Abstract

Linear regression is a linear approach for modelling the relationship between a response (dependent variable) and one or more independent variables. The case of one independent variable is called simple linear regression and for more than one independent variable, it is referred to as multiple linear regression. Gradient descent is an optimization algorithm that is used to find the optimized coefficients of the linear regression model by iteratively minimizing the error of the model.

In this report, a linear regression model is designed using gradient descent method to predict the housing prices in Boston based on 13 attributes. Further evaluation is conducted by comparing the predicted outputs vs the actual outputs and evaluating the Root Mean Squared Error (RMSE) and the R^2-Score of the model and the impact of learning rate on the gradient descent algorithm.

Based on the comparison findings described in this report, it is observed that the model yields a relatively high R^2 score of 0.758 and RMSE of 5.255. This is comparable to the R^2-score using the LinearRegression model from sklearn library. 76% of the variation in the output variable is explained by the input variables with a RMSE value of 5.255 for the predicted response.

Introduction

A linear regression model is designed using gradient descent method to predict the housing prices in Boston based on 506 instances with the following 13 attributes:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25000 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 centers
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 (output): Median value of owner-occupied homes in $1000's

The model parameters of the 13 attributes are estimated based on this dataset. Further elaboration on the gradient descent algorithm used and the evaluation metrics will be covered in the subsequent sections.

Methodology

The following methodology was employed in the completion of the tasks and objectives of this report.

Task 1:

Before using the dataset to build the linear regression model, checks were carried out to determine if data pre-processing steps were required for the given dataset. This included checking the number of data entries, data types, missing data. Example is shown in figure 1 below.

```
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

Figure 1. Displayed info of Boston dataset generated

Based on the data pre-processing checks done, there is no further need to modify the dataset as there is no missing values and the data is in numerical format (no structural errors).

Task 2:

The linear regression constructed is an extension of simple linear regression analysis (Boston University School of Public Health, 2013). As there are 13 input variables, 13 regression coefficient parameters need to be determined. These 13 parameters are required to accommodate the impact of all 13 input variables, and its individual regression equation, on the output variable. The linear regression equation is given as below:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 + b_7x_7 + b_8x_8 + b_9x_9 + b_{10}x_{10} + b_{11}x_{11} + b_{12}x_{12} + b_{13}x_{13}$$

Task 3:

The p-value of the variables was first found using statsmodels.api (figure 2) and the variance inflation factor (VIF) for each predictor was calculated for the input variables (figure 3).

OLS Regression Results

Dep. Variable:	MEDV	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Sat, 23 Apr 2022	Prob (F-statistic):	6.72e-135
Time:	20:53:39	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	36.4595	5.103	7.144	0.000	26.432	46.487
CRIM	-0.1080	0.033	-3.287	0.001	-0.173	-0.043
ZN	0.0464	0.014	3.382	0.001	0.019	0.073
INDUS	0.0206	0.061	0.334	0.738	-0.100	0.141
CHAS	2.6867	0.862	3.118	0.002	0.994	4.380
NOX	-17.7666	3.820	-4.651	0.000	-25.272	-10.262
RM	3.8099	0.418	9.116	0.000	2.989	4.631
AGE	0.0007	0.013	0.052	0.958	-0.025	0.027
DIS	-1.4756	0.199	-7.398	0.000	-1.867	-1.084
RAD	0.3060	0.066	4.613	0.000	0.176	0.436
TAX	-0.0123	0.004	-3.280	0.001	-0.020	-0.005
PTRATIO	-0.9527	0.131	-7.283	0.000	-1.210	-0.696
B	0.0093	0.003	3.467	0.001	0.004	0.015
LSTAT	-0.5248	0.051	-10.347	0.000	-0.624	-0.425

Omnibus:	178.041	Durbin-Watson:	1.078
Prob(Omnibus):	0.000	Jarque-Bera (JB):	783.126
Skew:	1.521	Prob(JB):	8.84e-171
Kurtosis:	8.281	Cond. No.	1.51e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.51e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Figure 2. Regression results summary based on OLS

	Var	Vif
10	PTRATIO	85.03
5	RM	77.95
4	NOX	73.89
9	TAX	61.23
6	AGE	21.39
11	B	20.10
8	RAD	15.17
7	DIS	14.70
2	INDUS	14.49
12	LSTAT	11.10
1	ZN	2.84
0	CRIM	2.10
3	CHAS	1.15

Figure 3. VIF values

Based on the variables with the lowest VIF (low degree of multicollinearity) and whose p-value is < 0.05 (significant), the 5 attributes chosen are CRIM (per capita crime rate), ZN (proportion of residential land zone lots), LSTAT (% lower status of the population), DIS (weighted distance to employment centers) and B (1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town). CHAS and RAD were excluded as they are categorical data (discrete values).

