PROJECT PHASE -2
A Predictive Assessment System for Diabetes Risk Factors

Mounika Pasupuleti
Engineering Science Data Science
University at Buffalo
Buffalo, New York
mpasupul@buffalo.edu

Sahithya Arveti Nagaraju
Engineering Science Data Science
University at Buffalo
Buffalo, New York
sarvetin@buffalo.edu

Sushmitha Manjunatha
Engineering Science Data Science
University at Buffalo
Buffalo, New York
smanjuna@buffalo.edu

***Abstract*—Diabetes has rapidly transformed from a relatively uncommon disease among a small number of people to a serious disease of global proportions affecting every age group. Diabetes Pathophysiology may be defined as a disease where the sugar levels in the blood are not well regulated leading to substantially deteriorated health and shortened life expectancy. In many cases treatment of diabetes comes with high costs which is an additional burden on the affected individual.**

The goal of this project is to investigate the most common correlation with diabetes, factors which help in understanding possible causative agents and their potential risk. This analysis is expected to aid in prevention of diabetes cases and promote healthy behavior.

In this regard, we expect to evaluate how these basic attributes such as work performance, nutrition, and health history could be used to estimate the probability of an individual being at risk of diabetes by employing complex machine learning models and analyzing data from health activities. This contribution is crucial as it addresses the growing health challenges like diabetes by providing valuable insights that enhances overall community health.

Index Terms: Diabetes, sugar levels, lifestyle, machine learning models.

I. DATA SOURCES

The dataset is taken from the Kaggle website. Below is the link:

https://www.kaggle.com/datasets/prosperchuks/health-dataset?select=diabetes_data.csv

The dataset consists of 70,707 rows and 19 columns in which there are 10 numeric features and 9 categorical features. Numeric features in our dataset are:

- Age
- HighChol
- CholCheck
- BMI
- HeartDiseaseorAttack
- PhysActivity
- HvyAlcoholConsump
- MentHlth
- PhyHlth
- HighBP

Categorical features in our dataset are:

- Sex
- Smoker
- Fruits
- Veggies
- GenHlth
- DiffWalk
- Stroke
- SugarConsumption
- Diabetes

Observe the data in the dataset:

```
[2] import pandas as pd
    import numpy as np
    data = pd.read_csv("impure_dataset_final.csv");
    data.head(5)
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	Male	0	1	26	Non Smoker	0	1.0	Does not Eat	
1	12	Male	1	1	26	Smoker	0	0.0	Eat	r
2	13	Male	0	1	26	Non Smoker	0	1.0	Eat	
3	11	Male	1	1	28	Smoker	0	1.0	Eat	
4	8	Female	0	1	29	Smoker	0	1.0	Eat	

```
[ ] data.shape

(70707, 19)
```

```
[ ] dataset_info = {
        "Number of rows in the dataset are": data.shape[0],
        "Number of columns in the dataset are ": data.shape[1],
        "Names of the columns": data.columns.tolist()
    }

    dataset_info

{'Number of rows in the dataset are': 70707,
 'Number of columns in the dataset are ': 19,
 'Names of the columns': ['Age',
  'Sex',
  'HighChol',
  'CholCheck',
  'BMI',
  'Smoker',
  'HeartDiseaseorAttack',
  'PhysActivity',
```

```
 'HvyAlcoholConsump',
 'GenHlth',
 'MentHlth',
 'PhysHlth',
 'DiffWalk',
 'Stroke',
 'HighBP',
 'SugarConsumption',
 'Diabetes']}
```

```
[ ] data.describe()
```

	Age	HighChol	CholCheck	HeartDiseaseorAttack	PhysActivity	HvyAlcol
count	70707.000000	70707.000000	70707.000000	70707.000000	70511.000000	69
mean	8.581116	0.525747	0.975264	0.147878	0.703124	
std	2.864696	0.499340	0.155320	0.354981	0.456885	
min	-31.000000	0.000000	0.000000	0.000000	0.000000	
25%	7.000000	0.000000	1.000000	0.000000	0.000000	
50%	9.000000	1.000000	1.000000	0.000000	1.000000	
75%	11.000000	1.000000	1.000000	0.000000	1.000000	
max	13.000000	1.000000	1.000000	1.000000	1.000000	

II. DATA CLEANING/PROCESSING

Uncleaned data contains duplicate entries or non-unique records, missing values and inconsistencies in the data. This data needs to be thoroughly cleaned to remove such issues and make the dataset fit for analysis.

A. Removing Duplicate Rows:

```
#1 Removing duplicate rows
print("Number of duplicated rows in the dataset are:", data.duplicated().sum())
data = data.drop_duplicates()
print("Number of rows after removing duplicated data are:", len(data))
```

```
Number of duplicated rows in the dataset are: 5554
Number of rows after removing duplicated data are: 65153
```

Using data.duplicated() method, we found 5,554 records having duplicated data. These duplicates were deleted and reduced the dataset from 70,707 rows to 65,153 rows.

B. Removing Missing Values:

At first, there were 6,702 missing values in the dataset concerning several different columns such as “BMI”, “Smoker”, “HvyAlcoholConsump”, “GenHlth” etc.,

Next, there were extreme cases of datasets deprecated due to the number of missing values (300 missing). After filtering the columns with 300 missing values are removed, which made dataset 64,286 rows. The rest of the categorical/numerical missing values that remain were replaced by mode/ mean respectively.

```
#2 Remove missing values
data.isnull().sum()
```

	0
Age	0
Sex	0
HighChol	0
CholCheck	0
BMI	987
Smoker	1013
HeartDiseaseorAttack	0
PhysActivity	196
Fruits	232
Veggies	215
HvyAlcoholConsump	932
GenHlth	989
MentHlth	947
PhysHlth	964
DiffWalk	227
Stroke	0
HighBP	0
SugarConsumption	0
Diabetes	0

dtype: int64

```
[79] total_missing_values = data.isnull().sum().sum()
     print("Total number of missing values in the dataset are:", total_missing_values)
```

```
Total number of missing values in the dataset are: 6702
```

```
[80] missingvalues_less_than_300 = data.isnull().sum() < 300
     columns_to_drop = data.columns[missingvalues_less_than_300]
     data.dropna(subset=columns_to_drop, inplace=True)
     print("Number of rows after removing columns with missing values less than 300 are:", len(data))
```

```
Number of rows after removing columns with missing values less than 300 are: 64286
```

C. Remove inconsistencies:

```
#3 Remove inconsistencies
for column in data.columns:
    unique_items = data[column].unique()
    print(f"Unique items in '{column}':", unique_items)
```

```
Unique items in 'Age': [  4  12  13  11   8   1   6   3  -1   7  10   9   5   2  -9 -31 -12 -21
  -3  -2  -7 -15 -19]
Unique items in 'Sex': ['Male' 'Female' 'M' 'F']
Unique items in 'HighChol': [0 1]
Unique items in 'CholCheck': [1 0]
Unique items in 'BMI': ['26' '28' '29' '18' '31' '32' '27' '24' '21' '58' '30' '20' '22' '38'
 '40' '25' '36' '47' '19' '37' '41' '23' '34' '35' '42' '17' '33' '44' nan
 '15' '52' '69' '56' '45' '39' '92' '98' '50' '46' '79' '48' '16' '63'
 '72' '54' '49' '68' '43' '84' '31.0 kg/m2' '53' '73' '76' '55' '51' '75'
 '57' '25.0 kg/m2' '60' '33.0 kg/m2' '77' '82' '67' '71' '61' '14' '81'
 '59' '29.0 kg/m2' '24.0 kg/m2' '86' '28.0 kg/m2' '13' '87' '65' '95' '89'
 '62' '64' '66' '85' '70' '83' '38.0 kg/m2' '34.0 kg/m2' '80' '21.0 kg/m2'
 '78' '26.0 kg/m2']
Unique items in 'Smoker': ['Non Smoker' 'Smoker' nan]
Unique items in 'HeartDiseaseorAttack': [0 1]
Unique items in 'PhysActivity': [1. 0.]
Unique items in 'Fruits': ['Does not Eat' 'Eat' 'No' 'Yes']
Unique items in 'Veggies': ['Eat' 'Does not Eat' 'No' 'Yes']
Unique items in 'HvyAlcoholConsump': [ 0.  1. nan]
Unique items in 'GenHlth': ['good' 'excellent' 'very good' 'fair' 'poor' nan]
```

```
Unique items in 'MentHlth': [ 5.  0.  7.  3.  4.  2. 30. 20.  1. 15. nan 25. 14. 28. 10.  6. 29. 26.
 12. 22. 13.  8.  9. 18. 16. 21. 17. 27. 24. 23. 11. 19.]
Unique items in 'PhysHlth': [30.  0. 10.  3.  6.  4. 15.  1.  2. 14.  7. 25. nan 21. 20.  5.  8. 22.
 23. 29. 12. 18. 28. 26. 24. 27. 11. 13. 16. 17.  9. 19.]
Unique items in 'DiffWalk': ['No' 'Yes' 'Y' 'N']
Unique items in 'Stroke': ['No' 'Yes' 'N' 'Y']
Unique items in 'HighBP': [1 0]
Unique items in 'SugarConsumption': ['High' 'Low']
Unique items in 'Diabetes': ['No' 'Yes']
```

```
[ ] data['Sex'] = data['Sex'].replace(['Male', 'Female'], ['M', 'F'])
    data['Fruits'] = data['Fruits'].replace(['Yes', 'No'], ['Eat', 'Does not Eat'])
    data['Veggies'] = data['Veggies'].replace(['Yes', 'No'], ['Eat', 'Does not Eat'])
    data['DiffWalk'] = data['DiffWalk'].replace(['Yes', 'No'], ['Y', 'N'])
    data['Stroke'] = data['Stroke'].replace(['Yes', 'No'], ['Y', 'N'])
    data['Diabetes'] = data['Diabetes'].replace(['Yes', 'No'], ['Y', 'N'])
    data['BMI'] = data['BMI'].astype(str).str.replace(' kg/m2', '').astype(float)
    for column in data.columns:
        unique_items = data[column].unique()
        print(f"Unique items in '{column}':", unique_items)
```

```
Unique items in 'Age': [  4  12  13  11   8   1   6   3  -1   7  10   9   5   2  -9 -31 -12 -21
  -3  -2  -7 -15 -19]
Unique items in 'Sex': ['M' 'F']
```

```
Unique items in 'HighChol': [0 1]
Unique items in 'CholCheck': [1 0]
Unique items in 'BMI': [26. 28. 29. 18. 31. 32. 27. 24. 21. 58. 30. 20. 22. 38. 40. 25. 36. 47.
 19. 37. 41. 23. 34. 35. 42. 17. 33. 44. nan 15. 52. 69. 56. 45. 39. 92.
 98. 50. 46. 79. 48. 16. 63. 72. 54. 49. 68. 43. 84. 53. 73. 76. 55. 51.
 75. 57. 60. 77. 82. 67. 71. 61. 14. 81. 59. 86. 13. 87. 65. 95. 89. 62.
 64. 66. 85. 70. 83. 80. 78.]
Unique items in 'Smoker': ['Non Smoker' 'Smoker' nan]
Unique items in 'HeartDiseaseorAttack': [0 1]
Unique items in 'PhysActivity': [1. 0.]
Unique items in 'Fruits': ['Does not Eat' 'Eat']
Unique items in 'Veggies': ['Eat' 'Does not Eat']
Unique items in 'HvyAlcoholConsump': [ 0.  1. nan]
Unique items in 'GenHlth': ['good' 'excellent' 'very good' 'fair' 'poor' nan]
Unique items in 'MentHlth': [ 5.  0.  7.  3.  4.  2. 30. 20.  1. 15. nan 25. 14. 28. 10.  6. 29. 26.
 12. 22. 13.  8.  9. 18. 16. 21. 17. 27. 24. 23. 11. 19.]
Unique items in 'PhysHlth': [30.  0. 10.  3.  6.  4. 15.  1.  2. 14.  7. 25. nan 21. 20.  5.  8. 22.
 23. 29. 12. 18. 28. 26. 24. 27. 11. 13. 16. 17.  9. 19.]
Unique items in 'DiffWalk': ['N' 'Y']
Unique items in 'Stroke': ['N' 'Y']
Unique items in 'HighBP': [1 0]
Unique items in 'SugarConsumption': ['High' 'Low']
Unique items in 'Diabetes': ['N' 'Y']
```

There are inconsistencies in data for instance ‘Sex’: ‘Male’, ‘M’, ‘Female’, ‘F’, ‘DiffWalk’, ‘Stroke’: ‘Yes’, ‘Y’, ‘No’, ‘N’ and in the ‘BMI’ column; being values measured in kg/m². This was also done for all inconsistencies to ensure they were uniform.

D. *Filling Null Values with Mean & Mode:*

```
#4 Filling null values  with mode and mean
for col in data.columns:
  if data[col].isnull().any():
    if data[col].dtype == 'object':
      data[col].fillna(data[col].mode()[0], inplace=True)
    else:
      data[col].fillna(data[col].mean(), inplace=True)
print("After imputing the null values in the dataset are:", data.isnull().sum().sum())
```

```
After imputing the null values in the dataset are: 0
```

After filling null values in the dataset. There should not be any missing values after replacing null values of categorical columns with the mode and numerical columns with the mean.

E. *Removing Negative Values:*

```
#5 Removing negative values in age column and before and after removing negative values
print("Number of negative values in the 'Age' column:", (data['Age'] < 0).sum())
data = data[data['Age'] >= 0]
print("Number of rows after removing negative values:", len(data))
```

```
Number of negative values in the 'Age' column: 11
Number of rows after removing negative values: 64275
```

There are a total of 11 negative values in ‘Age’ column which are needed to be removed. Therefore, after removal total no of remaining rows in the dataset are 64,275.

F. *Creating New Feature:*

```
# 6. Creating new attribute AgeGroup to classify as kids and teens
age_bins = [0, 10, 100]
age_labels = ['Kid', 'Teen']
data['Age_Group'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels)
print("Number of kids and teens in the dataset are:")
print(data['Age_Group'].value_counts())
data.head(5)
```

```
Number of kids and teens in the dataset are:
Age_Group
Kid     46796
Teen    17479
Name: count, dtype: int64
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	M	0	1	26.0	Non Smoker	0	1.0	Does not Eat	
1	12	M	1	1	26.0	Smoker	0	0.0	Eat	
2	13	M	0	1	26.0	Non Smoker	0	1.0	Eat	
3	11	M	1	1	28.0	Smoker	0	1.0	Eat	
4	8	F	0	1	29.0	Smoker	0	1.0	Eat	

Classifying individuals with respect to their age and creating a new column ‘Age_Group’ as follows:

- Kid: Age < 10
- Teen: Age ≥ 10

G. *Data Conversion:*

```
#7. Data type conversion for BMI column
data['BMI'] = data['BMI'].astype(int)
data['Age'] = data['Age'].astype(int)
data.head(5)
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Ve
0	4	M	0	1	26	Non Smoker	0	1.0	Does not Eat	
1	12	M	1	1	26	Smoker	0	0.0	Eat	
2	13	M	0	1	26	Non Smoker	0	1.0	Eat	
3	11	M	1	1	28	Smoker	0	1.0	Eat	
4	8	F	0	1	29	Smoker	0	1.0	Eat	

Converting the datatype of ’BMI’ and ’Age’ columns as integer so that it helps to make analysis easy by using mathematical techniques where numerical precision is required.

H. *BMI Group Classification:*

```
bmi_bins = [18, 25, 30, 40, 70]
bmi_labels = ['UnderWeight', 'HealthyWeight', 'OverWeight', 'Obese']
data['BMI_Group'] = pd.cut(data['BMI'], bins=bmi_bins, labels=bmi_labels)
print("Number of BMI groups in the dataset are:")
print(data['BMI_Group'].value_counts())
data.head(5)
```

```
Number of BMI groups in the dataset are:
BMI_Group
HealthyWeight    22293
OverWeight       20612
UnderWeight      15842
Obese             4764
Name: count, dtype: int64
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	F
0	4	M	0	1	26	Non Smoker	0	1.0	
1	12	M	1	1	26	Smoker	0	0.0	
2	13	M	0	1	26	Non Smoker	0	1.0	
3	11	M	1	1	28	Smoker	0	1.0	
4	8	F	0	1	29	Smoker	0	1.0	

Classifying individuals with respect to their BMI and creating a new feature named ‘BMI_Group’ by categorizing them with the labels ‘UnderWeight’, ’HealthyWeight’, ’OverWeight’ and ‘Obese’.

