Housing Price Prediction

USING ADVANCED REGRESSION TECHNIQUE

NAME: ABHISHEK DAS
ID: 1772359
SCHOOL: FACULTY OF ECMS
COURSE: BIG DATA PROJECT

ABSTRACT

House prices are dependent on various factors that continuously interact with each other maki ng its prediction a complex problem. The aim of this project was to predict this complex varia ble by understanding its dependencies on the governing factors. Three prediction models, lin ear regression, lasso regression and random forest was used in succession to model the proble m. The algorithms were selected in this sequence to understand and overcome any shortcomin g of the preceding algorithms. While different algorithms were implemented, the steps in each implementation remained the same, which was data analysis, pre-processing and model build up. This was implemented using Python. As expected, the prediction value kept increasing as the complexity of the model was increased with the highest R^2 value of 0.9834 for Random F orest model.

INTRODUCTION

Houses are one of the most fundamental human needs and their selection is a function of their price. There is no single accurate factor that measures the house price. Predicting the prices of the house not only concerns the buyer or real estate agent but also reflect the economy of a region. The goal of this project was to predict the sale price of houses in Ames, Iowa. The project description contains the *Ames Housing* dataset was composed by *Dean De Cock*.This data set has 79 features that are related to the house pricing and describe both qualitative and quantitative aspects of the house. These features provide us with almost all the information that typically determines the price of the house for the price.

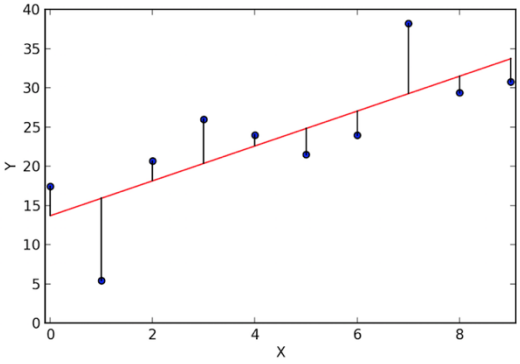
Our training data set included 1460 houses (i.e., observations) accompanied by 79 attributes (i.e., features, variables, or predictors) and the sales price for each house. Our testing set included 1459 houses with the same 79 attributes, but sales price was not included as this was our target variable.

The challenge of this project is to create a regression model which will be able to accurately estimate price of the house given the features.

METHODOLOGY

Linear Regression

It is one of the most popular and oldest machine learning algorithms. The core idea behind this model is that it assumes that the relationship between a dependent continuous variable and one or more independent variable is linear. In simple words, linear regression algorithm finds the relationship between the data we have and data we want to predict. It's mostly used to predict values (Sale price in our case) rather than classifying data. When we find a linear relation, or to be put in the mathematical terms, we try to fit a draw a line that "best fits" the data set. But how does our model do that. We first calculate the distance between the blue points and the line and find the sum of square of those distances. Then we would look for the minimum sum of squares to find the best fit. This process is also known as *least square method*.



Lasso Regressor

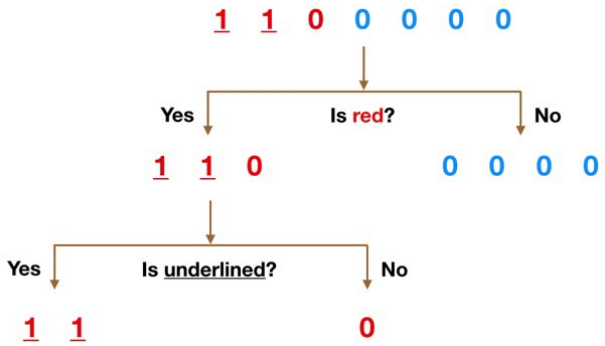
One of the common problems that occurs in Linear regression is *Overfitting*. But we want our model to *generalise* and perform well with the test data. To achieve that we implement something called Regression in our model. To perform this, we will modify our cost function by adding a "penalty" to the Residual Squared Sum (RSS). By adding this penalty, we are trying to decrease the value of the parameters that are less important to our data.

