**PHASE : 4**

WATER QUALITY ANALYSIS

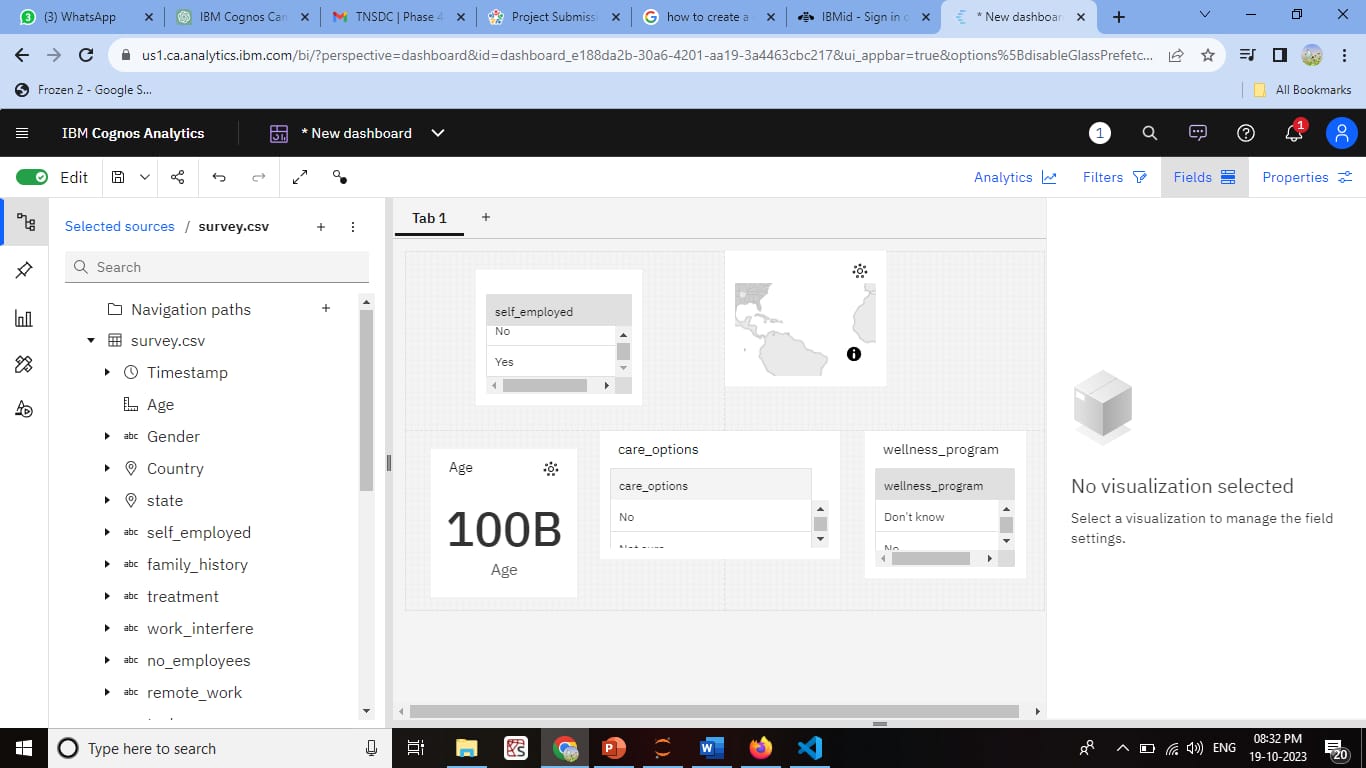
**1. \*\*Data Preparation\*\*:**

* Download the dataset from Kaggle: https://www.kaggle.com/datasets/adityakadiwal/water-potability.
* Import necessary libraries such as pandas, numpy, matplotlib, seaborn, and scikit-learn.
* Load the dataset into a pandas DataFrame.

**IBM Cognos Dashboard:**

Create a new dashboard in IBM Cognos that displays key insights.

For example:



**Integration with Python:**

For more complex analysis, you can integrate Python into IBM Cognos using the Python integration feature. Here's a simplified example of how to do this:

**Exploring the data**

* We have now dealt with the missing values in the data.
* Next, I want to **explore** the data.
* Does age makes one more likely to suffer a stroke? What about gender? Or BMI?
* These are all questions that can be explored and answered with some data visulization.
* First, let's look at the numeric/continuous variable distribution

