Name: Vaishnavi Chitloor Venkatesh Dr. Rama Krishna Koganti

CSE 5301 Data Analysis and Modeling Techniques Date: 05/03/2020

Diabetes Prediction using ML - Pima Dataset

Abstract:

Diabetes is considered as one of the deadliest and chronic diseases which causes an increase in blood sugar. Many complications occur if diabetes remains untreated and unidentified. The tedious identifying process results in visiting of a patient to a diagnostic center and consulting doctor. But the rise in machine learning approaches solves this critical problem. The motive of this study is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Therefore machine learning classification algorithms namely Decision Tree will be used in this experiment to detect diabetes at an early stage. Experiments are performed on Pima Indians Diabetes Database (PIDD) which is sourced from UCI machine learning repository. The performances of all the algorithms are evaluated on measures like Accuracy.

Introduction:

Machine learning is the scientific field dealing with the ways in which machines learn from experience. .The purpose of machine learning is the construction of computer systems that can adapt and learn from their experience. With the rise of Machine

Learning approaches we have the ability to find a solution to the issue, we have developed a system which has the ability to predict whether the patient has diabetes or not. Furthermore, predicting the disease early leads to treating the patients before it becomes critical. The aim of this research is to develop a system which can predict the diabetic risk level of a patient with a higher accuracy.

Problem:

Diabetes, is a group of metabolic disorders in which there are high blood sugar levels over a prolonged period. Symptoms of high blood sugar include frequent urination, increased thirst, and increased hunger. If left untreated, diabetes can cause many complications. Acute complications can include diabetic ketoacidosis, hyperosmolar hyperglycemic state, or death. Serious long-term complications include cardiovascular disease, stroke, chronic kidney disease, foot ulcers, and damage to the eyes.

This dataset is from Kaggle. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

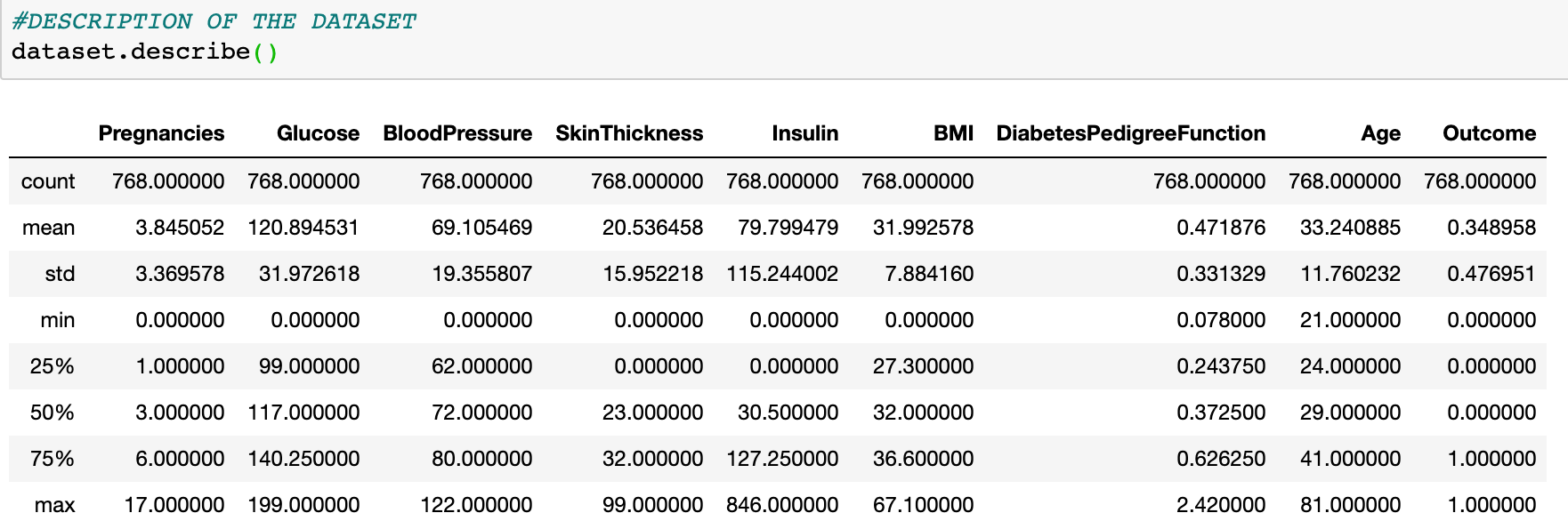
Objective:

I am building a machine learning model to accurately predict whether or not the patients in the dataset have diabetes or not.

