Orion Team Proudly Present :

ASTATINE FINAL PROJECT

Diabetes Indicators Classification Model

This project is supervised by:

DigitalSkola
Uncover The World of Digital Skills with Us

TEAM 7 - ORION

01
ISNEN HADI AL-GHOZALI

as Betelgeuse

02
KHUZIL AFWA

as Alnilam

03
MUHAMMAD ASKAR FATHIN

as RIGEL

04
SHERLLYA REBECCA EZRA

as BELLATRIX

05
TABITA KRISTINA MORA

as MEISSA

T O

01

DESCRIPTIVE PURPOSE

DESCRIPTIVE AND PURPOSE

ASTATINE is a data science project to detect prediabetes and diabetes conditions based on Behavioral Risk Factor Surveillance System (BRFSS) indicators. This project uses **diabetes _ binary _ health _ indicators _ BRFSS2015.csv** is a clean dataset of **253,680 survey responses** to the CDC's BRFSS2015. The target variable Diabetes_binary **has 2 classes**. 0 is for no diabetes, and 1 is for prediabetes or diabetes. This dataset has **21 feature variables** and is **not balanced**. This dataset is from https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?resource=download&select=diabetes_binary_health_indicators_BRFSS2015.csv

Before developing a **classification model**, **Exploratory Data Analysis (EDA)** and **Data Preprocessing** are first carried out. The EDA stage is aimed at identifying patterns, finding anomalies, testing hypotheses, and checking assumptions. Data Preprocessing is carried out to eliminate several problems that can interfere with data processing, such as data that is not normally distributed and data imbalance.

The prediction models developed in this project are:

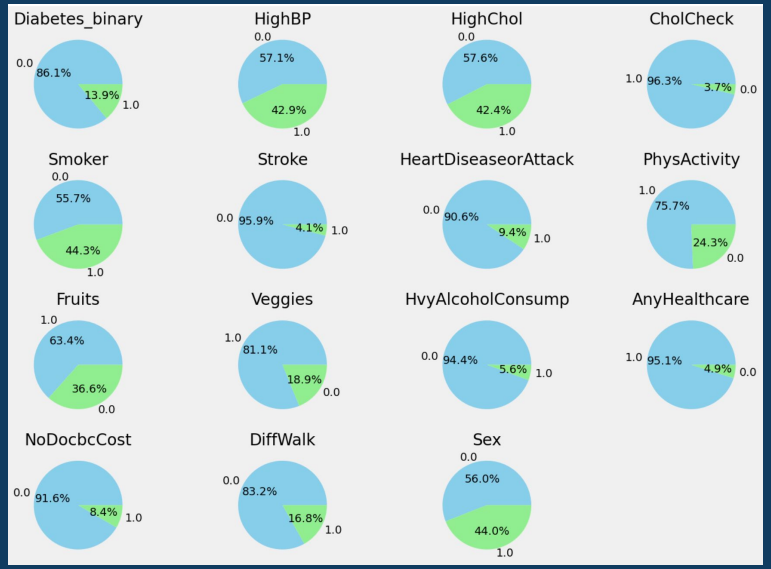
01	02	03	04	05
SVM	XGBOOST	RANDOM FOREST	NAIVE BAYES	ANN

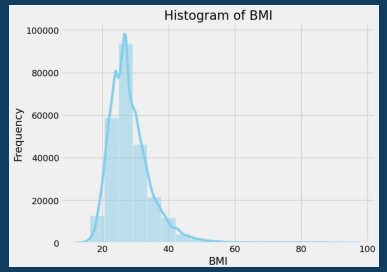
02

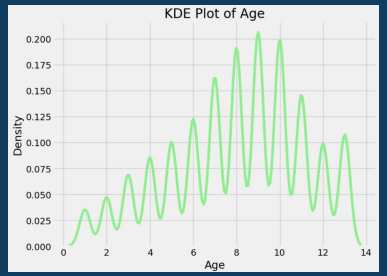
BASIC EXPLORATION

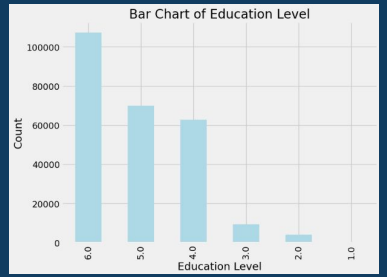
Check Data Distributions, The Outlier, Duplicated Data

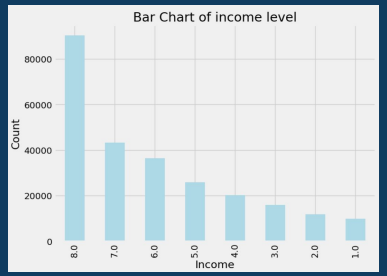
Check Data Distributions







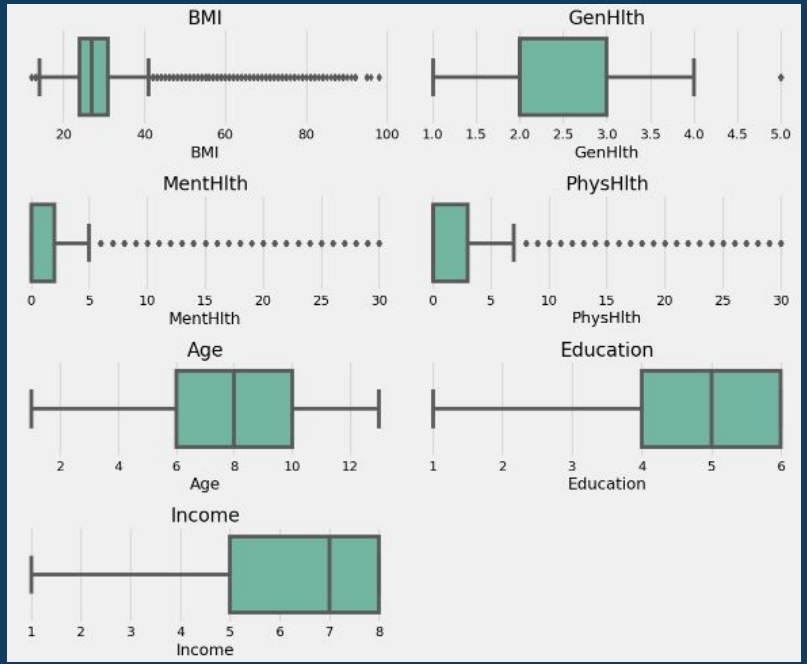





```
Shapiro-Wilk Test for BMI:
Test Statistic: 0.8717145323753357
p-value: 0.0
Sample does not look Gaussian (reject H0)


Shapiro-Wilk Test for Income:
Test Statistic: 0.8491994738578796
p-value: 0.0
Sample does not look Gaussian (reject H0)
```

Normality Test : In the Shapiro-Wilk test for BMI and Income, the obtained p-value of 0.0 suggests strong evidence to reject the null hypothesis. This indicates that the **BMI** and **Income data** is **not normally distributed.**

Check The Outliers

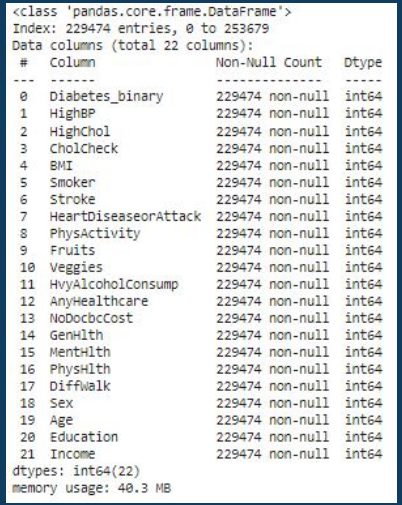


There's outlier in column BMI, GenHlth, MenHlth, PhysHlth, but all columns will transpose to categorical value.

