

# - Project 1: Diabetes prediction -

Author	Daniel Monti & Daniel Bürgler & Camille Serquet
Date	December 11, 2021
Course	D101722-HS2021-0: Technology and Diabetes Management
Professor	Stavroula Mougiakakou, PhD
Assistant	Matthias Fontanellaz, MSc, PhD Fellow

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>1</b>
<b>3</b>	<b>Methods</b>	<b>1</b>
3.1	Programming Language and used packages . . . . .	1
3.2	Decision Tree . . . . .	2
3.3	K-Nearest Neighbours . . . . .	2
3.4	Logistic Regression . . . . .	2
3.5	Naïve Bayes . . . . .	2
3.6	Support Vector Machines . . . . .	3
3.7	Stochastic Gradient Descent . . . . .	4
3.8	Cross Validation . . . . .	4
3.9	Model Comparisons . . . . .	4
<b>4</b>	<b>Data and Experiment setup</b>	<b>5</b>
4.1	Dataset . . . . .	5
4.2	Data preparation . . . . .	6
4.2.1	Dataset Splitting . . . . .	7
4.2.2	Data Scaling / Feature Scaling . . . . .	7
4.2.3	Imbalanced Data . . . . .	8
<b>5</b>	<b>Results</b>	<b>8</b>
<b>6</b>	<b>Discussion</b>	<b>9</b>

# 1 Introduction

Diabetes is a chronic metabolic disease associated with hyperglycemia. The chronic hyperglycemia results from defects in insulin secretion, insulin action, or both. Over time, diabetes leads to serious damage to the heart, eyes, kidney, nerves, and heart vessels. The prediction at an early stage of such a disease leads to improved treatment and mitigate the risk of long-term complications. Prediction Analysis uses statistics to create models that examine current and past data and predict potential future outcomes. When applied in health care, Prediction Analysis can help determine the patient’s risk of developing certain disease such as diabetes. It is therefore used as a tool to support medical decision-making and eliminate human efforts by supporting automation with minimum flaws [reference]. In current medical diagnosis method, three types of errors are still to be deplored: false-negative, false-positive, and unclassified type.

Classification algorithms need to be evaluated with evaluation metrics. The resulting accuracy gives a good appreciation of the algorithm and helps to detect which one suits best for a medical application. In this paper, the Pima Indian Diabetes Database [5] is prepared and applied to classification algorithms such as Logistic Regression, Naïve Bayes, Stochastic Gradient Descent, K-Nearest Neighbors, Decision Tree, and Support Vector Machine. All algorithms will be reviewed. Finally, a comparison between all algorithms’ evaluation metrics will be evaluated.

# 2 Related work

[2] DB

# 3 Methods

filler text here

## 3.1 Programming Language and used packages

This project is coded with Python version 3.9.9. the used packages can be seen in table 1.

Python Package	Version	Description
NumPy [3]	1.21.4	NumPy is the fundamental package for array computing with Python.
pandas [4]	1.3.4	pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool
matplotlib [1]	3.5.0	Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python
seaborn [7]	0.11.2	Seaborn is a library for making statistical graphics in Python
scikit-learn [6]	1.0.1	scikit-learn is a Python module for machine learning built on top of SciPy

Table 1: Used Python libraries

The source code for this project is hosted on GitHub and is openly available on the following webpage: <https://github.com/d-monti/diabetesPrediction>.

### 3.2 Decision Tree

Decision Tree is a tree-like model that contains conditional control statements. Each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf represents the outcome (a final decision). The splitting criterion at a decision node is generated by Attribute Selection Measure (ASM).

Decision Tree algorithm has multiple advantages as it does not require the normalization of data nor the scaling of the data, and the missing values do not affect the building process of the decision tree. Some drawbacks are the long time needed for training the model and the instability caused by a small change in the data.

The chosen model is the DecisionTreeClassifier. All default parameters are kept as so. The generation of the evaluation metrics gives an accuracy of [put result].

