**Machine Learning Model for Diabetes Prediction**

**Introduction**

This note attempts to address how Machine Learning [ML] can used to help with a common medical problem – Diabetes. Diabetes is an ever-growing problem worldwide and any insights that can be gleaned from data relating to past incidences can only be of benefit.

The aims of the project are to:

* Identify a suitable diabetes dataset
* Build a suitable ML model
* Analyse the model
* Make appropriate inferences

A Git repository has been set up locally at:

**C:\Users\Naresh\Portfolio.**

A Jupyter Notebook : **ML\_Portfolio.jpynb** shall be utilized for the project. Visual Code Studio will be used for the model build and analysis.

**Dataset:**

A dataset was searched for and the following was considered suitable for this project:

[Diabetes Health Indicators Dataset (kaggle.com)](https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_5050split_health_indicators_BRFSS2015.csv)

The dataset, a csv file is:

diabetes \_ binary \_ 5050split \_ health \_ indicators \_ BRFSS2015.csv

is a clean dataset of 70,692 survey responses to the CDC's BRFSS2015. It has an equal 50-50 split of respondents with no diabetes and with either prediabetes or diabetes. The target variable Diabetes\_binary has 2 classes. 0 is for no diabetes, and 1 is for prediabetes or diabetes. This dataset has 21 feature variables and is balanced.

The features are given below:

| Variable Name | Role | Type | Description | Missing Values |
| --- | --- | --- | --- | --- |
| ID | ID | Integer | Patient ID | no |
| Diabetes\_binary | Target | Binary | 0 = no diabetes 1 = prediabetes or diabetes | no |
| HighBP | Feature | Binary | 0 = no high BP 1 = high BP | no |
| HighChol | Feature | Binary | 0 = no high cholesterol 1 = high cholesterol | no |
| CholCheck | Feature | Binary | 0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years | no |
| BMI | Feature | Integer | Body Mass Index | no |
| Smoker | Feature | Binary | Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes | no |
| Stroke | Feature | Binary | (Ever told) you had a stroke. 0 = no 1 = yes | no |
| HeartDiseaseorAttack | Feature | Binary | coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes | no |
| PhysActivity | Feature | Binary | physical activity in past 30 days - not including job 0 = no 1 = yes | no |
| Fruits | Feature | Binary | Consume Fruit 1 or more times per day 0 = no 1 = yes | no |
| Veggies | Feature | Binary | Consume Vegetables 1 or more times per day 0 = no 1 = yes | no |
| HvyAlcoholConsump | Feature | Binary | Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week) 0 = no 1 = yes | no |
| AnyHealthcare | Feature | Binary | Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no 1 = yes | no |
| NoDocbcCost | Feature | Binary | Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no 1 = yes | no |
| GenHlth | Feature | Integer | Would you say that in general your health is: scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor | no |
| MentHlth | Feature | Integer | Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good? scale 1-30 days | no |
| PhysHlth | Feature | Integer | Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good? scale 1-30 days | no |
| DiffWalk | Feature | Binary | Do you have serious difficulty walking or climbing stairs? 0 = no 1 = yes | no |
| Sex | Feature | Binary | 0 = female 1 = male | no |
| Age | Feature | Integer | 13-level age category (\_AGEG5YR see codebook) 1 = 18-24, 2 = 25-29, 3 = 30-34, 4 = 35-39, 5 = 40-44, 6 = 45-49, 7 = 50-54, 8 = 55-59, 9 = 60-64, 10 = 65-69, 11 = 70-74, 12 = 75-79, 13 = 80 or older | no |
| Education | Feature | Integer | Education level (EDUCA see codebook) scale 1-6 1 = Never attended school or only kindergarten 2 = Grades 1 through 8 (Elementary) 3 = Grades 9 through 11 (Some high school) 4 = Grade 12 or GED (High school graduate) 5 = College 1 year to 3 years (Some college or technical school) 6 = College 4 years or more (College graduate) | no |
| Income | Feature | Integer | Income scale (INCOME2 see codebook) scale 1-8 1 = less than10,0005=lessthan10,0002= 5=𝑙𝑒𝑠𝑠𝑡ℎ𝑎𝑛35,000 8 = $75,000 or more | no |

In [5]:

This shows 22 features. The first feature is actually the target [Diabetes\_binary]. Hence we can regard this dataset consisting of 21 features and a target variable.

The first cell of the Jupyter notebook shows the size, first 5 rows and the data information. See Output 1 at the end of this paper.

