AI - BASED DIABETES PREDICTION SYSTEM

Phase 3: Development Part 1

The code starts by importing necessary libraries, including pandas, numpy, matplotlib, seaborn, and scikit-learn. It also disables warnings to prevent them from cluttering the output. Additionally, the 'pickle' library is imported for model serialization.Next, it loads a dataset ('diabetes.csv') into a Pandas DataFrame. This dataset includes various attributes such as glucose levels, BMI, blood pressure, pregnancies, skin thickness, insulin, diabetes pedigree function, age, and an outcome variable indicating whether a patient has diabetes or not.The data preprocessing step involves selecting specific features from the dataset (columns 1, 4, 5, and 7) and scaling them using Min-Max scaling to bring the values within a specified range, typically between 0 and 1. The scaled data is then split into features (X) and target labels (Y), which represent whether a patient has diabetes.The data is further split into training and testing sets using the train_test_split function from scikit-learn. The code proceeds to build a Support Vector Machine (SVM) classifier with a linear kernel using the training data.The model's accuracy is assessed on the test data, and the predicted outcomes are stored in 'Y_pred.' Additionally, the trained SVM model is serialized using 'pickle' and saved as 'model.pkl.' This allows for reusing the model without retraining it. A commented-out line demonstrates how to make predictions using the saved model.

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
dataset.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

dataset.shape

dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

dataset.describe().T

[6]:

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

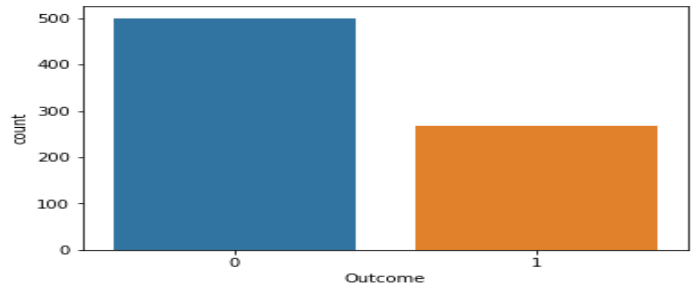
dataset.isnull().sum()

Out[7]:

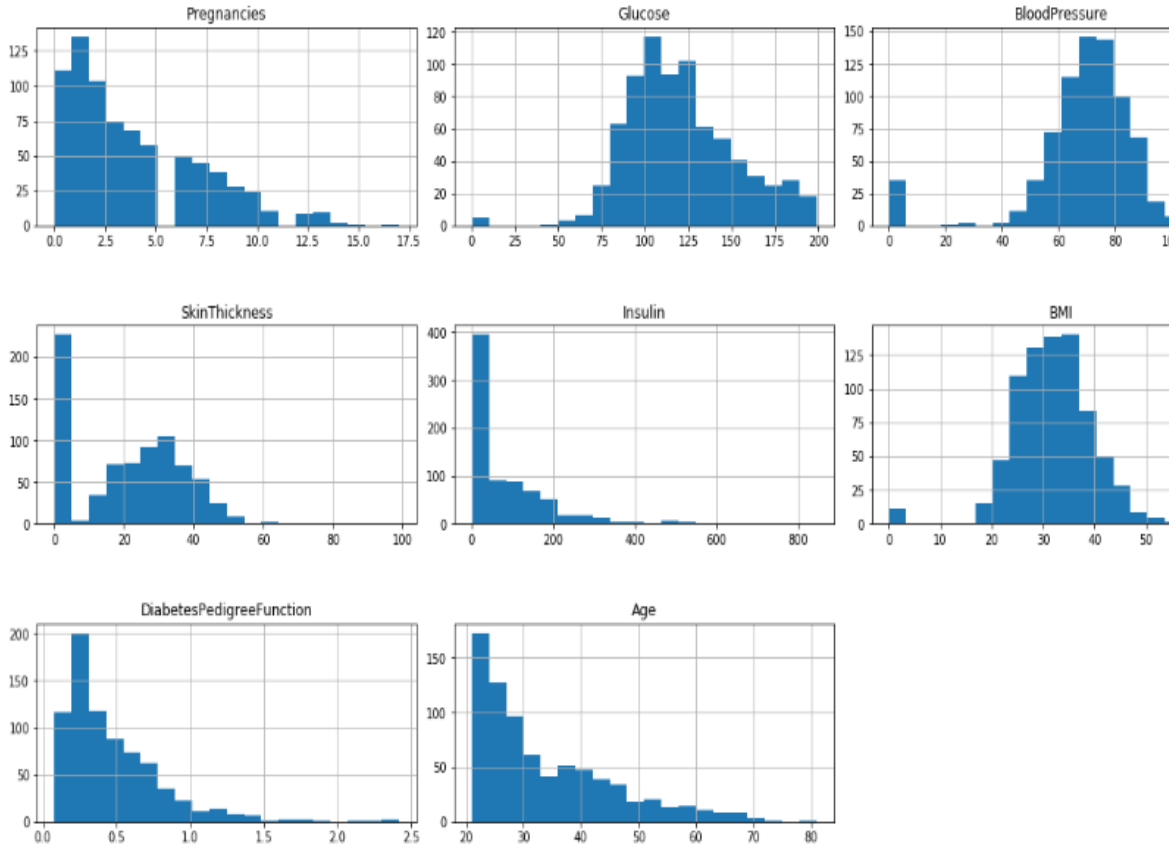
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

sns.countplot(x = 'Outcome',data = dataset)

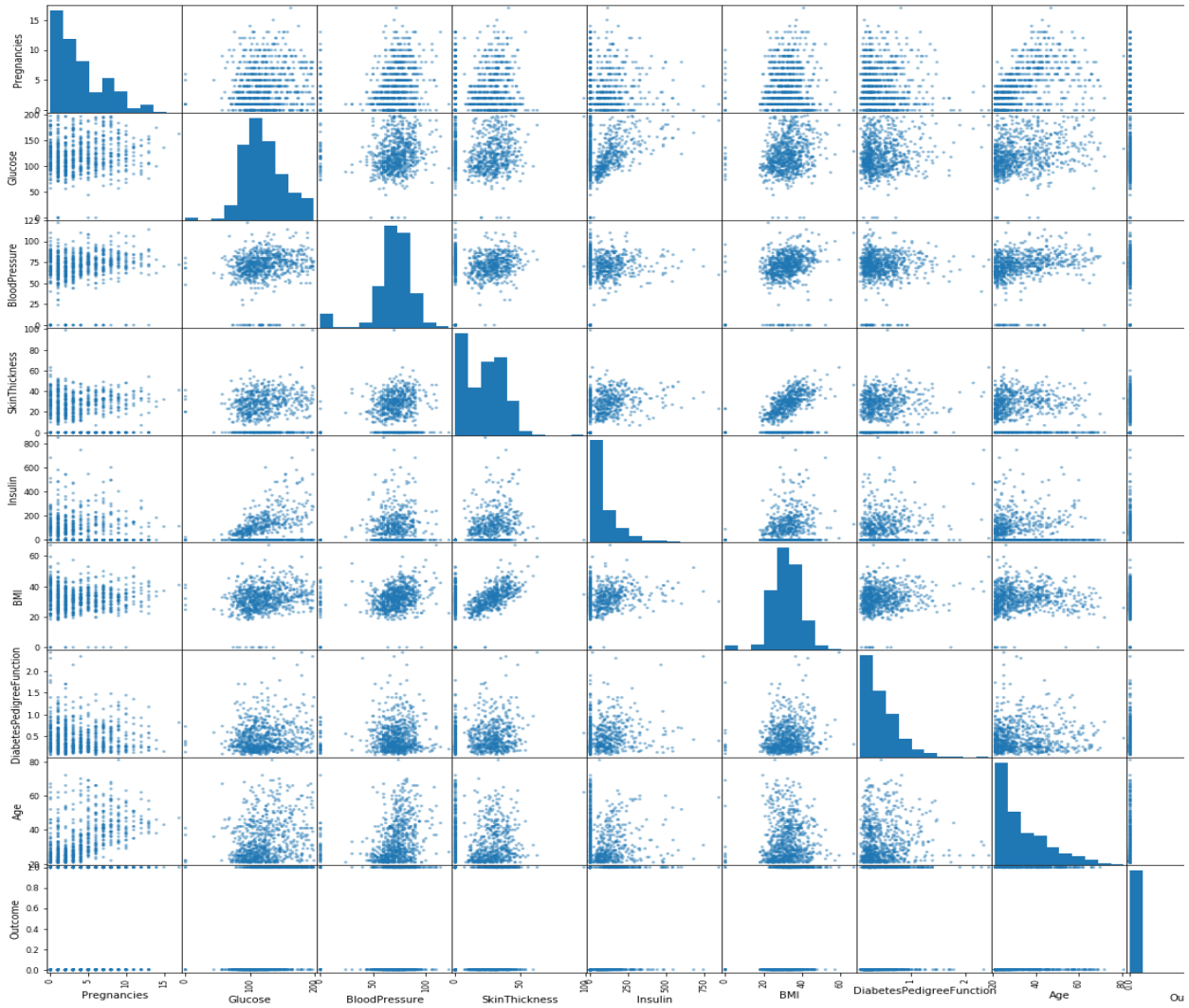
Out[8]: `<matplotlib.axes._subplots.AxesSubplot at 0x1976214b128>`

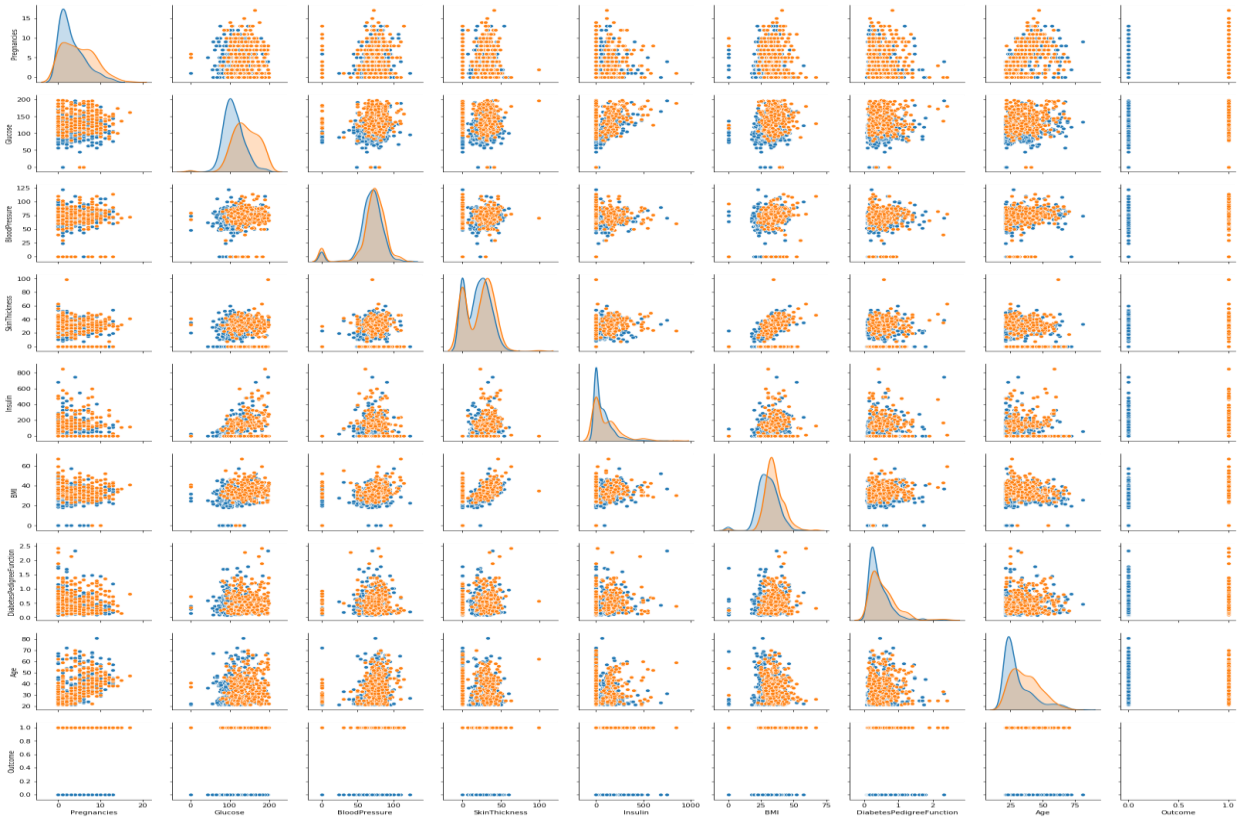


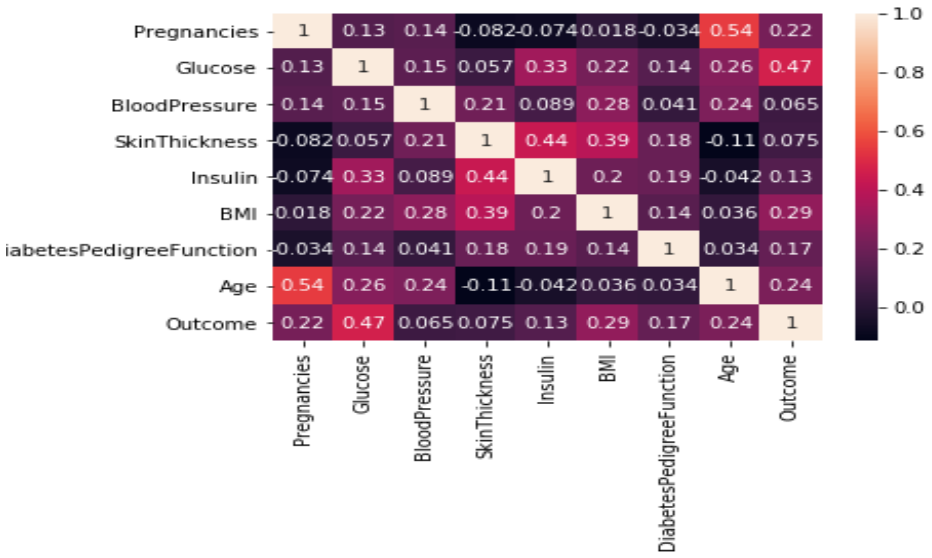
```
import itertools

col = dataset.columns[:8]
plt.subplots(figsize = (20, 15))
length = len(col)
for i, j in itertools.zip_longest(col, range(length)):
    plt.subplot((length/2), 3, j + 1)
    plt.subplots_adjust(wspace = 0.1,hspace = 0.5)
    dataset[i].hist(bins = 20)
    plt.title(i)
plt.show()
```



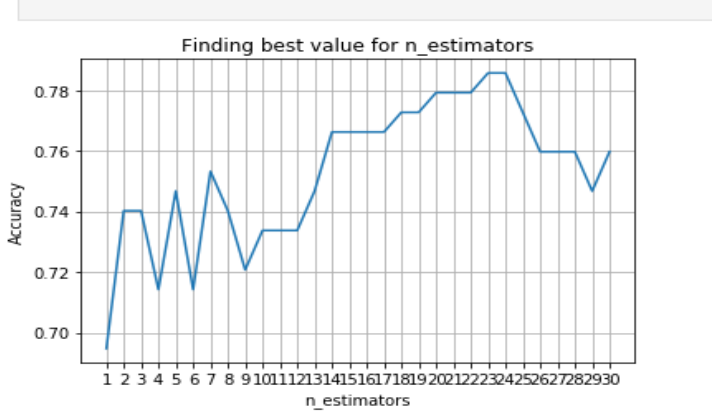

```
from pandas.tools.plotting import scatter_matrix
scatter_matrix(dataset, figsize = (20, 20));
sns.pairplot(data = dataset, hue = 'Outcome')
plt.show()
sns.heatmap(dataset.corr(), annot = True)
plt.show()
```


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome



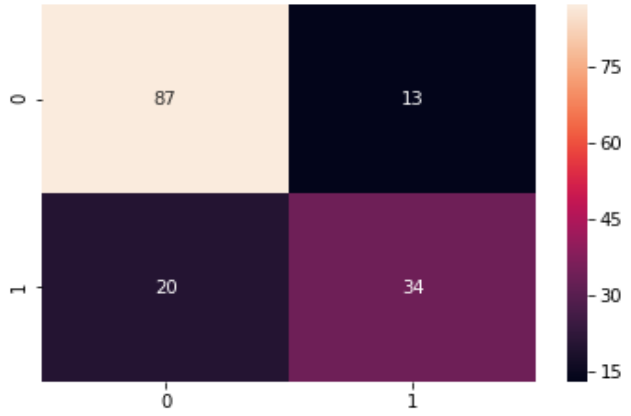

```
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier
X_axis = list(range(1, 31))
acc = pd.Series()
x = range(1,31)
for i in list(range(1, 31)):
    knn_model = KNeighborsClassifier(n_neighbors = i)
    knn_model.fit(X_train, Y_train)
    prediction = knn_model.predict(X_test)
    acc = acc.append(pd.Series(metrics.accuracy_score(prediction, Y_test)))
plt.plot(X_axis, acc)
plt.xticks(x)
plt.title("Finding best value for n_estimators")
plt.xlabel("n_estimators")
plt.ylabel("Accuracy")
plt.grid()
plt.show()
print('Highest value: ',acc.values.max())
```




```
Highest value:  0.7857142857142857
```

```
sns.heatmap(pd.DataFrame(cm), annot=True)
```

```
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x1976620cb70>
```




```
from sklearn.metrics import classification_report
print(classification_report(Y_test, Y_pred_knn))
```

	precision	recall	f1-score	support
0.0	0.81	0.87	0.84	100
1.0	0.72	0.63	0.67	54
micro avg	0.79	0.79	0.79	154
macro avg	0.77	0.75	0.76	154
weighted avg	0.78	0.79	0.78	154

DATASET LINK: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

CONCLUSION:

this code sets up a basic machine learning pipeline for diabetes prediction, including data preprocessing, model training, evaluation, and model serialization for future use. The accuracy score provides an indication of the model's performance, but more extensive evaluation and hyperparameter tuning may be necessary for a comprehensive assessment of the model's predictive capabilities.

DONED BY:

V.MUTHULAKSHMI

A.SUVETHA

M.SUBAPARVATHI

P.THEJASVANI