DIABETESHEALTHINDICATORS REPORT

Team members:-

- Samira Mostafa Matar

  Id:- 20201700367

- Rana Fawzy Gomaa

  Id:- 20201700270

- Alaa Adel Adam

  Id:-20201700136

- Khloud Mohamed El-Fateh

  Id:-20201700239

- Amany El-sayed Mohammed

  Id:-20201700139

- Marwa Nour El-Deen Abo-Sria

  Id:-20201700796

- Amar Alaa Awad Id:-

  20201700557

Reading data:-

readdatafromcsvfile.

Preprocessing:-

Cleaning data(remove duplicates – there is no null to be removed).

Data_coorelation:-

it shows the relation between features and dependency, is high if it equals to one and low if it equals to zero. our data is high correlation because it equals to one .

Data scaling:-

1-(min-max_scaller):

its sperate data frame into ranges and increase the accuracy

2-(standard_scaller):

its sperate data frame into ranges and increase the accuracy after split

Databalancing:we use over sampling (before balanced ((Counter({0.0: 194377, 1.0: 35097}), after balanced(Counter({0.0: 194377, 1.0: 194377})) to avoid underfitting and overfitting.

Feature selection:

we use(Linear Regression)to decrease the number of unrequired columns as a result to that the numbers of columns decrease from 22 columns to 7 columns we select 80% of the data for training to avoid overfitting and 20% of the data for testing to ensure accuracy of our predicting process.

Models

- **SVM:-**

supervised machine learning algorithm used for both classification and regression, it uses hyperplanes in high dimensional feature space ,The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

As a result to that process:-

```
----------------------------------------------------------SVM----------------------------------------------------------
[0. 1. 0. ... 1. 0. 0.]

 train score is : 0.733
test score is : 0.733
0.7328157912365187
              precision    recall  f1-score   support

         0.0       0.75      0.70      0.72     16798
         1.0       0.72      0.77      0.74     16791

    accuracy                           0.73     33589
   macro avg       0.73      0.73      0.73     33589
weighted avg       0.73      0.73      0.73     33589

confusion_matrix : [[11706  5092]
 [ 3869 12922]]
```

- **Logistic:-**

A model is used to solve classification problems ,that is used to predict the probability of a categorical dependent variable. In logistic

regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

As a result to that process:-

```
---------------------------------------------------------- LOGISTIC----------------------------------------------------------

 train score is :0.733
test score is :0.733
              precision    recall  f1-score   support

         0.0       0.75      0.70      0.72     16798
         1.0       0.72      0.76      0.74     16791

    accuracy                           0.73     33589
   macro avg       0.73      0.73      0.73     33589
weighted avg       0.73      0.73      0.73     33589

confusion_matrix : [[11821  4977]
 [ 3992 12799]]
```

- **Decision Tree:-**

Is a tree structure (a binary tree or a non-binary tree). Each non-leaf node represents a test on a feature attribute. Each branch represents the output of a feature attribute in a certain value range, and each leaf node stores a category.

It works for both continuous as well as categorical output variables.

attribute values records are distributed recursively. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

As a result to that process:-

```
---------------------------------------------------- DECISION TREE----------------------------------------------------------

 train score is : 0.918
test score is : 0.901
              precision    recall  f1-score   support

         0.0       0.85      0.97      0.91     16798
         1.0       0.97      0.83      0.89     16791

    accuracy                           0.90     33589
   macro avg       0.91      0.90      0.90     33589
weighted avg       0.91      0.90      0.90     33589

confusion_matrix : [[16317   481]
 [ 2834 13957]]
```

K Nearest Neighbor Algorithm:-

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most like the available categories.

As a result to that process:-

```
[0. 1. 0. ... 1. 0. 0.]
              precision    recall  f1-score   support

         0.0       0.83      0.93      0.88     16798
         1.0       0.92      0.80      0.86     16791

    accuracy                           0.87     33589
   macro avg       0.87      0.87      0.87     33589
weighted avg       0.87      0.87      0.87     33589

confusion_matrix : [[15638  1160]
 [ 3304 13487]]
```

Data combining:-

Stacking: It is an ensemble method that combines multiple models (classification or regression) via meta-model (meta-classifier or meta-regression). The base models are trained on the complete dataset, then the meta-model is trained on features returned (as output) from base models. The base models in stacking are typically different. The meta-model helps to find the features from base models to achieve the best accuracy.

As a result to that process:-

```
9.016043986483984
              precision    recall  f1-score   support

         0.0       0.75      0.71      0.73     16798
         1.0       0.73      0.77      0.75     16791

    accuracy                           0.74     33589
   macro avg       0.74      0.74      0.74     33589
weighted avg       0.74      0.74      0.74     33589
```

Decision Tree Regression:-

A model is used to solve Regression problems that used to select the important columns in the data frame to increase the accuracy.

```
model_tree=DecisionTreeRegressor(max_depth=6,random_state=0)
model_tree.fit(X=X_train,y=y_train)
# importance feature
importance =model_tree.feature_importances_
print(importance)
def plot_feature_importance(model3):
    plt.figure(figsize=(8,6))
    n_feature=21
    plt.barh(range(n_feature),model3.feature_importances_,align='center')
    plt.yticks(np.arange(n_feature),x)
    plt.xlabel("feature importance")
    plt.xlabel("feature")
    plt.ylim(-1,n_feature)
plot_feature_importance(model_tree)
plt.savefig('feature')
plt.show()
```

As a result to that process:-