```
pip install ydata-profiling
```

```
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/python3.10/dist-packages (from ydat
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (fro
Collecting multimethod<2,>=1.4 (from ydata-profiling)
  Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: typeguard<5,>=3 in /usr/local/lib/python3.10/dist-packages (from ydat
Collecting imagehash==4.3.1 (from ydata-profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in /usr/local/lib/python3.10/dist-packages (from yda
Collecting dacite>=1.8 (from ydata-profiling)
  Downloading dacite-1.8.1-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: numba<1,>=0.56.0 in /usr/local/lib/python3.10/dist-packages (from yda
Collecting PyWavelets (from imagehash==4.3.1->ydata-profiling)
  Downloading pywavelets-1.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from imagehash==4
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jin
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplot
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matp
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (fror
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from panda
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-packages (from phik<
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from request
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from statsmo
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.10/dist-packages (from visior
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.10/dist-packages (from visior
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-date
Downloading ydata_profiling-4.12.0-py2.py3-none-any.whl (390 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 390.6/390.6 kB 7.4 MB/s eta 0:00:00
Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.5/296.5 kB 13.8 MB/s eta 0:00:00
Downloading dacite-1.8.1-py3-none-any.whl (14 kB)
Downloading multimethod-1.12-py3-none-any.whl (10 kB)
Downloading phik-0.12.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (686 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 686.1/686.1 kB 18.6 MB/s eta 0:00:00
Downloading visions-0.7.6-py3-none-any.whl (104 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.8/104.8 kB 5.7 MB/s eta 0:00:00
Downloading pywavelets-1.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.5/4.5 MB 33.2 MB/s eta 0:00:00
Building wheels for collected packages: htmlmin
  Building wheel for htmlmin (setup.py) ... done
  Created wheel for htmlmin: filename=htmlmin-0.1.12-py3-none-any.whl size=27081 sha256=7e1740fa729
  Stored in directory: /root/.cache/pip/wheels/dd/91/29/a79cecb328d01739e64017b6fb9a1ab9d8cb1853098
Successfully built htmlmin
Installing collected packages: htmlmin, PyWavelets, multimethod, dacite, imagehash, visions, phik, y
Successfully installed PyWavelets-1.7.0 dacite-1.8.1 htmlmin-0.1.12 imagehash-4.3.1 multimethod-1.1
```

```
import seaborn as sns
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from ydata_profiling import ProfileReport
```

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Project Overview

The objective of this project is to develop a predictive model to estimate the likelihood of diabetes based on health-related features in the dataset. The process includes the following steps:

1. **Data Exploration**: Analyze the dataset to understand its structure, key features, and the target variable.
2. **Data Preprocessing**: Prepare the data by cleaning it, handling missing values and outliers, and applying normalization techniques.
3. **Feature Engineering**: Identify and select the most significant features that contribute to accurate predictions.
4. **Model Development**: Utilize Python libraries like Scikit-Learn to build machine learning models.
5. **Model Evaluation**: Measure the model's performance using metrics such as accuracy, precision, recall, and F1 score.

Data Overview

```
# Import the Diabetes dataset from the specified CSV file using Pandas
df1 = pd.read_csv('/content/drive/MyDrive/diabetes_012_health_indicators_BRFSS2015.csv')
df1
```

3MI	Smoker	Stroke	HeartDiseaseorAttack	PhysActiv
0.0	1.0	0.0	0.0	
5.0	1.0	0.0	0.0	
8.0	0.0	0.0	0.0	
7.0	0.0	0.0	0.0	
4.0	0.0	0.0	0.0	
...	...	...	...	
5.0	0.0	0.0	0.0	
8.0	0.0	0.0	0.0	
8.0	0.0	0.0	0.0	
3.0	0.0	0.0	0.0	
5.0	0.0	0.0	1.0	

