**CASE STUDY 5: TRAINER STATISTICS**

**DESCRIPTION:** Trainers are available in ARICH who have different skill set & background. While

many of them are in-house employees few others are consultants. Trainers are allocated to

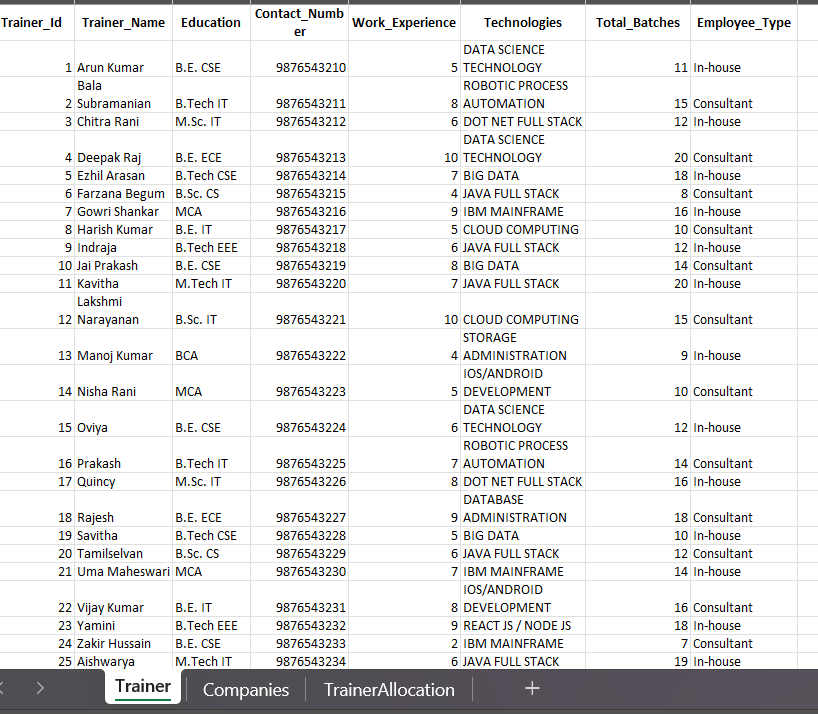
in-house as well as corporate training requirements based on demands & duration. Collect

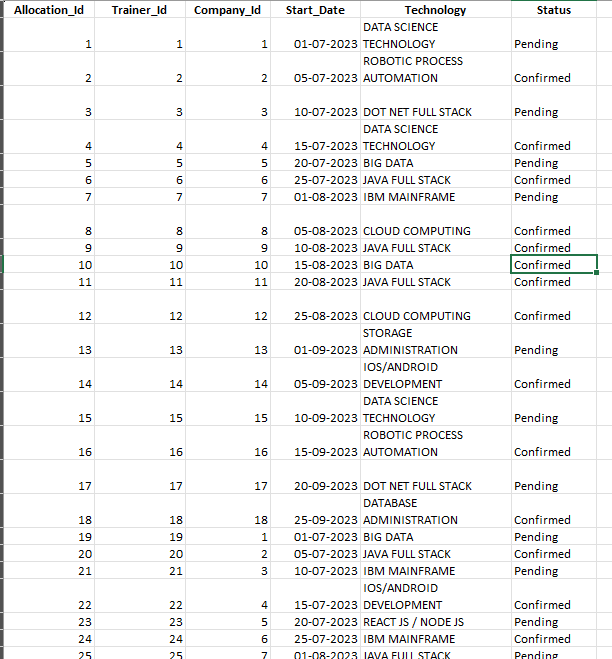
data relevant to this business case and do a projection on the demand for trainer on specific

technologies, matching skill sets of trainers & requirements fulfilled by in-house &

consultant trainers and in what time they are fulfilled and what up-skilling needed for

trainer based on requirement that could not be fulfilled.

