Journal of Applied Science and Education, 2022,
Vol. 02, Iss. 02, S. No. 003, pp. 1-12
ISSN (Online): 2583-1372

Diabetes Prediction using Machine Learning

Sania Faraz[1], Pawan Singh[2]

[1,2]Amity School of Engineering and Technology Lucknow, Amity University Uttar Pradesh, India
saniyafaraz18@gmail.com[1], pawansingh51279@gmail.com[2]

How to cite this paper: S. Faraz and P. Singh, "Diabetes Prediction using Machine Learning," *Journal of Applied Science and Education (JASE)*, Vol. 02, Iss. 02, S. No. 003, pp 1-12, 2022.

https://doi.org/10.54060/jase.v2i2.13

Received: 29/09/2022
Accepted: 16/11/2022
Published: 25/11/2022

Abstract

Machine Learning is a type of AI (also known as Artificial Intelligence) that makes the pc or computer act like individuals and learn more as they experience additional information from their client or user. So here in this report we got basic introduction about machine learning like actually what is it, what is its use, how it works and many more things. Thereafter, we discussed about the python the language, which is used for making project, important libraries such as pandas and numPy which is being is used for this particular project and we have also discussed about Support Vector machine which has been used as classifier. We have also talked about the linear and non-linear svm that is used to check the accuracy of the predictive system. For the implementation we have started with importing the dependencies numpy smp and pandas pd and for the analysis we have taken a csv Pima Indian diabetes dataset after that we have trained the model with the help of support vector classifier. For the model evaluation we have checked the accuracy of the training data and test data. Numerous people suffer from diabetes mellitus, one of the most serious diseases. Age, obesity, inactivity, genetic diabetes, a poor diet, high blood pressure, and other factors can all contribute to diabetes mellitus. Diabetes increases a person's risk of developing various illnesses, including heart disease, renal disease, stroke, vision problems, nerve damage, etc. So, here we will be building a system that can predict whether a person has diabetes or not with the help of Machine Learning.

Keywords

Diabetes mellitus, Diabetes prediction, Machine Learning, Python, Support Vector Machine

1. Introduction

Diabetes is a chronic condition brought on by either insufficient insulin production by the pancreas or inefficient insulin utilization by the body. A hormone called insulin controls blood sugar. Uncontrolled diabetes frequently results in hyperglycemia, or elevated blood sugar, which over time causes substantial harm to many different bodily systems, including the neurons and blood vessels. High blood sugar due to diabetes can damage our Kidneys, Eyes, Nerves and other organs as well. These disorders had a significant negative impact on quality of life. One of the most severe diseases, diabetes is spread worldwide.

This chronic disorder is a significant factor in adult fatalities worldwide. The cost of chronic illnesses is also a factor. Diagnosis of diabetes is regarded as a difficult subject for quantitative research. Worldwide, 451 million adults received diabetes treatment in 2017. The number of people with diabetes is expected to reach about 693 million by 2045, with half of them going untreated. Thus, the high prevalence of diabetes around the world has a detrimental economic effect on people, healthcare systems, and countries. Diabetes prediction model in the healthcare industry is a very useful system. Machine learning techniques are frequently used to predict diabetes, and they produce better results. We will be building a system that can predict whether a person has diabetes or not with the help of Machine Learning. This project is done in Python. In this project, we will be using one of the most important machine learning algorithms, Support Vector Machine models for prediction.

The organization of the paper is as follows: Section 2 of this paper explains the aim and purpose of the project. Section 3 is a literature review which provides detailed information about the disease and the techniques that are being used. In Section 4 workflow, code implementation, results and outcomes are discussed. The conclusion of this study is discussed in Section 5.

2. Aim and Purpose

The goal of this study is to create a system that, by fusing the findings of several machine learning approaches, can more accurately conduct early diabetes prediction for a patient. The project work reveals that our model is capable of accurately predicting diabetes with an accuracy of 95% or higher. Our findings demonstrate that Random Forest outperformed other machine learning methods in terms of accuracy.

