


Diabetes Prediction System

Dr. Arya S.
arya.subramonian@gmail.com

20/10/2020

Introduction

The aim of the project is building a **diabetes prediction system**. For this purpose, a machine learning model was trained using the **Pima Indian diabetes database**.¹ A **web-based app** was built to make use of this model to predict the risk of diabetes, using data obtained from the user interface. **IBM Watson Studio** and **Node-red** were utilized for this project. .

¹<https://www.kaggle.com/uciml/pima-indians-diabetes-database> 

Stages

- ▶ IBM account creation - for necessary services
- ▶ Data collection
- ▶ Data visualisation, data analysis
- ▶ Data cleaning
- ▶ Data transformation
- ▶ Model training, selection of the model with the highest accuracy
- ▶ Model deployment
- ▶ App building using Node-Red

IBM account - service creation

After creating an IBM Lite account, instances of services like **Watson Studio, Cloud Object Storage and Machine Learning** were created. A new project 'Diabetes Prediction System' was created in Watson Studio and the Machine Learning service was **associated with the project**. The Pima Indian diabetes database obtained from Kaggle was uploaded as an **asset of the project**.

ipython notebook

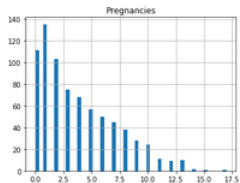
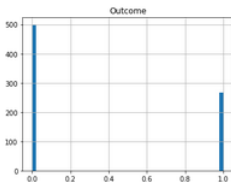
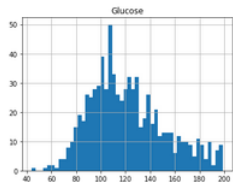
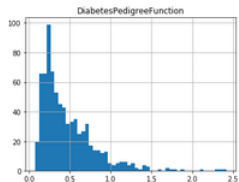
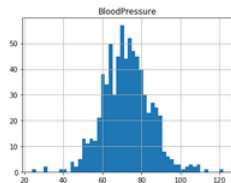
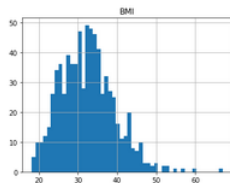
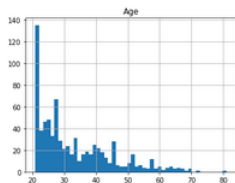
An **ipython notebook** was created for ML model training. **pandas library** was used for data handling. **scikitlearn library** was used for data transformation, splitting of data into training and test sets, and for training models.

Data cleaning

In some columns of the data, like BloodPressure, Insulin, Glucose etc. some entries are zeroes. In the case of Insulin and SkinThickness around / more than 30 percent entries are zeroes. Therefore, these two columns were dropped before training. The IBM **data refinery** was utilized to accomplish this. The edited dataset was also added as an asset to the project.

Data visualisation

pandas has many methods for the visualisation of dataframe objects. Out of the six remaining columns (Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, Age), which I used for training, the decision on how to scale the data was reached based on this visualisation.



Data transformation

'**StandardScaler**' was applied to the data corresponding to Glucose, BloodPressure and BMI, and '**MinMaxScaler**' to the data corresponding to Pregnancies, DiabetesPedigreeFunction and Age.

'**SimpleImputer**' was used to deal with missing data. Zero entries are wrong in these cases, therefore we replace these zero entries by null values and then perform imputation.

Data splitting and training

The data was split into training and test sets in the 80-20 ratio.

Five classifiers were tested -

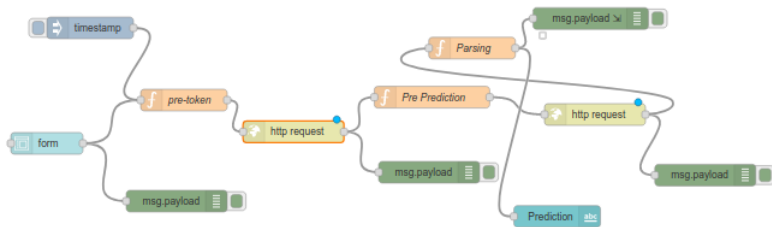
GradientBoostingClassifier, **RandomForestClassifier**, **SVC**, **KNeighborsClassifier** and **LogisticRegression**. Accuracies of the classifiers were compared.

The **highest accuracy** was obtained for **GradientBoostingClassifier**.

Model Deployment

Based on accuracy, GradientBoostingClassifier was chosen for our prediction purposes. The **model was deployed** using Watson Machine Learning credentials. A **scoring endpoint** was created and the deployed model was scored using test values. The **deployment id was used in Node-red** for creating the web-based app.

Node-red



In Node-red we make use of **inject** nodes, **form** nodes, **function** nodes, **http request** nodes and **debug** nodes for building our **flow**. Input data is obtained from the user via the interface, and this is passed to the deployed model and the prediction is displayed.

Summary

- ▶ **ML model for diabetes prediction** is built.
- ▶ The deployed model is used to **build a web-based app**.
- ▶ This is made possible using the **functionalities available on IBM Cloud**.