Public Health Awareness Campaign Analysis

Duration of preparation of analysis: 01-10-2023 to 30-10-2023

TEAM NAME :D²G²K

Overview

PROJECT OBJECTIVE

- ✓ **Identifying trends and patterns in public health awareness campaigns conducted.**
- ✓ **To provide better insights which can help government and organizations to improvise the quality of campaigns conducted toimplement an effective health awareness campaigns.**

DESIGN THINKING

Design thinking is a human-centered approach that involves:

- **Understanding users**
- **Challenging assumptions**
- **Redefining problems**
- **Creating innovative solutions**
- **Prototyping and testing solutions**
- **Iterating to improve solutions**

DESIGN PROCESS

1 COLLECT

In this step appropriate dataset has to be collected for the data analysis ,hence mental health prediction datset is collected for analysis

2 DATA PROCESSING

The mental health prediction dataset may contain raw data which has to be processed,to work further with it

3 CLEAN THE DATA

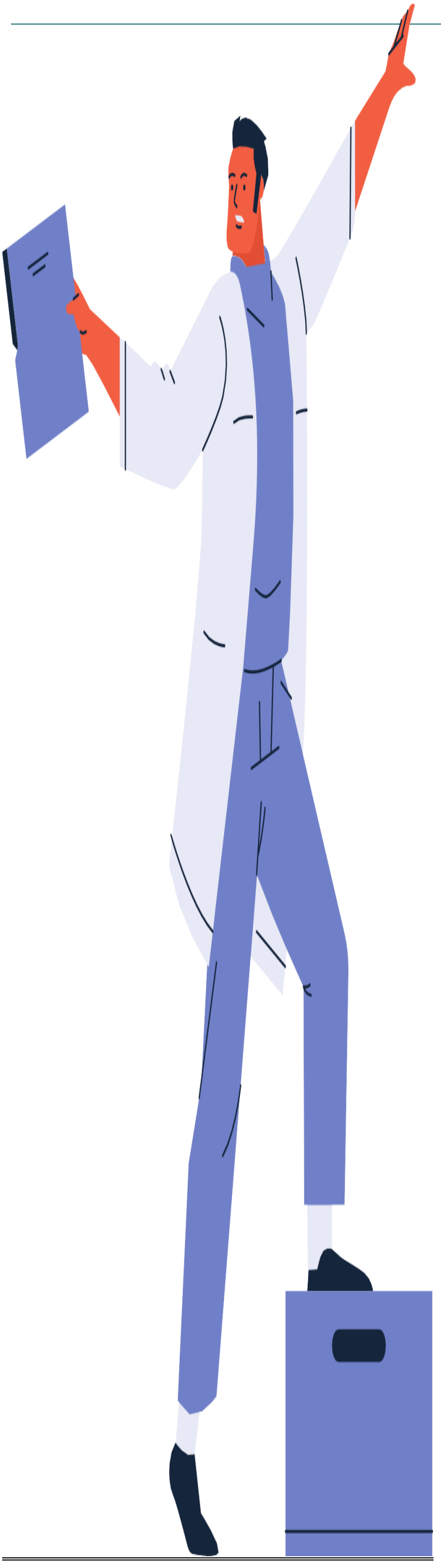
The raw data contains missing values and several error persist in the data which has to be cleansed to analyze the data of mental health prediction

4 DATA VISUALIZATION

This is the most important step in which we have to create visuals for public health awareness campaign using exploratory data analysis.

5 BUILDING THE PROJECT AND REPORTING

The project once after going through under all the procedures it has to be concluded by deducing the results from the mental health predication to estimate and adversely increase the performance of public health through campaigns



CODE INTEGRATION

1

DATA PROCESSING

The raw data has to be processed for anaysis of mental health prediction.

2

DATA CLEANING

The raw data may be prone to error hence it has to be cleansed .

3

DATA ANALYSIS

All the data has to be clearly analyzed in order to obtain and evaluate the performace of mental health prediction dataset

4

COMMUNICATION

This can be achieved by insights ,patterns and trends obtained from the mental health predcation dataset

DEVELOPMENT OF ANALYSIS

Procedure

Jupyter Notebook Code Implementation

Get hands-on experience with Jupyter Notebook to implement different aspects of the campaign.

Data Extraction in Notebook

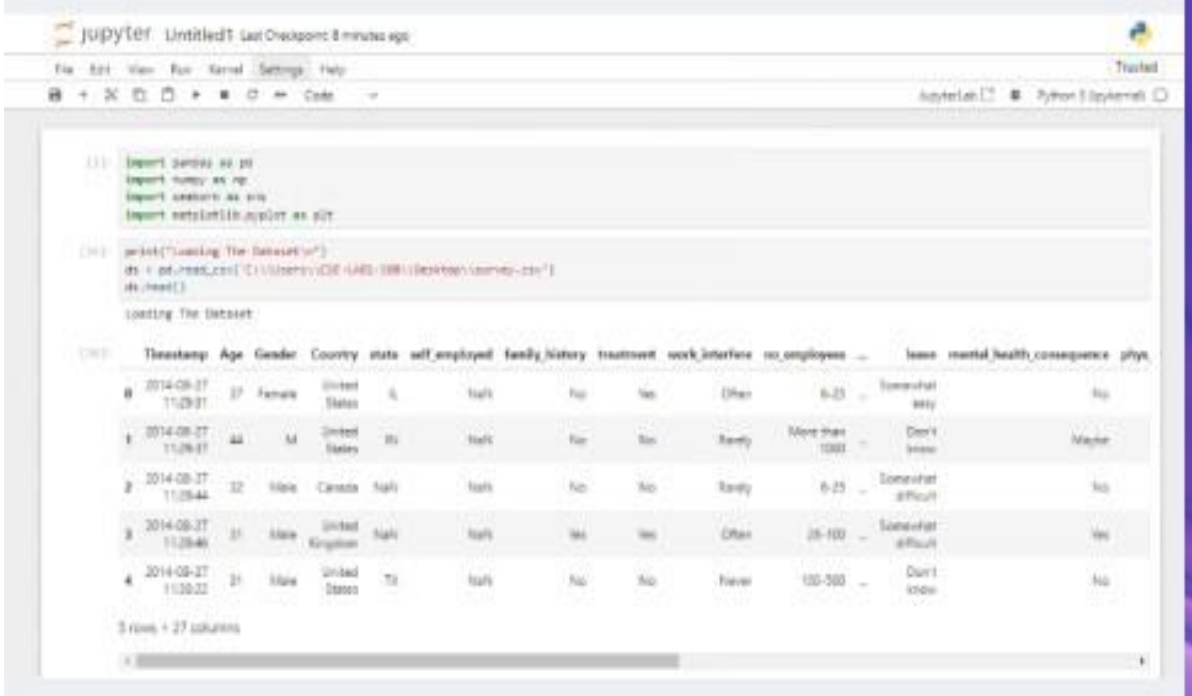
Learn how to effectively extract data using Jupyter Notebook for our campaign.

Machine Learning Algorithms

Discover the power of machine learning algorithms like Random Forest, K Classifier, CNN, KNN, and Gradient Descent for enhancing our campaign.

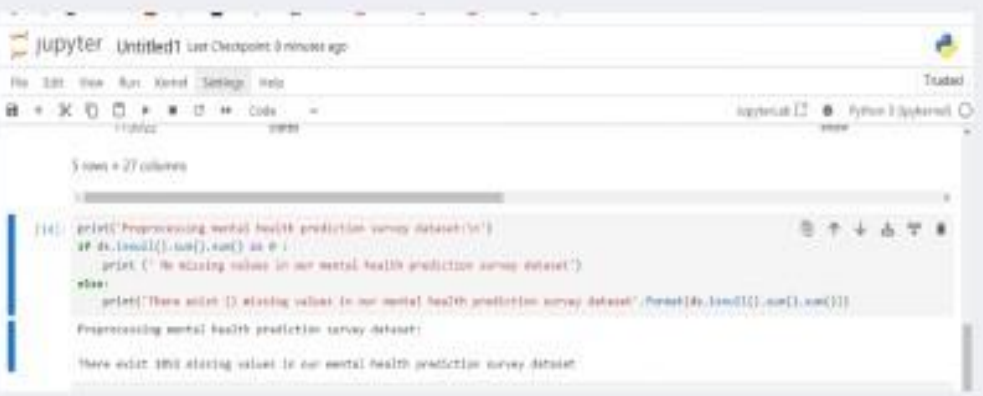
Loading the dataset

Before embarking on any analysis, it's crucial to load the public health awareness campaign dataset. By properly preparing the data, we can ensure reliable results and meaningful insights.



