MACHINE LEARNING

PROJECT REPORT

April 2024

Introduction:

This projects aim is to predict doctors fee against valuable influencing columns. This dataset is valuable for analyzing and understanding various factors influencing doctors fee, enabling researchers and analysts to develop models that accurately predict fee based on relevant input features.

Describing data

Description and illustration of of columns info **fig 1.1** & **fig 1.3**. Count of non null values **fig 1.2**. Count of unique values **fig 1.4** .

Doctor Name: *Doctors name*

City: *doctors city*

Specialization: *doctors specialization*

Doctor Qualification: *Doctors Qualifications*

Experience(Years): *explain the years of experience*

Total_Reviews: *Shows the total reviews of the doctor*

Patient Satisfaction Rate(%age): *the percentage of satisfaction of patient*

Avg Time to Patients(mins): *shows the patient average time in his session*

Wait Time(mins): *The time the patient waits to enter his reservation*

Hospital Address: *The addrsss of the hospital*

Doctors Link: *shows doctors link (his site)*

Fee(PKR): *shows the doctors fee*

fig 1.1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2386 entries, 0 to 2385
Data columns (total 12 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Doctor Name                      2386 non-null   object
 1   City                             2386 non-null   object
 2   Specialization                   2386 non-null   object
 3   Doctor Qualification             2386 non-null   object
 4   Experience(Years)                2386 non-null   float64
 5   Total_Reviews                    2386 non-null   int64
 6   Patient Satisfaction Rate(%age)  2386 non-null   int64
 7   Avg Time to Patients(mins)       2386 non-null   int64
 8   Wait Time(mins)                  2386 non-null   int64
 9   Hospital Address                 2386 non-null   object
 10  Doctors Link                     2386 non-null   object
 11  Fee(PKR)                         2386 non-null   int64
dtypes: float64(1), int64(5), object(6)
memory usage: 223.8+ KB
```

fig 1.2

	Experience(Years)	Total_Reviews	Patient Satisfaction Rate(%age)	Avg Time to Patients(mins)	Wait Time(mins)	Fee(PKR)
count	2386.000000	2386.000000	2386.000000	2386.000000	2386.000000	2386.000000
mean	11.844719	92.473177	96.657586	14.092205	11.264459	1201.016764
std	8.784449	282.162526	4.962300	2.722198	5.636885	831.760021
min	1.000000	0.000000	33.000000	3.000000	0.000000	0.000000
25%	6.000000	0.000000	94.000000	14.000000	10.000000	600.000000
50%	10.000000	8.000000	98.000000	14.000000	11.000000	1000.000000
75%	14.000000	54.000000	100.000000	15.000000	11.000000	1500.000000
max	53.000000	5147.000000	100.000000	50.000000	82.000000	10000.000000

fig 1.3

```
[ 6.  1. 11. 12. 15. 10. 33.  3.] ... etc
Unique categories in Total_Reviews: 361
[ 11   0   9  71 199  22   4] ... etc
Unique categories in Patient Satisfaction Rate(%age): 25
[100  94  96] ... etc
Unique categories in Avg Time to Patients(mins): 28
[19 14 10 18 16 15] ... etc
Unique categories in Wait Time(mins): 47
[ 6 11  0 10  2  9] ... etc
Unique categories in Fee(PKR): 48
[2000  500 1000  800 1500 2500] ... etc
------------------------------------------------------------
Unique categories in Doctor Name: 2190
Unique categories in City: 117
Unique categories in Specialization: 150
Unique categories in Doctor Qualification: 1041
Unique categories in Hospital Address: 1178
```

fig 1.4

Visualization of Bivariance & Univariance

Box plot **fig 2.1.** Histogram **fig 2.2.** Pair plot **fig 2.9.**

Scatter plot (years of experience against fees) **fig 2.3**. Scatter plot (average time of patients against fees) **fig 2.4**. Line plot Relationship between Avg Time to Patients and Wait Time **fig 2.5**. Pie chart patient satisfaction rate **fig 2.6.**

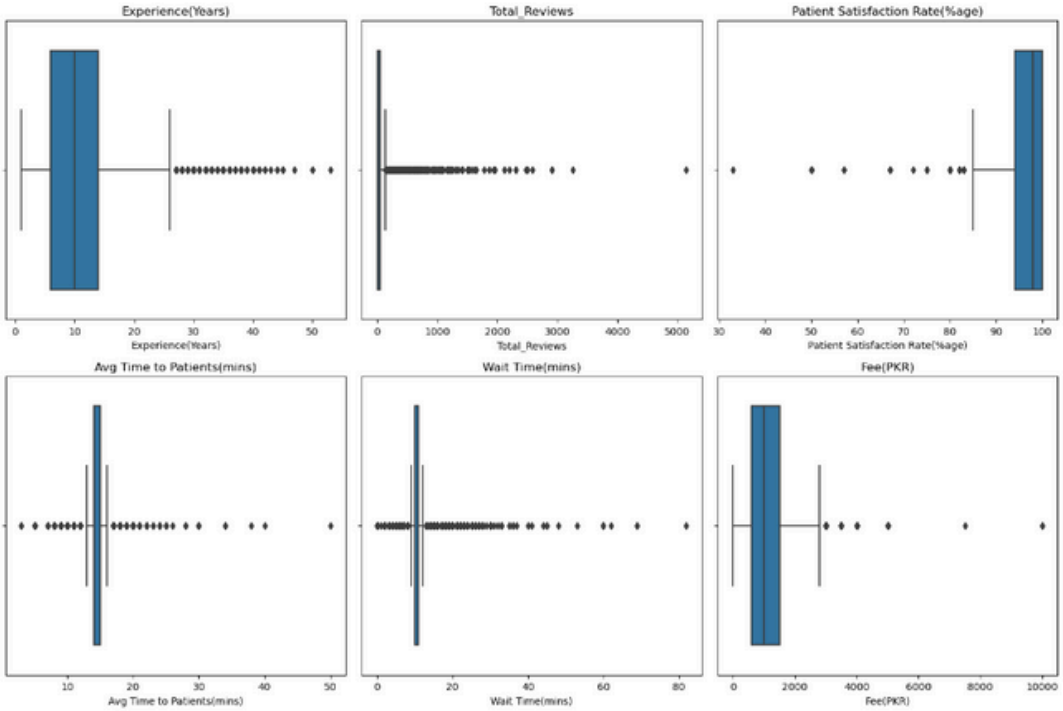

fig 2.1

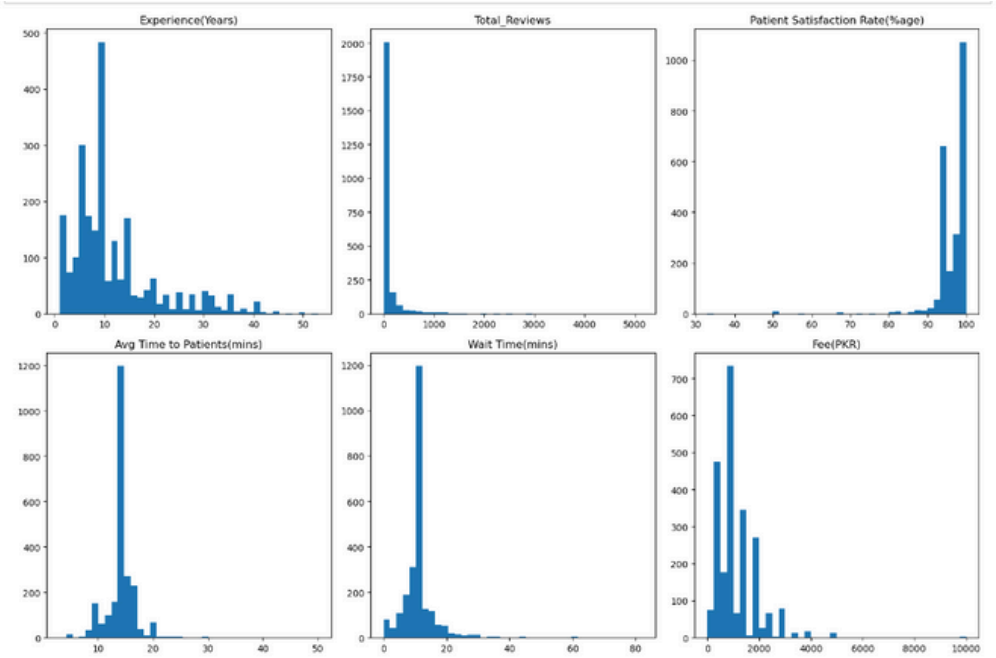

fig 2.2

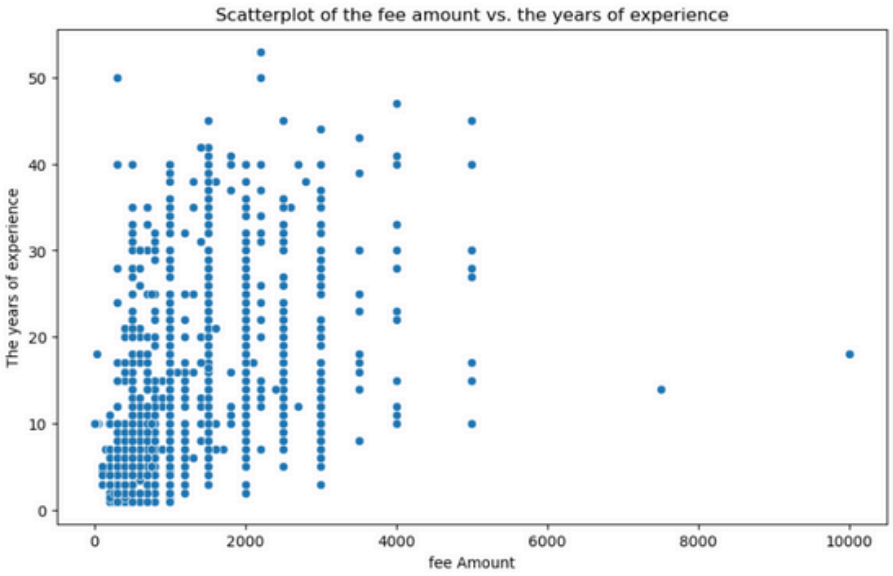

fig 2.3

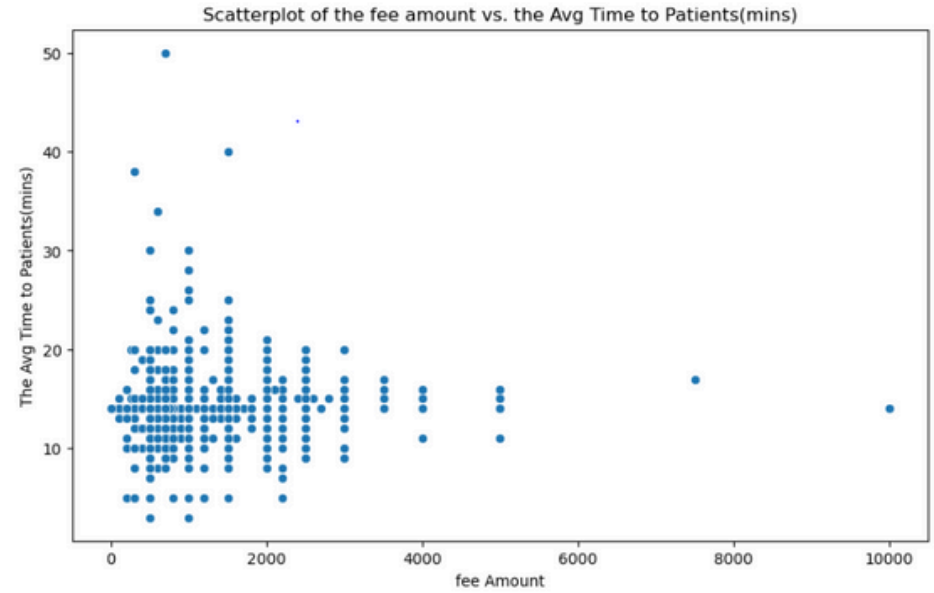

fig 2.4

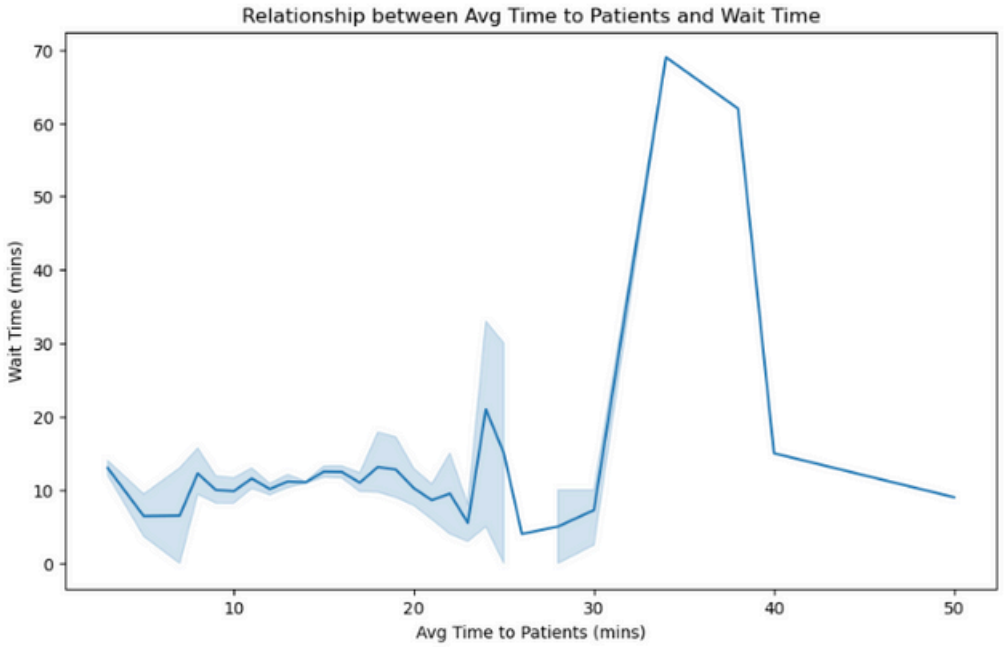

fig 2.5

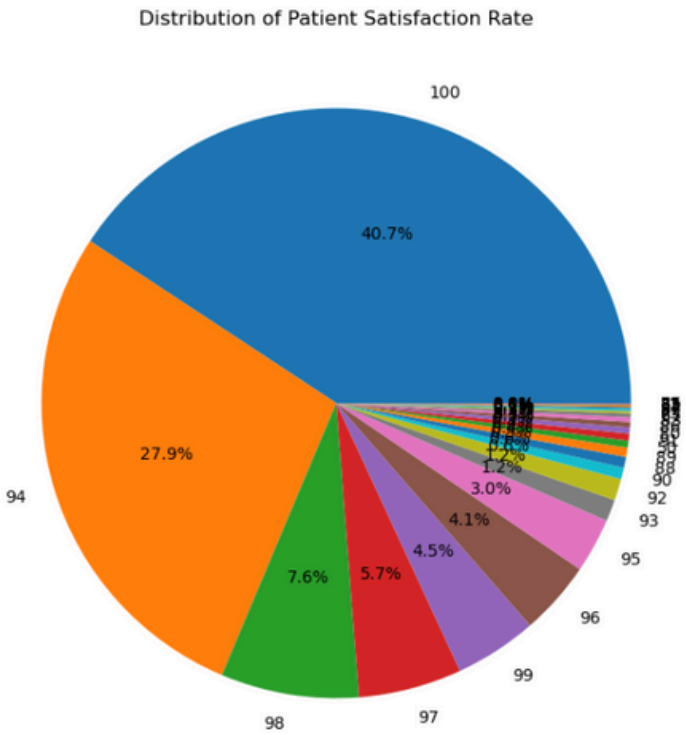

fig 2.6

Most Doctors are paid less than the average Fee. Furthermore, Qualifications and Degrees don't seem to have any effect on the Fee **fig 2.7.**

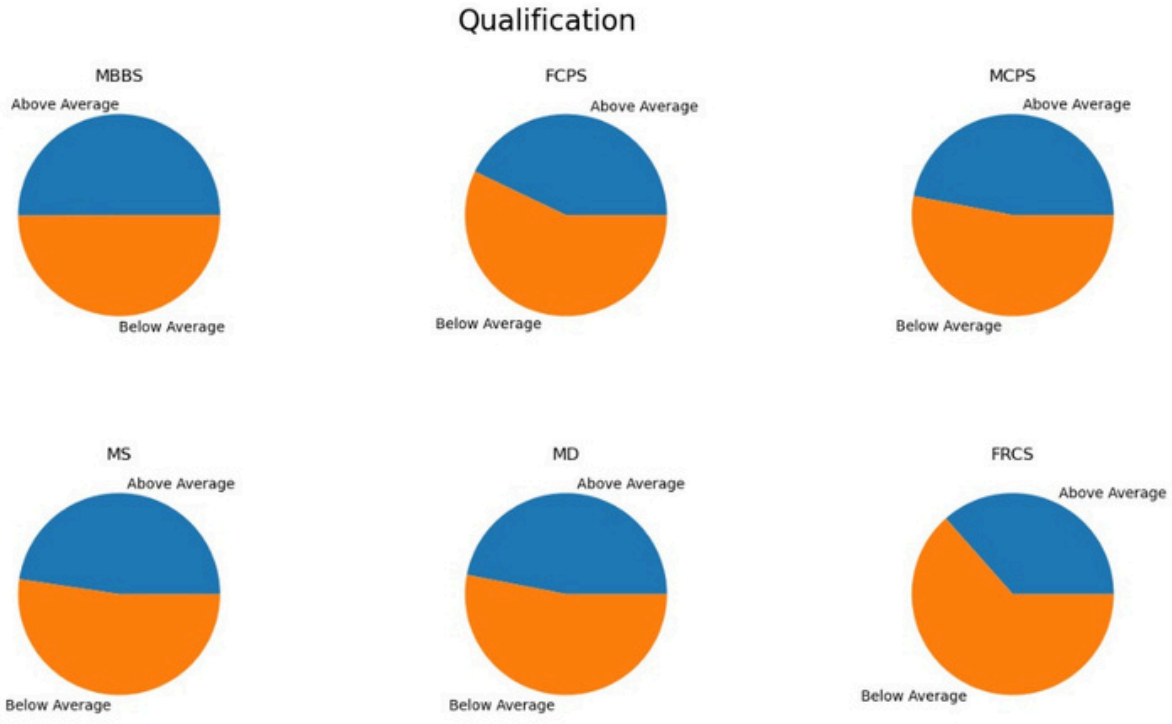

