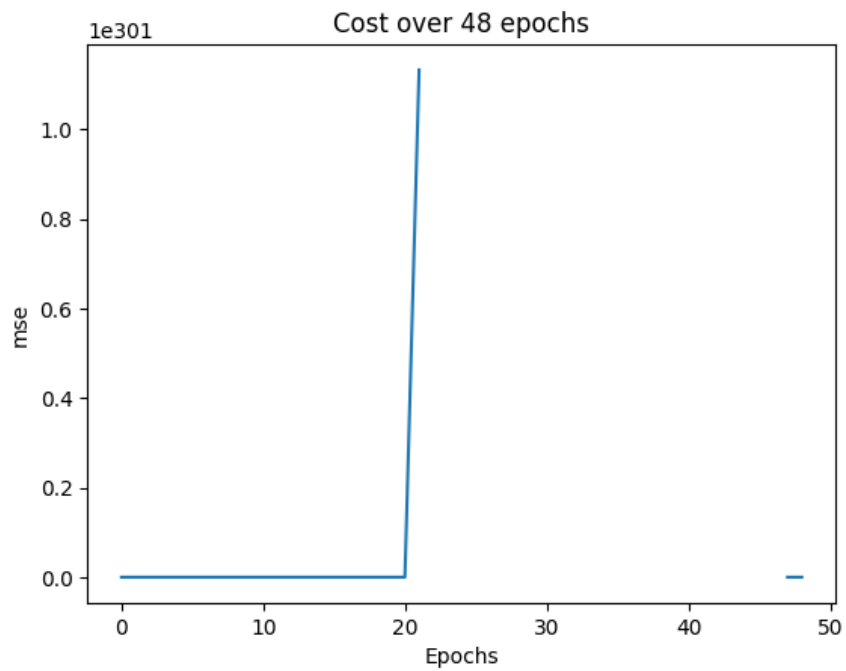


Question 1- Training without normalization

Attempting to train a model on the housing dataset as-is is futile. Every column of data is on a different scale, and the high magnitude of the values causes all sorts of errors. The model simply fails to train.

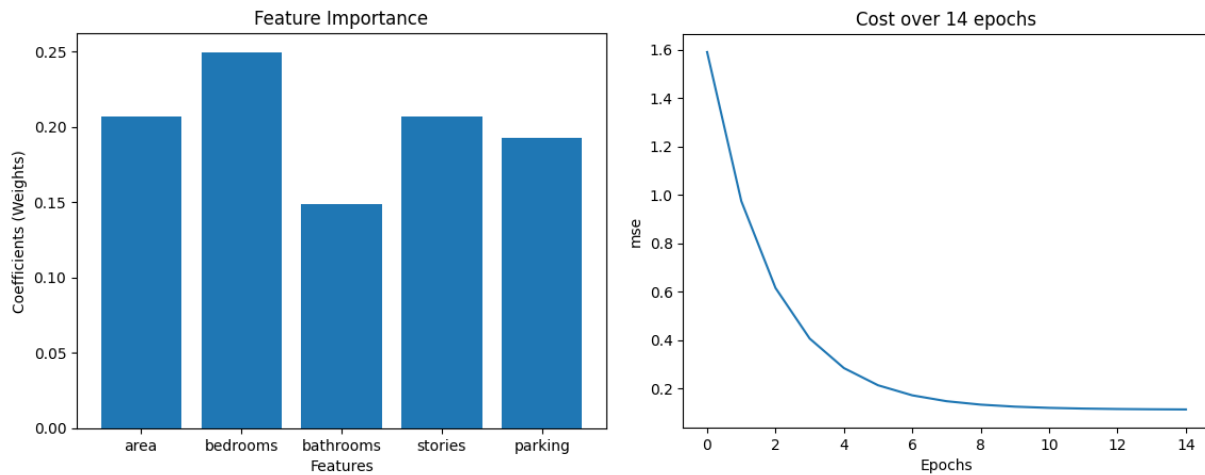


The exact same is true when trying to train multiple parameters without normalizing and standardizing the data.

