# CS 4375 Assignment 1

## Names of students in your group

1. Elias Khan
2. Tushar Wani

## Number of free late days used: 0

## Sources/references used:

1. <https://www.kaggle.com/code/yasserh/auto-mpg-prediction-comparing-top-ml-algorithms/notebook>
2. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>
3. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html#gallery-examples>

# Dataset

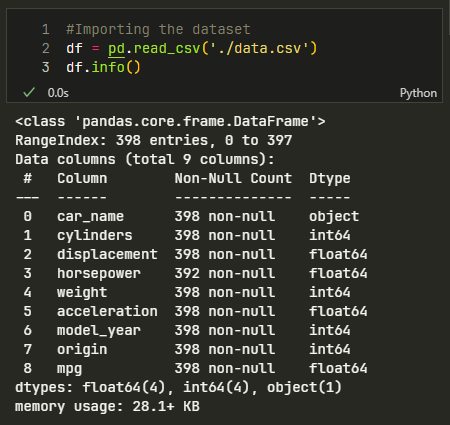
We used the Auto MPG dataset available on the [UC Irvine Machine Learning Repository](https://archive.ics.uci.edu/dataset/9/auto+mpg).

Created by Ross Quinlan and donated on 6th July, 1993.

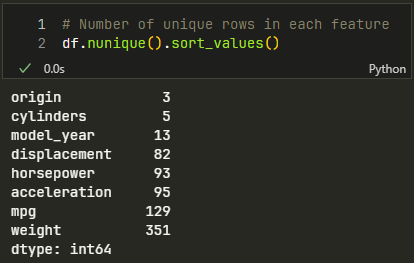
It is under a [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/legalcode) license.

# Data Analysis

The dataset has 9 features and 398 samples:



We can drop the car\_name column because we will not need it for training. Furthermore, we can drop the model year and origin, since they do not impact the mpg values. Now we can look at the number of unique rows in each feature:



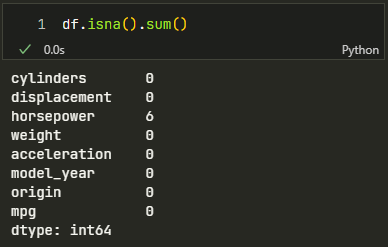
Now lets see if any of these are categorical features:



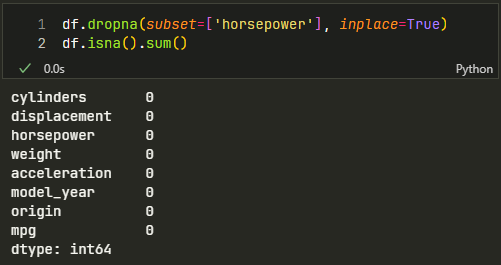
We can see that there aren’t really any categorical values in the dataset. So we do not need to convert any columns.

# Preprocessing

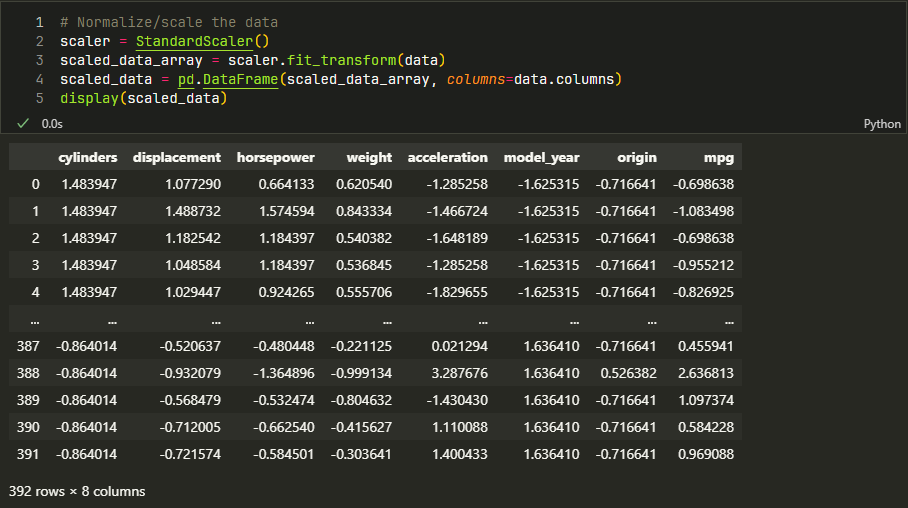
We can now check for empty values:



Since horsepower is the only column with empty values we can just drop those rows.