Basic Metrics

```
# Checking Shape of data
df1.shape
```

```
(253680, 22)
```

```
# Columns in data
df1.columns
```

```
Index(['Diabetes_012', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker',
       'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
       'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
       'Income'],
      dtype='object')
```

```
# Overview of the dataset structure
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Diabetes_012          253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   HeartDiseaseorAttack  253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
 14  GenHlth               253680 non-null  float64
 15  MentHlth              253680 non-null  float64
 16  PhysHlth              253680 non-null  float64
 17  DiffWalk              253680 non-null  float64
 18  Sex                   253680 non-null  float64
 19  Age                   253680 non-null  float64
 20  Education             253680 non-null  float64
 21  Income                253680 non-null  float64
dtypes: float64(22)
memory usage: 42.6 MB
```

Data Cleaning

```
# Check for missing or null values in the diabetes dataset and count them for each column
df1.isna().sum()
```

	0
Diabetes_012	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

dtype: int64

```
df1.groupby('Diabetes_012').apply(lambda x: x.count())
```

```
Warning: DataFrameGroupBy.apply operated on the grouping 
x.count())
```

CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	P
213703	213703	213703	213703	213703	
4631	4631	4631	4631	4631	
35346	35346	35346	35346	35346	

Descriptive Statistics

```
# Summary statistics for numerical columns
df1.describe(include = 'all').T
```

	count	mean	std	min	25%	50%	75%	max
Diabetes_012	253680.0	0.296921	0.698160	0.0	0.0	0.0	0.0	2.0
HighBP	253680.0	0.429001	0.494934	0.0	0.0	0.0	1.0	1.0
HighChol	253680.0	0.424121	0.494210	0.0	0.0	0.0	1.0	1.0
CholCheck	253680.0	0.962670	0.189571	0.0	1.0	1.0	1.0	1.0
BMI	253680.0	28.382364	6.608694	12.0	24.0	27.0	31.0	98.0
Smoker	253680.0	0.443169	0.496761	0.0	0.0	0.0	1.0	1.0
Stroke	253680.0	0.040571	0.197294	0.0	0.0	0.0	0.0	1.0
HeartDiseaseorAttack	253680.0	0.094186	0.292087	0.0	0.0	0.0	0.0	1.0
PhysActivity	253680.0	0.756544	0.429169	0.0	1.0	1.0	1.0	1.0
Fruits	253680.0	0.634256	0.481639	0.0	0.0	1.0	1.0	1.0
Veggies	253680.0	0.811420	0.391175	0.0	1.0	1.0	1.0	1.0
HvyAlcoholConsump	253680.0	0.056197	0.230302	0.0	0.0	0.0	0.0	1.0
AnyHealthcare	253680.0	0.951053	0.215759	0.0	1.0	1.0	1.0	1.0
NoDocbcCost	253680.0	0.084177	0.277654	0.0	0.0	0.0	0.0	1.0
GenHlth	253680.0	2.511392	1.068477	1.0	2.0	2.0	3.0	5.0
MentHlth	253680.0	3.184772	7.412847	0.0	0.0	0.0	2.0	30.0
PhysHlth	253680.0	4.242081	8.717951	0.0	0.0	0.0	3.0	30.0
DiffWalk	253680.0	0.168224	0.374066	0.0	0.0	0.0	0.0	1.0
Sex	253680.0	0.440342	0.496429	0.0	0.0	0.0	1.0	1.0
Age	253680.0	8.032119	3.054220	1.0	6.0	8.0	10.0	13.0
Education	253680.0	5.050434	0.985774	1.0	4.0	5.0	6.0	6.0
Income	253680.0	6.053875	2.071148	1.0	5.0	7.0	8.0	8.0

```
# Select numeric columns from the dataset
numeric_cols = df1.select_dtypes(include=['number']).columns
print("Numeric columns in the dataset:")
print(numeric_cols)
```

```
Numeric columns in the dataset:
Index(['Diabetes_012', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker',
       'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
       'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
       'Income'],
      dtype='object')
```

```
# Variance measures how much the data is spread out from the mean.
variance_values = df1[numeric_cols].var()
print("\nVariance for each numeric column:")
variance_values
```

```
Variance for each numeric column:
```

