Build a machine learning model to accurately classify whether or not the patients in the dataset have diabetes

Rachit Ban	**MT2019083**
Nisarg Shah	**MT2019071**
Aayushya Vadher	**MT2019003**

1. Dataset and Features

Given data set is ‘Pima Indian Diabetes

Feature Description:

Pregnancies: Number of times pregnant

Glucose: Plasma glucose concentration after 2 hours in an oral glucose tolerance test

BloodPressure: Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skinfold thickness (mm)

Insulin:2-Hour serum insulin (mu U/ml)

BMI: Body mass index (weight in kg/(height in m)^2)

DiabetesPedigreeFunction:

It provided some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient. This measure of genetic influence gives an idea of the hereditary risk one might have with the onset of diabetes mellitus.

2. Exploratory Data Analysis

Describe

```
Patient_data.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	742.000000	752.000000	768.000000	746.000000	768.000000	757.000000	768.000000	749.000000	768.000000
mean	3.866601	119.966097	68.886078	20.309879	79.799479	31.711151	0.471876	33.761336	0.348958
std	3.479971	32.367659	19.427448	15.974523	115.244002	8.544789	0.331329	12.297409	0.476951
min	-5.412815	0.000000	-3.496455	-11.945520	0.000000	-16.288921	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.100000	0.243750	24.000000	0.000000
50%	3.000000	116.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.000000	80.000000	32.000000	127.250000	36.500000	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

Head

```
Patient_data.head(10)
```

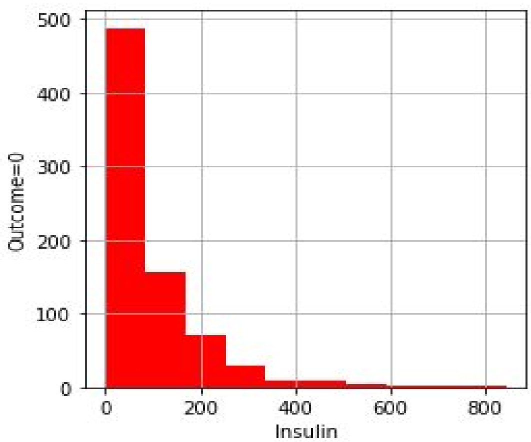
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6.0	148.0	72.000000	35.0	0	33.600000	0.627	50.000000	1
1	1.0	85.0	66.000000	29.0	0	26.600000	0.351	31.000000	0
2	8.0	183.0	64.000000	0.0	0	23.300000	0.672	32.000000	1
3	1.0	89.0	66.000000	23.0	94	19.179925	0.167	21.000000	0
4	0.0	137.0	40.000000	35.0	168	43.100000	2.288	33.000000	1
5	5.0	116.0	74.000000	0.0	0	25.600000	0.201	30.000000	0
6	3.0	78.0	43.869346	32.0	88	31.000000	0.248	26.000000	1
7	10.0	115.0	0.000000	0.0	0	35.300000	0.134	29.000000	0
8	2.0	197.0	70.000000	45.0	543	30.500000	0.158	NaN	1
9	8.0	125.0	96.000000	NaN	0	0.000000	0.232	68.636341	1

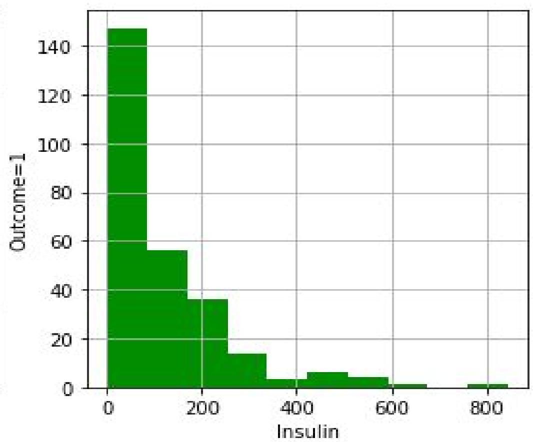
Null Values

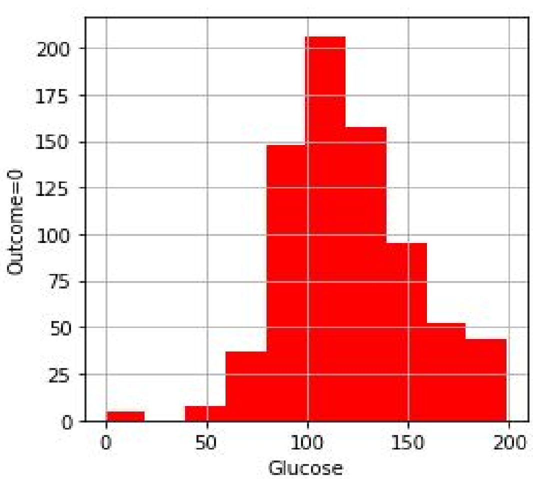
```
Pregnancies                 26
Glucose                     16
BloodPressure                0
SkinThickness               22
Insulin                      0
BMI                         11
DiabetesPedigreeFunction     0
Age                         19
Outcome                      0
dtype: int64
```

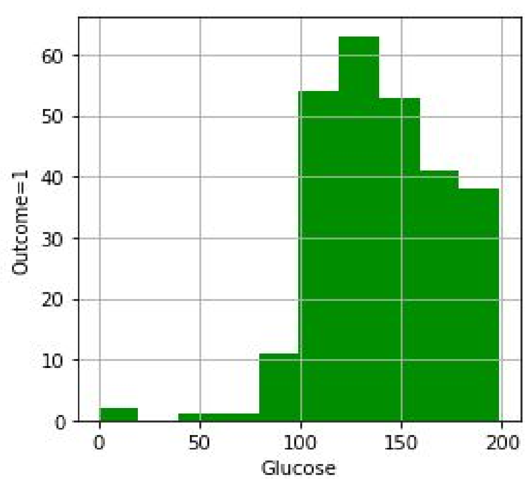
Zero Values