fig 2.7

Distribution plot with normalization, respectively average time to patients and wait time **fig 2.8.**

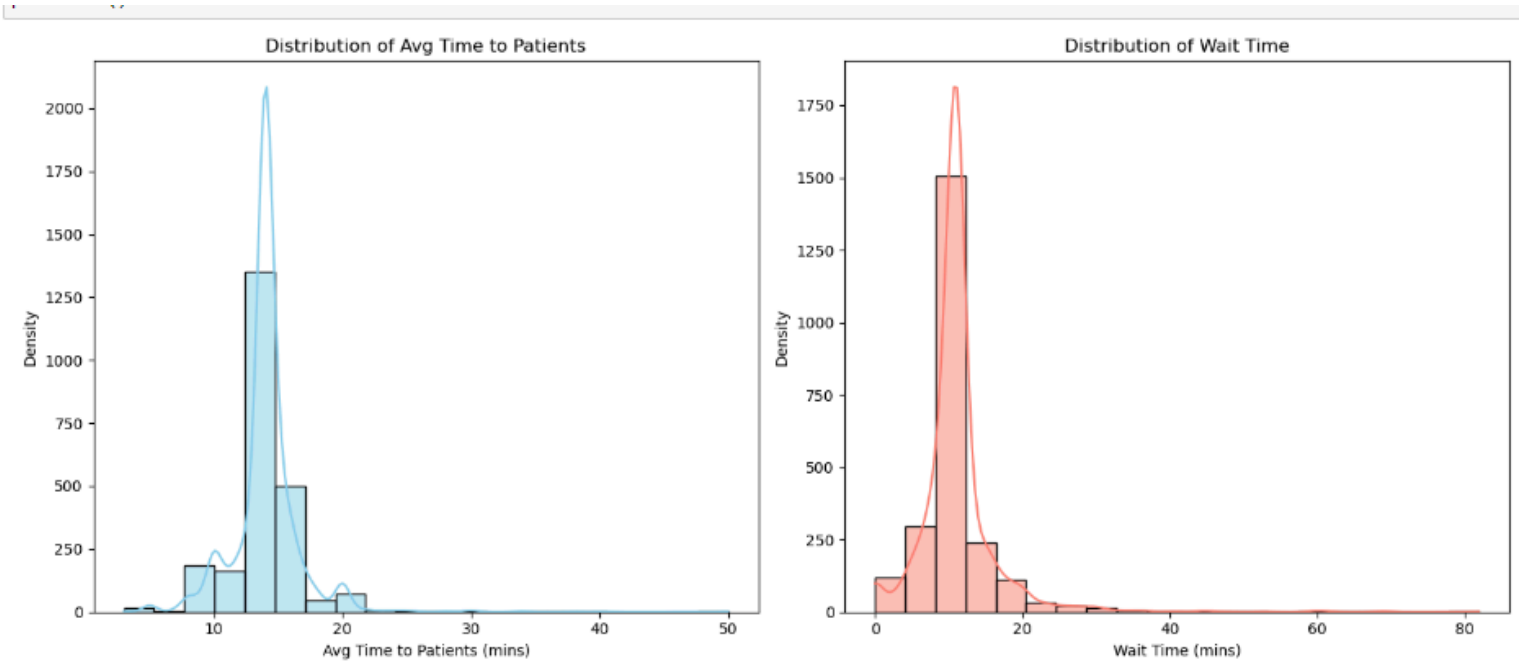

fig 2.8

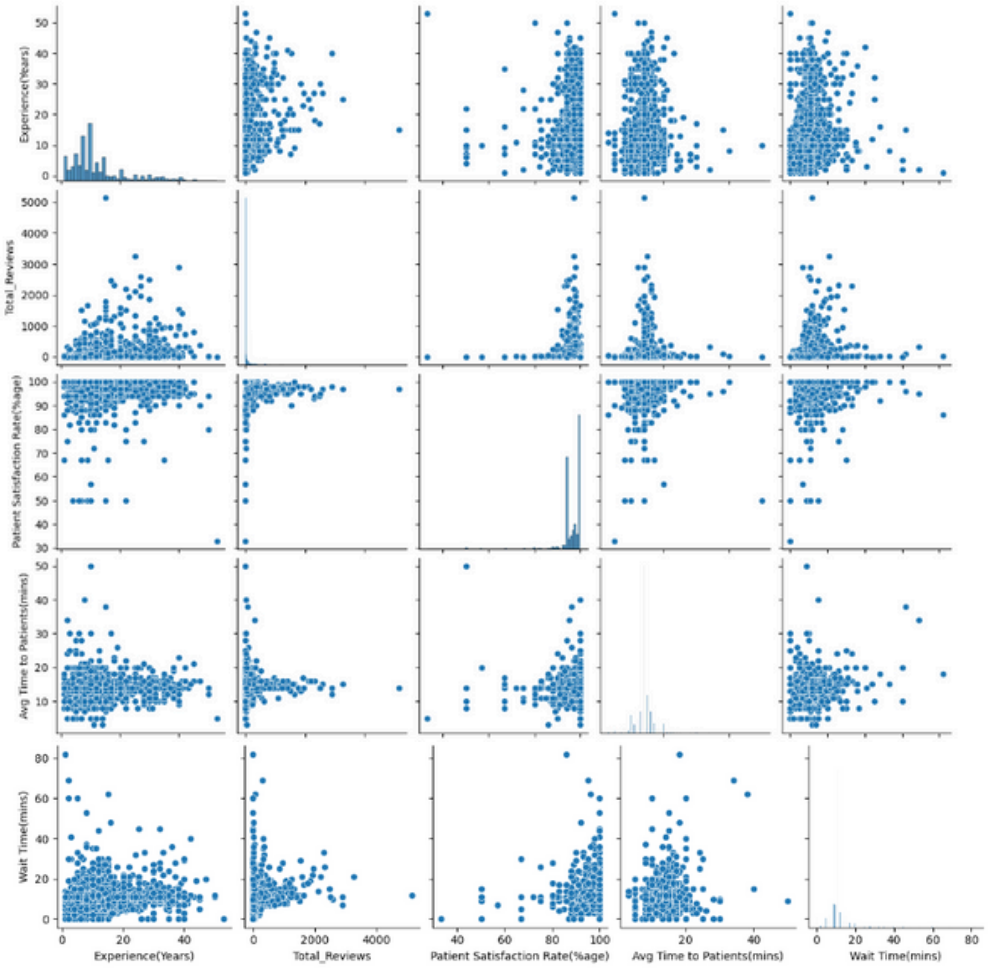

fig 2.9

Joint Plot between Avg Time to Patients and Wait Time **fig 2.10.**
Box Plot for patient Satisfaction rat**e fig 2.11**.

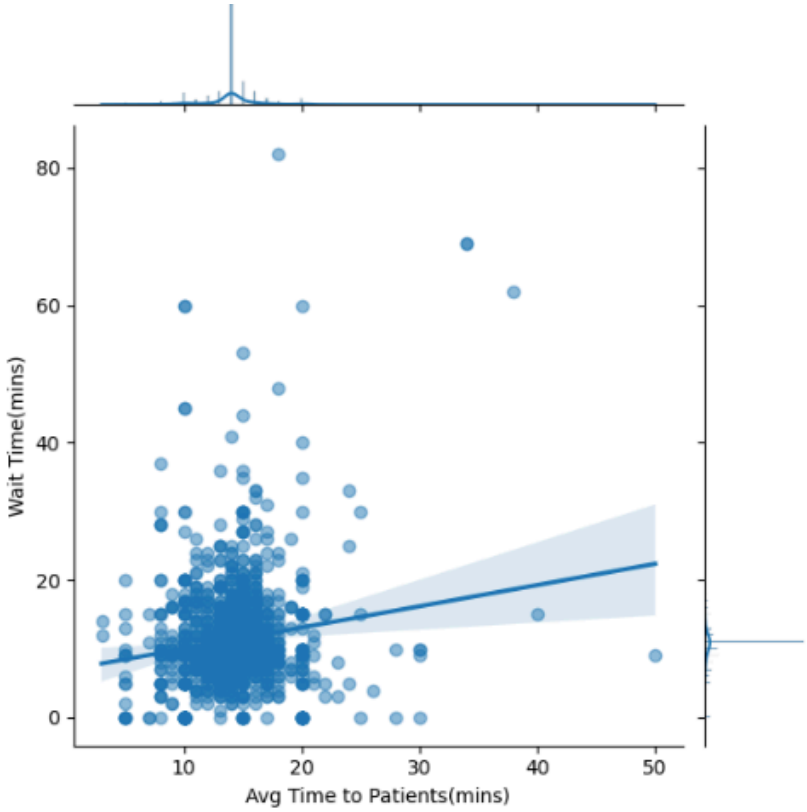

fig 2.10

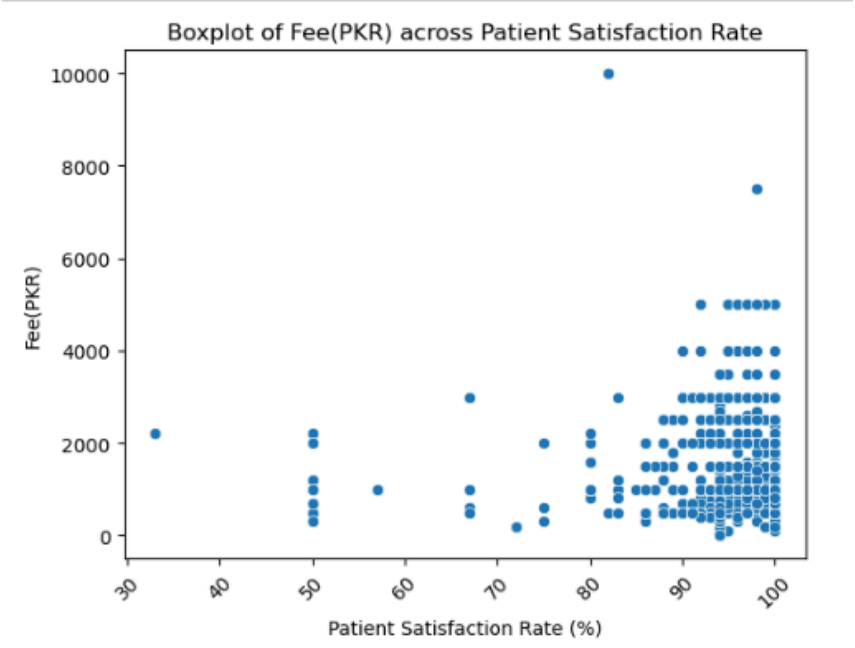

fig 2.11

Data Cleaning and Encoding

drop 22 rows for null values in fees and costing features columns doctor Name and index.

Qualification:

picked top 6 qualification in **fig 3.1** their names at **fig 3.3.** Number of people having FCPS & MCPS separately and their fees range **fig 3.4 ,**used **one hot encoder .**

Cities:

percentage of the city getting paid above average **fig 3.2 ,**used **target encoder.**

-Calculate Mean Target for Each Category
-Replace Categories with Mean Target Values

Specialization: using **target encoder**

-Calculate Mean Target for Each Category
-Replace Categories with Mean Target Values

Hospital Name: using **target encoder** to

-Calculate Mean Target for Each Category
-Replace Categories with Mean Target Values

and replace the no hospital address with zero as we consider it to be with no address.

doctor link: using one hot encoder as when doctor have a link I replace with one and when no link available put zero.

Outliers

We employed the Interquartile Range (IQR) method to identify outliers within our dataset. This method is particularly effective in detecting outliers within a continuous variable distribution ,Calculation of Quartiles (Q1, Q2, Q3): We computed the first quartile (Q1), median (Q2), and third quartile (Q3) for the variable of Fee which is our target q1=750,q2=1000,q3=1500 after that determination of Interquartile Range (IQR)
with this equation IQR=q3-q1 after that calculating :
Upper Threshold (range1): Q3 + 1.5 * IQR
Lower Threshold (range2): Q1 - 1.5 * IQR
This approach allows us to systematically detect outliers within the dataset by establishing threshold ranges beyond which values are considered unusual. It provides a robust means to identify potential anomalies that may impact the integrity of our analyses.

Robust encoding

This encoding method as it acts very good with outlier dataset and so it acts good with our dataset

