TITLE: ANALYZING MENTAL HEALTH IN TECH SURVEY USING BIGDATA AND DATA ANALYSIS

NAME: DISNEY SANDHYA GANDIKODA

STUDENT ID: 13135630

MODULE: 7153CEM-BIG DATA ANALYTICS AND DATA VISUALISATION

LECTURER: DR. MARWAN FUAD

Table of Contents

INTRODUCTION 3

DATA SET BRIEF 4

BACKGROUND STUDY 4

PLAN FOR DATA ANALYSIS AND VISUALISATION WITH SOFTWARE 5

INSTALLATION 6

DATA ANALYSIS 7

IMPORTING THE PACKAGES: 7

INITIALIZING THE SPARKCONTEXT: 7

LOADING DATASET: 8

DATASET CLEANING PROCESS: 9

SPLITTING THE DATASET: 16

BUILDING THE MODEL: 17

EVALUATING THE MODEL: 19

DATA VISUALIZATION 21

DATA VISUALIZATION USING TABLEAU. 21

1)PEOPLE RESPONDED TO THE SURVEY 22

2) RESPONSES BASED ON THE GENDER 23

3)RESPONSES BASED ON AGE 24

4) TREATMENT ANALYSIS 24

5) SUPPORT FROM THE COMPANY FOR MENTAL HEALTH 25

6) SHARING WITH SUPERVISOR'S VS CO-WORKERS 25

7) MENTAL HEALTH ISSUES DISCUSSIONS 26

CONCLUSION 26

REFERENCES **28**

APPENDIX **29**

INTRODUCTION

To get insight into the experiences of those working in the technology industry regarding their mental health, a great quantity of data must be gathered, processed, and analysed. This is known as "Analysing mental health in tech survey using big data and data analysis." This survey is designed to gather information on the mental health experiences of IT employees, such as their attitudes towards mental health, the prevalence of mental health problems, and the factors that contribute to poor mental health at work.

Big data refers to highly big and complicated data collections that must be processed and analysed using modern technology and analytical techniques. Big data in the field of mental health may include electronic health records, social media data, sensor data, and other kinds of information that might give insights into individuals' and populations' mental health.Data analysis for mental health includes finding insights and patterns from big and complicated data sets using statistical and computational tools. Analyzing patterns of mental health symptoms, identifying risk factors for mental health problems, or forecasting treatment results are all examples of what this involves.

Furthermore, data analysis may be utilised to provide customised mental health care for workers. Companies may design customised mental health programmes that are suited to individual needs and preferences by analysing data on employee preferences and behaviours. The use of machine learning algorithms to identify indicators of burnout among employees is one example of big data analysis for mental health in the IT business. Data on job demands, workload, and other parameters were utilised by researchers to construct models that may identify whether employees are at danger.

Big data analysis has the potential to revolutionise mental health assistance in the IT sector by giving insights into the causes and risk factors of mental health concerns, as well as generating personalised treatments that can enhance employee well-being and productivity.

DATA SET BRIEF

The dataset used in this coursework is 'survey.csv' which was collected by open sourcing mental illness organization for raising the awareness of mental health in tech industry. This dataset consists of 27 columns such as age, gender, country of residence etc. This survey aims to collect the information those who are suffering from mental illness and to create awareness among working individuals in tech sector. This survey was conducted in online and therefore 1,201 people was responded to this survey and according to their experiences the dataset was created and published in Kaggle. This dataset was taken from the Kaggle and the link for the dataset is: https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey

The datatypes which are included in this dataset are integers, floats, Boolean, Categorical (Text).

BACKGROUND STUDY

It is critical to have a solid understanding of the necessary background information in order to conduct a complete and informed survey on mental health in the IT sector. Here are some main areas of research that may be beneficial:

Prevalence of mental health issues in the general population: Understanding the prevalence of mental health issues in the general population can give a foundation for comparison with the tech industry. Anxiety, depression, bipolar disorder, and other mental health issues should be studied.

Mental health difficulties in the IT industry: Studies have indicated that tech professionals are more likely to suffer from mental health disorders such as burnout, depression, and anxiety.

Studies on the frequency of mental health concerns in the IT industry can help throw light on the unique challenges that tech professionals experience.

Factors that contribute: Long hours, high-pressure workplaces, and an overwork culture are all characteristics that might lead to mental health difficulties in the tech business. Research into these key aspects can aid in identifying areas for improvement.

influence on job performance: Mental health difficulties, such as absenteeism, poor productivity, and job dissatisfaction, can have a substantial influence on job performance. Understanding the impact of mental health concerns on job performance can aid in making the case for better mental health support in the IT sector.

Best practises in mental health care: There are various best practises that may be applied to help tech employees' mental health, such as providing mental health resources and support, fostering work-life balance, and eliminating stigma surrounding mental health concerns. The study of these best practises can give useful insights on how to enhance mental health assistance in the IT industry.

Office culture: A workplace's culture may have a substantial influence on employees' mental health. Workplace culture research in the technology business can assist identify areas where improvements can be done to support improved mental health outcomes.

Mental health treatment choices, like as counselling and medication, might serve to determine the sorts of services that should be made available to IT employees. Understanding the barriers to receiving mental health care can also help identify areas for improvement.

PLAN FOR DATA ANALYSIS AND VISUALISATION WITH SOFTWARE

In this course work,

- **PySpark** is used to analyze the dataset.
- **Classification** is performed in the data analysis.
- **Tableau** is used for the data visualization by importing the dataset into it.

INSTALLATION

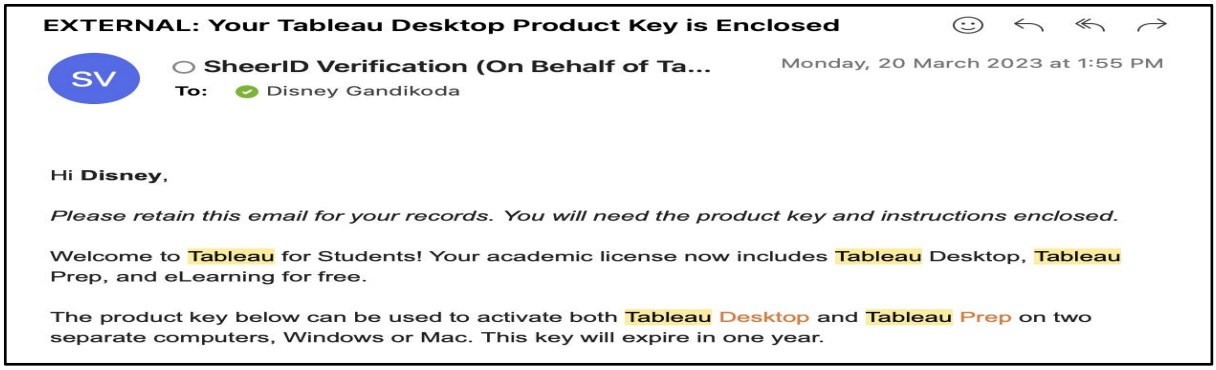
The tools that I have installed into my machine are Virtual environment, PySpark, Java, Pandas, Numpy and Juypterlab which are required for the data analysis and installed Tableau for data visualization. The screenshots of installation are provided below.

```
(base) disneysandhya@Disneys-MacBook-Air ~ % pipen install pyspark jupyter lab
zsh: command not found: pipen
(base) disneysandhya@Disneys-MacBook-Air ~ % pip3 install pipenv
Collecting pipenv
  Downloading pipenv-2023.3.20-py3-none-any.whl (2.9 MB)
     |████████████████████████████████| 2.9 MB 1.6 MB/s
Collecting setuptools>=67.0.0
  Downloading setuptools-67.6.1-py3-none-any.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 7.3 MB/s
Collecting virtualenv>=20.17.1
  Downloading virtualenv-20.21.0-py3-none-any.whl (8.7 MB)
     |████████████████████████████████| 8.7 MB 4.8 MB/s
Requirement already satisfied: certifi in ./opt/anaconda3/lib/python3.9/site-packages (from pipenv) (2021.10.8)
Collecting virtualenv-clone>=0.2.5
  Downloading virtualenv_clone-0.5.7-py3-none-any.whl (6.6 kB)
Requirement already satisfied: filelock<4,>=3.4.1 in ./opt/anaconda3/lib/python3.9/site-packages (from virtualenv>=20.17.1->pipenv) (3.6.0)
Collecting distlib<1,>=0.3.6
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
     |████████████████████████████████| 468 kB 6.2 MB/s
Collecting platformdirs<4,>=2.4
  Downloading platformdirs-3.2.0-py3-none-any.whl (14 kB)
Installing collected packages: platformdirs, distlib, virtualenv-clone, virtualenv, setuptools, pipenv
  Attempting uninstall: setuptools
    Found existing installation: setuptools 61.2.0
    Uninstalling setuptools-61.2.0:
      Successfully uninstalled setuptools-61.2.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
spyder 5.1.5 requires pyqt5<5.13, which is not installed.
spyder 5.1.5 requires pyqtwebengine<5.13, which is not installed.
conda-repo-cli 1.0.4 requires pathlib, which is not installed.
anaconda-project 0.10.2 requires ruamel-yaml, which is not installed.
Successfully installed distlib-0.3.6 pipenv-2023.3.20 platformdirs-3.2.0 setuptools-67.6.1 virtualenv-20.21.0 virtualenv-clone-0.5.7
(base) disneysandhya@Disneys-MacBook-Air ~ % pipenv install pyspark pandas numpy jupyterlab
Warning: the environment variable LANG is not set!
We recommend setting this in ~/.profile (or equivalent) for proper expected behavior.
Creating a virtualenv for this project...
Pipfile: /Users/disneysandhya/Pipfile
Using /Users/disneysandhya/opt/anaconda3/bin/python3 (3.9.12) to create virtualenv...
⠋ Creating virtual environment...created virtual environment CPython3.9.12.final.0-64 in 642ms
  creator CPython3Posix(dest=/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/disneysandhya/Library/Application Support/virtualenv)
    added seed packages: pip==23.0.1, setuptools==67.4.0, wheel==0.38.4
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

✔ Successfully created virtual environment!
Virtualenv location: /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ
```

```
Virtualenv location: /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ
Creating a Pipfile for this project...
Installing pyspark...
Resolving pyspark...
Installing...
Adding pyspark to Pipfile's [packages] ...
✔ Installation Succeeded
Installing pandas...
Resolving pandas...
Installing...
Adding pandas to Pipfile's [packages] ...
✔ Installation Succeeded
Installing numpy...
Resolving numpy...
Installing...
Adding numpy to Pipfile's [packages] ...
✔ Installation Succeeded
Installing jupyterlab...
Resolving jupyterlab...
Installing...
Adding jupyterlab to Pipfile's [packages] ...
✔ Installation Succeeded
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
✔ Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (4a231467861dc734c2ab09e42d00007284f98e43f8b7218fff9f830e9195fa77)!
Installing dependencies from Pipfile.lock (95fa77)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
(base) disneysandhya@Disneys-MacBook-Air ~ %
```

```
(base) disneysandhya@Disneys-MacBook-Air ~ % pipenv run pyspark
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
[Clang 12.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
23/04/14 03:41:56 WARN Utils: Your hostname, Disneys-MacBook-Air.local resolves to a loopback address: 127.0.0.1; using 192.168.1.177 instead (on interface en0)
23/04/14 03:41:56 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/04/14 03:41:57 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
23/04/14 03:41:57 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.4.0
      /_/

Using Python version 3.9.12 (main, Apr  5 2022 01:53:17)
Spark context Web UI available at http://192.168.1.177:4041
Spark context available as 'sc' (master = local[*], app id = local-1681440117591).
SparkSession available as 'spark'.
>>>
```

```
(base) disneysandhya@Disneys-MacBook-Air ~ % java -version
java version "17.0.6" 2023-01-17 LTS
Java(TM) SE Runtime Environment (build 17.0.6+9-LTS-190)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.6+9-LTS-190, mixed mode, sharing)
(base) disneysandhya@Disneys-MacBook-Air ~ %
```



EXTERNAL: Your Tableau Desktop Product Key is Enclosed

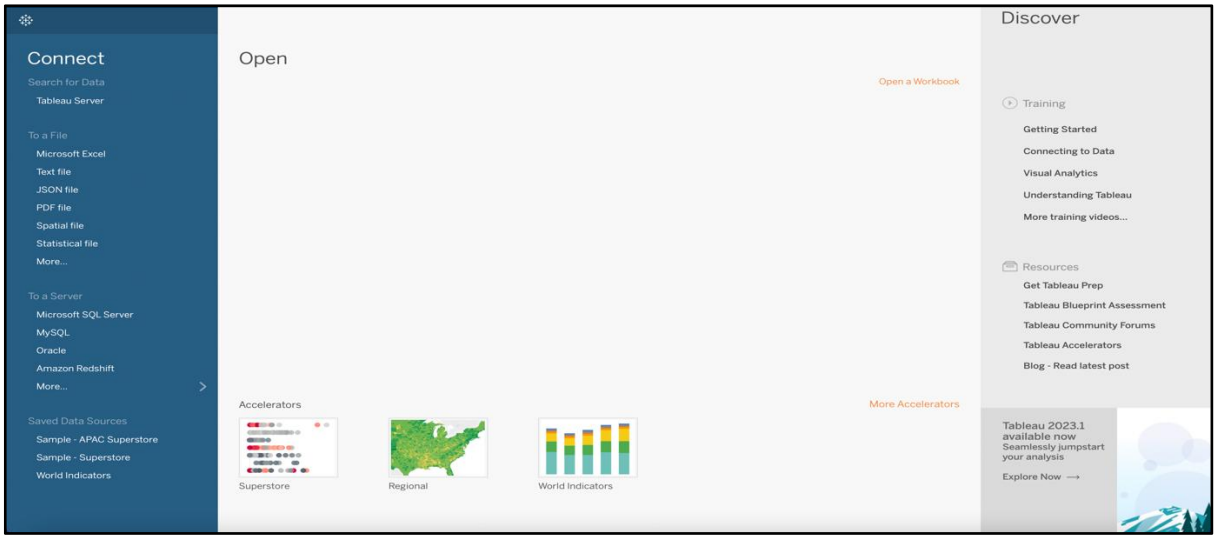
SV SheerID Verification (On Behalf of Ta... Monday, 20 March 2023 at 1:55 PM
To: Disney Gandikoda

Hi **Disney**,

Please retain this email for your records. You will need the product key and instructions enclosed.

Welcome to Tableau for Students! Your academic license now includes Tableau Desktop, Tableau Prep, and eLearning for free.

The product key below can be used to activate both Tableau Desktop and Tableau Prep on two separate computers, Windows or Mac. This key will expire in one year.



DATA ANALYSIS

Importing the packages:

Various packages are imported in the below code such as 'SparkContext', 'pyspark', 'SparkSession', 'Pandas', 'pyspark.ml',' VectorAssembler' and 'StringIndexer'.

```
#PACKAGES
from pyspark import SparkContext
```

Initializing the Sparkcontext:

Using the class 'SparkContext' , spark context is initialized as well as Local[2] is generated as the master URL. Using this, a local Spark cluster with two worker threads is created.

```
sc = SparkContext(master='local[2]')
```

```
23/04/14 03:54:30 WARN Utils: Your hostname, Disneys-MacBook-Air.local resolves to a loopback address: 127.0.0.1; using 192.168.1.177 instead (on interface en0)
```

```
#for spark UI
sc
```

SparkContext

Spark UI

Version	v3.4.0
Master	local[2]
AppName	pyspark-shell

Loading Dataset:

The PySpark SparkSession API is used to load the dataset. The read.csv() function is used to read the CSV file holding the survey data. The method accepts a few parameters: the path to the CSV file, header=True to indicate that the header information is contained in the first row of the CSV file, and inferSchema=True to indicate that PySpark should attempt to infer the data types of each column in the dataset.