Preprocessing the dataset

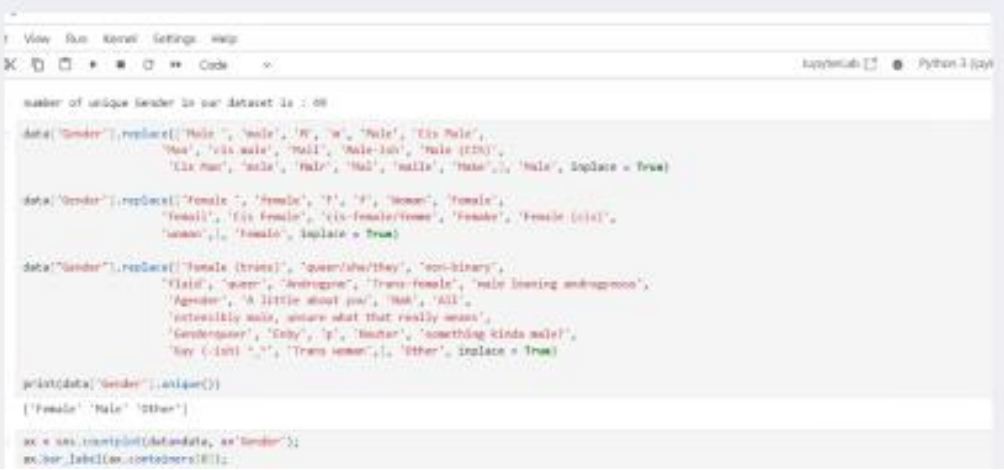
Data preprocessing is a fundamental stage in data mining that involves transforming raw data into a usable format for a machine learning model



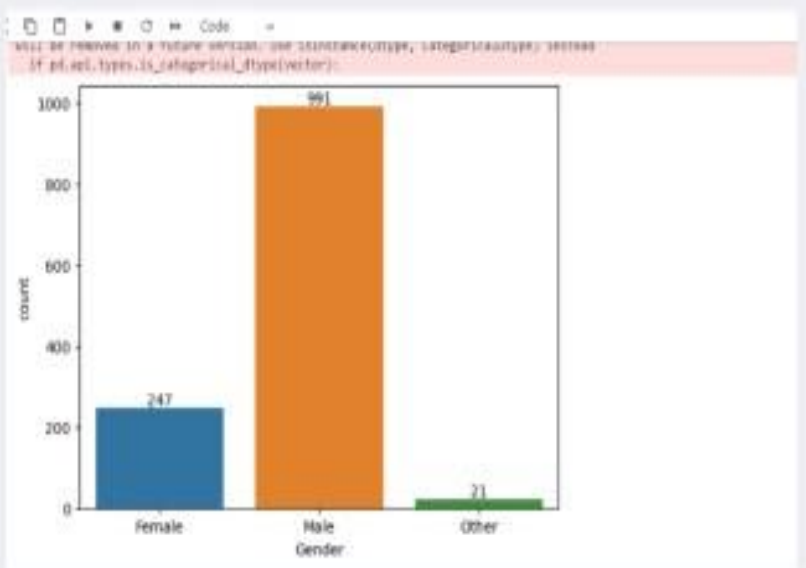
Data cleansing

Data is collected from survey dataset provided by Ibm data analytics link. Finding the unique categories of gender in order to remove the unwanted categories

Now cleaning:



After cleansing Insight can be obtained as



Working on the error data to ressolve it

```
there is 4 duplicated data:
--------------------------------------------------
0
```

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

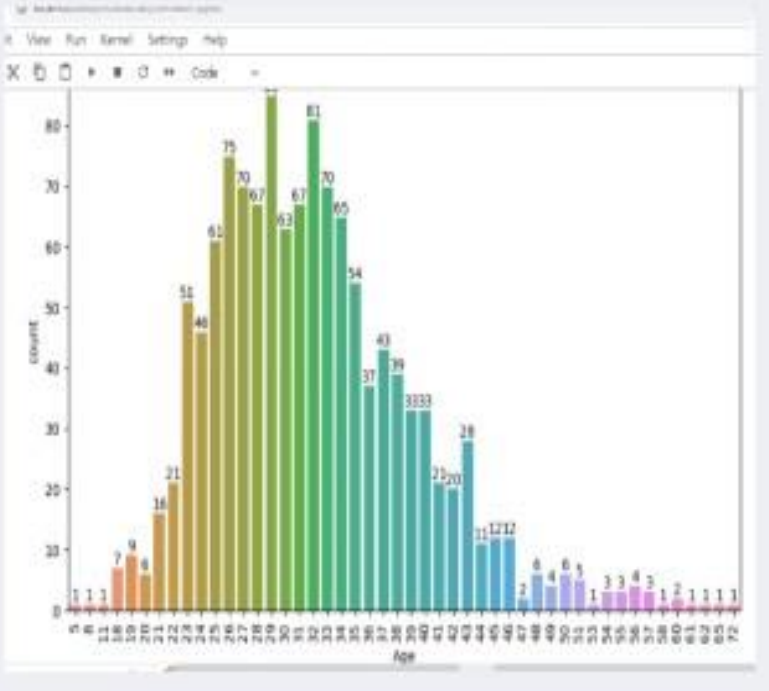
```
array([37, 44, 32, 31, 33, 35, 39, 42, 23, 29, 36, 27, 40, 41, 34, 30, 46, 50, 24, 18, 28, 26, 22, 19, 25, 45, 21, 49, 56, 60, 54, 55, 48,
       20, 57, 58, 47, 62, 51, 65, 64,  9, 53, 61,  8, 11, 73],
      dtype=int64)
```

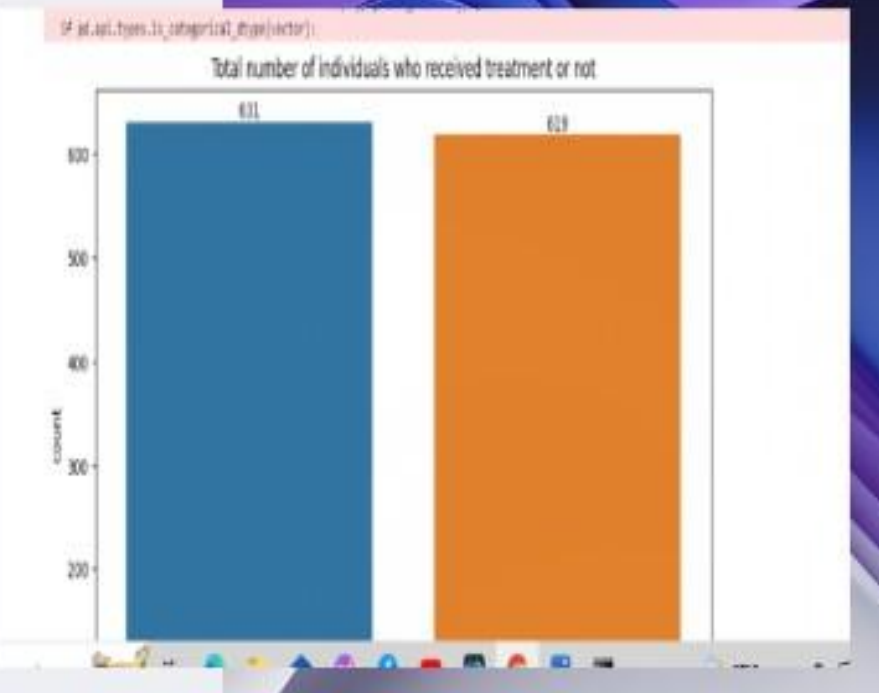
```
data.drop(data[data['Age']<0].index, inplace = True)
data.drop(data[data['Age']>99].index, inplace = True)

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

```
[37 44 32 31 33 35 39 42 23 29 36 27 40 41 34 30 46 50 24 18 28 26 22
 19 25 45 21 49 56 60 54 55 48 20 57 58 47 62 51 65 64  9 53 61  8 11 73]
```