Comparing the input attributes with the output attributes (figure 4 and 5), LSTAT is strongly negatively correlated with MEDV (figure 6). This indicates that the lower the percentage of lower status of the population, the higher the median value of owner-occupied homes (larger population able to own more expensive homes). In comparison, MEDV is not highly correlated with B, DIS (figure 7), ZN or CRIM. Another point to note is that majority of the data are 0 for ZN (figure 8). This could mean that a large proportion of residential land zoned for lots is below 25,000 square feet.

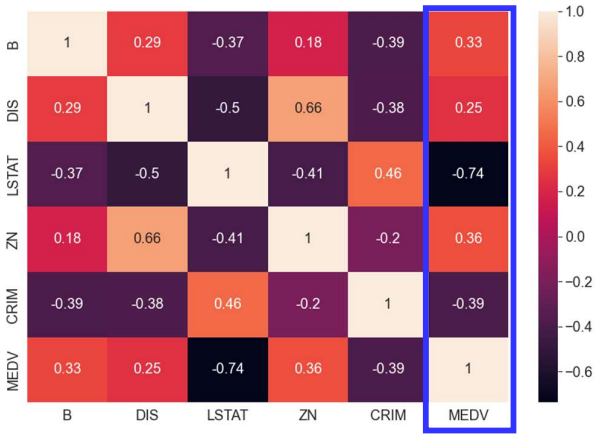

Figure 4. Correlation matrix of the 5 input attributes

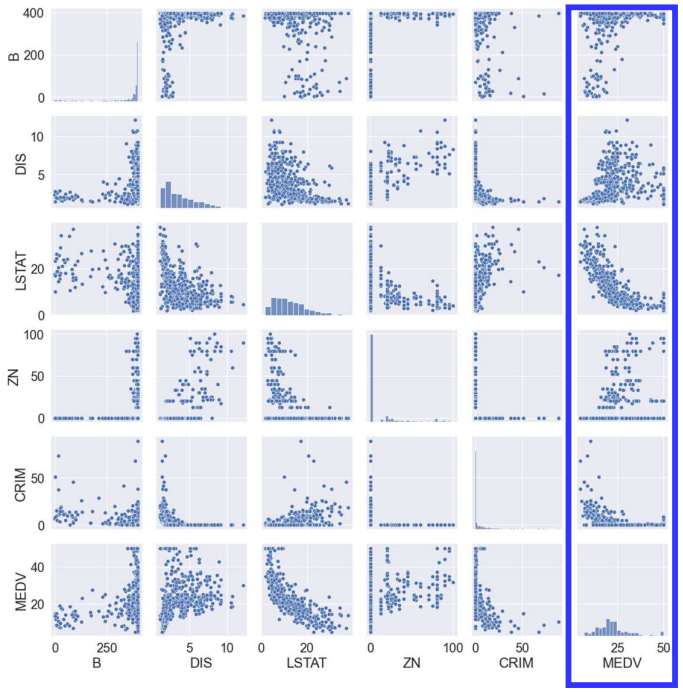

Figure 5. Pair plot of the 5 input attributes

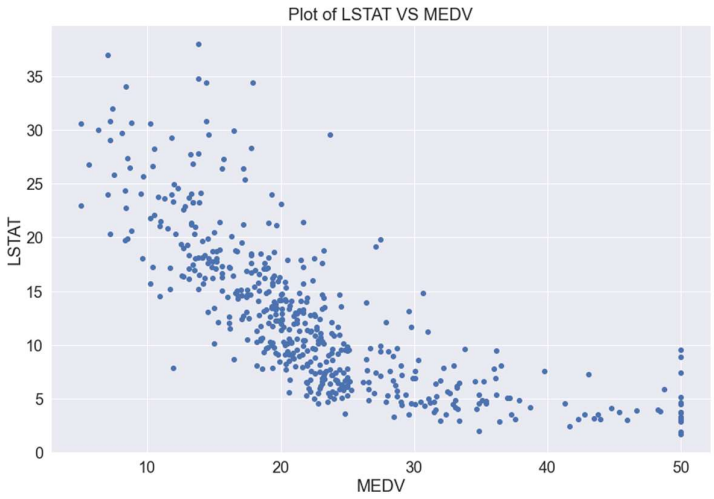

Figure 6. LSTAT vs output variable MEDV

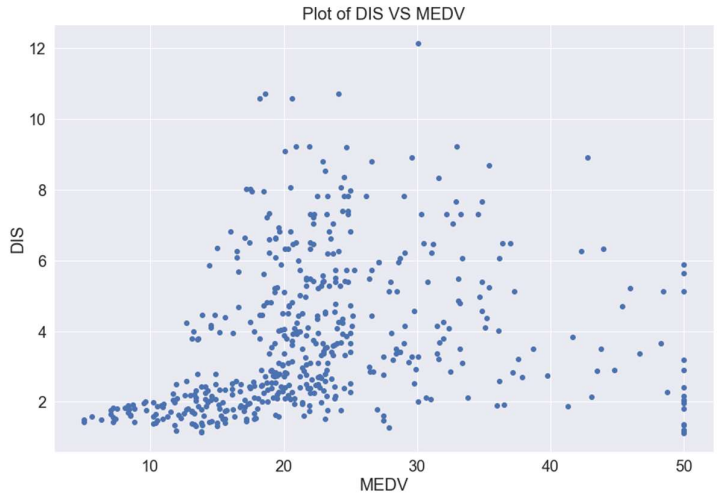

Figure 7. DIS vs output variable MEDV

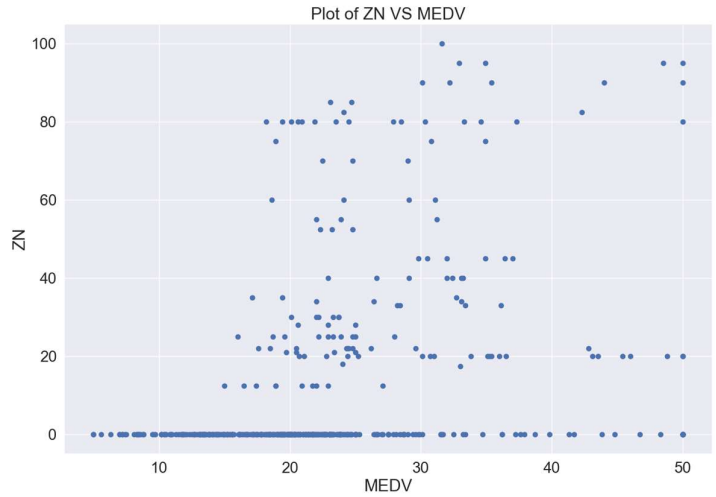

Figure 8. ZN vs output variable MEDV

Task 4:

Root Mean Squared Error (RMSE) represents the square root of the variance of the residuals and is a measure of how concentrate the data is around the line of best fit. The lower the RMSE, the better the fit. The formula is given as below (ZACH, 2019):

$$RMSE = \sqrt{\frac{\sum(\hat{y}_i - y_i)^2}{n}}$$

R^2-score is a measure of how well the linear regression model fits the dataset and tells us the proportion of the variance in the output variable that can be explained by the input variables (ZACH, 2019). R^2-score ranges from 0 to 1, with 1 indicating that the model is able to perfectly predict the output and 0 reflecting that the model is not able to explain the predictor variable at all. The formula is given as below (Coefficient of Determination, R-squared, n.d.):

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Task 5:

Gradient descent is an optimization algorithm that's used for training a machine learning model. Initial parameter values are first defined, and the gradient descent calculates the cost function based on these parameters. The algorithm then iteratively adjusts the parameter values such that the cost function is minimized. The aim of the algorithm is to find the set of parameter values that would result in the lowest cost function (Miller, 2018). The formula for cost function is given as below:

$$\frac{1}{2m}\sum_{i=1}^{m}(h(\theta^{(i)} - y^{(i)})^2$$

The step at which the gradient descends is determined by the learning rate (α) and this affects how fast the cost function converges. With low learning rates, the improvements will almost be linear, and it will take a long time for the function to converge. With high learning rates, the cost function will decay faster, and the cost function will reach convergence at a smaller number of iterations (figure 9). However, there is a possibility that the cost function is unable to find a global minimum as the algorithm has "overshoot" the optimized parameters. If the learning rate is very high, this might result in an exponentially increasing cost function instead. Therefore, an optimal value of the learning rate can be found by determining which learning rate results in the lowest cost function. The results of this comparison will be described in the "Results" section of this report.