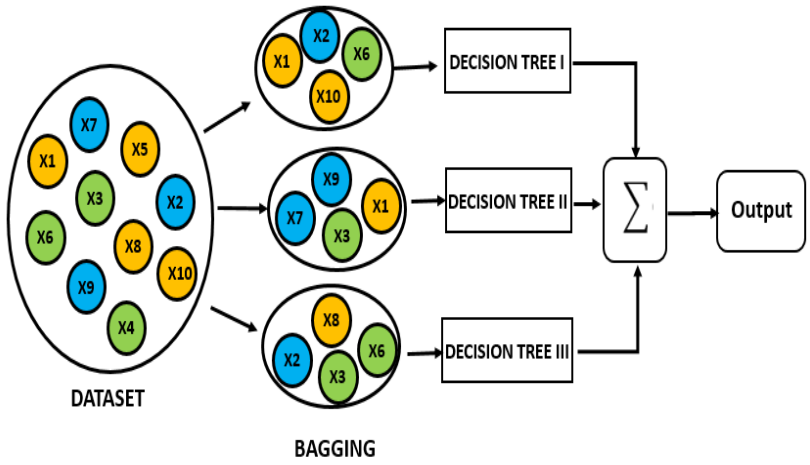
L1 Regularization or Lasso regression adds a penalty which is equal to the sum of absolute value of coefficient in the optimisation problem. One of the key differences between Lasso and other regressors is that it not only performs "coefficient shrinkage "but also select features (*Least Absolute Shrinkage and Selection Operator)*. If we choose a large value for the penalty, we can regress the coefficient values closer to zero.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Random Forest

The problems with our previous models were that it required a lot of engineering and regularization to prevent Overfitting. Feature engineering is a very complicated task and it requires some domain knowledge as well. To avoid this, we chose Random Forest algorithm. Random Forest is an *Ensembling* method and is quite robust and popular. The algorithm works on the core idea of *Decision Tree*. To understand the logic behind Decision Tree, let us look at the figure. As we can see, the model helps to split the observations through some computation. In real data, splits are done based on different methods like *Entropy* or *Chi-squared*. Random Forest is simply consisting of large number of such decision tress. A large number of such uncorrelated tress working together gives a better result than an individual model.





The Random Forest algorithm uses something called *Bagging*. In Bagging, we train each Decision Tree on a different set of data and we then sample the data set with replacement. In the figure,
We have a training dataset: X1 to X10. Random forest may create three decision trees and take inputs by sample the data from dataset and finally predicts results based on aggregation (in case of regression problem).

EXPERIMENT

Implementation

The implementation of this project was divided into 3 phases:
1.Exploratory Data Analysis
2.Data Pre-Processing
3.Building Models

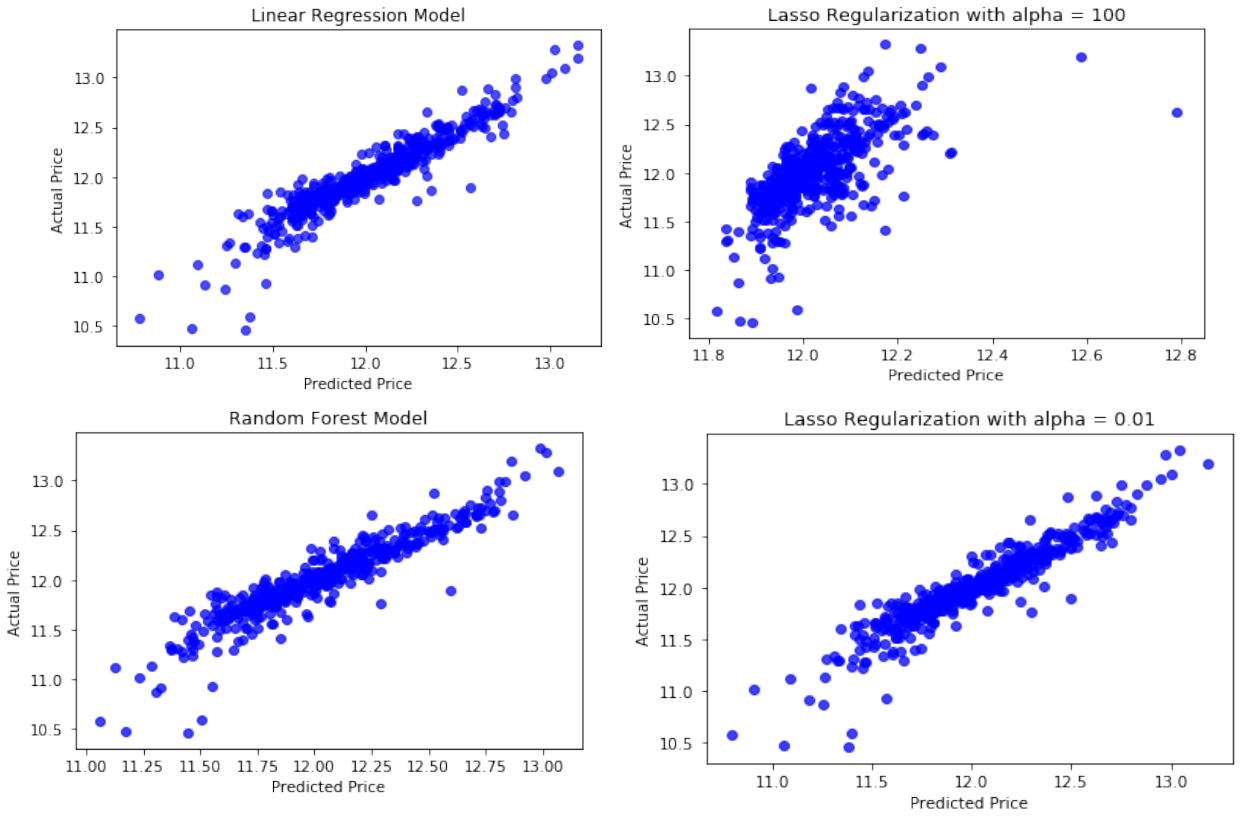
1.Exploratory Data Analysis: I believe it is one of the most important steps in any Data Science project. It is really necessary to understand the data first and gain as many insights possible from it. This process makes the rest of the job a bit easier and more efficient. To start with, I imported the necessary libraries (*pandas, NumPy, matplotlib)* and loaded the training and test data. To take a closer look at the data I used. *head ()* function. I used *train.info ()* to know about the columns and its data types. The data types used in our data set were float, int and object. Python provide us with a lot of visualisation libraries that help us to analyse our data better. The objective of our project was to predict the sale price. I explored our target variable with. *describe ()* function and also plot its distribution on a histogram. I found that data was more distributed for cheaper prices than expensive ones which make sense. I applied the log transformation to improve the linearity of our model. Simply rescaling our target variable will not change this. However, log transformations will help the data points to spread more uniformly in the graph. It is very important to select features in linear regression model that will help in predicting the target value. The *DataFrame.corr()* method helps in knowing the correlation between the columns of our data set. I further investigated the features that were highly correlated with the target variable with the help of scatter plot. These plots also helped in finding outliers in the data.