The dataframe that results is allocated to the variable df. The dataframe's show() function is used to display the first 20 rows of the dataset.

```
#loading packages
from pyspark.sql import SparkSession
```

```
#spark appname in UI
spark = SparkSession.builder.appName("Disney's ML code").getOrCreate()
```

```
#loading the dataset
df = spark.read.csv("Desktop/survey.csv",header=True,inferSchema=True)
```

```
#shows the dataset
df.show()
```

```
+--------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+------------+----------+--------
----+---------------+----------+----------+-----------------+-----------------------+-----------------------+------------+------------+----------------------
-+---------------------+-----------------+--------------+--------------------+
|     Timestamp|Age|Gender|       Country|state|self_employed|family_history|treatment|work_interfere|  no_employees|remote_work|tech_company|  benefits|care_opt
ions|wellness_program| seek_help| anonymity|              leave|mental_health_consequence|phys_health_consequence|   coworkers|  supervisor|mental_health_intervie
w|phys_health_interview|mental_vs_physical|obs_consequence|            comments|
+--------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+------------+----------+--------
----+---------------+----------+----------+-----------------+-----------------------+-----------------------+------------+------------+----------------------
-+---------------------+-----------------+--------------+--------------------+
|27/08/14 11:29| 37|Female| United States|   IL|           NA|            No|      Yes|         Often|        Jun-25|         No|         Yes|       Yes|    Not
sure|              No|       Yes|       Yes|     Somewhat easy|                     No|                     No|Some of them|        Yes|                     N
o|              Maybe|               Yes|           No|                 NA|
|27/08/14 11:29| 44|     M| United States|   IN|           NA|            No|       No|        Rarely|More than 1000|         No|          No|Don't know|
No|      Don't know|Don't know|Don't know|        Don't know|                    Maybe|                     No|         No|         No|                     No|
No|        Don't know|             No|                 NA|
|27/08/14 11:29| 32|  Male|        Canada|   NA|           NA|            No|       No|        Rarely|        Jun-25|         No|         Yes|        No|
No|              No|        No|Don't know|Somewhat difficult|                     No|                     No|         Yes|         Yes|                     Yes|
Yes|                 No|            No|                 NA|
|27/08/14 11:29| 31|  Male|United Kingdom|   NA|           NA|           Yes|      Yes|         Often|        26-100|         No|         Yes|        No|
Yes|              No|        No|        No|Somewhat difficult|                     Yes|                     Yes|Some of them|         No|                 Maybe
|              Maybe|               No|           Yes|                 NA|
|27/08/14 11:30| 31|  Male| United States|   TX|           NA|            No|       No|         Never|       100-500|        Yes|         Yes|       Yes|
No|      Don't know|Don't know|Don't know|        Don't know|                     No|                     No|Some of them|        Yes|                     Yes|
Yes|        Don't know|             No|                 NA|
|27/08/14 11:31| 33|  Male| United States|   TN|           NA|           Yes|       No|    Sometimes|        Jun-25|         No|         Yes|       Yes|    Not
sure|              No|Don't know|Don't know|        Don't know|                     No|                     No|         Yes|         Yes|                     N
o|              Maybe|        Don't know|            No|                 NA|
|27/08/14 11:31| 35|Female| United States|   MI|           NA|           Yes|      Yes|    Sometimes|        01-May|        Yes|         Yes|        No|
No|              No|        No|        No|Somewhat difficult|                    Maybe|                Maybe|Some of them|         No|                     No|
No|        Don't know|             No|                 NA|
|27/08/11 11:32| 39|     M|        Canada|   NA|           NA|            No|       No|         Never|        01-May|        Yes|         Yes|        No|
Yes|              No|        No|       Yes|        Don't know|                     No|                     No|         No|         No|                     No
|                 No|            No|                 No|
|27/08/14 11:32| 42|Female| United States|   IL|           NA|           Yes|      Yes|    Sometimes|       100-500|         No|         Yes|       Yes|
Yes|              No|        No|        No|    Very difficult|                    Maybe|                     No|         Yes|         Yes|                     No
|              Maybe|               No|            No|                 NA|
|27/08/14 11:32| 23|  Male|        Canada|   NA|           NA|            No|       No|         Never|        26-100|         No|         Yes|Don't know|
No|      Don't know|Don't know|Don't know|        Don't know|                     No|                     No|         Yes|         Yes|                 Maybe|
Maybe|                Yes|            No|                 NA|
|27/08/14 11:32| 31|  Male| United States|   OH|           NA|            No|      Yes|    Sometimes|        Jun-25|        Yes|         Yes|Don't know|
No|              No|        No|Don't know|        Don't know|                     No|                     No|Some of them|        Yes|                     No|
No|        Don't know|             No|                 NA|
```

```
|27/08/14 11:32| 29|  male|      Bulgaria|   NA|           NA|           No|      No|        Never|      100-500|       Yes|        Yes|Don't know|    Not
sure|              No|        No|Don't know|        Don't know|                No|                No|        Yes|        Yes|                Ye
s|                Yes|       Don't know|            No|                NA|
|27/08/14 11:33| 42|female| United States|   CA|           NA|          Yes|      Yes|    Sometimes|        26-100|        No|         No|      Yes|
Yes|              No|        No|Don't know|Somewhat difficult|                Yes|               Yes|        Yes|        Yes|                Maybe
|                Maybe|            No|            Yes|                NA|
|27/08/14 11:33| 36|  Male| United States|   CT|           NA|          Yes|      No|        Never|      500-1000|        No|        Yes|Don't know|    Not
sure|              No|Don't know|Don't know|       Don't know|                No|                No|        Yes|        Yes|                N
o|                No|       Don't know|            No|I'm not on my com...|
|27/08/14 11:33| 27|  Male|        Canada|   NA|           NA|           No|      No|        Never|       Jun-25|        No|        Yes|Don't know|    Not
sure|      Don't know|Don't know|Don't know|     Somewhat easy|                No|                No|Some of them|Some of them|                Mayb
e|                Yes|              Yes|            No|                NA|
|27/08/14 11:34| 29|female| United States|   IL|           NA|          Yes|      Yes|       Rarely|        26-100|        No|        Yes|      Yes|    Not
sure|              No|        No|Don't know|     Somewhat easy|                No|                No|        Yes|Some of them|                Mayb
e|              Maybe|       Don't know|            No|I have chronic lo...|
|27/08/14 11:34| 23|  Male|United Kingdom|   NA|           NA|           No|      Yes|    Sometimes|        26-100|        Yes|        Yes|Don't know|
No|      Don't know|Don't know|Don't know|         Very easy|                Maybe|                No|Some of them|         No|                Maybe|
Maybe|              No|            No|My company does p...|
|27/08/14 11:34| 32|  Male| United States|   TN|           NA|           No|      Yes|    Sometimes|       Jun-25|        No|        Yes|      Yes|
Yes|              No|Don't know|Don't know|       Don't know|                Maybe|                No|Some of them|        Yes|                No
|                No|              No|            No|                NA|
|27/08/14 11:34| 46|  male| United States|   MD|          Yes|          Yes|      No|    Sometimes|       01-May|        Yes|        Yes|      Yes|    Not
sure|              Yes|Don't know|       Yes|         Very easy|                No|                No|        Yes|        Yes|                N
o|                Yes|              Yes|            Yes|                NA|
|27/08/14 11:35| 36|  Male|        France|   NA|          Yes|          Yes|      No|           NA|       Jun-25|        Yes|        Yes|       No|
No|             Yes|        No|       Yes|     Somewhat easy|                No|                No|Some of them|Some of them|                Maybe|
Maybe|        Don't know|            No|                NA|
+---------------+---+------+---------------+-----+-------------+---------------+---------+---------------+---------------+-----------+-----------+----------+--------
----+----------------+----------+----------+-----------------+-----------------------+-----------------+-----------+-----------+----------------------
-+---------------------+-----------------+---------------+-------------------+
only showing top 20 rows
```

Dataset cleaning process:

In the Dataset cleaning process, some columns from the dataframe are chosen, and non-numeric values are transformed to numeric ones using StringIndexer. Using VectorAssembler, the specified columns are also connected into a single feature vector.

After importing the dataset, the following step is to filter the data to only include relevant columns.The select() function is used to eliminate the irrelevant columns from the dataset.

```
df = df.select('Age', 'Gender', 'Country', 'state', 'self_employed', 'family_history', 'treatment', 'work_interfere', 'no_employees',
               'remote_work', 'tech_company')
```

```
df.show(5)
```

```
+---+------+---------------+-----+-------------+---------------+---------+---------------+---------------+-----------+-----------+
|Age|Gender|        Country|state|self_employed|family_history|treatment|work_interfere|  no_employees|remote_work|tech_company|
+---+------+---------------+-----+-------------+---------------+---------+---------------+---------------+-----------+-----------+
| 37|Female| United States|   IL|           NA|             No|      Yes|         Often|        Jun-25|         No|        Yes|
| 44|     M| United States|   IN|           NA|             No|       No|        Rarely|More than 1000|         No|         No|
| 32|  Male|        Canada|   NA|           NA|             No|       No|        Rarely|        Jun-25|         No|        Yes|
| 31|  Male|United Kingdom|   NA|           NA|            Yes|      Yes|         Often|        26-100|         No|        Yes|
| 31|  Male| United States|   TX|           NA|             No|       No|        Never|       100-500|        Yes|        Yes|
+---+------+---------------+-----+-------------+---------------+---------+---------------+---------------+-----------+-----------+
only showing top 5 rows
```

Using the dtypes method, the below code determines the data types of each column. This is done to confirm that the data types match the intended kinds.

```
# Checking for datatypes
# InferSchema=True i.e before
df.dtypes
```

```
[('Timestamp', 'string'),
 ('Age', 'string'),
 ('Gender', 'string'),
 ('Country', 'string'),
 ('state', 'string'),
 ('self_employed', 'string'),
 ('family_history', 'string'),
 ('treatment', 'string'),
 ('work_interfere', 'string'),
 ('no_employees', 'string'),
 ('remote_work', 'string'),
 ('tech_company', 'string'),
 ('benefits', 'string'),
 ('care_options', 'string'),
 ('wellness_program', 'string'),
 ('seek_help', 'string'),
 ('anonymity', 'string'),
 ('leave', 'string'),
 ('mental_health_consequence', 'string'),
 ('phys_health_consequence', 'string'),
 ('coworkers', 'string'),
 ('supervisor', 'string'),
 ('mental_health_interview', 'string'),
 ('phys_health_interview', 'string'),
 ('mental_vs_physical', 'string'),
 ('obs_consequence', 'string'),
 ('comments', 'string')]
```

```
# InferSchema i.e after
df.dtypes
```

```
[('Timestamp', 'string'),
 ('Age', 'bigint'),
 ('Gender', 'string'),
 ('Country', 'string'),
 ('state', 'string'),
 ('self_employed', 'string'),
 ('family_history', 'string'),
 ('treatment', 'string'),
 ('work_interfere', 'string'),
 ('no_employees', 'string'),
 ('remote_work', 'string'),
 ('tech_company', 'string'),
 ('benefits', 'string'),
 ('care_options', 'string'),
 ('wellness_program', 'string'),
 ('seek_help', 'string'),
 ('anonymity', 'string'),
 ('leave', 'string'),
 ('mental_health_consequence', 'string'),
 ('phys_health_consequence', 'string'),
 ('coworkers', 'string'),
 ('supervisor', 'string'),
 ('mental_health_interview', 'string'),
 ('phys_health_interview', 'string'),
 ('mental_vs_physical', 'string'),
 ('obs_consequence', 'string'),
 ('comments', 'string')]
```

The printSchema() function is then used to print the dataset's schema. This is done to ensure that the columns have been properly typed and that the data is formatted as intended.

```
 # Checking for Schema
df.printSchema()
```

```
root
 |-- Timestamp: string (nullable = true)
 |-- Age: long (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- state: string (nullable = true)
 |-- self_employed: string (nullable = true)
 |-- family_history: string (nullable = true)
 |-- treatment: string (nullable = true)
 |-- work_interfere: string (nullable = true)
 |-- no_employees: string (nullable = true)
 |-- remote_work: string (nullable = true)
 |-- tech_company: string (nullable = true)
 |-- benefits: string (nullable = true)
 |-- care_options: string (nullable = true)
 |-- wellness_program: string (nullable = true)
 |-- seek_help: string (nullable = true)
 |-- anonymity: string (nullable = true)
 |-- leave: string (nullable = true)
 |-- mental_health_consequence: string (nullable = true)
 |-- phys_health_consequence: string (nullable = true)
 |-- coworkers: string (nullable = true)
 |-- supervisor: string (nullable = true)
 |-- mental_health_interview: string (nullable = true)
 |-- phys_health_interview: string (nullable = true)
 |-- mental_vs_physical: string (nullable = true)
 |-- obs_consequence: string (nullable = true)
 |-- comments: string (nullable = true)
```

The print(df.describe().show()) function is used to create the dataframe's summary statistics, which include the total, mean, standard deviation, as well as the lowest and highest numbers.

```
# summary
print(df.describe().show())
```

```
+-------+-------------+-------------------+-------------+-------+-----+-------------+-------------+---------+-------------+-------------+-----------+
-----------+----------+-----------+-----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+---
-----------------+-----------------+-----------+-----------+-----------+-----------+-----------+-----------+
|summary|     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|men
tal_health_interview|phys_health_interview|mental_vs_physical|obs_consequence|              comments|
+-------+-------------+-------------------+-------------+-------+-----+-------------+-------------+---------+-------------+-------------+-----------+
-----------+----------+-----------+-----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+---
-----------------+-----------------+-----------+-----------+-----------+-----------+-----------+-----------+
|  count|         1259|               1259|            1259|     1259| 1259|         1259|         1259|     1259|          1259|         1259|       1259|
1259|      1259|       1259|           1259|      1259|      1259|      1259|                       1259|                       1259|     1259|      1259|
1259|                1259|            1259|          1259|               1259|
|   mean|         null| 7.942814831135821E7|             null|     null| null|         null|          null|     null|          null|          null|       null|
null|      null|       null|           null|      null|      null|      null|                       null|                       null|     null|      null|
null|                null|            null|           null|                null|
| stddev|         null|2.8182994429819684E9|             null|     null| null|         null|          null|     null|          null|          null|       null|
null|      null|       null|           null|      null|      null|      null|                       null|                       null|     null|      null|
null|                null|            null|           null|                null|
|    min|01/02/16 23:04|              -1726|A little about you|Australia|   AL|          NA|           No|      No|            NA|        01-May|         No|
No|Don't know|         No|      Don't know|Don't know|Don't know|Don't know|                 Maybe|                 Maybe|      No|       No|
Maybe|               Maybe|       Don't know|            No|                |
|    max| 31/08/14 9:19|         99999999999|             woman| Zimbabwe|   WY|         Yes|          Yes|     Yes|     Sometimes|More than 1000|        Yes|
Yes|       Yes|        Yes|            Yes|      Yes|       Yes| Very easy|                       Yes|                       Yes|     Yes|      Yes|
Yes|                Yes|            Yes|            Yes|you rock for doin...|
+-------+-------------+-------------------+-------------+-------+-----+-------------+-------------+---------+-------------+-------------+-----------+
-----------+----------+-----------+-----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+---
-----------------+-----------------+-----------+-----------+-----------+-----------+-----------+-----------+

None
```

