**Doctor's Consultation Fees Prediction**

Introduction

Doctor consultation is an integral part of human life, and in this situation of COVID19 the world has become more proactive and cautious regarding individual health.

We are having a data from various geographical location of India, where we are trying to simplify the data to make a Machine learning algorithm for doctor’s fees prediction, we have various factors to study upon like qualification, experience, city, locality etc. Data set contains below mentioned features:

Qualification: Qualification and degrees held by the doctor

Experience: Experience of the doctor in number of years

Rating: Rating given by patients

Profile: Type of the doctor

Miscellaneous\_Info: Extra information about the doctor

Fees: Fees charged by the doctor (Target Variable)

Place: Area and the city where the doctor is located.

The data set have two files the train (5961 records) and test (1987 records).

Problem statement

Predict Doctor’s consultation fees.

Libraries

import numpy as np

import pandas as pd

import sklearn

import seaborn as sns

import matplotlib.pyplot as plt

sns.set()

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.metrics import r2\_score, mean\_squared\_error

from sklearn.metrics import mean\_absolute\_error

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear\_model import Lasso,Ridge

from sklearn.linear\_model import Lasso

from sklearn.model\_selection import GridSearchCV

import joblib

Analysis

We have the data available with us, lets analyse it to find what all information we can fetch.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5961 entries, 0 to 5960

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Qualification 5961 non-null object

1 Experience 5961 non-null object

2 Rating 2659 non-null object

3 Place 5936 non-null object

4 Profile 5961 non-null object

5 Miscellaneous\_Info 3341 non-null object

6 Fees 5961 non-null int64

dtypes: int64(1), object(6)

memory usage: 326.1+ KB

We get the information regarding the amount, type of data present in data set for train file we have 5961 entries in all and the data type is object and Fees column have integer data type. number of columns are 7.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1987 entries, 0 to 1986

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Qualification 1987 non-null object

1 Experience 1987 non-null object

2 Rating 897 non-null object

3 Place 1981 non-null object

4 Profile 1987 non-null object

5 Miscellaneous\_Info 1153 non-null object

dtypes: object(6)

memory usage: 93.3+ KB

We get the information regarding the amount, type of data present in data set For Test file we have 1987 entries in all and the data type is object and Fees column have integer data type. The number of columns are 6.

Out[14]:

Qualification 0

Experience 0

Rating 3302

Place 25

Profile 0

Miscellaneous\_Info 2620

Fees 0

dtype: int64

Qualification 0

Experience 0

Rating 1090

Place 6

Profile 0

Miscellaneous\_Info 834

dtype: int64

We are checking for null values if any and we have found the null values in train as well as test data. We have missing values in column of Rating, place and miscellaneous on test and train files.

9 years experience 323

10 years experience 294

11 years experience 288

8 years experience 282

12 years experience 279

...

66 years experience 1

58 years experience 1

65 years experience 1

61 years experience 1

59 years experience 1

Name: Experience, Length: 64, dtype: int64

We observe that very few doctors are having experience above 50 years, majority of doctors are having experience above 5 years.

[{"metadata":{"trusted":false},"cell\_type":"code","source":"test['Experience'].value\_counts()","execution\_count":20,"outputs":[{"data":{"text/plain":"10 110\n8 98\n9 96\n19 92\n7 89\n11 86\n13 83\n14 81\n12 79\n17 72\n18 70\n6 62\n15 57\n16 51\n5 50\n23 49\n21 48\n20 48\n29 46\n4 44\n22 37\n24 34\n3 33\n25 33\n26 32\n0 31\n28 29\n27 27\n34 27\n36 27\n31 26\n38 23\n30 18\n32 16\n44 16\n40 15\n37 14\n39 13\n35 13\n41 12\n33 11\n2 11\n46 11\n42 10\n43 10\n47 9\n49 8\n45 7\n48 6\n59 4\n50 3\n51 3\n52 2\n54 2\n53 1\n56 1\n65 1\nName: Experience, dtype: int64"},"execution\_count":20,"metadata":{},"output\_type":"execute\_result"}]}]

Out[16]:

BDS 488

BHMS 477

BAMS 471

MBBS 334

MBBS, MS - ENT 220

...

DVD, MD - Dermatology 1

BAMS, MS - Ayurvedic General Surgery, Ayurvedic panchkarma 1

MDS - Oral & Maxillofacial Surgery, BDS, Dip' NMD 1

BDS, MDS-Oral Pathology and Oral Microbiology, Certificate in Dental Implants 1

MBBS, Diploma in Dermatology, Diploma in Public Health 1

Name: Qualification, Length: 1420, dtype: int64

We understand that there are multiple backgrounds of education and qualification that the Doctors have in our data set. MBBS, BHMS, MD, Dermatology, BDS, MS etc are the qualifications.

10 110

8 98

9 96

19 92

7 89

11 86

13 83

14 81

12 79

17 72

18 70

6 62

15 57

16 51

5 50

23 49

21 48

20 48

29 46

4 44

22 37

24 34

3 33

25 33

26 32

0 31

28 29

27 27

34 27

36 27

31 26

38 23

30 18

32 16

44 16

40 15

37 14

39 13

35 13

41 12

33 11

2 11

46 11

42 10

43 10

47 9

49 8

45 7

48 6

59 4

50 3

51 3

52 2

54 2

53 1

56 1

65 1

Name: Experience, dtype: int64

100% 684

98% 290

99% 259

97% 241

96% 220

95% 178

94% 115

93% 109

90% 66

92% 66

91% 60

89% 42

88% 41

85% 27

82% 23

86% 21

83% 21

80% 19

77% 16

87% 14

79% 13

84% 13

81% 12

67% 12

76% 9

71% 9

75% 8

73% 8

74% 7

60% 7

78% 6

68% 5

56% 5

70% 4

69% 4

72% 3

57% 3

62% 2

55% 2

33% 2

63% 2

64% 2

36% 1

58% 1

47% 1

48% 1

53% 1

40% 1

45% 1

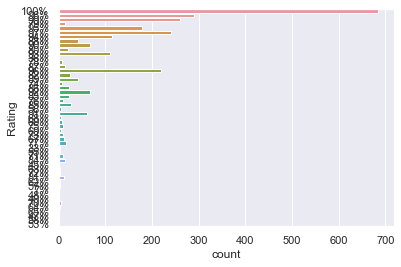
65% 1

7% 1

Name: Rating, dtype: int64

Above are the Rating count of the doctor. These rating are provided by the patients.

<AxesSubplot:xlabel='count', ylabel='Rating'>



We have the bar plot for ratings.

train["Place"] = train["Place"].str.split(",")

train["City"] = train["Place"].str[-1]

train["Place"] = train["Place"].str[0]

test["Place"] = test["Place"].str.split(",")

test["City"] = test["Place"].str[-1]

test["Place"] = test["Place"].str[0]

We are splitting the place into city, this will help us to segregate the place according to city. Here we observe that the data of Place column too is huge and not arranged properly hence we have tried to extract data of City rather than fetching the data of localities within the city.

Bangalore 1258

Mumbai 1219

Delhi 1185

Hyderabad 951

Chennai 855

Coimbatore 228

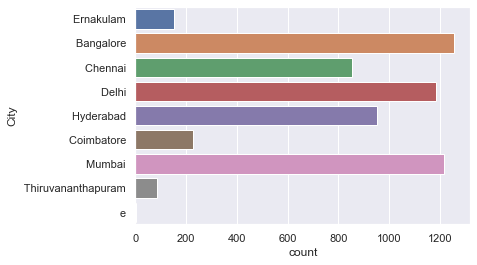
Ernakulam 153

Thiruvananthapuram 86

e 1

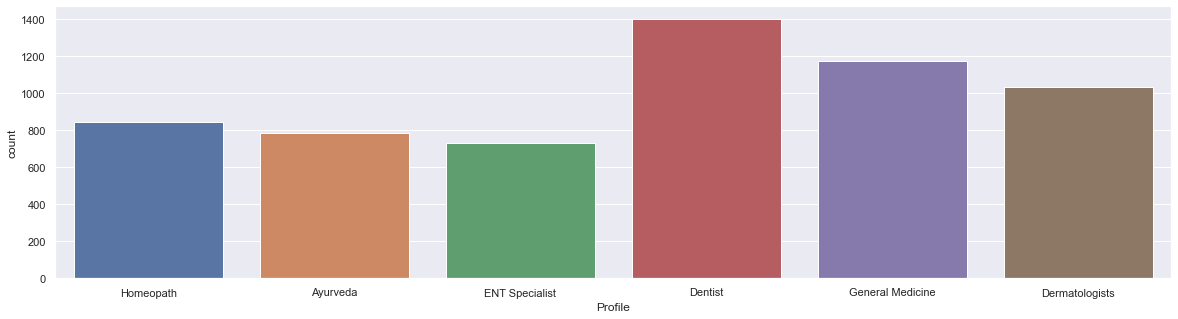
Name: City, dtype: int64

<AxesSubplot:xlabel='count', ylabel='City'>



Above is the bar graph of city.

