

## DEVELOPMENT PHASE PART 2

### Public health awareness compaign analysis

Date	24-10-2023
Team ID	1280
Project Name	Public health awareness compaign analysis

#### Table Of Content:

- 1.Introduction.
- 2.Data Preprocessing.
- 3.Data Visualization.
  - 3.1. Histogram & Distribution.
- 4.CognosVisualiation.
- 5.5.Conclusion

#### 1.Introduction:

In the development part 2 the project is Continued building the analysis by creating visualizations and building a predictive model. Using visualization libraries like Matplotlib, Seaborn the histograms were created. 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.

#### 2.Data Preprocessing:

In the second development phase, data preprocessing is a critical step. It involves cleaning the data by addressing missing values, duplicates, and outliers. Feature selection helps to choose the most relevant attributes, reducing dimensionality and enhancing model efficiency. Data transformation includes normalizing numerical data, encoding categorical variables, and handling date and time information appropriately

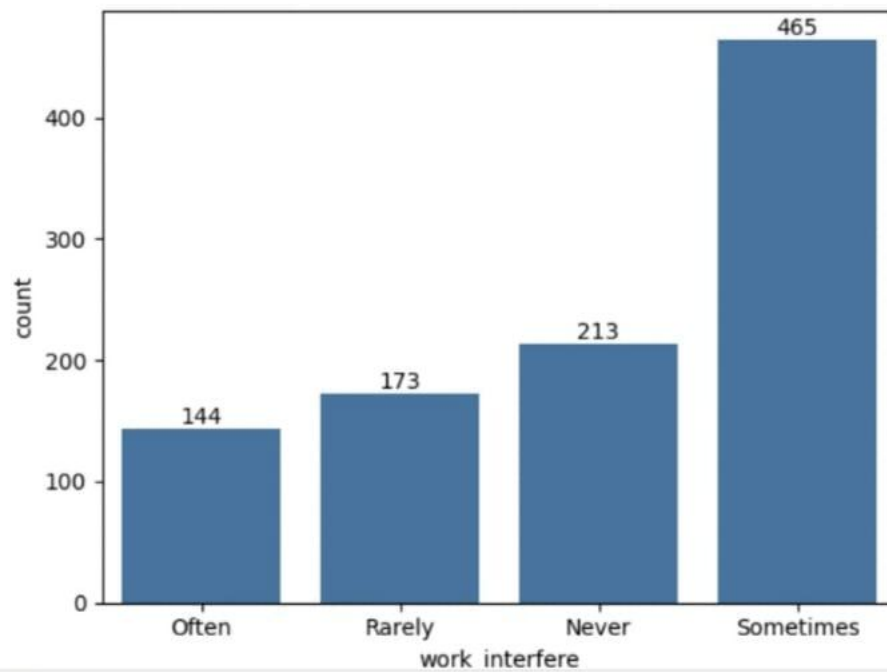
## Data visualization:

### 3.1 HISTOGRAM AND DISTRIBUTION.

Plotting the graph in relation with work interfere and country.

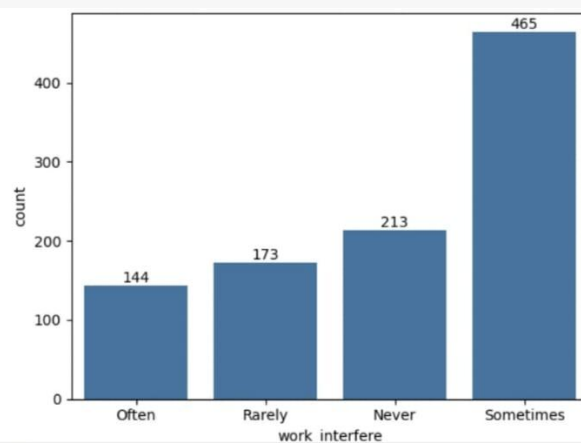
```
#Plot **work_interfere**  
ax = sns.countplot(data = data , x = 'work_interfere');  
#Add the value of each parametr on the Plot
```

```
ax.bar_label(ax.containers[0]);
```



```
ax = sns.countplot(data=data, x='work_interfere');  
ax.bar_label(ax.containers[0]);
```

