B. V. College Of Engineering

New Delhi

IN-HOUSE SUMMER TRAINING PROJECT

"Evaluating the Performance of Machine Learning Algorithms for Diagnosing Diabetes in Individuals"

MENTORS:

Anurag Agarwal

Pranav Das

Group:

Aditya Dogra (ECE-2)

Avi Bhatia (ECE-2)

Pranav Pushkar (CSE-M)

Mahesh Kr. Sharma (ECE-2)

Project Report

Evaluating the Performance of Machine Learning Algorithms for Diagnosing Diabetes in Individuals

Abstract

Application of machine learning algorithms for the diagnosis of diabetes has become a trending research area, as an effort to improve current techniques and methods used by healthcare institutions to determine the occurrence of diabetes in individuals is now given more attention than before.
The diabetes dataset is a binary classification problem where it needs to be analysed whether a patient is suffering from the disease or not on the basis of many available features in the dataset. Different methods and procedures of cleaning the data, feature extraction, feature engineering and algorithms to predict the onset of diabetes are used based for diagnostic measure on Pima Indians Diabetes Dataset.

Keywords

machine learning; Pima Indians Diabetes dataset; binary classification; features; feature extraction; feature engineering; support vector machine; MLP; neural networks; Decision tree; Linear regression heat map; pair plot; violin plot; feature importance.

Overview of Summer training

- The training of "Data Science with python and R" started on 17 July 2019.
- In the first week of the training we got to know about what is data science, then we got to learn the basics of python, saw some basic projects on how to apply data science using python. Later in the first week we learnt about NumPy, Pandas, Matplotlib.

NUMPY- NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open source module of Python which provides fast mathematical computation on arrays and matrices.

PANDAS-It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas is capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs.

MATPLOT-LIB Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

- In the second week we got to know about more libraries of python which helped in data visualisation like seaborn, Scikit-learn. Later we learnt about machine learning algorithms like supervised (Linear regression, logistic regression, knn, decision tree), unsupervised learning(clustering).
- In third week, we got introduction to R programming language. We learnt all the same data visualisation and machine learning activities with the help of R this time.
- We also got introduction to data mining, sentiment analysis of datasets using R by importing various built-in packages to implement those data mining techniques.
- Last week had some tests and presentations about the things we learnt.

1. Definition

1. Project Overview

Technological improvements in the field of science and health care has given rise to the application of computer aided systems and applications in handling medical and health issues. Medical practitioners which include doctors, laboratory specialists, nurses are faced with hundreds of situations where they have to make a decision about the health condition of their patients based on the patient's symptoms and signs.

With the increasing need for efficient health care and the need for timely decisions, it is obvious that the traditional method of sieving knowledge from records can no longer be sustained as this result to delay and errors in medical decisions. The amount of time required for laboratory diagnostic results to be available must be optimized for better health care and for timely decision making. Data mining as a branch of computer science has evolved to assist medical personnel in performing their functions more effectively. With the availability of large amount of patient information in health organizations, decision making as regards patient's condition can be more optimized and made faster through data mining knowledge discovery techniques. Many computational tools and algorithms have been recently developed to increase the experiences and abilities of physicians for taking decisions about different diseases.

In this study, an evaluation of the performance of machine learning classifiers in predicting diabetes disease in individuals is analysed using experimental and statistical procedures. Classification is a form of data analysis that extracts models describing important data classes. Such models called classifiers predict categorical (discrete, unordered) class labels [3]. Classification splits a dataset into mutually exclusive groups called a class based on suitable attributes [4]. Some of the numerous applications of classification include fraud detection, target marketing, performance prediction, manufacturing and medical diagnosis. Machine learning models can show the result of a patient's test with a pre-test probability (of the population), to predict or determine the chance of finding a particular disease. Therefore, the

aim of this study is to evaluate the performance of machine learning classifiers in predicting and diagnosing diabetes using historical patient data. The completed study will provide a clear understanding of the data mining process for medical diagnosis, and also a confidence level on the application of machine learning models on diabetic patient data to determine their status.

The Pima Indians dataset used for this study was obtained from the UCI machine learning repository. It is listed under the "Pima-Indians-diabetic" data name. The dataset was originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases. The data set is made up of 768 number of instances with nine (8) attributes (features) for each instance of the dataset. The attributes in the dataset are listed as follows, and table 3.1 shows a sample of the dataset.

