[DOCTOR FEE PREDICTION]

Milestone 1

12 May 2024

Machine Learning

Prof. Dina khatab

Team

AI_R_1
Nour sameh 2021170885
Nour madkour 2021170884
Rawan badr 2021170863
Menna maged 2021170875
Hoor hesham 2021170861

Data Overview

Data Description

- **Doctor Name**: The name of the doctor.
- **City**: The city where the doctor is located.
- **Specialization**: The area of specialization or medical expertise of the doctor.
- **Doctor Qualification**: The qualifications or degrees held by the doctor.
- **Experience (Years)**: The number of years of experience the doctor has.
- **Total Reviews**: The total number of reviews received by the doctor.
- **Patient Satisfaction Rate (%age)**: The percentage of patients satisfied with the doctor's services.
- **Avg Time to Patients (mins)**: The average time taken by the doctor to attend to patients.
- **Wait Time (mins)**: The average wait time for patients before being attended to by the doctor.
- **Hospital Address**: The address of the hospital where the doctor practices.
- **Doctors Link**: A link to the doctor's profile or information.
- **Fee (PKR)**: The consultation fee charged by the doctor .

```
 #   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
```

1.No null values but there may be missing values
2.data need to be encoded "Doctor Name" , "City", "Specialization" ,
"Doctor Qualification" ,"Hospital Address", "Doctors Link"

3. dropped duplicated values
4. Statistics for categorical columns.

	count	unique	top	freq
Doctor Name	2373	2190	Dr. Muhammad Amjad	4
City	2373	117	LAHORE	151
Specialization	2373	150	General Physician	406
Doctor Qualification	2373	1041	MBBS	332
Hospital Address	2373	1178	No Address Available	552
Doctors Link	2373	1606	No Link Available	645

found out that most freq in these 2 cols 'Hospital Address' & 'Doctors Link ' No Address Available & No Link Available So we may drop them and may not
5. Renamed columns to manipulate on the easily

Table of content

Doctor Fee Prediction

Team :

instruction : comment the visualisation in cities for running faster

Importing libraries and Reading Data

Data Description

Task 1: Explore and Familiarize with the Dataset:

1 Dive into the dataset to uncover any peculiarities or unexpected patterns that may influence linear regression modeling.

1.Doctor Name

Cleaning

Analysis

City

Cleaning

Analysis

Feature Engineering (Region)

3.Specialization

Cleaning

Analysis

Feature Engineering (Specialization Count)

Encoding

Encoding top 15 momkn 10 brdo shofo

Doctor Qualification

NLP

Cleaning

Analysis

Feature Engineering (Number of Qualification)

5.Hospital Address

Analysis

No Address Available

6.Doctors Link

Numerical Values

7.Experience(Years)

8.Total_Reviews

9.Patient Satisfaction Rate(%age)

10.Avg Time to Patients(min)

11.Wait Time(mins)

Feature Engineering (Total Time)

12.Fee(PKR)

1.Doctor Name

1.1 Cleaning

	count	unique	top	freq
Doctor Name	2373	2190	Dr. Muhammad Amjad	4

1.saw duplicates

```
Doctor Name
Dr. Muhammad Amjad                                4
Asst. Prof. Dr. Mujahid Israr                     4
Dr. Asim Munir Alvi Consultant Endocrinologist    3
Dr. Muhammad Saddiq Haris                         3
Asst. Prof. Dr. Shoaib Manzoor                    3
                                                 ..
Dr. Maria Anwar                                   2
Assoc. Prof. Dr. Mudassar Nazar                   2
Dr. Wasim Akram                                   2
Assoc. Prof. Dr. Fowad Shahzad                    2
Prof. Dr. Fawad Nasrullah                         2
Name: count, Length: 167, dtype: int64
```

2. saw freq of the repeated drs

```
Doctor Name: Assoc. Prof. Dr. Fowad Shahzad
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Assoc. Prof. Dr. Irfan Munir
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Assoc. Prof. Dr. Mudassar Nazar
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Assoc. Prof. Dr. Muhammad Farooq
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Asst. Prof. Dr. Adil Hassan Chang
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Asst. Prof. Dr. Alamzeb Khan
----------------------------------------
Number of rows: 2
----------------------------------------
Doctor Name: Asst. Prof. Dr. Asif Malik
```

Most from 2-4 repitition
the duplicated names maybe Similarity of names we will see that when see more relation with other columns like 'Link' , ‘City’, ‘Specialization’(refer to 1.2 in Dr names number(5))

1.2 Feature engnieering split (Doctor Names)To (Titles), (Names)

3. `extract_titles_and_name`
4. `find_empty_title_rows`

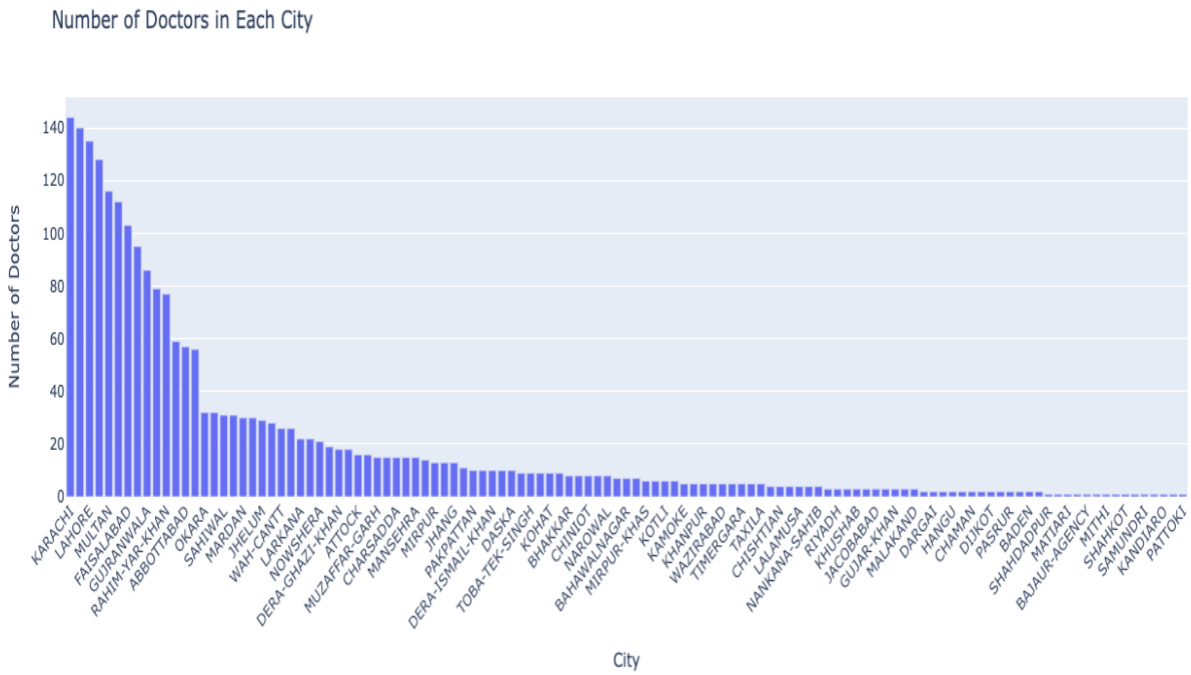
```
Titles
Dr               1899
Asst Prof Dr      247
Prof, Dr          148
Assoc Prof Dr      79
Name: count, dtype: int64
```

If not from these added to unique value ‘others’

5. 1st refered to dropped rows of dr that have same ['Doctor Name', 'Specialization','City'] and also 'Doctors Link' as it is wierd to have dr have same all things(80 rows)

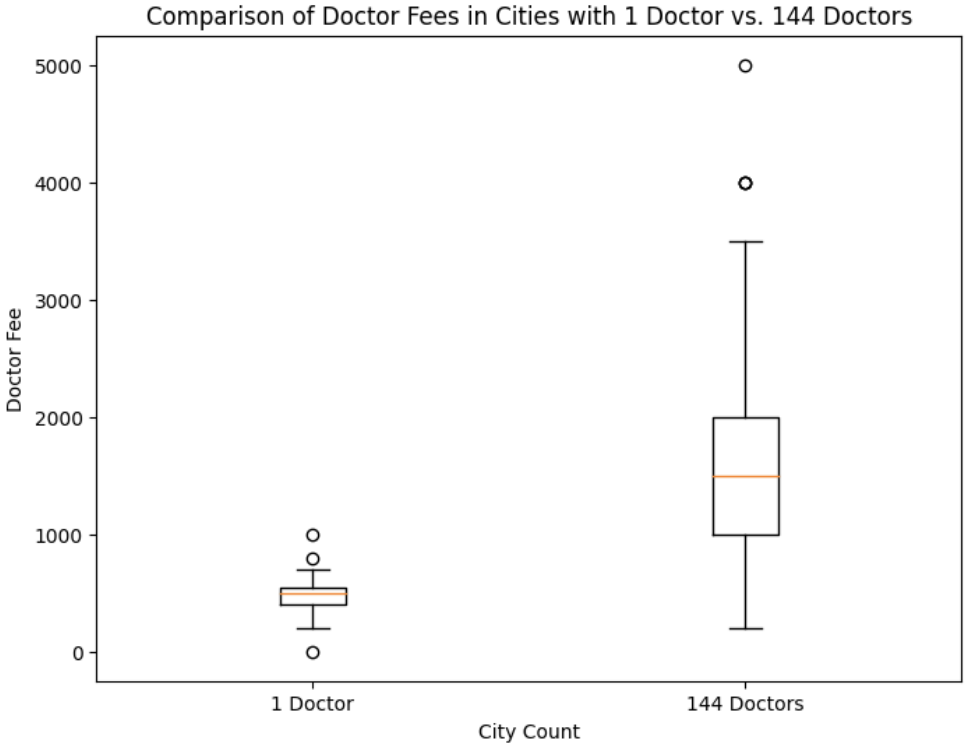
1.3 Analysis

1



In conclusion, the analysis reveals that Karachi has the highest number of doctors compared to other cities in the dataset. However, it's worth noting that several cities have only one doctor listed. This discrepancy in the distribution of doctors across cities might indicate variations in healthcare accessibility and resource allocation.

2.




```
Summary Statistics for Fees in Cities with 1 Doctor:
count      15.000000
mean      493.333333
std       240.436112
min         0.000000
25%       400.000000
50%       500.000000
75%       550.000000
max      1000.000000
Name: Fee, dtype: float64

Summary Statistics for Fees in Cities with 144 Doctors:
count     144.000000
mean     1524.305556
std       816.460597
min       200.000000
25%      1000.000000
50%      1500.000000
75%      2000.000000
max      5000.000000
Name: Fee, dtype: float64
```

Cities with 1 doctor have a fee range from 0 to 1000, whereas cities with 144 doctors have a wider fee range from $200 to 5000 This indicates greater variability in the fees charged by doctors in cities with a higher concentration of doctors.

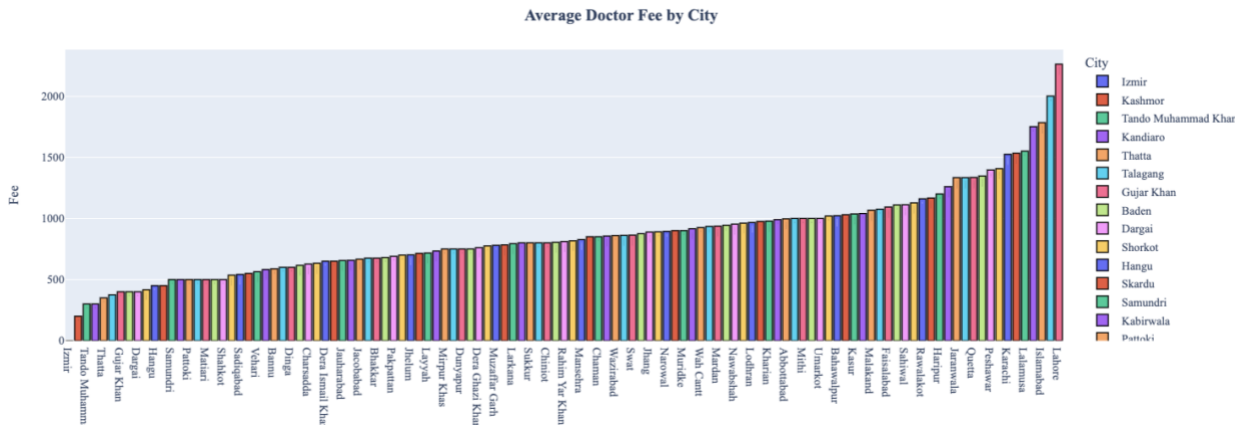
avg comparing
In cities with only 1 doctor, the mean fee charged is significantly lower at approximately 493.33 compared to cities with 144 doctors, where the mean fee is substantially higher at approximately 1524.31. This suggests that there is a noticeable difference in the average fees depending on the number of doctors in a city.

2.City

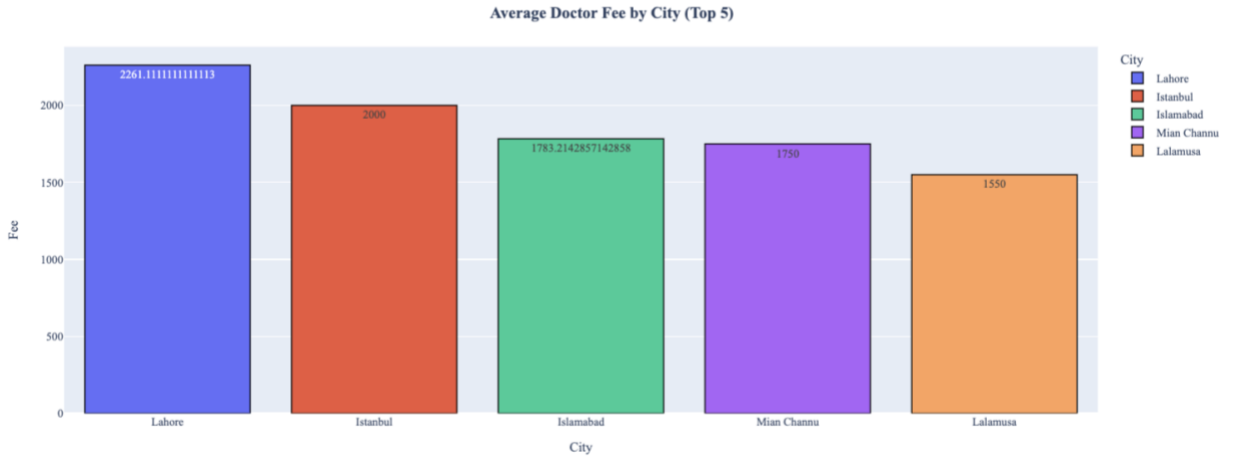
2.1 Cleaning

1.Removed '-' and made city lower case in order to see if there is any repeated city

2.2 Analysis



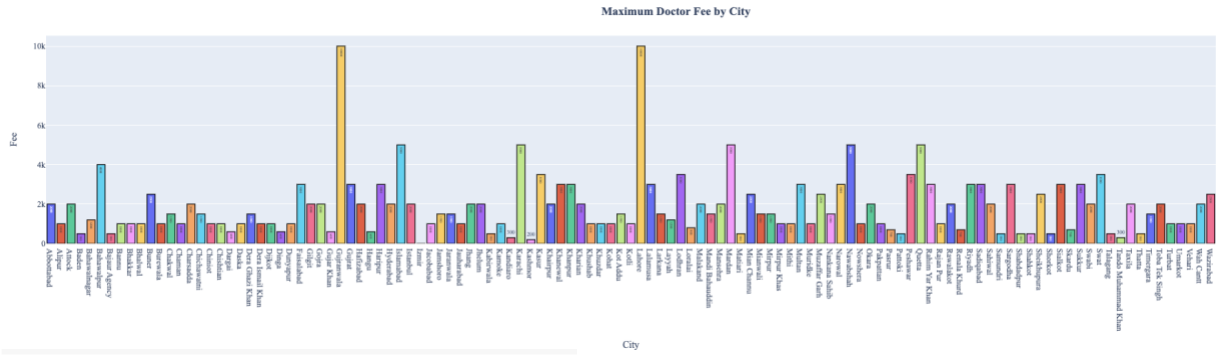
2. the avg fees are in LAHORE city lets dive into it and see fees of drs in this city , whilr in IZMIR is the lowes avg tends to 0



3.The fees vary considerable, with a standard deviation of approximately 1369.37, suggesting a wide range of fee amounts.
4. The city with the minimum average fees is Izmir with an average fee of $0.00.

	Doctor Name	City	Specialization	Doctor Qualification	Experience_Years	Total_Reviews	Patient_Satisfaction_Rate	Avg_time_per_Patient	Wait_Time	Hospital Address	Doctors Link	Fee	Titles
2025	Hayati Turker	Izmir	Eye Surgeon	M.D.	18.0	0	94	14	11	Netgoz Eye Hospital, YalÃ± Mahallesi, Izmir	No Link Available	0	Dr

5. this suggest that this dr giving charity for ppl as it is free for (Eye Surgeon)
`Patient_Satisfaction_Rate is 94`




```
Summary Statistics for Fees in GUJRANWALA:
count       86.000000
mean      1259.534884
std       1146.132019
min         20.000000
25%        700.000000
50%       1000.000000
75%       1500.000000
max      10000.000000
Name: Fee, dtype: float64
```

a wide range of fees charged by doctors in the city, with a minimum fee of 20 and a maximum fee of 10,000. The mean fee of approximately 1259.53 indicates the average cost of medical services

(IQR) of 800 (from 700 to 1500) suggests that the middle 50% of fees are relatively consistent, with the median fee (1000) falling within this range. This indicates that while there is considerable variability in fees, **a significant portion of doctors charge fees within a certain range**

2.3 Feature Engineering (Region)

Used folium library to see correlation between cities and desided to devide it to 6 regions


```
df['Region'].value_counts()
```

```
Region
Punjab Region           1420
KPK Region               389
Sindh Region             329
Balochistan Region       119
Kashmir Region            24
International Region      12
Name: count, dtype: int64
```

3.Specialization

2.1 Cleaning

```
[ ] df[['Specialization']].describe().T
```

	count	unique	top	freq
Specialization	2293	140	General Physician	406

1.Processed specialization to remove redundant values

```
from tabulate import tabulate
```

Captured more typos and mapped the corrected once

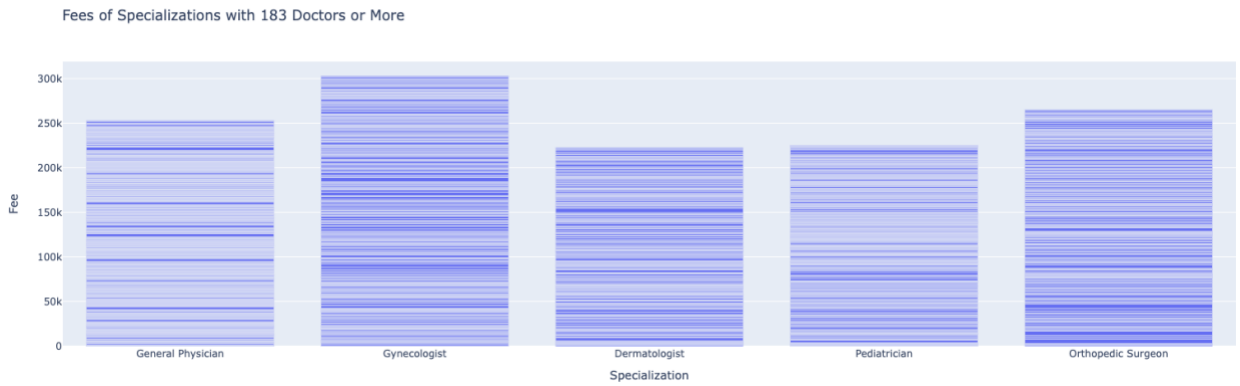
```
specialization_mapping = {
    "Pediatrician,Pediatric": "Pediatrician",
    "Lung Specialist": "Pulmonologist",
    "Eye Surgeon,Eye Specialist": "Ophthalmologist",
    "Sexologist": "Andrologist",
    "Cosmetic Surgeon,Dermatologist": "Cosmetic Dermatologist",
    "Internal Medicine Specialist,General Physician,Infectious
Diseases": "Infectious Disease Specialist",
}
```

```
df["Specialization"].nunique()
```

```
104
```

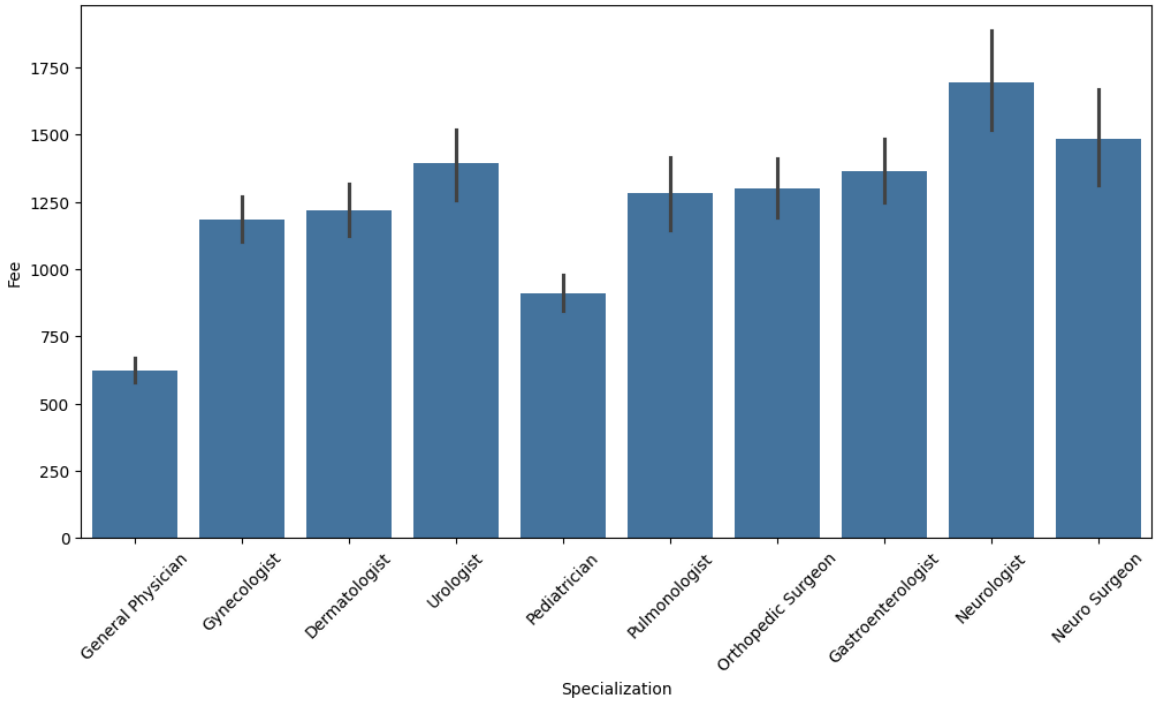
Went from 140 to 128 and that's good

2.2 Analysis



```
    print()
```

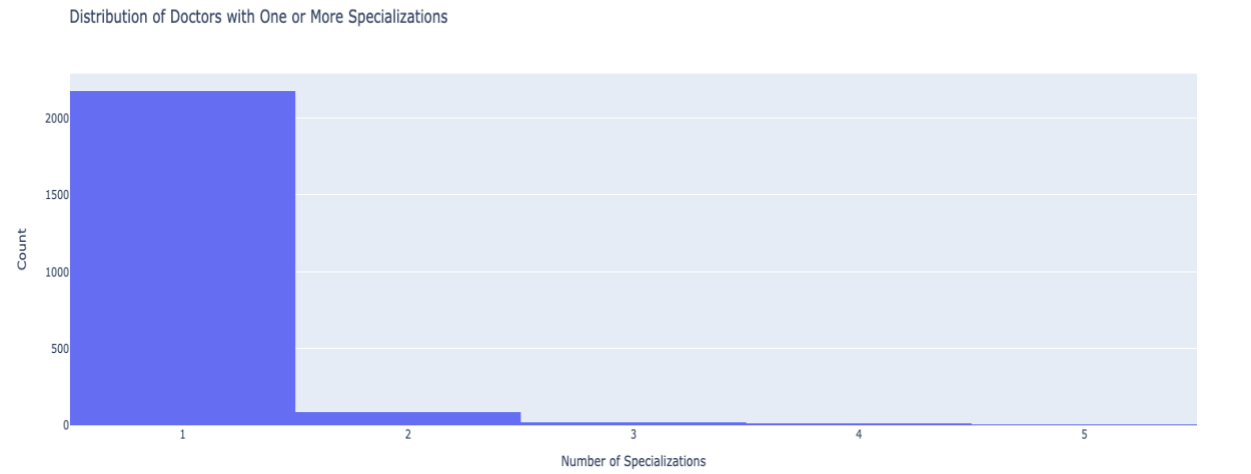
```
Specialization: General Physician
Maximum Fee: 4000
Minimum Fee: 0

Specialization: Gynecologist
Maximum Fee: 5000
Minimum Fee: 0

Specialization: Pediatrician
Maximum Fee: 3000
Minimum Fee: 5

Specialization: Orthopedic Surgeon
Maximum Fee: 5000
Minimum Fee: 0

Specialization: Dermatologist
Maximum Fee: 5000
Minimum Fee: 0
```



Drs with less experiene have highest rating and dr that are 47 yrs are not given a rate

Nephrologist,Diabetologist apears to have same max and min fees so lets view it

3.3 Feature Engineering (Specialization Count)

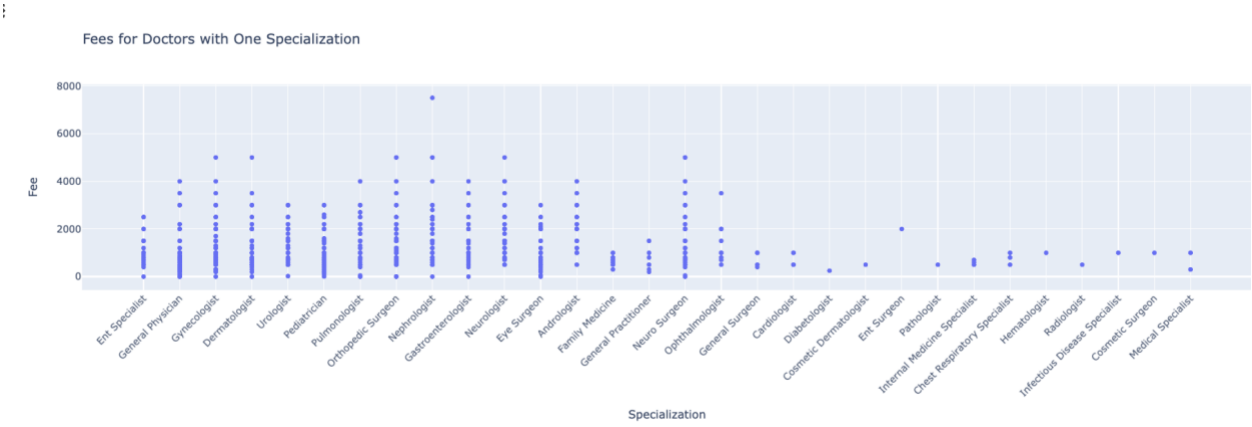


1 . Most drs have 1 or 2 specialization

Number of Dr with multiple specializations: 118

3. `Maximum number of specializations: 5`

`Minimum number of specializations:1`



2. fees varies but the Neurologist takes highest fees

`4 drs that have 1 or 2 specialization are the most ppl that take higer fees that is expected 3shan e7na bnb2a 3wziin nroo7 to specialized dr not  drs for everthing`

4. Doctor Qualification

	count	unique	top	freq
Doctor Qualification	2293	1014	MBBS	332

4.1 cleaning

Doctor Qualification	
MBBS	332
MBBS, FCPS	129
MBBS	78
MBBS,FCPS	68
MBBS, FCPS (Gastroenterology)	40
MBBS, FCPS (Obstetrics & Gynecology)	39
MBBS, FCPS (Orthopedic Surgery)	38
MBBS, FCPS (Dermatology)	33
MBBS, FCPS (Urology)	28
MBBS, FCPS (Pulmonology)	24
MBBS, FCPS (Neurology)	24
MBBS, FCPS (Neuro Surgery)	22
MBBS, FCPS (Pediatrics)	18
MBBS, MCPS	18
MBBS, FCPS (Nephrology)	17
MBBS, FCPS (Orthopaedic Surgery)	15
MBBS , FCPS	14
MBBS , FCPS (Obstetrics & Gynecology)	14
MD	14
MBBS, MCPS, FCPS	12
MBBS, DTCD	10
MBBS, MS (Urology)	10
MBBS , FCPS (Orthopedic Surgery)	10
MBBS, MD	10
MBBS, DCH	9
MBBS, FCPS (Obstetrics & Gynaecology)	9
MBBS , FCPS (Obstetrics & Gynaecology)	9
MBBS, FCPS, MCPS	8
MBBS, MS (Neurosurgery)	8
MBBS, FCPS (Neurosurgery)	8
MBBS , FCPS (Orthopedic Surgery)	7
MBBS, MCPS (Obstetrics & Gynaecology)	7
MBBS, FCPS (Ophthalmology)	7
MBBS, MCPS (Pediatrics)	7
MBBS , FCPS (Pediatrics)	7
MBBS, FCPS (Gastroentrology)	6
MBBS, MCPS (Dermatology)	6
MBBS , MS (Neurosurgery)	6
MBBS, RMP	6
MBBS, FCPS (Medicine)	6
MBBS, DOMS	5
MBBS, FCPS (Paediatrics)	5

there are typo error so will remove it

Tried nltk gave us 845 unique so prefered to do it manually

```
"""
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer

# Download NLTK resources (if not already downloaded)
#nltk.download('punkt')
#nltk.download('stopwords')
#nltk.download('wordnet')

# Initialize Lemmatizer and CountVectorizer
lemmatizer = WordNetLemmatizer()
vectorizer = CountVectorizer()

# Tokenization and Lemmatization
def preprocess_text(text):
    tokens = word_tokenize(text.lower())  # Tokenization and convert to lowercase
    tokens = [lemmatizer.lemmatize(token) for token in tokens]  # Lemmatization
    return ' '.join(tokens)

# Remove stopwords and perform lemmatization
stop_words = set(stopwords.words('english'))

# Apply preprocessing to each value in the 'Doctor Qualification' column
df['Cleaned Qualifications'] = df['Doctor Qualification'].apply(preprocess_text)

# Vectorization using Bag of Words
X = vectorizer.fit_transform(df['Cleaned Qualifications'])

# X now contains the vectorized representation of 'Doctor Qualification' column
"""
```

```python
def clean_qualifications(df):
    # Combine and update all replacements into a single dictionary
    replacements = {
        r'\bPhD\b': 'PHD', r'\bM\.D\.\b': 'MD', r'\bD\.M\.S\b': 'DMS',
        r'\bB\.Sc\.\b': 'BSC', r'\bM\.S\.\b': 'MS', r'\bM\.Phil\b': 'MPHIL',
        r'\bG\.A\.M\.S\b': 'GAMS', r'\(D\.H\.B\)': 'DHB', r'\(D\.Ac\)': 'PHD',
        r'Ophtamology': 'Ophthalmology', r'Gastroentrology': 'Gastroenterology',
        r'OtoRhinoLaryngology': 'Otorhinolaryngology', r'Paediatrics': 'Pediatrics',
        r'Pulmonology': 'Pulmonary', r'ENT': 'Otolaryngology', r'OrthopedicSurgery': 'Orthopedic Surgery',
        r'NeuroSurgery': 'Neurosurgery', r'Medicine': 'Internal Medicine',
        r'OBSTETRICS&GYNAECOLOGY': 'Obstetrics&Gynecology', r'Gynecology&amp;Obstetrics': 'Gynecology and Obstetrics',
        r'Genecology&amp;Obstetrics': 'Gynecology and Obstetrics', r'OtorhinolaryngologicENT': 'Otorhinolaryngologic,ENT',
        r'MasterOfSurgery': 'Master of Surgery', r'MD\d*': 'MD', r'MDGastroenterology': 'MD,Gastroenterology',
        r'FCPSPediatrics': 'FCPS,Pediatrics', r'MBBSMD': 'MBBS,MD', r'FRCSOrthopedics': 'FRCS,Orthopedics',
        r'MCPSGynae/Obs': 'MCPS(Gynecology/Obs)', r'MD-RMP': 'MD, RMP', r'Masters\(NeuroSurgeon\)': 'Masters, Neurosurgery',
        r'\(|\)': '', r'[^a-zA-Z,]': '', r'Ophthalmologist': 'Ophthalmology', r'GASTROENTEROLOGY': 'Gastroenterology',
        r'MCPS,': 'MCPS', r'M\.D': 'MD', 'MD 1': 'MD'
    }

    # Apply all replacements
    df['Doctor Qualification'] = df['Doctor Qualification'].replace(replacements, regex=True)

    # Additional replacements to handle specific concatenations
    concatenations = {
        r'FCPSOBSTETRICSampGYNAECOLOGY': 'FCPS,Obstetrics&Gynecology',
        r'FCPSOtolaryngology': 'FCPS,Otolaryngology',
        r'MCPSFCPS': 'MCPS,FCPS'
    }
    df['Doctor Qualification'] = df['Doctor Qualification'].replace(concatenations, regex=True)

    # Remove all unnecessary spaces, then remove spaces around commas
    df['Doctor Qualification'] = df['Doctor Qualification'].str.replace(r'\s+', '')
    df['Doctor Qualification'] = df['Doctor Qualification'].str.replace(r'\s*,\s*', ',', regex=True)
    df['Doctor Qualification'] = df['Doctor Qualification'].str.replace(r'\([^)]*\)', '', regex=True)


    # Enhanced cleaning function
    def enhance_cleaning(qualification):
        # Replace HTML entities and correct specific cases
        qualification = qualification.replace('&amp;', '&')
        qualification = re.sub(r'(?<!\w)([A-Z]+)(?!\w)', lambda x: x.group(1), qualification)
        qualification = qualification.replace('DiplomainTBandChestDiseases', 'DTBCD')

        # Split, sort, and remove duplicates
        parts = sorted(set(qualification.split(',')))  # Remove duplicates and sort
        return ','.join(parts)

    # Apply the enhanced cleaning function
    df['Doctor Qualification'] = df['Doctor Qualification'].apply(enhance_cleaning)

    return df
```

826 and that’s better than nltk

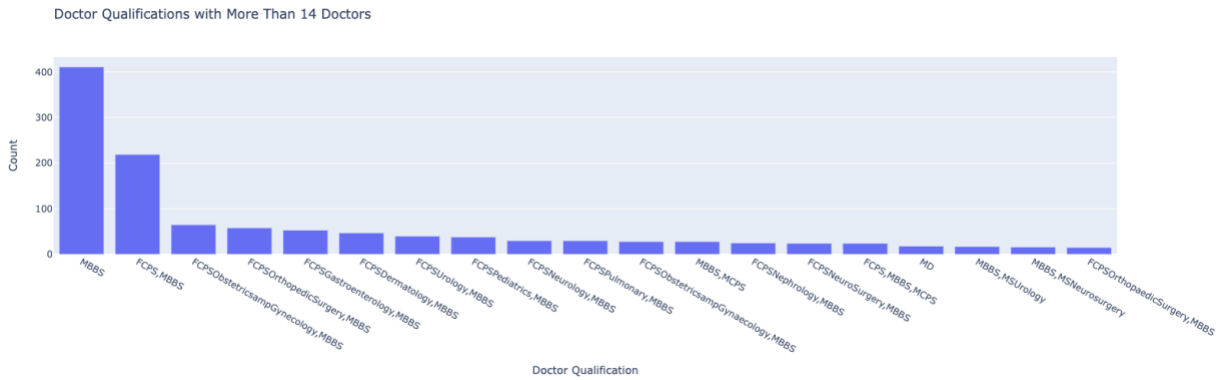
4.2 Analysis

```
#drs that dont have mbbs
#conclusion : 89 rows
```

```
Qualifications of rows where 'MBBS' is not present:
MD
MD,MSMasterofSurgery
DiplomaAnaesthsia,DiplomaChildHealthfromInstituteofHealthampManagementScienceIslamabad,MD
DHMS,DIPSexology,InternationalAffiliateMemberAmericanPsychologicalAssociationDivisionUSA,MScPsychology
DHMS
CISUK,CLDPUAE,CMTOSriLanka,CTOFUK,DIPSexology,MPHILPsychology,RHMPPak
CRSM,MD,MemberofAmericanSocietyforReproductiveInternalMedicine,PGFM
CertifiedSexologist,DiplomaPsychosexualampRelationshipTherapist,DiplomateSexologist,PHDHumanSexology,RHMP
MD
FCPS,PHDGastro
DoctorofInternalMedicineMD
MDBasicMedicalQualification
MD
MD
Doctorate
CertifiedUrodynamicist,MD,MSUrology
MSneurosurgeon
MD
MD,MSC
MCPS
MCPSPediatrics,MD
MD,MasterofSurgeryinOrthopedics
FCPSPediatrics,MD
FRACSOrthopedicSurgery,FRCSTraumaampOrth,MBCHBBachelorofInternalMedicineampBachelorofSurgery
DiplomainDiabetes,MD
DAC,DHMS,DIPSexology
DiplomaInDermatology,DoctorofInternalMedicineMD,MemberofMedicalDermatologysocietyUSAMMDSUSA
FCPSNeuroSurgery,MD
FACCUSA,FACPUSA,FASIMUSA,FRCPEdinburgh,FRCPLONDON,MRCPUK
MCPSPediatrics,MD
MDBASICMEDICALQUALIFICATION
MD
MD
MD,MastersNeuroSurgeon
FCPS,MD
MDBasicMedicalQualification
MD
FCPS,FRCS,MCPS
CRSMSexualampReproductiveInternalMedicine,FCPSUrology,MCPSGenSurgery,MD,MemberofPakistansocietyforAndrologyampSexualInternalMedicinePSA
MD
DiplomateAmericanBoardHairRestorationSurgery,MBB,MCPSDermatology
MD
DHMS
FCPSSURGERY,FCPSUROLOGY
FCPSUROLOGY,MD,MRCS
DIPLASTHUK,DPDUK,MDEurope,MRCGPUK
MD
MDGeneralPhysician
FCPSPediatrics,MCPSPediatrics,MD
MD
MD
DiplomaDermatology,MD,MasterofScienceinPublicHealth
FCPSII,MD
CertifiedUrodynamicist,MD,MSUrology
FCPSNeurology,FRCP,MD
MDRussia
FellowshipinVitreoretinalDiseasesandSurgery,Ophthalmology
```

- 96.2% of the drs have MBBs Qualification
- 3.8 % only dont have MBBs.

most of drs have MBBS (Bachelor of Medicine, Bachelor of Surgery)

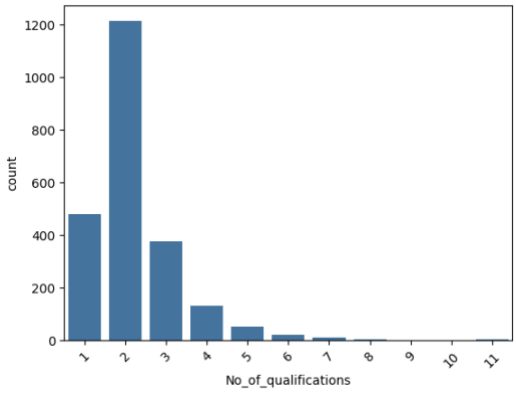


Most drs take this qualifications

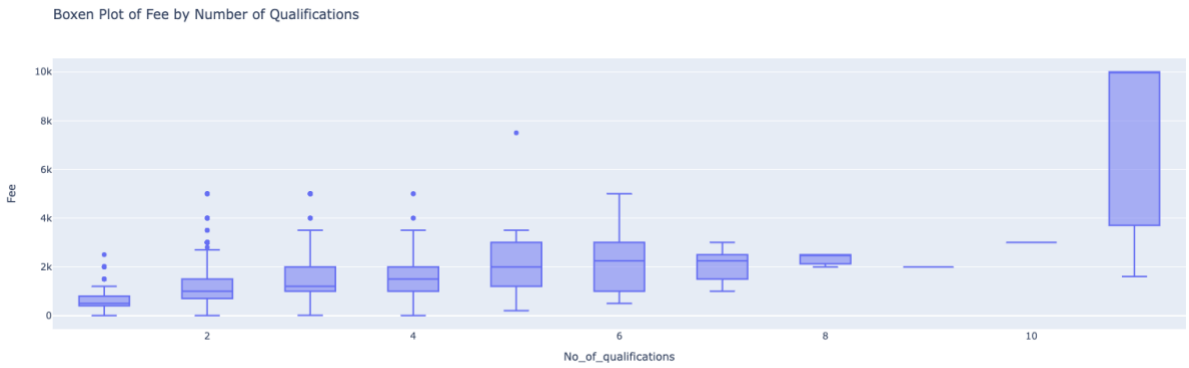
```
Maximum number of qualifications: 11
Minimum number of qualifications: 1
```

4.3 Feature Engineering (Number of Qualification)

No_of_qualifications	
2	1215
1	481
3	376
4	130
5	53
6	20
7	10
11	3
8	3
10	1
9	1

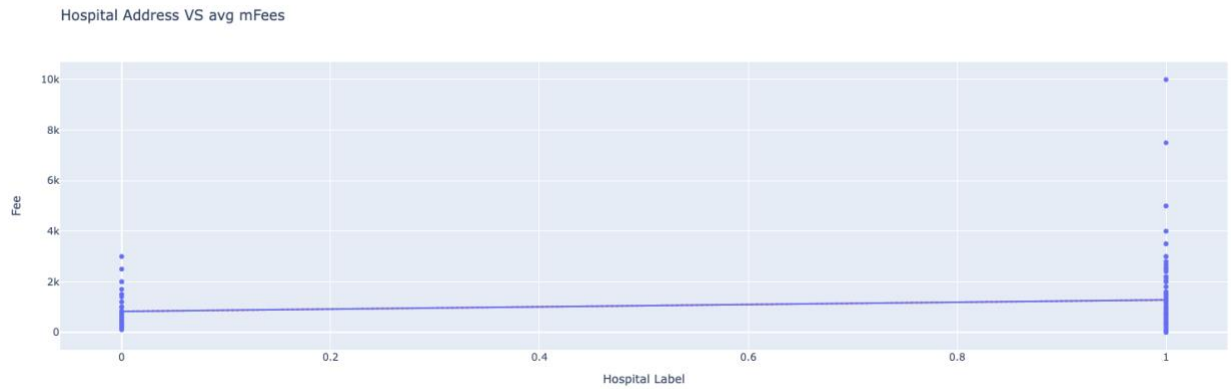


Distribution of Number of Qualifications



As the number of qualifications of doctors increase, So does there fees increases, but after 8 qualifications fees seem to decrease and then increase at 12

5.1 Hospital Address
feature engnieering has_hospital address

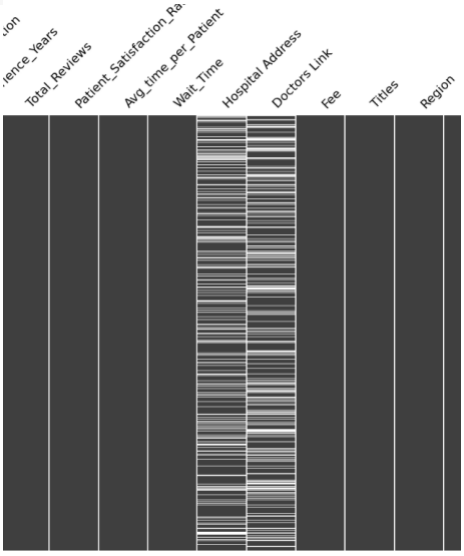


Those who have address seems to be easier to patient to go to so avg fees of drs are more

```
                Doctor Name
1             Haris Shakeel
8               Awais Ahmad
27          Anwisha Samreen
44                Saad Arif
48              Komal Azhar
...                     ...
2329             Maria Ijaz
2344              Asim Niaz
2365          Umber Mushtaq
2367  Hamraz Khan Yousaf Zai
2379        Muhammad Ikrama

[254 rows x 1 columns]
```

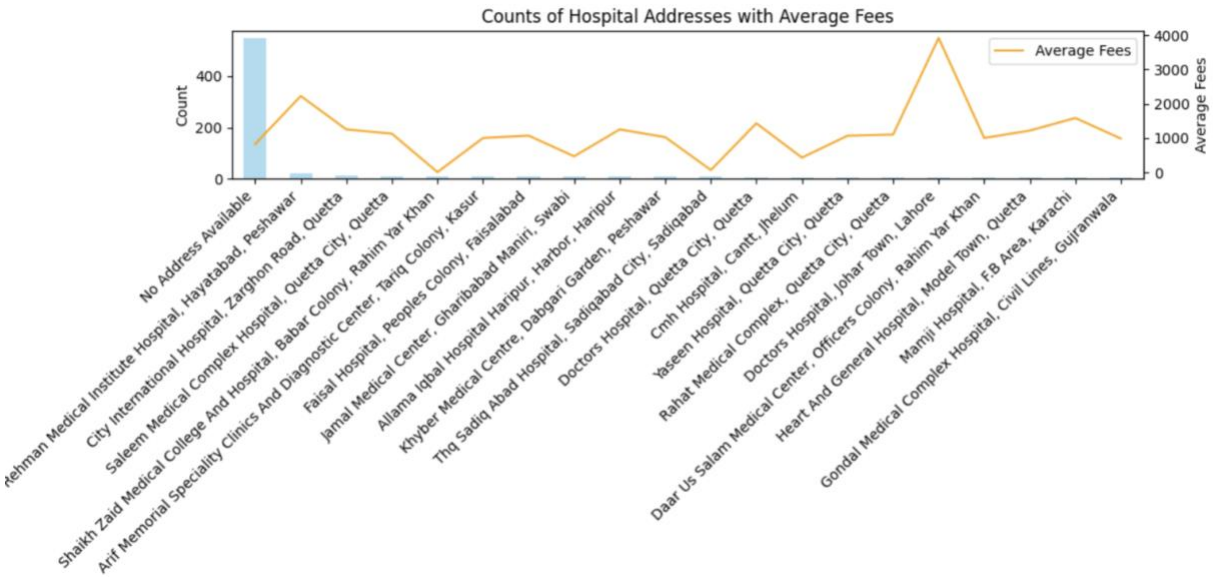
Drs that don't have both link nor hospital address `254`

`Wanted to see relation on msno`



5.1 Analysis

`Highest avg fees in` Doctors Hospital, Johar Town,Lahore hase the highest avg fees , it appeared 6 times in df



`Doctors with the highest fees at Doctors Hospital, Johar Town, Lahore :`

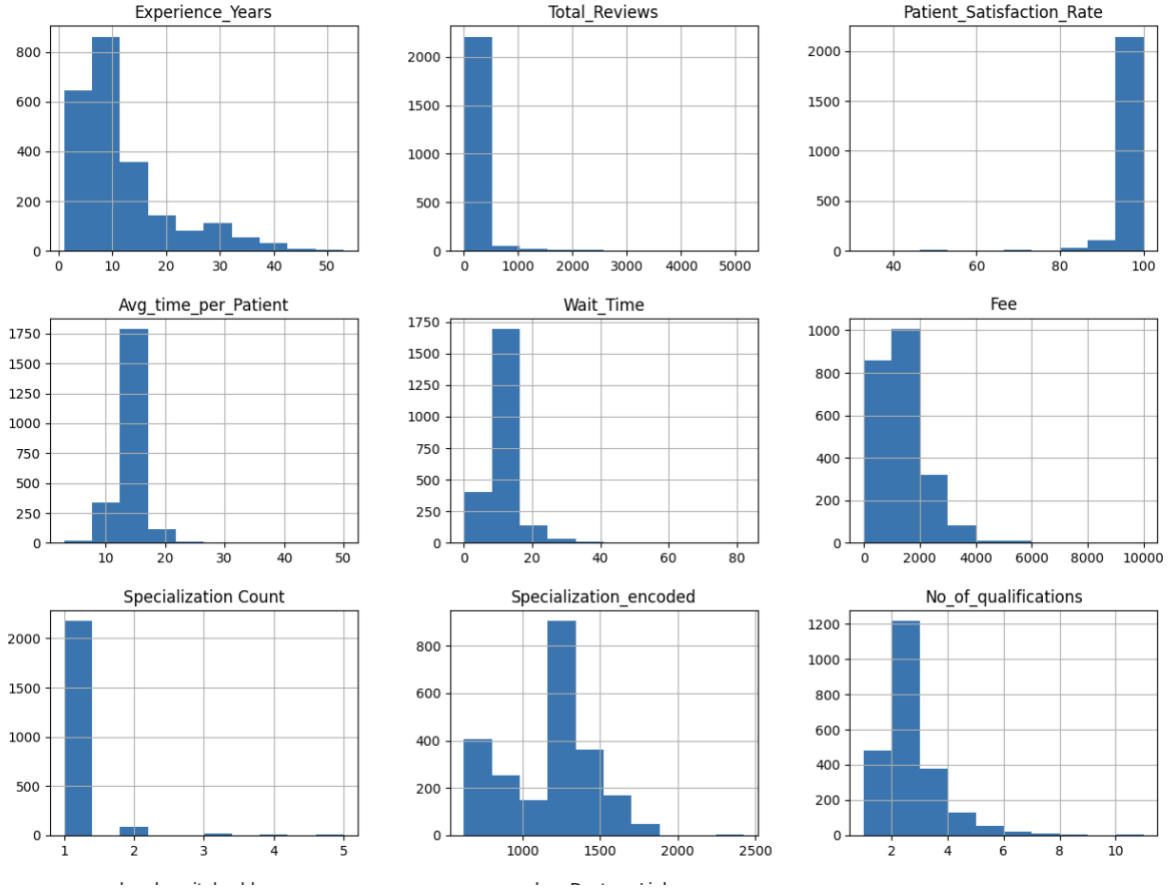
	Doctor Name	Fee
46	Ghazanfar Ali Shah	5000
96	Qurat Ul Ain Sajida	3500
839	Syed Shahzad Hussain Shah	3000
987	Tariq Sohail	4000
1767	Muhammad Bilal	3000
1959	Khurshid Alam	5000

6.Doctors Link

	count	unique	top	freq
Doctors Link	2293	1567	No Link Available	645

Replaced no link av by homepage `df['Doctors Link'] = df['Doctors Link'].replace('No Link Available', 'https://instacare.pk/')`

Numerical Values



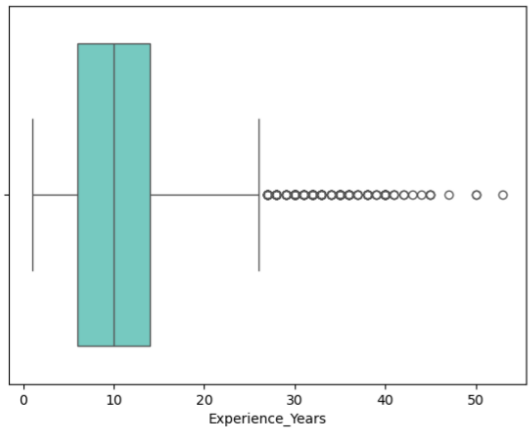
we concluded that

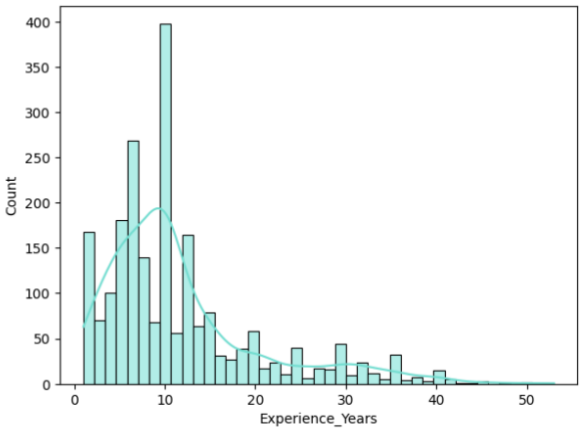
1. experience is somehow normally distributed
2. total_reviews is right skewed and the max is > 2000 & min is almost 0
3. try log to cancel skewness

7.Experience(Years)

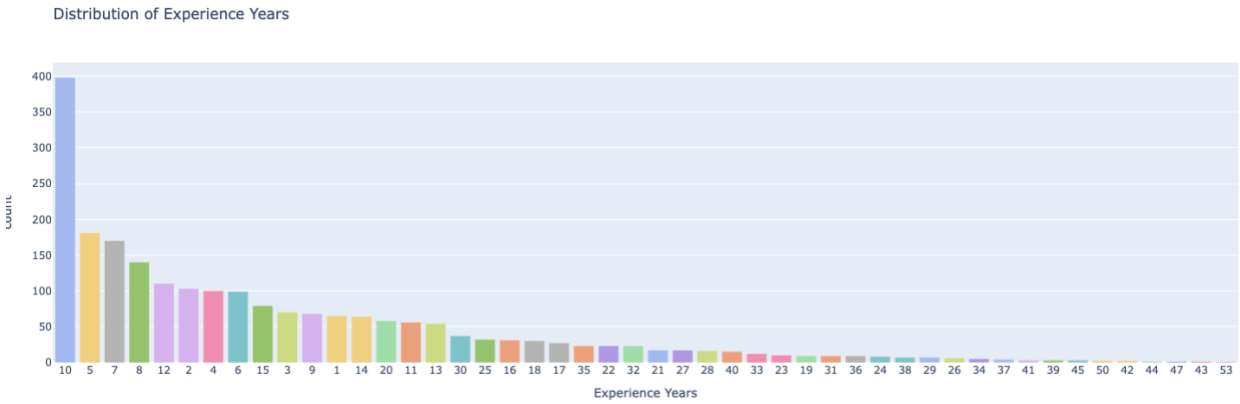
	count	mean	std	min	25%	50%	75%	max
Experience_Years	2209.0	11.660254	8.702791	1.0	6.0	10.0	14.0	53.0

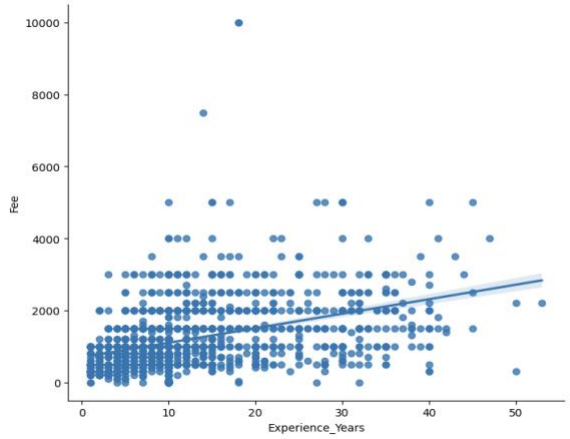
```
Minimum 1.0
Maximum 53.0
```



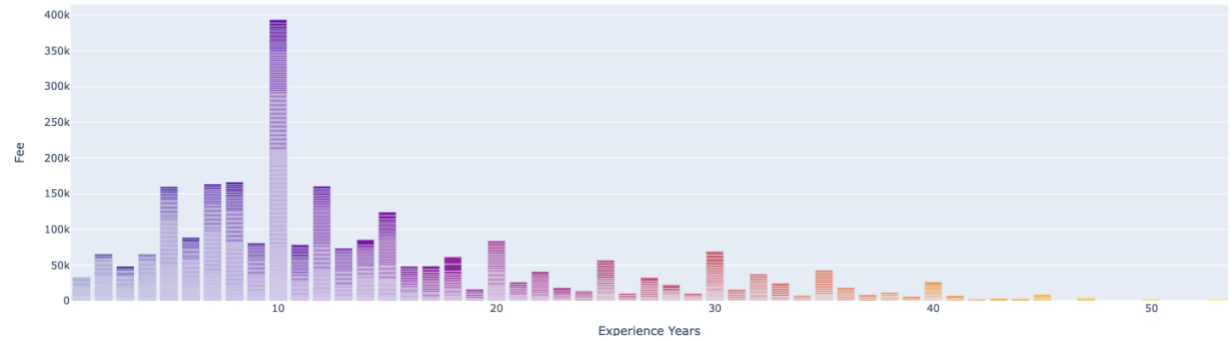


concluded that there are drs that have experience yrs 1.5 and 4.5 so will round them

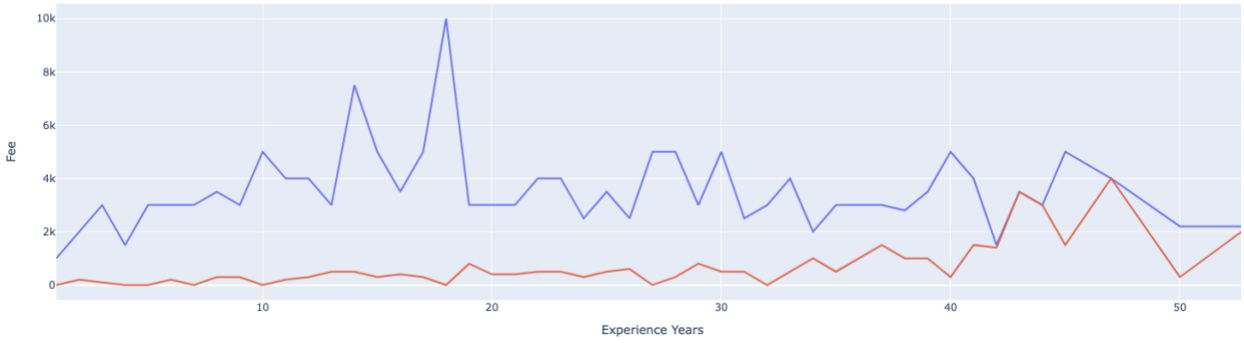




Relationship between Experience and Fee



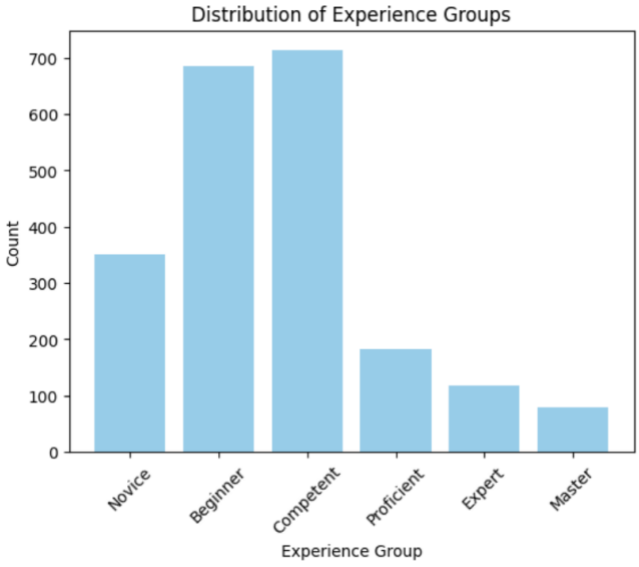
Maximum and Minimum Fees vs. Experience Years

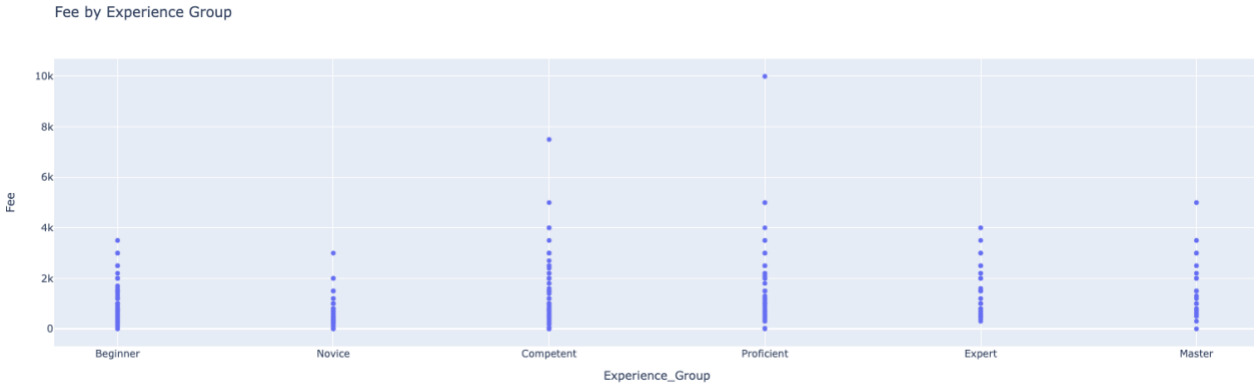


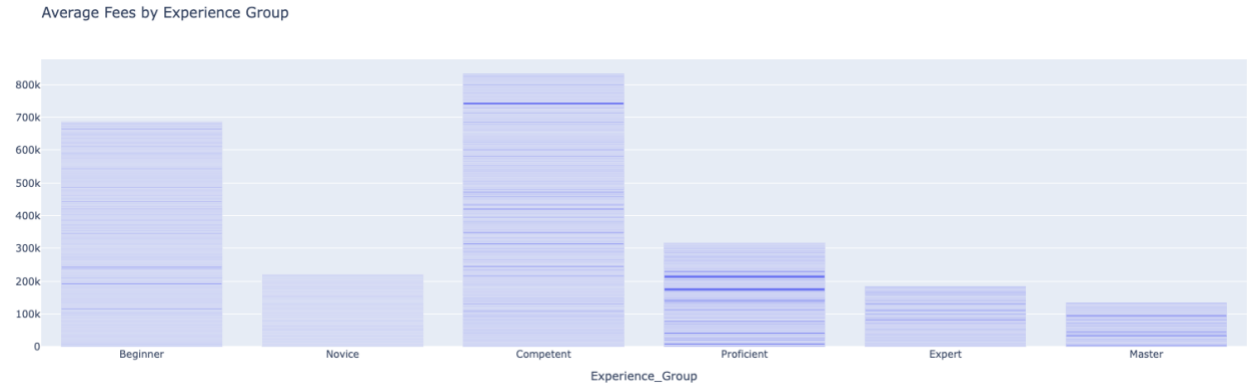
- AT max_fees : wee got that drs with experience 5 or more yrs are maximum then there is a drop at 24yrs then gets high again

- min_fees varies as exp varies to sum us fees is not that related to experience as there is a wide range in fees as it depends more on market demand , specialization ,Patient Demographics

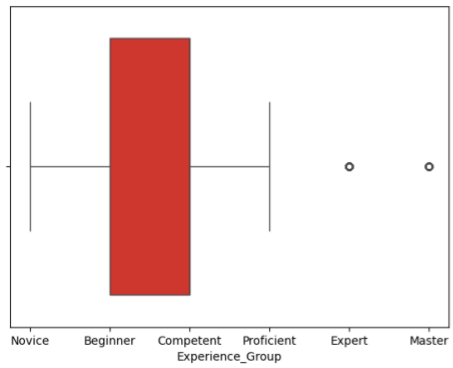
7.**Feature engnieering** (Experience_Group)

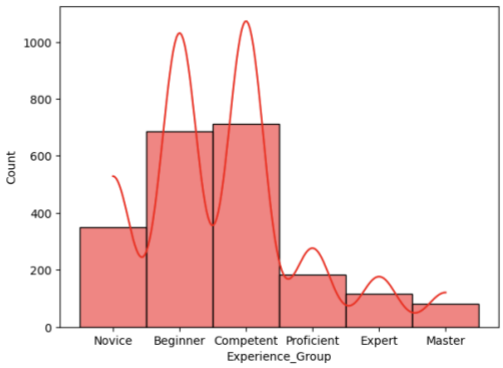






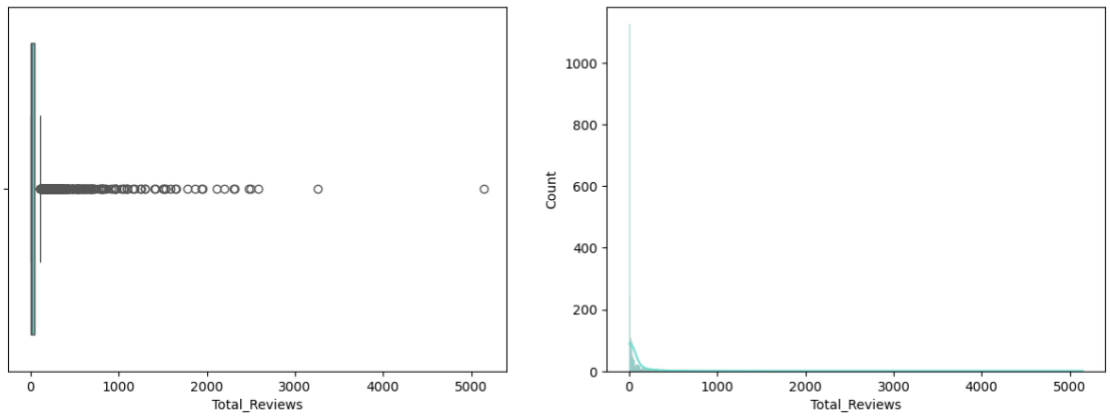

```
Minimum Novice
Maximum Master
```



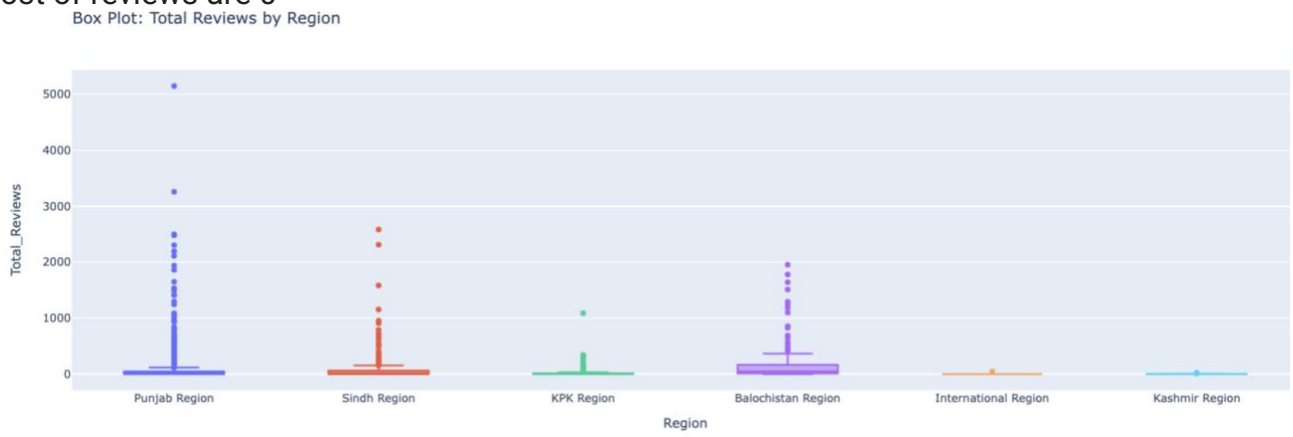


8.Total_Reviews

```
Minimum 0
Maximum 5147
```

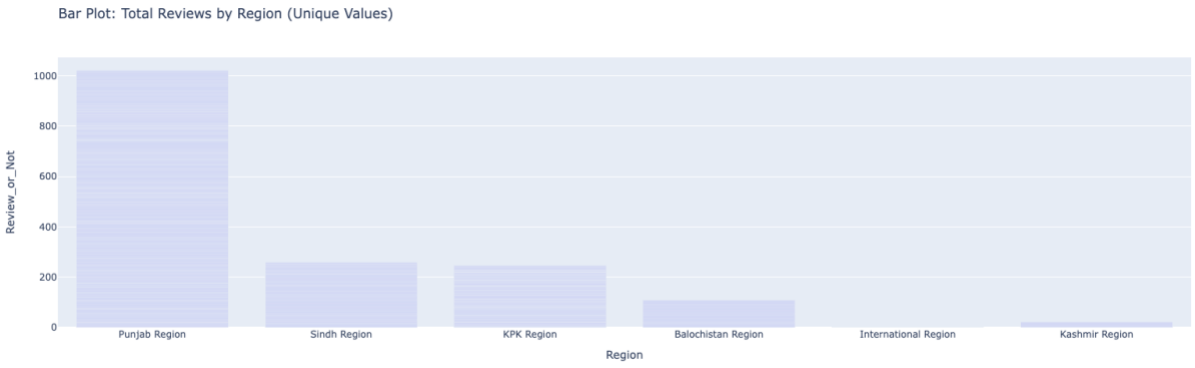


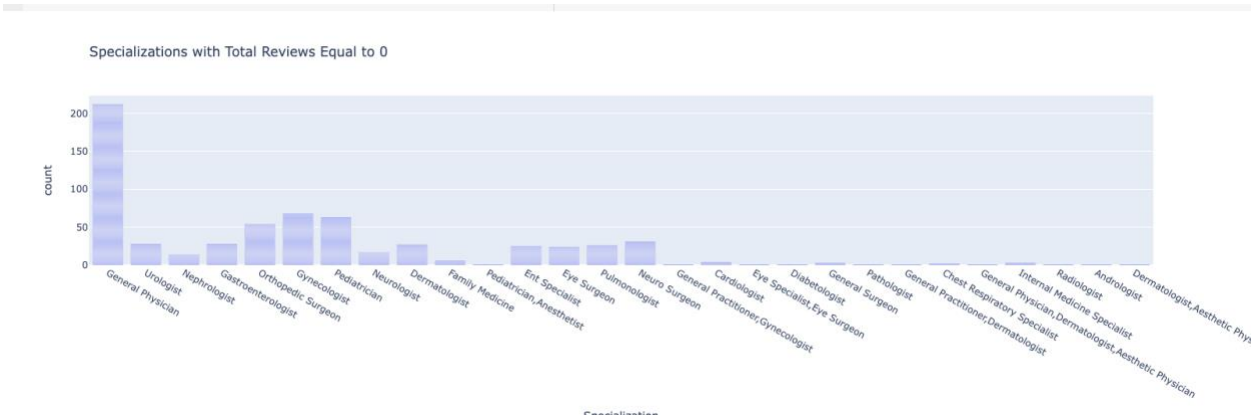
High diversion
most of reviews are 0

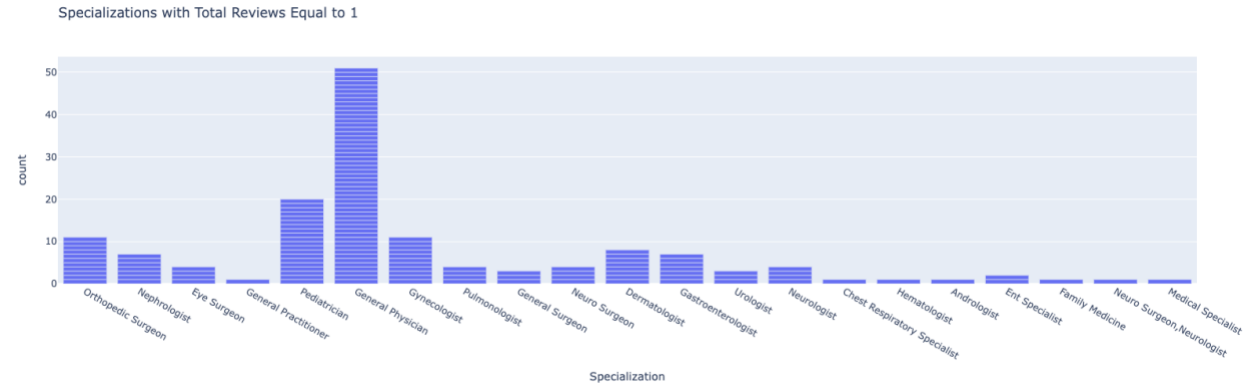


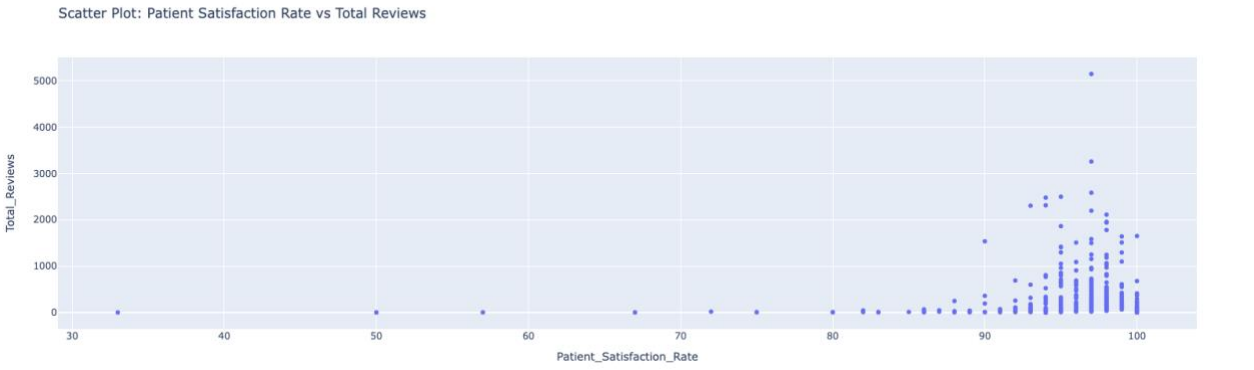
8.1Total_Reviewsfeature engnieering

```
df['Review_or_Not'] = (df['Total_Reviews'] > 0).astype(int)
```

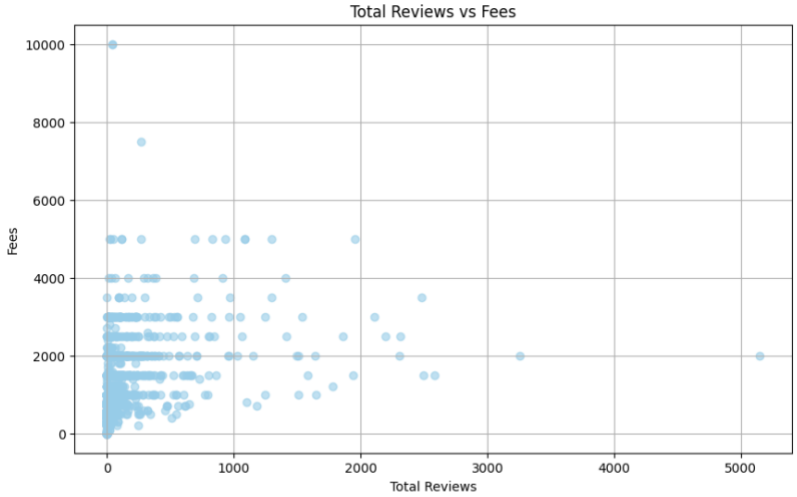








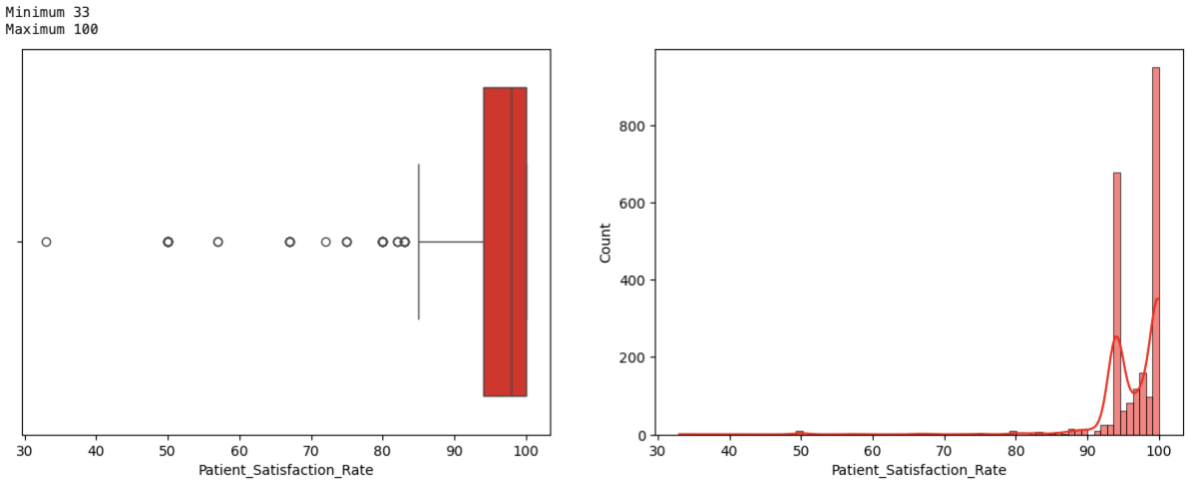
As patient satsfactiion rate increase total reviews increase too

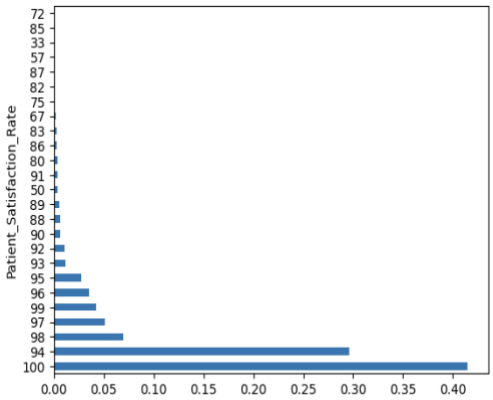


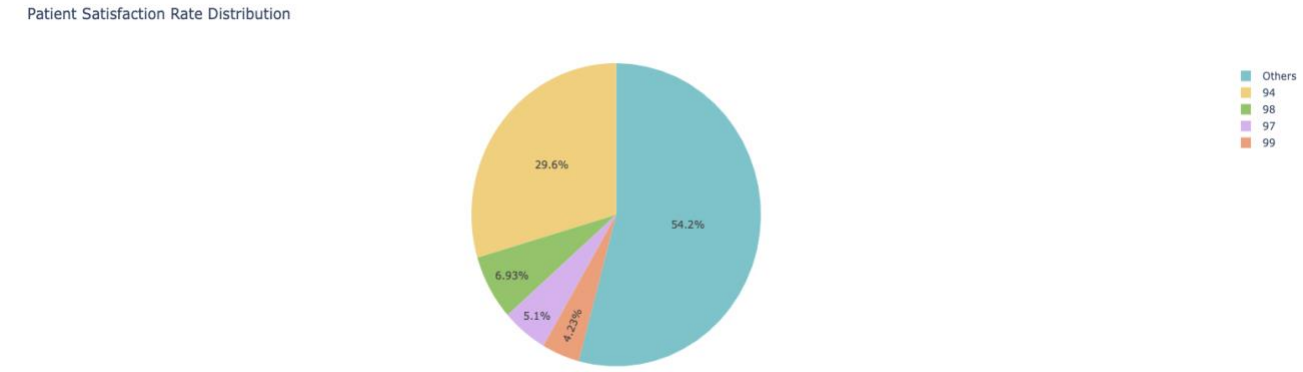

```
[ ] #total reviews doesnt correlate so much with fee
```

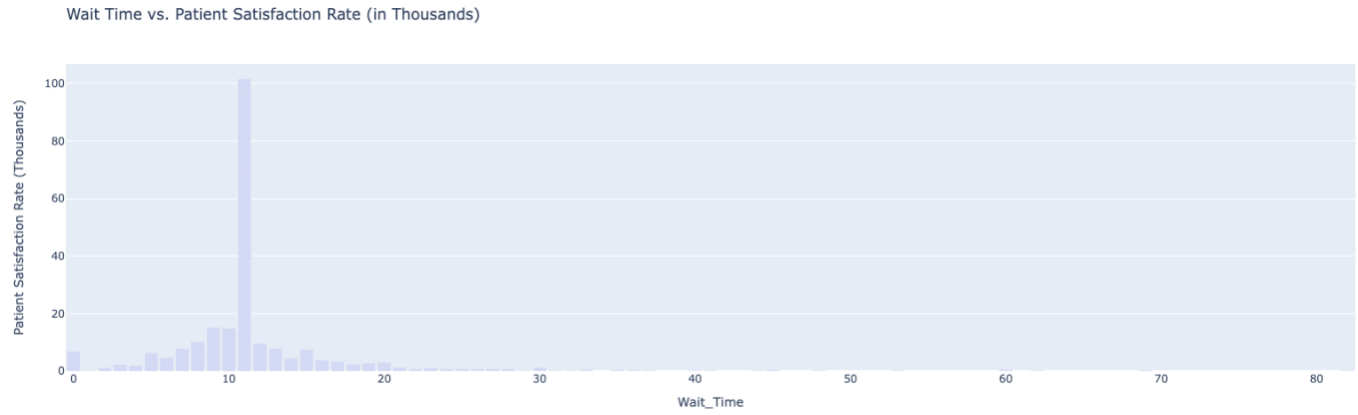
9.Patient Satisfaction Rate(%age)

Number of unique values: 25

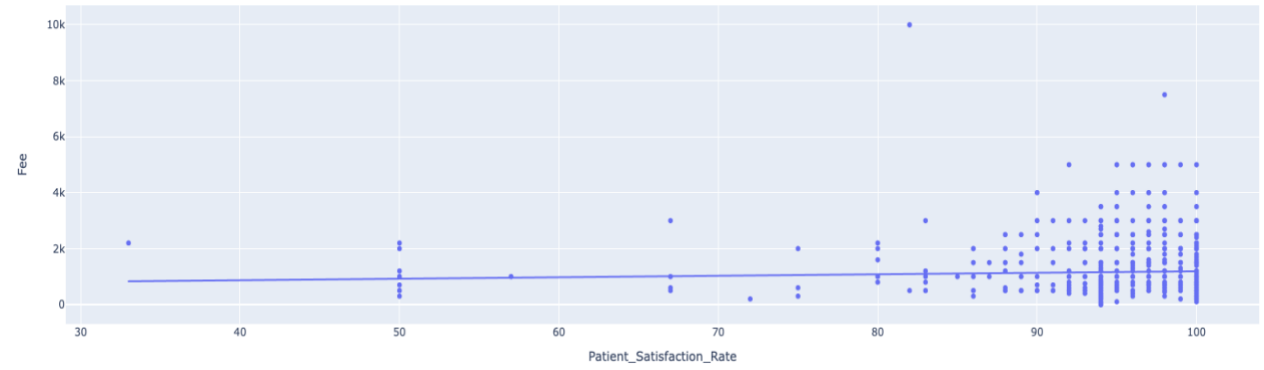






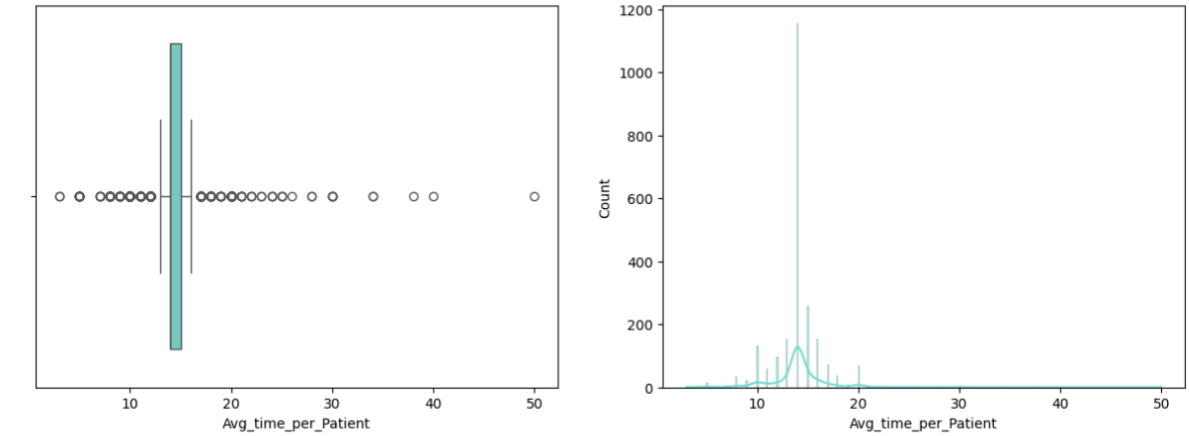


if the waiting time for the dr but the dr is good so it is worth the wait so it doesnt affect

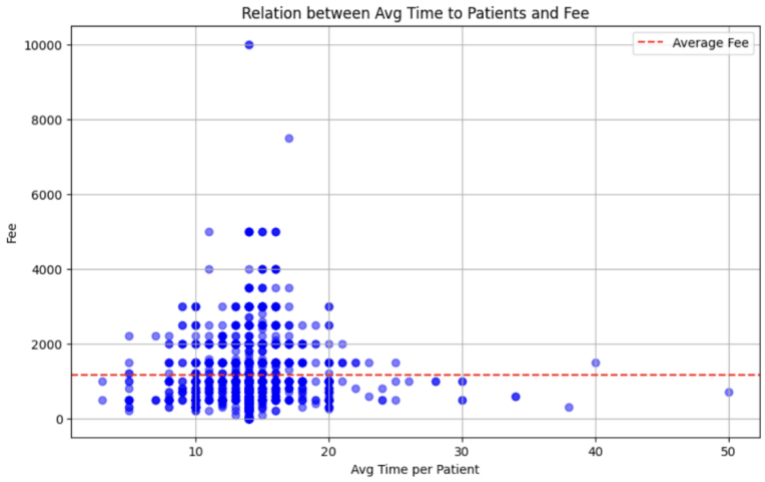


10.Avg Time to Patients(min)

```
Minimum 3
Maximum 50
```

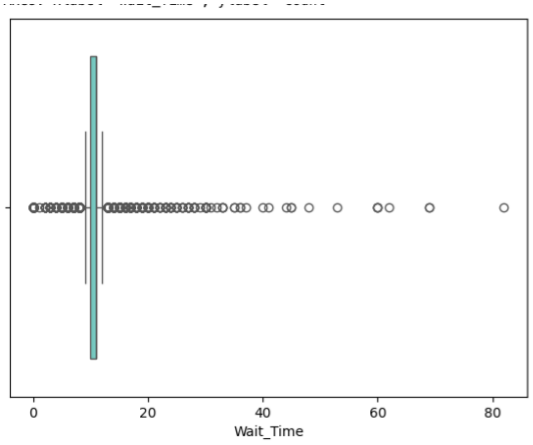


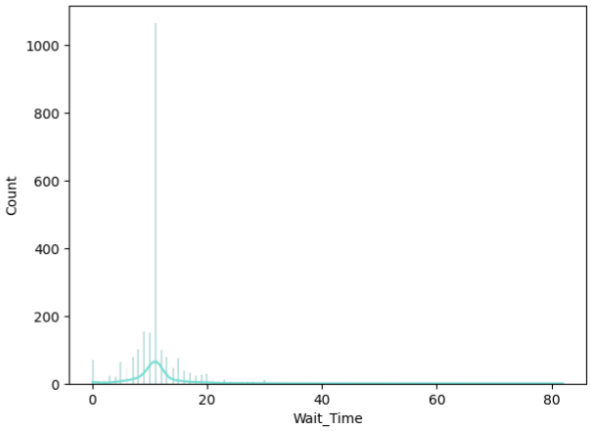
the majority of the patients spend of average less than 15 mins

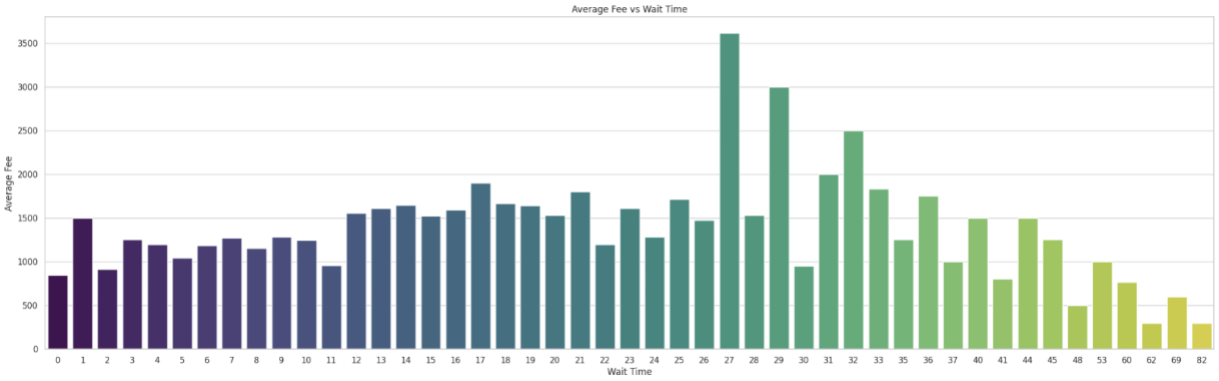


11.Wait Time(mins)

```
max val =  82
min val =  0
```







Feature Engineering (Total Time)

```
df['Total Time'] = df['Avg_time_per_Patient'] + df['Wait_Time']
```

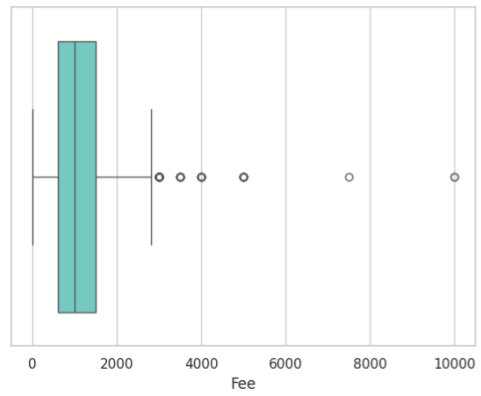
12.Feature Scaling

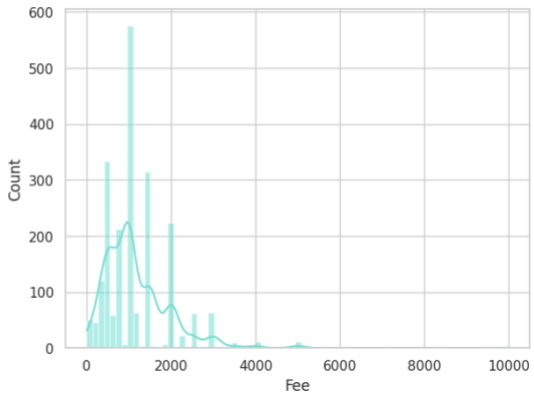
We used minmax Scaler

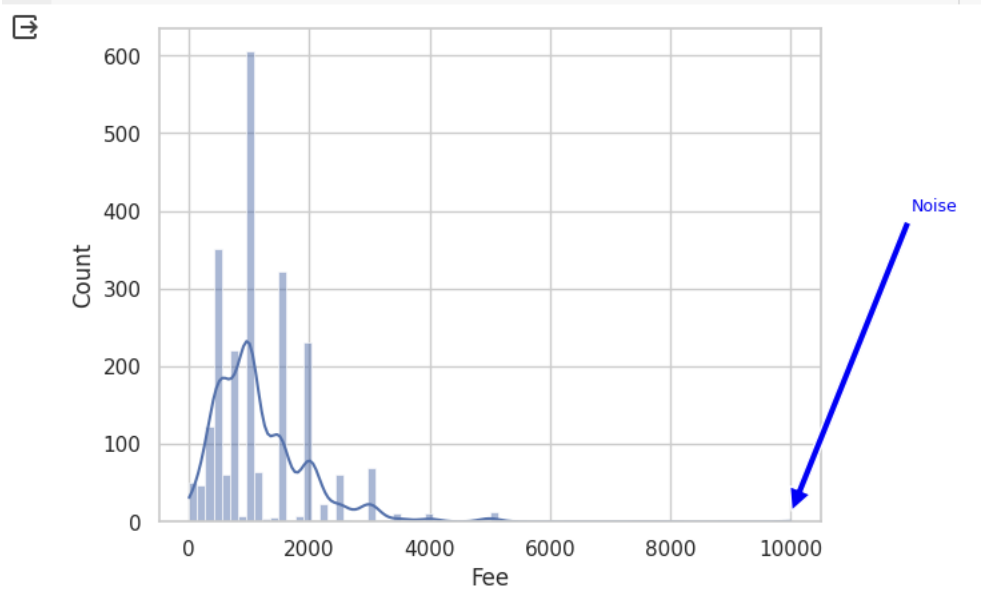
13Fee(Target)

```
print('Maximum',df['Fee'].max())
```

```
Minimum 0
Maximum 10000
```



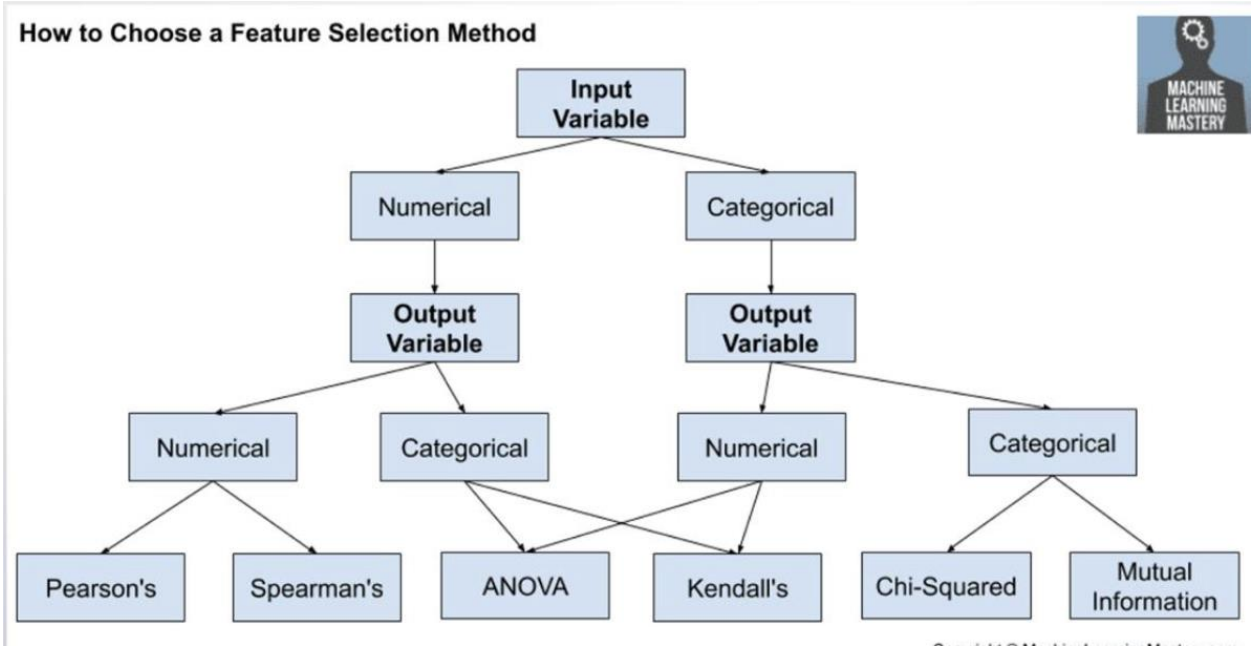




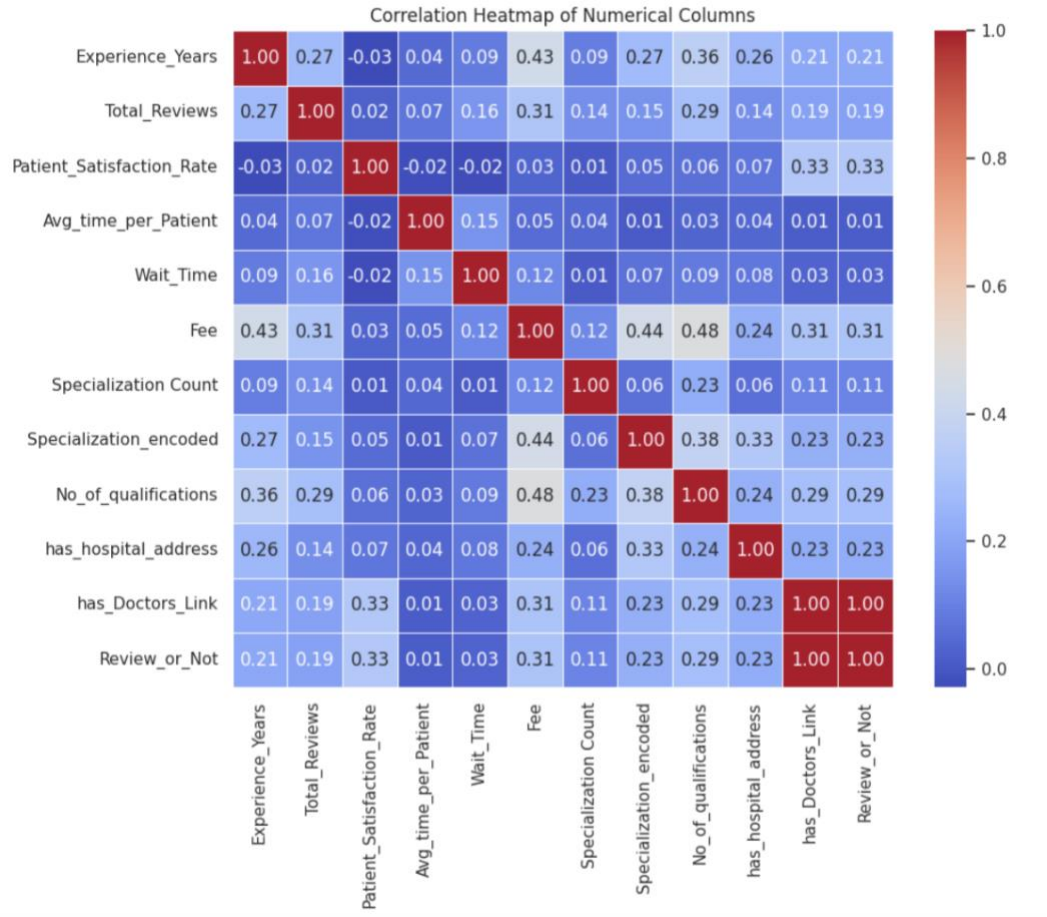
some dr dont take fees but others take 10000 majority takes 1500

14 Preprocessing

14.1 Feature Selection



14.1.1 Feature Selection (numerical values)



dropped

1.reviews_or_ not the feature Eng. col ass it gives same corr as total reviews

As If two variables are correlated, we can predict one from the other.

2 Patient_Satisfaction_Rate does not add additional information,

```
Pearson Correlation Table:
                      Feature  Pearson Correlation
0           Experience_Years             0.430926
1              Total_Reviews             0.305683
2  Patient_Satisfaction_Rate             0.030340
3                        Fee             1.000000
4       Specialization Count             0.125152
5           has_Doctors_Link             0.311233
6       has_hospital_address             0.240422
7                 Total Time             0.120838

Spearman Correlation Table:
                      Feature  Spearman Correlation
0           Experience_Years              0.487333
1              Total_Reviews              0.488316
2  Patient_Satisfaction_Rate              0.138393
3                        Fee              1.000000
4       Specialization Count              0.138437
5           has_Doctors_Link              0.361269
6       has_hospital_address              0.272006
7                 Total Time              0.148123
```

- Experience_Years has a Pearson correlation coefficient of approximately 0.431, indicating a moderate positive linear relationship with Fee

- Total_Reviews has a Pearson correlation coefficient of approximately 0.306, indicating a moderate positive linear relationship with Fee.

- Patient_Satisfaction_Rate has a Pearson correlation coefficient of approximately 0.030, indicating a weak positive linear relationship with Fee.

- Total Time has a Pearson correlation coefficient of approximately 0.121, indicating a very weak positive linear relationship with Fee.

14.1.2 Feature Selection (Categorical Data)

Cardinality Ratios

Is a way to see if we really care about this categorical column

```
Cardinality ratio for 'Doctor Name' column: 0.965142598460842
Cardinality ratio for 'City' column: 0.05251244907197827
Cardinality ratio for 'Specialization' column: 0.047080126754187414
Cardinality ratio for 'Doctor Qualification' column: 0.3739248528746039
Cardinality ratio for 'Hospital Address' column: 0.5255771842462653
Cardinality ratio for 'Doctors Link' column: 0.7093707559981892
Cardinality ratio for 'Titles' column: 0.0022634676324128564
Cardinality ratio for 'Region' column: 0.002716161158895428
```

1. **High Cardinality Ratios (> 0.5)**:
    - Columns like 'Doctor Name' and 'Hospital Address' have a high proportion of unique values relative to the dataset size.
    - High cardinality ratios may pose challenges for certain machine learning models, particularly those sensitive to high dimensionality.
2. **Moderate Cardinality Ratios (0.3 - 0.5)**:
    - The 'Doctor Qualification' column falls into this category, indicating a moderate number of unique values compared to the dataset size.
    - These columns may still be useful for encoding into numerical features or for grouping similar categories.
3. **Low Cardinality Ratios (< 0.1)**:
    - Columns such as 'City', 'Specialization', 'Titles', 'Region', 'Experience_Group', and 'Satisfaction_Category' exhibit low cardinality ratios.
    - Such columns are suitable candidates for various encoding techniques, including one-hot encoding, label encoding, or target encoding.

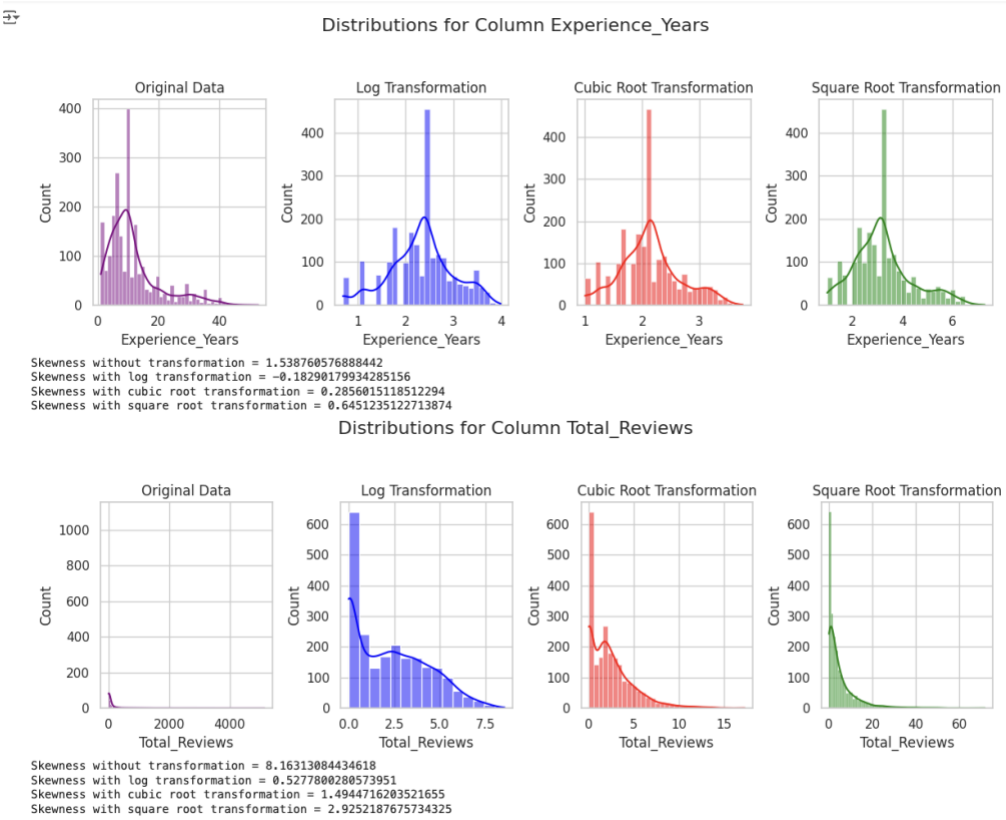
Surly we will drop Doctor Name as CR is almost 1 & Hospital Address, Doctors Link

Concatenated avg time per patient and wait time to total time so will drop too

15.1.Feature Scaling (Numerical Data)

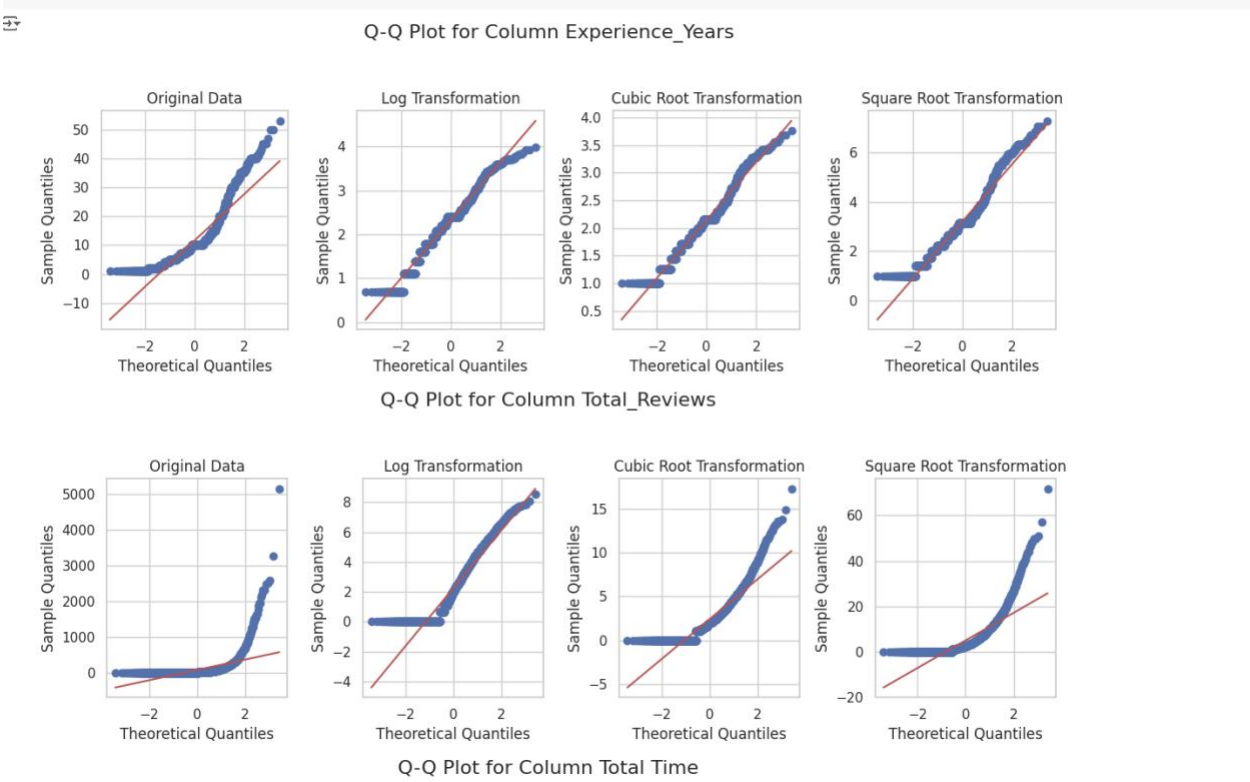
```
num_cols = ['Experience_Years','Total_Reviews','Total Time']
```

ideally, we want the skewness to be as close to zero as possible, indicating that the data is symmetrically distributed.



.

- The x-axis represents the quantiles of the theoretical distribution.
- The y-axis represents the quantiles of the dataset's distribution

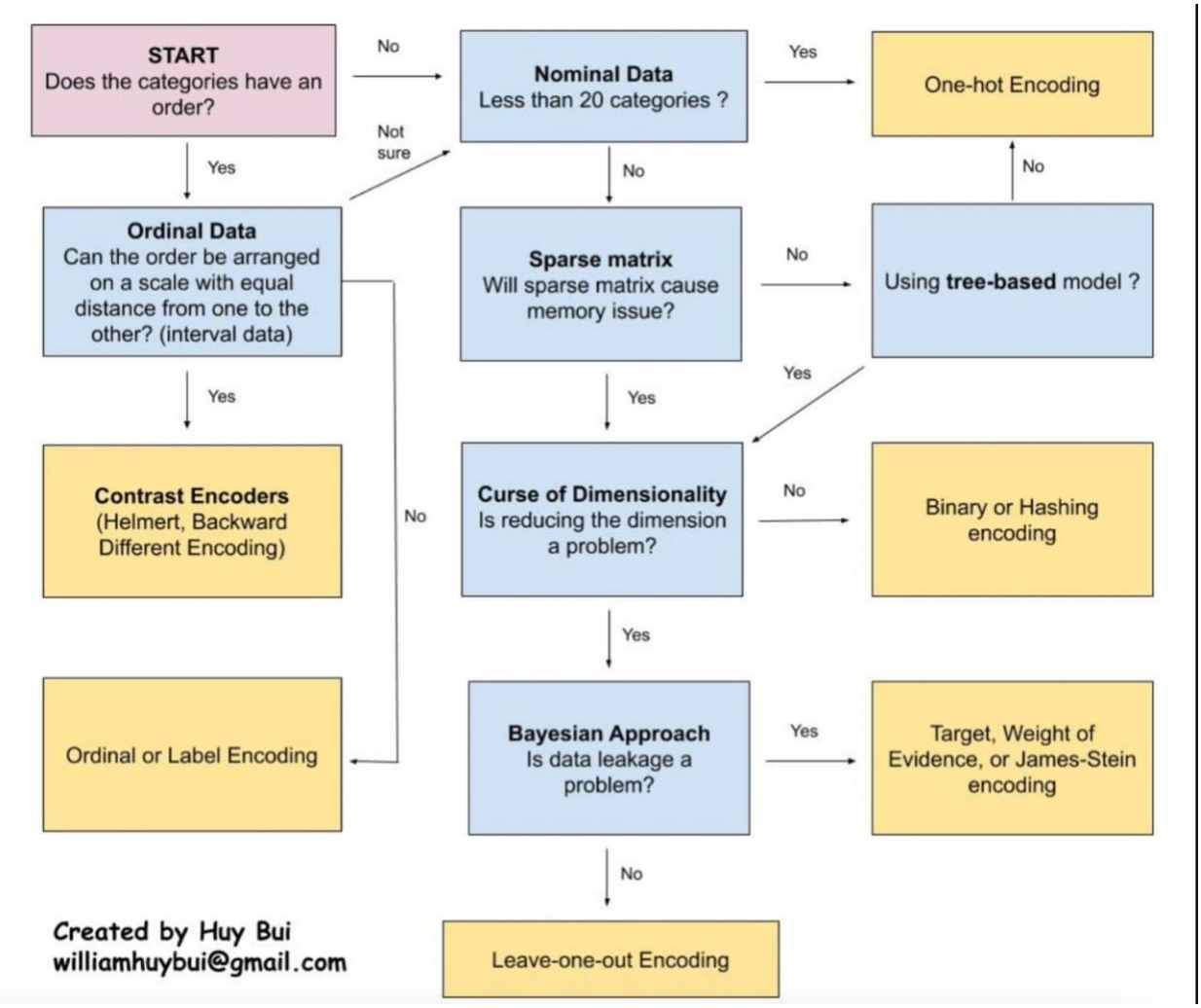


Applied log

```
otal outliers in column before transformation Experience_Years: 201, Percentage:
9.099139882299683%
-------------------------------------------------------------------------------------------------
---
Total outliers in column before transformation Total_Reviews: 331, Percentage:
14.984155726573109%
-------------------------------------------------------------------------------------------------
---
Total outliers in column before transformation Total Time: 569, Percentage: 25.758261656858306%
```

```
Total outliers in column aftar log transformation Experience_Years: 69, Percentage:
3.123585332729742%
-------------------------------------------------------------------------------------------------
--------------
Total outliers in column aftar log transformation Total_Reviews: 0, Percentage: 0.0%
-------------------------------------------------------------------------------------------------
--------------

Total outliers in column aftar log transformation Total Time: 632, Percentage:
28.610230873698505%
```

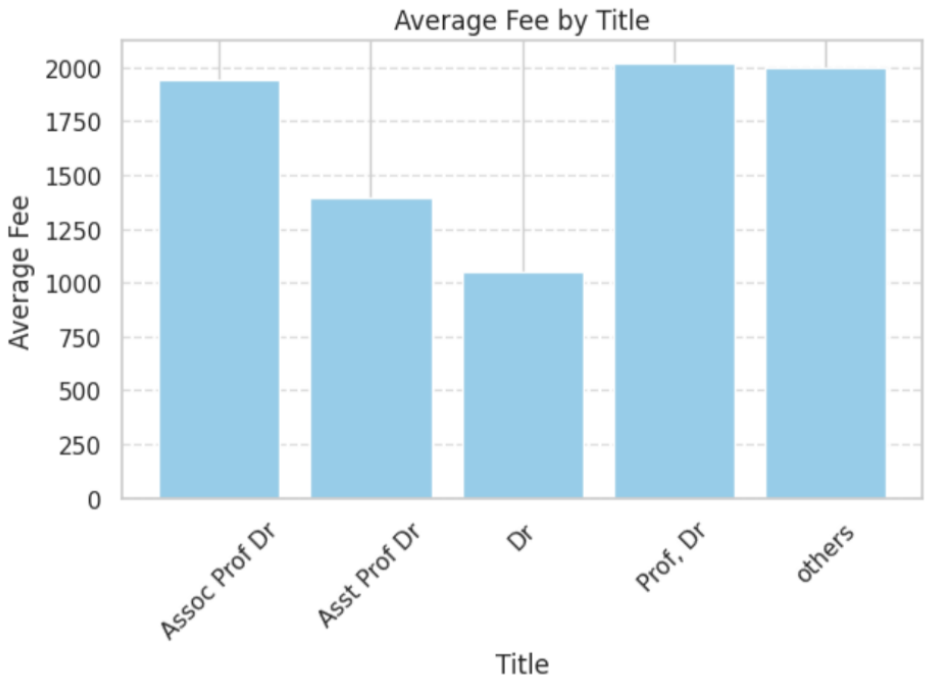
16.1Encoding data



'City', 'Specialization', 'Doctor Qualification', 'Titles'

- Title

Frequency encoder




```
Titles
Dr               1792
Asst Prof Dr      214
Prof, Dr          127
Assoc Prof Dr      74
others              2
```

```
# Target encoding for 'City'
```

- Specialization encoding

  Top 15 Onehot else others

```
Specialization Counts including 'Others':
Specialization
General Physician     392
Gynecologist          254
Pediatrician          246
Orthopedic Surgeon    192
Dermatologist         177
Others                158
Gastroenterologist    122
Pulmonologist         114
Neuro Surgeon         107
Urologist              95
Neurologist            82
Nephrologist           76
Ent Specialist         69
Eye Surgeon            61
Andrologist            47
Ophthalmologist        17
```

```
df.drop(['Specialization'], axis=1, inplace=True)
```

- `Doctor_Qualification_encoded`

  Target encoding

17Modeling and feature selection

```
Number of columns in the DataFrame: 26
//////////////////////////
Used spearsman for feature selection


//////////////////////////
```

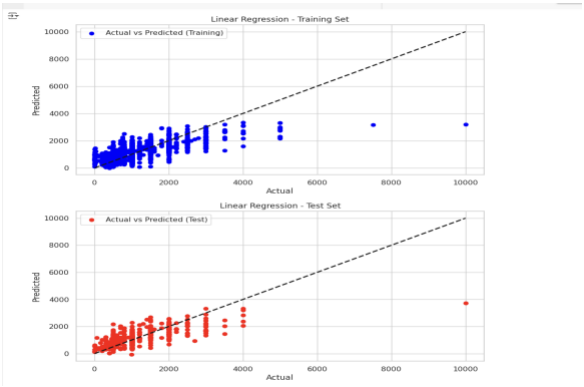
Models

```
 Linear Regression
Ridge Regression
Lasso Regression
Random Forest Regressor
AdaBoost Regressor
XGBoost Regressor
Gradient Boosting Regressor
```

#Models:

1. **Linear Regression**:

- Basic regression technique assuming a linear relationship between features and target.
- Simple to interpret and implemen
- struggled with capturing complex relationships in data.



2. **Random Forest Regressor**:

- Ensemble method averaging predictions of multiple decision trees.

- Handles both numerical and categorical data well.
- Robust to overfitting due to ensemble of trees and random feature selection.
- Can capture non-linear relationships and interactions between features.

3. **Ridge Regression**:

- Addresses multicollinearity by penalizing large coefficients.
- Helps mitigate overfitting by adding regularization term to cost function.

4. **Lasso Regression**:

- Performs feature selection by shrinking less important coefficients to zero.
- Removes less relevant features from the model.
- Can be computationally expensive for large datasets.

(we dropped it ,it gives bad accuracy with large amount of time in training)

5. **Gradient Boosting Regressor**:

- Ensemble method that sequentially combines weak learners (often decision trees).
- Designed for regression tasks, predicts numeric values.
- Improves model performance iteratively, focusing on misclassified observations.

6. **AdaBoost Regressor**:

- Ensemble learning method combining multiple weak learners.
- Focuses on improving model performance by giving more weight to misclassified observations in each iteration.

7. **XGBoost Regressior:** is a robust and versatile algorithm suitable for regression tasks, especially when dealing with large datasets. Its ability to handle complex relationships, scalability, and interpretability make it a preferred choice across various domains and applications.

Best Hyperparameters

1- **Linear Regression**: No hyperparameters to tune

2- **Ridge Regression**: Hyperparameter: `alpha` ->Represents the regularization strength.

3- **Lasso Regression**: Hyperparameter: `alpha->` Similar to Ridge regression However, Lasso also tends to shrink some coefficients all the way to zero, effectively performing feature selection.

4- **Random Forest Regressor**:

Hyperparameters: `max_depth`, `min_samples_split`, `n_estimators`

`max_depth`: Maximum depth of the trees in the forest.

`min_samples_split`: Minimum number of samples required to split an internal node.

`n_estimators`: Number of trees in the forest.

5- **AdaBoost Regressor**: Hyperparameters: `learning_rate`, `n_estimators`

`learning_rate`: Controls the contribution of each weak learner to the final prediction.`n_estimators`: Number of weak learners (decision trees) to train sequentially.

6- **XGBoost Regressor**: Hyperparameters: `learning_rate`, `max_depth`, `n_estimators`

`learning_rate`: Also known as the shrinkage parameter, controls the step size during optimization.

`max_depth`: Maximum depth of the trees in the boosting process.

`n_estimators`: Number of boosting rounds (iterations).

7- **Gradient Boosting Regressor:** Hyperparameters: `learning_rate`, `max_depth`, `n_stimators`

`learning_rate:` Represents the step size at each iteration

`max_depth`: Maximum depth of the trees in the boosting process.

`n_estimators`: Number of boosting rounds (iterations).

Used

1.randomized search cross-validation

2. We define a dictionary **models** that contains the names of the models as keys and their corresponding sklearn model objects as values. Each model has its own hyperparameters to optimize using randomized search.

3 Model Evaluation Loop

For each model, we perform the following steps:

1. Define hyperparameter distributions based on the model type.
2. Perform randomized search cross-validation to find the best hyperparameters.
3. Train the best model on the training data.
4. Evaluate the model on both training and testing data.
5. Store the results in dictionaries (**train_results** and **test_results**).

4 Spearman Correlation Coefficients

We calculate the Spearman correlation coefficients between each feature and the target variable ('Fee').

4Feature Selection

We select the top k features based on their correlation coefficients with the target variable.

5 Cross-Validation: Cross-validation is a technique used to assess how well a model generalizes to new,(folds)

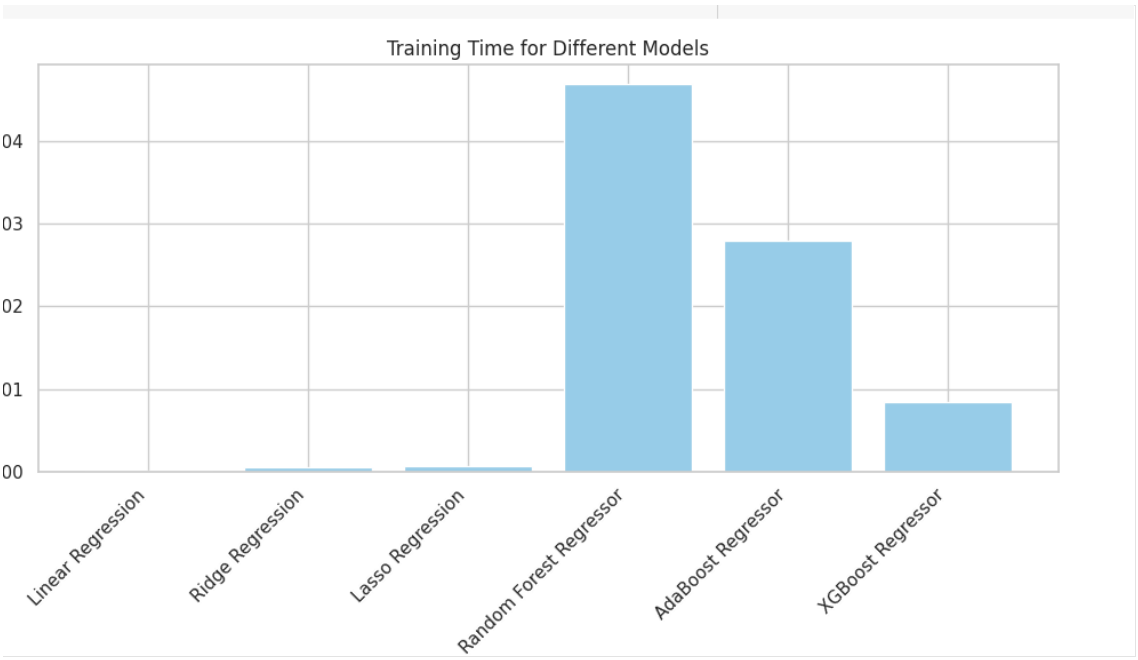
1. 6 **Select Best Hyperparameters**: Identify the hyperparameter combination with the best performance based on the evaluation metric.

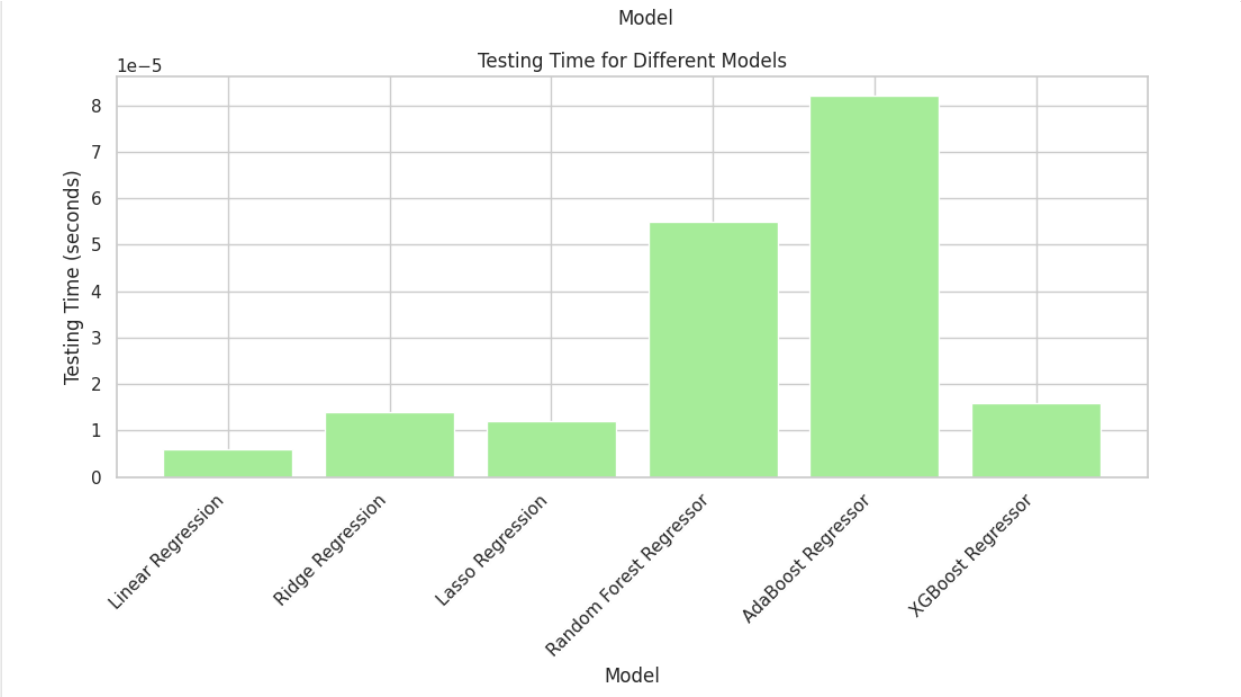
n_jobs parameter in the **RandomizedSearchCV**

make use of all available CPU cores to perform the randomized search in parallel, significantly speeding up the process

Model	Train MSE	Train RMSE	Train R2	Train Time (s)
Linear Regression	316286.195822	562.393275	0.51639	0.000104
Ridge Regression	316841.565279	562.886814	0.515541	0.000477
Lasso Regression	322891.087167	568.235063	0.506291	0.000648
Random Forest Regressor	66885.339848	258.622002	0.897731	0.046947
AdaBoost Regressor	209115.323433	457.291289	0.680257	0.027943
XGBoost Regressor	54712.793025	233.907659	0.916343	0.008357

Model	Test MSE	Test RMSE	Test R2	Test Time (s)
Linear Regression	377072.903288	614.062622	0.509471	0.000006
Ridge Regression	376730.928612	613.784106	0.509916	0.000014
Lasso Regression	380083.439004	616.509075	0.505554	0.000012
Random Forest Regressor	205149.607719	452.934441	0.733124	0.000055
AdaBoost Regressor	244964.114679	494.938496	0.681329	0.000082
XGBoost Regressor	193571.516324	439.967631	0.748185	0.000016


Training Time for Different Models
04
03
02
01
00
Linear Regression
Ridge Regression
Lasso Regression
Random Forest Regressor
AdaBoost Regressor
XGBoost Regressor


Model
Testing Time for Different Models
1e−5
Testing Time (seconds)
8
7
6
5
4
3
2
1
0
Linear Regression
Ridge Regression
Lasso Regression
Random Forest Regressor
AdaBoost Regressor
XGBoost Regressor
Model