I. *Label Encoding:*

```
#9. Label Encoding
from sklearn.preprocessing import LabelEncoder
lbl_encod = LabelEncoder()

for column in data.columns:
  if data[column].dtype == 'object' or 'category':
    data[column] = lbl_encod.fit_transform(data[column])
data.head(5)
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	.
0	3	1	0	1	13	0	0	1	0	1	
1	11	1	1	1	13	1	0	0	1	0	
2	12	1	0	1	13	0	0	1	1	0	
3	10	1	1	1	15	1	0	1	1	1	
4	7	0	0	1	16	1	0	1	1	1	

5 rows × 21 columns

Label Encoding helps in transforming a categorical variable into a numerical format which helps in effective use of machine learning models.

J. Dropping Features:

```
#10. Dropping Features
print("Number of High and Low values in SugarComputation:")
print(data['SugarConsumption'].value_counts())
data = data.drop(columns=['SugarConsumption'])
data.head(5)
```

```
Number of High and Low values in SugarComputation:
SugarConsumption
0    64271
1        4
Name: count, dtype: int64
```

	Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	V
0	3	1	0	1	13	0	0	1	0	
1	11	1	1	1	13	1	0	0	1	
2	12	1	0	1	13	0	0	1	1	
3	10	1	1	1	15	1	0	1	1	
4	7	0	0	1	16	1	0	1	1	

By using feature extraction, we dropped the ’SugarConsumption’ feature as it contains only 4 entries marked as ‘High’ making it low variance.

K. Removing Outliers:

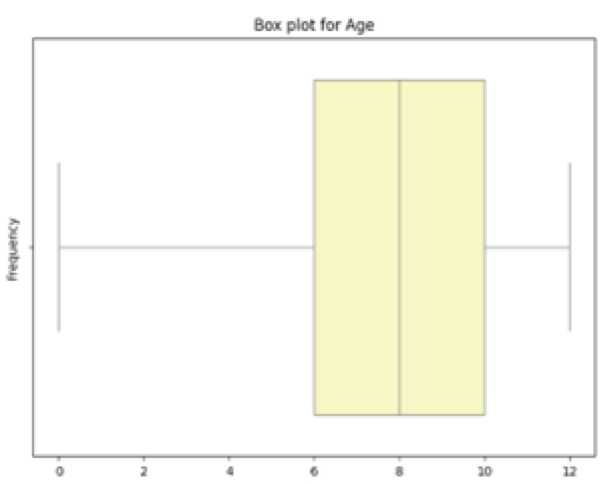
```
#11. Removing outliers
def remove_outliers(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    print(f"Lower bound for '{column}': {lower_bound}")
    print(f"Upper bound for '{column}': {upper_bound}")
    return data[(data[column] >= lower_bound) & (data[column] <= upper_bound)]

columns_to_clean = ['BMI', 'Age']
for col in columns_to_clean:
    data = remove_outliers(data, col)

