Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić

CodeOp DSPT10 - July 2023

codeop

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

Index

Index

Index

Index

Index

Index

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

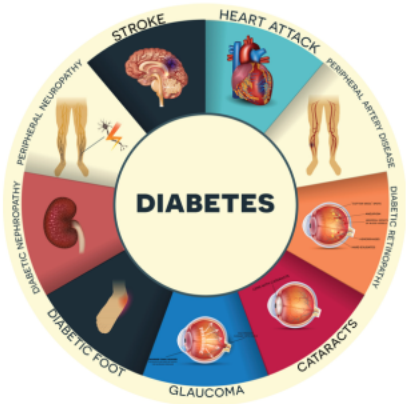
Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

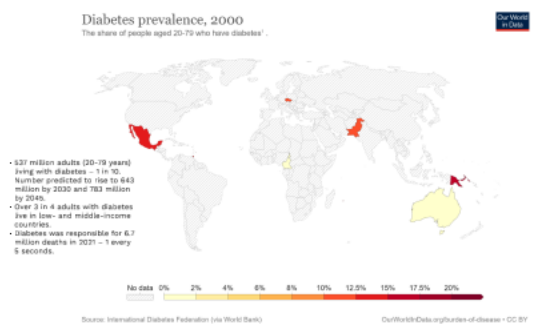
1- INTRODUCTION

What is Diabetes?



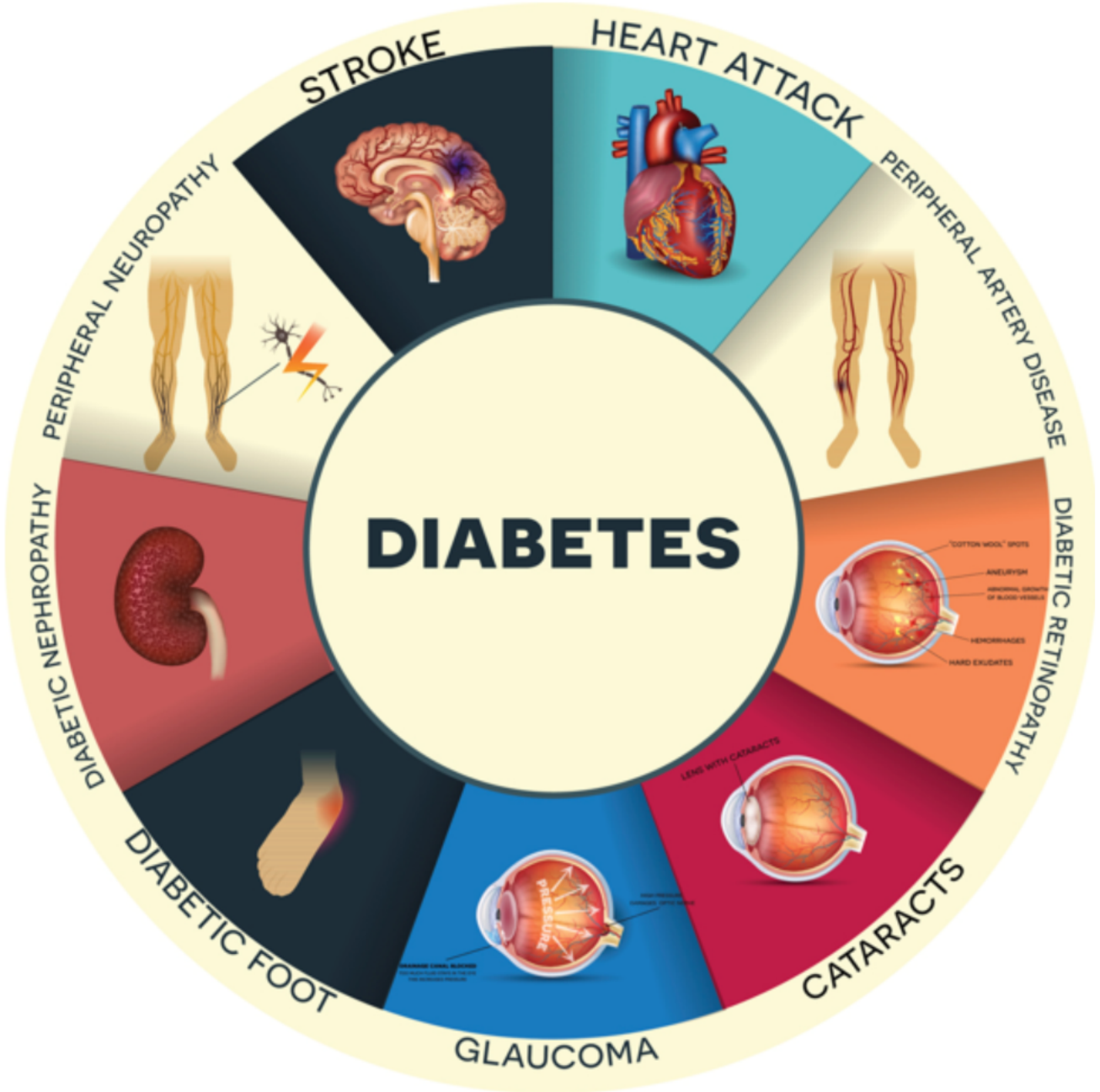
What can be done?

Why Study Diabetes



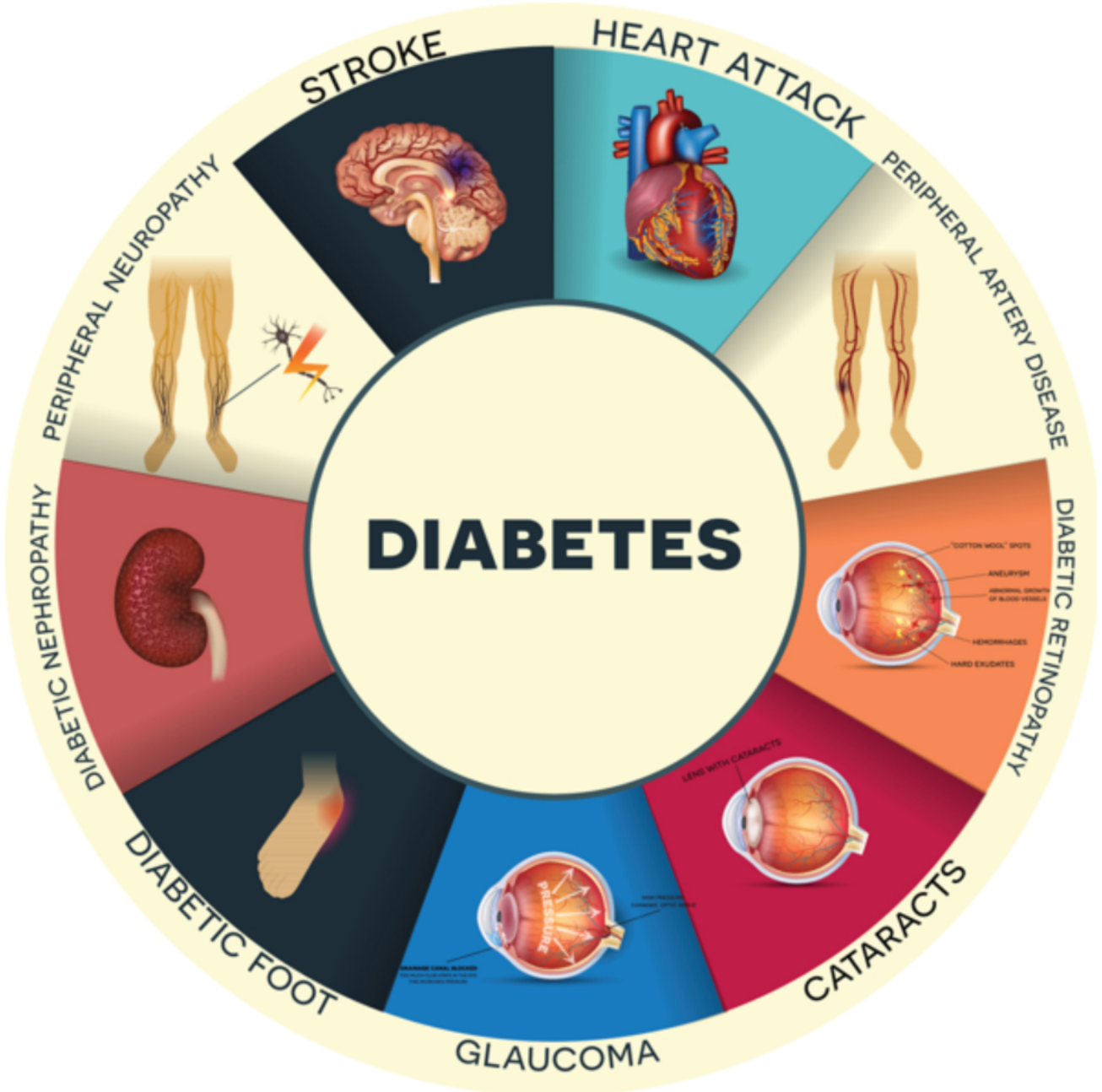
Study Objective

What is Diabetes?



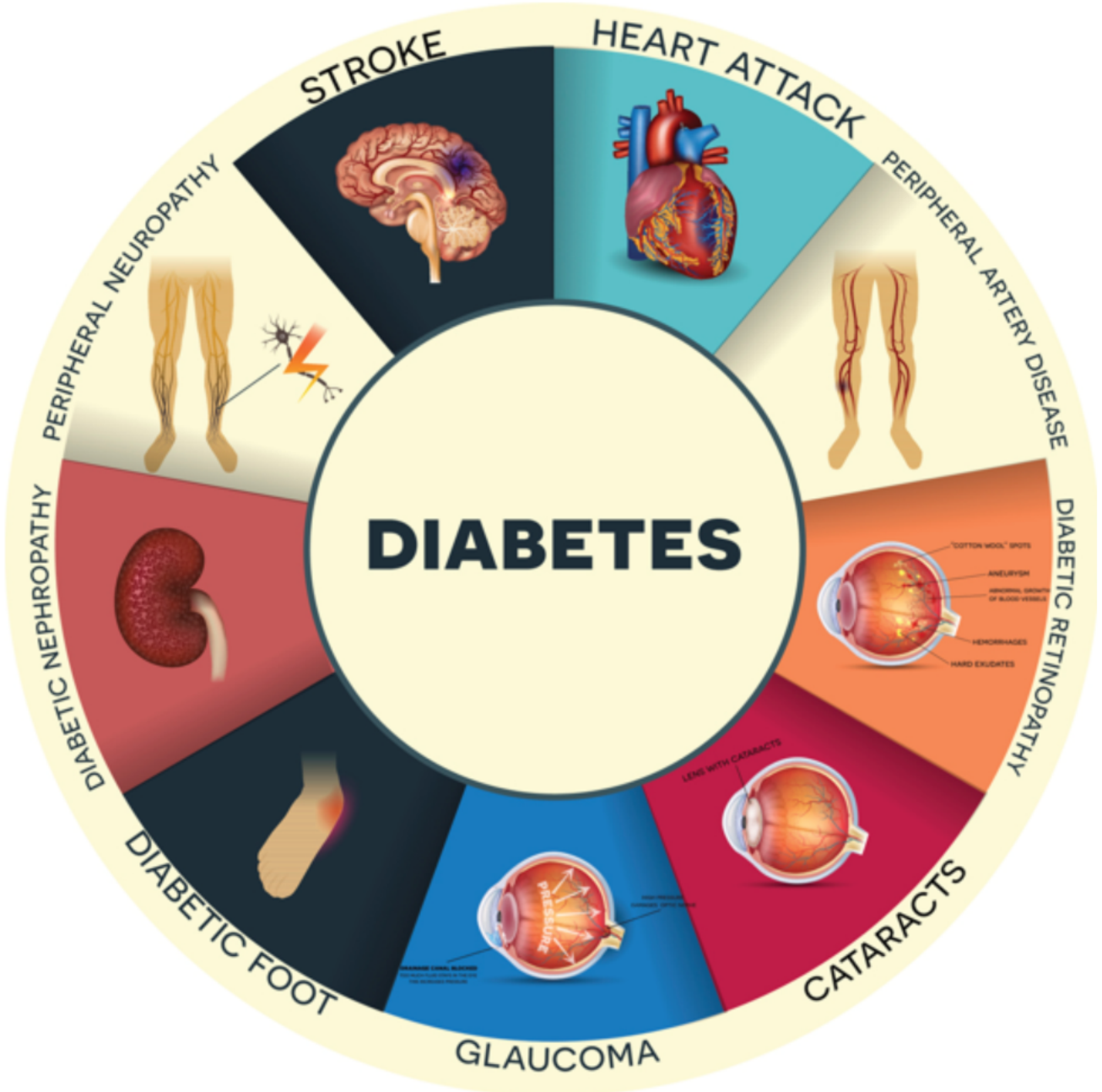
What is Diabetes?

- Endocrine disease



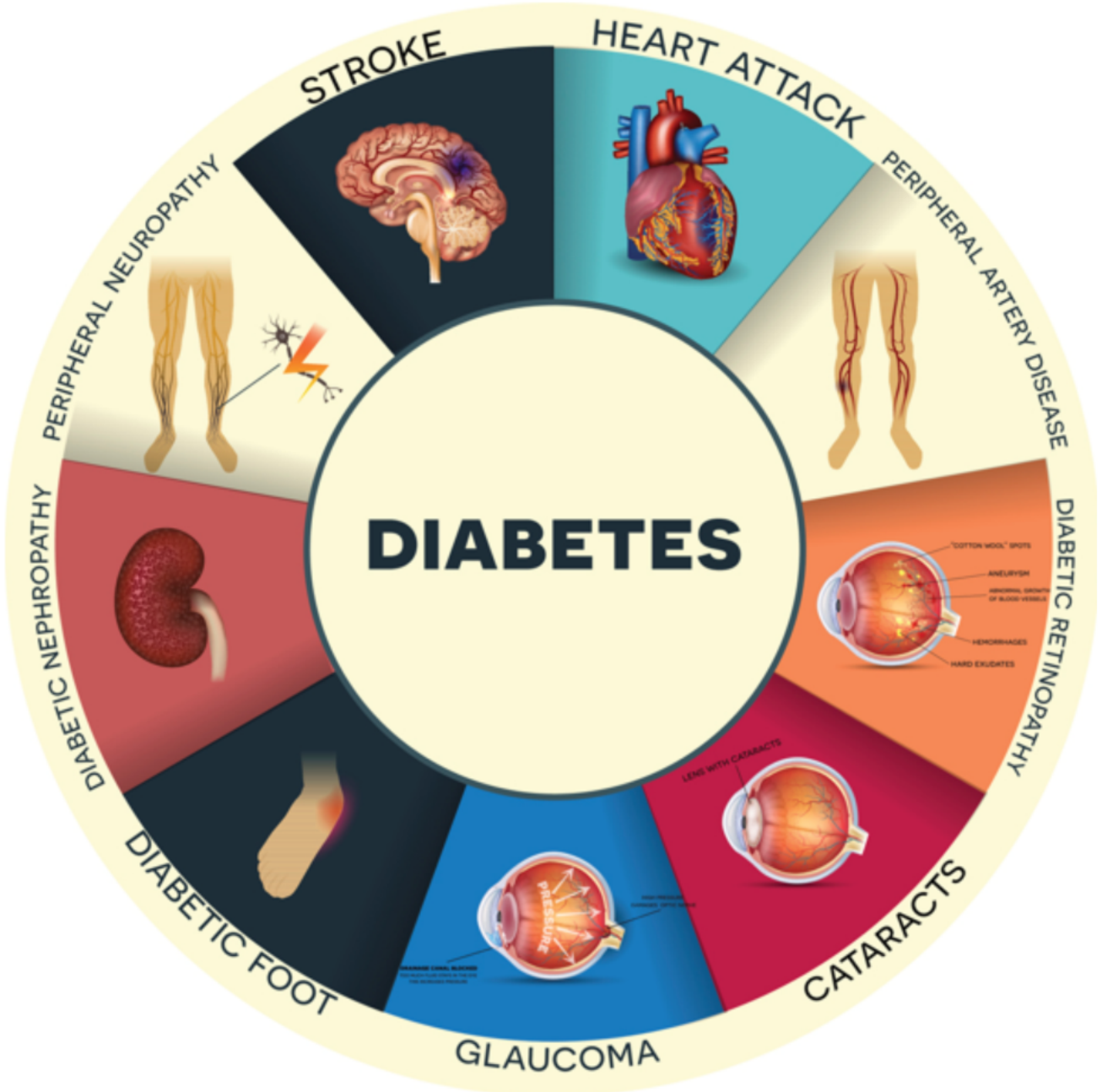
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar



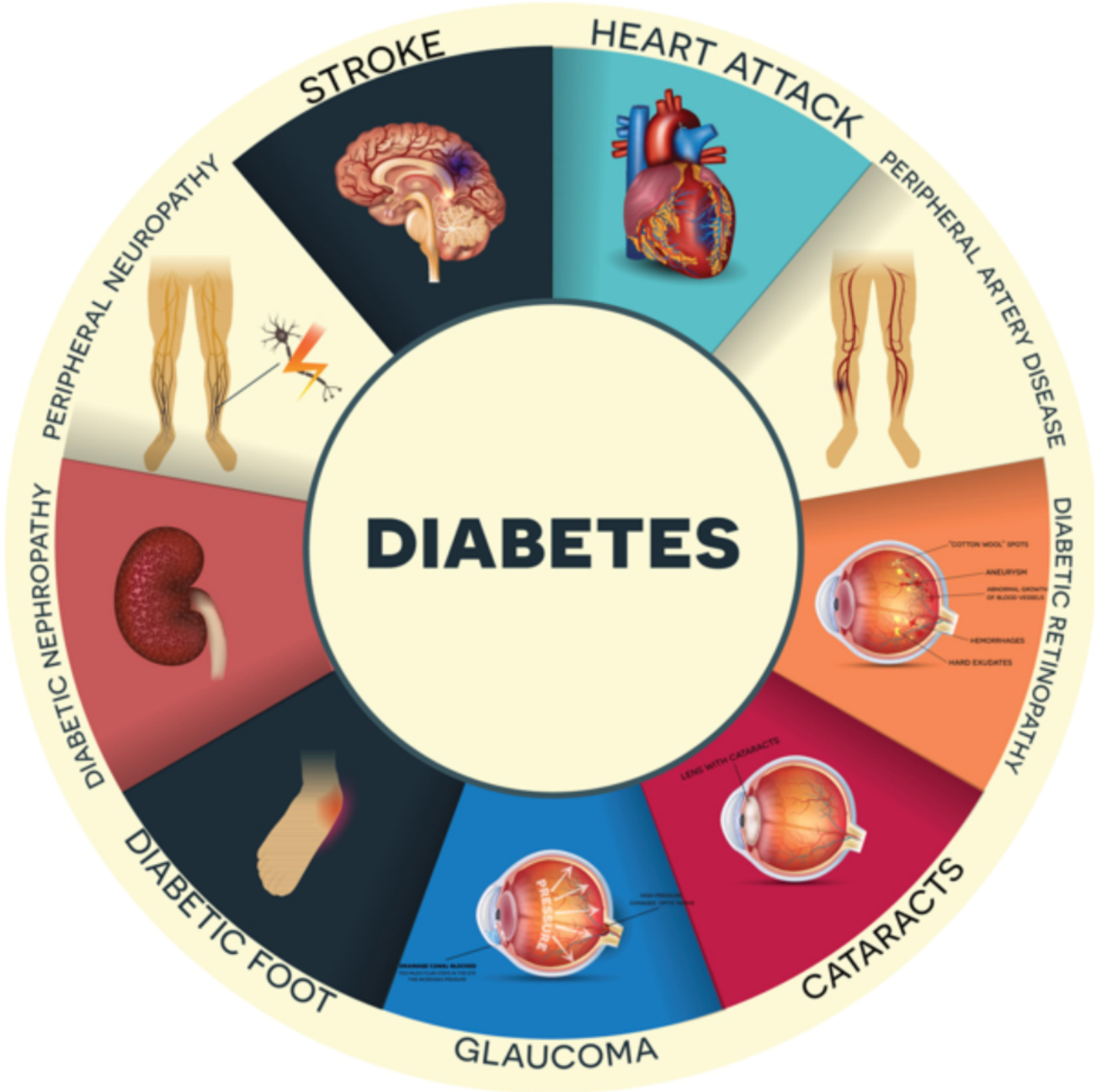
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood



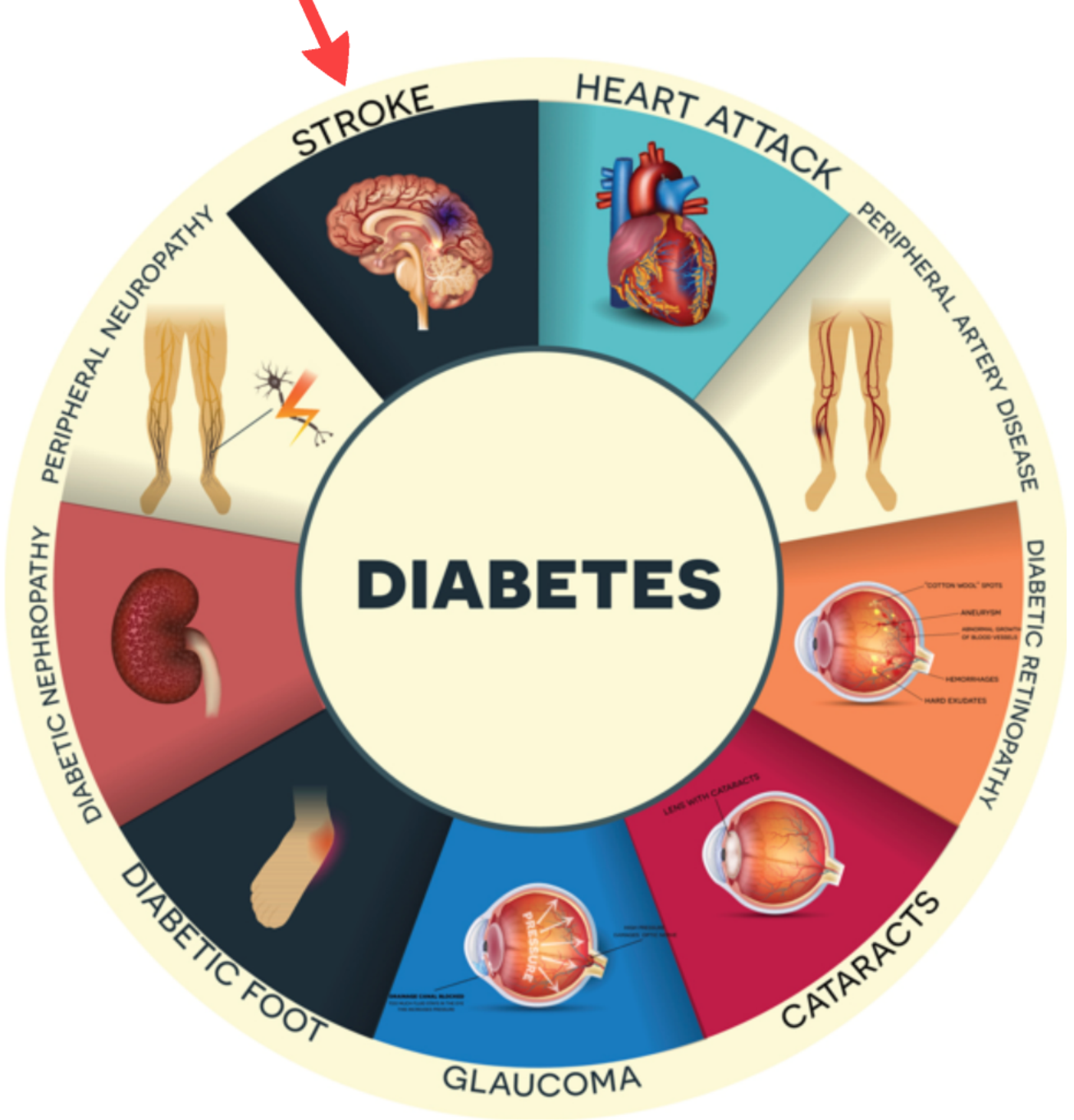
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



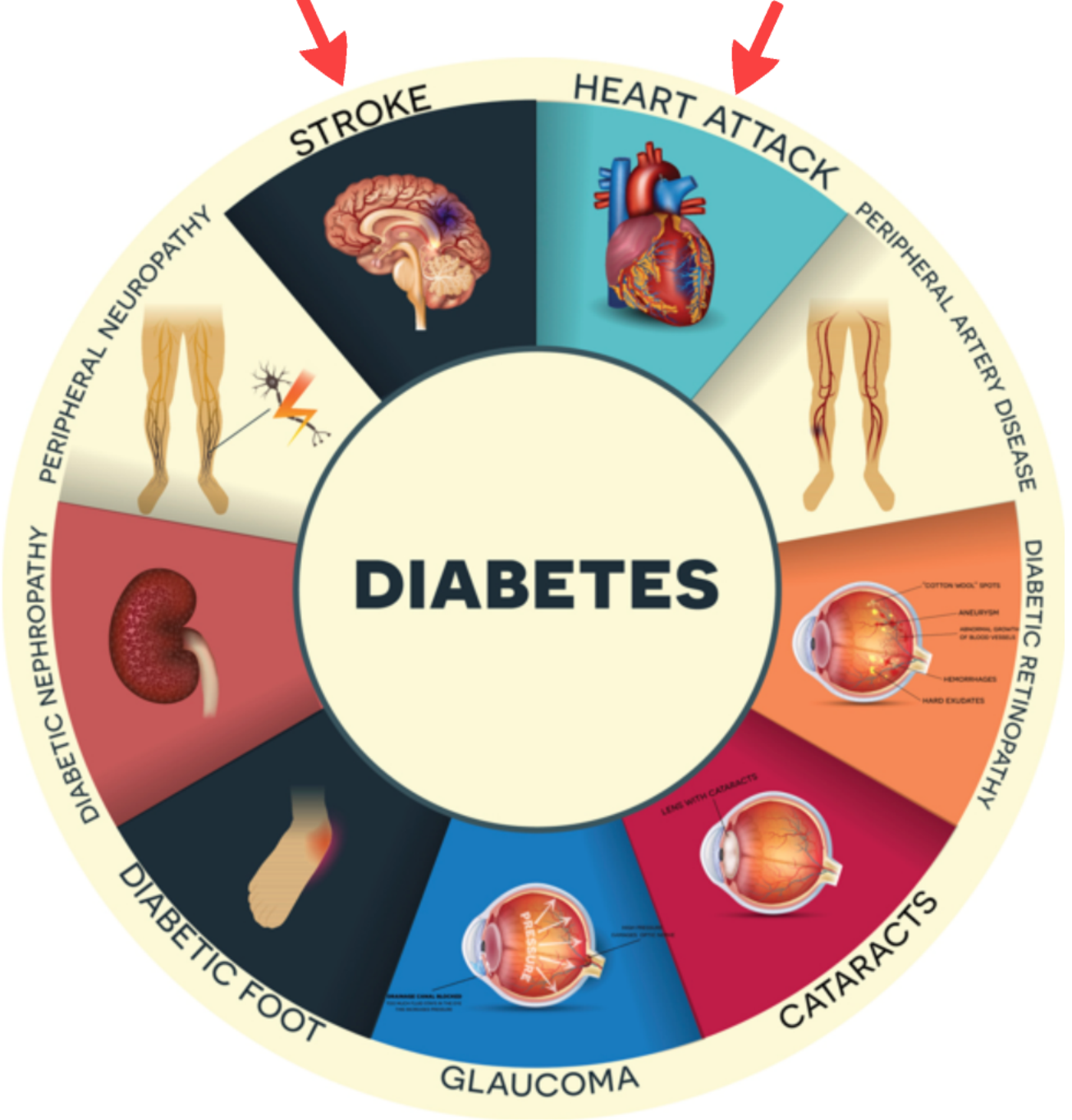
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



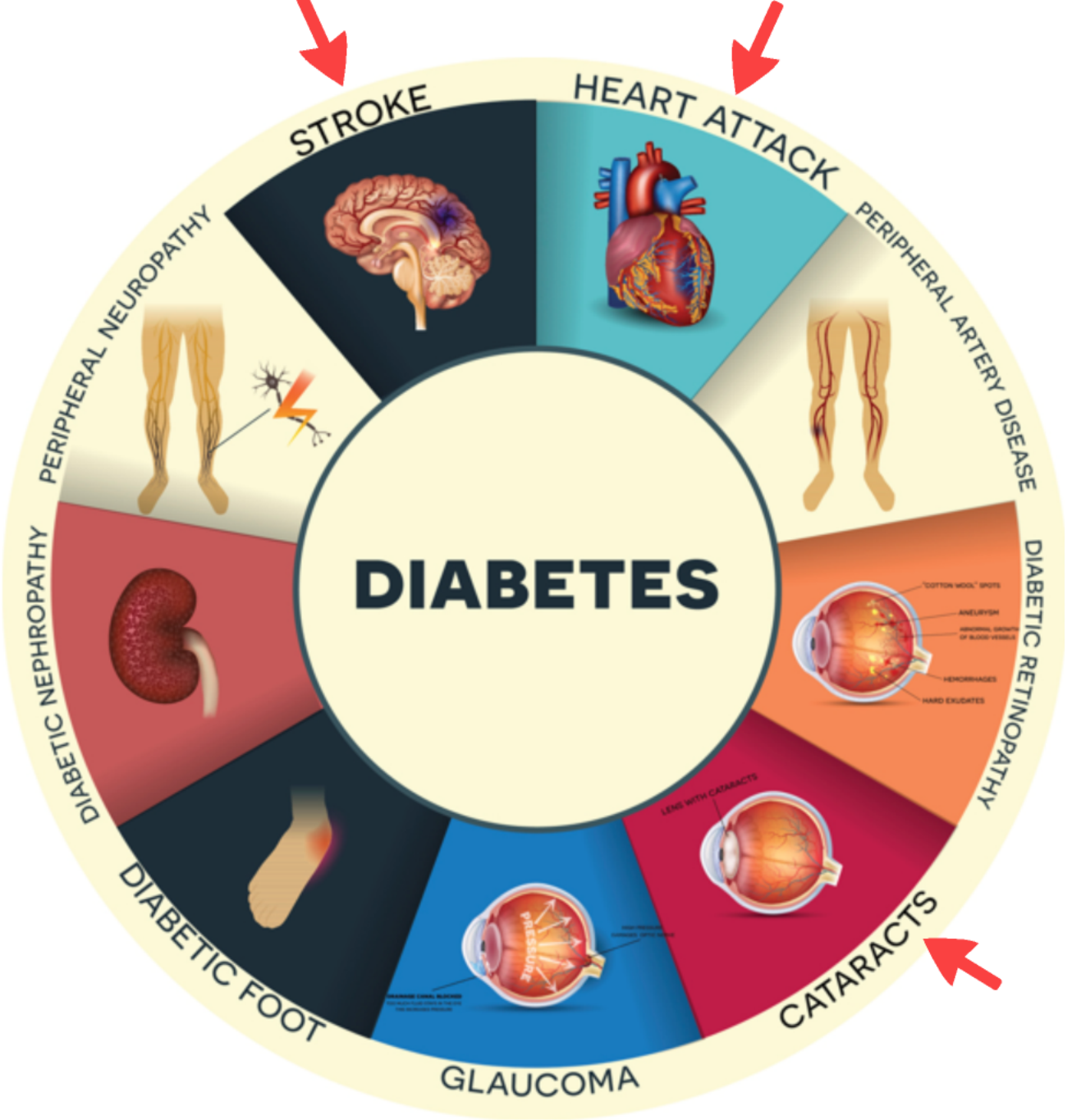
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