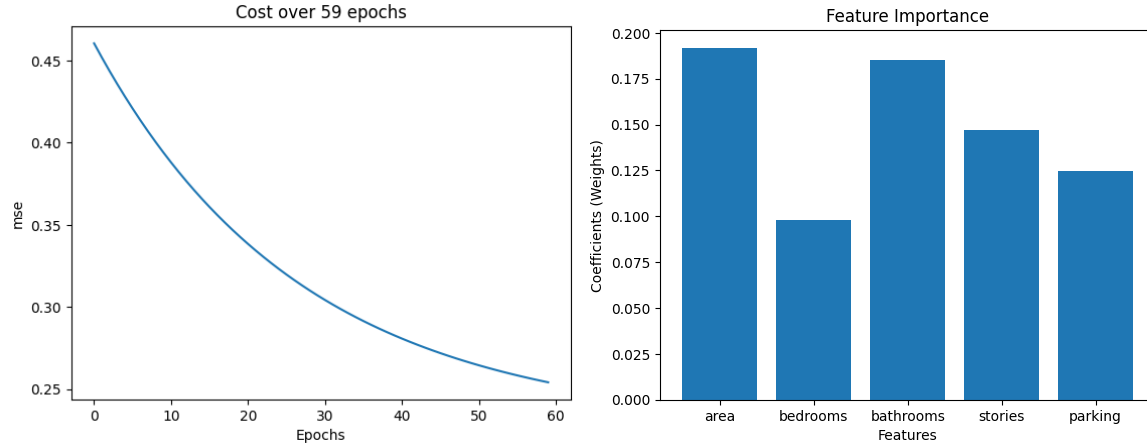
Question 2- Training with normalization and standardization

These are the graphs for training the model on a subset of variables (area, bedrooms, bathrooms, stories, and parking).

With normalization:

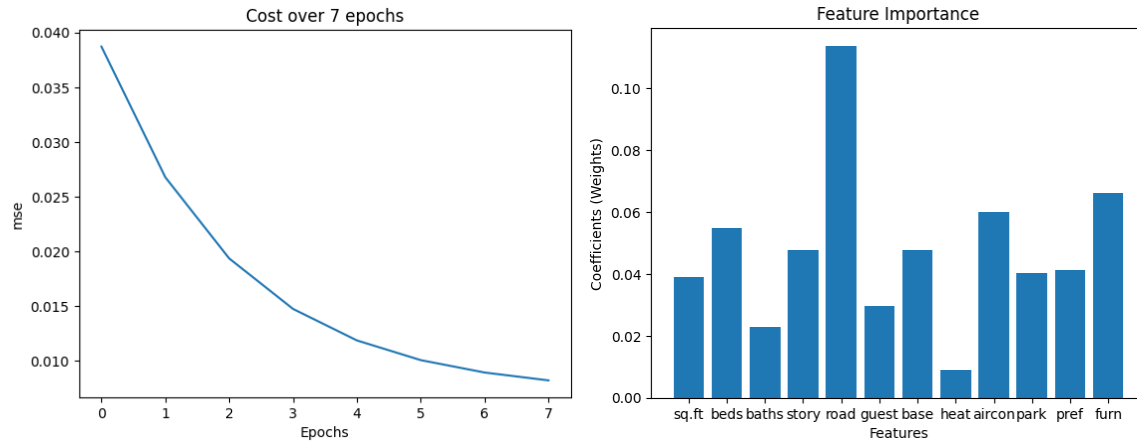


With standardization:

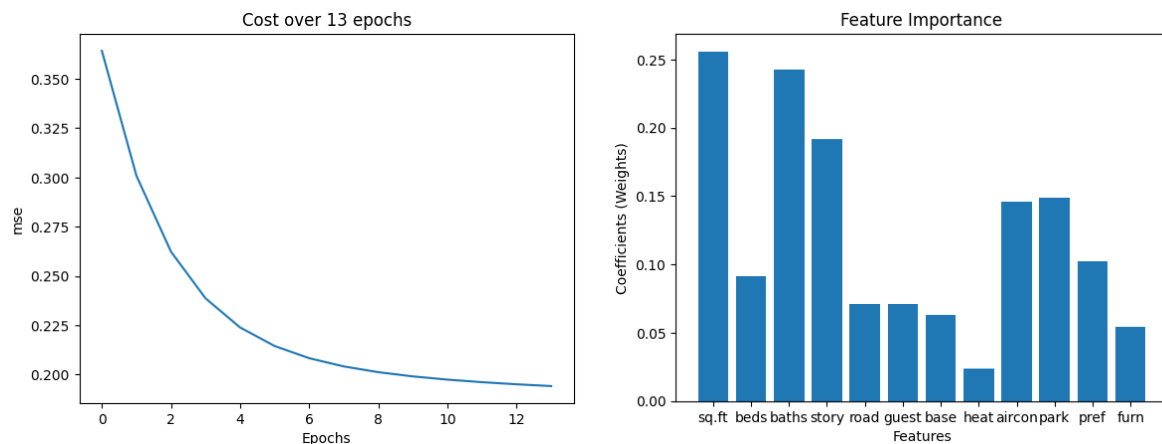


These are the graphs for training the model on all variables.

