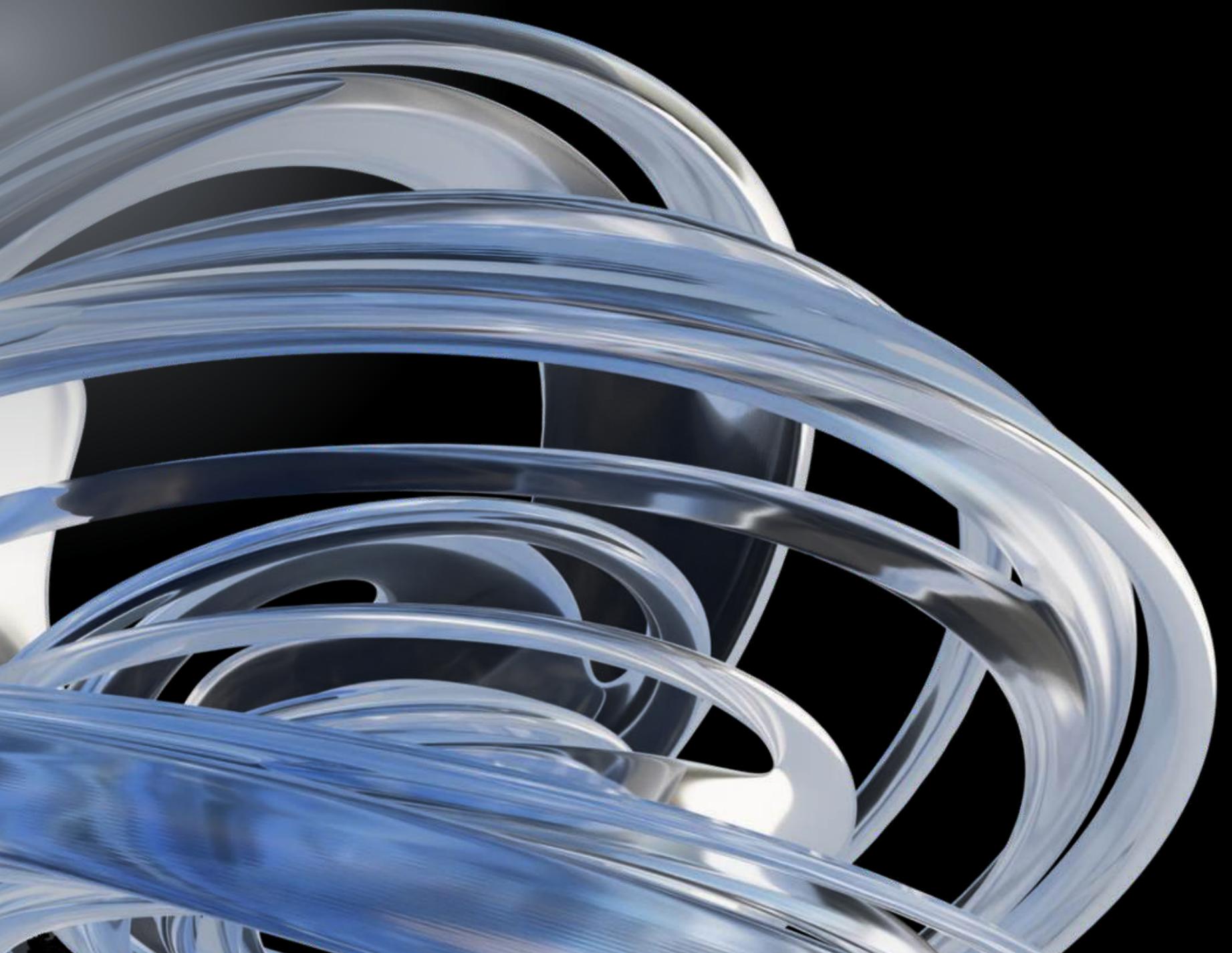


# Diabetes Predictor Modeling

**PRESENTED BY:**

Thi Nguyet Anh Che, Quan Pham, Kaveh Jalilian, Yi-Fang Chung



# Agenda

Project Overview

Data Wrangling and Cleaning

Exploratory Data Analysis (EDA)

Predictive Modeling

Model Evaluation & Results

Conclusion

## Data Source

The dataset [Diabetes 012 Health Indicators BRFSS 2015](#)  
contains healthcare statistics and lifestyle survey information  
about people in general along with their diagnosis of diabetes.

## Key Features

- Demographics variables (age, gender, race)
- Lifestyle factors (smoking, alcohol consumption, sleep patterns)
- Health indicators (BMI, general health rating, physical activity levels)
- Pre-existing conditions (high blood pressure, cholesterol levels)
- Diabetes diagnosis (target variable)

## **Business Question**

**How do lifestyle choices and physical health indicators correlate  
with diabetes risk?**

## **Objective**

**Identify key risk factors through data analysis  
Build a predictive model for early diabetes risk detection**

# Data Wrangling and Cleaning



# Data Structure Assessment

## Data Descriptions:

Column	Description
Diabetes_012	0 = no diabetes; 1 = prediabetes; 2 = diabetes
HighBP	0 = no high Blood Pressure; 1 = high Blood Pressure
HighChol	0 = no high Cholesterol; 1 = high Cholesterol
CholCheck	0 = no cholesterol check in 5 years; 1 = yes cholesterol check in 5 years
BMI	Body Mass Index
Smoker	Have you smoked at least 100 cigarettes in your entire life? 0 = no; 1 = yes
Stroke	(Ever told) you had a stroke? 0 = no; 1 = yes
HeartDiseaseorAttack	Coronary Heart Disease (CHD) or Myocardial Infarction (MI)? 0 = no; 1 = yes
PhysActivity	Physical activity in past 30 days - not including job? 0 = no; 1 = yes
Fruits	Consume Fruit 1 or more times per day? 0 = no; 1 = yes
Veggies	Consume Vegetables 1 or more times per day? 0 = no; 1 = yes
HvyAlcoholConsump	Heavy drinkers ( men more than 14 drinks per week or women more than 7 drinks per week)? 0 = no; 1 = yes
AnyHealthcare	Have any kind of health care coverage? 0 = no; 1 = yes
NoDocbcCost	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no; 1 = yes
GenHlth	Would you say that in general your health on scale 1-5 is: 1 = excellent, 2 = very good, 3 = good. 4 = fair, 5 = poor

MentHlth:	For how many days during the past 30 days was your mental health not good? scale 1-30 days
PhysHlth:	For how many days during the past 30 days was your physical health not good? scale 1-30 days
DiffWalk	Do you have serious difficulty walking or climbing stairs? 0 = no; 1 = yes
Sex	0 = female; 1 = male
Age	1 = 18-24 y/o; 2 = 25-29; 3 = 30-34; 4 = 35-39; 5 = 40-44; 6 = 45-49; 7 = 50-54; 8 = 55-59; 9 = 60-64; 10 = 65-69; 11 = 70-74; 12 = 75-79; 13 = 80 or older
Education	1 = Never attended school or only kindergarten; 2 = Grades 1-8; 3 = Grades 9-11; 4 = Grade 12 or GED (High school graduate); 5 = College 1-3 years (Some college or technical school); 6 = College 4 years or more
Income	1 = less than 10,000; 2 = less than 15,000; 3 = less than 20,000; 4 = less than 25,000; 5 = less than 35,000; 6 = less than 50,000; 7 = less than 75,000; 8 = 75,000 or more.

