Project

Certified Data Scientist

Predicting diabetes using different algorithms

Submitted in partial fulfilment of the requirements for the
EN ISO / IEC 17024 certification exam by

Kfenti Mbeng

mbengkfenti@t-online.de

Declaration

I hereby declare that I have completed this project work independently and without the involvement of third parties. References have been appropriately quoted.

Erkrath, 14.09.2022

Table of Contents

Background

Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin (type 1) or when the body cannot effectively use the insulin it produces (type 2). Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels [2]. According to the world health organisation (WHO), in 2014, 8.5% of adults aged 18 years and older had diabetes. In 2017 there were 9 million people with type 1 diabetes with the majority of them living in high-income countries. In 2019, diabetes was the direct cause of 1.5 million deaths and 48% of all deaths due to diabetes occurred before the age of 70 years [1]. From this statistics, it would be easy to imagine that the situation will continue to deteriorate if the causes of diabetes are not determined and remedial measures put in place. Whereas the causes of type 2 diabetes are related to excess body weight and physical inactivity, the causes for type 1 diabetes are not known. From the available data on past cases, patterns can be identified and investigated from which predictions can be made regarding subsequent cases.

Introduction

Data science involves principles, processes, and techniques for understanding phenomena or patterns or insights in data via the (automated) analysis of data [5]. Once this information is available, predictions using machine Learning (ML) tools on future likely outcomes can be made. This project makes use of Data Science methodologies. The programming language employed here is Python and its respective modules. During this project, CRISP- methodology of data mining will be adopted (see Fig. 1 below). Accordingly, a detail look into the dataset will begin the project and pre-processing strategies will be implemented, where necessary. Once the data is understood and processed, it will then be split into a training and test datasets. Models will be trained on training data and their performances or accuracies tested on the test data. The principle underlying each tested algorithm will be clearly stated before its application. A conclusion on the best performed algorithm on this dataset will be drawn and verified on the research question.

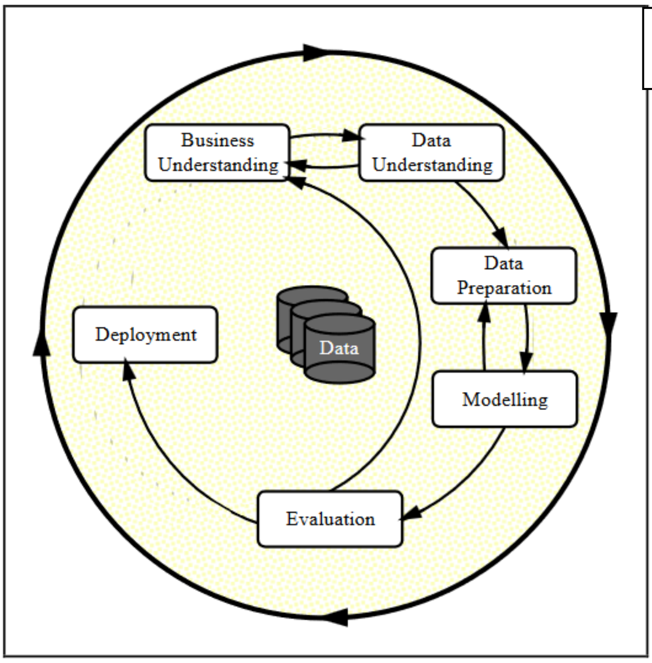


Fig. 1

Phases of the Current CRISP-DM Process Model for Data Mining

Objective

The objective of this project is to predict diabetes using different algorithms. Predictions are made based on available data. The dataset that is used in this project is downloaded from Kaggle [3]. In order to attain this objective, Data science and machine learning methodologies are implemented and a conclusion drawn as to which algorithm performs well with the dataset and what parameter or variable best influences the occurrence of diabetes.

Methodology

Since data and target are known, supervise learning methods will be employed. Contrary to Regression methods which address task with numerical continuous values, classification methods address research questions with discrete and categorical target value. Classification is concerned with building a model that separates data into distinct classes. This model is built by inputting a set of training data for which the classes are pre-labelled so that the algorithm can learn from. The model is then used by inputting a different dataset for which the labels/classes are withheld, allowing the model to predict their class membership based on what it has learned from the training set.

The answer to the research question – whether a person will have diabetes or not is a categorical value (Yes or No) and hence invites the application of Binary Classification models. The following classification algorithms are tried iteratively in the course of this work; K-Nearest Neighbors, Decision Tree, Logistic Regression, Gaussian Naives Bayes, Ensemble Methods specifically Random Forest, Stochastic gradient descent, Bagging and AdaBoost.