<AxesSubplot:xlabel='Profile', ylabel='count'>

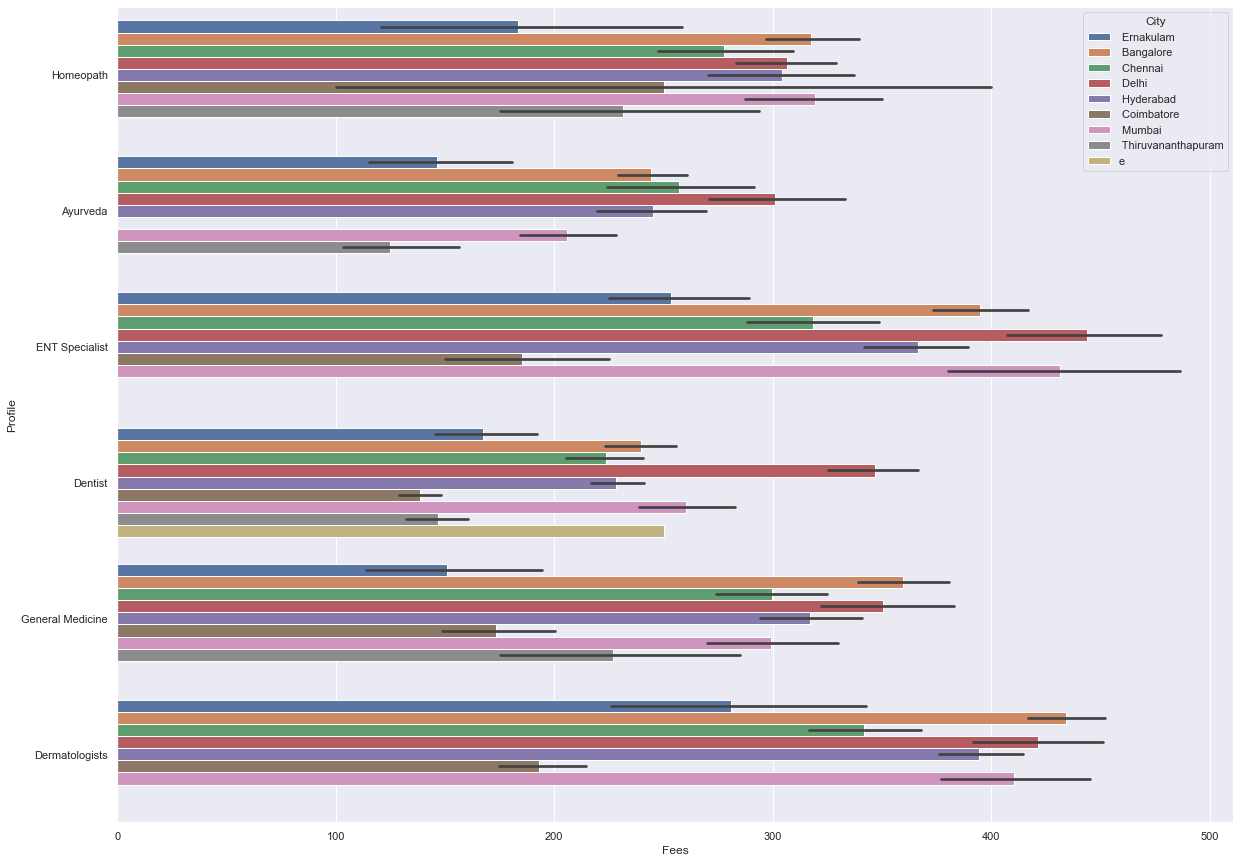


From above graph we observe that Dentist has the highest count followed by General medicine, Dermatologists, Homeopath, Ayurveda and last is ENT specialist.

plt.figure(figsize=(20,15))

sns.barplot('Fees','Profile',hue='City',data=train)

<AxesSubplot:xlabel='Fees', ylabel='Profile'>



From above graph we understand the relationship between the fees, city and Profile. We observe that the fees of ENT and dermatologist is high in cities of Bangalore, Mumbai and Delhi While Delhi is having highest fees in all the profiles.

def remove\_punctuation(x):

try:

x=x.str.replace('[^\w\s]','')

except:

pass

return x

test.apply(remove\_punctuation)

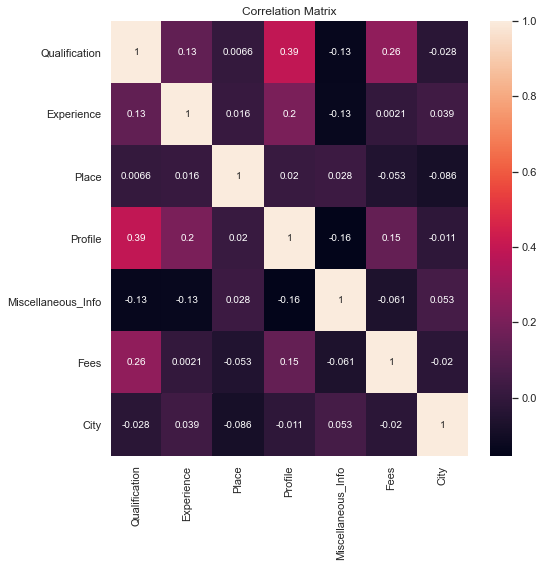
Using the command above we are removing the punctuations and special characters from the data set.

| **Qualification** | **Experience** | **Rating** | **Place** | **Profile** | **Miscellaneous\_Info** | **City** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | MBBS | 35 | NaN | Ghatkopar East | General Medicine | NaN | Mumbai |
| **1** | MBBS Diploma in Otorhinolaryngology DLO | 31 | NaN | West Marredpally | ENT Specialist | NaN | Hyderabad |
| **2** | MBBS DDVL | 40 | 70 | KK Nagar | Dermatologists | 70 4 Feedback KK Nagar Chennai | Chennai |
| **3** | BAMS | 0 | NaN | New Ashok Nagar | Ayurveda | NaN | Delhi |
| **4** | BDS MDS Conservative Dentistry Endodontics | 16 | 100 | Kanakpura Road | Dentist | General Dentistry Conservative Dentistry Cosme... | Bangalore |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1982** | BAMS Doctor of Medicine | 10 | NaN | Basavanagudi | Ayurveda | NaN | Bangalore |
| **1983** | MD Dermatology Venereology Leprosy MBBS | 4 | NaN | Choolai | Dermatologists | NaN | Chennai |
| **1984** | MBBS MD MS Obstetrics Gynaecology | 9 | NaN | Porur | General Medicine | NaN | Chennai |
| **1985** | BDS | 12 | 98 | Arekere | Dentist | RCT Root Canal Treatment Root Canal Treatment... | Bangalore |
| **1986** | MBBS MD Dermatology Venereology Leprosy | 8 | NaN | Pallikaranai | Dermatologists | 1 Feedback Pallikaranai Chennai 500 | Chennai |

1987 rows × 7 columns

Post removing the punctuations from data set we are performing the Label encoder process to convert the data to integer.

Now we will replace the null values with mode values.



We observe that the fees is having significant relationship with the features. We have the target defined that is fee hence we will use the Regression model in this data set.

LinearRegression

Mean absolute error: 147.90790961244474

r2 score is 0.09257073241264036

Root mean Squared Error: 182.75771343162717

Mean squared error is 33400.38181875676

RandomForestRegressor

RF Mean absolute error: 139.93958178284856

RF r2 score is 0.16010919811106328

RF Root mean Squared Error: 175.8250449054799

RF Mean squared error is 30914.446416014023

ls Mean absolute error: 139.93958178284856

ls r2 score is 0.16010919811106328

ls Root mean Squared Error: 175.8250449054799

ls Mean squared error is 30914.446416014023

From r2 values we conclude that LinearRegression is the best model.

Best Score: 0.09441119727132471

Best Params: {'alpha': 1}

We are getting best score of 0.0944 and best parameter is when alpha is 1.

Make sure to save the model using joblib.

Conclusion

Overall observation is that all the factors are critical and influence the fees of consultation. The Linear regression is the best model after comparing the r2 score and cross-validation difference with Random forest and lasso. Hence we are hyper parameter tunning the linear regression model and then saving the same.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*