INFO-I535 Final Project Fall 2023

Big Data Concepts

Mental Health in Tech

Shashwati Diware (sdiware@iu.edu)

INTRODUCTION:

In the fast-paced and dynamic landscape of the technology industry, the well-being of its workforce is a critical aspect often overlooked. This project delves into a comprehensive analysis of attitudes towards mental health and the prevalence of mental health disorders within the tech workplace. The dataset under examination originates from a survey, providing valuable insights into the perceptions and experiences of individuals working in the tech sector regarding mental health.

The dataset has parameters such as **age, gender**, and **country,** alongside workplace-specific details like **employment status, company size,** and **remote work** arrangements. Notably, the dataset also explores personal and organizational perspectives on **mental health**, touching upon **family history, treatment,** and the impact of **mental health conditions** on professional life.

In this project I have used fundamental ideas in big data management access to try and understand the complex dynamics of mental health in the tech industry as below:

1. Data Types and Sources
2. Cloud Computing
3. Virtualization
4. Lifecycles and Pipelines
5. Data Ingest and Storage
6. Data Processing and Analytics
7. Impact of Big Data (AI Fairness)

In this project, I initiated the data acquisition phase by sourcing a dataset from Kaggle, and for efficient management, I utilized MongoDB Atlas as the storage solution. Leveraging Python Jupyter notebooks, I seamlessly fetched and pre-processed the data, implementing necessary cleaning and refinement procedures to enhance its analytical suitability. The processed dataset was then pushed back into MongoDB for structured storage.

I conducted in-depth exploratory data analysis (EDA) using Jupyter notebooks and MongoDB Atlas Charts to gain profound insights into the dataset's patterns. The refined data was then stored in Google Cloud Platform (GCP) storage buckets, harnessing the cloud's scalability for further analysis. GitHub served as the central repository for the project, ensuring effective version control, collaboration, and knowledge-sharing within the technical community.

BACKGROUND:

This project explores the intersection of mental health and the professional field in the technology sector using a survey-derived dataset. The comprehensive dataset includes workplace-specific attributes and demographic information, providing insights into the complex relationship between mental health and the tech workplace. Key aspects investigated involve the availability of mental health benefits, the impact on professional life, and the willingness to discuss mental health with colleagues and managers.

Motivated by the dataset's richness, the project aims to uncover geographical disparities in mental health issues and attitudes within the tech industry. It seeks to identify predictors of mental health conditions and factors shaping attitudes in the workplace. By addressing these questions, the project contributes to the broader discourse on mental health in technology, paving the way for strategies and interventions to foster a healthier and more supportive work environment. This background sets the stage for a data-driven investigation into the perspectives of mental health in the rapidly evolving field of technology, emphasizing the significance of big data concepts.

• METHODOLOGY:

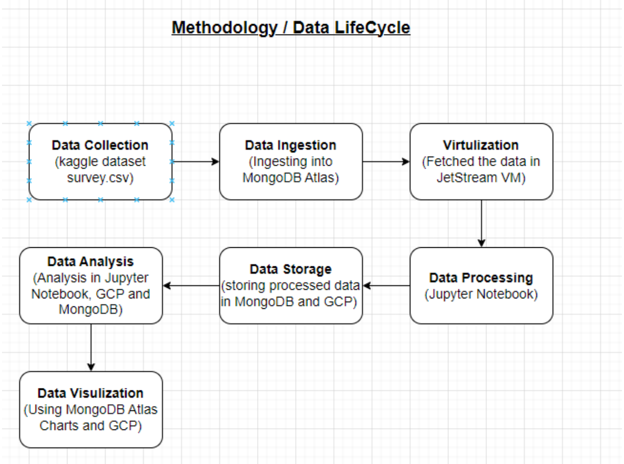

Methodology / Data LifeCycle
Data Collection (kaggle dataset survey.csv)
Data Ingestion (Ingesting into MongoDB Atlas)
Virtulization (Fetched the data in JetStream VM)
Data Processing (Jupyter Notebook)
Data Storage (storing processed data in MongoDB and GCP)
Data Analysis (Analysis in Jupyter Notebook, GCP and MongoDB)
Data Visulization (Using MongoDB Atlas Charts and GCP)

Fig: Complete methodology along with data lifecycle stages

Below are the processes involved for this project:

1. Data Collection:

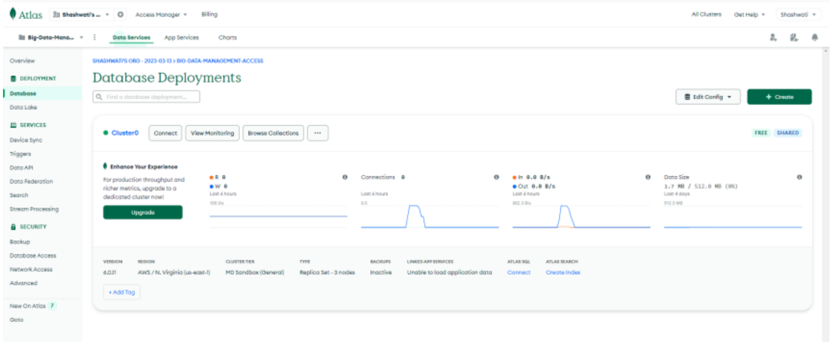
Source: The dataset was obtained from Kaggle.

2. Data Ingestion:

Storage Solution: MongoDB Atlas was employed for efficient data management.

Approach:

- Downloaded the dataset from Kaggle.
- Uploaded to MongoDB via MongoDB Compass with the help of user mongo, which is created explicitly to read/write the data in Mongo.
- Created a project with name Big-Data-Management-Access.
- Created a cluster with name Cluster0 in MongoDB in AWS /N Virginia (us-east-1).
- I have added a collection in database called survey.
- The collection is survey.techSurveyData.
- survey.techSurveyData has raw data.
- I have accessed this collection in Jupyter notebook via connection string.



SHASHWATI'S ORG - 2023-03-13 > BIG-DATA-MANAGEMENT-ACCESS

Database Deployments

Cluster0

Enhance Your Experience

For production throughput and richer metrics, upgrade to a dedicated cluster now!

R 0 / W 0 — Last 6 hours

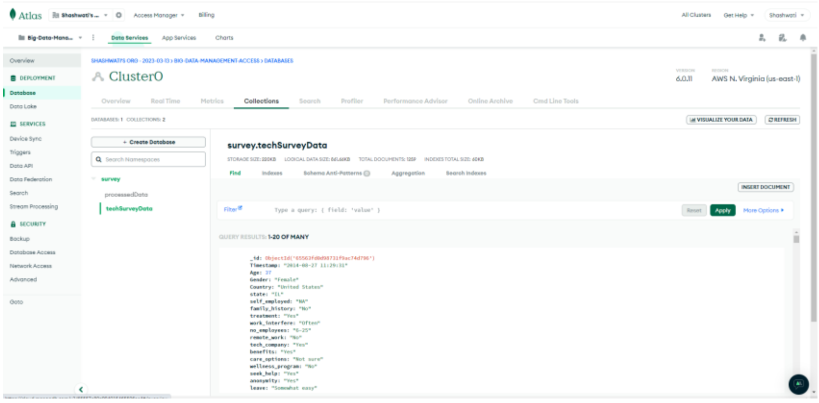
Connections 0 — Last 6 hours

In 0.0 B/s / Out 0.0 B/s — Last 6 hours

Data Size 1.7 MB / 512.0 MB (0%) — Last 4 days

VERSION 6.0.11 · REGION AWS / N. Virginia (us-east-1) · CLUSTER TIER M0 Sandbox (General) · TYPE Replica Set - 3 nodes · BACKUPS Inactive · LINKED APP SERVICES Unable to load application data · ATLAS SQL Connect · ATLAS SEARCH Create Index

Fig 1: Cluster0 in MongoDB



SHASHWATI'S ORG - 2023-03-13 > BIG-DATA-MANAGEMENT-ACCESS > DATABASES

Cluster0

VERSION 6.0.11 · REGION AWS N. Virginia (us-east-1)

DATABASES: 1 COLLECTIONS: 2

survey.techSurveyData

STORAGE SIZE: 220KB · LOGICAL DATA SIZE: 841.46KB · TOTAL DOCUMENTS: 1259 · INDEXES TOTAL SIZE: 60KB

QUERY RESULTS: 1-20 OF MANY

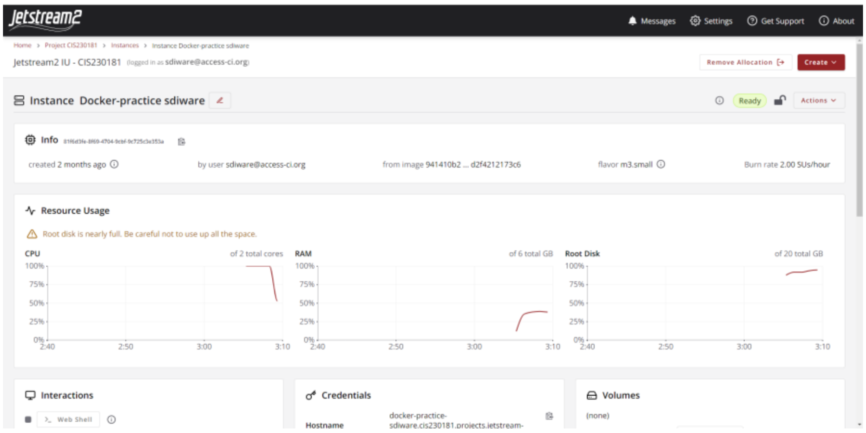
```
_id: ObjectId('65563fd0d98731f9ac74d796')
Timestamp: "2014-08-27 11:29:31"
Age: 37
Gender: "Female"
Country: "United States"
state: "IL"
self_employed: "NA"
family_history: "No"
treatment: "Yes"
work_interfere: "Often"
no_employees: "6-25"
remote_work: "No"
tech_company: "Yes"
benefits: "Yes"
care_options: "Not sure"
wellness_program: "No"
seek_help: "Yes"
anonymity: "Yes"
leave: "Somewhat easy"
```

Fig: Raw data uploaded in techSurveyData collection

3. Virtualization with Jetstream VM:

Approach:

- To execute my notebook in more computational strong resources for faster execution I used Jetstream VMs.
- I installed Jupyter notebook using docker image.
- After making this docker image up I launched the Jupyter in browser.
- I executed all my code in the Jupyter notebook on VMs to make use visualization.