The following are some of the features:

- Being able to obtain an accurate report.
- Diabetes diagnosis at an early stage can improve more effective treatment.
- We utilize data mining techniques to predict disease at an early stage.
- Significant attributes are used to predict diabetes, and the relationships between the various attributes are also described.

3. Literature Review

3.1 Diabetes

Numerous people suffer from diabetes mellitus, one of the most serious diseases. Age, obesity, inactivity, genetic diabetes, a poor diet, high blood pressure, and other factors can all contribute to diabetes mellitus. The majority of the food you consume is converted by your body into sugar (glucose), which is then released into your bloodstream. Your pancreas releases insulin when your blood sugar levels rise. In order for blood sugar to enter your body's cells and be used as energy, insulin functions like a key. Diabetes increases a person's risk of developing various illnesses, including heart disease, renal disease, stroke, vision problems, nerve damage, etc. Feeling tired and weak, urinating often, and losing weight are the symptoms of diabetes.

So, for the prediction we shall learn about the python libraries, support vector machine as a classifier. For the implementation of the project firstly we need a diabetes dataset, then we need to pre-process the data after preprocessing the data we need to split the data into train and test data and to for checking the accuracy of the system will use classifier and we need to split the data from the dataset to check the accuracy score.

3.1. Introduction to Machine Learning

Machine Learning is a field of study devoted to comprehending and developing "learning" methods, or methods that use data to enhance performance on a certain set of tasks. It is considered to be one of the parts of AI (Artificial Intelligence). Without being expressly taught to do so, machine learning algorithms create a model using sample data, also known as training data, in order to make decisions or predictions. Speech recognition, email filtering, computer vision, and other fields where it is impossible or impractical to create traditional algorithms for the required tasks, among many others, use machine learning methods [1]. Machine learning algorithms are trained to generate classifications or predictions using statistical techniques, revealing important insights in data mining operations. The decisions made as a result of these insights influence key growth indicators in applications and enterprises, ideally. Data scientists will be more in demand as big data develops and grows because they will be needed to help identify the most important business issues. Computational statistics, which is focused on making predictions with computers, is closely related to a subset of machine learning, but not all machine learning is statistical learning.

Modern ML (Machine Learning) has two purposes; the first one is to classify the data using established models, and the second one is to forecast future results using these models [2].

a) Data Mining

While machine learning concentrates on prediction, based on known qualities learnt from the training data, and data mining concentrates on the finding of (previously) unknown properties in the data, both techniques frequently use the same methodologies and have significant overlap. On the other hand, machine learning also uses data mining approaches as a pre-processing step or as an “unsupervised learning” to increase learner accuracy. Data mining uses a variety of machine learning techniques, albeit with distinct purposes.

b) Optimization

Optimization and machine learning are closely related since many learning problems are phrased as the minimization of a loss function on a training set of samples. The gap between the model's predictions and the actual problem occurrences is expressed by loss functions Such as: classification and to assign labels to instances [3].

c) Generalization

The distinction between machine learning and optimization stems from the generalization objective: although optimization methods can reduce loss of training sets, machine learning is focused on reducing loss on untried samples. Research on characterizing the generalization of different learning methods is ongoing, especially for the algorithms of deep learning.

d) Statistics

Machine learning and statistics are closely connected areas in terms of methodologies. Machine learning seeks generalizable predictive patterns. Some statisticians have incorporated machine learning techniques, creating what they refer to as statistical learning.

