DEVELOPMENT PHASE

PRODUCT SALES ANALYSIS PROJECT

Date	**19-10-2023**
Team ID	**1280**
Project Name	**Public health awareness compaign analysis**

Table of Contents

5	Overall Observation

1. Project Introduction :

In the development phase of a public health awareness campaign analysis, we delve into the intricacies of transforming ideas and strategies into actionable initiatives that can drive positive change. This critical phase represents the bridge between planning and execution, where the blueprint for the campaign comes to life. It involves creating the necessary infrastructure, content, and mechanisms to effectively reach the target audience, raise awareness, and inspire behavioral change.

During this phase, careful consideration is given to technological tools, data collection methods, and the ethical framework that underpins the campaign. The focus is on building the foundation for robust data analysis, interactive engagement, and comprehensive evaluation. As we navigate through the development phase, the campaign's potential to make a lasting impact on public health and well-being takes a tangible form, setting the stage for subsequent stages of implementation and analysis.

2. Problem Statement :

A problem statement is a concise, clear, and well-defined articulation of a specific issue or challenge that needs to be addressed. It serves as a starting point for any project, research, or problem-solving endeavor.

3. Data Pre-Processing :

- **Data Import:** Begin by importing your dataset using the appropriate libraries in Jupyter, such as Pandas. Load the dataset from the CSV file into a DataFrame.

- **Data Cleaning**: Describe the data cleaning steps in Jupyter. This might include handling missing values, identifying and managing outliers, and data validation. Use code snippets to demonstrate these processes..

- **Data Integration**: I have integrated data from jupyter and IBM cognos analytics.

- **Data Splitting:** I have split the data into training and testing sets for machine learning models. Include code for this process.

- **Data Visualization**: I visually explored the data in IBM cognos.

- **Data Export**: If you made any changes or created new datasets, mention if and how you exported the preprocessed data for further analysis.

Step 1: Import libraries

```
import pandas as pd
import numpy as np

import seaborn as sns
```

import matplotlib.pyplot as plt

step 2: Loading the data

data = pd.read_csv('/kaggle/input/mental-health-in-tech-survey/survey.csv')

data.head()

output:

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_health_consequence	phys_health_consequence	coworkers	supervisor	mental_health_interview	phys_health_interview	mental_vs_physical	obs_consequence	comments
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	No	Yes	Often	6-25	...	Somewhat easy	No	No	Some of them	Yes	No	Maybe	Yes	No	NaN
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	Maybe	No	No	No	No	No	Don't know	No	NaN
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	No	No	Rarely	6-25	...	Somewhat difficult	No	No	Yes	Yes	Yes	Yes	No	No	NaN
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	Yes	Yes	Some of them	No	Maybe	Maybe	No	Yes	NaN
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN	No	No	Never	100-500	...	Don't know	No	No	Some of them	Yes	Yes	Yes	Don't know	No	NaN

5 rows × 27 columns

Step 3: Preprocessing and Cleaning dataset:

There is a missing values in our data is present so we have to find how many missing values are present, so the code to find the missing values is executed.

```
          if data.isnull().sum().sum() == 0 :
    print ('There is no missing data in our dataset')
else:
    print('There is {} missing data in our dataset '.format(data.
isnull().sum().sum()))
```

output:

There is 1892 missing data in our dataset

```
frame = pd.concat([data.isnull().sum(), data.nunique(),
data.dtypes], axis = 1, sort= False)

frame
```

output:

	0	1	2
timestamp	0	**1246**	object
age	0	**53**	int64
gender	0	**49**	object
country	0	**48**	object
State	515	**45**	object
Self employed	0	**2**	object
Family history	0	**2**	object
Treatment	0	**2**	object
Work interfere	264	**4**	object
No employees	0	**6**	object
Remote work	0	**2**	object
Tech company	0	**2**	object
Benefits	0	**3**	object
Care options	0	**3**	object
Wellness program	0	**3**	object
Seek help	0	**3**	object
Anonymity	0	**3**	object
Leave	0	**5**	object