**DATAS FOR INPUT:**



**PYTHON PROGRAM:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LinearRegression, LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

# Load the data from Excel

trainers\_df = pd.read\_excel('trainers.xlsx', sheet\_name='Trainer')

companies\_df = pd.read\_excel('trainers.xlsx', sheet\_name='Companies')

allocations\_df = pd.read\_excel('trainers.xlsx', sheet\_name='TrainerAllocation')

# Check for validity and integrity of the data

print(trainers\_df.isnull().sum())

print(companies\_df.isnull().sum())

print(allocations\_df.isnull().sum())

# Check for missing values and fill with empty string if any

trainers\_df.fillna('', inplace=True)

companies\_df.fillna('', inplace=True)

allocations\_df.fillna('', inplace=True)

# Convert date columns to datetime format

companies\_df['Expected\_Start\_Date'] = pd.to\_datetime(companies\_df['Expected\_Start\_Date'])

allocations\_df['Start\_Date'] = pd.to\_datetime(allocations\_df['Start\_Date'])

# Apply mathematical formula

average\_work\_experience = trainers\_df['Work\_Experience'].mean()

max\_experience = trainers\_df['Work\_Experience'].max()

min\_experience = trainers\_df['Work\_Experience'].min()

print(f'Maximum Work Experience: {max\_experience} years')

print(f'Minimum Work Experience: {min\_experience} years')

print(f'Average Work Experience: {average\_work\_experience}')

# Aggregate the data to count the number of trainers for each level of work experience

experience\_count = trainers\_df['Work\_Experience'].value\_counts().sort\_index()

print(experience\_count)

"""# Create Training\_Gap feature in companies\_df

companies\_df['Training\_Gap'] = companies\_df['Trainers\_Count'] - allocations\_df.groupby('Company\_Id')['Trainer\_Id'].transform('count')

print(companies\_df)"""

# Descriptive statistics

print(trainers\_df.describe())

print(companies\_df.describe())

print(allocations\_df.describe())

# Distribution analysis

print("Mean of Work Experience:", trainers\_df['Work\_Experience'].mean())

print("Standard Deviation of Work Experience:", trainers\_df['Work\_Experience'].std())

print("Mode of Work Experience:", trainers\_df['Work\_Experience'].mode()[0])

print("Mean of Trainers Count:", companies\_df['Trainers\_Count'].mean())

print("Standard Deviation of Trainers Count:", companies\_df['Trainers\_Count'].std())

print("Mode of Trainers Count:", companies\_df['Trainers\_Count'].mode()[0])

#Trainers Background

data = {

    'Trainer\_Id': range(1, 26),

    'Work\_Experience':[5,8,6,10,7,4,9,5,6,8,7,10,4,5,6,7,8,9,5,6,7,8,9,2,6]

    }

df = pd.DataFrame(data)

# Work Experience of trainers

plt.figure(figsize=(10, 6))

sns.histplot(trainers\_df['Work\_Experience'], kde=True)

plt.title('Distribution of Work Experience')

plt.show()

# Plotting the bar plot

plt.figure(figsize=(25, 10))

sns.barplot(x=experience\_count.index, y=experience\_count.values, palette='viridis')

plt.title('Number of Trainers by Work Experience')

plt.xlabel('Work Experience (Years)')

plt.ylabel('Number of Trainers')

plt.xticks(rotation=0)

plt.show()

# Demand for trainers on specific technologies

technology\_demand = companies\_df.groupby('Technology').size()

plt.figure(figsize=(12, 8))

technology\_demand.plot(kind='bar')

plt.title('Demand for Trainers on Specific Technologies')

plt.xlabel('Technology')

plt.ylabel('Number of Companies')

plt.show()

# Matching skill sets of trainers

skill\_sets = trainers\_df['Technologies'].value\_counts()

plt.figure(figsize=(12, 8))

skill\_sets.plot(kind='bar')

plt.title('Skill Sets of Trainers')

plt.xlabel('Technologies')

plt.ylabel('Number of Trainers')

plt.show()

# Requirements fulfilled by in-house & consultant trainers

inhouse\_fulfillment = allocations\_df[allocations\_df['Status'] == 'Fulfilled'][allocations\_df['Trainer\_Id'].isin(trainers\_df[trainers\_df['Employee\_Type'] == 'In-house']['Trainer\_Id'])]

consultant\_fulfillment = allocations\_df[allocations\_df['Status'] == 'Fulfilled'][allocations\_df['Trainer\_Id'].isin(trainers\_df[trainers\_df['Employee\_Type'] == 'Consultant']['Trainer\_Id'])]