Data

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

* Pregnancies: Number of times pregnant
* Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
* BloodPressure: Diastolic blood pressure (mm Hg)
* SkinThickness: Triceps skin fold thickness (mm)
* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)^2)
* DiabetesPedigreeFunction: Diabetes pedigree function
* Age: Age (years)
* Outcome: Class variable (0 or 1)



While designing this particular model I tried to compare 2 classifiers namely: 1)Gradient Boosting Classifier

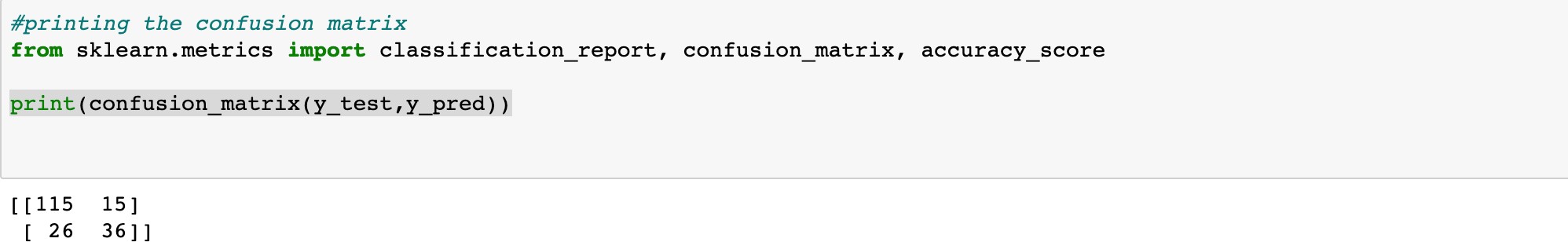
2)Random Forest Classifier

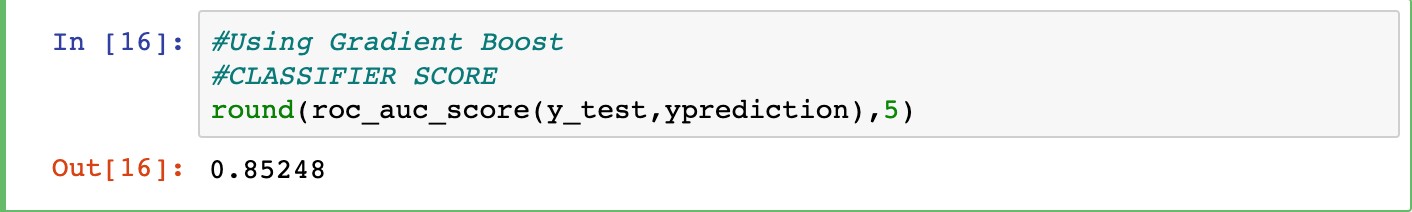
Gradient Boosting Classifier

Gradient boosting method creates step-wise process and increments the algorithm on basis of loss function. The errors are detected and rectified to improve the accuracy. Generally, boosting checks models which decrease the loss function obtained from trained samples. From these calculations the errors are measured and analyzed for optimal prediction of results. Loss function calculates the range of detected rate which compares with desired target. Onward stepwise process is most popular method for updating different with various attributes. The accuracy is optimized by reducing loss function and adding base learners at all stages

