FINAL PROJECT REPORT (DA EXPLORERS) :

HOUSE PRICE PREDICTION

Arjun Avadhani(PES2UG20CS901), Disha Singh(PES2UG20CS906) , Vaishnavi R Bhat(PES2UG20CS922)

1 Introduction and background – what is the problem area? Why is it important? What is the specific problem you seek to solve?

Our dataset comprises of around 4700 rows which has various past information about various flats, houses, apartments etc and its different features.
On the basis of these feature we will attempt to make an accurate price prediction.

The problem with which we are dealing is highly important as the rental industry is an evergreen field and an effective methodology to predict a future price will always be beneficial.

We aim to devise an efficient , unique and new method to predict future house prices.

Brief review of only the most relevant predecessor work; what limitations have you identified that you seek to address in your work? What are the assumptions you have made about the data/ problem area or the scope of the problem you seek to solve?

The most relevant predecessor work we did related to our project was a literature survey. Here are the details off the same :

As part of our literature survey, we reviewed 6 research papers which are relatively similar to the aim of our problem statement.

Papers we did survey on ;

1. Predicting the rental value of houses in Tanzania, Uganda and Malawi : Evaluations of hedonic pricing and ML approaches.
2. Housing Price Prediction via Improved ML techniques
3. Monitoring house rental price based on social media
4. Modelling house rent in atlanta metropolitan area using textual information and deep learning.
5. House Price Prediction using Random Forest Machine Learning Technique
6. House Price Prediction Using LSTM

Models used in each paper ;

Paper-1 : Ridge regression, LASSO, Tree Regression, Bagging, Random Forest and Boosting.

Paper-2 : XGBoost, LightGBM, Random Forest Technique.

Paper-3 : Hedonic regression model with six different machine learning algorithms

Paper-4 : A combination of linear non linear and ensemble algorithms are used

Paper-5 : Random Forest ML

Paper-6 : Long Short Term Memory

We can apply most of the above models to our dataset

LightGBM gave the best reults during our experimentation and was hence employed for dataset but even that had the following limitations :

1. Overfits the data to an extent.
2. Not as compatible or efficient with smaller datasets.

We have not made any assumptions of the data.

Proposed solution – an overview of the various components of your solution

1. The first and foremost preprocessing work we did was to ensure the chosen dataset has no null values.

2. The next step was to perform EDA.

We plotted the size attribute in a boxplot and inferred that our dataset has outliers. We also plotted two bar graphs (rent vs city and rent vs BHK) from which we inferred that Mumbai has the highest rent and that the rent progressively increases with increase in BHK respectively. We also plotted a bar graph (Rent vs furnishing status which showed that furnished houses have higher rent).

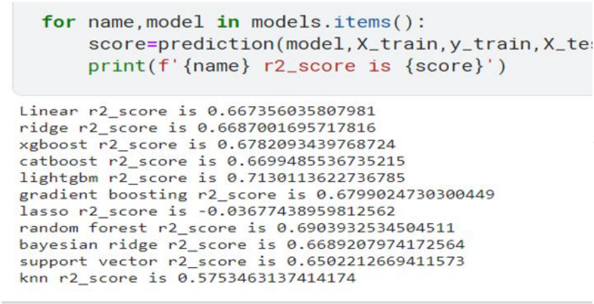
3. The next step was to plot a correlation heat map. This gives us information about linear and non linear relationships between variables. It essentially gives us a measure of how strongly correlated two variables are in a two dimensional plane. The correlation heat map was formed after dropping some features to reduce the probability of multi collinearity.

We employed multiple algorithms to get the most accurate results. Random Forest Regression, XGBoost, LightGBM etc were intensively tuned by the function GridSearchCV provided by scikit-learn to get accurate results. However, this increases the run time.

Random Forest was prone to overfitting.

During analysis, we observe that Lightgbm is giving the r2_score of mo**re** than 0.71 which is the highest amongst all the tested models.