# Checking missing values, unique values, and duplicates.

```
[ ] df.isnull().sum()
[ ] duplicates = df[df.duplicated()]
print(f"Number of Duplicates: {len(duplicates)}")

Number of Duplicates: 23968

[ ] duplicates
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
1242	1	1	1	1	27.0	1	0
1563	0	0	0	1	21.0	1	0
2700	0	0	0	1	32.0	0	0
3160	0	0	0	1	21.0	0	0
3332	0	0	0	1	24.0	0	0
...	...	...	...	...	...	...	...
253492	1	1	1	1	33.0	0	0
253550	0	0	0	1	25.0	0	0
253563	0	0	1	1	24.0	1	0

```
[ ] df.isnull().sum()
[ ] df.duplicated()

# Print unique values
cols = df.columns
for col in cols:
    print(col)

# get a list of unique values
unique = df[col].unique()
print(unique, '\n=====')
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
Diabetes_binary	0						
HighBP	0						
HighChol	0						
CholCheck	0						
BMI	0						
Smoker	0						
Stroke	0						
HeartDiseaseorAttack	0						
PhysActivity	0						
Fruits	0						
Veggies	0						
HvyAlcoholConsump	0						
AnyHealthcare	0						
NoDocbcCost	0						
GenHlth							
MentHlth							
PhysHlth							
DiffWalk							
Sex							
Age							
Education							
Income							

```
# Print unique values
cols = df.columns
for col in cols:
    print(col)

# get a list of unique values
unique = df[col].unique()
print(unique, '\n=====')
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
Diabetes_binary	0						
HighBP	0						
HighChol	0						
CholCheck	0						
BMI	0						
Smoker	0						
Stroke	0						
HeartDiseaseorAttack	0						
PhysActivity	0						
Fruits	0						
Veggies	0						
HvyAlcoholConsump	0						
AnyHealthcare	0						
NoDocbcCost	0						
GenHlth	0						
MentHlth	0						
PhysHlth	0						
DiffWalk	0						
Sex	0						
Age	0						
Education	0						
Income	0						

```
# Print unique values
cols = df.columns
for col in cols:
    print(col)