This shows that there are no nun-numeric categorical variables and the categorical numerical variables are all ordinal.

We also see that there are no null values from the information list (all columns have 70692 values).

**Feature Selection**

The next step is to see:

* Correlation between features if any [heat map]
* Correlation between each feature and the target [heatmap]

From the heatmaps we gather:

* The features are fairly independent of each other (Figure 1)
* There is only weak correlation between each feature and the target variable which seems to suggest that the occurrence of diabetes is not heavily dependent upon the features in question (Figure 2)

From the individual correlations [Figure 2] I aim to select features with a correlation greater than 0.2 for this exercise.

This list of features is:

HighBP

HighChol

BMI

HeartDiseaseorAttack

GenHlth

PhysHlth

DiffWalk

Age

Income

**Initial Analysis**

The model required is a classification model. For this I shall be choosing the following:

* Lasso classification
* Ridge classification

Before delving into a grid search to identify the best model, it is proposed to explore what would be a suitable range of the hyperparameter for each model.

The results for a wide range of c values for each are given in Output 2 below.

This suggests that very small values of c are not relevant for either the ridge or lasso models.

A sensible set would be in the range log(-3, +3) for both.

**Pipeline Build and Run**

Next cell in the Jupyter notebook shows the build of a pipeline with Logistic Regression as the estimator with liblinear as solver.

A gridsearch is then effected with a cv value of 5 and a search grid of [l1, l2] (Lasso and Ridge) over 50 C values each over a log scale.

The results are as Output 3 and Figure 3 below.

**Final Analysis:**

It can be seen that the best model is a Ridge classification model.

The accuracy of the model is fairly static at 74.4% across both, the training data as well as the test (validation) data. This shows that there is no under/over fitting.

The accuracy level in the 70% region is not exceptional and so not as reliable as we would like.

However, the coefficients and their sizes seem to suggest that:

BP and cholesterol level do have some bearing.

General health and heart attack have some significance.

The other features do not seem to have noticeable impact:

Age

BMI

Physical health

Difficulty walking

Income

The correlation maps did show that the relationship to the target variable was not very strong and in light of that a 70+% is considered fairly good.

There is always the possibility of factors not included in the dataset having a stronger bearing and so further studies may well prove more revealing.

**Output 1**

(70692, 22)

---

   Diabetes\_binary  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \

0              0.0     1.0       0.0        1.0  26.0     0.0     0.0

1              0.0     1.0       1.0        1.0  26.0     1.0     1.0

2              0.0     0.0       0.0        1.0  26.0     0.0     0.0

3              0.0     1.0       1.0        1.0  28.0     1.0     0.0

4              0.0     0.0       0.0        1.0  29.0     1.0     0.0

   HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare  \

0                   0.0           1.0     0.0  ...            1.0

1                   0.0           0.0     1.0  ...            1.0

2                   0.0           1.0     1.0  ...            1.0

3                   0.0           1.0     1.0  ...            1.0

4                   0.0           1.0     1.0  ...            1.0

   NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  Education  \

0          0.0      3.0       5.0      30.0       0.0  1.0   4.0        6.0

1          0.0      3.0       0.0       0.0       0.0  1.0  12.0        6.0

2          0.0      1.0       0.0      10.0       0.0  1.0  13.0        6.0

3          0.0      3.0       0.0       3.0       0.0  1.0  11.0        6.0

4          0.0      2.0       0.0       0.0       0.0  0.0   8.0        5.0

   Income

0     8.0

1     8.0

2     8.0

3     8.0

4     8.0

[5 rows x 22 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 70692 entries, 0 to 70691

Data columns (total 22 columns):

