Public Health Awareness Campaign

Project title : Public health awareness campaign using Data analytics

Phase 4 - project submission - development part2

M.Jini Mol

962221106059

INTRODUCTION:

Public health awareness campaigns play a vital role in informing and educating the general population about various health-related issues. These campaigns aim to raise awareness, promote healthy behaviors, and ultimately improve overall public health. In recent years, data analytics has emerged as a powerful tool in the field of public health, enabling organizations to make informed decisions, target interventions effectively, and measure the impact of their awareness campaigns. One popular tool for data analytics in public health campaigns is IBM Cognos.

IBM Cognos is a business intelligence and performance management software that can be effectively leveraged in public health campaigns to collect, analyze, and present data to support evidence-based decision-making.

Here's how to introduce a public health awareness campaign using data analytics with Cognos:

Define the Objective: Start by clearly defining the objective of your public health awareness campaign. Identify the specific health issue you aim to address, such as promoting vaccination, reducing the prevalence of a particular disease, or encouraging healthy lifestyle choices. Your objective should be specific, measurable, achievable, relevant, and time-bound (SMART).

Data Collection: Collect relevant data to support your campaign. This can include demographic information, health statistics, survey results, and information on the target population. Data can be obtained from various sources, such as public health agencies, surveys, electronic health records, and more.

Data source:

https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey

Data Preparation: Prepare and clean the collected data to ensure it is accurate and ready for analysis. This involves data cleaning, data transformation, and handling missing values. IBM Cognos offers data integration and transformation capabilities to streamline this process.

Data Analysis: Utilize IBM Cognos for data analysis. The software provides tools for data visualization, reporting, and dashboard creation. Use these features to uncover insights, trends, and patterns in the data, helping you understand the current state of the health issue and the target population's characteristics.

Target Audience Segmentation: Segment the target audience based on the insights gained from data analysis. This segmentation allows for personalized messaging and tailored interventions, increasing the campaign's effectiveness.

Campaign Planning: Develop a comprehensive campaign plan that includes messaging, communication channels, and a timeline for deployment. IBM Cognos can help in planning and tracking the campaign's progress with its reporting and dashboard features.

Monitoring and Evaluation: Implement the awareness campaign and continuously monitor its progress. Evaluate the impact of the campaign using key performance indicators (KPIs) and metrics derived from data analytics. Cognos enables real-time monitoring and reporting, facilitating data-driven decision-making throughout the campaign.

Adjust and Optimize: Based on the campaign's performance data, make necessary adjustments to the messaging and strategies. IBM Cognos can help you identify areas for improvement and optimize your campaign in real time.

Reporting and Communication: Use IBM Cognos to create reports and dashboards to communicate the campaign's results and impact to stakeholders, including government agencies, healthcare providers, and the general public.

Importing Libraries

In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


%matplotlib inline


df = pd.read_csv('survey.csv')
```

Understanding the Dataset

1. Total Number of Rows and Columns

In [ ]:

```
print(df.shape)
```

2. Understanding the Features

In [ ]:

```
df.head()
```

In [ ]:

```
df['supervisor'].value_counts()
```

In [ ]:

```
df['mental_health_consequence'].value_counts()
```

3. Data Types of Each Feature

In [ ]:

```
df.dtypes
```

In this dataset, only age is numerical. Every other feature is categorical.

4. Dataset Information

In [ ]:

```
print(df.info())
```

Data Preparation

1. Dropping Features

The features 'Timestamp', 'Country', 'state', and 'comments' will be dropped. 'Timestamp' and 'comments', which simply shows the date and time in which the respondent took the survey and the respondent's additional comments, do not provide any additional benefit to our observations. Although 'Country' definitely plays a role in the mental health of a person, it may result in inaccurate observations, as in this dataset, most of the respondents are from the USA, and there are several countries with only a single respondent, which may create biased observations. Due to this, we will also be dropping the 'state' feature.

In [ ]:

```
print(df['Country'].value_counts())
```

In [ ]:

```
print(df['state'].unique())
```

We will now drop the 4 features.

In [ ]:

```
df.drop(columns=['Timestamp', 'Country', 'state', 'comments'], inplace = True)
```

After dropping the variables, we check our dataset again to view the changes.

In [ ]:

```
df.head()
```

In [ ]:

```
print(df.info())
```

2. Preparing the 'Age' feature

We will first change the 'Age' feature to lowercase to match the rest of the features.

In [ ]:

```
df.rename(columns = {'Age': 'age'}, inplace=True)
```

Similarly, we check our dataset to view the changes.

In [ ]:

```
df.head()
```

A few of the values in 'age' are either too large or too small, which is probably the result of respondents writing random numbers.

In [ ]:

```
print(df['age'].unique())
```

As such, we will remove observations containing such values.

In [ ]:

```
df.drop(df[df['age'] < 18].index, inplace = True)
```

In [ ]:

```
df.drop(df[df['age'] > 100].index, inplace = True)
```

We check the dataset again, and as we can see, the ages make more sense now.

In [ ]:

```
df['age'].unique()
```

We removed 8 rows in the dataset. Previously there were a total of 1259 rows of data.

In [ ]:

```
print(df.info())
```

Due to the large number of varying ages, we will categorize the values into different age groups, and place them into a new column called 'age_group':

0-17: Teen

18-24: Young Adult

25-65: Adult

66-100: Elderly

In [ ]:

```
df['age_group'] = pd.cut(df['age'], [0, 17, 24, 65, 100],
labels = ["Teen", "Young Adult", "Adult", "Elderly"],
                    include_lowest=True)
```

Viewing our changes:

In [ ]:

```
df.columns
```

In [ ]:

```
df['age_group'].value_counts()
```

3. Preparing the 'Gender' feature

We will also change the 'Gender' feature to lowercase to match the rest of the features.

In [ ]:

```
df.rename(columns = {'Gender': 'gender'}, inplace=True)
```

We check the dataset to view our changes. We can now see that every feature is in lowercase.

In [ ]:

```
df.head()
```

There are plenty of random and nonsensical values in this feature, as shown below.

In [ ]:

```
print(df['gender'].unique())
```

We will first replace the values which are a misspelling of 'Male', or mean the same thing, with 'Male'.

In [ ]:

```
df['gender'].replace(['Male', 'Male ', 'male', 'M', 'm', 'Cis Male', 'Man', 'cis
male', 'Mail', 'Male-ish', 'Male (CIS)',
                    'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make'], 'Male', inpla
ce=True)
```

Viewing our changes:

In [ ]:

```
print(df['gender'].unique())
```

We then replace the values for 'Female'

In [ ]:

```
df['gender'].replace(['Female', 'Female ', 'female', 'F', 'f', 'Woman', 'femail'
, 'Cis Female', 'cis-female/femme', 'Femake',
                    'Female(cis)', 'woman'], 'Female', inplace=True)
```

Viewing our changes:

In [ ]:

```
print(df['gender'].unique())
```

We categorize every other gender under 'Others'.

In [ ]:

```
df['gender'].replace(['Trans-female', 'something kinda male?', 'queer/she/
they',
 'non-binary', 'Nah', 'Enby', 'fluid', 'Genderqueer', 'Androgyne', 'Agender',
'Guy (-ish) ^_^', 'male leaning androgynous', 'Trans woman', 'Neuter',
'Female (trans)', 'queer', 'Female (cis)', 'ostensibly male, unsure what th
at really means'], 'Others', inplace=True)
```

Viewing our changes:

In [ ]:

```
print(df['gender'].unique())
```

We can observe that majority of respondents are male.

In [ ]:

```
df['gender'].value_counts()
```

4. Preparing the 'self_employed' feature

For this feature, there are only 1233 entries. Since there are a total of 1251 rows in this dataset (after the previous changes), there are 18 null values.

In [ ]:

```
sum(df['self_employed'].value_counts())
```

In [ ]:

```
sum(df['self_employed'].isnull())
```

In [ ]:

```
df[df['self_employed'].isnull()]
```

We will assume that when the respondent was taking this survey, they skipped this question as they were currently employed by a company. As such, we will fill in the Null values with 'No'.

In [ ]:

```
df['self_employed'].fillna('No', inplace=True)
```

This feature now has 0 Null values:

In [ ]:

```
sum(df['self_employed'].value_counts())
```

In [ ]:

```
sum(df['self_employed'].isnull())
```

5. Preparing the 'work_interfere' feature

This feature has a total of 989 entries, meaning that it has 262 Null values.

In [ ]:

```
sum(df['work_interfere'].value_counts())
```

In [ ]:

```
sum(df['work_interfere'].isnull())
```

In [ ]:

```
df[df['work_interfere'].isnull()]
```

We will assume that the respondent skipped this question as they have never before felt like a mental health condition has disrupted them while working.

In [ ]:

```
df['work_interfere'].fillna('Never', inplace=True)
```

This feature now has 0 Null values:

In [ ]:

```
sum(df['work_interfere'].value_counts())
```

In [ ]:

```
sum(df['work_interfere'].isnull())
```

Final View of Features

In [ ]:

```
print(df.info())
```

Data Exploration

Univariate Analysis

1. Age Distribution of Respondents

In [ ]:

```
sns.set(style='whitegrid')
sns.displot(x='age', kde=True, data=df, height=5, aspect=1.5);
plt.xlabel('Age', labelpad=5)
plt.ylabel('Count', labelpad=5)
plt.title('Age Distribution of Respondents', pad=15)
```

We can see from the above plot that majority of the respondents of the survey are adults around the age of 30, which is unsurprising. In an annual developer survey conducted by Stackoverflow in 2022, 39% of respondents were within the age range of 25 to 34 years old. Despite this dataset being created in 2016, it still reflects similar information.

2. Gender Distribution of Respondents

In [ ]:

```
plt.figure(figsize=(10,3.5)) # Size of the figure
gp = sns.countplot(y='gender', data=df, palette=['violet', 'royalblue', 'seag
reen']);
plt.title('Gender Distribution of Respondents', pad=15)
plt.xlabel('Count', labelpad=10)
gp.set(ylabel=None);



total = len(df['gender'])
for bar in gp.patches:
    percentage = '{:.1f}%'.format(100 * bar.get_width()/total) # Get the bar
width as a percentage value
    x = bar.get_x() + bar.get_width() # Get x-axis position and width of ba
r
    y = bar.get_y() + bar.get_height()/1.75 # Get y-axis position and width
of bar
    gp.annotate(percentage, (x,y), size=9) # Display the percentage values
across all bars (patches)
```

We can see from the above plot that the highest number of respondents are male. This has been the case in the tech industry for several years, where there are a larger number of men as compared to women.

However, in recent years, this has changed. In a [2022 study]() by AnitaB.org, it was found that there were 27.6% of women in the tech industry, which when compared to this dataset (from 2016), is a 7.9% increase.

3. Respondents' Family History of Mental Illness

In [ ]:

```python
# Firstly, for each value in the feature, we need to count how many time
s each one appears
# We do so by first getting all the rows which are 'Yes', then counting th
e number of rows using len()
# We do the same for 'No'
yes = len(df[df['family_history'] == 'Yes'])
no = len(df[df['family_history'] == 'No'])

count = [yes, no]
labels = ['Yes', 'No']
colors = ['lightgrey', 'lightgreen']

