

Capstone Project

Diabetes Prediction using Supervised Machine Learning



July 24, 2022

Anand Rajan

# Abstract

This project aims to study health and lifestyle information of an observed set of people and its correlation with diabetes. Using supervised machine learning techniques, we attempt to replicate the results as closely as possible, so that pre-emptive help can be offered to people not wishing to suffer this chronic and debilitating disease. The dataset is split into 2 parts, namely, training and testing data. After performing adequate exploratory data analysis on all features, we are able to extract interesting insights. Following this, we will attempt to teach a Machine Learning model how to arrive at a similar conclusion based on given information.

Table of Contents

[Abstract 1](#_Toc109540112)

[Introduction 3](#_Toc109540113)

[Data Review 4](#_Toc109540114)

[Data Visualization and Manipulation 5](#_Toc109540115)

[Steps performed 8](#_Toc109540116)

[Results 10](#_Toc109540117)

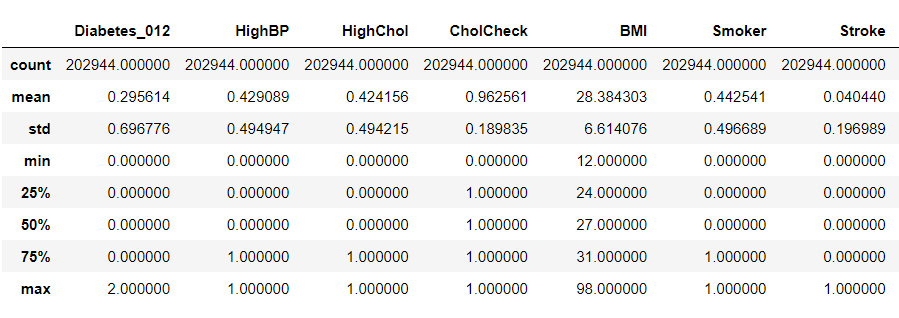
[Conclusion 14](#_Toc109540118)

# Introduction

Diabetes is the sixth leading cause of death in America today. Approximately 225,000 people die each year of diabetes complications. There are basically two types of diabetes. Type 1 is known as the childhood diabetes, but not necessarily limited to children. Type 1 diabetes is the more serious of the 2 types. People with this type require daily injections of insulin. Type 1 diabetes can appear suddenly to an individual who appeared to be otherwise healthy. Type 2 diabetes is often referred to as the adult diabetes. It is a metabolic disorder, non-insulin dependent. Type 2 develops slowly over time. Diabetes of either type can lead to other serious health problems. There is another type of diabetes known as type 3 that affects pregnant women. Even though found in pregnant women and can go away after birth, this type of diabetes can often lead to type 2 later in life. Each of these types of diabetes has one thing in common, that is high glucose levels in the blood.

We will study a dataset with all possible metrics and details, and will attempt to use 3 ML algorithms to accurately learn the patterns leading up to the subject’s diabetic status. This is a small, severely disproportionate and unbalanced dataset which could lead to learning difficulties as opposed to a larger dataset that contains balanced records of non-diabetic, pre-diabetic and diabetic people.

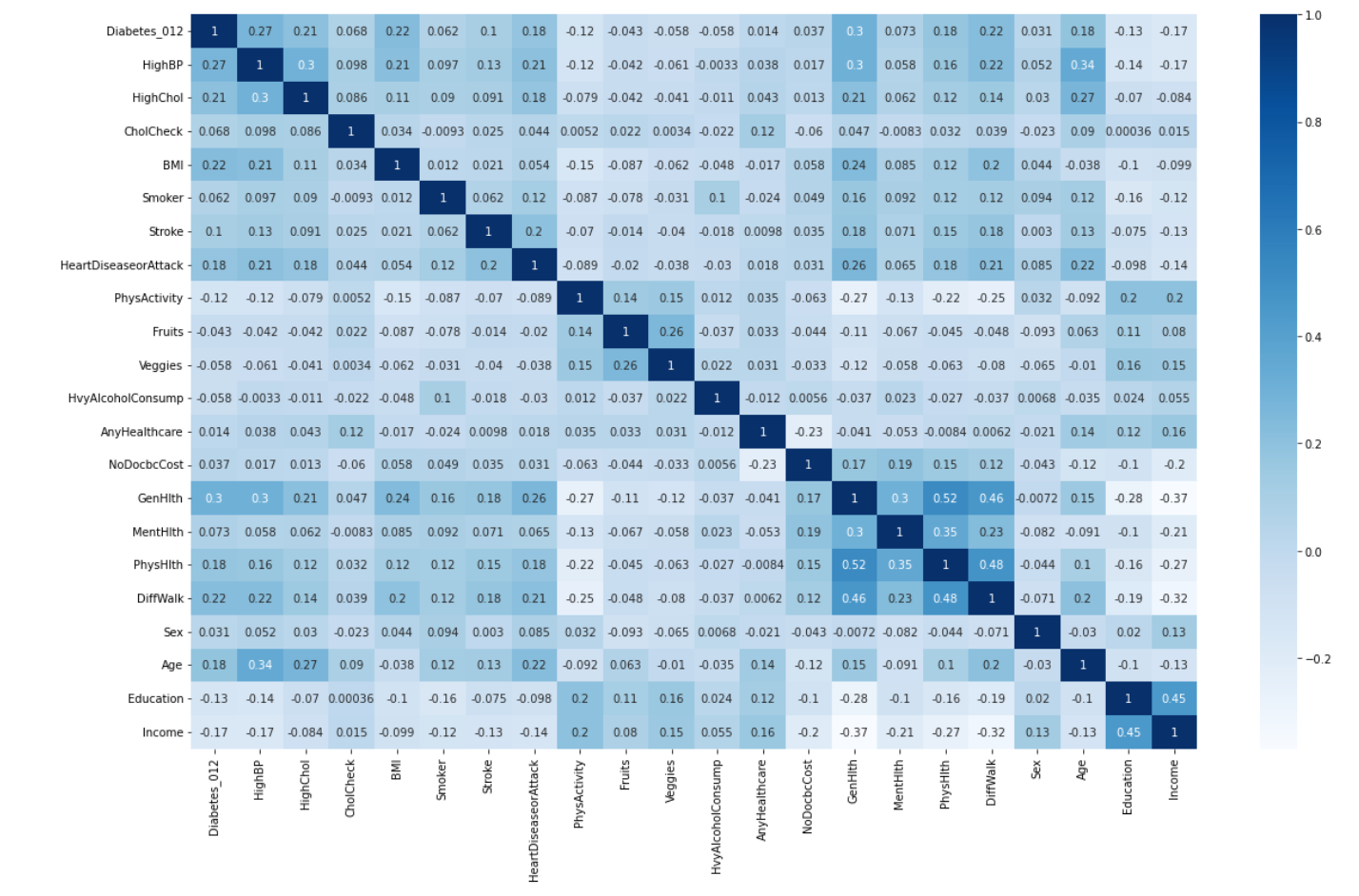
# Data Review



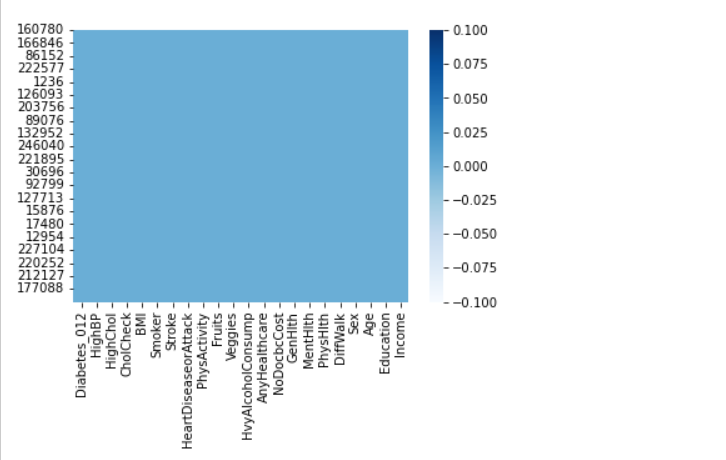
* In order to keep results as accurate as possible, we split the dataset into training and testing data at the very beginning. All our observations are based on training data.
* Our training dataset contains xxx records of which x are non-diabetic, x are pre-diabetic and x are diabetic.
* Fortunately, this dataset is completely filled without any missing data.
* Much of the data is pseudo factorized as most of the columns deal with ‘Yes’ and ‘No’ sort of data.
* BMI is a column which varies by the individual. We were able to find an astonishing outlier as high as 98. However, this did not affect the outcome across multiple cycles of modelling.

# Data Visualization and Manipulation

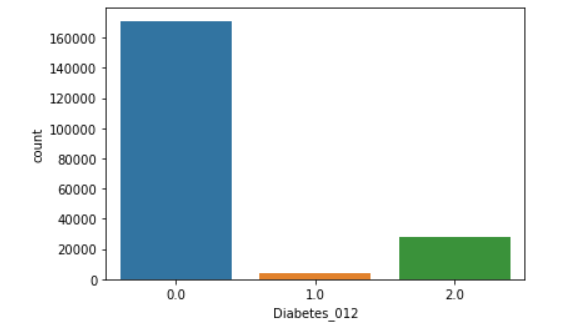
First, we shall have a look at the correlation matrix that shows us that all of the columns have some impact on other columns, hence showing correlation. Based on this inference, we are able to drop the PhysHlth column as it has a very similar effect as GenHlth.



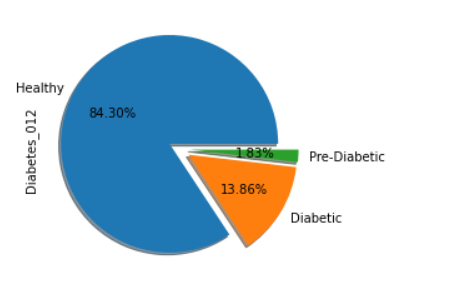
Following this, we find out that the dataset does not have any missing information. This is supported by the following graph. This indicates that the model can learn to the fullest extent that a dataset this size allows.



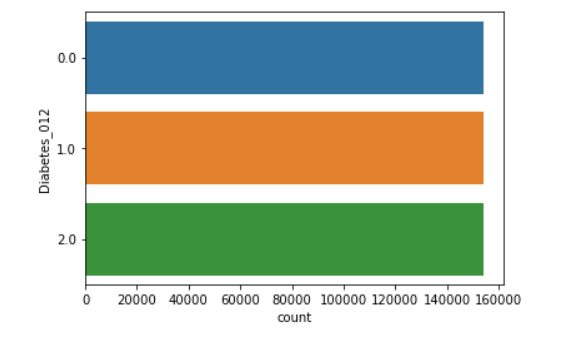
Next, we can see how the target/dependent variable is distributed. It is observed that the data contains an overwhelming number of records for non-diabetic people.



With the exploded pie chart, we are able to find a comparative metric on the uneven distribution.



By using the concept of oversampling, i.e., adding duplicated records of the minority factors in order to make the dataset be distributed equally among all 3 categories.



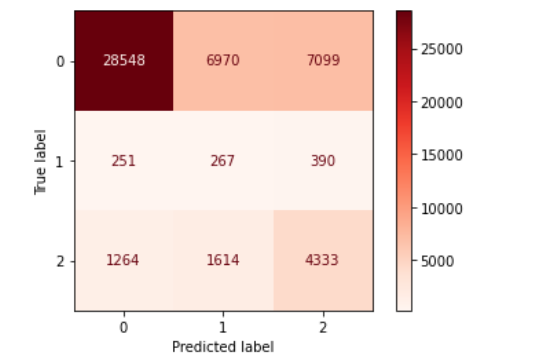
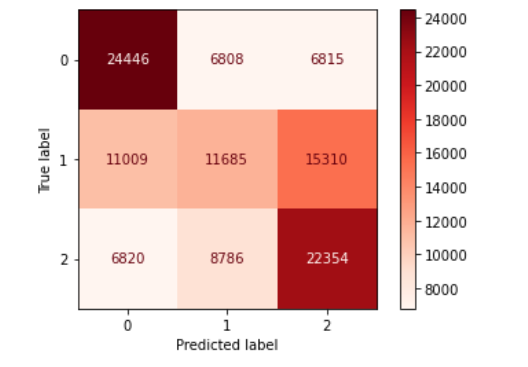
# Steps performed

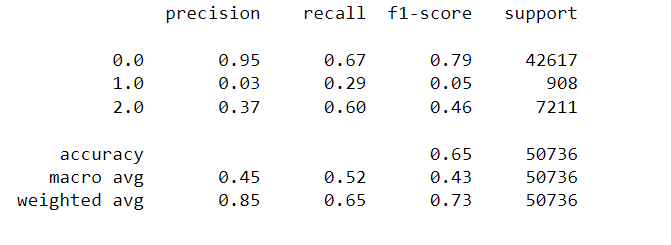
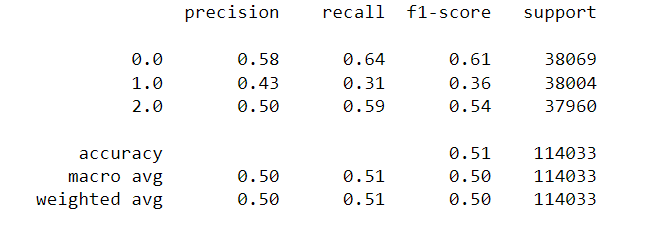
1. Installing the necessary packages.
   1. Pandas
   2. Pyplot
   3. Seaborn
   4. Sklearn
      1. Model\_selection
      2. Preprocessing
      3. Decomposition
      4. Linear\_model
      5. Naïve\_bayes
      6. Ensemble
      7. Metrics
2. Importing the required packages.
3. Loading the dataset.
   1. Created a pandas dataframe of dimensions: 253680 x 22.
4. Split the dataset into testing and training data.
   1. The need to do this is to test the model on raw data without any modifications for real world competence.
   2. Created a training dataframe of dimensions: 202944 x 22.
   3. Created a testing dataframe of dimensions: 50736 x 22.
5. Checking data for cleanliness and usability.
   1. No missing values found.
   2. 17123 duplicate values found and dropped.
   3. 'PhysHlth' dropped from both dataframes as it had closely similar impact as ‘GenHlth’.
6. Developing various plots for exploratory data analysis.
7. Oversample training dataset to balance data for learning.
   1. The new training dataframe is shaped at dimensions: 462408 x 21.
8. Split datasets into dependent and independent datasets.
   1. X\_train and y\_train were split using the dataframe.iloc functionality.
   2. X\_test and y\_test were split similarly.
9. Feature scaling depending on ML algorithm used.
   1. Use MinMaxScaler for Feature scaling for Logistic Regression and Naïve-Bayes Algorithm. Since there is no use of scaling in Random Forest, it is not executed while preparing to train RF model.
10. Build custom methods for code cleanliness and readability of modelling.
    1. Modelling and post modelling code, encapsulated in neat, coder friendly functions.
11. Build models.
    1. Logistic Regression – It is used to predict the probability of a categorical dependent variable.
    2. Naïve-Bayes - Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.
    3. Random Forest - It implements Breiman's random forest algorithm for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.
12. Predicting values.
    1. Predicted and test values are sent to an external file to record comparative results.
13. Creating confusion matrix and classification report for every model.
14. Interpreting results.

# Results

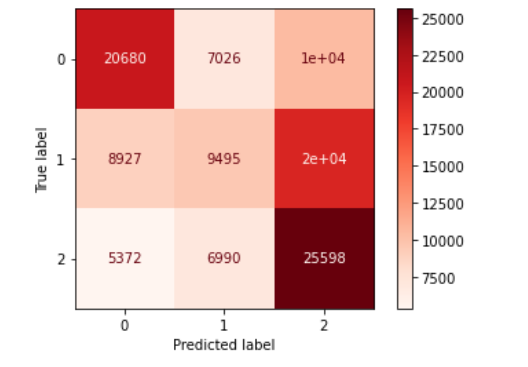
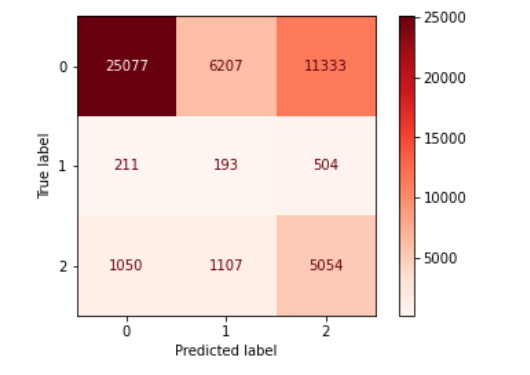
We have created a confusion matrix and classification report for every model. An interesting observation was that the results were greatly varied based on whether the oversampling was done before or after the Train Test split. This is interesting because while it may not work well in real world situations, in one of its cycles, the RF model (oversampling before TTS) for this dataset was able to achieve the highest accuracy of over 96% with almost perfect precision and recall.

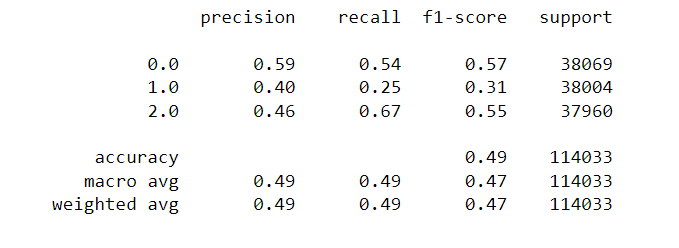
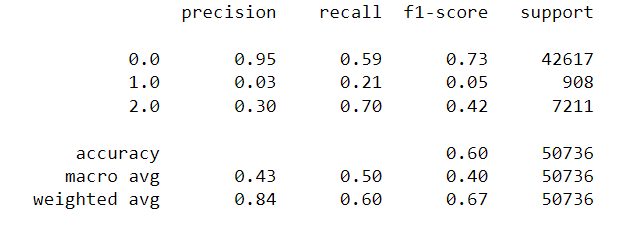
Logistic Regression (before and after)



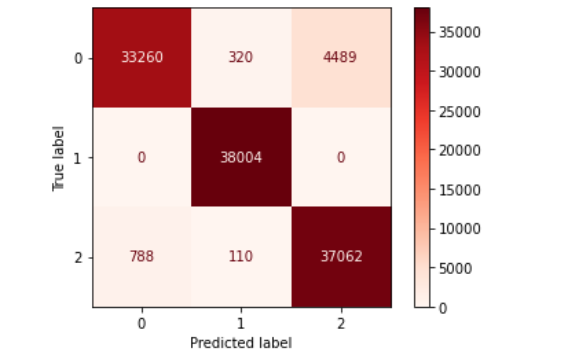
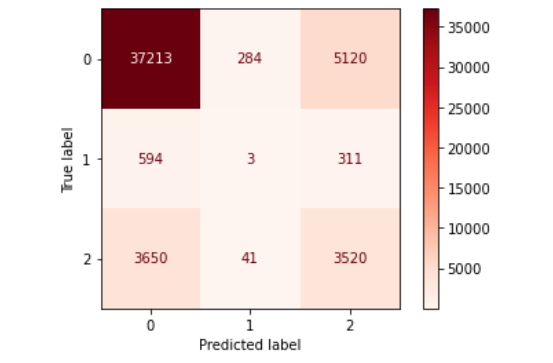


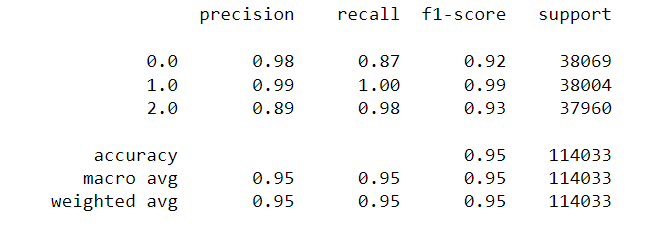
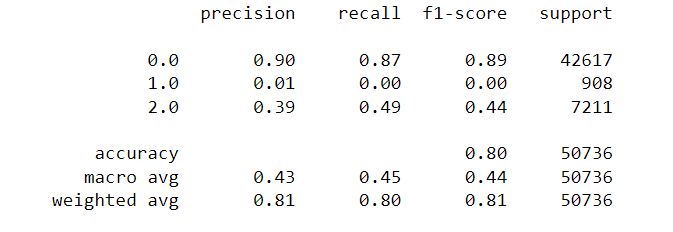
Naïve Bayes (before and after)

Random Forest (before and after)

Random Forest is the most consistent model in terms of higher accuracy.

|  |  |
| --- | --- |
| y\_test | y\_pred |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 2 | 2 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 2 | 2 |
| 0 | 0 |
| 2 | 2 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 2 | 0 |
| 2 | 2 |
| 1 | 0 |

As seen in the above results, bagging has its risks. Being lumped together in random sample sizes does reduce the chances of overfitting, but does not eliminate the risk. The dismal recall and precision of all the models to predict ‘1.0’ and a little less severely ‘2.0’, proves that unbalanced data can cause less than ideal training.

In factorized target data, we should not pay too much attention to accuracy as opposed to recall and precision of all target factors. As proven by the ‘split after being oversampled’ model, a balanced dataset could result in much more accurate predictions.

# Conclusion

Ensemble learning usually gives a better performance than a single model because it alleviates the overfitting problem and also it combines the strength of different models. However, it could incur longer runtime, and also the model logic is harder to interpret.

Random forest algorithm adopts the Bagging technique by randomly select the subset of the dataset for the different decision trees to fit. In addition, random forest algorithm also randomizes the features to be trained for the model which further reduces the chances of overfitting.

To conclude, I would like to say that Python as a programming language for analytics is very powerful and gives immense flexibility to the coder. It helped me to build models far easier than I would have in other languages.