Western Engineering

ECE 9063/ECE 9603 Data Analytics Foundations

Assignment 1

Assignment 1

Pair Number 10

Group Members:

Haoxuan Xu

Yuanxin Tuo

Problem Description

The problem that we want to address is about housing price prediction. Nowadays, housing is one of the most valuable assets that people would strive for. No matter it is about either selling or buying a house, we surely want to make the wisest choice in terms of its price. However, the price of a housing depends on many aspects, such as the size of rooms, neighborhood, yearbuilt, etc. Each of these aspects holds different weights on determining the price of a certain housing. Hence, it can be as hard as it sounds to predict its sale price. That's why, for this project we are trying to build models that forecast and predict the sale price of a housing.

Data for Modelling

The open-source dataset that we are using is from Kaggle. There are three sub-datasets that are available to us in which are a training set (training.csv), an output dataset that has actual sales price corresponding to each housing (sample_submission.csv), and a descriptional text file that illustrates the details about some attributes (data_description.txt).

1. Clearly indicate what attributes and/or parts you have used

According to different algorithms that we used, we chose slightly different attributes from the datasets.

- **SVR**: 'MSSubClass', 'LotFrontage','OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'GarageArea', 'GarageCars','FullBath', 'GrLivArea', '1stFlrSF', 'LotArea'
- **Linear Regression:** 'MSSubClass', 'LotFrontage','LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'GarageArea'
- **Random Forest Regression:** 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'GarageArea'

2. Indicate any normalization or transformation you performed on the data.

Prior to training our models, we performed data standardization. We applied a Python library, scikit-learn to perform this linear data transformation. Essentially, data standardization changes the scale of the data, which avoids associated issues, such as random features may be dominating others or features have vastly different ranges. Hence, this crucial step can improve the overall performances of our models and algorithms, such as speeding up convergence.

For our dataset, we have 79 different features and they range vastly different. For instance, we selected "YearBuilt" and "LotArea" as two of the features, and the fact that they have

vast different scales can bring a lot of issues. Hence, we performed data standardization, $Z = (X - \underline{X}) / \sigma$, where $\underline{X}$ is the mean and σ is the standard deviation of the data.

In addition to standardization, some of the features that we selected are missing data or not a number, therefore, we applied Data Imputation, SimpleImputer from scikit-learn to complete the missing values. The strategy we chose is to replace missing data with the median value of that column.

Background

We will use three methods to accomplish this task and they are:

1. SVR

As predicting housing price is more likely to be a regression problem, SVR is suitable for this regression task. Essentially, SVR aims to find an insensitive tube that covers as many data points as possible. Unlike the Least Square function, the data points within the tube won't be penalized or considered as errors, therefore, they don't contribute to formation of the tube. In that case, we need the data points outside the tube (support vectors) to form the tube. The incensitive tube essentially is the hyperplane plus some margins.

In order to find a hyperplane, kernel functions are crucial. As we are using a lot of features, it becomes hard to find a linear regression line, but in terms of different circumstances, there are different types of kernel to transform the data. In our case, we used Radial Basis Function (RBF) Kernel.

Once there is a hyperplane, the way to check how well this hyperplane fits the data, we use loss function or cost function to test it. For SVR, because it doesn't penalize all errors (the errors within the margin), the loss function for SVR is called incensitive loss function. The first part is to calculate the loss based on the difference between actual value and the predicted value, and if one data point is within the "tube", the loss is zero. The second part is a regularization term that avoids issues, such as overfitting.

2. Linear Regression

Linear regression models assume a linear relationship between the dependent variable (house price) and the characteristics of the independent variables (e.g. LotFrontage, age of the house, number of bedrooms, etc.). It can therefore be applied to problems where such a relationship exists.

Linear regression is relatively simple to understand and apply. The coefficients of the independent variables in the model can be interpreted directly. For example, a positive coefficient for the variable "number of bedrooms" indicates that the predicted house price will increase as the number of bedrooms increases.