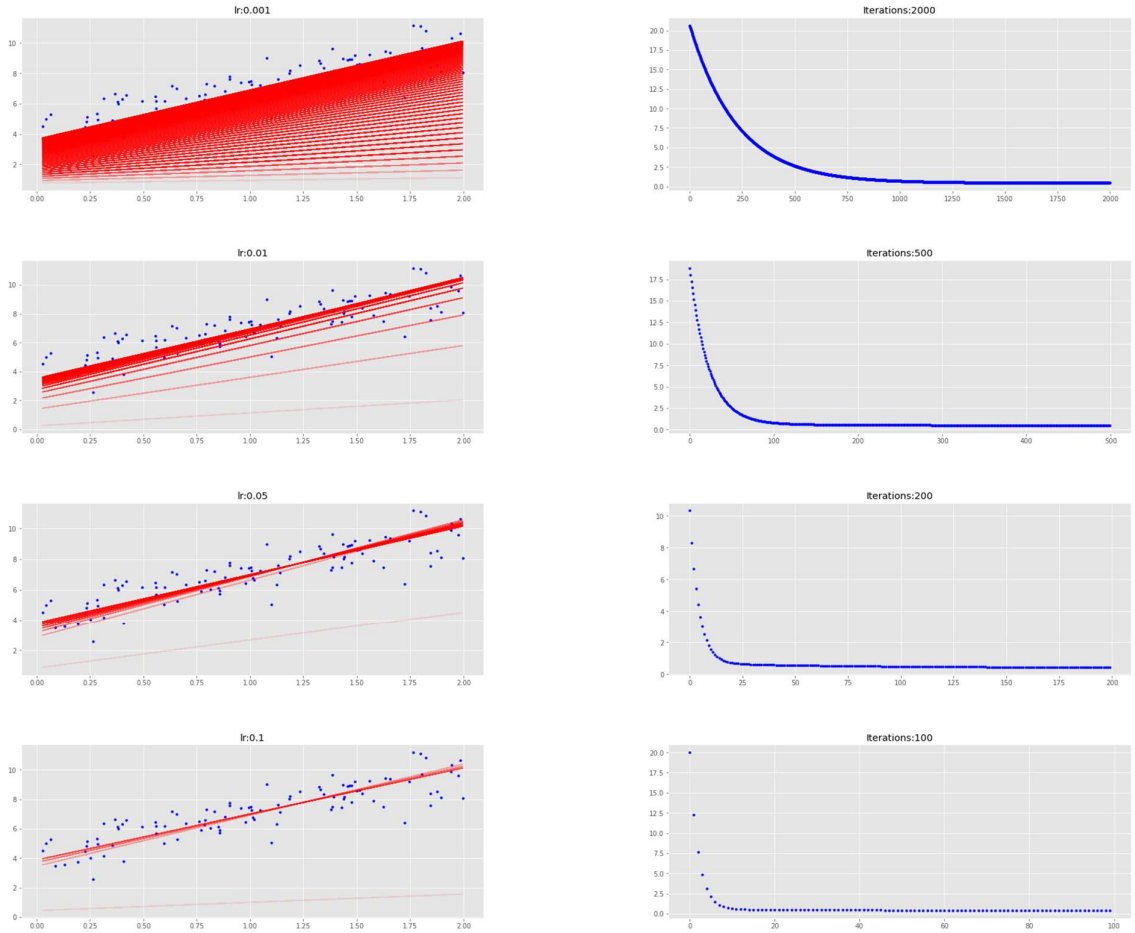

Figure 9. Comparison of learning rates and iterations on gradient descent

Task 6:

Min-max normalization was conducted on the input features. The formula for min-max normalization is given as:

$$normalized\ data = \frac{data - min}{\max - min}$$

Task 7:

The Boston dataset was split into a train dataset, comprising of 90% of the original dataset and the remaining 10% was used as the test dataset. The results were used in the comparison of the performance of the machine learning algorithm for this predictive modelling problem.

Results

The results obtained from the above tasks is described in the following sections below.

Task 5:

As mentioned previously, if the learning rate is very high, the cost function increases exponentially. With the gradient descent algorithm implemented, an example of a very high learning rate is 0.55 (figure 10).