Now we can scale the data and finish preprocessing. For our split, we did an 80/20 split, using 80% of the data for training and 20% for testing.



# Report

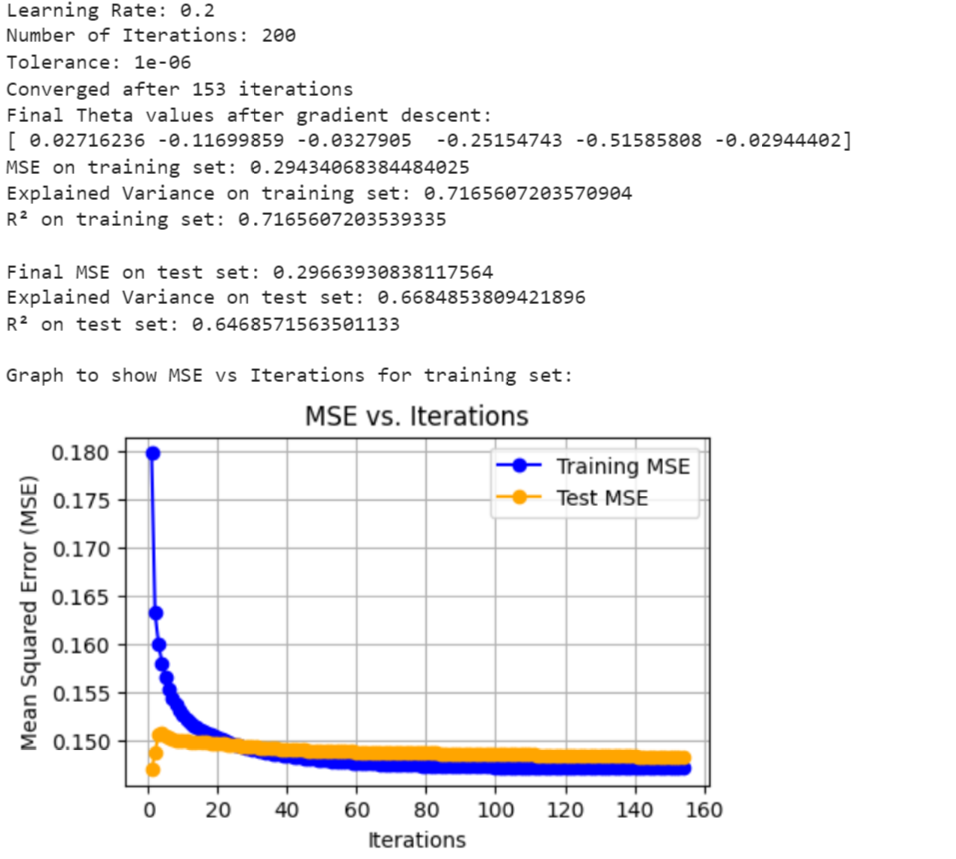
## Part I.

(Q)Answer this question: Are you satisfied that you have found the best solution? Explain:

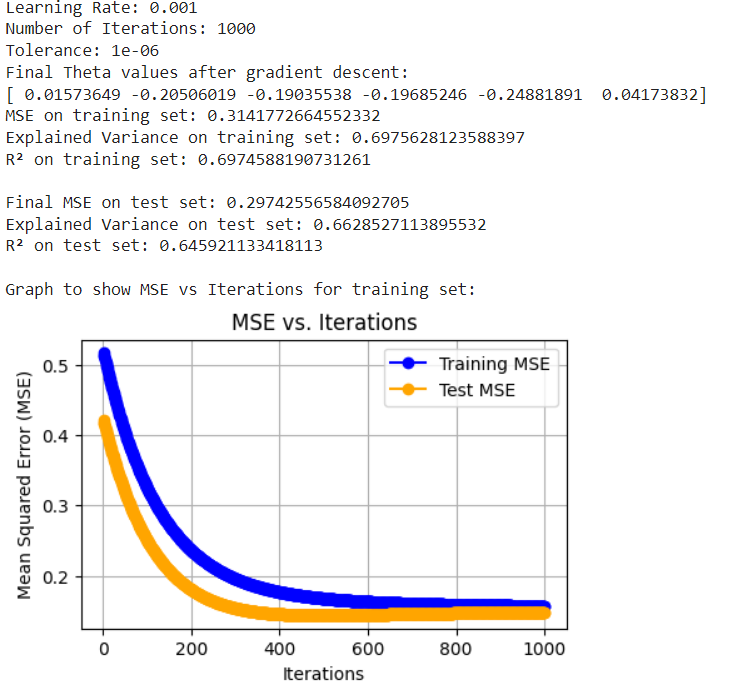
1. Yes, I was able to minimize the MSE as much as possible. Furthermore, my MSE using my gradient descent with optimal parameters is quite close to the MSE using the gradient descent packages. However, the R^2 values for my trials using test data were quite consistent and were not able to reach above 0.7. Most importantly, I was able to ensure that my model was trained adequately in order to avoid overfitting. The MSE for both training and testing were quite close.

Trials (7 total) :

Trial 1 (Best parameters):



Trial 2:



## Trial 3:

## 

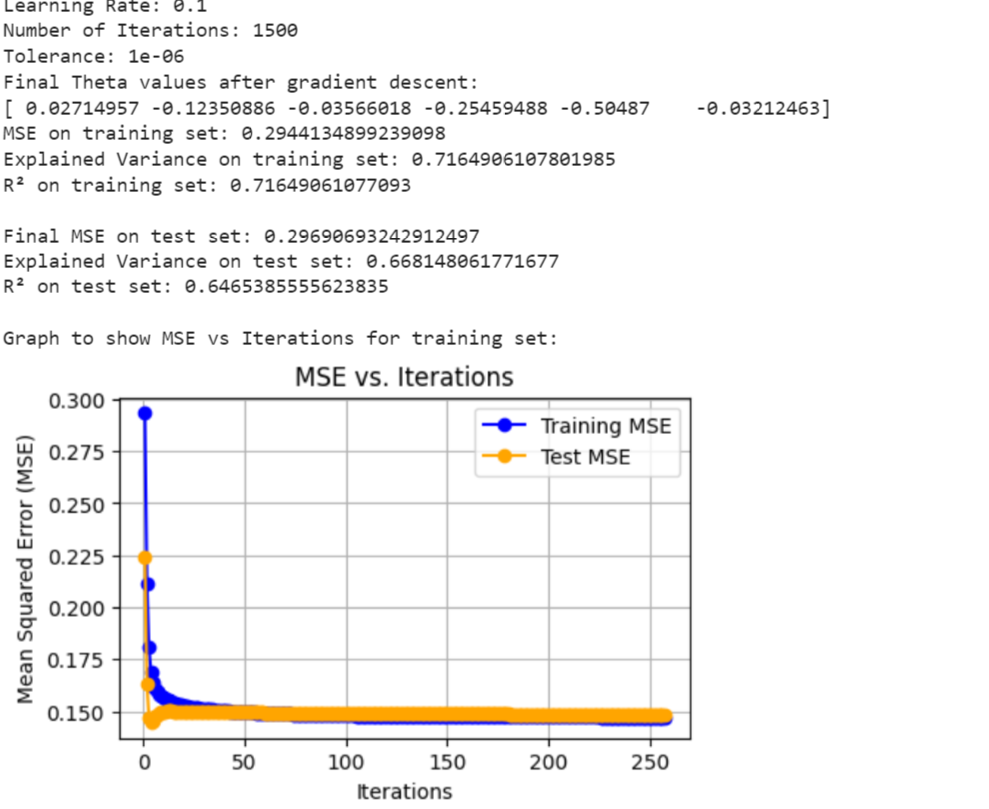
## 

## 

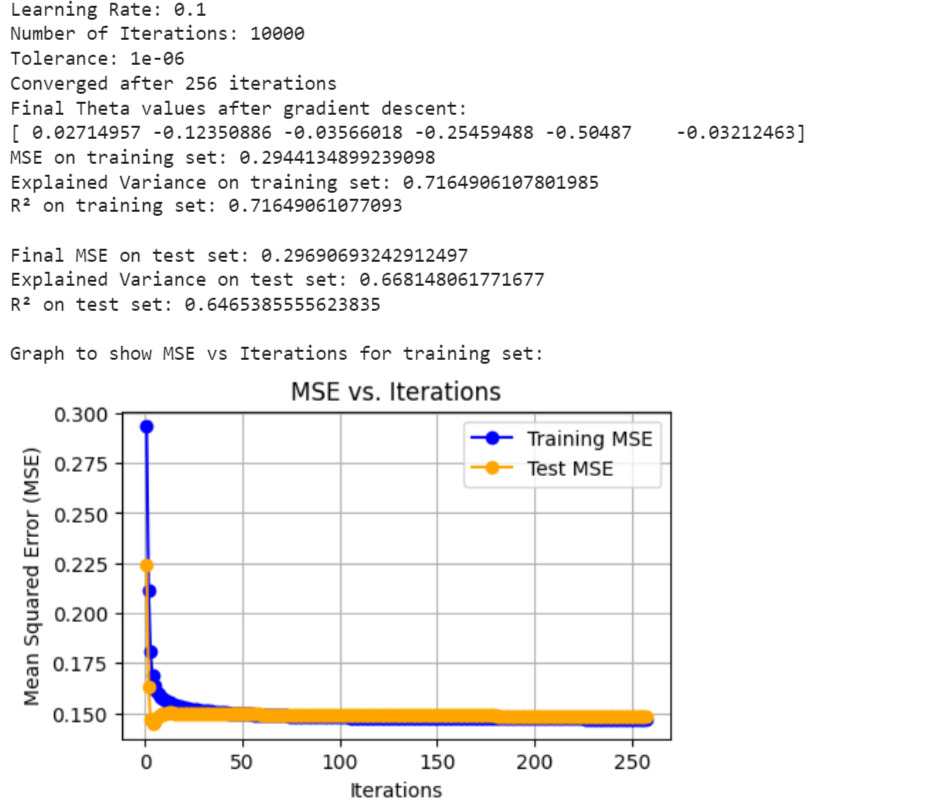
## 

## 

## Trial 4:

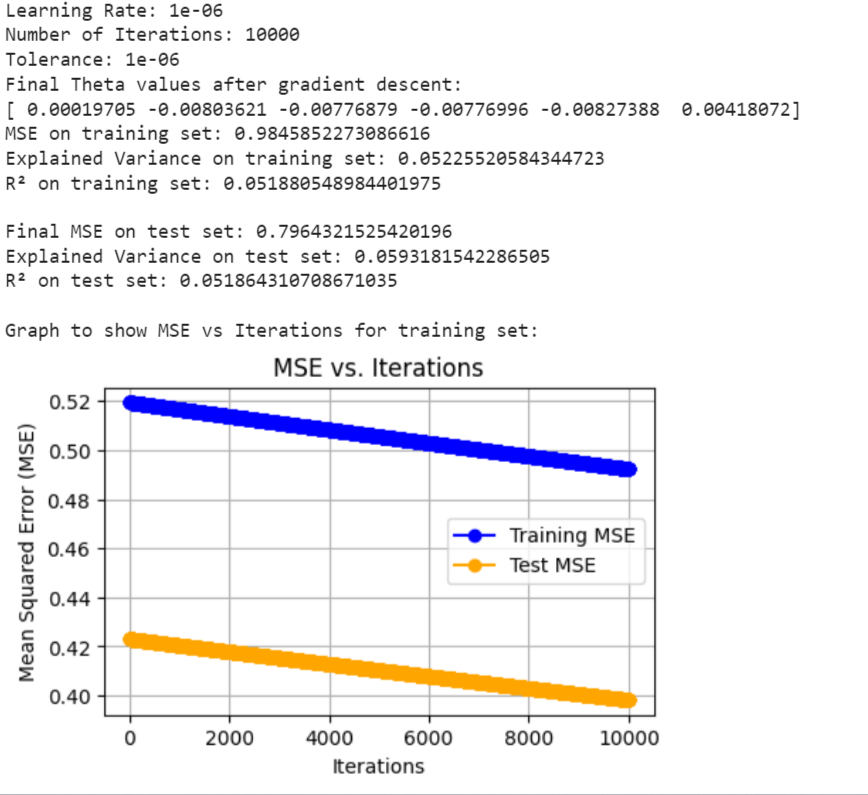


## Trial 5:

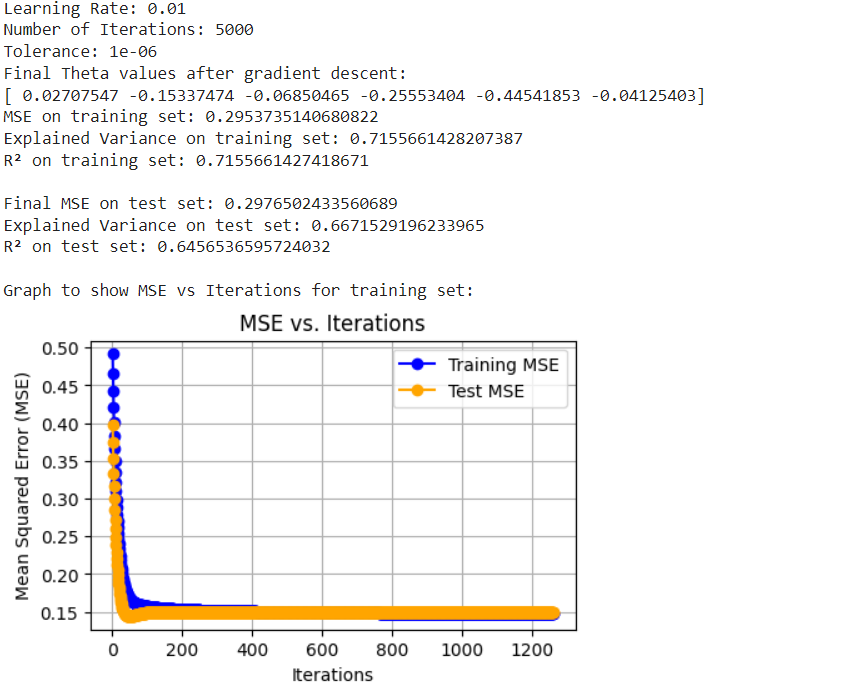


## Note: on trial 5, I noted the convergence point

Trial 6:



## Trial 7 :



## Graph to compare MPG (target) and a significant feature (horsepower)

Note: data is scaled

## 

## 

## 

## 

## 

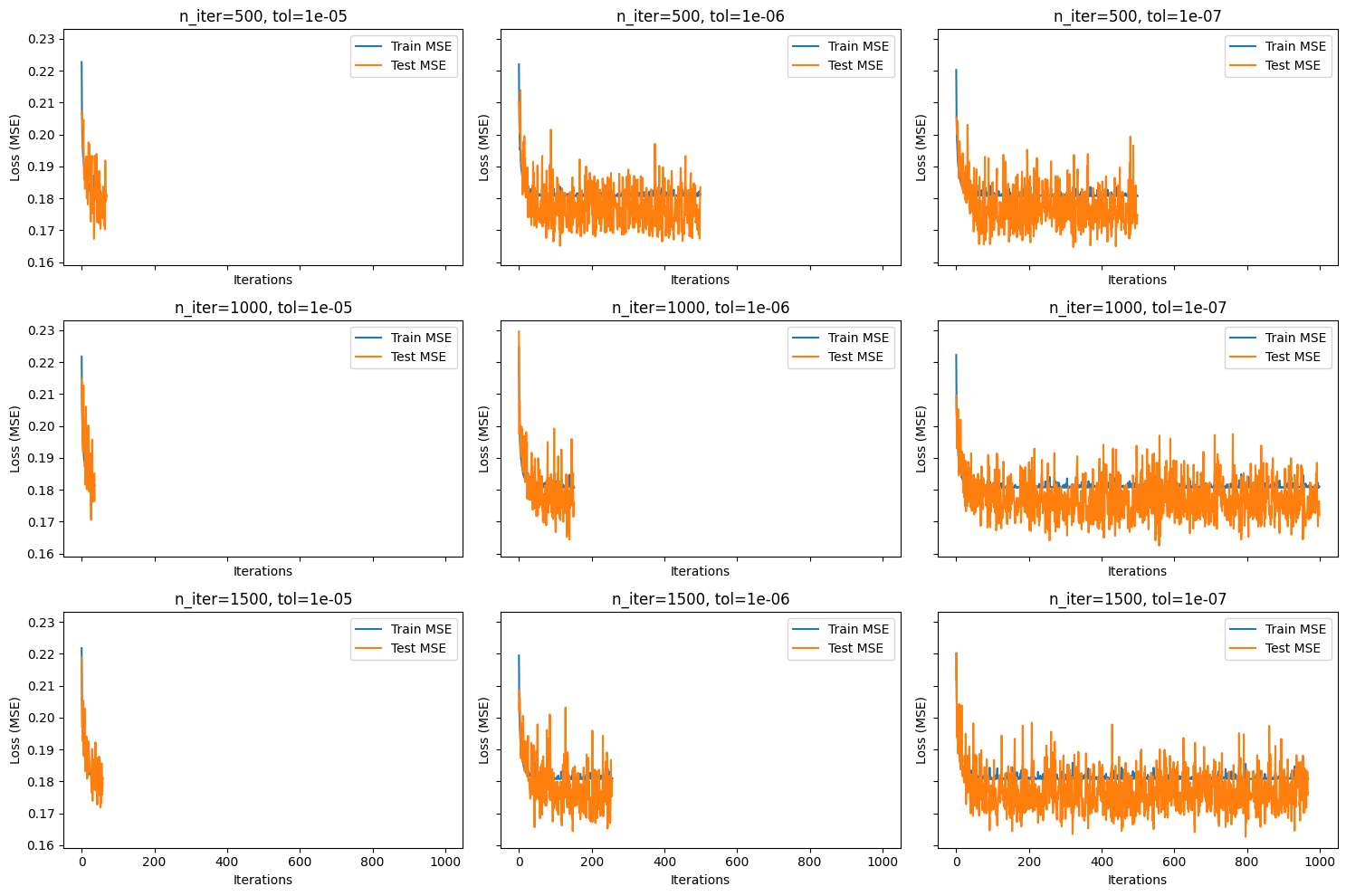
## 

## Part II.

Q) Are you satisfied that the package has found the best solution. How can you check? Explain.

A) I am satisfied by the best solution the package has found as can be seen below because the MSE and R^2 values I have gotten are fairly low and high respectively. That is what we want. Low error and R^2 to be closest to 1. This implies that the linear regression model captured the relationship between the features and the target variable appropriately.

Graph of trials and corresponding data as a json object:



Best Trial:

[

{

"n\_iter": 1000,

"tolerance": 1e-07,

"iterations": 1000,

"learning\_rate": "optimal",

"train\_mse": 0.18103311448869616,

"test\_mse": 0.1720483925127956,

"train\_explained\_variance": 0.825674287172178,

"test\_explained\_variance": 0.7963981215076104,

"r2": 0.7951800153899762,

"coefficients": [

-0.07365131439954227,

0.20026807189956727,

-0.10361437356256006,

-0.6546851353295307,

0.011478753590920557,

0.360310878345185,

0.1628768573967201

]

}

]

All Data:

[

{

"n\_iter": 500,

"tolerance": 1e-05,

"iterations": 70,

"learning\_rate": "optimal",

"train\_mse": 0.18093220454130998,

"test\_mse": 0.18106507018276416,

"train\_explained\_variance": 0.8257728104738532,

"test\_explained\_variance": 0.7857948545217767,

"r2": 0.7844458506900116,

"coefficients": [

-0.06148401737938368,

0.17151447048903887,

-0.10989940933805055,

-0.6586654175919348,

0.013480625516149055,

0.36094743915172633,

0.16939662712508965

]

},

{

"n\_iter": 500,

"tolerance": 1e-06,

"iterations": 500,

"learning\_rate": "optimal",

"train\_mse": 0.18122175662859794,

"test\_mse": 0.18351330061459326,

"train\_explained\_variance": 0.8254965317941108,

"test\_explained\_variance": 0.7832512487832806,

"r2": 0.7815312839681415,

"coefficients": [

-0.07830059663769509,

0.19454344660664102,

-0.11273252057433007,

-0.6696497455640522,

0.011178837374194351,

0.36326386026513796,

0.1737663621592197

]