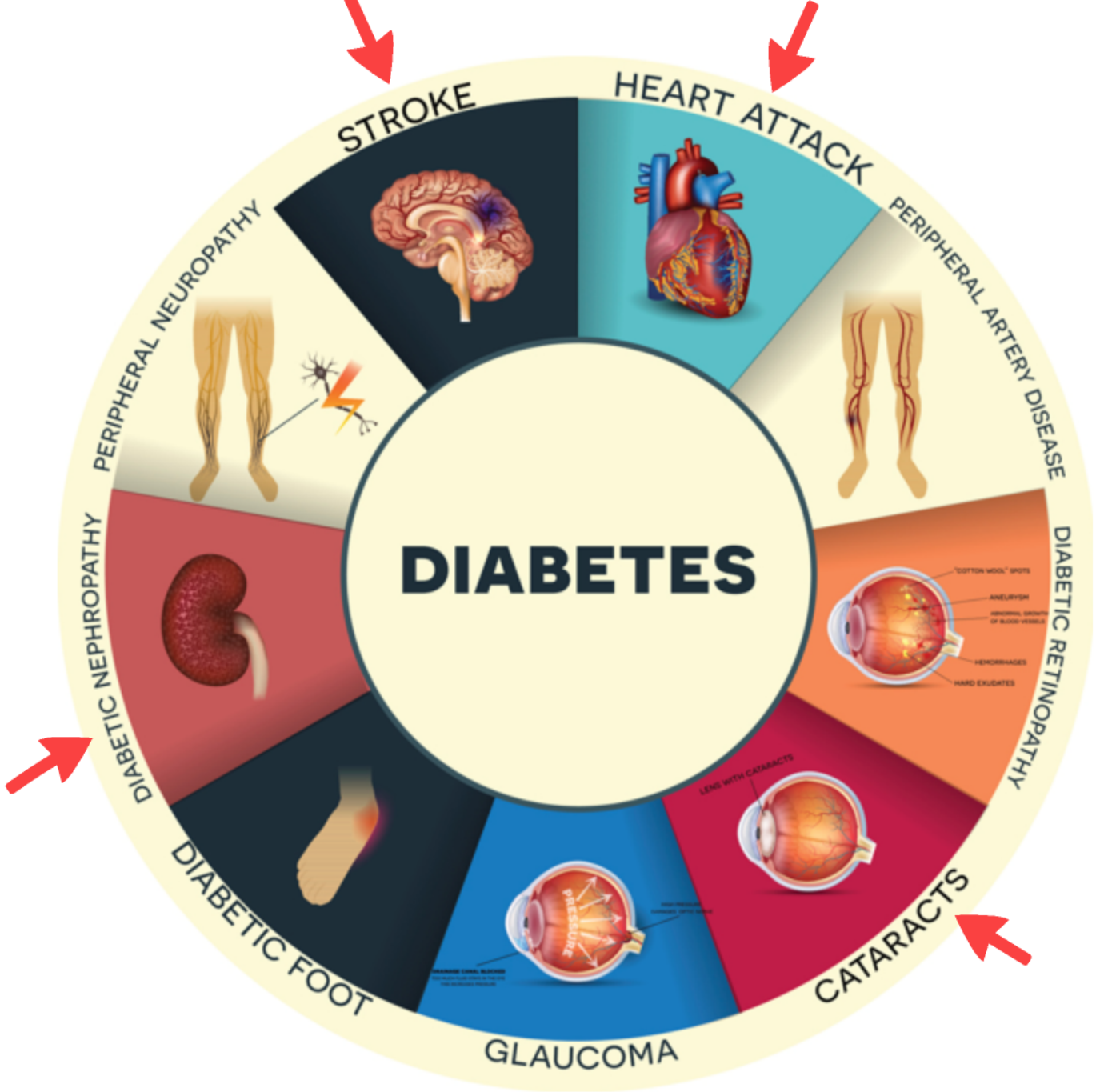
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



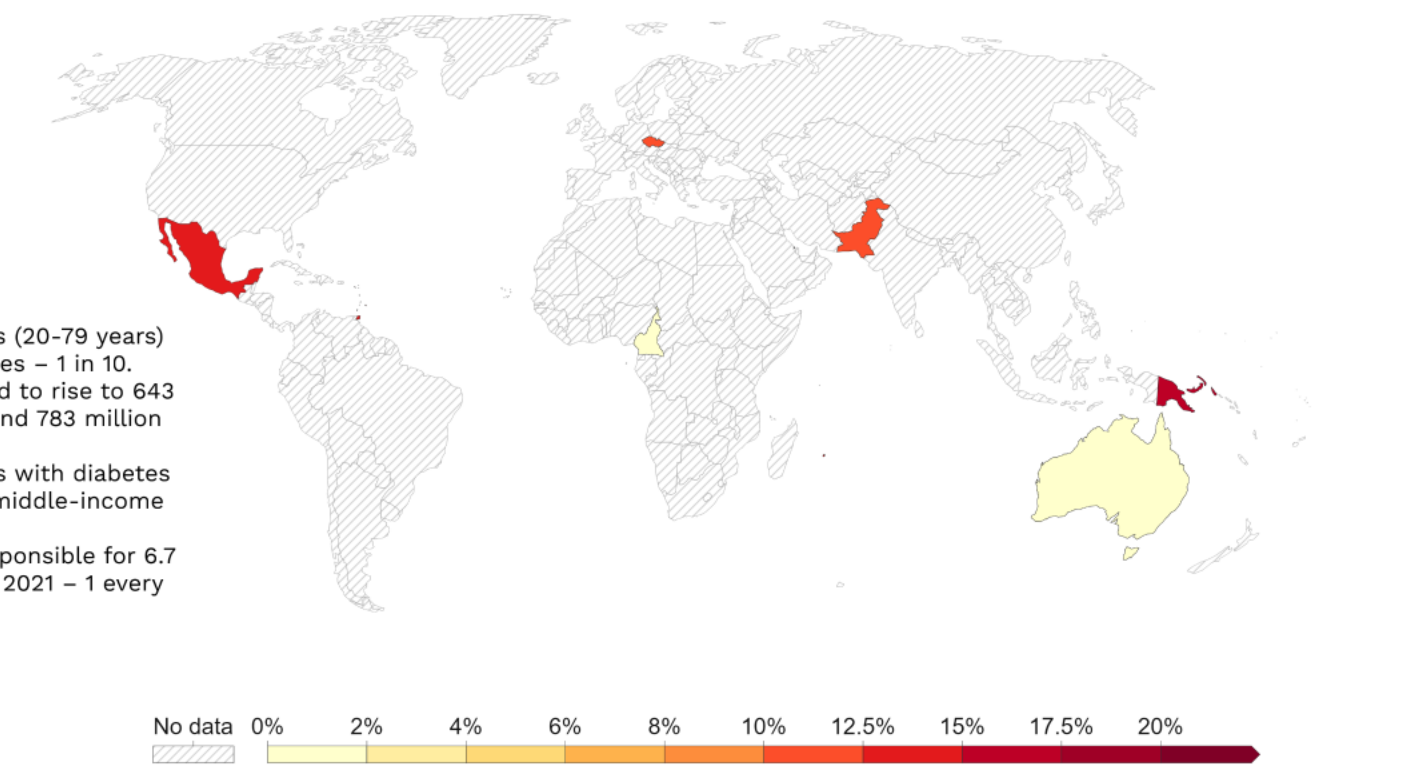
What is Diabetes?

- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



Why Study Diabetes

- 537 million adults (20-79 years) living with diabetes – 1 in 10. Number predicted to rise to 643 million by 2030 and 783 million by 2045.
- Over 3 in 4 adults with diabetes live in low- and middle-income countries.
- Diabetes was responsible for 6.7 million deaths in 2021 – 1 every 5 seconds.

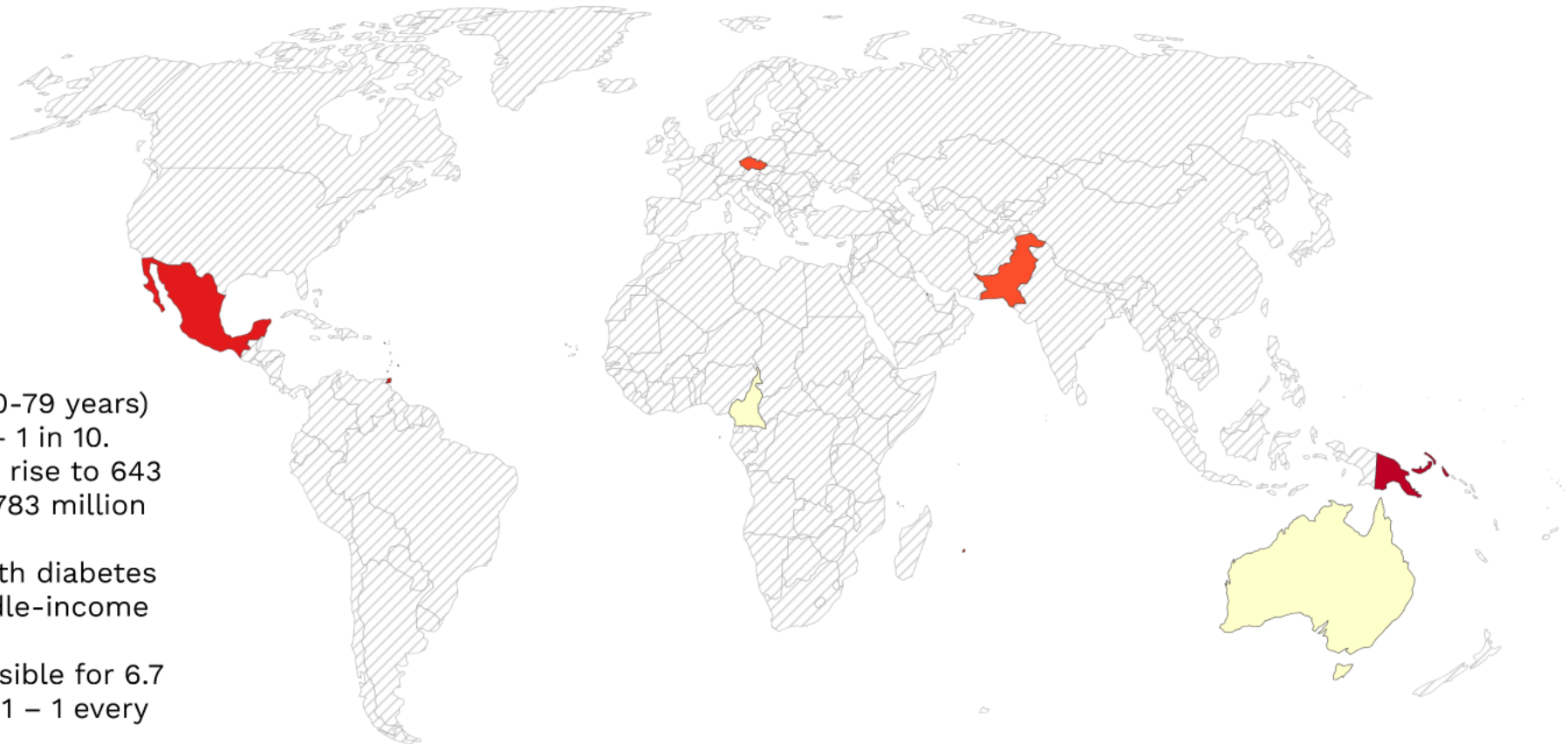


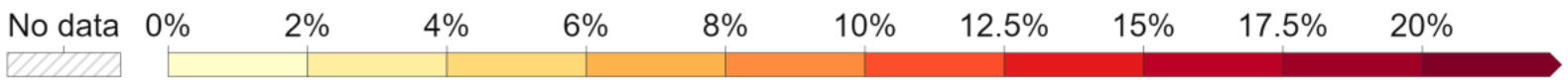
Diabetes prevalence, 2000

The share of people aged 20-79 who have diabetes[1] .

Our World in Data

- 537 million adults (20-79 years) living with diabetes – 1 in 10. Number predicted to rise to 643 million by 2030 and 783 million by 2045.
- Over 3 in 4 adults with diabetes live in low- and middle-income countries.
- Diabetes was responsible for 6.7 million deaths in 2021 – 1 every 5 seconds.



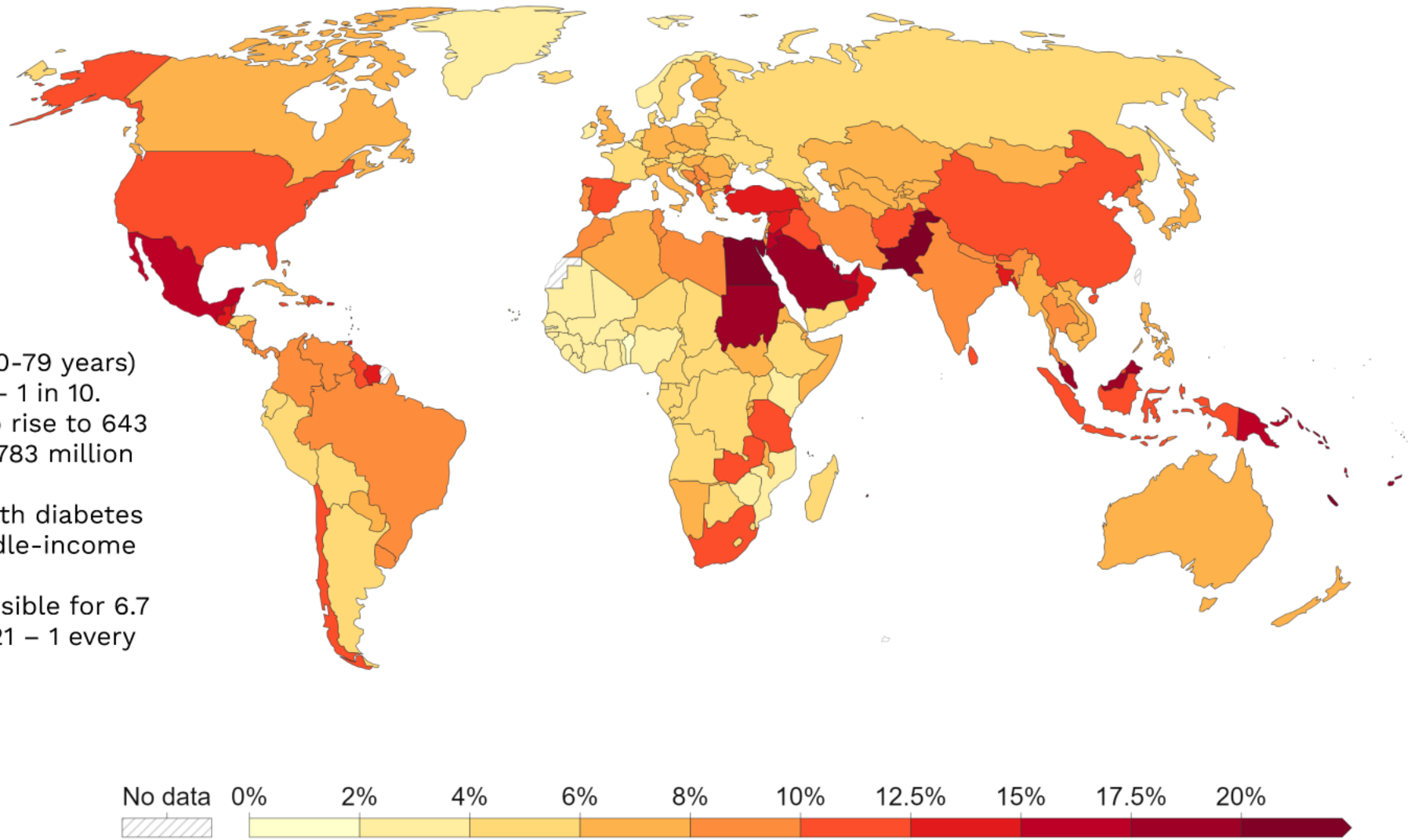


Source: International Diabetes Federation (via World Bank)

Diabetes prevalence, 2021

The share of people aged 20-79 who have diabetes[1] .

Our World in Data



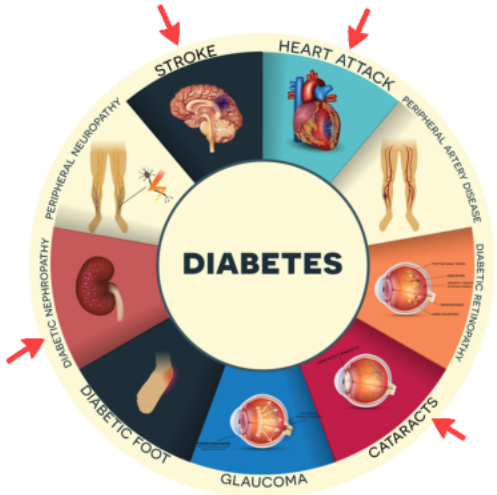
- 537 million adults (20-79 years) living with diabetes – 1 in 10. Number predicted to rise to 643 million by 2030 and 783 million by 2045.
- Over 3 in 4 adults with diabetes live in low- and middle-income countries.
- Diabetes was responsible for 6.7 million deaths in 2021 – 1 every 5 seconds.

Source: International Diabetes Federation (via World Bank)

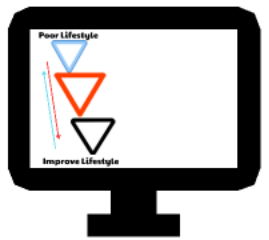
1- INTRODUCTION

What is Diabetes?

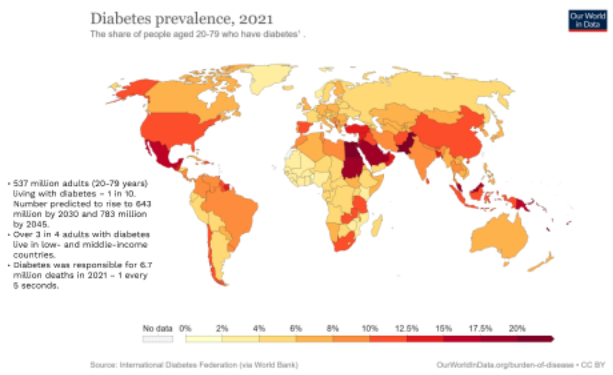
- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



What can be done?