· Number of Times Pregnant (preg)
· Plasma Glucose Concentration (plas)
· Diastolic Blood Pressure (mm Hg) (bld pres)
· Triceps Skin fold thickness (mm) (skin fold)
· 2-hour serum insulin (insulin)
· Body mass index (Kg/m2) (bmi)
· Diabetes Pedigree Function (pedi)
· Age(years) (age)
· Class Variable (class)

Here we have applied various machine learning algorithms on the following dataset and determined the accuracy of different machine learning algorithms. Then we applied various feature selection techniques with the help of correlation matrix and random forest classifier. With the help of random forest classifier, we obtained some essential features such as Bmi, Glucose, Age, Diabetes, Pedigree Function and outcome. Here outcome is a class variable which is actual label and here 1 denotes that person has diabetes and 0 denotes person does not have diabetes. Our dataset has 768 variables where 268 are those having where person has diabetes and remaining belong to those not having diabetes.

Basically, goal of this project is to emphasize on machine learning aspect of the problem. Hence, we are working with the labelled data. To test the performance of the model we are determining the accuracy and we are finding the best machine learning algorithm for our problem statement.

1.2Problem Statement

Diabetes mellitus is a chronic disease caused by inherited and/or acquired deficiency in production of insulin by the pancreas, or by the ineffectiveness of the insulin produced. Such a deficiency results in increased concentrations of glucose in the blood, which in turn damage many of the body systems, in particular the blood vessels and nerves.

Diabetes is deadly if there is no prompt diagnosis and offers a patient less lifeline of active living as it usually results to health complications such as heart attack, stroke, blindness, blindness, kidney failure, heart attack, stroke and lower limb amputation. Although early diagnosis has been identified as a major step in the fight against the dangers of diabetes, medical personnel are often in situations where analysis of a patient test result may pose a challenge.

Thus, the availability of an enhanced data mining algorithm for early diagnosis will definitely help to advert the dangers of late diagnosis and improve the management of diabetic cases.

We begin our investigation by analysing the dataset for any null values or any outliers. Then after calculating the accuracy for different machine learning algorithms and feature extraction we applied standard scaling method to ensure that all the features are treated equally when applying supervised learners

The next step is to establish a benchmark model that we can use to make performance comparisons. For this problem we define our benchmark model as naïve classifier, which classifies all the people having diabetes. We aim that our designed model should perform better than the naïve predictor. Since, this naïve prediction model does not consider any information to substantiate its claim, it helps establish a benchmark for whether a model is performing well.

Thereafter, we will investigate different supervised learning algorithms and determine which is best at modelling the data. The various algorithms that we will consider for evaluation are, Gaussian Naïve Bayes, Decision Tree, Support Vector Machines, Logistic Regression and K-nearest neighbour. We will compare each algorithm with naïve predictor to evaluate its performance.

1.3Metrics

We will use two evaluation metrics to quantify the performance of our solution as well as the benchmark model, viz., accuracy and F-beta score.

Accuracy= (number of samples correctly classified) / (total number of samples)

This can be a good metric, but suppose, if it's more detrimental for us if we are not able to correctly classify a diabetic patient, than a non-diabetic one. Therefore, the model's ability to recall all diabetic patients is more important than the model's ability to make precise prediction. In this scenario, we can use, Fβ score with β = 2, as it weighs recall more than precision. The general form is:

$$F\beta=(1+\beta^2).\ ((precision.\ recall)/((\beta^2.precision)+recall)).$$

2.Analysis

2.1Data Exploration

Pima Indian Diabetes dataset has 9 attributes in total. All the person in records are females and the number of pregnancies they have had has been recorded as the first attribute of the dataset. Second is the value of Plasma glucose concentration 2 hours in an oral glucose tolerance test and then is the Diastolic blood pressure (mm Hg), fourth in line is the Triceps skin fold thickness (mm), then is the 2-Hour serum insulin (mu U/ml), sixth is Body mass index (weight in kg/ (height in m) ^2) and then seventh is the Diabetes pedigree function and

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Table 1.First Five Records of the dataset

the second last value is that of the Age (years). The ninth column is that of the Class variable (0 or 1), 0 for no diabetes and 1 for the presence.

A brief description of the dataset, including parameters like mean, min, max for each column is given in table 2.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

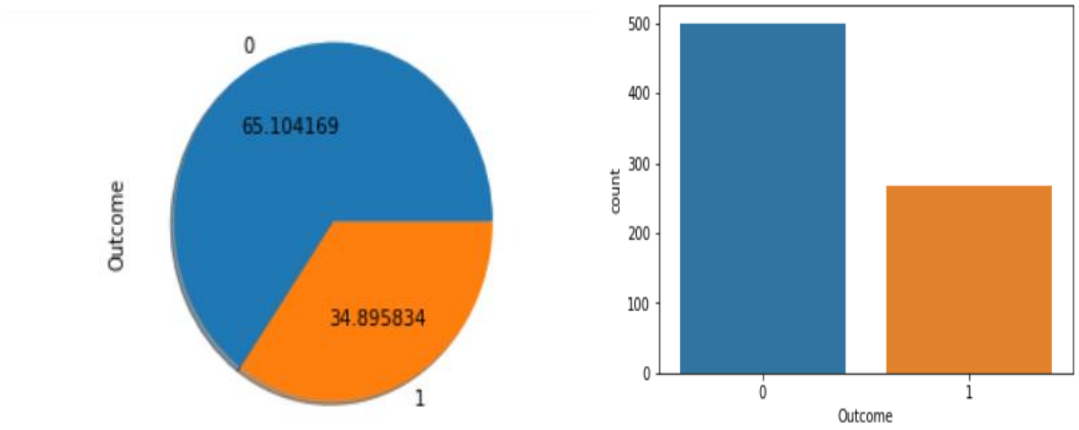
Table 2. Description of features of the dataset