# get a list of unique values
unique = df[col].unique()
print(unique, '\n=====')
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
Diabetes_binary	0						
HighBP	0						
HighChol	0						
CholCheck	0						
BMI	0						
Smoker	0						
Stroke	0						
HeartDiseaseorAttack	0						
PhysActivity	0						
Fruits	0						
Veggies	0						
HvyAlcoholConsump	0						
AnyHealthcare	0						
NoDocbcCost	0						
GenHlth	0						
MentHlth	0						
PhysHlth	0						
DiffWalk	0						
Sex	0						
Age	0						
Education	0						
Income	0						

```
# Print unique values
cols = df.columns
for col in cols:
    print(col)

# get a list of unique values
unique = df[col].unique()
print(unique, '\n=====')
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke
Diabetes_binary	0						
HighBP	0						
HighChol	0						
CholCheck	0						
BMI	0						
Smoker	0						
Stroke	0						
HeartDiseaseorAttack	0						
PhysActivity	0						
Fruits	0						
Veggies	0						
HvyAlcoholConsump	0						
AnyHealthcare	0						
NoDocbcCost	0						
GenHlth	0						
MentHlth	0						
PhysHlth	0						
DiffWalk	0						
Sex	0						
Age	0						
Education	0						
Income	0						

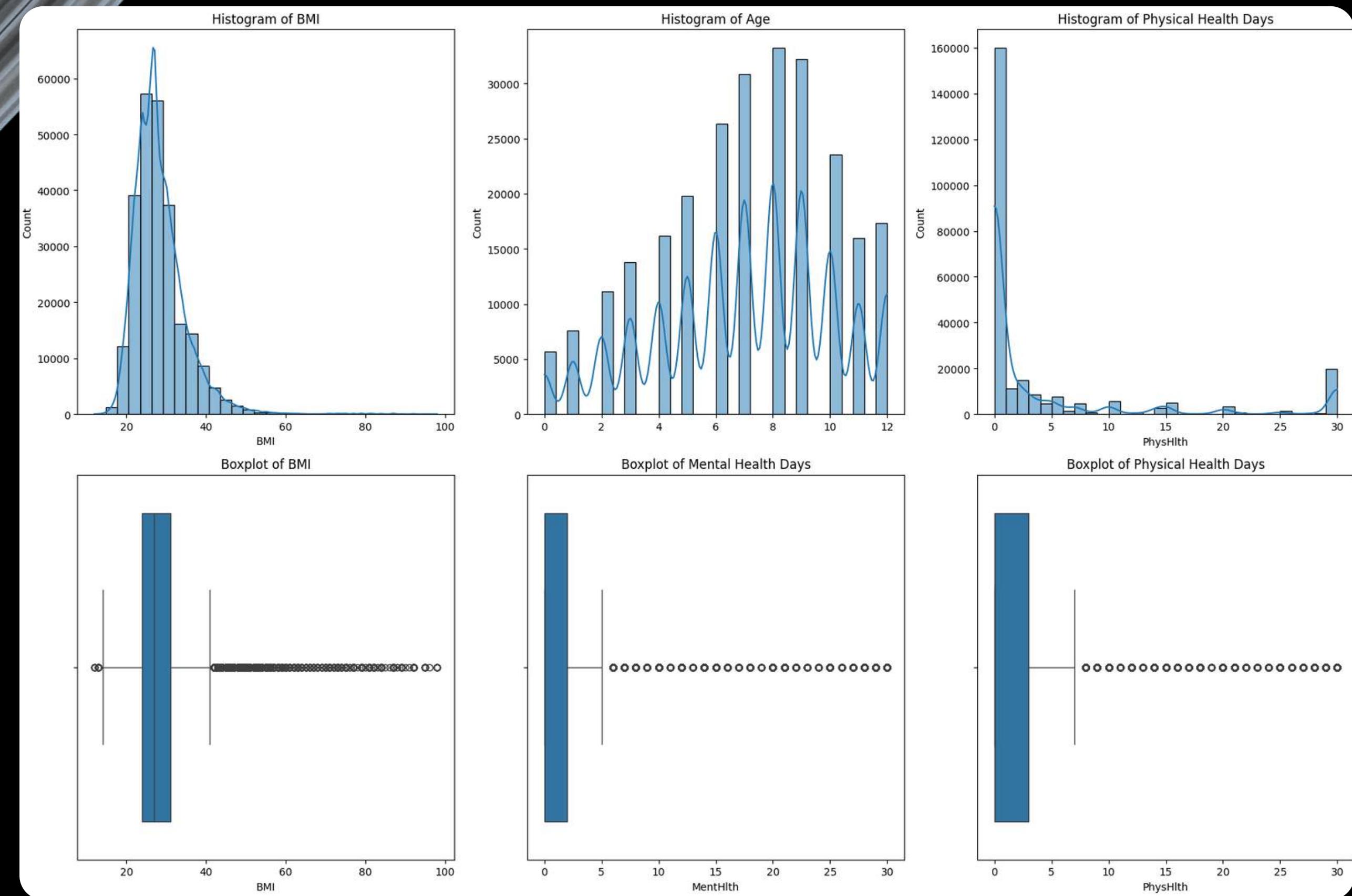


