AI BASED DIABETES PREDICTION SYSTEM

Phase 4 – Project Development

PROJECT TITLE	AI based diabetes prediction system
SKILLS TAKEN AWAY	❖ Python script ❖ EDA ❖ UI deployment
DOMAIN	Medical
TEAM MEMBERS	K. MADHUMITHA M.PAVITHRA M.ARTHI VARSHINI S. ROSHINI V.KANAGAVALLI

INTRODUCTION:

Diabetes is a health issue that affects how your body converts food into energy. The majority of the food you consume is broken down into sugar (also known as glucose) and released into your bloodstream. When your blood sugar rises, your pancreas releases insulin. Diabetes, if not managed carefully and continuously, can cause a buildup of sugars in the blood, increasing the risk of serious consequences such as stroke and heart disease. As a result, I decided to anticipate using Machine Learning in Python.

OBJECTIVES:

- Determine whether or not a person has diabetes.
- Discover the most telling signs of diabetes and experiment with several classification methods to get the best accuracy.

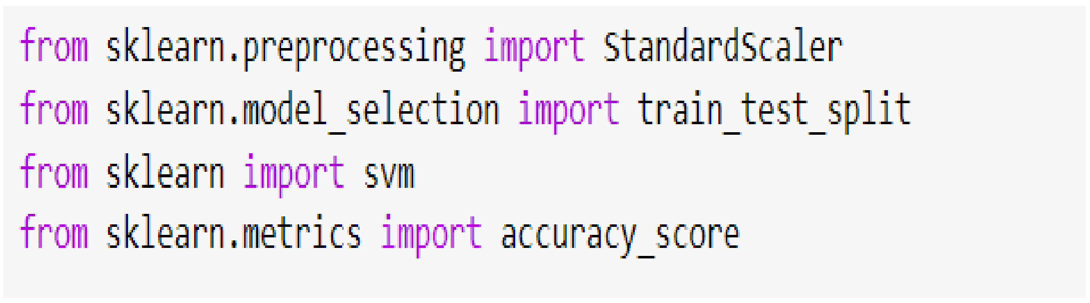
DETAILS ABOUT THE DATASET:

The datasets consist of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

- **Pregnancies**: Number of times pregnant
- **Glucose**: Plasma glucose concentration 2 hours in an oral glucose tolerance test
- **Blood Pressure**: Diastolic blood pressure (mm Hg)
- **Skin Thickness**: Triceps skin fold thickness (mm)
- **Insulin**: 2-Hour serum insulin (mu U/ml)
- **BMI**: Body mass index (weight in kg/ (height in m) ^2)
- **Diabetes Pedigree Function**: Diabetes pedigree function
- **Age**: Age (years)
- **Outcome**: Class variable (0 or 1)

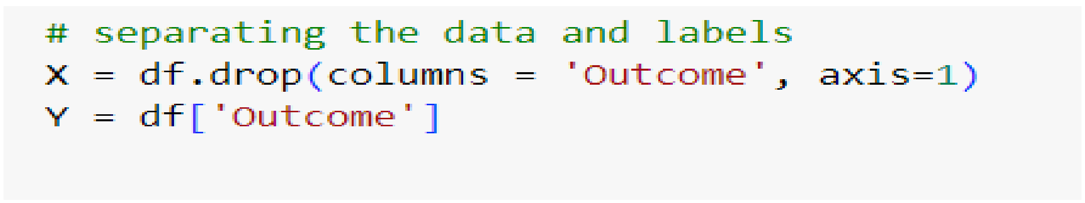
AI MODEL TRAINING AND EVALUATING:

✓ IMPORTING OTHER NECESSARY LIBRARIES:

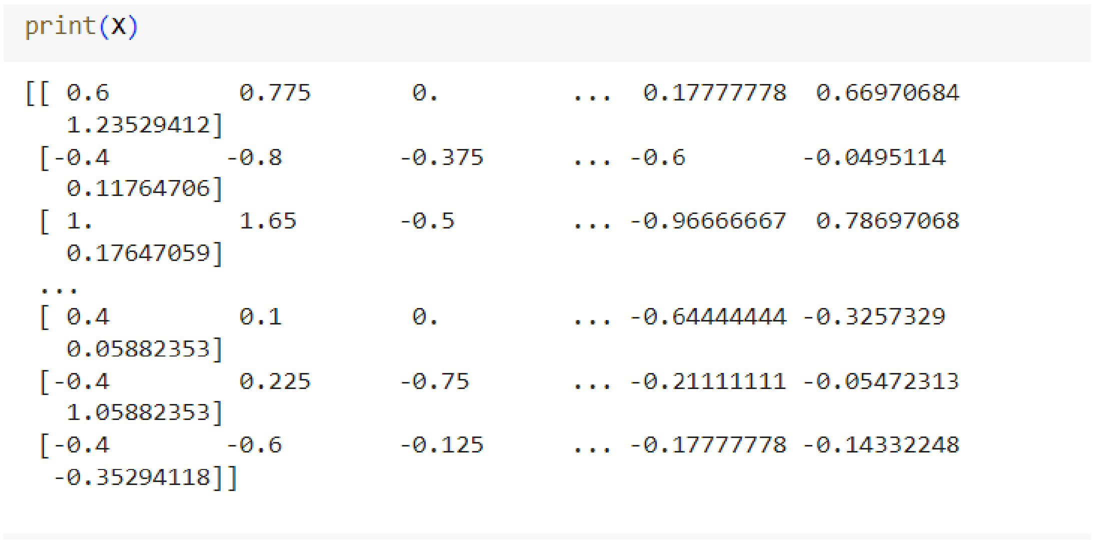


```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

✓ SEPARATING DATA AND LABELS:

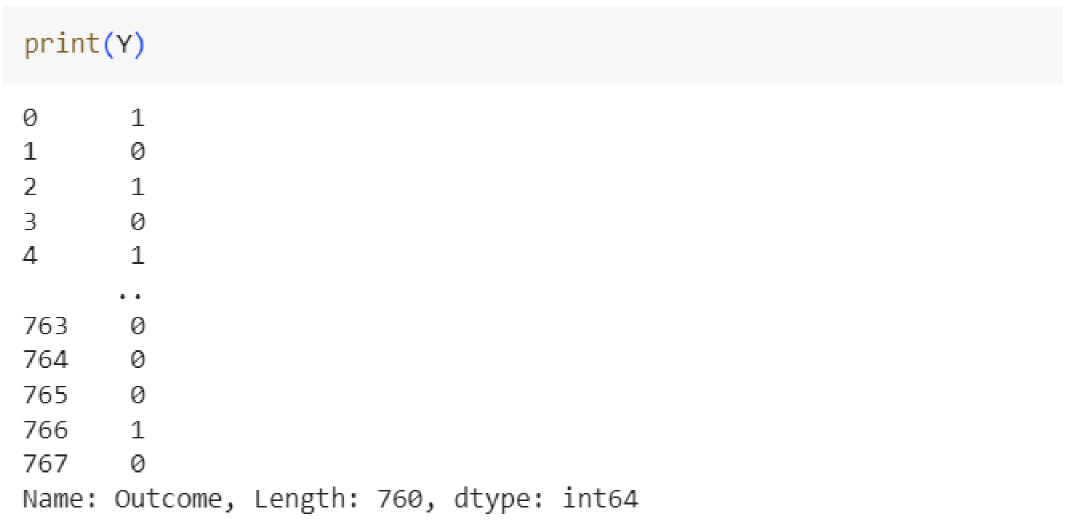


```
# separating the data and labels
X = df.drop(columns = 'Outcome', axis=1)
Y = df['Outcome']
```



```
print(X)
```

```
[[ 0.6         0.775       0.         ...  0.17777778  0.66970684
   1.23529412]
 [-0.4        -0.8        -0.375      ... -0.6        -0.0495114
   0.11764706]
 [ 1.         1.65       -0.5        ... -0.96666667  0.78697068
   0.17647059]
 ...
 [ 0.4         0.1         0.         ... -0.64444444 -0.3257329
   0.05882353]
 [-0.4         0.225      -0.75       ... -0.21111111 -0.05472313
   1.05882353]
 [-0.4        -0.6        -0.125      ... -0.17777778 -0.14332248
  -0.35294118]]
```

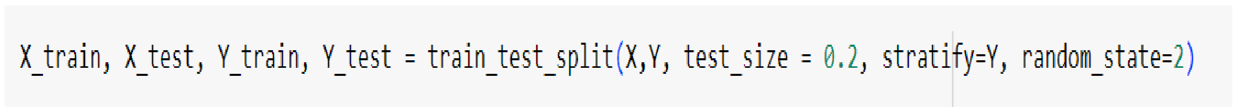


```
print(Y)
```

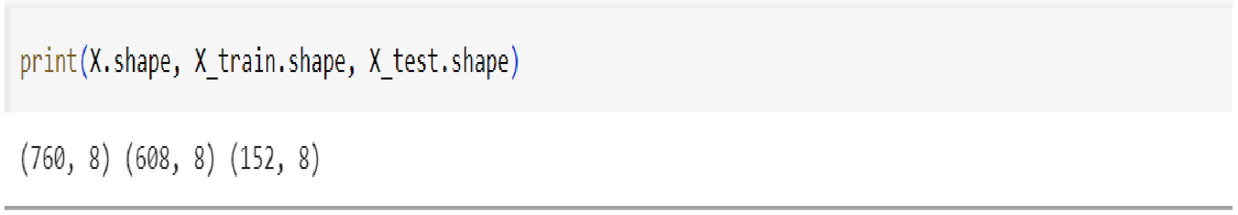
```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 760, dtype: int64
```

X represents the data and Y represents the label

✓ TRAIN TEST SPLIT:



```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)
```

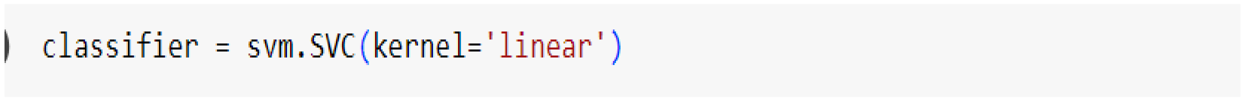


```
print(X.shape, X_train.shape, X_test.shape)
```

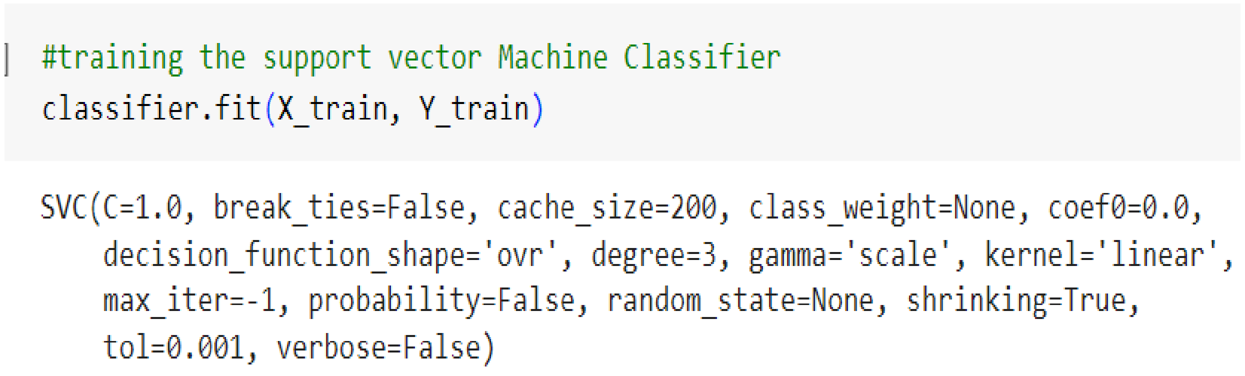
```
(760, 8) (608, 8) (152, 8)
```

✓ TRAINING THE MODEL:

Support vector Machine Classifier



```
classifier = svm.SVC(kernel='linear')
```



```
#training the support vector Machine Classifier
classifier.fit(X_train, Y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

✓MODEL EVALUATION:

Accuracy on training data:

```
# accuracy score on the training data
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy score of the training data : ', training_data_accuracy)
```

```
Accuracy score of the training data :  0.7861842105263158
```

Accuracy on test data:

```
# accuracy score on the test data
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy score of the test data : ', test_data_accuracy)
```

```
Accuracy score of the test data :  0.75
```

✓ MAKING A PREDICTIVE SYSTEM:

```
scaler = RobustScaler()
input_data = (5, 166, 72, 19, 175, 25.8, 0.587, 51)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)
scaler.fit(X_train)
std_data = scaler.transform(input_data_reshaped)
prediction = classifier.predict(std_data)
if prediction[0] == 0:
    print('The person is not diabetic')
else:
    print('The person is diabetic')
```

```
The person is diabetic
```

Thus AI model has been built and trained using machine learning algorithm and its performance have been evaluated.