Data

Before diving in to exploit the usefulness of various algorithms, the dataset is downloaded and exploit. As mentioned earlier, the dataset is obtained from Kaggle, titled: Diabetes Health Indicators Dataset. This dataset represents 253,680 survey responses to the CDC's BRFSS2015 (CDC = Centers for Disease Control and Prevention. BRFSS = Behavioral Risk Factor Surveillance System). The target variable - Diabetes_binary has 2 classes. 0 is for no diabetes, and 1 is for prediabetes or diabetes. This dataset has 21 independent feature variables. The following table shows what the variables and their respective values represent.

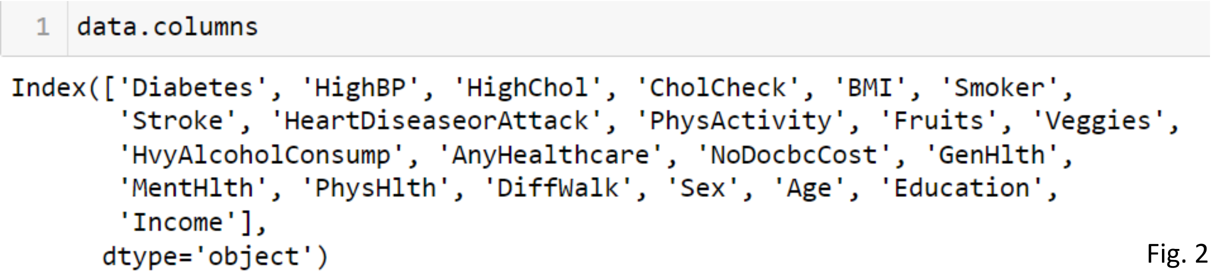
Feature	**Meaning of values**
HighBP	0 = no high BP, 1 = high BP
Highchol	0 = no high cholesterol, 1 = high cholesterol
cholcheck	0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years
Smoke	Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no, 1 = yes
Stroke	(Ever told) you had a stroke. 0 = no, 1 = yes
HeartDiseaseorAttack	coronary heart disease (CHD) or myocardial infarction (MI) 0 = no, 1 = yes
BMI	Body mass index measured values
Fruits	Consume Fruit 1 or more times per day 0 = no, 1 = yes
HeartDiseaseorAttack	coronary heart disease (CHD) or myocardial infarction (MI) 0 = no, 1 = yes
PhysActivity	physical activity in past 30 days - not including job 0 = no, 1 = yes
Veggies	Consume Vegetables 1 or more times per day 0 = no, 1 = yes
HvyAlcoholConsump	(adult men >=14 drinks per week and adult women>=7 drinks per week) 0 = no, 1 = yes
AnyHealthcare	Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no, 1 = yes
NoDocbcCost	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
GenHlth	Would you say that in general your health is: scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor
MentHlth	days of poor mental health scale 1-30 days

PhysHlth	physical illness or injury days in past 30 days scale 1-30
DiffWalk	Do you have serious difficulty walking or climbing stairs? 0 = no, 1 = yes
Sex	0 = female 1 = male
Age	level age category 1 = 18-24, 9 = 60-64, 13 = 80 or older
Education	Education level scale 1-6 1 = Never attended school or only kindergarten 2 = elementary etc
Income	Income scale (INCOME2 see codebook) scale 1-8 1 = less than $10,000, 5 = less than $35,000, 8 = $75,000 or more

Table 1. Meaning of values in the variables

Data understanding / exploitation / preparation

It involves understanding the type and distribution of data contained within each variable, imputing missing values, encoding categorical values, looking for the relationships between variables and how they vary relative to the outcome. According to kaggle, this dataset is clean meaning that it has been pre-processed. All categorical values as well as empty entries have been replaced and all the variables have an equal number of instances. See Fig. 2 below for column names, Fig.3 for the frequency distribution of the outcome in the target variable. Fig. 4 gives the column information;



```
data.columns
```

```
Index(['Diabetes', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker',
       'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
       'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
       'Income'],
      dtype='object')
```