	0
Diabetes_012	0.487427
HighBP	0.244960
HighChol	0.244243
CholCheck	0.035937
BMI	43.674839
Smoker	0.246771
Stroke	0.038925
HeartDiseaseorAttack	0.085315
PhysActivity	0.184186
Fruits	0.231976
Veggies	0.153018
HvyAlcoholConsump	0.053039
AnyHealthcare	0.046552
NoDocbcCost	0.077091
GenHlth	1.141644
MentHlth	54.950296
PhysHlth	76.002675
DiffWalk	0.139925
Sex	0.246442
Age	9.328262
Education	0.971751
Income	4.289652

dtype: float64

```
# Covariance measures how two variables move together. Positive covariance indicates that the variables ter
covariance_matrix = df1[numeric_cols].cov()
print("\nCovariance matrix:")
covariance_matrix
```

```
Covariance matrix:
```

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartD
Diabetes_012	0.487427	0.093848	0.072142	0.008940	1.035270	0.021820	0.014763	
HighBP	0.093848	0.244960	0.072940	0.009243	0.699142	0.023847	0.012653	
HighChol	0.072142	0.072940	0.244243	0.008024	0.348563	0.022414	0.009031	
CholCheck	0.008940	0.009243	0.008024	0.035937	0.043216	-0.000935	0.000904	
BMI	1.035270	0.699142	0.348563	0.043216	43.674839	0.045319	0.026276	
Smoker	0.021820	0.023847	0.022414	-0.000935	0.045319	0.246771	0.005995	
Stroke	0.014763	0.012653	0.009031	0.000904	0.026276	0.005995	0.038925	
HeartDiseaseorAttack	0.036762	0.030266	0.026094	0.002448	0.102122	0.016605	0.011698	
PhysActivity	-0.036539	-0.026608	-0.016554	0.000341	-0.417761	-0.018633	-0.005855	
Fruits	-0.014187	-0.009667	-0.009726	0.002178	-0.278571	-0.018582	-0.001272	
Veggies	-0.016105	-0.011862	-0.007708	0.000454	-0.160991	-0.005961	-0.003174	
HvyAlcoholConsump	-0.009307	-0.000453	-0.001314	-0.001036	-0.074176	0.011626	-0.000770	
AnyHealthcare	0.002321	0.004103	0.004503	0.004811	-0.026337	-0.002492	0.000374	
NoDocbcCost	0.006869	0.002385	0.001826	-0.003066	0.106804	0.006751	0.001907	
GenHlth	0.225720	0.158928	0.110060	0.009437	1.688945	0.086593	0.037511	
MentHlth	0.380423	0.207130	0.227390	-0.011756	4.179280	0.339505	0.102627	
PhysHlth	1.072973	0.695598	0.524562	0.052513	6.979457	0.504356	0.256184	
DiffWalk	0.058562	0.041400	0.026745	0.002878	0.487193	0.022756	0.013031	
Sex	0.010758	0.012827	0.007656	-0.002081	0.140909	0.023098	0.000292	
Age	0.394537	0.520688	0.411044	0.052295	-0.739105	0.183039	0.076512	
Education	-0.089825	-0.068968	-0.034493	0.000282	-0.677084	-0.079308	-0.014783	
Income	-0.247963	-0.175530	-0.087475	0.005598	-1.369699	-0.127515	-0.052549	

22 rows × 22 columns

```
# Correlation measures the strength and direction of the relationship between two variables. It ranges from
correlation_table = df1[numeric_cols].corr()
print("\nCorrelation table:")
correlation_table
```

Correlation table:

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartD
Diabetes_012	1.000000	0.271596	0.209085	0.067546	0.224379	0.062914	0.107179	
HighBP	0.271596	1.000000	0.298199	0.098508	0.213748	0.096991	0.129575	
HighChol	0.209085	0.298199	1.000000	0.085642	0.106722	0.091299	0.092620	
CholCheck	0.067546	0.098508	0.085642	1.000000	0.034495	-0.009929	0.024158	
BMI	0.224379	0.213748	0.106722	0.034495	1.000000	0.013804	0.020153	
Smoker	0.062914	0.096991	0.091299	-0.009929	0.013804	1.000000	0.061173	
Stroke	0.107179	0.129575	0.092620	0.024158	0.020153	0.061173	1.000000	
HeartDiseaseorAttack	0.180272	0.209361	0.180765	0.044206	0.052904	0.114441	0.203002	
PhysActivity	-0.121947	-0.125267	-0.078046	0.004190	-0.147294	-0.087401	-0.069151	
Fruits	-0.042192	-0.040555	-0.040859	0.023849	-0.087518	-0.077666	-0.013389	
Veggies	-0.058972	-0.061266	-0.039874	0.006121	-0.062275	-0.030678	-0.041124	
HvyAlcoholConsump	-0.057882	-0.003972	-0.011543	-0.023730	-0.048736	0.101619	-0.016950	
AnyHealthcare	0.015410	0.038425	0.042230	0.117626	-0.018471	-0.023251	0.008776	
NoDocbcCost	0.035436	0.017358	0.013310	-0.058255	0.058206	0.048946	0.034804	
GenHlth	0.302587	0.300530	0.208426	0.046589	0.239185	0.163143	0.177942	
MentHlth	0.073507	0.056456	0.062069	-0.008366	0.085310	0.092196	0.070172	
PhysHlth	0.176287	0.161212	0.121751	0.031775	0.121141	0.116460	0.148944	
DiffWalk	0.224239	0.223618	0.144672	0.040585	0.197078	0.122463	0.176567	
Sex	0.031040	0.052207	0.031205	-0.022115	0.042950	0.093662	0.002978	
Age	0.185026	0.344452	0.272318	0.090321	-0.036618	0.120641	0.126974	
Education	-0.130517	-0.141358	-0.070802	0.001510	-0.103932	-0.161955	-0.076009	
Income	-0.171483	-0.171235	-0.085459	0.014259	-0.100069	-0.123937	-0.128599	

22 rows × 22 columns

```
# Mode is the value that appears most frequently in a dataset.
mode_values = df1.mode().iloc[0]  # Using iloc[0] to get the mode for each column
print("Mode for each column:")
mode_values
```

```
Mode for each column:
```

	0
Diabetes_012	0.0
HighBP	0.0
HighChol	0.0
CholCheck	1.0
BMI	27.0
Smoker	0.0
Stroke	0.0
HeartDiseaseorAttack	0.0
PhysActivity	1.0
Fruits	1.0
Veggies	1.0
HvyAlcoholConsump	0.0
AnyHealthcare	1.0
NoDocbcCost	0.0
GenHlth	2.0
MentHlth	0.0
PhysHlth	0.0
DiffWalk	0.0
Sex	0.0
Age	9.0
Education	6.0
Income	8.0

dtype: float64

```
# Range is the difference between the maximum and minimum values in a dataset.
range_values = df1[numeric_cols].max() - df1[numeric_cols].min()
print("\nRange for each numeric column:")
range_values
```

```
Range for each numeric column:
```

	0
Diabetes_012	2.0
HighBP	1.0
HighChol	1.0
CholCheck	1.0
BMI	86.0
Smoker	1.0
Stroke	1.0
HeartDiseaseorAttack	1.0
PhysActivity	1.0
Fruits	1.0
Veggies	1.0
HvyAlcoholConsump	1.0
AnyHealthcare	1.0
NoDocbcCost	1.0
GenHlth	4.0
MentHlth	30.0
PhysHlth	30.0
DiffWalk	1.0
Sex	1.0
Age	12.0
Education	5.0
Income	7.0

dtype: float64

```
# Range is the difference between the maximum and minimum values in a dataset.
range_values = df1[numeric_cols].max() - df1[numeric_cols].min()
print("\nRange for each numeric column:")
range_values
```

```
Range for each numeric column:
```