Below code snippet uses the groupBy() function to do a value count analysis on the "Country" column. This is done to determine how frequently each unique value in the "Country" column occurs.

```
# Value Count
df.groupBy('Country').count().show()
```

```
+-------------+-----+
|      Country|count|
+-------------+-----+
|       Russia|    3|
|       Sweden|    7|
|  Philippines|    1|
|    Singapore|    4|
|      Germany|   45|
|       France|   13|
|       Greece|    2|
|      Belgium|    6|
|      Finland|    3|
|United States|  751|
|        India|   10|
|        China|    1|
|      Croatia|    2|
|      Nigeria|    1|
|        Italy|    7|
|       Norway|    1|
|        Spain|    1|
|      Denmark|    2|
|      Ireland|   27|
|     Thailand|    1|
+-------------+-----+
only showing top 20 rows
```

```
df.show(5)
```

```
+-------------+---+------+-------------+-----+-------------+-------------+---------+-------------+-------------+-----------+-----------+---------+--------
----+----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+-----------------------+
-----------------+-----------+-----------+-----------+-----------+-----------+-----------+
|     Timestamp|Age|Gender|       Country|state|self_employed|family_history|treatment|work_interfere|  no_employees|remote_work|tech_company|  benefits|care_opt
ions|wellness_program| seek_help| anonymity|             leave|mental_health_consequence|phys_health_consequence|   coworkers|supervisor|mental_health_interview|
phys_health_interview|mental_vs_physical|obs_consequence|comments|
+-------------+---+------+-------------+-----+-------------+-------------+---------+-------------+-------------+-----------+-----------+---------+--------
----+----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+-----------------------+
-----------------+-----------+-----------+-----------+-----------+-----------+-----------+
|27/08/14 11:29| 37|Female| United States|   IL|           NA|           No|      Yes|        Often|       Jun-25|         No|         Yes|       Yes|    Not
sure|              No|       Yes|       Yes|     Somewhat easy|                       No|                       No|Some of them|       Yes|                       No|
Maybe|               Yes|            No|      NA|
|27/08/14 11:29| 44|     M| United States|   IN|           NA|           No|       No|       Rarely|More than 1000|         No|          No|Don't know|
No|      Don't know|Don't know|Don't know|        Don't know|                    Maybe|                       No|         No|        No|                       No|
No|        Don't know|            No|      NA|
|27/08/14 11:29| 32|  Male|        Canada|   NA|           NA|           No|       No|       Rarely|       Jun-25|         No|         Yes|        No|
No|              No|        No|Don't know|Somewhat difficult|                       No|                       No|         Yes|       Yes|                       Yes|
Yes|                No|            No|      NA|
|27/08/14 11:29| 31|  Male|United Kingdom|   NA|           NA|          Yes|      Yes|        Often|       26-100|         No|         Yes|        No|
Yes|              No|        No|        No|Somewhat difficult|                       Yes|                       Yes|Some of them|        No|                     Maybe|
Maybe|               No|           Yes|      NA|
|27/08/14 11:30| 31|  Male| United States|   TX|           NA|           No|       No|        Never|       100-500|        Yes|         Yes|       Yes|
No|      Don't know|Don't know|Don't know|        Don't know|                       No|                       No|Some of them|       Yes|                       Yes|
Yes|        Don't know|            No|      NA|
+-------------+---+------+-------------+-----+-------------+-------------+---------+-------------+-------------+-----------+-----------+---------+--------
----+----------+----------+----------+----------+-----------------------+-----------------------+---------+---------+-----------------------+
-----------------+-----------+-----------+-----------+-----------+-----------+-----------+
only showing top 5 rows
```

```
import pyspark.ml
```

```
dir(pyspark.ml)
```

```
['Estimator',
 'Model',
 'Pipeline',
 'PipelineModel',
 'PredictionModel',
 'Predictor',
 'TorchDistributor',
 'Transformer',
 'UnaryTransformer',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 'base',
 'classification',
 'clustering',
 'common',
 'evaluation',
 'feature',
 'fpm',
 'image',
 'linalg',
 'param',
 'pipeline',
 'recommendation',
 'regression',
 'stat',
 'torch',
 'tree',
 'tuning',
 'util',
 'wrapper']
```

StringIndexer and VectorAssembler are imported and used in the below process.

```
#loading ml packages
from pyspark.ml.feature import VectorAssembler,StringIndexer
```

```
df.show(4)
```

```
+--------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+-----------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+----------------------+-----------+----------+-----------------------+
---------------------+-----------------+--------------+--------+
|     Timestamp|Age|Gender|       Country|state|self_employed|family_history|treatment|work_interfere|  no_employees|remote_work|tech_company|  benefits|care_opt
ions|wellness_program| seek_help| anonymity|             leave|mental_health_consequence|phys_health_consequence|   coworkers|supervisor|mental_health_interview|
phys_health_interview|mental_vs_physical|obs_consequence|comments|
+--------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+-----------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+----------------------+-----------+----------+-----------------------+
---------------------+-----------------+--------------+--------+
|27/08/14 11:29| 37|Female| United States|   IL|           NA|            No|      Yes|         Often|        Jun-25|        No|        Yes|       Yes|    Not
sure|              No|       Yes|       Yes|     Somewhat easy|                        No|                      No|Some of them|       Yes|                        No|
Maybe|               Yes|             No|      NA|
|27/08/14 11:29| 44|     M| United States|   IN|           NA|            No|       No|        Rarely|More than 1000|        No|          No|Don't know|
No|      Don't know|Don't know|Don't know|        Don't know|                     Maybe|                     No|         No|        No|                        No|
No|        Don't know|             No|      NA|
|27/08/14 11:29| 32|  Male|        Canada|   NA|           NA|            No|       No|        Rarely|        Jun-25|        No|        Yes|       No|
No|              No|        No|Don't know|Somewhat difficult|                        No|                      No|        Yes|       Yes|                       Yes|
Yes|                No|             No|      NA|
|27/08/14 11:29| 31|  Male|United Kingdom|   NA|           NA|           Yes|      Yes|         Often|        26-100|        No|        Yes|       No|
Yes|               No|        No|        No|Somewhat difficult|                        Yes|                      Yes|Some of them|       No|                     Maybe|
Maybe|               No|             Yes|      NA|
+--------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+-----------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+----------------------+-----------+----------+-----------------------+
---------------------+-----------------+--------------+--------+
only showing top 4 rows
```

```
#Unique Values for employment
df.select('self_employed').distinct().show()
```

```
+-------------+
|self_employed|
+-------------+
|           NA|
|           No|
|          Yes|
+-------------+
```

StringIndexer : It is a tool for converting category variables to numerical values. It takes a categorised column and assigns a number index to each distinct category. This is helpful for performing numerical analysis on categorical data. The list of categorical columns to encode is defined as cat_cols in the code. A StringIndexer is constructed for each column in cat_cols and fitted to the dataframe df using indexer.fit(df). The transform technique is then used to

the dataframe through encoder.transform(encoded_df) to generate a new dataframe encoded_df with each category column replaced with its associated index.

```
# Define the list of categorical columns to be encoded
cat_cols = ['Timestamp','Gender', 'Country', 'state', 'self_employed','family_history', 'treatment', 'work_interfere', 'no_employees', 'remote_work', 'tech_compa

# Create a list of StringIndexer stages, one for each categorical column
indexers = [StringIndexer(inputCol=col, outputCol=col+'_idx') for col in cat_cols]

# Fit the indexers to the dataframe
cat_Encoder = [indexer.fit(df) for indexer in indexers]

# Apply the indexers to the dataframe
encoded_df = df
for encoder in cat_Encoder:
    encoded_df = encoder.transform(encoded_df)
```

```
encoded_df.show(5)
```

```
+---------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+------------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+-----------------------+------------+----------+-----------------------+
-----------------+-----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
--+-----------------+----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
-----------------------+---------------------------+-------------+-------------+-----------------------+-----------------------+-----------------------+
--+-------------------+------------+
|     Timestamp|Age|Gender|       Country|state|self_employed|family_history|treatment|work_interfere|  no_employees|remote_work|tech_company|  benefits|care_opt
ions|wellness_program| seek_help| anonymity|             leave|mental_health_consequence|phys_health_consequence|   coworkers|supervisor|mental_health_interview|
phys_health_interview|mental_vs_physical|obs_consequence|comments|Timestamp_idx|Gender_idx|Country_idx|state_idx|self_employed_idx|family_history_idx|treatment_i
dx|work_interfere_idx|no_employees_idx|remote_work_idx|tech_company_idx|benefits_idx|care_options_idx|wellness_program_idx|seek_help_idx|anonymity_idx|leave_idx|
mental_health_consequence_idx|phys_health_consequence_idx|coworkers_idx|supervisor_idx|mental_health_interview_idx|phys_health_interview_idx|mental_vs_physical_i
dx|obs_consequence_idx|comments_idx|
+---------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+------------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+-----------------------+------------+----------+-----------------------+
-----------------+-----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
--+-----------------+----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
-----------------------+---------------------------+-------------+-------------+-----------------------+-----------------------+-----------------------+
--+-------------------+------------+
|27/08/14 11:29| 37|Female| United States|   IL|           NA|            No|      Yes|         Often|        Jun-25|        No|         Yes|       Yes|    Not
sure|              No|       Yes|       Yes|     Somewhat easy|                     No|                    No|Some of them|       Yes|                     No|
Maybe|              Yes|             No|      NA|         18.0|       2.0|        0.0|      7.0|             2.0|             0.0|         0.0|
4.0|            0.0|           0.0|           0.0|        0.0|            2.0|               0.0|           2.0|          1.0|      1.0|
0.0|                     0.0|          0.0|          0.0|                     0.0|                    0.0|                 1.0|               0.0|
0.0|
|27/08/14 11:29| 44|     M| United States|   IN|           NA|            No|       No|        Rarely|More than 1000|        No|         No|Don't know|
No|      Don't know|Don't know|Don't know|        Don't know|                  Maybe|                    No|          No|       No|                     No|
No|        Don't know|             No|      NA|         18.0|       3.0|        0.0|     10.0|             2.0|             0.0|         1.0|              3
.0|            2.0|           0.0|           1.0|        1.0|           0.0|               2.0|           1.0|          0.0|      0.0|
1.0|                     0.0|          1.0|          1.0|                     0.0|                    1.0|                 0.0|               0.0|
0.0|
|27/08/14 11:29| 32|  Male|        Canada|   NA|           NA|            No|       No|        Rarely|        Jun-25|        No|         Yes|        No|
No|              No|        No|Don't know|Somewhat difficult|                     No|                    No|         Yes|       Yes|                     Yes|
Yes|              No|       No|      NA|         18.0|       0.0|        2.0|      0.0|             2.0|             0.0|         0.0|       1.0|
3.0|            0.0|           0.0|           0.0|        2.0|            0.0|               0.0|           0.0|          0.0|      3.0|
0.0|                     0.0|          2.0|          0.0|                     2.0|                    2.0|                 2.0|               0.0|
0.0|
|27/08/14 11:29| 31|  Male|United Kingdom|   NA|           NA|           Yes|      Yes|         Often|       26-100|        No|         Yes|        No|
Yes|              No|       No|        No|Somewhat difficult|                    Yes|                   Yes|Some of them|        No|                  Maybe|
Maybe|              No|            Yes|      NA|         18.0|       0.0|        1.0|      0.0|             2.0|             1.0|         0.0|
4.0|            1.0|           0.0|           0.0|        2.0|            1.0|               0.0|           0.0|          2.0|      3.0|
2.0|                     2.0|          0.0|          1.0|                     1.0|                    0.0|                 2.0|               1.0|
0.0|
|27/08/14 11:30| 31|  Male| United States|   TX|           NA|            No|       No|         Never|      100-500|       Yes|         Yes|       Yes|
No|      Don't know|Don't know|Don't know|        Don't know|                     No|                    No|Some of them|       Yes|                     Yes|
Yes|        Don't know|             No|      NA|        335.0|       0.0|        0.0|      5.0|             2.0|             0.0|         1.0|
2.0|            3.0|           1.0|           0.0|        0.0|            0.0|               2.0|           1.0|          0.0|      0.0|
0.0|                     0.0|          0.0|          0.0|                     2.0|                    2.0|                 0.0|               0.0|
0.0|
+---------------+---+------+--------------+-----+-------------+--------------+---------+--------------+--------------+-----------+------------+----------+--------
----+----------------+----------+----------+-----------------+-------------------------+-----------------------+------------+----------+-----------------------+
-----------------+-----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
--+-----------------+----------------+--------------+--------------+-------------+-------------+-----------+-----------+-----------+-----------+---------+
-----------------------+---------------------------+-------------+-------------+-----------------------+-----------------------+-----------------------+
--+-------------------+------------+
only showing top 5 rows
```

```
print(encoded_df.columns)
```

```
['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', 'super
visor', 'mental_health_interview', 'phys_health_interview', 'mental_vs_physical', 'obs_consequence', 'comments', 'Timestamp_idx', 'Gender_idx', 'Country_idx', 's
tate_idx', 'self_employed_idx', 'family_history_idx', 'treatment_idx', 'work_interfere_idx', 'no_employees_idx', 'remote_work_idx', 'tech_company_idx', 'benefits
_idx', 'care_options_idx', 'wellness_program_idx', 'seek_help_idx', 'anonymity_idx', 'leave_idx', 'mental_health_consequence_idx', 'phys_health_consequence_idx',
'coworkers_idx', 'supervisor_idx', 'mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx', 'obs_consequence_idx', 'comments_idx']
```

```
encoded_df.dtypes
```

```
[('Timestamp', 'string'),
 ('Age', 'bigint'),
 ('Gender', 'string'),
 ('Country', 'string'),
 ('state', 'string'),
 ('self_employed', 'string'),
 ('family_history', 'string'),
 ('treatment', 'string'),
 ('work_interfere', 'string'),
 ('no_employees', 'string'),
 ('remote_work', 'string'),
 ('tech_company', 'string'),
 ('benefits', 'string'),
 ('care_options', 'string'),
 ('wellness_program', 'string'),
 ('seek_help', 'string'),
 ('anonymity', 'string'),
 ('leave', 'string'),
 ('mental_health_consequence', 'string'),
 ('phys_health_consequence', 'string'),
 ('coworkers', 'string'),
 ('supervisor', 'string'),
 ('mental_health_interview', 'string'),
 ('phys_health_interview', 'string'),
 ('mental_vs_physical', 'string'),
 ('obs_consequence', 'string'),
 ('comments', 'string'),
 ('Timestamp_idx', 'double'),
 ('Gender_idx', 'double'),
 ('Country_idx', 'double'),
 ('state_idx', 'double'),
 ('self_employed_idx', 'double'),
 ('family_history_idx', 'double'),
 ('treatment_idx', 'double'),
 ('work_interfere_idx', 'double'),
 ('no_employees_idx', 'double'),
 ('remote_work_idx', 'double'),
 ('tech_company_idx', 'double'),
 ('benefits_idx', 'double'),
 ('care_options_idx', 'double'),
 ('wellness_program_idx', 'double'),
 ('seek_help_idx', 'double'),
 ('anonymity_idx', 'double'),
 ('leave_idx', 'double'),
 ('mental_health_consequence_idx', 'double'),
 ('phys_health_consequence_idx', 'double'),
 ('coworkers_idx', 'double'),
 ('supervisor_idx', 'double'),
 ('mental_health_interview_idx', 'double'),
 ('phys_health_interview_idx', 'double'),
 ('mental_vs_physical_idx', 'double'),
 ('obs_consequence_idx', 'double'),
 ('comments_idx', 'double')]
```

```
df2 = encoded_df.select('Age','Gender_idx', 'Country_idx', 'state_idx', 'self_employed_idx', 'family_history_idx', 'treatment_idx', 'work_interfere_idx',
                        'no_employees_idx', 'remote_work_idx', 'tech_company_idx', 'benefits_idx', 'care_options_idx', 'wellness_program_idx', 'seek_help_idx',
                        'anonymity_idx', 'leave_idx', 'mental_health_consequence_idx', 'phys_health_consequence_idx', 'coworkers_idx', 'supervisor_idx',
                        'mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx', 'obs_consequence_idx', 'comments_idx')
```

```
df2.printSchema()
```

```
root
 |-- Age: long (nullable = true)
 |-- Gender_idx: double (nullable = false)
 |-- Country_idx: double (nullable = false)
 |-- state_idx: double (nullable = false)
 |-- self_employed_idx: double (nullable = false)
 |-- family_history_idx: double (nullable = false)
 |-- treatment_idx: double (nullable = false)
 |-- work_interfere_idx: double (nullable = false)
 |-- no_employees_idx: double (nullable = false)
 |-- remote_work_idx: double (nullable = false)
 |-- tech_company_idx: double (nullable = false)
 |-- benefits_idx: double (nullable = false)
 |-- care_options_idx: double (nullable = false)
 |-- wellness_program_idx: double (nullable = false)
 |-- seek_help_idx: double (nullable = false)
 |-- anonymity_idx: double (nullable = false)
 |-- leave_idx: double (nullable = false)
 |-- mental_health_consequence_idx: double (nullable = false)
 |-- phys_health_consequence_idx: double (nullable = false)
 |-- coworkers_idx: double (nullable = false)
 |-- supervisor_idx: double (nullable = false)
 |-- mental_health_interview_idx: double (nullable = false)
 |-- phys_health_interview_idx: double (nullable = false)
 |-- mental_vs_physical_idx: double (nullable = false)
 |-- obs_consequence_idx: double (nullable = false)
 |-- comments_idx: double (nullable = false)
```

VectorAssembler: It is used to join many columns together to form a single feature vector column. This is beneficial to input data into a machine learning system. The list of needed features is defined in the code as required_features.

```
required_features = ['Age','Gender_idx', 'Country_idx', 'state_idx', 'self_employed_idx', 'family_history_idx', 'treatment_idx', 'work_interfere_idx',
                     'no_employees_idx', 'remote_work_idx', 'tech_company_idx', 'benefits_idx', 'care_options_idx', 'wellness_program_idx', 'seek_help_idx',
                     'anonymity_idx', 'leave_idx', 'mental_health_consequence_idx', 'phys_health_consequence_idx', 'coworkers_idx', 'supervisor_idx',
                     'mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx', 'obs_consequence_idx', 'comments_idx']
```

```
df2 = df2.toPandas().replace('NA',0).astype(float)
```

```
type(df2)
```

```
pandas.core.frame.DataFrame
```

```
type(df)
```

```
pyspark.sql.dataframe.DataFrame
```

```
#converting to pyspark dataframe
new_df = spark.createDataFrame(df2)
```

```
new_df.show()
```

```
+----+----------+-----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------+-------------+---------+---------------------------+-------------------------+-------------+--
-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
| Age|Gender_idx|Country_idx|state_idx|self_employed_idx|family_history_idx|treatment_idx|work_interfere_idx|no_employees_idx|remote_work_idx|tech_company_idx|be
nefits_idx|care_options_idx|wellness_program_idx|seek_help_idx|anonymity_idx|leave_idx|mental_health_consequence_idx|phys_health_consequence_idx|coworkers_idx|su
pervisor_idx|mental_health_interview_idx|phys_health_interview_idx|mental_vs_physical_idx|obs_consequence_idx|comments_idx|
+----+----------+-----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------+-------------+---------+---------------------------+-------------------------+-------------+--
-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
|37.0|       2.0|        0.0|      7.0|             2.0|              0.0|         0.0|              4.0|             0.0|            0.0|             0.0|
0.0|           2.0|                 0.0|          2.0|         1.0|     1.0|                 0.0|                 0.0|         0.0|         0.0|
0.0|                 0.0|                 0.0|                 1.0|                 0.0|         0.0|
|44.0|       3.0|        0.0|     10.0|             2.0|              0.0|         1.0|              3.0|             2.0|            0.0|             1.0|
1.0|           0.0|                 2.0|         1.0|         0.0|     0.0|                 1.0|                 0.0|         1.0|
1.0|                 0.0|                 1.0|                 0.0|                 0.0|         0.0|
|32.0|       0.0|        2.0|      0.0|             2.0|              0.0|         1.0|              3.0|             0.0|            0.0|             0.0|
2.0|           0.0|                 0.0|         0.0|         0.0|     3.0|                 0.0|                 0.0|         2.0|
0.0|                 2.0|                 2.0|                 2.0|                 0.0|         0.0|
|31.0|       0.0|        1.0|      0.0|             2.0|              1.0|         0.0|              4.0|             1.0|            0.0|             0.0|
2.0|           1.0|                 0.0|         0.0|         2.0|     3.0|                 2.0|                 2.0|         0.0|
1.0|                 1.0|                 0.0|                 2.0|                 1.0|         0.0|
|31.0|       0.0|        0.0|      5.0|             2.0|              0.0|         1.0|              2.0|             3.0|            1.0|             0.0|
0.0|           0.0|                 2.0|         1.0|         0.0|     0.0|                 0.0|                 0.0|         0.0|
0.0|                 2.0|                 2.0|                 0.0|                 0.0|         0.0|
|33.0|       0.0|        0.0|      4.0|             2.0|              1.0|         1.0|              0.0|             0.0|            0.0|             0.0|
0.0|           2.0|                 0.0|         1.0|         0.0|     0.0|                 0.0|                 0.0|         2.0|
0.0|                 0.0|                 0.0|                 0.0|                 0.0|         0.0|
|35.0|       2.0|        0.0|     11.0|             2.0|              1.0|         0.0|              0.0|             4.0|            1.0|             0.0|
2.0|           0.0|                 0.0|         0.0|         2.0|     3.0|                 1.0|                 1.0|         0.0|
1.0|                 0.0|                 1.0|                 0.0|                 0.0|         0.0|
|39.0|       3.0|        2.0|      0.0|             2.0|              0.0|         1.0|              2.0|             4.0|            1.0|             0.0|
2.0|           1.0|                 0.0|         0.0|         1.0|     0.0|                 0.0|                 0.0|         1.0|
1.0|                 0.0|                 1.0|                 2.0|                 0.0|         0.0|
|42.0|       2.0|        0.0|      7.0|             2.0|              1.0|         0.0|              0.0|             3.0|            0.0|             0.0|
0.0|           1.0|                 0.0|         0.0|         2.0|     4.0|                 1.0|                 0.0|         2.0|
0.0|                 0.0|                 0.0|                 2.0|                 0.0|         0.0|
|23.0|       0.0|        2.0|      0.0|             2.0|              0.0|         1.0|              2.0|             1.0|            0.0|             0.0|
1.0|           0.0|                 2.0|         1.0|         0.0|     0.0|                 0.0|                 0.0|         2.0|
0.0|                 1.0|                 0.0|                 1.0|                 0.0|         0.0|
|31.0|       0.0|        0.0|      6.0|             2.0|              0.0|         0.0|              0.0|             0.0|            1.0|             0.0|
1.0|           0.0|                 0.0|         0.0|         0.0|     0.0|                 0.0|                 0.0|         0.0|
0.0|                 0.0|                 1.0|                 0.0|                 0.0|         0.0|
|29.0|       1.0|       18.0|      0.0|             2.0|              0.0|         1.0|              2.0|             3.0|            1.0|             0.0|
1.0|           2.0|                 0.0|         0.0|         0.0|     0.0|                 0.0|                 0.0|         2.0|
---|          ---|                 ---|         ---|         ---|    ---|                 ---|                 ---|         ---|
0.0|                 1.0|                 0.0|                 1.0|                 0.0|         0.0|
|31.0|       0.0|        0.0|      6.0|             2.0|              0.0|         0.0|              0.0|             0.0|            1.0|             0.0|
1.0|           0.0|                 0.0|         0.0|         0.0|     0.0|                 0.0|                 0.0|         0.0|
0.0|                 0.0|                 1.0|                 0.0|                 0.0|         0.0|
|29.0|       1.0|       18.0|      0.0|             2.0|              0.0|         1.0|              2.0|             3.0|            1.0|             0.0|
1.0|           2.0|                 0.0|         0.0|         0.0|     0.0|                 0.0|                 0.0|         2.0|
0.0|                 2.0|                 2.0|                 0.0|                 0.0|         0.0|
|42.0|       4.0|        0.0|      1.0|             2.0|              1.0|         0.0|              0.0|             1.0|            0.0|             1.0|
0.0|           1.0|                 0.0|         0.0|         0.0|     3.0|                 2.0|                 2.0|         2.0|
0.0|                 1.0|                 0.0|                 2.0|                 1.0|         0.0|
|36.0|       0.0|        0.0|     29.0|             2.0|              1.0|         1.0|              2.0|             5.0|            0.0|             0.0|
1.0|           2.0|                 0.0|         1.0|         0.0|     0.0|                 0.0|                 0.0|         2.0|
0.0|                 0.0|                 1.0|                 0.0|                 68.0|
|27.0|       0.0|        2.0|      0.0|             2.0|              0.0|         1.0|              2.0|             0.0|            0.0|             0.0|
1.0|           2.0|                 2.0|         1.0|         0.0|     1.0|                 0.0|                 0.0|         0.0|
2.0|                 1.0|                 2.0|                 1.0|                 0.0|         0.0|
|29.0|       4.0|        0.0|      7.0|             2.0|              1.0|         0.0|              3.0|             1.0|            0.0|             0.0|
0.0|           2.0|                 0.0|         0.0|         0.0|     1.0|                 0.0|                 0.0|         2.0|
2.0|                 1.0|                 0.0|                 0.0|                 0.0|         39.0|
|23.0|       0.0|        1.0|      0.0|             2.0|              0.0|         0.0|              0.0|             1.0|            1.0|             0.0|
1.0|           0.0|                 2.0|         1.0|         0.0|     2.0|                 1.0|                 0.0|         0.0|
1.0|                 1.0|                 0.0|                 2.0|                 0.0|         94.0|
|32.0|       0.0|        0.0|      4.0|             2.0|              0.0|         0.0|              0.0|             0.0|            0.0|             0.0|
0.0|           1.0|                 0.0|         1.0|         0.0|     0.0|                 1.0|                 0.0|         0.0|
0.0|                 0.0|                 1.0|                 2.0|                 0.0|         0.0|
|46.0|       1.0|        0.0|     23.0|             1.0|              1.0|         1.0|              0.0|             4.0|            1.0|             0.0|
0.0|           2.0|                 1.0|         1.0|         1.0|     2.0|                 0.0|                 0.0|         2.0|
0.0|                 0.0|                 2.0|                 1.0|                 1.0|         0.0|
|36.0|       0.0|        7.0|      0.0|             1.0|              1.0|         1.0|              1.0|             0.0|            1.0|             0.0|
2.0|           0.0|                 1.0|         0.0|         1.0|     1.0|                 0.0|                 0.0|         0.0|
2.0|                 1.0|                 0.0|                 0.0|                 0.0|         0.0|
+----+----------+-----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------+-------------+---------+---------------------------+-------------------------+-------------+--
-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
only showing top 20 rows
```

```
# Checking For DataTypes & Schema
new_df.printSchema()
```

```
root
 |-- Age: double (nullable = true)
 |-- Gender_idx: double (nullable = true)
 |-- Country_idx: double (nullable = true)
 |-- state_idx: double (nullable = true)
 |-- self_employed_idx: double (nullable = true)
 |-- family_history_idx: double (nullable = true)
 |-- treatment_idx: double (nullable = true)
 |-- work_interfere_idx: double (nullable = true)
 |-- no_employees_idx: double (nullable = true)
 |-- remote_work_idx: double (nullable = true)
 |-- tech_company_idx: double (nullable = true)
 |-- benefits_idx: double (nullable = true)
 |-- care_options_idx: double (nullable = true)
 |-- wellness_program_idx: double (nullable = true)
 |-- seek_help_idx: double (nullable = true)
 |-- anonymity_idx: double (nullable = true)
 |-- leave_idx: double (nullable = true)
 |-- mental_health_consequence_idx: double (nullable = true)
 |-- phys_health_consequence_idx: double (nullable = true)
 |-- coworkers_idx: double (nullable = true)
 |-- supervisor_idx: double (nullable = true)
 |-- mental_health_interview_idx: double (nullable = true)
 |-- phys_health_interview_idx: double (nullable = true)
 |-- mental_vs_physical_idx: double (nullable = true)
 |-- obs_consequence_idx: double (nullable = true)
 |-- comments_idx: double (nullable = true)
```

A VectorAssembler is built with required_features input columns and 'features' output columns. Using vec_assembler.transform(new_df), the transform method is performed to the dataframe new_df to build a new dataframe vec_df with the features integrated into a single column designated 'features'.

```
#VectorAssembler
vec_assembler =VectorAssembler(inputCols=required_features,outputCol='features')
```

```
vec_df = vec_assembler.transform(new_df)
```

```
vec_df.show(5)
```

```
+----+----------+----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------------+-------------+-------------+-------------+-------------+-----------------+-----------------+--
-----------+-------------------------+-----------------------+-----------------------+-----------------+-----------------+-----------------+-----------------+-----------------+
| Age|Gender_idx|Country_idx|state_idx|self_employed_idx|family_history_idx|treatment_idx|work_interfere_idx|no_employees_idx|remote_work_idx|tech_company_idx|be
nefits_idx|care_options_idx|wellness_program_idx|seek_help_idx|anonymity_idx|leave_idx|mental_health_consequence_idx|phys_health_consequence_idx|coworkers_idx|su
pervisor_idx|mental_health_interview_idx|phys_health_interview_idx|mental_vs_physical_idx|obs_consequence_idx|comments_idx|             features|
+----+----------+----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------------+-------------+-------------+-------------+-------------+-----------------+-----------------+--
-----------+-------------------------+-----------------------+-----------------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|37.0|       2.0|        0.0|      7.0|              2.0|               0.0|          0.0|               4.0|             0.0|            0.0|             0.0|
0.0|            2.0|                  0.0|          2.0|          1.0|      1.0|                  0.0|                         0.0|          0.0|
0.0|                      0.0|                  0.0|                  1.0|                 0.0|        0.0|(26,[0,1,3,4,7,12...|
|44.0|       3.0|        0.0|     10.0|              2.0|               0.0|          1.0|               3.0|             2.0|            0.0|             1.0|
1.0|            0.0|                  2.0|          1.0|          0.0|      0.0|                  1.0|                         0.0|          1.0|
1.0|                      0.0|                  1.0|                  0.0|                 0.0|        0.0|(26,[0,1,3,4,6,7,...|
|32.0|       0.0|        2.0|      0.0|              2.0|               0.0|          1.0|               3.0|             0.0|            0.0|             0.0|
2.0|            0.0|                  0.0|          0.0|          0.0|      3.0|                  0.0|                         0.0|          2.0|
0.0|                      2.0|                  2.0|                  2.0|                 0.0|        0.0|(26,[0,2,4,6,7,11...|
|31.0|       0.0|        1.0|      0.0|              2.0|               1.0|          0.0|               4.0|             1.0|            0.0|             0.0|
2.0|            1.0|                  0.0|          0.0|          2.0|      3.0|                  2.0|                         2.0|          0.0|
1.0|                      1.0|                  0.0|                  2.0|                 1.0|        0.0|(26,[0,2,4,5,7,8,...|
|31.0|       0.0|        0.0|      5.0|              2.0|               0.0|          1.0|               2.0|             3.0|            1.0|             0.0|
0.0|            0.0|                  2.0|          1.0|          0.0|      0.0|                  0.0|                         0.0|          0.0|
0.0|                      2.0|                  2.0|                  0.0|                 0.0|        0.0|(26,[0,3,4,6,7,8,...|
+----+----------+----------+---------+----------------+-------------------+-------------+-------------------+-----------------+----------------+-----------------+--
----------+----------------+-------------------+-------------------+-------------+-------------+-------------+-------------+-----------------+-----------------+--
-----------+-------------------------+-----------------------+-----------------------+-----------------+-----------------+-----------------+-----------------+-----------------+
only showing top 5 rows
```

Splitting the dataset:

Using the dataframe's randomSplit technique, the dataset is divided into training and testing sets. The randomSplit method accepts a list of floats representing the proportion of data to split. The list [0.7,0.3] specifies that 70% of the data will be assigned to the training set (train_df) and 30% to the testing set (test_df) in this example.

Train , Test split

```
train_df,test_df = vec_df.randomSplit([0.7,0.3])
```

```
train_df.count()
```

```
847
```

```
train_df.show(4)
```

```
+-----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+---------------+---------------+----------------+-
-----------+----------------+-------------------+-------------+-------------+---------+-------------------------+-------------------------+-------------+-
-------------+-------------------------+-------------------------+-------------------+-----------------+-----------+-------------------+
|  Age|Gender_idx|Country_idx|state_idx|self_employed_idx|family_history_idx|treatment_idx|work_interfere_idx|no_employees_idx|remote_work_idx|tech_company_idx|b
enefits_idx|care_options_idx|wellness_program_idx|seek_help_idx|anonymity_idx|leave_idx|mental_health_consequence_idx|phys_health_consequence_idx|coworkers_idx|s
upervisor_idx|mental_health_interview_idx|phys_health_interview_idx|mental_vs_physical_idx|obs_consequence_idx|comments_idx|             features|
+-----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+---------------+---------------+----------------+-
-----------+----------------+-------------------+-------------+-------------+---------+-------------------------+-------------------------+-------------+-
-------------+-------------------------+-------------------------+-------------------+-----------------+-----------+-------------------+
|-29.0|       0.0|        0.0|     12.0|             0.0|              0.0|          1.0|              1.0|            2.0|            1.0|             1.0|
0.0|            0.0|                2.0|          2.0|          0.0|      0.0|                         0.0|                         0.0|          0.0|
0.0|                        0.0|                       1.0|                 0.0|                0.0|        0.0|(26,[0,3,6,7,8,9,...|
| 18.0|       0.0|        0.0|      2.0|             0.0|              0.0|          1.0|              2.0|            1.0|            0.0|             0.0|
1.0|            2.0|                2.0|          1.0|          0.0|      2.0|                         0.0|                         0.0|          0.0|
2.0|                        0.0|                       0.0|                 1.0|                0.0|        0.0|(26,[0,3,6,7,8,11...|
| 18.0|       0.0|        0.0|     29.0|             0.0|              0.0|          0.0|              3.0|            4.0|            1.0|             0.0|
2.0|            0.0|                0.0|          0.0|          1.0|      2.0|                         0.0|                         0.0|          0.0|
1.0|                        0.0|                       1.0|                 0.0|                0.0|        0.0|(26,[0,3,7,8,9,11...|
| 18.0|       0.0|        5.0|      0.0|             0.0|              0.0|          1.0|              4.0|            0.0|            0.0|             0.0|
2.0|            2.0|                0.0|          0.0|          0.0|      3.0|                         2.0|                         1.0|          1.0|
2.0|                        0.0|                       1.0|                 2.0|                0.0|        0.0|(26,[0,2,6,7,11,1...|
+-----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+---------------+---------------+----------------+-
-----------+----------------+-------------------+-------------+-------------+---------+-------------------------+-------------------------+-------------+-
-------------+-------------------------+-------------------------+-------------------+-----------------+-----------+-------------------+
only showing top 4 rows
```

Building the model:

The models are constructed using pyspark.ml.classification. To train the model on the training set, the fit approach is utilised. The characteristics of Col parameter is set to 'features,' that refers to the column name holding the input features, and the label is set to 'features'.The Col argument is set to 'Country_idx,' which is the name of the column holding the target variable.

```
from pyspark.ml.classification import LogisticRegression,DecisionTreeClassifier
```

```
# Logistic regression Model
lr = LogisticRegression(featuresCol='features',labelCol='Country_idx')
```

```
lr_model = lr.fit(train_df)
```

```
y_pred = lr_model.transform(test_df)
```

```
y_pred.show()
```

```
+----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+----------------+---------------+----------------+--
----------+----------------+-------------------+-----------+-------------+-----------------+-----------------------+-----------------------+-----------+--
-----------+-------------------------+-----------------------+-----------------+-----------------+-----------+-----------------+-----------------+--
---+--------------------+----------+
| Age|Gender_idx|Country_idx|state_idx|self_employed_idx|family_history_idx|treatment_idx|work_interfere_idx|no_employees_idx|remote_work_idx|tech_company_idx|be
nefits_idx|care_options_idx|wellness_program_idx|seek_help_idx|anonymity_idx|leave_idx|mental_health_consequence_idx|phys_health_consequence_idx|coworkers_idx|su
pervisor_idx|mental_health_interview_idx|phys_health_interview_idx|mental_vs_physical_idx|obs_consequence_idx|comments_idx|             features|       rawPredict
ion|         probability|prediction|
+----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+----------------+---------------+----------------+--
----------+----------------+-------------------+-----------+-------------+-----------------+-----------------------+-----------------------+-----------+--
-----------+-------------------------+-----------------------+-----------------+-----------------+-----------+-----------------+-----------------+--
---+--------------------+----------+
|18.0|       1.0|        0.0|      5.0|             0.0|             0.0|         0.0|             0.0|            0.0|            0.0|            0.0|
1.0|            0.0|                0.0|          1.0|          0.0|      0.0|                        2.0|                    0.0|          1.0|
1.0|                0.0|                  0.0|                  0.0|                0.0|        0.0|(26,[0,1,3,11,14,...|[1575.09610769711...|[1.0,
0.0,0.0,0.0,...|       0.0|
|18.0|       1.0|       21.0|      0.0|             0.0|             0.0|         1.0|             1.0|            5.0|            1.0|            1.0|
1.0|            0.0|                0.0|          1.0|          0.0|      2.0|                        2.0|                    0.0|          1.0|
1.0|                0.0|                  0.0|                  0.0|                0.0|        0.0|(26,[0,1,2,6,7,8,...|[-5710.5897371804...|[0.0,
0.0,0.0,0.0,...|      17.0|
|18.0|       2.0|        1.0|      0.0|             0.0|             1.0|         0.0|             0.0|            4.0|            1.0|            0.0|
1.0|            2.0|                2.0|          1.0|          0.0|      0.0|                        1.0|                    1.0|          0.0|
2.0|                1.0|                  0.0|                  0.0|                0.0|       31.0|(26,[0,1,2,5,8,9,...|[918.784730203314...|[7.19
036833613034...|       1.0|
|19.0|       0.0|        2.0|      0.0|             0.0|             1.0|         0.0|             0.0|            4.0|            0.0|            0.0|
2.0|            0.0|                0.0|          0.0|          0.0|      0.0|                        1.0|                    1.0|          0.0|
1.0|                0.0|                  0.0|                  0.0|                0.0|        0.0|(26,[0,2,5,8,11,1...|[407.358738562259...|[7.58
470016351249...|       4.0|
|19.0|       1.0|       43.0|      0.0|             0.0|             0.0|         0.0|             0.0|            0.0|            0.0|            0.0|
1.0|            0.0|                0.0|          0.0|          0.0|      1.0|                        2.0|                    2.0|          0.0|
1.0|                0.0|                  1.0|                  0.0|                0.0|        0.0|(26,[0,1,2,11,16,...|[-12965.233297791...|[0.0,
0.0,0.0,0.0,...|      37.0|
|21.0|       0.0|        1.0|      0.0|             0.0|             0.0|         0.0|             0.0|            0.0|            0.0|            1.0|
1.0|            2.0|                0.0|          0.0|          2.0|      2.0|                        2.0|                    0.0|          0.0|
1.0|                0.0|                  0.0|                  2.0|                0.0|        0.0|(26,[0,2,10,11,12...|[660.093717746780...|[1.49
528511671710...|       1.0|
|21.0|       0.0|        1.0|      0.0|             1.0|             0.0|         1.0|             4.0|            4.0|            1.0|            1.0|
2.0|            1.0|                0.0|          0.0|          1.0|      4.0|                        2.0|                    0.0|          0.0|
0.0|                0.0|                  0.0|                  2.0|                0.0|        0.0|(26,[0,2,4,6,7,8,...|[1021.93498930451...|[9.53
790905190843...|       1.0|
|21.0|       0.0|        2.0|      0.0|             0.0|             0.0|         0.0|             0.0|            0.0|            1.0|            0.0|
2.0|            1.0|                0.0|          0.0|          0.0|      3.0|                        0.0|                    0.0|          0.0|
2.0|                1.0|                  0.0|                  1.0|                0.0|        0.0|(26,[0,2,9,11,12,...|[425.527366378040...|[2.30
415542165067...|       3.0|
|21.0|       3.0|        0.0|      1.0|             0.0|             1.0|         1.0|             1.0|            2.0|            0.0|            0.0|
0.0|            1.0|                1.0|          2.0|          1.0|      0.0|                        0.0|                    0.0|          0.0|
0.0|                0.0|                  0.0|                  1.0|                0.0|        0.0|(26,[0,1,3,5,6,7,...|[981.958049114167...|[1.0,
4.4125635370...|       0.0|
|21.0|       5.0|        0.0|      1.0|             0.0|             1.0|         0.0|             3.0|            1.0|            0.0|            0.0|
1.0|            2.0|                2.0|          0.0|          0.0|      1.0|                        2.0|                    0.0|          0.0|
1.0|                0.0|                  2.0|                  0.0|                0.0|        0.0|(26,[0,1,3,5,7,8,...|[1302.28798540120...|[1.0,
6.7779627162...|       0.0|
|22.0|       0.0|        0.0|      1.0|             0.0|             0.0|         1.0|             1.0|            2.0|            0.0|            0.0|
0.0|            2.0|                1.0|          1.0|          1.0|      0.0|                        0.0|                    0.0|          0.0|
0.0|                0.0|                  0.0|                  1.0|                0.0|        0.0|(26,[0,3,6,7,8,12...|[1012.81575361205...|[1.0,
1.1107380466...|       0.0|
|22.0|       0.0|        5.0|      0.0|             0.0|             1.0|         0.0|             0.0|            0.0|            1.0|            0.0|
2.0|            2.0|                0.0|          1.0|          0.0|      1.0|                        0.0|                    0.0|          0.0|
0.0|                1.0|                  2.0|                  1.0|                0.0|        0.0|(26,[0,2,5,9,11,1...|[-834.44140634330...|[0.0,
0.0,1.425246...|      11.0|
|22.0|       0.0|        6.0|      0.0|             1.0|             1.0|         0.0|             0.0|            0.0|            1.0|            0.0|
1.0|            2.0|                1.0|          2.0|          0.0|      1.0|                        1.0|                    0.0|          0.0|
0.0|                0.0|                  2.0|                  2.0|                1.0|        0.0|(26,[0,2,4,5,9,11...|[-1068.0695741382...|[0.0,
0.0,1.179607...|      16.0|
|22.0|       3.0|        5.0|      0.0|             1.0|             0.0|         1.0|             2.0|            4.0|            0.0|            0.0|
1.0|            2.0|                0.0|          0.0|          0.0|      0.0|                        0.0|                    1.0|          0.0|
0.0|                1.0|                  2.0|                  2.0|                1.0|        0.0|(26,[0,1,2,4,6,7,...|[-269.04014388898...|[0.0,
2.6055002429...|       3.0|
|22.0|       3.0|        8.0|      0.0|             0.0|             0.0|         0.0|             2.0|            2.0|            0.0|            0.0|
1.0|            2.0|                0.0|          0.0|          0.0|      4.0|                        2.0|                    1.0|          1.0|
1.0|                1.0|                  0.0|                  2.0|                0.0|        0.0|(26,[0,1,2,7,8,11...|[-1354.6162738124...|[0.0,
0.0,3.722029...|       8.0|
|22.0|       5.0|        0.0|      2.0|             0.0|             0.0|         1.0|             1.0|            2.0|            0.0|            0.0|
0.0|            0.0|                1.0|          2.0|          1.0|      1.0|                        0.0|                    0.0|          1.0|
2.0|                0.0|                  1.0|                  1.0|                0.0|        0.0|(26,[0,1,3,6,7,8,...|[1340.30728451767...|[1.0,
1.5280379290...|       0.0|
|23.0|       0.0|        0.0|      9.0|             0.0|             1.0|         1.0|             0.0|            3.0|            0.0|            0.0|
1.0|            0.0|                0.0|          1.0|          0.0|      0.0|                        1.0|                    0.0|          0.0|
1.0|                0.0|                  0.0|                  2.0|                0.0|        0.0|(26,[0,3,5,6,8,11...|[2185.12148045137...|[1.0,
0.0,0.0,0.0,...|       0.0|
|23.0|       0.0|        0.0|     44.0|             0.0|             0.0|         1.0|             1.0|            1.0|            0.0|            0.0|
1.0|            0.0|                2.0|          0.0|          0.0|      0.0|                        0.0|                    0.0|          2.0|
0.0|                0.0|                  0.0|                  0.0|                0.0|        0.0|(26,[0,3,6,7,8,11...|[7308.53820853885...|[1.0,
0.0,0.0,0.0,...|       0.0|
|23.0|       0.0|        1.0|      0.0|             0.0|             0.0|         1.0|             2.0|            1.0|            0.0|            0.0|
0.0|            2.0|                1.0|          1.0|          0.0|      2.0|                        0.0|                    0.0|          0.0|
0.0|                1.0|                  0.0|                  1.0|                0.0|        0.0|(26,[0,2,6,7,8,12...|[683.228463880841...|[3.05
533784701213...|       1.0|
|23.0|       0.0|        1.0|      0.0|             2.0|             0.0|         0.0|             0.0|            1.0|            1.0|            0.0|
1.0|            0.0|                2.0|          1.0|          0.0|      2.0|                        1.0|                    0.0|          0.0|
1.0|                1.0|                  0.0|                  2.0|                0.0|       94.0|(26,[0,2,4,8,9,11...|[1163.16738443225...|[1.28
749628062523...|       2.0|
+----+----------+-----------+---------+----------------+-----------------+-------------+-----------------+----------------+---------------+----------------+--
----------+----------------+-------------------+-----------+-------------+-----------------+-----------------------+-----------------------+-----------+--
-----------+-------------------------+-----------------------+-----------------+-----------------+-----------+-----------------+-----------------+--
---+--------------------+----------+
only showing top 20 rows
```

