Springboard

thecapstoneproject

PIMA INDIANS-PREDICT THE ONSET OF DIABETES

Author : Saurabh Anand Date: Nov, 15 2017

Springboard

Contents

Springboard

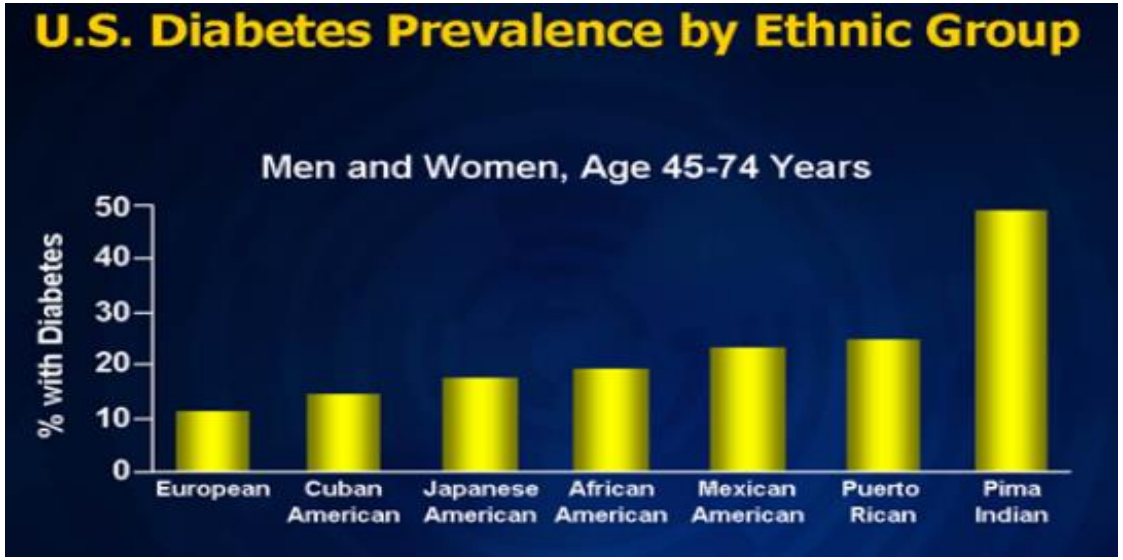
Background and Description

Background

Diabetes is a group of metabolic diseases in which there are high blood sugar levels over a prolonged period

Description

- ✓ Project emphasizes on onset of diabetes in Pima Indians
- ✓ Dataset gathered from UCI ML repository
- ✓ Data content of at least 21 yr old female of Pima Indian heritage
- ✓ Depicts the onset of diabetes in various ethnic group in US, Pima Indians being the highest