Mental health consequences	0	**3**	object
Phys health consequences	0	**3**	object
Co workers	0	**3**	object
Supervisor	0	**3**	object
Mental health interview	0	**3**	object
Physical health interview	0	**3**	object
Obs consequences	0	**2**	object
Comments	**1095**	**160**	object

Step 4: Checking the unique data in columns

```
     #Check unique data in gender columns
print(data['Gender'].unique())
print('')
print('-'*75)
print('')
#Check number of unique data too.
print('number of unique Gender in our dataset is :'
, data['Gender'].nunique())
```

output:

['Female' 'M' 'Male' 'male' 'female' 'm' 'Male-ish' 'maile' 'Trans-female'

'Cis Female' 'F' 'something kinda male?' 'Cis Male' 'Woman' 'f' 'Mal'

'Male (CIS)' 'queer/she/they' 'non-binary' 'Femake' 'woman' 'Make' 'Nah'

'All' 'Enby' 'fluid' 'Genderqueer' 'Female ' 'Androgyne' 'Agender'

'cis-female/femme' 'Guy (-ish) ^_^' 'male leaning androgynous' 'Male
'

'Man' 'Trans woman' 'msle' 'Neuter' 'Female (trans)' 'queer'

'Female (cis)' 'Mail' 'cis male' 'A little about you' 'Malr' 'p' 'femail'

'Cis Man' 'ostensibly male, unsure what that really means']

---

number of unique Gender in our dataset is : 49

.

Step 5: Check for gender problem

```
data['Gender'].replace(['Male ', 'male', 'M', 'm', 'Male', 'Cis Male',
                        'Man', 'cis male', 'Mail', 'Male-ish', 'Male (CIS)',
                        'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make',], 'Male',
inplace = True)


data['Gender'].replace(['Female ', 'female', 'F', 'f', 'Woman',
'Female',
```

```
                'femail', 'Cis Female', 'cis-female/femme', 'Femake',
'Female (cis)',
                'woman',], 'Female', inplace = True)


data["Gender"].replace(['Female (trans)', 'queer/she/they', 'non-
binary',
                'fluid', 'queer', 'Androgyne', 'Trans-female', 'male leaning
androgynous',
                 'Agender', 'A little about you', 'Nah', 'All',


                 'Genderqueer', 'Enby', 'p', 'Neuter', 'something kinda
male?',
                 'Guy (-ish) ^_^', 'Trans woman',], 'Other', inplace =
True)


print(data['Gender'].unique())
```

output:

```
['Female' 'Male' 'Other']
```

Step 6: check for duplicated data

```
if data.duplicated().sum() == 0:
    print('There is no duplicated data:')
```

```
else:
    print('Tehre is {} duplicated data:'.format(data.duplicated().sum()))
    #If there is duplicated data drop it.
    data.drop_duplicates(inplace=True)


print('-'*50)
print(data.duplicated().sum())
```

output:

There is 4 duplicated data:

--

0

Step 7: Split the data to train and test

```
from sklearn.model_selection import train_test_split

#I wanna work on 'treatment' column.
X = data.drop(columns = ['treatment'])
y = data['treatment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

print(X_train.shape, y_train.shape)
print('-'*30)
print(X_test.shape, y_test.shape)
print('_'*30)
```

output:

(937, 23) (937,)

```
-----------------------------
(313, 23) (313,)
```

Step 8: Random Forest Classifier

```
steps_rfc = [('Scaler', StandardScaler()),
             ('clf', RFC(n_estimators = 40))]



clf_rfc = Pipeline(steps=steps_rfc)



clf_rfc.fit(X_train, y_train)

y_pred_rfc = clf_rfc.predict(X_test)
print('RFC accuracy: ', accuracy_score(y_true=y_test,
    y_pred=y_pred_rfc)*100)
```

output:

```
RFC accuracy:  69.6485623003195
```

4. Data Visualization:

CHART-1

Chart Insights were not computed because this visualization is based on clipped data. Consider applying a filter to reduce the number of records, and to prevent the data from being clipped, before creating the visualization.

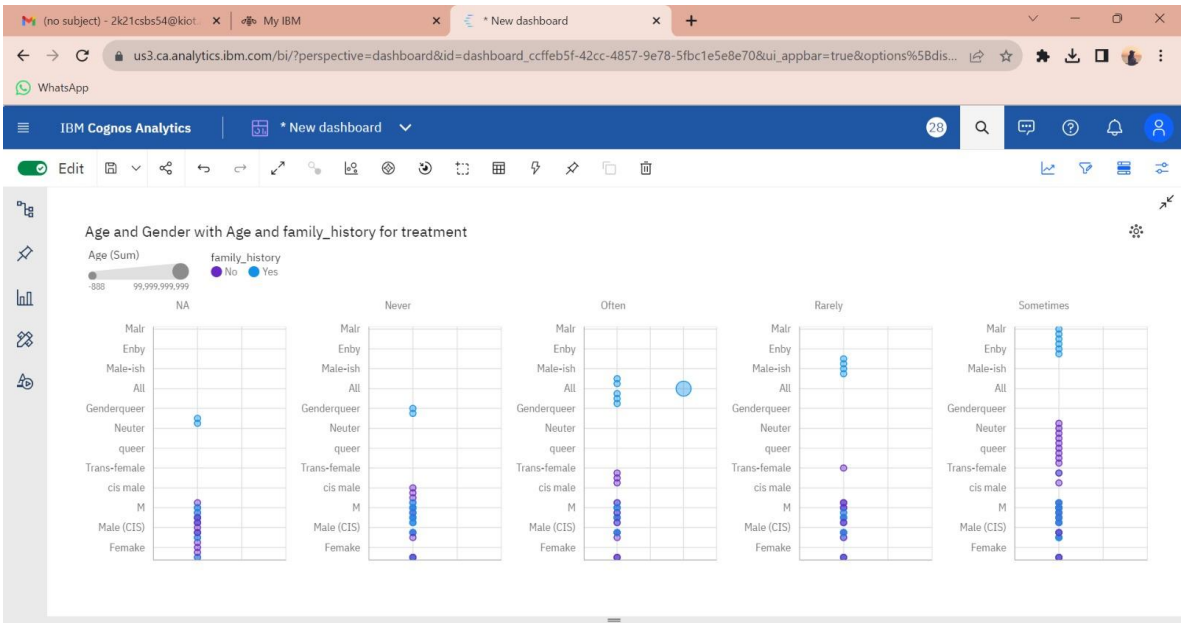


CHART-2

Chart Insights were not computed because this visualization is based on clipped data. Consider applying a filter to reduce the number of records, and to prevent the data from being clipped, before creating the visualization.

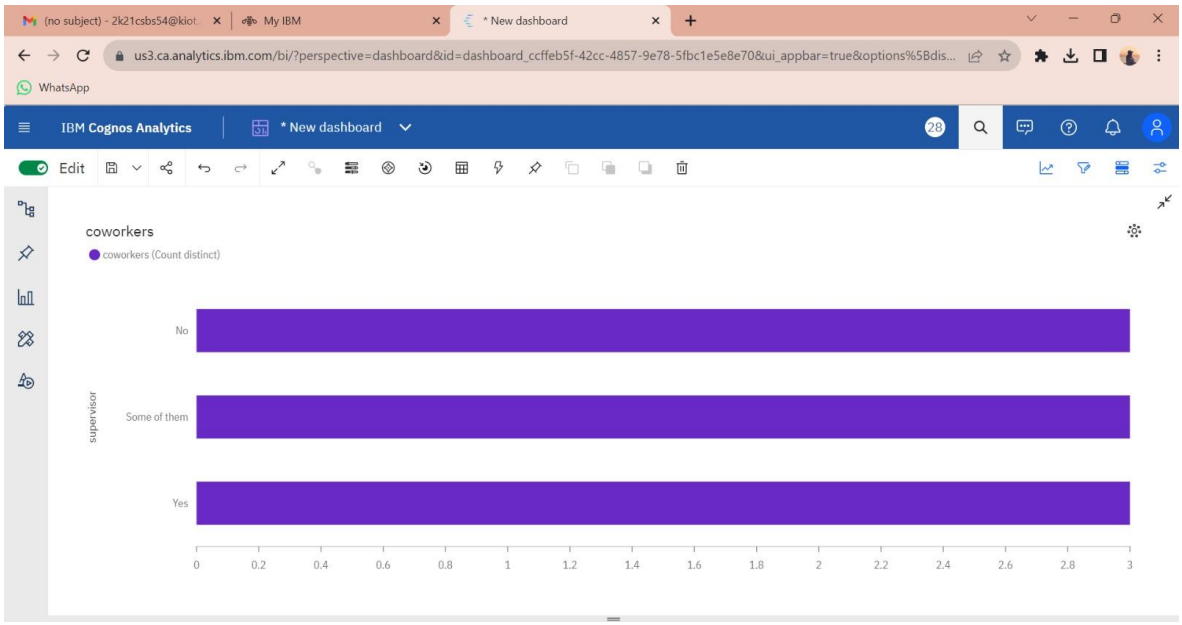
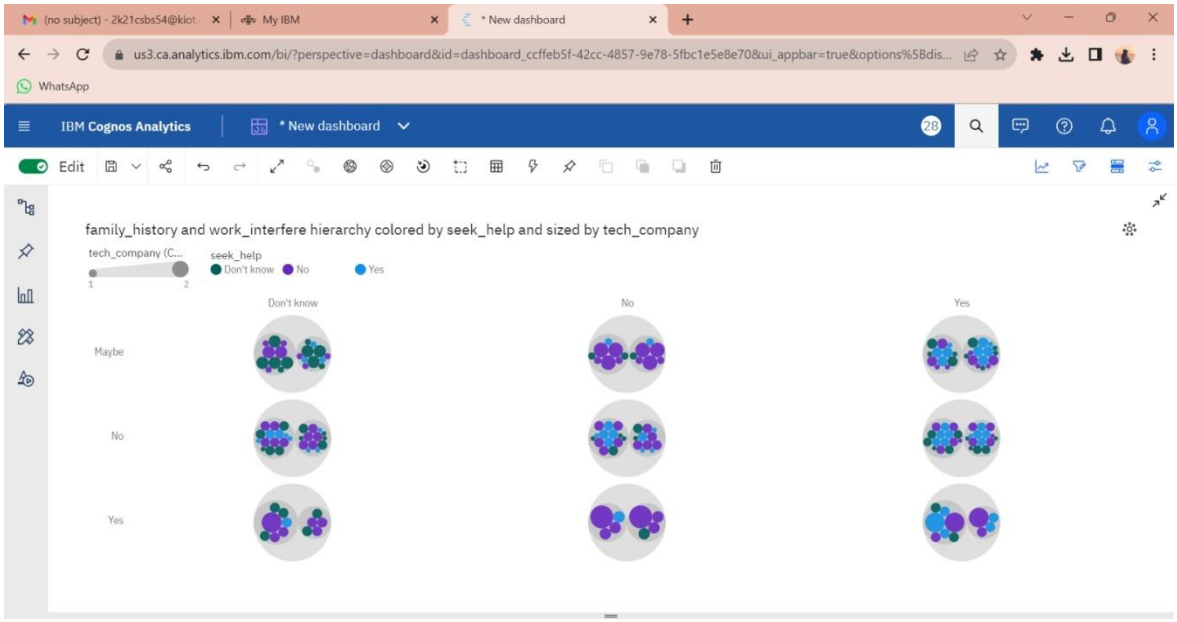


Chart 3



Overall observation:

During the development phase of a public health awareness campaign analysis, several notable observations come to light. First and foremost, the integration of data from diverse sources reveals the complexity and richness of the dataset, setting the stage for comprehensive analysis. Cleaning efforts unveil the extent of data quality issues, such as missing values and inconsistencies, while transformations like scaling and encoding begin to reshape the data distribution. Observations regarding data reduction and feature engineering highlight the trade-offs between dimensionality and informative features. Ethical considerations concerning data privacy and security become evident, with safeguards put in place to protect user information. The division of data into training and testing sets offers insights into data representativeness. Visualizations showcase trends and outliers, and the

technology stack, including Jupyter for preprocessing and IBM Cognos for advanced analysis, plays a pivotal role. Additionally, the development phase prompts critical assessments of campaign strategy, personalized content, and real-time engagement, shaping the foundation for effective public health awareness campaigns.