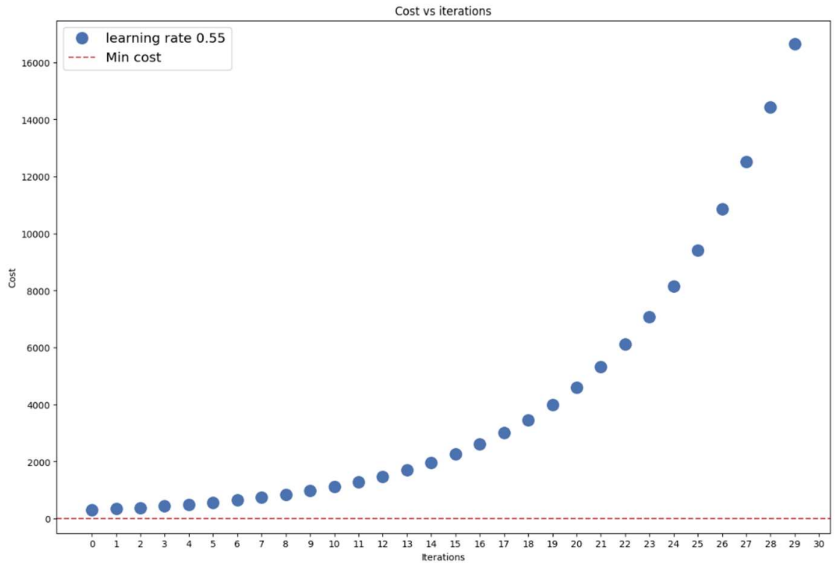

Figure 10. Plot of cost function vs iterations with a learning rate of 0.55

By comparing the cost functions vs iterations of different learning rates, 0.001, 0.003, 0.01, 0.03, 0.1 and 0.3, a learning rate of 0.03 is chosen as a good learning rate as it approaches the lowest cost function (figure 11) without being overly aggressive (figure 12).

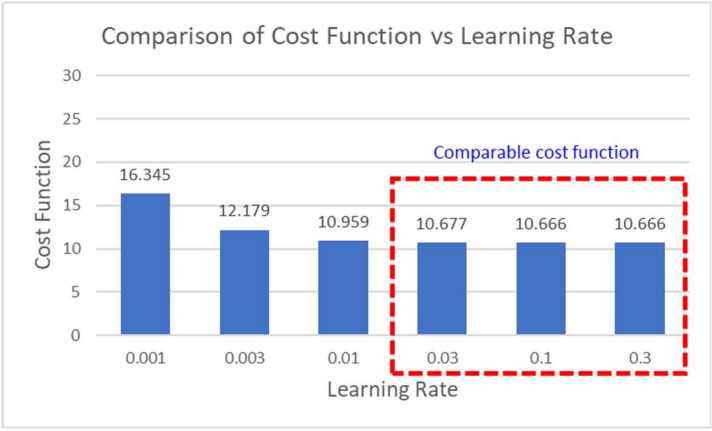

Figure 11. Comparison of cost function vs learning rates

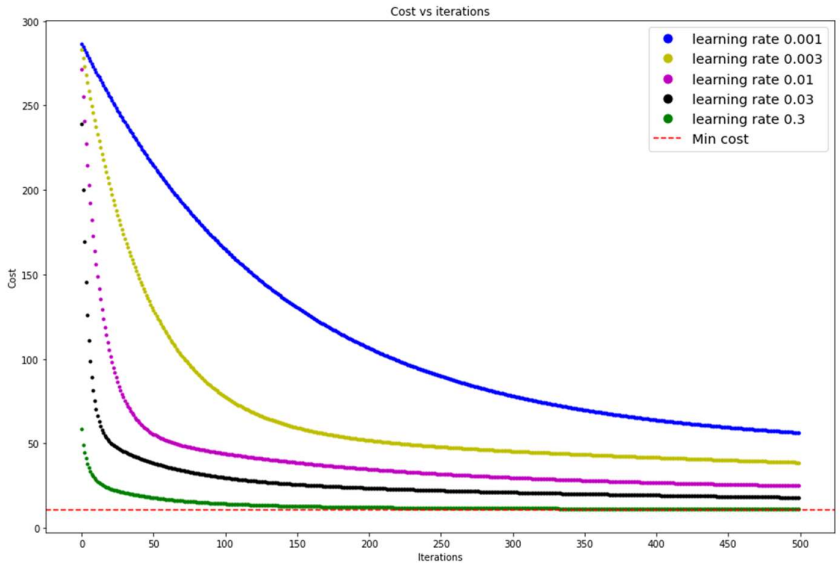

Figure 12. Graph of cost function vs iterations for different learning rates

Note that if number of iterations is too little, the algorithm will prematurely end before convergence is reached (figure 13). The number of iterations must be larger than a threshold value (Ogura, 2017), where the change in the cost function in subsequent iterations is less than 10^{-3} (figure 14).

