GROUP 4

PROJECT TITLE

House Price Prediction using Advanced Regression Techniques

Project Members:

1. Asavari Dawkhar

2. Musavir Arafath

3. Ramya

4. Shubham Debnath

Under Guidance of:

-Jithesh Kurian

CONTENTS:

1. ABSTRACT

This study demonstrates the comparison of different models for prediction of house price which assists house buyer, house seller or real estate agent to make better forecast of the house prices bearing in mind various features. The different Machine Learning algorithms employed to predict the house prices are Random Forest Regressor, Lasso Regressor, Linear Regressor, XGBoost Regressor, Ridge and ElasticNet Regressor; of which XGBoost yielded the most optimal results. Hence the main objective of this project is to predict the sales price of a house and so the housing dataset of 2919 records is obtained from Kaggle dataset on which grid search method is used to find the best model.

2. INTRODUCTION

House is one of the most essential needs in human life’s. Demand for houses grew rapidly over the years as people’s living standards improved. The housing markets have a positive impact on country’s currency, which is an important factor in economy scale. Many international organizations and human rights also emphasized about house importance. House is profoundly rooted in the economic, financial, and political structure of each country.

When people first think of buying a house/selling it to Real estate they study trends and other related stuff. People do this so they can look for a house which contains everything they need. While doing these people make a note of the price which goes with these houses.

In this way both people and companies are drawn to this market, which presents many profit opportunities that come from housing demands worldwide. These demands are influenced by several factors, such as demography, economy, and politics. For this reason, the analysis of such markets has been challenging for data scientists and ML engineers around the world, as they must take into account a wide range of fields, to come up with accurate results to customers and stakeholders.

The prospective home buyer considers several factors such as House style, Garage Type, Pool area, Neighborhood, Overall Condition etc., Regression techniques are widely used to build a model based on these factors to predict the house price.

Modelling uses machine learning algorithms, where machine learns from the data and use them to predict a new data. The most frequently used model for predictive analysis is regression. In this study, we have made an attempt to find best house price prediction regression model for Kaggle dataset.

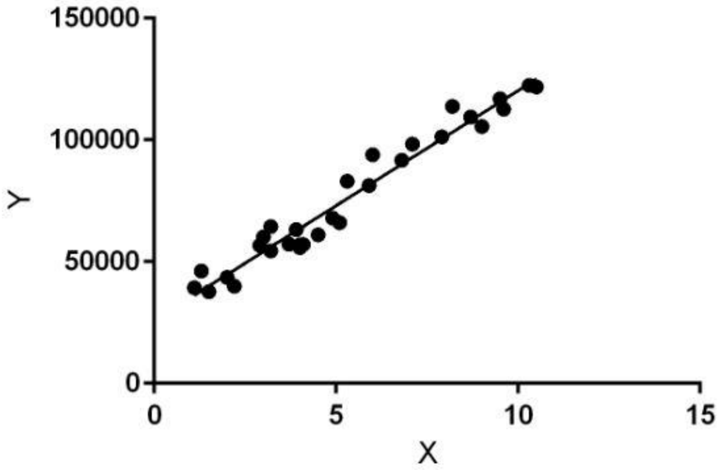
Machine learning offers a set of algorithms which allow the outcome prediction process to become more accurate without explicitly programming the software applications. ML modelling involves various steps such as Data Collection, Preparation of the data, Choosing the model, Model Training, Evaluating the model, Parameter Tuning, Predicting the result for specified data

For modelling we have considered six prediction models, they are Linear regression model, Random Forest Regression model, Lasso model, XGBoost regression model, Ridge and ElasticNet regression model. A comparative study was carried out with evaluation metrics. Based on metrics we decide on the model get fit into that model, after which we can use the model to forecast the monetary value of a particular house.

