

# Feature Extraction and Modeling on PIMA Data

**Samarth Gupta**

samarthgupta0309@gmail.com  
Dept. of Information Technology  
Vellore Institute of Technology

**Nikhil Katta**

nikhilkatta10@gmail.com  
Dept. of Information Technology  
Vellore Institute of Technology

**Kush Vashisth**

kushvashisthkv@gmail.com  
Dept. of Computer Science  
BITS Edu Campus, Vadodara

**Abstract** – Diabetes is a chronic disease with a number of possible contributing factors. In this research, the researchers use multiple ensemblers to improve the accuracy of the models used in the previous studies while using the PIMA dataset. Using feature engineering the accuracy of the K Neighbors Classifier, Random Forest Classifier, SVC, Logistic Regression, Decision Tree Classifier, Bagging, Deep Learning Grid-Search.

**Keywords**– Data Mining, Feature Engineering, Type -2 Diabetes, Weighted Diabetic Risk Score, K-fold Cross Validation.

## I. INTRODUCTION

Glucose is what provides our cells energy. Insulin, a hormone made by the pancreas, helps glucose from food to be used by the cells for energy. However, if the body doesn't make enough (or any) insulin or doesn't use the insulin produced well, glucose remains in the blood and never reaches the cells, this condition is called Diabetes. Having too much glucose in the blood is harmful for the body as it can increase the chances of health problems such as heart diseases, kidney diseases, eye problems, nerve damage, foot problems and many more.

The estimated number of diabetics (including Type 1, Type 2, Gestational, Monogenic and Cystic Fibrosis-Related Diabetes) in India is estimated to be around 40 Million (over 30 Million have been diagnosed) making it the country with the second highest number of cases in one country in the World after China. The ninth edition of IDF Diabetes Atlas estimated that India will remain in the second slot up to 2045. It also predicts that the number of Diabetes cases in India will be over 134 Million in the next 25 years.

Out of the 40 Million diabetes cases currently estimated in India, only 30 Million have been diagnosed. This implies that about 10 Million people are still unaware about diabetic conditions making them even more vulnerable to the health problems associated with diabetes.

In diabetes a delay in taking action can be fatal. In addition to this, Dr. Sangeeta Kashyap an endocrinologist at Cleveland Clinic in an interview with Healthline mentioned, "If you follow the advice of your doctors and nutritionist and make an effort to lose weight, diabetes can be reversed by normalizing your blood sugar levels without medication early in the course of the disease, that is the first three to five years," making early detection of diabetes even more rewarding than it already was.

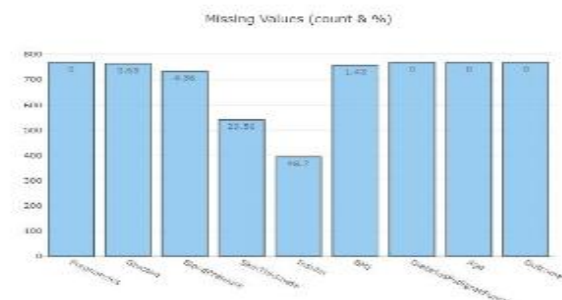
Feature Engineering is choosing the best from within the existing set of features without considerably transforming them.

Techniques for feature Engineering:

1. **Filter Method:** Chi-Square analysis/mutual information statistical Technique to extract some important features Embedded Method: Relevant score is given to build Machine Learning which decides important features .
2. **Feature Extraction :** Input features which are often unrecognizable but represent patterns are transformed into derived features.
3. **Feature Combination :** Some features are combined with one another to work better.
4. In this Research Paper we will use Feature Extraction with Feature combination mainly on the PIMA diabetic dataset.
5. All data can be distributed into Train, Test and **Validation ;**
6. Validation is used to evaluate different candidate models as if only Test and Train will be there best model will be chosen for Test. Overfitting in Test data is possible. In this paper we have used K-fold Cross Validation.

### Problem with PIMA diabetic Dataset :

The PIMA contains the missing data, which is coded in the form of zeros . Out of 768 cases in PIMA, the missing value is found for 5 patients with glucose value of zero, 11 patients had a body mass index of zero, 28 patients had a diastolic blood pressure of zero, 192 others had skin fold thickness readings of zero, and 140 others had serum insulin levels of 0, which is biologically impossible.