Hence,We Select the LightGBM. Light gradient boosting technique gives an RMSE of 0.358 and this RMSE value is much lesser than most other Machine Learning algorithms .



```
for name,model in models.items():
    score=prediction(model,X_train,y_train,X_te
    print(f'{name} r2_score is {score}')
```

```
Linear r2_score is 0.667356035807981
ridge r2_score is 0.6687001695717816
xgboost r2_score is 0.6782093439768724
catboost r2_score is 0.6699485536735215
lightgbm r2_score is 0.7130113622736785
gradient boosting r2_score is 0.6799024730300449
lasso r2_score is -0.03677438959812562
random forest r2_score is 0.6903932534504511
bayesian ridge r2_score is 0.6689207974172564
support vector r2_score is 0.6502212669411573
knn r2_score is 0.5753463137414174
```

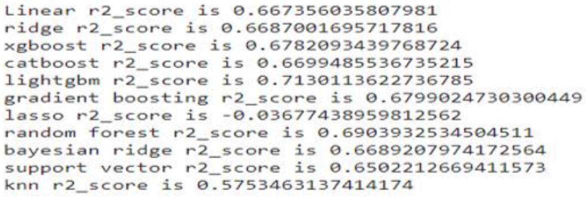
<u>Evaluation :</u>

Evaluation metrics used in our project are R squared and Root mean squared error .

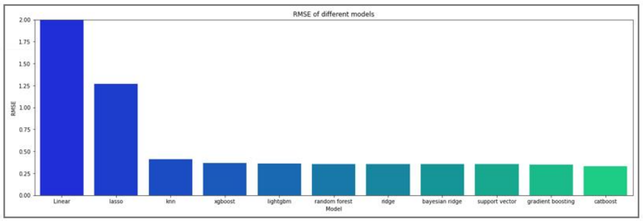
Using the Light gbm method we have found that:

1. The R squared value is greater than 0.71 which is higher compared to all the other tested models.

2. The RMSE value is found to be 0.358 which is lesser than all the other tested models



```
Linear r2_score is 0.667356035807981
ridge r2_score is 0.6687001695717816
xgboost r2_score is 0.6782093439768724
catboost r2_score is 0.6699485536735215
lightgbm r2_score is 0.7130113622736785
gradient boosting r2_score is 0.6799024730300449
lasso r2_score is -0.03677438959812562
random forest r2_score is 0.6903932534504511
bayesian ridge r2_score is 0.6689207974172564
support vector r2_score is 0.6502212669411573
knn r2_score is 0.5753463137414174
```



Answer to questions from peer review

1. How are you dealing with outliers?

We are applying log transformations which normalize the data set.

2. What are the disadvantages of your model?

The model tends to overfit the data to an extent and is neither very compatible nor efficient with smaller datasets.

3. What are the hyper parameters of your method?

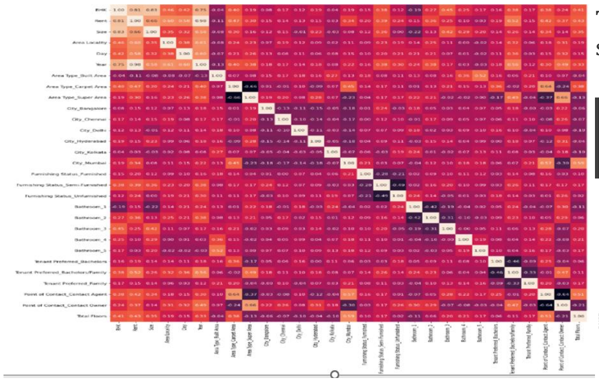
Max depth, min data in leaf, bagging fraction, feature fraction, early-stopping round, lambda , Min-gain-to-split, Max-cat-group

Detailed explanation of each component and what you have

The first component of our code was the EDA which has been explained in detail before.

The most significant component of our feature engineering stage is the correlation heat map.

The correlation heat map was essential in reducing the probability of multi collinearity.



The major component of our code is the model we selected after careful evaluation and deliberation : The Light Gradient Boosting Method.

