Housing Price Prediction

Done by *Vijayaraghavan S*

Introduction

BUSINESS PROBLEM FRAMING

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors to the world's economy. It is a very large market and there are various companies working in the domain.

Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improve their marketing strategies and focus on changing trends in house sales and purchases. Predictive modeling, Market mix modeling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

We are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

CONCEPTUAL BACKGROUND OF THE DOMAIN PROBLEM

A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below.

The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

1. Which variables are important to predict the price of a variable?
2. How do these variables describe the price of the house?

REVIEW OF LITERATURE

Based on the sample data provided to us from our client database where we have understood that the company is looking at prospective properties to buy houses to enter the market. The data set explains it is a regression problem as we need to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

Also, we have other independent features that would help to decide which all variables are important to predict the price of the variable and how do these variables describe the price of the house.

MOTIVATION FOR THE PROBLEM UNDERTAKEN

Our main objective in doing this project is to build a model to predict house prices with the help of other supporting features. We are going to predict by using Machine Learning algorithms. The sample data is provided to us from our client database. In order to improve the selection of customers, the client wants some predictions that could help them in further investment and improvement in the selection of customers.

House Price Index is commonly used to estimate the changes in housing prices. Since housing price is strongly correlated to other factors such as location, area, population, it requires other information apart from HPI to predict individual housing prices. There has been a considerably large number of papers adopting traditional machine learning approaches to predict housing prices accurately, but they rarely concern themselves with the performance of individual models and neglect the less popular yet complex models.

As a result, to explore various impacts of features on prediction methods, this paper will apply both traditional and advanced machine learning approaches to investigate the difference among several advanced models. This paper will also comprehensively validate multiple techniques in model implementation on regression and provide an optimistic result for housing price prediction.

ANALYTICAL PROBLEM FRAMING

MATHEMATICAL/ ANALYTICAL MODELING OF THE PROBLEM

We are building a model in Machine Learning to predict the actual value of the prospective properties and decide whether to invest in them or not. So, this model will help us to determine which variables are important to predict the price of variables & also how do these variables describe the price of the house. This will help to determine the price of houses with the available independent variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns.

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For specific mathematical reasons, this allows the researcher to estimate the conditional expectation of the dependent variable when the independent variables take on a given set of values.

Regression analysis is also a form of predictive modeling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modeling, and finding the causal effect relationship between the variables.

DATA SOURCES AND THEIR FORMATS

Dataset provided by Flip Robo Technologies consists of Train and Test datasets.
I used the train dataset which 1168 rows 81 Columns

DATA PREPROCESSING DONE

Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model. Therefore, it is the first and crucial step while creating a machine learning model. I have used the following preprocessing steps:

1. Loading the dataset in a Dataframe.
2. Checked the number of rows and columns and the data type details.
3. Checked the unique values to drop the unnecessary columns.
4. Checked for NA & Empty Columns and replaced with necessary data as per the supporting documentation provided with the dataset.
5. Checked and dropped duplicate values.
6. Separated categorical column names and numeric column names in separate list variables for ease in visualization.
7. Through Visualization analyzed the data with Count Plot and Scatter Plot.
8. With the help of the ordinal encoder, the technique converted all object datatype columns to a numeric datatype.
9. Checked for Outliers & Skewness.
10. Separated feature and label data to ensure feature scaling is performed avoiding any kind of bias-ness.
11. Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details.
12. Finally created a regression model function along with evaluation metrics to pass through various model formats.

DATA INPUTS- LOGIC- OUTPUT RELATIONSHIPS

When we loaded the training dataset, we had to go through various data pre-processing steps to understand what was given to us and what we were expected to predict for the project. When it comes to the logical part the domain expertise of understanding how real estate works and how we are supposed to cater to the customers came in handy to train the model with the modified input data. In the Data Science community, there is a saying “Garbage In Garbage Out” therefore we had to be very cautious and spent almost 80% of our project building time in understanding each and every aspect of the data how they were related to each other as well as our target label.

With the objective of predicting hosing sale prices accurately we had to make sure that a model was built that understood the customer priorities trending in the market imposing those norms when a relevant price tag was generated. I tried my best to retain as much data possible that was collected but I feel discarding columns that had lots of missing data was good. I did not want to impute data and then cause a bias-ness in the machine learning model from values that did not come from real people.

IDENTIFICATION OF POSSIBLE PROBLEM-SOLVING APPROACHES (METHODS)

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data and EDA to check the correlation of independent and dependent features. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models.

For this project, we need to predict the sale price of houses, which means our target column is continuous so this is a regression problem. I have used various regression algorithms and tested them for prediction. By doing various evaluations I have selected Extra Trees Regressor as the best suitable algorithm for our final model as it is giving good r2-score and the least difference in r2-score and CV-score among all the algorithms used. Other regression algorithms are also giving me good accuracy but some are over-fitting and some are under-fitting the results which may be because of fewer amounts of data.