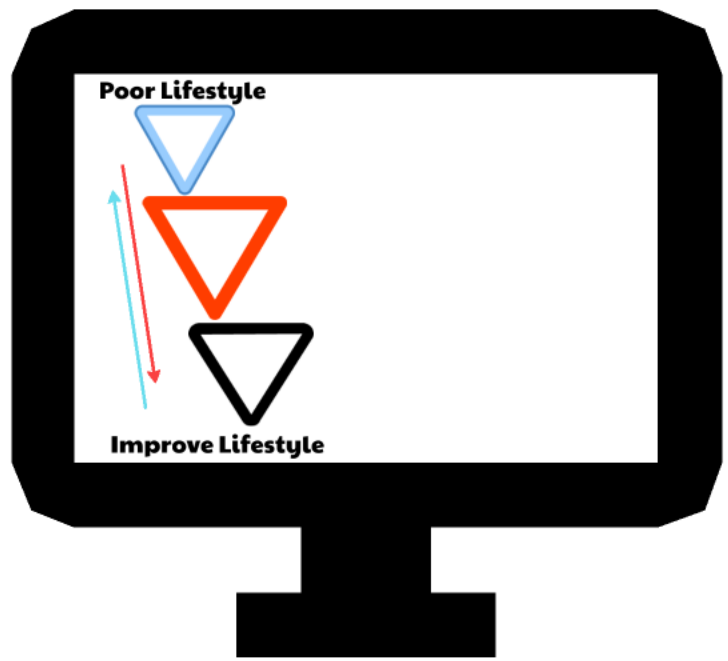


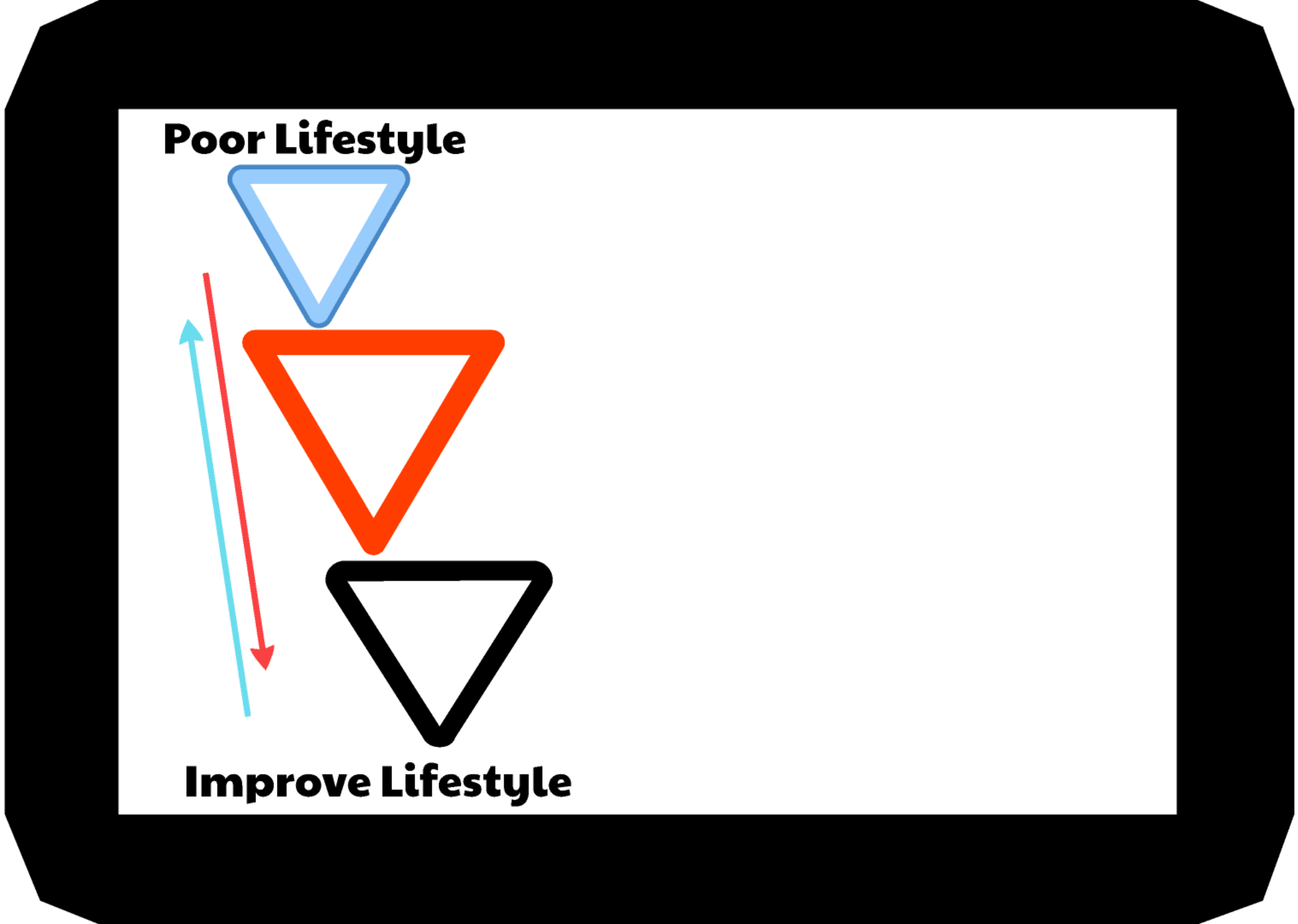
Why Study Diabetes

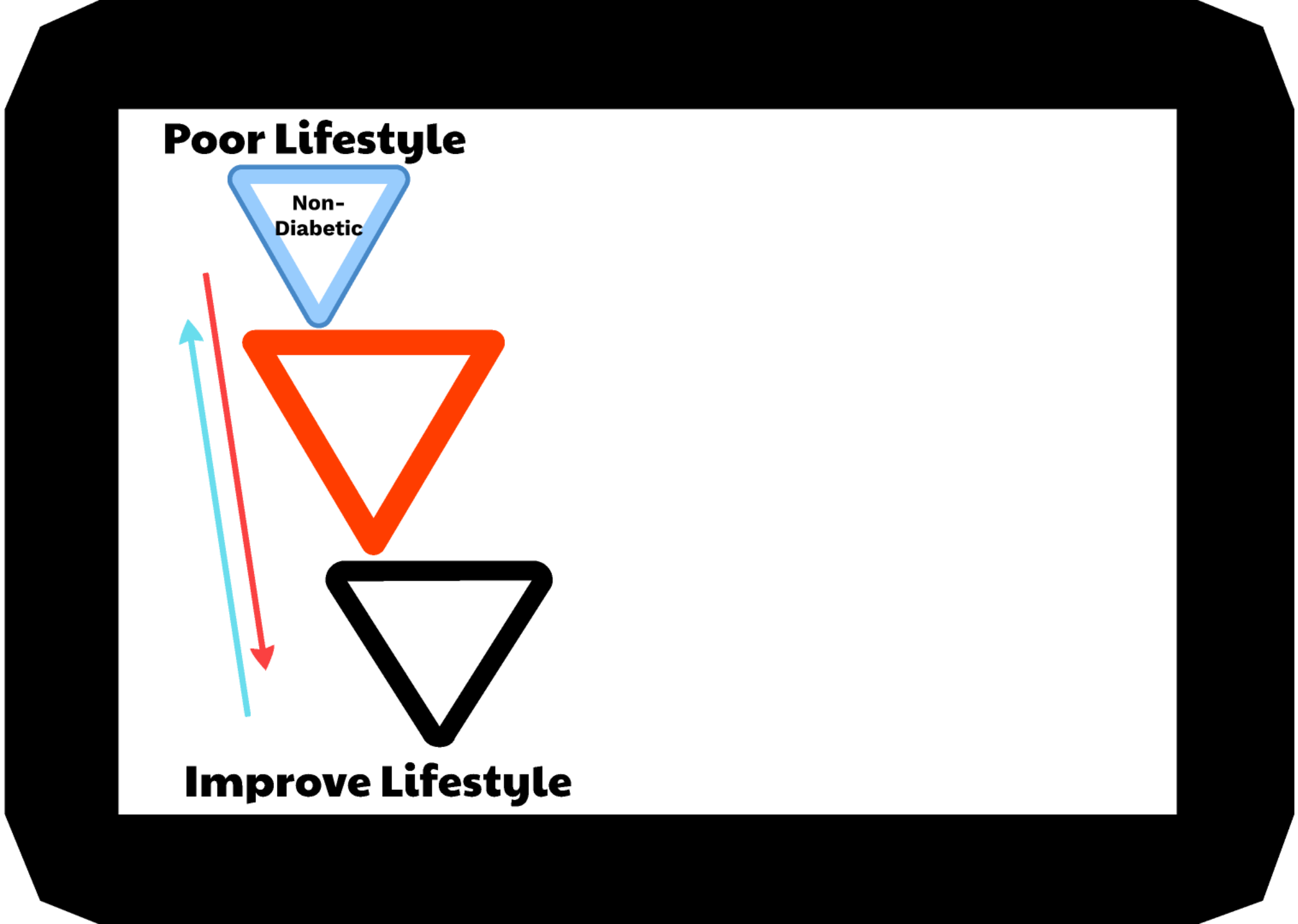


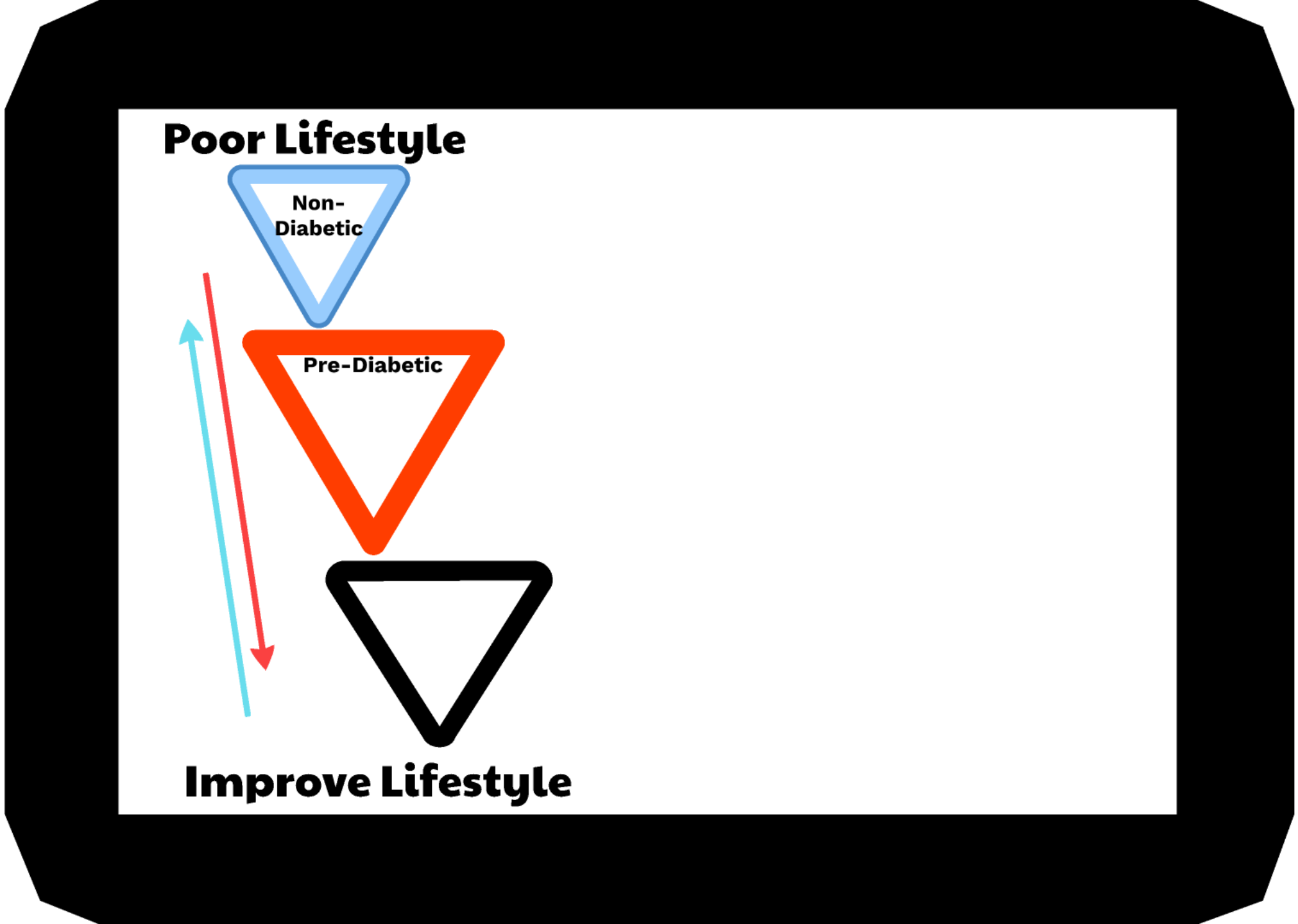
Study Objective

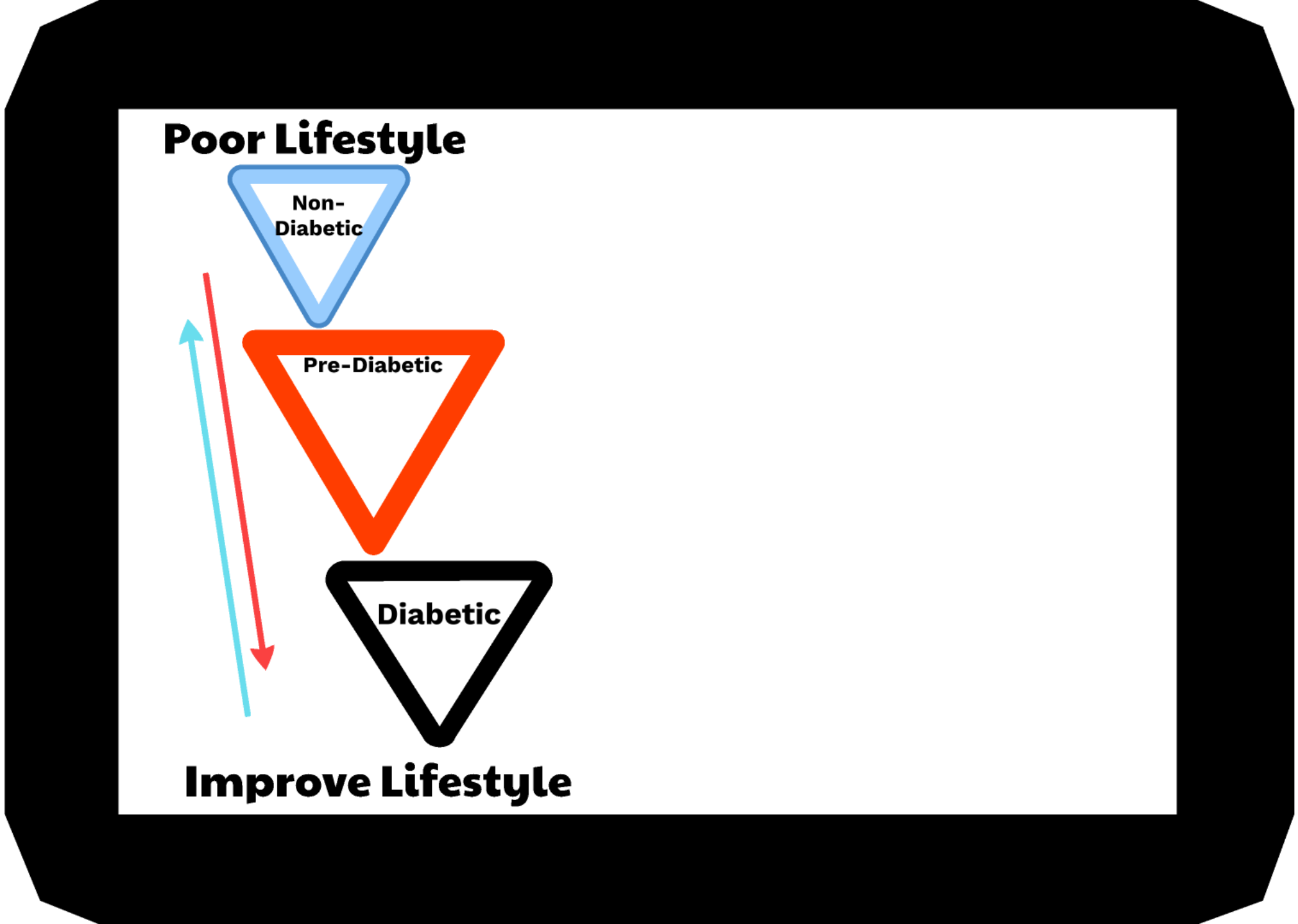
What can be done?

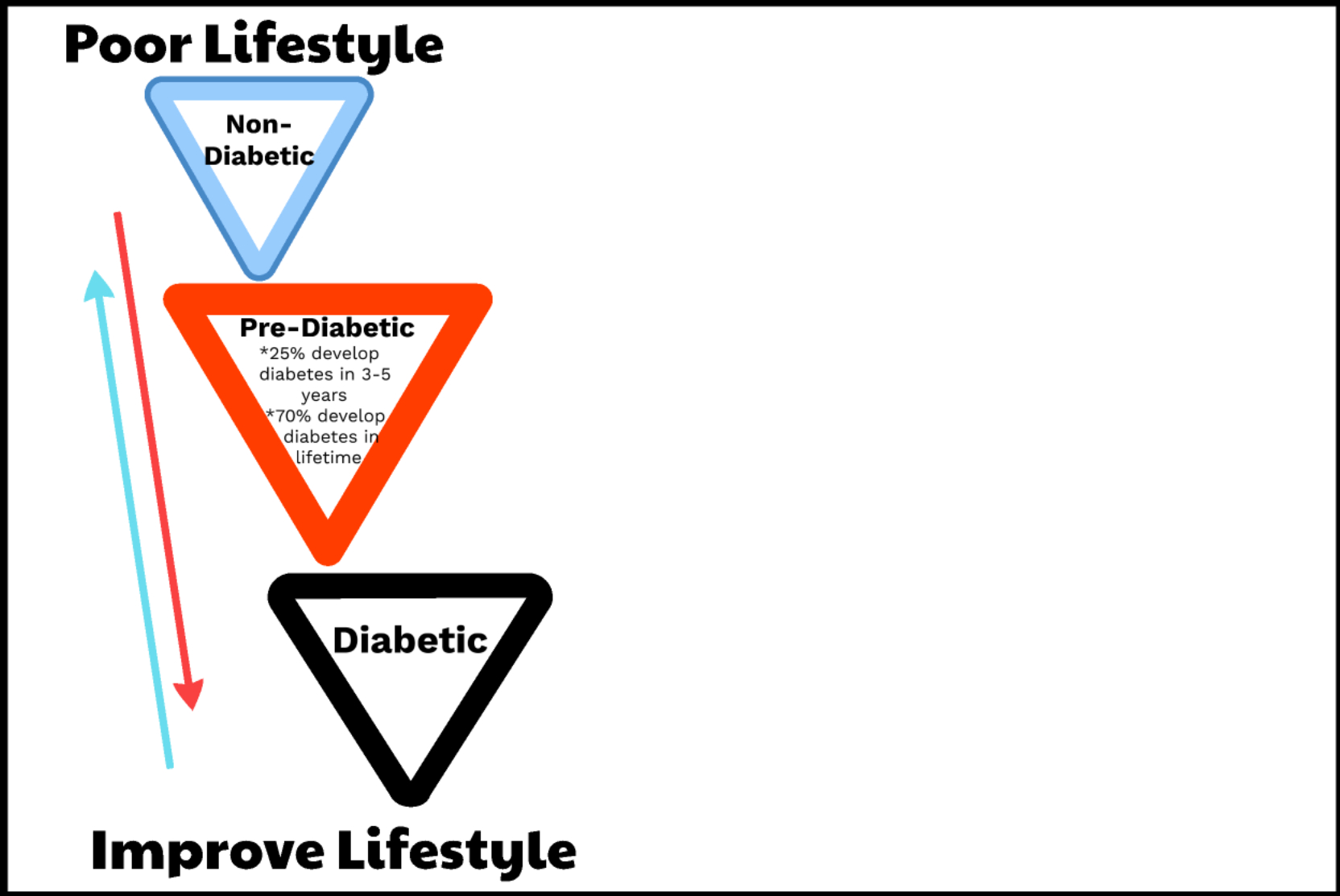


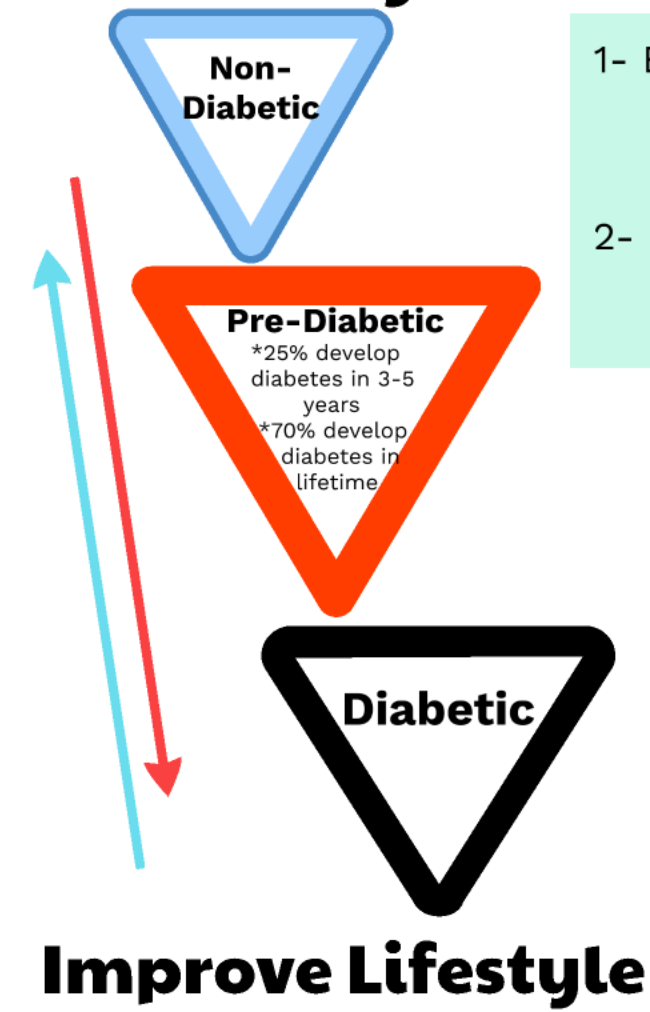

Poor Lifestyle
Improve Lifestyle


Poor Lifestyle
Non-
Diabetic
Improve Lifestyle


Poor Lifestyle
Non-
Diabetic
Pre-Diabetic
Improve Lifestyle

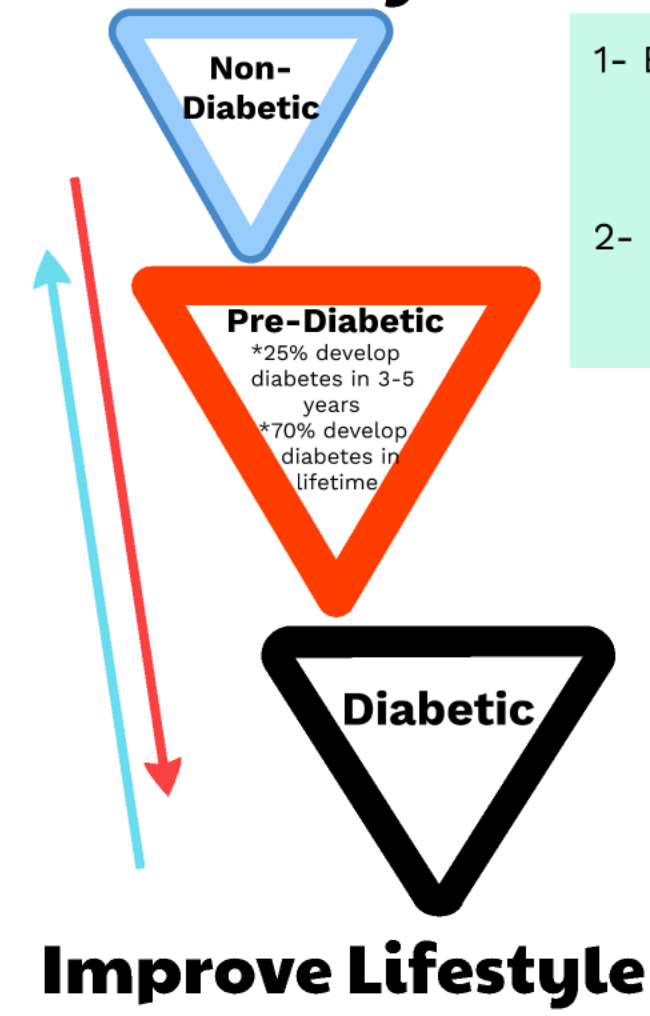

Poor Lifestyle
Non-
Diabetic
Pre-Diabetic
Diabetic
Improve Lifestyle


Poor Lifestyle
Non-Diabetic
Pre-Diabetic
*25% develop diabetes in 3-5 years
*70% develop diabetes in lifetime
Diabetic
Improve Lifestyle



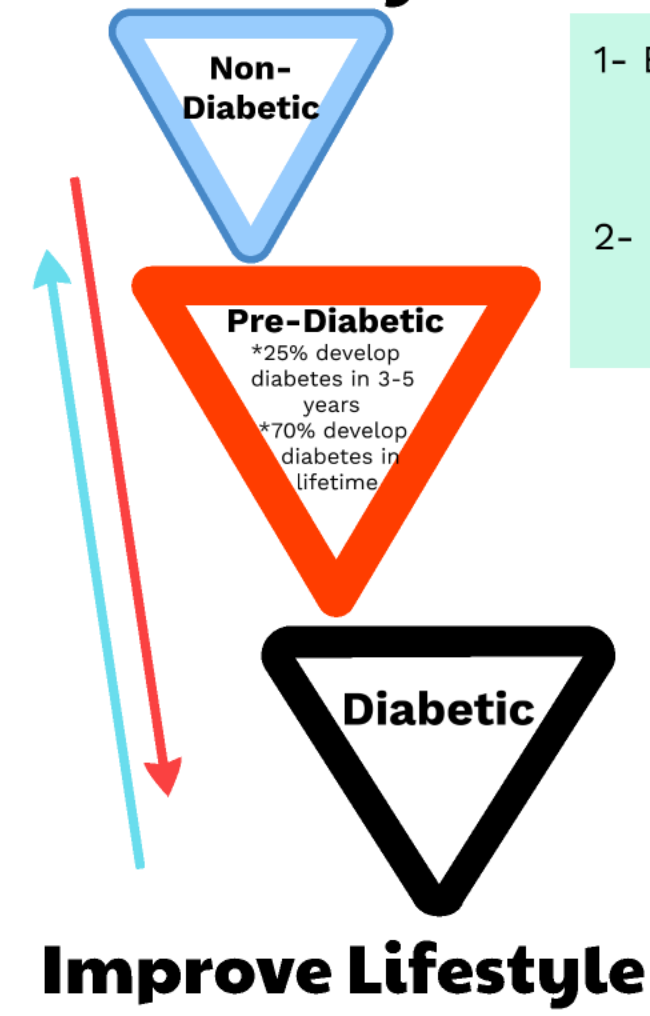
1- Blood work markers:
- Blood glucose (HbA1c)
- Fat profile (TG, Col, etc)

2- Physical changes:
- Body Mass Index (BMI)
- Age



1- Blood work markers:
- Blood glucose (HbA1c)
- Fat profile (TG, Col, etc)

2- Physical changes:
- Body Mass Index (BMI)
- Age



1- Blood work markers:
- Blood glucose (HbA1c)
- Fat profile (TG, Col, etc)

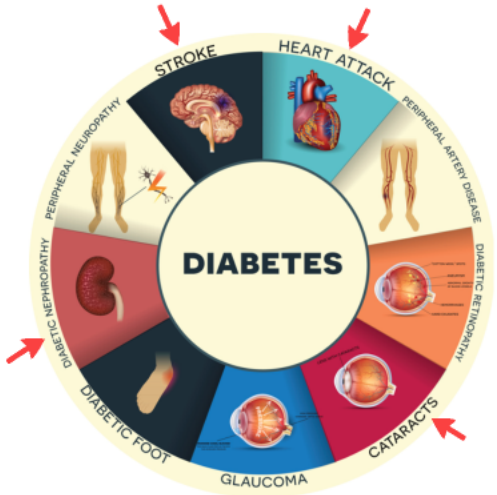
2- Physical changes:
- Body Mass Index (BMI)
- Age

- Identifying **what factors** put an individual at risk
- **Early accurate diagnosis** is especially important:
  - According to CDC **1 in 3 Americans** are pre-diabetic
  - More than **80% don't even know** they have it!
- Progression of pre-diabetes to diabetes can be **slowed** or **reversed** by lifestyle changes or medication

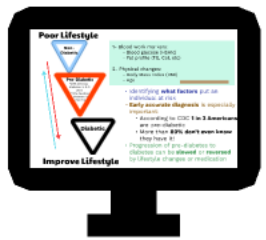
1- INTRODUCTION

What is Diabetes?

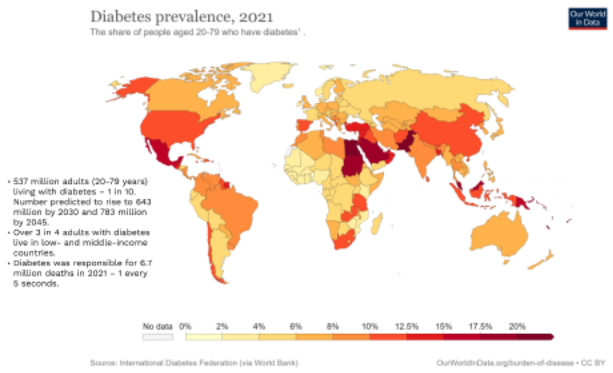
- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



What can be done?

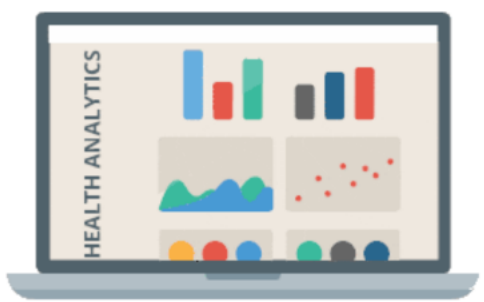


Why Study Diabetes



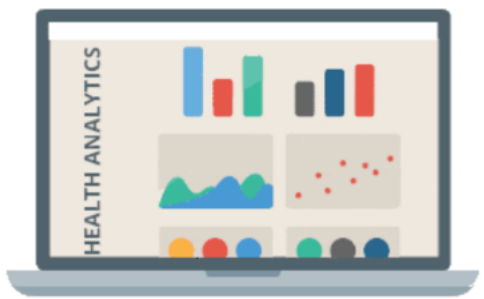
Study Objective

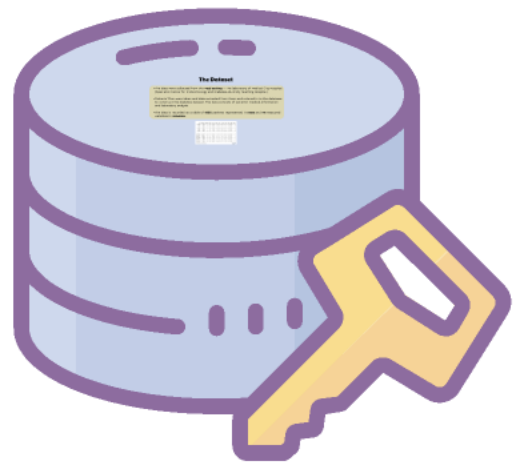
Study Objective



Study Objective

"**Design a Predictive Diabetes Model** that can estimate the **likelihood of diabetes** development based on the provided factors".

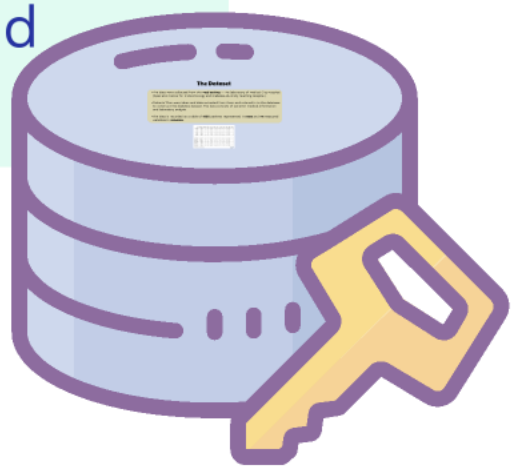




Study Objective

"**Design a Predictive Diabetes Model** that can estimate the **likelihood of diabetes** development based on the provided factors".

Such a model can provide **valuable insights** for **healthcare professionals** and **researchers** in developing targeted interventions and preventative strategies.



The Dataset

- The data were collected from the **Iraqi society** – The laboratory of Medical City Hospital (Specialist Center for Endocrinology and Diabetes-Al-Kindy Teaching Hospital).
- Patients' files were taken and data extracted from them and entered in to the database to construct the diabetes dataset. The data consists of patients' medical information and laboratory analysis.
- The data is recorded as a table of **1000** patients represented in **rows** and **14** measured variables in **columns.**

	ID	No_Pation	Gender	AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	CLASS
0	502	17975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
1	735	34221	M	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	23.0	N
2	420	47975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
3	680	87656	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
4	504	34223	M	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	21.0	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	200	454317	M	71	11.0	97	7.0	7.5	1.7	1.2	1.8	0.6	30.0	Y
996	671	876534	M	31	3.0	60	12.3	4.1	2.2	0.7	2.4	15.4	37.2	Y
997	669	87654	M	30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Y
998	99	24004	M	38	5.8	59	6.7	5.3	2.0	1.6	2.9	14.0	40.5	Y
999	248	24054	M	54	5.0	67	6.9	3.8	1.7	1.1	3.0	0.7	33.0	Y

1000 rows × 14 columns

	ID	No_Pation	Gender	AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	CLASS
0	502	17975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
1	735	34221	M	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	23.0	N
2	420	47975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
3	680	87656	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
4	504	34223	M	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	21.0	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	200	454317	M	71	11.0	97	7.0	7.5	1.7	1.2	1.8	0.6	30.0	Y
996	671	876534	M	31	3.0	60	12.3	4.1	2.2	0.7	2.4	15.4	37.2	Y
997	669	87654	M	30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Y
998	99	24004	M	38	5.8	59	6.7	5.3	2.0	1.6	2.9	14.0	40.5	Y
999	248	24054	M	54	5.0	67	6.9	3.8	1.7	1.1	3.0	0.7	33.0	Y

1000 rows × 14 columns

Variable	Description
ID	Patient ID
No_of Pation	Patient number
Gender	Male or Female
Age	Age of patient in years
Urea	Nitrogenous end-product derived from dietary protein and tissue protein turnover
Cr	Creatinine is the nitrogenous end-product from muscle and protein metabolism and is an important indicator of kidney health
HbA1c	A form of haemoglobin chemically linked to sugar and is a measure to test the **average blood sugar** over the past 90 days
Chol	**Cholesterol** is a type of lipid (fat) that helps the body perform many important functions and comes from our liver and our diet - it travels through the blood on lipoproteins HDL and LDL
TG	**Triglycerides** come from our diet and are a type of lipid that circulate in our blood that the body uses for energy

	Range
	-
	-
	-
	-
tissue protein turnover	Normal range is 1.8 to 7.1 mmol urea/l
protein metabolism and is an	Normal range is 61.9 to 114.9 µmol/L for men is 53 to 97.2 µmol/L for women
neasure to test the **average blood**	Measured as a percentage, a normal reading is between 4.0-5.6%, pre-diabetic 5.7-6.4%, and diabetic over 6.4%
rm many important functions and ood on lipoproteins HDL and LDL	A normal healthy reading for total serum cholesterol should be below 5 mmol/L
t circulate in our blood that the body	Less than 1.7 mmol/L is considered normal, 1.8-2.2 mmol/L borderline high, 2.3-5.6 mmol/L high, and very high 5.7 mmol/L and above

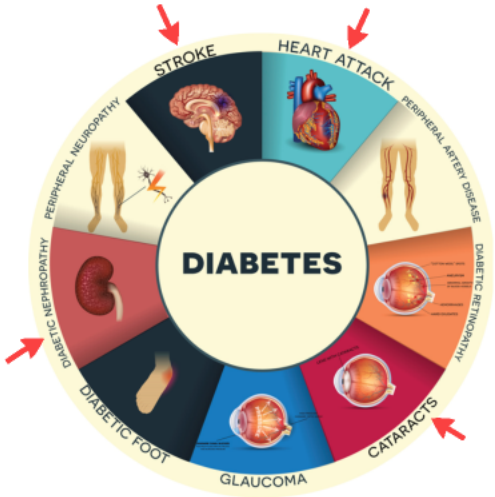
HbA1c	A form of haemoglobin chemically linked to sugar and is a measure to test the **average blood sugar** over the past 90 days
Chol	**Cholesterol** is a type of lipid (fat) that helps the body perform many important functions and comes from our liver and our diet - it travels through the blood on lipoproteins HDL and LDL
TG	**Triglycerides** come from our diet and are a type of lipid that circulate in our blood that the body uses for energy
HDL	**High-density lipoprotein** is known as the "good" cholesterol - it absorbs cholesterol in the blood and takes it to the liver to be flushed out the body
LDL	**Low-density lipoprotein** is known as the "bad" cholesterol - too much can build up in the walls of our blood vessels and cause health problems
VLDL	**Very-low-density-lipoprotein** is a type of LDL and carries triglycerides to the body's cells and tissues - too much VLDL can build up in the walls of our blood vessels
BMI	**Body mass index** is a measure of body fat based on height and weight measured in kg/m^2
Class	Refers to diagnosis as **N** as non-diabetic, **P** as predicted-diabetic (pre-diabetic), and **Y** as diabetic

neasure to test the **average blood**	Measured as a percentage, a normal reading is between 4.0-5.6%, pre-diabetic 5.7-6.4%, and diabetic over 6.4%
rm many important functions and ood on lipoproteins HDL and LDL	A normal healthy reading for total serum cholesterol should be below 5 mmol/L
t circulate in our blood that the body	Less than 1.7 mmol/L is considered normal, 1.8-2.2 mmol/L borderline high, 2.3-5.6 mmol/L high, and very high 5.7 mmol/L and above
ol - it absorbs cholesterol in the	HDL should ideally be above 1 mmol/L for men and 1.2 mmol/L for women
- too much can build up in the walls	LDL should ideally be below 3 mmol/L
iglycerides to the body's cells and od vessels	A normal VLDL level is anything up to 0.77 mmol/L
and weight measured in kg/m^2	A BMI of less than 18.5 is underweight, 18.5-24.9 is healthy, 25.0-29.9 is overweight, and greater than 30.0 is obese
betic (pre-diabetic), and **Y** as diabetic	

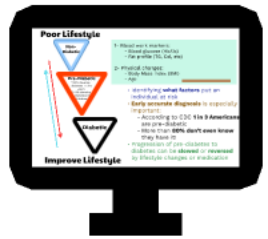
1- INTRODUCTION

What is Diabetes?

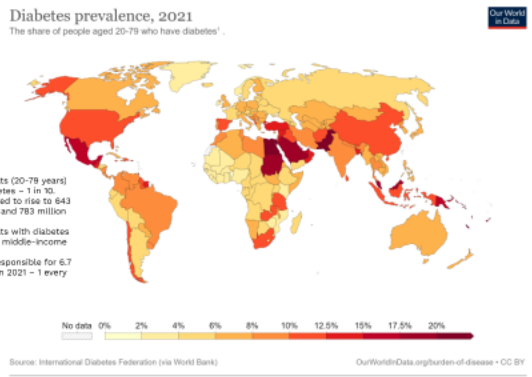
- Endocrine disease
- Sustained **high** blood sugar
- Type 1 :
  - 10% of cases
  - No insulin production
  - Autoimmune disease, complex pathogenesis, poorly understood
- Type 2 :
  - > 90% of cases
  - High insulin production
  - No insulin response by cells
  - Primarily a lifestyle disease, but genetics and family history play a role



What can be done?



Why Study Diabetes



Study Objective

"**Design a Predictive Diabetes Model** that can estimate the **likelihood of diabetes** development based on the provided factors".

Such a model can provide **valuable insights** for **healthcare professionals** and **researchers** in developing targeted interventions and preventative strategies.

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

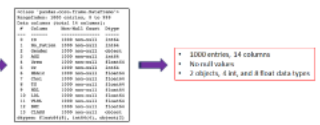
codeop

Index

2- DATA PRE-PROCESSING & FEATURE ENGINEERING

1. Overview of dataset

data.info()



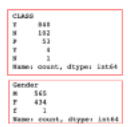
- 1000 entries, 14 columns
- No null values
- 2 objects, 4 int, and 8 float data types

2. We were particularly interested to see **how many categories** exist in **"CLASS" & "Gender"** columns as **'object'** data types, and **number of patients** in each category.