# Upskilling needed for trainers based on unfulfilled requirements

unfulfilled\_requirements = allocations\_df[allocations\_df['Status'] != 'Fulfilled']

upskilling\_needed = unfulfilled\_requirements.groupby('Technology').size()

plt.figure(figsize=(12, 8))

upskilling\_needed.plot(kind='bar')

plt.title('Upskilling Needed Based on Unfulfilled Requirements')

plt.xlabel('Technology')

plt.ylabel('Number of Unfulfilled Requirements')

plt.show()

**OUTPUT:**

Maximum Work Experience: 10 years

Minimum Work Experience: 2 years

Average Work Experience: 6.68

Work\_Experience No\_of\_People

2 1

4 2

5 4

6 5

7 4

8 4

9 3

10 2

Trainer\_Id Work\_Experience Total\_Batches

count 25.000000 25.000000 25.000000

mean 13.000000 6.680000 13.840000

std 7.359801 1.994158 3.771383

min 1.000000 2.000000 7.000000

25% 7.000000 5.000000 11.000000

50% 13.000000 7.000000 14.000000

75% 19.000000 8.000000 16.000000

max 25.000000 10.000000 20.000000

Company\_Id Trainers\_Count Expected\_Start\_Date Duration Training\_Gap

count 18.000000 18.000000 18 18.000000 18.000000

mean 9.500000 4.500000 2024-08-12 16:00:00 27.500000 3.111111

min 1.000000 3.000000 2024-07-01 00:00:00 20.000000 1.000000

25% 5.250000 4.000000 2024-07-21 06:00:00 25.000000 2.250000

50% 9.500000 4.500000 2024-08-12 12:00:00 27.500000 3.000000

75% 13.750000 5.000000 2024-09-04 00:00:00 30.000000 4.000000

max 18.000000 6.000000 2024-09-25 00:00:00 35.000000 5.000000

std 5.338539 0.985184 NaN 4.925922 1.182663

Allocation\_Id Trainer\_Id Company\_Id Start\_Date

count 25.000000 25.000000 25.000000 25

mean 13.000000 13.000000 7.960000 2023-08-04 18:14:24

min 1.000000 1.000000 1.000000 2023-07-01 00:00:00

25% 7.000000 7.000000 4.000000 2023-07-15 00:00:00

50% 13.000000 13.000000 7.000000 2023-08-01 00:00:00

75% 19.000000 19.000000 12.000000 2023-08-25 00:00:00

max 25.000000 25.000000 18.000000 2023-09-25 00:00:00

std 7.359801 7.359801 5.263712 NaN

Mean of Work Experience: 6.68

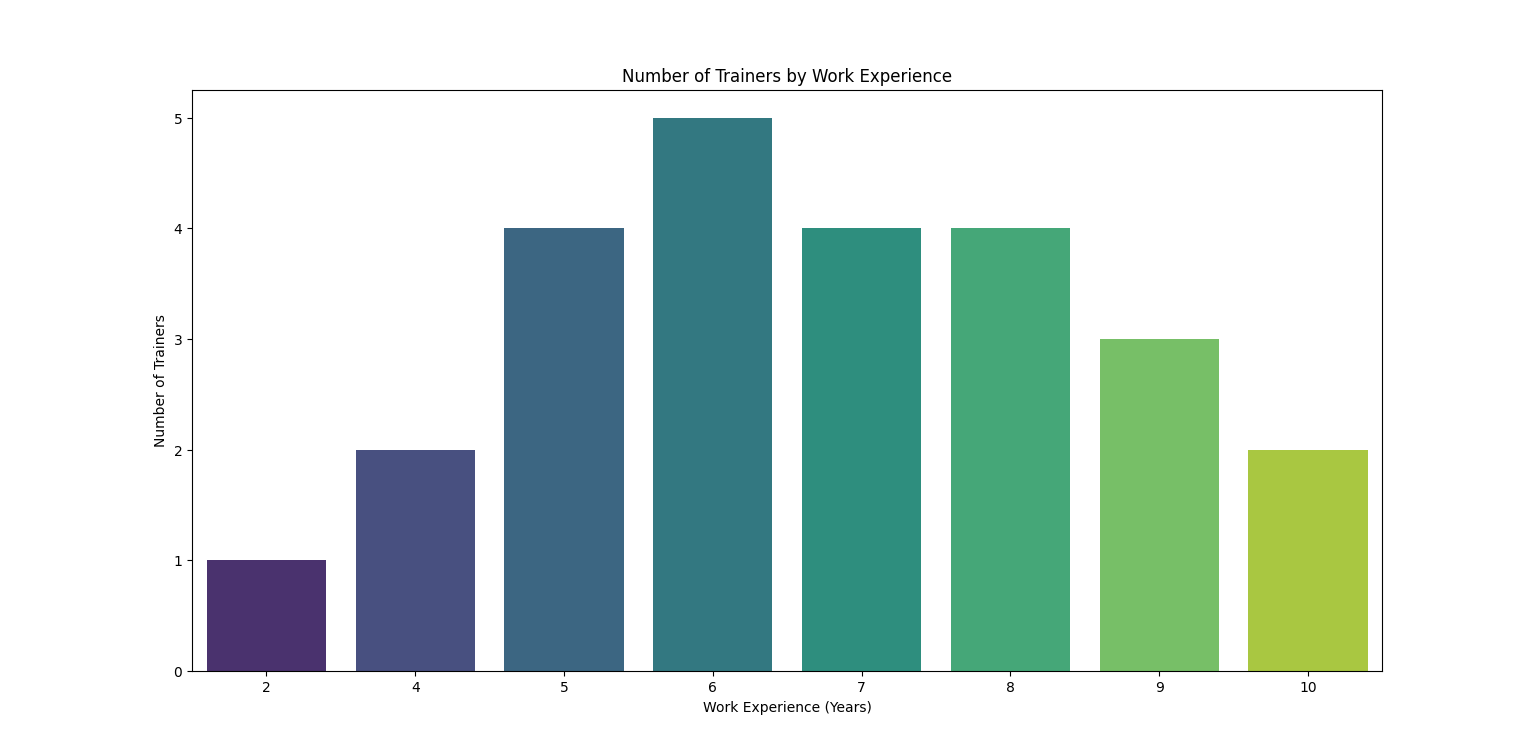
Standard Deviation of Work Experience: 1.99415813481947