3. Random Forest Regression

Random Forest regression can handle large signs with small samples and is suitable for datasets with a large number of variables, such as in a house price prediction project.

This method has many advantages. Firstly, it can identify the most important predictor variables in a dataset. For example, in the house price prediction project, the most important predictor variables were the overall quality of the house, the size of the living space above ground, and the total basement area. Furthermore, it is resistant to overfitting, which can be a common problem in house price forecasting due to the large number of potential explanatory variables. Last but not least, random forest regression allows us to model the complex nonlinear relationships often found in housing data.

Methodology

Firstly, we loaded the training dataset by reading CSV files. According to how differently features relate to our target value (SalePrice), we selected features for each algorithm.

After loading the data, we performed data standardization as a step of data preparation and a lot of the algorithms can be benefiting from data standardization. Besides, we didn't choose text data but only numerical data, so we didn't apply models, such as Vector Space Models. Additionally, we applied data imputation to handle missing values.

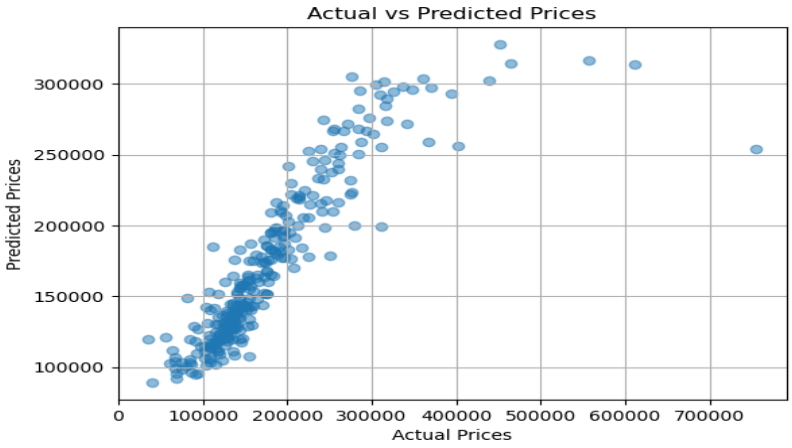
In addition to data preparation, we initially split the dataset into the training set and the test set by 80:20. Then performed data imputation to handle missing values so that we could create and train the model using different algorithms. In terms of which algorithm it is, we adjusted the parameters and tried different kernels. We need to evaluate and select the models after prediction, so we choose to K-fold cross validation. We referenced the lecture notes and decided to use 5-fold cross validation. Besides, we also seeked ways to tune hyperparameters (the regularization term and gamma) of our algorithms. We did not optimize the data in assignment 1. We will use stochastic gradient descent and Adam to optimize the data in the next assignment.

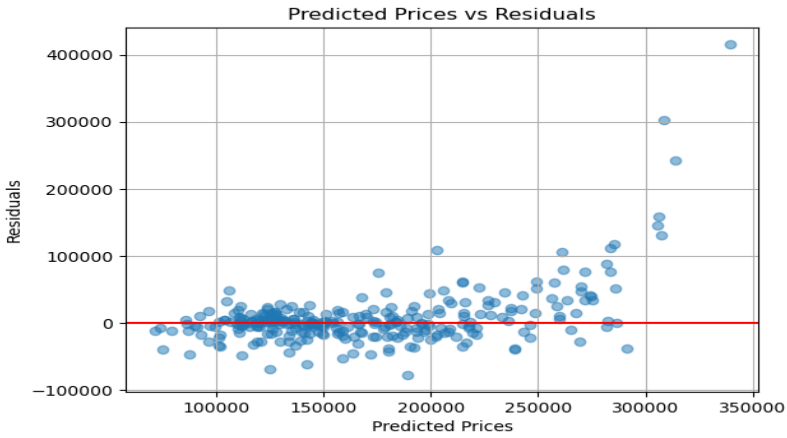
For evaluation, we calculated the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for each algorithm for the sake of accuracy measures and comparisons. To have better visualized insights of how each algorithm performed, we plotted out several graphs, such as residual plot, etc.