Home > Project CIS230181 > Instances > Instance Docker-practice sdiware

Jetstream2 IU - CIS230181 (logged in as sdiware@access-ci.org)

Instance Docker-practice sdiware

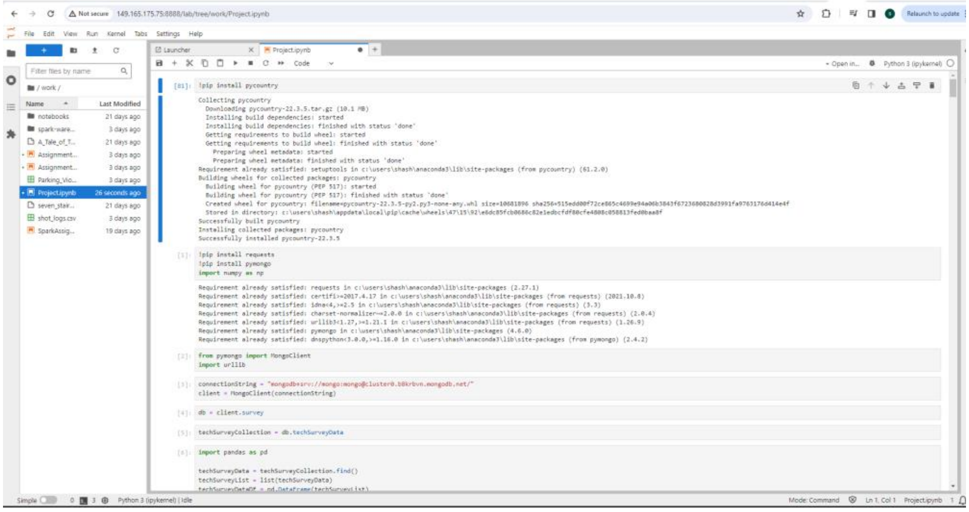
Info 81f6d3fe-8f69-4704-9cbf-9c725c3e353a

created 2 months ago · by user sdiware@access-ci.org · from image 941410b2 ... d2f4212173c6 · flavor m3.small · Burn rate 2.00 SUs/hour

Resource Usage

Root disk is nearly full. Be careful not to use up all the space.

Fig: Jetstream VM


Not secure 149.165.175.75:8888/lab/tree/work/Project.ipynb
Relaunch to update
File Edit View Run Kernel Tabs Settings Help
Launcher
Project.ipynb
Code
Open in...
Python 3 (ipykernel)
Filter files by name
/ work /
Name
Last Modified
notebooks 21 days ago
spark-ware... 3 days ago
A_Tale_of_T... 21 days ago
Assignment... 3 days ago
Assignment... 3 days ago
Parking_Vio... 3 days ago
Project.ipynb 26 seconds ago
seven_stair... 21 days ago
shot_logs.csv 3 days ago
SparkAssig... 19 days ago
Simple 0 3 Python 3 (ipykernel) | Idle
Mode: Command Ln 1, Col 1 Project.ipynb 1

```
[81]: !pip install pycountry

Collecting pycountry
  Downloading pycountry-22.3.5.tar.gz (10.1 MB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Requirement already satisfied: setuptools in c:\users\shash\anaconda3\lib\site-packages (from pycountry) (61.2.0)
Building wheels for collected packages: pycountry
  Building wheel for pycountry (PEP 517): started
  Building wheel for pycountry (PEP 517): finished with status 'done'
  Created wheel for pycountry: filename=pycountry-22.3.5-py2.py3-none-any.whl size=10681896 sha256=515edd00f72ce865c4699e94a06b3843f6723680028d3991fa9763176d414e4f
  Stored in directory: c:\users\shash\appdata\local\pip\cache\wheels\47\15\92\e6dc85fcb0686c82e1edbcfdf80cfe4808c058813fed0baa8f
Successfully built pycountry
Installing collected packages: pycountry
Successfully installed pycountry-22.3.5

[1]: !pip install requests
     !pip install pymongo
     import numpy as np

Requirement already satisfied: requests in c:\users\shash\anaconda3\lib\site-packages (2.27.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shash\anaconda3\lib\site-packages (from requests) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\shash\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\shash\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\shash\anaconda3\lib\site-packages (from requests) (1.26.9)
Requirement already satisfied: pymongo in c:\users\shash\anaconda3\lib\site-packages (4.6.0)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\shash\anaconda3\lib\site-packages (from pymongo) (2.4.2)

[2]: from pymongo import MongoClient
     import urllib

[3]: connectionString = "mongodb+srv://mongo:mongo@cluster0.b8krbvn.mongodb.net/"
     client = MongoClient(connectionString)

[4]: db = client.survey

[5]: techSurveyCollection = db.techSurveyData

[6]: import pandas as pd

     techSurveyData = techSurveyCollection.find()
     techSurveyList = list(techSurveyData)
     techSurveyDataDF = pd.DataFrame(techSurveyList)
```

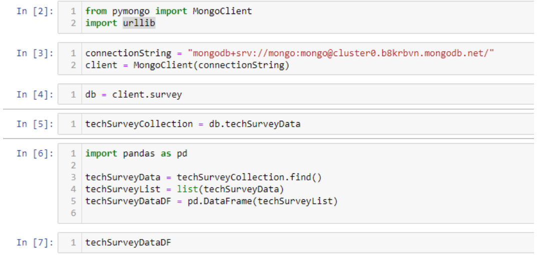
Fig: Jupyter notebook running on VM instance

4. Data Access in Jupyter:

Libs Used: pymongo, MongoClient, urllib, requests.

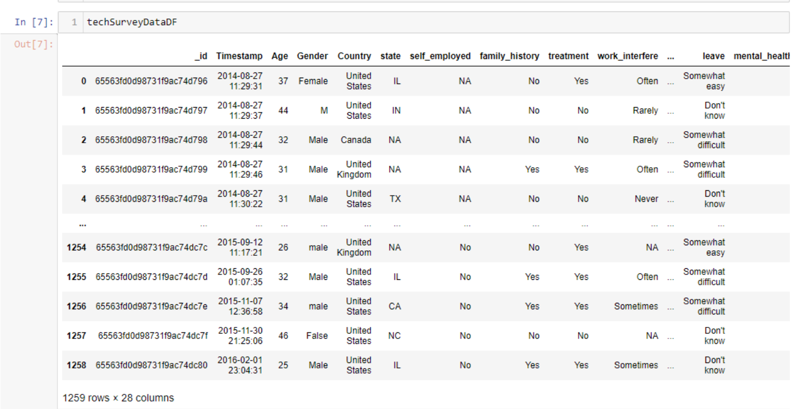
Approach:

- I used above mentioned libraries to fetch the data stored in mongo DB.
- With the help of connection string and username/password for the cluster I was able to fetch the data in Python notebook.
- This data was raw, I did further processing on which is discussed below in preprocessing stage.



```
In [2]: from pymongo import MongoClient
        import urllib

In [3]: connectionString = "mongodb+srv://mongo:mongo@cluster0.b8krbvn.mongodb.net/"
        client = MongoClient(connectionString)

In [4]: db = client.survey

In [5]: techSurveyCollection = db.techSurveyData

In [6]: import pandas as pd

        techSurveyData = techSurveyCollection.find()
        techSurveyList = list(techSurveyData)
        techSurveyDataDF = pd.DataFrame(techSurveyList)


In [7]: techSurveyDataDF
```

Fig: Code to fetch the data from MongoDB into Python Notebook



```
In [7]: techSurveyDataDF
```

Out[7]:

	_id	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	...	leave	mental_healtl
0	65563fd0d98731f9ac74d796	2014-08-27 11:29:31	37	Female	United States	IL	NA	No	Yes	Often	...	Somewhat easy	
1	65563fd0d98731f9ac74d797	2014-08-27 11:29:37	44	M	United States	IN	NA	No	No	Rarely	...	Don't know	
2	65563fd0d98731f9ac74d798	2014-08-27 11:29:44	32	Male	Canada	NA	NA	No	No	Rarely	...	Somewhat difficult	
3	65563fd0d98731f9ac74d799	2014-08-27 11:29:46	31	Male	United Kingdom	NA	NA	Yes	Yes	Often	...	Somewhat difficult	
4	65563fd0d98731f9ac74d79a	2014-08-27 11:30:22	31	Male	United States	TX	NA	No	No	Never	...	Don't know	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1254	65563fd0d98731f9ac74dc7c	2015-09-12 11:17:21	26	male	United Kingdom	NA	No	No	Yes	NA	...	Somewhat easy	
1255	65563fd0d98731f9ac74dc7d	2015-09-26 01:07:35	32	Male	United States	IL	No	Yes	Yes	Often	...	Somewhat difficult	
1256	65563fd0d98731f9ac74dc7e	2015-11-07 12:36:58	34	male	United States	CA	No	Yes	Yes	Sometimes	...	Somewhat difficult	
1257	65563fd0d98731f9ac74dc7f	2015-11-30 21:25:06	46	False	United States	NC	No	No	No	NA	...	Don't know	
1258	65563fd0d98731f9ac74dc80	2016-02-01 23:04:31	25	Male	United States	IL	No	Yes	Yes	Sometimes	...	Don't know	

1259 rows × 28 columns

Fig: Data Fetched from the MongoDB into Python notebook

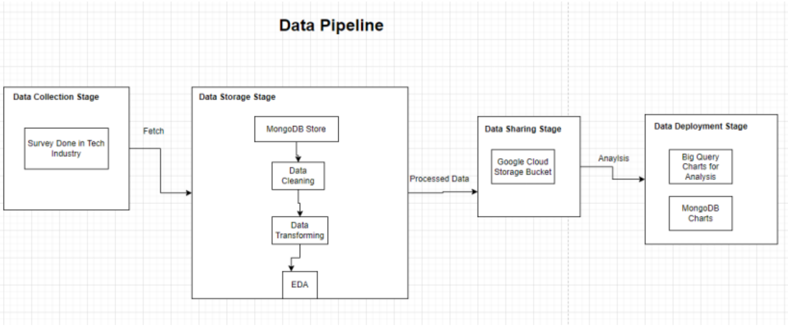


Fig: Data Pipeline

5. Data Pre-Processing:

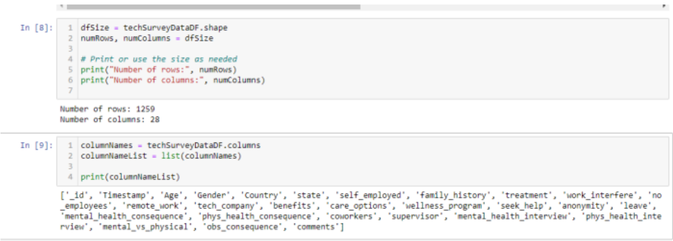
Tools Used: Python Jupyter notebooks.

