**Doctor Consultation Fee Dataset**

**Problem Statement:**

We all have been in a situation where we have to decide about the doctor consultation. Every time we have to check the consultation fee of the doctor but we can not find the fee of all the doctors. But we can prepare a model for predicting the fee of the doctor by passing some information of the doctor.

This dataset contains the information about the doctor and their respective fee for consultation. We have to prepare a machine learning model for predicting the doctor consultation fee by passing some information about the doctors. This dataset contains the two sub dataset one for training and one for testing. The training dataset contains 7 columns and 5961 rows and testing contains 6 columns and 1987 rows. Following are the variables of this dataset:-

* **Qualification:** This column contains the information about the qualification of the doctor
* **Experience:** Experience of doctor in years
* **Rating:** Rating of doctor given by past patients out of 100
* **Place:** Location of the doctor clinic
* **Profile:** Profile of the doctor
* **Miscellaneous\_Info:** Other information about the doctor
* **Fees:** Consultation fee charged by the doctor

There are 7 columns in this dataset. The target or y variable in this dataset is Fees and others are x variable. Y variable is dependent variable and x variable is independent variable which means the value of y is always dependent on the value of x.

This is a Regression problem because the target variable in this dataset is a continues variable. Machine learning model for continues variable can be prepare by the regression process. In this dataset the target variable Fees is a continues variable so we use the regression process for this dataset.

**Data Analysis:**

First we read the csv file into our jupyter notebook using pd.read\_csv method and check the 5 rows of the data using df.head() function. After checking the data we used the df.info() function for getting the information about the datatype of this dataset. We found that all independent variable are object type and fees are integer type. We check the all qualification type of the doctor and also check the different different location of the doctor clinic. We check the unique values for the profile of the doctor.

**EDA:**

EDA is known as the exploratory data analysis. In this analysis it is shown that which variable has the most impact on the target variable and how a column is performing in predicting the target variable. First we separate the qualification the qualification column and find the different different qualification of the doctor.

After separating the qualification now we have to convert the experience column in the integer datatype by removing the year experience and change the data type using df.astype() function. Rating is also a object type in this dataset we have to convert it into integer by removing the % sign and using the df.astype() function. Now we have to deal with place column there are many place in this dataset with the locality as well. We have to remove the locality of the city and we are going to use only city column for predicting the fees of doctor. We split the place column and find the city of the doctor. Finally we check the unique value for the profile of the doctor there are 7 type of profile in this dataset.

Now we have to deal with the null values of this dataset. We can check the null values of the dataset by using the df.isnull().sum() function. After checking the null values we find that rating and miscellaneous\_info have the null values. We have drop the miscellaneouss\_info column from both of the dataset training and testing and the null values of the Rating column is filed by the mean value of the rating using df.fillna() function. Now our dataset does not contain any null values.

After ploting the graphs we found that most common fee charged by the doctor is 100 rupee and Delhi is charged by the highest fee. According to the profile of the doctor ENT specialist and Dermatologist charged the maximum fees. After anlysing the data we found that young doctors charged more than the old doctors.

**Pre – Processing Pipeline:**

After getting the proper knowledge of dataset now we forward to the data cleaning and data preprocessing.

First we drop the miscellaneouss\_info column as it is not playing an important role in predicting the Doctor consultation Fees. We drop the miscellaneouss\_info column by using df.drop() function.

Now we have to deal with the categorical variable of the dataset. For predicting the target variable using machine learning model data should be in numeric form. We use the LabelEncoder for converting categorical variable into numeric form. We can import labelencoder from siktlearn and can be use as df(‘col’)=labelencoder.fit\_transform(df(‘col’)). In this dataset Qualification, Place and Profile are categorical variables. We convert them into numeric form by using labelencoder.

Now we have to check the relation between variables of the dataset by using the df.corr() function. We found that these variable are related with each other and fees is also relative to these variable. Qualification is the most affected column in this dataset with fees.

Now we will separate the data into x\_train and y\_train and x\_test variable. X\_train variable includes all the independent variable of the dataset for training and y\_train variable includes only dependent variable that is Fees in this dataset.

X\_test includes the values for the testing the datset.

We use the standardscaler function for making the independent value scaler. StandardScaler can be imported from sklearn.preprocessing. we use standardscaler by using this code :- x=StandardScaler.fit\_transform(x\_train).

**Building Machine Learning Model:**

We are using the regression model for this dataset as target variable of this dataset is a continues variable. First we import all the necessary library for the machine learning model from the sikitlearn. After importing all the libraries we first split the data into train test split.

**Linear Regression:**

Linear regression is a regression model of machine learning for preparing a machine learning model of a regression problem. First we import the linear regression from sikitlean and create a variable lr=LinearRegression(). Now we have to fit the model by using lr.fit(x\_train,y\_train) and we can predict the fees value by using the lr.predict(x\_test) these prediction are for the values of the y\_train. We save these prediction and use this model in future.

**Lasso Regression:**

Lasso regression is a regression model of machine learning for preparing a machine learning model of a regression problem. First we import the lasso regression from sikitlean and create a variable lsr=LassoRegressor(). We use the function GridSearchCV for getting the best parameter of lasso regression. Now we have to fit the model by using lsr.fit(x\_train,y\_train) and we can predict the fees value by using the lsr.predict(x\_test) these prediction are for the values of the y\_train. We save these prediction and use this model in future.

**KNeighborsRegressor:**

KNeighbors regression is a regression model of machine learning for preparing a machine learning model of a regression problem. First we import the Kneighbors regression from sikitlean.neighbors and create a variable knn=KneighborsRegressor(). We use the function GridSearchCV for getting the best parameter of kneighbors regressor. Now we have to fit the model by using knn.fit(x\_train,y\_train) and we can predict the fees value by using the knn.predict(x\_test) these prediction are for the values of the y\_train. We save these prediction and use this model in future.

**GradientBoostingRegressor:**

GradientBoostingRegressor is a regression model of machine learning for preparing a machine learning model of a regression problem. First we import the GradientBoostingRegressor from sikitlean.ensemble and create a variable gb= GradientBoostingRegressor (). We use the function GridSearchCV for getting the best parameter of GradientBoostingRegressor. Now we have to fit the model by using gb.fit(x\_train,y\_train) and we can predict the fees value by using the gb.predict(x\_test) these prediction are for the values of the y\_train. We save these prediction and use this model in future.

**RandomForestRegressor:**

RandomForestRegressor is a regression model of machine learning for preparing a machine learning model of a regression problem. First we import the RandomForestRegressor from sikitlean.ensemble and create a variable rf= RandomForestRegressor (). We use the function GridSearchCV for getting the best parameter of RandomForestRegressor. Now we have to fit the model by using rf.fit(x\_train,y\_train) and we can predict the fees value by using the rf.predict(x\_test) these prediction are for the values of the y\_train. We save these prediction and use this model in future.

**Concluding Remarks:**

In simple words we can say that this dataset contains the information about the doctors and by using this dataset we can predict the consultation fee of the doctor by just passing the some information. This model is very useful when we have to check the consultation fee of the doctor.