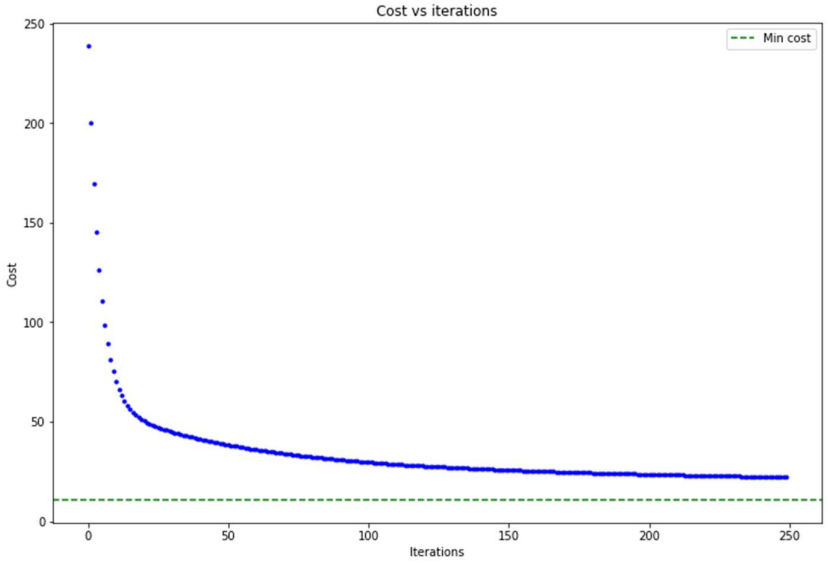

Figure 13. Example of convergence not being reached due to insufficient iterations

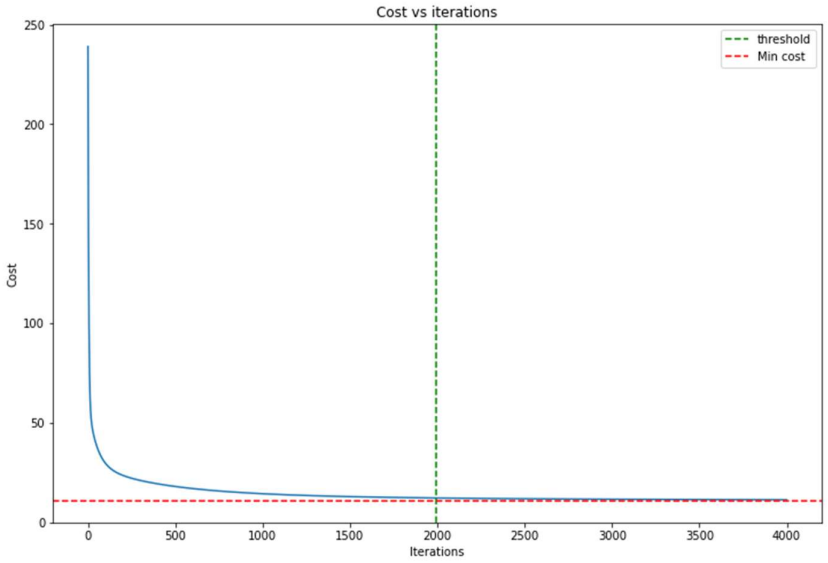

Figure 14. Example of threshold point where convergence is declared

Task 8:

Based on the learning rate of 0.03, the gradient descent algorithm was run to compute the parameters of the linear regression equation. Figure 15 shows a graph of the cost function vs the number of iterations using a learning rate of 0.03.