Springboard

Contents

Springboard

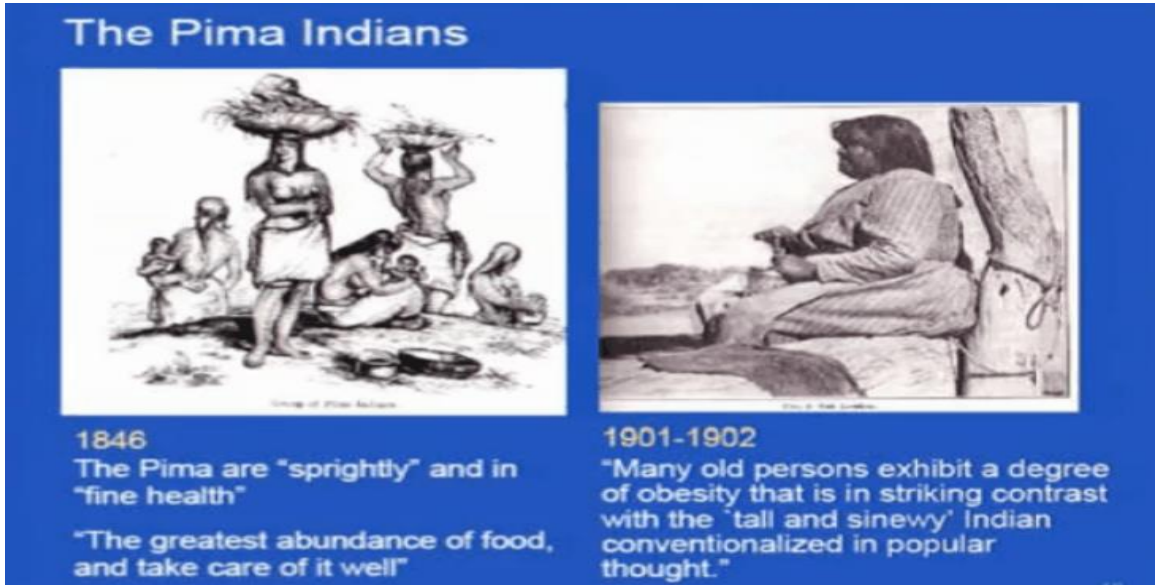
Objective and Target Audience

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

PIMA Indians Female diabetics-

- ✓ 768 Female diabetics patient info in dataset
- ✓ Predict the probability that individual females have diabetes
- ✓ Detect subgroups of characteristics that are at higher risk of diabetes