3.2 Python

Python is Object oriented programming language which is simple to code when compared to other programming languages also it is interpreted language which means that its code is executed line by line. Python language may be utilized for various things including AI and ML, visualization of data, analytical data etc. [4]. Python has established itself as a language capable of solving any problem, regardless of its nature. Python is an OOP language, and it is interpreted and interactive. Modules, exceptions, dynamic typing, extremely high-level dynamic data types, and classes are all included. Python is a powerful programming language with a simple syntax. It has too many system calls and libraries, as well as several window systems, and it may be extended in C or C++ Python can also be utilized as an extension language for programs that require a configurable

user interface. Python is designed to be a language that is simple to read. Its formatting is visually clean and frequently replaces punctuation with English keywords. It does not employ curly brackets to separate blocks, in contrast to many other languages, and semicolons are permitted but infrequently used after statements. Compared to C or Pascal, it features fewer syntactic exceptions and special circumstances. Python has established itself as a standard in data science, allowing data analysts and other experts to utilize it to perform intricate statistical computations, design machine learning algorithms, handle and analyze data, among other activities [5].

3.3 Support Vector Machine

One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed as Support Vector Machine [6].

a) Hyperplane

In n-dimensional space, there may be several lines or decision boundaries used to separate the classes, but we must identify the optimum decision boundary that best aids in classifying the data points. hyperplane of the SVM is a name for this optimal boundary.

b) Support Vectors

Support vectors are the vectors that are closest to the hyperplane and have the greatest influence on where the hyperplane is located. By utilizing these support vectors, we increase the classifier's margin [7]. The hyperplane's location will vary if the support vectors are deleted. These are the ideas that aid in the development of our SVM.

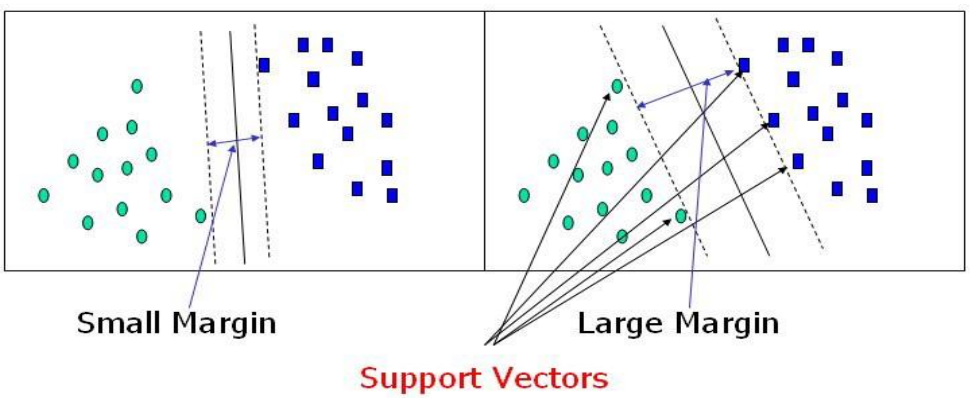

Figure 1. Support Vectors

c) Linear SV

Consider a dataset with two tags blue and green, two features (x1 and x2), and two tags. We need a classifier that can identify whether the pair of coordinates (x1, x2) is blue or green. Since it is a two-dimensional space, we may easily distinguish between these two classes by utilizing merely a straight line. But these classes may be divided by more than one line.

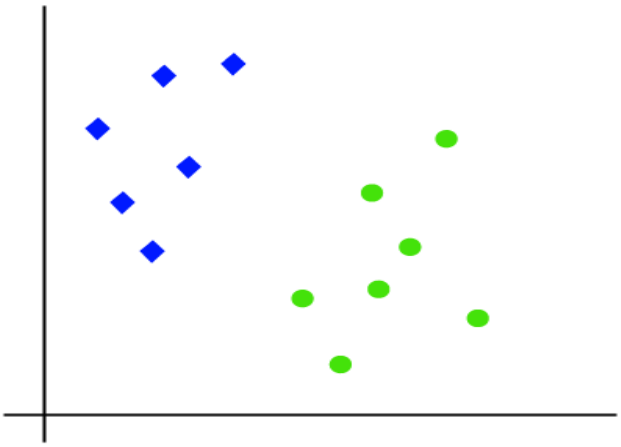

Figure 2. Linear SVM

d) Non-Linear SVM

We can use We can use a straight line to divide data that is organized linearly, but we are unable to do so with non-linear data. We must therefore add another dimension in order to separate these data values. We have used the two dimensions x and y for linear data, so we will add the third-dimension z for non-linear data [8].