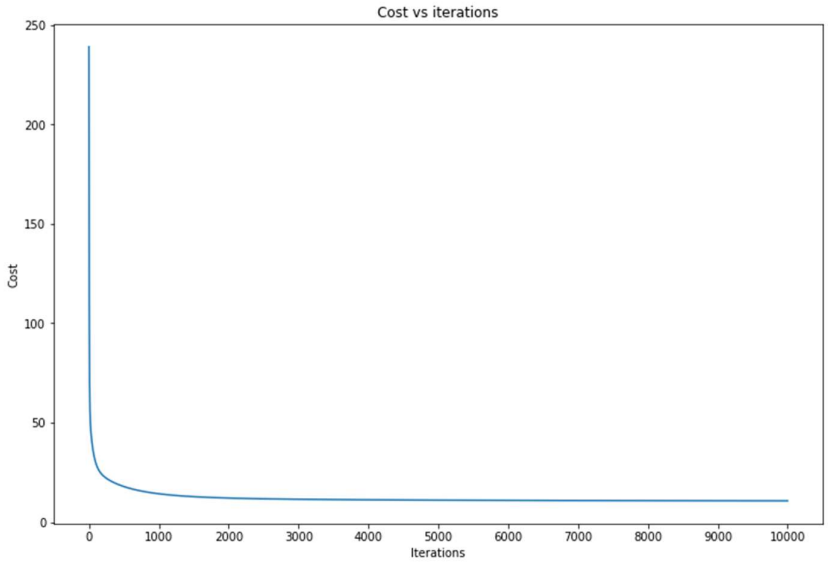

Figure 15. Plot of cost function vs iterations with a learning rate of 0.03

Using the parameters obtained, the predicted output variables for the train dataset and the test dataset were calculated. Plotting predicted_y_train VS actual_y_train and predicted_y_test VS actual_y_test (figure 16), we observe that most of the data points are clustered around the black line (perfect prediction, i.e. predicted y equals to actual y) and our model is able to predict the output for majority of the points.

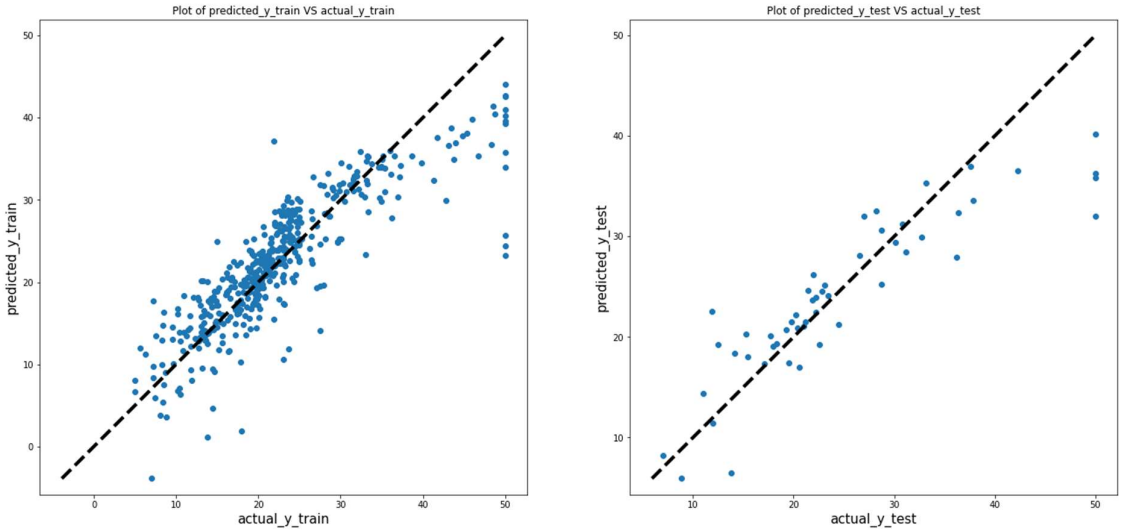

Figure 16. Plot of predicted_y_train VS actual_y_train (left) and plot of predicted_y_test VS actual_y_test (right)

Comparing with the RMSE and R^2-score of both the train and test dataset (figure 17), we observe that the R^2-score is close to the upper limit of 1 at around 0.75 for both the train and test data. The linear regression model fits the train and test dataset and is able to explain the predictor variable. However, the model will predict the response with a RMSE value of around 5 (relatively high compared to the output variable dataset). As the RMSE value for the test dataset is higher than the training dataset, this could indicate overfitting.

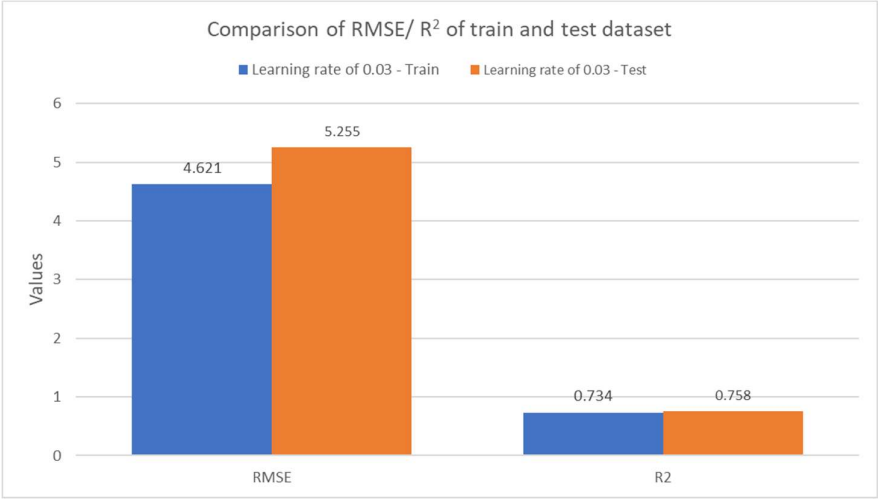

Figure 17. Comparison of RMSE/ R^2 of train and test dataset

Task 9:

A linear regression model using sklearn library's LinearRegression model was built using the same train and test dataset. The accuracy of the model and the sklearn model is comparable for both the train and test dataset as seen in figure 18 below.

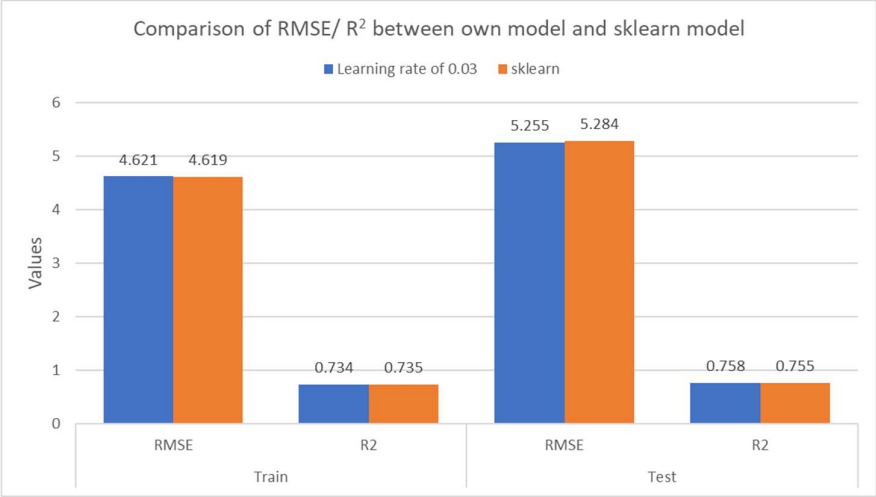

Figure 18. Comparison of RMSE/ R^2 of own model and sklearn model

Conclusion

The results predicted from the linear regression algorithm designed in this report is comparable to the results predicted by the sklearn linear regression model. From the results, the model is able to explain the predictor value with 0.758 accuracy. However, the predicted response has an RMSE value of around 5.255.

References

Boston University School of Public Health. (17 January, 2013). *Multiple Linear Regression Analysis*. Retrieved from https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_multivariable/bs704_multivariable7.html

Coefficient of Determination, R-squared. (n.d.). Retrieved from https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html

Miller, L. (11 January, 2018). *Machine Learning week 1: Cost Function, Gradient Descent and Univariate Linear Regression*. Retrieved from Medium: https://medium.com/@lachlanmiller_52885/machine-learning-week-1-cost-function-gradient-descent-and-univariate-linear-regression-8f5fe69815fd

Ogura, M. (19 December, 2017). *Intuition (and maths!) behind multivariate gradient descent*. Retrieved from Towards Data Science: https://towardsdatascience.com/machine-learning-bit-by-bit-multivariate-gradient-descent-e198fdd0df85

Sirohi, K. (20 January, 2019). *Beginner: Cost Function and Gradient Descent*. Retrieved from Towards Data Science: https://towardsdatascience.com/machine-leaning-cost-function-and-gradient-descend-75821535b2ef

ZACH. (24 February, 2019). *What is a Good R-squared Value?* Retrieved from statology: https://www.statology.org/good-r-squared-value/