EDA, Feature Engineering and Analysis

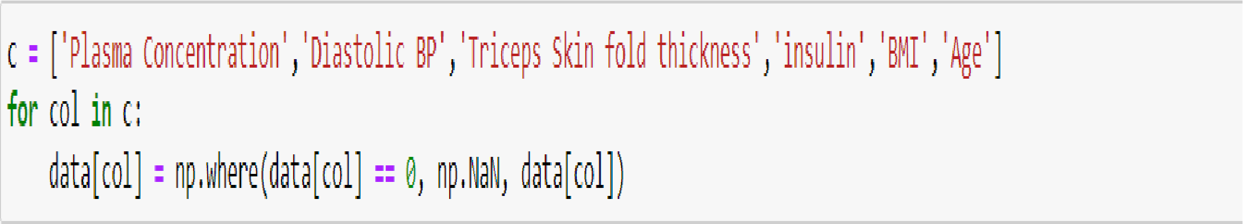
About the dataset:

1. All the patients are females, 21 years or older of Pima Indian Heritage
2. Dataset contains 768 Rows and 8 columns, out of this first 7 columns are independent features, and the 7th column is the dependent variable showing if the patient is diabetic or not.
3. Features:
    I. Number of times Pregnant
    II. Plasma Concentration
    III. Diastolic BP
    IV. Triceps Skin fold thickness
    V. Insulin
    VI. BMI
    VII. Age

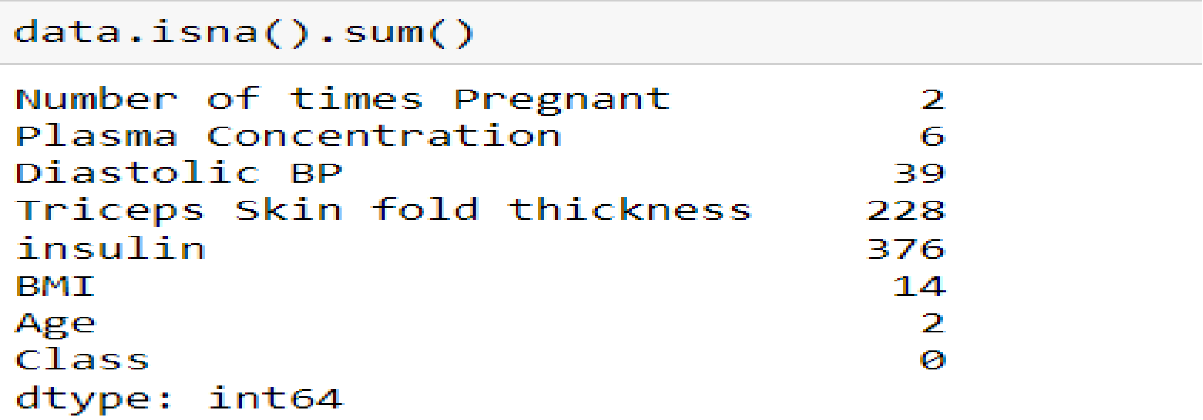
Feature Engineering:

- As all the features contains some value as ‘0’ and we cannot have ‘0’ Plasma Concentration, Diastolic BP, Triceps Skin fold thickness, Insulin, BMI and we doesn’t have very large dataset So I will be treating all the ‘0’ value as NULL.

- Now we can check how many null values we have, and we will treat each feature individually.



```
c = ['Plasma Concentration','Diastolic BP','Triceps Skin fold thickness','insulin','BMI','Age']
for col in c:
    data[col] = np.where(data[col] == 0, np.NaN, data[col])
```



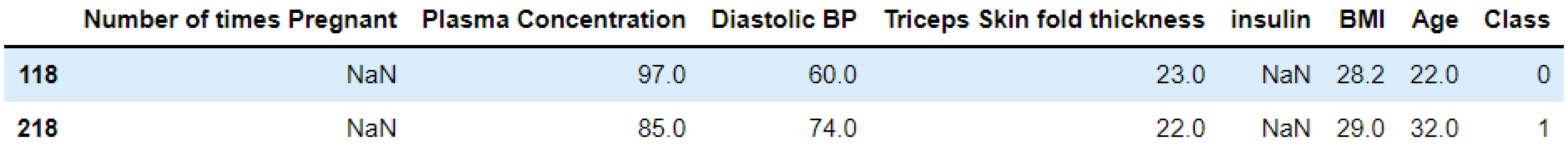
```
data.isna().sum()
```

```
Number of times Pregnant          2
Plasma Concentration              6
Diastolic BP                     39
Triceps Skin fold thickness     228
insulin                         376
BMI                              14
Age                               2
Class                             0
dtype: int64
```

Feature Engineering:

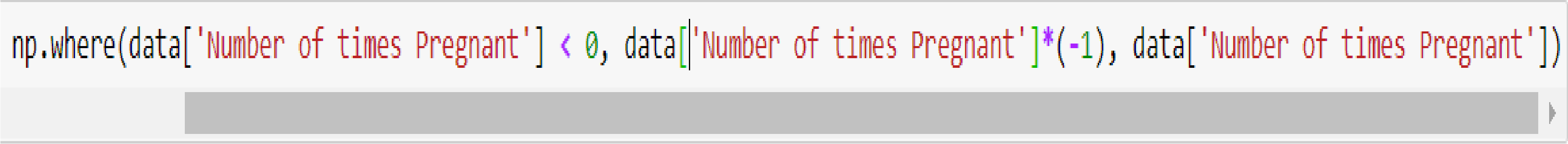
1. Number of times Pregnant

- Contains 2 NaN and one value as -13
- As number of 1 is 135 and 0 is 113 and Null value has the Class 1 and 0 both
- So to be on the safer side I replaced both NaN as 0



	Number of times Pregnant	Plasma Concentration	Diastolic BP	Triceps Skin fold thickness	insulin	BMI	Age	Class
118	NaN	97.0	60.0	23.0	NaN	28.2	22.0	0
218	NaN	85.0	74.0	22.0	NaN	29.0	32.0	1

- Next I treated the -13 as the Mod value as I consider this as a typo error.

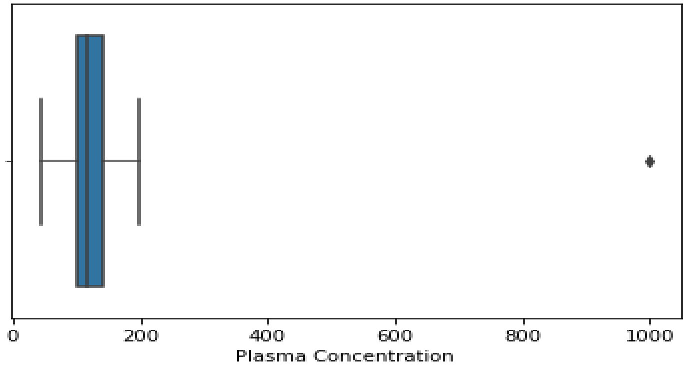


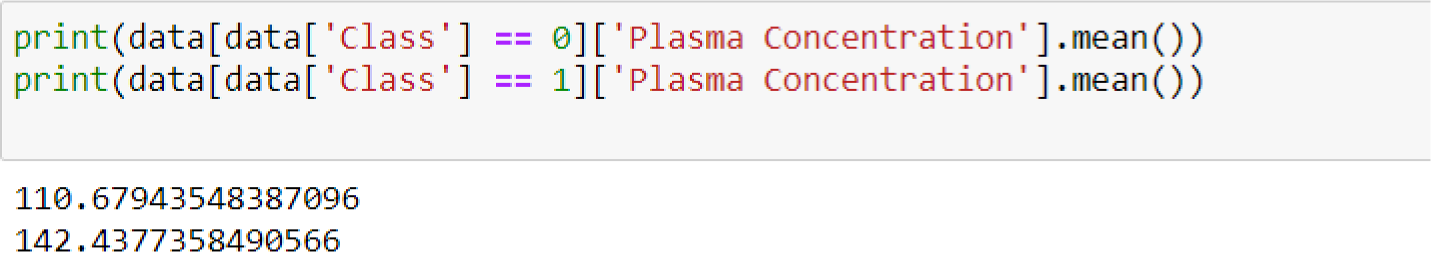
```
np.where(data['Number of times Pregnant'] < 0, data['Number of times Pregnant']*(-1), data['Number of times Pregnant'])
```

Feature Engineering:

2. Plasma Concentration

- Contains 6 NaN and an outlier
- I treted Outlier as the null value
- Now I have 7 Null values, as 4 as Class 0 and 3 has 1
- So I replaced both the values with the Mean of both the Classes





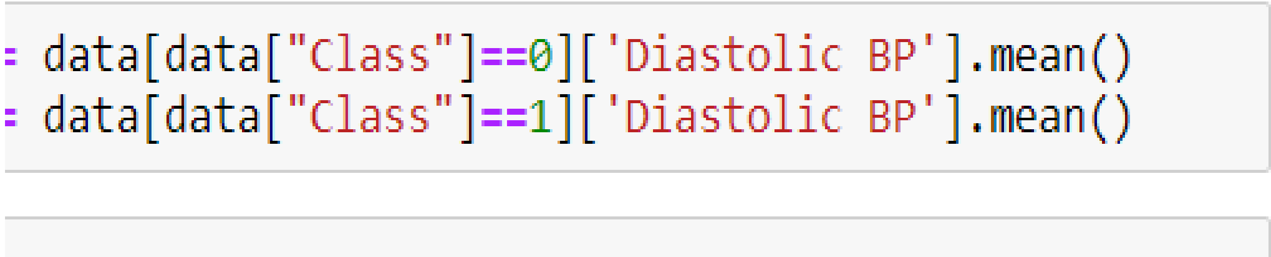
```
print(data[data['Class'] == 0]['Plasma Concentration'].mean())
print(data[data['Class'] == 1]['Plasma Concentration'].mean())
```

```
110.67943548387096
142.4377358490566
```

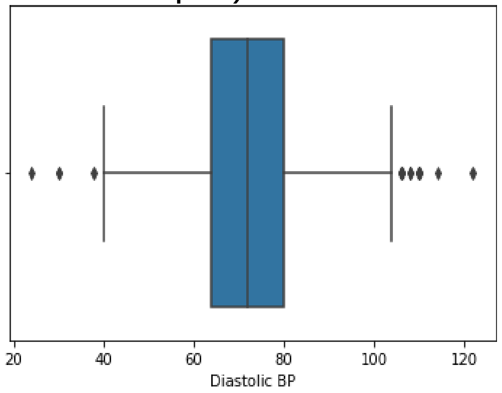
Feature Engineering:

3. 'Diastolic BP’

- Contains 39 NaN and it also contains 4 values less than 40 which could possible be the wrong and also they are the outliers
- So I treted these value as the null value
- Now I have 43 Null values, Class 1 = 19 and Class 0 = 24
- I replaced both the values with the Mean of both the Classes (Both Mean and Median are equal) and is



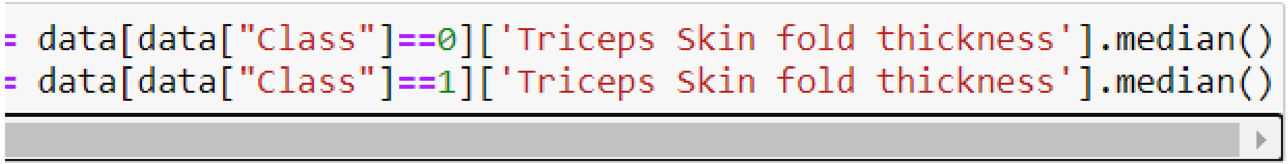
```
= data[data["Class"]==0]['Diastolic BP'].mean()
= data[data["Class"]==1]['Diastolic BP'].mean()
```



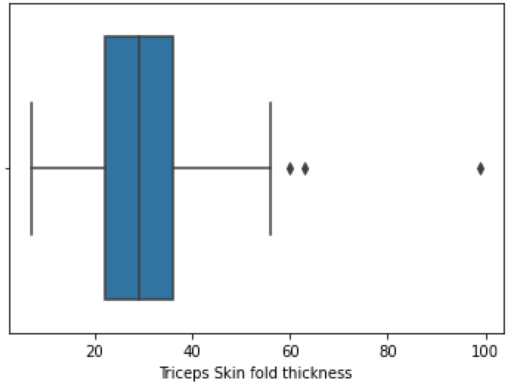
Feature Engineering:

4. Triceps Skin fold thickness

- Contains 228 NaN and it also contains 3 values as outliers, which I consider them as NaN and treat it as other nulls.
- I replaced both the values with the Median as it is a continuous values for both the classes and this feature is not symmetric.

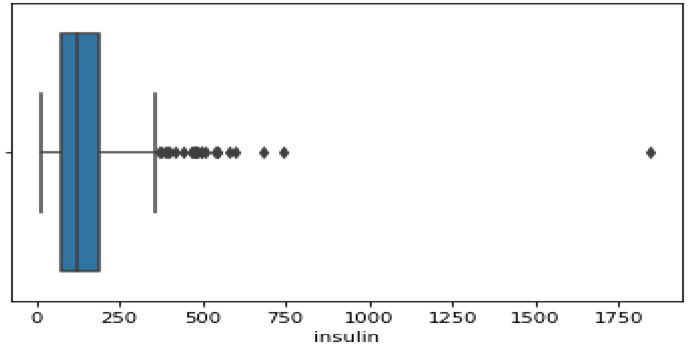


```
= data[data["Class"]==0]['Triceps Skin fold thickness'].median()
= data[data["Class"]==1]['Triceps Skin fold thickness'].median()
```

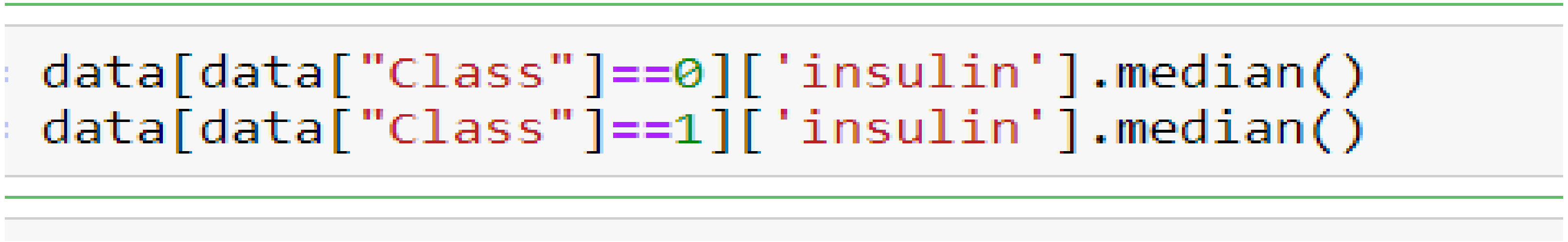


Feature Engineering:

5. Insulin



- Contains 376 NaN and it also contains lots of ourliers.
- I replaced all the values greater than 350 as null and treated them as null.
- Now I have 401 nulls.
- I replaced both the classes values with the Median as it is a continuous values for both the classes and this feature is not symmetric.



```
data[data["Class"]==0]['insulin'].median()
data[data["Class"]==1]['insulin'].median()
```

Feature Engineering:

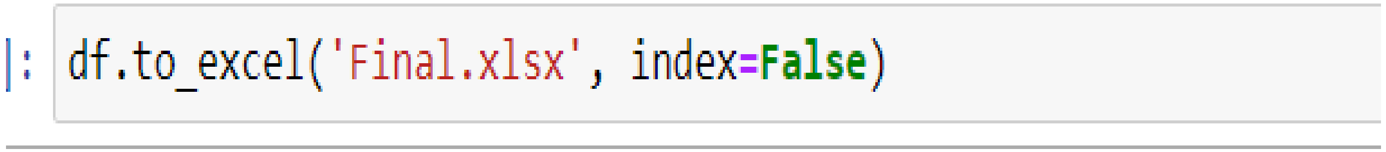
6. BMI

- Contains 14 NaN.
- I replaced both the classes values with the Median as it is a continuous values for both the classes and this feature is not symmetric.

7. Age

- Contains 14 NaN and a value less than 0 and I treated it as a null value as the age of the patient was 21 and above so this doesn't look like a typo error.
- I replaced the null value with the Mode of the Age (22) as this is a skewed feature.

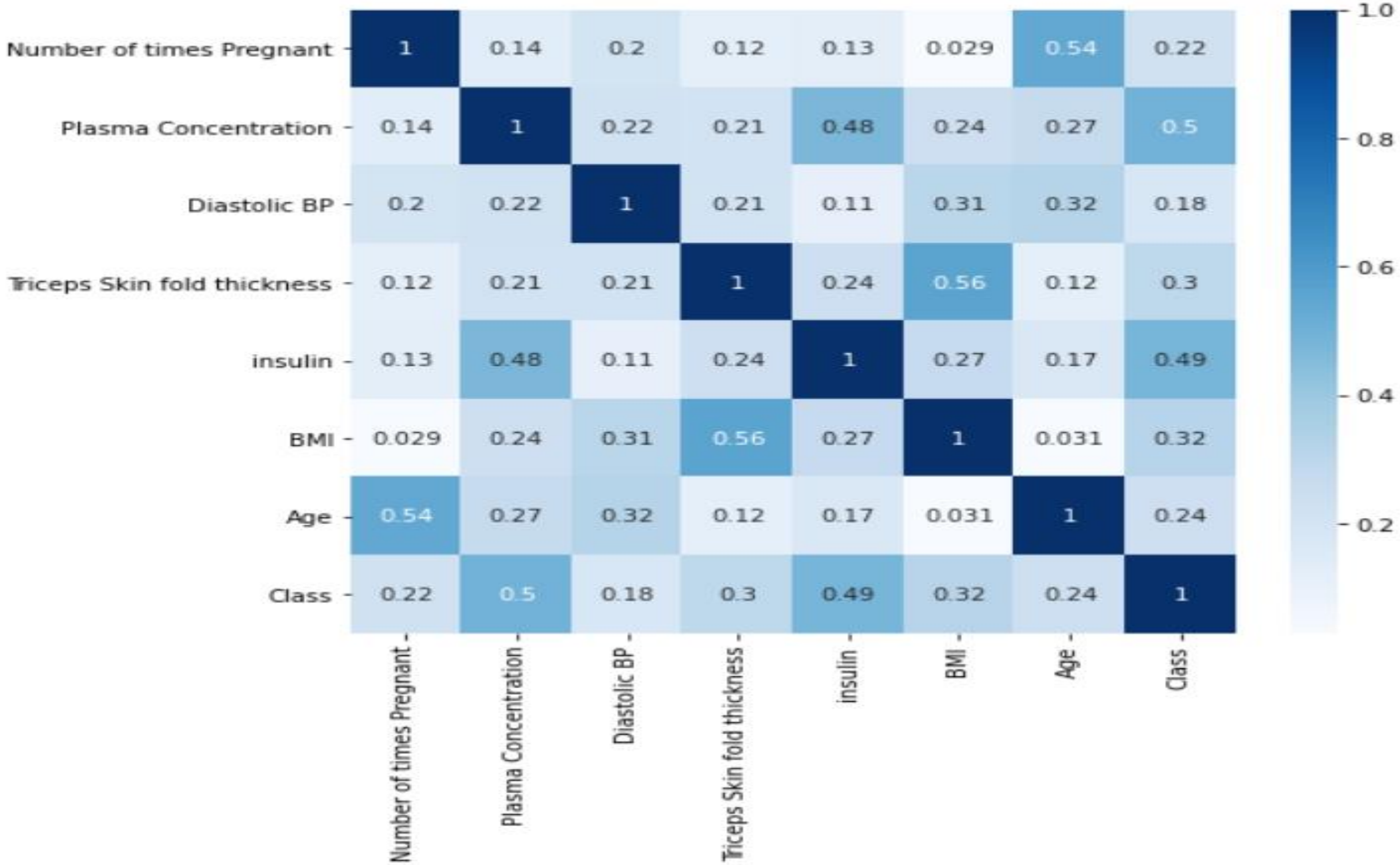
With this my final cleaned data was ready and I exported the data in an Excel file.



```
]: df.to_excel('Final.xlsx', index=False)
```

Analysis:

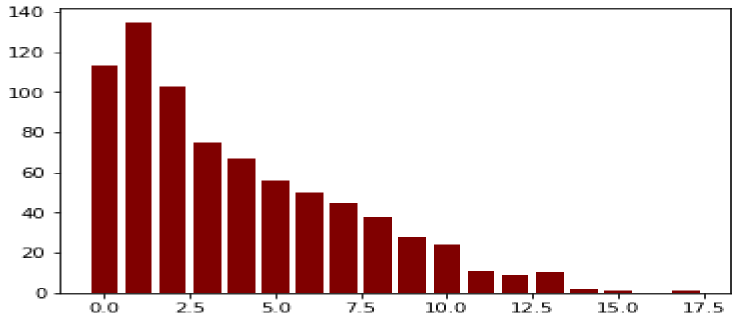
- Checking for the correlation between the features and I found that there is no correlation between any feature

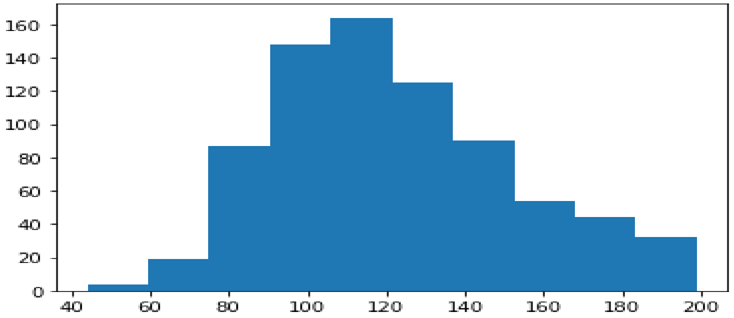


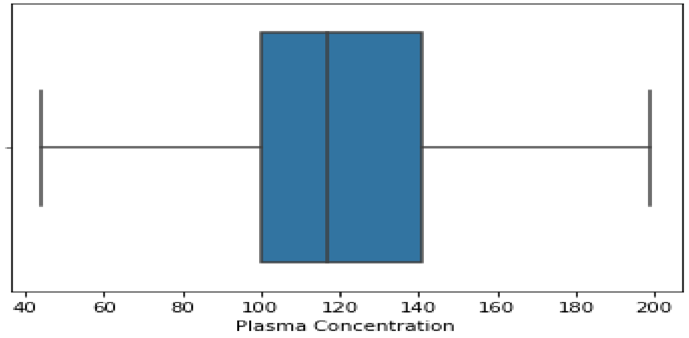
	Number of times Pregnant	Plasma Concentration	Diastolic BP	Triceps Skin fold thickness	insulin	BMI	Age	Class
Number of times Pregnant	1	0.14	0.2	0.12	0.13	0.029	0.54	0.22
Plasma Concentration	0.14	1	0.22	0.21	0.48	0.24	0.27	0.5
Diastolic BP	0.2	0.22	1	0.21	0.11	0.31	0.32	0.18
Triceps Skin fold thickness	0.12	0.21	0.21	1	0.24	0.56	0.12	0.3
insulin	0.13	0.48	0.11	0.24	1	0.27	0.17	0.49
BMI	0.029	0.24	0.31	0.56	0.27	1	0.031	0.32
Age	0.54	0.27	0.32	0.12	0.17	0.031	1	0.24
Class	0.22	0.5	0.18	0.3	0.49	0.32	0.24	1

Analysis:

- 50% of the women got pregnant 3 or less time and 25% of the women got pregnant over 6 times
- 34 women got pregnant over 10 times

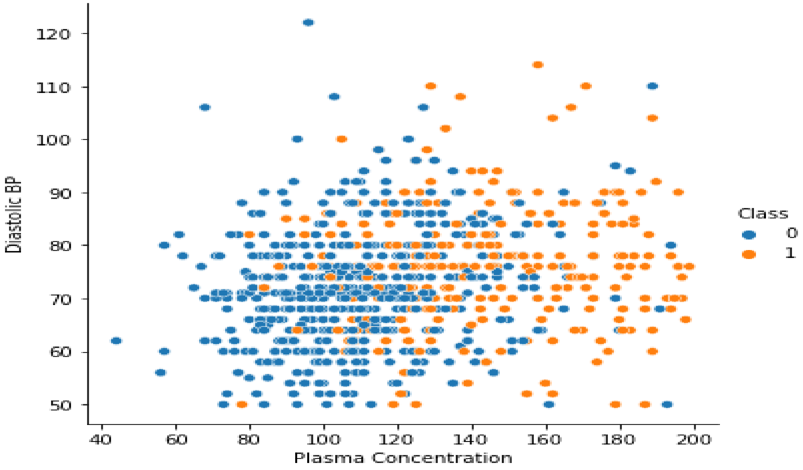






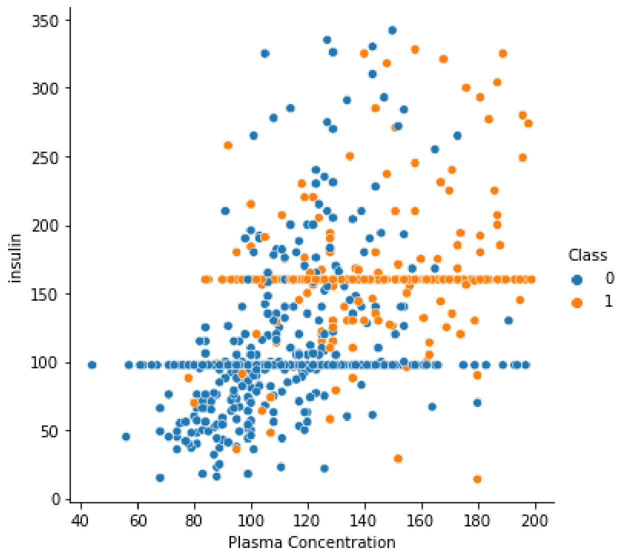
- More than 75% of the women has the plasma concentration between 80 and 160 and 50% of the patient have plasma concentration between 100 & 141.\

- Diastolic BP vs Plasma Concentration relation shows that
  - With higher Diastolic BP and High Plasma Concentration the chance of Diabetes Increases.
  - High Plasma Concentration has very high chances of getting Diabetes

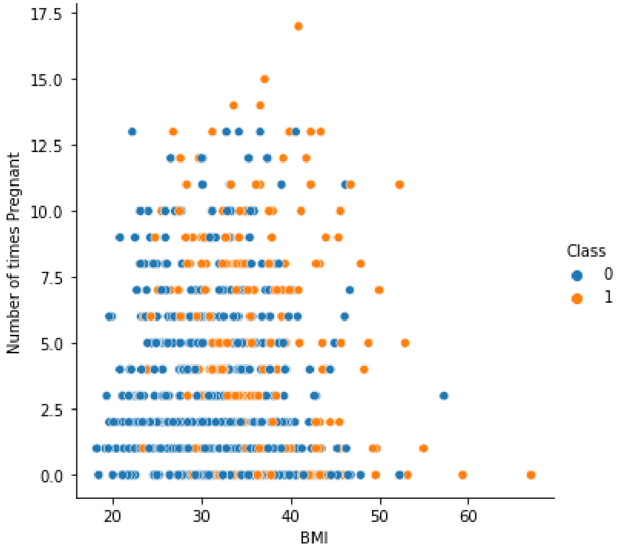


Analysis:

- Relation between Insulin and Plasma Concentration shows that :
  - People with Higher Insulin has high chances of Diabetes
  - Most of the patient with insulin over 150 has most of the Diabetes patients

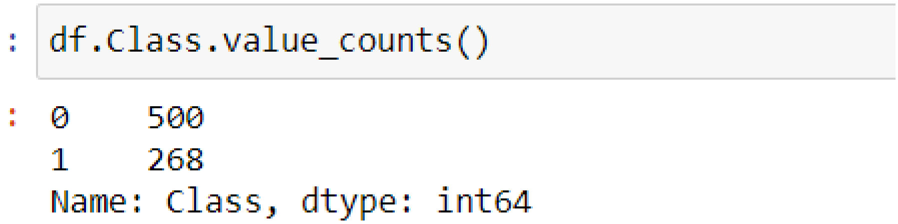


- Number of Pregnancies has no relation with he Diabetes
  - As the sample size is low for the pregnancies over 10 we can not say if the population can have same impact.
  - But as per the data with higher numbers of pregnancies patients do have Diabetes.

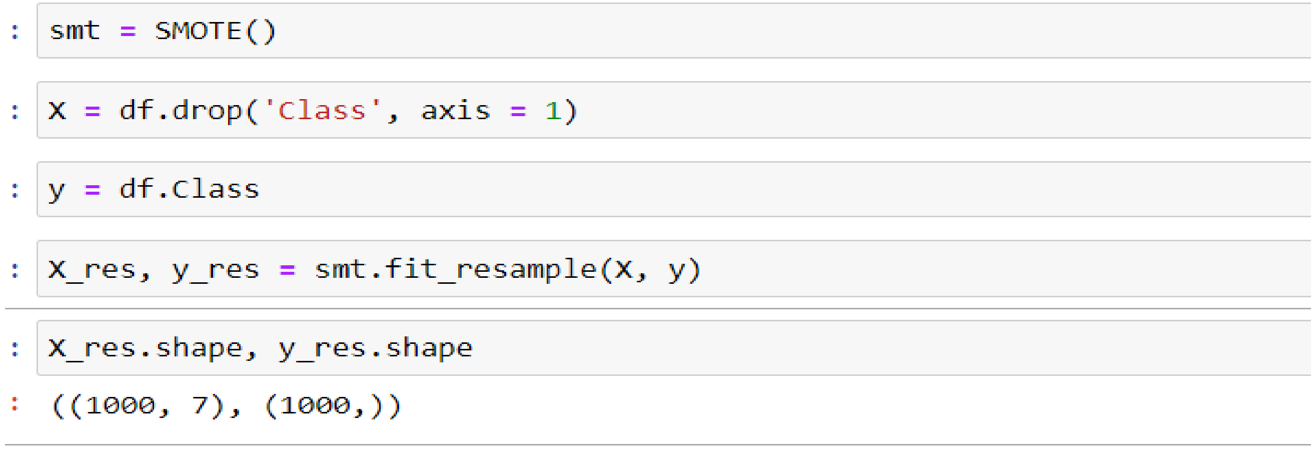


Predictive Model Building:

As this was a Imbalanced dataset I will prepare a Balances dataset using SMOTE by increasing the number of count values



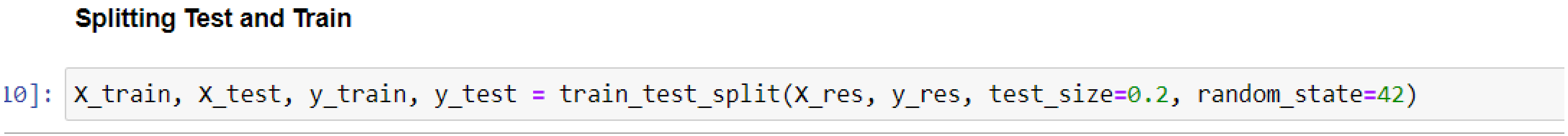
```
: df.Class.value_counts()

: 0    500
  1    268
  Name: Class, dtype: int64
```



```
: smt = SMOTE()

: X = df.drop('Class', axis = 1)

: y = df.Class

: X_res, y_res = smt.fit_resample(X, y)

: X_res.shape, y_res.shape

: ((1000, 7), (1000,))
```

Predictive Model Building:

Splitting the dataset into Test and Train with 80% training data and 20% Testing data.



Splitting Test and Train

```
10]: X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)
```

I have trained Multiple models:

- KNN
- Logistic Regression
- Decision Tree
- Random Forest

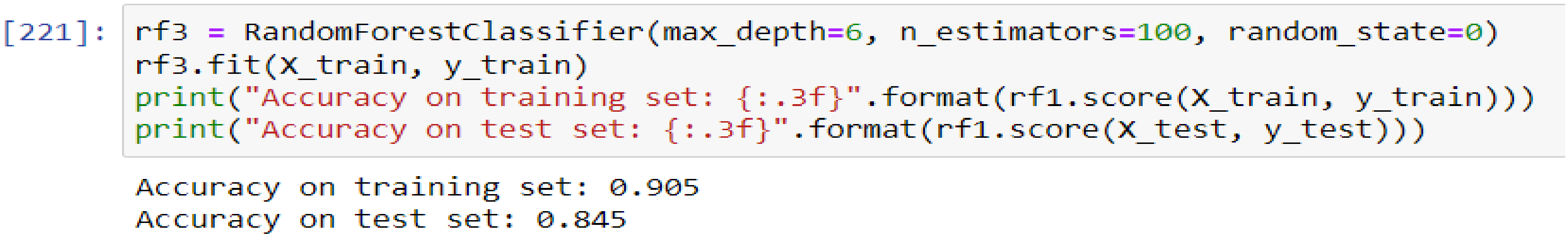
Predictive Model Building:

I have trained all the model with the training dataset and validated the data using the Testing dataset

For random forest and Decision Tree I have done a small hyperparameter optimization and selected the model with the best accuracy.

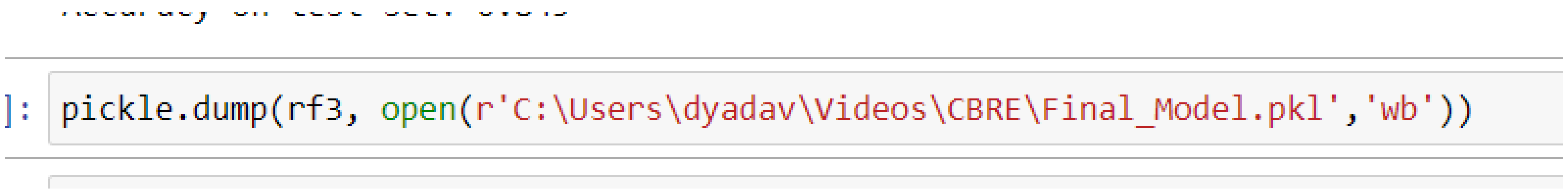
At the end after training all the models and Validating all the models I finally decided Random Forest as the best model as it gave me the best of the accuracy with the training as well as Testing data.

Also there was a good trade off between Bias and Variance.



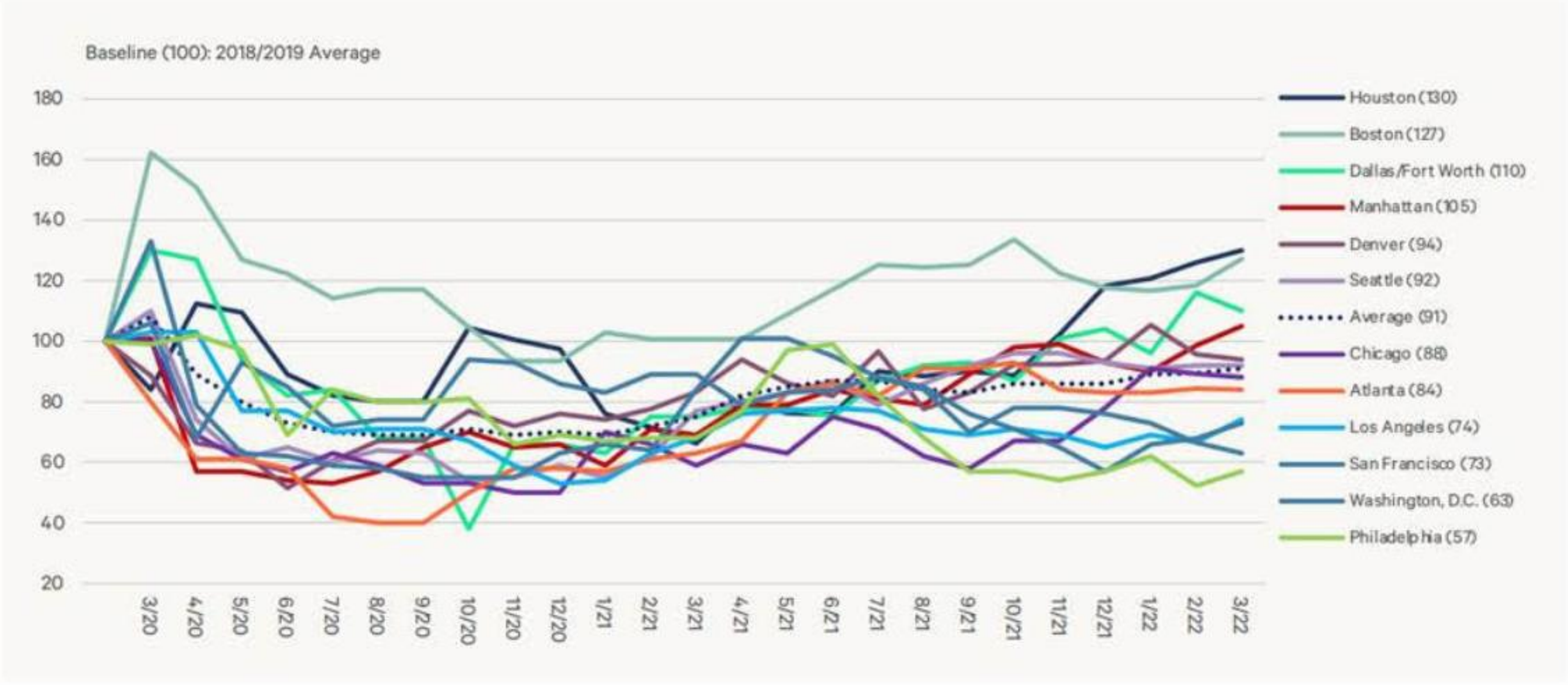
```
[221]: rf3 = RandomForestClassifier(max_depth=6, n_estimators=100, random_state=0)
       rf3.fit(X_train, y_train)
       print("Accuracy on training set: {:.3f}".format(rf1.score(X_train, y_train)))
       print("Accuracy on test set: {:.3f}".format(rf1.score(X_test, y_test)))

       Accuracy on training set: 0.905
       Accuracy on test set: 0.845
```

Finally I saved the model using Pickle for future Use.



```
]: pickle.dump(rf3, open(r'C:\Users\dyadav\Videos\CBRE\Final_Model.pkl','wb'))
```

Q2. This Viz is annoying and messy. Are there better ways to visualize it?



Q2. This Viz is annoying and messy. Are there better ways to visualize it?

Yes there could be multiple ways to visualize this chart:

1. As this line chart shows the comparison for 12 entities over the period of 25 Months we can add as slicer for filtering the entities with the entity Name.
    1. We can compare 2 or 3 entities at a time
    2. We can even select 1 for validating its performance over the period
2. Secondly we can add a slicer with the greater than or less than the average value and based on that we can filter out and compare the performance of entities below the average and above the average.
3. We can also use the sparklines for each entity for the visual comparison of these entities.
4. Or we can use multiple lines placed parallel side by side in the 4*3 matrix.
5. We can also use the table format and highlight the table based on their values over the period of time, like higher value will be darker and lower value will be lighter.

All the supporting files has been attached with the email.

Feel free to ask any question.

Thank You