	0
Diabetes_012	2.0
HighBP	1.0
HighChol	1.0
CholCheck	1.0
BMI	86.0
Smoker	1.0
Stroke	1.0
HeartDiseaseorAttack	1.0
PhysActivity	1.0
Fruits	1.0
Veggies	1.0
HvyAlcoholConsump	1.0
AnyHealthcare	1.0
NoDocbcCost	1.0
GenHlth	4.0
MentHlth	30.0
PhysHlth	30.0
DiffWalk	1.0
Sex	1.0
Age	12.0
Education	5.0
Income	7.0

dtype: float64

```
# Convert any potential strings in numeric columns to NaN (if they exist)
df[numeric_cols] = df1[numeric_cols].apply(pd.to_numeric, errors='coerce')
Q1 = df1[numeric_cols].quantile(0.25)
Q3 = df1[numeric_cols].quantile(0.75)
IQR = Q3 - Q1
print("\nInterquartile Range (IQR) for each numeric column:")
IQR
```

```
Interquartile Range (IQR) for each numeric column:
```

	0
Diabetes_012	0.0
HighBP	1.0
HighChol	1.0
CholCheck	0.0
BMI	7.0
Smoker	1.0
Stroke	0.0
HeartDiseaseorAttack	0.0
PhysActivity	0.0
Fruits	1.0
Veggies	0.0
HvyAlcoholConsump	0.0
AnyHealthcare	0.0
NoDocbcCost	0.0
GenHlth	1.0
MentHlth	2.0
PhysHlth	3.0
DiffWalk	0.0
Sex	1.0
Age	4.0
Education	2.0
Income	3.0

dtype: float64

```
# Detecting outliers based on IQR
outliers = df1[((df1[numeric_cols] < (Q1 - 1.5 * IQR)) | (df1[numeric_cols] > (Q3 + 1.5 * IQR))).any(axis=1
print("Outliers:", outliers)
```

```
Outliers:         Diabetes_012  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \
0                0.0     1.0       1.0        1.0  40.0     1.0     0.0
1                0.0     0.0       0.0        0.0  25.0     1.0     0.0
2                0.0     1.0       1.0        1.0  28.0     0.0     0.0
6                0.0     1.0       0.0        1.0  30.0     1.0     0.0
7                0.0     1.0       1.0        1.0  25.0     1.0     0.0
...              ...     ...       ...        ...   ...     ...     ...
253675           0.0     1.0       1.0        1.0  45.0     0.0     0.0
253676           2.0     1.0       1.0        1.0  18.0     0.0     0.0
253677           0.0     0.0       0.0        1.0  28.0     0.0     0.0
253678           0.0     1.0       0.0        1.0  23.0     0.0     0.0
253679           2.0     1.0       1.0        1.0  25.0     0.0     0.0

        HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare  \
0                        0.0           0.0     0.0  ...            1.0
1                        0.0           1.0     0.0  ...            0.0
2                        0.0           0.0     1.0  ...            1.0
6                        0.0           0.0     0.0  ...            1.0
7                        0.0           1.0     0.0  ...            1.0
```

```
...                    ...          ...     ...  ...           ...
253675                 0.0          0.0     1.0  ...           1.0
253676                 0.0          0.0     0.0  ...           1.0
253677                 0.0          1.0     1.0  ...           1.0
253678                 0.0          0.0     1.0  ...           1.0
253679                 1.0          1.0     1.0  ...           1.0

        NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  \
0               0.0      5.0      18.0      15.0       1.0  0.0   9.0
1               1.0      3.0       0.0       0.0       0.0  0.0   7.0
2               1.0      5.0      30.0      30.0       1.0  0.0   9.0
6               0.0      3.0       0.0      14.0       0.0  0.0   9.0
7               0.0      3.0       0.0       0.0       1.0  0.0  11.0
...             ...      ...       ...       ...       ...  ...   ...
253675          0.0      3.0       0.0       5.0       0.0  1.0   5.0
253676          0.0      4.0       0.0       0.0       1.0  0.0  11.0
253677          0.0      1.0       0.0       0.0       0.0  0.0   2.0
253678          0.0      3.0       0.0       0.0       0.0  1.0   7.0
253679          0.0      2.0       0.0       0.0       0.0  0.0   9.0

        Education  Income
0             4.0     3.0
1             6.0     1.0
2             4.0     8.0
6             6.0     7.0
7             4.0     4.0
...           ...     ...
253675        6.0     7.0
253676        2.0     4.0
253677        5.0     2.0
253678        5.0     1.0
253679        6.0     2.0

[168001 rows x 22 columns]
```

```
outliers
```

Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActiv
0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	
0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	
0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	
0.0	1.0	0.0	1.0	30.0	1.0	0.0	0.0	
0.0	1.0	1.0	1.0	25.0	1.0	0.0	0.0	
...	...	...	...	...	...	...	...	
0.0	1.0	1.0	1.0	45.0	0.0	0.0	0.0	
2.0	1.0	1.0	1.0	18.0	0.0	0.0	0.0	
0.0	0.0	0.0	1.0	28.0	0.0	0.0	0.0	
0.0	1.0	0.0	1.0	23.0	0.0	0.0	0.0	
2.0	1.0	1.0	1.0	25.0	0.0	0.0	1.0	

ws × 22 columns

Data Visualization

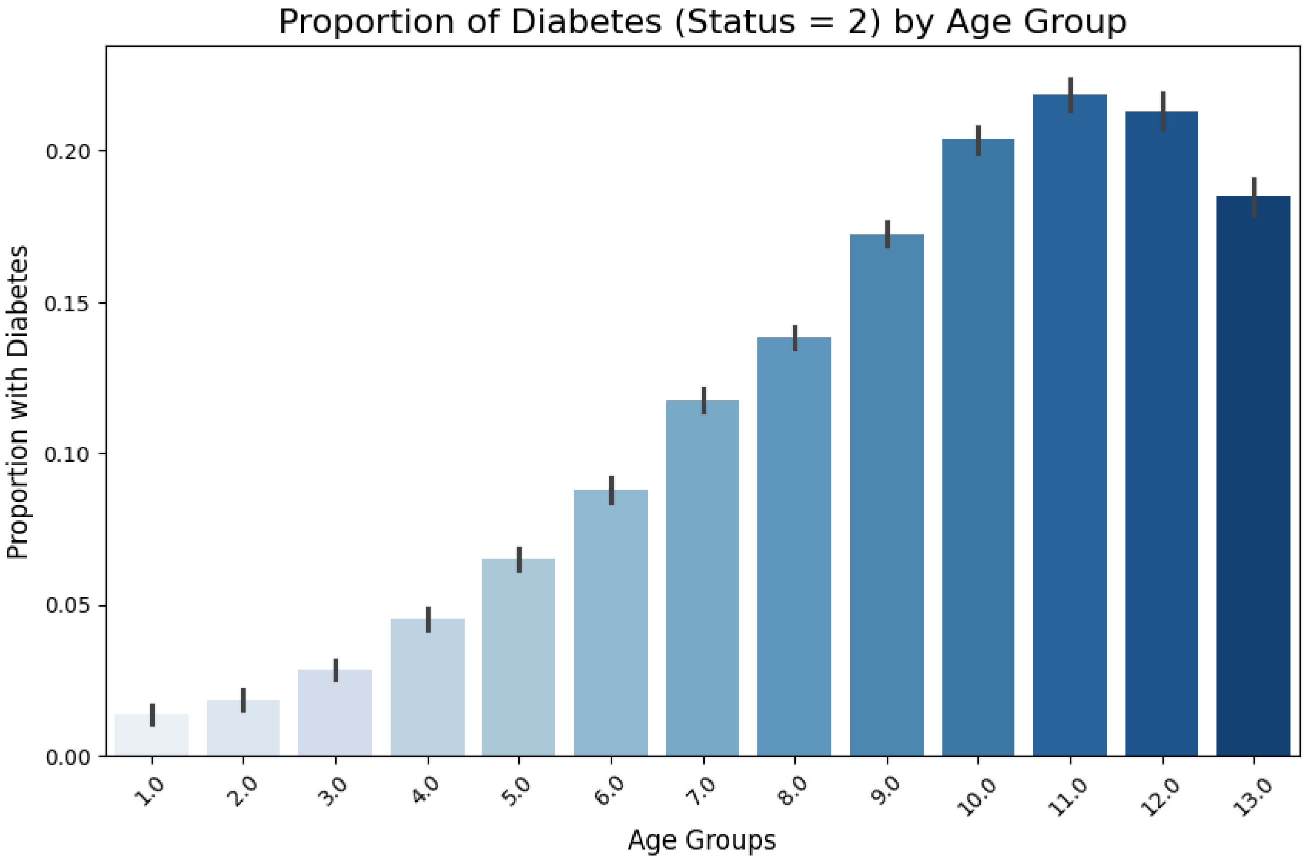
```
# 1. Diabetes Status by Age Group
plt.figure(figsize=(10, 6))
```

```
sns.barplot(x='Age', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palette='Blues')
plt.title('Proportion of Diabetes (Status = 2) by Age Group', fontsize=16)
plt.xlabel('Age Groups', fontsize=12)
plt.ylabel('Proportion with Diabetes', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-36-46aeb23355b2>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.barplot(x='Age', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palette='Blu
```