Check and Drop Duplicated Data

There were **24,206 duplicate rows** in the DataFrame, but after performing the duplicate removal operation, there were **no more duplicate rows**, resulting in the total number of rows **decreasing** to **229,474** from the initial **253,680**. This step is crucial for data cleaning and ensuring the integrity of the data used in analysis.

Transform The Data Into Integer



```
<class 'pandas.core.frame.DataFrame'>
Index: 229474 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes_binary       229474 non-null  int64
 1   HighBP                229474 non-null  int64
 2   HighChol              229474 non-null  int64
 3   CholCheck             229474 non-null  int64
 4   BMI                   229474 non-null  int64
 5   Smoker                229474 non-null  int64
 6   Stroke                229474 non-null  int64
 7   HeartDiseaseorAttack  229474 non-null  int64
 8   PhysActivity          229474 non-null  int64
 9   Fruits                229474 non-null  int64
 10  Veggies               229474 non-null  int64
 11  HvyAlcoholConsump     229474 non-null  int64
 12  AnyHealthcare         229474 non-null  int64
 13  NoDocbcCost           229474 non-null  int64
 14  GenHlth               229474 non-null  int64
 15  MentHlth              229474 non-null  int64
 16  PhysHlth              229474 non-null  int64
 17  DiffWalk              229474 non-null  int64
 18  Sex                   229474 non-null  int64
 19  Age                   229474 non-null  int64
 20  Education             229474 non-null  int64
 21  Income                229474 non-null  int64
dtypes: int64(22)
memory usage: 40.3 MB
```

As a result, all columns in the DataFrame now have an integer data type, with no missing values.

03

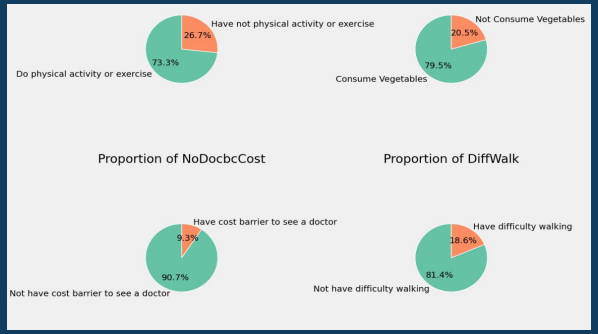
EDA

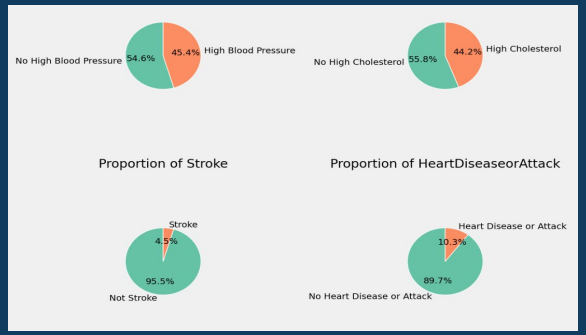
EXPLORATORY DATA ANALYSIS

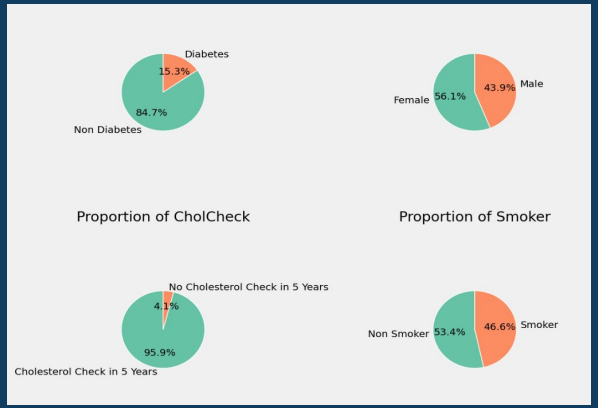
Univariate & Bivariate Analysis

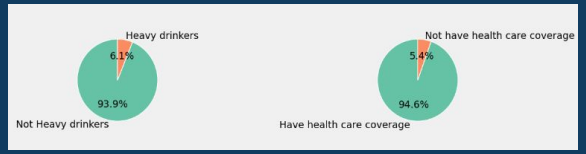
Univariate Analysis

Binary Variables Proportion





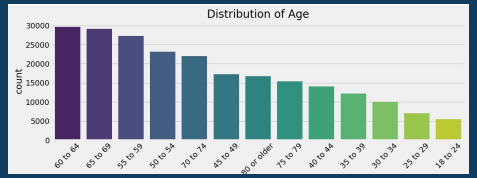




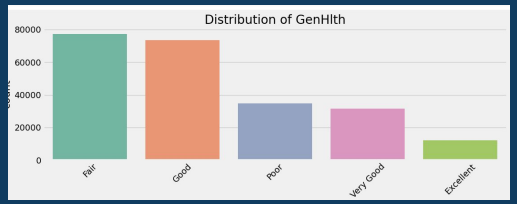
The percentage of individuals who smoke/have high blood pressure/high cholesterol is above 40%, whereas the lowest percentage is for individuals who have had a stroke or have not had their cholesterol checked in the last 5 years, which is below 5%.

Univariate Analysis

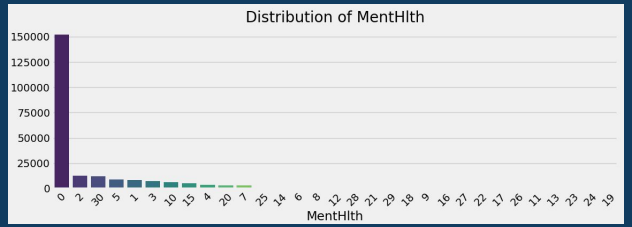
Categorical Variables Proportion



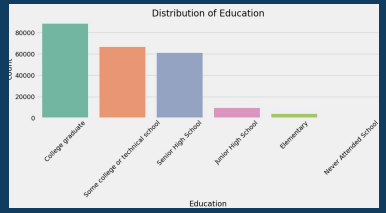
The majority of individuals fall within the age range of **55-64 years**, while the least percentage is found in the age range of **18-44 years.**



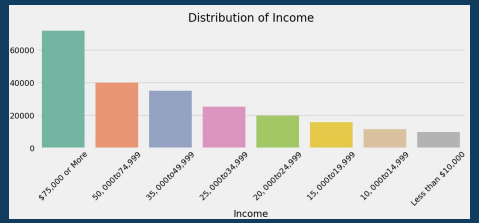
The distribution of general health is **highest** in the **"fair"** category, followed by **"good".**



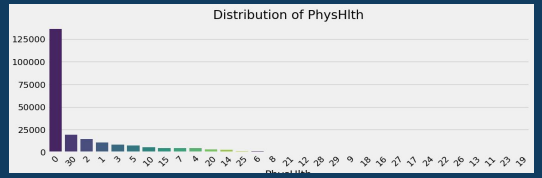
The **highest** distribution of mental health is observed at the value of **0**, followed by **2** and **30.**



The **highest** frequency of education level is observed in **"college student"**, followed by **"some college or technical school"**, while the **lowest** frequency is found in **"never attended school"**.