print("Number of rows after removing outliers:", len(data))
```

```
Lower bound for 'BMI': 0.0
Upper bound for 'BMI': 32.0
Lower bound for 'Age': 0.0
Upper bound for 'Age': 16.0
Number of rows after removing outliers: 62168
```

By using IQR method, we can find the outliers using upper and lower bounds. After removing these outliers, we finally have 62,168 rows in our dataset.

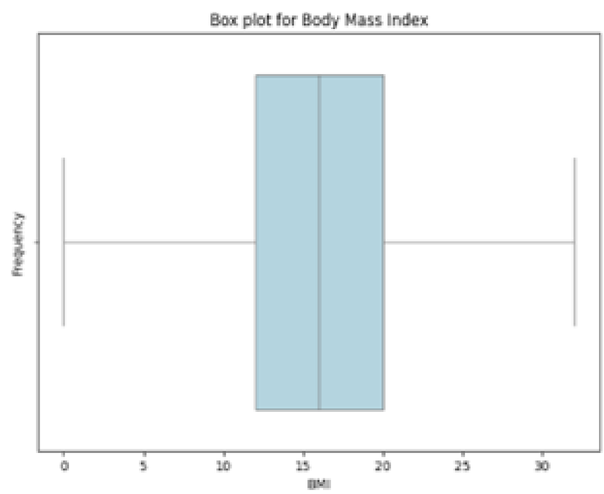
III. EXPLORATORY DATA ANALYSIS

A. Box Plots:



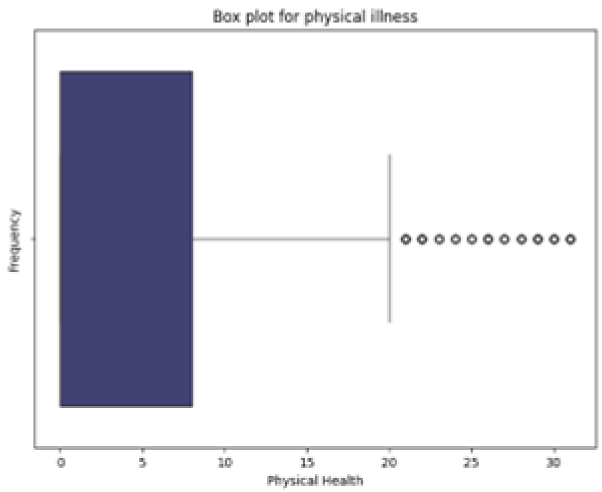
Observations:

- The median of the age distribution is 8, implying that the average population is relatively homogenous.
- There is no possible reason for concerns toward the existence of outliers, as the whiskers reach irritably both the lower and upper advents without any glaring irregularities.
- The central IQR was found to range between 6 and 10 with the latter edge extending the middle range of 50 percent of the data.
- The length of the whiskers on either side is about the same, thus implying that data is normally distributed with no considerable skewness.



Observations:

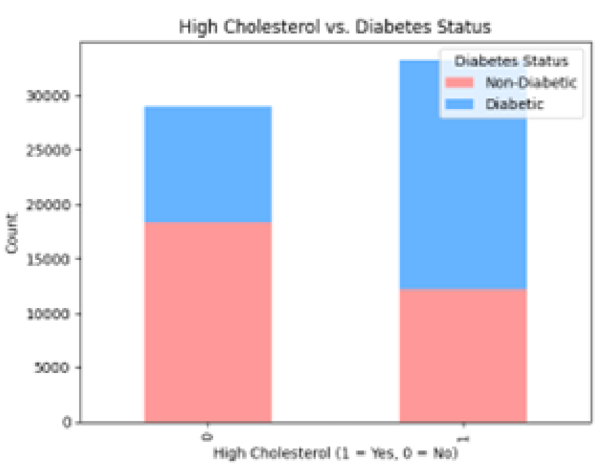
- The median BMI value seems to be in the middle of the lower range on the BM index with a value of about 17.
- The concept of outliers is not in this data as evidenced by the absence of any points that are beyond the whiskers.
- The IQR is roughly between 15 and 20; thus middle 50values.
- ‘Whiskers’ extends in equal lengths on the diagrams both to the right and left sides, which suggests that there are no significant distortions in the distribution of the BMI values.

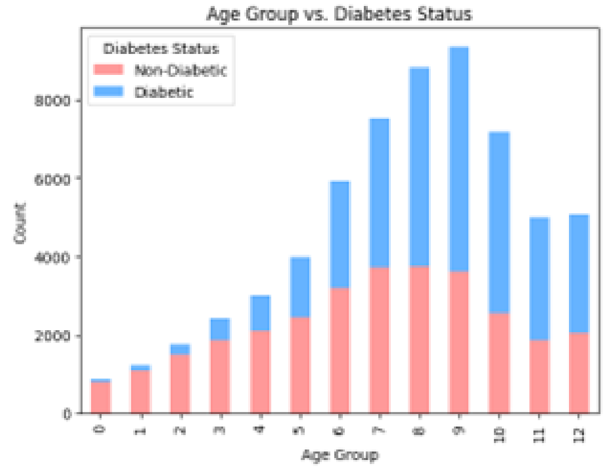


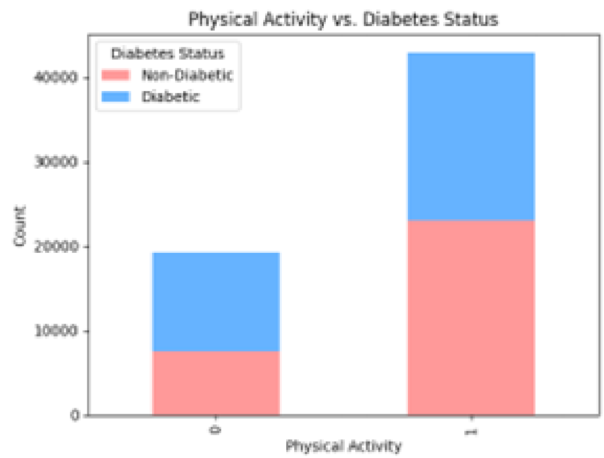
Observations:

- In the second box plot, the distribution of physical health appears to be negative as many people report low value mostly in keeping physical wellbeing more so within (0 to 5 days).
- For example, the median is about 2-3 days, suggesting that people do not tend to report a lot of physical health problems.
- Some maximum values which are more than 15 days there are many outliers outside the whiskers diagrams this illustrates that some people were ill or injured for more days than others.
- In contrast, the range shown by the spread of the whiskers was even wider indicating that respondents reported experiencing different levels of physical health.

B. Bar Plots:

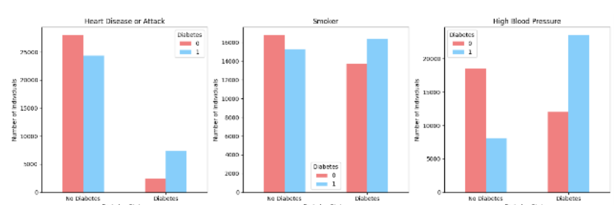






Observations:

- From the first graph above, it can be inferred that people with a high daily cholesterol intake have a higher chance of diabetes than those without a high daily cholesterol intake.
- The second graph very clearly shows that as a person grows older, the chances of having diabetes increase with the highest numbers lying in the middle age categories.
- The third graph conclusively shows that those who participate in various forms of physical activities are at a lesser risk of acquiring diabetes as opposed to those who are inactive.

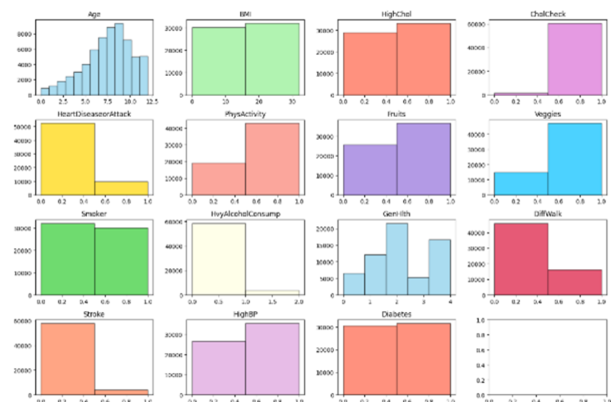


Observations:

- From the first graph, it seems that people suffering from diabetes have the higher chances of having either heart disease or one in the form of heart attack, however, it is the total number who do not suffer from diabetes who have a higher number of cases.

- In the other graph, it is shown that the propensity to smoke is almost the same among people with diabetes and without diabetes, therefore, it cannot be concluded that smokers with diabetes are more than nonsmokers.
- The third graph says that high blood pressure, or hypertension, seems to strike people with diabetes more than those without, thus linking diabetes and hypertensions heavy-handedly.

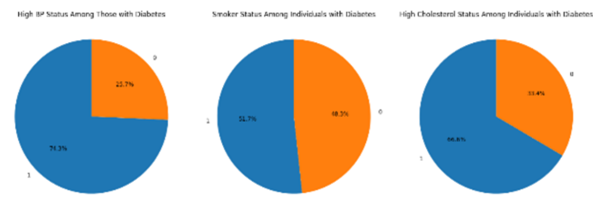
C. Histograms:



Observations from the Multiple Histogram Plots:

- Considering the sample demographics, there are more middle-aged respondents with the distribution pattern observed to be concentrated in the mid-age groups, a range which is probably between 6 and 9 units. This suggests age distribution is skewed.
- BMI parameters are equally distributed, showing good representation of respondents in both categories.
- Many of the respondents have reported having high cholesterol which indicates a greater number of persons suffering from the condition that the sample represent.
- Since most of the respondents have undergone a cholesterol check, it follows that cholesterol screening is a common practice among the respondents.
- Most of the respondents have never had a case of heart disease or heart attack. This reveals a good cardiovascular health status amongst the sample.
- A larger number of my respondents go out to exercise, this shows increased levels of physical activity among the respondents.
- While a slightly larger portion of the respondents consumes fruits on regular basis, a considerable portion does not, therefore these have varied dietary practices.
- Regular vegetable consumption is to some extent higher as more respondents report to regular meal with vegetables.
- Smokers are slightly fewer than non-smokers indicating that the smoking status is almost evenly spread within the population sample.
- It is uncommon to have heavy drinkers as most respondents indicate only moderate or lower drinking levels.
- Health ratings are corroborated by majority of the people with moderate health rations accordingly and is perceived on average level.
- While a small subgroup does walk, difficulty in mobilization is not prevalent.

D. Pie Charts:



- High Blood Pressure (BP) Status Among Those with Diabetes:
  - 74.3high blood pressure.
  - 25.7not have high blood pressure.

Observation:

A large proportion of people with diabetes (nearly three-quarters) also suffer from high blood pressure, which suggests a strong link between diabetes and high BP.

- Smoker Status Among Individuals with Diabetes:
  - 51.7smokers.
  - 48.3non-smokers.

Observation:

Slightly more than half of the individuals with diabetes are smokers, which is concerning as smoking can exacerbate diabetes-related complications. There is an almost equal split between smokers and non-smokers in this population.

- 3. High Cholesterol Status Among Individuals with Diabetes:
  - 66.6high cholesterol.
  - 33.4not have high cholesterol.

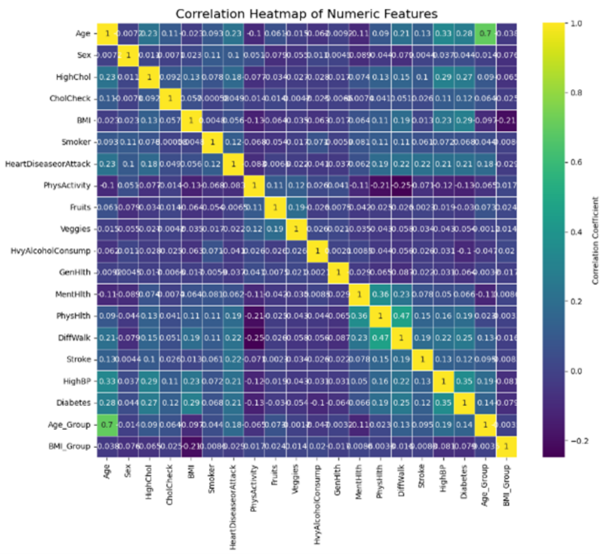
Observation:

A significant majority (two-thirds) of individuals with diabetes also have high cholesterol. This indicates a high comorbidity of cholesterol problems in diabetic individuals, increasing their risk for cardiovascular complications.

General Observations:

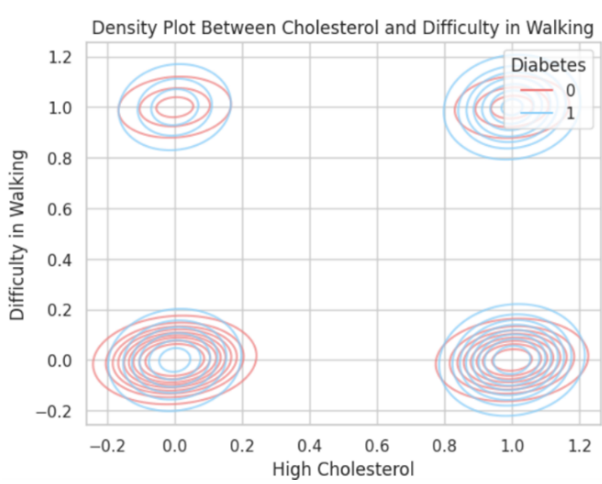
- Comorbidities: The high percentages of people with diabetes suffering from high blood pressure (74.3%) and high cholesterol (66.6%) indicate that these comorbidities are common in diabetic populations.
- Smoking Risk: More than half (51.7%) of diabetic individuals are smokers, which could further increase the risk of cardiovascular diseases and complications associated with diabetes.

E. Heatmap:



- Age correlates with high blood pressure, high cholesterol, and diabetes, indicating age-related increases in these conditions.
- BMI is linked to high cholesterol and high blood pressure, showing that higher body weight contributes to both.
- General health and physical health are closely related, with poor general health strongly tied to difficulty walking and poor physical health.
- High BP, BMI, high cholesterol, and age are the strongest predictors of diabetes in the dataset.
- Physical activity is positively associated with better general health and physical health, but negatively related to difficulty walking.