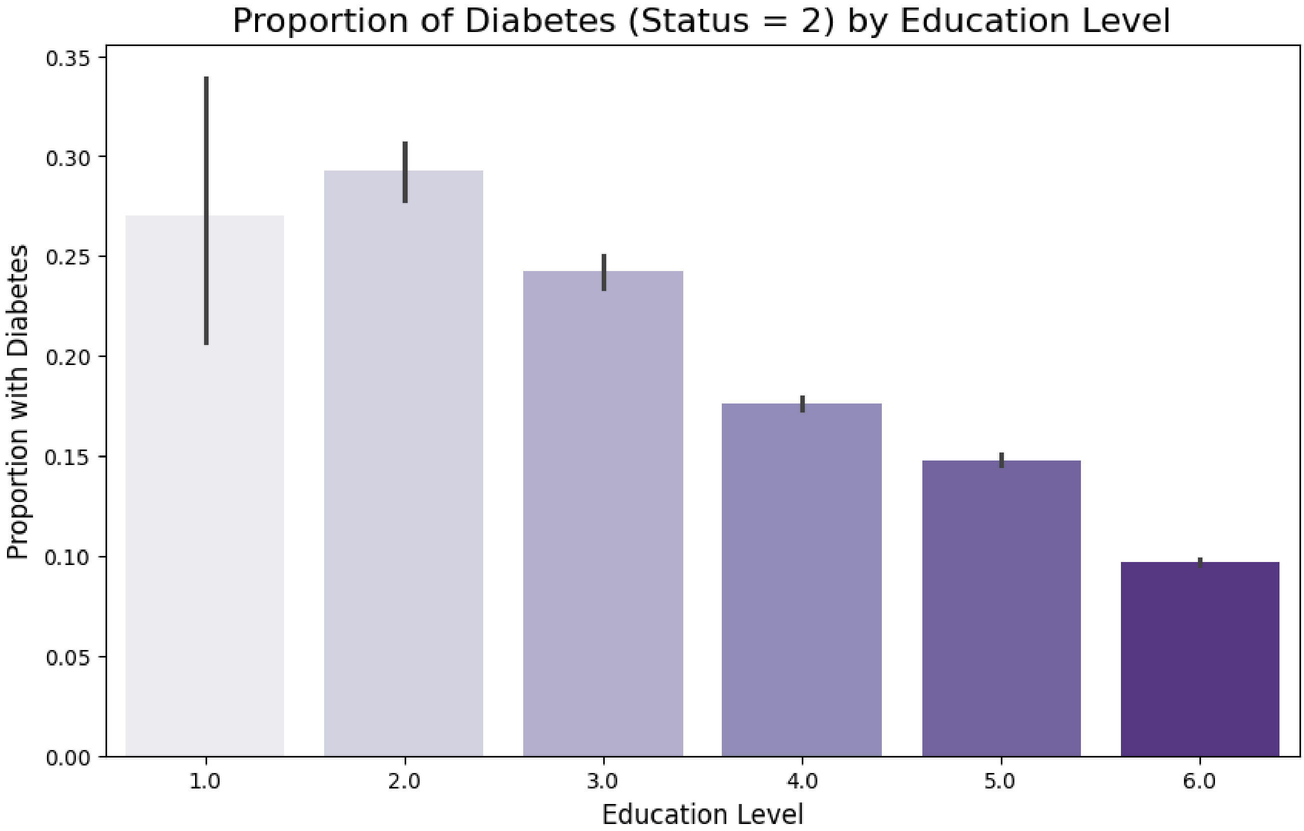



```
# 2. Diabetes Status by Education Level
plt.figure(figsize=(10, 6))
sns.barplot(x='Education', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palette='Pur
plt.title('Proportion of Diabetes (Status = 2) by Education Level', fontsize=16)
plt.xlabel('Education Level', fontsize=12)
plt.ylabel('Proportion with Diabetes', fontsize=12)
plt.show()
```

```
<ipython-input-38-ea88460d5906>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.barplot(x='Education', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palett
```

