AI BASED DIABETES PREDICTION SYSTEM

PHASE-5 Documentation & Submission

Introduction:

Type 2 diabetes (T2D) is defined as hyperglycemia caused by abnormal insulin action, resulting in metabolic malfunction in energy generation from ingested glucose (Association Citation2020). T2D is achronic disease that is increasing the levels of mortality, morbidity, severe complications, and health expenditure** (Khan et al. Citation2020; Zhou et al. Citation2016). Unfortunately, despite this growing concern (Khan et al. Citation2020), multiple studies have reported that reducing blood glucose levels is challenging (Adu et al. Citation2019; Carls et al. Citation2017; Schlender et al. Citation2017). The rate of diabetes diagnosis in the Republic of Korea (ROK) is rapidly increasing, similar to global trends. Currently, there is no permanent treatment for T2D. The World Health Organization (WHO) has highlighted the implications of the early detection and screening of noncommunicable diseases (NCDs) as cost-effective interventions (WHO(World Health Organization).This study aimed to develop various ML prediction methods for the future state of prediabetes using current real-world data from Korean medical examination records. Furthermore, we aimed to identify crucial risk factors for prediabetes and diabetes.

Materials and Methods:

- **Ethics Declarations:**

  Clinical and laboratory variables were collected from the clinical data warehouse platform and the electronic medical records in Ulsan University Hospital. This study is categorized under the records-based retrospective research; therefore, informed consent by participants was not required. The study was reviewed and the protocol approved by the Institutional Human Experimentation Committee Review Board of Ulsan University Hospital, Republic

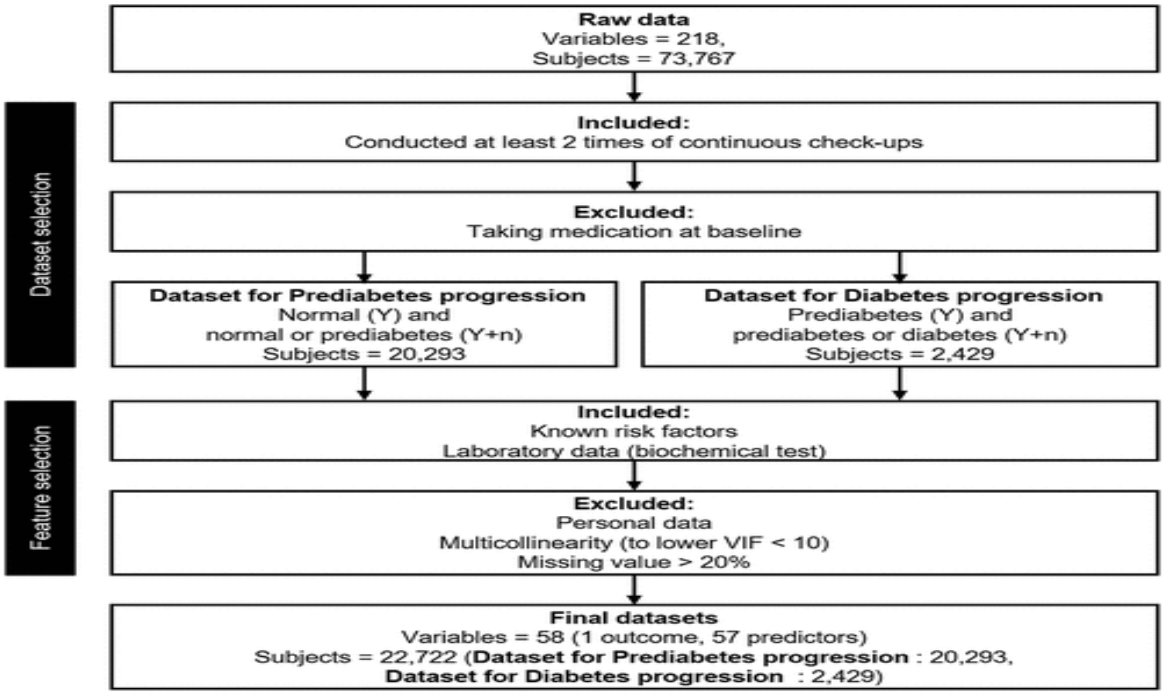
of Korea (UUH 2020-09-003). The study was conducted in accordance with the ethical standards set forth in the 1964 Declaration of Helsinki.

- **Participants:**

In this study, the dataset contained 133,387 instances collected from 73,767 subjects, and each instance had 218 variables. Variables that included the subjective opinions of the medical staff were excluded. From the total data, 57 variables were ultimately selected, excluding variables with missing values amounting to 20% or more, and variables with suspected multicollinearity.

- **Study Design:**

shows the variables used in the study and the process of selecting study subjects, which was as follows. First, 33,784 subjects who had undergone two or more health checkups were selected. Second, fasting blood glucose was considered to be the diabetes screening criterion: when it was >125 mg/dL, the patient was diagnosed as diabetic, and when it was <100 mg/dL, the patient was considered to be normal and borderline, and was classified as prediabetic. Finally, 33,403 participants met the inclusion criteria. In this study, the datasets from normal to prediabetic, and from prediabetic to diabetic were analyzed separately.



- **Cost-Sensitive Learning:**

ML models were appropriately predicted when the ratios of each class were similar. If the class ratio was imbalanced, the algorithm learned to predict most of the classes in a biased manner. Consequently, many studies have been conducted to solve the problem of class imbalance .In this study, we used cost-sensitive learning. Increased cost is the main result of wrong predictions. The goal of cost-sensitive learning is to minimize the cost of the model on the dataset. Therefore, the classifier is set to respond sensitively to the minority class by assigning a low weight to the majority class while setting high weights for the misclassification of the minority class without generating the data to reduce the bias with regard to the majority class.

Code:

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from collections import Counter

import os

from sklearn.preprocessing import QuantileTransformer

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score

from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier,
VotingClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.svm import SVC
```

```
from sklearn.model_selection import GridSearchCV, cross_val_score, StratifiedKFold, learning_curve, train_test_split

df = pd.read_csv("../input/pima-indians-diabetes-database/diabetes.csv")

df.info()

df.head()

df.isnull().sum()

df['Glucose'] = df['Glucose'].replace(0, df['Glucose'].mean())

df['BloodPressure'] = df['BloodPressure'].replace(0, df['BloodPressure'].mean()) # There are 35 records with 0 BloodPressure in dataset

df['BMI'] = df['BMI'].replace(0, df['BMI'].median())

df['SkinThickness'] = df['SkinThickness'].replace(0, df['SkinThickness'].median())

df['Insulin'] = df['Insulin'].replace(0, df['Insulin'].median())

df.describe()df.describe()

plt.figure(figsize=(13,10))

sns.heatmap(df.corr(),annot=True, fmt = ".2f", cmap = "coolwarm")

plt.figure(figsize=(13,6))

g = sns.kdeplot(df["Pregnancies"][df["Outcome"] == 1],

    color="Red", shade = True)

g = sns.kdeplot(df["Pregnancies"][df["Outcome"] == 0],

    ax =g, color="Green", shade= True)

g.set_xlabel("Pregnancies")

g.set_ylabel("Frequency")

g.legend(["Positive","Negative"])

sns.countplot('Outcome', data = df)

plt.figure(figsize=(10,6))

sns.violinplot(data=df, x="Outcome", y="Glucose",
```

```
        split=True, inner="quart", linewidth=1)
plt.figure(figsize=(13,6))
g = sns.kdeplot(df["Glucose"][df["Outcome"] == 1], color="Red", shade = True)
g = sns.kdeplot(df["Glucose"][df["Outcome"] == 0], ax =g, color="Green", shade= True)
g.set_xlabel("Glucose")
g.set_ylabel("Frequency")
g.legend(["Positive","Negative"])
plt.figure(figsize=(20,10))
sns.scatterplot(data=df, x="Glucose", y="BMI", hue="Age", size="Age")
model = RandomForestClassifier(random_state=42)
tuned_parameters = {
   'n_estimators': [200, 500],
   'max_features': ['auto', 'sqrt', 'log2'],
   'max_depth' : [4,5,6,7,8],
   'criterion' :['gini', 'entropy']
}
cv = StratifiedKFold(n_splits = 2, random_state = 1, shuffle = True)
grid_search = GridSearchCV(estimator = model, param_grid = tuned_parameters, cv = cv, scoring =
'accuracy', error_score = 0)
grid_result = grid_search.fit(x_train, y_train)
analyze_grid_result(grid_result)
y_pred = logi_result.predict(x_test)
print(classification_report(y_test, y_pred))
x_test['pred'] = y_pred
print(x_test)
```

Output:

Importing Data:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #    Column                    Non-Null Count  Dtype
---   ------                    --------------  -----
 0    Pregnancies               768 non-null    int64
 1    Glucose                   768 non-null    int64
 2    BloodPressure             768 non-null    int64
 3    SkinThickness             768 non-null    int64
 4    Insulin                   768 non-null    int64
 5    BMI                       768 non-null    float64
 6    DiabetesPedigreeFunction  768 non-null    float64
 7    Age                       768 non-null    int64
 8    Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

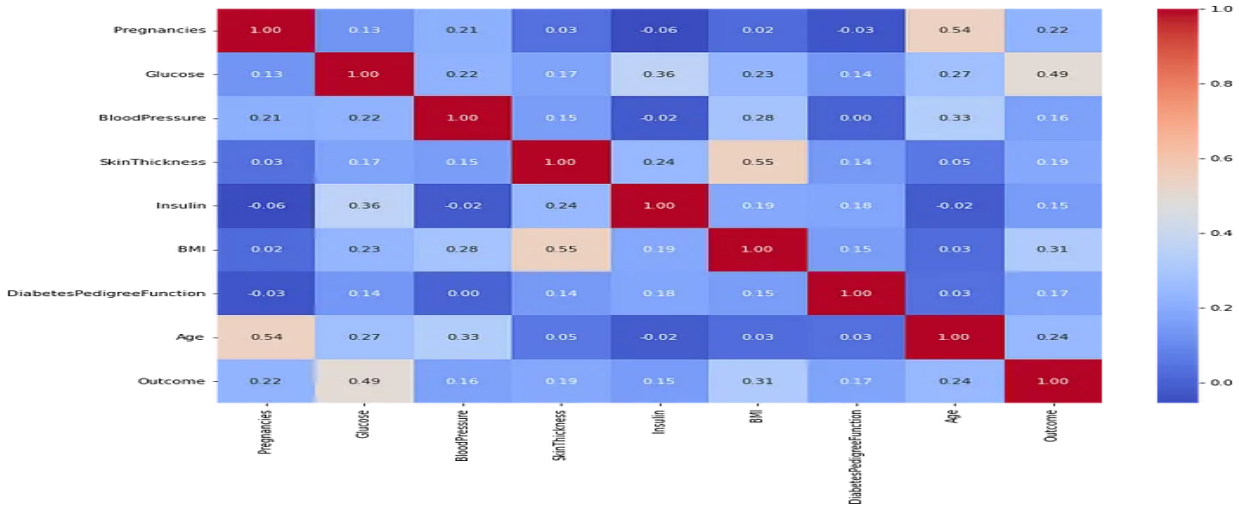
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35	30.5	33.6	0.627	50	1
1	1	85.0	66.0	29	30.5	26.6	0.351	31	0
2	8	183.0	64.0	23	30.5	23.3	0.672	32	1
3	1	89.0	66.0	23	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35	168.0	43.1	2.288	33	1

Missing Value Analysis:

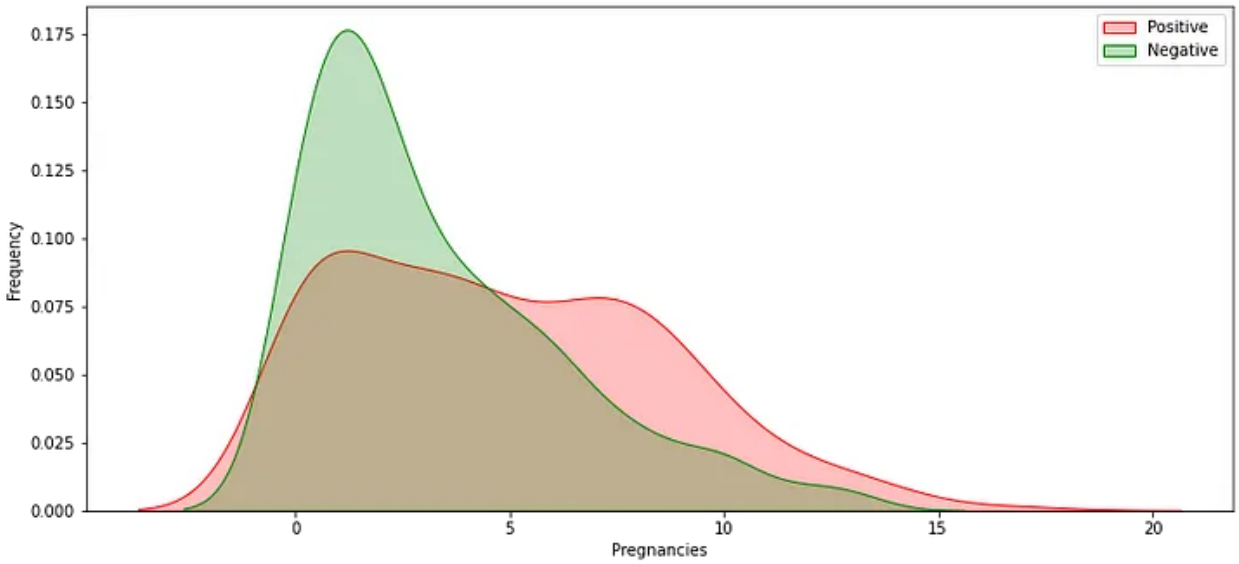
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.681605	72.254807	27.334635	94.652344	32.450911	0.471876	33.240885	0.348958
std	3.369578	30.436016	12.115932	9.229014	105.547598	6.875366	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	23.000000	30.500000	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	31.250000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

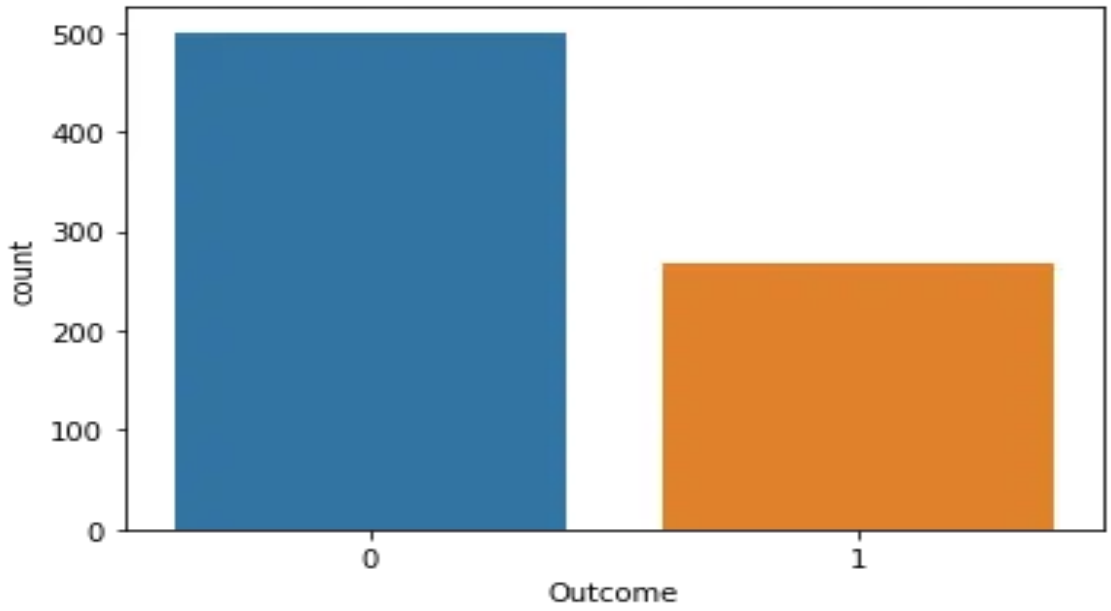
Exploratory Data Analysis:



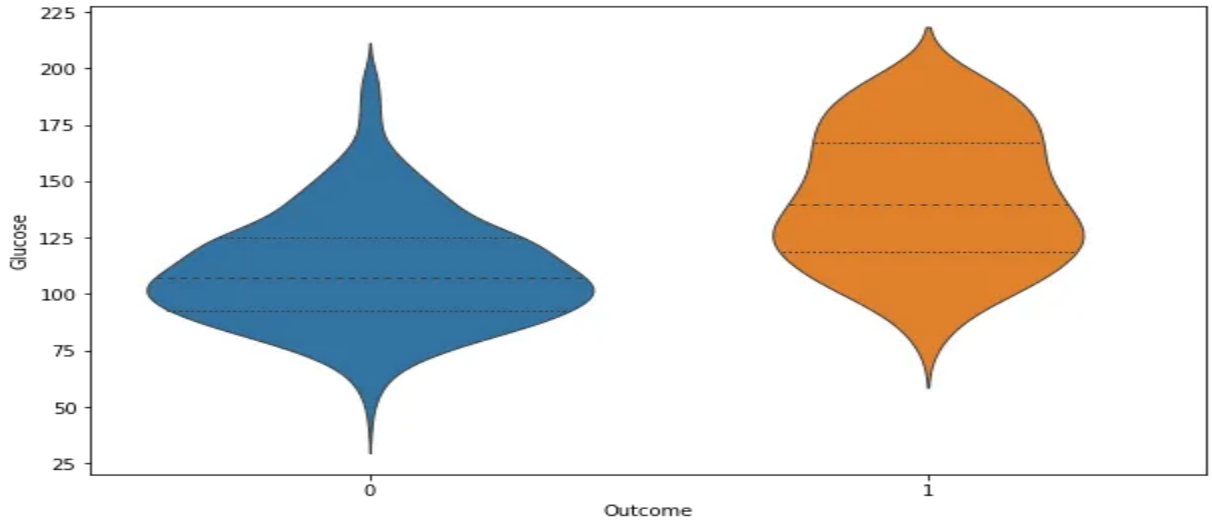
Pregnancies:

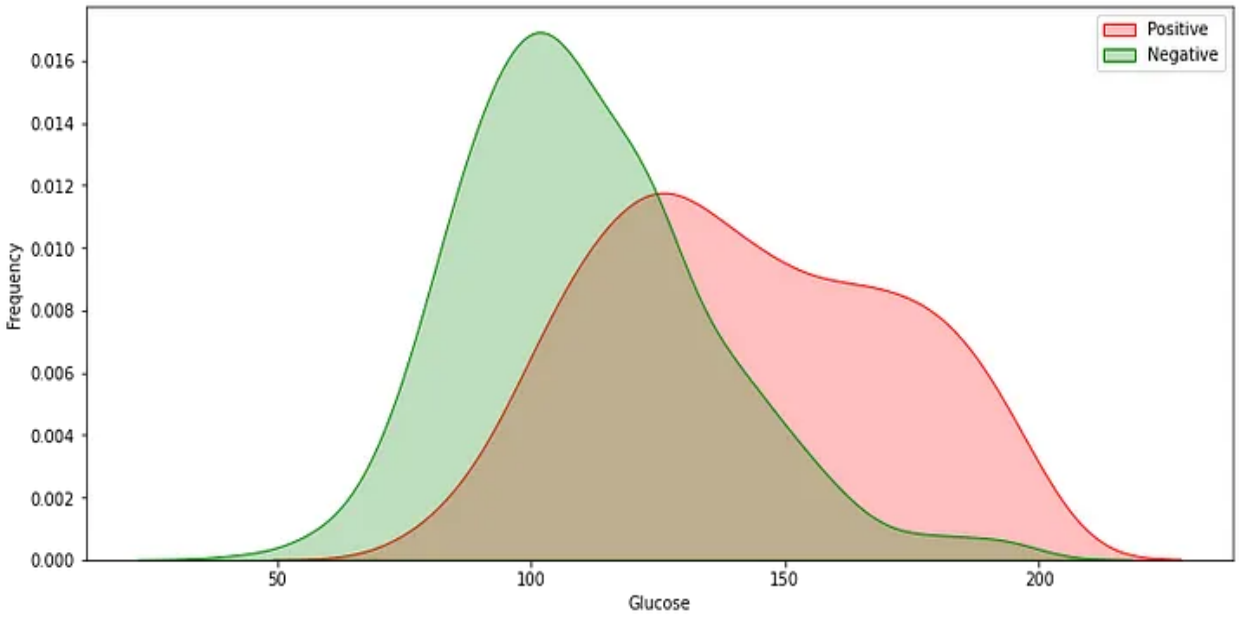


Outcome:

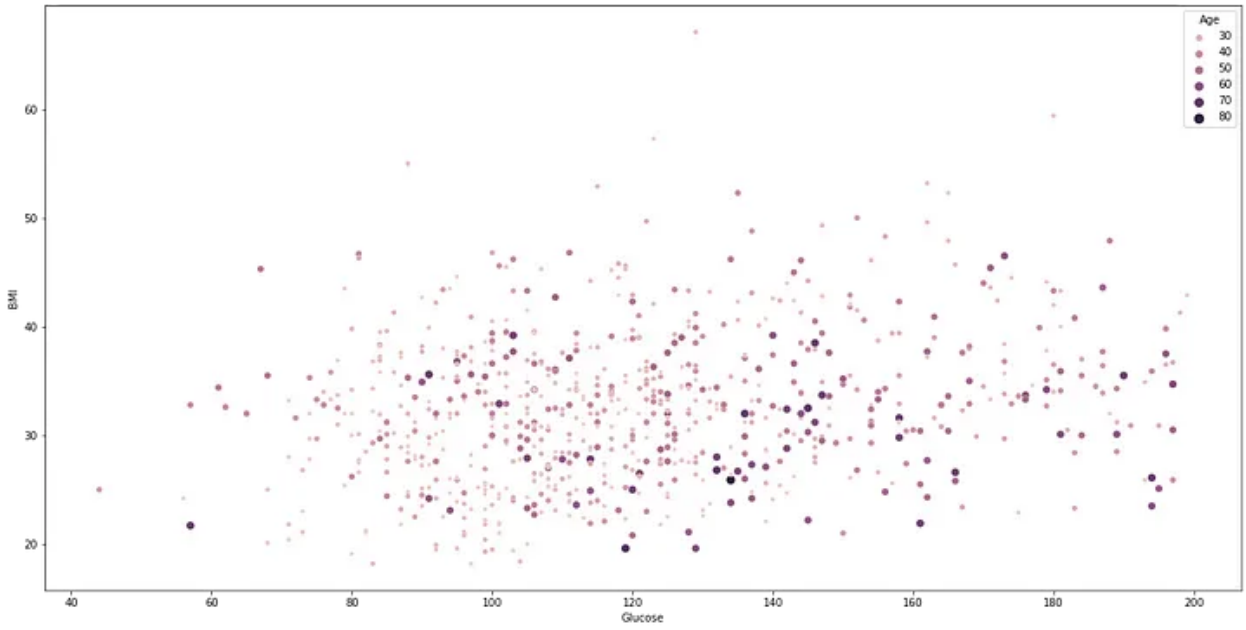


Glucose:





Explore Glucose vs BMI vs Age:



RandomForestClassifier:

```
Tuned hyperparameters: (best parameters)  {'criterion': 'entropy',
'max_depth': 5, 'max_features': 'log2', 'n_estimators': 200}
Accuracy : 0.7663648051875454
Detailed classification report:

              precision    recall  f1-score   support

           0       0.78      0.83      0.80       147
           1       0.66      0.58      0.62        83

    accuracy                           0.74       230
   macro avg       0.72      0.70      0.71       230
weighted avg       0.73      0.74      0.74       230
```

Test Predictions:

	precision	recall	f1-score	support
0	0.78	0.84	0.81	147
1	0.68	0.58	0.62	83
accuracy			0.75	230
macro avg	0.73	0.71	0.72	230
weighted avg	0.74	0.75	0.74	230

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	pred
236	7	181.0	84.0	21	192.0	35.9	0.586	51	1
715	7	187.0	50.0	33	392.0	33.9	0.826	34	1
766	1	126.0	60.0	23	30.5	30.1	0.349	47	0
499	6	154.0	74.0	32	193.0	29.3	0.839	39	1
61	8	133.0	72.0	23	30.5	32.9	0.270	39	1
...	...	...	...	...	...	...	...	...	...
189	5	139.0	80.0	35	160.0	31.6	0.361	25	0
351	4	137.0	84.0	23	30.5	31.2	0.252	30	0
120	0	162.0	76.0	56	100.0	53.2	0.759	25	1
108	3	83.0	58.0	31	18.0	34.3	0.336	25	0
637	2	94.0	76.0	18	66.0	31.6	0.649	23	0

Conclusion:

An AI-based diabetes prediction system has the potential to contribute to early detection and prevention of diabetes by analyzing various factors and patterns in data. While these systems can provide valuable insights, it's essential to consider the following:

- Data quality and representativeness are crucial for accurate predictions. Robust data collection processes and quality control measures are necessary to ensure the system's effectiveness.AI-based systems should be used as decision support tools, with medical professionals interpreting the results and making informed decisions based on their expertise. Compliance with data protection regulations and ethical guidelines is essential to maintain patient privacy and trust.Continuous improvement is necessary to enhance the system's performance and reliability over time. Regular updates, validation, and evaluation are crucial to ensure accurate predictions.