 #   Column                Non-Null Count  Dtype

---  ------                --------------  -----

 0   Diabetes\_binary       70692 non-null  float64

 1   HighBP                70692 non-null  float64

 2   HighChol              70692 non-null  float64

 3   CholCheck             70692 non-null  float64

 4   BMI                   70692 non-null  float64

 5   Smoker                70692 non-null  float64

 6   Stroke                70692 non-null  float64

 7   HeartDiseaseorAttack  70692 non-null  float64

 8   PhysActivity          70692 non-null  float64

 9   Fruits                70692 non-null  float64

 10  Veggies               70692 non-null  float64

 11  HvyAlcoholConsump     70692 non-null  float64

 12  AnyHealthcare         70692 non-null  float64

 13  NoDocbcCost           70692 non-null  float64

 14  GenHlth               70692 non-null  float64

 15  MentHlth              70692 non-null  float64

 16  PhysHlth              70692 non-null  float64

 17  DiffWalk              70692 non-null  float64

 18  Sex                   70692 non-null  float64

 19  Age                   70692 non-null  float64

 20  Education             70692 non-null  float64

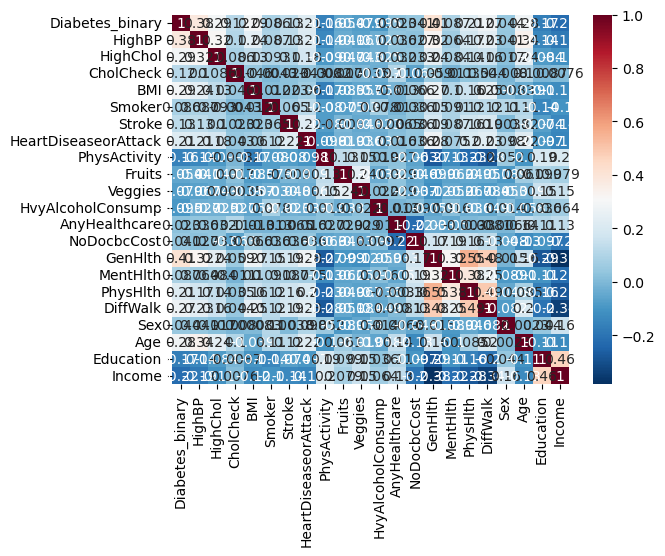
 21  Income                70692 non-null  float64

dtypes: float64(22)

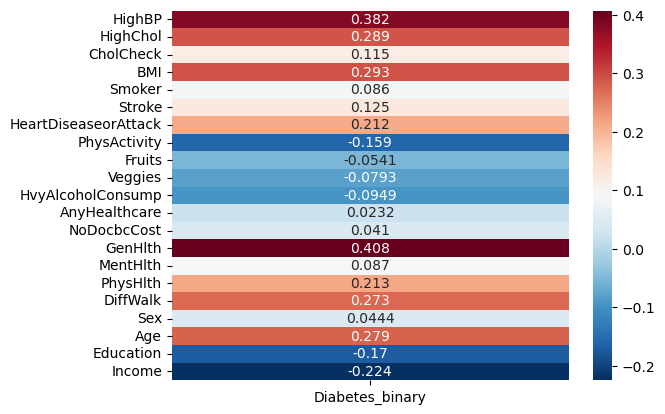
memory usage: 11.9 MB

None

**Figure 1**



**Figure 2**



**Output 2**

Ridge C: 1e-05

Train score: 0.7042758256474094

Test score: 0.7016918463192441

Ridge C: 0.001

Train score: 0.7441483241856693

Test score: 0.7435070446443728

Ridge C: 1

Train score: 0.7442237688375866

Test score: 0.7441860465116279

Ridge C: 10

Train score: 0.7442237688375866

Test score: 0.7442426300005658

Ridge C: 100

Train score: 0.7442237688375866

Test score: 0.7442426300005658

Ridge C: 1000

Train score: 0.7442426300005658

Test score: 0.7442992134895038

Lasso C: 1e-05

Train score: 0.5027820215394482

Test score: 0.49165393538165564

Lasso C: 0.001

Train score: 0.7080669194062506

Test score: 0.715045549708595

Lasso C: 1

Train score: 0.7442614911635451

Test score: 0.7442992134895038

Lasso C: 10

Train score: 0.7442614911635451

Test score: 0.7442426300005658

Lasso C: 100

Train score: 0.7442614911635451

Test score: 0.7442426300005658

Lasso C: 1000

Train score: 0.7442803523265245

Test score: 0.7441860465116279

**Output 3**

Best pipeline:

Pipeline(steps=[('regr',

LogisticRegression(C=0.0517947467923121, max\_iter=1000,

solver='liblinear'))])

The best regression model is:

LogisticRegression(C=0.0517947467923121, max\_iter=1000, solver='liblinear')

Best score is: 0.744544441686486

Best params {'regr\_\_C': 0.0517947467923121, 'regr\_\_penalty': 'l2'}

Best coefficients: [[ 0.74360309 0.58067569 0.06942512 0.31894908 0.55239387 -0.00932032 0.12162908 0.14134744 -0.07241724]]

Test data score: 0.7448084648899451

(Best mode)

**Figure 3 Coefficients**

