diabetes\_dataset = pd.read\_csv(r"C:\Users\mohsi\IML pro\diabetes.csv")

pd.read\_csv

diabetes\_dataset.head()

diabetes\_dataset.shape

diabetes\_dataset.describe()

diabetes\_dataset['Outcome'].value\_counts()

diabetes\_dataset.groupby('Outcome').mean()

X = diabetes\_dataset.drop(columns = 'Outcome', axis=1)

Y = diabetes\_dataset['Outcome']

print(X)

print(Y)

scaler = StandardScaler()

scaler.fit(X)

StandardScaler(copy=True, with\_mean=True, with\_std=True)

standardized\_data = scaler.transform(X)

print(standardized\_data)

X = standardized\_data

Y = diabetes\_dataset['Outcome']

print(X)

print(Y)

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)

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

classifier.fit(X\_train, Y\_train)

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)

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)

input\_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input\_data to numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the array as we are predicting for one instance

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

# standardize the input data

std\_data = scaler.transform(input\_data\_reshaped)

print(std\_data)

prediction = classifier.predict(std\_data)

print(prediction)

if (prediction[0] == 0):

print('The person is not diabetic')

else:

print('The person is diabetic')