We note the following points from the above description: -

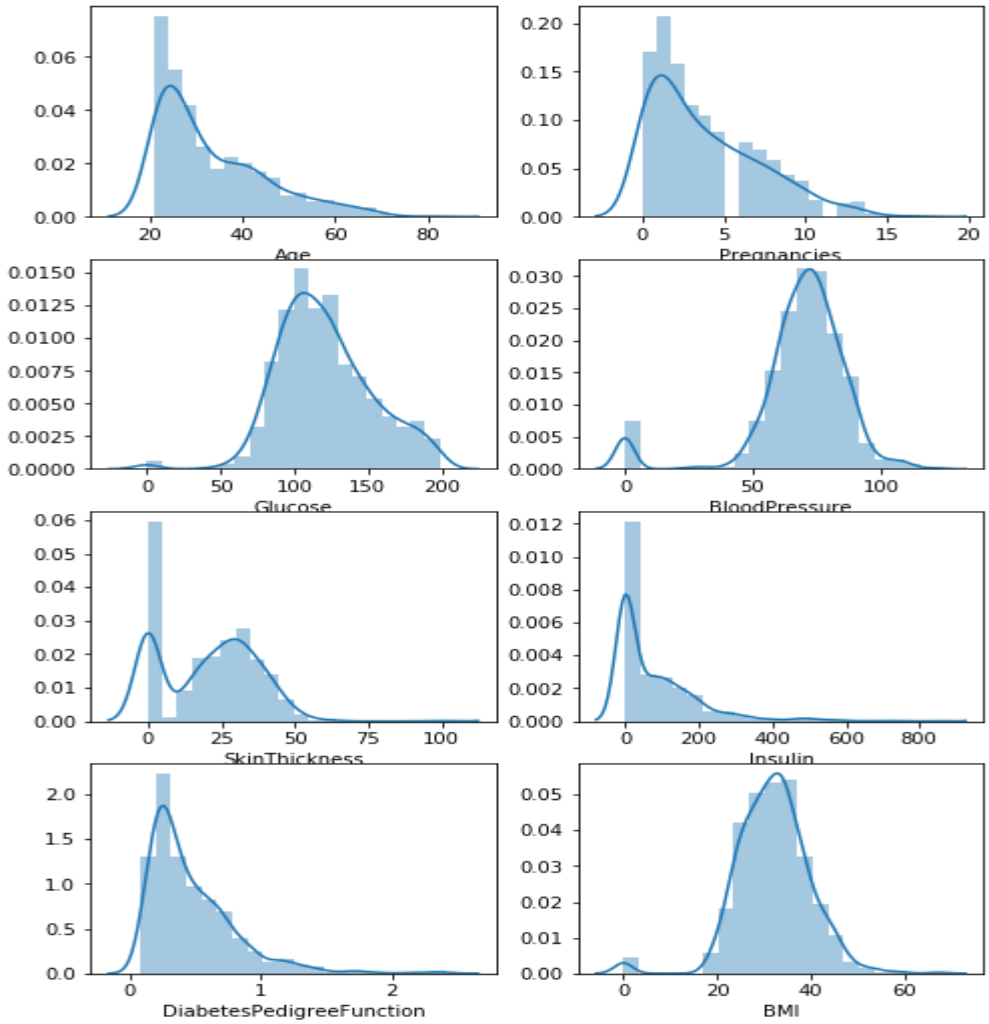
1.All the above feature have different mean and standard deviation.
2. Our target feature ‘outcome’ is binary valued, with value ‘0’ for not having diabetes, and ‘1’ for having diabetes.
3. We also observe from the exploration of data that it does not have any missing value.