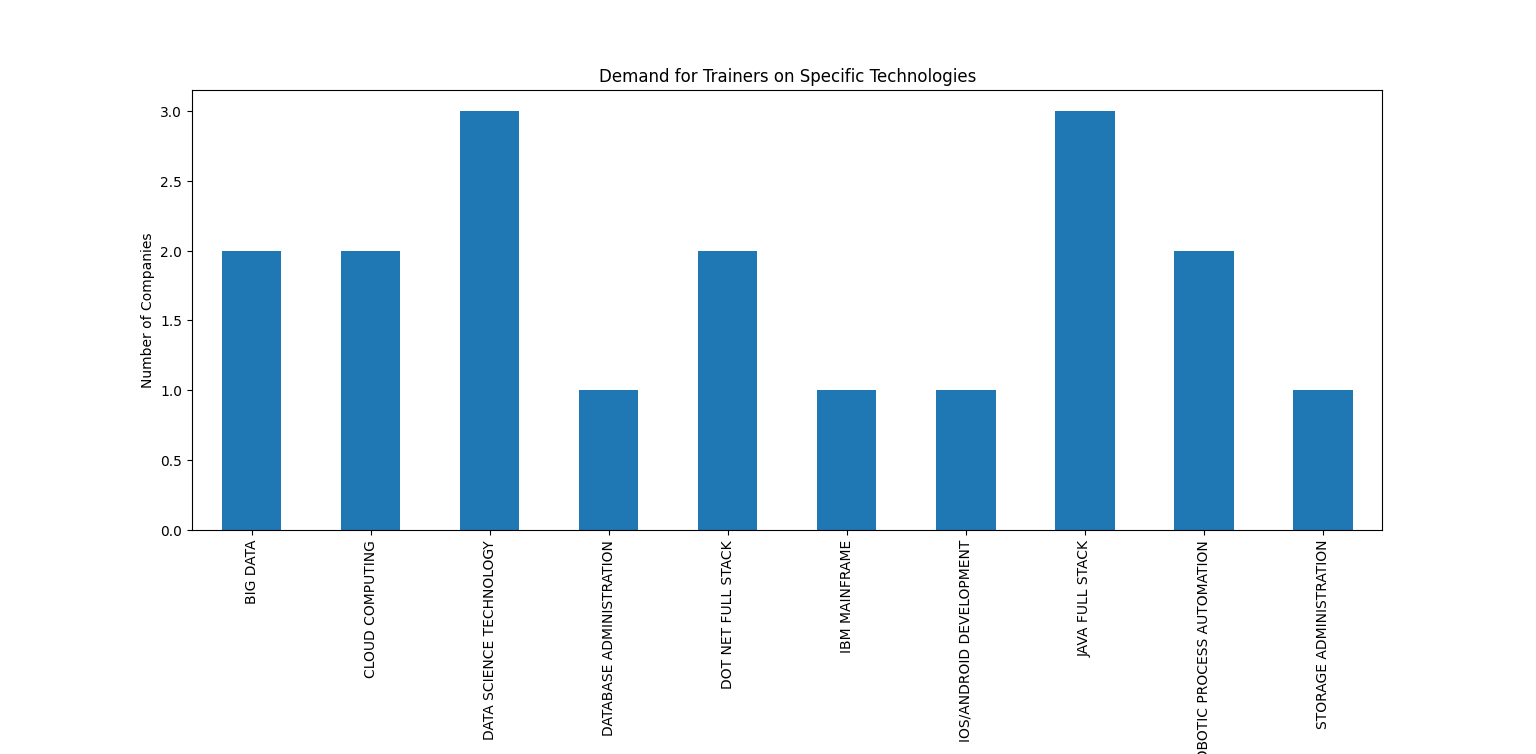