Approach:

- Above figure shows data pipeline for this project.
- Data preprocessing involves task such as data cleaning, transforming, filtering out unnecessary data.
- Pushed the processed dataset back into MongoDB for structured storage.

Step 1: Data Cleaning:

- In in this step, I identified and addressed null values in the dataset.
- Converted initial representations of null values from "NA" to actual null values to ensure proper recognition and handling in subsequent data cleaning steps.
- Below is the structure of dataset.

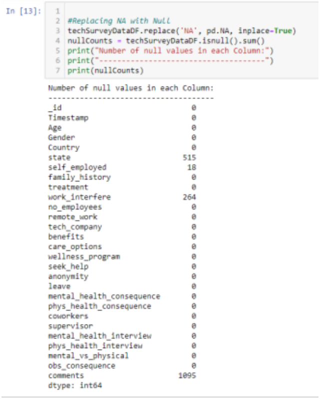


In [8]:
```
dfSize = techSurveyDataDF.shape
numRows, numColumns = dfSize

# Print or use the size as needed
print("Number of rows:", numRows)
print("Number of columns:", numColumns)

```
```
Number of rows: 1259
Number of columns: 28
```

In [9]:
```
columnNames = techSurveyDataDF.columns
columnNameList = list(columnNames)

print(columnNameList)
```
```
['_id', 'Timestamp', 'Age', 'Gender', 'Country', 'state', 'self_employed', 'family_history', 'treatment', 'work_interfere', 'no
_employees', 'remote_work', 'tech_company', 'benefits', 'care_options', 'wellness_program', 'seek_help', 'anonymity', 'leave',
'mental_health_consequence', 'phys_health_consequence', 'coworkers', 'supervisor', 'mental_health_interview', 'phys_health_inte
rview', 'mental_vs_physical', 'obs_consequence', 'comments']
```

Fig: Above is the structure of dataset and columns in dataset

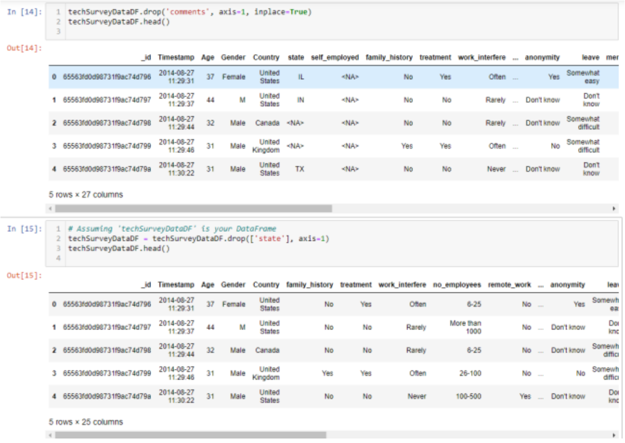
- Below are the NULL values in the original dataset.



In [13]:
```

#Replacing NA with Null
techSurveyDataDF.replace('NA', pd.NA, inplace=True)
nullCounts = techSurveyDataDF.isnull().sum()
print("Number of null values in each Column:")
print("-----------------------------------")
print(nullCounts)
```
```
Number of null values in each Column:
-------------------------------------
_id                          0
Timestamp                    0
Age                          0
Gender                       0
Country                      0
state                      515
self_employed               18
family_history               0
treatment                    0
work_interfere             264
no_employees                 0
remote_work                  0
tech_company                 0
benefits                     0
care_options                 0
wellness_program             0
seek_help                    0
anonymity                    0
leave                        0
mental_health_consequence    0
phys_health_consequence      0
coworkers                    0
supervisor                   0
mental_health_interview      0
phys_health_interview        0
mental_vs_physical           0
obs_consequence              0
comments                  1095
dtype: int64
```

Fig: NULL values in original dataset

- I dropped ‘state’ and ‘comments’ columns because they had 40% and 85% null values resp.
- This approach helps in streamlining the dataset by removing columns that contain a significant amount of missing data, promoting a more focused and relevant analysis.



In [14]:

```
techSurveyDataDF.drop('comments', axis=1, inplace=True)
techSurveyDataDF.head()
```

Out[14]:

	_id	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	...	anonymity	leave	mer
0	65563fd0d98731f9ac74d796	2014-08-27 11:29:31	37	Female	United States	IL	<NA>	No	Yes	Often	...	Yes	Somewhat easy	
1	65563fd0d98731f9ac74d797	2014-08-27 11:29:37	44	M	United States	IN	<NA>	No	No	Rarely	...	Don't know	Don't know	
2	65563fd0d98731f9ac74d798	2014-08-27 11:29:44	32	Male	Canada	<NA>	<NA>	No	No	Rarely	...	Don't know	Somewhat difficult	
3	65563fd0d98731f9ac74d799	2014-08-27 11:29:46	31	Male	United Kingdom	<NA>	<NA>	Yes	Yes	Often	...	No	Somewhat difficult	
4	65563fd0d98731f9ac74d79a	2014-08-27 11:30:22	31	Male	United States	TX	<NA>	No	No	Never	...	Don't know	Don't know	

5 rows × 27 columns

In [15]:

```
# Assuming 'techSurveyDataDF' is your DataFrame
techSurveyDataDF = techSurveyDataDF.drop(['state'], axis=1)
techSurveyDataDF.head()
```

Out[15]:

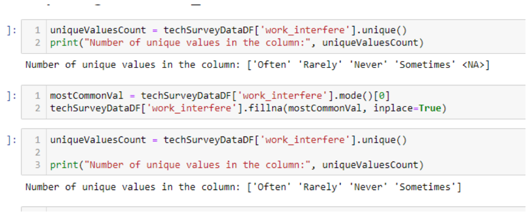
	_id	Timestamp	Age	Gender	Country	family_history	treatment	work_interfere	no_employees	remote_work	...	anonymity	lea
0	65563fd0d98731f9ac74d796	2014-08-27 11:29:31	37	Female	United States	No	Yes	Often	6-25	No	...	Yes	Somewh ea
1	65563fd0d98731f9ac74d797	2014-08-27 11:29:37	44	M	United States	No	No	Rarely	More than 1000	No	...	Don't know	Do kn
2	65563fd0d98731f9ac74d798	2014-08-27 11:29:44	32	Male	Canada	No	No	Rarely	6-25	No	...	Don't know	Somewh diffic
3	65563fd0d98731f9ac74d799	2014-08-27 11:29:46	31	Male	United Kingdom	Yes	Yes	Often	26-100	No	...	No	Somewh diffic
4	65563fd0d98731f9ac74d79a	2014-08-27 11:30:22	31	Male	United States	No	No	Never	100-500	Yes	...	Don't know	Do kn

5 rows × 25 columns

Fig: DataFrame after dropping state and comments column

Step 2: Data Transformation:

- In this step I have done few transformations on certain columns for better understanding of the data.
- Firstly, I replaced the NA values in work_interfere to modal value of that column because NA values count was not that significant but dropping the column was not an option.



```
]:  uniqueValuesCount = techSurveyDataDF['work_interfere'].unique()
    print("Number of unique values in the column:", uniqueValuesCount)
```

```
Number of unique values in the column: ['Often' 'Rarely' 'Never' 'Sometimes' <NA>]
```

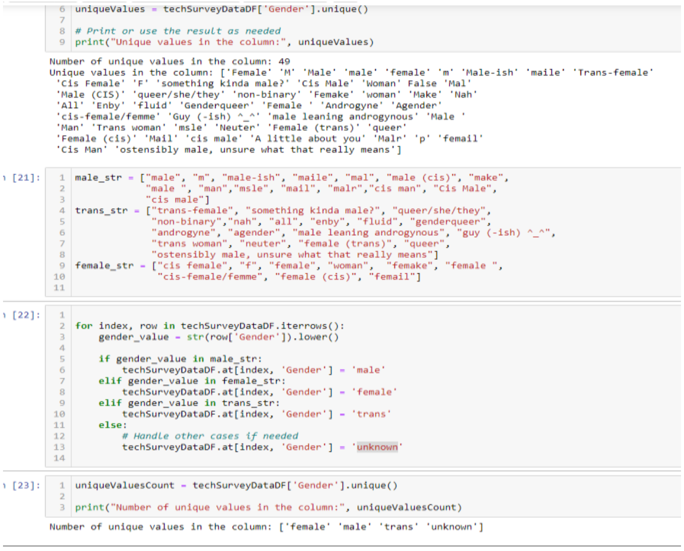
```
]:  mostCommonVal = techSurveyDataDF['work_interfere'].mode()[0]
    techSurveyDataDF['work_interfere'].fillna(mostCommonVal, inplace=True)
```

```
]:  uniqueValuesCount = techSurveyDataDF['work_interfere'].unique()

    print("Number of unique values in the column:", uniqueValuesCount)
```

```
Number of unique values in the column: ['Often' 'Rarely' 'Never' 'Sometimes']
```

Fig: Replace “NA” value in work_interfere column

- Secondly, the gender column had total 49 unique values, out of which some didn’t even make sense. Hence for better understanding I categorize the gender column into 4 different categories such as male, female, trans and unknown.



```
uniqueValues = techSurveyDataDF['Gender'].unique()

# Print or use the result as needed
print("Unique values in the column:", uniqueValues)
```

```
Number of unique values in the column: 49
Unique values in the column: ['Female' 'M' 'Male' 'male' 'female' 'm' 'Male-ish' 'maile' 'Trans-female'
 'Cis Female' 'F' 'something kinda male?' 'Cis Male' 'Woman' False 'Mal'
 'Male (CIS)' 'queer/she/they' 'non-binary' 'Femake' 'woman' 'Make' 'Nah'
 'All' 'Enby' 'fluid' 'Genderqueer' 'Female ' 'Androgyne' 'Agender'
 'cis-female/femme' 'Guy (-ish) ^_^' 'male leaning androgynous' 'Male '
 'Man' 'Trans woman' 'msle' 'Neuter' 'Female (trans)' 'queer'
 'Female (cis)' 'Mail' 'cis male' 'A little about you' 'Malr' 'p' 'femail'
 'Cis Man' 'ostensibly male, unsure what that really means']
```

```
male_str = ["male", "m", "male-ish", "maile", "mal", "male (cis)", "make",
            "male ", "man","msle", "mail", "malr","cis man", "Cis Male",
            "cis male"]
trans_str = ["trans-female", "something kinda male?", "queer/she/they",
             "non-binary","nah", "all", "enby", "fluid", "genderqueer",
             "androgyne", "agender", "male leaning androgynous", "guy (-ish) ^_^",
             "trans woman", "neuter", "female (trans)", "queer",
             "ostensibly male, unsure what that really means"]
female_str = ["cis female", "f", "female", "woman",  "femake", "female ",
              "cis-female/femme", "female (cis)", "femail"]

