**Dataset:**

[Pima Indians Diabetes Database | Kaggle](https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database)

**About Dataset:**

The Pima population has been under study by the National Institute of Diabetes and Digestive and Kidney Diseases at intervals of 2 years.

(The [**Pima**](https://en.wikipedia.org/wiki/Pima_people) are a group of **Native Americans** living in Arizona. A genetic predisposition allowed this group to survive normally to a diet poor of carbohydrates for years.

In the recent years, because of a sudden shift from traditional agricultural crops to processed foods, together with a decline in physical activity, made them develop **the highest prevalence of type 2 diabetes** and for this reason they have been subject of many studies.)

The dataset includes data from **768** women of at least 21 years old of Pima Indian heritage with **8** characteristics based on certain **diagnostic measurements** included in the dataset, in particular:

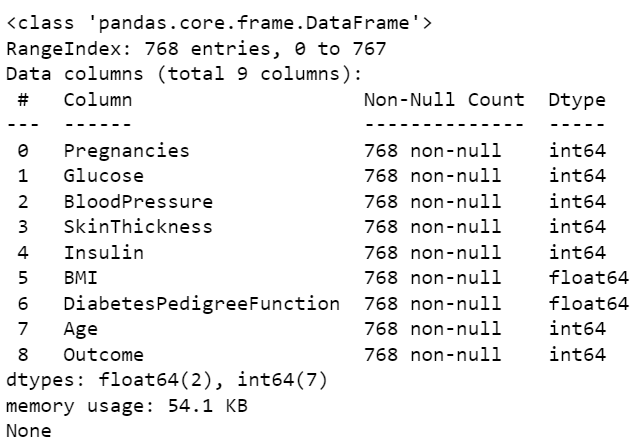
1. **Pregnancies:** No. of times been pregnant.
2. **Glucose:** Plasma Glucose Concentration (mg/dl)
3. **Blood Pressure:** Diastolic Blood Pressure(mmHg)
4. **Skin Thickness:** A value used to estimate body fat. Normal Triceps Skin Fold Thickness in women is 23mm. Higher thickness leads to obesity and chances of diabetes increases.
5. **Insulin:** 2-Hour Serum Insulin (mu U/ml)
6. **BMI:** Body Mass Index (weight in kg/ height in m2)
7. **Diabetes Pedigree Function:** It provides information about diabetes history in relatives and genetic relationship of those relatives with patients. Higher Pedigree Function means patient is more likely to have diabetes.
8. **Age:** Age (years)
9. **Outcome:** Class Variable (0 or 1) where ‘0’ denotes patient is not having diabetes and ‘1’ denotes patient having diabetes.

**To Achieve:**

Predicting the onset of diabetes based on diagnostic measures.

**Preparing Data:**

1. Analyzing the Data:
2. Information about columns-

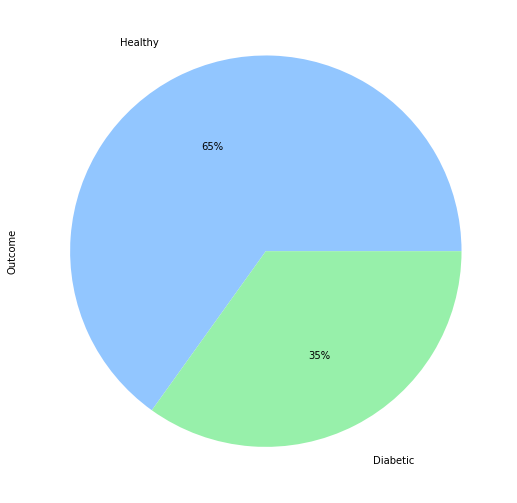


* All the attributes are **NUMERICAL** valued (as the data type is int and float).
* There are **no NULL** values.

1. Visualization of Target Variable-

Samples of healthy people: 500

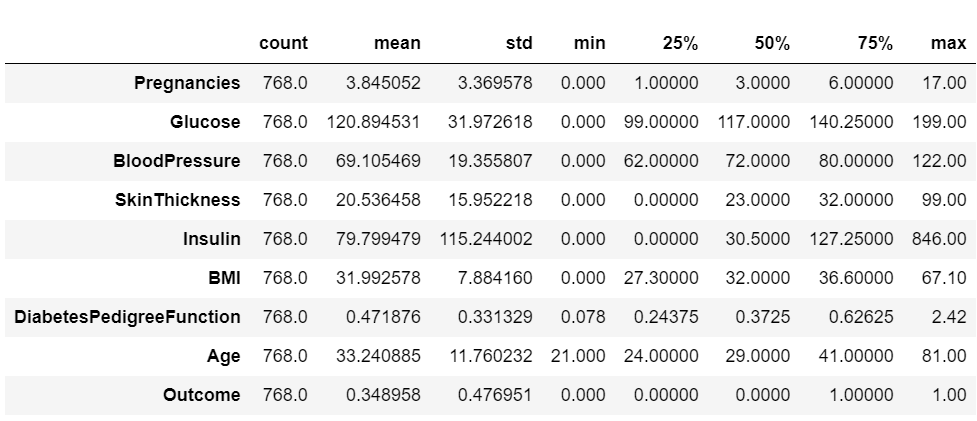
Samples of diabetic people: 268

* Classes are imbalanced.

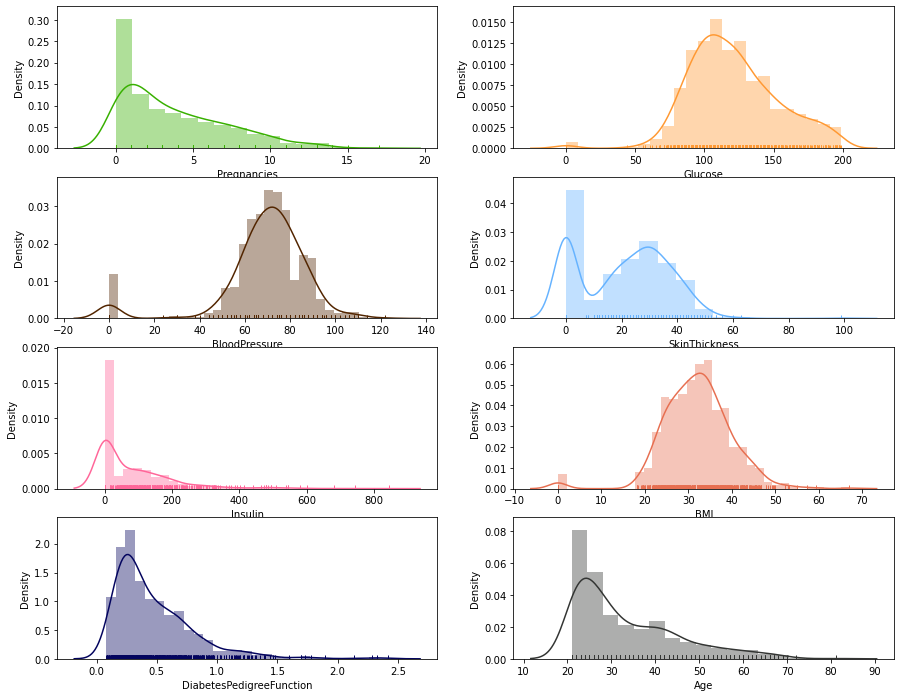
1. Checking for Duplicates-

* Result shows **no duplicate** values.

1. **Statistical Summary-**

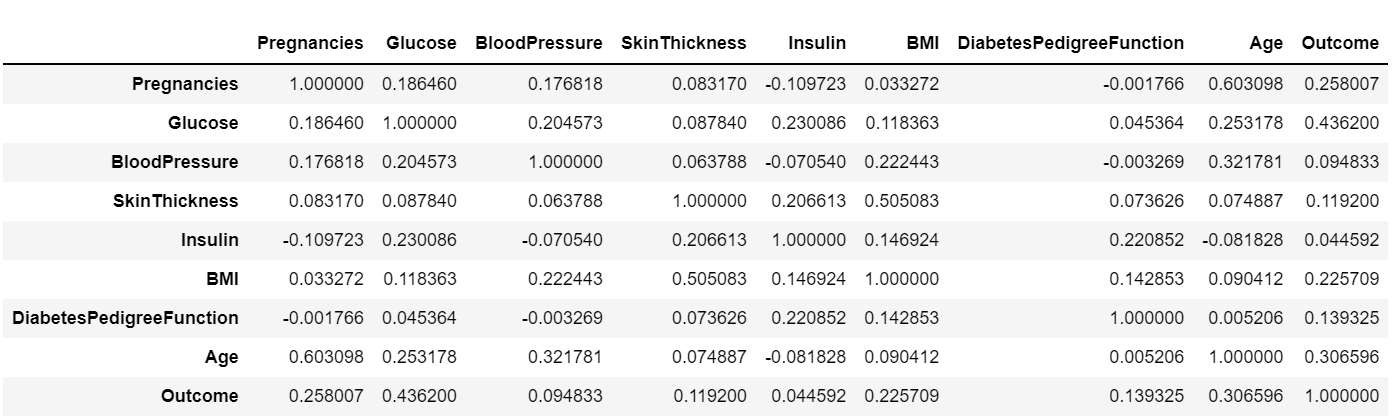
* Count column shows that there are **no Empty and NULL** values.
* Features like Insulin, BMI, Blood Pressure, Glucose and Skin Thickness cannot have "minimum" value of 0, meaning that there are **Zero values** in our dataset which are to be handled.
* Maximum values of most of the columns like Insulin, BMI, Skin thickness, etc are impossible to have showing that there are **outliers** in the data.

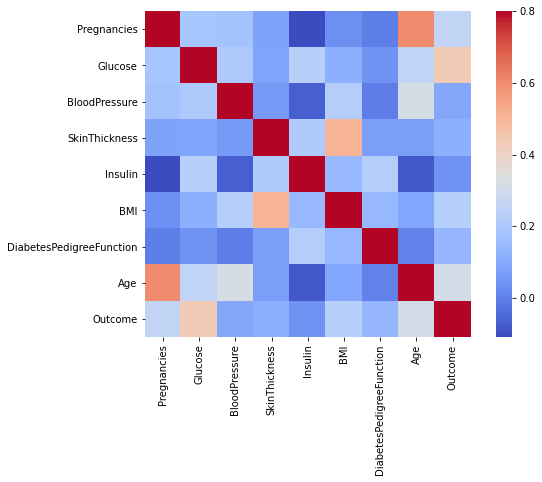
1. **Distribution Of Data-**

****

* Blood Pressure, BMI, Glucose are Normally Distributed.
* Pregnancies, Insulin, Diabetes Pedigree Function and Age are Right Skewed.
* Insulin, Skin Thickness and Pregnancies have quite huge number of Zero valued columns.

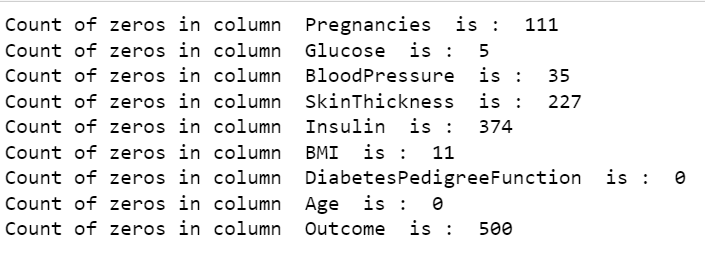
1. Correlation between features:





* Final outcome is not correlated with most of the features except Glucose (which has a correlation of 0.4).
* Also, none of the two features are correlated with each other except Age with Pregnancies and BMI with Skin thickness. So, most of the features are independent of each other.

1. Cleaning and Manipulating Data:
2. Zero Values:
3. Count of Zero Values-

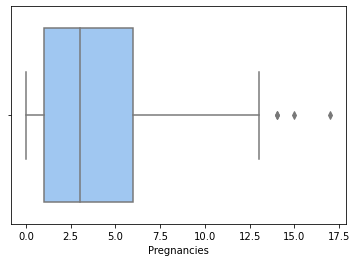
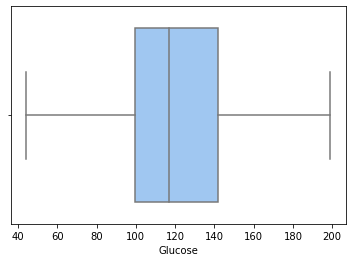


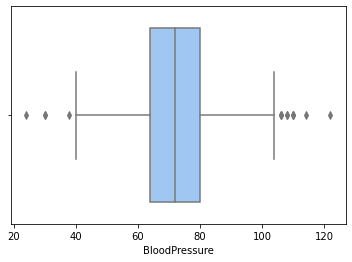
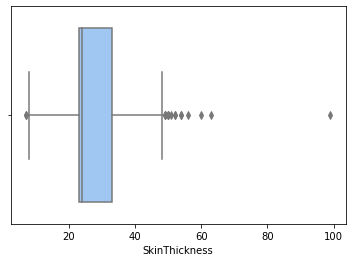
* Dataset consists of women of at least 21 years so it is unlikely that everyone would have been pregnant at least once, so ignoring the zero values in pregnancies column.
* As the Insulin and Skin Thickness values with zero are quite high and removing them would lead to huge data loss so it is better to replace them with a value.
* As Blood pressure & Glucose are critical for determining diabetes so it should not be filled with invalid values
* Also, their zero values are very less, so it is better to remove them.

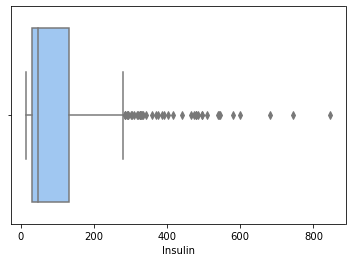
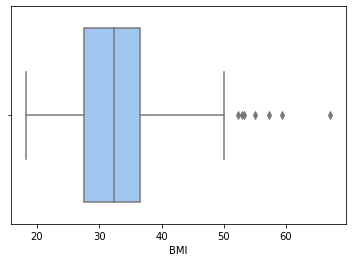
1. Handling Zero Values-

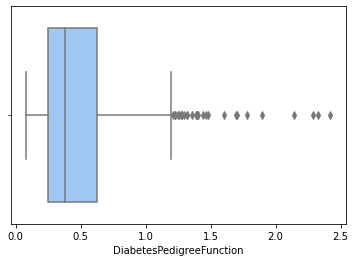
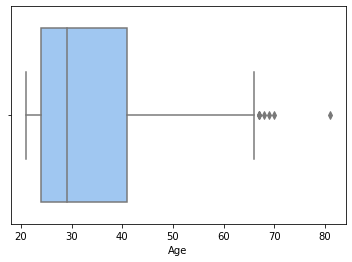
* Replacing Zeroes values of Insulin and Skin Thickness values with median values of respective features.
* Removing Zero values of Glucose, Blood Pressure and BMI.

1. Outliers-
2. Detecting Outliers:

* Outliers in Insulin, Age, Skin Thickness, Glucose, etc are to be handled.

1. Handling Outliers:
2. Z-Score:

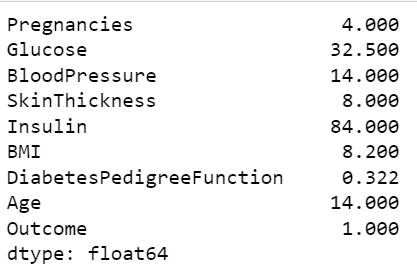
* Used a threshold of 2 or -2 that is if Z-score value is greater than or less than 2 or -2 respectively, that data point will be identified as outlier.



* Even after Z-score corrections, we seem to have outliers so performing IQR score method to remove them.

1. IQR score:

* Difference between 75th and 25th percentiles that is upper and lower quartile.
* IQR score for each feature-

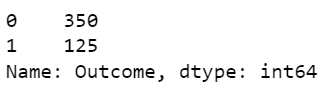


* After filtering out the outliers:

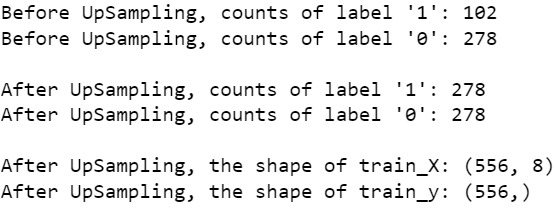


1. Balancing Classes:

* Classes are imbalanced.



* Implementing Over Sampling to make minority class equal to majority class created duplicates which led to overfitted in this case.
* Also, Working with Down Sampling left with a very small dataset than it is already which underfitted the model.
* So, Using SMOTE (Synthetic Minority Oversampling Technique) since it retains and doesn't duplicate as well.



1. Feature Scaling-

* Technique to standardize the independent features present in the data in a fixed range.
* It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.
* Standard Scaler: It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.
* In our case for example the Age ranges from 20 to 80 years old, while the number of times a patient has been pregnant ranges from 0 to 17. For this reason, we need to apply a proper transformation.

**Algorithms considered and why:**

1. Logistic Regression -

* As we are dealing with classification of discrete Outcomes. This is a good choice.
* Since, we do not have duplicates and Each Observation is independent of each other.
* Since we also have minimal collinearity among the independent variables.

**Since, Logistic regression assumes that there are no extreme outliers and The Sample Size is Sufficiently Large which doesn’t hold in our case. So, extending our approach to other approaches.**

1. Gaussian Naïve Bayes -

* **Independence of features** holds true for most of the features. (Also, we have **less data**)
* Since most of the predictors have an **equal effect on the outcome**, so choosing this model might perform better than other models.
* **Since, we are also working with the medical data, it is important to take account of all the evidence from the attributes** to make the final prediction and Naïve Bayes is good at using all the available information.

**A big data set is required for making reliable predictions of the probability of each class. We can use this with small data set but the precision will be altered.**

1. Support Vector Machine -

* Since we have **limited amount of**[**data** to analyze](https://monkeylearn.com/data-analysis/) and SVM performs better with limited number of samples.
* Since our target classes are well separated and not overlapping.

**SVMs do not perform well on highly skewed/imbalanced data sets.**

1. Decision Trees-

// Since through Data processing, we have already balanced the classes//

* As it is an effective tool that helps understand how the decisions are taken visually through a form of tree.
* [Outliers](https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm) have less significance on the decision tree’s data.
* Requires little data preparation.

**Since, Decision Trees are prone to overfitting and extremely small changes in data also can result in a drastically different tree.**

**However, using multiple decision trees and combining their results will do great which is called ensemble.**

1. Random forest-

* Avoids and prevents overfitting by using multiple decision trees.

**But They do not fit elaborate and highly detailed things well when the sample size is low and poor performance on imbalanced datasets. So, using other method called Boosting.**

1. Gradient Boosting-

* Gradient Boosting performs well when you have unbalanced data.

1. K-Nearest Neighbor-

* **It is Sensitive to outliers.**

**Challenges:**

1. **Pool of Resources:**

* While it is true that the internet is full of resources that offer a wide understanding of ML and there is also access to near unlimited learning material for most disciplines.
* But I was often lost in this pool of knowledge.

1. **Dilemma of choosing:**

* As there is a famous phrase saying “No Free Lunch” in ML, it’s true that there is no fastened rule on what algorithm to choose just with the problem statement.
* There are a lot of algorithms to choose from and instead of using a random algorithm but deciding upon an algorithm just with the data was quite challenging as a beginner.

1. **Always place for more Improvisation:**

**Achievements:**

1. **Appropriate path:**

* With all the unlimited resources scattered all over internet, First I had to break down the journey into parts to make things simpler and started to develop a basic understanding of ML terms and concepts before using them in algorithms.
* Its all about figuring out which method is going to work for you, whether you like their way of teaching or prefer a different method or style of learning.

1. **Filtering and choosing:**
   * **I have understood that choosing the right algorithm lies within understanding what sort of problem we are working with and what form of algorithms works best for each issue.**
   * **Most importantly, how the data is distributed and what sort of data we are working with.**
2. **Practical Usage:**

* Understanding a lot of new theoretical topics and implementing practically and seeing results in a matter of few weeks was quite a good progress that I have observed.

**My Learning:**

1. **Workflow:**