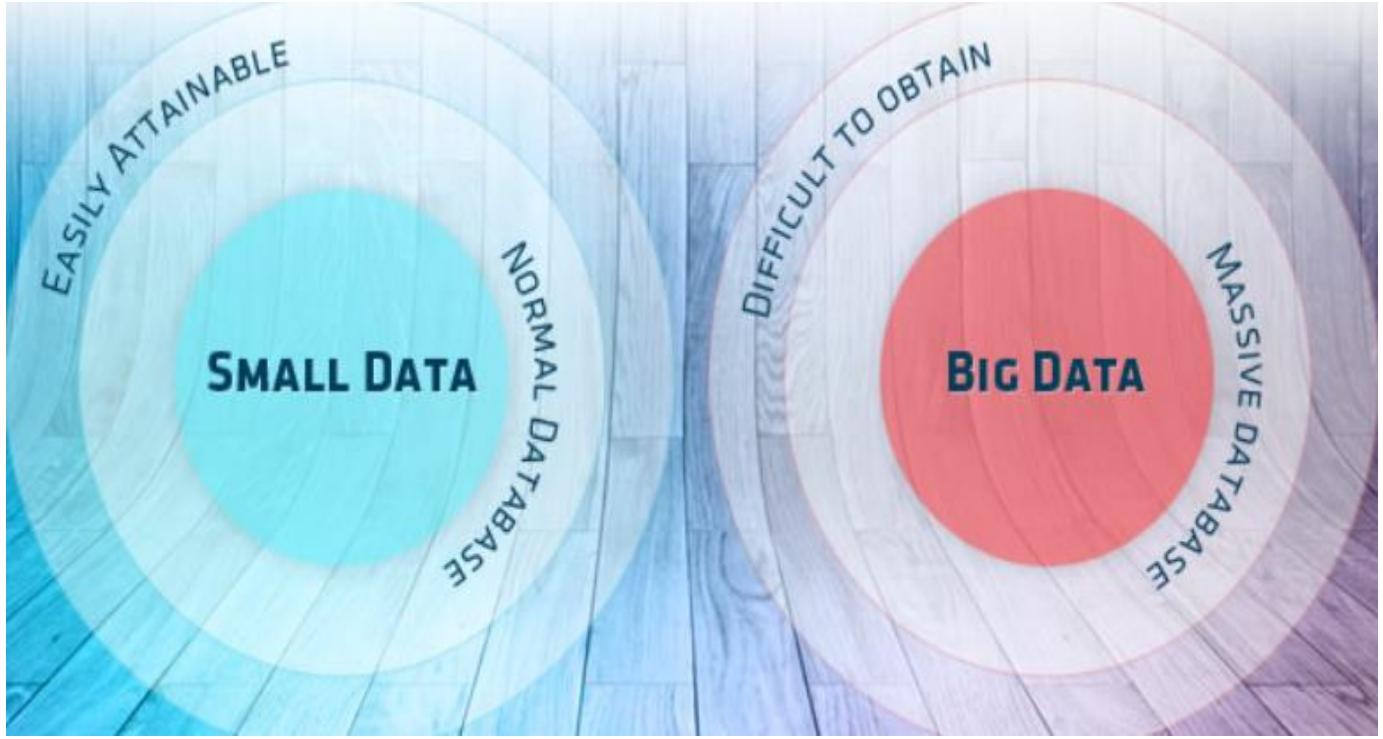
Springboard

Contents

Springboard

Data Acquisition- Small dataset! Why?

- ✓ Dataset for 768 patients with 9 variables
- ✓ Researcher argued "***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***"
- ✓ Dataset to understand the correlations among number of variables using various machine learning models.



Springboard

Contents

Springboard

Data Description

Dataset for 768 patients with 9 variables

- ✓ **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

Springboard

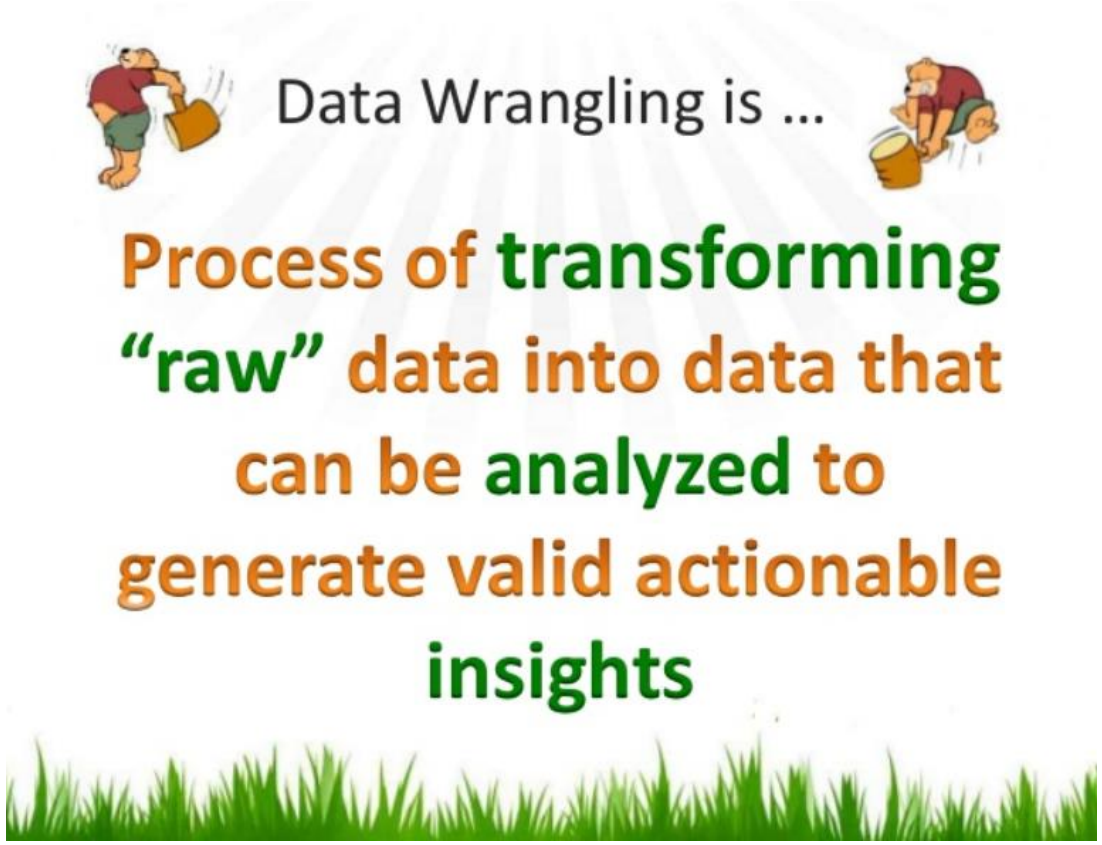
Contents

Springboard

Data Wrangling

Steps performed-

- ✓ CSV files was loaded with 768 records
- ✓ Validated shape of the data
- ✓ 500 negative instances (65.1%) and 258 positive instances (34.9%) was found
- ✓ Some attributes where a zero value exist seem to be errors in the data (e.g. Glucose, BMI and blood pressure)
- ✓ 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.



