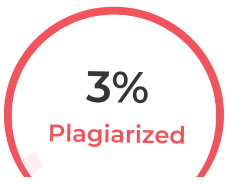
Dec 08, 2022

Plagiarism Scan Report



Characters:3791	Words:575
Sentences:30	Speak Time: 5 Min

Excluded URL	None

Content Checked for Plagiarism

Step 6: Use a test set to evaluate the Classifier model for the aforementioned machine learning algorithm. Comparative evaluation of the experimental performance outcomes for each classifier is step seven. Step 8: Select the highest performing algorithm after analysis based on various metrics. Data pre-processing The most crucial phase is data pre-processing. Most data pertaining to healthcare has missing values and other contaminants that can affect the data's usefulness. Data pre-processing is done to increase the quality and efficacy of the results from the mining process. This procedure is crucial for accurate results and good prediction when applying machine learning techniques to the dataset. the results information from the initial diabetes. To train the data only on the specified parameters, csv was separated. [6] Fig 4.1 Heatmap after pre-processing data Training and Testing Training Before training the data we need to split the data into train dataset and test dataset which was done using the following line of code X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2) After splitting the data, now we have to train the model for train dataset so that the model with the help of training can predict diabetes result for various other data’s. The model was trained using support vector machine algorithm. The final training of the model was done using fitting the data to our model. classifier = svm.SVC(kernel='linear') classifier.fit(X_train, Y_train) Testing so as of now the model has been trained and now its time to test the model on trained dataset as well as test dataset and to compare the accuracies of both the result t avoid any kind of overfitiing problem. X_train_prediction = classifier.predict(X_train) training_data_accuracy = accuracy_score(X_train_prediction, Y_train) X_test_prediction = classifier.predict(X_test) test_data_accuracy = accuracy_score(X_test_prediction, Y_test) Building the model Now as our model is trained so we must write a code snippet which will evaluate whether the person has diabetes or not. V RESULTS AND DISCUSSION The accuracy on training data was 78% and the accuracy on testing data was 77% as we can see the difference between accuracies of training and testing data is very less so we can say that this is not the case of overfitting and our model is good to go to find the result. The accuracy which was achieved on sample data set was nearly 78%. We have used SVM algorithm to predict whether a person is diabetic or not. SVM is a supervised machine learning algorithm used for both classification and regression. Our main goal here is to classify from given

dataset whether a person is not. VI CONCLUSION Designing and implementing a Diabetes Prediction Using Machine Learning Method was the primary goal of this project (SVM). Data validation accuracy was 78%. For the primary detection of diabetes, the accuracy is adequate. Now that the model has been extracted, it may be used in a user interface. A person may use it primarily depending on their needs. VII REFERENCES [1] fig 3.1 javatpoint [2] K.vijaykumar, B.lavanya, I.Nirmala, S.sofia caroline, “random forest algorithm for prediction of diabetes” [3] Jingyu xue, Fancho min, Fengying ma, “research on diabetes prediction based on machine learning” [4] Aishwarya Mujumdar, Dr. vaidehi V, “diabetes prediction using machine learning algorithm” [5] Mitushi Soni, Dr. sunita verma “diabetes prediction using various machine learning techniques” [6] Nahla B, Andrew “intelligible support vector machines for diagnosis of diabetes mellitus” [7] Dheeraj sheety “diabetes detection” [8] Tejas joshi, Pramila chawan, “diabetes prediction using ML” [9] Kaggle (diabetes dataset)

Sources

3% Plagiarized

Diagnose-That/urinary_inflammation.py at main · shrutityagi4102 ...Diabetes Prediction - GitHub

https://github.com/shrutityagi4102/Diagnose-That/blob/main/urinary_inflammation.py