```

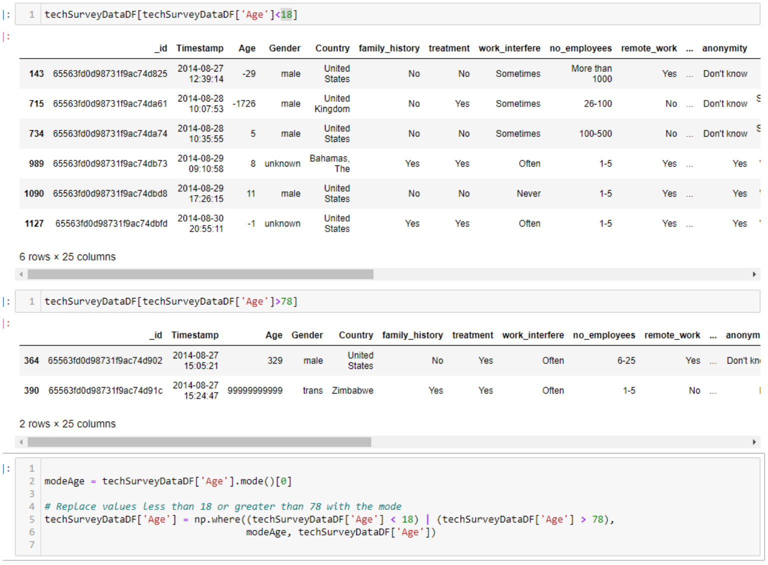
```

for index, row in techSurveyDataDF.iterrows():
    gender_value = str(row['Gender']).lower()

    if gender_value in male_str:
        techSurveyDataDF.at[index, 'Gender'] = 'male'
    elif gender_value in female_str:
        techSurveyDataDF.at[index, 'Gender'] = 'female'
    elif gender_value in trans_str:
        techSurveyDataDF.at[index, 'Gender'] = 'trans'
    else:
        # Handle other cases if needed
        techSurveyDataDF.at[index, 'Gender'] = 'unknown'

```

```
uniqueValuesCount = techSurveyDataDF['Gender'].unique()

print("Number of unique values in the column:", uniqueValuesCount)
```

```
Number of unique values in the column: ['female' 'male' 'trans' 'unknown']
```

Fig: Transformation of Gender column

- In third step, I found out inconsistencies in Age field such as some values were negative and some where way too big which is not possible in real life scenario. Hence, I replaced them with mode value of Age column as below.



```
techSurveyDataDF[techSurveyDataDF['Age']<18]
```

	_id	Timestamp	Age	Gender	Country	family_history	treatment	work_interfere	no_employees	remote_work	...	anonymity
143	65563fd0d98731f9ac74d825	2014-08-27 12:39:14	-29	male	United States	No	No	Sometimes	More than 1000	Yes	...	Don't know
715	65563fd0d98731f9ac74da61	2014-08-28 10:07:53	-1726	male	United Kingdom	No	Yes	Sometimes	26-100	No	...	Don't know
734	65563fd0d98731f9ac74da74	2014-08-28 10:35:55	5	male	United States	No	No	Sometimes	100-500	No	...	Don't know
989	65563fd0d98731f9ac74db73	2014-08-29 09:10:58	8	unknown	Bahamas, The	Yes	Yes	Often	1-5	Yes	...	Yes
1090	65563fd0d98731f9ac74dbd8	2014-08-29 17:26:15	11	male	United States	No	No	Never	1-5	Yes	...	Yes
1127	65563fd0d98731f9ac74dbfd	2014-08-30 20:55:11	-1	unknown	United States	Yes	Yes	Often	1-5	Yes	...	Yes

6 rows × 25 columns

```
techSurveyDataDF[techSurveyDataDF['Age']>78]
```

	_id	Timestamp	Age	Gender	Country	family_history	treatment	work_interfere	no_employees	remote_work	...	anonym
364	65563fd0d98731f9ac74d902	2014-08-27 15:05:21	329	male	United States	No	Yes	Often	6-25	Yes	...	Don't kn
390	65563fd0d98731f9ac74d91c	2014-08-27 15:24:47	99999999999	trans	Zimbabwe	Yes	Yes	Often	1-5	No	...	

2 rows × 25 columns

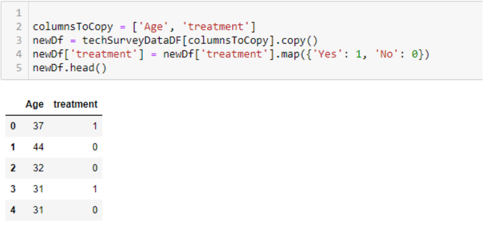
```

modeAge = techSurveyDataDF['Age'].mode()[0]

# Replace values less than 18 or greater than 78 with the mode
techSurveyDataDF['Age'] = np.where((techSurveyDataDF['Age'] < 18) | (techSurveyDataDF['Age'] > 78),
                                   modeAge, techSurveyDataDF['Age'])

```

Fig: Replacing inconsistencies in Age column with mode value of that column

Step 3: AI Fairness and bias mitigation

Lib used: aif360

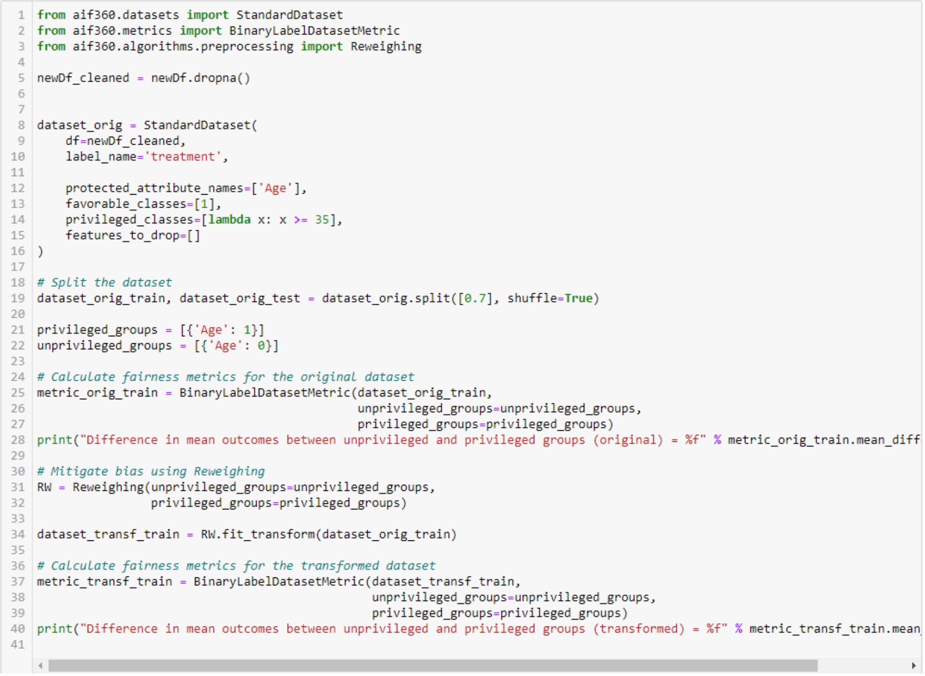
- I performed AI Fairness to check bias and mitigate the bias with Age as protected attribute and treatment column as label name.
- I set the age>35 as privileged group and I split the input dataframe with test-train split of 30-70%.
- I found that this age group >35 is bias towards treatment in the given dataset with mean value as -0.086 that is they are receiving these much favourable results.
- Hence to mitigate this I utilized Reweighing Algorithm from aif360, and score dropped to 0.00, which means there is no bias anymore.



```
columnsToCopy = ['Age', 'treatment']
newDf = techSurveyDataDF[columnsToCopy].copy()
newDf['treatment'] = newDf['treatment'].map({'Yes': 1, 'No': 0})
newDf.head()
```

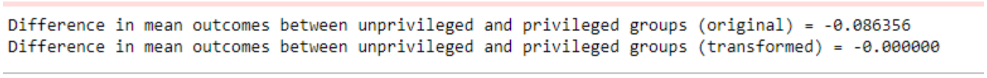
	Age	treatment
0	37	1
1	44	0
2	32	0
3	31	1
4	31	0

Fig: Mapping the Age >35 with 1 and Age<35 with 0



```
from aif360.datasets import StandardDataset
from aif360.metrics import BinaryLabelDatasetMetric
from aif360.algorithms.preprocessing import Reweighing

newDf_cleaned = newDf.dropna()


dataset_orig = StandardDataset(
    df=newDf_cleaned,
    label_name='treatment',

    protected_attribute_names=['Age'],
    favorable_classes=[1],
    privileged_classes=[lambda x: x >= 35],
    features_to_drop=[]
)

# Split the dataset
dataset_orig_train, dataset_orig_test = dataset_orig.split([0.7], shuffle=True)

privileged_groups = [{'Age': 1}]
unprivileged_groups = [{'Age': 0}]

# Calculate fairness metrics for the original dataset
metric_orig_train = BinaryLabelDatasetMetric(dataset_orig_train,
                                             unprivileged_groups=unprivileged_groups,
                                             privileged_groups=privileged_groups)
print("Difference in mean outcomes between unprivileged and privileged groups (original) = %f" % metric_orig_train.mean_diff

# Mitigate bias using Reweighing
RW = Reweighing(unprivileged_groups=unprivileged_groups,
                privileged_groups=privileged_groups)

dataset_transf_train = RW.fit_transform(dataset_orig_train)

# Calculate fairness metrics for the transformed dataset
metric_transf_train = BinaryLabelDatasetMetric(dataset_transf_train,
                                               unprivileged_groups=unprivileged_groups,
                                               privileged_groups=privileged_groups)
print("Difference in mean outcomes between unprivileged and privileged groups (transformed) = %f" % metric_transf_train.mean
```

Fig: Code for bias detection and mitigating the bias



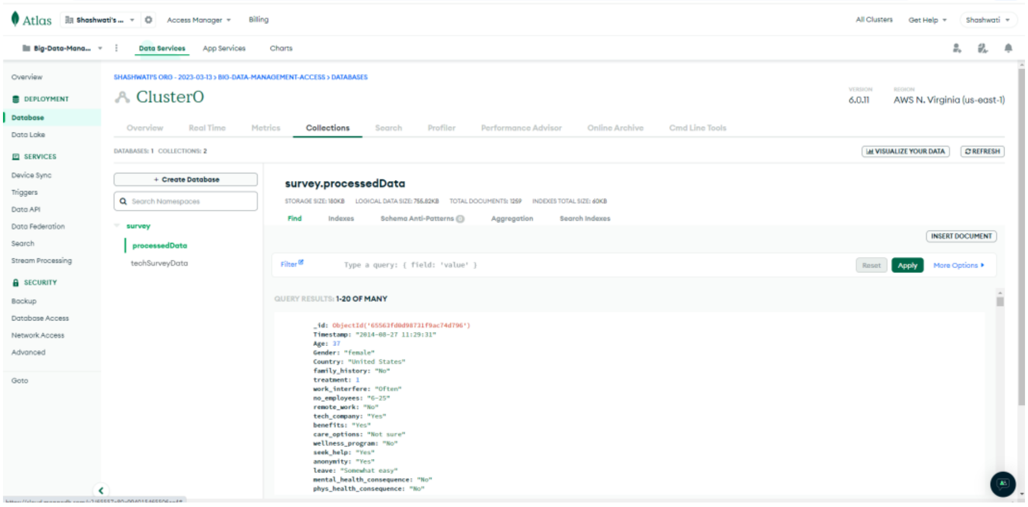
```
Difference in mean outcomes between unprivileged and privileged groups (original) = -0.086356
Difference in mean outcomes between unprivileged and privileged groups (transformed) = -0.000000
```

Fig: Final output before and after mitigating the bias

6. Uploaded processed data to MongoDB:

Approach:

- Followed the same process and cluster as mentioned in Step 2 Data Ingestion Step to upload the processed data to MongoDB Atlas.
- survey.processedData collection has processed data after preprocessing.
- I have accessed this collection in Jupyter notebook via connection string.
- Performed additional analysis which will be discussed in results.



SHASHWATI'S ORG - 2023-03-13 › BIG-DATA-MANAGEMENT-ACCESS › DATABASES

Cluster0

VERSION 6.0.11 REGION AWS N. Virginia (us-east-1)

DATABASES: 1 COLLECTIONS: 2

survey
- processedData
- techSurveyData

survey.processedData

STORAGE SIZE: 180KB LOGICAL DATA SIZE: 755.82KB TOTAL DOCUMENTS: 1259 INDEXES TOTAL SIZE: 60KB

```
Type a query: { field: 'value' }
```

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('65563fd0d98731f9ac74d796')
Timestamp: "2014-08-27 11:29:31"
Age: 37
Gender: "female"
Country: "United States"
family_history: "No"
treatment: 1
work_interfere: "Often"
no_employees: "6-25"
remote_work: "No"
tech_company: "Yes"
benefits: "Yes"
care_options: "Not sure"
wellness_program: "No"
seek_help: "Yes"
anonymity: "Yes"
leave: "Somewhat easy"
mental_health_consequence: "No"
phys_health_consequence: "No"
```

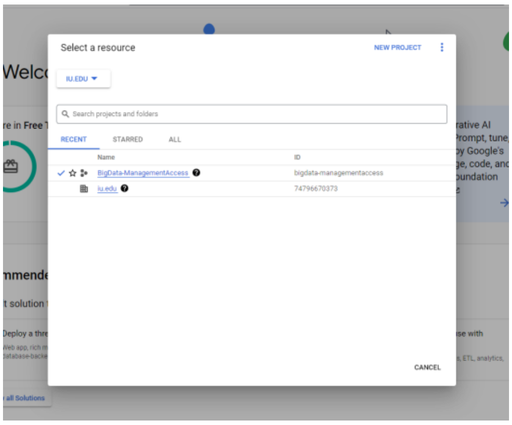
Fig: Processed data uploaded in survey.processedData collection

7. **Uploaded processed data to Google Cloud Storage:**

Platform: Google Cloud Platform (GCP) storage buckets.

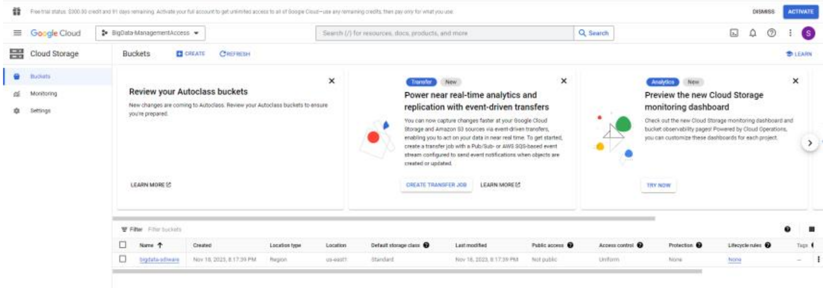
Approach:

- Created storage bucket in GCP and uploaded the processed data in storage buckets.
- Stored the refined data to, leveraging the scalability of the cloud for additional analysis and processing.
- Later fetched this same data in BigQuery for further analysis.



Name	ID
BigData-ManagementAccess	bigdata-managementaccess
iu.edu	74796670373

Fig: Project BigData-ManagementAccess created in GCP



Free trial status: $300.00 credit and 91 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

Review your Autoclass buckets

New changes are coming to Autoclass. Review your Autoclass buckets to ensure you're prepared.

Power near real-time analytics and replication with event-driven transfers

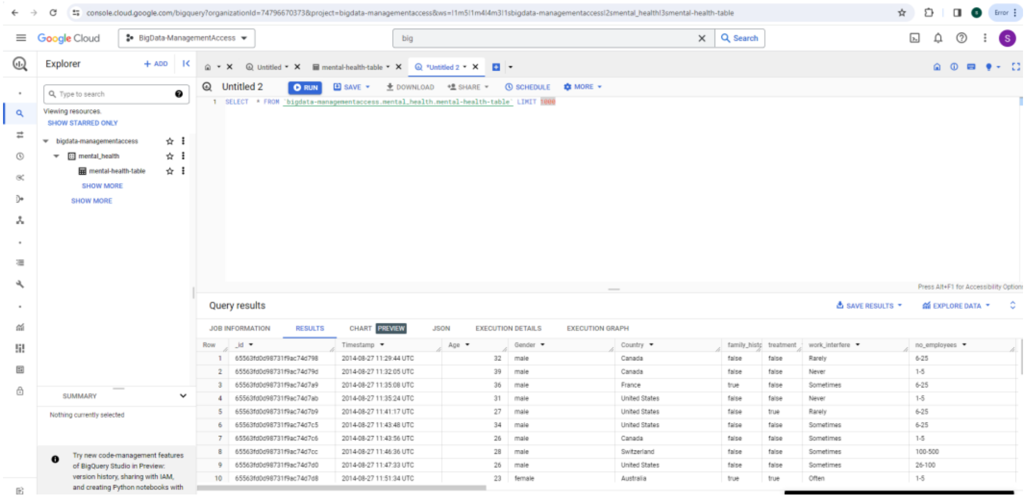
You can now capture changes faster at your Google Cloud Storage and Amazon S3 sources via event-driven transfers, enabling you to act on your data in near real time. To get started, create a transfer job with a Pub/Sub- or AWS SQS-based event stream configured to send event notifications when objects are created or updated.

Preview the new Cloud Storage monitoring dashboard

Check out the new Cloud Storage monitoring dashboard and bucket observability pages! Powered by Cloud Operations, you can customize these dashboards for each project.

Name	Created	Location type	Location	Default storage class	Last modified	Public access	Access control	Protection	Lifecycle rules	Tags
bigdata-sdware	Nov 18, 2023, 8:17:39 PM	Region	us-east1	Standard	Nov 18, 2023, 8:17:39 PM	Not public	Uniform	None	None	–

Fig: Google Cloud Storage bucket created to store processed data



```
SELECT * FROM `bigdata-managementaccess.mental_health.mental-health-table` LIMIT 1000
```

Row	_id	Timestamp	Age	Gender	Country	family_histc	treatment	work_interfere	no_employees
1	65563fd0d98731f9ac74d798	2014-08-27 11:29:44 UTC	32	male	Canada	false	false	Rarely	6-25
2	65563fd0d98731f9ac74d79d	2014-08-27 11:32:05 UTC	39	male	Canada	false	false	Never	1-5
3	65563fd0d98731f9ac74d7a9	2014-08-27 11:35:08 UTC	36	male	France	true	false	Sometimes	6-25
4	65563fd0d98731f9ac74d7ab	2014-08-27 11:35:24 UTC	31	male	United States	false	false	Never	1-5
5	65563fd0d98731f9ac74d7b9	2014-08-27 11:41:17 UTC	27	male	United States	false	true	Rarely	6-25
6	65563fd0d98731f9ac74d7c5	2014-08-27 11:43:48 UTC	34	male	United States	false	false	Sometimes	6-25
7	65563fd0d98731f9ac74d7c6	2014-08-27 11:43:56 UTC	26	male	Canada	false	false	Sometimes	1-5
8	65563fd0d98731f9ac74d7cc	2014-08-27 11:46:36 UTC	28	male	Switzerland	false	false	Sometimes	100-500
9	65563fd0d98731f9ac74d7d0	2014-08-27 11:47:33 UTC	26	male	United States	false	false	Sometimes	26-100
10	65563fd0d98731f9ac74d7d8	2014-08-27 11:51:34 UTC	23	female	Australia	true	true	Often	1-5

Fig: BigQuery platform to fetch the data from Processed Data csv

8. Exploratory Data Analysis (EDA):

Step 1: Exploratory Data Analysis (EDA) via Jupyter Notebook:

Tools Used: Python Jupyter notebook.

Approach:

- Downloaded the processed data from MongoDB and conducted comprehensive visualization and exploratory data analysis.

Step 2: Exploratory Data Analysis (EDA) via MongoDB Atlas Charts:

Approach:

- On the processed data uploaded to MongoDB Atlas after data preprocessing implemented charts via MongoDB Dashboard
- Utilized MongoDB Atlas Charts for additional insights.

Step 3: Exploratory Data Analysis via GCP Big Query:

Tools Used: Google Cloud Platform Big Query

Approach:

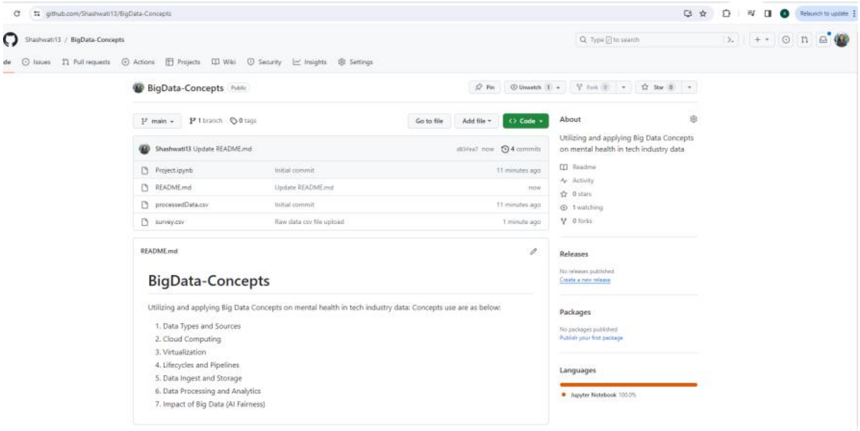
- Linked the storage bucket to big query.
- Implemented few queries for additional analysis in Big Query.

9. Version Control and Collaboration:

Repository: GitHub.

Benefits:

- Ensured version control for the entire project.
- URL: https://github.com/Shashwati13/BigData-Concepts



BigData-Concepts Public

Shashwati13 Update README.md d83fea7 now 4 commits

Project.ipynb	Initial commit	11 minutes ago
README.md	Update README.md	now
processedData.csv	Initial commit	11 minutes ago
survey.csv	Raw data csv file upload	1 minute ago

README.md

BigData-Concepts

Utilizing and applying Big Data Concepts on mental health in tech industry data: Concepts use are as below:

1. Data Types and Sources
2. Cloud Computing
3. Virtualization
4. Lifecycles and Pipelines
5. Data Ingest and Storage
6. Data Processing and Analytics
7. Impact of Big Data (AI Fairness)

About

Utilizing and applying Big Data Concepts on mental health in tech industry data

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

Fig: GitHub Repository for the project

RESULTS:

- Using all the steps taken until now I have found some interesting facts from the data.
- I extracted few hidden trends and analysis after processing the data.
- I have performed analysis and visualization in GCP Big Query, Jupyter Notebook and MongoDB Atlas Dashboard.
- Below are some interesting results:

1. <u>MongoDB Chart Dashboard and Analysis:</u>

Query 1: Age group vs how many benefits:

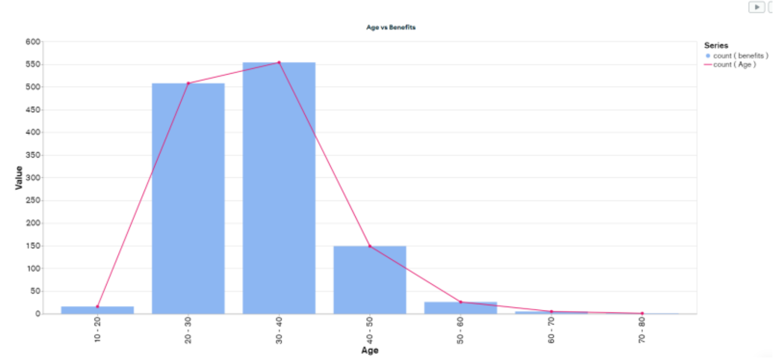


Fig: Age Group vs No of Benefits

- Above chart visualizes Age group vs Benefits.
- This plot tells us which age group has what number of benefits out of total benefits.
- As seen above age group 30-40 has maximum number of benefits.

Query 2: Country wise mental health case count

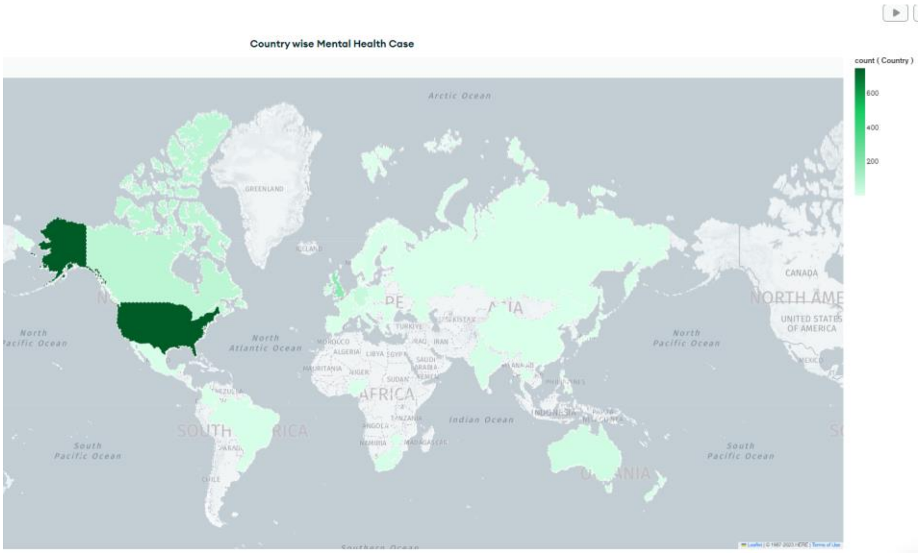


Fig: Country vs Mental Health Case Count

- Above plot shows country wise count of mental health cases.
- As the colour goes towards darker side that means cases count is more there.
- And as seen in above graph United States has darker shade of green than other countries, which means mental health case count in that region is significantly higher than neighbouring countries.

Query 3: Below query shows care_options availability according to employees.

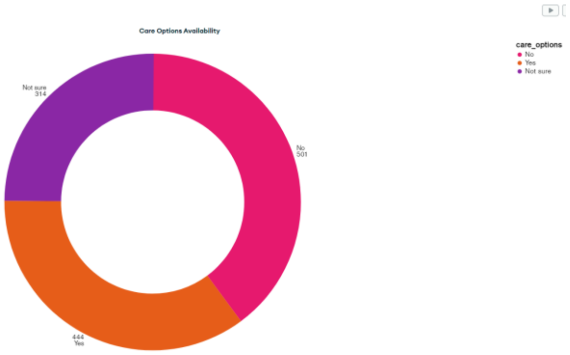


Fig: Above chart shows care options available distribution

- Above pie donut chart shows care options available distribution.
- Care_options column corresponds to whether the tech company where employees work has care options for mental health.
- And as seen in above chart max value is 501 which is for answer "No".
- From this we can infer more companies don't have care options as compared to count of companies who have.

Query 4: Below chart gives distribution of employees belonging to tech company have mental health consequences.

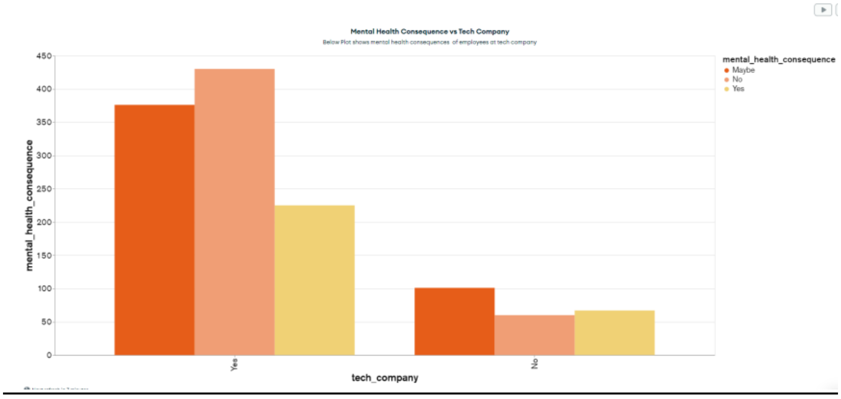


Fig: Tech Company vs mental health consequences distribution

- Above chart represents employee belonging to tech company vs not belonging to tech company mental health consequences count.
- As seen in the chart, most people are from tech companies and the count of mental health consequences is there is maximum for these people than the people who do not belong to tech companies.
- Apart from these we can also deduce for employee belonging to tech companies how many have mental health consequence? How many do not have? and how many are unsure about it?

Query 5: Below chart shows distribution of age group vs if they seek help or not.

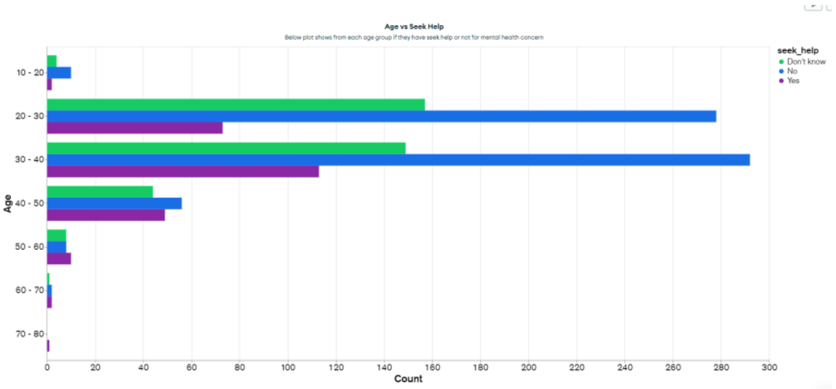


Fig: Age Group vs Seek_Help

- Above chart shows distribution of age group vs count whether they seek help or not?
- From the above chart we can see for age group 30-40 has maximum value of No, meaning, people from this age group do not seek help at all.
- From the above chart we can see for age group 20-30 has maximum value of unsure that means people from this age group are not sure if they have seek help at all.

Query 6: Mean Age Value vs Gender

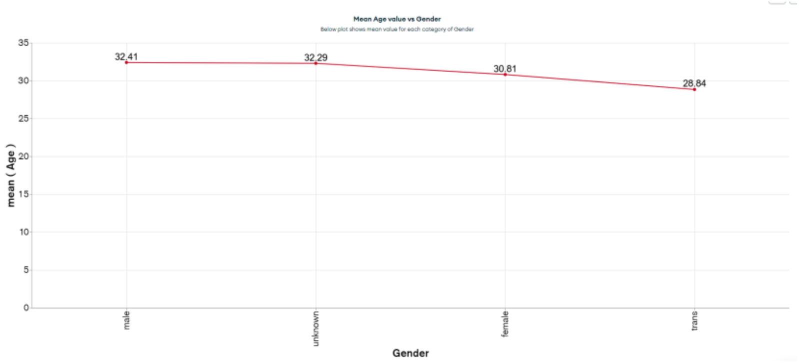
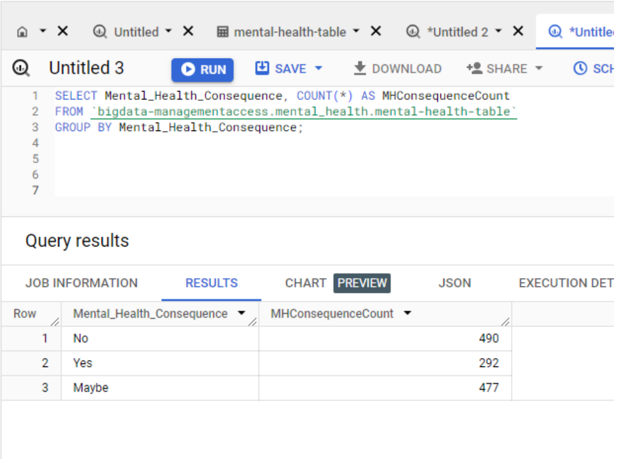


Fig: Mean Age Value vs Gender

- Above chart is Mean Age value vs Gender distribution.
- From that we can see mean age value for male is 32.41, 30.81 for female resp.

2. GCP BIG Query Analysis and Charts:

Query 1: Mental Health Consequences distribution



```
SELECT Mental_Health_Consequence, COUNT(*) AS MHConsequenceCount
FROM `bigdata-managementaccess.mental_health.mental-health-table`
GROUP BY Mental_Health_Consequence;
```

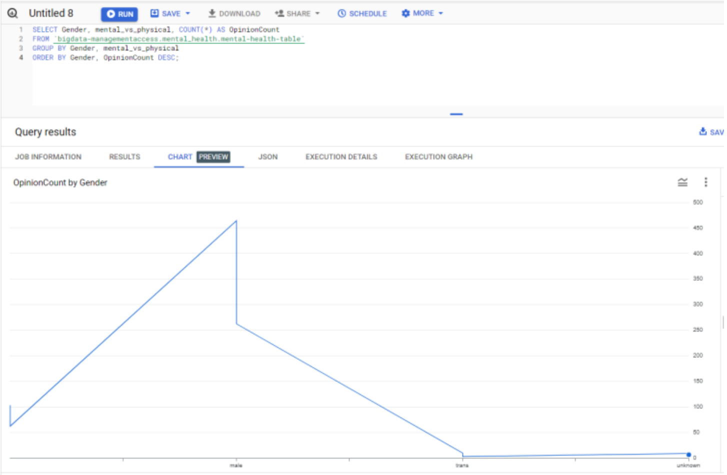
Query results

Row	Mental_Health_Consequence	MHConsequenceCount
1	No	490
2	Yes	292
3	Maybe	477

Fig: Mental Health Consequences distribution

- The query returns a summary of the count of occurrences for each unique value in the Mental_Health_Consequence column, showing how many times each consequence appears in the dataset.
- It helps analyze and understand the distribution of different mental health consequences reported by survey participants, providing insights into the prevalence and patterns of such consequences within the surveyed population.

Query 2: Gender vs Mental Physical Opinion Count Distribution

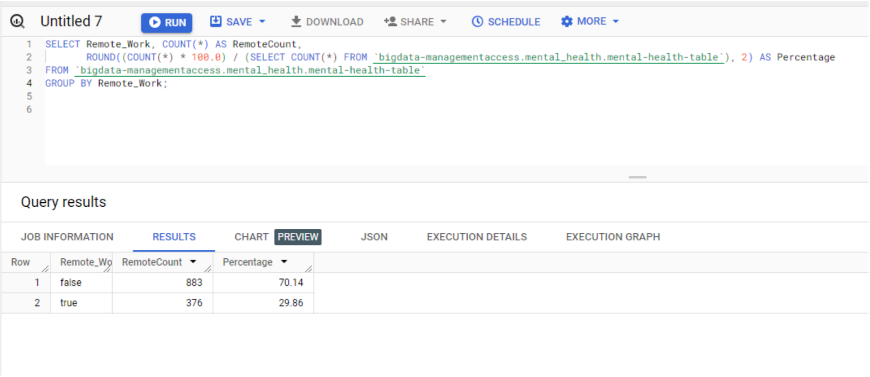


```sql
SELECT Gender, mental_vs_physical, COUNT(*) AS OpinionCount
FROM `bigdata-managementaccess.mental_health.mental-health-table`
GROUP BY Gender, mental_vs_physical
ORDER BY Gender, OpinionCount DESC;
```

Fig: Gender vs Mental Physical Opinion Count

- The query helps to analyse and understand how opinions on mental versus physical health disclosure vary across different genders.
- By grouping and counting occurrences, it provides insights into the distribution of opinions within each gender category.
- The results, ordered by the count of occurrences, highlight the dominant opinions within each gender category as shown in the chart above.

Query 3: Remote Work Percentage Distribution



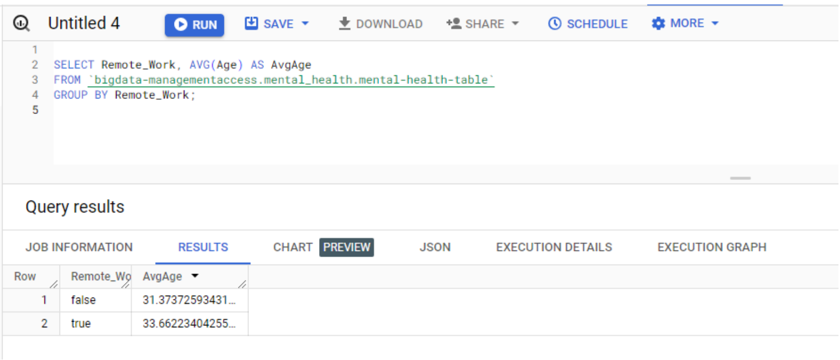
```sql
SELECT Remote_Work, COUNT(*) AS RemoteCount,
       ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM `bigdata-managementaccess.mental_health.mental-health-table`), 2) AS Percentage
FROM `bigdata-managementaccess.mental_health.mental-health-table`
GROUP BY Remote_Work;
```

Row	Remote_Wo	RemoteCount	Percentage
1	false	883	70.14
2	true	376	29.86

Fig: Remote Work Percentage Distribution

- We can use this analysis to understand the prevalence of remote work in the tech industry, which may be relevant for workforce planning, policy development, and adapting to evolving work trends.
- Policymakers and researchers can use the data to gain insights into the broader trends of remote work within the context of mental health in the surveyed population.

Query 4: Remote Work vs Average Age:



```
SELECT Remote_Work, AVG(Age) AS AvgAge
FROM `bigdata-managementaccess.mental_health.mental-health-table`
GROUP BY Remote_Work;
```

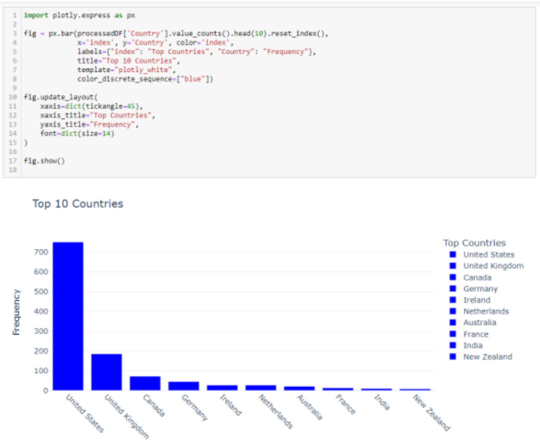
Row	Remote_Wo	AvgAge
1	false	31.37372593431...
2	true	33.66223404255...

Fig: Remote Work vs Average Age

- The results will show the average age of respondents grouped by whether they work remotely or not.
- This analysis helps understand potential age-related patterns or differences in remote work adoption within the surveyed population.

3. Jupyter Notebook Anaylsis:

Query 1: Top 10 Countries from the Mental Health Survey

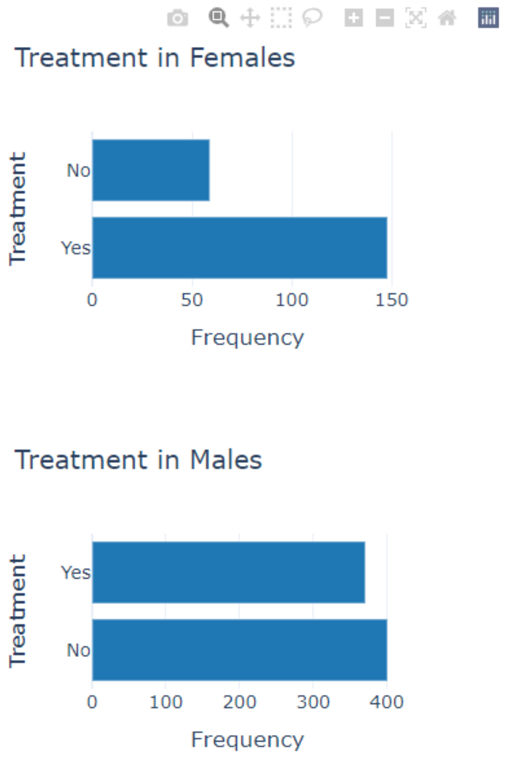


```
import plotly.express as px

fig = px.bar(processedDF['Country'].value_counts().head(10).reset_index(),
             x='index', y='Country', color='index',
             labels={"index": "Top Countries", "Country": "Frequency"},
             title="Top 10 Countries",
             template="plotly_white",
             color_discrete_sequence=["blue"])

fig.update_layout(
    xaxis=dict(tickangle=45),
    xaxis_title="Top Countries",
    yaxis_title="Frequency",
    font=dict(size=14)
)

fig.show()
```

Fig: Top 10 Countries who participated in the survey

- Above chart is country wise distribution of employees who participated in the mental health in tech survey.
- From above chart we can say most employees were from United States and New Zealand had the least.

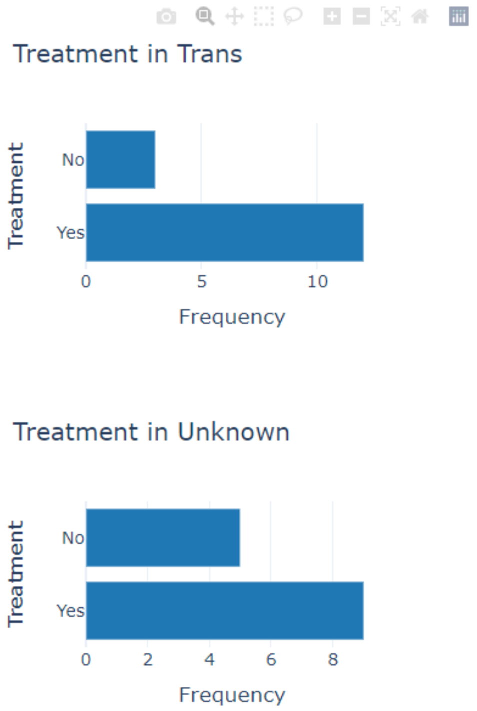
Query 2: Below chart and code does the analysis on number treatments per gender.



Treatment in Females

Treatment in Males

Fig: Number of treatments vs Gender (Male, Female)



Treatment in Trans

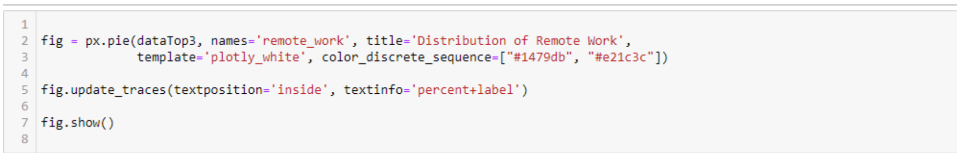
Treatment in Unknown

Fig: Number of treatments vs Gender (Trans, Unkown)

- Above chart and code does the analysis on number of treatments done for each gender.
- From first chart we can see a greater number of females have received treatment.
- From second chart we can see greater number of males have not received treatment.

- Similarly, for other categories of gender we can see treatment distribution.

Query 3: Remote Work Distribution Chart



```

fig = px.pie(dataTop3, names='remote_work', title='Distribution of Remote Work',
             template='plotly_white', color_discrete_sequence=["#1479db", "#e21c3c"])

fig.update_traces(textposition='inside', textinfo='percent+label')

fig.show()

```

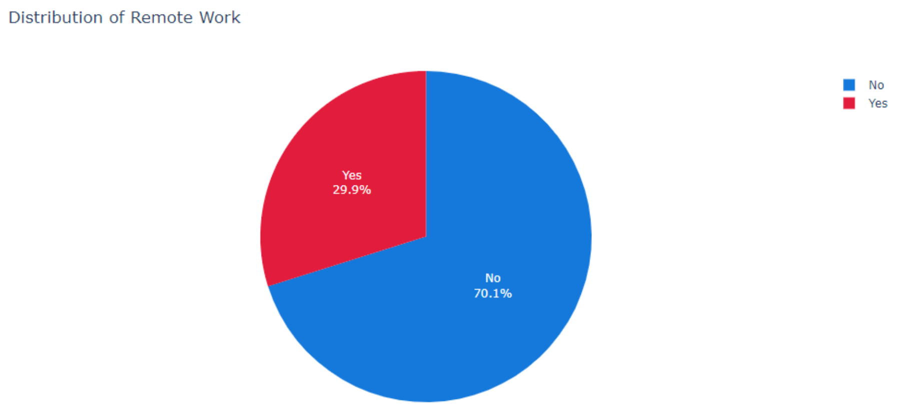
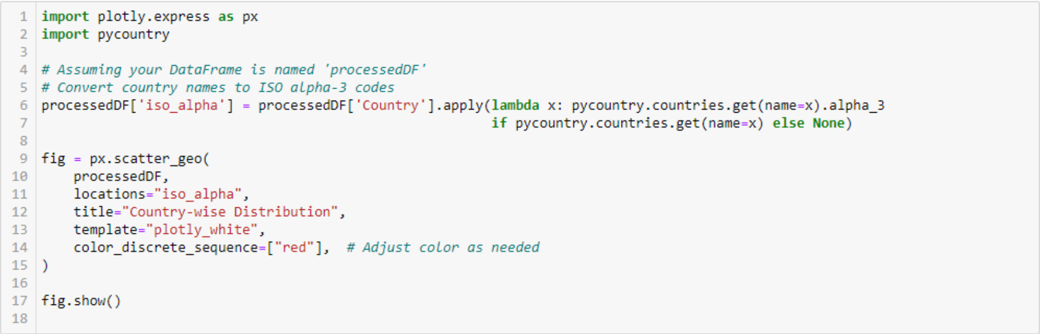


Fig: Remote Work Distribution

- From above chart we can see a greater number of employees does not work remotely than who work remotely. The percentage is 70.1: 29.9 %.

Query 4: Demographic Distribution Chart



```
import plotly.express as px
import pycountry

# Assuming your DataFrame is named 'processedDF'
# Convert country names to ISO alpha-3 codes
processedDF['iso_alpha'] = processedDF['Country'].apply(lambda x: pycountry.countries.get(name=x).alpha_3
                                                        if pycountry.countries.get(name=x) else None)


fig = px.scatter_geo(
    processedDF,
    locations="iso_alpha",
    title="Country-wise Distribution",
    template="plotly_white",
    color_discrete_sequence=["red"],  # Adjust color as needed
)

fig.show()

```

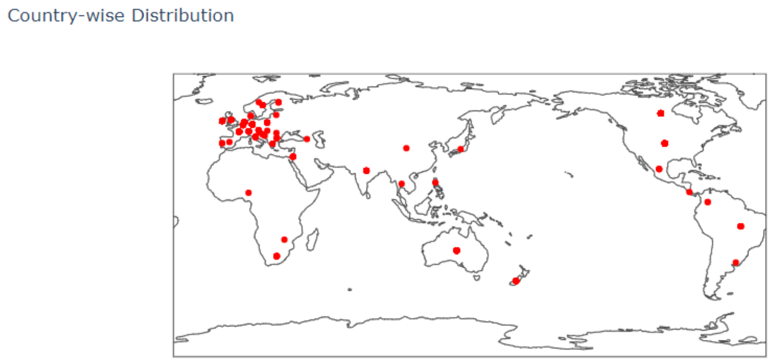


Fig: Country Wise Distribution

- From above map chart, we can infer which demographic location has cases of mental health cases in tech industry.
- As seen majorly US locations are covered in red, that is because the survey is taken by more people from this region than rest.

• DISCUSSION:

As we navigate the findings uncovered in this exploration of mental health within the tech workplace, several noteworthy insights come to the light. Many factors such as remote work, seek help option, taking leaves, supporting teammates, treatment availability is important and has huge impact on mental health in tech industry. To get in-depth analysis and trends I went through several process which were implemented as part of data lifecycle and pipeline. I performed preprocessing on the raw data, removed null values, filled mode values for lesser null values, reduce the values in a column for better understanding and removed inaccurate data. After processing, during exploratory analysis I got many interesting facts but out of all those below are some major ones.

1. Mental health survey for tech industry was taken to understand how tech is affecting the mental health of its employees and there are many factors to it.
2. During the analysis I found variations in attitudes towards mental health across different countries and states. I investigated whether there are noticeable patterns in the frequency of mental health issues based on geographic locations.
3. Major mental health employees belonged to USA almost more than 50% of data of employees came from tech companies in the USA and least came from New Zealand.
4. I also analysed workplace dynamics such as company size, remote work policies, and tech industry affiliation, on mental health attitudes and conditions and how the self-employment status of individuals correlates with their responses regarding mental health. And majorly people who didn't work remotely were facing some mental health consequences the ratio of remote work to in office work was 29.9 % to 70.1%.
5. I performed many ways of analysis of age group with other factors such as seek help, suffering through mental health consequences, and the majorly affected age group seems 20-40 years old with almost 50% contributing towards it. Age group such as 50-60 were least affected hardly 10%.
6. From the data the availability and utilization of mental health benefits, wellness programs, treatments and care options within the tech workplace was analysed and showed positive results towards health mindset of employees. And care options availability was 35% and no care options were 40% and unsure were 25%.

Challenges Faced:

- → The dataset was very inconsistence. While working I had to convert the irrelevant data such as gender column had 49 unique value some of them didn't even make any sense. So, converting this irrelevant data was really challenging.
- → Working with google cloud platform for google storage functionality was challenging.
- → There was potential bias in the dataset, such as underrepresentation of age >35 groups, which affected the findings. And to address this I used a new concept which I learned in the same class AI 360 lib, it turned out challenging to use it.
- → While launching my docker image I was facing issue for the token which is generated, but after rewriting the docker-compose.yaml the issue got resolved.
- → There were many data inconsistencies such as Age column, it had negative values along with large values which are not possible in real life. So, converting this data was a challenge.

CONCLUSION:

In concluding this exploration into the intersection of mental health and the dynamic landscape of the tech workplace, several key insights have emerged. The dataset has provided a nuanced glimpse into the attitudes, challenges, and regional dynamics that shape the mental well-being of individuals within the technology industry. And with this project I got work on various Big Data Concepts and I really enjoyed doing hands-on on cloud technology. I implemented Virtualization, Cloud Computing, Data Ingestion, Data Storage, Data Pipeline and Lifecyle. I got to work with modern technologies such as Docker, Google Cloud Platform and Jetstream right from raw data to processed data. I deduced some valuable insights from the data generated after preprocessing. And I can conclude that age group of 20-40 are majorly affected by mental health consequences.
People who work from office are more likely to have mental health issues are compared to ones who work remotely. I also found out demographically how mental health in tech is affected. All these can be helpful to improve the conditions.

REFERENCES:

- AI Fairness - https://iu.instructure.com/courses/2169303/assignments/15072898
- https://www.youtube.com/watch?v=IHUJ3g01xmI
- https://link.springer.com/article/10.1007/s42979-022-01613-z
- https://cloud.google.com/bigquery/docs/visualize-jupyter
- https://cloud.google.com/bigquery/docs