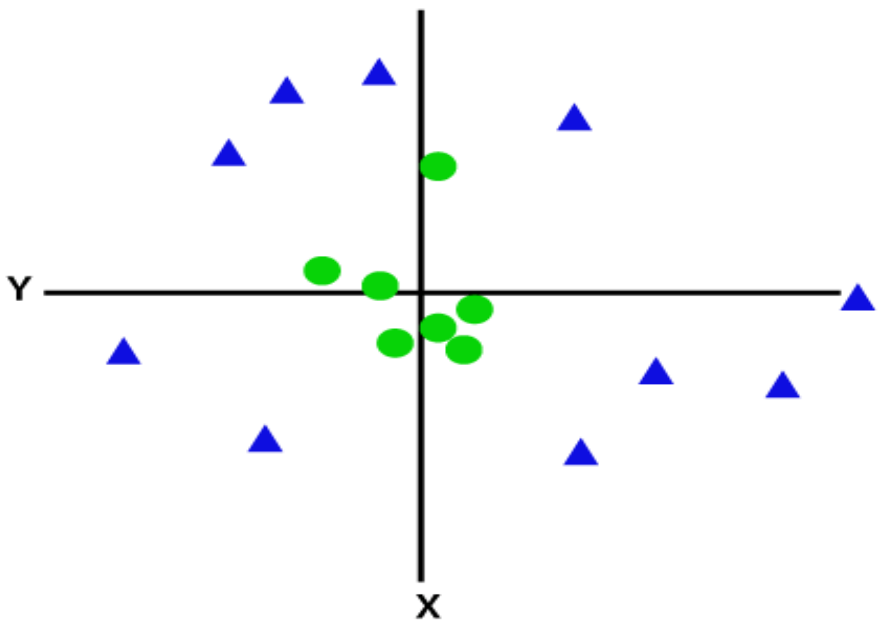

Figure 3. Non-Linear SVM

3.4 Python Implementation of SVM

We will now use Python to develop the SVM algorithm. The same user data dataset that we used for the KNN Classification and logistic Regression will be used here. Now that we are familiar with the fundamentals of SVM, let's try to use Python to build it. With Scikit Learns svm package, implementation is easy to understand and follows the same logic as the abovementioned understanding [9].

3.5 Libraries used

a) NumPy

Numerical Python is referred to as NumPy. The Python package NumPy is used to manipulate arrays. Additionally, it has matrices, Fourier transform, and functions for working in the area of linear algebra. The equivalent of arrays in Python are lists, although they take a long time to execute. The goal of NumPy is to offer array objects that are up to 50 times faster than conventional Python lists. The NumPy array object is referred to as ND array, and it has a number of supporting methods that make using ND array relatively simple.

b) Pandas

Open-source library designed primarily for working quickly and logically with relational or labelled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library. It is built on the top of NumPy library, many NumPy structures are utilized or duplicated in Pandas. Pandas generates data that is frequently used as input for SciPy's statistical analysis, SciPy's graphing routines, and Scikit-machine Learns learning algorithms. Any text editor can be used to run the Pandas program; however, Jupiter Notebook is preferred because it allows you to only run the code in a specific cell rather than the entire file [10].