```
BMI:  10
Glucose:  5
Insulin:  374
SkinThickness:  215
BloodPressure:  32
Age:  0
Diabetes Pedigree Function:  0
Pregnancies:  106
```





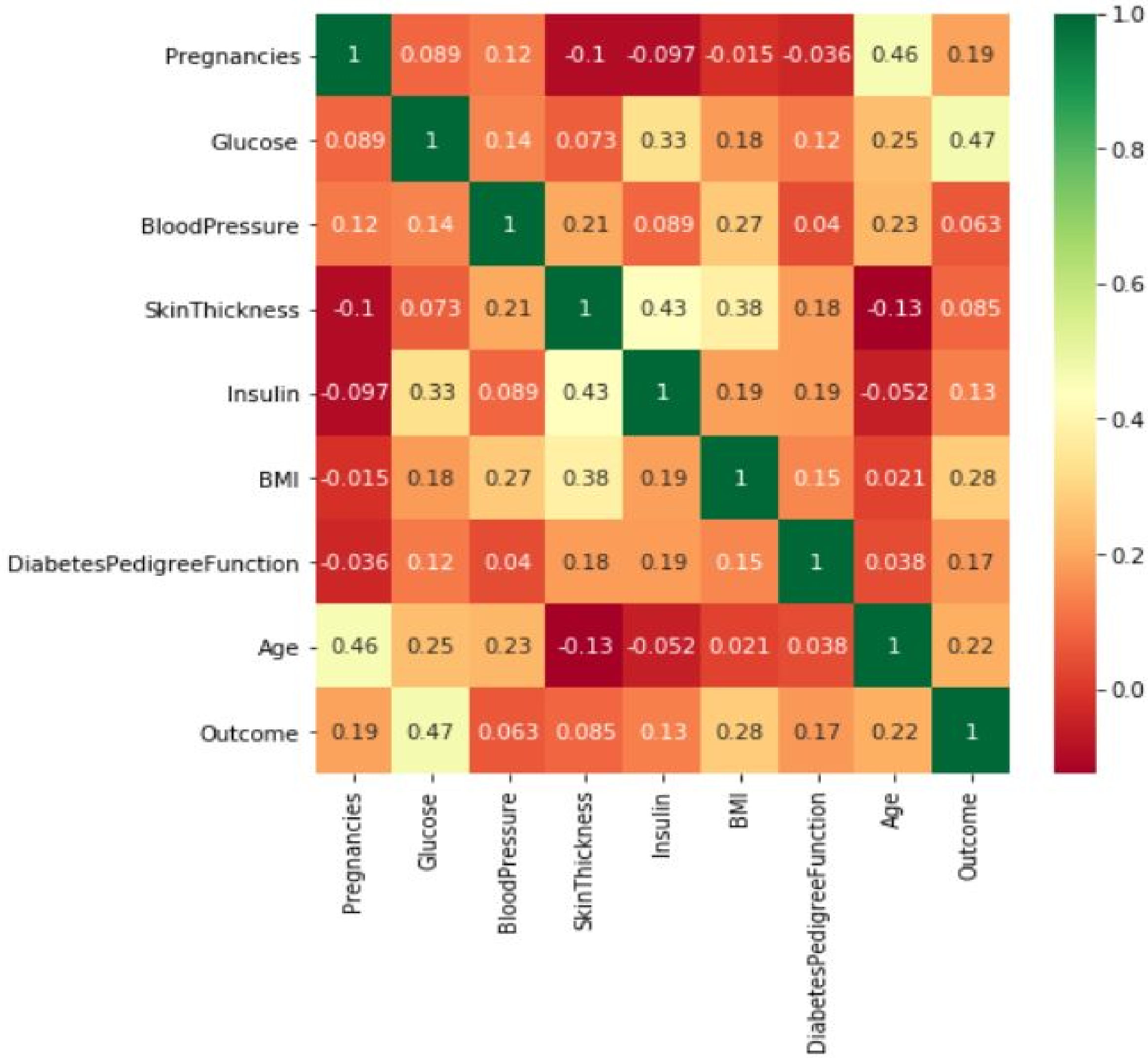




Plotting distribution graph for Glucose and Insulin with respect to outcome 0 and 1 will help to determine whether to use mean or median to replace false values.

If the distribution of values are spread normally then mean is preferred else if it is skewed than median is preferred

Correlation Matrix



The correlation matrix shows relation between features.From above we can conclude that the outcome heavily depends on Glucose.

3. Data Cleaning

From the Exploratory Data Analysis, we note that there are some invalid and missing data values occurring in data set:

1. None of Pregnancies can be -ve
2. Blood Pressure cannot be -ve or 0
3. Any person has minimum skin thickness of 0.5 mm (using domain knowledge) so it cannot be 0 or -ve
4. Age cannot be in floating point or 0(assuming age is in years)
5. BMI cannot be 0 or -ve (using domain knowledge)
6. Glucose and Insulin(for Outcome 0) cannot be 0 (using domain knowledge)
7. How we handled null and 0 values for Glucose and Insulin ?

#Functions Used For Data Cleaning

1. **impute_values_for_pragn:** If Pregnancies col. has -ve or floating values than function replaces values with mode of pregnancies.
2. **impute_values_for_bp:**If BloodPressure col. has -ve or 0 or nil than make it to 0 or integer than replace it with mean of blood pressure Here mean is used because from distribution plot of Blood Pressure we conclude that the curve is distributed equally.
3. **impute_age**:If Age col. has value 0 or nil than replace it with the median of the distribution as from the distribution plot of ages, frequent occurrence of ages between 21-29 is observed