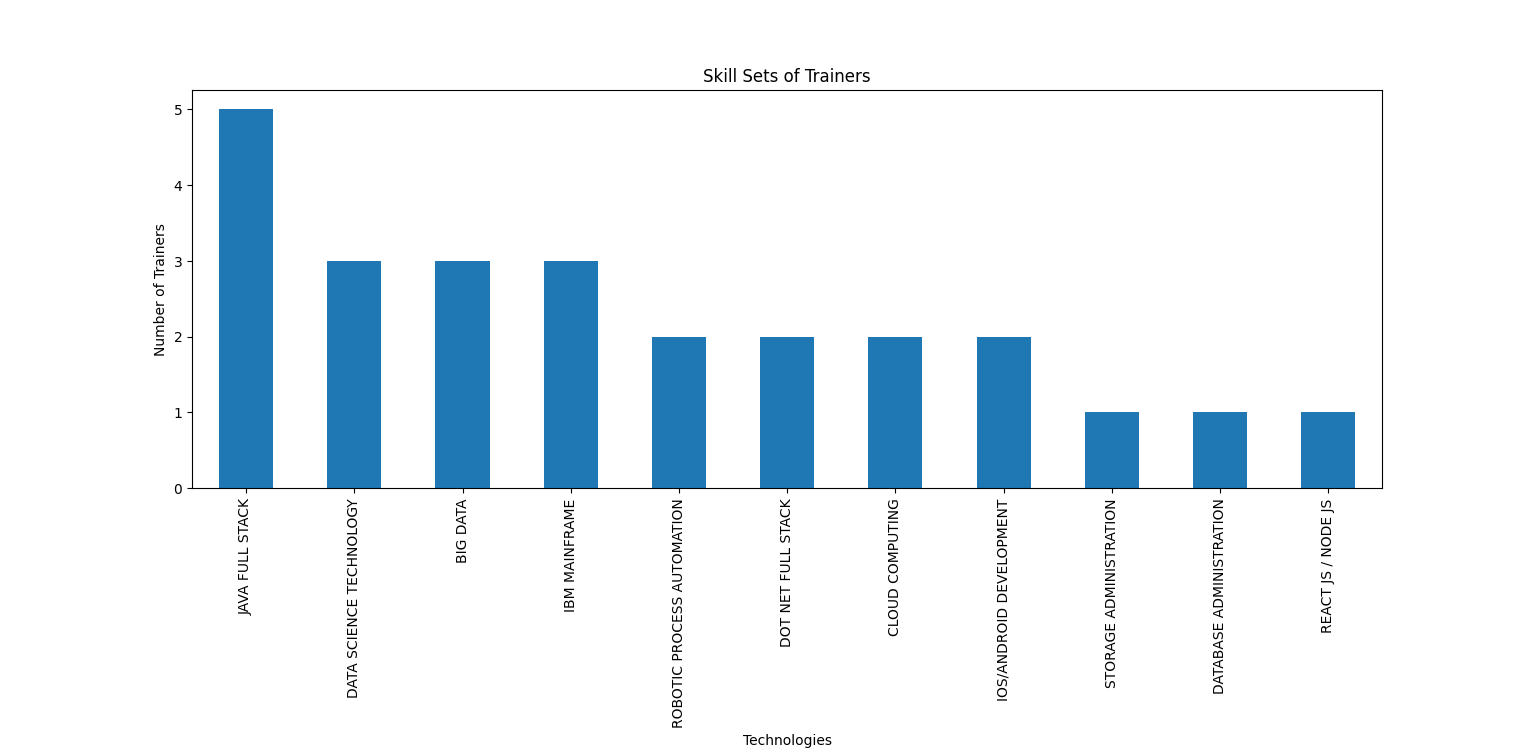
Mode of Work Experience: 6