The transform technique is used to forecast the target variable for the test_df data once the model has been trained. The predicted values are saved in y_pred, a dataframe that contains the predicted target variable values as well as other columns such as rawPrediction,

probability, and prediction. The show method is then used to extract only the relevant columns (Country_idx, rawPrediction, probability, and prediction) and display them.

```
print(y_pred.columns)
```

```
['Age', 'Gender_idx', 'Country_idx', 'state_idx', 'self_employed_idx', 'family_history_idx', 'treatment_idx', 'work_interfere_idx', 'no_employees_idx', 'remote_w
ork_idx', 'tech_company_idx', 'benefits_idx', 'care_options_idx', 'wellness_program_idx', 'seek_help_idx', 'anonymity_idx', 'leave_idx', 'mental_health_consequen
ce_idx', 'phys_health_consequence_idx', 'coworkers_idx', 'supervisor_idx', 'mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx',
'obs_consequence_idx', 'comments_idx', 'features', 'rawPrediction', 'probability', 'prediction']
```

```
y_pred.select('Country_idx','rawPrediction', 'probability', 'prediction').show()
```

```
+-----------+--------------------+--------------------+----------+
|Country_idx|       rawPrediction|         probability|prediction|
+-----------+--------------------+--------------------+----------+
|        0.0|[1575.09610769711...|[1.0,0.0,0.0,0.0,...|       0.0|
|       21.0|[-5710.5897371804...|[0.0,0.0,0.0,0.0,...|      17.0|
|        1.0|[918.784730203314...|[7.19036833613034...|       1.0|
|        2.0|[407.358738562259...|[7.58470016351249...|       4.0|
|       43.0|[-12965.233297791...|[0.0,0.0,0.0,0.0,...|      37.0|
|        1.0|[660.093717746780...|[1.49528511671710...|       1.0|
|        1.0|[1021.93498930451...|[9.53790905190843...|       1.0|
|        2.0|[425.527366378040...|[2.30415542165067...|       3.0|
|        0.0|[981.958049114167...|[1.0,4.4125635370...|       0.0|
|        0.0|[1302.28798540120...|[1.0,6.7779627162...|       0.0|
|        0.0|[1012.81575361205...|[1.0,1.1107380466...|       0.0|
|        5.0|[-834.44140634330...|[0.0,0.0,1.425246...|      11.0|
|        6.0|[-1068.0695741382...|[0.0,0.0,1.179607...|      16.0|
|        5.0|[-269.04014388898...|[0.0,2.6055002429...|       3.0|
|        8.0|[-1354.6162738124...|[0.0,0.0,3.722029...|       8.0|
|        0.0|[1340.30728451767...|[1.0,1.5280379290...|       0.0|
|        0.0|[2185.12148045137...|[1.0,0.0,0.0,0.0,...|       0.0|
|        0.0|[7308.53820853885...|[1.0,0.0,0.0,0.0,...|       0.0|
|        1.0|[683.228463880841...|[3.05533784701213...|       1.0|
|        1.0|[1163.16738443225...|[1.28749628062523...|       2.0|
+-----------+--------------------+--------------------+----------+
only showing top 20 rows
```

Evaluating the model:

Following the prediction of the labels for the test data, the model's performance is evaluated. This code's assessment measures include accuracy, precision, recall, and F1 score.

MulticlassClassificationEvaluator: This evaluator is used to calculate the anticipated label correctness. It accepts as input the label column and the metric name and returns the accuracy value.

MulticlassMetrics: This class is employed in the computation of other evaluation metrics such as precision, recall, and F1 score. It uses RDD (Resilient Distributed Dataset) to store the expected and real labels and gives ways for calculating metrics.

The MultiClassificationEvaluator is used in the following code to calculate the correctness of the predicted labels. The evaluate method accepts as input the anticipated data and returns the accuracy score.

The accuracy, recall, and F1 score of the predicted labels are then calculated using MulticlassMetrics. These metrics are computed using MulticlassMetrics' precision, recall, and fMeasure algorithms.

The precision, recall, and F1 score for the label with index 1 are computed. These measures show how effectively the model performs in detecting the positive class. The accuracy of the model reveals how well it performs overall in predicting the labels for the test data.

Model Evaluation

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

```
#Checking For Accuracy
multi_evaluator = MulticlassClassificationEvaluator(labelCol='Country_idx',metricName='accuracy')
```

```
multi_evaluator.evaluate(y_pred)
```

```
0.808252427184466
```

```
# Precision,F1 Score,Recall : Classification Report
```

```
from pyspark.mllib.evaluation import MulticlassMetrics
```

```
lr_metric = MulticlassMetrics(y_pred['Country_idx', 'prediction'].rdd)
```

```
/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/pyspark/sql/context.py:157: FutureWarning: Deprecated in 3.0.0.
Use SparkSession.builder.getOrCreate() instead.
  warnings.warn(
```

```
dir(lr_metric)
```

```
['__class__',
 '__del__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_java_model',
 '_sc',
 'accuracy',
 'call',
 'confusionMatrix',
 'fMeasure',
 'falsePositiveRate',
 'logLoss',
 'precision',
 'recall',
 'truePositiveRate',
 'weightedFMeasure',
 'weightedFalsePositiveRate',
 'weightedPrecision',
 'weightedRecall',
 'weightedTruePositiveRate']
```

```
print("Accuracy",lr_metric.accuracy)
```

```
Accuracy 0.808252427184466
```

```
print("Precision",lr_metric.precision(1.0))
print("Recall",lr_metric.recall(1.0))
print("F1Score",lr_metric.fMeasure(1.0))
```

```
Precision 0.8873239436619719
Recall 0.9692307692307692
F1Score 0.9264705882352942
```

```
# Saving Model
lr_model.save("Disneylr_model_30")
lr_model.write().save("Disneylr_model")
```

Therefore, the code above performs loading, cleaning, splitting, training, and evaluating of the dataset using PySpark.

The logistic regression model has an accuracy of 0.808, which indicates it identified 80.8% of the test data correctly.

The model's accuracy of category 1 is 0.887, which means that the model correctly predicted 88.7% of all instances as class 1.

The recall for class 1 is 0.969, implying that the model successfully identified 96.9% of the true class 1 cases.

The model has an F1-score of 0.926 in class 1 which represents the harmonic mean of accuracy and recall. It delivers a single score that balances precision and recall, with 1 being the greatest and 0 being the worst.

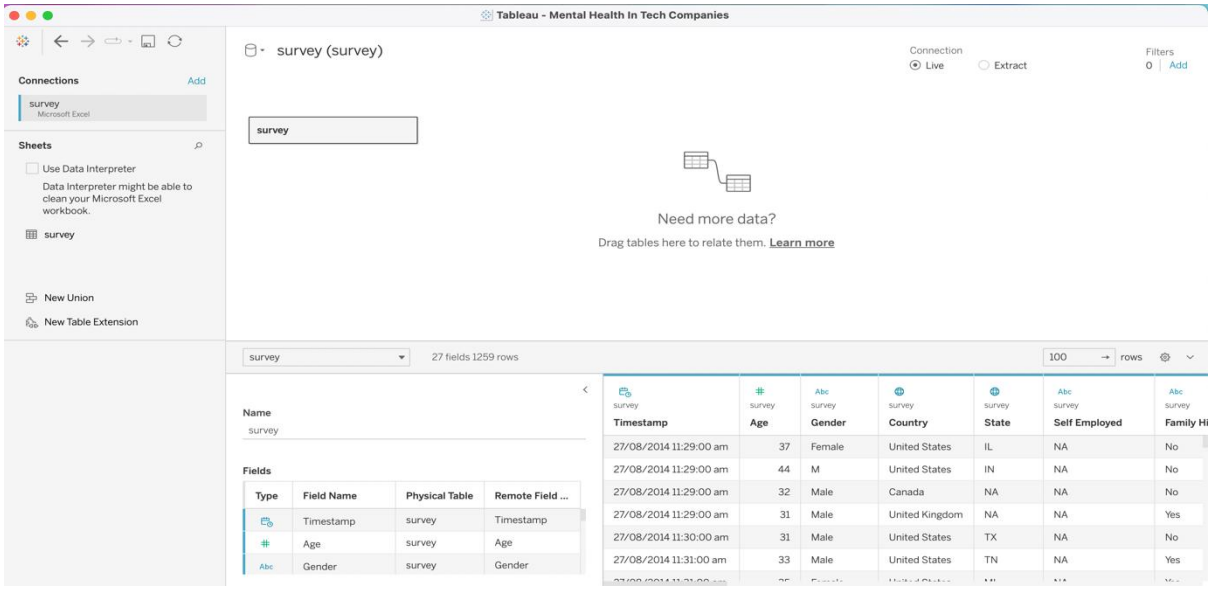
In terms of accuracy and precision, the logistic regression model did pretty well overall.

DATA VISUALIZATION

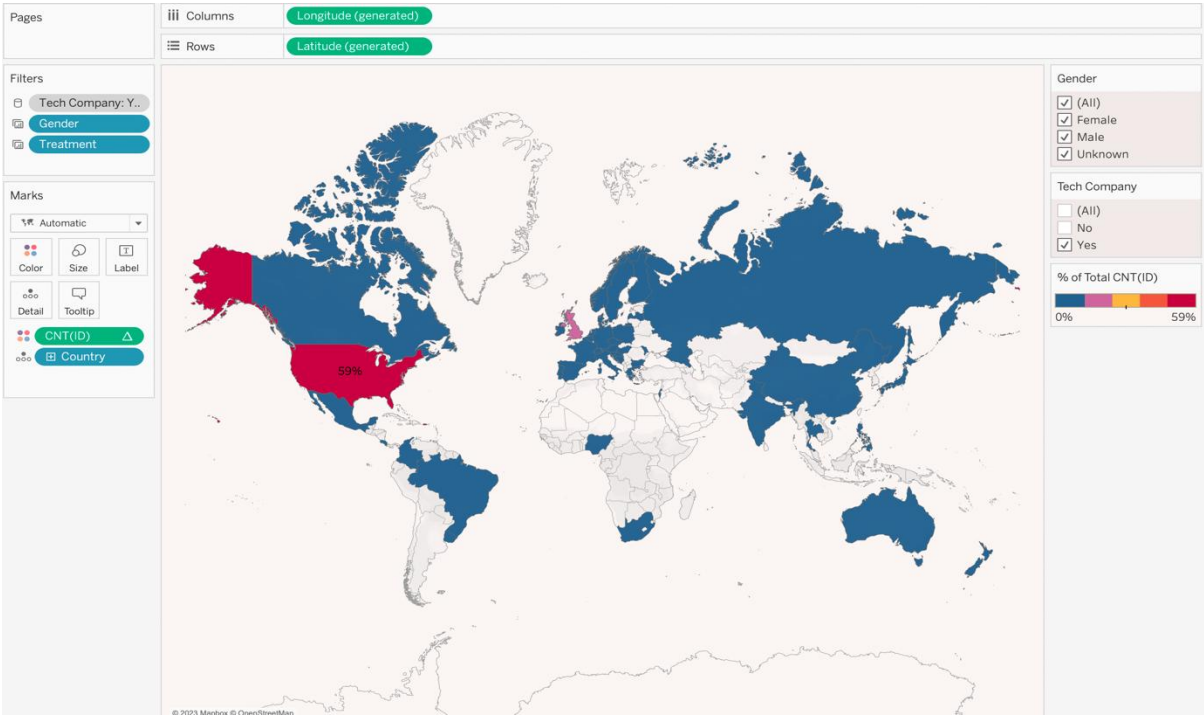
Data visualization using Tableau.

After the Tableau installation, open the tableau desktop and load the survey data in the 'To a File' session shown in the desktop.

The survey data is loaded and the Data source will be displayed in the screen which is provided in the below:

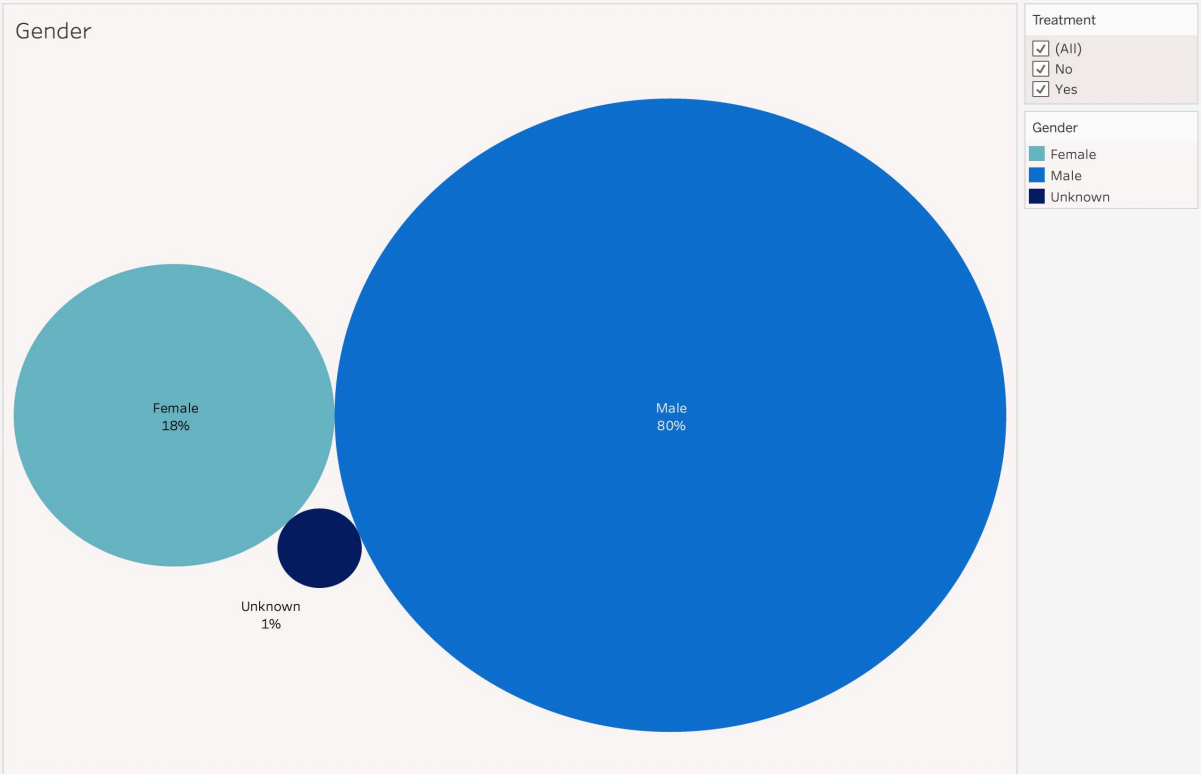


1)People responded to the survey



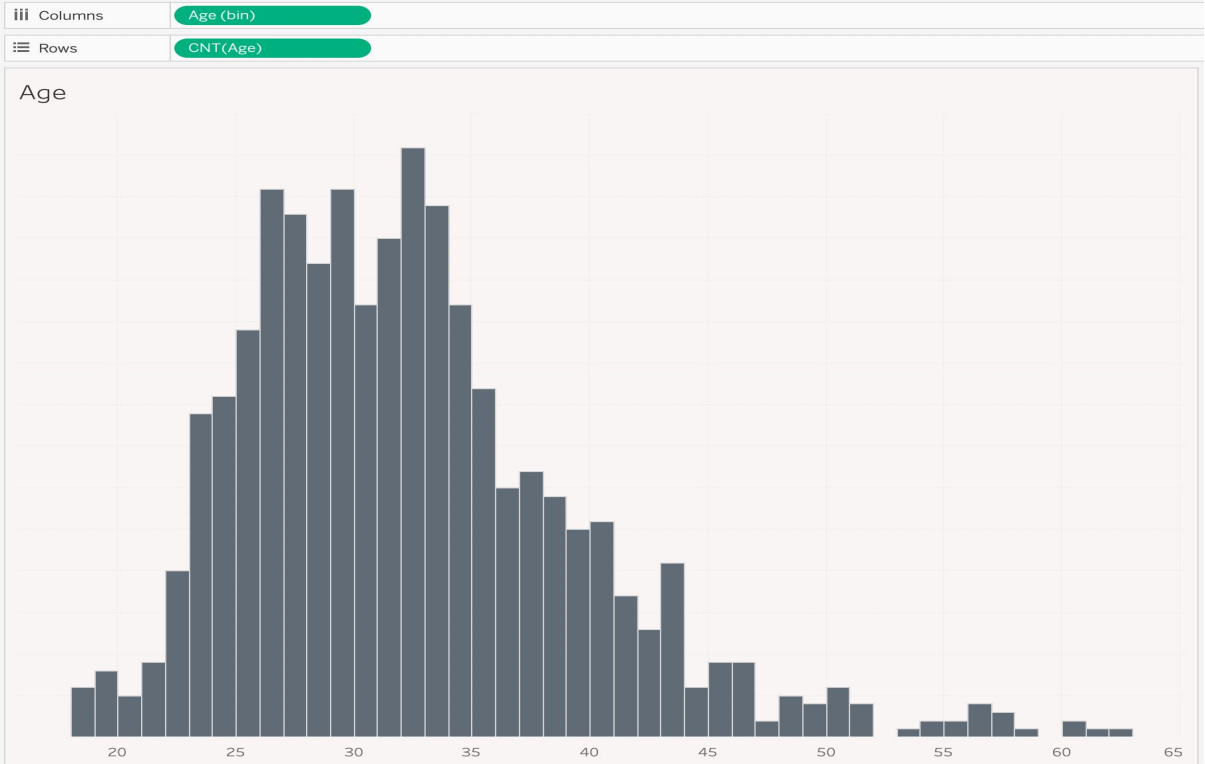
This is a map visualization shows that people from United states i.e. 59% who are in tech field responded to this survey in top and the second highest is the united kingdom i.e. 14%.