With normalization:



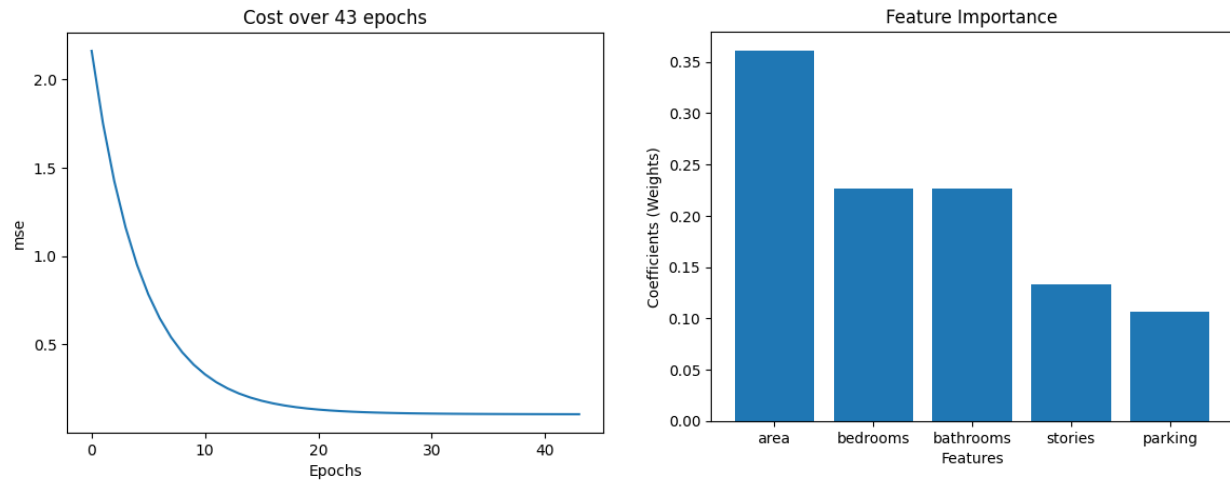
With standardization:



Normalization consistently produces better models. The final cost for the normalization training is always lower than the standardization training. Although, the normalization training seems to heavily prefer “road” for an unknown reason. There are some irregularities that are not present in the standardized models.

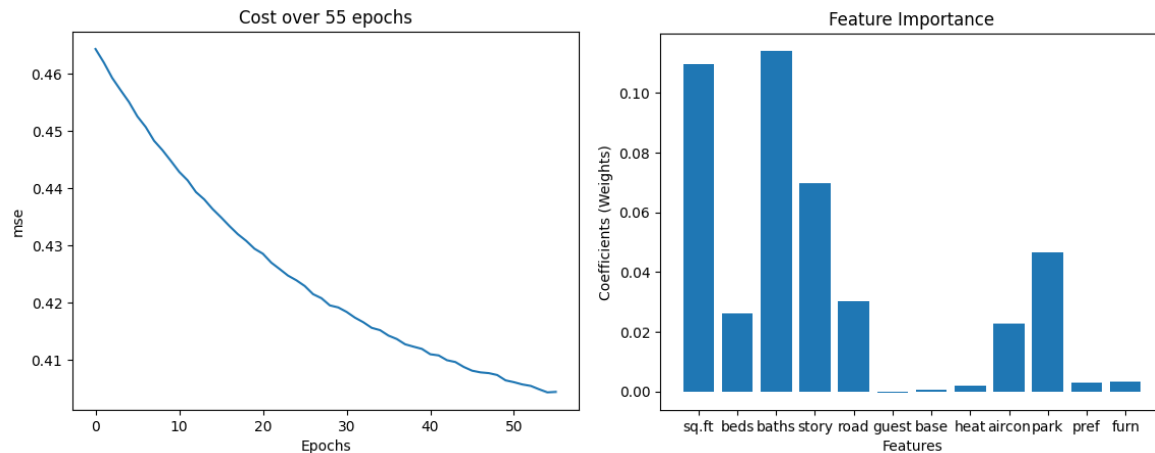
Question 3- Training with Lasso regulation (parameter penalty)

With normalization (subset of variables):



With standardization (subset of variables):

With standardization (all variables):



The lasso regularization adds cost for the current sum of the magnitudes of each theta value. This encourages the model to start eliminating parameters that don't have as much of an effect on the price. Guest, base, heart, preferred area, and furnished were all zeroed out. If you turn up the lambda constant (which determines the multiplier that's added to the cost

function), the model becomes inaccurate, but the only 4 parameters remaining will be the same subset of variables that we have been training on throughout this exercise.

Report link: [Anu78/intro-to-ml-hw: homework for ECGR-4105 @ uncc \(github.com\)](https://github.com/Anu78/intro-to-ml-hw: homework for ECGR-4105 @ uncc)