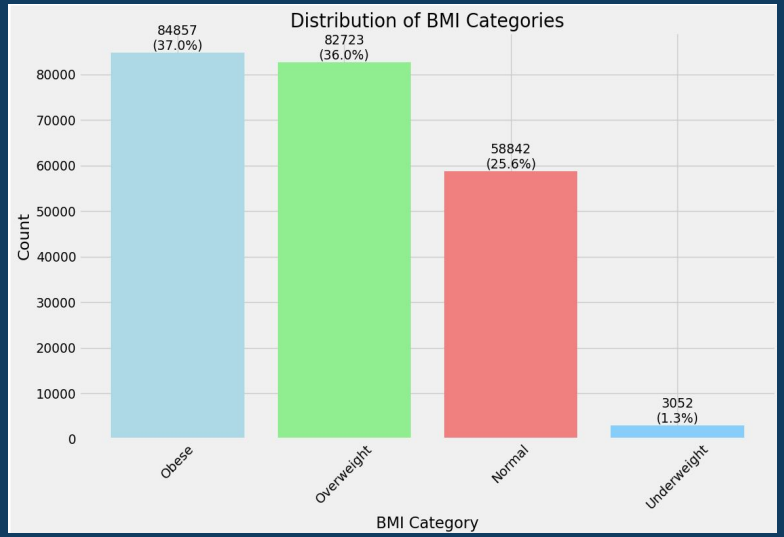
The **highest** frequency of income level is in the category **"$75,000 or More"**, while the **lowest** frequency is in the category **"Less than"**.



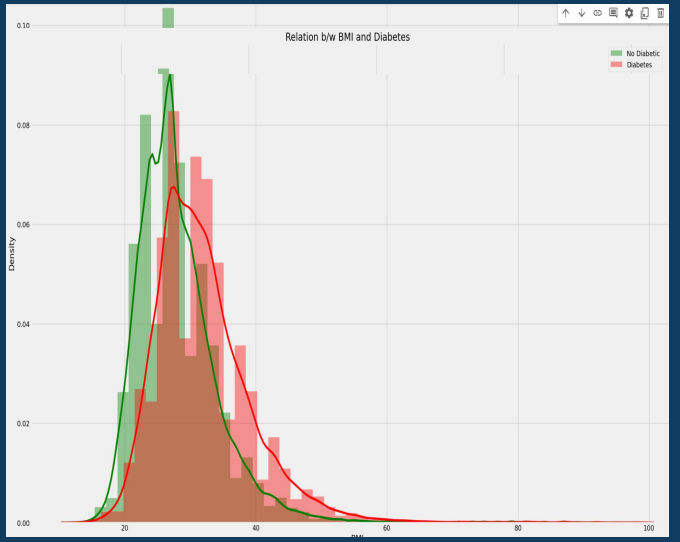
The **highest** distribution of physical health is observed at the value of **0**, followed by **30, 2, 1 etc.**

Univariate Analysis

Numerical Variables Proportion

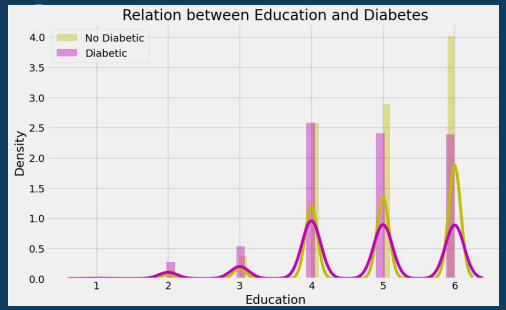


The majority of individuals in this dataset experience overweight to obesity, with only 25.6% having a normal BMI.

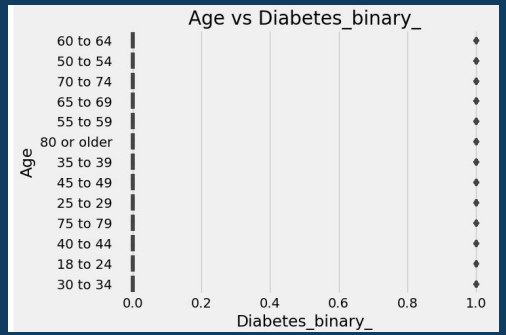


There is a significant relationship between BMI and diabetes.

Univariate Analysis

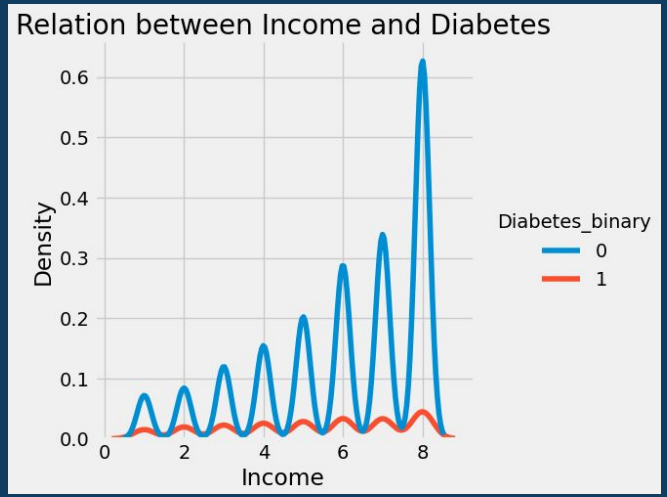


Most people have a high level of education, and those with higher levels of education tend to experience better overall health.



As the age increases, the chances of diabetes also commonly increases.

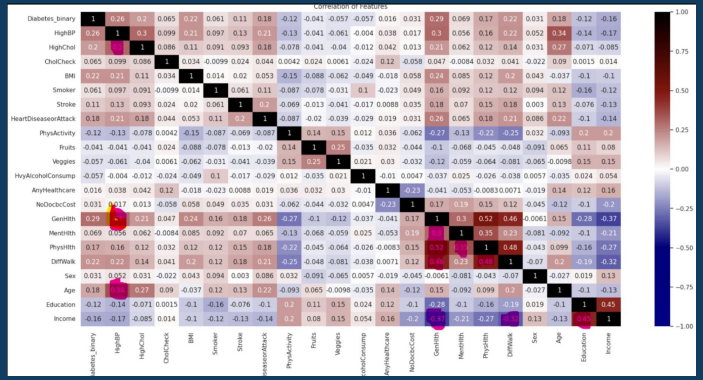
Heart Disease Correlation Factor

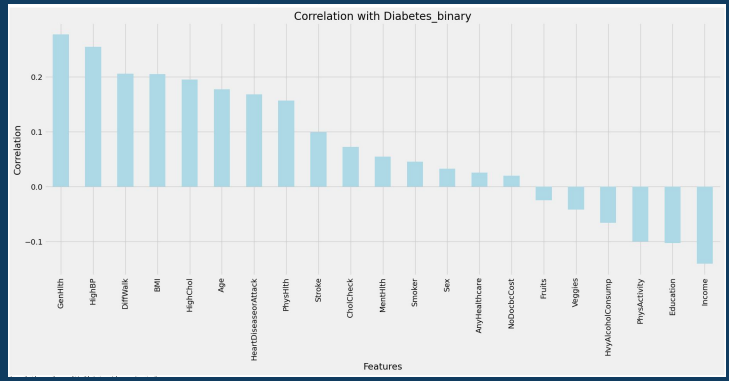


As the age increases, the chances of diabetes also commonly increases.

