1. System name: Diabetes prediction system
2. Motivation:

   The reason why I want to create a diabetes prediction system is to improve healthcare and empower the ability that people managing their health. More and more people have diabetes in these few years, if diabetes can be detected and predicted earlier, then it can greatly reduce the health risks of patients. The diabetes prediction system can help doctors and professional medical staff to analyze the risks of the disease by the patient's clinical and lifestyle factors, and the medical staff can provide early warning and personalized prevention advice through the system.

   Overall, the motivation behind creating a diabetes prediction system is to improve individual health outcomes, enable proactive management, optimize healthcare resources, and contribute to scientific research for better prevention and management of diabetes.
3. Purpose:

   The diabetes prediction system can be used in several areas, such as scientific research, medical institutions, and health departments. For scientific research, the researchers can analyze these data by my diabetes prediction system, and they investigate the features that cause diabetes. It is useful for improving the strategy that manages this disease and advancing ways to prevent diabetes. Also, for medical institutions, it is useful to identify high-risk populations, and provide early warning and risk assessment for doctors and patients. Predicted outcomes can guide further investigations and interventions that can help reduce a patient's risk of developing diabetes.
4. Relative work:

   There is a similar system that is developed in Taiwan that can predict the diabetes, and find out patients who already have metabolic syndrome, intervene in the control early, and help the public prevent the occurrence of diseases. Enter 9 kinds of body indicators, and the system will tell the doctor whether the patient has a “metabolic syndrome” called insulin resistance, allowing the doctor to accurately identify potential diabetic patients or high-risk groups for cardiovascular diseases. And the features that this prediction system take for training the machine model are patient’s age, sex, race, fasting blood glucose, glycated hemoglobin, triglycerides, total cholesterol, and high-density cholesterol.

   There is no difference of the main goal between my system and the system that is developed in Taiwan, but the features that my system take are gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, blood

glucose level instead of age, sex, race, fasting blood glucose, glycated hemoglobin, triglycerides, total cholesterol, and high-density cholesterol.

5. System description:

```
from mlxtend.plotting import plot_decision_regions
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

I use plot_decision_regions, seaborn, and matplotlib packages to plot the graph in order to help me to visualize the data and analyze the data. Moreover, I use NumPy and pandas' packages to help me deal with the data easily, NumPy and pandas have a lot of powerful functions that can help us to do some operation of the data, such as replace the data with mean or median, and calculate the data for number of nan or sum of some specific index, the index can be string or integer, etc.

```
diabetes_data_copy = diabetes_data.copy(deep = True)
diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] \
= diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0,np.NaN)

## showing the count of Nans
print(diabetes_data_copy.isnull().sum())
```

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

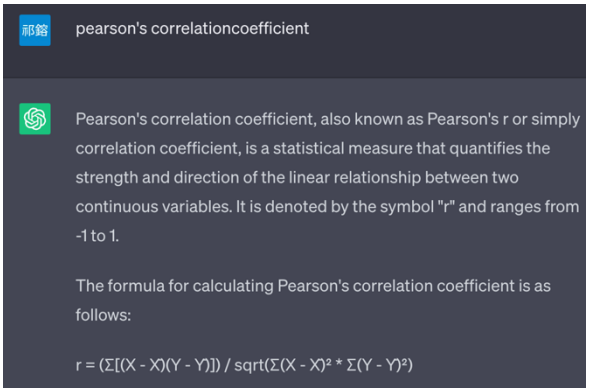
Because zero has no meaning in column Glucose, BloodPressure, SkinThickness, Insulin, and BMI, we use nan to replace zero. It would be easier to replace the meaningless value with proper value, such as mean or median.

Reference:

Pearson’s correlation coefficient:

It can help us to find the relation between two features, the value of pearson’s correlation coefficient can between +1 to -1, 1 means they are highly related, o means they are not related.

Part of the conversation:



Scaling the data:

The reason why we need to scale the data is how the feature with greater range with overshadow or diminish the smaller feature completely and this will impact the performance of all distance-based model as it will give higher weightage to variables which have higher magnitude. But it is not always correct that scaling the data because if we scale the data in wrong situation, the distribution may be broken.

The stratify of training model:

Stratify parameter makes a split so that the proportion of values in the sample produced will be the same as the proportion of values provided to parameter stratify.

After analyzing the data, we can start to train the model with the stratify that has the same distribution as the outcome of the data, and list all the condition with different value k, then finding the k value that the training model has highest accuracy.

```python
#importing train_test_split
from sklearn.model_selection import train_test_split    "sklearn": Unknown word.
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42, stratify=y)


from sklearn.neighbors import KNeighborsClassifier    "sklearn": Unknown word.


test_scores = []
train_scores = []

for i in range(1,15):

    knn = KNeighborsClassifier(i)
    knn.fit(X_train,y_train)

    train_scores.append(knn.score(X_train,y_train))
    test_scores.append(knn.score(X_test,y_test))
```

```
## score that comes from testing on the datapoints that were split in the beginning to be used for testing s
max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]
print('Max test score {} % and k = {}'.format(max_test_score*100,list(map(lambda x: x+1, test_scores_ind))))
```

```
Max test score 76.5625 % and k = [11]
```

Analyze the performance of the model:

Confusion Matrix:

The confusion matrix is a technique used for summarizing the performance of a classification algorithm i.e. it has binary outputs.

Classification Report:

Report which includes Precision, Recall, F1-Score.

ROC-AUC:

ROC (Receiver Operating Characteristic) Curve tells us about how good the model can distinguish between two things (e.g., If a patient has a disease or no). Better models can accurately distinguish between the two. Whereas a poor model will have difficulties in distinguishing between the two things.

And there is another approach that building and evaluating a model, it is called "Hyper Parameter Optimization", the use of it is shown in follow.

```
#import GridSearchCV
from sklearn.model_selection import GridSearchCV    "sklearn": Unknown wo
#In case of classifier like knn the parameter to be tuned is n_neighbors
param_grid = {'n_neighbors':np.arange(1,50)}
knn = KNeighborsClassifier()
knn_cv= GridSearchCV(knn,param_grid,cv=5)
knn_cv.fit(X,y)

print("Best Score:" + str(knn_cv.best_score_))
print("Best Parameters: " + str(knn_cv.best_params_))
```

```
Best Score:0.7721840251252015
Best Parameters: {'n_neighbors': 25}
```

6. Conclusion and reflection:

For my system, I think the strength is that it can improve the medical system because it helps doctors to predict whether the patient will have diabetes or not based on their body features, and curing earlier for fear that it become danger or more serious. Moreover, people and doctors can learn how to prevent diabetes by the graph I made for analyzing the data. But the weakness that the system has is the accuracy is not good enough, I think we can take more features that improve the system's accuracy, and take more data for training the system.