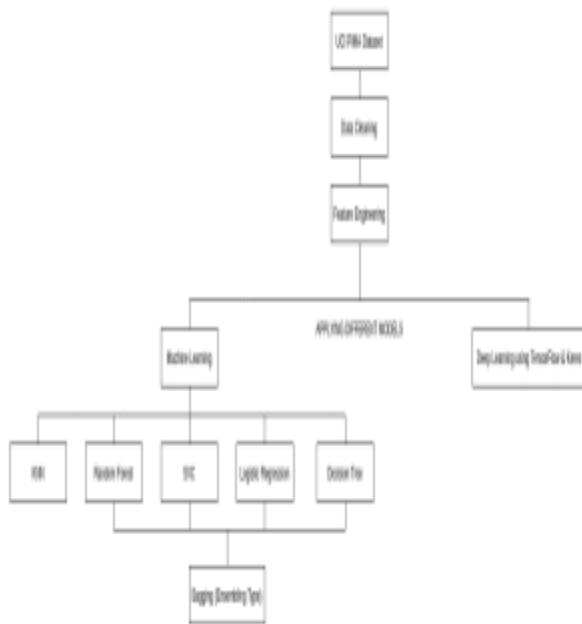


Fig.1. Approach Flow-chart.

## II. ABOUT DATA SET

The PIMA dataset contains 768 instances of women of ages 21 and above having these 8 attributes:

1. Number of pregnancies.
2. Plasma glucose concentration for 2 hours in an oral glucose tolerance test.
3. Diastolic blood pressure in mm of Hg.
4. Triceps skin fold thickness in mm.
5. 2-Hour serum insulin in mu U/ml.
6. Body mass index
7. Diabetes Pedigree Function (It provided some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient).
8. Age in years

On examining the data, the following observations can be drawn:

The number of pregnancy and age are integers.

Wherever a zero is present, there seems to be an error in the data.

Certain attributes like BMI, blood pressure and glucose are normally distributed.

Certain attributes like age, Diabetes Pedigree Function, insulin and pregnancy are exponentially distributed.

Using extraTree classifiers we find that the most important attributes are glucose, age, BMI, Diabetes Pedigree Function and pregnancy.

## III. EXPERIMENTS AND RESULTS

In recent years, research work has been done using base classifiers and classifier ensemble alone on various

medical data sets including the Pima Indian Diabetes Dataset. The accuracy of most of these classifiers is in the range of 66.6% to 77.7%. The highest accuracy of Logdisc algorithm is 77.7%.

In the Research paper, Exploring data Analysis is used with different features and domain knowledge by which we modified the data (Feature Extraction and Feature Combination)

They have been fixed with a median of features depending on their outcome (this has been taken out from given data (PIMA)).

Insulin: 102.5 for a healthy person and 169.5 for a diabetic person

Glucose: 107 for healthy person and 140 for a diabetic person

Skin thickness: 27 for healthy and 32 for diabetic

Blood Pressure: 70 for healthy and 74.5 for diabetic person

BMI: 30.1 for healthy and 34.3 for diabetic person

### Scatter plot of Glucose V/S Age

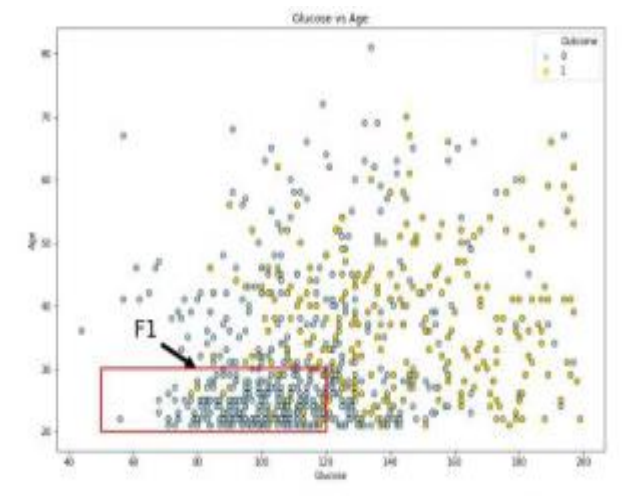


Fig.2. Glucose V/S Age.

On studying the scatter plot we discover that most non-diabetic females in the study are concentrated in the glucose  $\leq 120$  and age  $\in [20,30]$ .

According to Wikipedia "The body mass index (BMI) or Quetelet index is a value derived from the mass (weight) and height of an individual. The BMI is defined as the body mass divided by the square of the body height, and is universally expressed in units of  $\text{kg}/\text{m}^2$ , resulting from mass in kilograms and height in metres."  $30 \text{ kg}/\text{m}^2$  is the limit to obesity. If BMI is greater than or equal to 30, F2 would be 0. Else F2 will be 1.