Mean of Trainers Count: 4.5

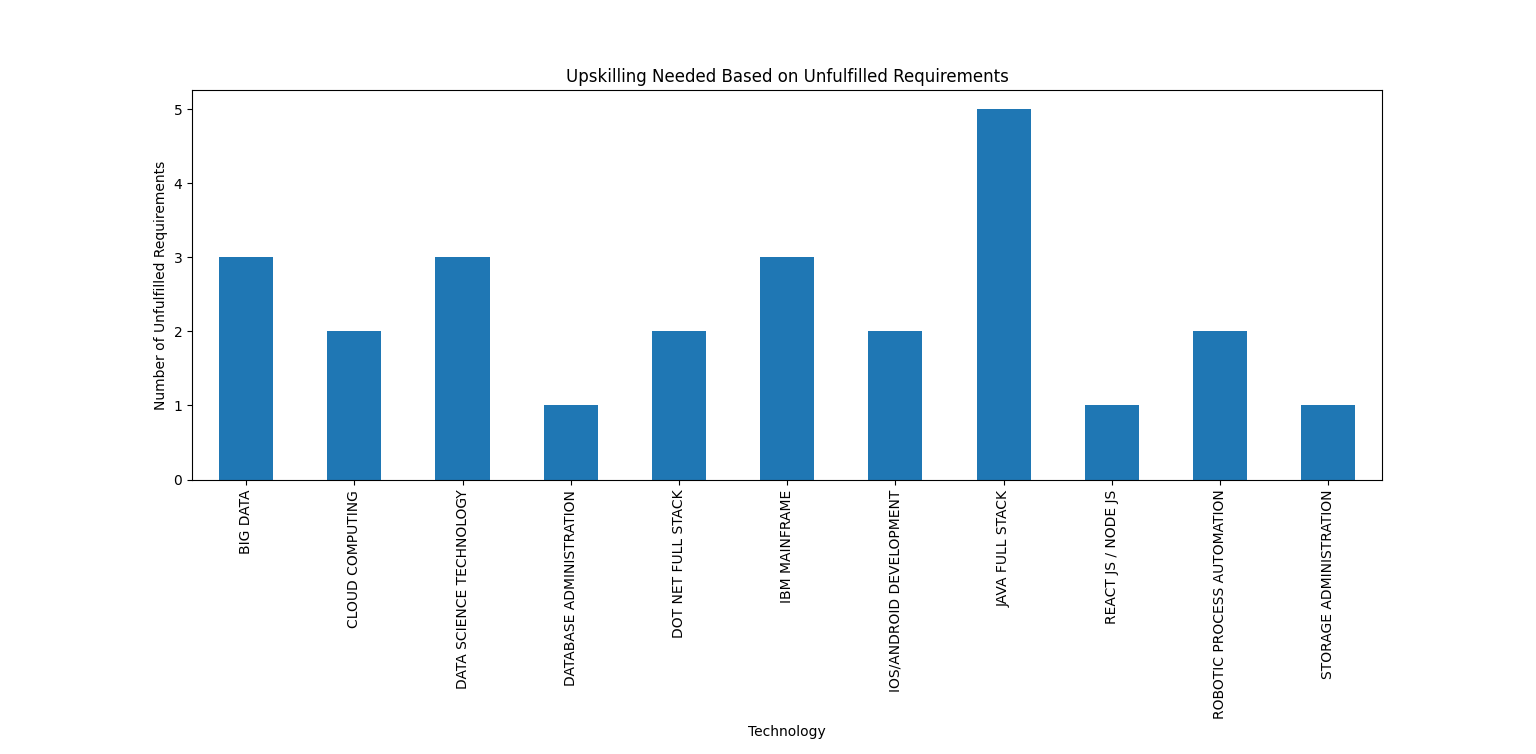
StandardDeviation of Trainers Count: 0.985184366143778

Mode of Trainers Count: 4

**DATA VISUALISATION:**





****

**Overall Insights:**

* **Demand for Courses Among Students:**
  + High demand for Data Science Technology and Java Full Stack courses.
  + Courses related to Big Data, React JS / Node JS, and Networking and Cyber Security are also popular among students.
  + Technologies like Robotic Process Automation and Cloud Computing are gaining traction.
  + Some technologies have shorter fulfillment times, indicating efficient training processes, while others may require strategic planning to meet demands promptly.
* **Industry Needs:**
  + IT Services and Consulting firms have the highest demand for trainers.
  + Specific technologies in demand include Data Science Technology, Dot Net Full Stack, and Database Administration.
  + Big Data and Cloud Computing are critical areas where industries are seeking skilled professionals.
  + Companies are increasingly looking for trainers with expertise in Emerging Technologies such as AI/ML, Cyber Security, and DevOps.
* **Recruitment Trends:**
  + Consultant Trainers are more in demand for niche and specialized training requirements.
  + In-house Trainers are preferred for ongoing and long-term training programs.
  + Fulfillment Time for training demands varies, with some technologies taking longer to fulfill due to high demand or complexity.
  + Technologies with faster fulfillment times include Java Full Stack and Database Administration.
  + Skill Gaps identified in areas like Robotic Process Automation and Cloud Computing suggest a need for up-skilling existing trainers in these domains

**Upskilling Recommendations:**

Based on the analysis of the current demand for technologies and the gaps identified in the skill sets of trainers, the following upskilling recommendations are made for ARICH trainers:

1. **Emerging Technologies:**
   * **Data Science and Machine Learning:** Given the high demand, trainers should undergo advanced courses and certifications in Data Science, AI, and Machine Learning.
   * **Cloud Computing:** Trainers should be proficient in platforms like AWS, Azure, and Google Cloud. Certifications such as AWS Certified Solutions Architect or Google Cloud Professional Data Engineer would be beneficial.
2. **Programming Languages and Frameworks:**
   * **Full Stack Development:** Trainers need to master modern frameworks such as React, Angular, Node.js, and Django. Full Stack Development bootcamps and certifications would help.
   * **Mobile Development:** There is a growing need for expertise in iOS and Android development. Trainers should aim for certifications and hands-on projects in Swift (iOS) and Kotlin (Android).