#Capstone1- Pima Indians-Predict the Onset of Diabetes

Introduction

The population for this study was the Pima Indian population near Phoenix, Arizona. The population has been under continuous study since 1965 by the National Institute of Diabetes and Digestive and Kidney Diseases because of its high incidence rate of diabetes.

For the purposes of this dataset, diabetes was diagnosed according to World Health Organization Criteria, which stated that if the 2 hour post-load glucose was at least 200 mg/dl at any survey exam or if the Indian Health Service Hospital serving the community found a glucose concentration of at least 200 mg/dl during the course of routine medical care.

Given the medical data we can gather about people, we should be able to make better predictions on how likely a person is to suffer the onset of diabetes, and therefore act appropriately to help. We can start analyzing data and experimenting with algorithms that will help us study the onset of diabetes in Pima Indians.

Background on Diabetes and Description

Diabetes is a group of metabolic diseases 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. To study the reason that leading to diabetes, a cluster of dataset about Pima Indian Diabetes was used. It is consisted of 8 predict variables and 1 response variable.

This project emphasizes on predicting the onset of diabetes in Pima Indians. Dataset has been taken from UCI ML repository. Source for this data is from National Institute of Diabetes and Digestive and Kidney Diseases. This contains data in particular, for all patients who are females and at least 21 years old of Pima Indian heritage.

Fig1- Depicts the onset of diabetes in various ethnic group in US, Pima Indians being the highest

Target Audience

The research people is Pima Indian Female diabetics, which are 768 in the dataset used. The goal of this project is to predict the probability that individual females have diabetes and detect subgroups of characteristics that are at higher risk of diabetes .The higher risk subgroups contain four variables, pregnancies, glucose, BMI and Diabetes pedigree function

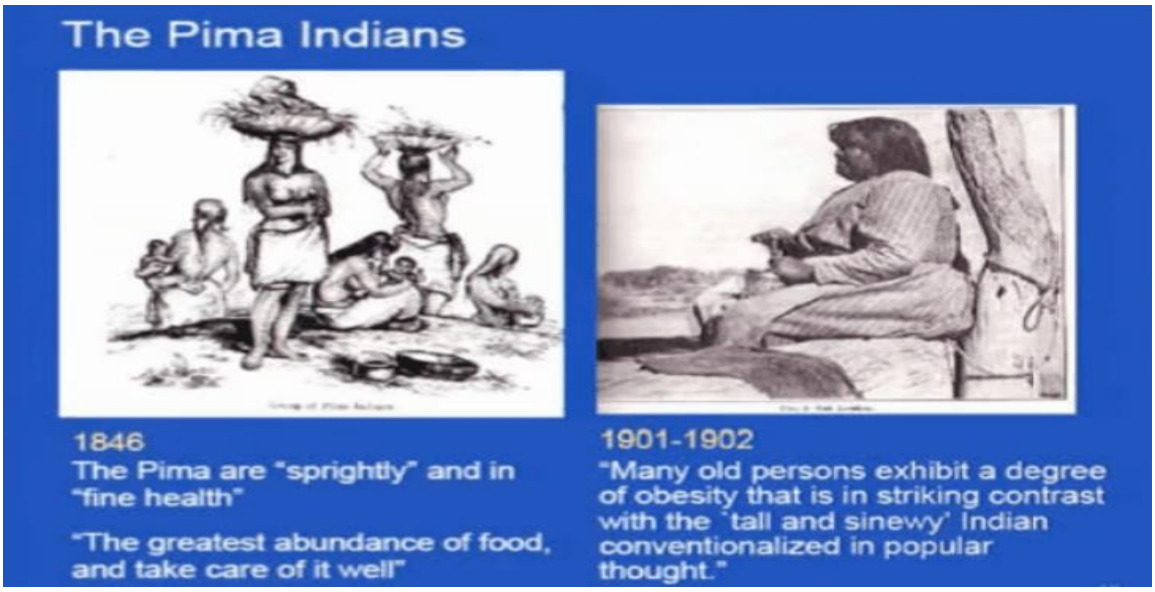


Fig2- Depicts the health history of Pima Indians

Objective

Final objective of the analysis is to make predictions on whether a person is to suffer the diabetes given the features in the dataset by determining the accuracy using various Machine learning models.

Data Acquisition- Small dataset! Why?

Dataset has been taken from UCI ML repository .Using dataset for 768 patients, 9 variables were taken to fit in a different statistics model to predict the probability that individual females have diabetes using Train and test dataset (80:20)
Our study begins with an in-depth look of how researchers that used the same dataset approached the same problem. It helped in gaining understanding of the data and pave the way for my project,

In 1988, Smith, Everhart, Dickson, Knowler, and Johannes performed an evaluation of using an early neural network model, to forecast the onset of diabetes mellitus in a high-risk population of Pima Indians. They argued that the neural network approach would provide strong results when *"the sample size is small, the form of underlying functional relationship is not known, and*

the underlying functional relationships involve complex interactions and inter correlations among a number of variables"

So, I went further and started analyzing the dataset to understand the correlations among number of variables using various machine learning models.

Data Description

We have 768 instances and the following 8 attributes of female patients at least 21 years old of Pima Indian heritage-

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

```
1 read_diab.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Fig3- Data description of Pima Indian dataset used from Python Notebook

Data Wrangling

Steps required to cleanse and modify the data into a necessary format for analysis.

1. CSV files was loaded with 768 records
2. Validated the size of the dataset.

```
In [6]:  1 read_diab.shape
Out[6]: (768, 9)
```

3. Counting the people with and Without diabetes (Out of 768 records). Upon examining the distribution of class values, it was noticed that there are 500 negative instances (65.1%) and 258 positive instances (34.9%).

```
In [17]: 1 #COUNTING THE PEOPLE WITH AND WITHOUT DIABETES- Out of 768 Pima Indian female, almost 3/4th are without diabetes
         2 read_diab.groupby("Outcome").size()
Out[17]: Outcome
         0    500
         1    268
         dtype: int64
```

4. The pregnancies and age attributes are integers.
5. The population is generally young, less than 50 years old.
6. Some attributes where a zero value exist seem to be errors in the data (e.g. Glucose, BMI and blood pressure)
7. Filled in missing values, mostly ‘0’ observed for Glucose, BMI and blood pressure. To fill in missing values, mode was used for each attributes for better analysis purpose.

Post data wrangling missing values were filled and same was used in prediction of accuracy

```
In [23]: 1 read_diab.describe()
```

Out[23]:

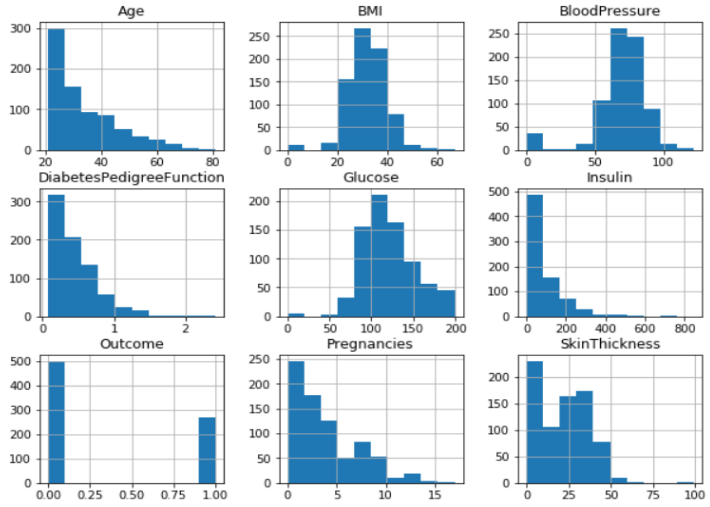
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.542318	72.295573	20.536458	79.799479	32.450911	0.471876	33.240885	0.348958
std	3.369578	30.488277	12.106756	15.952218	115.244002	6.875366	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	0.000000	0.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.375000	64.000000	0.000000	0.000000	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Exploratory Data Analysis

Most of the initial data analysis was regarding the shape of the dataset and exploration of each attributes in the data.

```
In [6]: read_diab.shape
Out[6]: (768, 9)

In [7]: read_diab.hist(figsize=(10,8))
        plt.figure()
        plt.show()
```




```
<matplotlib.figure.Figure at 0xbe8d908>
```

Fig- Initial visualization of data for each attributes

Multiple visualization plot were used for data analysis, below is the snippet for violin and Heat map visualization-

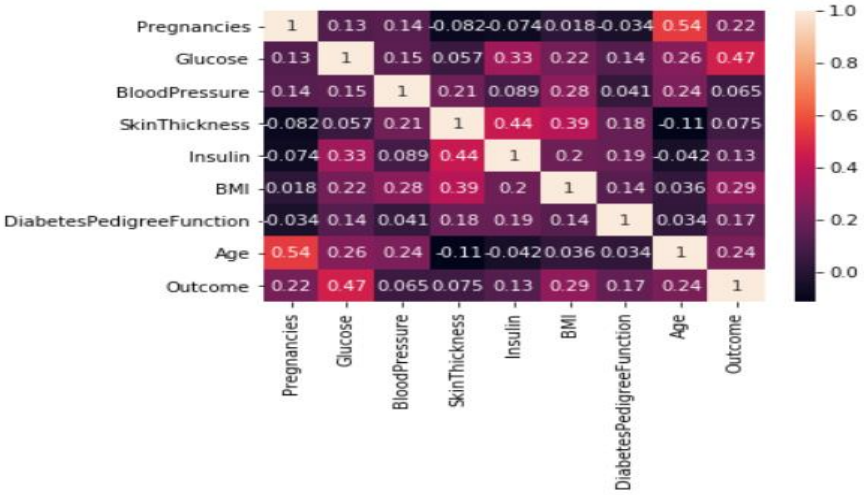


Fig- Visualization for Co-relation of each attributes using Heat map

Violin Plot Visulaization

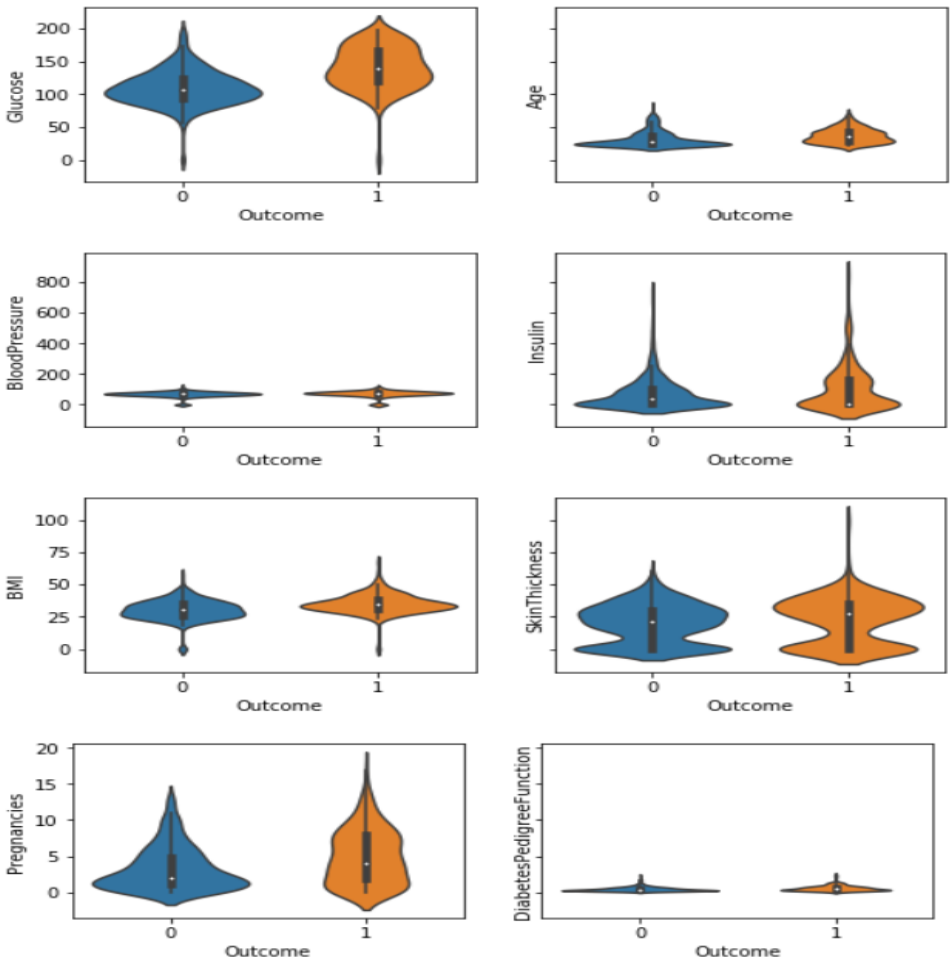


Fig :-Violin plot visualization for 8 different attributes wrt to Outcomes of diabetes

Inference from above visualizations-

1. Graph shows that the glucose ranges from 44 to 200 and there isn't much difference on outcome comparison.
2. But age and outcome comparison shows that the aged person are there who has '0' outcomes which helps to draw the thought that there is more chances of diabetes among young group of pima indians females.
3. Insulin level shows that person having onset of diabetes has high insulin level.
4. Also there seems to be an outlier in skin thickness for people having positive outcome.
5. Looking onto pregnancies plot it seems like there are more pregnant Pima indian females which has onset of diabetes.
6. Seems to be no obvious relationship between age and onset of diabetes.
7. Seems no obvious relationship between pedigree function and onset of diabetes.
8. It suggest that diabetes is not hereditary, or that the Diabetes Pedigree Function needs work.

MACHINE LEARNING

The primary goal of the machine learning (ML) section is to identify the best algorithm to predict the accuracy of onset of diabetes with various ML models. This is an unsupervised ML exercise and will focus on testing and understanding key parameters as relates to the exploratory data analysis findings above.

For validating the accuracy through each ML model, data has been split into Train and Test (80:20)