3. METHOD

In this project we are implementing different machine learning algorithms such as Linear Regression, Random Forest, XGBoost Regressor, Ridge Regressor, Lasso Regressor and ElasticNet Regressor.

a) **Linear Regression Algorithm**: It is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

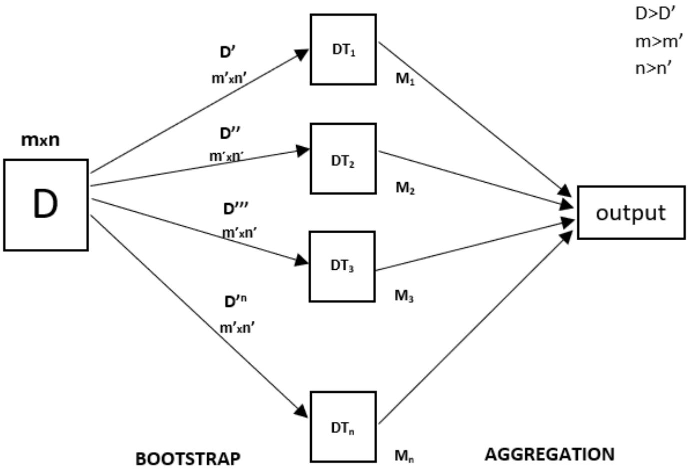


Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2 . x$$

b) **Random Forest Algorithm:** Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn’t depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.



A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

c) **XGBoost Regressor Algorithm:** XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners.The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values. The most common loss functions in XGBoost for regression problems is reg: Linear, and that for binary classification is reg: Logistics.

d) Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods. XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancels out and better one sums up to form final good predictions.

e) **Ridge Regressor Algorithm:** In Ridge regression, we add a penalty term which is equal to the square of the coefficient. The *L2* term is equal to the square of the magnitude of the coefficients. We also add a coefficient to control that penalty term. In this case if is zero then the equation is the basic OLS else if then it will add a constraint to the coefficient. As we increase the value of this constraint causes the value of the coefficient to tend towards zero. This leads to both low variance (as some coefficient leads to negligible effect on prediction) and low bias (minimization of coefficient reduce the dependency of prediction on a particular variable).

$$L_{ridge} = argmin_{\hat{\beta}}\left(\|Y-\beta*X\|^2+\boxed{\lambda*\|\beta\|_2^2}\right)$$

f) **Lasso Regressor Algorithm:** Lasso regression stands for Least Absolute Shrinkage and Selection Operator. It adds penalty term to the cost function. This term is the absolute sum of the coefficients. As the value of coefficients increases from *0* this term penalizes, cause model, to decrease the value of coefficients in order to reduce loss. The difference between ridge and lasso regression is that it tends to make coefficients to absolute zero as compared to Ridge which never sets the value of coefficient to absolute zero.

$$L_{lasso} = argmin_{\hat{\beta}}\left(\|Y-\beta*X\|^2+\boxed{\lambda*\|\beta\|_1}\right)$$

g) **ElasticNet Regressor Algorithm:** Sometimes, the lasso regression can cause a small bias in the model where the prediction is too dependent upon a particular variable. In these cases, elastic Net is proved to better it combines the regularization of both lasso and Ridge. The advantage of that it does not easily eliminate the high collinearity coefficient.

$$\hat{\beta} \equiv \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1).$$

GridSearchCV: It is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. As mentioned above, the performance of a model significantly depends on the value of hyperparameters. Note that there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

GridSearchCV is a function that comes in Scikit-learn's(or SK-learn) model_selection package.So an important point here to note is that we need to have Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

4. PROCEDURE

Step 1: Collecting Data

In this project we collected dataset from Kaggle. In which it has train dataset of 1460 rows and 81 columns and test dataset of 1460 rows and 80 columns.

Step 2: Data Exploration

In the first section of the project, we did an exploratory analysis (using heatmap) on the dataset and examined the observations. From that observations on train dataset and test dataset, checked null values using isnull() and obtained the output using heatmap.