- Smoking negatively correlates with fruit and vegetable consumption, indicating poorer diet choices among smokers.

F. Density plot between Cholesterol and Difficulty in Walking:



Observations:

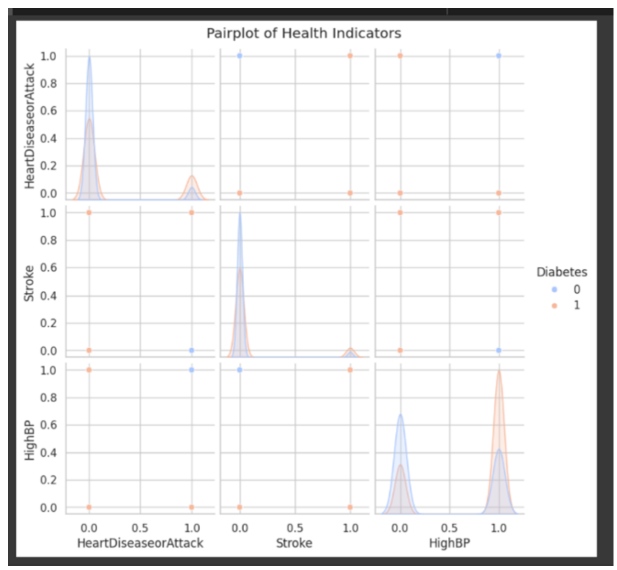
The plot clearly delineates two different density contour clusters and presents a clear relation of high cholesterol with the difficulty in walking. Those having had greater cholesterol levels have been seen to have difficulties in walking.

Impact of Diabetes:

- The plot suggests that diabetes patients are more likely to be hypercholesterolemic and have difficulty in walking and the excursion is extended further.
- Diabetes is probably not the sole cause of difficulty in walking since there is a clear overlap of the two groups.
- For cholesterol levels, it is observed that most of the scores are not clustered but rather a majority of them fall at the peak around 0.8.
- For difficulty in walking as well, only a few are apportioning around the concentration of 0.2.

This indicates that there is an indelible link between high cholesterol levels, diabetes and the inability to walk.

G. Pair plot of Health indicators:

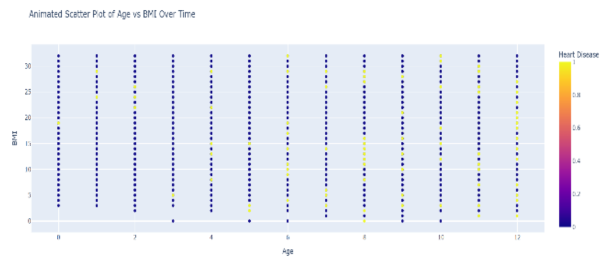


Observations:

- Positive Correlations: There is a positive correlation in all three health conditions, and each one will increase the risk for the others.
- Strongest Association: Heart disease/attack and high blood pressure have the most closely related association.
- Skewed Distributions: The distributions of heart disease/attack and stroke are left-skewed, while the high blood pressure values are dispersed.

High blood pressure acts as a common risk factor between heart disease/attack and stroke, while the prevalence of both heart disease/attack and stroke is comparably low to that of high blood pressure.

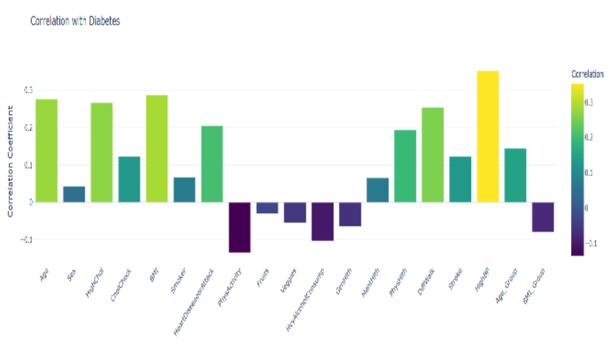
H. Scatter Plot of Age vs BMI Over Time:



Observations:

- Age Range: The dataset encompasses a wide age bracket, ranging from early childhood to teenage. This infers a heterogeneous child population sample.
- Distribution of BMI: The graph peaks in the middle range between 15-25. That is indicative of at least a sizeable percentage of the population falling within a healthy range.
- There is a general trend of rising BMI with age, more so in later years. It shows that weight gain could be related to the factors of lifestyle, such as reduced physical activity and change in diet, through the years.
- Higher values of BMI are associated with increased risks of heart disease, again wellknown, which gives emphasis to the need for keeping weight at as healthy a level as possible.
- Animation of the plot could show possible changes in the relationship of age to BMI with heart disease over time.

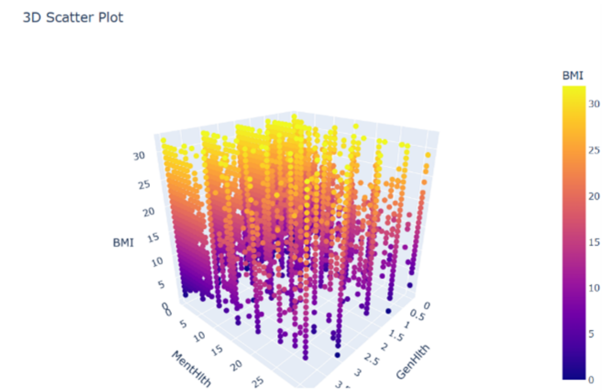
I. Correlation Bar Plot:



- Strong Positive Correlation: The important risk factors that are associated with diabetes include age, high cholesterol, BMI, difficulty walking, and high blood pressure. This has brought the need to attend to these factors through lifestyle changes, medical interventions, and preventive measures.
- Positive Correlation-Moderate: At this level, are considered the following as moderate risks: sex, cholesterol checks, smoking, heart disease/attack, mental health, physical health, and stroke. While these do not relate as highly with diabetes as the above-named key risk factors, they do add to the overall risk.
- Negative Correlation: Physical activity, fruits, vegetables, heavy alcohol consumption, general health, and the group of BMI are associated with lower risk of diabetes. To prevent diabetes, healthy lifestyle habits can be promoted.

J. 3D Scatter Plot:

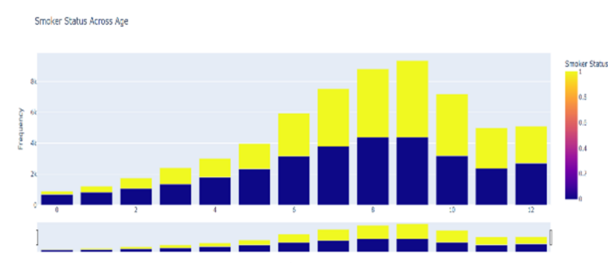
K. Correlation Bar Plot:



This plot shows a complex interrelationship among these three variables.

- There is a slight negative correlation between BMI and mental health. It could be interpreted that for those people having higher values of BMI, their scores of mental health may be lower, though the relation is not exactly strong.
- The correlation for BMI in relation to general health is less clear. There seems to be a slight negative correlation, but it is less than the relationship of BMI with mental health.

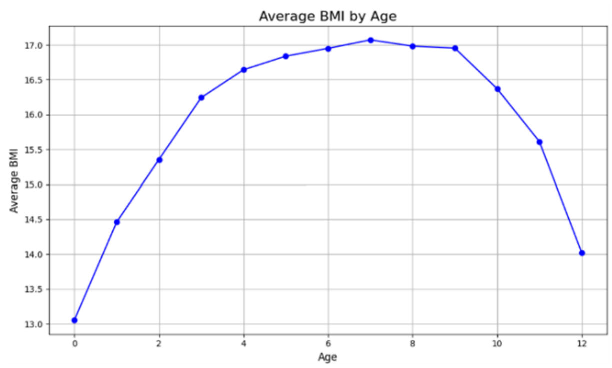
L. Stacked bar graph for Smoker Status Across Age Graph:



The height of each bar will show the number of people in that age group who are smokers and who are not smokers.

- Increasing Smoking Prevalence with Age: The graph above clearly indicates a very strong positive correlation between age and smoking prevalence. As age increases, so does the probability of being a smoker.
- Peak Smoking Rates: It reaches the peak in the mid-to-late adult years, around 8-10 years. This may indicate that smoking initiation or continuation is more vulnerable in these age groups.

M. Line Graph of BMI vs Age:



Observations:

- The graph shows a clear U-shaped curve, indicating that average BMI increases from early childhood to middle age, reaches a peak, and then declines in later adulthood.
- The highest average BMI is observed in midadulthood (around age 7-8). This suggests that individuals in this age group tend to have the highest body mass index.
- In early childhood (ages 0-2), average BMI is relatively low. During adolescence (ages 3-6), BMI increases steadily, reflecting the typical growth spurt.
- In late adulthood (ages 10-12), average BMI starts to decline. This could be attributed to various factors, such as decreased physical activity, changes in metabolism, or health conditions.

IV. ALGORITHMS

In this project, we applied six significant algorithms to predict diabetes risk. The selected algorithms include both commonly discussed methods and those not covered in class.

Algorithms Applied:

- K-Nearest Neighbors (KNN)
- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- Random Forest
- Extreme Gradient Boosting (XGBoost)
- Decision Tree

A. K-Nearest Neighbors (KNN)

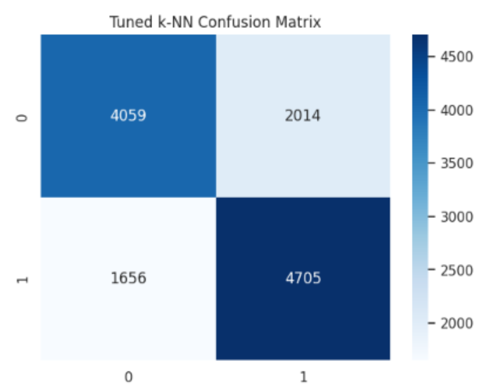
There are several reasons to select the KNN algorithm for our diabetes prediction model. One key advantage is that it captures complex, non-linear relationships in the dataset. Diabetes prediction relies on comparing multiple parameters, including physical activity, smoking status, BMI, age, and more.

Additionally, KNN is robust against outliers, a common issue in healthcare datasets due to variations in individual health. This resilience ensures that the model's predictions remain reliable despite the presence of atypical data points. Furthermore, the interpretability of KNN is a significant benefit in healthcare contexts, as it provides transparency in decision-making by determining outcomes based on the majority class of the nearest neighbors.

- Tuning: In our investigation, Initially we got accuracy 0.68 and then we did two key hyperparameters that influence the results obtained from K-Nearest Neighbors were targeted: the number of neighbors-n_neighbors-and a metric of distance. This is resolved through systematic testing for n_neighbors, namely, 3, 5, 7, and 9, whereas distance metric evaluation considered Euclidean, Manhattan, and Minkowski. A thorough cross-validation using GridSearchCV can find the best configuration. This was particularly when applying the 5 neighbors with the Minkowski distance metric. This therefore points to the importance of adjustment in hyperparameters in order to improve performance in KNN algorithms for our classification problem in diabetes.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 70.5%.

KNN Accuracy & Report:

```
/usr/local/lib/python3.10/dist-packages/numpy/ma/core.py:2820: RuntimeWarning:

invalid value encountered in cast

Best KNN Parameters: {'metric': 'manhattan', 'n_neighbors': 9}
Tuned KNN Accuracy: 0.7048415634550427
Training Report
              precision    recall  f1-score   support

           0       0.77      0.72      0.74     24420
           1       0.75      0.80      0.77     25314

    accuracy                           0.76     49734
   macro avg       0.76      0.76      0.76     49734
weighted avg       0.76      0.76      0.76     49734

Testing Report
              precision    recall  f1-score   support

           0       0.71      0.67      0.69      6073
           1       0.70      0.74      0.72      6361

    accuracy                           0.70     12434
   macro avg       0.71      0.70      0.70     12434
weighted avg       0.71      0.70      0.70     12434
```



B. Naive Bayes

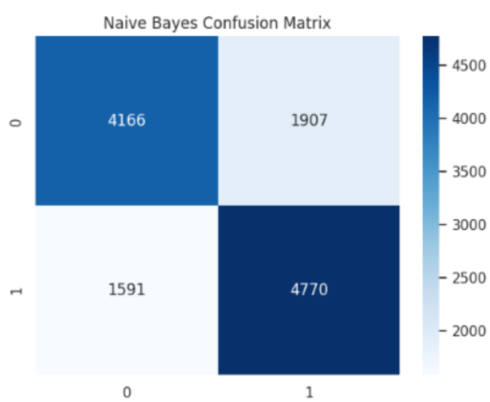
Using the Naive Bayes algorithm for diabetes prediction presents several benefits that align well with the characteristics of the dataset. One of its primary advantages is its ability to handle categorical data effectively, making it particularly appropriate for features like gender, smoking status, history of heart disease, and stroke, which are crucial risk factors in diabetes-related datasets. Given that diabetes can lead to severe health complications if not identified early, accurately predicting risk is essential. Naive Bayes assumes that the features are independent, simplifying the computation and modeling process, especially when dealing with the categorical aspects of certain variables.

The probabilistic nature of the model allows for clear insights into how predictions are made, which is vital for users to understand the model's logic. This transparency is especially important in the context of diabetes prediction, where trusting the model's conclusions can significantly impact healthcare decisions, potentially leading to timely interventions. Additionally, Naive Bayes boasts a quick training time, making it an excellent option for projects that are resource-limited or require prompt results, such as those involving diabetes datasets.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 71.9%.

Naive Bayes Accuracy & Report:

```
Naive Bayes Accuracy: 0.7186746018980216
Training Report
              precision    recall  f1-score   support

           0       0.72      0.68      0.70     24420
           1       0.71      0.75      0.73     25314

    accuracy                           0.71     49734
   macro avg       0.71      0.71      0.71     49734
weighted avg       0.71      0.71      0.71     49734

Testing Report
              precision    recall  f1-score   support

           0       0.72      0.69      0.70      6073
           1       0.71      0.75      0.73      6361

    accuracy                           0.72     12434
   macro avg       0.72      0.72      0.72     12434
weighted avg       0.72      0.72      0.72     12434
```



C. Logistic Regression

Logistic regression is a widely used supervised machine learning technique designed to estimate the likelihood of binary outcomes, such as whether a patient is diagnosed with diabetes.

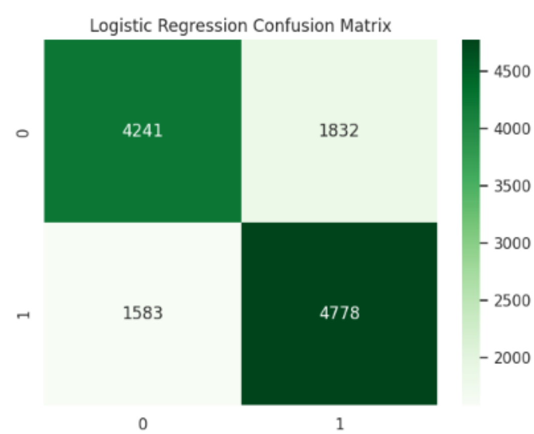
The main strength of logistic regression is its straightforwardness. The model provides clear insights into how different variables contribute to predicted outcomes, making it easier for users to understand the relationship between predictors and diabetes probability Coefficients from the logistic regression equation represent the odds associated with each predictor.

Furthermore, logistic regression is known for its complexity. In general, it performs better when applied to new data, as it is less likely to overfit the training data. This characteristic is particularly important in real-world applications, where the training data may not fully reflect the state of the data used to predict. By balancing accuracy, logistic regression is a reliable tool in the healthcare context for diabetes prediction.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe accuracy is 72.5%.

Logistic Regression Accuracy & Report:

```
Logistic Regression Accuracy: 0.72534984719318
Training Report
              precision    recall  f1-score   support

           0       0.73      0.69      0.71     24420
           1       0.72      0.75      0.73     25314

    accuracy                           0.72     49734
   macro avg       0.72      0.72      0.72     49734
weighted avg       0.72      0.72      0.72     49734

Testing Report
              precision    recall  f1-score   support

           0       0.73      0.70      0.71      6073
           1       0.72      0.75      0.74      6361

    accuracy                           0.73     12434
   macro avg       0.73      0.72      0.72     12434
weighted avg       0.73      0.73      0.73     12434
```



D. Support Vector Machines (SVM)

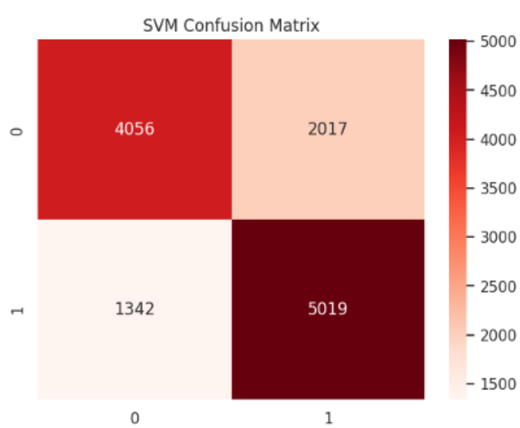
One of the robust supervised learning techniques that form quite ideal choices for the classification of diabetes status, a factor very critical in managing a chronic condition affecting millions across the globe, is Support Vector Machines. Among the key merits of SVM is handling high-dimensional data with great effectiveness, which comes in pretty useful considering the scope of risk factors associated with diabetes-anything from obesity to age to family and lifestyle choices. By using kernel functions, SVM is able to model complex nonlinear relationships between features that can reveal patterns in the dataset for indicating a person's susceptibility to diabetes.

Besides this, SVM is resistant to overfitting when the number of features outnumbers the samples, which also is very common in medical datasets. This property ensures that the model generalizes well for unseen data, crucial for proper prediction in real-world applications. The clear margin of separation obtained while using SVM enhances interpretability by allowing health professionals to more easily understand how the model differentiates between diabetic versus nondiabetic patients. Overall, the support vector machine will make correct predictions that would enable the doctor or physician to make prudent decisions in terms of assessment and management of the risk of diabetes to improve patient outcomes.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 73%.

SVM Accuracy & Report:

```
SVM Accuracy: 0.7298536271513592
Training Report
              precision    recall  f1-score   support

           0       0.75      0.66      0.70     24420
           1       0.71      0.79      0.74     25314

    accuracy                           0.72     49734
   macro avg       0.73      0.72      0.72     49734
weighted avg       0.73      0.72      0.72     49734

Testing Report
              precision    recall  f1-score   support

           0       0.75      0.67      0.71      6073
           1       0.71      0.79      0.75      6361

    accuracy                           0.73     12434
   macro avg       0.73      0.73      0.73     12434
weighted avg       0.73      0.73      0.73     12434
```



E. Random Forest

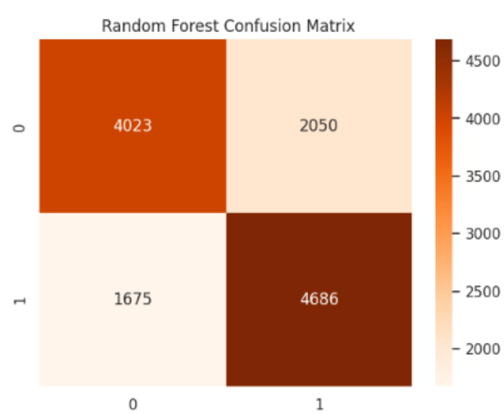
Random Forest is particularly effective at modeling the complex relationships inherent in datasets, making it essential for accurately assessing the diverse risk factors associated with diabetes. By creating multiple decision trees and aggregating their predictions through ensemble learning, Random Forest enhances the model's resilience, effectively managing the variability and noise that often characterize healthcare data.

Another significant advantage of Random Forest is its ability to evaluate feature importance, which is crucial in the context of diabetes prediction. This capability provides valuable insights into the significance of various factors, such as BMI, blood pressure, and physical activity, allowing for a clearer understanding of how each parameter contributes to diabetes risk. In healthcare settings, where transparency is vital, this analysis of feature importance fosters trust and comprehension among healthcare providers and stakeholders.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 70.0%.

Random Forest Accuracy & Report:

```
Random Forest Accuracy: 0.7004182081389738
Training Report
              precision    recall  f1-score   support

           0       0.97      0.98      0.98     24420
           1       0.98      0.97      0.98     25314

    accuracy                           0.98     49734
   macro avg       0.98      0.98      0.98     49734
weighted avg       0.98      0.98      0.98     49734