Fig. 2

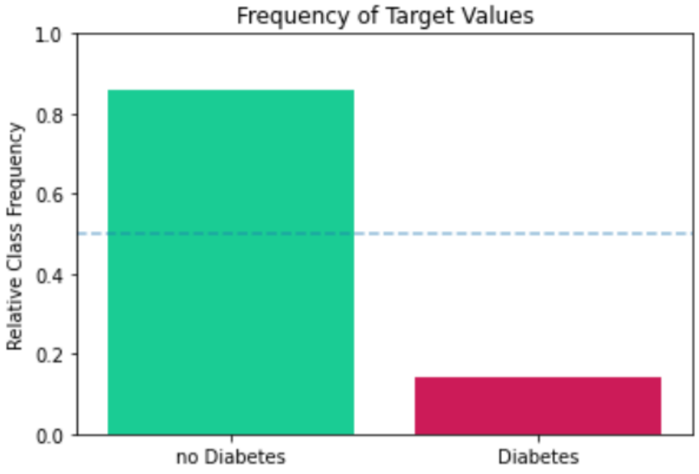
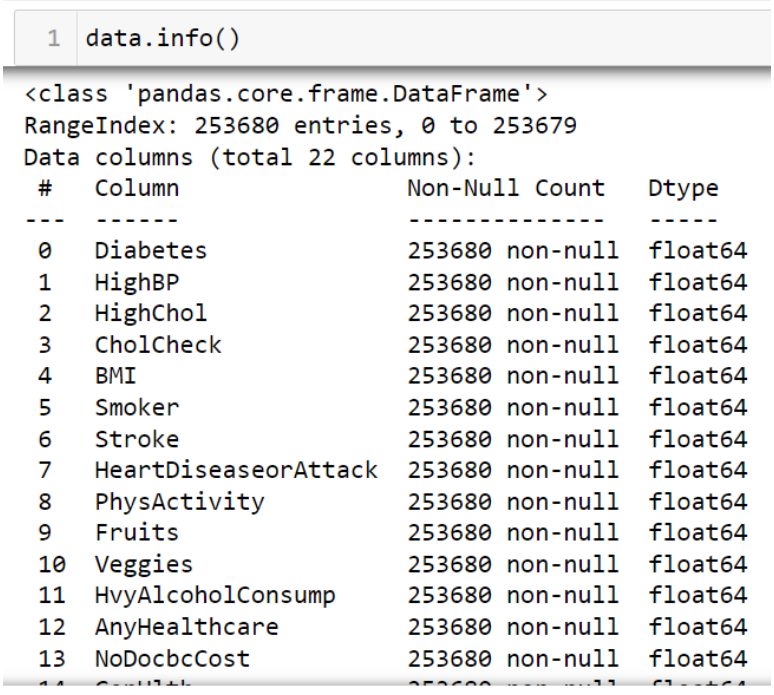


Fig. 3

From the distribution above (Fig. 3), it can be seen that, 86% of respondents are negative (with no Diabetes) while 14% are positive (with Diabetes). These results should later on reflect on the accuracy once the models are trained.



```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes              253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   HeartDiseaseorAttack  253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
```

Fig. 4

To understand the inter-relationship between the variables particularly the target variable, Pearson correlation is used. The Pearson correlation measures the strength of the linear relationship between two variables. It has a value between -1 to 1, with a value of -1 meaning a total negative linear correlation, 0 being no correlation, and + 1 meaning a total positive correlation.