**OUTPUT:**

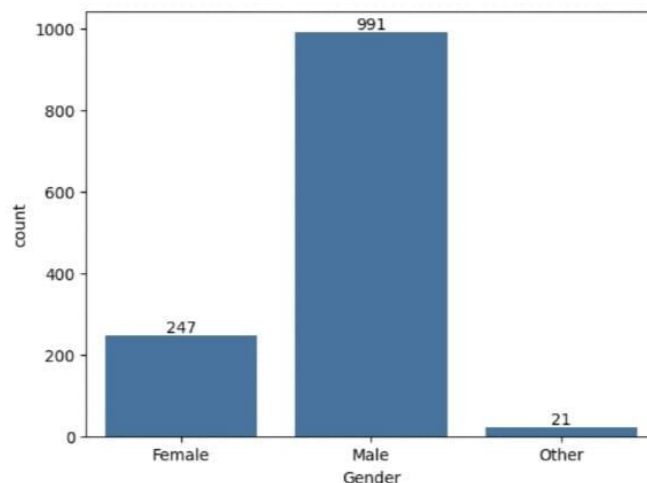


#Plot Genders column after cleaning and new categorizing

```
ax = sns.countplot(data=data, x='Gender');
```

```
ax.bar_label(ax.containers[0]);
```

**OUTPUT:**



### Observation

The visualization will help assess whether the data cleaning and categorization have effectively organized the gender data or if there are any remaining issues or imbalances in the gender categories.

This plot is useful for identifying patterns in gender distribution and ensures that the data preprocessing steps have successfully transformed the 'Gender' column into a more manageable and informative format for analysis.

```
data['Age'].unique()
```

```
array([ 37,  44,  32,  31,  33,
        35,  39,  42,  23,  29,
        36,  27,  46,  41,  34,
        30,  40,  38,  50,  24,
        18,  28,  26,  22,  19,
        25,  45,  21, -29,  43,
        56,  60,  54, 329,  55,
        9999999999, 48,  20,  57,  58,
```

```

47,      62,      51,      65,      49,
-1726,    5,      53,      61,      8,
11,      -1,      72], dtype=int64)

```

## Observation:

By examining the unique values, you can identify the different ages represented in the dataset.

This information is helpful for understanding the age distribution within the dataset and may be crucial for various analyses and modeling tasks.

It also allows you to check for any potential data entry errors or outliers in the 'Age' column, which can be addressed during data preprocessing.

```

#We had a lot of nonsense answers in the Age column too
#This filtering will drop entries exceeding 100 years and those
indicating negative values.
data.drop(data[data['Age']<0].index, inplace = True)
data.drop(data[data['Age']>99].index, inplace = True)

print(data['Age'].unique())
[37 44 32 31 33 35 39 42 23 29 36 27 46 41 34 30 40 38 50 24 18 28 26
 22
 19 25 45 21 43 56 60 54 55 48 20 57 58 47 62 51 65 49  5 53 61  8 11
 72]

```

```

#Let's see the Age distribution in this dataset.

