i.strip().replace("\n", "") for i in lst])  
 return df  
  
skill\_cols = ['COMMON\_SKILLS\_NAME', 'SOFTWARE\_SKILLS\_NAME', 'SPECIALIZED\_SKILLS\_NAME']  
  
for col in skill\_cols:  
 ba\_da\_data\_pd = clean\_skills\_column(ba\_da\_data\_pd, col)  
 ds\_data\_pd = clean\_skills\_column(ds\_data\_pd, col)

# Our thought process is to visualize first the top 10 skills for each role and based on how many times they appear   
# Respective job postings and then get granular  
def process\_skills(df, role, skill\_col):  
 temp = df[['NAICS\_2022\_2\_NAME', 'SALARY', skill\_col]].copy()  
 temp['ROLE'] = role  
 temp = temp.rename(columns={skill\_col: 'SKILLS', 'NAICS\_2022\_2\_NAME': 'INDUSTRY'})  
 temp = temp.explode('SKILLS')  
 temp = temp.dropna(subset=['SKILLS'])  
 temp['SKILL\_TYPE'] = skill\_col.replace('\_NAME', '').title().replace('\_', ' ')  
 return temp.rename(columns={'SKILLS': 'SKILL'})  
  
skill\_data = []  
for col in skill\_cols:  
 skill\_data.append(process\_skills(ba\_da\_data\_pd, 'BA/DA', col))  
 skill\_data.append(process\_skills(ds\_data\_pd, 'Data Scientist', col))  
  
all\_skills = pd.concat(skill\_data, ignore\_index=True)

ba\_da\_top\_skills = {}  
ds\_top\_skills = {}  
  
roles\_data = {  
 'BA/DA': all\_skills[all\_skills['ROLE'] == 'BA/DA'],  
 'Data Scientist': all\_skills[all\_skills['ROLE'] == 'Data Scientist']  
}  
  
for role, role\_df in roles\_data.items():  
 skill\_type\_dict = {}  
 for skill\_type in role\_df['SKILL\_TYPE'].unique():  
 temp\_df = role\_df[role\_df['SKILL\_TYPE'] == skill\_type]  
 top\_10\_series= (  
 temp\_df['SKILL']  
 .value\_counts()  
 .nlargest(10)  
 )  
 top\_10\_df = top\_10\_series.reset\_index()  
 top\_10\_df.columns = ['SKILL', 'COUNT']  
 skill\_type\_dict[skill\_type] = top\_10\_df  
   
 if role == 'BA/DA':  
 ba\_da\_top\_skills = skill\_type\_dict  
 else:  
 ds\_top\_skills = skill\_type\_dict

import plotly.graph\_objects as go  
  
def create\_bubble\_chart(role\_name, role\_data):  
 skill\_bubble = go.Figure()  
 buttons = []  
 trace\_count = 0  
   
 for i, (skill\_type, df) in enumerate(role\_data.items()):  
 visible = [False] \* len(role\_data)  
 visible[i] = True # show only the selected skill type initially  
   
 skill\_bubble.add\_trace(go.Scatter(  
 x=df['SKILL'],  
 y=df['COUNT'],  
 mode='markers+text',  
 text=df['SKILL'],  
 textposition='top center',  
 textfont= dict(size = 7),  
 marker=dict(  
 size=df['COUNT'],  
 sizemode='area',  
 sizeref=2. \* df['COUNT'].max() / (40. \*\* 2),  
 sizemin=10  
 ),  
 name=skill\_type,  
 visible=True if i == 0 else False  
 ))  
  
 buttons.append(dict(  
 label=skill\_type,  
 method='update',  
 args=[  
 {'visible': [j == i for j in range(len(role\_data))]},  
 {'title': f"{role\_name} - {skill\_type}"}  
 ]  
 ))  
  
 skill\_bubble.update\_layout(  
 title=f"{role\_name} - {list(role\_data.keys())[0]}",  
 updatemenus=[dict(  
 buttons=buttons,  
 direction="down",  
 showactive=True,  
 x=0.1,  
 y=1.15  
 )],  
 xaxis\_title="Skill",  
 yaxis\_title="Count",  
 height=600  
 )  
  
 return skill\_bubble  
  
# Create and display  
skill\_bubble\_ba\_da = create\_bubble\_chart("BA/DA", ba\_da\_top\_skills)  
skill\_bubble\_ds = create\_bubble\_chart("Data Scientist", ds\_top\_skills)  
  
skill\_bubble\_ba\_da.write\_html("./visualizations/skill\_bubble\_ba.html")  
skill\_bubble\_ba\_da.write\_image("./visualizations/skill\_bubble\_ba.png")  
skill\_bubble\_ds.write\_html("./visualizations/skill\_bubble\_ds.html")  
skill\_bubble\_ds.write\_image("./visualizations/skill\_bubble\_ds.png")  
  
#skill\_bubble\_ba\_da.show()  
#skill\_bubble\_ds.show()

**Analysis Top 10 skills** - The above plot highlights top 10 skills from each skill type (Common, Specialized and Software skills), this will give a full overview of what is required to be successful in the role and match the demands of the current labor market in these domains. In terms of common skills, we can see that common skills or so to say soft skills like communication, Management and Problem solving are in both roles but the presence is considerably higher in Data scientist. Communication as expected is at the highest because both these roles require high collaboration. Strangely for data scientist, the postings that require Python is less but that could also be because some postings have the tendency to include Python libraries instead of using the tool name because they are looking for specific candidates who should know these libraries. Computer Science seems to be one of the highest requirement for data science which can be influenced by the Information and Tech sector. On BA/DA roles, Dashboarding and Business Intelligence tools are on higher scale because that is what the role entails, there is consistent use of SQL and BI tools alongwith some use of Python.

# Average Min years of experience required across Industries for BA/DA and Data Science positions  
  
# Calculating average experience (minimum) per industry for each role  
avg\_exp\_by\_industry\_role = (  
 combined\_df.groupby(['ROLE', 'NAICS\_2022\_2\_NAME'])  
 .agg(AVG\_MIN\_EXPERIENCE=('MIN\_YEARS\_EXPERIENCE', 'mean'))  
 .reset\_index()  
 .rename(columns={'NAICS\_2022\_2\_NAME': 'INDUSTRY'})  
)  
  
ba\_df = avg\_exp\_by\_industry\_role[avg\_exp\_by\_industry\_role['ROLE'] == 'BA/DA'].sort\_values(by='AVG\_MIN\_EXPERIENCE', ascending=False)  
ds\_df = avg\_exp\_by\_industry\_role[avg\_exp\_by\_industry\_role['ROLE'] == 'Data Scientist'].sort\_values(by='AVG\_MIN\_EXPERIENCE', ascending=False)  
  
hbar\_ba = px.bar(  
 ba\_df,  
 x='AVG\_MIN\_EXPERIENCE',  
 y='INDUSTRY',  
 orientation='h',  
 title='Avg Min Years of Experience by Industry – BA/DA',  
 labels={'AVG\_MIN\_EXPERIENCE': 'Avg Min Experience', 'INDUSTRY': 'Industry'},  
 height=800  
)  
hbar\_ba.update\_layout(yaxis=dict(tickfont=dict(size=10)))  
  
hbar\_ds = px.bar(  
 ds\_df,  
 x='AVG\_MIN\_EXPERIENCE',  
 y='INDUSTRY',  
 orientation='h',  
 title='Avg Min Years of Experience by Industry – Data Scientist',  
 labels={'AVG\_MIN\_EXPERIENCE': 'Avg Min Experience', 'INDUSTRY': 'Industry'},  
 height=800  
)  
  
hbar\_ds.update\_layout(yaxis=dict(tickfont=dict(size=10)))  
hbar\_ba.write\_html("./visualizations/avg\_exp\_indba.html")  
hbar\_ba.write\_image("./visualizations/avg\_exp\_indba.png", width=1000, height=800, scale=2)  
hbar\_ds.write\_html("./visualizations/avg\_exp\_indds.html")  
hbar\_ds.write\_image("./visualizations/avg\_exp\_indds.png", width=1000, height=800, scale=2)  
  
#hbar\_ba.show()  
#hbar\_ds.show()

**Average Experience by Industry Analysis** - When it comes to skills analysis, it is not just if a candidate knows the skill but how much experience is required also is critical. As per [@Singh2024DataAnalytics], he and his co-authors had explained that the industry at this point in time is valuing experience of using the skills on real-world data than just backing of academic projects. Hence, as we step into the job market in search of an ideal job it is critical to know what are the demands of the companies in our preferred industries (Healthcare, Marketing and Finance) in terms of the number of years of experience. Although the average minimum years experience for all these 3 industry groups is above 3, we can speculate this is likely caused because of some of the executive positions failling under these industries. However, it is important to note that even after removing those outliers we should be speculating something aroun 0-2 years range and that highlights the importance of collaborating on real-world projects, internships and putting in the work beyond the academics.

# Machine Learning: Random Forest for predicting Salary

# Converting pandas dataframes back into spark   
ba\_da\_data\_df = spark.createDataFrame(ba\_da\_data\_pd)  
ds\_data\_df = spark.createDataFrame(ds\_data\_pd)  
  
# Keeping only common columns in both dataframes  
cols\_ba = set(ba\_da\_data\_df.columns)  
cols\_ds = set(ds\_data\_df.columns)  
  
common\_cols = list(cols\_ba.intersection(cols\_ds))  
  
ml\_ba\_da\_df = ba\_da\_data\_df.select(common\_cols)  
ml\_ds\_data\_df = ds\_data\_df.select(common\_cols)  
  
# Combining both dataframes for ML  
ml\_combined\_df = ml\_ba\_da\_df.unionByName(ml\_ds\_data\_df)

# Cleaning the target variable and features in the dataset: Feature Engineering

from pyspark.sql.functions import col, pow  
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler, CountVectorizer  
from pyspark.ml import Pipeline  
  
# Dropping rows with NA values in the columns that will be used in the model   
ml\_df = ml\_combined\_df.dropna(subset=["SALARY", "MIN\_YEARS\_EXPERIENCE", "SPECIALIZED\_SKILLS\_NAME",  
 "EDUCATION\_LEVELS"])  
  
# Using CountVectorizer instead of OneHotencoder to convert list of skills into a sparse vector  
cv = CountVectorizer(inputCol="SPECIALIZED\_SKILLS\_NAME", outputCol="SKILLS\_VEC", vocabSize = 1000, binary =True )  
  
# Encoding these categorical variables   
indexers = StringIndexer(inputCol = "EDUCATION\_LEVELS", outputCol = "EDU\_IDX", handleInvalid = "skip")  
  
# Using vector assembler to assemble features (for GLR and Random forest)  
assembler = VectorAssembler(inputCols = ["MIN\_YEARS\_EXPERIENCE"] + ["EDU\_IDX"] + ["SKILLS\_VEC"],  
 outputCol = "features")  
# Building the pipeline of target and features   
cv\_model = cv.fit(ml\_df)  
ml\_df = cv\_model.transform(ml\_df)  
  
indexers\_model = indexers.fit(ml\_df)  
ml\_df = indexers\_model.transform(ml\_df)  
  
data = assembler.transform(ml\_df)  
  
# Displaying the final structure  
data.select("SALARY", "features").show(5, truncate = True)

**Feature Selection** - For continuous variables I have used minimum experience required for a job posting because that will be an indicator for junior and more senior level roles as we have some executive roles as well. For categorical variables, we have selected specialized skills because those skills ideally set the salary, for eg if we know SQL plus AWS services and how to work with RDS then we might command higher salary then those that know SQL and Excel but no cloud knowledge. Also we used Vectorizer instead of one Hot encoding just to limit the number of features and keep the computation stable.

# Splitting the prepared data into Training and Test set

train\_data, test\_data = data.randomSplit([0.8, 0.2], seed = 317)  
print((train\_data.count()), len(train\_data.columns))  
print((test\_data.count()), len(test\_data.columns))

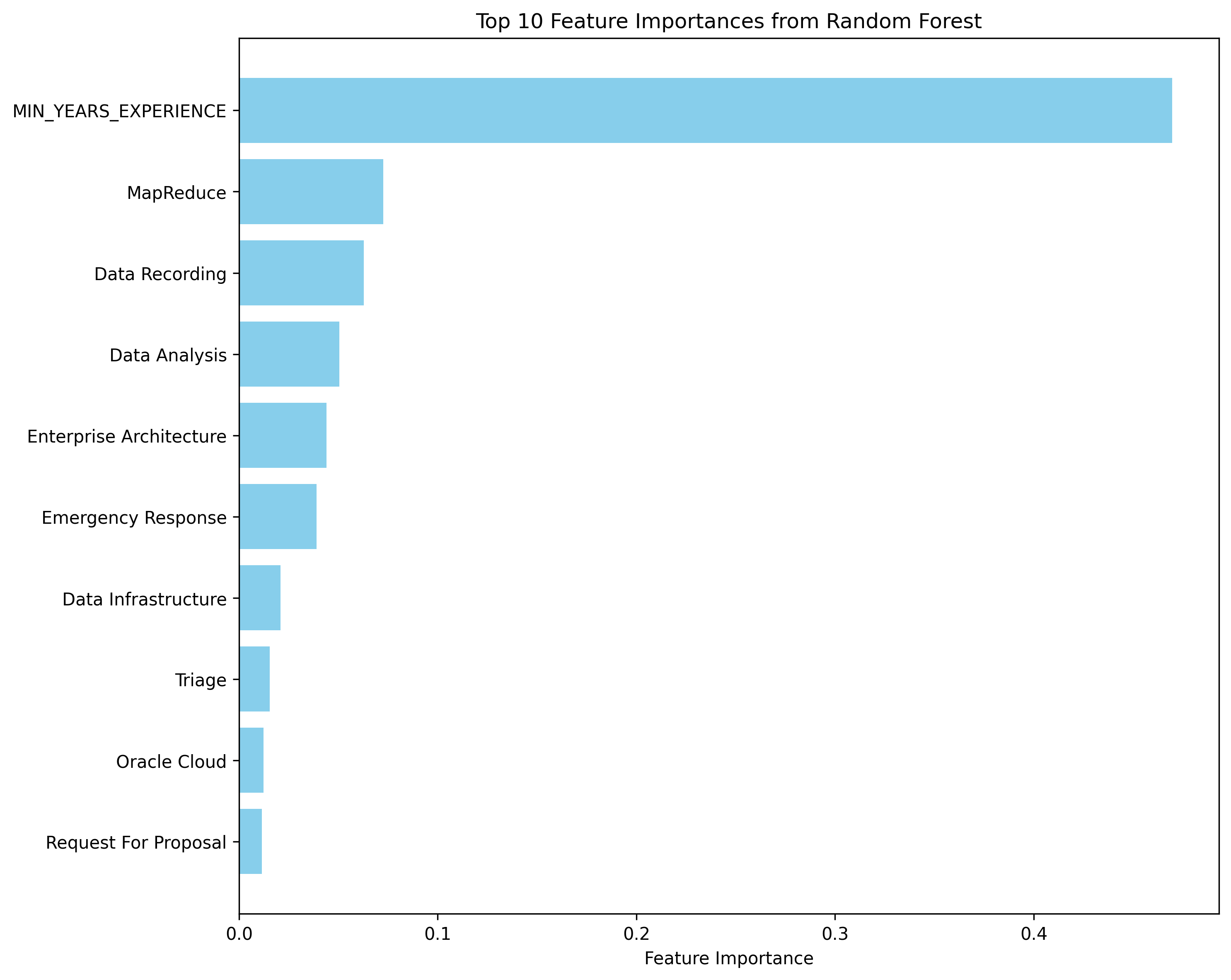
# Excecuting Random Forest Model

from pyspark.ml.regression import RandomForestRegressor  
  
# Initiating the model  
rf = RandomForestRegressor(featuresCol = "features", labelCol = "SALARY", numTrees = 150, maxDepth = 6, seed = 317)  
  
# Training the Random Forest model  
rf\_model = rf.fit(train\_data.select("SALARY", "features"))  
  
# Generating predictions  
rf\_preds = rf\_model.transform(train\_data.select("SALARY", "features"))

skill\_features = cv\_model.vocabulary # List of strings  
all\_feature\_names = ["MIN\_YEARS\_EXPERIENCE", "EDU\_IDX"] + skill\_features  
  
importances = rf\_model.featureImportances.toArray()  
  
feature\_df = pd.DataFrame({  
 "Feature": all\_feature\_names,  
 "Importance": importances  
}).sort\_values(by="Importance", ascending=False)

# Visualizing Feature Importance

top\_n = 10  
top\_features = feature\_df.head(top\_n)  
  
plt.figure(figsize=(10, 8))  
plt.barh(top\_features['Feature'], top\_features['Importance'], color='skyblue')  
plt.xlabel("Feature Importance")  
plt.title(f"Top {top\_n} Feature Importances from Random Forest")  
plt.gca().invert\_yaxis()  
plt.tight\_layout()  
plt.savefig("./visualizations/rf\_feature\_importance.png", dpi=300, bbox\_inches='tight')  
  
#plt.show()



Top 10 Rf imp

# Extracting evaluation metrics

from pyspark.ml.evaluation import RegressionEvaluator  
evaluator = RegressionEvaluator(labelCol="SALARY", predictionCol="prediction")  
  
rmse = evaluator.setMetricName("rmse").evaluate(rf\_preds)  
r2 = evaluator.setMetricName("r2").evaluate(rf\_preds)  
  
metrics\_df = pd.DataFrame({  
 'Metric': ['Root Mean Squared Error (RMSE)', 'R² (Coefficient of Determination)'],  
 'Value': [round(rmse, 2), round(r2, 4)]  
})  
metrics\_df.to\_csv("./data/rf\_metrics\_table.csv")  
print(metrics\_df.to\_string(index=False))

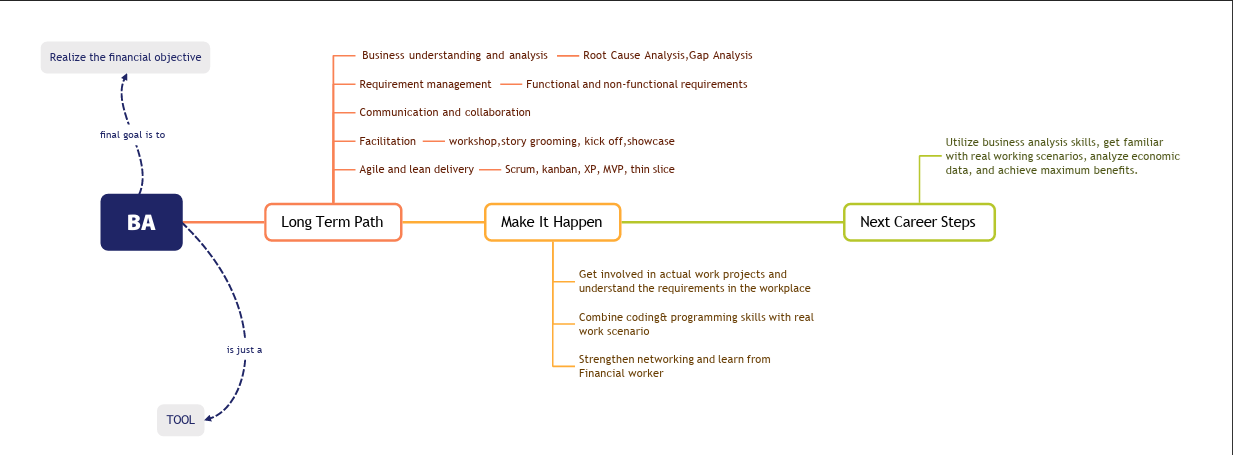
bklaboqou ./data/rf\_metrics\_table.csv

**RF Evaluation & Analysis** - From extracting the feature importances we know that minimum years experience plays a strong driver in predicting salary, however in terms of others like Enterprise Architecture, Data analysis and MapReduce are specialized skills and since we are using both Business Analyst and Data Scientist job postings to train this it is hard to evaluate if this would affect the salaries of both the positions in the same way. Mostly Mapreduce and cloud architecture are not the skills that are expected from a data analyst so this is likely attributed to Data Scientist. From the results it seems it would have been better to train the model and then make predictions using it separately for both positions, this would have limited the number of features and would have made the model more streamlined. If we look at the RMSE of the model it is deviating $29.6K from the actual salary values while making predictions which is acceptable considering the real-world factors where only skill sets and experience alone won’t determine the final value of the salary. However, these are strong drivers and from the R2 value the model is doing much better than capturing the mean value of the salary in the data range, however we would better look at RMSE values and features context than R2 for picking a model.

# Conclusion

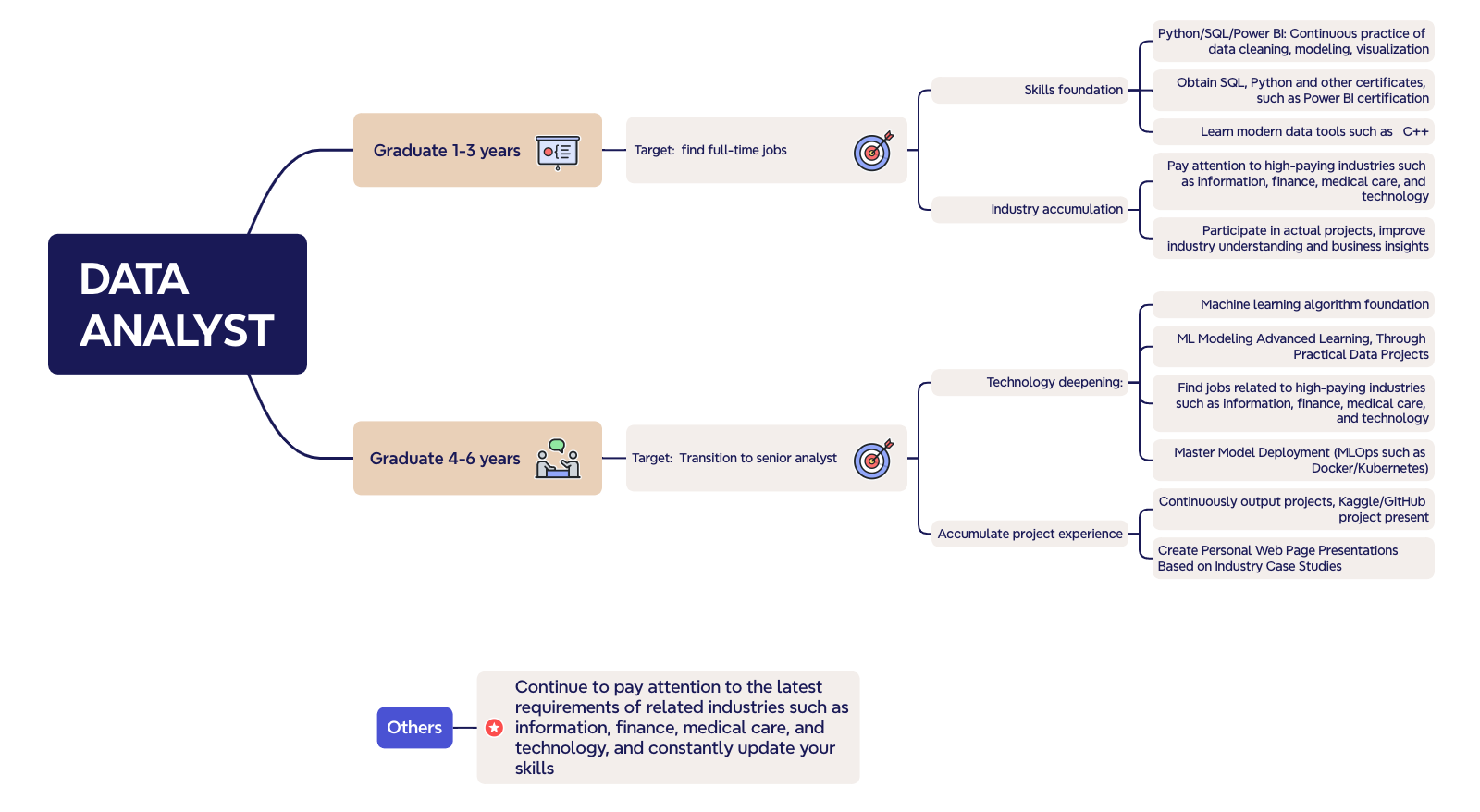
Summarizing the analysis from the various processes that we conducted above, we can briefly conclude that the market is extremely competitive at this stage, even though we did not look at specific AI trends from skills perspective we know that it has pushed the standards in the market up for data analysts as well by some margin. SQL is the core of data analysis be it data scientist or analyst, this tool becomes the first access point to data and is essential to condense the data into smaller subsets relatively to process it further of ML experimentation or insights reporting through dashboards. From an experience standpoint, simply completing Masters with generalized projects won’t make the cut, there has to be specific strategic learning process to get ahead of the clutter in the labor market. Lastly, in terms of further analysis we could further group the skills into subsets and use some NLP techniques to capture what specific skills in an industry can be mapped to either of these positions and what is common irrespective of the vertical. Although we are able to make some general statements about this but interesting patterns could be discovered from this analysis.

# Xinyi Dong (Sheryl) Carrer plan



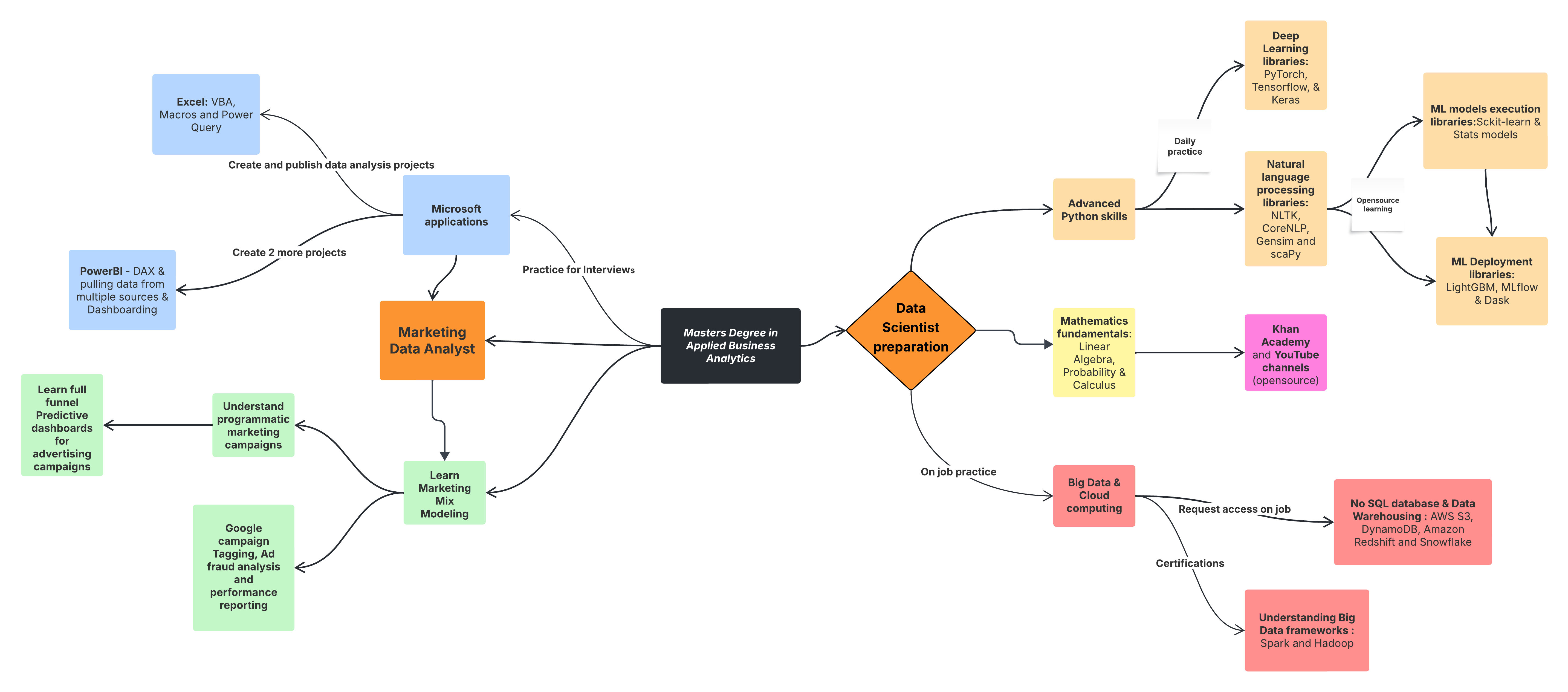
Sheryl’s career path

# Yao Ma (April) Career plan



Yao’s career path

# Hiten’s Career plan



Hiten’s career path

# Conclusion:

Each of our plan is based on the industry requirements as well as our the way we have envisioned our career paths for 3-5 years down the line. One element however is common in all of the career paths, the specific steps chosen are driven by the conclusions from research and quantitative analysis that was conducted. We found that constant learning and willingness to attain expertise in more than one skills is what is in demand in the current labor market. However, with the advent of AI soft skills like critical thinking, creativity and unconventional approach to problem solving is more at the forefront and we are striving to inculcate that approach in our daily practice.