2.2Exploratory Visualization

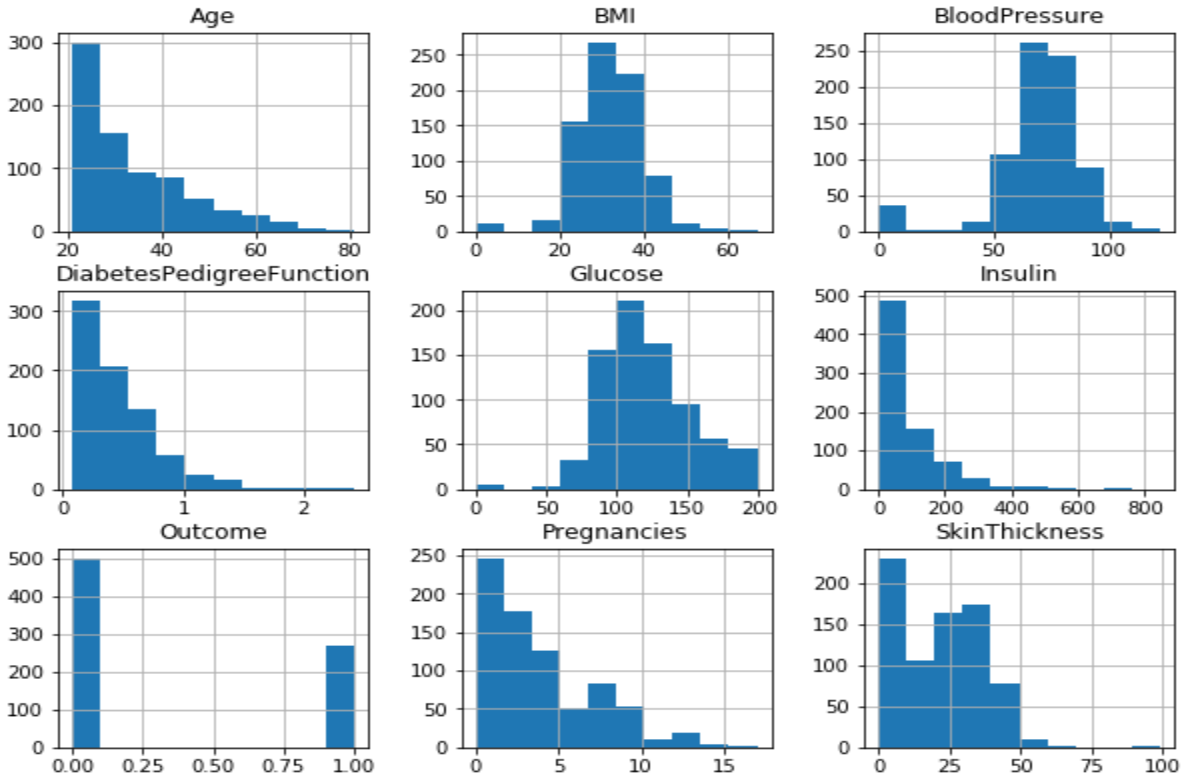
A dataset may sometimes contain at least one feature whose values tend to lie near a single number, but will also have a non-trivial number of vastly larger or smaller values than that single number. Algorithms can be sensitive to such distributions of values and can underperform if the range is not properly normalized.



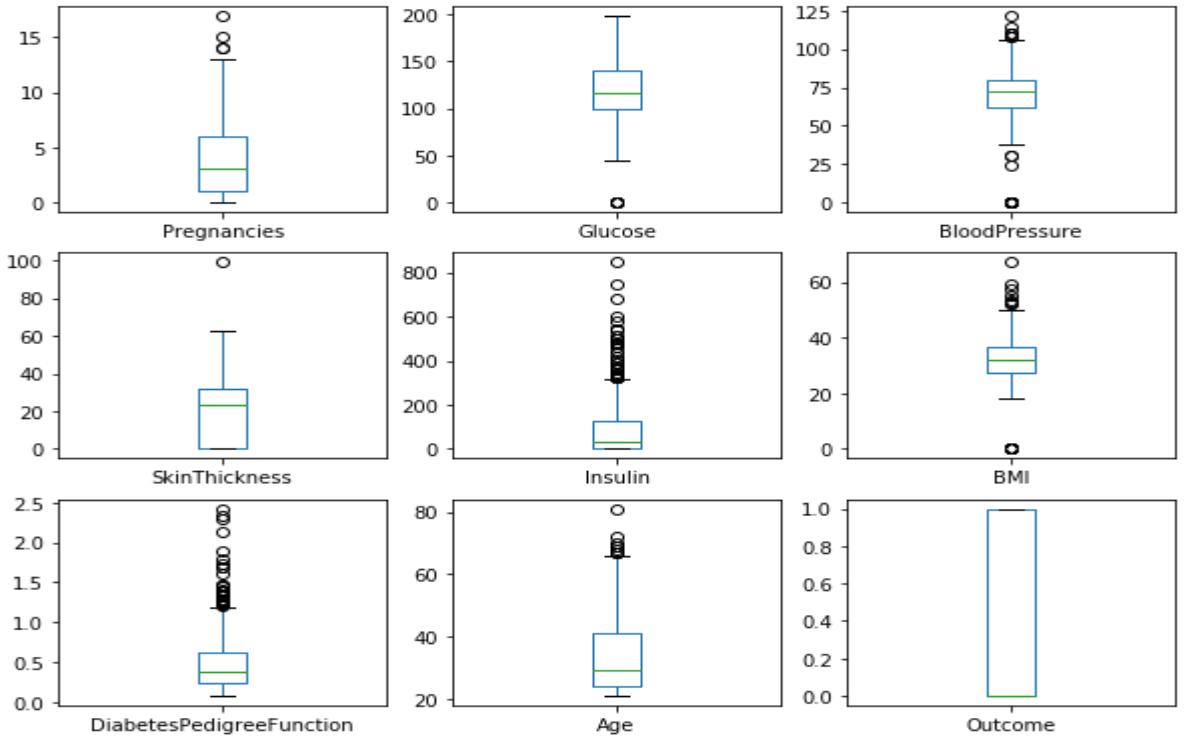
Above is a pie and bar plot of same representation and that is the total proportion of patients with diabetes(1 or Yellowish brown) and patients without diabetes(0 or Blue).