Fig. Before removing null value

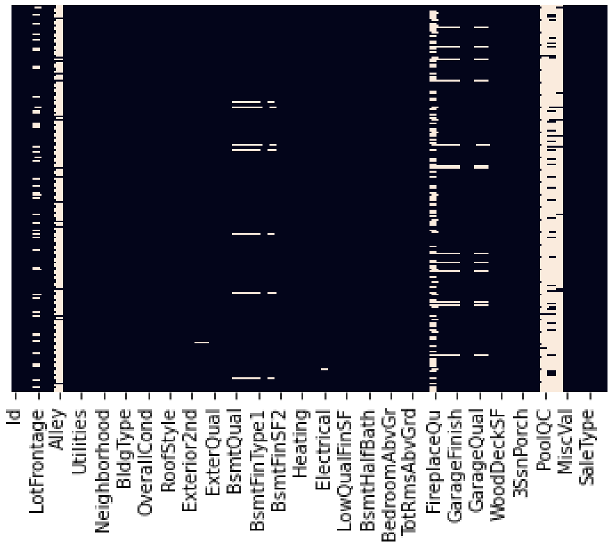
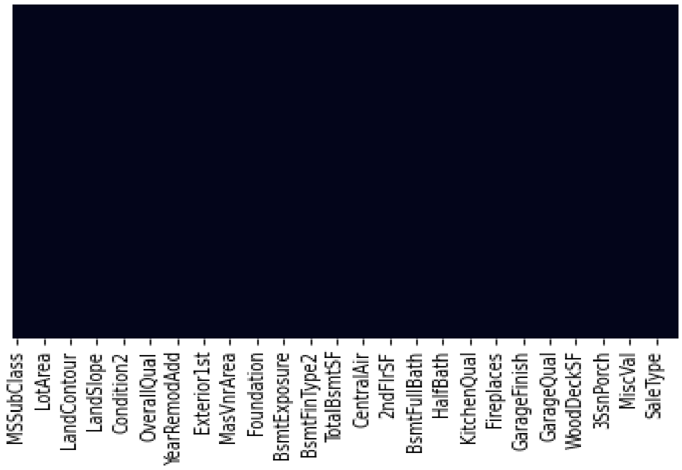


Fig. After removing null value



After that we concatenated test dataset and train dataset and deleted the duplicate columns. In next step we converted categorical features into numerical type using one hot encoding.

Step 3: Splitting Dataset into test and train

```
df_train= final_df.iloc[:1460,:]
df_test= final_df.iloc[1460:,:]
```

After splitting, we find x and y variable for developing model.

Step 4: Modelling

The algorithms we are using are as follows:

1.Linear Regression

2. Random Forest Algorithm

3. Rigde Regressor

4. Lasso Regressor

5. ElasticNet Regressor

6. XGBoost Regressor

We comparatively studied and implemented above mentioned algorithms by giving the clean dataset and the house prices are predicted.

Firstly we fit the model using linear Regression and finding the accuracy.

Step 5: Evaluating Model’s Performance

We are using GridSearchCV method for evaluating model performance. In this final section of the project, we are finding model name, best accuracy, best parameter.

Model   best_score   best_params

1. Lasso 0.701868{'selection': 'random'}
2. Ridge 0.824644{'solver': 'auto'}
3. ElasticNet 0.793055{'selection': 'random'}
4. linear_regression 0.686780{}
5. random_forest 0.849241{'criterion': 'squared_error', 'n_estimators':..
6. XGBoost 0.883842{'verbosity': 0}

After analysing this result, we acknowledge that XGBoost Regressor gave highest accuracy, so we train our model using XGBoost Regressor and predict the final result.

```
#predicting
mdl=xgboost.XGBRegressor()
mdl.fit(X_train,y_train)
y_pred=mdl.predict(X_train)
y_pred[1435]
```

Result:

```
179536.23
```

Cross verifying:

```
df_train['SalePrice'].iloc[1435,]
```

Result:

```
174000.0
```

5. RESULTS AND DISCUSSION

We have considered six prediction models, namely Linear regression model, Random Forest regression model, Lasso model, XGBoost regression model, Ridge and ElasticNet regression model. Then each model is stated with their hyperparameters in Grid Search CV process; after which each models scores such as best_score and best_parameters are evaluated, from which it is evident that XGBoost regressor yielded satisfactory score of 0.883842 (when its verbosity=0), Random Forest regressor yielded a score of 0.849241 (when criterion= 'squared_error', n_estimators= 10) and linear regression yielded the least score of 0.686780. As the XGBoost regressor yielded the best score among all the models, so the training data is fitted into this model training and then this model is used for predicting the price of specified house.