Best Parameters are: criterion: gini, splitter: best

### 3.3 K-Nearest Neighbours

**Standardization needed!** Supervised neighbors-based learning comes in two flavors: classification for data with discrete labels, and regression for data with continuous labels. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The k-factor is the number of samples/neighbours. It can be defined by the user. Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular. The optimal choice for the k-factor depends on the data. In general a larger k suppresses the effects of noise, but makes the classification boundaries less distinct. There are different methods to find the nearest neighbor. The algorithms used are brute force, K-D Tree or Ball Tree. The optimal choice of algorithm depends on multiple factors: number of samples and dimensionality D, Data Structure, number of neighbours requested and number of query points

### 3.4 Logistic Regression

I honestly don't understand shit. the l1, l2 is confusing

Logistic regression is a linear model used for classification. The probabilities describing the possible outcomes of a single trial are modeled by a logistic function. This implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional l1 and l2 or Elastic-Net regularization. The regularization l1 and l2 both solve optimization problems. Whereas the Elastic-Net is a combination of both. An advantage of regularization is that it improves numerical stability. No regularization amounts to setting C to a very high value

### 3.5 Naïve Bayes

Naive Bayes (NB) is a probabilistic classification technique based on Bayes' theorem with the assumption of independence between all features in a class. Bayes' theorem gives the probability that an event happens with the knowledge of another event with the condition that they are not interdependent. In a Gaussian Naive Bayes classifier, each values are continuous and assumed to be normally distributed.

NB is a good classifier when it comes to multi-class prediction dataset and when the independence between all classes remains true. One big advantage of NB is that it requires a small amount of training data to estimate the test data which leads to a short training period. A drawback of this implementation is the assumption of independent predictors which, in real life, is almost impossible to achieve.

The chosen model is the GaussianNB(). Both parameters *priors* and *var\_smoothing* are kept as default. The generation of the evaluation metrics gives an accuracy of [put result].

### 3.6 Support Vector Machines

Support vector machines (SVMs) are supervised learning methods used for classification, regression, and outliers detection.

An advantage of SVMs is that it is effective in high dimensional spaces, even in cases where the number of dimensions is greater than the number of samples. SVMs are also memory efficient because they use a subset of training points in the decision function (called support vectors). Another advantage is that SVMs is versatile in which different Kernel functions can be specified for the decision function. One disadvantage of SVMs is that if the number of features is much greater than the number of samples, the choice of Kernel functions and the regularization term become crucial to avoid overfitting. Another drawback is that SVMs do not directly provide probability estimates; these are calculated using time expensive k-fold cross-validation.

In the SVMs model we can chose between the following four kernel functions, the figure 1 illustrates the differences between the kernels.

Linear	$\langle x, x' \rangle$
Rbf	$\exp \left( -\gamma \ x - x'\ ^2 \right)$ , where $\gamma$ is specified by parameter gamma
Polynomial	$(\gamma \langle x, x' \rangle + r)^d$ , where $d$ is specified by parameter degree, $r$ by coefo
Sigmoid	$\tanh (\gamma \langle x, x' \rangle + r)$ , where $r$ is specified by coefo

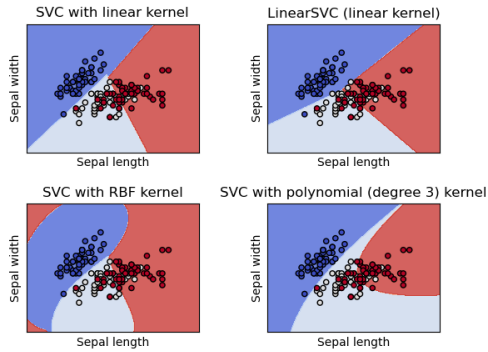


Table 2: Support Vector Machines Kernels

Figure 1: Support Vector Machines Kernels [6]

For this model the test and train data will be tested scaled and non scaled, this means training an SVMs over the scaled and non-scaled data leads to the generation of different models. To tune the SVMs model the following parameters were tested with a grid search and cross validation with 5 time folding. The underlined parameters are the best-performing ones, and we used them in this project. The scaled test dataset was used, because it scored a higher accuracy. The best parameters can vary for every run of the source code.

Kernel	C	gamma	class_weight	coef0	degree
Linear	0.9, 1, 2, 10		balanced		
<u>Rbf</u>	0.9, <u>1</u> , 2, 10	<u>scale</u> , auto	<u>balanced</u>		
Polynomial	0.9, 1, 2, 10	scale	balanced	0.0	2, 3, 4, 5, 6, 7, 8
Sigmoid	0.9, 1, 2, 10	scale, auto	balanced		

Table 3: Support Vector Machine grid search used test parameters

### 3.7 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fit linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

The advantages of Stochastic Gradient Descent are its efficiency and the ease of the implementation with a lots of opportunities for code tuning. One drawback of SGD is that it requires a number of hyperparameters such as the regularization parameter and the number of iterations. SGD is also sensitive to feature scaling, and therefore this model was tested with the scaled and non scaled dataset. This model has a lot of parameter to test, following a list of all tested parameters. The underlined parameters are the best-performing ones, and we used them in this project. The scaled test dataset was used, because it scored a higher accuracy. The best parameters can vary for every run of the source code.

Parameters	Values
loss	hinge, log, <u>modified_huber</u> , <u>squared_hinge</u> , perceptron, <u>squared_error</u> , <u>huber</u> , <u>epsilon_insensitive</u> , <u>squared_epsilon_insensitive</u>
penalty	l1, <u>l2</u> , elasticnet
class_weight	<u>balanced</u>
l1_ratio	0 – 1 in 10 steps (best: <u>0.0</u> )
alpha	0.0001 – 0.1 (best: <u>0.01</u> )

Table 4: Stochastic Gradient Descent grid search used test parameters

### 3.8 Cross Validation

Alli algorithm sind mit cross validation getestet worda, k-fold = 5 -< meh got z lang und brint nid viel  
**Burd TODO** Grid search for SVM, KNN and random search for the rest.

### 3.9 Model Comparisons

To compare the models against each other, the best parameters are automatically chosen, and with them, another cross-validation with 5 times folding is done. The accuracy of all models is then used to create a boxplot, which allows us to compare the model against each other quickly. The accuracy only demonstrates to us a small part of the performance of a model. Therefore also the Specificity, Sensitivity and the F1 value are evaluated. Following the equations to get these values.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$F_1 = \frac{\text{True Positive}}{\text{True Positive} + \frac{1}{2}(\text{False Positive} + \text{False Negative})}$$

## 4 Data and Experiment setup

### 4.1 Dataset

The Pima Indians Diabetes Database [5] is used to build a model that accurately predict whether or not the patients in the dataset have diabetes or not. The datasets contains several medical predictor variables and one target variable; the output. All patients registered in this dataset are females, take part of the Pima Indian heritage, and are at least 21 years. The dataset contains 768 patients.

#### Predictor variables and target variable of the dataset

- **Pregnancies** : number of times pregnant
- **Glucose** : plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **BloodPressure** : diastolic blood pressure (mmHg)
- **SkinThickness** : triceps skin fold thickness (mm)
- **Insulin** : 2-Hour serum insulin (muU/ml)
- **BMI** : Body mass index (weight in kg/(height in m)<sup>2</sup>)
- **DiabetesPedigree** : diabetes pedigree function (A function that scores the likelihood of diabetes based on family history)
- **Age** : age (years)
- **Outcome** : class variables(0 or 1)

Multiple statistical tools allows a clear overview of the dataset. As some algorithm require the independancy of all variabls, the coorelation between them are calculated and shown in figure 2. The figure 3 show the histogram of the dataset.

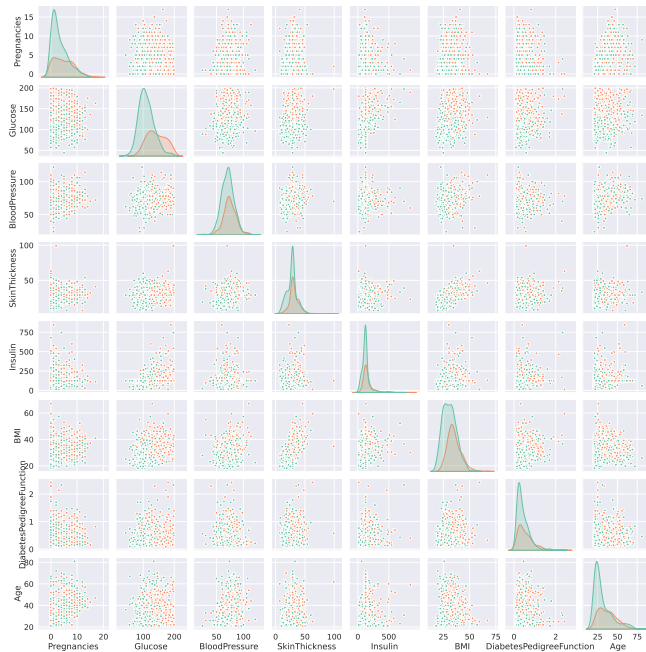


Figure 2: Correlation of the Dataset

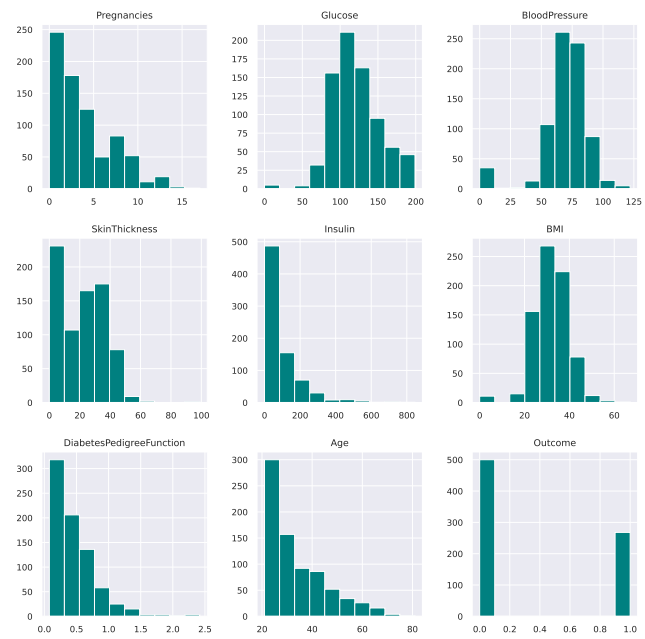
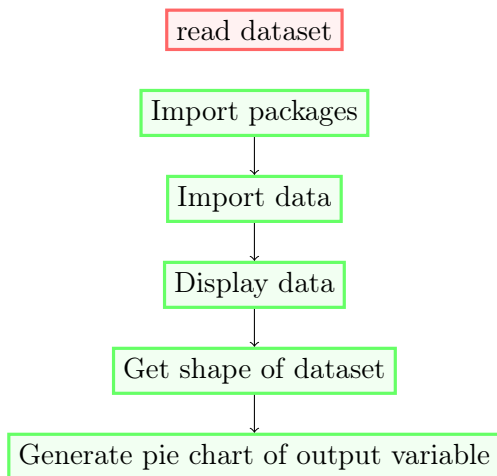


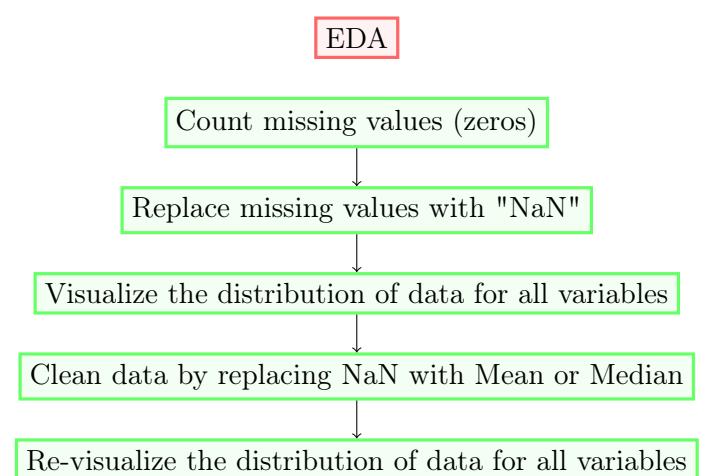
Figure 3: Histogram of the Dataset

## 4.2 Data preparation

The reading of the dataset is realised with multiples steps as shown in figure [4a]. The raw data needs to be prepared in order to be applicable for all classification algorithm used in this project. All missing values, denoted with the value "0" needs to be replaced and the imbalanced of the dataset needs to be checked. To realized these steps, an Exploratory Data Analysis (EDA), shown in figure [4b] needs to be followed.



(a) Reading dataset.



(b) Exploratory Data Analysis (EDA)

Figure 4: Dataset import and preparation

First we have a look at the dataset to have an overview, shown in figure 3. The summary of all variables is generated and gives useful information such as the number of entries, the mean, the standard

deviation, the minimum value, the quartiles, and the maximum value. shown in figure 5.

The output variable is shown in figure 6. This variable is assumed to be exact in the sense that there is no false positive nor false negative. From this statement, the goal is to reach the exact same output using all the predictors given in the dataset.

By looking at the data in table 5, it is easy to see that some "0" values do not make sense. For example, a value of zero in the BMI or in the Blood pressure variable would indicate that the patient is not existing or dead. The replacement of the zero values is made by first analyzing the distribution of the data. If the distribution is symmetric, the missing values are replaced with the mean value. If the distribution shows a lot of outliers, the missing values are replaced by the median value. The distribution and box plots are generated (see figure 5) to permit the decision. Once the missing values are replaced, those figures are re-generated to assure a similar distribution as before the cleaning. Table 5 shows the amount of the "0" values and how they were handled.

Data column	Amount of "0" values	Chosen replacement method
Pregnancies	0	None
Glucose	5	Mean
Blood Pressure	35	Mean
Skin Thickness	227	Median
Insulin	374	Median
BMI (Body Mass Index)	11	Median
Diabetes Pedigree Function	0	None
Age	0	None
Outcome	0	None

Table 5: Amount of zero values in dataset

#### 4.2.1 Dataset Splitting

The dataset needs to be split into a test dataset and a training dataset to train and test the machine learning models. In our case, the test dataset is 25% of the entire dataset. We use the training dataset to fit the machine learning model and the test dataset to evaluate the machine learning model. When we split a dataset, there are two competing concerns. Our parameter estimates have more significant variance if we have less training data, and our performance statistics will have more significant variance if we have less testing data. Therefore the 25% is an excellent choice and the default in scikit-learn library.

Both sets will be further split into an x-set and a y-set, where the y-set contains the outcome, if a person has diabetes or not (True or False), the x set contains all measured patient data. If the model is fitted with the training set, the test set will evaluate the model performance.

#### 4.2.2 Data Scaling / Feature Scaling

Data scaling or feature scaling is used in some machine learning algorithms. If there is a vast difference between numbers sizes, the model could assume that bigger numbers are more significant than smaller ones. Therefore all values are normalized or standardized.



In our case, data scaling was used for the SVMs and SGD **more? dani?** algorithms, in both cases, the scaled data had better accuracy than the non-scaled data. Data scaling also helps some models (for example, SGD) with the training time, significantly improving the converge time.

### 4.2.3 Imbalanced Data

The Pima Indians Diabetes Database [5] is quite imbalanced. This can be seen in figure 6, the outcome has way more non-diabetes patients (diabetes: 34.9%, No diabetes: 65.1%) than patients with diabetes. Some algorithms have problems with imbalanced data, for example, the SVMs algorithm. Suppose the imbalanced data are not handled in the SVMs algorithm. It will predict every patient as a non-diabetes patient and, therefore, have a pretty good accuracy because 65.1% of the patients are non-diabetic. This gives us a specificity of 100% and a sensitivity of 0%. Therefore, the model is useless. The imbalanced data need to be checked and handled if the algorithm requires it.