Above is a distribution plot for our dataset. a density is scaled so that the area under the curve is 1, so no individual bin. will be taller than 1. all the attributes are accordingly plotted along x-axis.



Above is histogram plot which gives us a statistical measure of the complete dataset and in what frequency as well as range all the attributes lie.



Above is a box plot of the dataset that we want to work on. it is a measure of how well distributed is the data in a data set. it divides the data set into three quartiles. this graph represents min, max, median, first and third quartile in the data set.

2.3Algorithms and Techniques

Five supervised learning approaches are selected for this problem. These algorithms are chosen such that their approaches are fundamentally different from each other, so that we can cover a wide spectrum of possible approaches. We consider the following five algorithms for analysis, and would compare their performance with our benchmark model.

1. SVM
2. Logistic Regression
3. KNN
4. Decision tree
5. Gaussian Naïve Bayes

1. Support Vector Classifier

Support Vector Machine is a supervised learning technique that constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Some of the various parameters that it manipulates, include:

- kernel: It defines the type of kernel to be used, like, 'linear', 'poly', 'rbf', 'sigmoid'.
- C: Penalty parameter of the error term.
- degree: Degree of the polynomial kernel function ('poly'). Default is 3, ignored by other kernels

Advantages:

- Effective in high dimensional spaces.
- Performs well with non-linear decision boundaries if appropriate kernel is used.
- Also effective where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different kernel functions can be specified for the decision function.

Disadvantages:

- If the number of features is much greater than the number of samples, the method is likely to give poor performances.
- Expensive to train.

2.Logistic Regression

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Logistic regression is going to work for the current problem, since it is a binary classification problem.

2.3.3 KNN

kNN classified an object by a majority vote of the object's neighbours, in the space of input parameter. The object is assigned to the class which is most common among its **k (an integer specified by human) nearest neighbour**.
It is a **non-parametric, lazy algorithm**. It's non-parametric since it does not make any assumption on data distribution (the data does not have to be normally distributed). It is lazy since it does not really learn any model and make generalization of the data (It does not train some parameters of some function where input X gives output y).
So strictly speaking, this is not really a learning algorithm. It simply classifies objects based on **feature similarity** (feature = input variables).

Advantages: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

Disadvantages: Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

3.Decision Tree

Decision Tree, as its name says, makes decision with tree-like model. It splits the sample into two or more homogeneous sets (leaves) based on the most significant differentiators in your input variables. To choose a differentiator (predictor), the algorithm considers all features and does a binary split on them (for categorical data, split by cat; for continuous, pick a cut-off threshold). It will then choose the one with the least cost (i.e. highest accuracy), and repeats recursively, until it successfully splits the data in all leaves (or reaches the maximum depth)

Advantages: Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

2.3 Benchmark Model

A benchmark model is defined to provide a threshold and compare the performance of the solution obtained by the trained model. For the Diabetes identification problem, we will define our benchmark to be a naïve classifier, which classifies all the people having diabetes. The accuracy of this naïve classifier will be the number of samples of people having diabetes to the total number of samples in the data. Our model should perform far better than the naïve approach to be any worthy for intended use. Since, this naïve prediction model does not consider any information to substantiate its claim, it helps establish a benchmark for whether a model is performing well.
We calculate the accuracy of the naïve predictor model as follows:

Accuracy= Samples correctly classified/Total Number of Samples

$$= \frac{No\ of\ people\ with\ diabetes}{Total\ Number\ of\ samples} = \frac{268}{768} = 0.3489$$

So, for the benchmark smodel had an accuracy of 0.3489.