2.Data Pre-Processing: As the name suggest, this process involved cleaning our data. It is also a starting point for feature engineering. I started by removing the outliers that was visualised earlier *isnull.sum. ()* methods helps to identify the null values in each column. I also did some feature engineering in data like Street. Our algorithm works better with numeric data. As such, I used the method of one-hot-encoding to add numerical values (Boolean values in our case) in the column. *pd.get_dummies ()* helped me in achieving this task. Feature engineering is a very crucial task and it also requires some knowledge about the domain of the project. Feature engineering needs to be applied in both training and testing data set. After wrangling with the data, I checked for the rest of the missing data again and used the method of interpolation to fill the missing values with an average value. I used *DataFrame.interpolate* for this task.

3.Building Models: This was the final phase of the project. I assigned the features in X and target variable to Y. The data was then split into 70 % for validation and 30 % for testing (which was the requirement of this project) using *train_test_split ()*. The data was first trained using a simple Linear Regressor using *linear_model.LinearRegression()*. Then we trained it using Lasso regularizer, linear *model.Lasso(alpha=alpha).* We used different values of alpha . to check the accuracy and how well it fit the data. We finally used RandomForestRegressor *RandomForestRegressor(n_estimators=300, random_state=0)* to train our data and get better accuracy. Here *n_estimators* determines the number of trees or base trainers.

Finding and analysis

MODEL	R^2 SCORE	RMSE
Linear Regressor	0.88418	0.0183
Lasso Regressor	0.88335	0.0185
Random Forest Regressor	0.9834	0.0210



The data was first trained with a Linear regressor and it could be observed that the model performs quite well and could predict values for test data as R^2 and RMSE values are quite descent. The data was then trained with a Lasso Regularizer to shrink the influence of less important features and sometimes making it zero. It could be seen that the accuracy of the model was improved. It is very important to hyper tune the parameter alpha for our Lasso regressor. An alpha as small as 0.01 to 1 gives us significant improvement. On the other hand, high values of alpha (alpha =100) can lead to underfitting. Finally, we trained our model with Random Forest. This model doesn't require any scaling of features and has an inbuilt regularizer which can take care of the overfitting. As expected, we could observe that there is a vast improvement in the results.

CONCLUSION

According to our analysis, we can say that features like Overall Quality, Living Area Square Feet have the greatest statistical impact in predicting the sale price of the house. To solve our regression problem, a linear regressor gives us a good fit on the dataset. The shortcomings of linear regressor can be overcome by using some other algorithms, like lasso regression. From the results, it could be seen that Random Forest gave significant improvement in the accuracy in comparison to other algorithms. Understanding the domain and the motivation for prediction can utilised to select an algorithm that is best suited to the problem at hand. There is still a lot of future scope for this project. For instance, if the categorical data is explored and handled more efficiently, the accuracy might be improved further. We can also examine different variations to optimize Rain Forest Algorithm, such as considering the splits of nodes, the requirements of leaf nodes, etc.