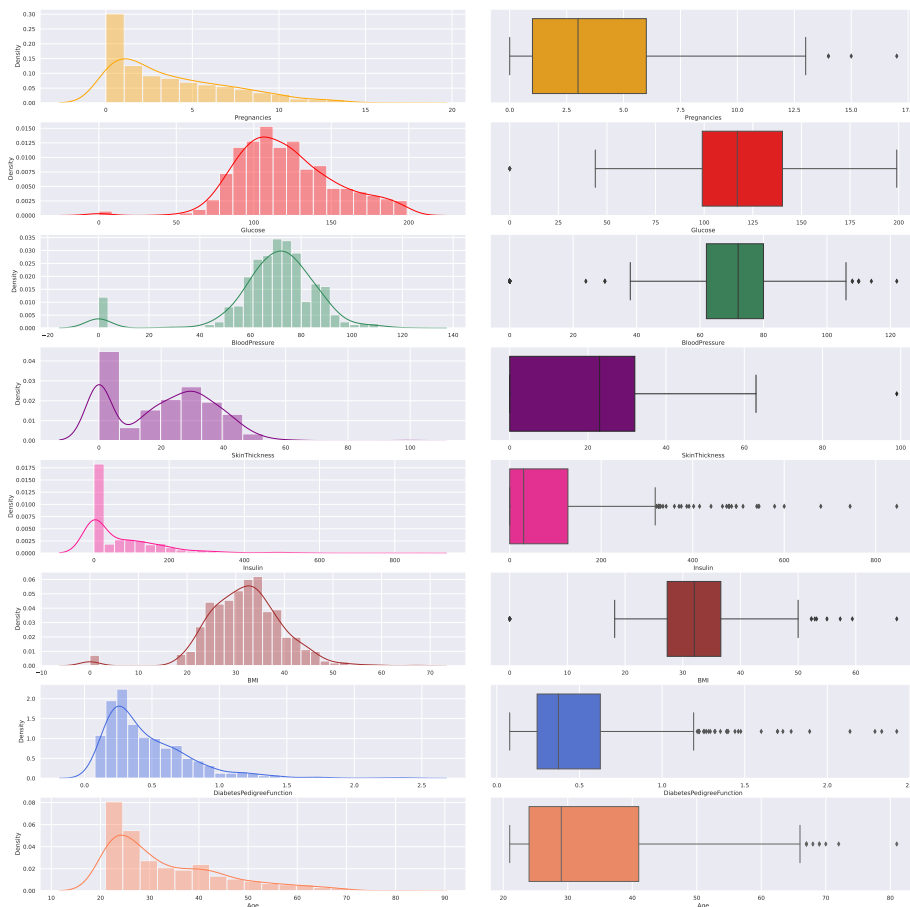


Figure 5: Dataset distribution and boxplots

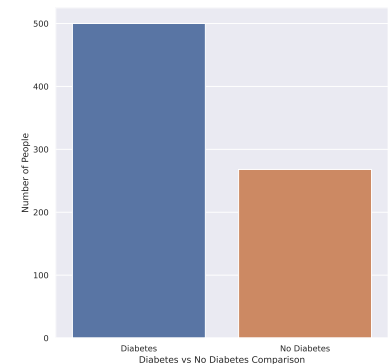


Figure 6: Dataset Outcome

## 5 Results

**TODO real output**

Model	Accuracy	Specificity	Sensitivity	F1
Stochastic Gradient Descent	0.670996	0.931507	0.223529	0.333333
Support Vector Machines	0.774892	0.897260	0.564706	0.648649
Naive Bayes	0.770563	0.842466	0.647059	0.674847
Decision Trees	0.727273	0.815068	0.576471	0.608696
Logistic Regression	0.779221	0.897260	0.576471	0.657718
K Nearest Neighbors	0.779221	0.863014	0.635294	0.679245

