

Doctor's Consultation Fees Prediction

Submitted by:

Vivek Singh

Batch NO:1827

**ACKNOWLEDGMENT**

Through this project I would like to Acknowledge my view towards The Data Trained Institute for their continuous support and Guidance which Give me a Confidence to complete the Assigned Project.

Through this I would like to thanks each and every Members of Data Trained Institute in completion of the project.

**INTRODUCTION**

Doctor's Consultation Fees Prediction

We all have been in situation where we have to go to doctor in emergency for treatment.

Its very difficult to analyse the consultation fee which is mostly higher.

As a data scientist we can do better if we have records like Qualification, experience. On the basis of which we can build a model. and predict the consulting charges..

**Data Description**:

Size of the training data : 5961 records

and testing dta:1987 records.

* Importing the required libraries:

import pandas as pd

import numpy as np

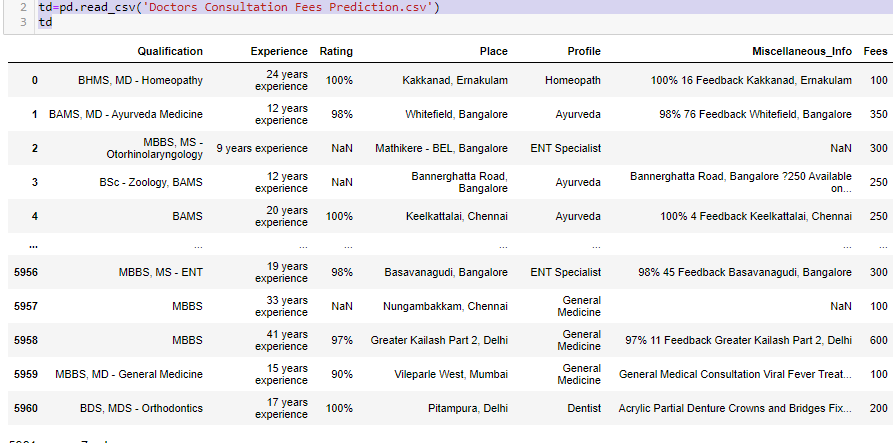
import seaborn as sns

import matplotlib.pyplot as plt

* Loading the train data set:

td=pd.read\_csv('Doctors Consultation Fees Prediction.csv')

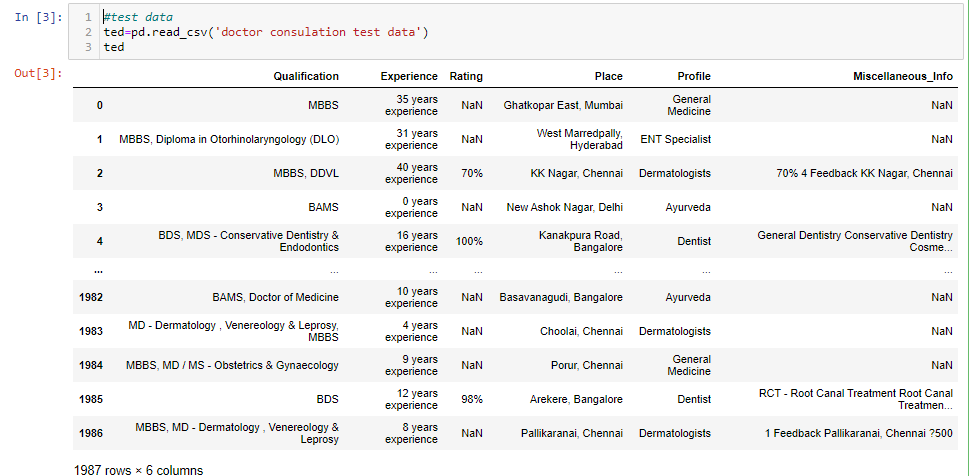
td



* we can clearly see our data set is consist of 5961 rows × 7 columns
* **Loading the test data**

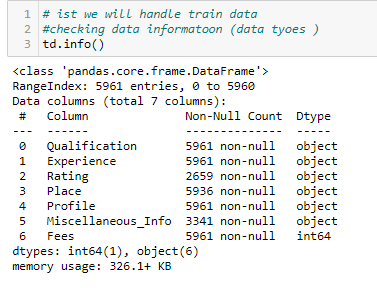
ted=pd.read\_csv('doctor consulation test data')

ted

****

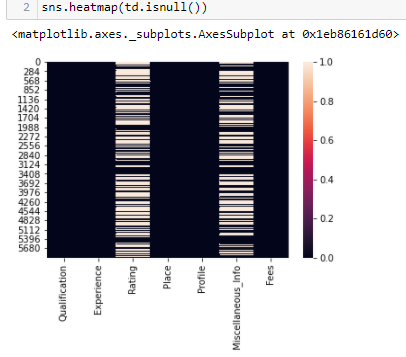
* we can clearly see our data set is consist of 1987 rows × 6columns

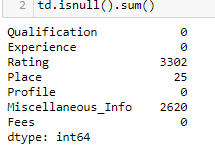
* **next we will check train data details**

****

from above we can clearly see the and its information either it is having null values or not and checking the data types its Object ,float or Int type data

* **Checking the null values of the columns**

****

****

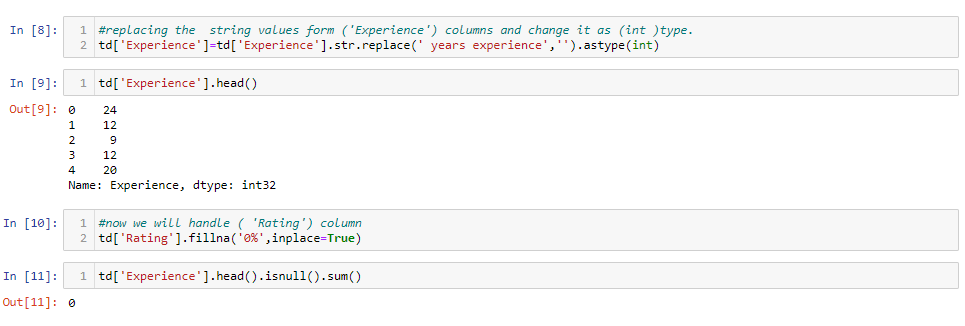
Observation:

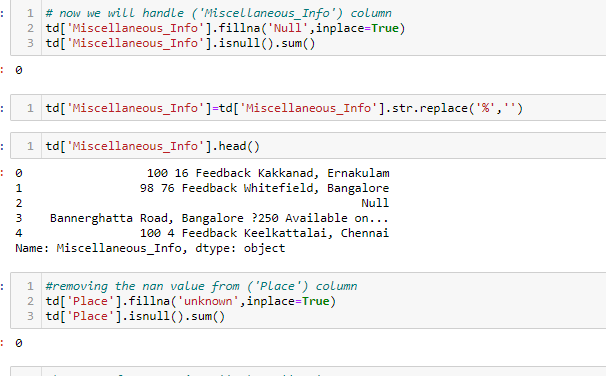
From above we clearly see Rating have 3302 null values

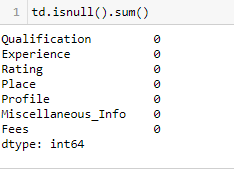
Place having 25 null values

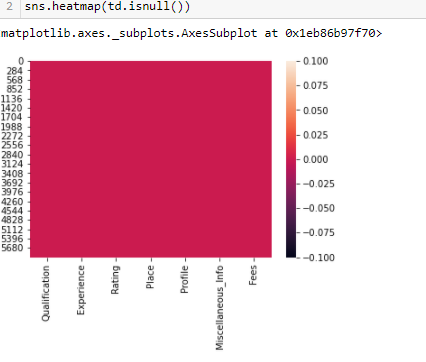
Miscellaneous info having 2620 null values

* Here we can clearly see our most of the columns are actually int in nature but there are sum marks like ‘,% etc.





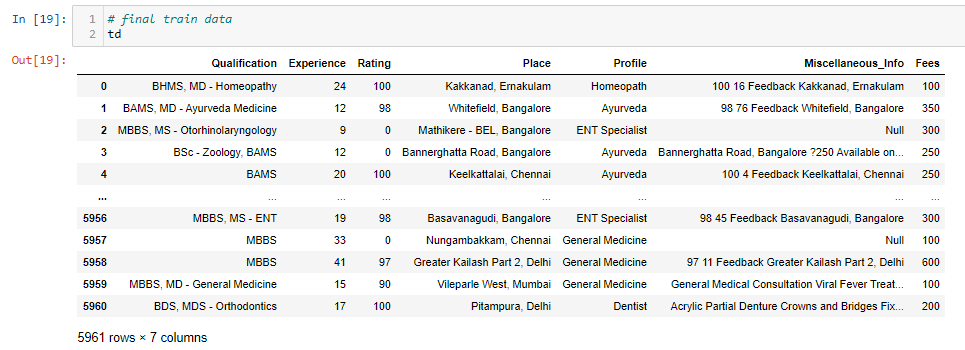


****

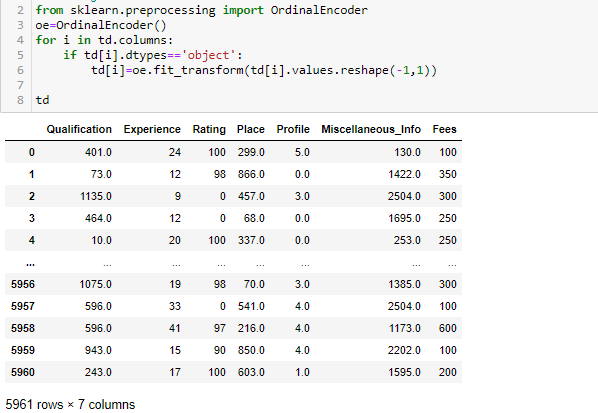
**Observations:**

* Here we have change the un required values like %, , etc from our columns.
* Changes the columns nature which is nonobject types in nature
* Last we can clearly see heat map with zero null values

**Final Train data :**

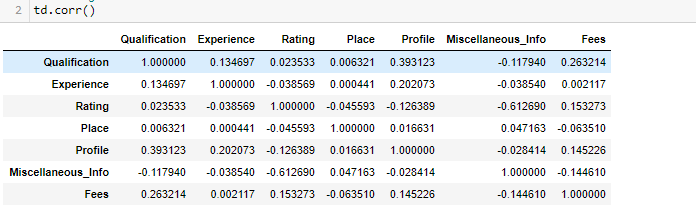
****

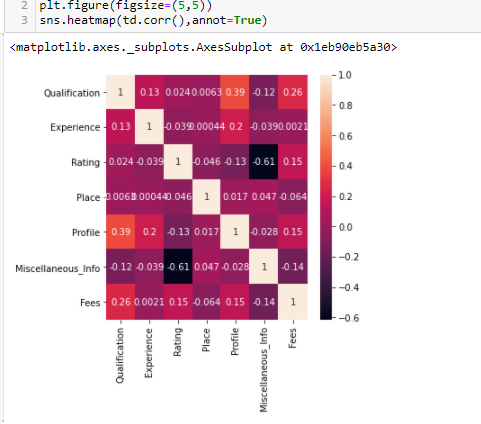
* Above we can see final train data after handling the null values and all unrequired like % , ‘ etc.
* **Encoding the all object data into numeric form**

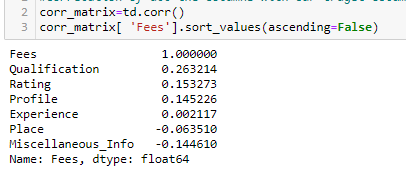


**OBSERVATION:** From we can clearly see that our object columns are changed into numeric form.

* **Checking the Correlation of all the data :**

****





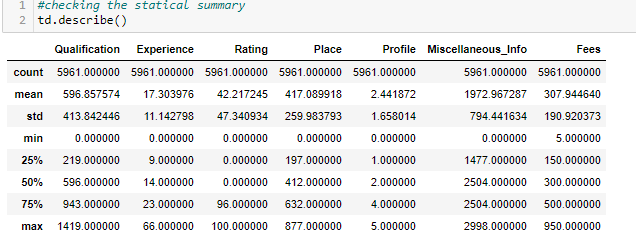
**Observation:**

* from above we can clearly see how the data correlated with each other some of positively correlated and some are negatively correlated.
* We can also find out the correlation of every columns with others remaining column via using correlation matrix
* From above we can clearly see the correlation of the column ‘Fees’ with the remaining columns
* ‘Fees’ is having highly correlated with Qualification

* ‘Fees’ Least correlated with Miscellaneous\_info

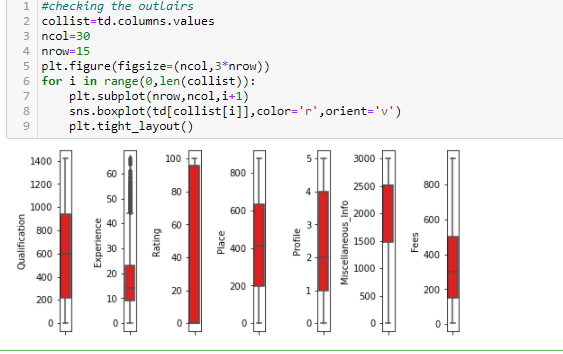
* **Statistical Summary:**

in this method we well see the statistical summary of each columns of our data set



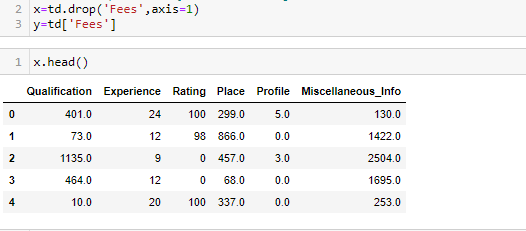
* From above we can easily analyse the out lairs skewness, mean, median, Standard deviation of our data set.

**Out lairs Detection :**



From above figure we can clear see there are no outliars in our data set except columns  {‘Experience}

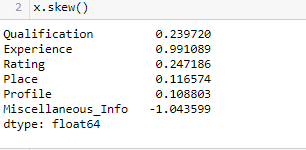
* **Features and Target data Split :**
* In the the case of dta split we will sepreate the feature and target data.
* We can mainly do splitting by using locking , method or column dropping.



**Observation:**

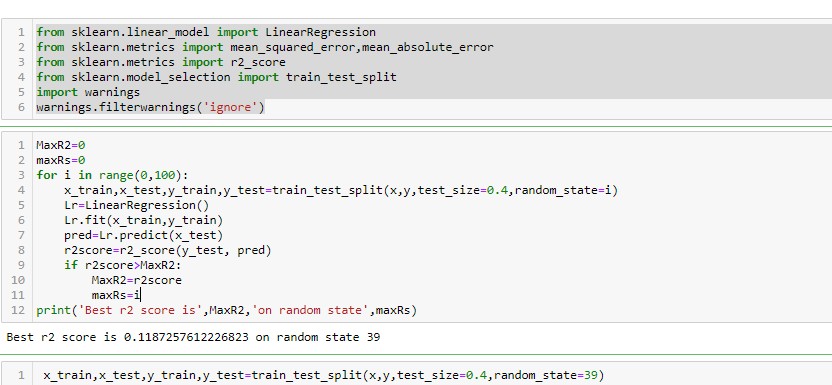
Here for feature data selection we dropped the target columns. And we can easily split our feature and target data which his denoted by x and y.

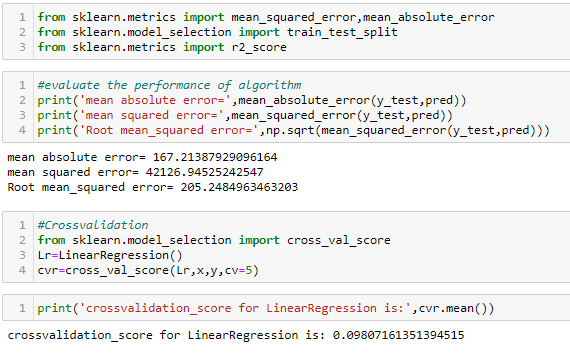
* **Skewness:** before training and testing our data set we need to check the skewness of our data set.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

* Here we will check the skewness value of the our x data set i.e(±0.5)
* Here we can clearly see most the values of numerical data type lies in range i.e(±0.5) except Experience columns
* Before training we must have to check both our x and y data must be into 2D from. we change data into 2d form by using **reshape method** .
* As we can see our target variable is continuous type so we will use LinearRegression.
* **Loading the model & Best random state selection:**

****

****

**Observation:**

* From we can clearly see best random state is 0.4 because r2\_score is 0.11
* Cv score is also low 0.098
* **Now we will train our data in other model and check its r2\_score** :
* **Ensemble method**



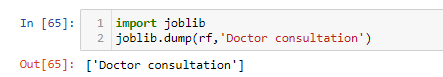
* **Observation:**

Here the Diffrence between r2score and cv score is almost about zero in case of RandomforestRegressor. So we will consider this as best model.

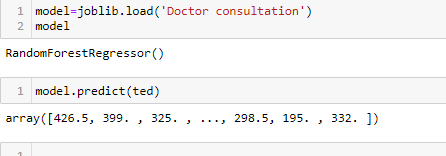
* **Saving the best model:**

After the training selection of the best model we have to save it for predicting the values .

We can save our model by using pickle **pickle or joblib**



* From above we can clearly see joblib method the save the model.
* We can recall it for prediction the values.
* **Now we will handle the test data:**
* We will follow the same steps as we have followed earlier in our train data .
* After all the data cleasing process we will recall our saved model and will predict the test data.
* **Loading the saved model :**



**Observation:**

After all the cleansing process removing the null values   
 etc we Load our saved model and predict our test data

Predicted result is shown in above fig.

Thank you.