

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Data Science  Internship Report  Diabetes Prediction Health App | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | *HIMANSHU BOBADE* Bharati Vidyapeeth College Of Engineering, Navi Mumbai |  | | | |
|  | | | | Date of Joining: May 18th,2021 Mentor:Y. VishnuVardhan Date of completion of internship:  June 20th ,2021  Exposys Data labs |  | | | |
|  | | |  | | |  | | |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | |
|  | | DOMAIN: DATA SCIENCE | | |  | |
|  | |  |  |  |  | |
|  | INTRODUCTION Diabetes, a global public health problem, is now emerging as a pandemic and by the year 2025, three-quarters of the world’s 300 million adults with diabetes will be in non-industrialized countries and almost a third in India and China alone. Today, the prevalence of diabetes in the urban metros of India is approaching the figures reported in the affluent migrant Indians.  There certainly is a need of urgency to prevent the disease at it’s earliest stage and this can be done with the help of Machine Learning. It is possible to detect the early symptoms of diabetes with the help of appropriate data and thus can be prevented. Machine learning techniques like Classification can help in identifying this disease.  The Project mainly highlightes machine learning algorithms like Random-Forest, Decision Tree, Logistic Regression, Support Vector Machine(SVMs), Logistic Discriminant Analysis and K-Nearest Neighbours.  The goal of this project is to find the best algorithm with best accuracy in order the chance/probability of having diabetes with given set of data as the input.  **METHODS AND DATA**  For this project, the most relevant Dataset available on internet is taken into consideration. It is from Kaggle.com (<https://www.kaggle.com/uciml/pima-indians-diabetes-database>).  The data consists of a parameters like: Pregnancy, Glucose, Blood Pressure, Insulin, Skin Thickness, BMI, Diabetes-Pedigree-function, Age and the outcome that is whether he/she has diabetes or not. All the values are decimals and integers but also have irregularities in the data. The data preprocessing measures have been taken to get rid of the irregularities.  Further, The data is divided into two datasets:  • Training data  • Test data    There are a total 8 data columns and 1 outcome column and 2000 rows in total. In this project we have used Random-Forest, Decision Tree, Logistic Regression, Support Vector Machine(SVMs), Logistic Discriminant Analysis and K-Nearest Neighbours and found out the algorithm which works the best.  For deployment purpose, we have used FLASK framework which is a python framework which can be used for small size projects efficiently.  Pickle module is used to serve as a trained data model for the deployment purpose.  **DATA CLEANING AND PREPROCESSING**  This dataset needed some cleanings and modification. Besides some feature representation should be done.  This data mainly have outliers and irregularities in the following parameters:   * Glucose: Glucose cannot be zero. * Blood Pressure: cannot be zero and cannot be greater than 250. * SkinThickness : cannot be zero * Insulin: cannot be zero * BMI: cannot be zero * Diabetes-Pedigree-function : cannot be negative * Age: Cannot be zero * Outcome : cannot be integers other than 0 and 1.   The mean of the whole column is taken I order to replace an irregularity as 0 with its mean value.  Eg. Insulin – 0 is replaced with 126.0  Now, the data is normalized and ready to be fed.  The data is splitted into train and test dataset according to the necessity of every algorithm. Generally, it is 20-25% of the total dataset for testing and rest for training purpose.  **ALGORITHMS**  Random Forest Algorithm:  Random forest is a [supervised learning algorithm](https://builtin.com/data-science/supervised-learning-python). The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.  **Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.**  One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:  two tree random forest  Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor.  **Decision Tree**  Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.  Decision_Tree (2)  Support Vector Machines  The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.  https://miro.medium.com/max/600/0*9jEWNXTAao7phK-5.pnghttps://miro.medium.com/max/600/0*0o8xIA4k3gXUDCFU.pngPossible hyperplanes  To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.  The [Logistic Function](https://en.wikipedia.org/wiki/Logistic_function)  Also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.  1 / (1 + e^-value)  Where e is the [base of the natural logarithms](https://en.wikipedia.org/wiki/E_(mathematical_constant)) (Euler’s number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform. Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.  Logistic Function K-Nearest Neighbor(KNN)K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.  * K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. * K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. * K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. * It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset   K-Nearest Neighbor(KNN) Algorithm for Machine Learning  Linear discriminant analysis  Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in [statistics](https://en.wikipedia.org/wiki/Statistics) and other fields, to find a [linear combination](https://en.wikipedia.org/wiki/Linear_combination) of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier), or, more commonly, for [dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) before later [classification](https://en.wikipedia.org/wiki/Statistical_classification).  LDA is closely related to [analysis of variance](https://en.wikipedia.org/wiki/Analysis_of_variance) (ANOVA) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), which also attempt to express one [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) as a linear combination of other features or measurements. However, ANOVA uses [categorical](https://en.wikipedia.org/wiki/Categorical_variable) [independent variables](https://en.wikipedia.org/wiki/Independent_variables) and a [continuous](https://en.wikipedia.org/wiki/Continuous_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), whereas discriminant analysis has continuous [independent variables](https://en.wikipedia.org/wiki/Independent_variables) and a categorical dependent variable (*i.e.* the class label).  [Logistic regression](https://en.wikipedia.org/wiki/Logistic_regression) and [probit regression](https://en.wikipedia.org/wiki/Probit_regression" \o "Probit regression) are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method.  **Algorithm Efficiency Comparison:**    This is an actual Comparison output from the algorithms tested.    **Deployment**  Flask is a web application framework written in Python. It has multiple modules that make it easier for a web developer to write applications without having to worry about the details like protocol management, thread management, etc.  Flask gives is a variety of choices for developing web applications and it gives us the necessary tools and libraries that allow us to build a web application. The dataflow goes as shown:    **Conclusion**  The project helps to find an efficient way to calculate or predict is the person has diabetes or not. Also, I got to learn a lot about various algorithms and also flask framework for deployment purposes.  **Reference**  Google: <https://google.com>  StackOverFlow: <https://stackoverflow.com>  GeeksForGeeks: <https://geeksforgeeks.com>  Kaggle:  <https://www.kaggle.com/uciml/pima-indians-diabetes-database> | | | | |  |