LightGBM is a fast, distributed, high-performance gradient boosting framework that is based on the decision tree learning algorithm and is used for ranking, classification and many other machine learning tasks.

Even after tuning the parameters, there is no much difference in the accuracy and the voting model. Therefore, this is the best model we can get.

Light gradient boosting has a better performance compared to other Machine Learning algorithms and is comparitively more efficient .

LightGBM splits the tree leaf-wise as opposed to other boosting algorithms that grow tree level-wise. It chooses the leaf with maximum delta loss to grow. Since the leaf is fixed, the leaf-wise algorithm has lower loss compared to the level-wise algorithm.

The main parameters of the Light GBM method are:

num_leaves: controls complexity of the tree model.

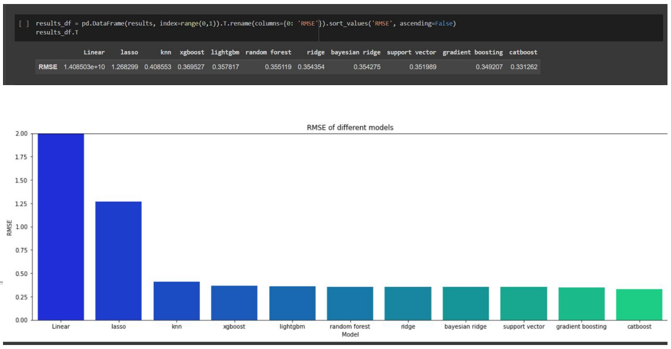
min_data_in_leaf: avoids over fitting in a leaf wise tree.

max_depth: limits the tree depth explicitly.

Experimental results and a detailed explanation of all the insights you have gained into the data

Here are some of the experimental results :

The RMSE values of all the tested models is shown in the following snippets:

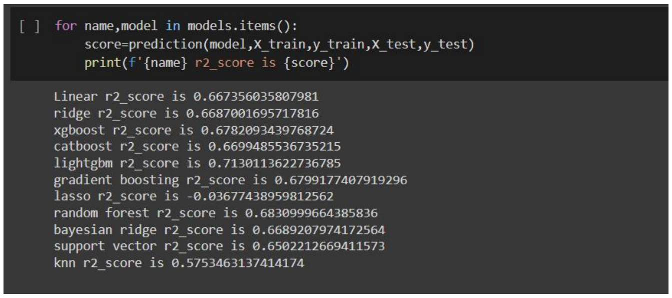


```
[ ] results_df = pd.DataFrame(results, index=range(0,1)).T.rename(columns={0: 'RMSE'}).sort_values('RMSE', ascending=False)
    results_df.T
```

```
              Linear     lasso       knn   xgboost  lightgbm  random forest     ridge  bayesian ridge  support vector  gradient boosting  catboost
RMSE  1.408503e+10  1.268299  0.408553  0.369527  0.357817       0.355119  0.354354        0.354275        0.351989           0.349207  0.331262
```

We have used RMSE because Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.This tells us heuristically that RMSE can be thought of as some kind of (normalized) distance between the vector of predicted values and the vector of observed values which clearly gives us the error difference required.

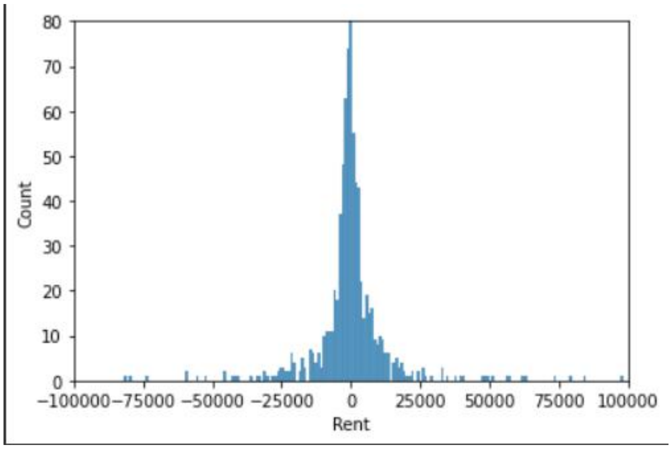
Here are the r2 scores for all the models:



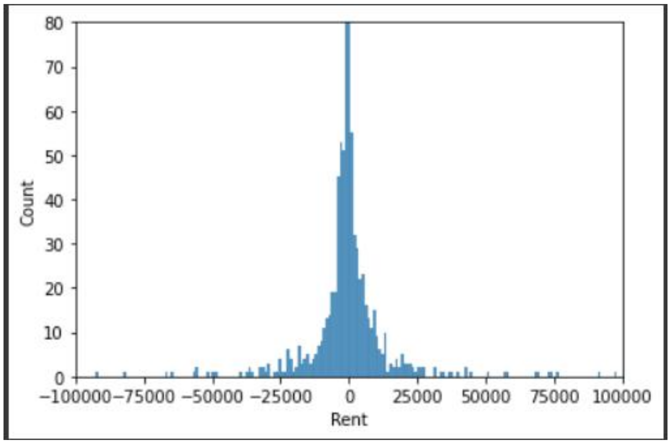
```
[ ] for name,model in models.items():
        score=prediction(model,X_train,y_train,X_test,y_test)
        print(f'{name} r2_score is {score}')

    Linear r2_score is 0.667356035807981
    ridge r2_score is 0.6687001695717816
    xgboost r2_score is 0.6782093439768724
    catboost r2_score is 0.6699485536735215
    lightgbm r2_score is 0.7130113622736785
    gradient boosting r2_score is 0.6799177407919296
    lasso r2_score is -0.03677438959812562
    random forest r2_score is 0.6830999664385836
    bayesian ridge r2_score is 0.6689207974172564
    support vector r2_score is 0.6502212669411573
    knn r2_score is 0.5753463137414174
```

Error of the predicted Y with the true Y :

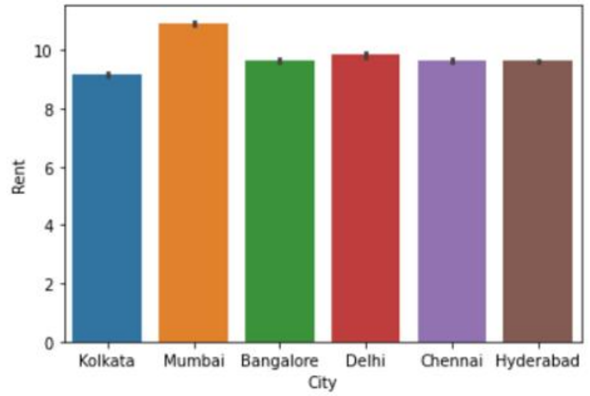


The same error after tuning the parameters :

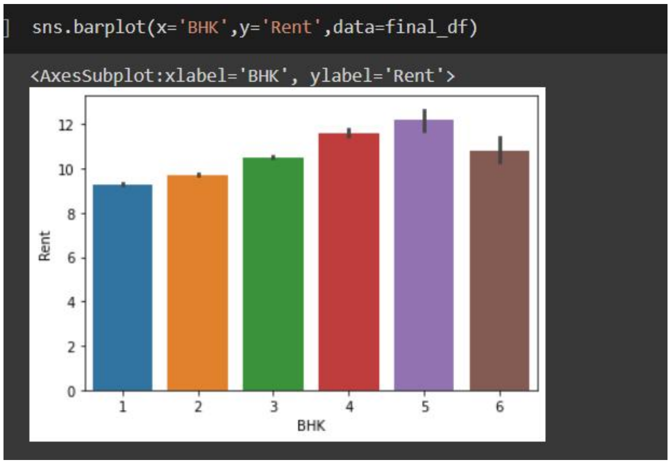


Even after tuning the parameters, there is no much differnence in the accuracy and the voting model. Therefore, this is the best model we can get.

These are some of the insights we have gained from the data :



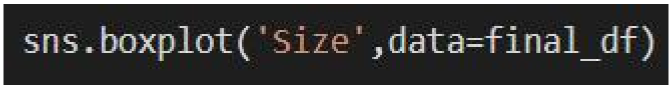
1.As per the code execution, Mumbai clearly has the highest rent.