3.METHODOLOGY

3.1 Implementation

3.1.1 Splitting(1st)

The reason we perform splitting is to test the trained model over samples which it has never seen before. This way, we make sure that the model has extracted classification patterns from the training samples and has not memorized them. For the given problem, to split our data, we use a function train_test_split available in model_selection module of sklearn library. This function does two tasks
which are important in the current context.

1. It shuffles the dataset so that both training and test sets have nearly equal number of samples
from both classes.
2. After shuffling, it performs the split. We can specify what fraction of the total data to be included in the training or test set.

We make split such that training set has 75% samples and test set has 25% samples.

3.1.2 Creating training and prediction pipeline

We define a method called 'train_split' that takes as input the following parameters: learner, sample_size, X_train, y_train, X_test, y_test. It returns the accuracy and on training and test set.

The algorithms that were applied were linear SVM, radial SVM, Logistic regression, K-nearest neighbours, Decision tree and naive Bayes.

Their Accuracies are as follows:

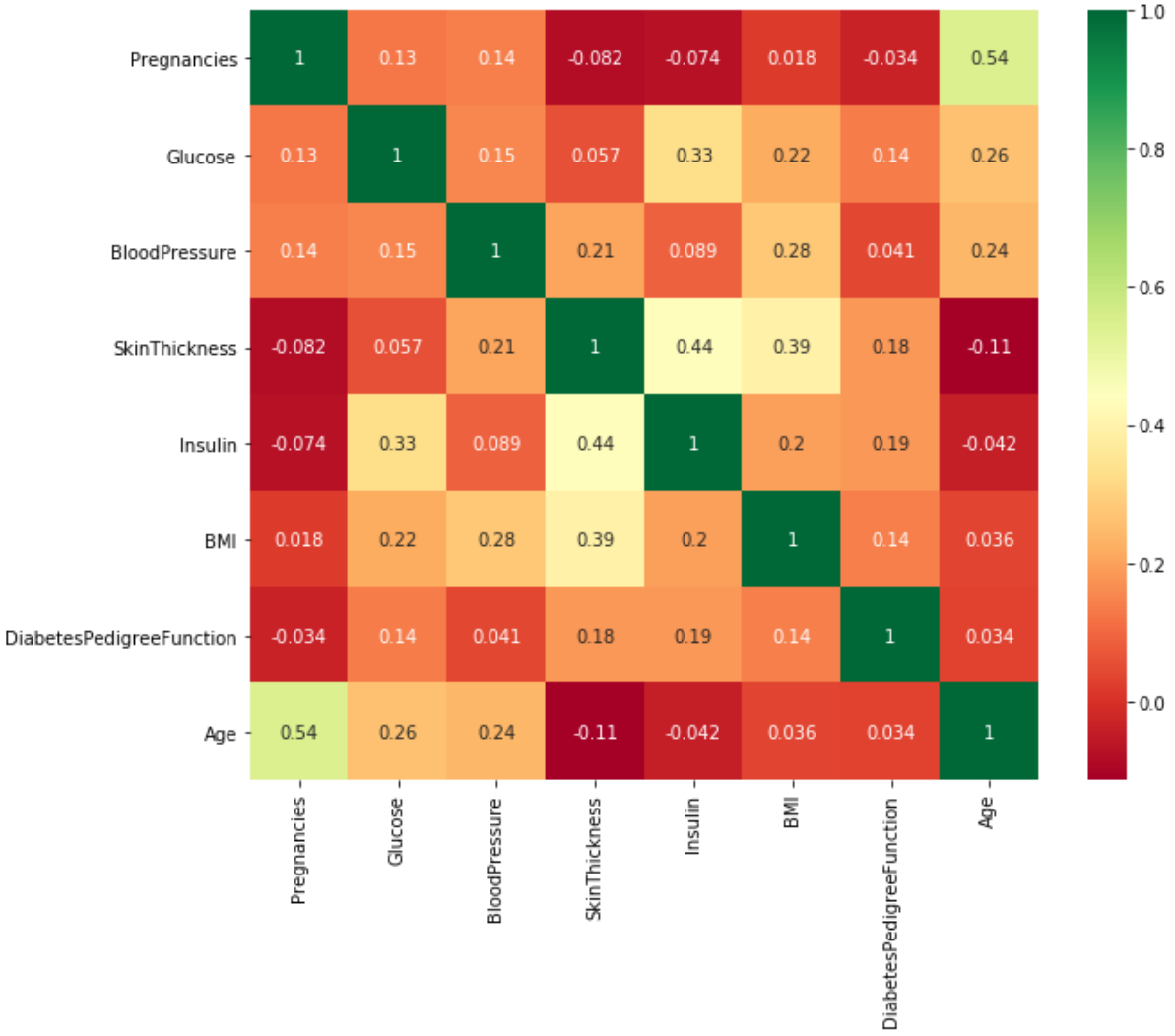
	Accuracy
Linear Svm	80.208333
Radial Svm	67.708333
Logistic Regression	80.729167
KNN	75.520833

Decision Tree	72.916667
Naive Bayes	76.562500

The above algorithms are not giving very high accuracy. This can be improved by using Feature Selection and using only relevant features.