4. Result and Discussion

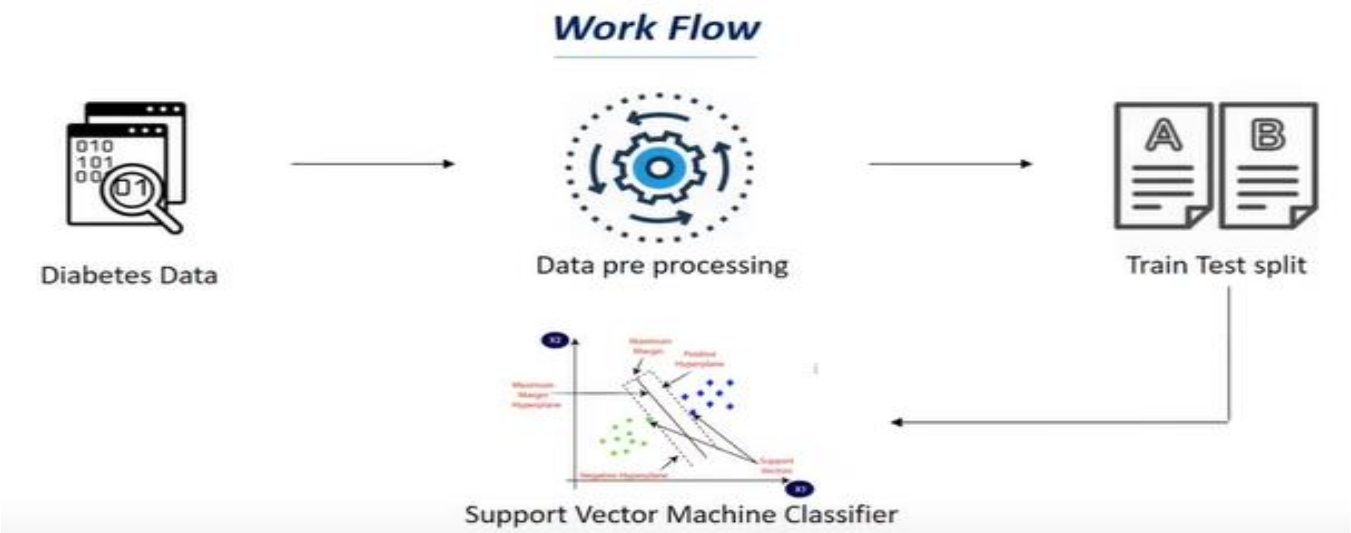

Figure 4. Workflow chart

4.1 Importing the Dependencies

Importing NumPy smp and pandas pd [11]. Now, we need to standardize the data, so we need a standardizer function from sklearn dot pre-processing inputs standard scalar. This standard data will be used to standardize the data to a common range. Now, we will be using train test split to change our data into training data and test data, will use svm which stands for support vector machine, and we need to test accuracy score from sklearn dot matrix to import accuracy score.

```
[ ] import numpy as np
    import pandas as pd
    from sklearn.preprocessing import StandardScaler
    from sklearn.model_selection import train_test_split
    from sklearn import svm
    from sklearn.metrics import accuracy_score
```

Figure 5 (a). Importing the independent Pandas and NumPy.

4.2 Data collection and Analysis

Tools for gathering, interpreting, and presenting data for a variety of applications and sectors are referred to as analysis and data collection tools. For usage in practically every industry, numerous programs and procedures have been developed, from manufacturing & quality control to research teams and data collection businesses [12].

4.3 PIMA Diabetes dataset

	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	0	64	0	23.3	0.672	32	1
3	1	89	94	66	23	94	0.167	21	0
4	0	137	35	40	35	168	2.288	33	1

```
[ ] # number of rows and Columns in this dataset
    diabetes_dataset.shape
```

Figure 5(b). showing the number of rows and cplumns in the dataset

(768, 9)

```
[ ] # loading the diabetes dataset to a pandas DataFrame
    diabetes_dataset = pd.read_csv('/content/diabetes.csv')

[ ] pd.read_csv?

    # printing the first 5 rows of the dataset
    diabetes_dataset.head()
```

Figure 6. loading the diabetes dataset to a pandas Data frame

Diabetes_dataset [‘outcome’]. Value_counts()

Separating the data and labels

X = diabetes_dataset.drop(columns = 'Outcome', axis=1)

Y = diabetes_dataset['Outcome']

Name: Outcome, dtype: int64

0 --> Non-Diabetic

1 --> Diabetic

separating the data and labels

X = diabetes_dataset.drop(columns = 'Outcome', axis=1)

Y = diabetes_dataset['Outcome']

4.4 Data Standardization

Data normalization is the procedure we use to scale all of the data to the same level. This will assist us in data analysis and model feed-back. This is the mathematical foundation for the data standardization procedure. A dataset must be scaled so that the standard deviation is 1 and the mean value is 0, which is known as standardization.