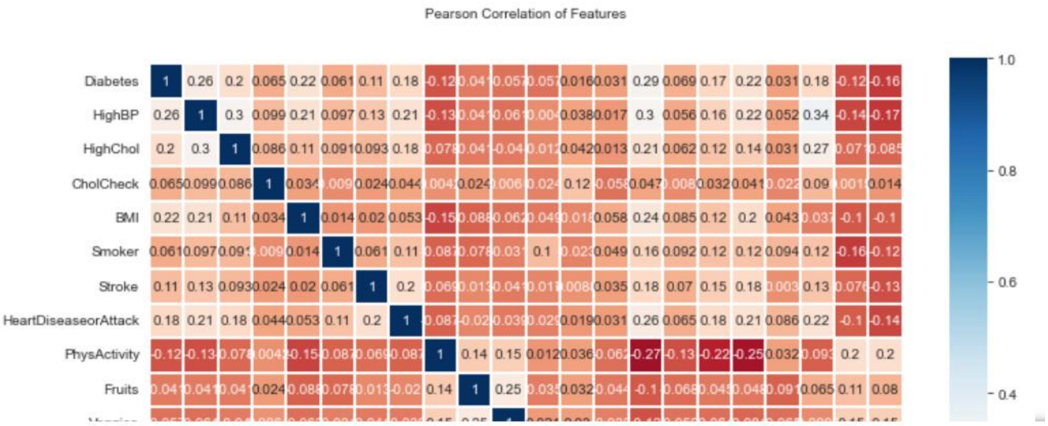


Fig. 5a

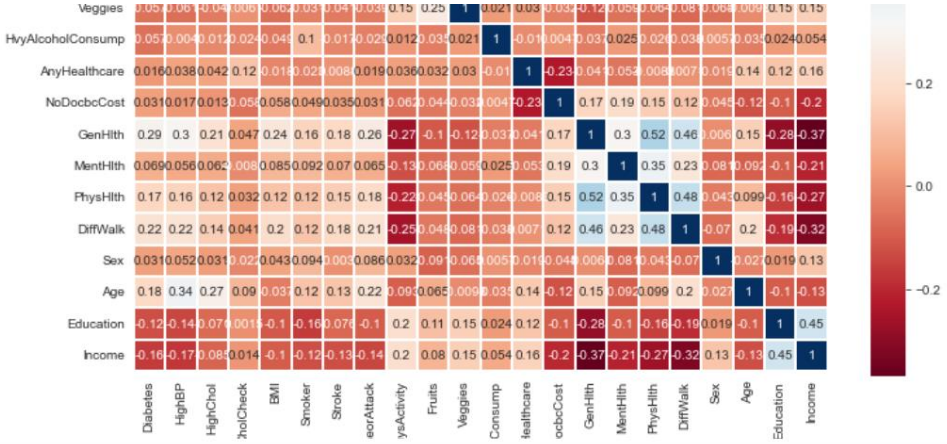


Fig. 5b

From the correlation matrix it can be observed that;

HighBP, Highchol, BMI, Heartdisease, GenHlth, physHlth, DiffWalk, Income and Age seem to have an influence on diabetes though minimal. Only variables that relate with Diabetes and have a Pearson coefficient above 0,1 are viewed to have a significant influence. It is worth noting that, the other variables with Pearson coefficient less than 0,1 are not ignored.

Model training and evaluation

Now that the data is prepared and understood, the respective models can be trained. Before training the models, the dataset is split into the train and test datasets. The models are trained on the train set and the performances checked by introducing “new” (test) dataset whose actual values are known and compared to the predicted values. The performance of the models is evaluated with the help of the following classification metrics; Accuracy, Precision and Recall, Receiver Operation Characteristics (ROC) and the area under the ROC curve.

Logistic regression

In logistic regression the probability of an outcome is predicted by applying the logistic function or a sigmoid function to a linear regression. The outcome value is between 0 and 1 such that any data point with a probability value above the line is classified to class 1 while data points with probability below the line is classified to class 0.

A logistic regression model based on the given data is trained and the results obtained are explained with the help of classification metrics. Fig. 6 below is a confusion matrix. The confusion matrix function evaluates classification accuracy by computing the confusion matrix with each row corresponding to the true class. In the matrix, the diagonal elements are the true positive (TP) that is, an outcome where the model correctly predicts the positive class. True negative (TN) count for a class is an outcome where the model correctly predicts the negative class. The false positive (FP) is an outcome where the model incorrectly predicts the positive class. Finally, the false negative (FN) is an outcome where the model incorrectly predicts the negative class [6].

Accuracy = {(TP+TN)/ total number of predictions} = 86%

Precision (positive predictive value) = {TP/ (TP+FP)} = 56%

Recall (also called sensitivity) = {(TP/(TP+FN)} = 16%

F1 Score = {2TP/(2TP+FN+FP)} = 24%

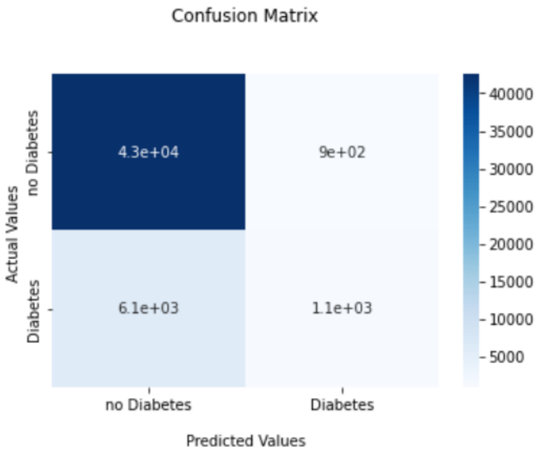
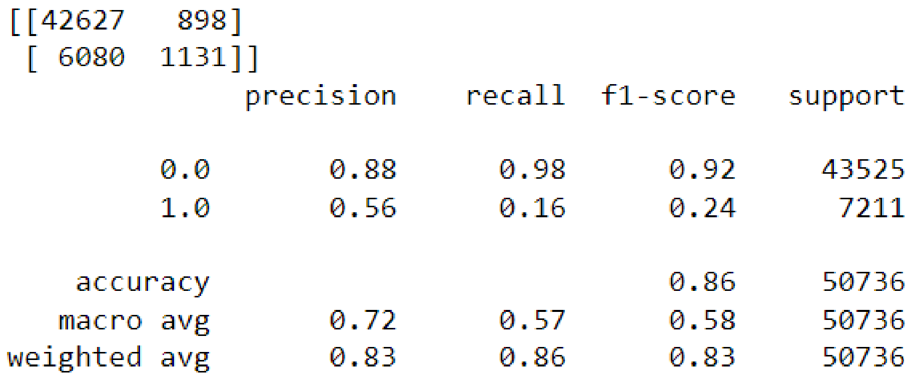


Fig. 6a



```
[[42627   898]
 [ 6080  1131]]
              precision    recall  f1-score   support

         0.0       0.88      0.98      0.92     43525
         1.0       0.56      0.16      0.24      7211

    accuracy                           0.86     50736
   macro avg       0.72      0.57      0.58     50736
weighted avg       0.83      0.86      0.83     50736
```

Fig. 6b

ROC (Receiver Operation Characteristics) curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. The best possible prediction method would yield a point in the upper left corner of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). See Fig.7

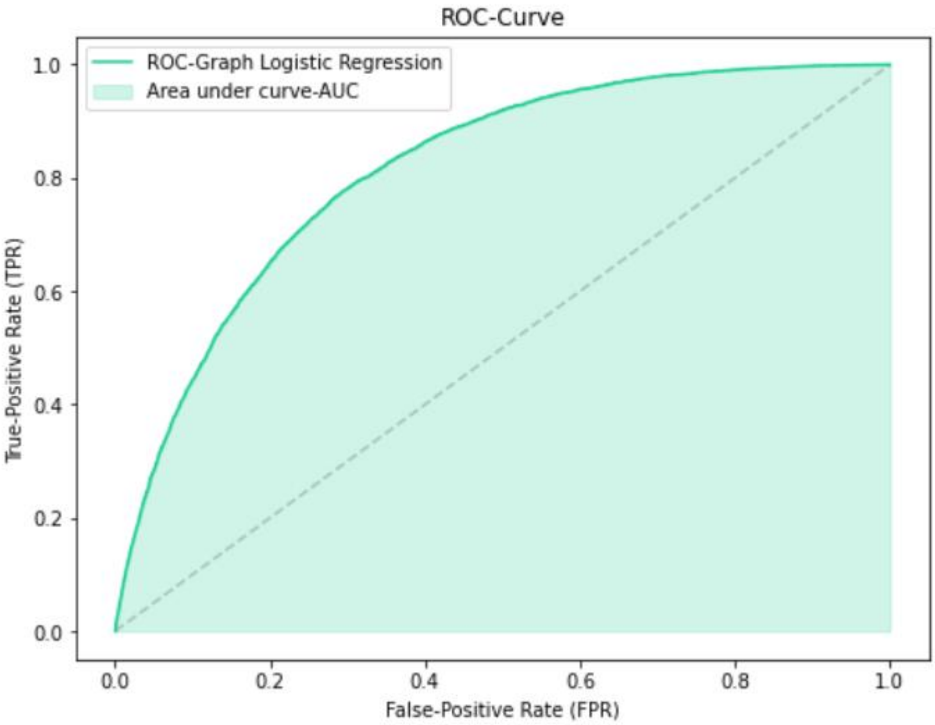


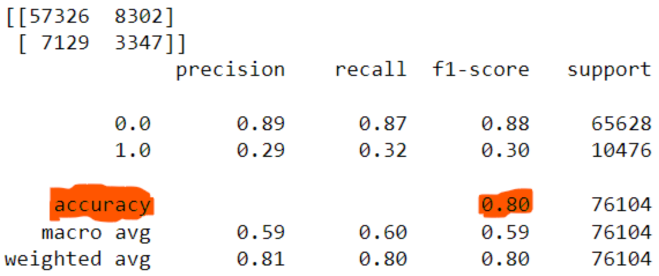
Fig. 7

ROC is a probability curve and AUC (Area Under Curve) represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 class as 0 and 1 class as 1. From Fig.7 above, the non-thresholded decision values given by the: `LogisticRegression(solver="liblinear").fit(X, y)` returns a value of 0,82 meaning that the model has successfully distinguished the two classes with a success rate of 82%.

Decision Tree Classifier

A Decision Tree estimates the probability that an instance belongs to a particular class k. First it traverses the tree to find the leaf node for this instance, and then it returns the ratio of training instances of class k in this node. `The DecisionTreeClassifier` has a few other parameters that regularize the Decision Tree: `min_samples_split` (the minimum number of samples a node must have before it can be split), `min_samples_leaf` (the minimum number of samples a leaf node must have), `min_weight_fraction_leaf` (same as min_samples_leaf but expressed as a fraction of the total number of weighted instances), `max_leaf_nodes` (maximum number of leaf nodes), `and max_features` (maximum number of features that are evaluated for splitting at each node [8].

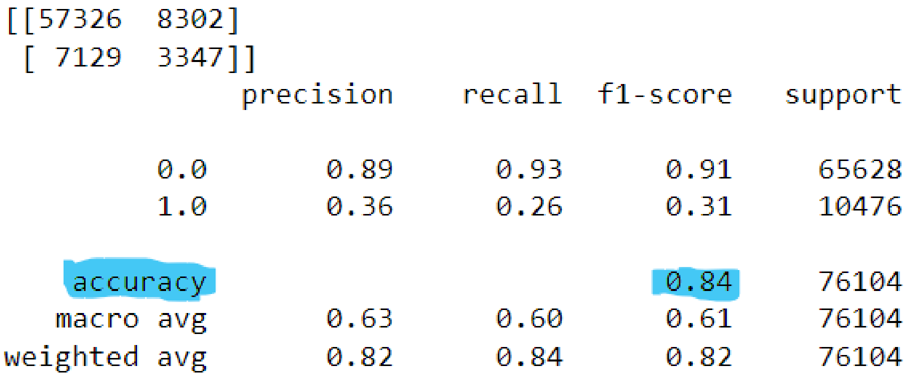
Following this principle, a decision tree classifier is trained resulting in the following output:



```
[[57326  8302]
 [ 7129  3347]]
              precision    recall  f1-score   support

         0.0       0.89      0.87      0.88     65628
         1.0       0.29      0.32      0.30     10476

    accuracy                           0.80     76104
   macro avg       0.59      0.60      0.59     76104
weighted avg       0.81      0.80      0.80     76104
```

Fig. 8a

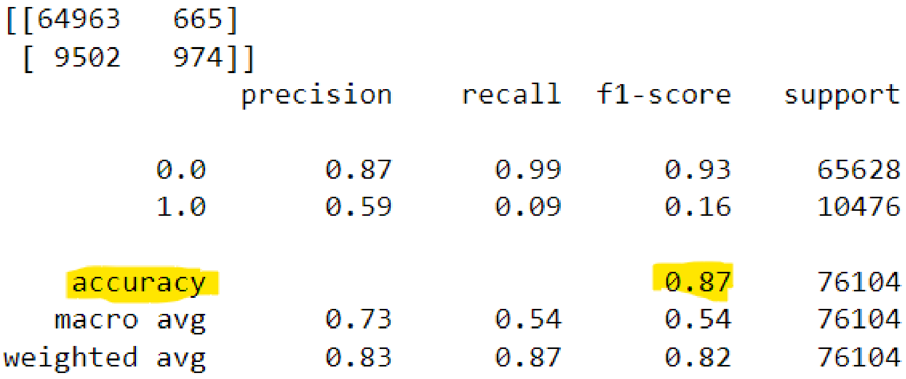
To avoid repeating what has been discussed already under other models,only accuracy will be given a closer attention to show how the metrics are optimised using different parameters in subsequent models. From the results above, accuracy is 80%. An attempt to regularize the tree with the introduction of `criterion`, `max_depth and min_samples_leaf`, improved the accuracy to 84%. See Fig. 9 below;



```
[[57326  8302]
 [ 7129  3347]]
              precision    recall  f1-score   support

         0.0       0.89      0.93      0.91     65628
         1.0       0.36      0.26      0.31     10476

    accuracy                           0.84     76104
   macro avg       0.63      0.60      0.61     76104
weighted avg       0.82      0.84      0.82     76104
```

Fig. 8b

A further optimization process using `GridSearchCV`, is implemented. This improves the accuracy to 87%. See Fig. 9c below;



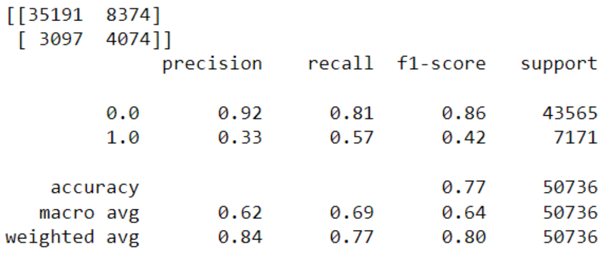
```
[[64963   665]
 [ 9502   974]]
              precision    recall  f1-score   support

         0.0       0.87      0.99      0.93     65628
         1.0       0.59      0.09      0.16     10476

    accuracy                           0.87     76104
   macro avg       0.73      0.54      0.54     76104
weighted avg       0.83      0.87      0.82     76104
```

Fig. 8c

Naive Bayes classifier

Naive Bayes classifiers are probabilistic classifiers (able to predict, given an observation of an input, a probability distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to) which assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