Springboard

Contents

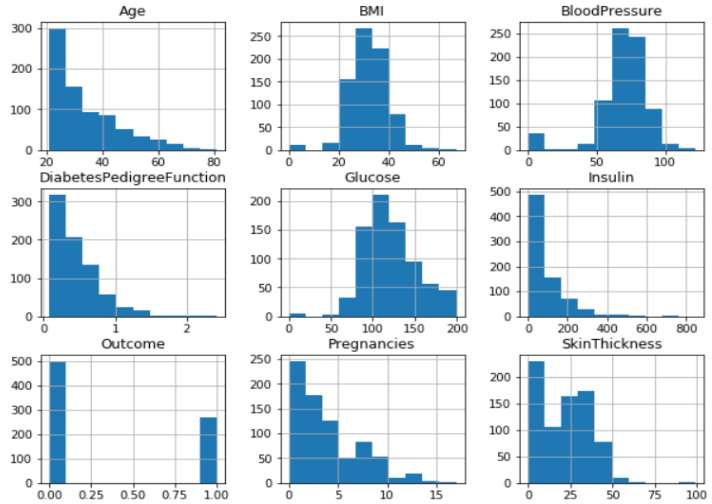
Springboard

Exploratory Data Analysis

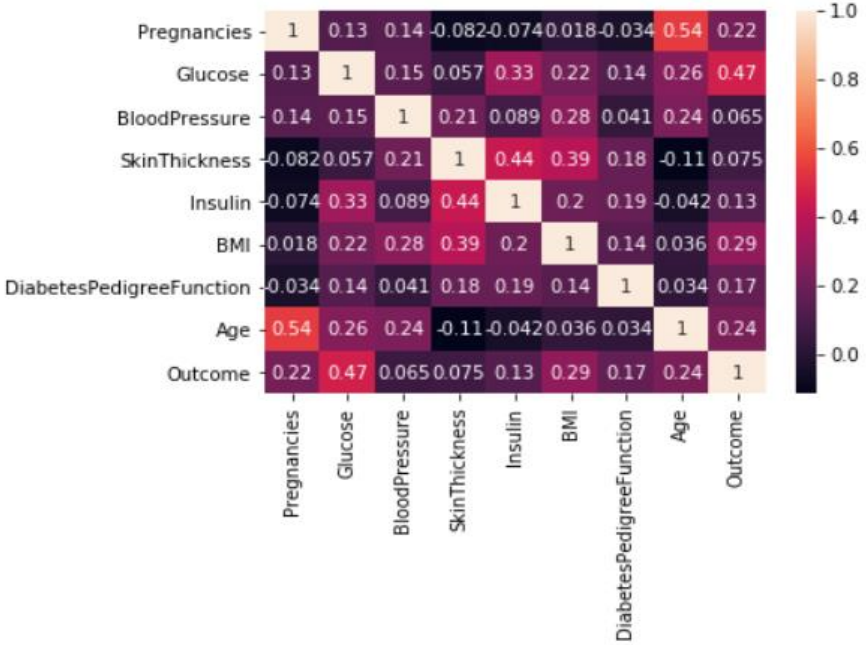
Most of the initial data analysis was regarding the shape of the dataset and exploration of each attributes in the data.

```
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>
```