	Model	**best_score**	**best_params**
0	Lasso	0.701868	{'selection': 'random'}
1	Ridge	0.824644	{'solver': 'auto'}
2	ElasticNet	0.793055	{'selection': 'random'}
3	linear_regression	0.686780	{}
4	random_forest	0.849241	{'criterion': 'squared_error', 'n_estimators': 20}
5	XGBoost	0.883842	{'verbosity': 0}

```
    },
```

6. Sample Code

Data Collection and data preparation are the first and foremost steps before training a model. After which the major part is implementation of various model and predicting the best model out of those models.

```
# Models with their hyperparameters.
from sklearn.linear_model import LinearRegression,Ridge,Lasso,ElasticNet
from sklearn.ensemble import RandomForestRegressor
import xgboost

model_params={
    'Lasso':{
        'model': Lasso(),
        'params':{
            'selection':['cyclic','random']
        }
    },

  'Ridge':{
          'model': Ridge(),
          'params':{
              'solver':['auto','lsqr','saga','lbfgs'],
          }
      },

  'ElasticNet':{
          'model': ElasticNet(),
          'params':{
              'selection':['cyclic','random'],
          }
      },

    'linear_regression':{
        'model': LinearRegression(),
        'params':{}
    },

    'random_forest':{
        'model':RandomForestRegressor(),
        'params':{
            'n_estimators':[1,5,10,20],
            'criterion':['squared_error','absolute_error'],
        }
    },

      'XGBoost':{
          'model':xgboost.XGBRegressor(),
          'params':{
              'verbosity':[0,1,2]
          }
      }
}

# Hypertuning the Models:
from sklearn.model_selection import GridSearchCV
scores=[]

for model_name, mp in model_params.items():
  clf=GridSearchCV(mp['model'], mp['params'], cv=5)
  clf.fit(X_train,y_train)

  scores.append({
      'model': model_name,
      'best_score': clf.best_score_,
      'best_params':clf.best_params_
})
scores
```

7. CONCLUSION

This project mainly concentrates on comparison of different Machine Learning algorithms for house price prediction analysis. From the above results, it is clear that XGBoost has high score and Linear regression has least score when compared to all other algorithms. With further improvements in the XGBoost model will give better results in predicting the house price.

8. FUTURE ENHANCEMENT

In future, We want to enhance accuracy of our model using multilinear algorithm and using deep learning concepts as our model has still chance to improve score

9. BIBLIOGRAPHY

1. A Hybrid Regression Technique for House Prices Prediction Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh

2. House Price Prediction Using Regression Techniques: A Comparative Study 1 CH.Raga Madhuri, 2 Anuradha G, 3 M.Vani Pujitha 1,3Assistant Professor, 2Associate Professor 1,2,3Department of CSE, VRSEC, Vijayawada 1chragamadhuri@vrsiddhartha.ac.in, 2ganuradha@vrsiddhartha.ac.in, 3pujitha.vani@gmail.com

3. House Resale Price Prediction Using Classification Algorithms 1 P. Durganjali, 2 M. Vani Pujitha. 1M.Tech student, 2Assistant Professor Department of Computer Science and Engineering V R Siddhartha Engineering College, Vijayawada.

4. Housing Price Prediction Based on CNN Yong Piao School of Software Dalian University of Technology Dalian, China Ansheng Chen School of Software Dalian University of Technology Dalian, China Zhendong Shang Land Resources and Housing Information Center Dalian, China

5. Advanced Regression Techniques Based Housing Price Prediction Model. Sahar Abbasi

6. House Price Prediction using a Machine Learning Model: A Survey of Literature.