Evaluation/Results

Graph results obtained with different algorithms.

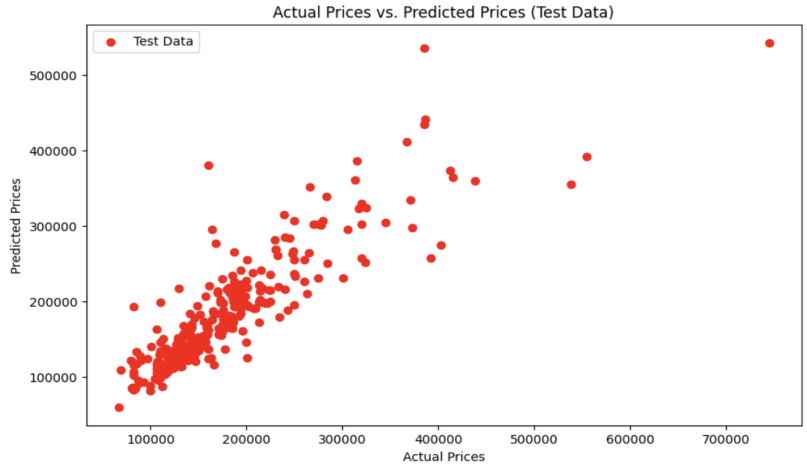
SVR





Linear Regression

Random Forest Regression



1. Metrics and evaluation process

5-Fold Cross-validation

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. k-fold cross-validation is a good method to evaluate the model design, not a particular training. Because we re-trained the model of the same design with different training sets.

SVR

#Train/Test Split 80:20

```
RMSE without using 5-fold cross validation: $47649.74964039872
 MAE without using 5-fold cross validation: $22760.47893119769
```

#Applying k-fold cross-validation (k=5)

```
Average RMSE: $41940.978086326206
 Average MAE: $23261.86196977298
```

Linear Regression

```
Fold 1 MAE: 30364.58, RMSE: 50118.06
Fold 2 MAE: 26845.30, RMSE: 37870.21
Fold 3 MAE: 30586.48, RMSE: 45040.59
Fold 4 MAE: 28682.14, RMSE: 44065.76
Fold 5 MAE: 27159.68, RMSE: 37724.32
Mean MAE: 28727.64
Mean RMSE: 42963.79
```

Random Forest Regression

```
Fold 1 MAE: 25307.80, RMSE: 39406.27
Fold 2 MAE: 22505.99, RMSE: 35149.78
Fold 3 MAE: 26105.84, RMSE: 37870.06
Fold 4 MAE: 22682.01, RMSE: 34043.49
Fold 5 MAE: 23779.30, RMSE: 35409.33
Mean MAE: 24076.19
Mean RMSE: 36375.79
```

In addition to calculating RMSE and MAE, we plotted a residual graph that you can see above, which observes how the predicted values deviates from the actual values. Apparently, there are a few datapoints with vastly large residuals and causing the larger RMSE. There can be a number of reasons, such as data errors from the given dataset, inefficient hyperparameter tuning, feature selection, etc.

2. Results

a. Result Comparison

With these above results, we can see that all three models predict roughly the same outcomes. Through the results of MAE we can get that SVR model has the best performance in house price prediction among the three models. From the results of RMSE, we can see that Random Forest Regression has the better prediction. It is worth noting that the number of features in the SVR part is more than Linear Regression and Random Forest Regression, which may be a part of influencing the results and we will optimize the steps in the subsequent assignments.

b. Discussion & Comments

From the perspective of the results, none of the three algorithms performed well in terms of MAE and RMSE. To some degree, the models tend to be considered as underfitting or poor models without further optimizing steps, such as parameter tuning, feature selection, etc. Most of the actual target values(SalePrice) are around $200k, an approximate $40k error is certainly not optimal, therefore, we are looking forward to optimizing the models during the second assignment.

References

[1] “Sklearn.Impute.SimpleImputer.” *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html. Accessed 18 Oct. 2023.