Table 6: Model comparison results

todo text

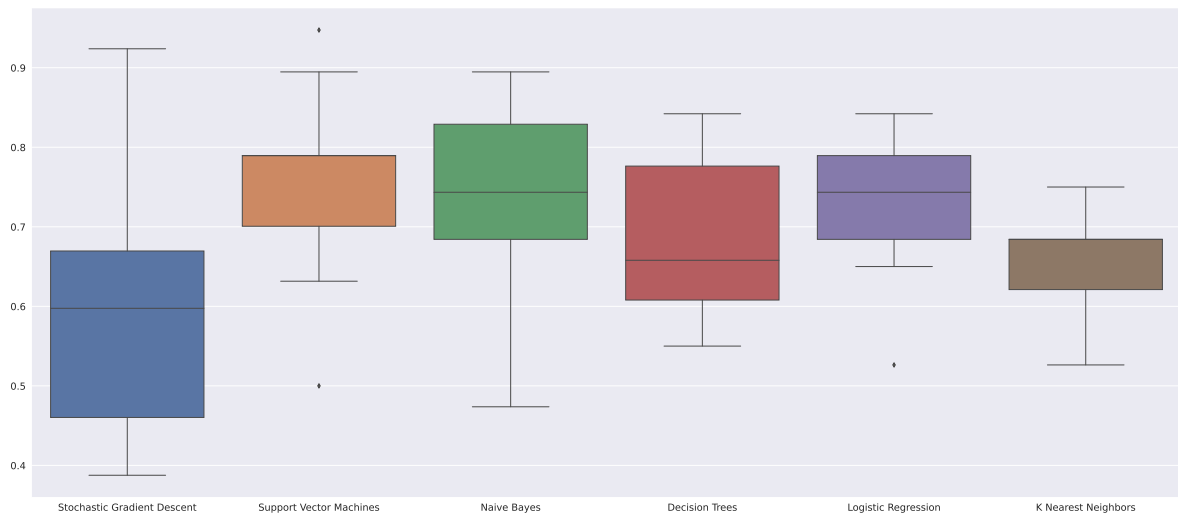


Figure 7: Model Comparison

6 Discussion

Diabetes is a well-known chronic metabolic disease that results from defects in insulin secretion, insulin action, or both. The long-term complications related to diabetes could be critical. To avoid dramatic situations, the diagnosis of diabetes should be made as early as possible. To this day, Prediction Analysis has been used to assist medical decision-making. What differs between each algorithm used to predict disease such as diabetes is the accuracy found in the evaluation metrics. By evaluating the accuracy of multiple algorithms, one can find the one that suits the most in a medical application such as diabetes diagnosis. In order to do this comparison, the Pima Indian Diabetes Database is prepared and applied to classification algorithms such as Logistic Regression, Naïve Bayes, Stochastic Gradient Descent, K-Nearest Neighbors, Decision Tree, and Support Vector Machine. The result shows that the best accuracy is reached with [??????] with an accuracy of [?????]. The result is explained by a good ratio of number of correct predictions to the total number of input samples. The database includes only women and is therefore limited to female diagnosis only. Future work could include a study of male-diagnosed patients to develop a more inclusive diagnosis method.

TODO -> a ccuracy of arounf 77% is really bad not? its a bit better than guessing, shall we write something about that? WHO criteria of  $\geq 80\%$  sensitivity and  $\geq 97\%$  specificity -> example requirments who coivd tests

## References

- [1] Matplotlib: Visualization with Python. <https://matplotlib.org/>. Accessed: 09.12.2021.
- [2] Aishwarya Mujumdar and V Vaidehi. “Diabetes Prediction using Machine Learning Algorithms”. In: *Procedia Computer Science* 165 (2019), pp. 292–299. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2020.01.047>.
- [3] Numpy: The fundamental package for scientific computing with Python. <https://numpy.org/>. Accessed: 09.12.2021.
- [4] Panda: open source data analysis and manipulation tool. <https://pandas.pydata.org/>. Accessed: 09.12.2021.
- [5] Pima Indians Diabetes Database. <https://www.kaggle.com/uciml/pima-indians-diabetes-database>. Accessed: 09.12.2021.
- [6] scikit-learn: Machine Learning in Python. <https://scikit-learn.org/>. Accessed: 09.12.2021.
- [7] Seaborn: Statistical data visualization. <https://seaborn.pydata.org/>. Accessed: 09.12.2021.