"CLASS" column = **"target variable"** classifies patients in 3 categories:
Non-diabetic (N), Pre-diabetic (P) and **Diabetic(Y)**

"Gender" column classifies patients in 2 categories:
Female (F) and **Male (M)**

value_counts()



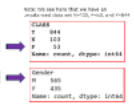
Redundant categories were found

3. Redundant categories were found, due to typing error!

So we replaced:

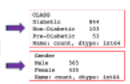
Any spaces (" ") with an empty string ("") — str.strip()

Lowercase letters with uppercase ones — str.upper()



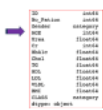
4. For easier readability of the code we replaced the values in "CLASS" and "Gender as following:
"N" to "Non-Diabetic"
"P" to "Pre-Diabetic"
"Y" to "Diabetic"
"F" to "Female"
"M" to "Male"

map()



5. We then converted the "CLASS" and "Gender" columns from 'object' to 'category' data type.

astype()



6. Two numerical columns as "CLASS_Category" from the "CLASS" column and "Gender_Category" from "Gender" column were added for feasibility of further analysis and machine learning algorithms.

The code we used allowed us to explicitly define the mapping between categories and numerical values:

map('category codes')

```
{'Non-Diabetic': 0, 'Pre-Diabetic': 1, 'Diabetic': 2}
```

With this code numerical values were automatically assigned to each category:

```
{0: 'Female', 1: 'Male'}
```

factorize()

7. Remove unnecessary columns - the "ID" and "No-Pation" columns.

drop()

1. Overview of dataset

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   ID          1000 non-null   int64
 1   No_Pation   1000 non-null   int64
 2   Gender      1000 non-null   object
 3   AGE         1000 non-null   int64
 4   Urea        1000 non-null   float64
 5   Cr          1000 non-null   int64
 6   HbA1c       1000 non-null   float64
 7   Chol        1000 non-null   float64
 8   TG          1000 non-null   float64
 9   HDL         1000 non-null   float64
 10  LDL         1000 non-null   float64
 11  VLDL        1000 non-null   float64
 12  BMI         1000 non-null   float64
 13  CLASS       1000 non-null   object
dtypes: float64(8), int64(4), object(2)
```

- 1000 entries, 14 columns
- No null values
- 2 objects, 4 int, and 8 float data types

2. We were particularly interested to see **how many categories** exist in **“CLASS”** & **“Gender”** columns as **‘object’** data types, and **number of patients** in each category.

“CLASS” column = **"target variable"** classifies patients in 3 categories:
Non-diabetic (N), **Pre-diabetic (P)** and **Diabetic(Y)**

“Gender” column classifies patients in 2 categories:
Female (F) and **Male (M)**

value_counts()

```
CLASS
Y      840
N      102
P       53
Y        4
N        1
Name: count, dtype: int64
```

```
Gender
M    565
F    434
f      1
Name: count, dtype: int64
```

Redundant categories were found

3. Redundant categories were found, due to typing error!

So we replaced:

Any spaces (“ ”) with an empty string (“”)

str.strip()

Note: We see here that we have an unbalanced data set N=103, P=53, and Y=844

```
CLASS
Y    844
N    103
P     53
Name: count, dtype: int64
```

Lowercase letters with uppercase ones

str.upper()

```
Gender
M    565
F    435
Name: count, dtype: int64
```

4. For easier readability of the code we replaced the values in “CLASS” and “Gender as following:
“N” to “Non-Diabetic”
“P” to “Pre-Diabetic”
“Y” to “Diabetic”
“F” to “Female”
“M” to “Male”

.map()

```
CLASS
Diabetic          844
Non-Diabetic      103
Pre-Diabetic       53
Name: count, dtype: int64
```

```
Gender
Male       565
Female     435
Name: count, dtype: int64
```

5. We then converted the “CLASS” and “Gender” columns from ‘object’ to ‘category’ data type.

.astype()

```
ID                int64
No_Pation         int64
Gender         category
AGE               int64
Urea            float64
Cr                int64
HbA1c           float64
Chol            float64
TG              float64
HDL             float64
LDL             float64
VLDL            float64
BMI             float64
CLASS          category
dtype: object
```

6. Two numerical columns as “CLASS_Category” from the “CLASS” column and “Gender_Category” from “Gender” column were added for feasibility of further analysis and machine learning algorithms.

```
CLASS          category
dtype: object
```

6. Two numerical columns as “CLASS_Category” from the “CLASS” column and “Gender_Category” from “Gender” column were added for feasibility of further analysis and machine learning algorithms.

The code we used allowed us to explicitly define the mapping between categories and numerical values:

```
{'Non-Diabetic': 0, 'Pre-Diabetic': 1, 'Diabetic': 2}
```

.map().astype(int)

CLASS	CLASS_Category
Non-Diabetic	0
Non-Diabetic	0
Non-Diabetic	0
Non-Diabetic	0
Non-Diabetic	0

With this code numerical values were automatically assigned to each category:

```
{0: 'Female', 1: 'Male'}
```

.cat.codes

Gender_Category
0
1
0
0
1

7. Remove unnecessary columns - the “ID” and “No-Pation” columns.

.drop()

	Gender	AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	CLASS	CLASS_Category	Gender_Category
0	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
1	Male	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	23.0	Non-Diabetic	0	1
2	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
3	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
4	Male	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	21.0	Non-Diabetic	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	Male	71	11.0	97	7.0	7.5	1.7	1.2	1.8	0.6	30.0	Diabetic	2	1
996	Male	31	3.0	60	12.3	4.1	2.2	0.7	2.4	15.4	37.2	Diabetic	2	1
997	Male	30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Diabetic	2	1
998	Male	38	5.8	59	6.7	5.3	2.0	1.6	2.9	14.0	40.5	Diabetic	2	1
999	Male	54	5.0	67	6.9	3.8	1.7	1.1	3.0	0.7	33.0	Diabetic	2	1

1000 rows × 14 columns

	Gender	AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	BMI	CLASS	CLASS_Category	Gender_Category
0	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
1	Male	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	23.0	Non-Diabetic	0	1
2	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
3	Female	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	Non-Diabetic	0	0
4	Male	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	21.0	Non-Diabetic	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	Male	71	11.0	97	7.0	7.5	1.7	1.2	1.8	0.6	30.0	Diabetic	2	1
996	Male	31	3.0	60	12.3	4.1	2.2	0.7	2.4	15.4	37.2	Diabetic	2	1
997	Male	30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Diabetic	2	1
998	Male	38	5.8	59	6.7	5.3	2.0	1.6	2.9	14.0	40.5	Diabetic	2	1
999	Male	54	5.0	67	6.9	3.8	1.7	1.1	3.0	0.7	33.0	Diabetic	2	1

1000 rows × 14 columns

Diabetes Prediction Model

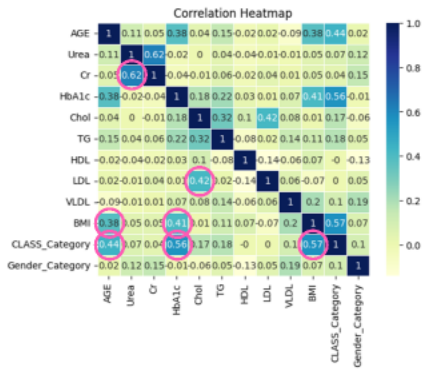
Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

3- EXPLORATORY DATA ANALYSIS (EDA)

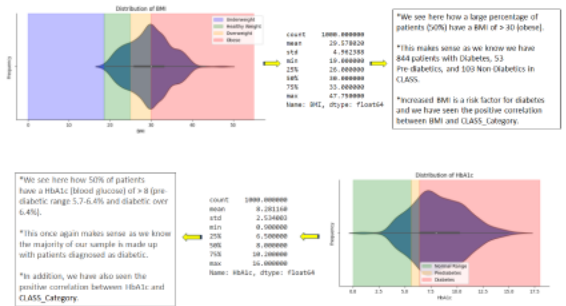


We used a Pearson´s correlation with a heatmap to look at correlations and strength of correlations between our variables.

What we see here, and relevance to analysis:

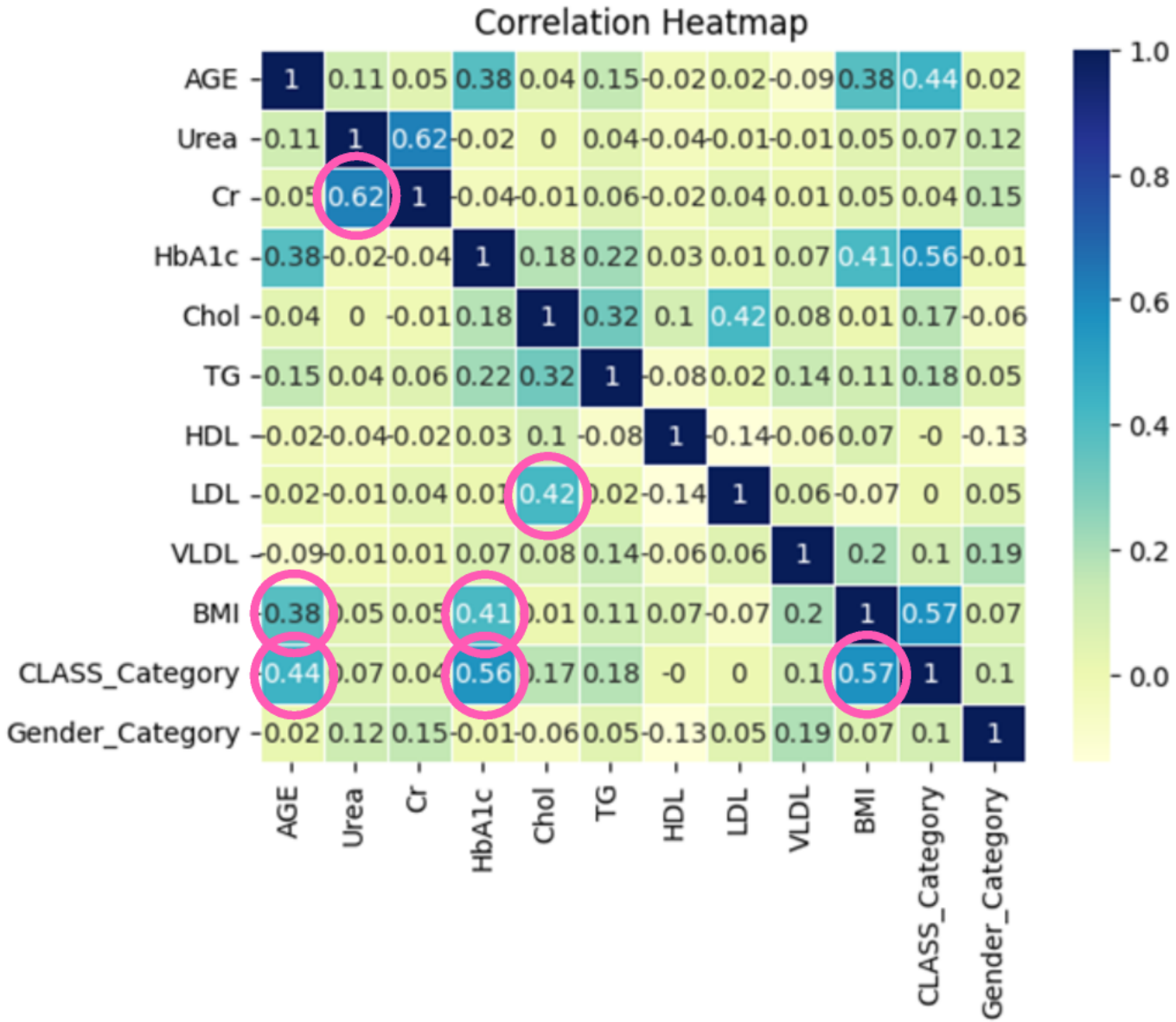
- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.
- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.
- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).

Let's take a look at the distribution and normal ranges of BMI and HbA1c in a little more detail:



Conclusions from EDA...

Interactive Plots With Diabetes Dataset



We used a Pearson´s correlation with a heatmap to look at correlations and strength of correlations between our variables.

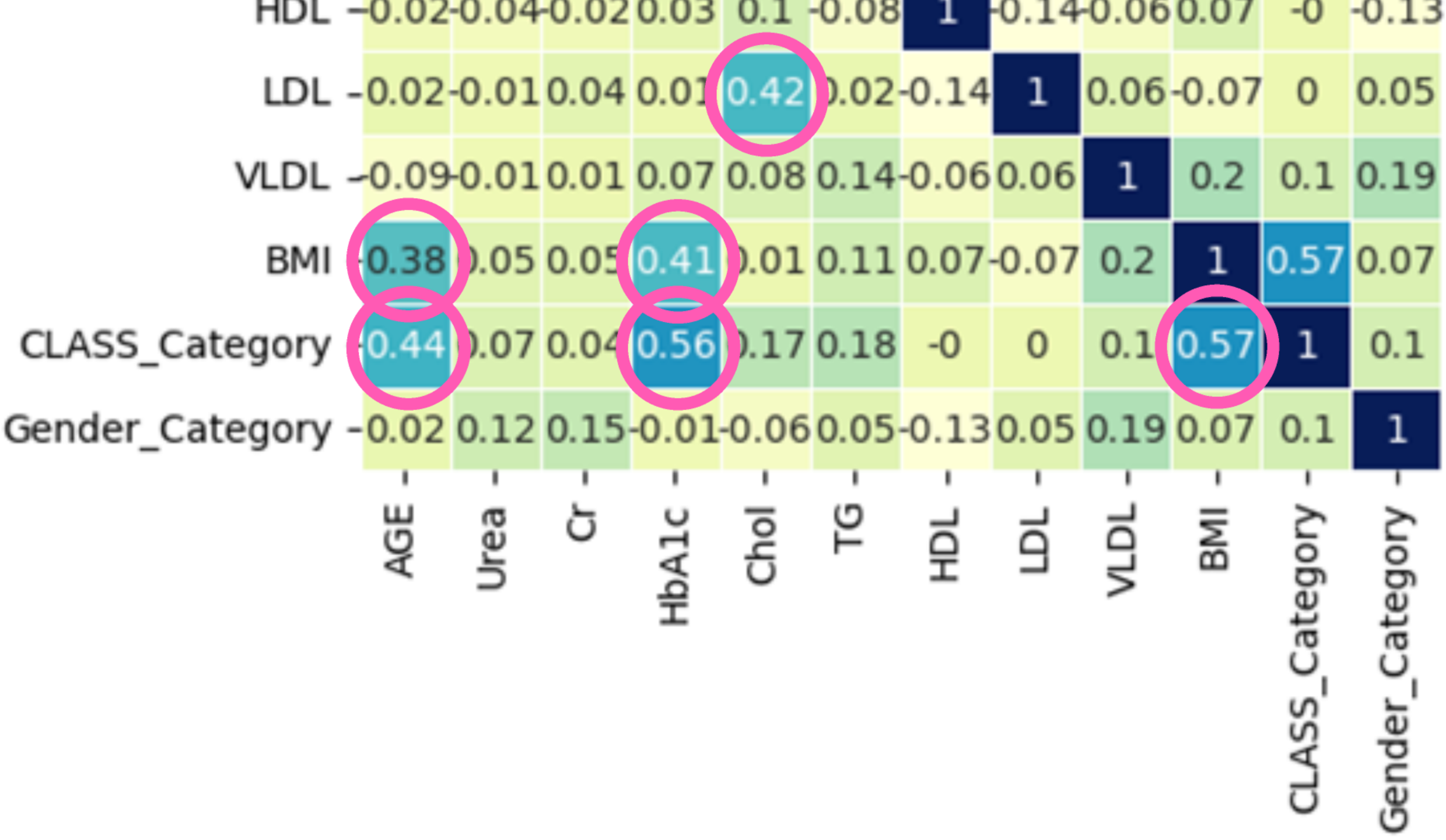
What we see here, and relevance to analysis:

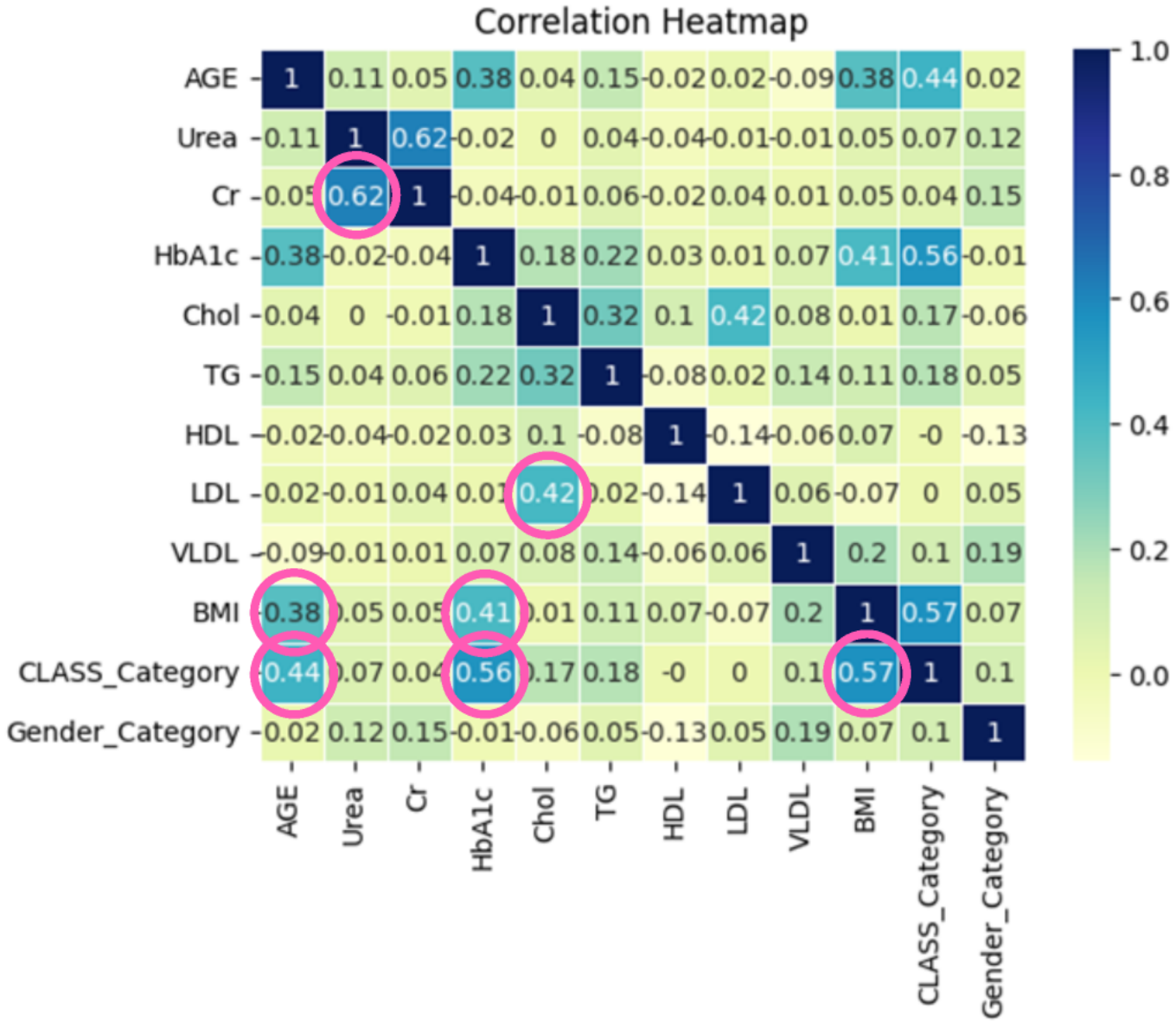
- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.
- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.
- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).

- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.

- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.

- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).


HDL -0.02 -0.04 -0.02 0.03 0.1 -0.08 1 -0.14 -0.06 0.07 -0 -0.13
LDL -0.02 -0.01 0.04 0.01 0.42 0.02 -0.14 1 0.06 -0.07 0 0.05
VLDL -0.09 -0.01 0.01 0.07 0.08 0.14 -0.06 0.06 1 0.2 0.1 0.19
BMI 0.38 0.05 0.05 0.41 0.01 0.11 0.07 -0.07 0.2 1 0.57 0.07
CLASS_Category 0.44 0.07 0.04 0.56 0.17 0.18 -0 0 0.1 0.57 1 0.1
Gender_Category -0.02 0.12 0.15 -0.01 -0.06 0.05 -0.13 0.05 0.19 0.07 0.1 1
AGE
Urea
Cr
HbA1c
Chol
TG
HDL
LDL
VLDL
BMI
CLASS_Category
Gender_Category

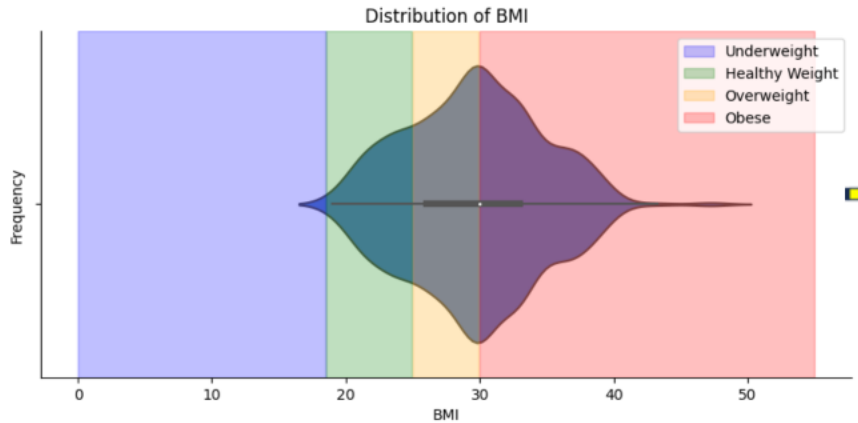


We used a Pearson´s correlation with a heatmap to look at correlations and strength of correlations between our variables.

What we see here, and relevance to analysis:

- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.
- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.
- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).

Let's take a look at the distribution and normal ranges of BMI and HbA1c in a little more detail:




```
count    1000.000000
mean       29.578020
std         4.962388
min        19.000000
25%        26.000000
50%        30.000000
75%        33.000000
max        47.750000
Name: BMI, dtype: float64
```

*We see here how a large percentage of patients (50%) have a BMI of > 30 (obese).

*This makes sense as we know we have 844 patients with Diabetes, 53 Pre-diabetics, and 103 Non-Diabetics in CLASS.

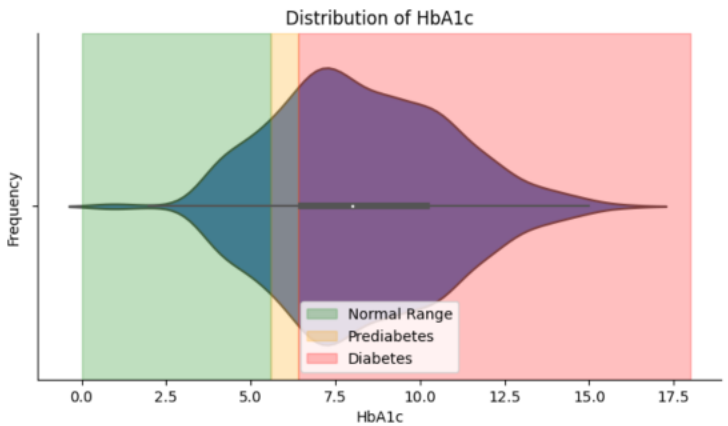
*Increased BMI is a risk factor for diabetes and we have seen the positive correlation between BMI and CLASS_Category.

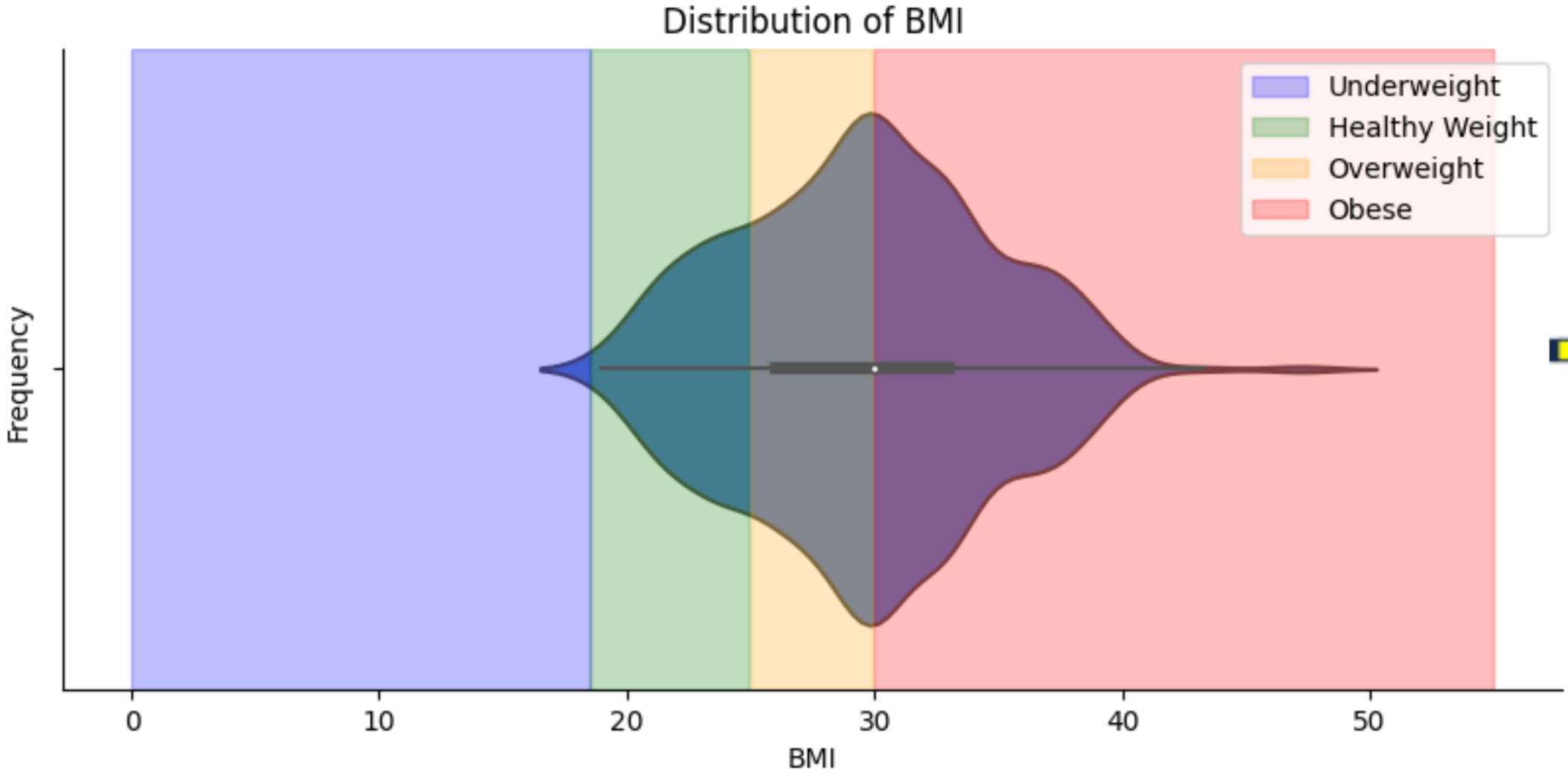
*We see here how 50% of patients have a HbA1c (blood glucose) of > 8 (pre-diabetic range 5.7-6.4% and diabetic over 6.4%).

*This once again makes sense as we know the majority of our sample is made up with patients diagnosed as diabetic.

*In addition, we have also seen the positive correlation between HbA1c and CLASS_Category.

```
count    1000.000000
mean        8.281160
std         2.534003
min         0.900000
25%         6.500000
50%         8.000000
75%        10.200000
max        16.000000
Name: HbA1c, dtype: float64
```






```
count
mean
std
min
25%
50%
75%
max
Name:
```

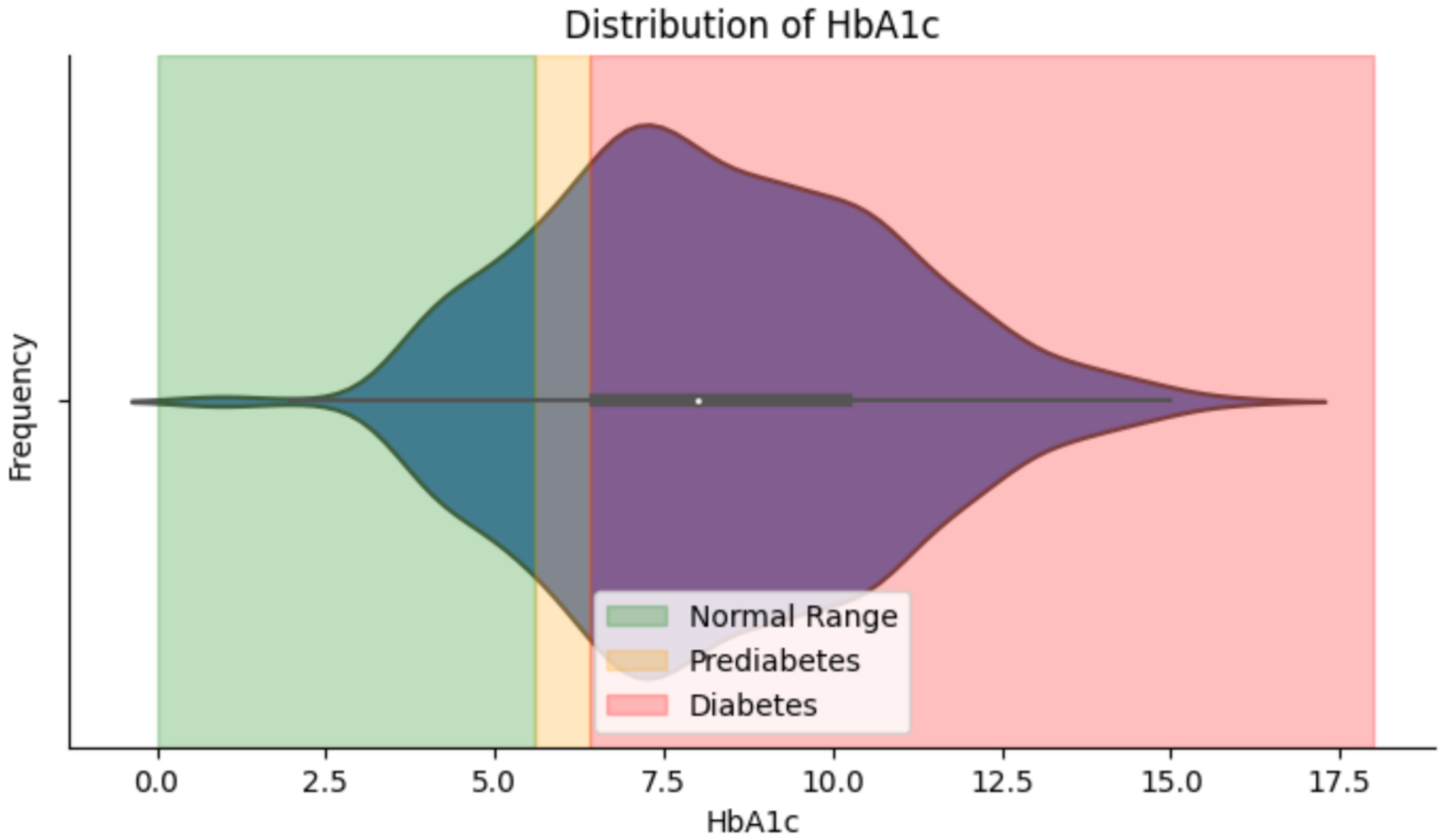
```
count    1000.000000
mean       29.578020
std         4.962388
min        19.000000
25%        26.000000
50%        30.000000
75%        33.000000
max        47.750000
Name: BMI, dtype: float64
```

*We see here how a large percentage of patients (50%) have a BMI of > 30 (obese).

*This makes sense as we know we have 844 patients with Diabetes, 53 Pre-diabetics, and 103 Non-Diabetics in CLASS.

*Increased BMI is a risk factor for diabetes and we have seen the positive correlation between BMI and CLASS_Category.

000000
281160
534003
900000
500000
000000
200000
000000
type: float64



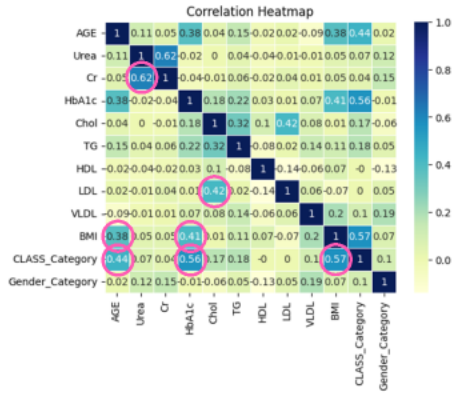
*We see here how 50% of patients have a HbA1c (blood glucose) of > 8 (pre-diabetic range 5.7-6.4% and diabetic over 6.4%).

*This once again makes sense as we know the majority of our sample is made up with patients diagnosed as diabetic.

*In addition, we have also seen the positive correlation between HbA1c and CLASS_Category.