After splitting the data into train and test dataset, the model is trained using the Gaussian Naives Bayes classifier and the results are presented in the confusion matrix and classification report below.



```
[[35191  8374]
 [ 3097  4074]]
              precision    recall  f1-score   support

         0.0       0.92      0.81      0.86     43565
         1.0       0.33      0.57      0.42      7171

    accuracy                           0.77     50736
   macro avg       0.62      0.69      0.64     50736
weighted avg       0.84      0.77      0.80     50736
```

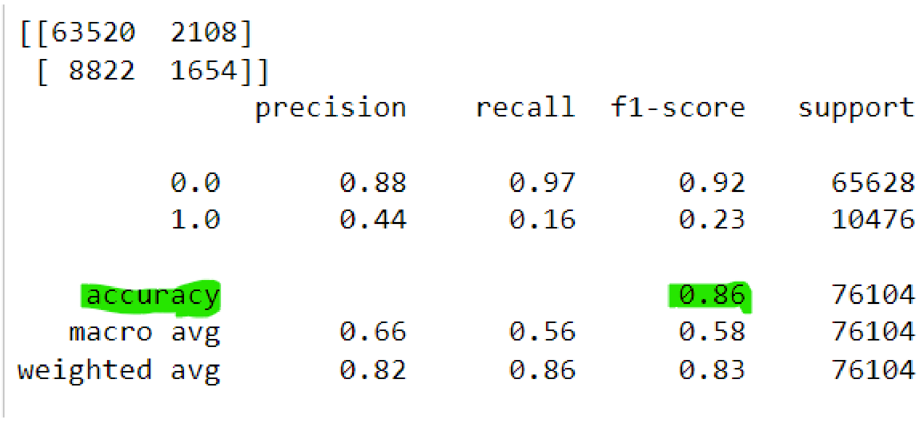
Fig. 9

The accuracy score of this model is 77%, and less than that of the logistic regression. It is worth noting that the accuracy can be optimised but will not be dealt with in this work.

K- Nearest Neighbors (KNN)

KNN models classify an unknown data point based on the classes of the data points that are in the immediate vicinity to it. As such the distances between the unknown and its nearest K-neighbors are of paramount importance in determining the performance of the model. The optimal K value is determined iteratively. See Fig.11 below.

A model training with an arbitrary K = 9 value returns an accuracy of 86% as shown in the classification report below in Fig. 10



```
[[63520  2108]
 [ 8822  1654]]
              precision    recall  f1-score   support

         0.0       0.88      0.97      0.92     65628
         1.0       0.44      0.16      0.23     10476

    accuracy                           0.86     76104
   macro avg       0.66      0.56      0.58     76104
weighted avg       0.82      0.86      0.83     76104
```

Fig. 10

Fig. 11 below shows how the accuracy of the KNN model is dependent on the K-value. K = 25 appears to be the best K-value to achieve maximum performance of the model with an accuracy of about 87%. It must be noted here that, it took more than two hours for this model to train with different K-values.

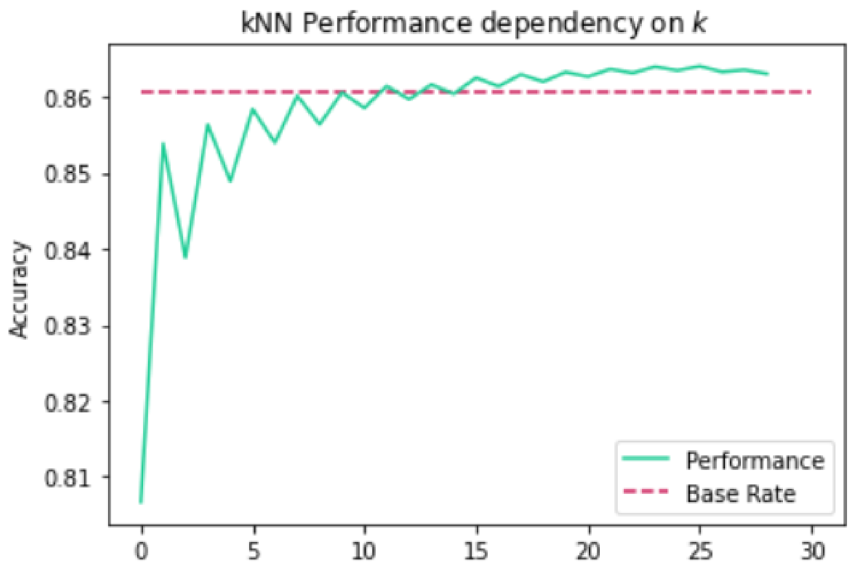
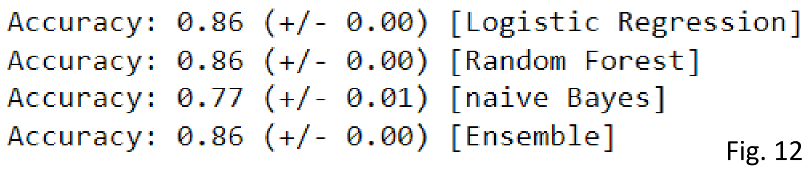