```
plt.figure(figsize = (30,6))
age_range_plot = sns.countplot(data = data, x = 'Age');
age_range_plot.bar_label(age_range_plot.containers[0]);
plt.xticks(rotation=90);
```

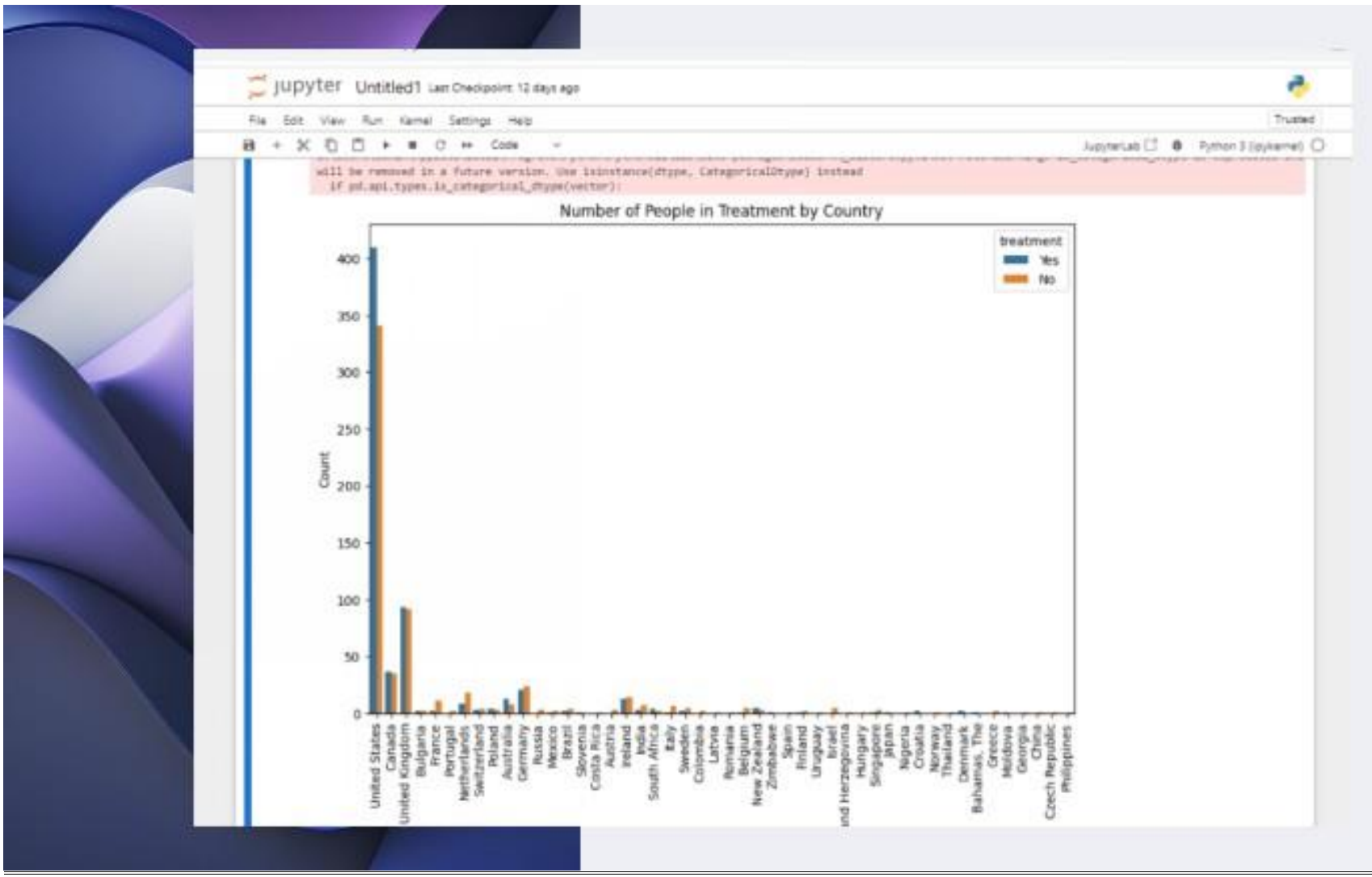
After resolving insights obtained can be





Exploring Mental Health Prediction Data

In addition to the public health awareness campaign analysis, we'll look into mental health prediction datasets. By combining these two domains, we can uncover fascinating insights and potential correlations.



Analyzing Mental Health Prediction Data

By applying advanced analytical techniques to the mental health prediction datasets, we'll unravel key findings and observations. Prepare to be surprised by the interplay between public health awareness and mental health.

Data Collection

The data has to be collected for analysis of public health awareness campaigns, The datasets have to be reliable and valid for futheranalysis and research.

Gather historical data on past campaigns. This data should include various features like campaign type, target audience, messaging,channels used, timing, and the outcome (e.g., conversion rate, ROI).

Deliverables

Deliverable 1:
Statistical analysis report on the public health awareness campaigns

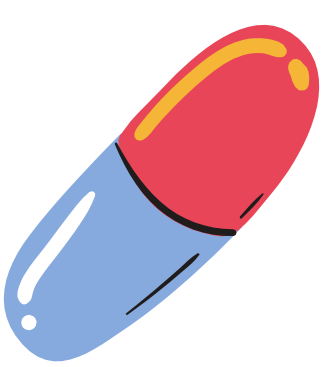
Deliverable Description: It can be done by IBM Cognos Analytics by creating data visualization.

Deliverable 2:
Insights and patterns observed from the data.

Deliverable Description: Based upon the mental health prediction dataset obtain patterns and trends for the analysis andunderstanding the campaign effectiveness.

Deliverable 3:
Providing solution to sort out the problem arises by specifying the essential ways

Deliverable Description: By the analysis solution can be found by applying few methods.



Data Visualization With IBM COGNOS

Continue Building the Analysis

1. **Create Visualizations**

   Utilize IBM Cognos to transform raw data into insightful visualizations that showcase campaign performance.

2. **Integrate Code**

   Enhance your analysis by integrating Python to perform advanced data analysis and generate valuable insights.

3. **Enhance Campaign Analysis**

   With IBM Cognos and Python, take your analysis to the next level by calculating engagement rates and conducting demographic analysis.

Design Engaging Dashboards and Reports

Visualize Campaign Reach

Design dynamic dashboards that provide a comprehensive view of your public health campaign's reach and audience engagement.

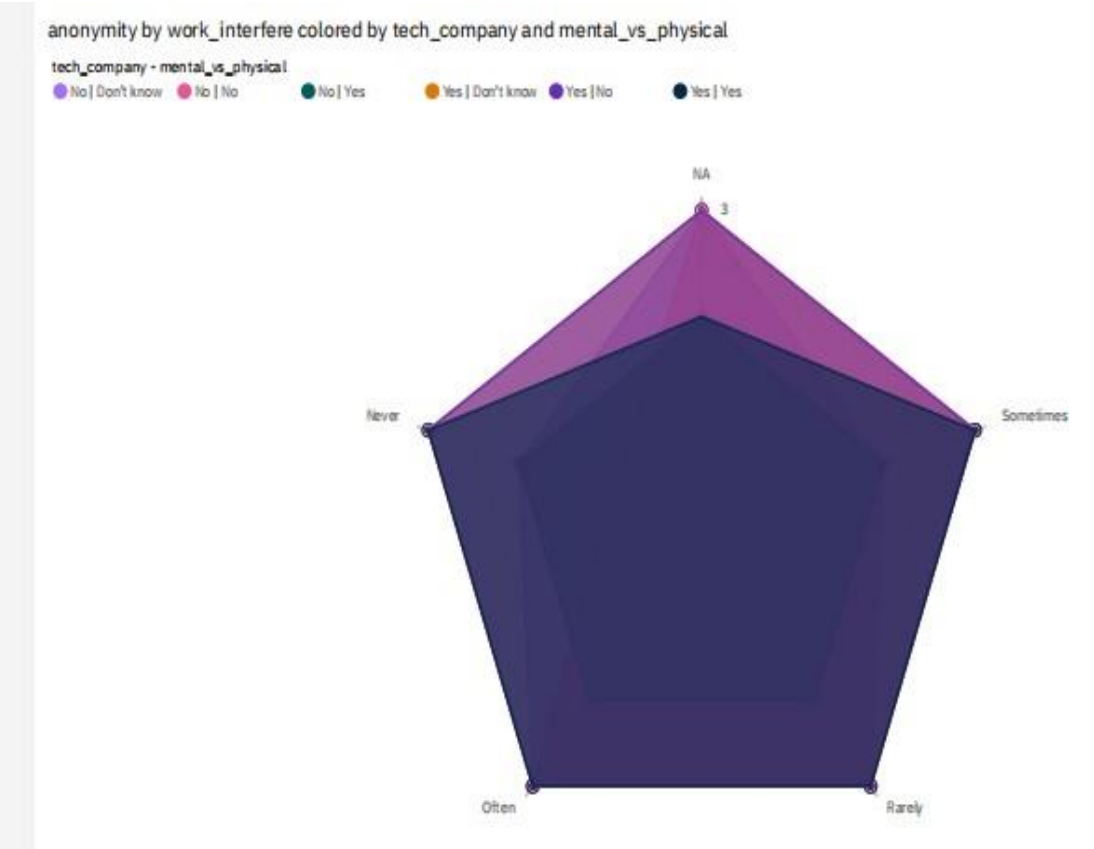
Elevate Awareness Levels

Create impactful reports that capture the effectiveness of your awareness initiatives and drive increased awareness among the target audience.

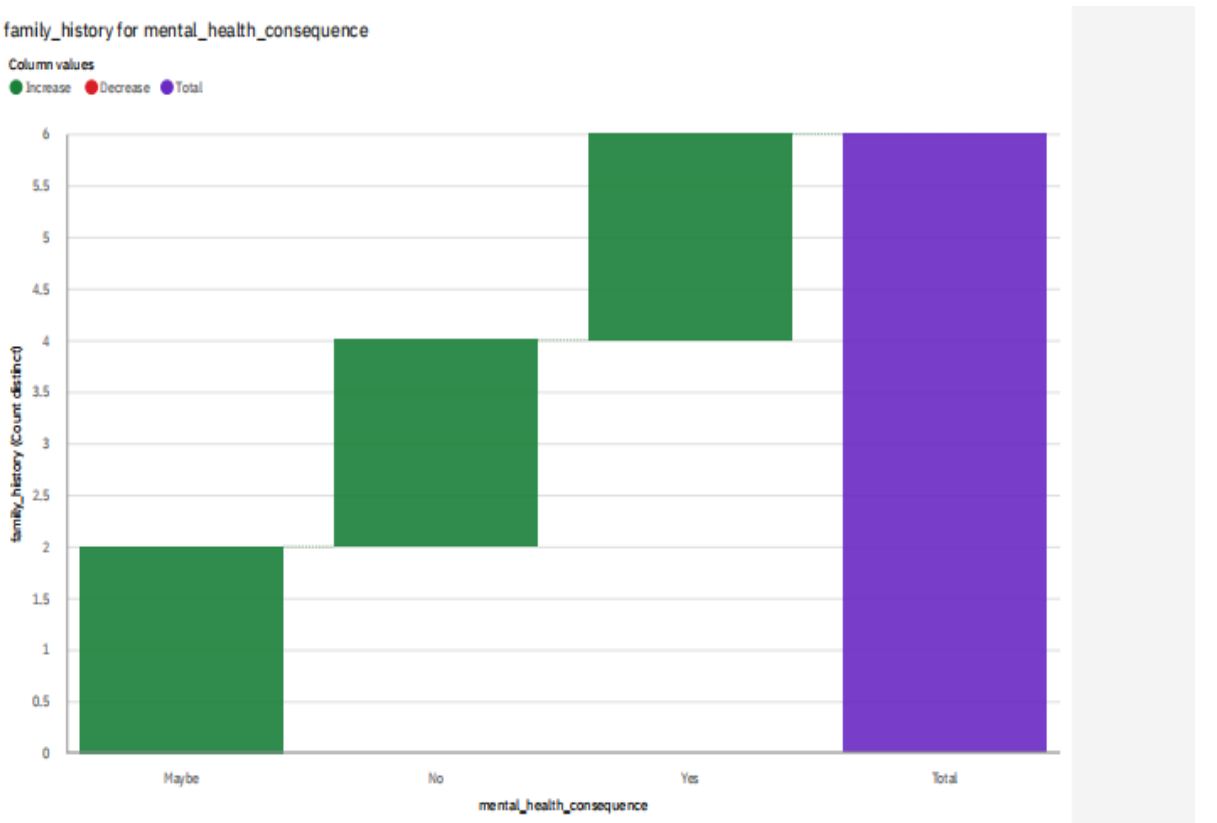
Measure Impact Metrics

Track key impact metrics through visually compelling dashboards that empower stakeholders to make data-driven decisions.

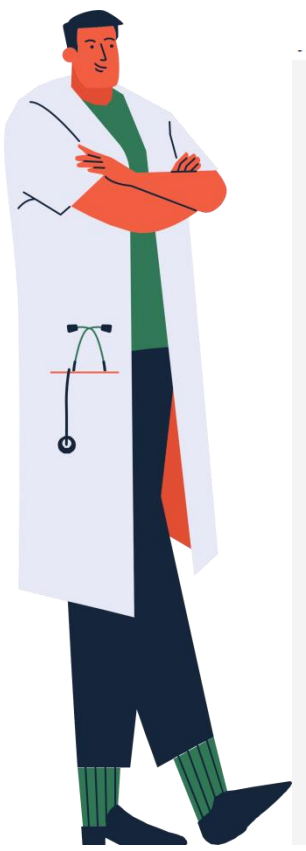
Anonymity by work-interfere for tech company and mental vs physical teach company

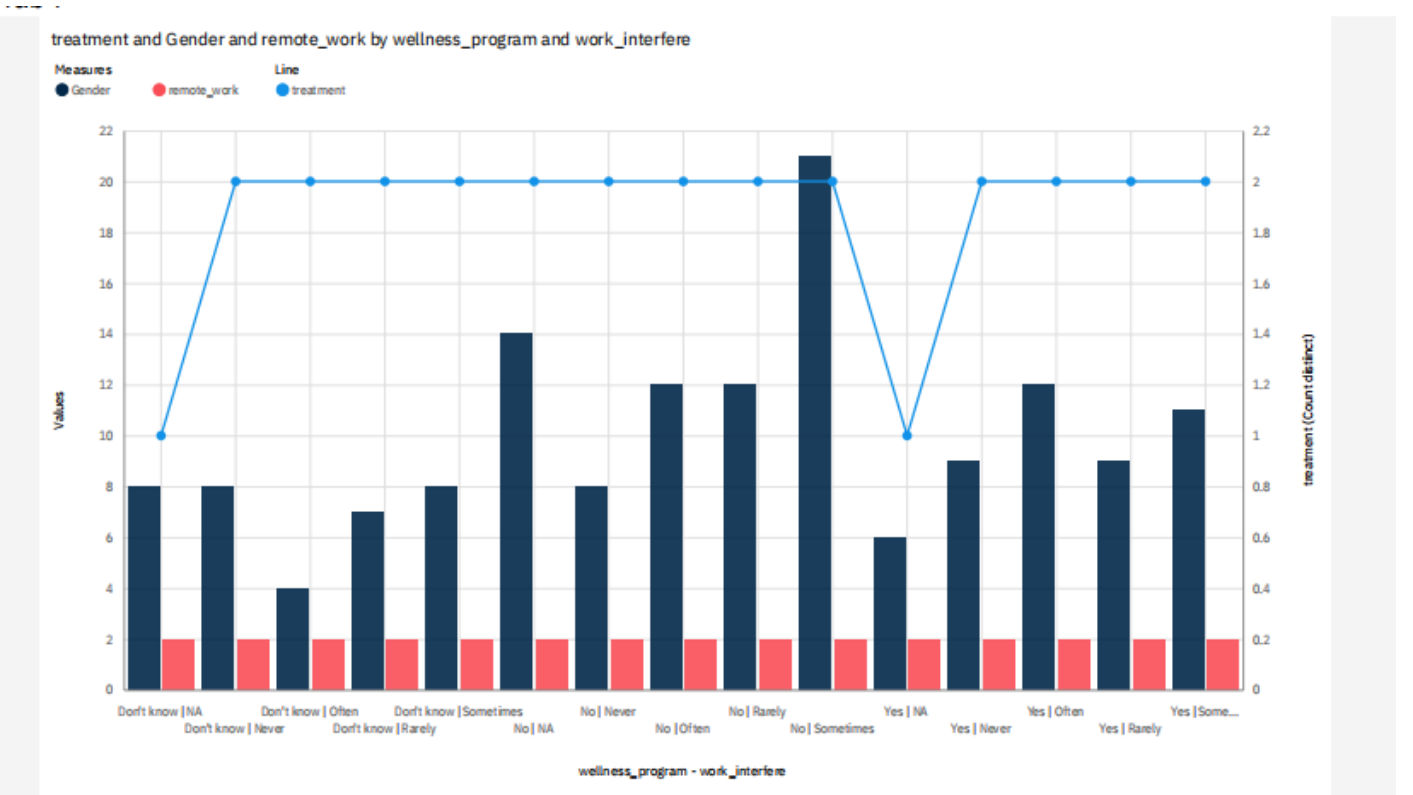


Family history for Mental Health sequence



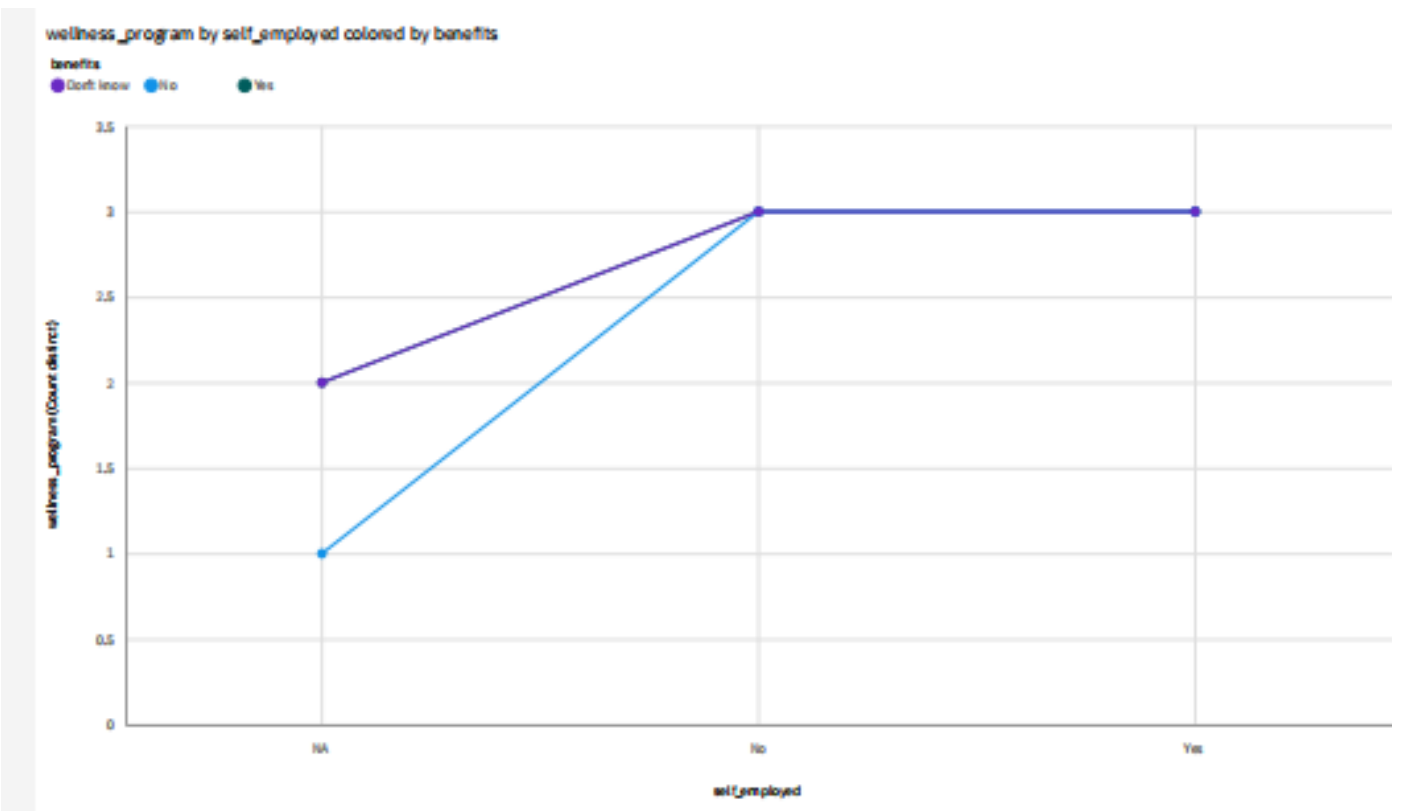
Treatment,Gender and remote work by wellness program and work interfere



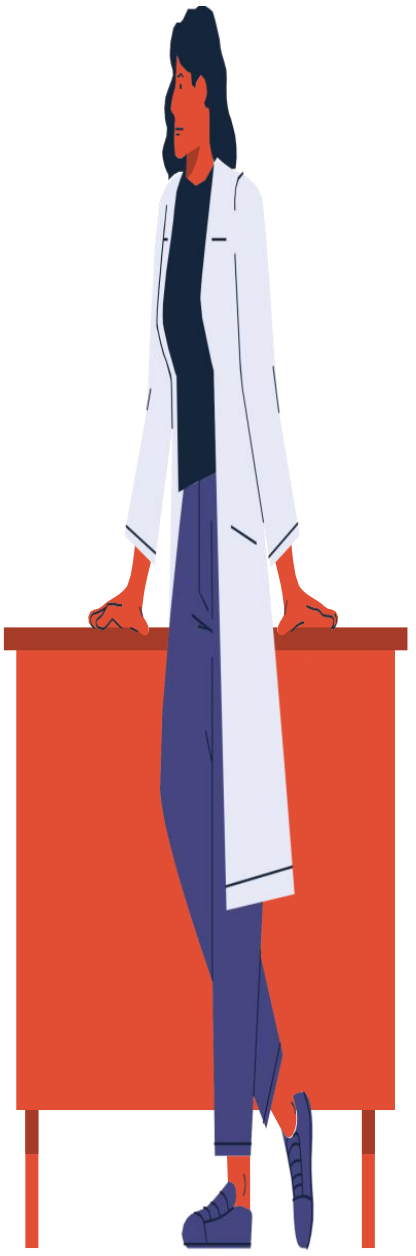


z

Wellness program by self employees by benefits

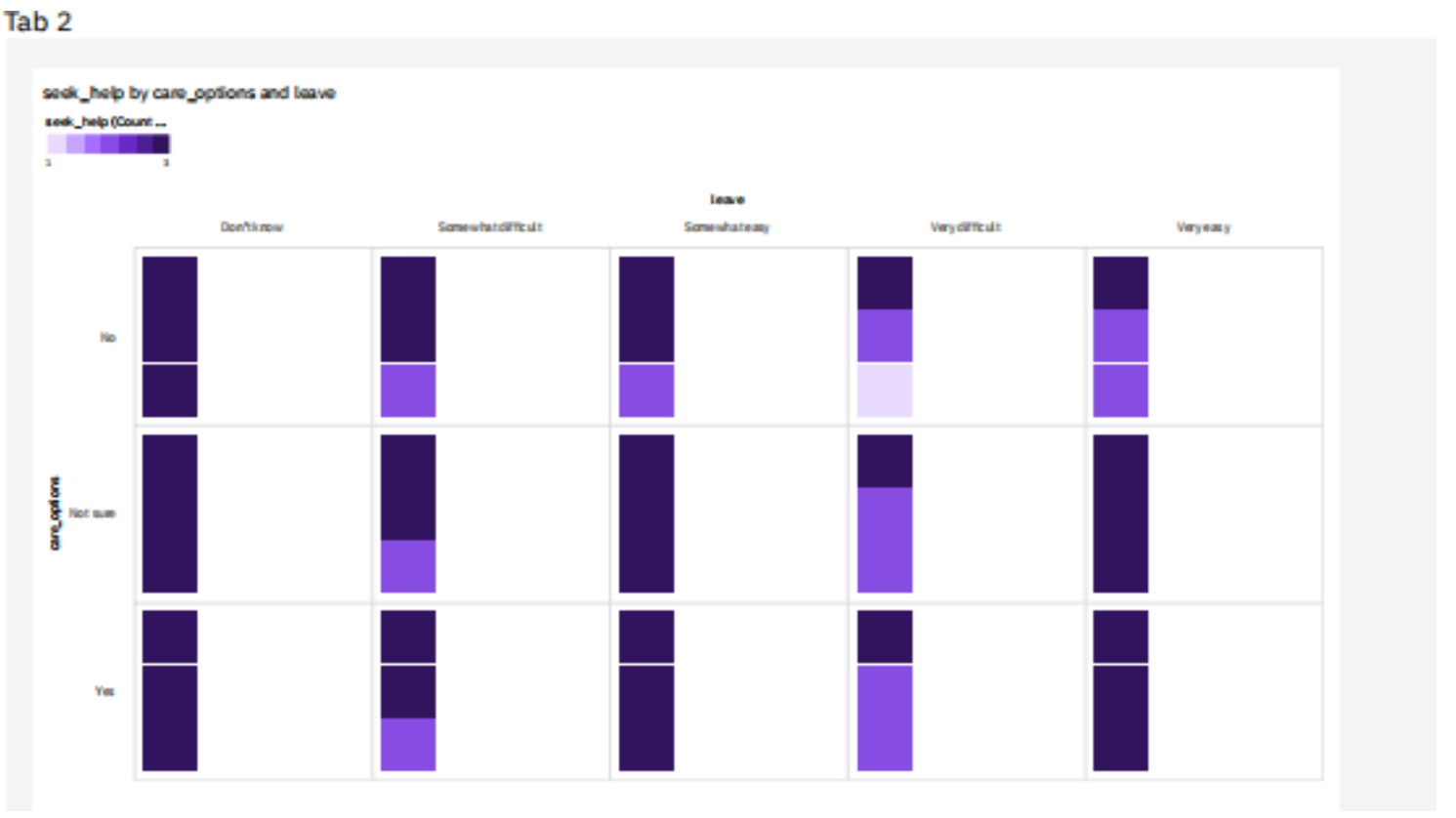


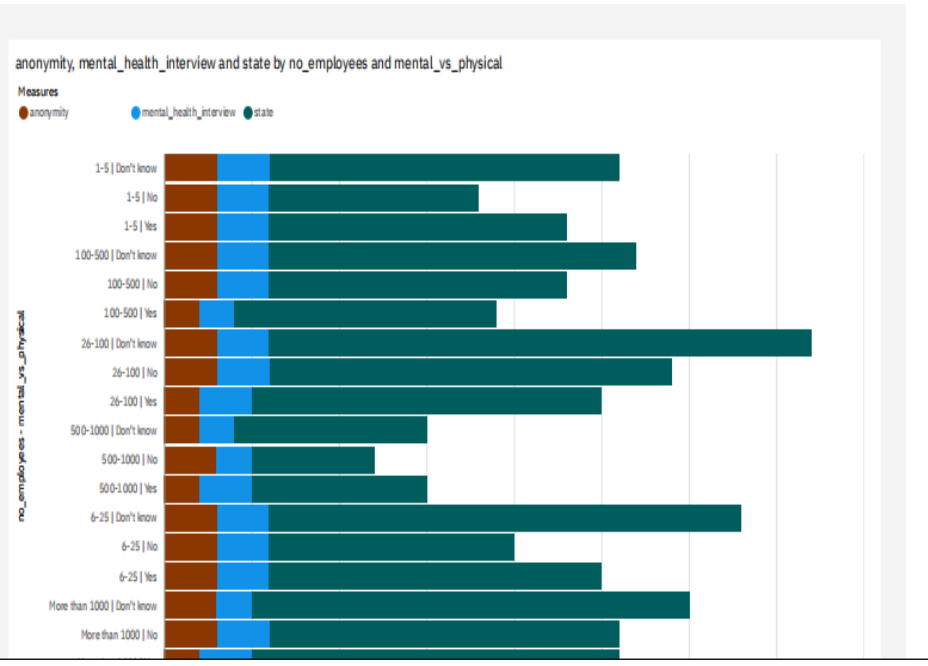
Anonymi Coworkers hierarchy by self employed and sized by wellness program



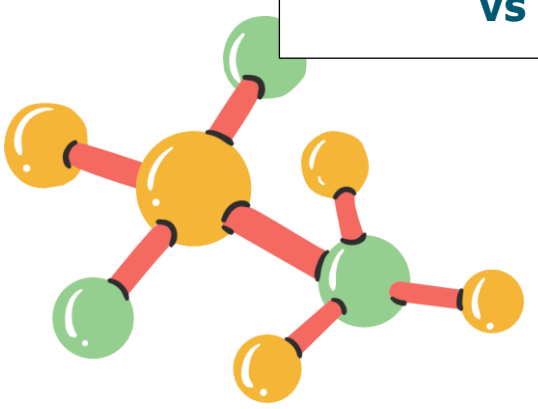
Anonymity

Anonymity by work-interfere for tech company and mental vs physical teach company





Anonymity by work-interfere for tech company and mental vs physical teach company



Insights on Measure Campaign Effectiveness

Measuring the effectiveness of a marketing campaign is crucial to assess its impact and determine whether it achieved its goals. To gainvaluable insights into the campaign's effectiveness, consider the following key metrics and methods:

Key Performance Indicators (KPIs):

- Define clear, measurable goals and KPIs before launching the campaign. KPIs can include sales, website tramp, conversion rates,leads generated, or brand awareness.

Conversion Rates:

- Track the conversion rate from different stages of your campaign, such as click-through rates, sign-up rates, and purchase rates.This provides insights into how well your campaign is performing at each step of the customer journey.

Return on Investment (ROI):Calculate the ROI by comparing the campaign's cost to the revenue generated as a result. This is a fundamental metric forassessing the financial success of the campaign

Customer Acquisition Cost (CAC):

Measure how much it costs to acquire a new customer through the campaign. This helps determine the campaign's emciency inacquiring new business.

Customer Lifetime Value (CLV):

Assess the long-term value of customers acquired through the campaign. A high CLV can justify higher acquisition costs.

Attribution Models:

Use attribution models to understand how different touchpoints contribute to conversions. Multi-touch attribution models canhelp

- Measure the campaign's effect on customer retention and loyalty. It's often cheaper to retain existing customers than acquirenew ones.

Offline Metrics:

- If applicable, measure the campaign's impact on in-store tramc, phone inquiries, or other offline interactions.

Customer Feedback and Complaints:

- Pay attention to customer feedback and complaints related to the campaign to identify areas for improvement.

Post-Campaign Analysis:

- After the campaign, conduct a thorough post-mortem analysis to review what worked, what didn't, and what lessons can beapplied to future campaigns.

Website Analytics:

- Analyze website data to track visitor behavior, page views, bounce rates, and conversion paths. Tools like Google Analytics canprovide valuable insights.

A/B Testing:

- Conduct A/B tests to compare different campaign elements (e.g., headlines, images, CTA buttons) to determine which versionsperform best.

Social Media Engagement:

- Monitor likes, shares, comments, and other engagement metrics on social media platforms to gauge the campaign's impact onsocial channels.

Email Marketing Metrics:

- For email campaigns, track open rates, click-through rates, conversion rates, and unsubscribe rates to measure email campaigneffectiveness.

Customer Surveys:

- Collect feedback from customers to gauge their satisfaction and whether the campaign influenced their decision-making.

Heatmaps and User Session Recordings:

- Tools like Hotjar or Crazy Egg can provide insights into how users interact with your website, helping you identify areas forimprovement.

Customer Segmentation:

- Analyze the campaign's impact on different customer segments to understand which groups responded most positively.

Brand Tracking:

- Monitor brand metrics like brand awareness, perception, and sentiment before and after the campaign to assess its impact onbrand equity.

Competitive Analysis:

- Compare your campaign performance to that of your competitors to understand your position in the market.