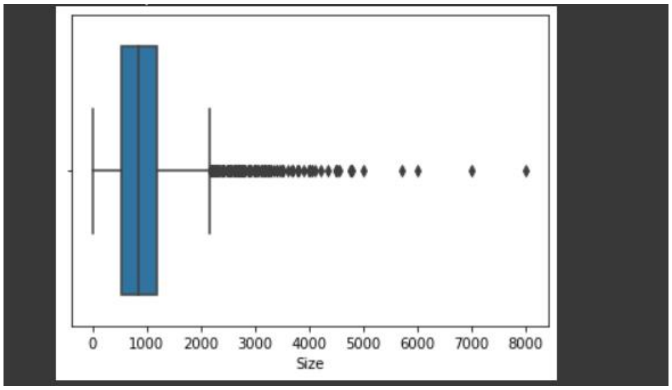
```
sns.barplot(x='BHK',y='Rent',data=final_df)
```

```
<AxesSubplot:xlabel='BHK', ylabel='Rent'>
```

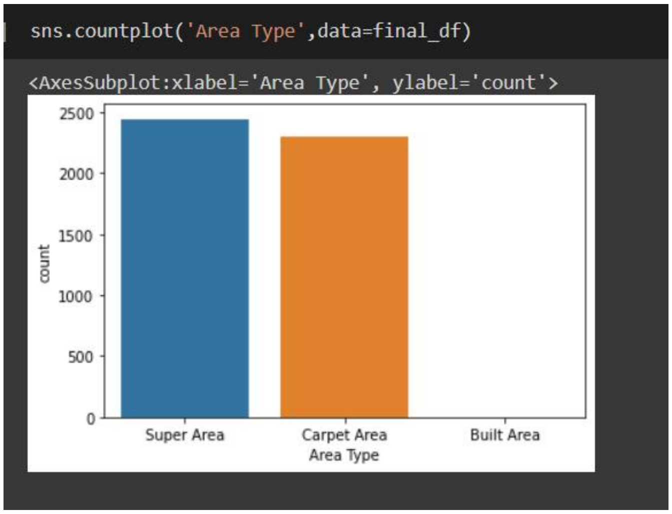
2.Clearly, 5 BHK houses have the highest rent. The increase in rent is directly proportional to the increase in BHK barring 6 BHK houses.



```
sns.boxplot('Size',data=final_df)
```



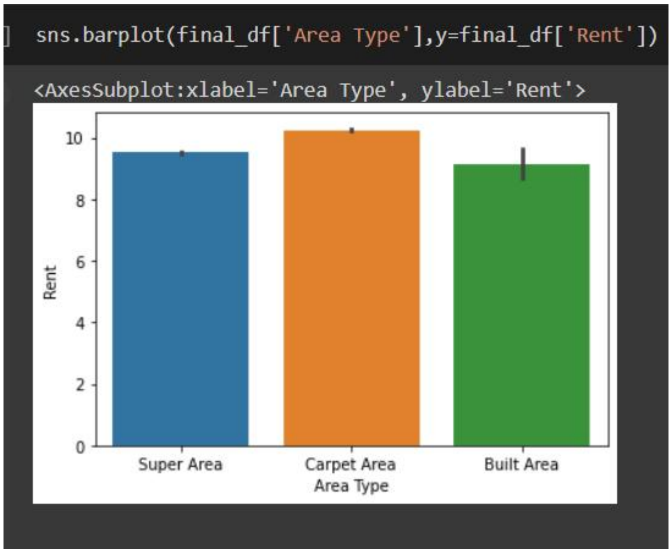
3.Greater the size of the house, greater is the rent. But there are a lot of outliers found while inferring this.



```
sns.countplot('Area Type',data=final_df)
```

```
<AxesSubplot:xlabel='Area Type', ylabel='count'>
```

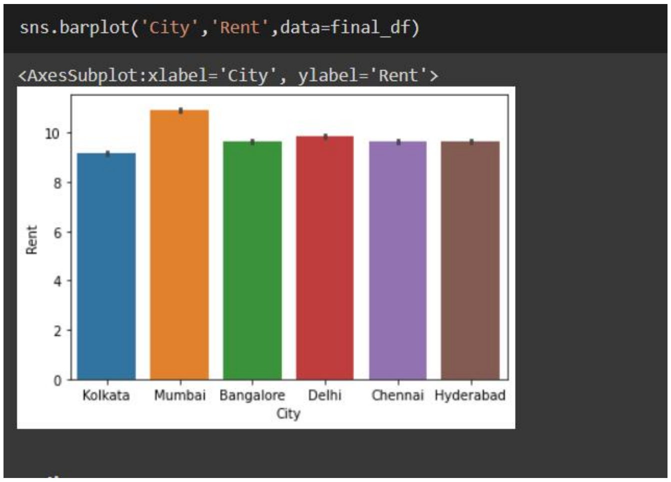
4. The number of super areas is higher than the rest and Built Area contains only two houses for Rent in the Dataset.



```
sns.barplot(final_df['Area Type'],y=final_df['Rent'])
```

```
<AxesSubplot:xlabel='Area Type', ylabel='Rent'>
```

5. Built areas are the cheapest while carpet areas are the costliest.



```
sns.barplot('City','Rent',data=final_df)
```

```
<AxesSubplot:xlabel='City', ylabel='Rent'>
```

6. The average rent of Mumbai is the highest.

The model we have selected works extremely well with larger datasets but fails when the dataset is smaller. However, this is not a concern in this case as the selected dataset is very large .

Conclusions

House rent prediction finds multiple applications in the real world . The value of rent depends on multiple factors and all of them need to be factored in , manually doing this is close to impossible and hence using MI algorithms we can efficiently predict the rent of any house .
Many MI algorithms have already been implemented in the past to predict the house rent values with promising accuracy values .

However, Light Gradient Boosting Method is a relatively new model in this field (although it gives commendable results) and is hence why we worked with it extensively in this project.

The rental industry is a growing one and will continue to remain so . Factors influencing rental value keep changing and hence new MI algorithms help in accurate predictions with the trend and hence house rent prediction finds numerous applications.

Contribution of each member

1. Arjun (PES2UG20CS901) :

   Helped in formulating the code and refining it , inferred the code and helped in creating the ppt for peer review , performed literature survey on 2 papers and formulated this final project report.

2. Disha (PES2UG20CS906) :

   Helped in formulating the code and refining it, inferred the code and helped in creating the ppt for peer review, performed literature survey on 2 papers and answered the literature survey questions.

3. Vaishnavi (PES2UG20CS922)

   Main formulator/executor of code, performed literature survey on 2 papers and represented the compiled 6 papers done by the team in an IEEE format based report.

References

1. https://www.researchgate.net/publication/349227005_
2. (PDF) Housing Price Prediction via Improved Machine L
3. https://pdf.sciencedirectassets.com/271740/1-s2.0-S02
   Token=IQoJb3JpZ2luX2VjECQaCXVzLWVhc3QtMSJIMEYC
   94SZX%2FoJkcrgtN3pW8SMqlmEDLspaTcCwM6vEa%2Fp
   %2FPvHm8SGToORdparOIYAQr2FZj8YjYDiwiPFeiTNnSdg
   2C4elkMAJNbiRt9je6N9qFahaKKWN9sC58uWeFWBDT7z
   bJhylT3qXtMIKev5kGOqgBXTKJRyhqgoRR4GHV45ADZSb
   zaFKt&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
   Signature=c49665f589bda544753a9fed6e74faaaeb23f7
   9444-d996dcf1fab9&sid=ea8dc5b5113c894ea13955001
4. https://www.mdpi.com/2220-9964/8/8/349
5. https://paperswithcode.com/paper/house-price-prediction
6. (PDF) House Price Prediction Using Random Forest Mac