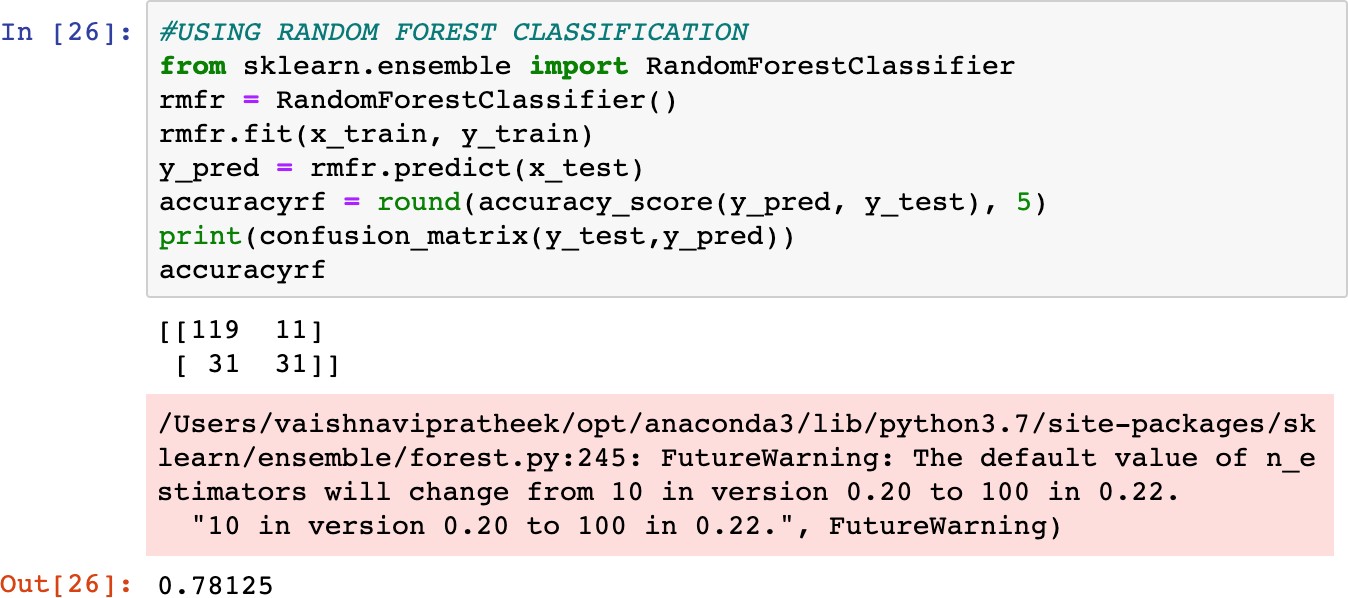
Steps for gradient Boosting method

1. Consider a sample of target values as P
2. Estimate the error in target values
3. Update and adjust the weights to reduce error M
4. P[x]=p[x]+alpha M[x]
5. Model Learners are analysed and calculated by loss function F
6. Repeat steps till desired & target result P





Random Forest Classification Results:



Conclusion:

Thus, we can see that the Gradient Bosting Classifier gives the best performance with 85.24%, followed by Random Forest with 78.1%

References:

1. https://towardsdatascience.com/pima-indian-diabetes-prediction-7573698bd5fe
2. https://towardsdatascience.com/machine-learning-workflow-on-diabetes-data- part-01-573864fcc6b8
3. ht[tps://www.researchgate.net/publication/](http://www.researchgate.net/publication/) 301335647\_Cascaded\_Modeling\_for\_PIMA\_Indian\_Diabetes\_Data

4) https://link.springer.com/article/10.1007/s40200-020-00520-5

5)https://bmcendocrdisord.biomedcentral.com/articles/10.1186/s12902-019-0436-6 6)https://ieeexplore.ieee.org/document/8355130 7)https://link.springer.com/chapter/10.1007/978-3-319-13102-3\_6 8)ht[tps://www.datacamp.com/community/tutorials/random-forests-classifier-python](http://www.datacamp.com/community/tutorials/random-forests-classifier-python)

Appendix :

Project Code:

import pandas as pd import numpy as np import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.model\_selection import train\_test\_split, cross\_val\_score from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_curve from sklearn.preprocessing import binarize

from sklearn.feature\_selection import SelectKBest from sklearn.feature\_selection import chi2

import warnings

warnings.filterwarnings("ignore", category=DeprecationWarning) #LOADING THE DATASET

dataset = pd.read\_csv('/Users/vaishnavipratheek/Downloads/diabetes.csv') dataset.head()

#DESCRIPTION OF THE DATASET

dataset.describe()

data = dataset.iloc[:,0:8]

outcome = dataset.iloc[:,8] x,y = data,outcome

#DISTRIBUTION OF DATASET INTO TRAINING AND TESTING SETS

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,random\_state=0) #COUNTING THE POSITIVE AND NEGATIVE VALUES

print(y\_test.value\_counts())

#MEAN OF THE TESTING DISTRIBUTION

print(1- y\_test.mean())

#PARAMETER EVALUATION WITH GSC VALIDATION

gbe = GradientBoostingClassifier(random\_state=0) parameters={

'learning\_rate': [0.05, 0.1, 0.5],

'max\_features': [0.5, 1],

'max\_depth': [3, 4, 5]

}

gridsearch=GridSearchCV(gbe,parameters,cv=100,scoring='roc\_auc') gridsearch.fit(x,y)

print(gridsearch.best\_params\_) print(gridsearch.best\_score\_)

#ADJUSTING DEVELOPMENT THRESHOLD

gbi = GradientBoostingClassifier(learning\_rate=0.05,max\_depth=3,max\_features=0.5,random\_state=0

)

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,random\_state=0) gbi.fit(x\_train,y\_train)

#STORING THE PREDICTION

yprediction = gbi.predict\_proba(x\_test)[:,1] #PLOTTING THE PREDICTIONS

plt.hist(yprediction,bins=10)

plt.xlim(0,1)

plt.xlabel("Predicted Proababilities") plt.ylabel(“Frequency")

#printing the confusion matrix

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

print(confusion\_matrix(y\_test,y\_pred)) #Using Gradient Boost

#CLASSIFIER SCORE

round(roc\_auc\_score(y\_test,yprediction),5) #USING RANDOM FOREST CLASSIFICATION

from sklearn.ensemble import RandomForestClassifier rmfr = RandomForestClassifier()

rmfr.fit(x\_train, y\_train) y\_pred = rmfr.predict(x\_test)

accuracyrf = round(accuracy\_score(y\_pred, y\_test), 5) print(confusion\_matrix(y\_test,y\_pred))

accuracyrf