### Scatter plot of Pregnancies V/S Age

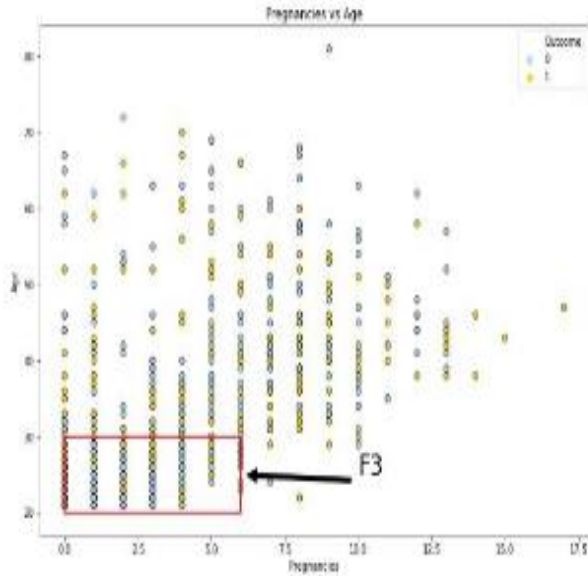


Fig.3. Pregnancy V/S Age

On studying the scatter plot we discover that most non-diabetic females in the study are concentrated in the Pregnancies  $\in [0,6]$  and Age  $\in [20,30]$ .

### Scatter plot of Glucose v/s Blood Pressure

On studying the scatter plot we discover that most non-diabetic females in the study are concentrated in the Glucose  $\leq 105$  and BloodPressure  $\leq 80$  mm of Hg

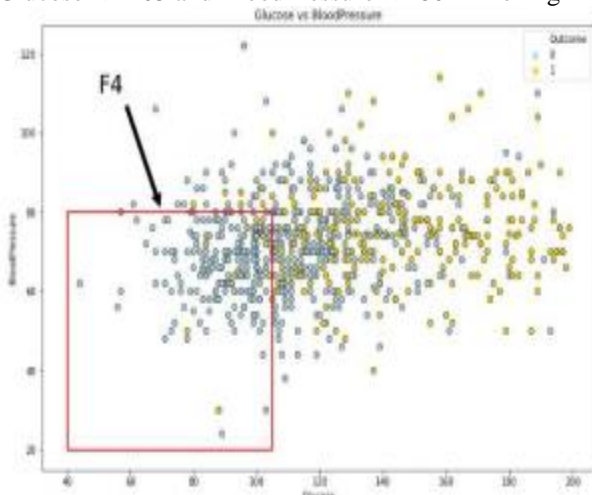


Fig.4. Glucose V/S Blood Pressure.

If Skin Thickness is less than 20 mm, F5 would be 1  
Else F5 will be 0.

### Scatter plot of Skin Thickness v/s BMI

On studying the scatter plot we discover that most non-diabetic females in the study are concentrated in the Skin Thickness  $\leq 20$  mm and BMI  $< 30$ .

### Scatter plot of Glucose v/s BMI

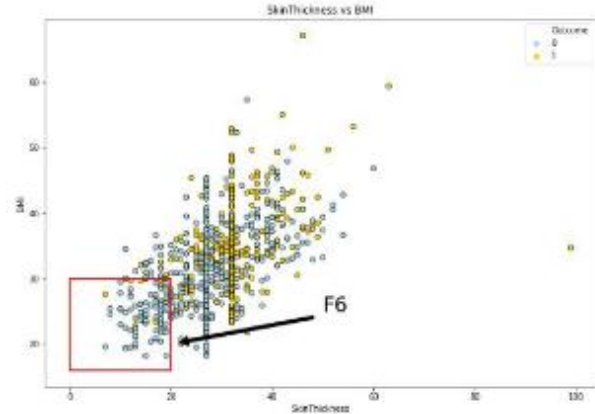


Fig.5. Skin-Thickness V/S BMI.

On studying the scatter plot we discover that most non-diabetic females in the study are concentrated in the Glucose  $\leq 105$  and BMI  $\leq 30$

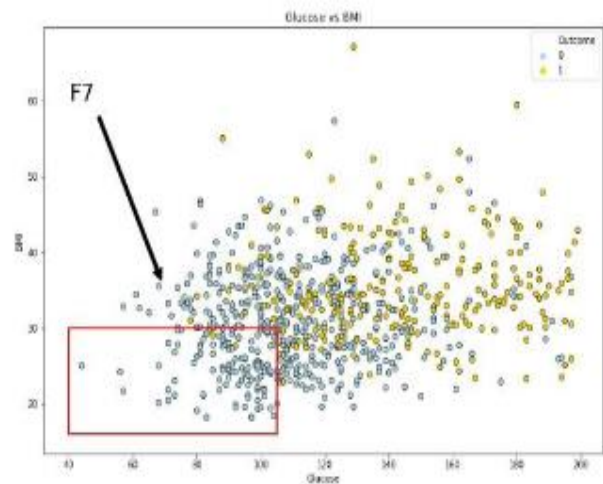


Fig.6. Glucose V/S BMI.

If Insulin is less than 200  $\mu$ U/ml, F8 would be 1. Else F8 will be 0.

If Blood Pressure is less than 80 mm of Hg, F9 would be 1. Else F9 will be 0.

If Pregnancies is less than 4 and not equal to 0, F10 would be 1. Else F10 will be 0.

F11 = data['BMI'] \* data['Skin Thickness']

Using Threshold feature engineering

If F11 is more than 1034 change it to 1 else 0

F12 = data['Pregnancies'] / data['Age']

F13 = data['Glucose'] / data['Diabetes Pedigree Function']

F14 = data['Age'] \* data['Diabetes Pedigree Function']

F15 = data['Age'] / data['Insulin']

Using these new features extracted from the previous ones we can give more weightage to certain attributes and make accuracy more than the previous paper , we use cross validation k-fold method for determining the accuracy(precision and recall)

We finally get accuracy of 88.67% with 28- neighbours (KNN algorithm)

Best Score:0.8867413632119515  
Best Parameters: {'knn\_n\_neighbors': 28}

Some of the base classifier and compared with the bagging algorithm

Accuracy: 0.8737 (+/- 0.0089) [Normal RandomForestClassifier]  
Accuracy: 0.8346 (+/- 0.0137) [Bagging RandomForestClassifier]  
  
Accuracy: 0.8281 (+/- 0.0217) [Normal SVC]  
Accuracy: 0.7930 (+/- 0.0224) [Bagging SVC]  
  
Accuracy: 0.7657 (+/- 0.0212) [Normal LogisticRegression]  
Accuracy: 0.7995 (+/- 0.0286) [Bagging LogisticRegression]  
  
Accuracy: 0.8646 (+/- 0.0231) [Normal DecisionTreeClassifier]  
Accuracy: 0.8463 (+/- 0.0136) [Bagging DecisionTreeClassifier]

Training our model with deep neural network we came with accuracy of 87.7%

We tried different models

To find the best\_batch\_size and best\_epochs required we made a model of layer (8,16,8) on kernel\_initializer = 'normal', activation='relu' optimizer = adam

```
# define the grid search parameters
batch_size = [16, 32, 64,128]
epochs = [2, 5, 10]
```

Best result got the accuracy of 85.68  
Best Epoch = 5 best batch size 16

```
Best: 0.8568033337593078, using {'batch_size': 16, 'epochs': 5}
0.8242594083677360 (0.03360411486802116) with: {'batch_size': 16, 'epochs': 2}
0.8568033337593078 (0.015089823304916936) with: {'batch_size': 16, 'epochs': 5}
0.8529326915740967 (0.025850539061531837) with: {'batch_size': 16, 'epochs': 10}
0.7371954798698426 (0.00439038277536853) with: {'batch_size': 32, 'epochs': 2}
0.8294711828231811 (0.044483596451909635) with: {'batch_size': 32, 'epochs': 5}
0.8555046319961540 (0.030702018504136057) with: {'batch_size': 32, 'epochs': 10}
0.7370002500163452 (0.10200189666090684) with: {'batch_size': 64, 'epochs': 2}
0.8368071301468266 (0.03271805294220727) with: {'batch_size': 64, 'epochs': 5}
0.8529326915740967 (0.023812870335068154) with: {'batch_size': 64, 'epochs': 10}
0.6668449163496809 (0.08174265477895343) with: {'batch_size': 128, 'epochs': 2}
0.7345895886421203 (0.08707833630754575) with: {'batch_size': 128, 'epochs': 5}
0.8386130213737480 (0.037284079368725334) with: {'batch_size': 128, 'epochs': 10}
```