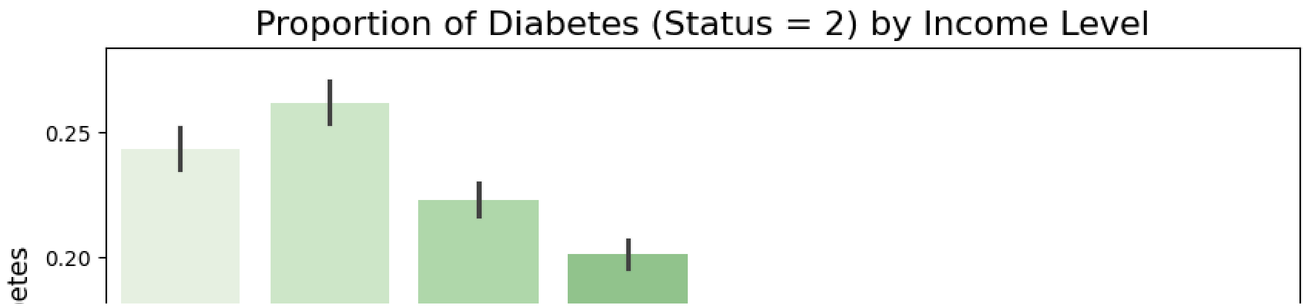



```
# 3. Diabetes Status by Income Level
plt.figure(figsize=(10, 6))
sns.barplot(x='Income', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palette='Greens
plt.title('Proportion of Diabetes (Status = 2) by Income Level', fontsize=16)
plt.xlabel('Income Level', fontsize=12)
plt.ylabel('Proportion with Diabetes', fontsize=12)
plt.show()
```

```
<ipython-input-39-4b1038bbbe2e>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.barplot(x='Income', y='Diabetes_012', data=df1, estimator=lambda x: sum(x == 2)/len(x), palette='
```




```
# 4. Diabetes and Multiple Health Issues (HighBP, HighChol, Stroke, HeartDiseaseorAttack)
df1['HealthIssuesCount'] = df1[['HighBP', 'HighChol', 'Stroke', 'HeartDiseaseorAttack']].sum(axis=1)

plt.figure(figsize=(10, 6))
sns.boxplot(x='Diabetes_012', y='HealthIssuesCount', data=df1, palette='coolwarm')
plt.title('Number of Health Issues by Diabetes Status', fontsize=16)
plt.xlabel('Diabetes Status', fontsize=12)
plt.ylabel('Number of Health Issues', fontsize=12)
plt.show()
```

```
<ipython-input-40-1b1f1054a3fb>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.boxplot(x='Diabetes_012', y='HealthIssuesCount', data=df1, palette='coolwarm')
```