# **EXPLORATORY DATA ANALYSIS**

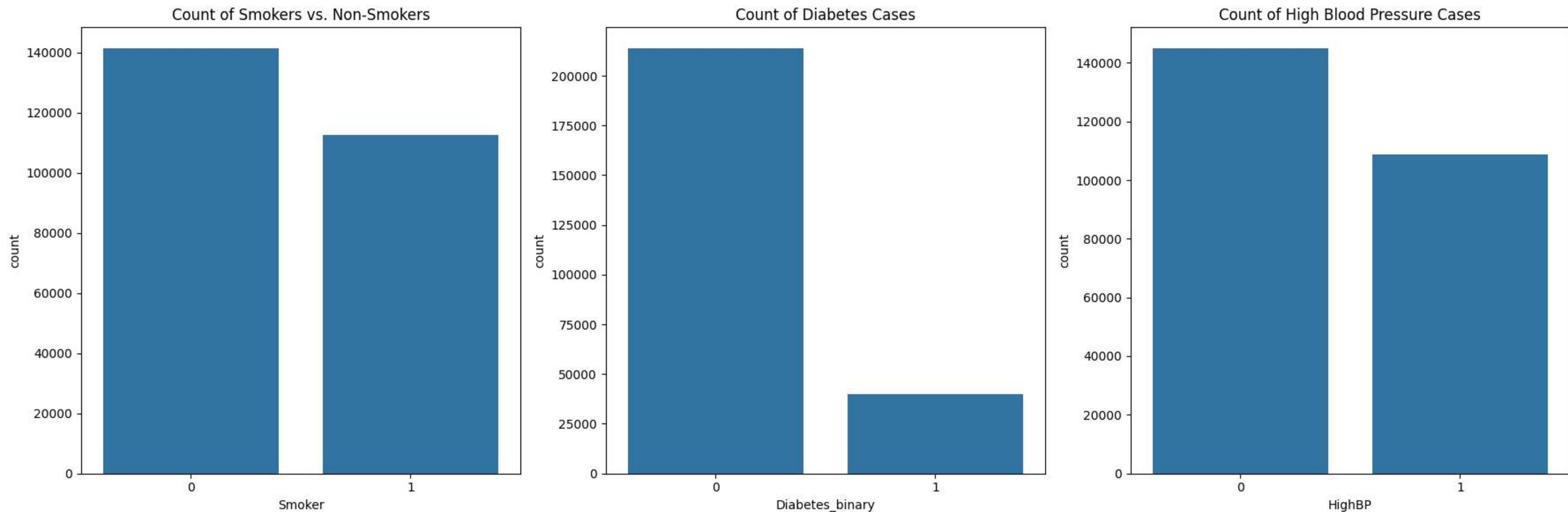
# Statistic Summary

	count	mean	std	min	25%	50%	75%	max	Skewness
<b>Diabetes_binary</b>	253680.0	0.157588	0.364355	0.0	0.0	0.0	0.0	1.0	1.879563
<b>HighBP</b>	253680.0	0.429001	0.494934	0.0	0.0	0.0	1.0	1.0	0.286904
<b>HighChol</b>	253680.0	0.424121	0.494210	0.0	0.0	0.0	1.0	1.0	0.307075
<b>CholCheck</b>	253680.0	0.962670	0.189571	0.0	1.0	1.0	1.0	1.0	-4.881271
<b>BMI</b>	253680.0	28.382364	6.608694	12.0	24.0	27.0	31.0	98.0	2.122004
<b>Smoker</b>	253680.0	0.443169	0.496761	0.0	0.0	0.0	1.0	1.0	0.228810
<b>Stroke</b>	253680.0	0.040571	0.197294	0.0	0.0	0.0	0.0	1.0	4.657340
<b>HeartDiseaseorAttack</b>	253680.0	0.094186	0.292087	0.0	0.0	0.0	0.0	1.0	2.778742
<b>PhysActivity</b>	253680.0	0.756544	0.429169	0.0	1.0	1.0	1.0	1.0	-1.195546
<b>Fruits</b>	253680.0	0.634256	0.481639	0.0	0.0	1.0	1.0	1.0	-0.557500
<b>Veggies</b>	253680.0	0.811420	0.391175	0.0	1.0	1.0	1.0	1.0	-1.592239
<b>HvyAlcoholConsump</b>	253680.0	0.056197	0.230302	0.0	0.0	0.0	0.0	1.0	3.854132
<b>AnyHealthcare</b>	253680.0	0.951053	0.215759	0.0	1.0	1.0	1.0	1.0	-4.181116
<b>NoDocbcCost</b>	253680.0	0.084177	0.277654	0.0	0.0	0.0	0.0	1.0	2.995290
<b>MentHlth</b>	253680.0	3.184772	7.412847	0.0	0.0	0.0	2.0	30.0	2.721148
<b>PhysHlth</b>	253680.0	4.242081	8.717951	0.0	0.0	0.0	3.0	30.0	2.207395

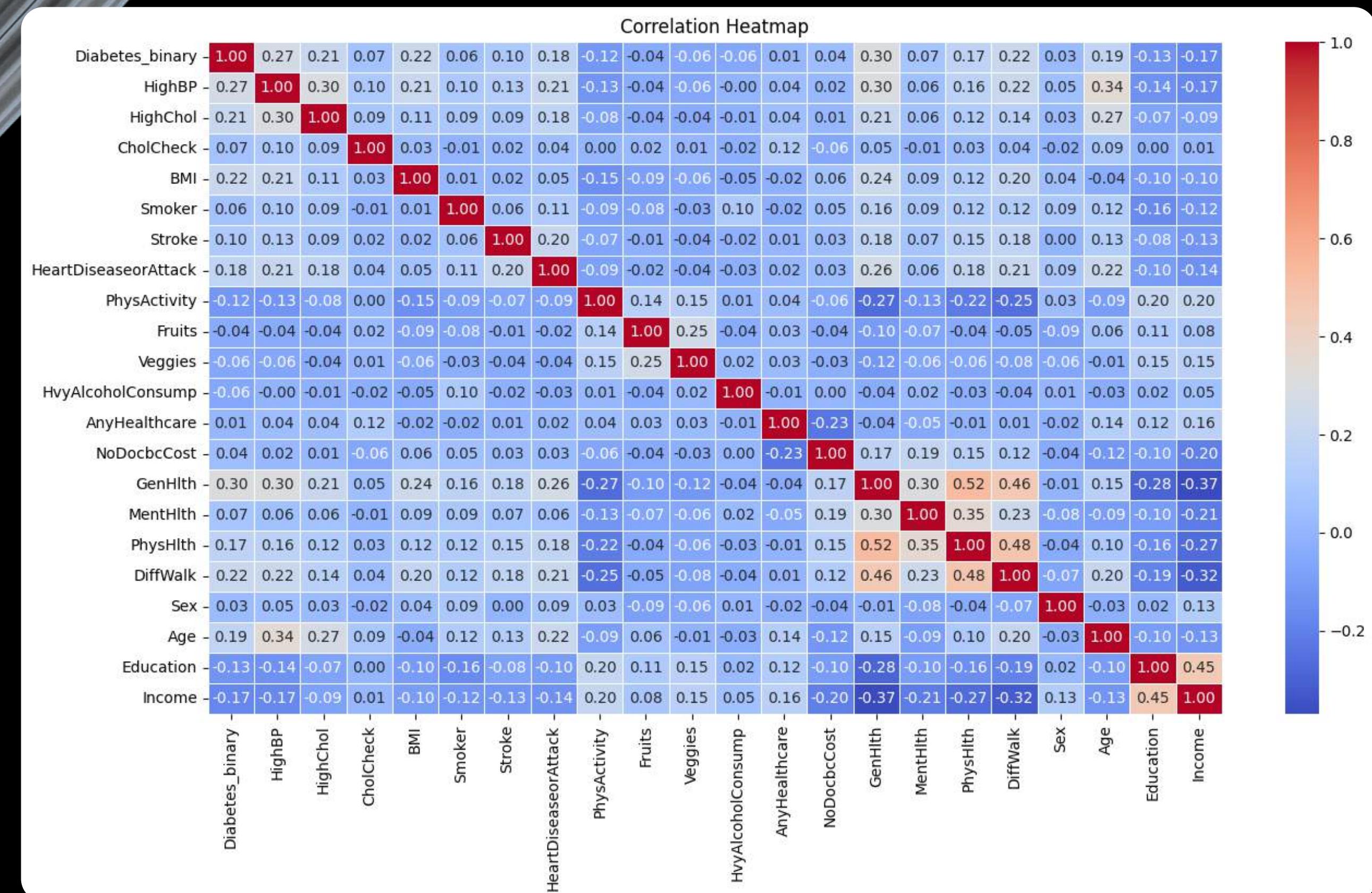
# Distribution Plots and Box Plots



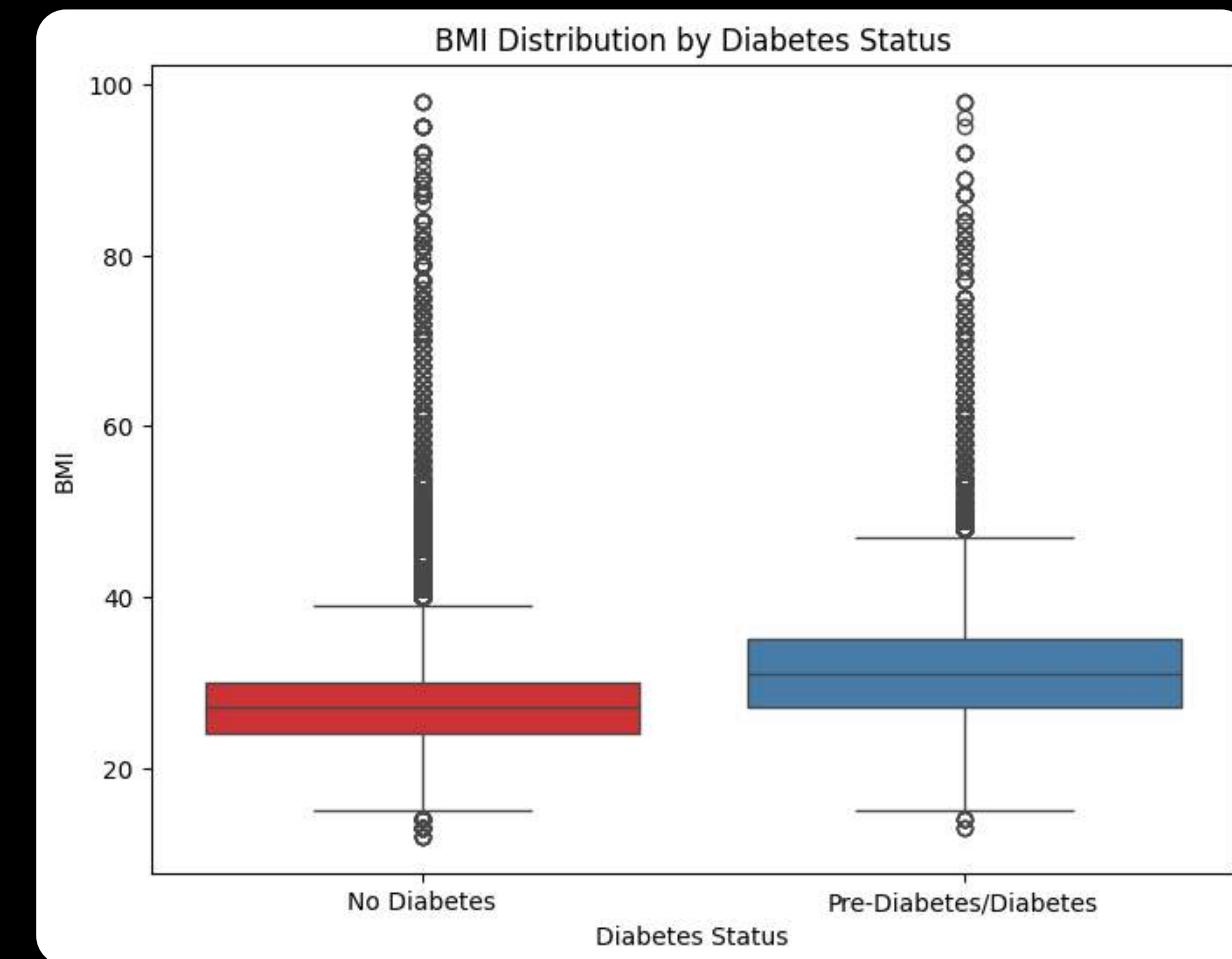
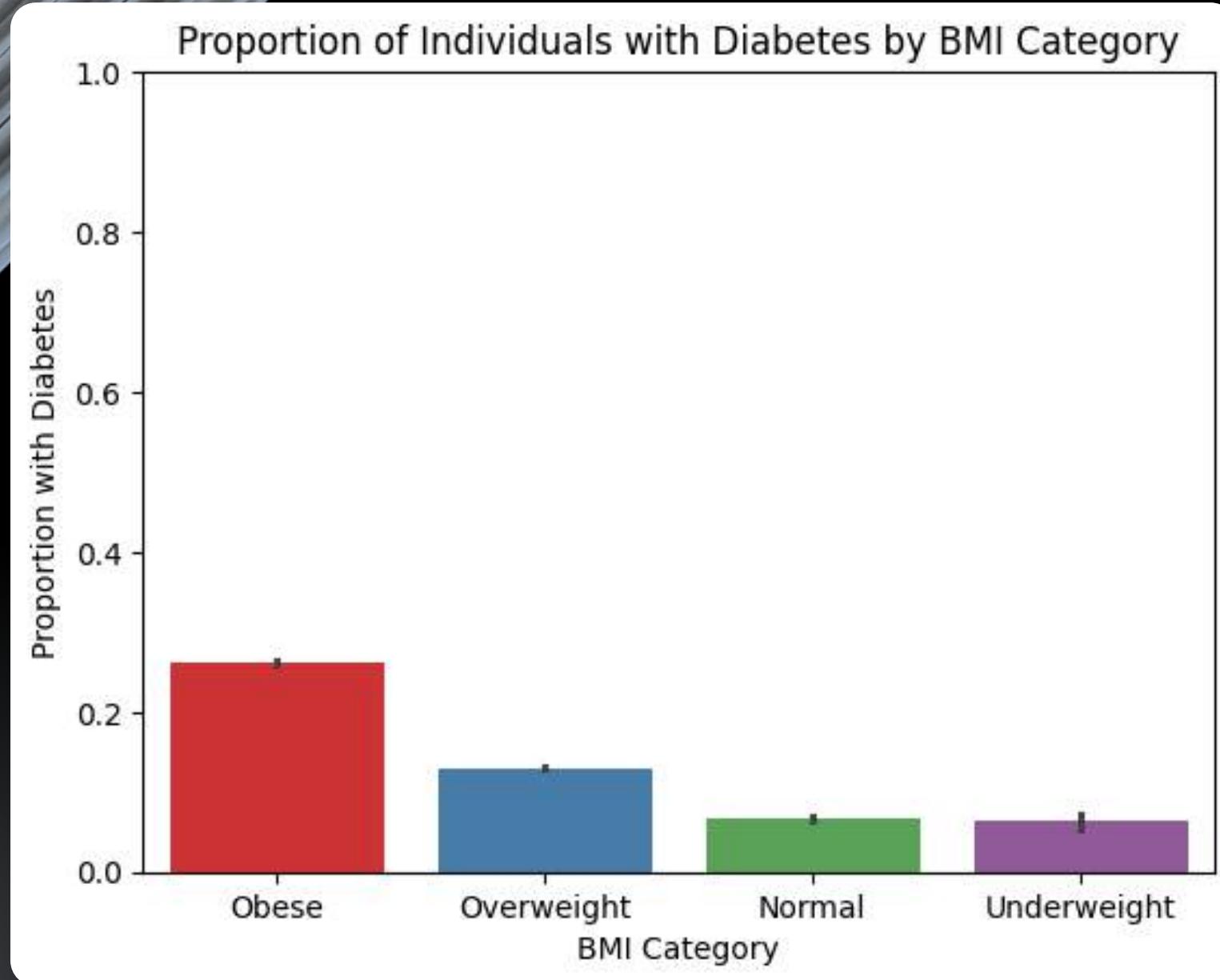
## Count Plots



# Correlation Heatmap

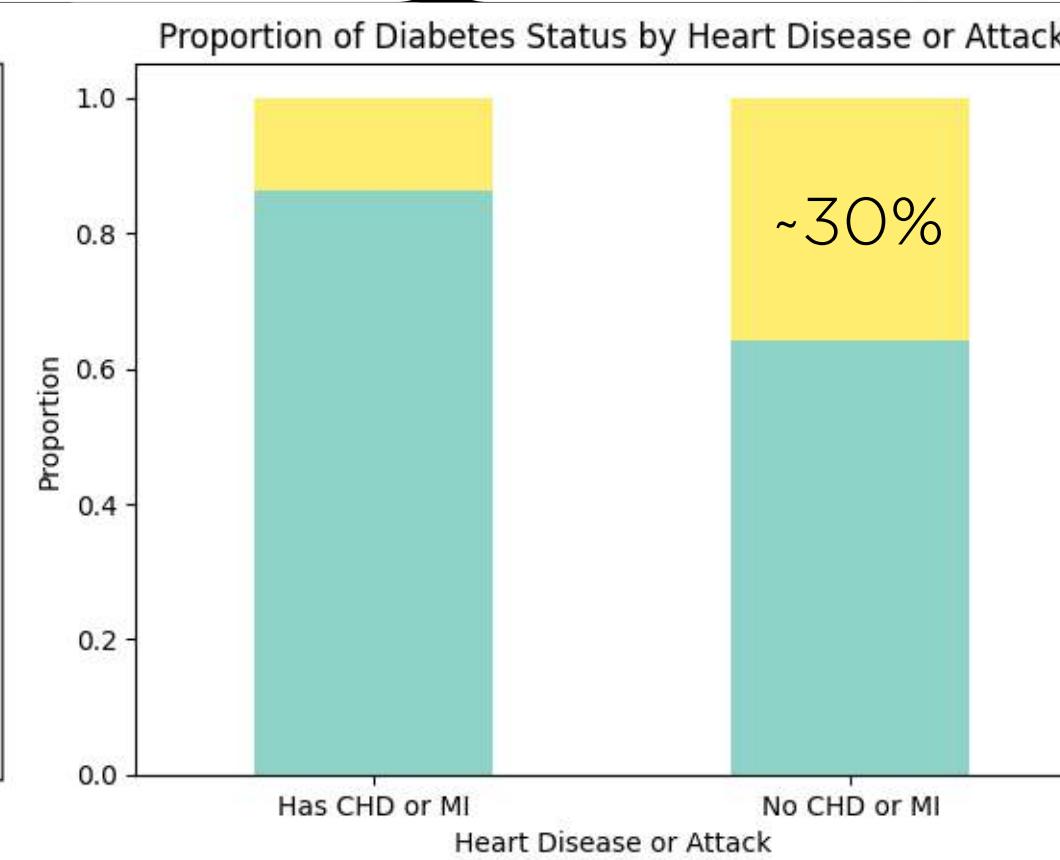
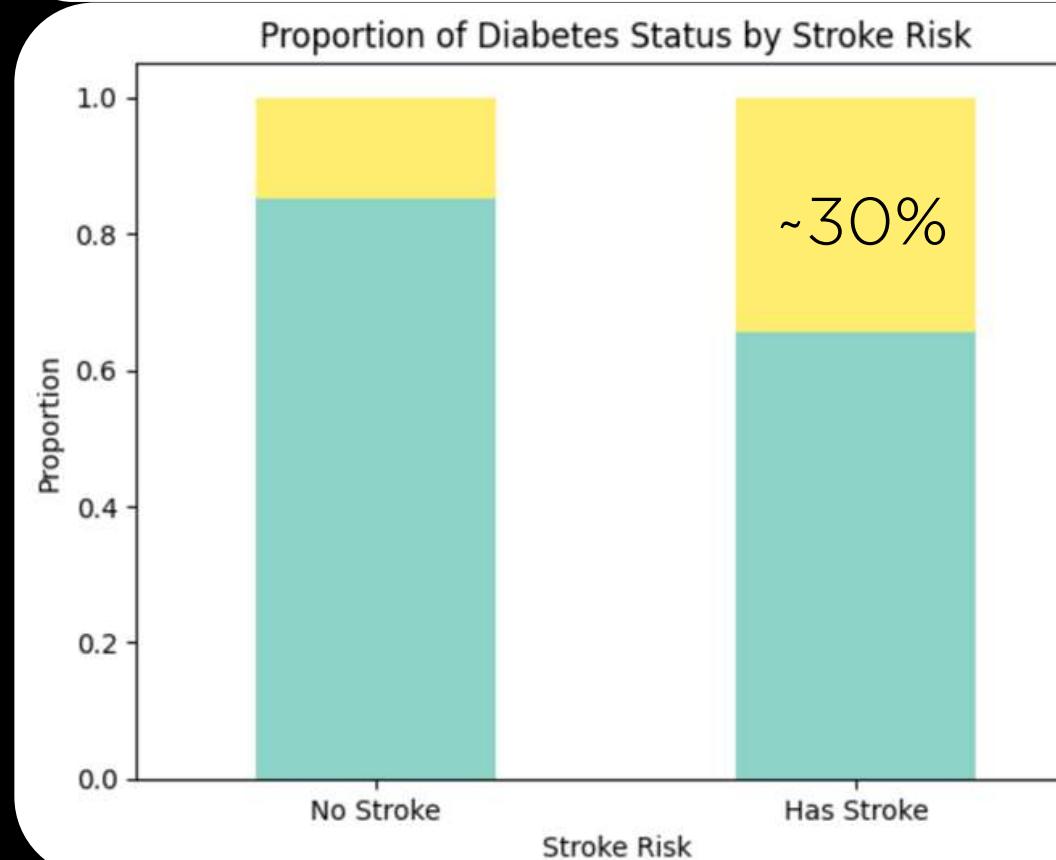
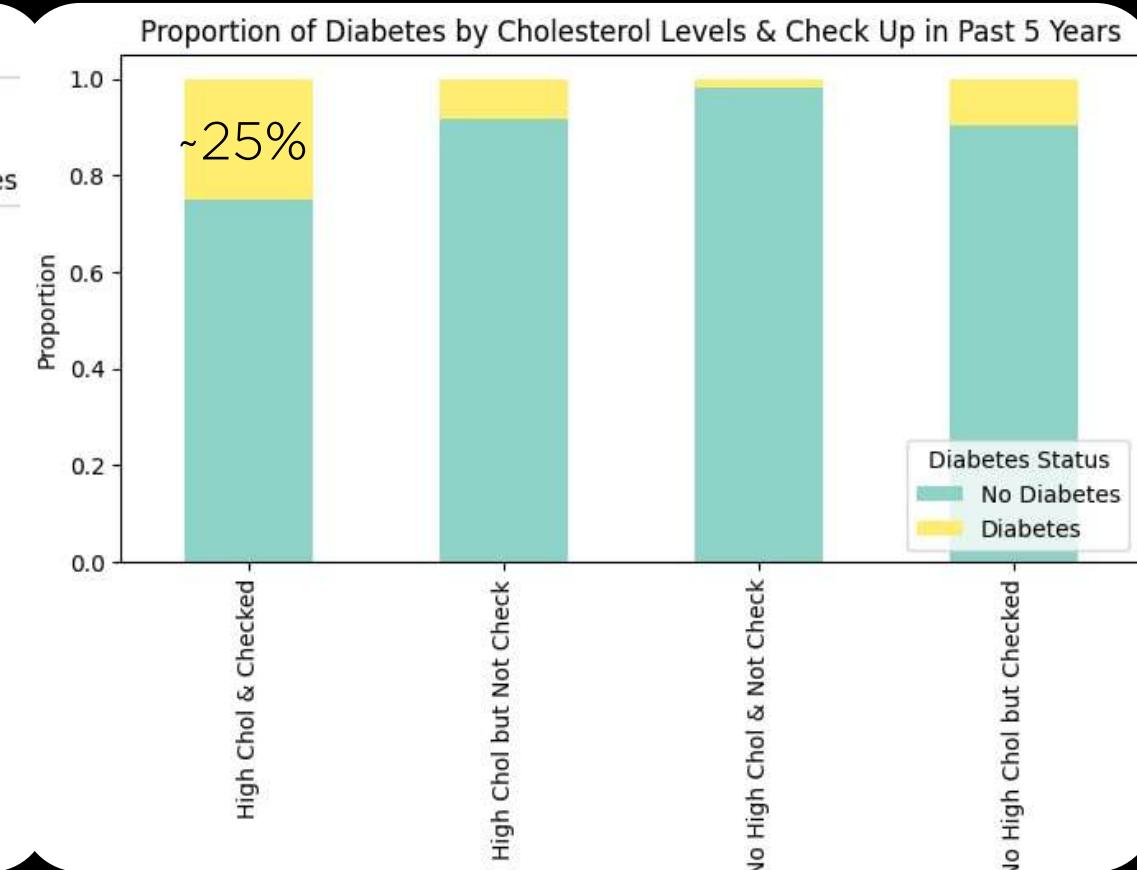
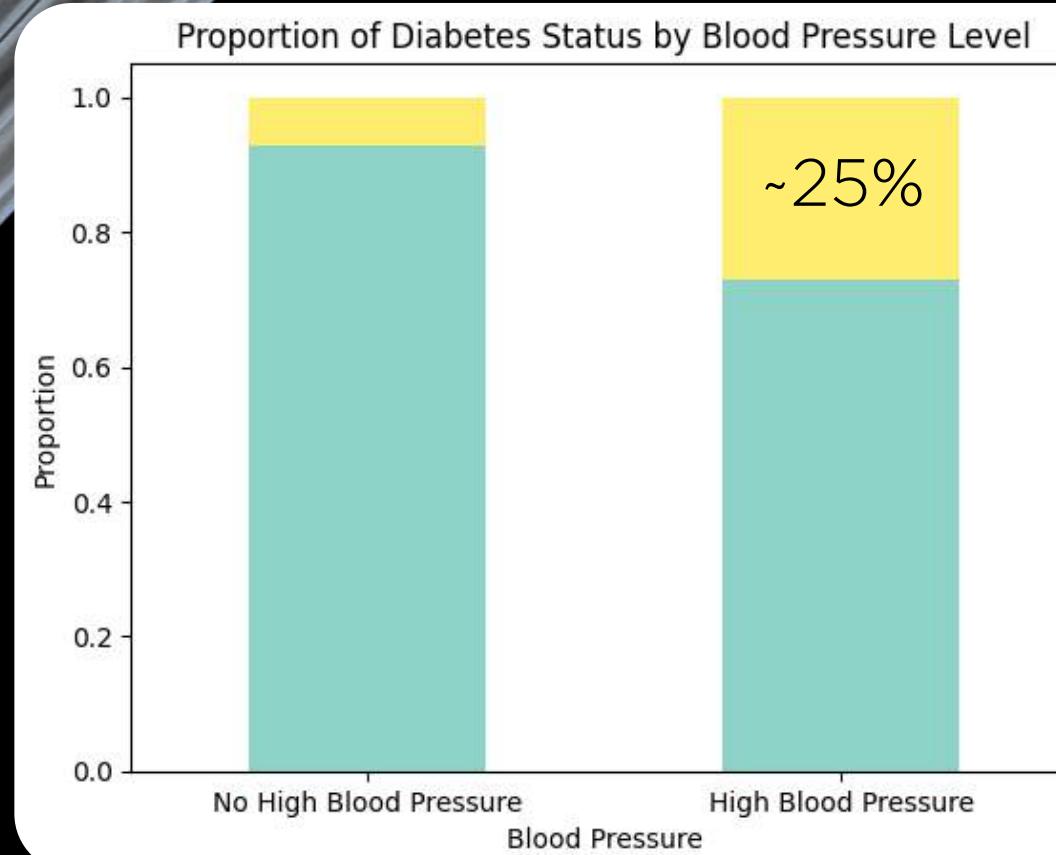


# Health Indicators with Diabetes Risk

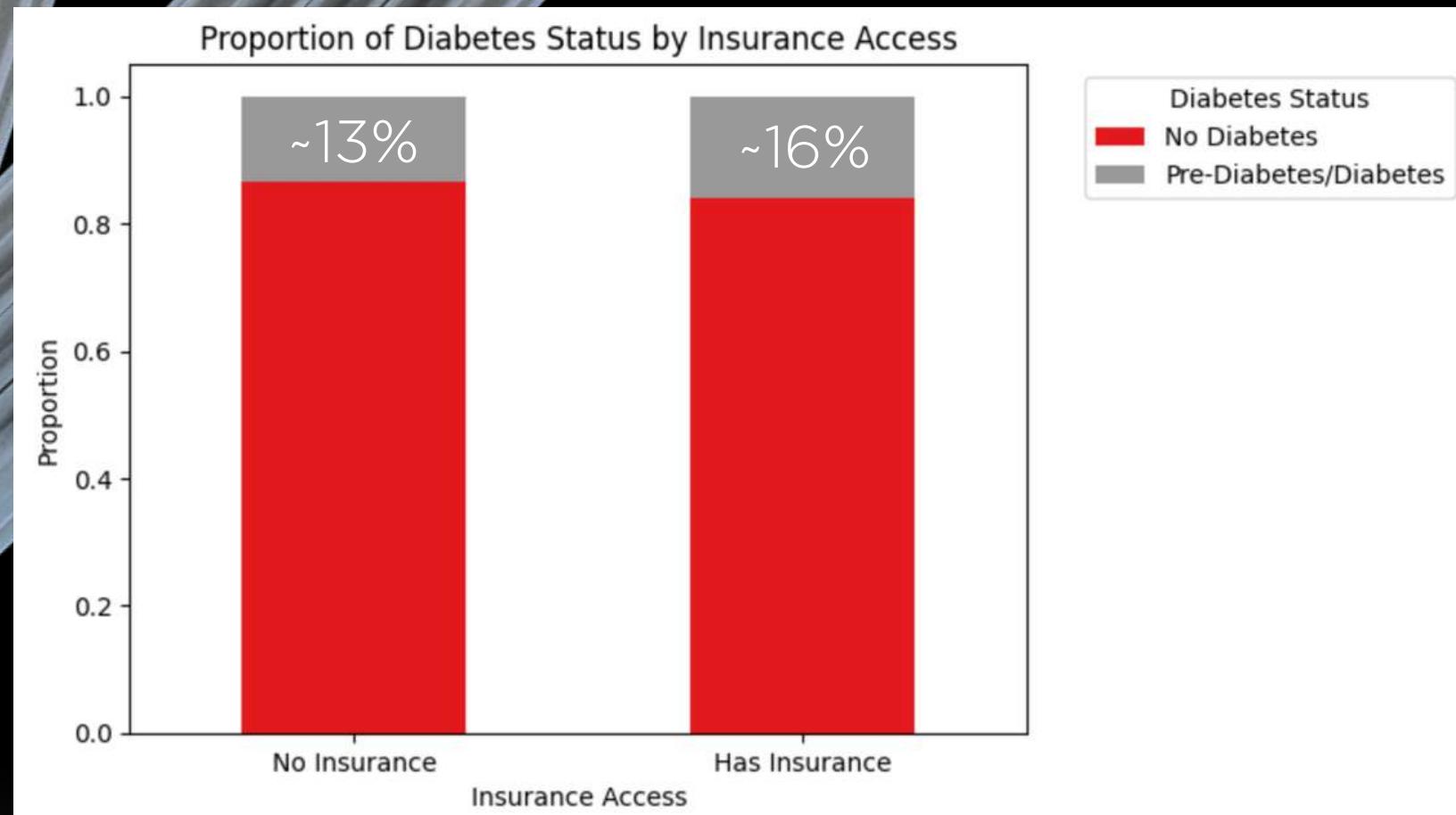


- Statistically Significant Difference: The two-sample t-test ( $p < 0.05$ ) shows that individuals with diabetes have a higher mean BMI (31.80) than those without (27.74).
- Clinical & Practical Implications: Obesity remains a key modifiable risk factor. As BMI climbs above 25 (Overweight) and above 30 (Obese), the prevalence of diabetes rises sharply.

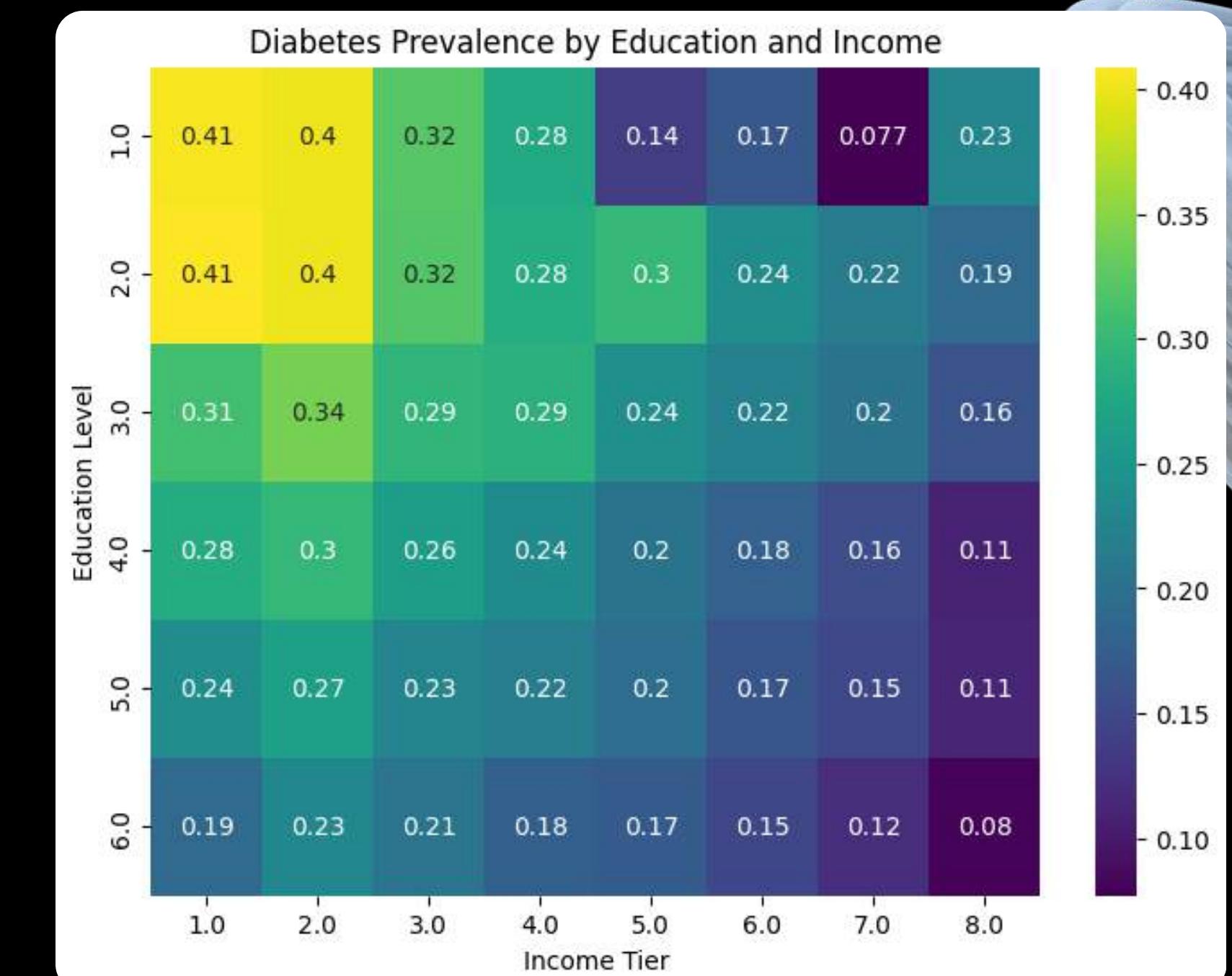
# Health Indicators with Diabetes Risk



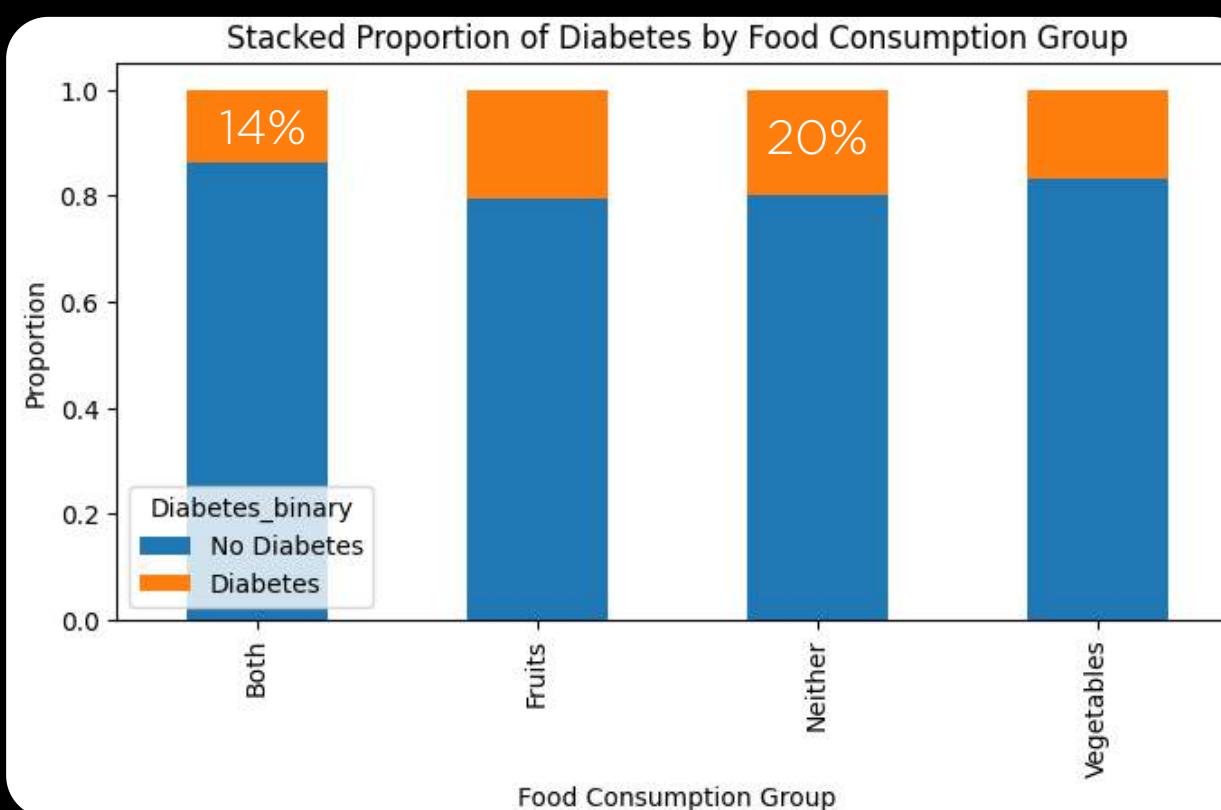
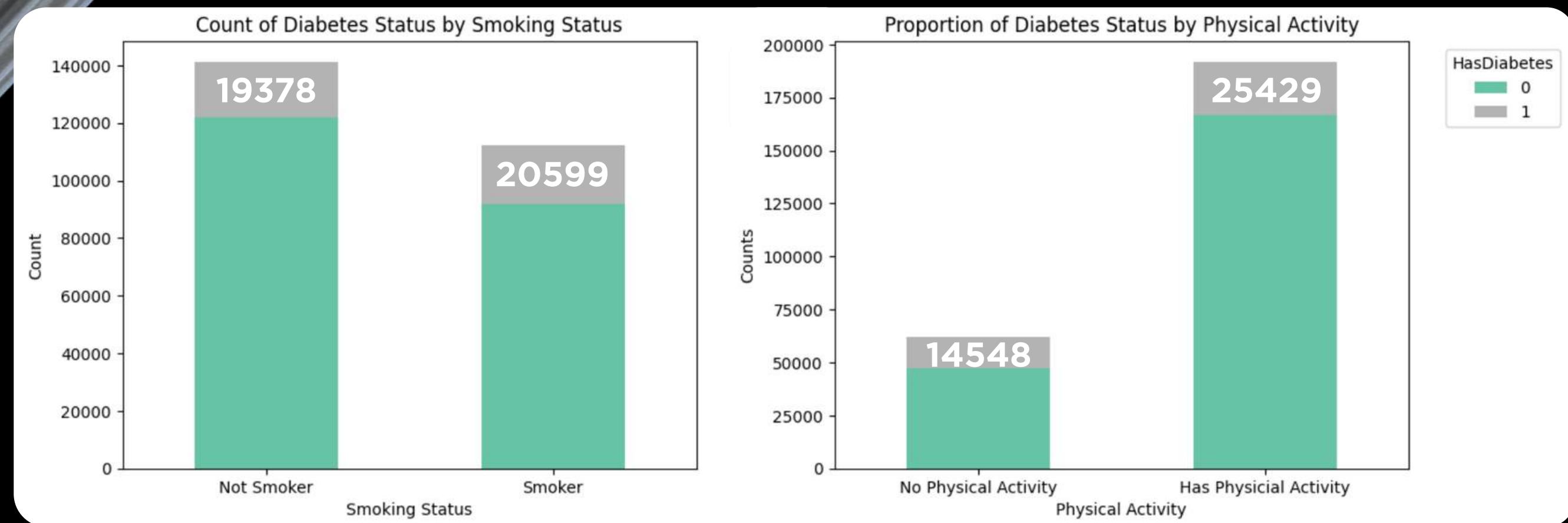
# Social Determinants of Health and Diabetes



Diabetes_binary	0	1
AnyHealthcare	0	1
0	0.865024	0.134976
1	0.841248	0.158752



# Lifestyle Indicators and Diabetes



# Predictive Modeling

Models  
Tested

Logistic Regression

Decision Tree

Random Forest

XGBoost

Handling  
Class  
Imbalance

Undersampling

SMOTE

# Predictive Modeling

## Handling imbalance using *SMOTE*

	Model	Accuracy	Precision	Recall	F1-score
0	Logistic Regression	0.704707	0.617479	0.703638	0.614648
1	Decision Tree	0.720974	0.571877	0.603550	0.575960
2	Random Forest	0.730829	0.588563	0.629396	0.594985
3	XGBoost	0.712564	0.614290	0.692982	0.615122

**BEST MODEL : XGBoost model with the highest F1-score (0.615122)**

- The F1-score provides a balanced view of both precision and recall.
- This balance is crucial in medical diagnostics, where both false positives and false negatives can have serious consequences.
- The F1-score works well on skewed datasets, making it more reliable than accuracy.
- It ensures that neither precision nor recall is disproportionately favored, which is vital in healthcare decisions.

# Predictive Modeling

## Handling imbalance using *undersampling*

	Model	Accuracy	Precision	Recall	F1-score
0	Logistic Regression	0.739567	0.740075	0.739617	0.739455
1	Decision Tree	0.654396	0.654424	0.654378	0.654363
2	Random Forest	0.734314	0.735666	0.734397	0.733977
3	XGBoost	0.521574	0.683296	0.522609	0.388100

**BEST MODEL : Logistic Regression model with the highest F1-score (0.739455)**

- The F1-score provides a balanced view of both precision and recall.
- This balance is crucial in medical diagnostics, where both false positives and false negatives can have serious consequences.
- The F1-score works well on skewed datasets, making it more reliable than accuracy.
- It ensures that neither precision nor recall is disproportionately favored, which is vital in healthcare decisions.

# Feature Selection with SFS

Final Model Comparison:

	Model Selection	Type	Accuracy	Precision	Recall	F1-score
0	Logistic Regression	Forward	0.7408	0.7294	0.7637	0.7462
1	Logistic Regression	Backward	0.7395	0.7284	0.7620	0.7448

**Feature Selection for Logistic Regression model(Forward Selection):**  
('HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'HeartDiseaseorAttack',  
'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'GenHlth',  
'MentHlth', 'DiffWalk', 'Sex', 'Age', 'Education', 'Income')

# Predictive Modeling

## Logistic Regression Model & Forward Selection

	Pred:0	Pred:1
Actual:0	8621	3399
Actual:1	2848	9119