Giving the same condition (initializer and kernel) and best\_batch\_size = 16 and best\_epoch = 5 to finding learning rate and dropout\_rate

```
Best: 0.8516424775123597, using {'dropout_rate': 0.0, 'learn_rate': 0.01}
0.811273394657135 (0.04106256312429461) with: {'dropout_rate': 0.0, 'learn_rate': 0.001}
0.8516424775123597 (0.01808539486052797) with: {'dropout_rate': 0.0, 'learn_rate': 0.01}
0.8130697862625122 (0.03549497418270532) with: {'dropout_rate': 0.0, 'learn_rate': 0.1}
0.7539597630500794 (0.05573953405029379) with: {'dropout_rate': 0.2, 'learn_rate': 0.001}
0.8450900424880902 (0.028083415281606437) with: {'dropout_rate': 0.2, 'learn_rate': 0.01}
0.69277650811787415 (0.06393203459300921) with: {'dropout_rate': 0.2, 'learn_rate': 0.1}
0.7902259511947631 (0.03421140506832245) with: {'dropout_rate': 0.4, 'learn_rate': 0.001}
0.841227400302087 (0.03084023308234416) with: {'dropout_rate': 0.4, 'learn_rate': 0.01}
0.6511506427688598 (0.05244526932680711) with: {'dropout_rate': 0.4, 'learn_rate': 0.1}
0.653773021697998 (0.05900473721744527) with: {'dropout_rate': 0.6, 'learn_rate': 0.001}
0.7786181211471558 (0.06694674531187101) with: {'dropout_rate': 0.6, 'learn_rate': 0.01}
0.6511506427688598 (0.05244526932680711) with: {'dropout_rate': 0.6, 'learn_rate': 0.1}
```

Similarly finding best activation function

```
Best: 0.8685765266418457, using {'activation': 'tanh', 'init': 'uniform'}
0.8477293968200683 (0.01449314504616638) with: {'activation': 'softmax', 'init': 'uniform'}
0.8490026354789734 (0.01068070435695357) with: {'activation': 'softmax', 'init': 'normal'}
0.6511506427688598 (0.05244526932680711) with: {'activation': 'softmax', 'init': 'zero'}
0.8425345897674561 (0.03035086649088564) with: {'activation': 'relu', 'init': 'uniform'}
0.8067760494782111 (0.11966826905344665) with: {'activation': 'relu', 'init': 'normal'}
0.6511506427688598 (0.05244526932680711) with: {'activation': 'relu', 'init': 'zero'}
0.8685765266418457 (0.01114477683347632) with: {'activation': 'tanh', 'init': 'uniform'}
0.863477687835693 (0.030822853357140165) with: {'activation': 'tanh', 'init': 'normal'}
0.6511506427688598 (0.05244526932680711) with: {'activation': 'tanh', 'init': 'zero'}
0.8230455756187439 (0.04762461611516047) with: {'activation': 'linear', 'init': 'uniform'}
0.8243183304971610 (0.050153469132595874) with: {'activation': 'linear', 'init': 'normal'}
0.6511506427688598 (0.05244526932680711) with: {'activation': 'linear', 'init': 'zero'}
```

Best\_activation function: tan(h) and best\_initializer: uniform

Now binding them together for best number of neurons in different layer

```
Best: 0.8815211057662964, using {'neuron1': 32, 'neuron2': 32, 'neuron3': 16}
Best: 0.8815211057662964, using {'neuron1': 32, 'neuron2': 32, 'neuron3': 16}
```

MODEL	ACCURACY
KNN	88.67%
Random Forest Classifier	87.37%
Bagging Random Forest Classifier	83.46%
SVC	82.81%
Bagging SVC	79.30%
Normal Logistic Regression	76.57%
Bagging Logistic Regression	79.95%
Decision Tree Classifier	86.46%
Bagging Decision Tree Classifier	84.63%
MLP {32,32,16}	88.15%

## REFERENCES

- [1]. "ENN-Ensemble based Neural Network method for Diabetes Classification" by G L Aruna Kumari, Padmaja P and Jaya Suma G.
- [2]. "Prediction of Diabetes using Ensemble Techniques"
- [3]. By Prema N S, Varshith V, Yogeswar J

- [4]. Designing a Model to Detect Diabetes using Machine Learning By Ms Komal Patil, Dr. S.D Sawarkar and Mrs Swati Narwane
- [5]. Ensemble Learning Model for Diabetes Classification By Nongyao Nai-arun and Punnee Sittidech.
- [6]. Improved Diabetes Prediction Model for Predicting Type-II Diabetes By Sai Poojitha Nimmagadda, Sagar Yeruva and Rakesh Siempu.
- [7]. Predicting Diabetes in Medical Datasets using Machine Learning Techniques By Uswa Ali Zia, Naeem Khan
- [8]. Deep Learning approach for diabetes prediction using PIMA Indian dataset By Huma Naz and Sachin Ahuja
- [9]. Predicting modelling and analytics for diabetes using a machine learning approach By Kaur H and Kumari V.
- [10]. Machine learning and data mining methods in diabetes research by Kavakiotis Tsave, Osalifoglou A, Maglaveras, Vlhavas and Chouvardha.