Standardization is helpful when your data have different scales and the algorithm you're employing, like logistic regression, linear regression or linear discriminant analysis, does assume that your data have a Gaussian distribution. It is also known as Python data scaling.

```
[ ] scaler = StandardScaler()

[ ] scaler.fit(X)

    StandardScaler()

[ ] standardized_data = scaler.transform(X)

    print(standardized_data)
```

Figure 7. Standardizing the data with the help of Standard Scaler Function

4.5 Train Test split

Sort matrices or arrays into test subsets and train at random. Quick tool that encapsulates next(ShuffleSplit) and input validation () [13]. Splitting (and possibly subsampling) data with a oneliner requires only the calls split (X, y) and input applcation data.

```
[ ]  X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)

[ ]  print(X.shape, X_train.shape, X_test.shape)
```

Figure 8. splitting the data into train test split data.

4.6 Training the model

For training our model, we will create a variable called as classifier which is support vector classifier (svc). Then we need to represent another parameter which is kernel and now we will fit the training data to the classifier [14].

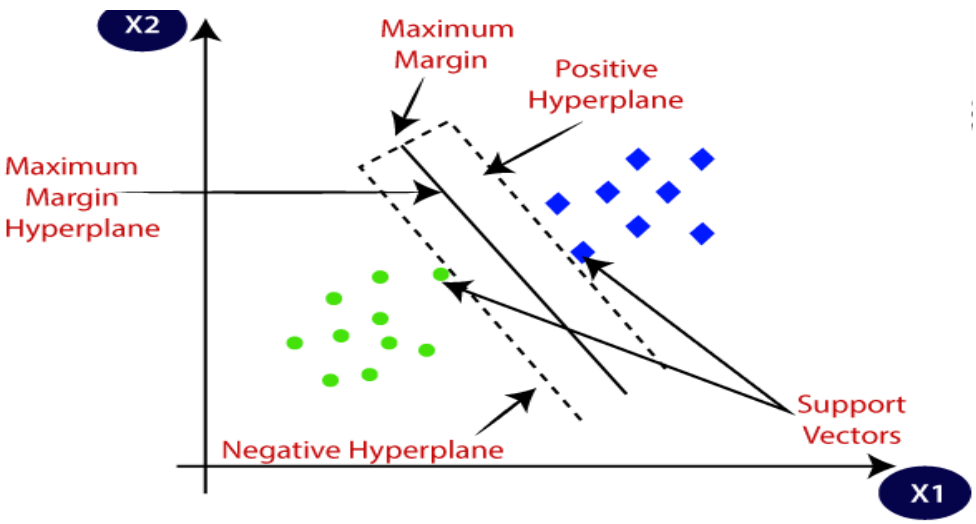

Figure 9. two classified different categories

```
[ ]  classifier = svm.SVC(kernel='linear')

[ ]  #training the support vector Machine Classifier
     classifier.fit(X_train, Y_train)

     SVC(kernel='linear')
```

Figure 10. Training the model

4.7 Model evaluation

Accuracy score

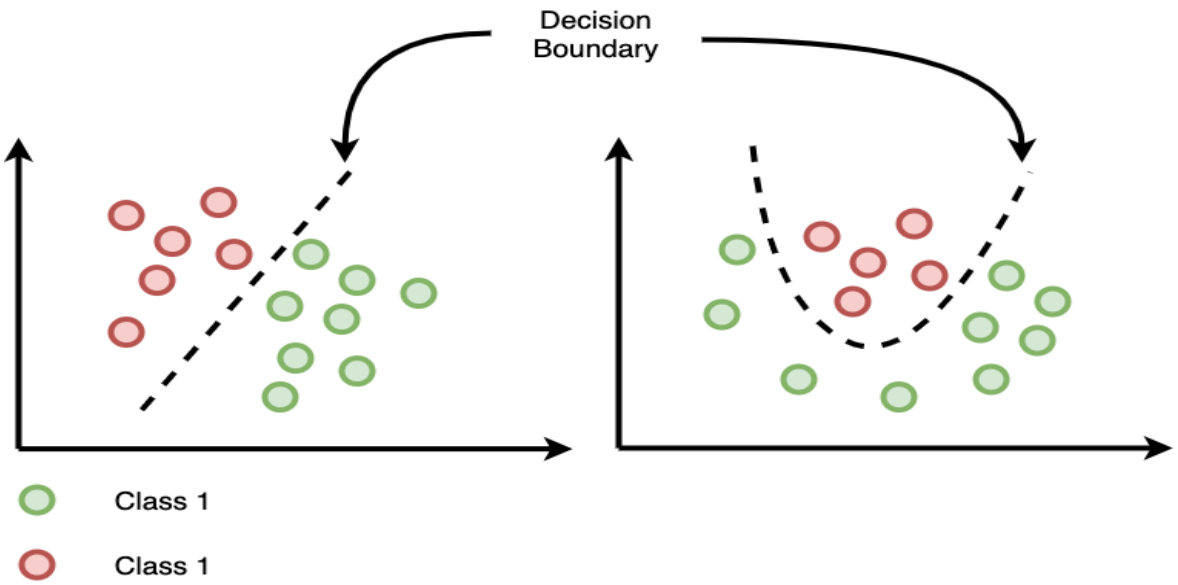

Figure 11. showing the decision boundary which can be linear and non-linear both

```
[ ] # accuracy score on the training data
    X_train_prediction = classifier.predict(X_train)
    training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy score of the training data : ', training_data_accuracy)
```

Figure 12. checking the accuracy score of the training data

```
[ ] # accuracy score on the test data
    X_test_prediction = classifier.predict(X_test)
    test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[ ] print('Accuracy score of the test data : ', test_data_accuracy)
```

Figure 13. checking the accuracy score of the testing data.

```
[ ]  input_data = (5,166,72,19,175,25.8,0.587,51)

     # changing the input_data to numpy array
     input_data_as_numpy_array = np.asarray(input_data)

     # reshape the array as we are predicting for one instance
     input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

     # standardize the input data
     std_data = scaler.transform(input_data_reshaped)
     print(std_data)

     prediction = classifier.predict(std_data)
     print(prediction)

     if (prediction[0] == 0):
       print('The person is not diabetic')
     else:
       print('The person is diabetic')
```

Figure 14. Making a Predictive system

```
[[ 0.3429808    1.41167241   0.14964075 -0.09637905   0.82661621 -0.78595734
  0.34768723   1.51108316]]
The person is diabetic.
```

5. Conclusion

Studied about the support vector machine algorithm. So, in supervised machine learning model, we feed the data to our machine learning model and machine learning model learns from the data and its respective label. We need to pre-process the data where we will try to analyze the data and we need to standardize this data so that all this data lies in the same range so, once we pre-process the data, we will split the data into training and testing the data so will train machine learning model with training data and then we try to find the accuracy score of our model with the help of test and it will tell us that how well our model is performing. for the analysis we have taken a csv Pima Indian diabetes dataset. For the implementation we have imported all the dependencies. I have imported NumPy for making empire respond us for creating the data frame and we have a standard scalar function to stabilize the data and then will add train split to split the data into training and test data and we have to import the support vector machine model from scalar and then we have to have trained the model with the help of support vector classifier. For the model evaluation we have checked the accuracy of the training data and test data. Import this accuracy score for evaluating the model. Now, will provide the random values and the system can predict whether a person has diabetes or not. With this study we acquired around 79% accuracy.