# Customizing the pie chart
plt.figure(figsize=(8,4))
explode = (0, 1, 1) # Only the second slice will explode
pc = plt.pie(count, labels=labels, autopct='%1.1f%%', startangle=90, colo
rs=colors)
plt.title('Family History of Mental Illness');
```

From this, we can see that almost 40% of respondents have a family history of mental illness. According to a 2017 study by the Arctic University of Norway, it was discovered that children with parents who had a severe mental illness had up to a 50% chance of developing a mental illness, and a 32% chance of developing a severe mental illness (bipolar disorder, major depressive disorder, schizophrenia, etc). We will look further into this when performing bivariate analysis.

4. Mental Health Resources

In [ ]:

```
df['leave'].value_counts().index
```

In [ ]:

```
plt.figure(figsize=(8,5)) # Size of the figure

# Using value_counts(), we get the count of each answer in descending o
rder, we then use .index to get an Index object, which
# we later pass into the order parameter of the countplot, sorting the plo
t in descending order
order = df['leave'].value_counts().index

plt.title('Taking Leave for Mental Health Issue', pad=15);
mp = sns.countplot(x='leave', data=df, order=order, palette='gist_heat')
plt.ylabel('Count', labelpad=10)
mp.set(xlabel=None);
```

From the above plot, we can see that most respondents do not know whether they are even allowed to take leave for a mental health issue, and there are also quite a number who find it hard to do so, which may be due to the social stigma surrounding mental issues. Companies should learn to change the way they handle mental health, especially due to the prevalence of mental health issues as a result of the COVID-19 pandemic. This 2022 study has shown that during the quarantine period, increased stress levels resulted in depressive symptoms and anxiety disorders amongst employees. It was also reported that even after the pandemic, there were individuals who suffered from post-traumatic stress disorder (PTSD).

Bivariate Analysis

1. Relationship between Family History and Treatment

In [ ]:

```
sns.countplot(x='family_history', data=df, hue='treatment', palette=['red
', 'gray'])
leg = plt.legend(loc='best', title='Seek Treatment')
leg._legend_box.align = "left"
plt.xlabel('Family History of Mental Illness', labelpad=10)
plt.ylabel('Count', labelpad=10)
plt.title('Relationship between Family History and Treatment', pad=15);
```

As mentioned in the 4th plot of univariate analysis, people with a family history of mental illness have a higher chance of developing mental illnesses in their lifetime. As shown in this plot, this proves true, as we can see that there are a higher number of people who have a family history of mental illness and have seeked treatment.

2. Relationship between Age and Treatment

In [ ]:

```
sns.violinplot(x='treatment', y='age', data=df, palette='Set1')
plt.title('Relationship between Age and Treatment', pad=15);
plt.xlabel('Seeked Treatment', labelpad=10)
plt.ylabel('Age', labelpad=10)
```

As seen in the 1st plot of univariate analysis, most of the people working in tech are around the age of 30 years old. Based on the size of the 'violins', we can see that there is only a slightly smaller amount of people who have not seeked treatment. There are also respondents of an older age (70 and above) who have seeked treatment for mental illness. According to a 2021 national survey conducted by the Substance Abuse and Mental Health Services Administration (SAMHSA), adults aged 18 to 25 years old had the highest prevalence of mental illnesses (33.7%), as compared to that of adults aged 26 to 49 years old (28.1%).

3. Relationship between Anonymity and Treatment

In [ ]:

```
# order = df['work_interfere'].value_counts().index
```

```
sns.countplot(x='obs_consequence', hue='treatment', data=df, palette='S
et1')
leg = plt.legend(loc='best', title='Seek Treatment')
leg._legend_box.align = "left"
plt.xlabel('Observed Consequences', labelpad=10)
plt.ylabel('Count', labelpad=10);
plt.title('Relationship between Anonymity and Treatment', pad=15)
```

Unfortunately, there are actually quite a number of respondents who have observed negative consequences for coworkers with mental health conditions. Even so, there are also many who have placed a greater importance on their health, and seeked treatment for their mental health conditions.

4. Relationship between Importance of Mental Health (mental_vs_physical) and Leave

In [ ]:

```
plt.figure(figsize=(8,5)) # Size of the figure
mvp = df[((df['mental_vs_physical'] == 'Yes') | (df['mental_vs_physical']
== 'No')) & (df['leave'] != "Don't know")]['leave']
test = df[((df['mental_vs_physical'] == 'Yes') | (df['mental_vs_physical']
== 'No')) & (df['leave'] != "Don't know")]['mental_vs_physical']

order = df[((df['mental_vs_physical'] == 'Yes') | (df['mental_vs_physical']
== 'No')) & (df['leave'] != "Don't know")]['leave'].value_counts().index
sns.countplot(y=mvp, data=df, order=order, hue=test, palette=['green', 'r
ed'])
```

```
plt.xlabel('Count', labelpad=10)
plt.ylabel('Taking Leave for Mental Health', labelpad=20)
plt.title('Relationship between mental_vs_physical and Leave', pad=15)

leg = plt.legend(loc='best', title='Mental Health Important')
leg._legend_box.align = "center"
```

This plot is very interesting. We can clearly see that for companies which place a higher importance on mental health, it is easier for employees to take leave for their mental health. Whereas for companies that do not place such an importantfor mental health, it is hard for its employees to take leave for their mental health. Companies should learn to place a higher importance on the mental health of their employees, as it affects their personal well-being at work, and may even affect their productivity.

5. Relationship between Number of Employees and Observed Consequences

In [ ]:

```
plt.figure(figsize=(10,5)) # Size of the figure
order = ['1-5', '6-25', '26-100', '100-500', '500-1000', 'More than 1000']
ax = sns.countplot(x='no_employees', hue='coworkers', data=df, order=o
rder, palette=['dodgerblue', 'maroon', 'limegreen'])
sns.move_legend(ax, "upper left", bbox_to_anchor=(1, 1))
plt.xlabel('Number of Employees', labelpad=10)
plt.ylabel('Count', labelpad=10);
plt.title('Relationship between Number of Employees and Observed Con
sequences', pad=15);
```

This plot tells us that the bigger the company (in terms of number of employees), the less likely an employee is to discuss about a mental health issue with their coworkers. It might be due to the fact that the employees in smaller companies are closer to one another, or that they might be more open with each other. Larger companies may have a strict work culture, and employees who feel restricted by them may not want to share about their personal issues with others.

Feature Engineering:

Feature engineering is a critical aspect of data analytics and machine learning in public health awareness campaigns. It involves creating new features or modifying existing ones from your raw data to improve the performance of your predictive models and provide more valuable insights for your campaign. Here are some feature engineering techniques that can be applied to public health data:

Temporal Features:

Extract time-based features like day of the week, month, season, or year. This can help identify patterns and trends related to public health issues.

Aggregation and Summary Statistics:

Compute summary statistics (mean, median, mode, variance, etc.) for relevant variables within specific time frames or geographical regions. For instance, calculate the average number of cases per month in a particular region.

Lagged Variables:

Create lagged features, such as the number of cases in the previous month, to account for time dependencies and trends.

Geospatial Features: Utilize geographical information to create features like proximity to healthcare facilities, population density, or socioeconomic indicators for specific regions. Geospatial data can be crucial in public health campaigns targeting specific areas.

Categorical Variable Encoding:

Convert categorical variables into numerical representations through techniques like one-hot encoding, label encoding, or embeddings. This enables machine learning algorithms to work with categorical data.

Text and NLP Features:

If your data includes textual information, you can perform natural language processing (NLP) and extract features like sentiment, key phrases, or word frequency to gain insights into public sentiment and awareness related to the health issue.

Interaction and Polynomial Features:

Create interaction features by multiplying or dividing two or more relevant variables. Additionally, introduce polynomial features to capture non-linear relationships in the data.

Domain-Specific Features:

Incorporate features that are specific to the public health issue you're addressing. For example, if working on a campaign related to disease outbreaks, you might create features related to the pathogen's characteristics or the availability of vaccines.

Feature Scaling:

Normalize or scale features to bring them to a common range. Scaling ensures that no single feature dominates the model's learning process.

Missing Data Handling:

Create binary indicators to flag missing data, which can sometimes be informative. This allows the model to distinguish between available and missing values.

Feature Selection:

Employ techniques like recursive feature elimination or feature importance scores to select the most relevant features, especially if you have a high-dimensional dataset.

Feature Extraction:

Use dimensionality reduction techniques like Principal Component Analysis (PCA) or t-SNE to extract meaningful features from high-dimensional data.

Feature Crosses:

Combine two or more features to capture interactions. For example, multiplying age and income might reveal a significant feature if your campaign relates to socioeconomic factors.

Health Indicators and Composite Features:

Calculate health-related indices or composite features, such as BMI, which can be valuable in public health campaigns focusing on nutrition and fitness.

Feature engineering is an iterative process that involves experimenting with different transformations and selecting the most informative features for your specific campaign. The goal is to make your data more amenable to machine learning and provide actionable insights that drive your public health awareness efforts.

Model Training:

Training a public health awareness campaign involves several key steps and considerations. Below is a general guide on how to train a model for a public health awareness campaign:

1. **Define Objectives and Goals**:
    - Clearly define the objectives of your public health awareness campaign. What do you want to achieve? Who is your target audience? What behavior change or awareness are you aiming for?
2. **Gather Data**:
    - Collect data relevant to your campaign. This may include health statistics, demographic information, and historical campaign data. You'll need this data to train your model effectively.
3. **Choose a Machine Learning Model**:
    - Depending on your campaign's goals, you may choose different machine learning models. For instance, for predictive modeling, you might use regression or time series analysis. For classification tasks, you might use decision trees, random forests, or deep learning models.
4. **Feature Engineering**:
    - Prepare your data by selecting relevant features (variables) that are likely to influence the outcome of your campaign.

This can involve data cleaning, transformation, and normalization.

5. **Data Splitting**:

    - Split your data into training, validation, and test sets. The training set is used to train the model, the validation set is used for hyperparameter tuning, and the test set is used to evaluate model performance.

6. **Model Training**: Train your chosen machine learning model on the training data. This involves feeding the model input data and target labels (if supervised learning) and adjusting model parameters to minimize the error.

7. **Hyperparameter Tuning**:

    - Optimize the hyperparameters of your model to improve its performance. You can use techniques like cross-validation and grid search.

.Evaluation:

- Evaluate your model's performance using the validation and test datasets. Common evaluation metrics for public health campaigns may include accuracy, precision, recall, F1 score, or area under the ROC curve, depending on the nature of your campaign.

8. **Interpretation**:
   - Understand the insights gained from the model. What factors influence public health behavior or awareness? This can help you fine-tune your campaign strategy.
9. **Deployment**:
   - Deploy your model in a real-world setting. This might involve integrating it with a website, mobile app, or other digital platforms where you can reach your target audience.
10. **Monitoring and Feedback**:
    - Continuously monitor the performance of your campaign. Gather feedback from the target audience and the community to make necessary adjustments and improvements.
11. **Ethical Considerations**:
    - Be aware of ethical considerations in using data and machine learning for public health. Ensure that your campaign respects privacy, is fair, and doesn't discriminate against any group.
12. **Adaptation**:
    - Public health is dynamic, and awareness campaigns may need to adapt to changing circumstances, new information,

or emerging health issues. Be ready to update your model and campaign as needed.

13. **Collaboration**:

    - Collaboration with public health experts, community leaders, and organizations is often crucial for the success of public health campaigns. Work closely with these stakeholders to ensure your campaign aligns with expert guidance and community needs.

Remember that public health campaigns may involve complex social and behavioral factors, so machine learning models are just one tool among many in your toolkit for addressing public health challenges. Combining data-driven insights with community engagement and expert knowledge is often the most effective approach.