Social Listening:

- Use social listening tools to monitor online conversations and sentiment related to your campaign or brand.

Insights on Future Guide Strategies

Remember that the choice of metrics and methods for measuring campaign effectiveness will depend on your specific goals and the
nature of your campaign. Be prepared to adapt your measurement approach based on the campaign's objectives and the channels used.Regularly tracking and analyzing these metrics will help you refine your marketing strategies and improve future campaigns

Planning future campaign strategies requires a combination of historical data analysis, market research, and a forward-lookingapproach. Here are some insights and strategies to guide your future campaigns:

Data-Driven Decision Making:

- Continue to collect and analyze data from your past campaigns to identify trends and patterns. Use this data to inform yourfuture strategies.

Customer Segmentation:

- Segment your target audience based on demographics, behavior, and preferences. This allows for more personalized andeffective campaigns.

Multi-Channel Marketing:

- Utilize multiple marketing channels to reach your audience where they are. This may include email, social media, contentmarketing, SEO, and paid advertising.

Content Marketing:

- Develop valuable, relevant, and high-quality content to engage your audience. Content can include blog posts, videos, infographics, and more.

Marketing Automation:

- Implement marketing automation tools to streamline repetitive tasks, nurture leads, and improve overall emciency.

Personalization:

- Leverage personalization techniques to tailor your marketing messages to individual customers. Personalized campaigns tend toperform better.

A/B Testing:

- Continue to A/B test various elements of your campaigns, such as headlines, CTAs, and visuals, to optimize your messaging.

Customer Journey Mapping:

- Map out the customer journey to understand the touchpoints and interactions customers have with your brand. Use this insightto create more effective campaigns.

Mobile Optimization:

- Ensure your campaigns are optimized for mobile devices, as mobile usage continues to grow.

User-Generated Content:

- Encourage your customers to create and share content about your products or services. User-generated content can be highlyinfluential.

Social Media Engagement:

- Stay active on social media platforms where your audience is present. Engage with your audience, respond to comments, andcreate shareable content.

Influencer Marketing:

- Collaborate with influencers in your industry who can promote your products or services to their followers.

SEO and SEM:

- Invest in search engine optimization (SEO) and search engine marketing (SEM) to ensure your website ranks well on searchengines and captures organic and paid transaction.

Customer Feedback:

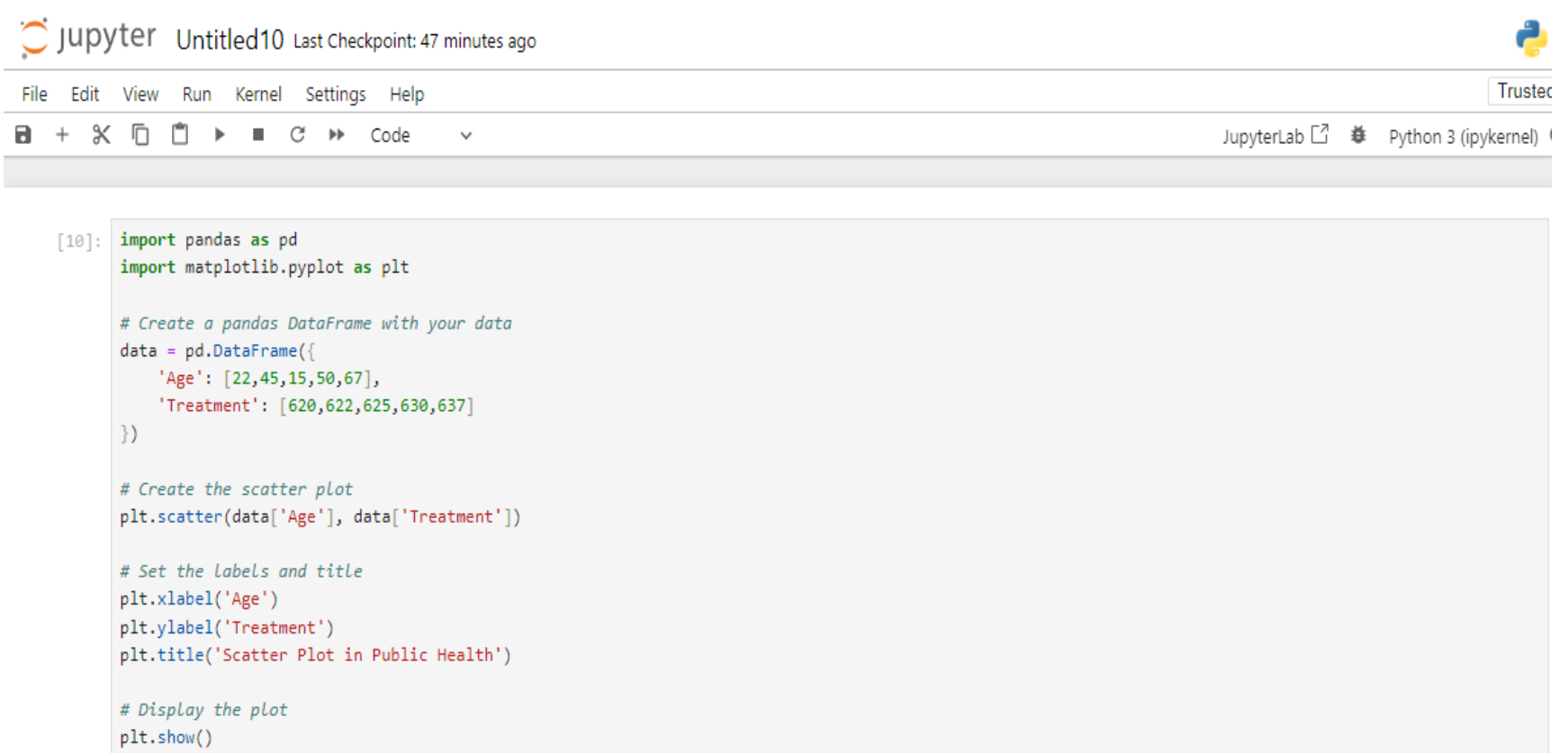
- Continuously gather and analyze customer feedback to make improvements to your products, services, and campaigns.

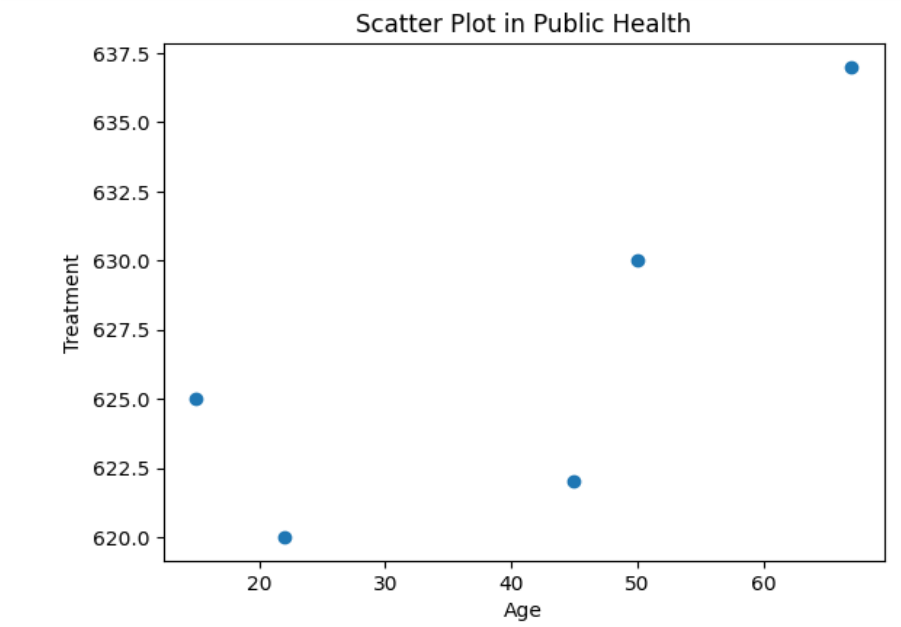
Competitor Analysis:

- Keep an eye on your competitors and analyze their marketing strategies. Identify gaps and opportunities that you can leverage.

Measurement and Analytics:

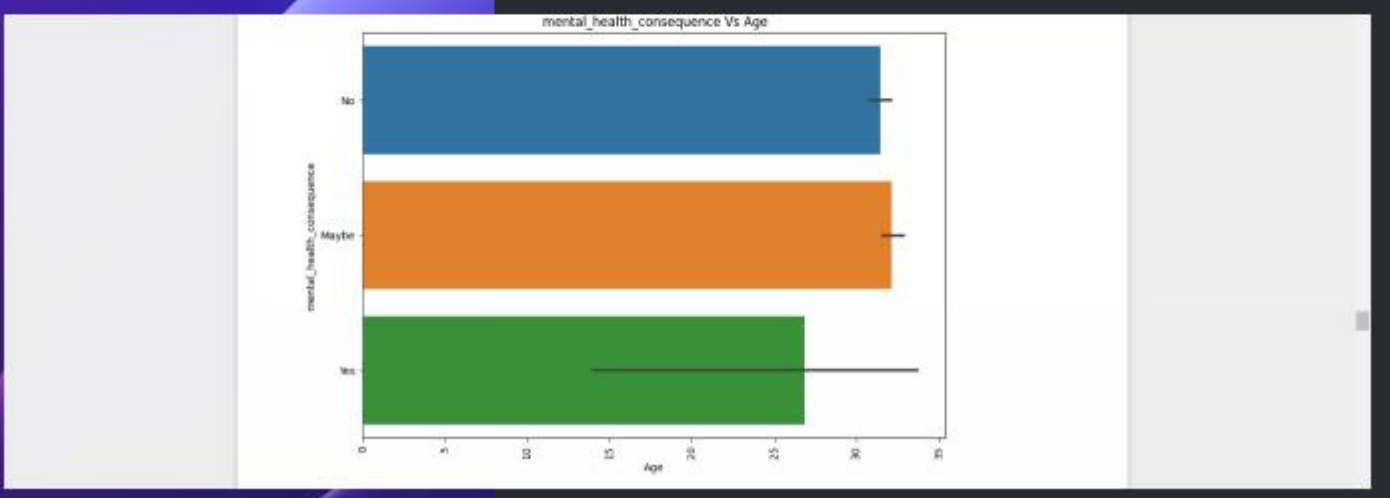
- Establish clear key performance indicators (KPIs) for your campaigns and regularly measure and analyze the results. Use theseinsights to refine future strategies resonate with your customers.





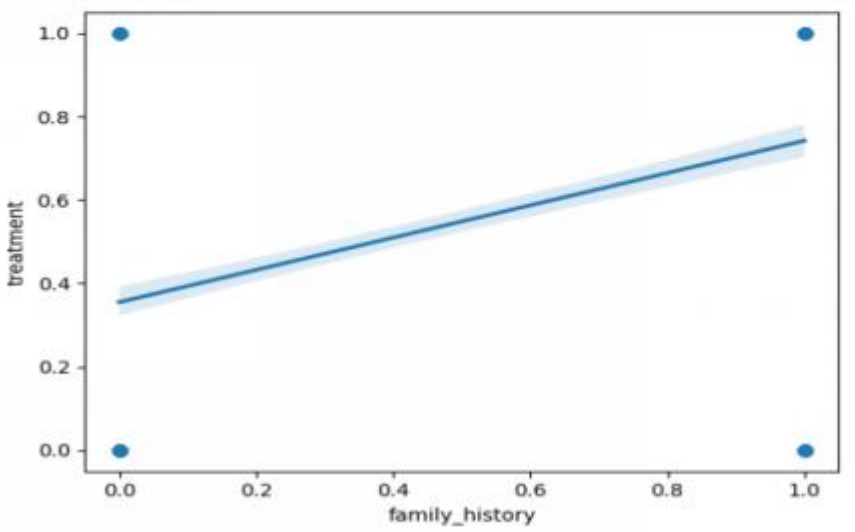
Mental health consequences Vs

```
[6]: plt.figure(figsize = (10,7))
     sns.barplot(data = df,x='Age',y='mental_health_consequence')
     plt.xticks(rotation=90)
     plt.title(" mental_health_consequence Vs Age")
     plt.show()
```



Treatment with family history

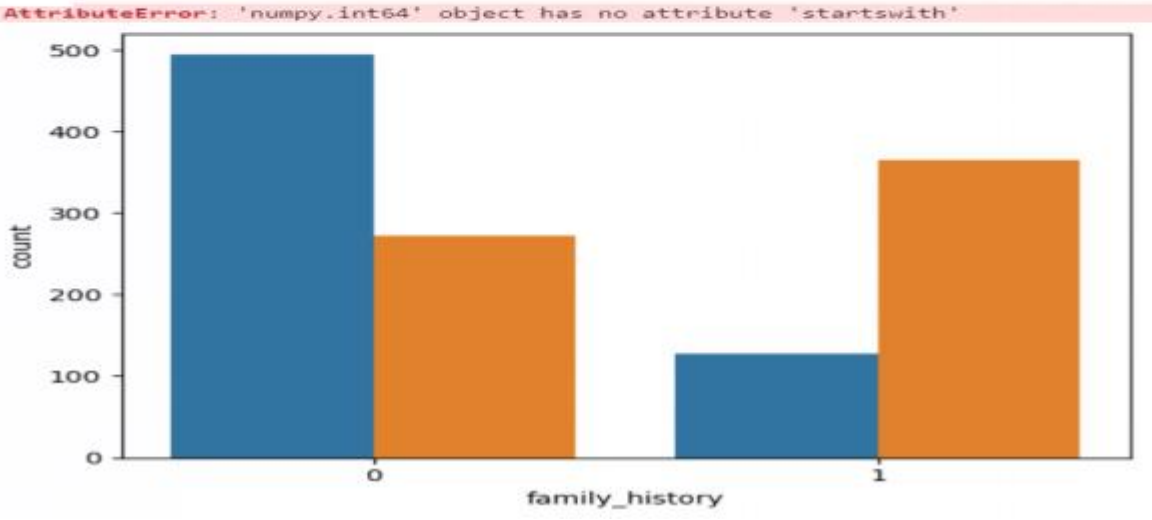
```
[21]: df["family_history"] = df["family_history"].replace({"Yes": 1, "No": 0})
      df["treatment"] = df["treatment"].replace({"Yes": 1, "No": 0})
      var = sns.regplot(x=df["family_history"].astype(float), y=df["treatment"].astype(float), data=df)
```



How is the treatment By

```
[22]: sns.countplot(x="family_history", hue="treatment", data=df)
      plt.title("Treatment by Family History")
      plt.xlabel("Family History")
      plt.ylabel("Count")
      plt.show()
```

```
AttributeError: 'numpy.int64' object has no attribute 'startswith'
```



Linear Regression Algorithm

```
[8]: from sklearn.linear_model import LinearRegression
     from sklearn.model_selection import train_test_split,cross_val_score
     from sklearn.metrics import mean_squared_error,r2_score
     from sklearn.preprocessing import OneHotEncoder
     from sklearn.preprocessing import StandardScaler
     from sklearn.linear_model import LogisticRegression
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
     from sklearn.metrics import confusion_matrix

[9]: # create OneHotEncoder object
     encoder = OneHotEncoder(handle_unknown='ignore')

     # transform categorical variable 'Country' into numerical
     X = encoder.fit_transform(df[['Country']])

     # assign target variable as numerical variable
     y = df['Age'].values

[10]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)

[11]: model = LinearRegression()
      model.fit(X_train, y_train)

[11]: LinearRegression
      LinearRegression()
```

```
y_pred = model.predict(X_test)

print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))

Training Accuracy : [illegible]
Testing Accuracy : -[illegible]

lin_reg = LinearRegression().fit(X_train,y_train)

lin_reg.coef_[0]

-3.434825[illegible]

lin_reg.intercept_

31.3015[illegible]

print("{0}+{1}*Happiness Level".format(lin_reg.intercept_,lin_reg.coef_[0]))

31.3015[illegible]+-3.434825[illegible]*Happiness Level

y_pred = lin_reg.predict(X_test)

y_pred[0:10]

array([27.8667231 , 31.8891976 , 32.9663[illegible], 16.7481593 , 27.8667231 ,
       28.8127[illegible], 32.9663[illegible], 32.9663[illegible], 32.9663[illegible], 32.9663[illegible]])

np.sqrt(mean_squared_error(y_test,y_pred))

18.2981[illegible]
```

Logistic regression ,K Neighbour classifier, K-nearest Neighbour and Desision tree classifier

```
y_pred = lr_model.predict(X_test)
print("Logistic Regression:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred, average='macro'))
print("Recall:", recall_score(y_test, y_pred, average='macro'))
print("F1 Score:", f1_score(y_test, y_pred, average='macro'))

Logistic Regression:
Accuracy: 0.0582010582010582
Precision: 0.00727127[illegible]
Recall: 0.0300340909090909[illegible]
F1 Score: 0.0100577943753213[illegible]
C:\Users\admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metric
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_divisio
  _warn_prf(average, modifier, msg_start, len(result))

knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
knn= KNeighborsClassifier().fit(X_train,y_train)

y_pred = knn.predict(X_test)
print("K-Nearest Neighbors:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred, average='macro'))
print("Recall:", recall_score(y_test, y_pred, average='macro'))
print("F1 Score:", f1_score(y_test, y_pred, average='macro'))

K-Nearest Neighbors:
Accuracy: 0.06613756613756613
Precision: 0.00605221[illegible]
Recall: 0.0404040404040404[illegible]
F1 Score: 0.01038349[illegible]
C:\Users\admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metric
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_divisio
  _warn_prf(average, modifier, msg_start, len(result))

dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)

DecisionTreeClassifier
DecisionTreeClassifier()
```

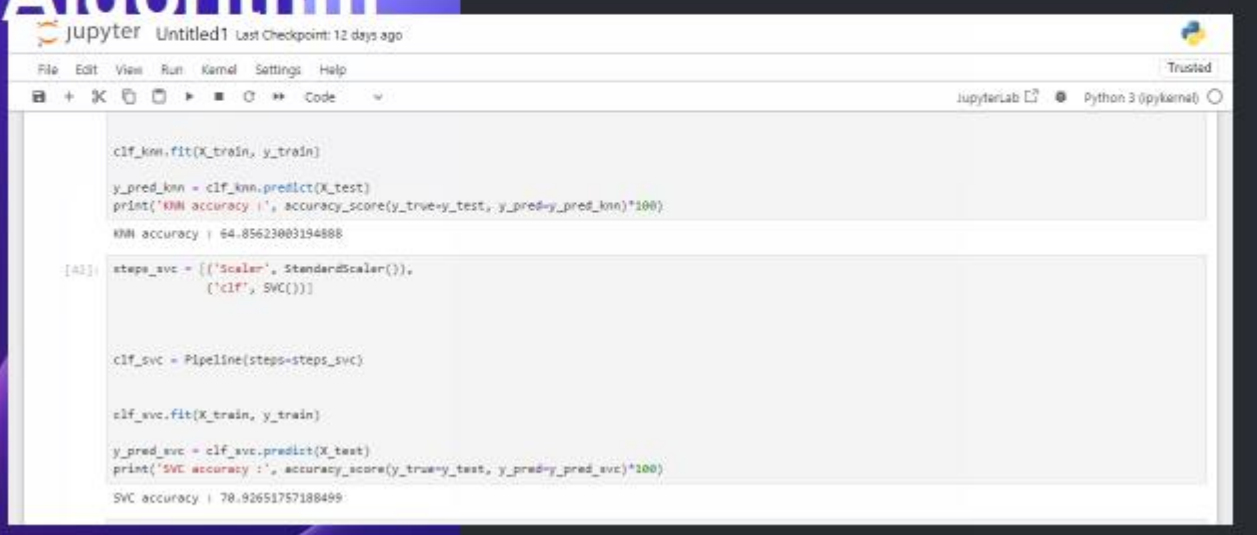
Decision Tree Classifier and Random Forest Classifier

localhost:8888/notebooks/Untitled1.ipynb

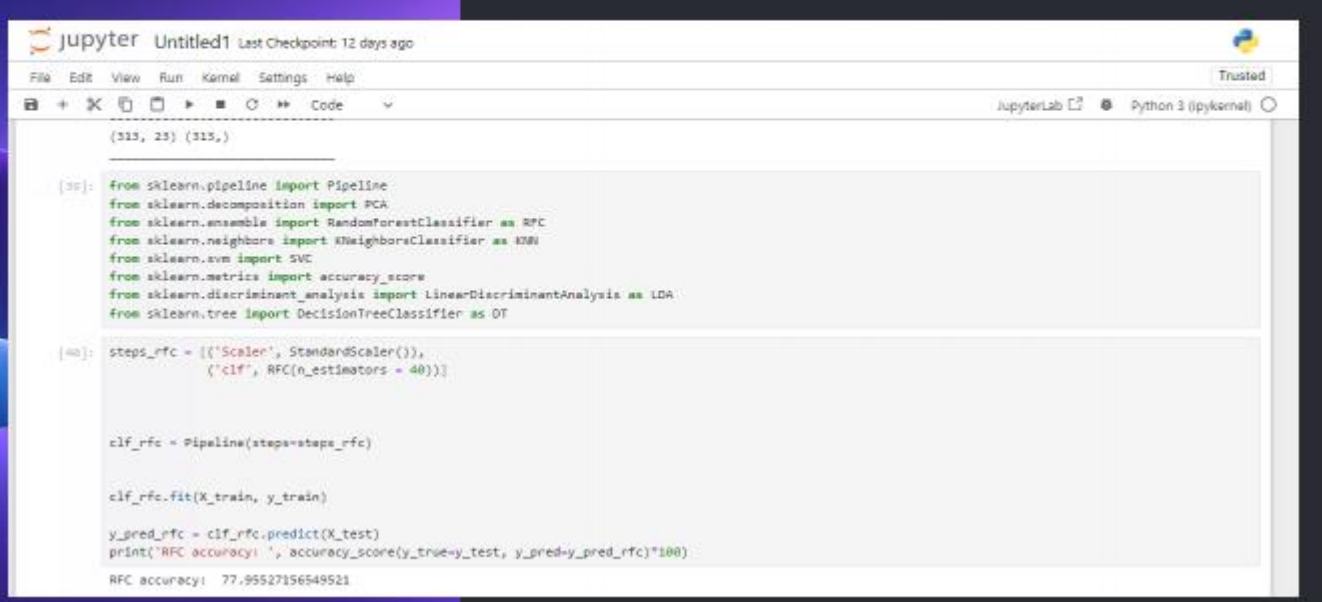
```
DecisionTreeClassifier()

[31]: y_pred = dt.predict(X_test)
      print("Decision Tree:")
      print("Accuracy:", accuracy_score(y_test, y_pred))
      print("Precision:", precision_score(y_test, y_pred, average='macro'))
      print("Recall:", recall_score(y_test, y_pred, average='macro'))
      print("F1 Score:", f1_score(y_test, y_pred, average='macro'))

Decision Tree:
Accuracy: 0.05555555555555555
Precision: 0.031256589750237065
Recall: 0.028838012477718 36
F1 Score: 0.011735262527079731
C:\Users\admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision is
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

[32]: # RandomForestClassifier
      rf = RandomForestClassifier()
      rf.fit(X_train, y_train)

[32]: RandomForestClassifier
      RandomForestClassifier()

[33]: y_pred = rf.predict(X_test)
      print("Random Forest:")
      print("Accuracy:", accuracy_score(y_test, y_pred))
      print("Precision:", precision_score(y_test, y_pred, average='macro'))
      print("Recall:", recall_score(y_test, y_pred, average='macro'))
      print("F1 Score:", f1_score(y_test, y_pred, average='macro'))

Random Forest:
Accuracy: 0.05555555555555555
Precision: 0.03267502190061717
Recall: 0.028888381229881602
F1 Score: 0.011529884903889 21
```

Test and Train Model Algorithm

Support Vector Classifier Algorithm



Random Forest Classifier



Training Accuracy and Testing

```
[23]: scaler = StandardScaler()
      df["age_scaled"] = scaler.fit_transform(df[["Age"]])

[24]: df.drop("Age", axis=1, inplace=True)

[25]: lr_model = LogisticRegression()
      lr_model.fit(X_train, y_train)

[25]: - LogisticRegression
      LogisticRegression()

[26]: y_pred = lr_model.predict(X_test)
      print("Training Accuracy :", lr_model.score(X_train, y_train))
      print("Testing Accuracy :",lr_model.score(X_test, y_test))
      lr_reg =  LogisticRegression().fit(X_train,y_train)
      lr_reg.coef_[0]
      lr_reg.intercept_
      print("{0}+{1}*Happiness Level".format(lr_reg.intercept_,lr_reg.coef_[0] ))
      df["family_history"] = df["family_history"].replace({"Yes": 1, "No": 0})
      df["treatment"] = df["treatment"].replace({"Yes": 1, "No": 0})
      sns = sns.regplot(x=df["family_history"].astype(float), y=df["treatment"].astype(float),logistic=True, ci=None, data=df)
      #  # plot results
      sns.set(style="whitegrid")
      plt.title("Treatment by family History")
      plt.xlabel("Family History")
      plt.ylabel("Treatment")
      plt.show()

Training Accuracy : 0.8225879682179419
Testing Accuracy : 0.8582810562010582
[-1.8645815  -2.17268627 -2.17268627 -2.17268627 -0.3725089   0.14533899
 -0.50912899  0.47630814  1.03927915  1.66130954  1.43032312  1.1998357
  2.41381959  2.38794448  2.19120785  2.17440162  1.91497125  2.28666527
  1.73819573  2.04846439  1.97777644  1.33533162  1.22678314  1.49888748
  1.35769631  1.14095555  0.7403819   0.21712347  0.43428462  0.81954331
 -0.4428397   0.05543766 -0.64573418 -1.61754097 -1.12139968 -0.83149755
 -0.67837561 -0.84921969 -2.17268627 -1.38662782 -1.0128323  -1.12139968
 -2.17268627 -2.17268627 -2.17268627 -1.65567533 -2.17268627 -2.17268627]+[-0.01614695 -0.00230238  0.         -0.00680279  0.         -0.00368135
 -0.0031677  -0.05041312 -0.00121391 -0.00215601 -0.00118812 -0.00233959
 -0.00114187 -0.00110929 -0.00232    -0.00586396  0.         -0.03247706
 -0.00232077 -0.00111886 -0.00001851 -0.02285126 -0.0021486  -0.00334509
 -0.00126133 -0.00111248 -0.00134692  0.         -0.02057366 -0.00733055
  0.         -0.00114187 -0.0011284  -0.00428735 -0.00220418 -0.00111248
 -0.00215601 -0.00120638 -0.00122856 -0.00483147  0.         -0.00547092
 -0.00593914 -0.00111391  0.71757744 -0.46891685 -0.00110655  0.        ]*Happiness Level
```

K Nearest Neighbour Algorithm

Decision Tree Algorithm

Conclusion

Hence public health awareness plays a major role in every peoples life to lead a disease free healthier life ,through analysis of our data we get to know how efficiently and effectively these campaigns have worked to promote health awareness among public.Through the analysis we obtain the performance and results through visuals with a better readability.