References

[1]. I. D. Apostolopoulos, N. I. Papandrianos, E. I. Papageorgiou, and D. J. Apostolopoulos, “Artificial Intelligence methods for identifying and localizing abnormal Parathyroid Glands: A review study,” Mach. Learn. Knowl. Extr., vol. 4, no. 4, pp. 814–826, 2022.

[2]. X. Zhou, I. W. Tsang, and J. Yin, “LADDER: Latent boundary-guided adversarial training,” Mach. Learn., 2022.

[3]. T. Mercieca, J. G. Vella, and K. Vella, “Initial optimization techniques for the Cube Algebra Query Language: The relational model as a target,” Int. J. Data Warehous. Min., vol. 18, no. 1, pp. 1–17, 2022.

[4]. A. Thomas et al., “A Study to assess Psychosocial Parental Stress during COVID-19 Pandemic in a selected rural community at Kottayam district,” Int. J. Nurs. Educ. Res., pp. 211–215, 2022.

[5]. T. P. Lestari, “Analisis Text Mining pada Sosial Media Twitter Menggunakan Metode Support Vector Machine (SVM) dan Social Network Analysis (SNA),” Jurnal Informatika Ekonomi Bisnis, pp. 65–71, 2022.

[6]. K. Mostafaei, S. Maleki, and B. Jodeiri, “A new gold grade estimation approach by using support vector machine (SVM) and back propagation neural network (BPNN)- A Case study: Dalli deposit, Iran,” Research Square, 2022.

[7]. T. Ebina, T. Arai, H. Toh, and Y. Kuroda, “1P472 Domain linker prediction using a Support Vector Machine(SVM)(23. Bioinformatics, genomics and proteomics (I),Poster Session,Abstract,Meeting Program of EABS &BSJ 2006),” Seibutsu Butsuri, vol. 46, no. supplement2, p. S264, 2006.

[8]. S. Lee et al., “Comparison of west Nile virus and yellow fever virus using apriori algorithm, decision tree, and support vector machine(SVM),” Int. J. Mach. Learn. Comput., vol. 6, no. 2, pp. 155–159, 2016.

[9]. J. Dr. Menyhárt and J. H. Gomes Da Costa Cavalcanti, “LSI with Support Vector Machine for Text Categorization – a practical example with Python,” Int. J. Eng. Manag. Sci., vol. 6, no. 3, 2021.

[10]. S. Zhou and W. Zhou, “Unified SVM algorithm based on LS-DC loss,” Mach. Learn., 2021..

[11]. A. Astorino, A. Fuduli, G. Giallombardo, and G. Miglionico, “SVM-based multiple instance classification via DC optimization,” Algorithms, vol. 12, no. 12, p. 249, 2019.

[12]. S. M. Ganie, M. B. Malik, and T. Arif, “Performance analysis and prediction of type 2 diabetes mellitus based on lifestyle data using machine learning approaches,” J. Diabetes Metab. Disord., vol. 21, no. 1, pp. 339–352, 2022.

[13]. S. Mishra and Research Scholar, Department of Electronics and Instrumentation Engineering, Odisha University of Technology and Research, Bhubaneswar (Odisha), India., “A comparative analysis of diabetes prediction using different machine learning algorithms,” Indian Journal of Artificial Intelligence and Neural Networking, vol. 2, no. 5, pp. 1–7, 2022.

[14]. J. N. V. R. S. Kumar, K. H. Kumar, A. Haleem, B. Sivaranjani, B. N. T. Kiran, and S. Prameela, “IBM auto AI bot: Diabetes mellitus prediction using machine learning algorithms,” in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022.

[15]. A. Viloria, Y. Herazo-Beltran, D. Cabrera, and O. B. Pineda, “Diabetes diagnostic prediction using vector support machines,” Procedia Comput. Sci., vol. 170, pp. 376–381, 2020.