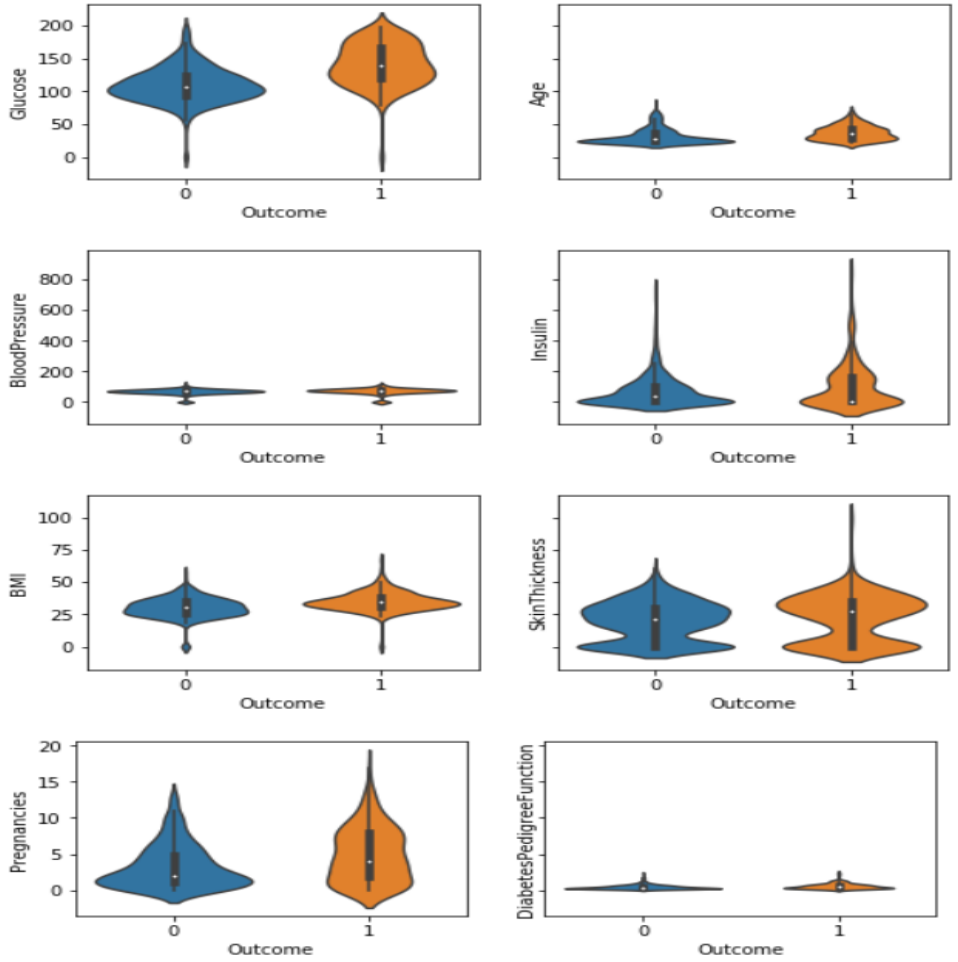


Springboard

Exploratory Data Analysis(Contd)

Inference-

- ✓ Graph shows that the glucose ranges from 44 to 200 and there isn't much difference on outcome comparison.
- ✓ 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.
- ✓ Insulin level shows that person having onset of diabetes has high insulin level.
- ✓ Also there seems to be an outlier in skin thickness for people having positive outcome.
- ✓ Looking onto pregnancies plot it seems like there are more pregnant Pima Indian females which has onset of diabetes.
- ✓ Seems to be no obvious relationship between age and onset of diabetes.
- ✓ Seems no obvious relationship between pedigree function and onset of diabetes.
- ✓ It suggest that diabetes is not hereditary, or that the Diabetes Pedigree Function needs work



Springboard

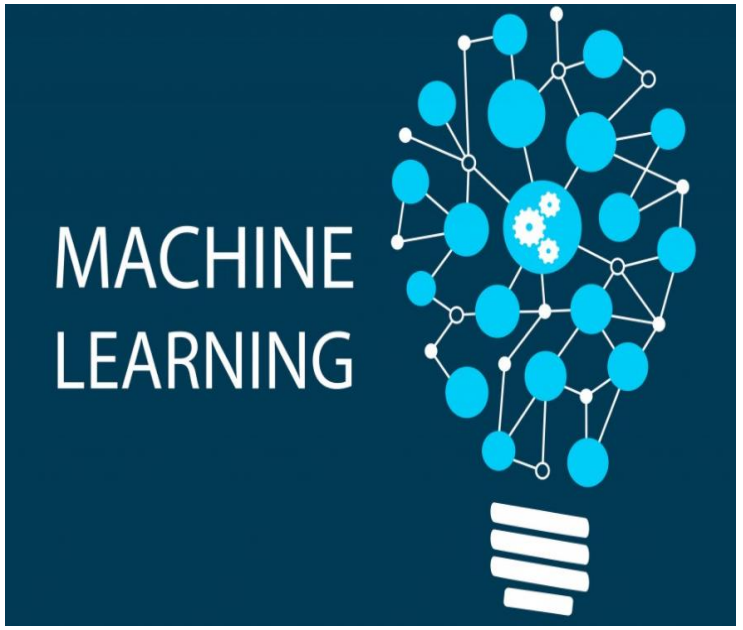
Contents

Springboard

Machine Learning

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

- ✓ Logistics Regression
- ✓ Linear Discriminant Analysis
- ✓ K-Nearest Neighbor
- ✓ Decisions Tree Classifier
- ✓ Random Forest
- ✓ Gaussian Naive Bayes
- ✓ Support Vector Machines



Springboard

Machine Learning(Contd)- 80:20(Train and Test Split)

- ✓ 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
- ✓ 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
```

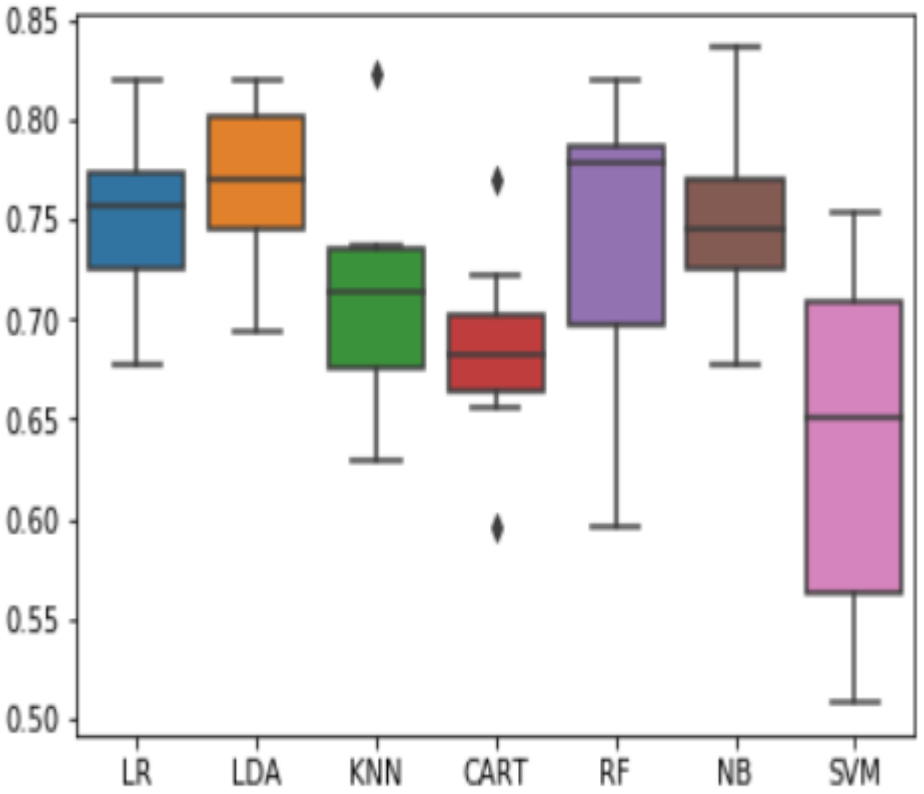
Springboard

Machine Learning(Contd)- 80:20 (Train and Test Split)

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

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
```



Springboard

Machine Learning(Contd)-Test data fit

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

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

```

```
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%

Springboard

Contents

Springboard

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%.

Springboard