Testing Report
              precision    recall  f1-score   support

           0       0.71      0.66      0.68      6073
           1       0.70      0.74      0.72      6361

    accuracy                           0.70     12434
   macro avg       0.70      0.70      0.70     12434
weighted avg       0.70      0.70      0.70     12434
```



F. XGBoost

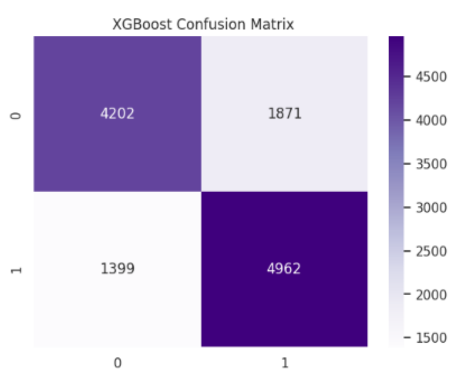
XGBoost yields highly accurate predictions of diabetes, a chronic disease characterized by high levels of blood glucose, together with serious subsequent health complications. This algorithm works miracles on complex data and can also reflect nonlinear relationships between various risk factors such as age, BMI, and physical activity, critical in diabetic risk assessment. Utilizing gradient boosting, XGBoost ensembles a large number of weak learners to generate a strong predictive model while integrating techniques for regularization that avoid overfitting.

Also, XGBoost handles missing values and feature importance scores, thus enabling healthcare practitioners to identify important risk factors. Such a scalable and fast algorithm is suitable for large datasets and helps to make timely predictions in the clinical environment. Customizable hyperparameters allow XGBoost to be fine-tuned for the peculiarities in diabetes prediction, therefore turning even more effective for healthcare applications.

- Tuning: Since we achieved an accuracy of more than 70%, we decided not to proceed with tuning the hyperparameter further. This suggests that while hyperparameter tuning can often enhance model performance, our model's performance in classifying diabetes in our data was already satisfactory.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 73.7%.

XGBoost Accuracy & Report:

```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:158: UserWarning:

[20:56:59] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.


 XGBoost Accuracy: 0.7370114202991797
Training Report
              precision    recall  f1-score   support

           0       0.80      0.73      0.76     24420
           1       0.76      0.82      0.79     25314

    accuracy                           0.78     49734
   macro avg       0.78      0.78      0.78     49734
weighted avg       0.78      0.78      0.78     49734

Testing Report
              precision    recall  f1-score   support

           0       0.75      0.69      0.72      6073
           1       0.73      0.78      0.75      6361

    accuracy                           0.74     12434
   macro avg       0.74      0.74      0.74     12434
weighted avg       0.74      0.74      0.74     12434
```



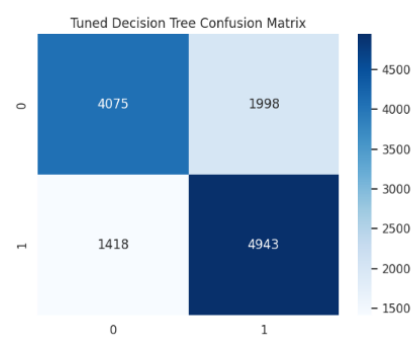
G. Decision Tree

Decision Trees are effective for predicting diabetes, capturing complex relationships between risk factors like age, BMI, and physical activity. Their intuitive structure allows for clear, interpretable outcomes, making them valuable in clinical settings. The algorithm can handle missing values and provides insights into feature importance, helping identify key diabetes risk factors. While they may overfit, techniques like pruning and ensemble methods can enhance their accuracy, making Decision Trees adaptable for various healthcare applications.

- Tuning: In our investigation, we initially achieved an accuracy of 0.62 with the Decision Tree model. We focused on three key hyperparameters that significantly impact the model's performance: max_depth, min_samples_split, and min_samples_leaf. We systematically evaluated various configurations for these parameters, testing max_depth values of None, 5, 10, and 15, as well as min_samples_split options of 2, 5, and 10, and min_samples_leaf settings of 1, 2, and 4.
  Using GridSearchCV for comprehensive cross-validation, we identified the optimal configuration that improved the model's accuracy to 72.6. This process highlights the importance of adjusting hyperparameters to enhance model performance in our diabetes classification problem.
- Effectiveness: From the cross-validation training and testing and classification report, we can observe the accuracy is 72.5%.

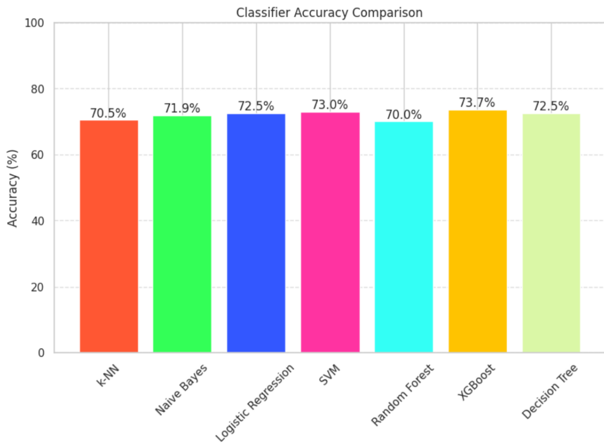
Decision Tree Accuracy & Report:

```
Best Decision Tree Parameters: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2}
Tuned Decision Tree Accuracy: 0.7252694225510696
Training Report
              precision    recall  f1-score   support

           0       0.77      0.69      0.73     24420
           1       0.73      0.80      0.76     25314

    accuracy                           0.75     49734
   macro avg       0.75      0.75      0.75     49734
weighted avg       0.75      0.75      0.75     49734

Testing Report
              precision    recall  f1-score   support

           0       0.74      0.67      0.70      6073
           1       0.71      0.78      0.74      6361

    accuracy                           0.73     12434
   macro avg       0.73      0.72      0.72     12434
weighted avg       0.73      0.73      0.72     12434
```



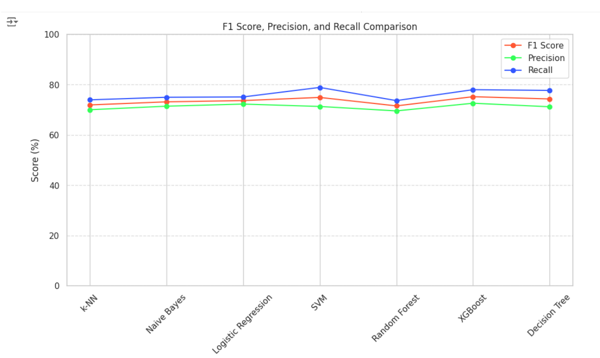
V. EVALUATION

The bar graph compares the accuracy of different classification algorithms (K-NN, Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and Decision Tree) on a diabetes prediction problem.



- XGBoost and Decision Tree achieved the highest accuracy of 73.7% and 72.5%, respectively, indicating that these algorithms might be the most suitable for this particular diabetes prediction task.
- K-NN and Naive Bayes didn't perform well, with accuracies of 70.5% and 71.9%, respectively.
- Logistic Regression, SVM, and Random Forest had accuracies in the mid-range, with values between 72.5% and 73.0%.

The below line graph compares the F1 score, precision, and recall of different classification algorithms (K-NN, Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and Decision Tree) on a diabetes prediction problem.



- XGBoost and Decision tree generally exhibits the best performance across all three metrics, achieving the highest F1 score, precision, and recall. This suggests that XGBoost is a well-rounded algorithm for this particular diabetes prediction task.
- Logistic Regression and SVM achieve decent F1 scores but have lower precision and recall compared to XGBoost and Decision Tree.

Overall, the graph suggests that XGBoost and Decision Tree are promising algorithms for diabetes prediction.

REFERENCES

[1] https://scikit-learn.org/stable/modules/tree.html
[2] https://www.datacamp.com/tutorial/xgboost-in-python
[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassif

Peer Evaluation Form for Final Group Work

CSE 487/587B

- Please write the names of your group members.

Group member 1 : Sahithya Arveti Nagaraju

Group member 2 : Sushmitha Manjunatha

Group member 3 : Mounika Pasupuleti

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	Group member 1	Group member 2	Group member 3
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude.	5	5	5
Contributes significantly to the success of the project .	5	5	5
TOTAL	20	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%) :

Group member 1 :

33.333

Group member 2 :

33.33

Group member 3 :

33.33