```

```

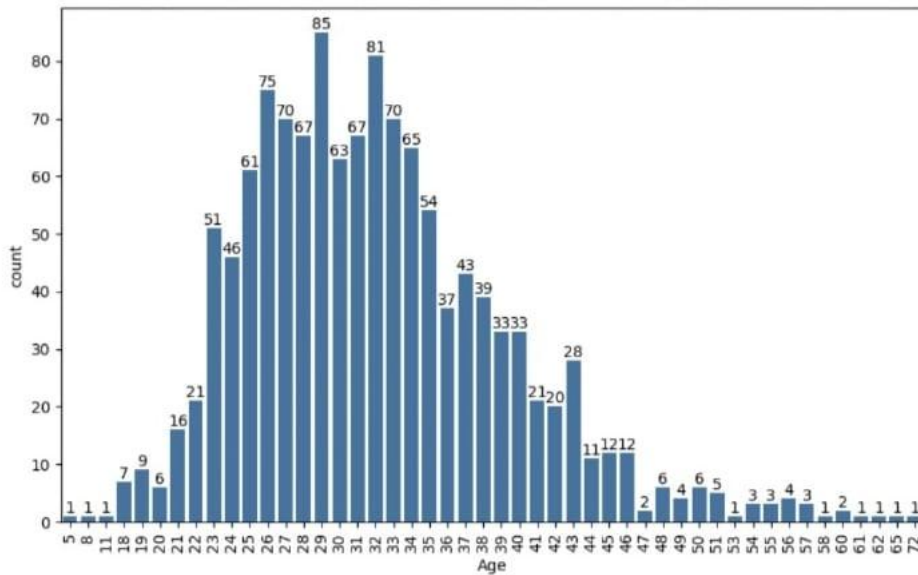
plt.figure(figsize = (10,6))

age_range_plot = sns.countplot(data = data, x = 'Age');

age_range_plot.bar_label(age_range_plot.containers[0]);

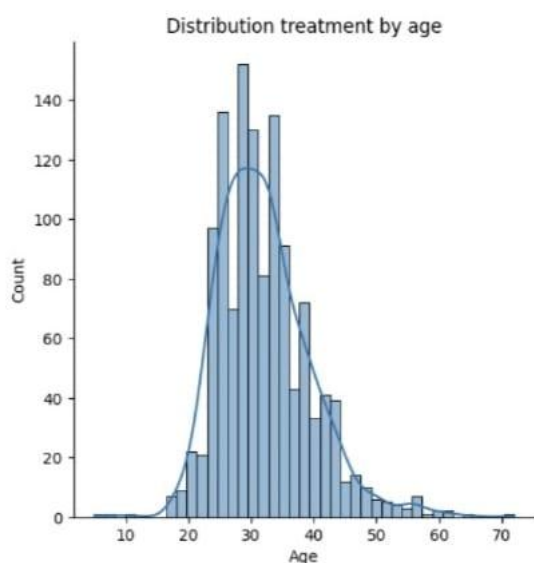
plt.xticks(rotation=90);

```



#In this plot moreover on Age distribution we can see treatment distribution by age

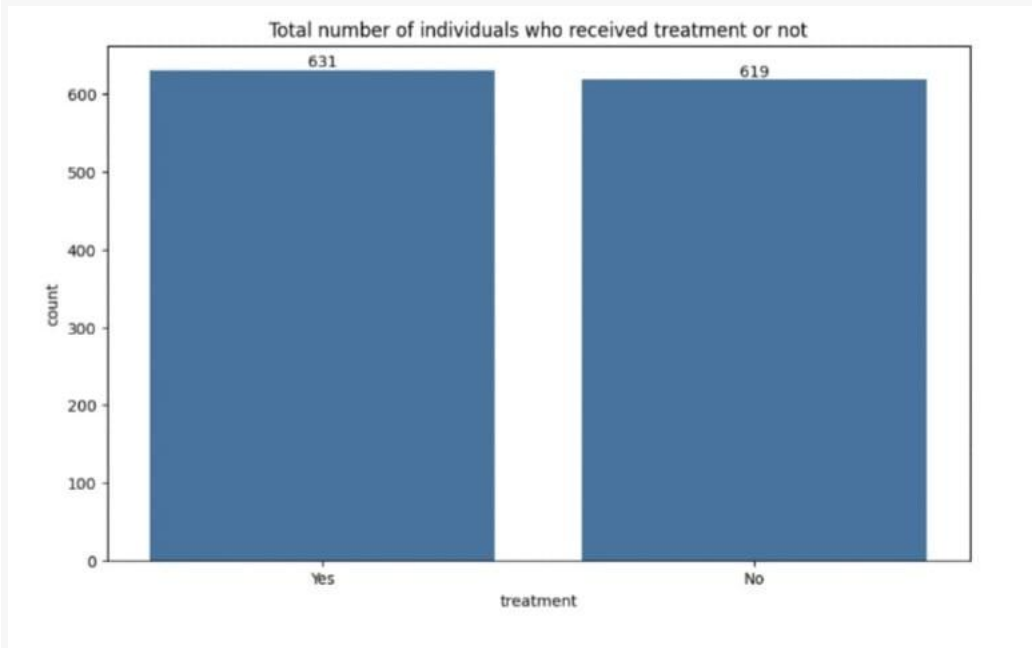
```
plt.figure(figsize=(10, 6));
sns.displot(data['Age'], kde = 'treatment');
plt.title('Distribution treatment by age');
```



#In this plot We can see Total number of individuals who received treatment or not.