```
In [21]: X = read_diab.ix[:,0:8]
         Y = read_diab["Outcome"]
         from sklearn import model_selection
         X_train, X_test, Y_train, Y_test= model_selection.train_test_split(X, Y, test_size=0.2)

In [22]: len(X_train)
Out[22]: 614

In [25]: len(X_test)
Out[25]: 154

In [27]: len(Y_train)
Out[27]: 614

In [29]: len(Y_test)
Out[29]: 154
```

Machine Learning Algorithms

For this study, we'll take a look at the performance of below algorithms:

1. Logistics Regression
2. Linear Discriminant Analysis
3. K-Nearest Neighbor
4. Decisions Tree Classifier
5. Random Forest
6. Gaussian Naive Bayes
7. Support Vector Machines

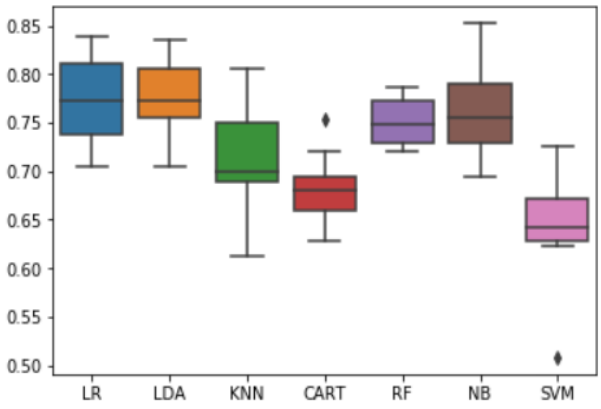
Above ML algorithm will be applied first on train data and the depending upon the performance of the model, we will fit test data to validate the accuracy.

On applying all of the above ML algorithms using Train data-

```
In [32]: results = []
         names = []
         for name,model in models:
             kfold = model_selection.KFold(n_splits=10)
             cv_result = model_selection.cross_val_score(model,X_train,Y_train, cv = kfold,scoring = "accuracy")
             kfold = model_selection.KFold(n_splits=10)
             names.append(name)
             results.append(cv_result)
         for i in range(len(names)):
             print(names[i],results[i].mean())
```

```
LR 0.770386039133
LDA 0.775145425701
KNN 0.715097831835
CART 0.680909571655
RF 0.75248545743
NB 0.760682178741
SVM 0.641591750397
```

Below is the box plot depicting the performance of all algorithms-



Based on above visualization, as these are above 75% accuracy, so I planned to fit test data into LR, LDA and RF model. Out of all best performance comes out from Random forest classifier model.

Random Forest Classifier

```
:  1 nest = [10,20,50,100,200]
   2 for i in nest:
   3         random_forest = RandomForestClassifier(n_estimators=i)
   4         random_forest.fit(X_train, Y_train)
   5         acc_random_forest = round(random_forest.score(X_train, Y_train) * 100, 2)
   6         acc_test = round(random_forest.score(X_test, Y_test) * 100, 2)
   7         print("n_estimators",i,acc_random_forest,acc_test)
   8         sns.boxplot(acc_test)
   9
```

```
n_estimators 10 98.7 74.68
n_estimators 20 99.51 76.62
n_estimators 50 100.0 73.38
n_estimators 100 100.0 79.22
n_estimators 200 100.0 75.97
```

Highest accuracy score using RF on Test data upto 79.22%

Result Summary and Conclusion-

After performing a cross-validation on the dataset, I focused on analyzing the algorithms which has more accuracy.

Based on testing, accuracy will determine the percentage of instances that were correctly classified by the algorithm. This was an important start of my analysis since gave me a baseline of how each algorithm performs.

I believe it was very interesting to see how our algorithms predict on this scale.

The data here suggests that Logistic Regression, LDA and RF performs the best on the standard, while other performed comparatively low. However, there is no clear winner between any of the algorithms.

I strongly believe that all algorithms will perform rather similarly because we are dealing with a small dataset for classification. However, the 3 algorithms should all perform better than the class baseline prediction that gave an accuracy above 75%.