```
count    1000.000000
mean        8.281160
std         2.534003
min         0.900000
25%         6.500000
50%         8.000000
75%        10.200000
max        16.000000
Name: HbA1c, dtype: float6
```

3- EXPLORATORY DATA ANALYSIS (EDA)

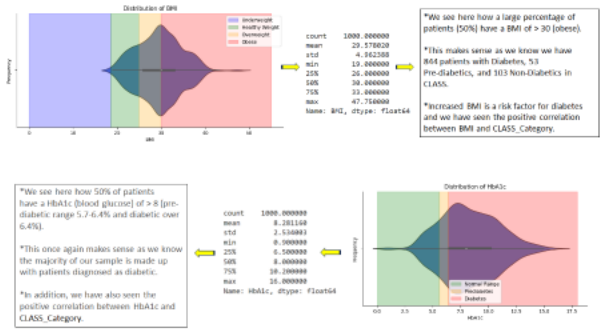


We used a Pearson´s correlation with a heatmap to look at correlations and strength of correlations between our variables.

What we see here, and relevance to analysis:

- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.
- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.
- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).

Let's take a look at the distribution and normal ranges of BMI and HbA1c in a little more detail:



Conclusions from EDA...

Interactive Plots With Diabetes Dataset

Scatterplot

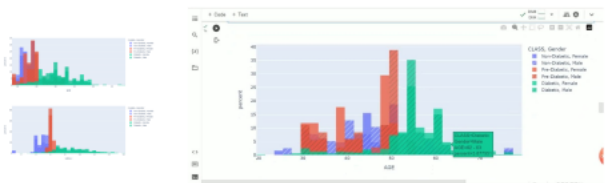
Histogram

Catplot

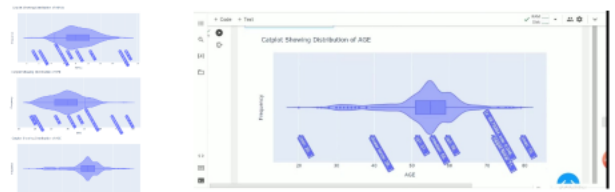
Interactive Plots With Diabetes Dataset

Scatterplot

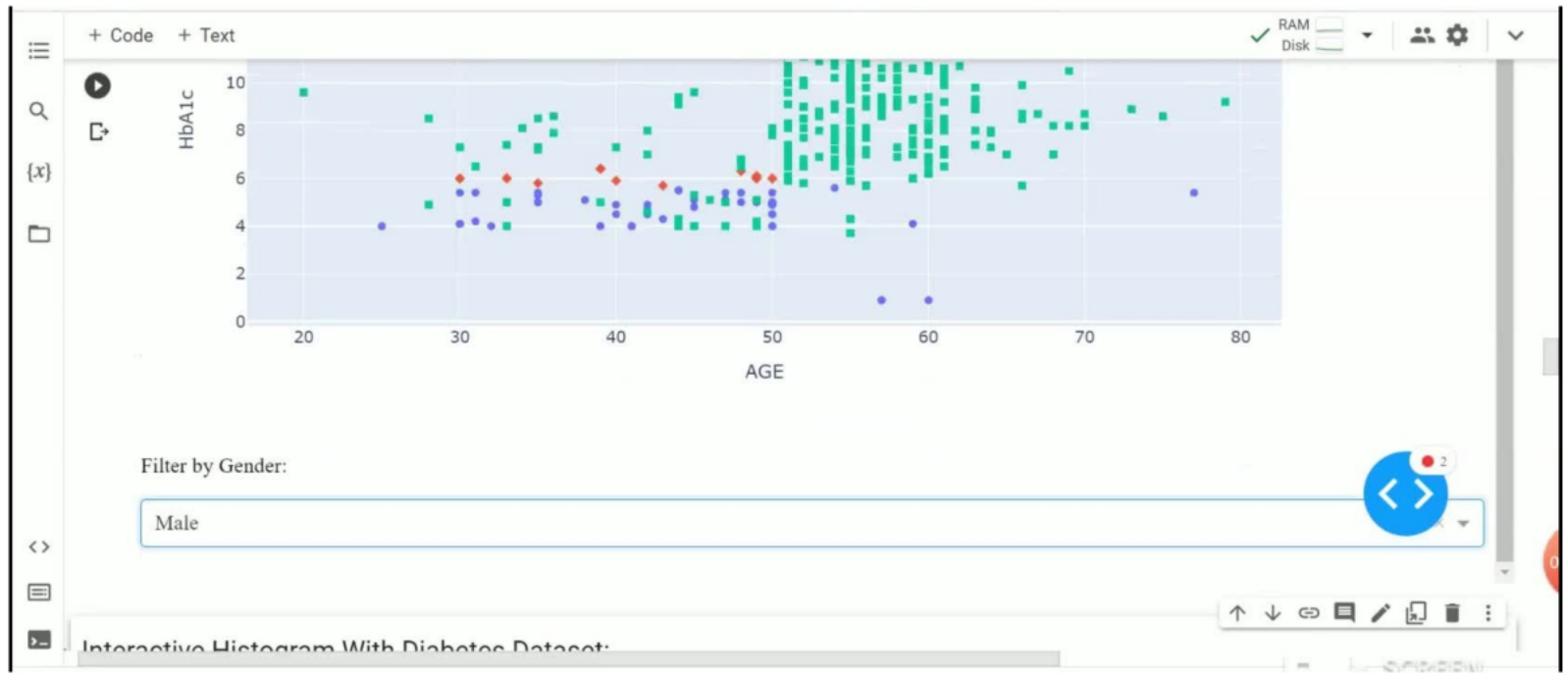
Histogram

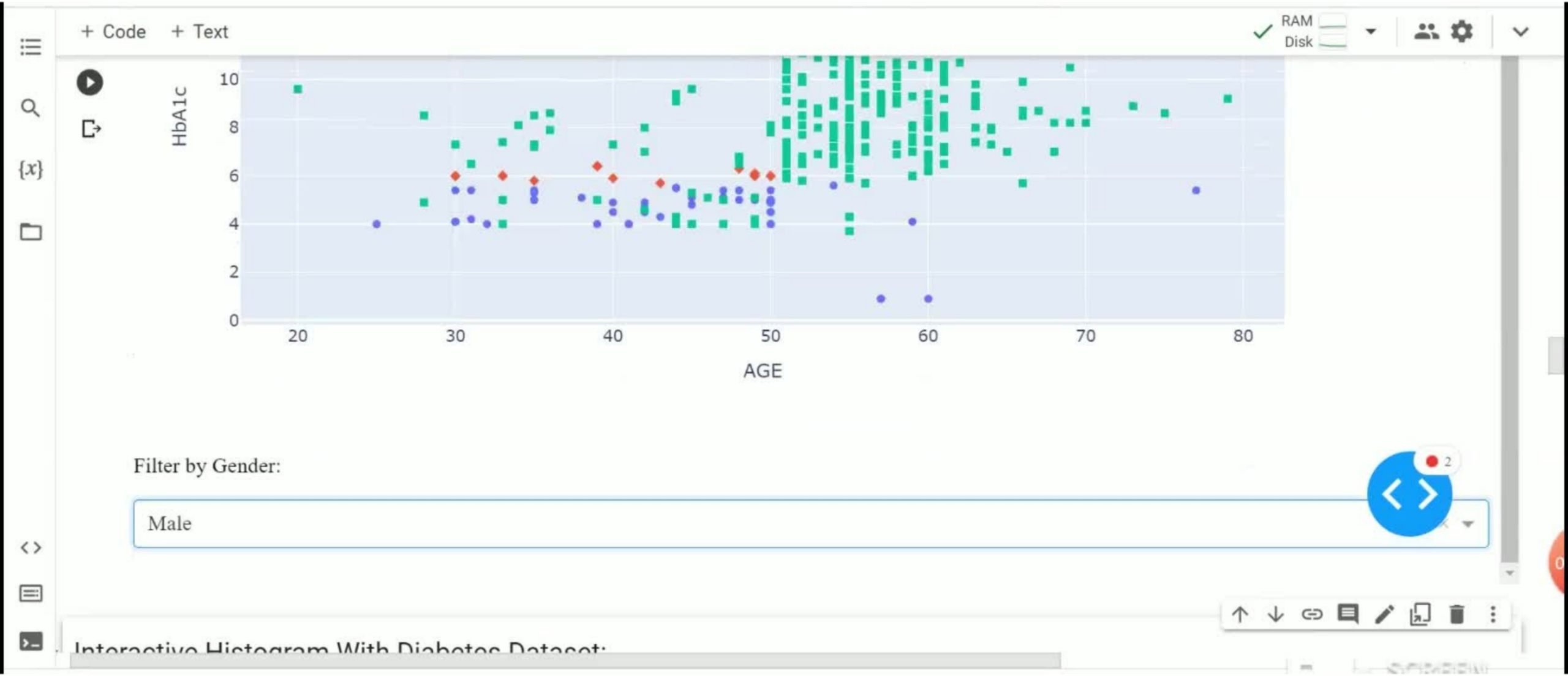


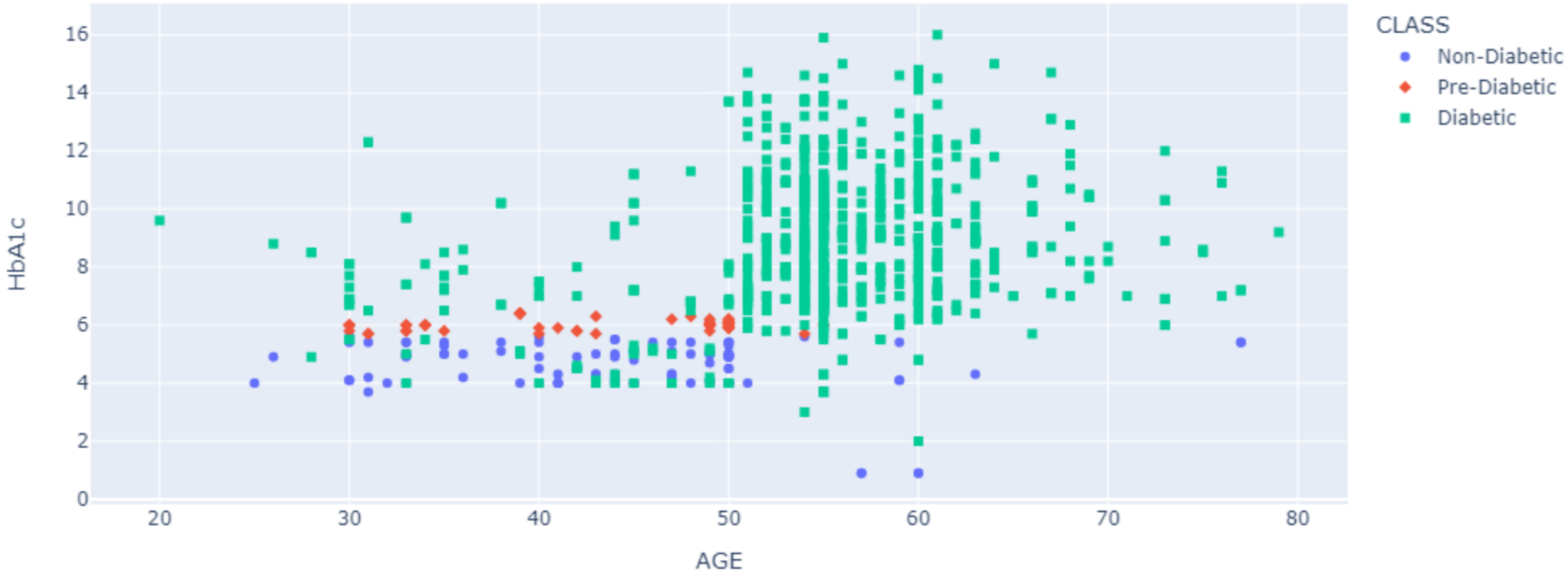
Catplot

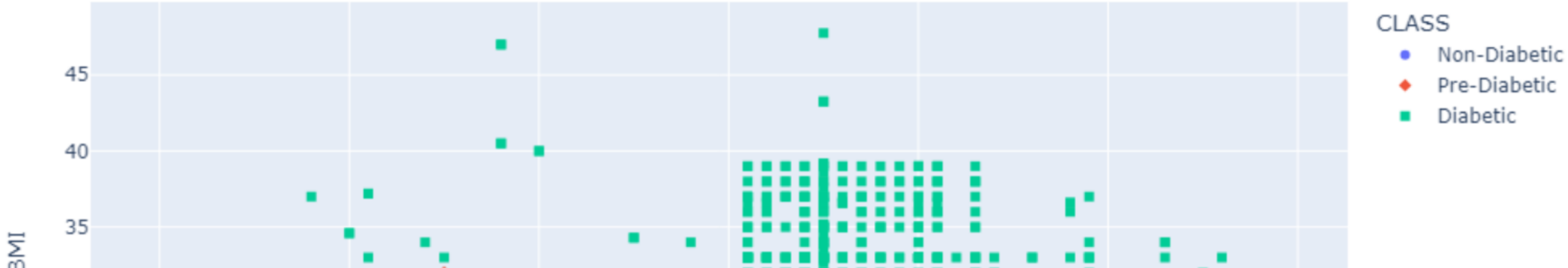


Scatterplot

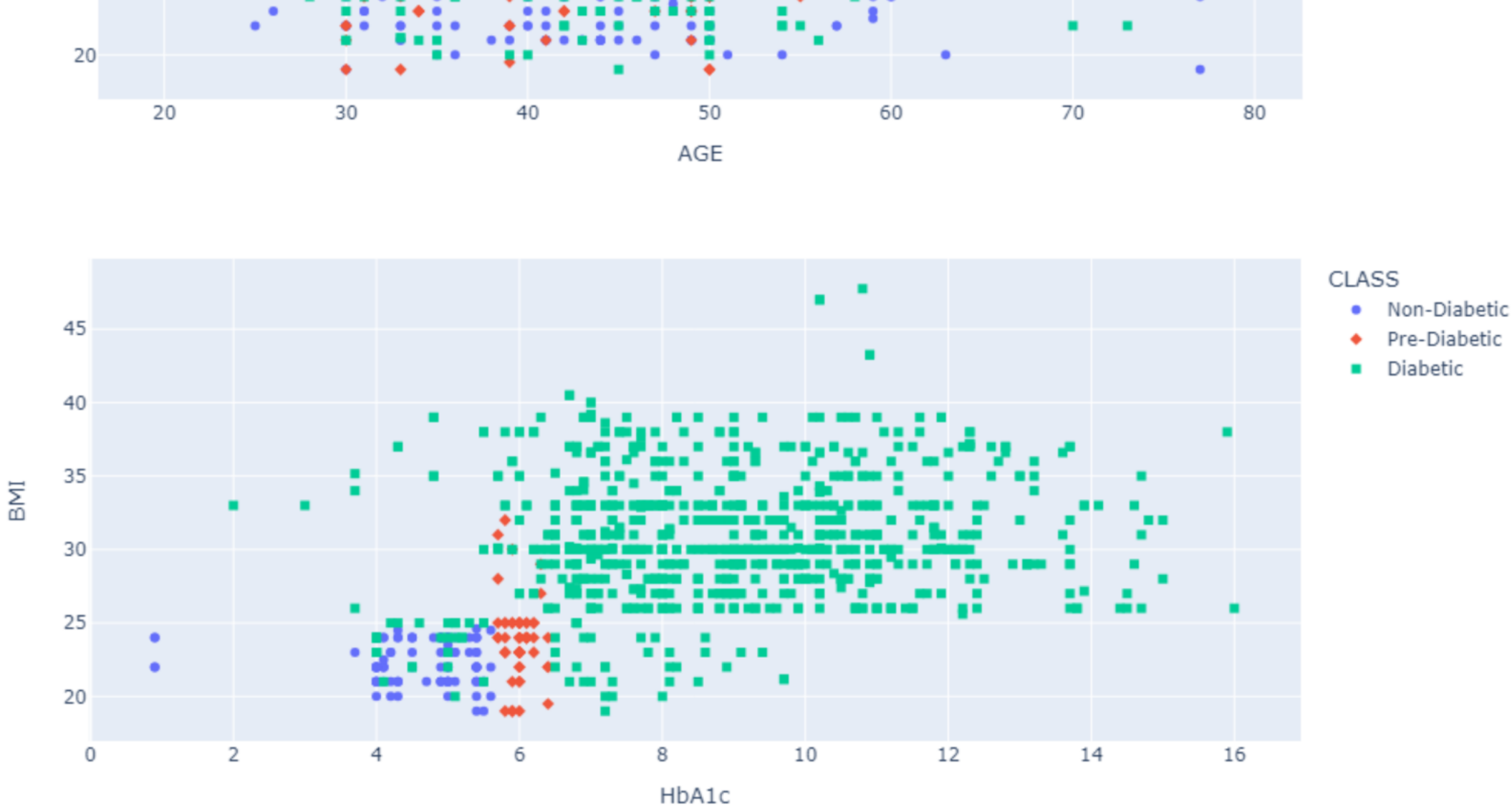



+ Code
+ Text
RAM
Disk
HbA1c
10
8
6
4
2
0
20
30
40
50
60
70
80
AGE
Filter by Gender:
Male
Interactive Histogram With Diabetes Dataset:


HbA1c
AGE
CLASS
Non-Diabetic
Pre-Diabetic
Diabetic


BMI
CLASS
Non-Diabetic
Pre-Diabetic
Diabetic

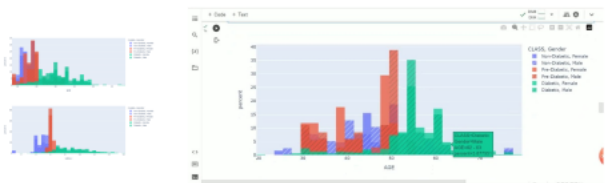

HbA
0
2
4
6
8
20
30
40
50
60
70
80
AGE
BMI
20
25
30
35
40
45
20
30
40
50
60
70
80
AGE
CLASS
Non-Diabetic
Pre-Diabetic
Diabetic


20
20
30
40
50
60
70
80
AGE
BMI
45
40
35
30
25
20
0
2
4
6
8
10
12
14
16
HbA1c
CLASS
Non-Diabetic
Pre-Diabetic
Diabetic

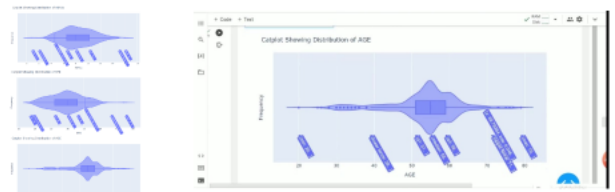
Interactive Plots With Diabetes Dataset

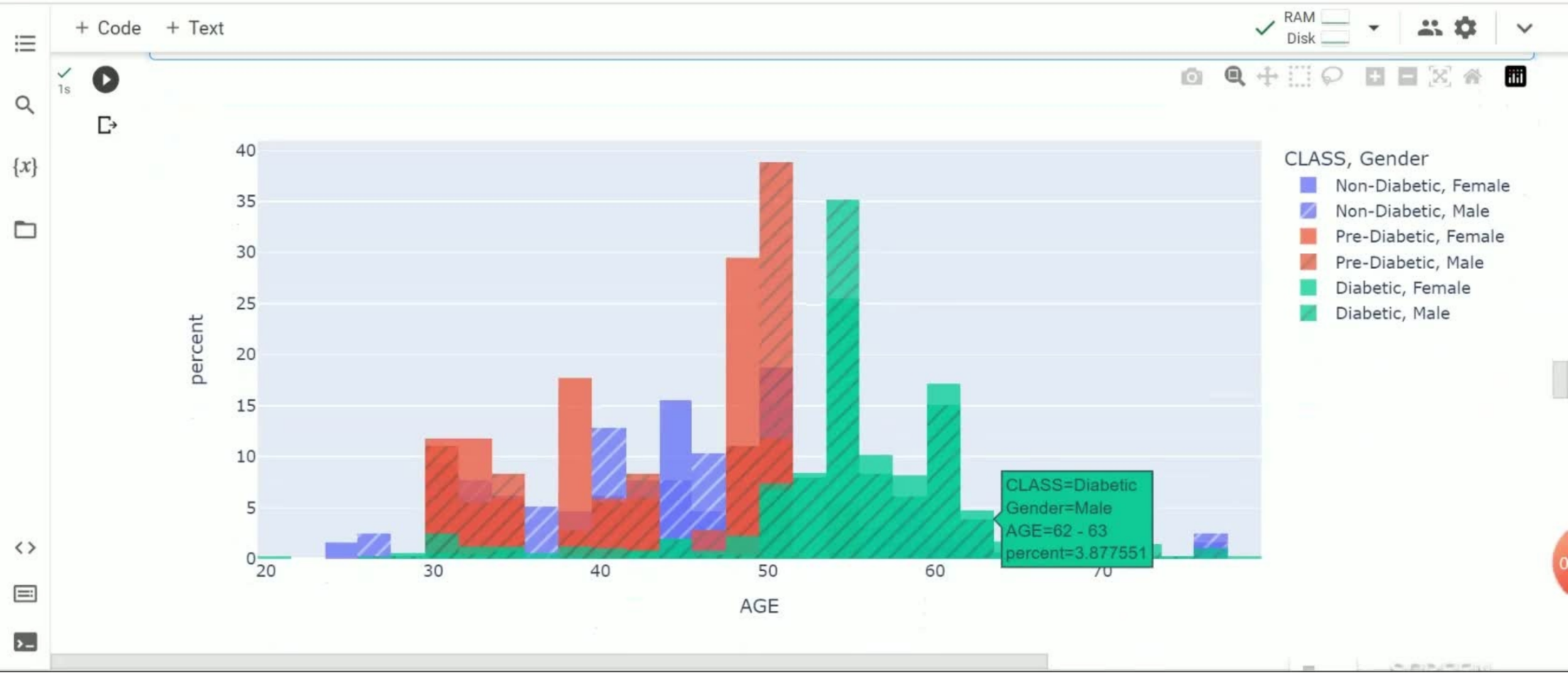
Scatterplot

Histogram

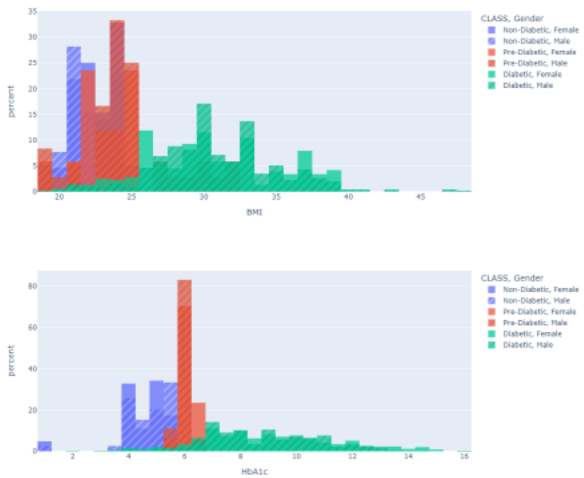


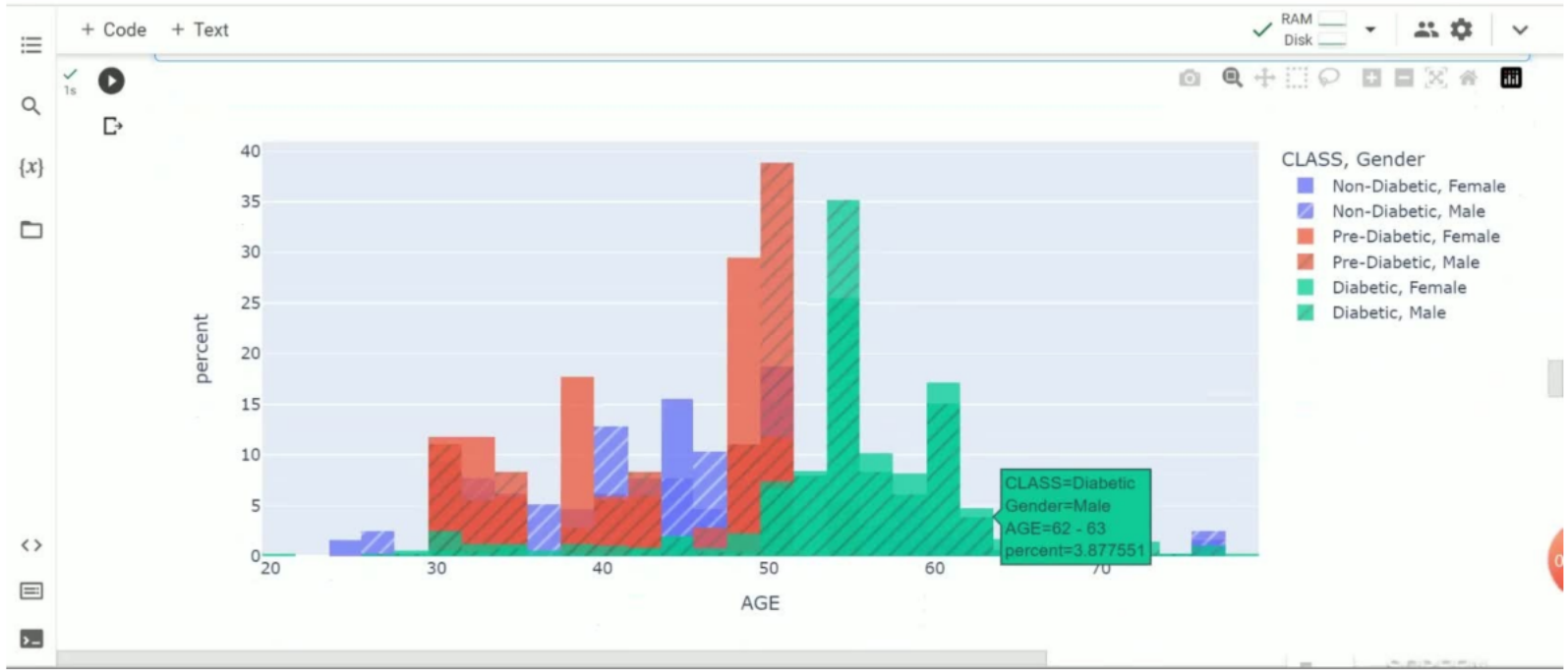
Catplot

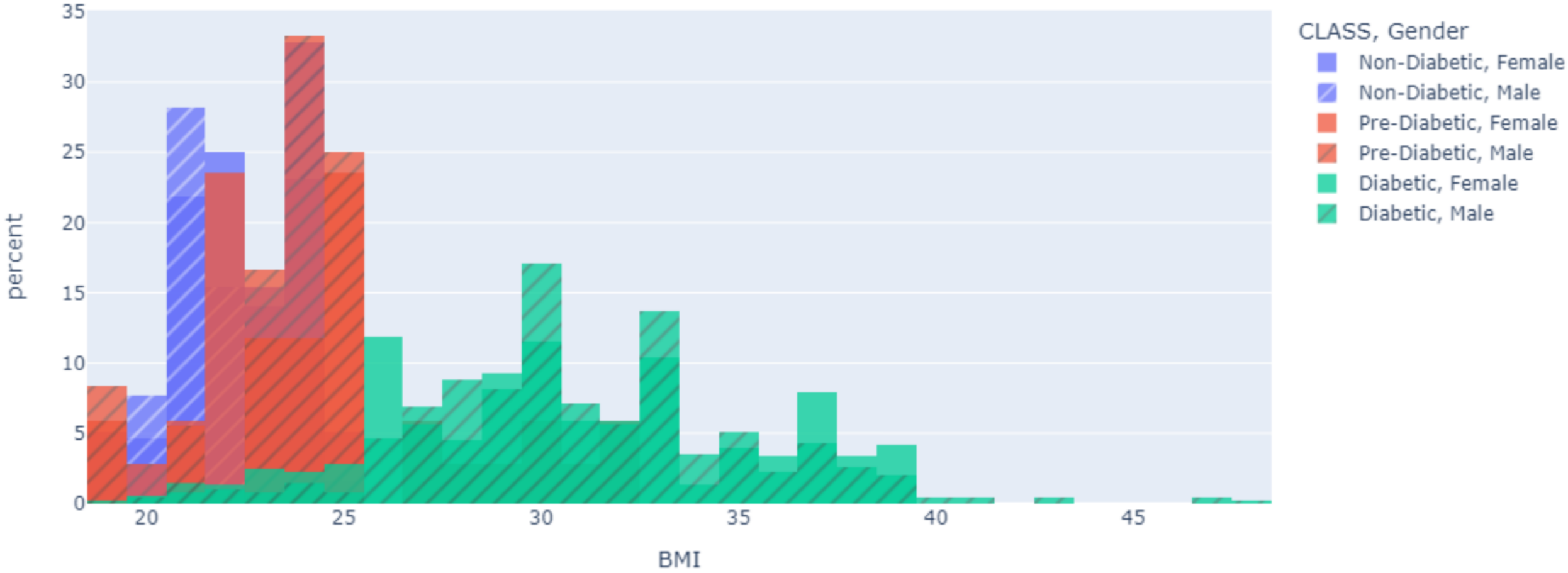


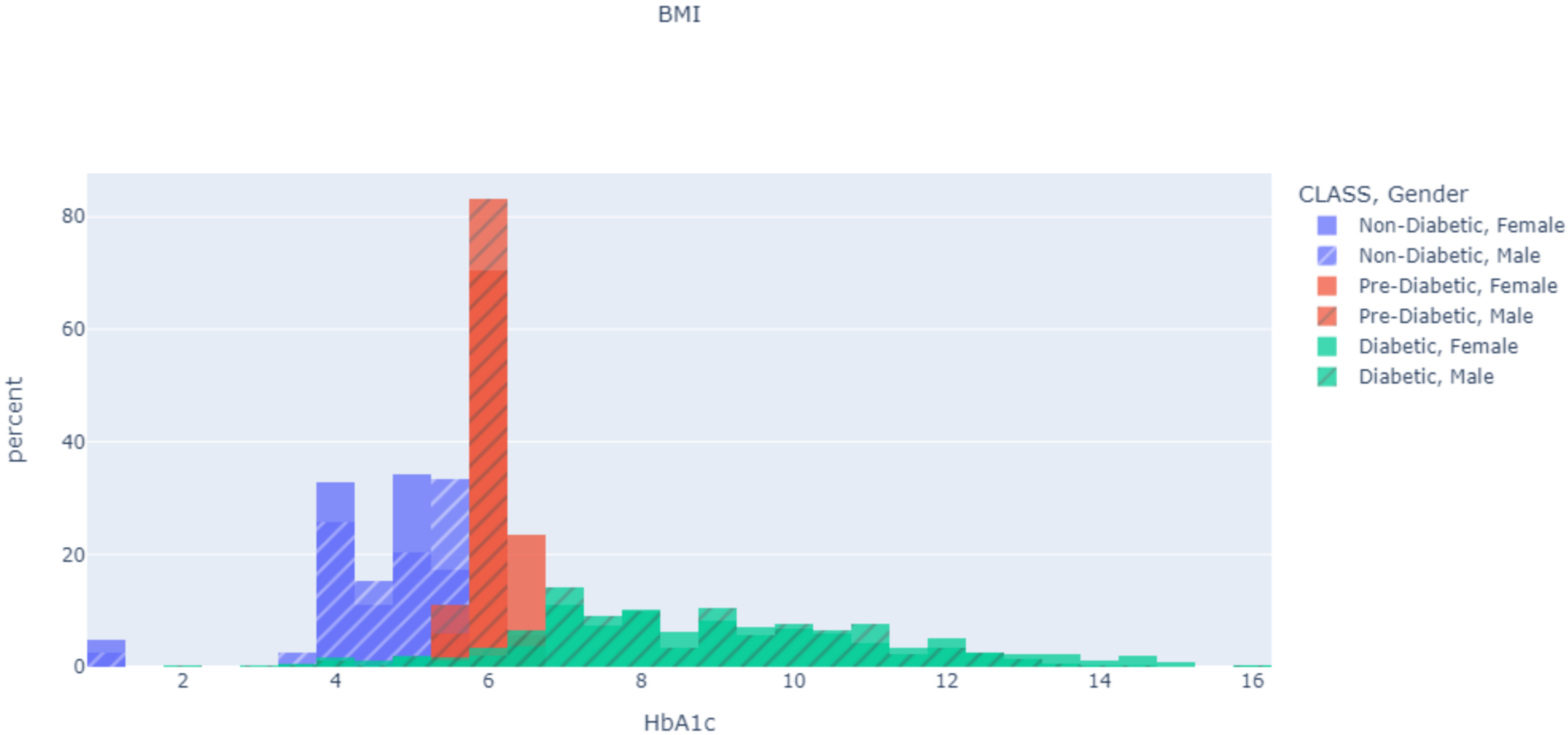

+ Code
+ Text
RAM
Disk
1s
CLASS, Gender
Non-Diabetic, Female
Non-Diabetic, Male
Pre-Diabetic, Female
Pre-Diabetic, Male
Diabetic, Female
Diabetic, Male
CLASS=Diabetic
Gender=Male
AGE=62 - 63
percent=3.877551
percent
AGE
0
5
10
15
20
25
30
35
40
20
30
40
50
60
70

Histogram


CLASS, Gender
Non-Diabetic, Female
Non-Diabetic, Male
Pre-Diabetic, Female
Pre-Diabetic, Male
Diabetic, Female
Diabetic, Male
percent
BMI
HbA1c


+ Code
+ Text
RAM
Disk
CLASS, Gender
Non-Diabetic, Female
Non-Diabetic, Male
Pre-Diabetic, Female
Pre-Diabetic, Male
Diabetic, Female
Diabetic, Male
percent
AGE
CLASS=Diabetic
Gender=Male
AGE=62 - 63
percent=3.877551

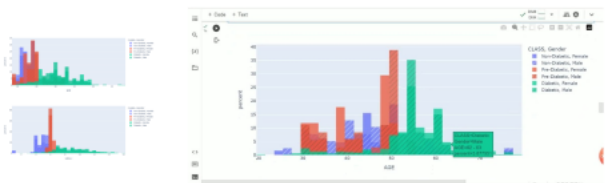

CLASS, Gender
Non-Diabetic, Female
Non-Diabetic, Male
Pre-Diabetic, Female
Pre-Diabetic, Male
Diabetic, Female
Diabetic, Male
percent
0
5
10
15
20
25
30
35
20
25
30
35
40
45
BMI


BMI
percent
0
20
40
60
80
2
4
6
8
10
12
14
16
HbA1c
CLASS, Gender
Non-Diabetic, Female
Non-Diabetic, Male
Pre-Diabetic, Female
Pre-Diabetic, Male
Diabetic, Female
Diabetic, Male

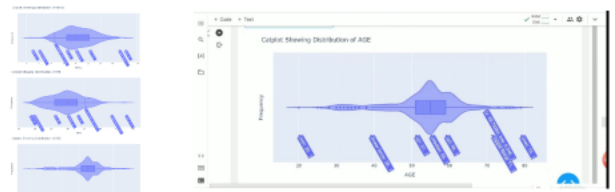
Interactive Plots With Diabetes Dataset

Scatterplot

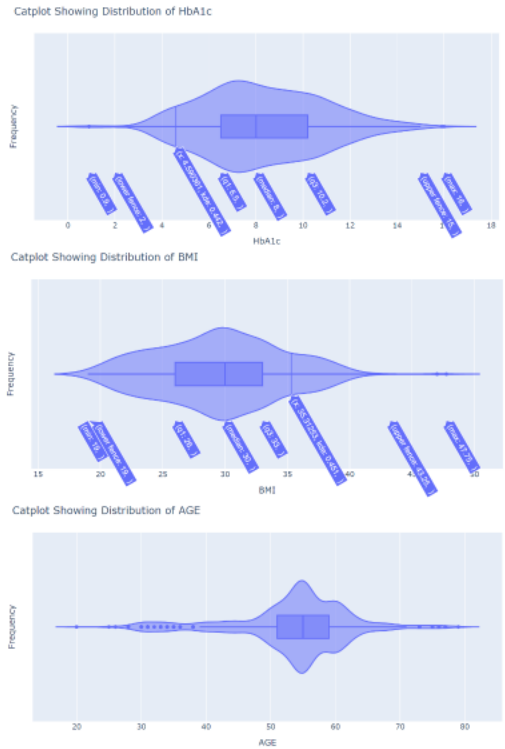
Histogram



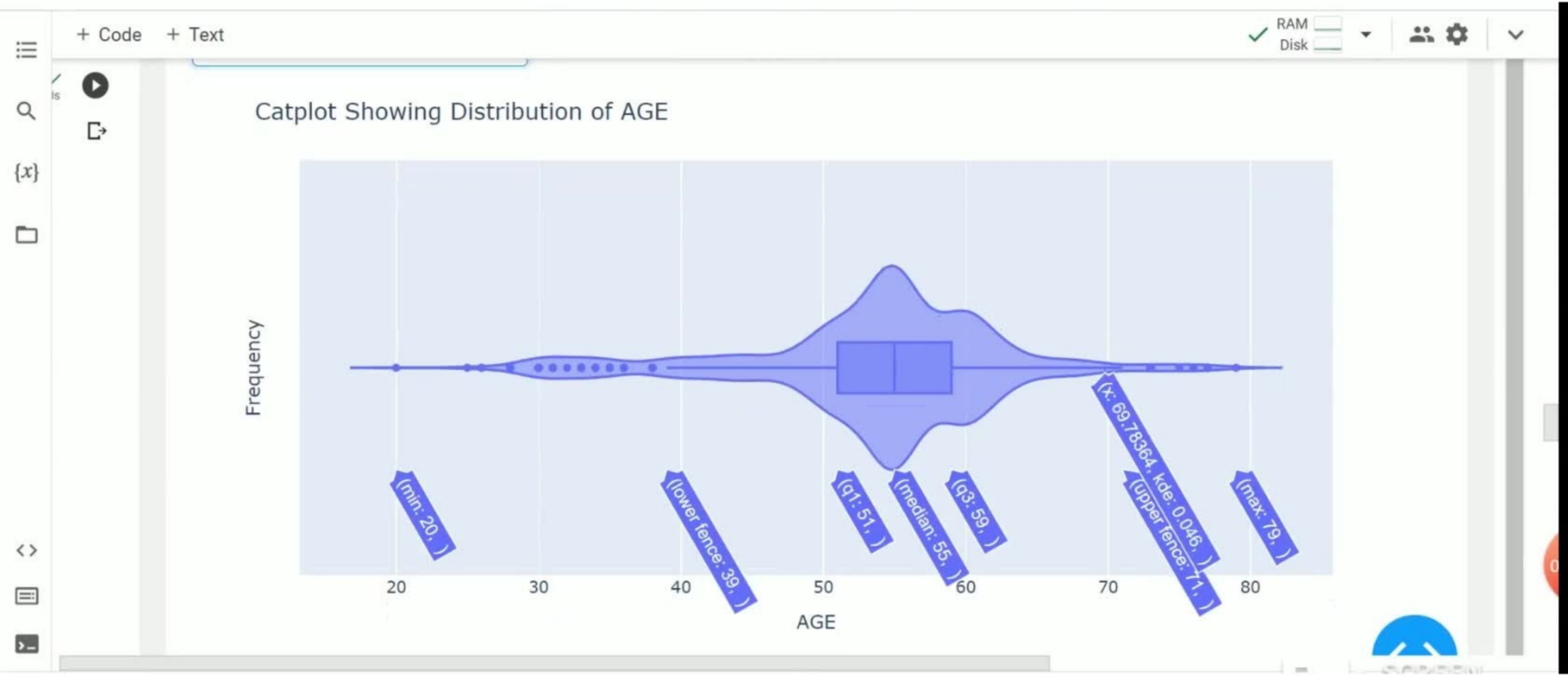
Catplot



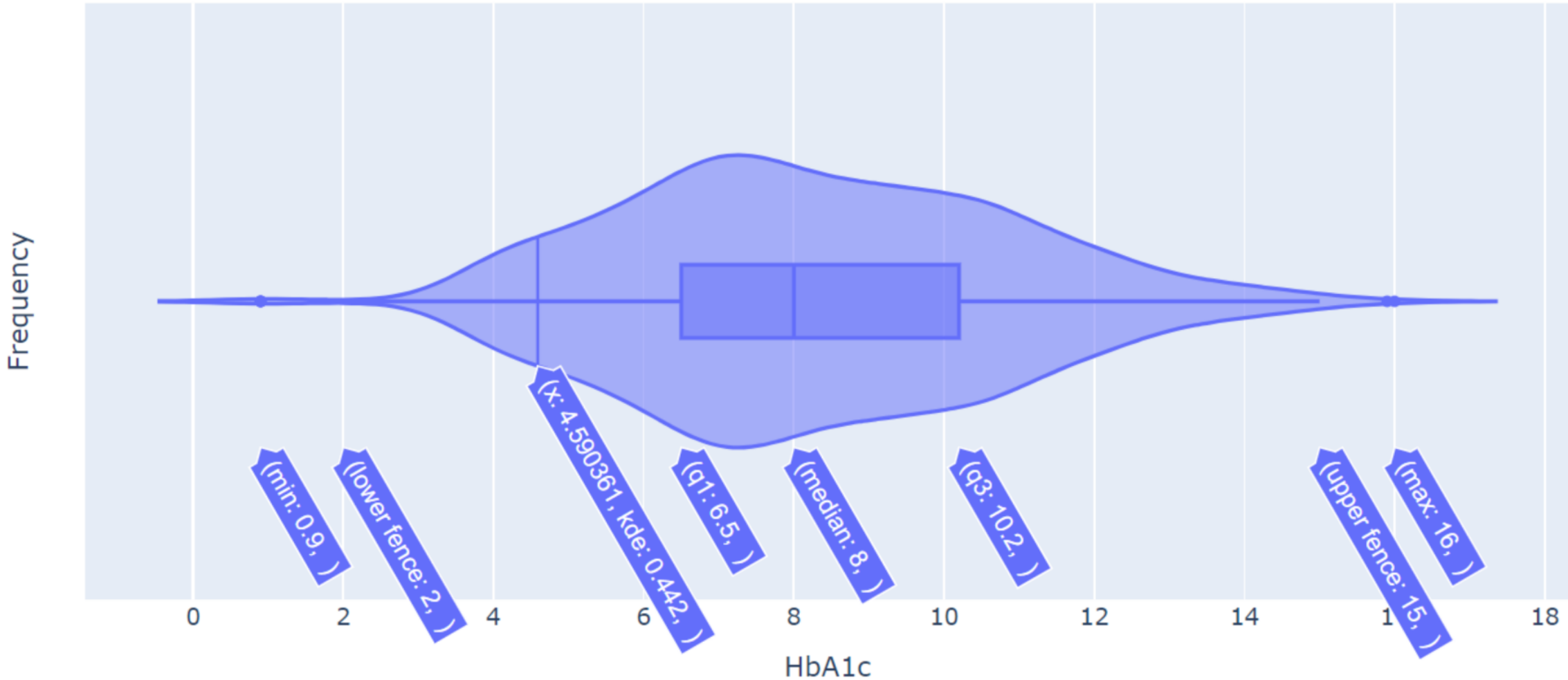
Catplot


Catplot Showing Distribution of HbA1c
Frequency
HbA1c
Catplot Showing Distribution of BMI
Frequency
BMI
Catplot Showing Distribution of AGE
Frequency
AGE

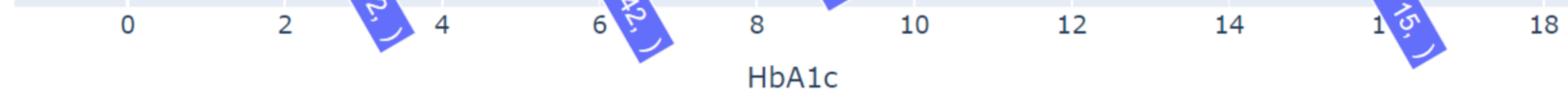

+ Code
+ Text
RAM
Disk
Catplot Showing Distribution of AGE
Frequency
(min: 20,)
(lower fence: 39,)
(q1: 51,)
(median: 55,)
(q3: 59,)
(x: 69.783364, kde: 0.046,)
(upper fence: 71,)
(max: 79,)
20
30
40
50
60
70
80
AGE


+ Code
+ Text
RAM
Disk
Catplot Showing Distribution of AGE
Frequency
(min: 20,)
(lower fence: 39,)
(q1: 51,)
(median: 55,)
(q3: 59,)
(x: 69.78364, kde: 0.046,)
(upper fence: 71,)
(max: 79,)
20
30
40
50
60
70
80
AGE

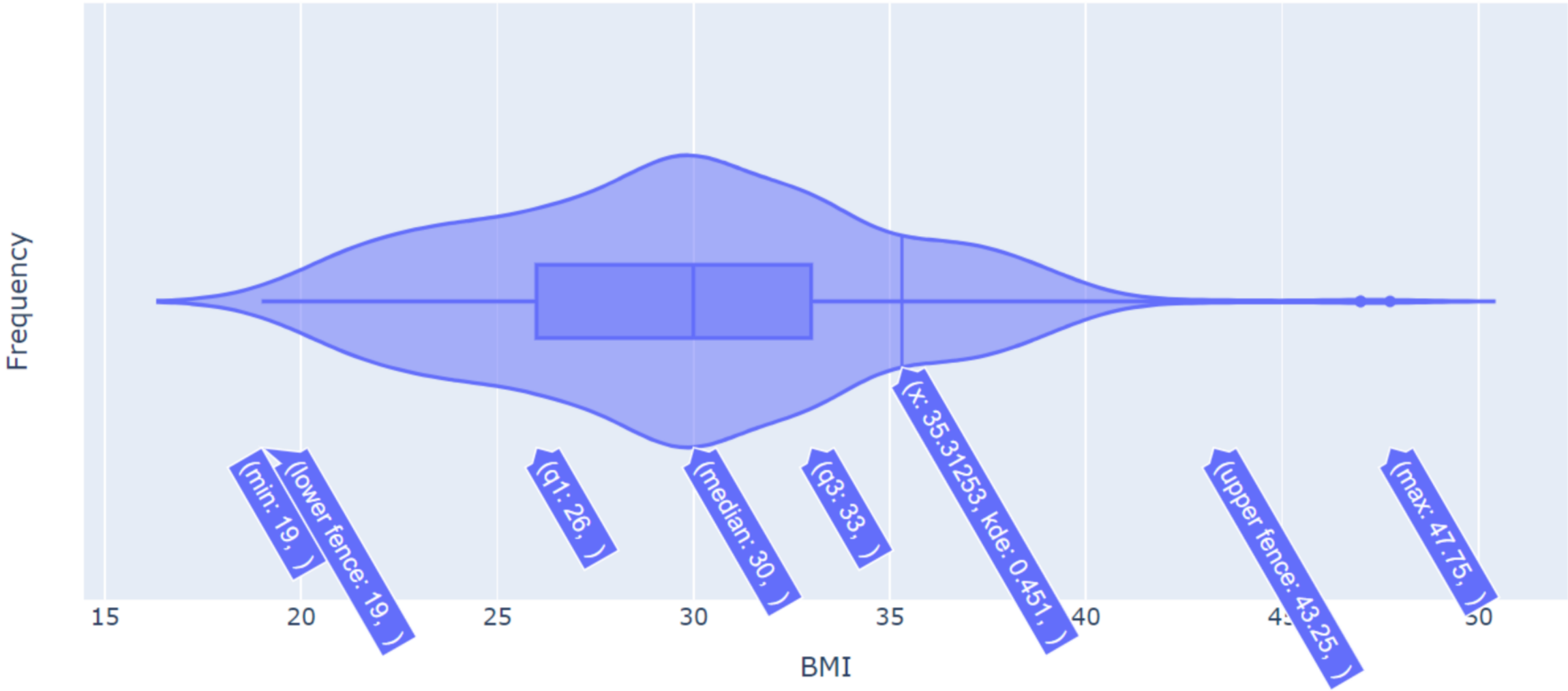
Catplot Showing Distribution of HbA1c



Catplot Showing Distribution of BMI

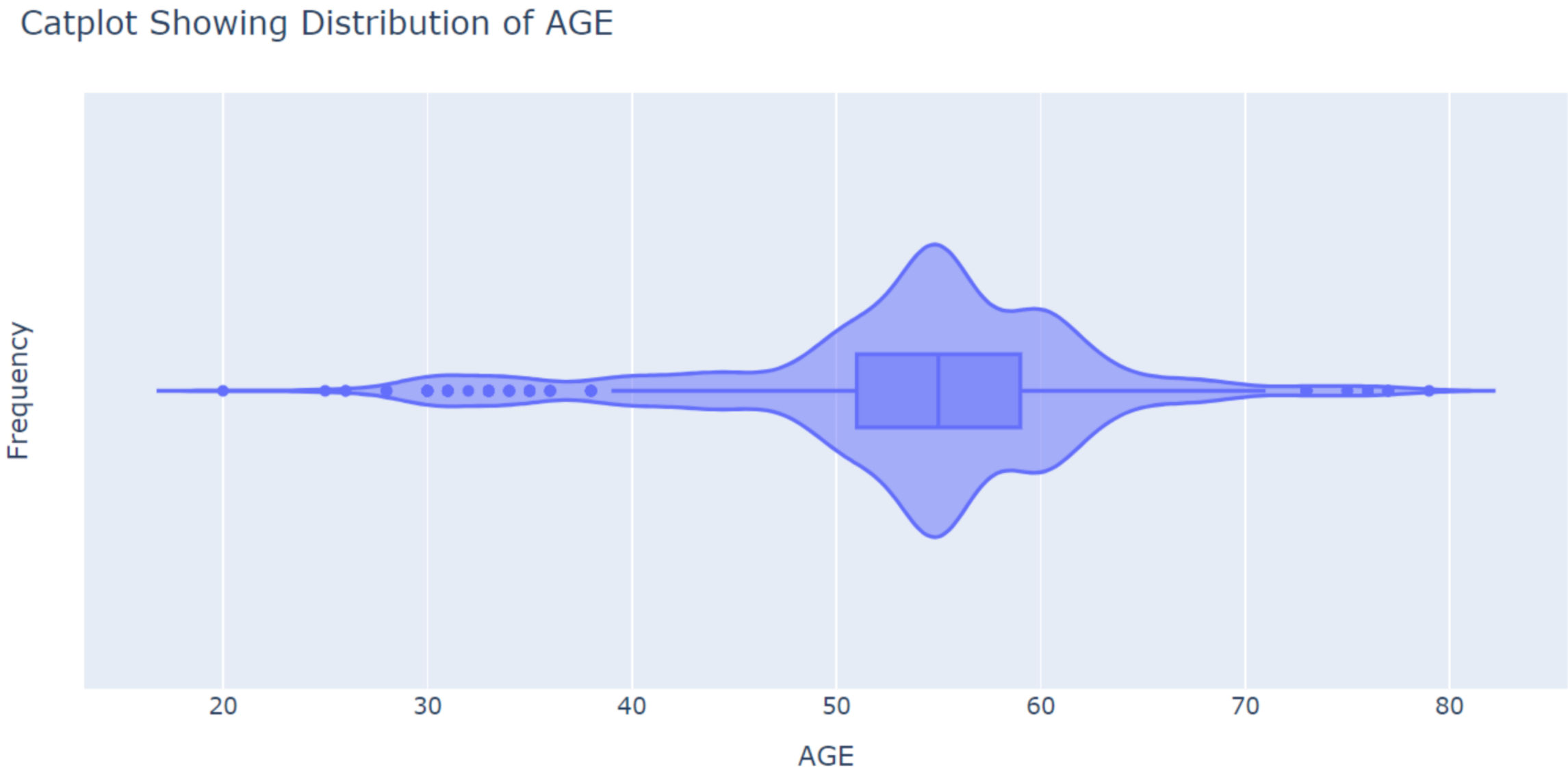


Catplot Showing Distribution of BMI

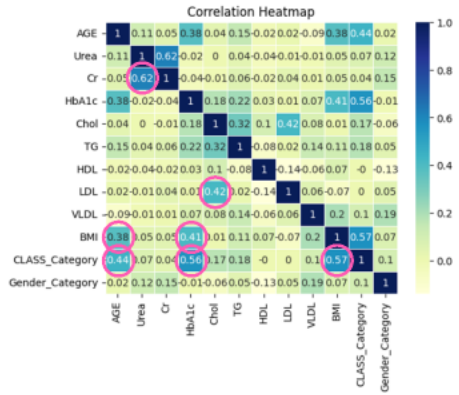


Catplot Showing Distribution of AGE


15
20
25
30
35
40
45
50
BMI


Catplot Showing Distribution of AGE
Frequency
20
30
40
50
60
70
80
AGE

3- EXPLORATORY DATA ANALYSIS (EDA)

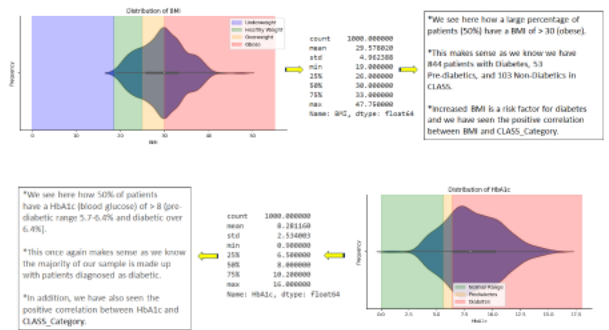


We used a Pearson´s correlation with a heatmap to look at correlations and strength of correlations between our variables.

What we see here, and relevance to analysis:

- **BMI** and **CLASS_Category** showed a moderate positive correlation (**0.57**) – this is not unexpected due to the risk of increased BMI and diabetes.
- **HbA1c** and **CLASS_Category** showed a moderate positive correlation (**0.56**) – this too is not unexpected because of high blood sugar characteristic in diabetes patients.
- **Age** and **CLASS_Category** showed a moderate positive correlation (**0.44**).

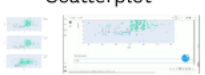
Let's take a look at the distribution and normal ranges of BMI and HbA1c in a little more detail:



Conclusions from EDA...

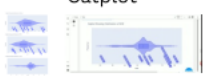
Interactive Plots With Diabetes Dataset

Scatterplot



Histogram

Catplot



Conclusions from EDA...

Conclusions from EDA...

* Results using Pearson´s correlation suggest that **BMI**, **HbA1c**, and **age** are moderately positively correlated with CLASS_Category i.e. **an increase in these variables correlates with an increased risk of diabetes**. This was not unexpected as these variables are all known risk factors for diabetes.

Conclusions from EDA...

* Results using Pearson´s correlation suggest that **BMI**, **HbA1c**, and **age** are moderately positively correlated with CLASS_Category i.e. **an increase in these variables correlates with an increased risk of diabetes**. This was not unexpected as these variables are all known risk factors for diabetes.

* Interestingly no correlation was seen between gender and CLASS_Category – suggesting that **males and females are equally as likely to develop diabetes** (literature generally shows type 2 diabetes to be more prevalent in men).

Conclusions from EDA...

* Results using Pearson´s correlation suggest that **BMI**, **HbA1c**, and **age** are moderately positively correlated with CLASS_Category i.e. **an increase in these variables correlates with an increased risk of diabetes**. This was not unexpected as these variables are all known risk factors for diabetes.

* Interestingly no correlation was seen between gender and CLASS_Category – suggesting that **males and females are equally as likely to develop diabetes** (literature generally shows type 2 diabetes to be more prevalent in men).

* The dataset has a skewed distribution within the **BMI**, **HbA1c** and **age** variables, with **a large percentage of patients falling into the 'obese', 'diabetic glucose', and 'older' (median 55 years) range,** respectively. This, however, was not unexpected as most of the sample were diagnosed diabetes patients.

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

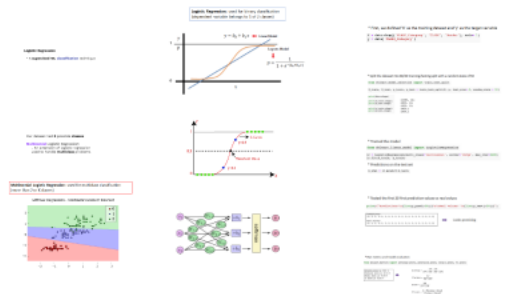
Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

4- MACHINE LEARNING & DEEP LEARNING

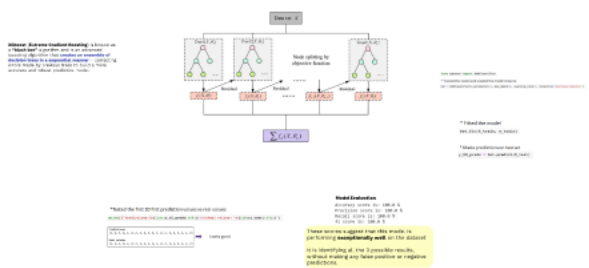
4.1 Logistic Regression - Classification



Machine learning (ML)
- is a subfield of **artificial intelligence** that focuses on developing and training algorithms to **learn patterns** and **make decisions** from data without being explicitly trained to do so.
- The aim is to ultimately develop models that can make **accurate predictions or decisions on new unseen data.**

4.1 Logistic Regression - Classification
4.2 - Normalization of the data - MinMaxScaler

4.3 XGboost



4.4 NEURAL NETWORKS

Machine learning (ML)

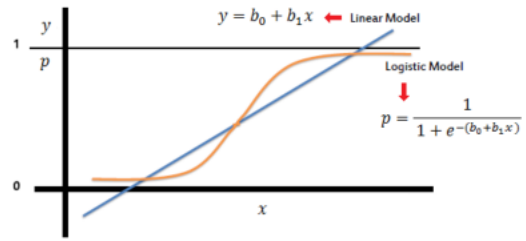
- is a subfield of **artificial intelligence** that focuses on developing and training algorithms to **learn patterns** and **make decisions** from data without being explicitly trained to do so.
- The aim is to ultimately develop models that can make **accurate predictions or decisions on new unseen data.**

4.1 Logistic Regression - Classification

Logistic Regression

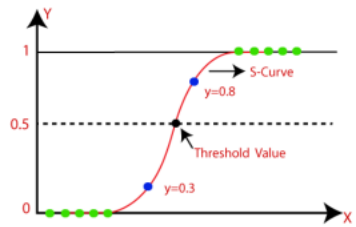
- A **supervised ML classification** technique

Logistic Regression: used for binary classification (dependent variable belongs to 1 of 2 classes)



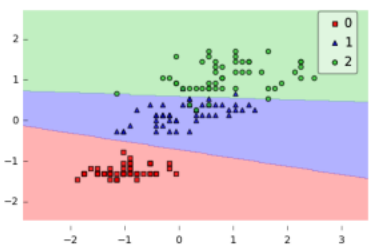
Our dataset had **3** possible **classes**

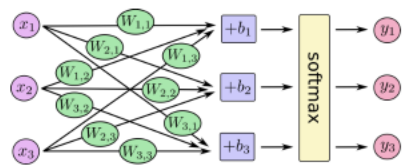
Multinomial Logistic Regression:
- An extension of logistic regression used to handle **multiclass** problems.



Multinomial Logistic Regression: used for multiclass classification (more than 2 or K classes)

Softmax Regression - Stochastic Gradient Descent





* First, we defined 'X' as the training dataset and 'y' as the target variable

```
X = data.drop(['CLASS_Category', 'CLASS', 'Gender'], axis= 1)
y = data['CLASS_Category']
```

* Split the dataset into 80/20 training/testing split with a random state of 50

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state = 50)

print(data.shape)

print(X_train.shape)
print(X_test.shape)

print(y_train.shape)
print(y_test.shape)
```

```
(1000, 14)
(800, 11)
(200, 11)
(800,)
(200,)
```

* Trained the model

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=2600)
lr.fit(X_train, y_train)
```

* Predictions on the test set

```
y_pred = lr.predict(X_test)
```

* Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_pred[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

➡ Looks promising

* Ran metrics and model evaluation

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
Accuracy score is: 93.5 %
Precision score is: 67.51 %
Recall score is: 66.84 %
f1 score is: 65.63 %
```

➡

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Logistic Regression

- A **supervised ML classification** technique

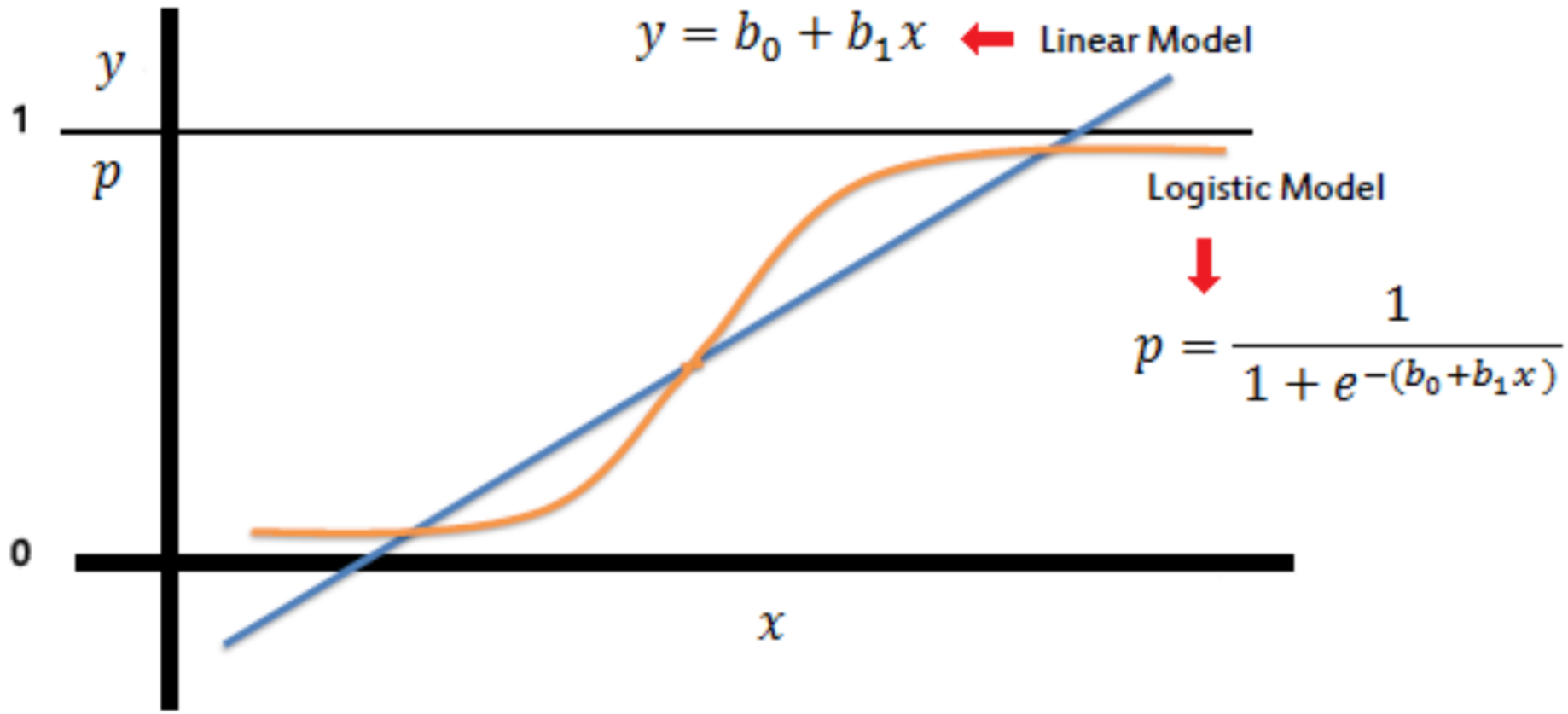
Logistic Regression

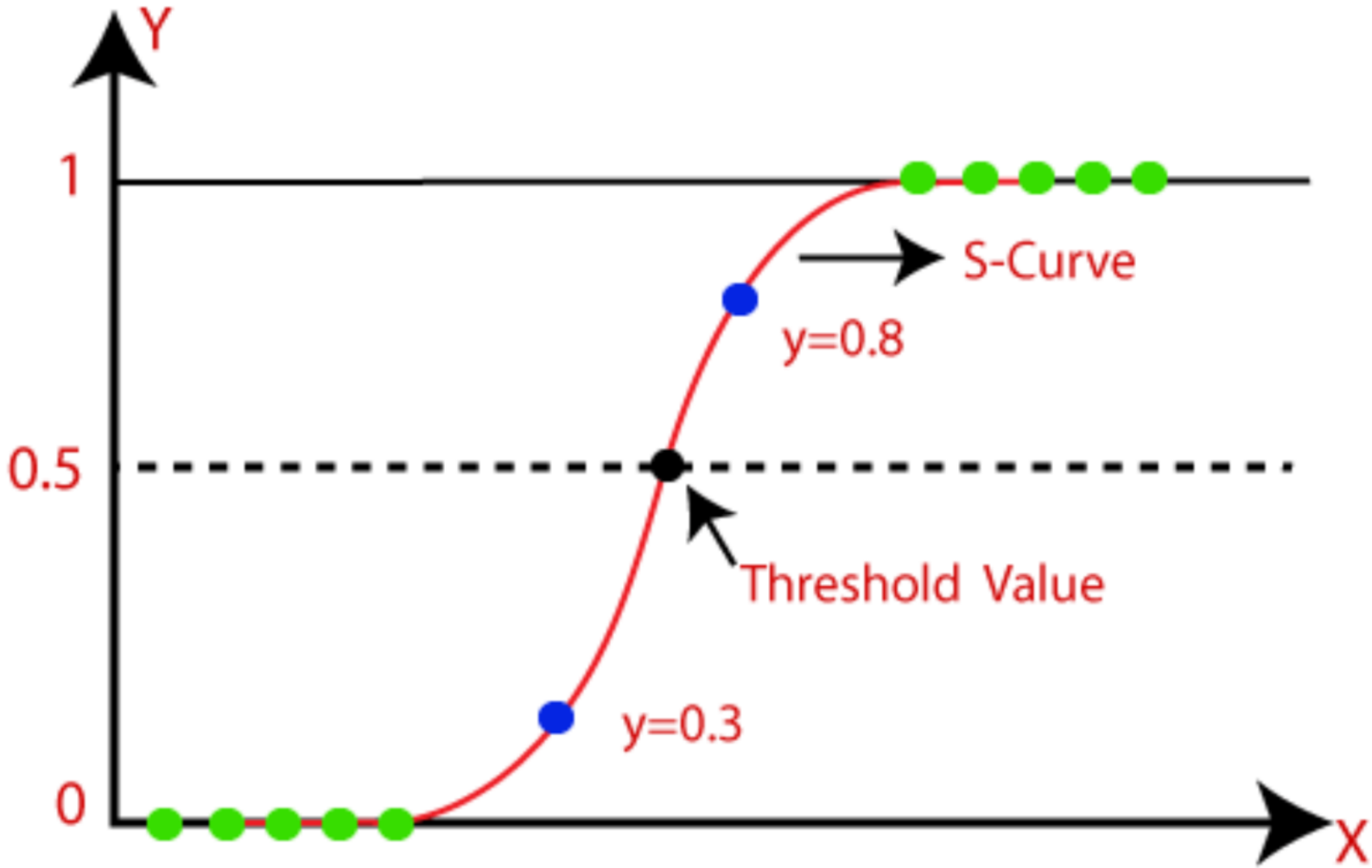
- A **supervised ML classification** technique
- Takes a set of predictor variables (**independent variables**)

Logistic Regression

- A **supervised ML classification** technique
- Takes a set of predictor variables (**independent variables**)
- Predicts the **probability** of an observation belonging to a certain class or category (**dependent variables**)

Logistic Regression: used for binary classification (dependent variable belongs to 1 of 2 classes)



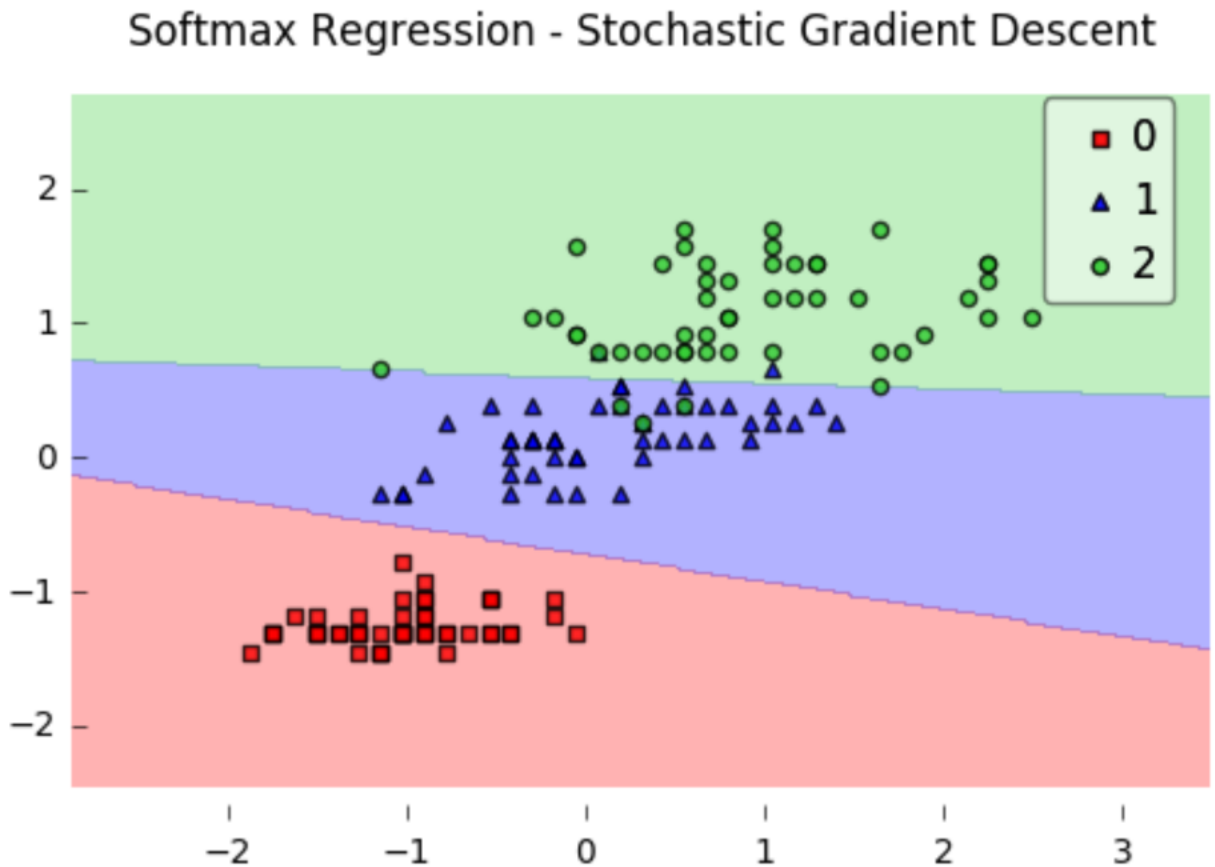

Y
1
0.5
0
S-Curve
y=0.8
Threshold Value
y=0.3
X

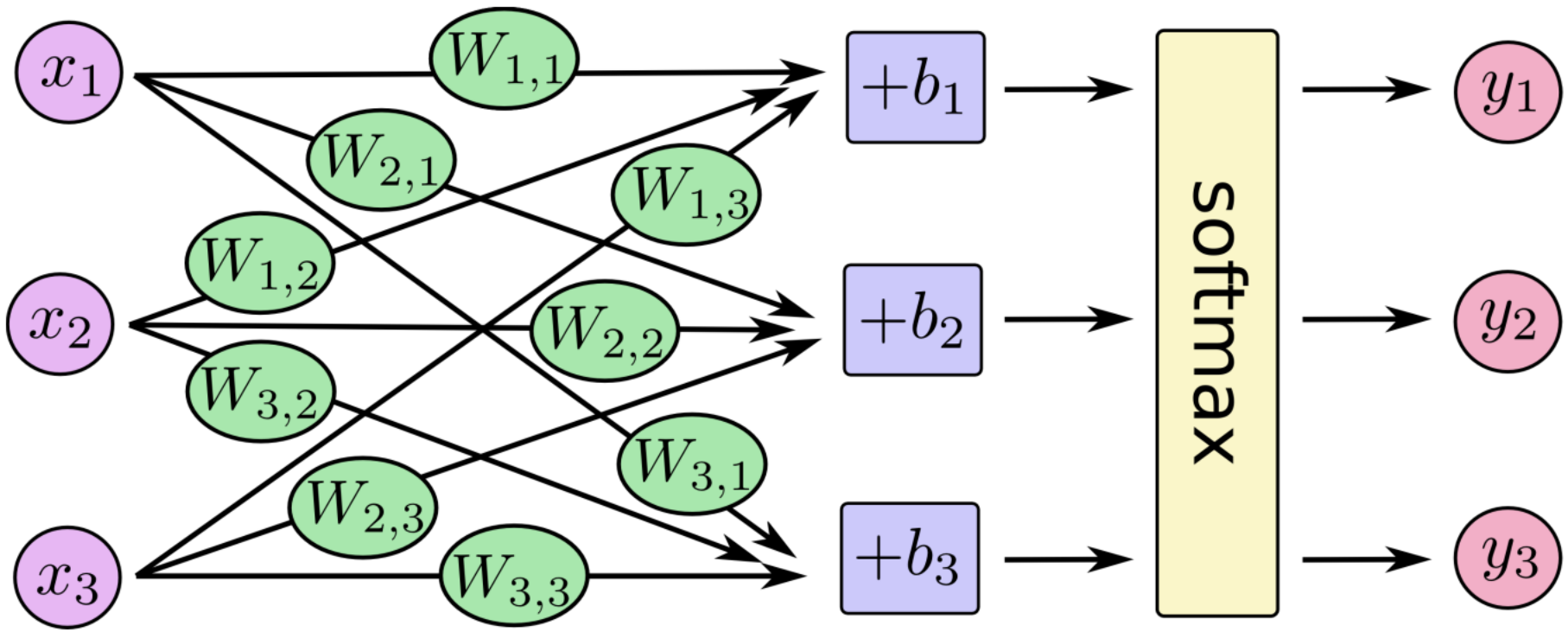
Our dataset had **3** possible **classes**

Multinomial Logistic Regression:

- An extension of logistic regression used to handle **multiclass** problems.

Multinomial Logistic Regression: used for multiclass classification (more than 2 or K classes)




x_1
x_2
x_3
$W_{1,1}$
$W_{2,1}$
$W_{1,2}$
$W_{1,3}$
$W_{2,2}$
$W_{3,2}$
$W_{3,1}$
$W_{2,3}$
$W_{3,3}$
$+b_1$
$+b_2$
$+b_3$
softmax
y_1
y_2
y_3

* First, we defined 'X' as the training dataset and 'y' as the target variable

```
X = data.drop(['CLASS_Category', 'CLASS', 'Gender'], axis= 1)
y = data['CLASS_Category']
```

* Split the dataset into 80/20 training/testing split with a random state of 50

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state = 50)
```

```
print(data.shape)

print(X_train.shape)
print(X_test.shape)

print(y_train.shape)
print(y_test.shape)
```

```
(1000, 14)
(800, 11)
(200, 11)
(800,)
(200,)
```

* Trained the model

```
from sklearn.linear_model import LogisticRegression
```

```
lr = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=2600)
lr.fit(X_train, y_train)
```

* Predictions on the test set

```
y_pred = lr.predict(X_test)
```

```
y_pred = lr.predict(X_test)
```

* Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_pred[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

➡ Looks promising

* Ran metrics and model evaluation

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
Accuracy score is: 93.5 %
Precision score is: 67.51 %
Recall score is: 64.06 %
f1 score is: 65.63 %
```

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Relatively high **accuracy score** – making correct predictions for most of the instances.
(Note: with unbalanced dataset high accuracy may be misleading)

Relatively high **accuracy score** – making correct predictions for most of the instances.
(Note: with unbalanced dataset high accuracy may be misleading)

Precision score is relatively low indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.
(only 67.51% of all predicted diabetes cases were actually diabetic.)

Relatively high **accuracy score** – making correct predictions for most of the instances.
(Note: with unbalanced dataset high accuracy may be misleading)

Precision score is relatively low indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.
(only 67.51% of all predicted diabetes cases were actually diabetic.)

Recall score also relatively low which indicates model wasn't great at detecting all the actual diabetic cases in the dataset (only 64.06%) i.e. missing a considerable amount of diabetic cases.

Relatively high **accuracy score** – making correct predictions for most of the instances.
(Note: with unbalanced dataset high accuracy may be misleading)

Precision score is relatively low indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.
(only 67.51% of all predicted diabetes cases were actually diabetic.)

Recall score also relatively low which indicates model wasn't great at detecting all the actual diabetic cases in the dataset (only 64.06%) i.e. missing a considerable amount of diabetic cases.

F1 score shows moderate balance between correctly identifying diabetes cases and minimizing false positives (i.e. between precision and recall) HOWEVER highlights the need for further optimization.

4.1 Logistic Regression – Classification

4.2 - Normalization of the data - MinMaxScaler

* We then normalized our data using **MinMaxScaler** to see if this would improve the performance of our Logistic Regression model (in our case normalization may help in particular with scaling of features with different ranges)

```
from sklearn.preprocessing import MinMaxScaler
```

* Created a MinMaxScaler object

```
scaler = MinMaxScaler()
```

* Fit and transform the data – columns to be normalized selected, transformation scales the columns, and result of transformation stored in new dataframe

```
X_norm = pd.DataFrame(scaler.fit_transform(X[['Urea', 'Cr', 'BBalc', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL']]),
columns=X[['Urea', 'Cr', 'BBalc', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL']].columns)
X_norm = pd.concat([X_norm, X[['Age', 'BMI', 'Gender_Category']]], axis=1)
```

* Training/testing split on the X_norm dataset

```
X_norm_train, X_norm_test, y_norm_train, y_norm_test = train_test_split(X_norm, y, test_size=.2, random_state = 42)
```

* Trained the model

```
from sklearn.linear_model import LogisticRegression

lr_norm = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=700)
lr_norm.fit(X_norm_train, y_norm_train)
```

* Predictions on the test set

```
y_norm_pred = lr_norm.predict(X_norm_test)
```

*Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_norm_pred[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

➡ Some errors

* Ran metrics and model evaluation

```
Accuracy score is: 92.5 %
Precision score is: 95.9 %
Recall score is: 65.92 %
f1 score is: 64.5 %
```

➡

* We then normalized our data using **MinMaxScaler** to see if this would improve the performance of our Logistic Regression model (in our case normalization may help in particular with scaling of features with different ranges)

```
from sklearn.preprocessing import MinMaxScaler
```

* Created a MinMaxScaler object

```
scaler = MinMaxScaler()
```

* Fit and transform the data – columns to be normalized selected, transformation scales the columns, and result of transformation stored in new dataframe

```
X_norm = pd.DataFrame(scaler.fit_transform(X[['Urea', 'Cr', 'HbA1c', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL']]),
columns=X[['Urea', 'Cr', 'HbA1c', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL']].columns)
```

```
X_norm = pd.concat([X_norm, X[['AGE', 'BMI', 'Gender_Category']]], axis=1)
```

* Training/testing split on the X_norm dataset

```
X_norm_train, X_norm_test, y_norm_train, y_norm_test = train_test_split(X_norm, y, test_size=.2, random_state = 42)
```

* Trained the model

```
from sklearn.linear_model import LogisticRegression
```

```
lr_norm = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=700)
lr_norm.fit(X_norm_train, y_norm_train)
```

* Predictions on the test set

```
y_norm_pred = lr_norm.predict(X_norm_test)
```

*Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_norm_pred[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

➡ Some errors

* Ran metrics and model evaluation

```
Accuracy score is: 92.0 %
Precision score is: 69.9 %
Recall score is: 65.92 %
f1 score is: 64.5 %
```

- Similarly to before, relatively high **accuracy score** – making correct predictions for most of the instances.

- Similarly to before, relatively high **accuracy score** – making correct predictions for most of the instances.

- **Precision score** is relatively low (slightly higher after normalization) indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.

- Similarly to before, relatively high **accuracy score** – making correct predictions for most of the instances.

- **Precision score** is relatively low (slightly higher after normalization) indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.

- **Recall score** also relatively low (very slightly higher after normalization) – indicates model wasn't great at detecting all the actual diabetic cases i.e. missing a considerable amount of diabetic cases.

- Similarly to before, relatively high **accuracy score** – making correct predictions for most of the instances.

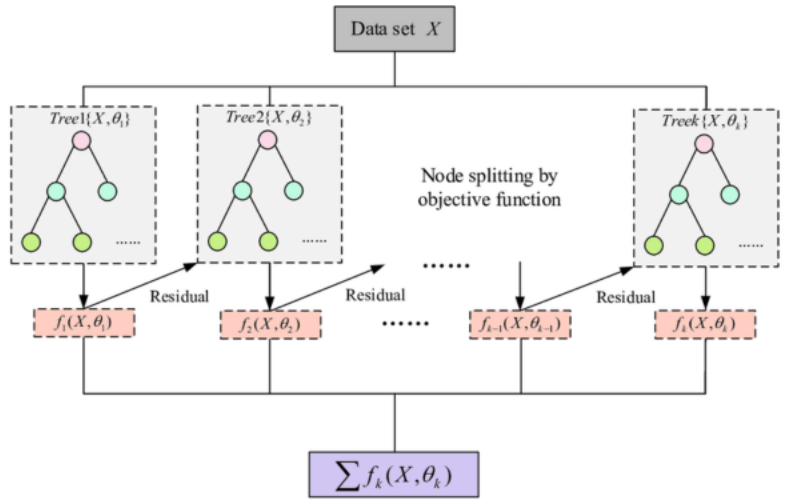
- **Precision score** is relatively low (slightly higher after normalization) indicative that model may be prone to false positives i.e. identifying non-diabetic cases as diabetic.

- **Recall score** also relatively low (very slightly higher after normalization) – indicates model wasn't great at detecting all the actual diabetic cases i.e. missing a considerable amount of diabetic cases.

- **F1 score** is slightly lower than before normalization and shows moderate balance between correctly identifying diabetes cases and minimizing false positives, HOWEVER, once again, this highlights the need for further optimization and a new strategy.

4.3 XGboost

XGboost (**Extreme Gradient Boosting**) is known as a **"black box"** algorithm and is an advanced boosting algorithm that **creates an ensemble of decision trees in a sequential manner** – correcting errors made by previous trees to build a more accurate and robust predictive model.




```
from xgboost import XGBClassifier
```

* Trained the model and created the model instance

```
bst = XGBClassifier(n_estimators=3, max_depth=3, learning_rate=1, objective='multiple:logistic')
```

* Fitted the model

```
bst.fit(X_train, y_train)
```

* Made predictions on test set

```
y_XG_preds = bst.predict(X_test)
```

*Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_XG_preds[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

➡ Looks good

Model Evaluation

```
Accuracy score is: 100.0 %
Precision score is: 100.0 %
Recall score is: 100.0 %
f1 score is: 100.0 %
```

These scores suggest that this model is performing **exceptionally well** on the dataset

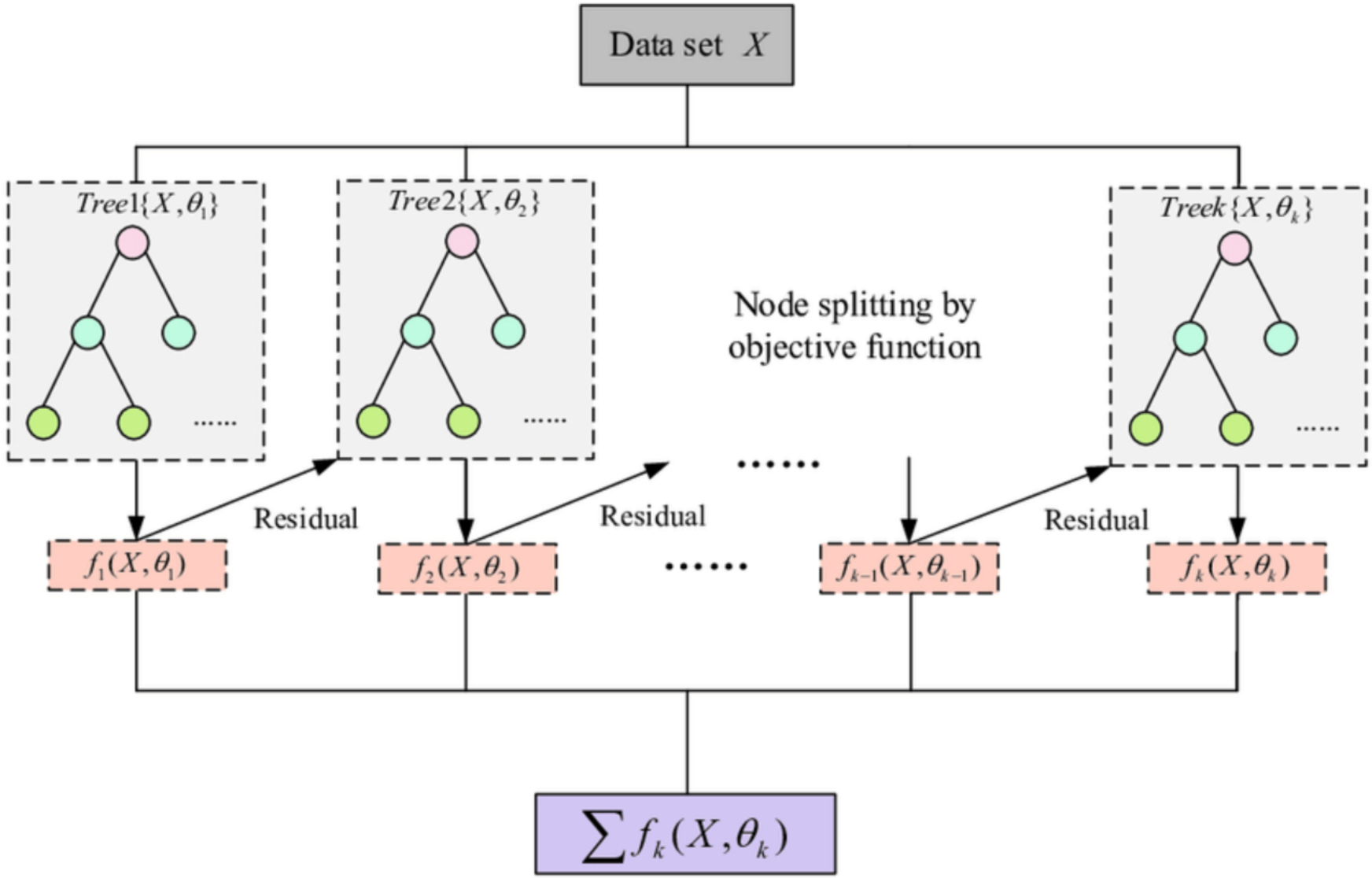
It is identifying all the 3 possible results, without making any false positive or negative predictions.

XGboost (**Extreme Gradient Boosting**) is known as a **“black box”** algorithm and is an advanced boosting algorithm that **creates an ensemble of decision trees in a sequential manner** – correcting errors made by previous trees to build a more accurate and robust predictive model.

XGboost (**Extreme Gradient Boosting**) is known as a **“black box”** algorithm and is an advanced boosting algorithm that **creates an ensemble of decision trees in a sequential manner** – correcting errors made by previous trees to build a more accurate and robust predictive model.

It uses a gradient boosting algorithm to

- **minimize the residual/loss** when adding new models.
- It is able to **handle complex data**, **reduce overfitting**, and
- provide **high prediction performance**.


Data set X
$Tree1\{X,\theta_1\}$
$Tree2\{X,\theta_2\}$
$Treek\{X,\theta_k\}$
Node splitting by objective function
Residual
Residual
Residual
$f_1(X,\theta_1)$
$f_2(X,\theta_2)$
$f_{k-1}(X,\theta_{k-1})$
$f_k(X,\theta_k)$
$\sum f_k(X,\theta_k)$

```
from xgboost import XGBClassifier
```

* Trained the model and created the model instance

```
bst = XGBClassifier(n_estimators=3, max_depth=3, learning_rate=1, objective='multiple:logistic')
```

* Fitted the model

```
bst.fit(X_train, y_train)
```

* Fitted the model

```
bst.fit(X_train, y_train)
```

* Made predictions on test set

```
y_XG_preds = bst.predict(X_test)
```

*Tested the first 20 first prediction values vs real values

```
print(f"Predictions:\n{list(y_XG_preds[:20])}\n\nReal values: \n{list(y_test[:20])}")
```

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

Looks good

Model Evaluation

```
Accuracy score is: 100.0 %
Precision score is: 100.0 %
Recall score is: 100.0 %
f1 score is: 100.0 %
```

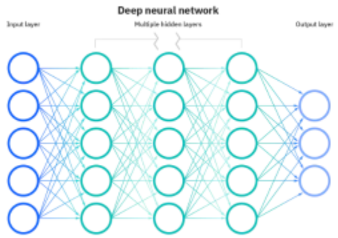
These scores suggest that this model is performing **exceptionally well** on the dataset

It is identifying all the 3 possible results, without making any false positive or negative predictions.