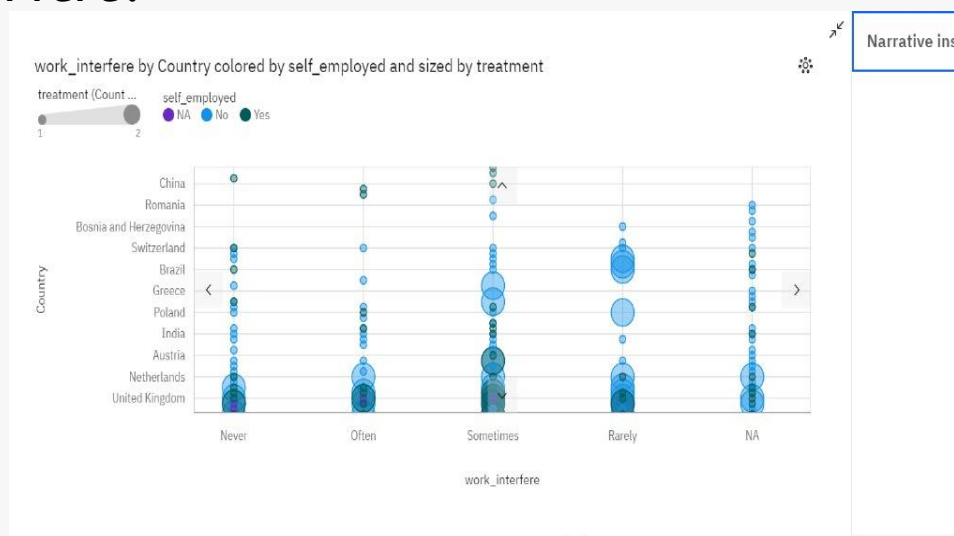
```
plt.figure(figsize = (10,6));
```

```
treat = sns.countplot(data = data, x =
'treatment');
treat.bar_label(treat.containers[0]);
plt.title('Total number of individuals who received
treatment or not');
```



## IBM COGNOS VISUALIZATION:

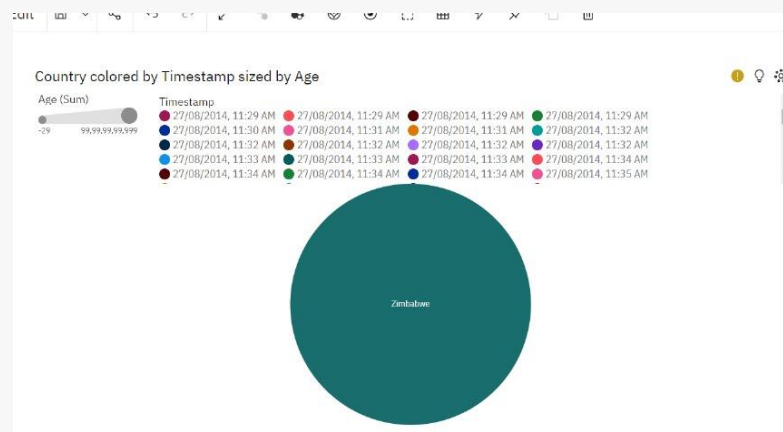
**Visualization between country and work interfere:**



## Visualization between country and gender:



## visualization between country and age:



## Visualization between country, treatment and age:



## Conclusion

In conclusion, the data preprocessing steps undertaken in the second phase of our project have been pivotal in preparing the dataset for analysis and model development. These steps, which include data cleaning, feature selection, transformation, and proper data splitting, have laid the foundation for robust and accurate modeling. Additionally, addressing class imbalances and conducting feature engineering further improves the quality of our data.

Moreover, the integration of IBM Cognos for visualization techniques has been instrumental in creating insightful and interactive data visualizations. This platform's capabilities have enabled us to communicate our findings effectively, allowing for better decision-making and a deeper understanding of the data.

By following these data preprocessing best practices and leveraging IBM Cognos for visualization, we are well-positioned to deliver meaningful insights and drive informed actions in our project.