3.1.3. Feature Extraction/Selection

1)A lot many features can affect the accuracy of the algorithm.
2)Feature Extraction means to select only the important features in-order to improve the accuracy of the algorithm.
3)It reduces training time and reduces overfitting
4)We can choose important features in 2 ways:
a) Correlation matrix--> selecting only the uncorrelated features.
b) Random Forest Classifier--> It gives the importance of the features



All the features look to be uncorrelated. So we cannot eliminate any features just by looking at the correlation matrix.

Thus, we will now use random forest classifier method for feature selection/extraction. from sklearn.ensemble we imported RandomForestClassifier method and then determined feature importance of each feature as follows:

Glucose	0.242098
BMI	0.172574
Age	0.135220
DiabetesPedigreeFunction	0.128324
BloodPressure	0.092903
Pregnancies	0.086774
SkinThickness	0.073109
Insulin	0.068999

The important features are: Glucose, BMI, Age, DiabetesPedigreeFunction.

3.1.4 Standardisation

There can be a lot of deviation in the given dataset. An example in the dataset can be the BMI where it has 248 unique values. This high variance has to be standardised. Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

From sklearn.preprocessing we imported StandardScaler method and then performed standard scaling on our data.

3.1.5 Splitting(2^{nd})

This time we again split our data into training and test set after selecting useful features and performing standardization.
Here we also used test size of 0.25 in which training set has 75% samples and test set has 25% samples.

3.1.6 Predictive Analysis

After than we determined new accuracies for different classifiers such as linear svm, radial svm, Logistic Regression, KNN, Decision Tree and Naïve Bayes.
The results of the following are as follows:

	New Accuracy	**Accuracy**	**Increase**
Linear Svm	78.125000	80.208333	-2.083333
Radial Svm	77.083333	67.708333	9.375000
Logistic Regression	77.604167	80.729167	-3.125000
KNN	75.520833	75.520833	0.000000
Decision Tree	73.437500	72.916667	0.520833
Naive Bayes	78.125000	76.562500	1.562500

The above table shows the new accuracy after feature selection. Here we observed that there is a decrease in accuracy for linear svm and logistic regression classifiers by -2% and -3% whereas there is an increase in the accuracy of Radial Svm, Decision Tree and Naïve Bayes by 9%,0.52%,1.56% respectively.

4.Results

We considered six classifiers problems like linear svm, radial svm, Logistic Regression, KNN, Decision Tree and Naïve Bayes for the given problem. We evaluated the performance of these classifiers using accuracy metric. We also performed K-fold cross validation to a achieve a generalized model. In the end we plotted we also plotted boxplot to compare different classifiers.

4.1 K-Fold Cross Validation

Many a times, the data is imbalanced, i.e. there may be a high number of class1 instances but a smaller number of other class instances. Thus, we should train and test our algorithm on each and every instance of the dataset. Then we can take an average of all the noted accuracies over the dataset.

1)The K-Fold Cross Validation works by first dividing the dataset into k-subsets.

2)Let's say we divide the dataset into (k=5) parts. We reserve 1 part for testing and train the algorithm over the 4 parts.

3)We continue the process by changing the testing part in each iteration and training the algorithm over the other parts. The accuracies and errors are then averaged to get a average accuracy of the algorithm.

This is called K-Fold Cross Validation.

4)An algorithm may underfit over a dataset for some training data and sometimes also overfit the data for other training set. Thus, with cross-validation, we can achieve a generalized model.

Here we imported KFold from sklearn.model_selection for k-fold cross validation and cross_val_score from sklearn.model_selection. Taking k=10, we split our data into 10 parts.

For Example: -If k=5 the dataset will be divided into 5 equal parts and the below process will run 5 times, each time with a different holdout set.
1. Take the group as a holdout or test data set
2. Take the remaining groups as a training data set
3. Fit a model on the training set and evaluate it on the test set
4. Retain the evaluation score and discard the model
At the end of the above process Summarize the skill of the model using the sample of model evaluation scores.

