Housing Price Prediction

Ishan Acharyya
1901077

Problem Definition

- The Problem is to predict the price of houses using the given dataset
- We will be using the “Boston Dataset”
- The goal of this statistical analysis is to help us understand the relationship between house features(13 in case of Boston housing dataset such as crim,rm,zn,age etc.) and how the variables are used to predict house price.Our aim here is to to predict the reasonable housing price with these aspects of the houses by using the Boston Housing dataset.

Objective:

- Predict the house price
- Use different models in terms of minimizing the difference between predicted and actual rating.

Dataset Description

- The Dataset is derived from information collected by the U.S. Census Service concerning housing in the area of Boston Mass.
- The dataset contains a total of **506** cases.

There are **14** attributes in each case of the dataset. They are:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's

- We will be predicting the “MEDV” value

Literary Survey

These papers deliver an insight of Linear Regression being used in real life:

LINEAR REGRESSION ANALYSIS TO PREDICT THE LENGTH OF THESIS COMPLETION

Linear Regression Model Development for Analysis of Asymmetric Copper-Bisoxazoline Catalysis

LINEAR REGRESSION ANALYSIS TO PREDICT THE LENGTH OF THESIS COMPLETION

- Students noticed that it takes more time in completing their thesis than the time given to them.
- These are caused by several factors, such as, students who are working, working hours that do not support the implementation of thesis preparation, students who already have families and other factors.
- Universities have to plan special methods or strategies to reduce the number of students who don't complete their thesis within the given time.
- In this paper, length of time for completion of college student thesis is predicted by utilizing data mining and a simple linear regression approach.

- Simple linear regression analysis can have a simple structure using 1 independent variable, namely the average inhibiting factor (Working Status, Working Hours, Work Sip, Guidance Media, Status) (X1).
- The number of days of thesis completion being the dependent variable (Y).
- This can be accepted and used as a tool in predicting the completion time of student thesis work in universities.
- the simple linear regression equation model is: Y = 280.450 + 1.650 X

Linear Regression Model Development for Analysis of Asymmetric Copper-Bisoxazoline Catalysis

- In this paper, Multivariate linear regression (MLR) analysis is used to unify and correlate different categories of asymmetric Cub Isoxazoline (BOX) catalysis.
- Statistical tools and extensive molecular featurization have guided the development of an inclusive linear regression model, providing a predictive platform and readily interpretable descriptors.

- This workflow also permitted the development of a complementary linear regression model correlating analogous BOX-catalyzed reactions employing Ni, Fe, Mg, and Pd complexes
- Overall, this strategy highlights the utility of MLR analysis in exploring mechanistically driven correlations across a diverse chemical space in organometallic chemistry and presents an applicable workflow for related ligand classes

Experiment Description

Using two different models, Housing price predictions have been made.

1) `Batch Gradient Descent`
2) `Stochastic Gradient Descent`

We have also used Linear Regression model from SKlearn Library as a benchmark and verified results using Kfold-cross validation.

*All experiments have been run using tuned HyperParameters

Results

Studying Batch Gradient Descent:

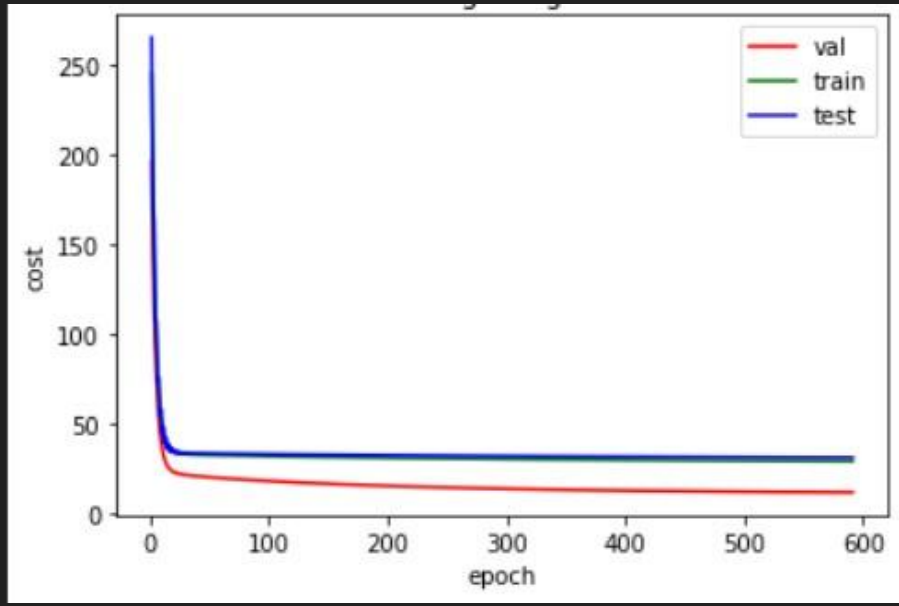
Split:
Train:Validation:Test
30 : 10 : 60

After generating 5000 groups of epoch,alpha and rho combinations, we observe that:

```
Best values are generated at : 1006
Mean Squared Error for best group gets generated as : 19.8114650788416
Next step will be to analyse the model for overfitting, underfitting, performance.
```

Results(analysis)

We obtain final model results as shown in the figure after checking overfitting using validation dataset.




```
Final Model Results :
Test MSE 20.6667332
Train MSE 18.62004493
Validation MSE 11.2610343
```

Result analysis:

Insignificant Difference between Train, Test MSE and good perf, thus no Underfitting/Overfitting of model.

Small difference between Validation and Train MSE, no Overfitting of Hyperparameters.

Result

Studying Stochastic Gradient Descent:

Split:
Train:Validation:Test
30 : 10 : 60

After generating 5000 groups of epoch,alpha and rho combinations, we observe that:

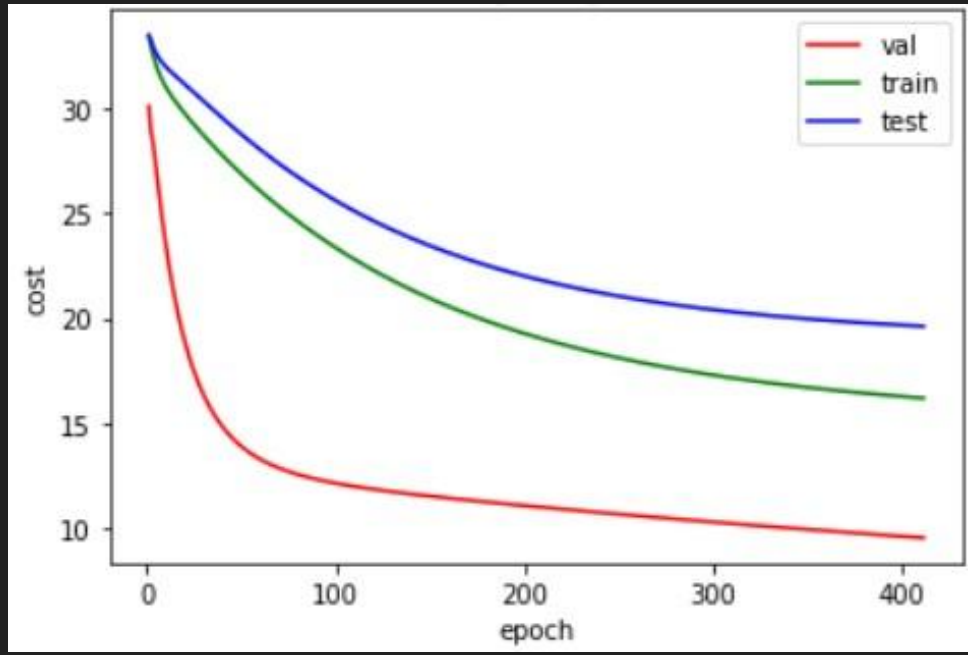
```
Best values are generated at : 2756
Mean Squared Error for best group gets generated as :161567980989464 3
Next step will be to analyse the model for overfitting, underfitting, performance.
```

Result(analysis)

We obtain final model results as shown in the figure after checking overfitting using validation dataset.

Since the number of samples are less and Stochastic Gradient Descent doesn't provide any significant improvements, we will continue to use Batch Gradient Descent for further experiments.

```
Final Model Results :
Test MSE 19.61212346
Train MSE 16.19283591
Validation MSE 9.547732916
```



RESULT ANALYSIS:

Insignificant Difference between Train, Test MSE and good perf, thus no Underfitting/Overfitting of models

Small difference between Validation and Train MSE, no Overfitting of Hyperparameters

The MSE is lower compared to Batch Gradient Descent, but the runtime is significantly greater, making Batch gradient descent better for this dataset, which is small

K-fold cross validation for Batch GD

MSE for each Fold and Average for Batch Gradient Descent using K-fold cross validation

k=1	44.48440457	50.18838939
k=2	43.85539293	52.30223996
k=3	46.69652906	43.2976008
k=4	45.22015235	47.08378704
Avg	45.06411973	48.2180043

Split:
Validation using random 30pc
Train:Test
25 : 75

RESULT ANALYSIS:

The average error is near the Train, Test MSE, verifying that our Model has no bias.
However, the MSE has gone up, indication slight underfitting.
No overfitting, since performance is comparable for both sets

SKlearn Model

Using sklearn's linear regression model as a benchmark

Train:Test = 30:70

Test MSE 14.366525126333983

Train MSE 6.103884478675774

Analyzing the effect of K-fold validation using SKLearn Library

k=1	40.17586023	43.73010596
k=2	39.28718049	47.51023235
k=3	40.73223014	41.97628022
k=4	41.4283319	40.35673631
Avg	40.40590069	43.39333871

RESULT ANALYSIS:

The average error is near the Train, Test MSE, verifying that our Model has no bias .

However, the MSE has gone up significantly, indicating slight underfitting with the given split.

No overfitting can be observed.