},

{

"n\_iter": 500,

"tolerance": 1e-07,

"iterations": 500,

"learning\_rate": "optimal",

"train\_mse": 0.1807315182407331,

"test\_mse": 0.17479249506123498,

"train\_explained\_variance": 0.8259638337182644,

"test\_explained\_variance": 0.7933628302306084,

"r2": 0.7919132191500876,

"coefficients": [

-0.07242870439830587,

0.20085997431824099,

-0.10391532063029812,

-0.6660632734386901,

0.014521683812451124,

0.3623164877962172,

0.16367448646508861

]

},

{

"n\_iter": 1000,

"tolerance": 1e-05,

"iterations": 37,

"learning\_rate": "optimal",

"train\_mse": 0.1814585599202738,

"test\_mse": 0.1765501452195474,

"train\_explained\_variance": 0.8252642248737255,

"test\_explained\_variance": 0.7912916306915107,

"r2": 0.789820773686823,

"coefficients": [

-0.03359585447403059,

0.12881342716833788,

-0.11693044306307804,

-0.6186401895740973,

-0.006458550331250777,

0.3629235986579452,

0.1614261739789525

]

},

{

"n\_iter": 1000,

"tolerance": 1e-06,

"iterations": 153,

"learning\_rate": "optimal",

"train\_mse": 0.1807056679498067,

"test\_mse": 0.17583593998954958,

"train\_explained\_variance": 0.8259878173236495,

"test\_explained\_variance": 0.7921556323788947,

"r2": 0.7906710199581195,

"coefficients": [

-0.06859841252168682,

0.19745327847004648,

-0.10839522408963365,

-0.6666331736717821,

0.011782634747584227,

0.3607696760002406,

0.1652298107067877

]

},

{

"n\_iter": 1000,

"tolerance": 1e-07,

"iterations": 1000,

"learning\_rate": "optimal",

"train\_mse": 0.18103311448869616,

"test\_mse": 0.1720483925127956,

"train\_explained\_variance": 0.825674287172178,

"test\_explained\_variance": 0.7963981215076104,

"r2": 0.7951800153899762,

"coefficients": [

-0.07365131439954227,

0.20026807189956727,

-0.10361437356256006,

-0.6546851353295307,

0.011478753590920557,

0.360310878345185,

0.1628768573967201

]

},

{

"n\_iter": 1500,

"tolerance": 1e-05,

"iterations": 59,

"learning\_rate": "optimal",

"train\_mse": 0.1809583476125205,

"test\_mse": 0.18063329525251043,

"train\_explained\_variance": 0.825749222005573,

"test\_explained\_variance": 0.7862857566787738,

"r2": 0.7849598696429232,

"coefficients": [

-0.05676315988012552,

0.14657678905863877,

-0.10036331281479112,

-0.648561441188719,

0.012565789618579706,

0.3602093990524945,

0.16426577904462056

]

},

{

"n\_iter": 1500,

"tolerance": 1e-06,

"iterations": 258,

"learning\_rate": "optimal",

"train\_mse": 0.18092016715223244,

"test\_mse": 0.1801212886345394,

"train\_explained\_variance": 0.8257876515472728,

"test\_explained\_variance": 0.7870830806653801,

"r2": 0.7855694027288267,

"coefficients": [

-0.06876812489336624,

0.19729883215466096,

-0.11058814936612171,

-0.6741688008255706,

0.01738549247106434,

0.36722047578616823,

0.1659310408270187

]

},

{

"n\_iter": 1500,

"tolerance": 1e-07,

"iterations": 969,

"learning\_rate": "optimal",

"train\_mse": 0.1807161237039362,

"test\_mse": 0.1766222945777917,

"train\_explained\_variance": 0.8259851394782782,

"test\_explained\_variance": 0.7913391477389472,

"r2": 0.7897348813966982,

"coefficients": [

-0.0811280885663703,

0.2007489387278843,

-0.10460446388117191,

-0.6648808588358757,

0.009171694503857021,

0.3630570836329743,

0.16521579271123316

]

}

]