Bivariate Analysis

Feature Selection






```
Correlation values with Diabetes_binary (sorted):
GenHlth                0.276940
HighBP                 0.254318
DiffWalk               0.205302
BMI                    0.205086
HighChol               0.194944
Age                    0.177263
HeartDiseaseorAttack   0.168213
PhysHlth               0.156211
Stroke                 0.099193
CholCheck              0.072523
MentHlth               0.054153
Smoker                 0.045504
Sex                    0.032724
AnyHealthcare          0.025331
NoDocbcCost            0.020048
Fruits                -0.024805
Veggies               -0.041734
HvyAlcoholConsump     -0.065950
PhysActivity          -0.100404
Education             -0.102686
Income                -0.140659
dtype: float64
```

Strong Correlation: GenHlth, HighBP, DiffWalk, BMI

Moderate Corellation with positive relation: HighChol, Age, HeartDiseaseorAttack, PhysHlth, Stroke, CholCheck, MentHlth

Moderate Corellation with negative relation: Income, Education, PhysActivity, HvyAlcoholConsump

Weak Correlation: Smoker, Sex, AnyHealthcare, NoDocbcCost, Fruits, Veggies

Bivariate Analysis

VIF and ANOVA Test

In the result, the "**const**" variable has the **highest VIF** value, indicating high multicollinearity with other variables in the dataset. However, most other variables have **VIF** values close to 1, indicating **low multicollinearity**. Nevertheless, the "**GenHlth**" variable stands out with a relatively **high VIF** value of **1.741**, suggesting multicollinearity with other variables.

const	109.425291
Diabetes_binary	1.182154
HighBP	1.315161
HighChol	1.166374
CholCheck	1.035970
BMI	1.141796
Smoker	1.076125
Stroke	1.077944
HeartDiseaseorAttack	1.170400
PhysActivity	1.130550
Fruits	1.097950
Veggies	1.098136
HvyAlcoholConsump	1.027834
AnyHealthcare	1.109935
NoDocbcCost	1.135686
GenHlth	1.741508
MentHlth	1.221789
PhysHlth	1.594308
DiffWalk	1.513943
Sex	1.076736
Age	1.359039
Education	1.272148
Income	1.431806

dtype: float64

	0	1	2	3	4	5	6	7	8	9
0	1	1	40	0	5	15	1	9	4	3
1	0	0	25	0	3	0	0	7	6	1
2	1	1	28	0	5	30	1	9	4	8

After conducting the **ANOVA test**, the **top 10 features** were selected from the data for further modeling. The selected feature matrix has dimensions of **229,474 rows** (samples) and **10 columns** (selected features).

Bivariate Analysis

Chi Square Test

	Feature	Score
15	PhysHlth	97988.761672
3	BMI	15507.736174
14	MentHlth	11419.584750
18	Age	8539.906340
0	HighBP	8098.548237
16	DiffWalk	7875.496177
13	GenHlth	7671.732832
6	HeartDiseaseorAttack	5822.145697
1	HighChol	4869.312739
20	Income	3377.099257
5	Stroke	2156.678382
10	HvyAlcoholConsump	937.401148
7	PhysActivity	617.563886
19	Education	479.112939
4	Smoker	253.826098
17	Sex	137.837135
12	NoDocbcCost	83.662830
9	Veggies	82.098846
8	Fruits	54.688897
2	CholCheck	48.904140
11	AnyHealthcare	7.949731

```
<class 'pandas.core.frame.DataFrame'>
Index: 229474 entries, 0 to 253679
Data columns (total 17 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes_binary       229474 non-null  int64
 1   HighBP                229474 non-null  int64
 2   HighChol              229474 non-null  int64
 3   BMI                   229474 non-null  int64
 4   Smoker                229474 non-null  int64
 5   Stroke                229474 non-null  int64
 6   HeartDiseaseorAttack  229474 non-null  int64
 7   PhysActivity          229474 non-null  int64
 8   HvyAlcoholConsump     229474 non-null  int64
 9   GenHlth               229474 non-null  int64
 10  MentHlth              229474 non-null  int64
 11  PhysHlth              229474 non-null  int64
 12  DiffWalk              229474 non-null  int64
 13  Sex                   229474 non-null  int64
 14  Age                   229474 non-null  int64
 15  Education             229474 non-null  int64
 16  Income                229474 non-null  int64
dtypes: int64(17)
memory usage: 39.6 MB
```

	count	mean	std	min	25%	50%	75%	max
Diabetes_binary	229474.0	0.152945	0.359936	0.0	0.0	0.0	0.0	1.0
HighBP	229474.0	0.454343	0.497912	0.0	0.0	0.0	1.0	1.0
HighChol	229474.0	0.441640	0.496584	0.0	0.0	0.0	1.0	1.0
BMI	229474.0	28.687507	6.789204	12.0	24.0	27.0	32.0	98.0
Smoker	229474.0	0.465800	0.498830	0.0	0.0	0.0	1.0	1.0
Stroke	229474.0	0.044816	0.206899	0.0	0.0	0.0	0.0	1.0
HeartDiseaseorAttack	229474.0	0.103336	0.304398	0.0	0.0	0.0	0.0	1.0
PhysActivity	229474.0	0.733042	0.442371	0.0	0.0	1.0	1.0	1.0
HvyAlcoholConsump	229474.0	0.060791	0.238947	0.0	0.0	0.0	0.0	1.0
GenHlth	229474.0	2.601820	1.064962	1.0	2.0	3.0	3.0	5.0
MentHlth	229474.0	3.509866	7.717643	0.0	0.0	0.0	2.0	30.0
PhysHlth	229474.0	4.681219	9.050877	0.0	0.0	0.0	4.0	30.0
DiffWalk	229474.0	0.185751	0.388906	0.0	0.0	0.0	0.0	1.0
Sex	229474.0	0.439087	0.496277	0.0	0.0	0.0	1.0	1.0
Age	229474.0	8.085068	3.094451	1.0	6.0	8.0	10.0	13.0
Education	229474.0	4.979741	0.992989	1.0	4.0	5.0	6.0	6.0
Income	229474.0	5.888615	2.092888	1.0	4.0	6.0	8.0	8.0

The top 10 best features selected from the data based on **the chi-square test.** These features are chosen for their ability to predict the target variable effectively. After feature selection, **several columns are removed** from the original dataset, namely ["**Fruits**", "**Veggies**", "**NoDocbcCost**", "**CholCheck**", "**AnyHealthcare**"]. Following this removal, information and descriptive statistics about the remaining data are displayed, providing insights into the data types, non-null counts, and summary statistics.

04

DATA PRE-PROCESSING

Operational Variable &
Treatment for Imbalance
Data and Data Scaling

T 4

OPERATIONAL VARIABLE

	HighBP	HighChol	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	HvyAlcoholConsump	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
0	1	1	40	1	0	0	0	0	5	18	15	1	0	9	4	3
1	0	0	25	1	0	0	1	0	3	0	0	0	0	7	6	1
2	1	1	28	0	0	0	0	0	5	30	30	1	0	9	4	8
3	1	0	27	0	0	0	1	0	2	0	0	0	0	11	3	6
4	1	1	24	0	0	0	1	0	2	3	0	0	0	11	5	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
253675	1	1	45	0	0	0	0	0	3	0	5	0	1	5	6	7
253676	1	1	18	0	0	0	0	0	4	0	0	1	0	11	2	4
253677	0	0	28	0	0	0	1	0	1	0	0	0	0	2	5	2
253678	1	0	23	0	0	0	0	0	3	0	0	0	1	7	5	1
253679	1	1	25	0	0	1	1	0	2	0	0	0	0	9	6	2

229474 rows × 16 columns

In this section, the **independent variables X** are separated by removing the "**Diabetes_binary**" column from the DataFrame df. The dependent variable Y is then defined as the "**Diabetes_binary**" column from the same DataFrame. Finally, the independent variables X are displayed. This process essentially separates the predictor variables from the target variable for further analysis.

TREATMENT OF IMBALANCE DATA AND DATA SCALING

```
Y.value_counts()
```

```
Diabetes_binary
0    194377
1     35097
Name: count, dtype: int64
```

```
y_sm.value_counts()
```

```
Diabetes_binary
0    35097
1    35097
Name: count, dtype: int64
```

```
y_sm.shape , x_sm.shape
```

```
((70194,), (70194, 16))
```

In this section:

1. Frequency of each class in the target variable Y is counted.
2. NearMiss resampling technique is initialized with version 1 and 10 nearest neighbors.
3. Resampling is applied to the feature matrix X and the target variable Y, resulting in resampled feature matrix x_sm and resampled target variable y_sm.
4. The shapes of the resampled matrices are printed, indicating the number of samples generated.
5. Counts of each class in the resampled target variable y_sm are displayed, showing balanced distribution.
6. Data is split into training and testing sets with a test size of 0.3 and using random state 37

```
scalar = StandardScaler()
X_train = scalar.fit_transform(X_train)
X_test = scalar.fit_transform(X_test)
```

In this process, the features in the training set (X_train) are standardized using StandardScaler, which calculates the mean and standard deviation of each feature and scales them accordingly. The same scaling parameters are then applied to standardize the testing set (X_test) to maintain consistency in feature scaling between the training and testing data.

05

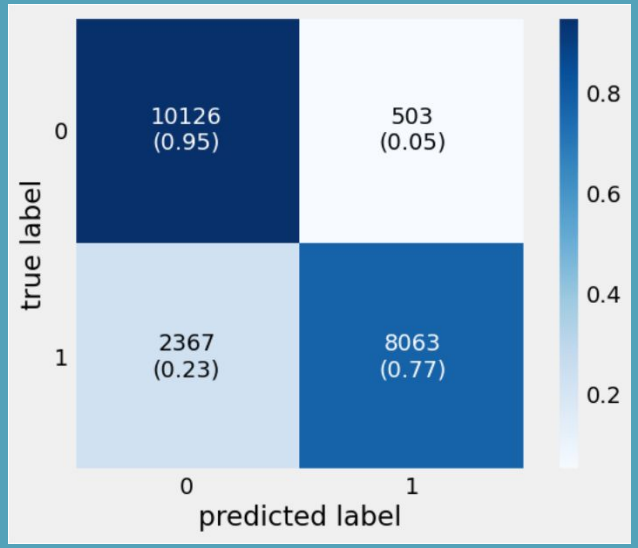
MACHINE LEARNING MODEL

The five models tested are Support Vector Machine (SVM), XGBoost, Random Forest, Naive Bayes, and Artificial Neural Network (ANN)

SVM

SVM Evaluation
Training set score: 0.8646
Test set score: 0.8637
Mean Squared Error : 0.1362837741583171
Root Mean Squared Error : 0.36916632316385134

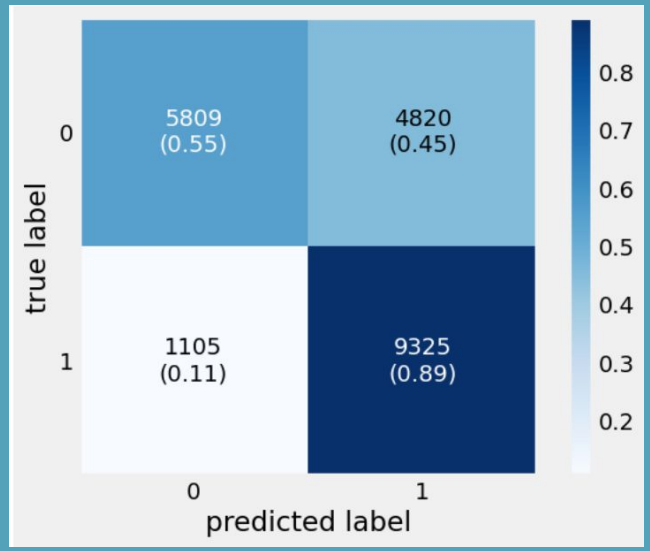
	precision	recall	f1-score	support
0	0.81	0.95	0.88	10629
1	0.94	0.77	0.85	10430
accuracy			0.86	21059
macro avg	0.88	0.86	0.86	21059
weighted avg	0.88	0.86	0.86	21059



XGBOOST

XGBoost Evaluation
Training set score: 0.8725
Test set score: 0.7186
Mean Squared Error : 0.28135239090175224
Root Mean Squared Error : 0.5304266121734016

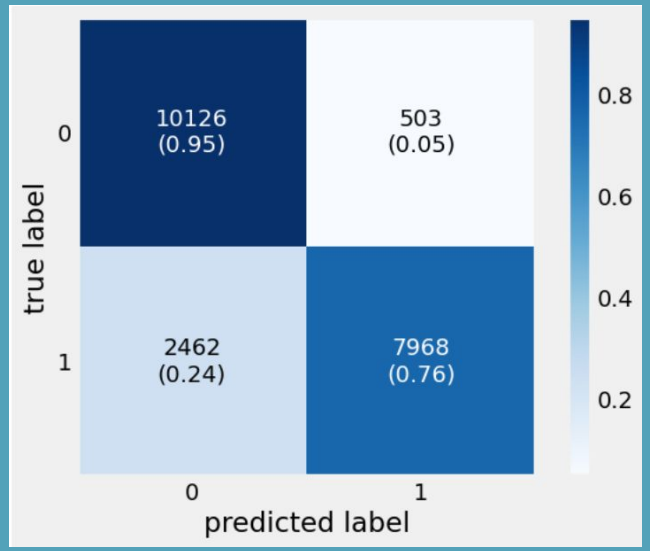
	precision	recall	f1-score	support
0	0.84	0.55	0.66	10629
1	0.66	0.89	0.76	10430
accuracy			0.72	21059
macro avg	0.75	0.72	0.71	21059
weighted avg	0.75	0.72	0.71	21059



RANDOM FOREST EVALUATION

Random Forest Evaluation
Training set score: 0.8689
Test set score: 0.8592
Mean Squared Error : 0.1407949095398642
Root Mean Squared Error : 0.37522647766364275

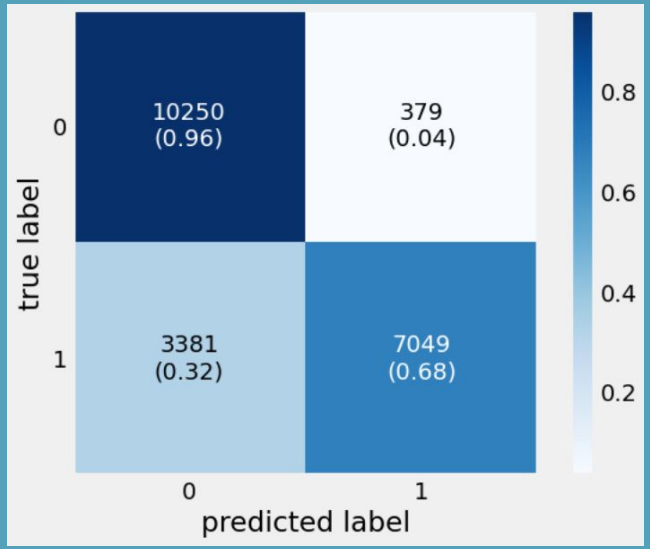
	precision	recall	f1-score	support
0	0.80	0.95	0.87	10629
1	0.94	0.76	0.84	10430
accuracy			0.86	21059
macro avg	0.87	0.86	0.86	21059
weighted avg	0.87	0.86	0.86	21059



NAIVE BAYES

Naive Bayes Evaluation
Training set score: 0.8203
Test set score: 0.8215
Mean Squared Error : 0.178545989838074
Root Mean Squared Error : 0.42254702677698963

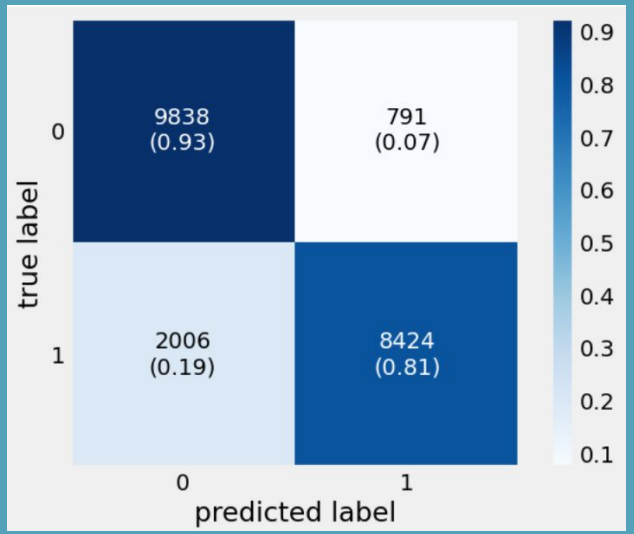
	precision	recall	f1-score	support
0	0.75	0.96	0.85	10629
1	0.95	0.68	0.79	10430
accuracy			0.82	21059
macro avg	0.85	0.82	0.82	21059
weighted avg	0.85	0.82	0.82	21059



ANN

```
ANN Evaluation
Training set accuracy: 0.8737
Test set accuracy: 0.8672
659/659 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
Mean Squared Error: 0.1328
Root Mean Squared Error: 0.3644
```

	precision	recall	f1-score	support
0	0.83	0.93	0.88	10629
1	0.91	0.81	0.86	10430
accuracy			0.87	21059
macro avg	0.87	0.87	0.87	21059
weighted avg	0.87	0.87	0.87	21059



TUNING BEST MODEL (ANN)

```
Best: 0.869095 using {'batch_size': 32, 'epochs': 20, 'optimizer': 'adam'}
0.864414 (0.001805) with: {'batch_size': 16, 'epochs': 5, 'optimizer': 'adam'}
0.865045 (0.002918) with: {'batch_size': 16, 'epochs': 5, 'optimizer': 'rmsprop'}
0.867549 (0.002072) with: {'batch_size': 16, 'epochs': 10, 'optimizer': 'adam'}
0.868546 (0.002033) with: {'batch_size': 16, 'epochs': 10, 'optimizer': 'rmsprop'}
0.866714 (0.002635) with: {'batch_size': 16, 'epochs': 15, 'optimizer': 'adam'}
0.864883 (0.002567) with: {'batch_size': 16, 'epochs': 15, 'optimizer': 'rmsprop'}
0.867630 (0.001034) with: {'batch_size': 16, 'epochs': 20, 'optimizer': 'adam'}
0.866083 (0.001016) with: {'batch_size': 16, 'epochs': 20, 'optimizer': 'rmsprop'}
0.864191 (0.002736) with: {'batch_size': 32, 'epochs': 5, 'optimizer': 'adam'}
0.865554 (0.002384) with: {'batch_size': 32, 'epochs': 5, 'optimizer': 'rmsprop'}
0.866531 (0.001015) with: {'batch_size': 32, 'epochs': 10, 'optimizer': 'adam'}
0.867121 (0.003037) with: {'batch_size': 32, 'epochs': 10, 'optimizer': 'rmsprop'}
0.867976 (0.001830) with: {'batch_size': 32, 'epochs': 15, 'optimizer': 'adam'}
0.866796 (0.000456) with: {'batch_size': 32, 'epochs': 15, 'optimizer': 'rmsprop'}
0.869095 (0.000868) with: {'batch_size': 32, 'epochs': 20, 'optimizer': 'adam'}
0.867284 (0.000350) with: {'batch_size': 32, 'epochs': 20, 'optimizer': 'rmsprop'}
0.863458 (0.001802) with: {'batch_size': 64, 'epochs': 5, 'optimizer': 'adam'}
0.861545 (0.003583) with: {'batch_size': 64, 'epochs': 5, 'optimizer': 'rmsprop'}
0.867447 (0.001233) with: {'batch_size': 64, 'epochs': 10, 'optimizer': 'adam'}
0.866368 (0.002242) with: {'batch_size': 64, 'epochs': 10, 'optimizer': 'rmsprop'}
0.866022 (0.002494) with: {'batch_size': 64, 'epochs': 15, 'optimizer': 'adam'}
0.868017 (0.001457) with: {'batch_size': 64, 'epochs': 15, 'optimizer': 'rmsprop'}
0.868444 (0.001720) with: {'batch_size': 64, 'epochs': 20, 'optimizer': 'adam'}
0.866877 (0.003579) with: {'batch_size': 64, 'epochs': 20, 'optimizer': 'rmsprop'}
0.861789 (0.002680) with: {'batch_size': 128, 'epochs': 5, 'optimizer': 'adam'}
0.861463 (0.001990) with: {'batch_size': 128, 'epochs': 5, 'optimizer': 'rmsprop'}
0.866165 (0.002797) with: {'batch_size': 128, 'epochs': 10, 'optimizer': 'adam'}
0.866246 (0.002539) with: {'batch_size': 128, 'epochs': 10, 'optimizer': 'rmsprop'}
0.865371 (0.001472) with: {'batch_size': 128, 'epochs': 15, 'optimizer': 'adam'}
0.867549 (0.001570) with: {'batch_size': 128, 'epochs': 15, 'optimizer': 'rmsprop'}
0.868057 (0.001252) with: {'batch_size': 128, 'epochs': 20, 'optimizer': 'adam'}
0.866816 (0.001847) with: {'batch_size': 128, 'epochs': 20, 'optimizer': 'rmsprop'}
```

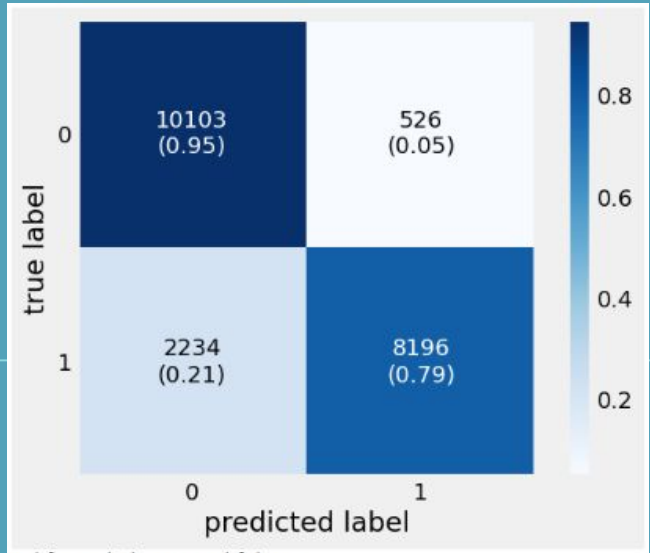
In this section, a grid search cross-validation is performed to optimize the hyperparameters of an Artificial Neural Network (ANN) model. The hyperparameters tested include different optimizers (Adam and RMSprop), batch sizes (16, 32, 64, and 128), and numbers of epochs (5, 10, 15, and 20). The best performing combination of hyperparameters is determined based on the highest accuracy score obtained during cross-validation. The grid search results show the mean test accuracy along with the standard deviation for each combination of hyperparameters tested. The best performing combination is identified as using Adam optimizer with a batch size of 32 and 20 epochs, achieving an accuracy of 86.91%.

RUNNING TUNING MODEL ANN

```
Epoch 1/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - accuracy: 0.8170 - loss: 0.3955 - val_accuracy: 0.8559 - val_loss: 0.3294
Epoch 2/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8589 - loss: 0.3266 - val_accuracy: 0.8645 - val_loss: 0.3208
Epoch 3/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8655 - loss: 0.3157 - val_accuracy: 0.8653 - val_loss: 0.3136
Epoch 4/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8673 - loss: 0.3129 - val_accuracy: 0.8654 - val_loss: 0.3116
Epoch 5/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8694 - loss: 0.3092 - val_accuracy: 0.8697 - val_loss: 0.3074
Epoch 6/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8734 - loss: 0.3026 - val_accuracy: 0.8643 - val_loss: 0.3120
Epoch 7/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8704 - loss: 0.3024 - val_accuracy: 0.8687 - val_loss: 0.3050
Epoch 8/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8728 - loss: 0.3013 - val_accuracy: 0.8701 - val_loss: 0.3004
Epoch 9/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 6s 2ms/step - accuracy: 0.8747 - loss: 0.2953 - val_accuracy: 0.8707 - val_loss: 0.3000
Epoch 10/10
1536/1536 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8713 - loss: 0.2981 - val_accuracy: 0.8689 - val_loss: 0.3009
```

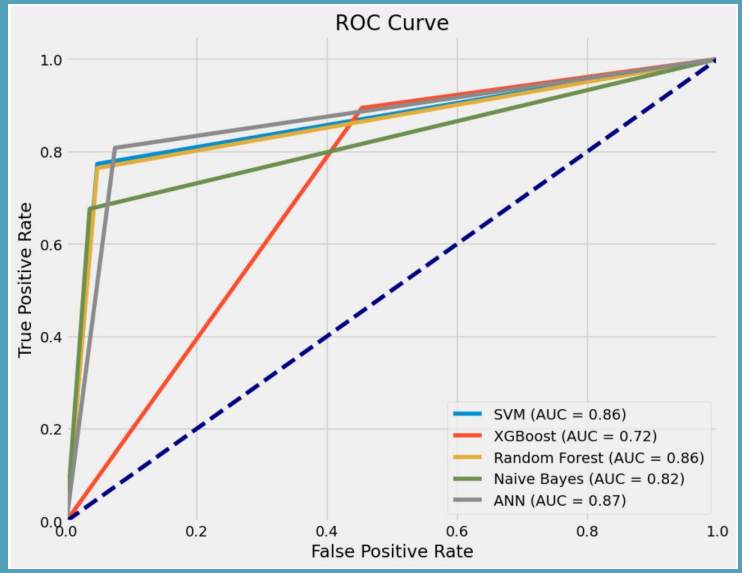
The Artificial Neural Network (ANN) model attained a training accuracy of 87.49% and a test accuracy of 86.89%, suggesting robust predictive capability. However, slight overfitting was observed as the training accuracy surpassed the test accuracy, indicating potential room for model refinement.

```
Training set accuracy: 0.8749
Test set accuracy: 0.8689
659/659 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
Mean Squared Error: 0.1311
Root Mean Squared Error: 0.3620
              precision    recall  f1-score   support

           0       0.82      0.95      0.88     10629
           1       0.94      0.79      0.86     10430

    accuracy                           0.87     21059
   macro avg       0.88      0.87      0.87     21059
weighted avg       0.88      0.87      0.87     21059
```




```
Model saved at: ann2_model.keras
```

RUNNING TUNING MODEL ANN



	Model	Training set accuracy	Test set accuracy	MSE	RMSE
4	ANN	0.873675	0.867183	0.132817	0.364441
0	SVM	0.864618	0.863716	0.136284	0.369166
2	Random Forest	0.868892	0.859205	0.140795	0.375226
3	Naive Bayes	0.820250	0.821454	0.178546	0.422547
1	XGBoost	0.872494	0.718648	0.281352	0.530427

The Area under the Curve (AUC) is a metric used to evaluate the performance of a classification model. It measures the ability of the model to distinguish between positive and negative classes across various thresholds. Here's the interpretation of the AUC scores for each model:

1. SVM (Support Vector Machine): The SVM model has an AUC of 0.8628. This indicates that the SVM model performs relatively well in distinguishing between positive and negative classes, with an overall good discriminatory ability.
2. XGBoost: The XGBoost model has an AUC of 0.7203. While this AUC score is lower compared to the SVM model, it still suggests that the XGBoost model has some ability to differentiate between positive and negative classes, although it may not be as strong as the SVM model.
3. Random Forest: The Random Forest model achieves an AUC of 0.8583. This indicates that the Random Forest model performs well in terms of discriminatory ability, similar to the SVM model, with a relatively high AUC score.
4. Naive Bayes: The Naive Bayes model has an AUC of 0.8201. While this AUC score is lower compared to the SVM and Random Forest models, it still suggests that the Naive Bayes model has a decent ability to distinguish between positive and negative classes.
5. ANN (Artificial Neural Network): The ANN model achieves an AUC of 0.8666. This indicates that the ANN model performs relatively well in distinguishing between positive and negative classes, with a high discriminatory ability similar to the SVM and Random Forest models.