The Cross-validation scores table is as follows:

	CV Mean
Linear Svm	78.125000
Radial Svm	77.083333
Logistic Regression	77.604167
KNN	75.520833
Decision Tree	73.437500
Naive Bayes	78.125000

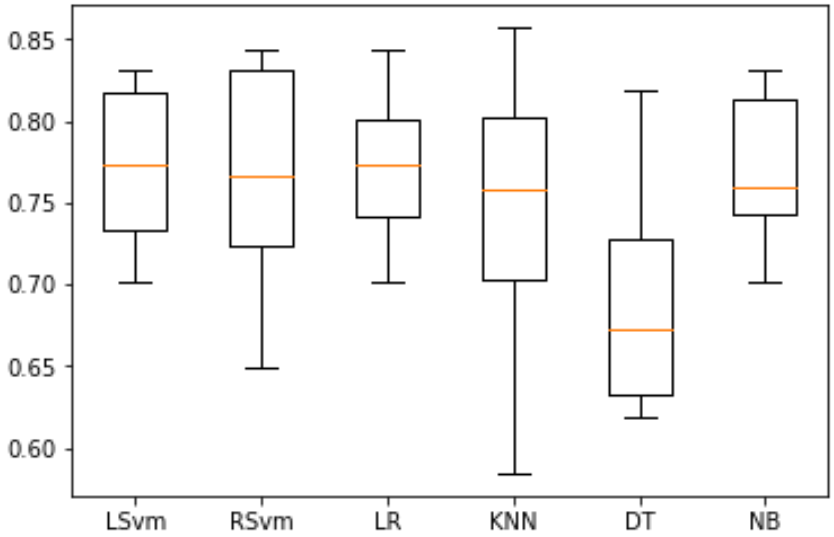


Fig.Boxplot comparing different classifiers

From this boxplot we infer that for our model Svm perform the best while Decision Tree performs the worst.

4.2 Justification

The benchmark model for the given problem is a naïve predictor which predicts every person is having diabetes. The evaluation metrics for the benchmark was computed as follows:

Accuracy = 0.3489,

The evaluation metrics for our solution model of Linear Svm Classifier is as follows:

Accuracy = 0.78125

This shows that our solution model way outperforms the benchmark in terms of evaluation metrics.

Therefore, our solution model is way stronger than the benchmark model, as it uses domain information to make informed predictions and is also quite robust in its performance. We believe the model can be adopted for real use, and even mobile apps can be developed for detection of diabetes in individuals.

5.Conclusion

5.1 Reflection

The problem began with identifying an interesting dataset that could solve a real-world problem. Pima Indian dataset available on UCI machine learning repository was a good

choice. We performed statistical analysis and reflected upon the features of the dataset. We then made some visualizations which reflected upon the relationship among various features. Further, we split the data into training and test set. We took 25% of data to be in test set. We also made a benchmark model. A naïve predictor that classified all people having diabetes was taken to be as benchmark. We defined metrics that we used to compare the performance of our model. We took accuracy and to be the performance metrics.
Then we chose six algorithms like linear svm, radial svm, KNN, Logistic Regression, Decision Tree and Naïve Bayes and calculated the accuracies. Then to improve the accuracy we performed feature extraction using random forest classifier and correlation matrix. We also cross validated our results to get a generalized model and to overcome the problem of overfitting.
It was interesting to note that algorithms like Svm and even naïve Bayes performed the best. The results are sufficient to firm our belief in the fact that it is the data, not the algorithms, that play the most decisive role.
Though we did not face any difficulty in executing the project, we would like to express our gratitude for the publisher of the dataset, without which it would have been almost impossible for us to carry out this project.
To finally conclude, I want to state that I am satisfied with the results, and feel that the final model has fulfilled my expectations for an accurate and robust solution.

5.2 Improvement

Though we have got good results, there are still chances of improvement, as in a healthcare like this, we always aim to be 100% accurate. Possibly, we can improve by having larger dataset which captures more distinguishing patterns, and exploring other algorithms like Neural Networks, which we did not touch in this project.

Future study can focus on gathering new dataset that will present new insight and knowledge to enhance the prediction of diabetes using data mining technique. Also fine tuning techniques can be used to improve the performance of the models while means of handling imbalance class data can also be explored.

6.References

1.http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/diabetes.html
2.Alaa Elsayad, Mahmoud Fakhr, “Diagnosis ofCardiovascular Diseases with Bayesian Classifiers”,Journal of Computer Sciences 11 (2), pp. 274 – 282.
3. WHO,"Diabetes Factsheet," WHO Media Center, updatedNov. 2017. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs312/en/. [Accessed: Dec. 8, 2017].
4. J. Han, M. Kamber, J. Pei, Data Mining Concepts and Techniques 3rd ed, Morgan Kaufmann Publishers, USA, 2012.