2) Responses based on the Gender



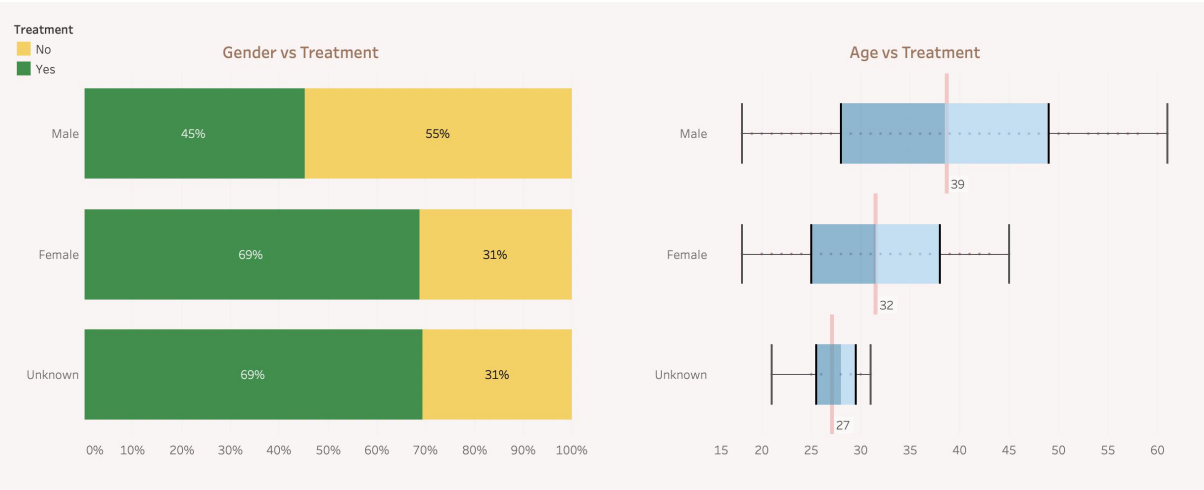
This is a Bubble chart visualization of the responses to the survey based on the Gender overall the world. Such that according to the above visualization 80% of the male has responded to the survey, 18% of the female and remaining 1% are unknown.

3)Responses based on age



This is a Histogram visualization that shows the responses recorded in the survey according to the age. Therefore the above histogram shows that the average age recorded is 32.

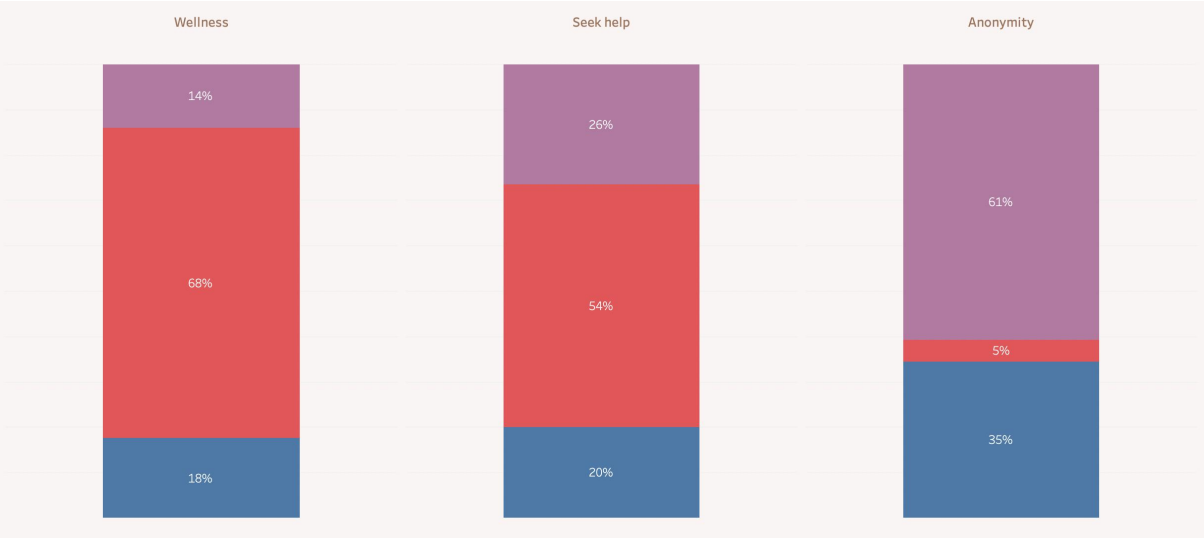
4) Treatment analysis



The stacked bar chart visualization shows the treatment analysis based on Gender vs Treatment such that the graphs explains that ‘female employees seek a lot of treatment for their mental health condition’.

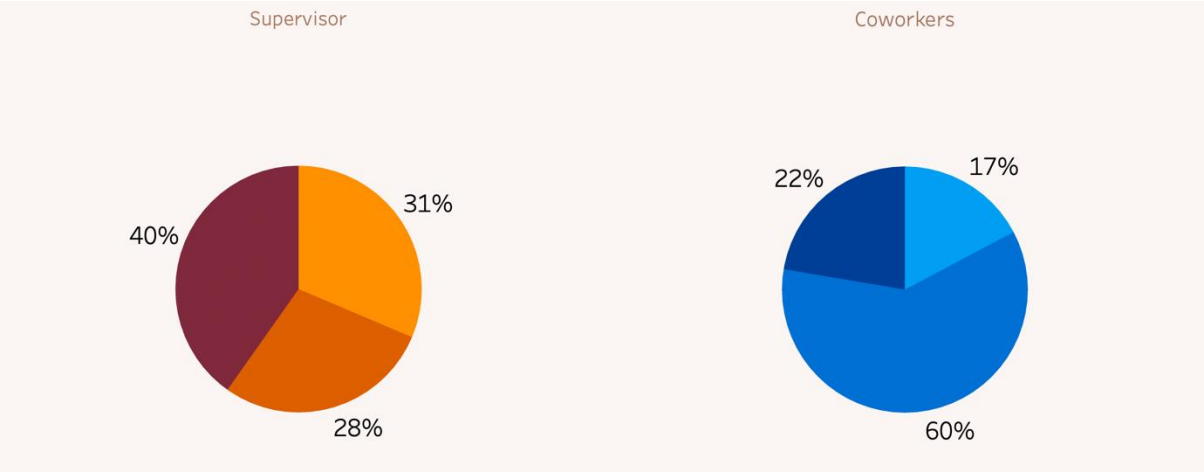
The box and whiskers visualization shows the treatment analysis based on the age vs Treatment such that the graph explains that ‘female are seeking for treatment at younger age for their mental health condition’.

5) Support from the company for mental health



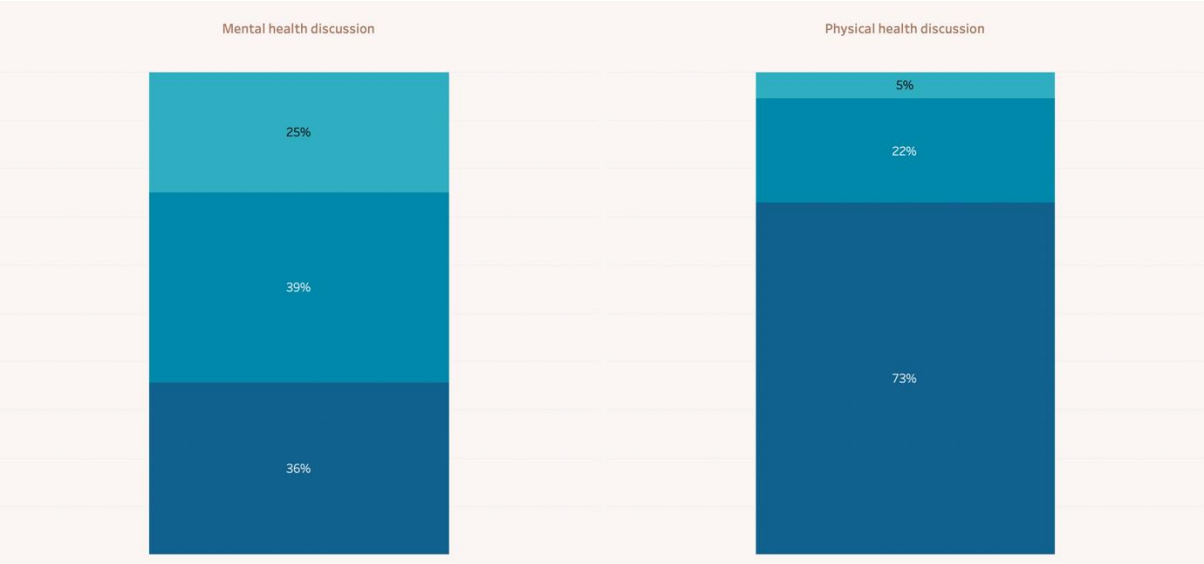
The above stacked bar charts visualize the company support for the employees regarding their mental health. Therefore, the graph1 shows that 'Mental health is incorporated into company wellness programmes.', graph2 shows that ‘Access to information on mental health difficulties and how to get treatment’ and graph3 shows that ‘Anonymity is protected when receiving mental health care’.

6) Sharing with supervisor’s vs co-workers



The pie charts visualize that the discussions about mental health between employee-supervisors and employee-coworkers. As a result of the graph, employees are more ready to disclose their mental health difficulties with their peers than with their bosses.

7) Mental health issues Discussions



The above stacked bar charts visualize the physical and mental health discussions such that 'The majority of people do not believe that addressing physical health concerns would have a negative impact'.

(Tableau.com, 2023)

CONCLUSION

The employment of Pyspark and machine learning in the IT industry can bring useful insights into mental health. Patterns and trends may be detected by analysing survey data, which may assist to better understand the mental health difficulties experienced by IT employees.

The following are some probable outcomes of such an analysis:

- Mental health concerns are common in the IT industry: survey results may show that a considerable proportion of tech professionals are suffering from anxiety, depression,

or other mental health disorders. This might indicate a need for increased industry support and resources to solve these concerns.

- Certain variables may lead to mental health difficulties in the technology industry: It may be feasible to detect connections between certain workplace characteristics (such as long hours, high-pressure conditions, or a lack of work-life balance) and the occurrence of mental health difficulties using machine learning algorithms. These findings might help shape workplace policies and practises targeted at lowering stress and increasing well-being.
- The survey's findings may reflect that many IT workers are uncomfortable addressing mental health difficulties with their colleagues or bosses, implying a need for further education and awareness-raising about these issues inside the sector.
- Demographic characteristics may influence mental health: Machine learning algorithms can help in determining whether particular demographic aspects, such as gender, age, or ethnicity, are linked to greater incidence of mental health concerns in the IT sector. This might assist to direct resources and support to individuals who are most vulnerable.
- Working from home may have an influence on one's mental health: Many IT workers have shifted to remote employment because of the COVID-19 outbreak. Machine learning algorithms might aid in determining whether this transition has affected mental health and whether there are special issues related with remote employment that must be addressed.

In the end, Pyspark and machine learning have the potential to be a powerful tool for assessing mental health data in the tech sector and driving actions to improve mental health and well-being among tech workers. The usage of Pyspark and machine learning can assist in shedding light on the complicated topic of mental health in the IT business. Patterns and

trends may be detected by analysing survey data, which may assist to drive workplace policies and practises targeted at fostering greater mental health and well-being for IT employees. They may put light on the complicated connections that exist between mental health, professional variables, and demographic factors in the technology industry. These results might help to guide the development of focused treatments aimed at improving the mental health and well-being of technology workers.

REFERENCES

Jesse, Jc. (2023). *pyspark-tutorials*. [online] GitHub. Available at: https://github.com/Jcharis/pyspark-tutorials#readme [Accessed 21 Apr. 2023].

Simplilearn.com. (n.d.). *20+ Tableau Charts with Uses and its Application for 2022*. [online] Available at: https://www.simplilearn.com/tutorials/tableau-tutorial/tableau-charts [Accessed 21 Apr. 2023].

Tableau.com. (2023). Available at: https://public.tableau.com/app/profile/almullaonline/viz/MentalHealthInTechCompanies/Story1 [Accessed 21 Apr. 2023].

www.kaggle.com. (n.d.). *Mental Health in Tech Survey*. [online] Available at: https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey?select=survey.csv [Accessed 21 Apr. 2023].

APPENDIX

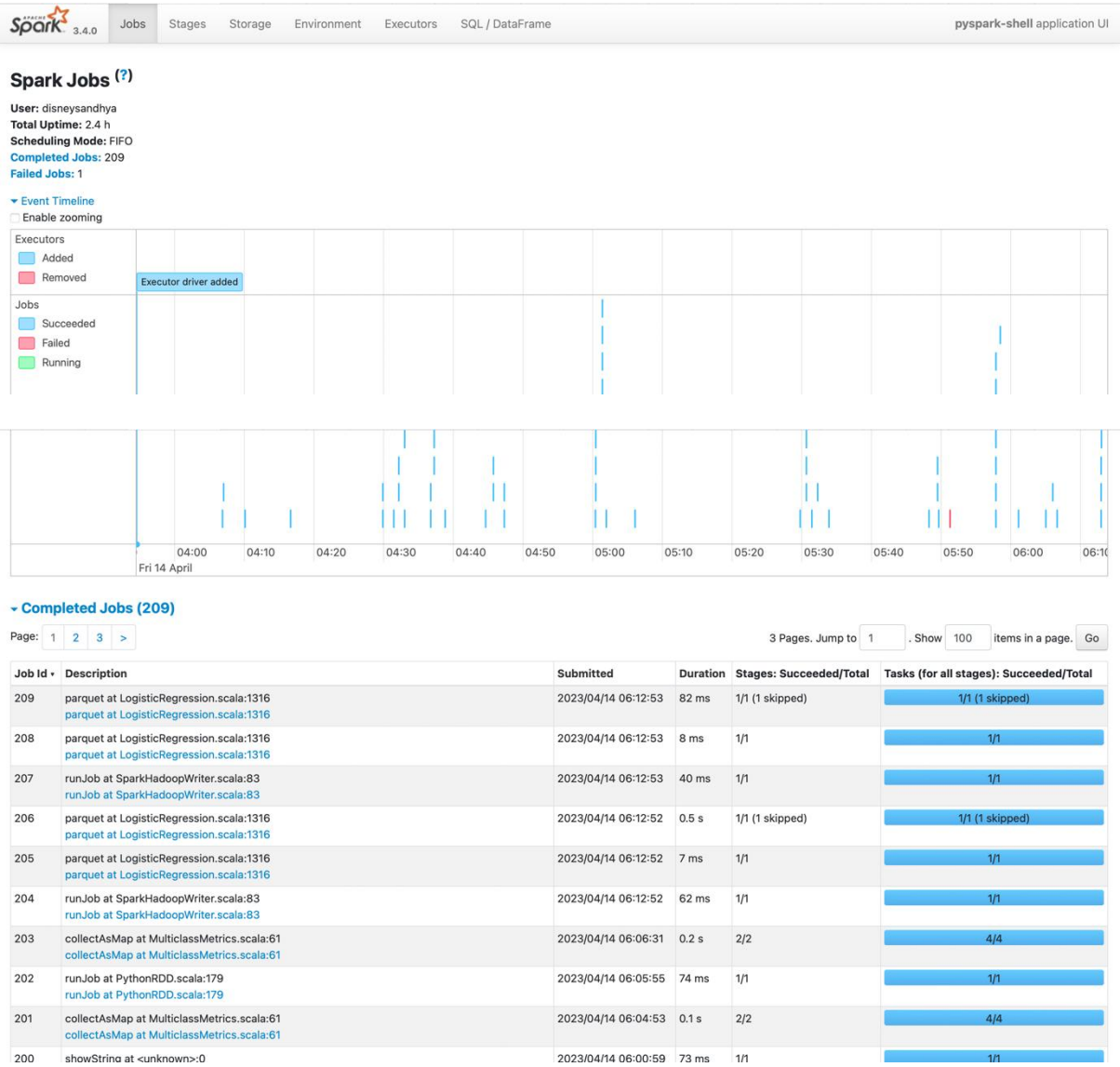
Code:

```
#PACKAGES
from pyspark import SparkContext
sc = SparkContext(master='local[2]')
#for spark UI
sc
#loading packages
from pyspark.sql import SparkSession
#spark appname in UI
spark = SparkSession.builder.appName("Disney's ML code").getOrCreate()
#loading the dataset
df = spark.read.csv("Desktop/survey.csv",header=True,inferSchema=True)
#shows the dataset
df.show()
#checking for coloumns
print(df.columns)
df = df.select('Age', 'Gender', 'Country', 'state', 'self_employed', 'family_history', 'treatment', 'work_interfere',
'no_employees', 'remote_work', 'tech_company')
df.show(5)
# Checking for datatypes
# InferSchema=True i.e before
df.dtypes
# InferSchema i.e after
df.dtypes
 # Checking for Schema
df.printSchema()
 # summary
print(df.describe().show())
# Value Count
df.groupBy('Country').count().show()
df.show(5)
import pyspark.ml
dir(pyspark.ml)
#loading ml packages
from pyspark.ml.feature import VectorAssembler,StringIndexer
df.show(4)
#Unique Values for employment
df.select('self_employed').distinct().show()
# Define the list of categorical columns to be encoded
cat_cols = ['Timestamp','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_interview', 'mental_vs_physical', 'obs_consequence', 'comments']
# Create a list of StringIndexer stages, one for each categorical column
indexers = [StringIndexer(inputCol=col, outputCol=col+'_idx') for col in cat_cols]
# Fit the indexers to the dataframe
cat_Encoder = [indexer.fit(df) for indexer in indexers]
# Apply the indexers to the dataframe
encoded_df = df
for encoder in cat_Encoder:
    encoded_df = encoder.transform(encoded_df)
```

```
encoded_df.show(5)
print(encoded_df.columns)
encoded_df.dtypes
df2 = encoded_df.select('Age','Gender_idx', 'Country_idx', 'state_idx', 'self_employed_idx', 'family_history_idx',
'treatment_idx',
'work_interfere_idx', 'no_employees_idx', 'remote_work_idx', 'tech_company_idx', 'benefits_idx',
'care_options_idx', 'wellness_program_idx', 'seek_help_idx','anonymity_idx', 'leave_idx',
'mental_health_consequence_idx', 'phys_health_consequence_idx', 'coworkers_idx',
'supervisor_idx','mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx',
'obs_consequence_idx', 'comments_idx')
df2.printSchema()
required_features = ['Age','Gender_idx', 'Country_idx', 'state_idx', 'self_employed_idx', 'family_history_idx',
'treatment_idx', 'work_interfere_idx', 'no_employees_idx', 'remote_work_idx', 'tech_company_idx', 'benefits_idx',
'care_options_idx', 'wellness_program_idx', 'seek_help_idx','anonymity_idx', 'leave_idx',
'mental_health_consequence_idx', 'phys_health_consequence_idx', 'coworkers_idx', 'supervisor_idx',
'mental_health_interview_idx', 'phys_health_interview_idx', 'mental_vs_physical_idx', 'obs_consequence_idx',
'comments_idx']
df2 = df2.toPandas().replace('NA',0).astype(float)
type(df2)
type(df)
#converting to pyspark dataframe
new_df = spark.createDataFrame(df2)
new_df.show()
 # Checking For DataTypes & Schema
new_df.printSchema()
#VectorAssembler
vec_assembler =VectorAssembler(inputCols=required_features,outputCol='features')
vec_df = vec_assembler.transform(new_df)
vec_df.show(5)
train_df,test_df = vec_df.randomSplit([0.7,0.3])
train_df.count()
train_df.show(4)
from pyspark.ml.classification import LogisticRegression,DecisionTreeClassifier
# Logistic regression Model
lr = LogisticRegression(featuresCol='features',labelCol='Country_idx')
lr_model = lr.fit(train_df)
y_pred = lr_model.transform(test_df)
y_pred.show()
print(y_pred.columns)
y_pred.select('Country_idx','rawPrediction', 'probability', 'prediction').show()
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
#Checking For Accuracy
multi_evaluator = MulticlassClassificationEvaluator(labelCol='Country_idx',metricName='accuracy')
multi_evaluator.evaluate(y_pred)
from pyspark.mllib.evaluation import MulticlassMetrics
lr_metric = MulticlassMetrics(y_pred['Country_idx', 'prediction'].rdd)
dir(lr_metric)
print("Accuracy",lr_metric.accuracy)
print("Precision",lr_metric.precision(1.0))
print("Recall",lr_metric.recall(1.0))
print("F1Score",lr_metric.fMeasure(1.0))
 # Saving Model
lr_model.save("Disneylr_model_30")
lr_model.write().save("Disneylr_model")
```

(Jesse, 2023)

Related screen shots of my work from Spark UI :



Related screenshots from Terminal:

```
~ — jupyter_mac.command — python • jupyter_mac.command                    ~ — pipenv shell
(base) disneysandhya@Disneys-MacBook-Air ~ % pipenv shell
Launching subshell in virtual environment...
 . /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/bin/activate
(base) disneysandhya@Disneys-MacBook-Air ~ %  . /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/bin/activate
(disneysandhya) (base) disneysandhya@Disneys-MacBook-Air ~ % pipenv run jupyter lab
[I 2023-04-14 03:48:24.380 ServerApp] Package jupyterlab took 0.0000s to import
[I 2023-04-14 03:48:24.383 ServerApp] Package jupyter_server_fileid took 0.0019s to import
[I 2023-04-14 03:48:24.388 ServerApp] Package jupyter_server_terminals took 0.0056s to import
[I 2023-04-14 03:48:24.645 ServerApp] Package jupyter_server_ydoc took 0.2568s to import
[I 2023-04-14 03:48:24.646 ServerApp] Package nbclassic took 0.0000s to import
[W 2023-04-14 03:48:24.647 ServerApp] A `_jupyter_server_extension_points` function was not found in nbclassic. Instead, a `_jupyter_server_extension_paths` function was found and will be used for now. Th
is function name will be deprecated in future releases of Jupyter Server.
[I 2023-04-14 03:48:24.648 ServerApp] Package notebook_shim took 0.0000s to import
[W 2023-04-14 03:48:24.648 ServerApp] A `_jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `_jupyter_server_extension_paths` function was found and will be used for now
. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-04-14 03:48:24.650 ServerApp] jupyter_server_fileid | extension was successfully linked.
[I 2023-04-14 03:48:24.652 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2023-04-14 03:48:24.655 ServerApp] jupyter_server_ydoc | extension was successfully linked.
[I 2023-04-14 03:48:24.658 ServerApp] jupyterlab | extension was successfully linked.
[I 2023-04-14 03:48:24.660 ServerApp] nbclassic | extension was successfully linked.
[I 2023-04-14 03:48:24.822 ServerApp] notebook_shim | extension was successfully linked.
[I 2023-04-14 03:48:24.977 ServerApp] notebook_shim | extension was successfully loaded.
[I 2023-04-14 03:48:24.977 FileIdExtension] Configured File ID manager: ArbitraryFileIdManager
[I 2023-04-14 03:48:24.978 FileIdExtension] ArbitraryFileIdManager : Configured root dir: /Users/disneysandhya
[I 2023-04-14 03:48:24.978 FileIdExtension] ArbitraryFileIdManager : Configured database path: /Users/disneysandhya/Library/Jupyter/file_id_manager.db
[I 2023-04-14 03:48:24.980 FileIdExtension] ArbitraryFileIdManager : Successfully connected to database file.
[I 2023-04-14 03:48:24.980 FileIdExtension] ArbitraryFileIdManager : Creating File ID tables and indices with journal_mode = DELETE
[I 2023-04-14 03:48:24.985 FileIdExtension] Attached event listeners.
[I 2023-04-14 03:48:24.985 ServerApp] jupyter_server_fileid | extension was successfully loaded.
[I 2023-04-14 03:48:24.986 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2023-04-14 03:48:24.987 ServerApp] jupyter_server_ydoc | extension was successfully loaded.
[I 2023-04-14 03:48:24.987 LabApp] JupyterLab extension loaded from /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyterlab
[I 2023-04-14 03:48:24.987 LabApp] JupyterLab application directory is /Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/share/jupyter/lab
[I 2023-04-14 03:48:24.989 ServerApp] jupyterlab | extension was successfully loaded.

  _   _                _                _
 | | | |_ __   __ _ | |___| |_ ___
 | |_| | '_ \/ _` / _` |  _/ _ \
  \___/| .__/\__,_\__,_|\__\___|
       |_|

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.

https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

[I 2023-04-14 03:48:24.992 ServerApp] nbclassic | extension was successfully loaded.
[I 2023-04-14 03:48:24.993 ServerApp] The port 8888 is already in use, trying another port.
[I 2023-04-14 03:48:24.994 ServerApp] Serving notebooks from local directory: /Users/disneysandhya
[I 2023-04-14 03:48:24.994 ServerApp] Jupyter Server 2.5.0 is running at:
[I 2023-04-14 03:48:24.994 ServerApp] http://localhost:8889/lab?token=37a79540e632a77f1d084c9ce0c82c9e6185628b48fa5189
[I 2023-04-14 03:48:24.994 ServerApp]     http://127.0.0.1:8889/lab?token=37a79540e632a77f1d084c9ce0c82c9e6185628b48fa5189
[I 2023-04-14 03:48:24.994 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2023-04-14 03:48:24.999 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/disneysandhya/Library/Jupyter/runtime/jpserver-5542-open.html
    Or copy and paste one of these URLs:
        http://localhost:8889/lab?token=37a79540e632a77f1d084c9ce0c82c9e6185628b48fa5189
        http://127.0.0.1:8889/lab?token=37a79540e632a77f1d084c9ce0c82c9e6185628b48fa5189



[I 2023-04-14 03:48:26.783 LabApp] Generating new user for token-authenticated request: 529c7dd7c50a442e9c2e963af47e6f0f
[I 2023-04-14 03:48:28.746 LabApp] Build is up to date
[I 2023-04-14 03:48:29.543 ServerApp] Kernel started: 067adc82-5c03-497e-ab6d-e02fd4a5f8e6
[I 2023-04-14 03:48:30.150 ServerApp] Connecting to kernel 067adc82-5c03-497e-ab6d-e02fd4a5f8e6.
[I 2023-04-14 03:48:30.152 ServerApp] Connecting to kernel 067adc82-5c03-497e-ab6d-e02fd4a5f8e6.
[I 2023-04-14 03:48:30.154 ServerApp] Connecting to kernel 067adc82-5c03-497e-ab6d-e02fd4a5f8e6.
[W 2023-04-14 03:48:30.190 ServerApp] Got events for closed stream <zmq.eventloop.zmqstream.ZMQStream object at 0x7fd582c34bb0>
[I 2023-04-14 03:50:29.308 ServerApp] Saving file at /Untitled.ipynb
[W 2023-04-14 03:51:16.693 ServerApp] 409 PATCH /api/contents/Untitled.ipynb?1681440676687 (::1): File already exists: BIGDATA ML CODE.ipynb
[W 2023-04-14 03:51:16.693 ServerApp] wrote error: 'File already exists: BIGDATA ML CODE.ipynb'
    Traceback (most recent call last):
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/tornado/web.py", line 1713, in _execute
        result = await result
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyter_server/services/contents/handlers.py", line 151, in patch
        model = await ensure_async(cm.update(model, path))
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyter_core/utils/__init__.py", line 182, in ensure_async
        result = await obj
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyter_server/services/contents/manager.py", line 901, in update
        await self.rename(path, new_path)
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyter_server/services/contents/manager.py", line 888, in rename
        await self.rename_file(old_path, new_path)
      File "/Users/disneysandhya/.local/share/virtualenvs/disneysandhya-7I8Ae6ZQ/lib/python3.9/site-packages/jupyter_server/services/contents/filemanager.py", line 1045, in rename_file
        raise web.HTTPError(409, "File already exists: %s" % new_path)
    tornado.web.HTTPError: HTTP 409: Conflict (File already exists: BIGDATA ML CODE.ipynb)
[W 2023-04-14 03:51:16.699 ServerApp] 409 PATCH /api/contents/Untitled.ipynb?1681440676687 (529c7dd7c50a442e9c2e963af47e6f0f@::1) 9.35ms referer=http://localhost:8889/lab
[I 2023-04-14 03:54:29.350 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
23/04/14 03:54:30 WARN Utils: Your hostname, Disneys-MacBook-Air.local resolves to a loopback address: 127.0.0.1; using 192.168.1.177 instead (on interface en0)
23/04/14 03:54:30 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/04/14 03:54:31 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
23/04/14 03:54:31 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
[I 2023-04-14 03:56:29.392 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
[I 2023-04-14 03:58:29.460 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
[I 2023-04-14 04:00:29.518 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
[I 2023-04-14 04:02:29.566 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
[I 2023-04-14 04:06:29.615 ServerApp] Saving file at /MLWITHPYSPARK.ipynb
```