4. **impute_values_for_skinth**: If SkinThickness col. has <0.5mm or nil than replace it with the median of the distribution because from distribution plot of SkinThickness, we can see the graph is more skewed towards a particular range.
5. **impute_glucose**:If outcome is 0 than Glucose should be less than 140 and if outcome is 1 than Glucose should be more than 140. For

Glucose mean is used because from distribution plot mean will be a better guess as graph is distributed equally.

6. **impute_insulin**:If outcome is 0 than Insulin should be between 16 to 166.If outcome is 1 than Insulin should be less than 16.For Insulin median is used because the distribution plot for insulin shows skewness so median would be a better choice.
7. **impute_bmi**:BMI cannot be -ve or 0 or null so replacing it with median as from the graph of BMI vs outcome it can be seen that values are more concentrated between 20 to 40.

4. Model building and Analysis

Logistic Regression is used in case of binary classification. Here we need to predict whether a patient has diabetes or not depending on whether outcome is 0 or 1. So this problem is similar to binary classification.

The Logistic Regression

The logistic model is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

Classification Report

	precision	recall	f1-score	support
0	0.86	0.92	0.89	100
1	0.83	0.72	0.77	54
accuracy			0.85	154
macro avg	0.84	0.82	0.83	154
weighted avg	0.85	0.85	0.85	154

Confusion Matrix

	Predicted Outcome 0	Predicted Outcome 1
Actual Outcome 0	92 (TP)	8 (FN)
Actual Outcome 1	16 (FP)	38 (TN)

5. Conclusion

As observed Logistic Regression gives accuracy of 85.06%