```
{'FCPS': 1342,
 'FRCS': 75,
 'M.S': 4,
 'MCPS': 304,
 'professor': 3,
 'DTCD': 24,
 'MS': 110,
 'FACP': 14,
 'FACS': 19,
 'MBBS': 2209,
 'RMP': 20,
 'CFP (USA)': 1,
 'Certified in Covid 19 +': 1,
 'Certified (Aesthetic Medicine)': 1,
 'Master Of Surgery (Urology)': 2,
 'Training Geriatric Medicine UK': 1,
 'CHPE': 2,
 'MRCOG': 9,
 'General Physician': 1,
```

fig 3.1

Count	Ratio	Count
LAHORE	0.78	151
ISLAMABAD	0.76	147
KARACHI	0.61	151
MULTAN	0.59	126
SIALKOT	0.55	58
PESHAWAR	0.54	134
GUJRAT	0.43	35
GUJRANWALA	0.41	91
FAISALABAD	0.38	112
SAHIWAL	0.35	31
SARGODHA	0.33	55
QUETTA	0.31	118
OKARA	0.29	34
ABBOTTABAD	0.25	60
BAHAWALPUR	0.21	77
HYDERABAD	0.18	94
RAHIM-YAR-KHAN	0.13	77
SWABI	0.1	31

fig 3.2

Extra insights: all cities with rate higher than 50% are listed on Wikipedia's Top 10 (Cities by GDP) list. https://en.wikipedia.org/wiki/List_of_Pakistani_administrative_units_by_gross_state_product

MBBS : Bachelor of Medicine, Bachelor of Surgery

FCPS : Fellow of College of Physicians and Surgeons

MCPS : Membership of the College of Physicians and Surgeons

MS : Master of Surgery

MD : Doctor of Medicine

FRCS : Fellowship of the Royal Colleges of Surgeons

fig 3.3

fig 3.4

Reference: difference between FCPS & MCPS. https://en.wikipedia.org/wiki/List_of_Pakistani_administrative_units_by_gross_state_product.

According to this source, doctors with FCPS are paid more than doctors with MCPS. this'll be verified later.

Feature Selection

Anova Feature selection for discreet value and choose top 6 columns due to their distinguished high values using Kbest method which selects best features to train and test on it based on the number of feature we put in the K value **fig 4.1**

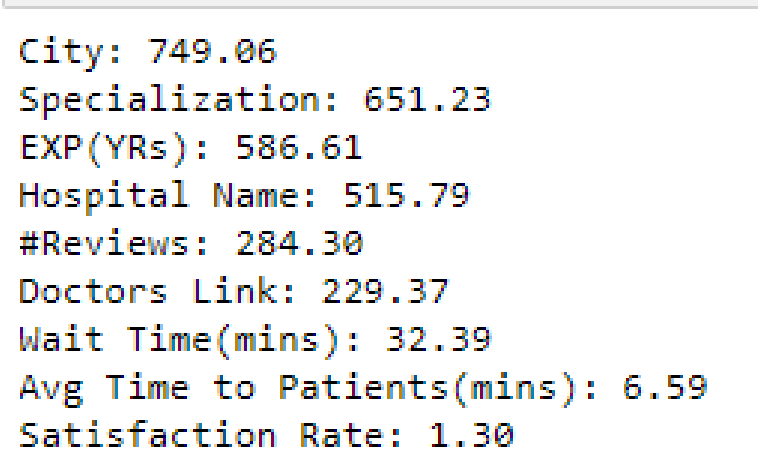

fig 4.1

Train,Test value

Train test split method with test value 30% and train value 70% and cross validation 389.5.

```
from sklearn.model_selection import train_test_split

from sklearn.svm import  SVR
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from math import sqrt


X_train, X_test, y_train, y_test = train_test_split(X_, y_, train_size=0.7, random_state=42)
```

[98] ✓ 0.0s

```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

[99] ✓ 0.0s

```
(1558, 12)
(669, 12)
(1558,)
(669,)
```

```
from sklearn.model_selection import GridSearchCV

RFR_params = {
    'n_estimators': [90, 100,110, 120],
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 3],
    'min_samples_leaf' : [2,3],
}

GSCV = GridSearchCV(RandomForestRegressor(), cv=4, param_grid=RFR_params, scoring='r2')
GSCV.fit(X_train, y_train)
```

✓ 1m 23.1s

Machine Learning Models

Tree Based Models

A variety of Tree Based models were employed in this project, to reach the optimal accuracy metrics, those metrics were:

- Root Mean Square Error (RMSE)
- R^2 score

the best values for these metrics were reached using:

- Random Forest
- XGBoost
- AdaBoost (the base estimator was tree based)
- Extra Tree Regressor
- Voting Regressor (which combined XG, RF, and Extra Tree)

Furthermore, hyperparameter tuning was applied on each of these models to get the best possible hyper-parameters, reaching an **RMSE** of **269** and an **R^2** of **0.78** on the Voting Regressor.

1- Random Forest

```
RFR1 = RandomForestRegressor( max_depth= 10, min_samples_leaf= 3, min_samples_split= 2, n_estimators= 50)
RFR1.fit(X_train, y_train)
evaluate_model(RFR1)
```

✓ 0.4s

```
train rmse: 195.79805105037656
train r2 score: 0.885237134324329
--------------------
------------------
test rmse: 272.85568662027197
test r2 score: 0.7773835397344715
```

2- XGBoost

```
XG = GradientBoostingRegressor(max_depth=3, min_samples_split=4, min_samples_leaf=1, n_estimators=100, random_state = 102)
XG.fit(X_train, y_train)
evaluate_model(XG)
```

✓ 0.1s

```
train rmse: 248.37774880542474
train r2 score: 0.8153241833989537
--------------------
-----------------
test rmse: 270.7613866926286
test r2 score: 0.7807878036387337
```

3- Extra Tree

```
from sklearn.ensemble import ExtraTreesRegressor
ETR = ExtraTreesRegressor(max_depth=10, min_samples_split=2, min_samples_leaf=1,
                          n_estimators=65, random_state = 102, oob_score=True, bootstrap=True)

ETR.fit(X_train, y_train)
evaluate_model(ETR)
```

✓ 0.1s

```
train rmse: 215.72110948424344
train r2 score: 0.8606939604121102
--------------------
------------------
test rmse: 287.2162650442616
test r2 score: 0.7533339866103301
```

4- Voting Regressor

```
from sklearn.ensemble import VotingRegressor

vr = VotingRegressor(
    estimators=[('XG', XG), ('ETR', ETR), ('RFR', RFR2),],
)

vr.fit(X_train, y_train)

evaluate_model(vr)
```

✓ 0.7s

```
train rmse: 229.71093691790801
train r2 score: 0.8420396812610825
__________________
------------------
test rmse: 269.0791481049653
test r2 score: 0.7835032694297744
```

5- AdaBoost

```
from sklearn.ensemble import AdaBoostRegressor

ADA = AdaBoostRegressor(base_estimator=XG , learning_rate=0.01, n_estimators=100, random_state=102)

ADA.fit(X_train, y_train)
evaluate_model(ADA)
```

✓ 11.3s

```
train rmse: 233.4005231266453
train r2 score: 0.8369246552632413
__________________
------------------
test rmse: 271.0462889954535
test r2 score: 0.7803262391725315
```

Conclusion:

Our investigation into the dataset began with the assumption that the data would follow a normal distribution. However, to our surprise, the data did not adhere to this expectation, which posed initial challenges in achieving high accuracy. Despite applying various methods to correct the data distribution, these issues persisted, leading to lower-than-expected accuracy. Through our rigorous analysis and modeling efforts, particularly in developing a predictive model for estimating doctor fees, we focused on key factors such as the doctor's years of experience, specialization, city location, and patient feedback. These factors were hypothesized to significantly influence the fees charged, and our findings confirmed their crucial roles in shaping the fee structure.

Finally, it was concluded that Voting Regressor ;combination of extra tree regressor, XGboost, Random Forest.
RMSE: 269
R^2: 0.78

- Mahmoud Sayed AI3
- Bassem Mahmoud AI3
- Basant Badr AI3
- Youssef Osama ROB3
- Aya Ahmed Sayed ROB3
- Mostafa Samy ROB3