**program**

variables = [variable for variable **in** df.columns if variable **not** **in** ['id','stroke']]

conts = ['age','avg\_glucose\_level','bmi']

fig = plt.figure(figsize=(12, 12), dpi=150, facecolor='#fafafa')

gs = fig.add\_gridspec(4, 3)

gs.update(wspace=0.1, hspace=0.4)

background\_color = "#fafafa"

plot = 0

for row **in** range(0, 1):

for col **in** range(0, 3):

locals()["ax"+str(plot)] = fig.add\_subplot(gs[row, col])

locals()["ax"+str(plot)].set\_facecolor(background\_color)

locals()["ax"+str(plot)].tick\_params(axis='y', left=False)

locals()["ax"+str(plot)].get\_yaxis().set\_visible(False)

for s **in** ["top","right","left"]:

locals()["ax"+str(plot)].spines[s].set\_visible(False)

plot += 1

plot = 0

for variable **in** conts:

sns.kdeplot(df[variable] ,ax=locals()["ax"+str(plot)], color='#0f4c81', shade=True, linewidth=1.5, ec='black',alpha=0.9, zorder=3, legend=False)

locals()["ax"+str(plot)].grid(which='major', axis='x', zorder=0, color='gray', linestyle=':', dashes=(1,5))

*#locals()["ax"+str(plot)].set\_xlabel(variable) removed this for aesthetics*

plot += 1

ax0.set\_xlabel('Age')

ax1.set\_xlabel('Avg. Glucose Levels')

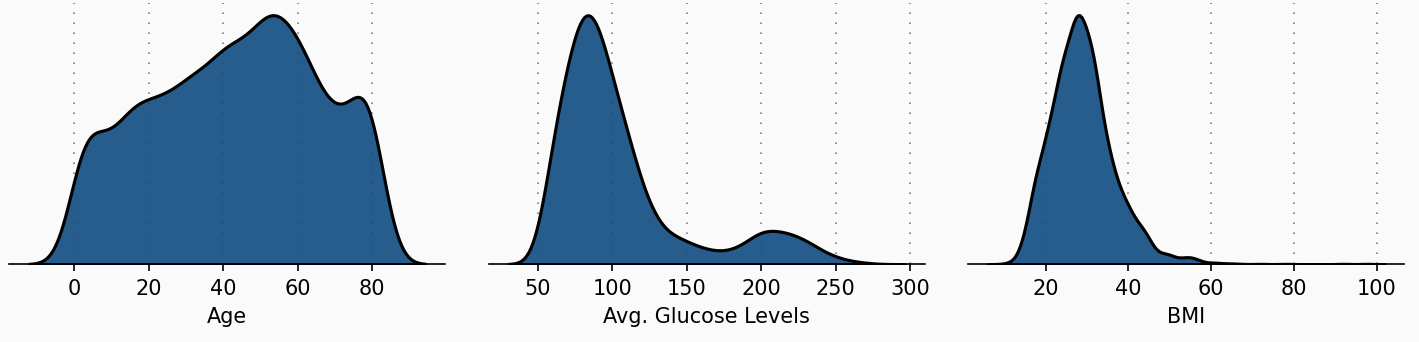
ax2.set\_xlabel('BMI')

ax0.text(-20, 0.022, 'Numeric Variable Distribution', fontsize=20, fontweight='bold', fontfamily='serif')

ax0.text(-20, 0.02, 'We see a positive skew in BMI and Glucose Level', fontsize=13, fontweight='light', fontfamily='serif')

plt.show()

**output**

****

**program**

str\_only = df[df['stroke'] == 1]

no\_str\_only = df[df['stroke'] == 0

fig = plt.figure(figsize=(10,16),dpi=150,facecolor=background\_color)

gs = fig.add\_gridspec(4, 2)

gs.update(wspace=0.5, hspace=0.2)

ax0 = fig.add\_subplot(gs[0, 0:2])

ax1 = fig.add\_subplot(gs[1, 0:2])

ax0.set\_facecolor(background\_color)

ax1.set\_facecolor(background\_color)

sns.regplot(no\_str\_only['age'],y=no\_str\_only['avg\_glucose\_level'],

color='lightgray',

logx=True,

ax=ax0)

sns.regplot(str\_only['age'],y=str\_only['avg\_glucose\_level'],

color='#0f4c81',

logx=True,scatter\_kws={'edgecolors':['black'],

'linewidth': 1},

ax=ax0)

ax0.set(ylim=(0, None))

ax0.set\_xlabel(" ",fontsize=12,fontfamily='serif')

ax0.set\_ylabel("Avg. Glucose Level",fontsize=10,fontfamily='serif',loc='bottom')

ax0.tick\_params(axis='x', bottom=False)

ax0.get\_xaxis().set\_visible(False)

for s **in** ['top','left','bottom']:

ax0.spines[s].set\_visible(False)

sns.regplot(no\_str\_only['age'],y=no\_str\_only['bmi'],

color='lightgray',

logx=True,

ax=ax1)

sns.regplot(str\_only['age'],y=str\_only['bmi'],

color='#0f4c81', scatter\_kws={'edgecolors':['black'],

'linewidth': 1},

logx=True,

ax=ax1)

ax1.set\_xlabel("Age",fontsize=10,fontfamily='serif',loc='left')

ax1.set\_ylabel("BMI",fontsize=10,fontfamily='serif',loc='bottom')

for s **in** ['top','left','right']:

ax0.spines[s].set\_visible(False)

ax1.spines[s].set\_visible(False)

ax0.text(-5,350,'Strokes by Age, Glucose Level, and BMI',fontsize=18,fontfamily='serif',fontweight='bold')

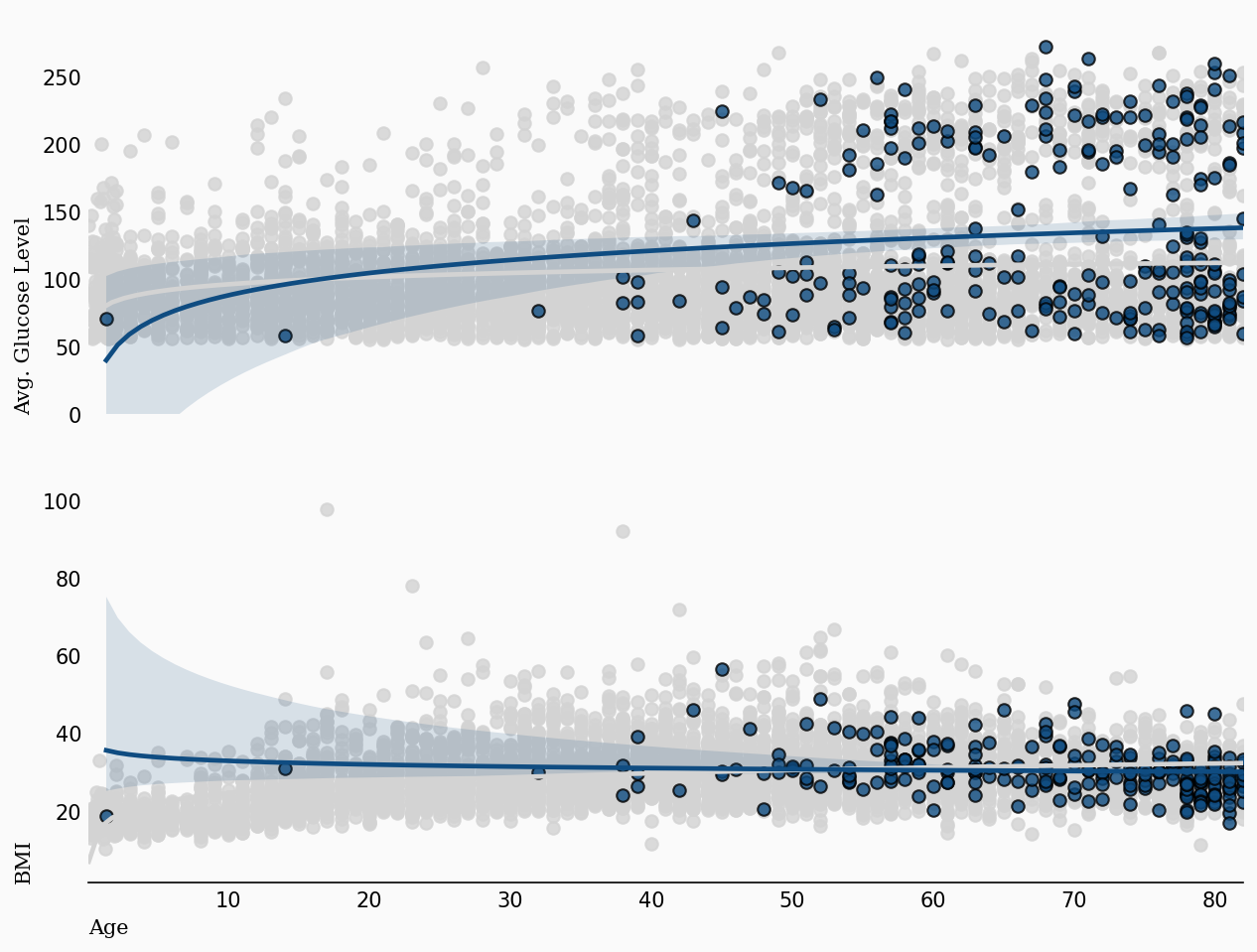
ax0.text(-5,320,'Age appears to be a very important factor',fontsize=14,fontfamily='serif')

ax0.tick\_params(axis=u'both', which=u'both',length=0)

ax1.tick\_params(axis=u'both', which=u'both',length=0)

plt.show()

**Output**



**conclusion**

* This is a general guide to building a predictive model for water potability using the provided dataset. You can explore other machine learning algorithms, feature engineering, and hyperparameter tuning to improve the model's performance further.