In order to get good performance as well as accuracy and to check my model from over-fitting and under-fitting, I have made use of the K-Fold cross-validation and then hyperparameter tuned the final model.

Once I was able to get my desired final model I ensured to save that model before I loaded the testing data and started performing the data pre-processing as the training dataset and obtaining the predicted sale price values out of the Regression Machine Learning Model.

TESTING OF IDENTIFIED APPROACHES (ALGORITHMS)

The algorithms used on training and test data are as follows:

- Linear Regression Model
- Ridge Regularization Regression Model
- Lasso Regularization Regression Model
- Support Vector Regression Model
- Decision Tree Regression Model
- Random Forest Regression Model
- K Nearest Neighbours Regression Model
- Gradient Boosting Regression Model
- Ada Boost Regression Model
- Extra Trees Regression Model

Run and Evaluate selected models

I used a total of 10 Regression Models after choosing the random state amongst 1-1000 numbers. Then I even defined a function for getting the regression model trained and evaluated. The code for the models is as shown below.

```
maxAccu=0
maxRS=0

for i in range(1, 1000):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=i)
    lr=LinearRegression()
    lr.fit(X_train, Y_train)
    pred = lr.predict(X_test)
    r2 = r2_score(Y_test, pred)

    if r2>maxAccu:
        maxAccu=r2
        maxRS=i

print("Best R2 score is", maxAccu,"on Random State", maxRS)
```

Machine Learning Model for Regression with Evaluation Metrics

```
# Regression Model Function

def reg(model, X, Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=340)

    # Training the model
    model.fit(X_train, Y_train)

    # Predicting Y_test
    pred = model.predict(X_test)

    # RMSE - a lower RMSE score is better than a higher one
    rmse = mean_squared_error(Y_test, pred, squared=False)
    print("RMSE Score is:", rmse)

    # R2 score
    r2 = r2_score(Y_test, pred, multioutput='variance_weighted')*100
    print("R2 Score is:", r2)

    # Cross Validation Score
    cv_score = (cross_val_score(model, X, Y, cv=5).mean())*100
    print("Cross Validation Score:", cv_score)

    # Result of r2 score minus cv score
    result = r2 - cv_score
    print("R2 Score - Cross Validation Score is", result)
```

```
# Linear Regression Model

model=LinearRegression()
reg(model, X, Y)
```

```
RMSE Score is: 24450.992145552078
R2 Score is: 88.95133158521756
Cross Validation Score: 75.37421522245968
R2 Score - Cross Validation Score is 13.577116362757877
```

```
# Ridge Regularization

model=Ridge(alpha=1e-2, normalize=True)
reg(model, X, Y)
```

```
RMSE Score is: 24384.088996243536
R2 Score is: 89.01171190976207
Cross Validation Score: 75.66075392124897
R2 Score - Cross Validation Score is 13.350957988513102
```

```
# Lasso Regularization

model=Lasso(alpha=1e-2, normalize=True, max_iter=1e5)
reg(model, X, Y)
```

```
RMSE Score is: 24475.030001884254
R2 Score is: 88.92959693724507
Cross Validation Score: 75.37415144768174
R2 Score - Cross Validation Score is 13.555445489563326
```

```
# Support Vector Regression

model=SVR(C=1.0, epsilon=0.2, kernel='poly', gamma='auto')
reg(model, X, Y)
```

```
RMSE Score is: 76633.41213348275
R2 Score is: -8.530872394841428
Cross Validation Score: -6.2196135410372255
R2 Score - Cross Validation Score is -2.311258853804202
```

```
# Decision Tree Regressor

model=DecisionTreeRegressor(criterion="poisson", random_state=111)
reg(model, X, Y)
```

```
RMSE Score is: 62543.048104857946
R2 Score is: 27.710527744040427
Cross Validation Score: 47.92460187087371
R2 Score - Cross Validation Score is -20.21407412683328
```

```
# Random Forest Regressor

model=RandomForestRegressor(max_depth=2, max_features="sqrt")
reg(model, X, Y)
```

```
RMSE Score is: 41506.22877132564
R2 Score is: 68.16216566149387
Cross Validation Score: 64.04181550034377
R2 Score - Cross Validation Score is 4.1203501611501
```

```
# K Neighbors Regressor

KNeighborsRegressor(n_neighbors=2, algorithm='kd_tree')
reg(model, X, Y)
```

```
RMSE Score is: 40231.139456865574
R2 Score is: 70.08826296941496
Cross Validation Score: 63.461941681712105
R2 Score - Cross Validation Score is 6.626321287702851
```

```
# Gradient Boosting Regressor

model=GradientBoostingRegressor(loss='quantile', n_estimators=200, max_depth=5)
reg(model, X, Y)
```

```
RMSE Score is: 34595.1729989324
R2 Score is: 77.88189165329096
Cross Validation Score: 79.78739827409618
R2 Score - Cross Validation Score is -1.9055066208052125
```

```
# Ada Boost Regressor

model=AdaBoostRegressor(n_estimators=300, learning_rate=1.05, random_state=42)
reg(model, X, Y)
```

```
RMSE Score is: 33262.51891928025
R2 Score is: 79.55311111528376
Cross Validation Score: 78.90999304267208
R2 Score - Cross Validation Score is 0.6431180726116708
```

```
# Extra Trees Regressor

model=ExtraTreesRegressor(n_estimators=200, max_features='sqrt', n_jobs=6)
reg(model, X, Y)
```

```
RMSE Score is: 24077.911164495334
R2 Score is: 89.28592743883011
Cross Validation Score: 84.67181508628198
R2 Score - Cross Validation Score is 4.614112352548133
```