```
Testing Model Prediction

[111]: 1 # Making predictions for patient with
2 #'HighBP':1, 'HighChol':1, 'CholCheck':1, 'BMI':28, 'Smoker':0, 'Stroke':1 'HeartDiseaseorAttack':1, 'PhysActivity':1, 'Fruit':0, 'Veggies':0,
3 #'HvyAlcoholConsump':0, 'AnyHealthcare':1, 'NoDocbcCost: 0', 'GenHlth':4, 'MenHlth':5', 'PhysHlth':15, 'DiffWalk':1, 'Sex':1, 'Age: 5', 'Education: 5', 'Income: 5'
4
5 new_sample_values = [[1, 1, 1, 28, 0, 1, 1, 1, 0, 0,
6                 0, 1, 0, 4, 5, 15, 1, 1,
7                 5, 5, 5]]
8
9 new_sample_raw = pd.DataFrame(new_sample_values, columns=original_features)
10
11 # Scale the new sample using the same scaler
12 new_sample_scaled = pd.DataFrame(scaler.transform(new_sample_raw),
13                                     columns=original_features)
14
15 # Apply the feature selection transformation to get only the selected features
16 new_sample_sfs = sfs_fw.transform(new_sample_scaled)
17
18 # Predict the probability
19 probabilities = log_reg_us.predict_proba(new_sample_sfs)
20
21 print("Probabilities:", probabilities)
22
23 outcome = log_reg_us.predict(new_sample_sfs)
24 print("Outcome:", outcome)

Probabilities: [[0.25225306 0.74774694]]
Outcome: [1]
```

- In our selected model, FP is higher than FN
- Even though we aim for a balance in precision and recall, having a higher FP is more ideal in this case, as it can prompt further diagnostic testing for patients
- After using Forward Selection, our F1 score improved to 0.746
- Upon prediction, probability for Class 0 (Non-Diabetic) is 0.25 | Class 1 (Pre/Diabetic) is 0.75

# Prescriptive Modeling

```
# Get the predicted risk using the trained model
risk = log_reg_us.predict_proba(new_sample_sfs)[:, 1][0]

# Categorize risk
if risk < risk_threshold["low"]:
    risk_category = "Low"
elif risk < risk_threshold["medium"]:
    risk_category = "Medium"
else:
    risk_category = "High"

# Risk-Based Recommendations
recommendation = []
if risk_category == "Low":
    recommendation.append(
        "Your diabetes risk is low. Please maintain a healthy lifestyle, engage in frequent physical activity,\n"
        "and follow regular check-ups to promote your overall well-being.")
elif risk_category == "Medium":
    recommendation.append(
        "Your diabetes risk is medium. Please make modifications in your lifestyle and diets to mitigate this risk.\n"
        "A screening or check-up with your Primary Care Physician is highly recommended for early detection of potential health issues.")
else:
    recommendation.append(
        "Your diabetes risk is high. Please consult with medical professionals accordingly.\n"
        "In the meantime, consider managing your risk by modifying your lifestyle, diets, and physical activities.")
```

- Provide recommendations based on patient's risk (low, medium, high)
- Recommendations based on trustworthy resources from CDC, American Diabetes Association, National Library of Medicine

```
# --- Prescriptive Rules (for Medium or High risk) ---
if risk_category in ["Medium", "High"]:
    if input_data.get('BMI', 0) > 25:
        recommendation.append(
            "Your BMI indicates overweight or obese status, which increases your risk of developing diabetes.\n"
            "Consider following a weight management plan, such as managing your diet intake and introducing regular exercise to your daily routine.")

    if input_data.get('PhysActivity', 0) == 0:
        recommendation.append(
            "Physical activity is fundamental in diabetes management.\n"
            "The American Diabetes Association recommends at least 150 minutes of moderate-to-vigorous aerobic activity per week.\n"
            "Consider scheduling short, regular walks or other activities to build up your routine.")

    if input_data.get('Fruits', 0) == 1 and input_data.get('Veggies', 0) == 0:
        recommendation.append(
            "While fruits are a healthy source of nutrients, excessive fruit intake (over 2 servings/day) without adequate vegetables\n"
            "could result in higher energy intake. Consider incorporating more vegetables along with fruits for a balanced diet.")

    if input_data.get('Fruits', 0) == 0 and input_data.get('Veggies', 0) == 0:
        recommendation.append(
            "A nutritious, balanced diet is essential for blood sugar management.\n"
            "Consider adding fruits—especially berries—and a variety of vegetables, including dark green and cruciferous options, to your meals.")

    if input_data.get('Smoker', 0) == 1:
        recommendation.append(
            "As smoking increases the risk of type 2 diabetes by 30–40%, quitting smoking can contribute to better blood sugar control.\n"
            "Discuss available nicotine replacement therapies with a healthcare provider.")

    if input_data.get('HighBP', 0) == 1:
        recommendation.append(
            "For those with high blood pressure, following the DASH (Dietary Approaches to Stop Hypertension) plan can help mitigate diabetes risk.\n"
            "This includes limiting sodium intake and focusing on nutrient-rich foods.")

return risk, recommendation
```

# Prescriptive Modeling

```
▶ 1 example_patient = {  
2     'HighBP': 1,  
3     'HighChol': 1,  
4     'CholCheck': 1,  
5     'BMI': 28,  
6     'Smoker': 0,  
7     'Stroke': 1,  
8     'HeartDiseaseorAttack': 1,  
9     'PhysActivity': 1,  
10    'Fruits': 1,  
11    'Veggies': 1,  
12    'HvyAlcoholConsump': 0,  
13    'AnyHealthcare': 1,  
14    'NoDocbcCost': 0,  
15    'GenHlth': 4,  
16    'MentHlth': 5,  
17    'PhysHlth': 5,  
18    'DiffWalk': 0,  
19    'Sex': 1,  
20    'Age': 5,  
21    'Education': 5,  
22    'Income': 5  
23 }  
24  
25 predicted_risk, suggested_actions = prescribe_interventions(  
26     example_patient,  
27     risk_threshold={"low": 0.3, "medium": 0.5})  
28  
29 print("Predicted Diabetes Risk: {:.1%}".format(predicted_risk))  
30 print("Recommended Interventions:")  
31 for action in suggested_actions:  
32     print("-", action)  
  
→ Predicted Diabetes Risk: 72.2%  
Recommended Interventions:  
- Your diabetes risk is high. Please consult with medical professionals accordingly.  
In the meantime, consider managing your risk by modifying your lifestyle, diets, and physical activities.  
- Your BMI indicates overweight or obese status, which increases your risk of developing diabetes.  
Consider following a weight management plan, such as managing your diet intake and introducing regular exercise to your daily routine.  
- For those with high blood pressure, following the DASH (Dietary Approaches to Stop Hypertension) plan can help mitigate diabetes risk.  
This includes limiting sodium intake and focusing on nutrient-rich foods.
```

# INSIGHTS

1

**Top Health & Lifestyle Indicators Correlates with Diabetes**

Blood Pressure, Cholesterol, BMI, and Mobility, Mental Well-Beings, Food & Alcohol assumptions

2

**Social Determinants of Health Interact Differently with Diabetes**

While lower Income & Education levels showed high correlation with diabetes, health care access and equity have mix indication of diabetes prevalence

3

**Class Imbalance & Survey Bias & Feature Informativeness Influenced Models' Performance**

Despite applying different techniques and models, we still cannot raise our F1 scores to above 0.9 (which is essential in healthcare)



**THANK  
YOU**