4.4 NEURAL NETWORKS

Neural Networks (NN)

- A class of **machine learning** models inspired by the structure and function of the **human brain**.
- They consist of interconnected nodes or **"neurons"** that receive input data, process it, and then pass the output to other nodes.




```python
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
```

Sequential Model

- A type of neural network
- Designed to process input data in a sequential manner
- Multiple layers of nodes are organized in a linear, sequential way forming a feedforward network.

* Converted labels in the target variable (y_train) using one-hot encoding

```python
y_NN_train = to_categorical(y_train)
```

```python
print(y_train[:3])
```

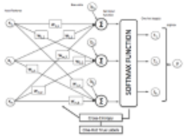
```
886    2
488    2
265    2
Name: CLASS_Category, dtype: int64
```

```python
print(y_NN_train[:3])
```

```
[[0. 0. 1.]
 [0. 0. 1.]
 [0. 0. 1.]]
```

* Trained the keras model

```python
model = Sequential()
model.add(Dense(200, activation='relu', input_shape=(11,)))
model.add(Dense(200, activation='relu'))
model.add(Dense(3, activation='softmax'))
```




```python
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 200)               2400

 dense_1 (Dense)             (None, 200)               40200

 dense_2 (Dense)             (None, 3)                 603

=================================================================
Total params: 43,203
Trainable params: 43,203
Non-trainable params: 0
_________________________________________________________________
```

* Compiled the keras model

```python
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
    )
```

* Fitted the keras model on the dataset

```python
model.fit(
    X_train,
    y_NN_train,
    epochs=500,
    batch_size=30
)
```


```
Epoch 494/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0018 - accuracy: 1.0000
Epoch 495/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0013 - accuracy: 1.0000
Epoch 496/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0013 - accuracy: 1.0000
Epoch 497/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0018 - accuracy: 1.0000
Epoch 498/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0027 - accuracy: 1.0000
Epoch 499/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0029 - accuracy: 1.0000
Epoch 500/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0025 - accuracy: 1.0000
<keras.callbacks.History at [illegible]>
```

* Evaluation of the keras model

```python
loss, accuracy = model.evaluate(X_train, y_NN_train)
```

```
Accuracy:100.00%
Loss: 0.0015
```

* Class predictions with the model

```python
predictions = model.predict(X_test)
predictions[:5]
```

```
7/7 [==============================] - 0s 2ms/step
array([[[illegible]],
       [[illegible]],
       [[illegible]],
       [[illegible]],
       [[illegible]]], dtype=float32)
```

* Class predictions with the model but round them into the closest integer

```python
predictions = predictions.round()
predictions[:5]
```

```
array([[0., 0., 1.],
       [0., 0., 1.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 0., 1.]], dtype=float32)
```

* First 10 predictions vs real values

```
Predictions:   Real values:
[[0. 0. 1.]    [[0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [1. 0. 0.]     [1. 0. 0.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]     [0. 0. 1.]
 [0. 0. 1.]]    [0. 0. 1.]]
```

* Changed categorical values into integers as it was in the y_test

```python
y_NN_pred = []

for prediction in predictions:
    max_index = np.argmax(prediction)
    y_NN_pred.append(max_index)

y_NN_pred[:5]
```

```
[2, 2, 2, 0, 2]
```

* First 20 prediction values vs real values

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

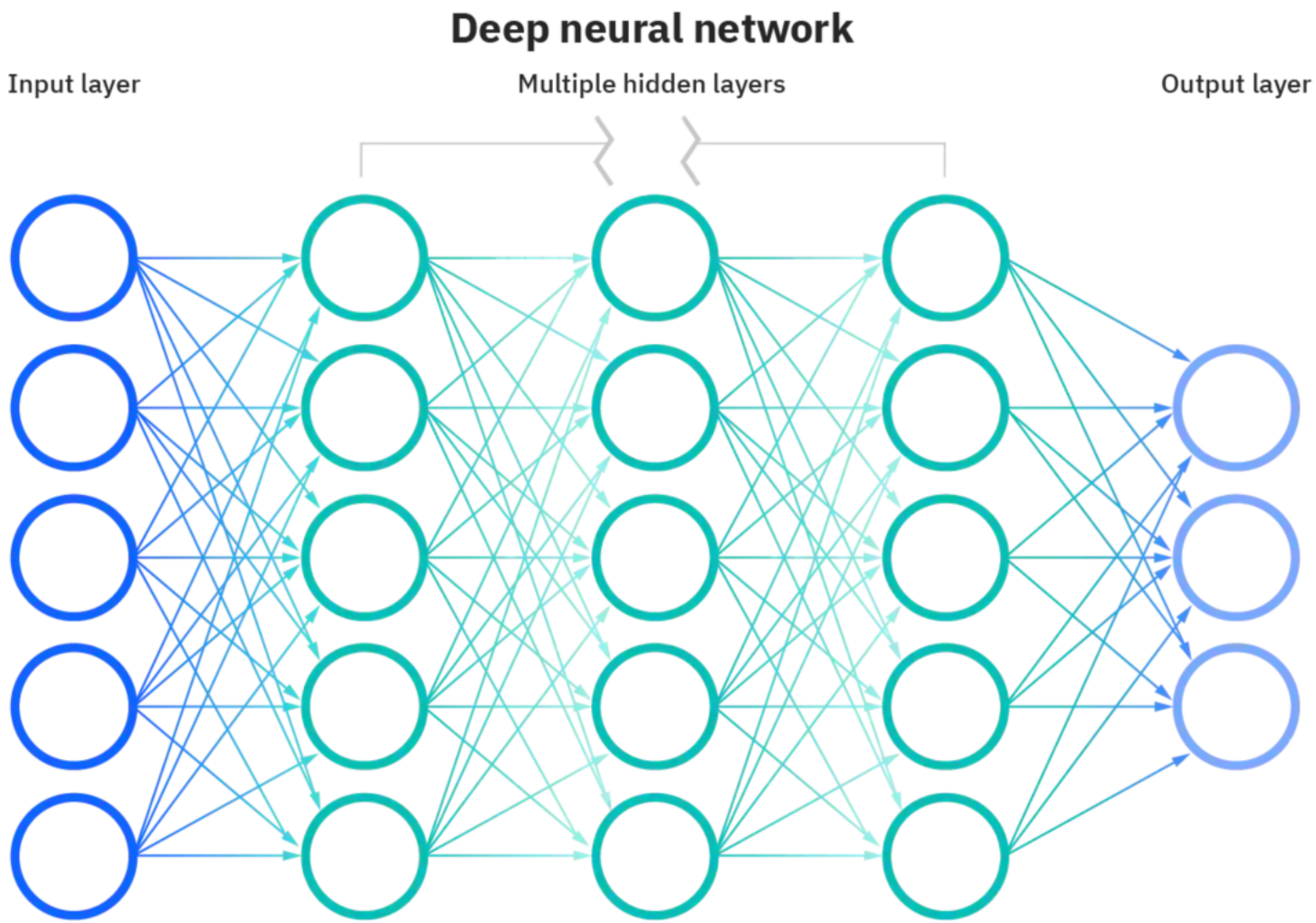
* Metrics and model evaluation

```
Accuracy score is: 95.5 %
Precision score is: 96.61 %
Recall score is: 95.5 %
F1 score is: 95.7 %
```

Neural Networks (NN)

- A class of **machine learning** models inspired by the structure and function of the **human brain**.

- They consist of interconnected nodes or **“neurons”** that receive input data, process it, and then pass the output to other nodes.


Deep neural network
Input layer
Multiple hidden layers
Output layer

```
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
```

Sequential Model

- A type of neural network
- Designed to process input data in a sequential manner
- Multiple layers of nodes are organized in a linear, sequential way forming a feedforward network.

* Converted labels in the target variable (y_train) using one-hot encoding

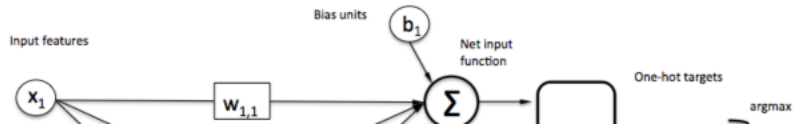
```
y_NN_train = to_categorical(y_train)
```

```
print(y_train[:3])
```

```
886    2
488    2
265    2
Name: CLASS_Category, dtype: int64
```

```
print(y_NN_train[:3])
```

```
[[0. 0. 1.]
 [0. 0. 1.]
 [0. 0. 1.]]
```



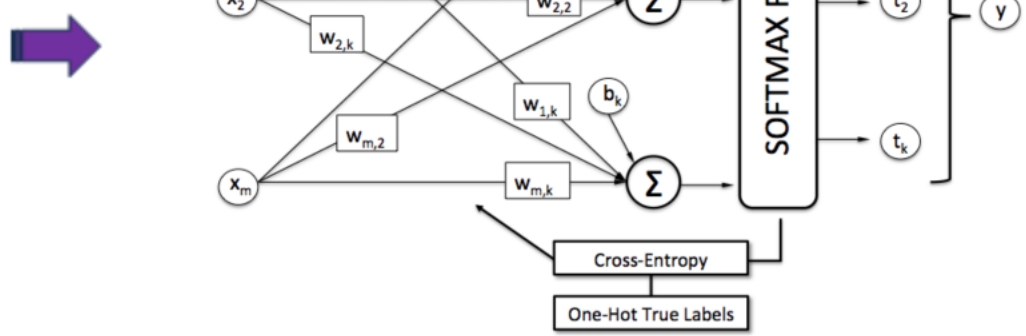
```
Name: CLASS_Category, dtype: int64
```

```
print(y_NN_train[:3])
```

```
[[0. 0. 1.]
 [0. 0. 1.]
 [0. 0. 1.]]
```

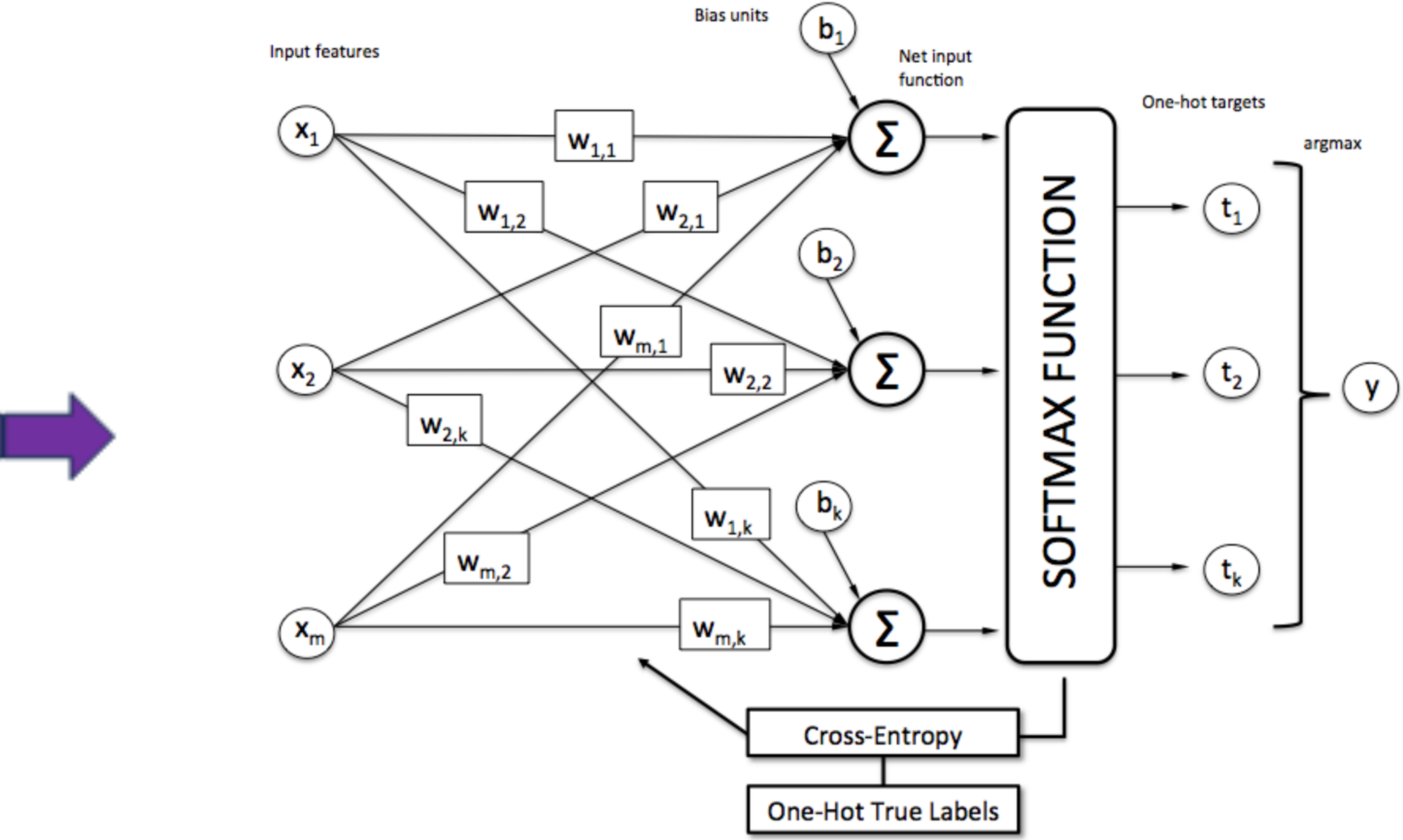
* Trained the keras model

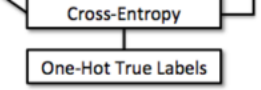
```
model = Sequential()
model.add(Dense(200, activation='relu', input_shape=(11,)))
model.add(Dense(200, activation='relu'))
model.add(Dense(3, activation='softmax'))
```




```
model.summary()
```

```
Model: "sequential"
```


Bias units
Input features
Net input function
One-hot targets
argmax
SOFTMAX FUNCTION
Cross-Entropy
One-Hot True Labels




```
model.summary()
```

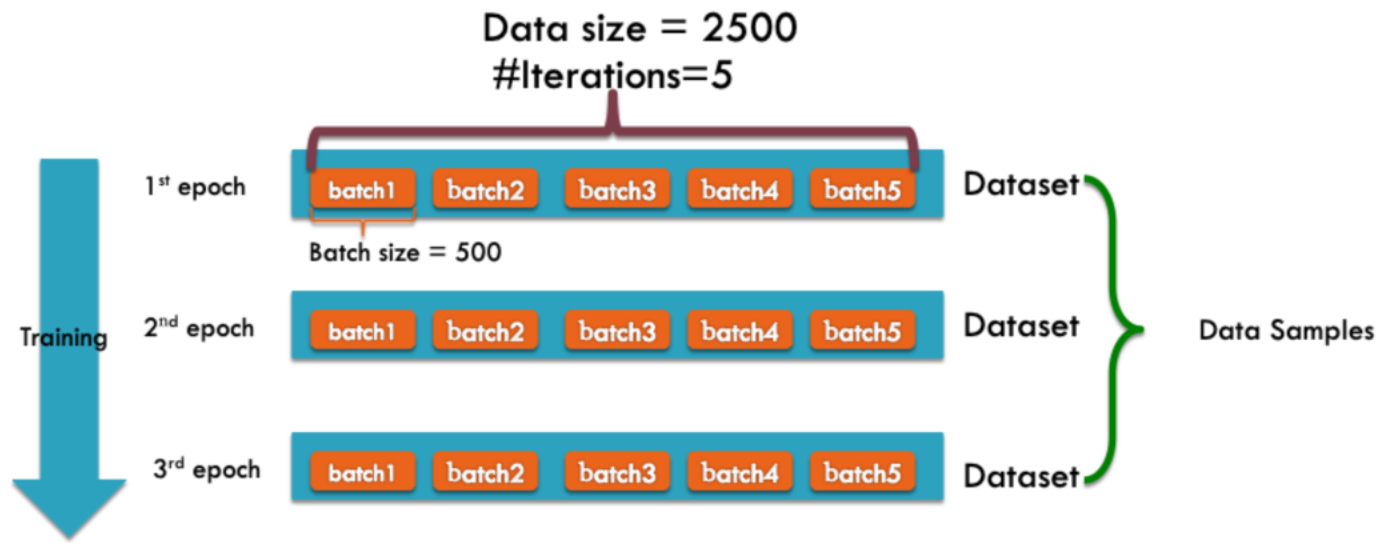
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 200)               2400

 dense_1 (Dense)             (None, 200)               40200

 dense_2 (Dense)             (None, 3)                 603

=================================================================
Total params: 43,203
Trainable params: 43,203
Non-trainable params: 0
_________________________________________________________________
```

* Compiled the keras model

```
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
    )
```

* Fitted the keras model on the dataset

```
model.fit(
    X_train,
    y_NN_train,
    epochs=500,
    batch_size=30
)
```



```
Epoch 494/500
27/27 [==============================] - 0s 4ms/step - loss: 0.0036 - accuracy: 1.0000
Epoch 495/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0043 - accuracy: 1.0000
Epoch 496/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0033 - accuracy: 1.0000
Epoch 497/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0028 - accuracy: 1.0000
Epoch 498/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0027 - accuracy: 1.0000
Epoch 499/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0029 - accuracy: 1.0000
Epoch 500/500
27/27 [==============================] - 0s 3ms/step - loss: 0.0025 - accuracy: 1.0000
<keras.callbacks.History at 0x7d9fc11e19c0>
```

* Evaluation of the keras model

```
loss, accuracy = model.evaluate(X_train, y_NN_train)
```

```
Accuracy:100.00%
Loss: 0.0015
```

* Class predictions with the model

```
predictions = model.predict(X_test)
predictions[:5]
```

```
7/7 [==============================] - 0s 2ms/step
array([[1.0368682e-10, 2.9687493e-05, 9.9997032e-01],
       [1.9128512e-16, 2.3144937e-06, 9.9999774e-01],
       [1.6650246e-19, 2.4478529e-28, 1.0000000e+00],
       [9.9999857e-01, 1.0152958e-06, 3.1146052e-07],
       [4.9838639e-10, 2.4898136e-05, 9.9997509e-01]], dtype=float32)
```

* Class predictions with the model but round them into the closest integer

```
predictions = predictions.round()
predictions[:5]
```

```
array([[0., 0., 1.],
       [0., 0., 1.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 0., 1.]], dtype=float32)
```

* First 10 predictions vs real values

```
Predictions:  Real values:
[[0. 0. 1.]   [[0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [1. 0. 0.]    [1. 0. 0.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]    [0. 0. 1.]
 [0. 0. 1.]]   [0. 0. 1.]]
```

* Changed categorical values into integers as it was in the y_test

```
y_NN_pred = []

for prediction in predictions:
    max_index = np.argmax(prediction)
    y_NN_pred.append(max_index)

y_NN_pred[:5]
```

```
[2, 2, 2, 0, 2]
```

* First 20 prediction values vs real values

```
Predictions:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Real values:
[2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

* Metrics and model evaluation

```
Accuracy score is: 95.5 %
Precision score is: 96.61 %
Recall score is: 95.5 %
F1 score is: 95.7 %
```

- High **accuracy score** – making correct predictions for most of the instances.

- High **accuracy score** – making correct predictions for most of the instances.

- High **precision score** is indicative that model is making very few false positive predictions i.e. very few non-diabetic cases as diabetic (96.61% of all predicted diabetes cases were actually diabetic).

- High **accuracy score** – making correct predictions for most of the instances.

- High **precision score** is indicative that model is making very few false positive predictions i.e. very few non-diabetic cases as diabetic (96.61% of all predicted diabetes cases were actually diabetic).

- **Recall score** also high – indicates model very good at detecting all the actual diabetic cases in the dataset (only 95.5%) i.e. missing very few diabetic cases.

- High **accuracy score** – making correct predictions for most of the instances.

- High **precision score** is indicative that model is making very few false positive predictions i.e. very few non-diabetic cases as diabetic (96.61% of all predicted diabetes cases were actually diabetic).

- **Recall score** also high – indicates model very good at detecting all the actual diabetic cases in the dataset (only 95.5%) i.e. missing very few diabetic cases.

- **F1 score** shows good balance between correctly identifying diabetes cases and minimizing false positives

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

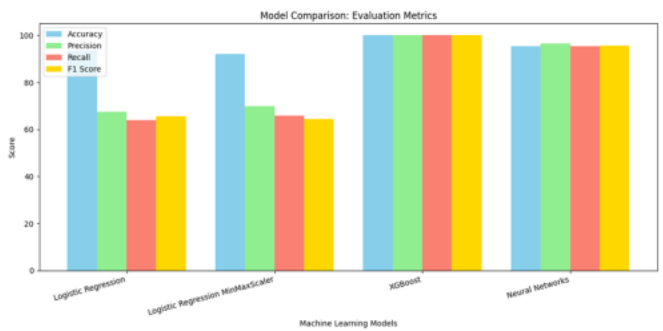
Index

5 -CONCLUSIONS

Quick backtrack to study objective...

We aimed to design a **predictive diabetes model** that could estimate the likelihood of diabetes development based on the provided factors.

SUMMARY OF RESULTS



REFERENCES

1. https://www.who.int/news-room/fact-sheets/detail/diabetes
2. https://www.cdc.gov/diabetes/managing/problems
3. DiMeglio LA, Evans-Molina C, Oram RA. Type 1 diabetes. Lancet. 2018 Jun 16;391(10138):2449-2462. doi: 10.1016/S0140-6736(18)31320-5. PMID: 29916386; PMCID: PMC6661119.
4. Risérus U, Willett WC, Hu FB. Dietary fats and prevention of type 2 diabetes. Prog Lipid Res. 2009 Jan;48(1):44-51. doi: 10.1016/j.plipres.2008.10.002. Epub 2008 Nov 7. PMID: 19032965; PMCID: PMC2654180.
5. https://bmcmedicine.biomedcentral.com/articles/10.1186/s12916-017-0901x
6. Tabish SA. Is Diabetes Becoming the Biggest Epidemic of the Twenty-first Century? Int J Health Sci (Qassim). 2007 Jul;1(2):V-VIII. PMID: 21475425; PMCID: PMC3068646
7. https://diabetesatlas.org
8. Tabák AG, Herder C, Rathmann W, Brunner EJ, Kivimäki M. Prediabetes: a high-risk state for diabetes development. Lancet. 2012 Jun 16;379(9833):2279-90. doi: 10.1016/S0140-6736(12)60283-9. Epub 2012 Jun 9. PMID: 22683128; PMCID: PMC3891203.
9. Carris NW, Magness RR, Labovitz AJ. Prevention of Diabetes Mellitus in Patients With Prediabetes. Am J Cardiol. 2019 Feb 1;123(3):507-512. doi: 10.1016/j.amjcard.2018.10.032. Epub 2018 Nov 6. PMID: 30528418; PMCID: PMC6350898.
10. https://www.cdc.gov/diabetes/basics/prediabetes
11. https://data.mendeley.com/datasets/wj9rwkp9c2/1

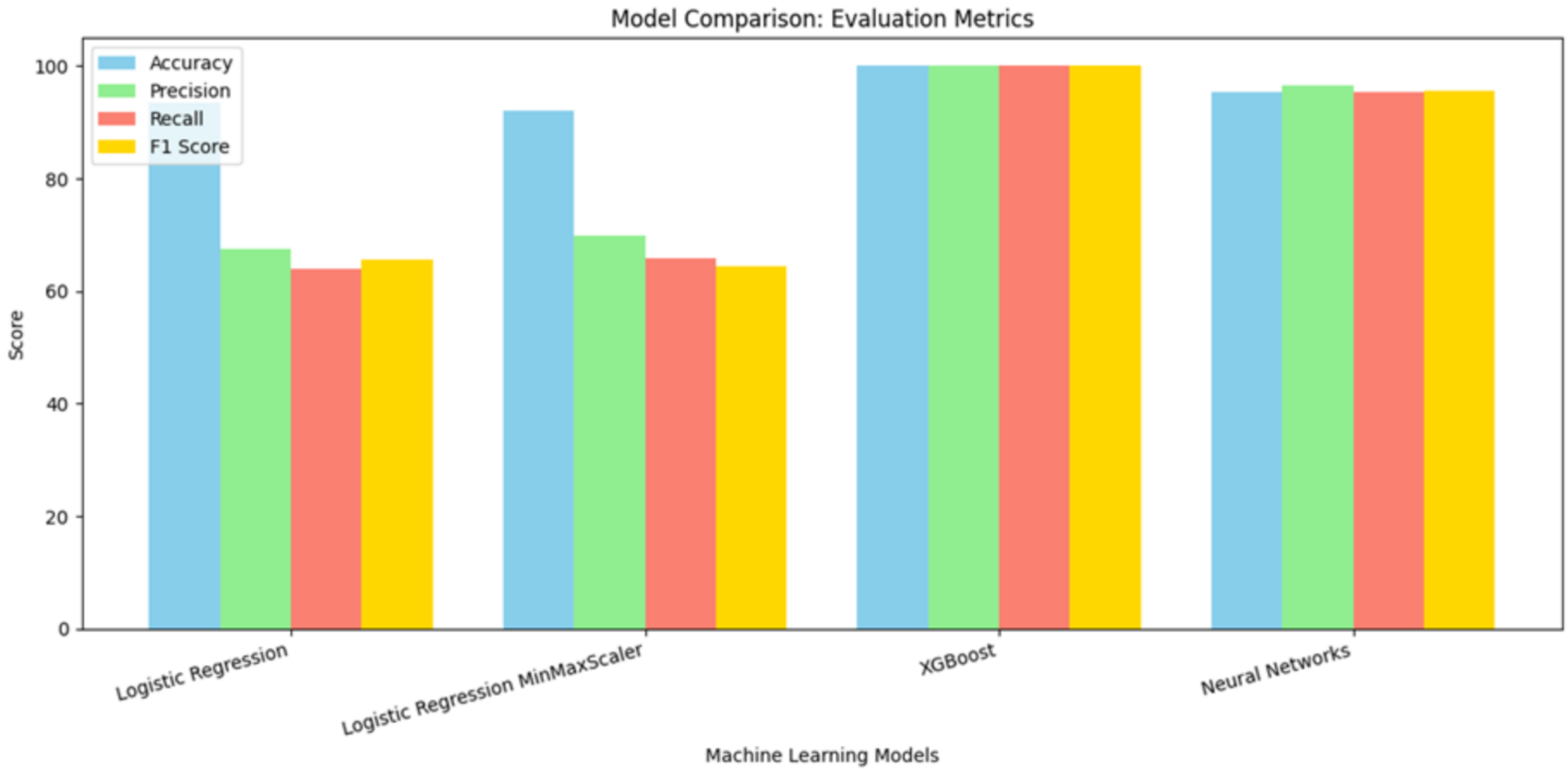
Big thanks to FilLipa and Shandy!

Contact details:
daniellavuletic218@gmail.com
n.ashoori1989@gmail.com
Nastaran Ashoori on Linkedin

Quick backtrack to study objective...

We aimed to design a **predictive diabetes model** that could estimate the likelihood of diabetes development based on the provided factors.

SUMMARY OF RESULTS



- According the our dataset and analysis, the **variables** with the strongest correlation, and therefore most likely to put people at **risk** for the development of **diabetes**, include an increased body mass index (**BMI**), **HbA1c** (blood glucose) and **age**.

- According the our dataset and analysis, the **variables** with the strongest correlation, and therefore most likely to put people at **risk** for the development of **diabetes**, include an increased body mass index (**BMI**), **HbA1c** (blood glucose) and **age**.

- Our machine learning models were able to successfully predict whether a patient belonged to the non-diabetic, pre-diabetic or diabetic class, based on the given variables. The **XGboost** model performing the best.

- According the our dataset and analysis, the **variables** with the strongest correlation, and therefore most likely to put people at **risk** for the development of **diabetes**, include an increased body mass index (**BMI**), **HbA1c** (blood glucose) and **age**.

- Our machine learning models were able to successfully predict whether a patient belonged to the non-diabetic, pre-diabetic or diabetic class, based on the given variables. The **XGboost** model performing the best.

- Further studies would benefit from an increased **sample size** and perhaps a more **balanced dataset**. A more **globally representative** sample would also add value.

REFERENCES

1. https://www.who.int/news-room/fact-sheets/detail/diabetes
2. https://www.cdc.gov/diabetes/managing/problems
3. DiMeglio LA, Evans-Molina C, Oram RA. Type 1 diabetes. Lancet. 2018 Jun 16;391(10138):2449-2462. doi: 10.1016/S0140-6736(18)31320-5. PMID: 29916386; PMCID: PMC6661119.
4. Risérus U, Willett WC, Hu FB. Dietary fats and prevention of type 2 diabetes. Prog Lipid Res. 2009 Jan;48(1):44-51. doi: 10.1016/j.plipres.2008.10.002. Epub 2008 Nov 7. PMID: 19032965; PMCID: PMC2654180.
5. https://bmcmedicine.biomedcentral.com/articles/10.1186/s12916-017-0901x
6. Tabish SA. Is Diabetes Becoming the Biggest Epidemic of the Twenty-first Century? Int J Health Sci (Qassim). 2007 Jul;1(2):V-VIII. PMID: 21475425; PMCID: PMC3068646
7. https://diabetesatlas.org
8. Tabák AG, Herder C, Rathmann W, Brunner EJ, Kivimäki M. Prediabetes: a high-risk state for diabetes development. Lancet. 2012 Jun 16;379(9833):2279-90. doi: 10.1016/S0140-6736(12)60283-9. Epub 2012 Jun 9. PMID: 22683128; PMCID: PMC3891203.
9. Carris NW, Magness RR, Labovitz AJ. Prevention of Diabetes Mellitus in Patients With Prediabetes. Am J Cardiol. 2019 Feb 1;123(3):507-512. doi: 10.1016/j.amjcard.2018.10.032. Epub 2018 Nov 6. PMID: 30528418; PMCID: PMC6350898.
10. https://www.cdc.gov/diabetes/basics/prediabetes
11. https://data.mendeley.com/datasets/wj9rwkp9c2/1

Big thanks to FilLipa and Shandy!

Contact details:
daniellavuletic218@gmail.com
n.ashoori1989@gmail.com
Nastaran Ashoori on Linkedin

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index

Diabetes Prediction Model

Using Machine Learning Regression Algorithms and Neural Networks

Presentation by Nastaran Ashoori & Daniella Vuletić
CodeOp DSPT10 - July 2023

codeop

Index