Fig. 11

Ensemble Methods

The accuracy score of the following Ensemble methods will be compared. RandomForestClassifiers, BaggingClassifiers, AdaBoostClassifiers, StochasticGradientBoosting and VotingClassifier. While Bagging methods build several instances of a black-box estimator on random subsets of the original training set and then aggregate their individual predictions to form a final prediction, AdaBoost fits a sequence of weak learners such as small decision trees, on repeatedly modified versions of the data [10]. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. Based on this background information, their models were trained and they return the following results; Fig.12 shows the results of a VotingClassifier on three models.

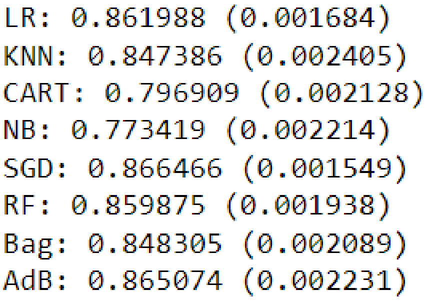


```
Accuracy: 0.86 (+/- 0.00) [Logistic Regression]
Accuracy: 0.86 (+/- 0.00) [Random Forest]
Accuracy: 0.77 (+/- 0.01) [naive Bayes]
Accuracy: 0.86 (+/- 0.00) [Ensemble]
```

Fig. 12

Conclusion and recommendations

All the models that have been trained and tested show above 75% accuracy score, meaning that they can be used to predict diabetes based on the available data as well as would rightly predict on similar future data. According to the CrossValidation results below (Fig.13), AdaBoost and StochasticGradientDescent seem to perform better than the other models with an accuracy score of 86% each. However, a conclusion must be drawn with caution because it has also been demonstrated here that the performance of KNN can be optimized to 86% which is not represented here.



```
LR: 0.861988 (0.001684)
KNN: 0.847386 (0.002405)
CART: 0.796909 (0.002128)
NB: 0.773419 (0.002214)
SGD: 0.866466 (0.001549)
RF: 0.859875 (0.001938)
Bag: 0.848305 (0.002089)
AdB: 0.865074 (0.002231)
```

Fig. 13

With respect to the feature importance as seen on Pearson correlation matrix, it could be observed that some few features might have highly contributed to the outcome more than others. Hence, the possibility of a person becoming diabetic in relation to his behavioural habits seems to be influenced by general health condition (GenHlth), Age, Cholesterol check in five years (Cholcheck) , High blood Pressure(HighBP).

It is curious to find out why High cholesterol seems to have no impact on diabetes meanwhile cholesterol check in five years seems to influence the possible occurrence of diabetes.

This work is not exhaustive and gives room for further in-depth investigation on the performances of the classification algorithms tested here. Support Vector Machine algorithm could not be tested due to the extensive time needed to train the model.

References

[1] https://www.who.int/news-room/fact-sheets/detail/diabetes

[2] Emerging Risk Factors Collaboration, Sarwar N, Gao P, Seshasai SR, Gobin R, Kaptoge S, Di Angelantonio E, Ingelsson E, Lawlor DA, Selvin E, Stampfer M, Stehouwer CD, Lewington S, Pennells L, Thompson A, Sattar N, White IR, Ray KK, Danesh J.

Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies. Lancet. 2010 Jun 26;375(9733):2215-22. doi: 10.1016/S0140-6736(10)60484-9. Erratum in: Lancet. 2010 Sep 18;376(9745):958. Hillage, H L [corrected to Hillege, H L]. PMID: 20609967; PMCID: PMC2904878.

[3]https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators dataset?select=diabetes_binary_health_indicators_BRFSS2015.csv

[4] Practical Machine Learning with Python. A Problem-Solver's Guide to Building Real-World Intelligent Systems. 2018. by Dipanjan Sarkar, Raghav Bali,Tushar Sharma

[5] Data Science for Business by Foster Provost and Tom FawcettCopyright © 2013 Foster Provost and Tom Fawcett.

[6] Data Science and Machine Learning Mathematical and Statistical Methods. Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, Radislav Vaisman. 8th May 2022

[7] Swets, John A.; Signal detection theory and ROC analysis in psychology and diagnostics : collected papers, Lawrence Erlbaum Associates, Mahwah, NJ, 1996

[8] Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron

Copyright © 2017 Aurélien Géron

[9] https://scikit-learn.org/stable/modules/ensemble.html

[10] https://en.wikipedia.org/wiki/Bootstrap_aggregating

List of Figures: