Capstone Project-1

1. **Title of the Project :- Property Price Prediction**

2. **Brief on the Project :-**

Property Prices fluctuate a lot in real world. The motive behind a property price prediction project revolves around addressing the challenges and inefficiencies in real estate markets. It also helps us to capture opportunities in this market at the right time. This can be a win-win situation for all, the Buyers, the Sellers and the Investors as well.

- Helps **Buyers** to determine if the property is priced fairly or not.
- Helps **Sellers** to set competitive prices for their properties
- Helps **Investors** in identifying undervalued properties or evaluation potential return on investment.

It may also help in automating the process of price estimation, which otherwise requires manual analysis and consultation with experts. Even enables us to analyse thousands of properties simultaneously.

a. Helps lenders and financial institutions assess the risk of granting loans or mortgages based on accurate property valuation. Investors can mitigate risks by identifying overpriced properties.

b. Enhances property listing platforms by providing price suggestions for new listings. Even allows for adjustments based on real-time data, similar to pricing strategies in other industries like travel.

3. Project Deliverables:

In this project, we are going to create a model for Property Prices. Here, we will analyse data, find insights and proceed accordingly. We were provided data from the Iot Academy Team. Here, we need to predict the prices of the properties with different locations and parameters.

- Our model will analyse the data and find insights from within.
- This model will identify patterns, trends and correlation of independent variables with our target variable.
- When our model is ready, it will be able to predict the prices of the properties with different parameters.

Under supervised machine learning, we will employ the Decision Tree technique. Here in decision tree, a tree like structure is formed and the leaf nodes give the prices for the particular property with given parameters. This model gets the most accurate when we take the optimal value of ccp alpha, which saves the model to get either overfitted or underfitted.

We calculate Errors in numeric terms unlike in case of Classification problem and our main aim is to reduce it to predict the prices with the most amount of accuracy.

We will also look for VIF values for each variable to see if any variable is very highly co-related with other independent variables or not. Here, the correlation between independent variables is not high and even the data doesn't seem to contain only high valued or only lowed valued properties. So, we can directly carry out EDA without bothering much on these issues,

We hoped that Linear Regression would give the best accuracy but Decision Tree came out with the best accuract after applying Hyperparameter Tuning and it made our testing accuracy to jump from only around 50% to cross 90% accuracy mark. This shows the true strength of Decision tree after applying HYperparameter tuning.

4. Resources:

- **Data set source:**

  The source of the dataset is Kaggle.

- **Software:**

  Jupyter Notebook is used to create the Machine Learning Algorithm.

5. Individual Details:

a) Name: Manoj Sancheti

b) E-mail ID: manojsancheti4848@gmail.com

c) Contact Details: 8753952285

6. Milestones:

- **Define a Problem:**

Unpredictable price fluctuations in Property Prices, lack of knowledge of perfect buying or selling price of properties has made it really tough for both the parties to figure out the correct selling or buying prices of properties according to that time's market rates which creates a sense of doubt in both the parties.

Also, it helps financial institutions to figure out rates of multiple properties simultaneously and also helps in reducing their risks by analyzing the optimal pricing and hence increasing their profits.

- **Get the Data:**

  The Dataset is a csv file as attached.

  House Price.csv

- **Explore and Pre-Process Data:**

  This Dataset contains 29451 rows and 12 columns.

  I chose Jupyter Notebook to create my Supervised Machine Learning Model.

- **Create Features:**

  Following are the features for the given Data set.

  - POSTED_BY : Category marking who has listed the property, either Builder, Dealer or Owner.
  - UNDER_CONSTRUCTION: Either the property is fully ready to be delivered or it is under construction. 0 means it is ready and 1 means it is under construction.
  - RERA: If the property is RERA approved or not. 0 means not approved and 1 means approved.
  - BHK_NO: Number of rooms in a property. It has values ranging from 1-20.
  - BHK_OR_RK: It signifies the type of property, if it is Bedroom_Hall_Kitchen or Room_Kitchen.
  - SQUARE_FT: It tells about the total area of the property in square feet.
  - READYTOMOVE: Category marking if the flat or property is ready to move in or not. 0 means not ready and 1 means ready to move.
  - RESALE: Category marking if the property is fresh or is up for resale. 0 means fresh property and 1 means resale property.
  - ADDRESS: It has the full address of the property, its locality, area and city.

- LONGITUDE: It contains the Logitude coordinates of the property.
- LATITUDE: It contains the Latitude coordinates of the property.

- **EDA:**

  While performing the EDA, following are our observations:

  - None of the values in the entire dataset have NULL value.
  - From the describe function, we get to know that most of the houses in our dataset are under-construction and mostly having 2 BHK facility accommodation
  - There are 401 duplicate rows in the dataset, so we remove them from our dataset.
  - After showcasing the chart for RESALE column , we get to know that most of the properties in our dataset are for resale.
  - Then, we split the ADRESS column in two different columns, one comprising the locality and the other comprising the city of the property naming them as LOCALITY and CITY.
  - Then , we remove the ADDRESS column as it is of no use now and also LOCALITY column as the data for each locality is very less or in many cases only in range of 1-2.
  - Then we plot the heatmap of the numeric variables conveying about the correlation of different variables with different variables. Here, we get to know that there is a small correlation between our Target variable TARGET(PRICE_IN_LACS) and SQUARE_FT and other variables are either have very slight positive relation or negative relation with our target variable.
  - We use LabelEncoder on POSTED_BY, BHK_OR_RK, LOCALITY and CITY, to convert categorical data to numeric data, so we can apply different operations on them and use those variables to train data.
  - Then, we plot boxplot to see if there are outliers in the dataset or not. We get some outliers in various variables but there can be properties present with such parameters, so we try to reduce the skewness of the data now.

- Then ,we check for the skewness in the data of different variales, and here we get to know that SQUARE_FT variable has very high skewnesss even more than 100 and also a very high negative skewness in LATITUDE. So, we apply Logrithemic Transformation to SQUARE_FT variable and YeoJohnson Transformation to LATITUDE variable to reduce skewness.

- **Create Model:**

  We split the model in the ratio of 70:90 before creating our model, to test the accuracy of our model.

  Then used various Regression Algorithms on our dataset to see which gives us the best accuracy link:

  - Linear Regression
  - Decision Tree
  - Random Forest
  - KNN
  - Gradient Boosting Regressor
  - Xboost

- **Model Evaluation:**

  a) Performance Metrics are evaluated for each model to determine which model will give better predictions in testing

Sl. No.	Model Name	Training R2 Value	Testing R2 Value
1	Linear Regression	0.512	0.485
2	Decision Tree with Hyperparamenter Tuning	0.977	0.935
3	Random Forest	0.982	0.925
4	KNN	0.871	
5	Gradient Boosting Regressor	0.975	0.939
6	Xboost	0.855	0.857

7. Report Writing:

The target variable or the dependent variable i.e., Price of the properties is classified as a Regression problem where our model predict the prices of the properties according to the following paramenters taken into account.

The dataset consists of 29451 rows and 12 columns representing the locality, city or RERA certified and other parameters. There is no single null value present in the dataset. Plotting the parameters on chart and performing describe function, we get very useful insights as most of the houses are of 2 BHK and and most of the house prices are under 1Cr rupees and very less houses are there with premium pricing.

There is a slight positive correlation between pricing of the house and its total area, prices increases with increase in area of the house and no much correlation is there of pricing with other independent variables.

The data has many outliers in dependent variable, so there is very less accuracy found in Linear Regression. In Decision tree also, the situation was same which changed only after choosing the best ccp alpha value of the tree and using preferred Hyperparameter Tuning to increase accuracy.

In this project, we explored 6 different model to see which model performs best and reduce our error and increase accuracy.

From the above table, we can conclude that among all the models we used to test the accuracy, Decision Tree performed the best of all after uing hyperparamenter tuning on our model to increase its testing accuracy and it really worked out successfully .

The worst model among all is Linear Regression as its r2 value doesn’t even cross 50%.

And the best model among all is , Decision Tree which also showed very less accuracy before without tuning our data, but tuning method just turned the table down and came out with the best R2 value.

So, considering all the factors and the accuracy of our different models, I recommend to use Decision Tree Regressor Model to further increase its accuracy and R2 value as it works best with its best ccp_alpha value and the prescribed different values of the our variables we get after tuning our model with different tuning methods.