KEY METRICS FOR SUCCESS IN SOLVING PROBLEM UNDER CONSIDERATION

The key metrics used here were r2_score, cross_val_score, MAE, MSE, and RMSE. We tried to find out the best parameters and also to increase our scores by using Hyperparameter Tuning and we will be using the GridSearchCV method.

Cross-Validation:

Cross-validation helps to find out the overfitting and underfitting of the model. In the cross-validation, the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part is 20% of the full dataset. While running the Cross-validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for training data. This way we will get the first estimate of the model quality of the dataset.

In a similar way, further iterations are made for the second 20% of the dataset is held as a holdout set and the remaining 4 parts are used for training data during the process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross-validation process to get the remaining estimate of the model quality.

R2 Score:

It is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

Mean Squared Error (MSE):

MSE of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors – that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. RMSE is the Root Mean Squared Error.

Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

Hyperparameter Tuning:

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.

We are not aware of optimal values for hyperparameters that would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically. This selection procedure for hyperparameters is known as Hyperparameter Tuning. We can do tuning by using GridSearchCV.

GridSearchCV is a function that comes in Scikit-learn (or SK-learn) model selection package. An important point here to note is that we need to have the 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.

Hyper parameter tuning

```
# Choosing Extra Trees Regressor

fmod_param = {'n_estimators' : [100, 200, 300],
              'criterion' : ['squared_error', 'mse', 'absolute_error', 'mae'],
              'n_jobs' : [-2, -1, 1],
              'random_state' : [42, 111, 340]
             }

#After comparing all the regression models I have selected Extra Trees Regressor as my best model and have listed down it's parameters above referring the sklearn webpage.
```

```
GSCV = GridSearchCV(ExtraTreesRegressor(), fmod_param, cv=5)
GSCV.fit(X_train,Y_train)
```

```
GridSearchCV(cv=5, estimator=ExtraTreesRegressor(),
             param_grid={'criterion': ['squared_error', 'mse', 'absolute_error',
                                       'mae'],
                         'n_estimators': [100, 200, 300], 'n_jobs': [-2, -1, 1],
                         'random_state': [42, 111, 340]})
```

```
GSCV.best_params_
```

```
{'criterion': 'mse', 'n_estimators': 100, 'n_jobs': -2, 'random_state': 42}
```

```
Final_Model = ExtraTreesRegressor(criterion='mse', n_estimators=100, n_jobs=-2, random_state=42)
Model_Training = Final_Model.fit(X_train, Y_train)
fmod_pred = Final_Model.predict(X_test)
fmod_r2 = r2_score(Y_test, fmod_pred, multioutput='variance_weighted')*100
print("R2 score for the Best Model is:", fmod_r2)
```

```
R2 score for the Best Model is: 82.2385387640393
```

It is possible that there are times when the default parameters perform better than the parameters list obtained from the tuning and it only indicates that there are more permutations and combinations that one needs to go through for obtaining better results.

Post model building and choosing the appropriate model I went ahead and loaded the testing dataset. After applying all the data.

Pre-processing steps as the training dataset I was then able to get the predicted sale price results. Since the values were in array format, I converted them into a data frame and merged it with the original testing data frame that consisted of only of our feature columns. Once the testing dataset with feature columns and the predicted label was formed, I exported the values in a comma-separated values file to be accessed as needed.

Learning Outcomes of the Study in respect of Data Science:

The above study helps one to understand the business of the real estate. How the price is changing across the properties. With the Study, we can tell how multiple real estate amenities like swimming pool, garage, pavement and lawn size of Lot Area, and type of Building raise decides the cost. With the help of the above analysis, one can sketch the needs of a property buyer and according to need, we can project the price of the property.

ACKNOWLEDGMENT:

I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement.