House Price Prediction

**Problem Statement:**

The housing market is one of the most critical sectors of the economy, and accurately predicting house prices is essential for various stakeholders, including homebuyers, sellers, real estate agents, and financial institutions. However, predicting house prices accurately is challenging due to the complex interplay of numerous factors, including location, size, amenities, economic indicators, and market trends.

**Data Set:**

Kaggle Competition-> House Price – Advance Regression Technique (in CSV)

**Data Loading:**

Using pandas (pd.read())

**Data Analysis**

Now we start analying data, so for processing and analying we join both train and test data so we don’t have to perfom opteration twicw it can we done in single time

Now we do Exploratory Data Analysis (EDA) (is an analysis approach that identifies general patterns in the data)in our data so after using info() command we can see there are my feature who have more than 50% NULL Value so now we have to handle this values

Lets list out the numerical and object type data

**Data Cleaning**Methods for cleaning:

* Ignore missing values row/delete row (sometimes loss valueable data)
* Fill missing value manually
* Global constant adding
* Measure of central tendency(mean, midian, mode)
* Measure of central tendency for each class
* Most probable value(ML algo) (most effective and time consumeing and costly)

**Handleing missing value**

There two algo knn and naïve baiyes doesn’t care about missing value while traing the model as it don’t take missing value in trainng

But in sklearn library they fails

**Drop column/fearutee**

So how we decide which column/feature to drop, so for this we create one threshold value for we can say a we set a limit range this will vary in project to project according to the domain knowledge

* First I check data with more than 50% of missing values

[Alley 93.216855

PoolQC 99.657417

Fence 80.438506

MiscFeature 96.402878]  
then we goto the data decription and get to know that this missing values actulay represetnt that it don’t have that feature in this house hence no need to drop the feature just add some constant NA(represent no)

* Now with 20% to 50%
* Then 5% to 20% hence with all this observation we conclude not to drop and feature and with this we can definetly say that there is no need to remove fetaue with less than 5% null value

Just don’t drop feature only based on threshold before droping always check out the data description for proper decision

**Imputating missing value**

First we check the missing feature in categorial data: missing feature 23

Then for int: missing feature 0

Then for float: missing feature 12

Now to start with imputation we take one by one feature to impute

Lets take **MSZoning** is a categorical data(object) so ussally or say normal to impute categorical data we take most repeting value from the data or in statistically we take mode value of coulum as in central tendency .hence for this feature RL is mode value

Here we store all imputating value in some varivble then in some file like json so during deployment if any null value came then it can be directly hanlded with the json file

we never make changes directly to data base we take copy of it and then we manipulate

now lets take Alley: as we can see alley is a type of road acces to property hence instead of droping we just add **NA** as it means no acces to road also store that constant in variable firstly in original data NA values was set as NULL meaning was same but now we convert it to object value NA   
  
now LotFrontage : so as we know when data is in numercal form we normally take mean or median for imputating missing value. Always have a habit of visualization fearutre before diging into it to get proper knowledge, we visualize data using box and hist graph and we can see data has outliers and right skew hence because of this we take median value instead of mean   
  
now utilities: also a object data types hence mode

now MasVnrType 0.822199, MasVnrArea 0.787941: in this we can see missing values are almost similar so there might be chace to have common pattern in both hecne we use heat map to check

now basements: in this there 5 categorical and 6 numerical feature given in dataset in that we see some pattern like when categorical is null means numerical is 0 vice versa  
  
same way we do for all other feature and impute them

**Feature Transformation:**

**Numerical to categorical-** Here we are going to convert some numerical feature into categorical because its numeric sequence is not varying the house price as its just a mapped relation for eg feature like, MSSubClass as its just a mapped for object data, year built etc . here feature like overallQual have numeric feature but its numeric values will afftect house price as it’s a rating value hence we will not chnge it to category form.

Mosold feature Is a month given as 1(jan) 2(feb).. etc so with python we directly convert it into months like January, February…etc and rest other we just convert them in string

**Categotical to Numerical:** Here we first change some ordinal category feature to numerical like feature have objects like excellent, good, fair or poor type of data to numerical data using ordinal encoding

Now we gonna convery nominal object to numerical feature using one hot encoding to do this we use pandas function get\_dummies to convert object feature to one hot encoding and we only take k-1 classes of each feature (as if all k-1 class value is 0 then its automicallty the last class is there)

**Split data for training and testing**

Lets divide df\_encode into x\_train, y\_train and x\_test according to original data sizes

**Working on Feature Scaling**there are multiple method for feature scaling but here we are using standardization method using sklearn librabry

**Model Selection**As we know that is project is baed on Advanc regression technique hecne we will dhcek out some model and check its accuracy

Usally now we use x\_train and y\_train to tain model and x\_test and y\_test to get the score but here we don’t have y\_test so normaly people devide train data into 80 % and 20% data for training and testing but instead we apply cross validation on tain data after that the model which have high accuracy so we used it for model training  
Aftet testing we found that ['GradientBoostingRegressor', [0.868993820347623]], has the highest accuracy score hence we gonna use this

**Model Training**Gradient Boosting Regressor model is trained and test data is sent to it to get the desire output

Model is stored in pickle file