In summary, the SVM, Random Forest, and ANN models demonstrate relatively strong discriminatory abilities, as indicated by their higher AUC scores, while the XGBoost and Naive Bayes models also show reasonable performance, albeit with slightly lower AUC scores.

RUNNING TUNING MODEL ANN

```
Enter value for HighBP (0 for No, 1 for Yes): 1
Enter value for HighChol (0 for No, 1 for Yes): 1
Enter value for BMI (Numeric value, based on formula): 25
Enter value for Smoker (0 for No, 1 for Yes): 1
Enter value for Stroke (0 for No, 1 for Yes): 1
Enter value for HeartDiseaseorAttack (0 for No, 1 for Yes): 1
Enter value for PhysActivity (0 for No, 1 for Yes): 0
Enter value for HvyAlcoholConsump (0 for No, 1 for Yes): 0
Enter value for Sex (0 for Female, 1 for Male): 1
Enter value for GenHlth (Range: 1-5): 3
Enter value for MentHlth (Numeric value between 0-30): 2
Enter value for PhysHlth (Numeric value between 0-30): 2
Enter value for DiffWalk (0 for No, 1 for Yes): 1
Enter value for Age (Range: 1-13): 4
Enter value for Education (Range: 1-6): 3
Enter value for Income (Range: 1-8): 3
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 309ms/step
```

	HighBP	HighChol	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	HvyAlcoholConsump	Sex	GenHlth	MentHlth	PhysHlth	DiffWalk	Age	Education	Income
0	1.0	1.0	25.0	1.0	1.0	1.0	0.0	0.0	1.0	3.0	2.0	2.0	1.0	4.0	3.0	3.0

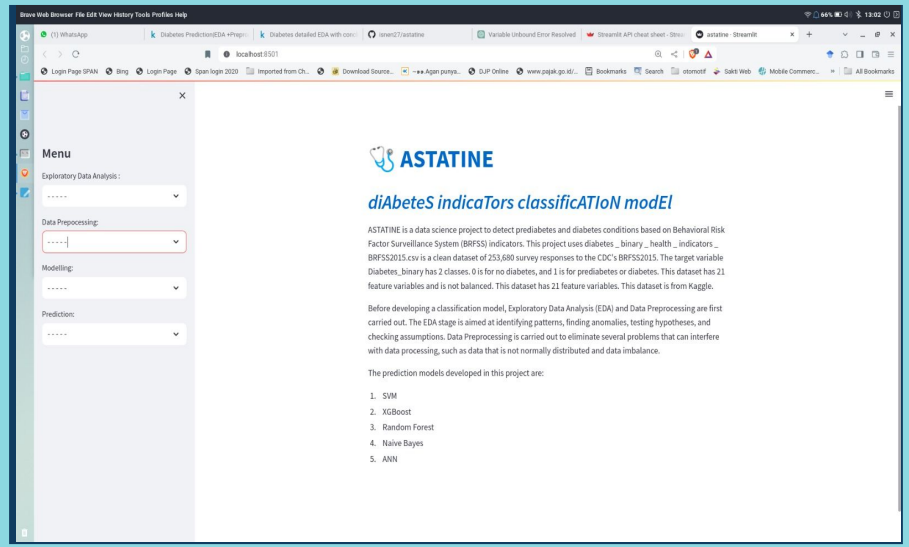
```
Based on our research model, it is predicted that you have a Prediabetes/Diabetes status. We recommend you see a doctor soon.
```

The code successfully allows for input of health feature values from users, then predicts the likelihood of prediabetes or diabetes based on a previously trained artificial neural network model. The prediction results are displayed to users along with a recommendation to consult a doctor if there is a risk of diabetes detected.

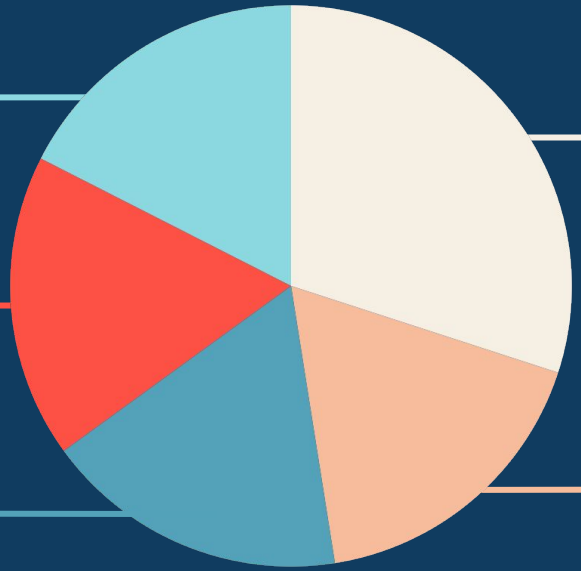
06

DEPLOYMENT MODEL

https://astatine.streamlit.app/



CONTRIBUTION IN PERCENTAGES



Tabita Mora Kristina
17,5%

Sherllya Rebecca Ezra
17,5%

Muhammad Askar F.
17,5%

Isnen Hadi Al-Ghozali
30,0%

Khuzil Afwa
17,5%

JOB LIST

Final Project							
		Keterangan	Kak Isnen	Kak Tabita	Kak Askar	Kak Afwa	Kak Ezra
	Google Colab	**19/04/2024**					
	EDA	data info, data describe, missing value, visualisasi data		X	X		
	Data pre-processing	regularisasi data, cek outlier, distribusi data, data anomali, korelasi, feature selection		X	X		
	Deadline	**15/04/2024**					
	Model training	ML Traditional :kNN, Decision Tree (C.45), Random Forest ML Deep Learning eksperimen untuk Boosting atau majority voting tambahkan algoritma Apriori untuk menunjukkan gejala/kondisi yg dapat menyebabkan diabetes	X			X	
	Model evaluation	convution metric (recall, precision, accuracy, F-1 Score), ROC, AUC					X
	Streamlit						
	Modul Inisiasi		X	X			
	Modul EDA		X		X		
	Modul Dara Prepocessing		X			X	
	Deadline	**24/04/2023**					
	Modul Model Training		X				X
	Modul Evaluation Model		X				
	Deadline	**25/04/2023**					
	Presentasi	**Buat slide untuk project**	**X**		**X**		
	Deadline	**28/04/2024**					

THANK YOU

– ORION TEAM