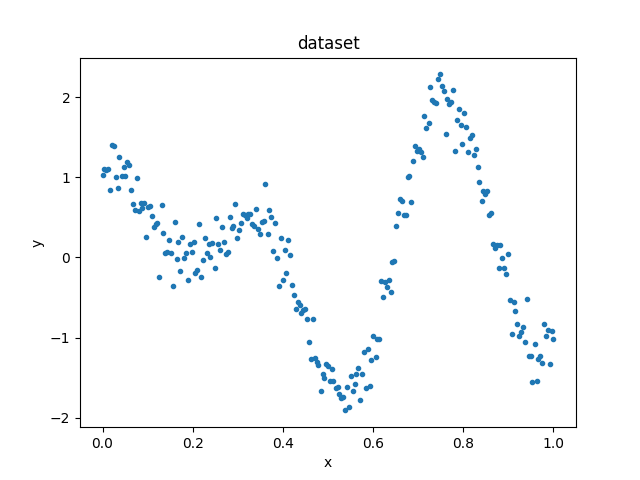
Q1

problem: performing linear regression on a signal

**a- plot the data**

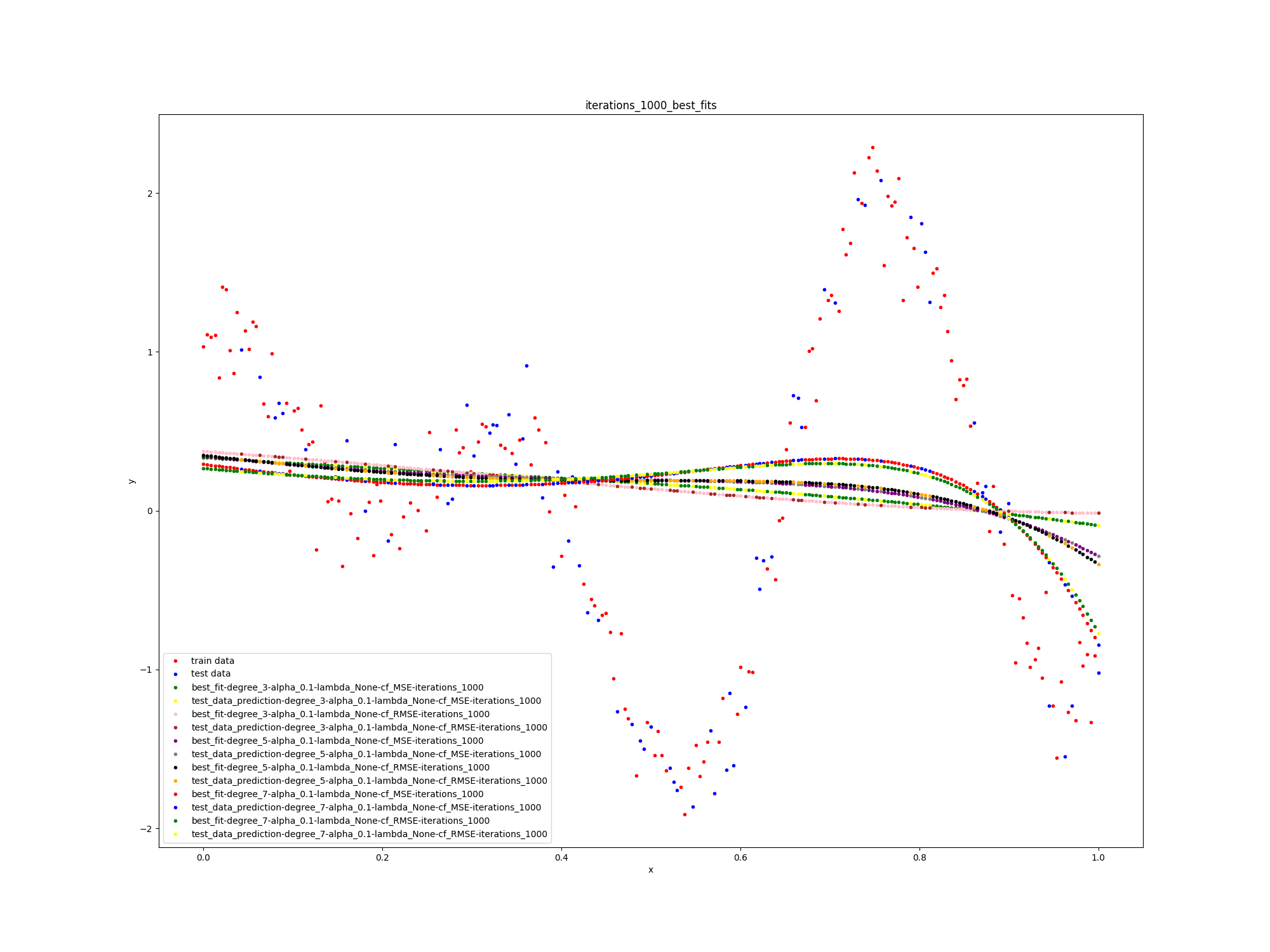
**b- use gradient descent algorithm to fit a model on data by adding polynomial features with degrees 3, 5, and 7. report the training and testing costs. repeat the algorithm for 1000 and 10000 iterations. does overfitting occur? in which cases? use cost functions MSE and RMSE to evaluate the models and compare the results.**

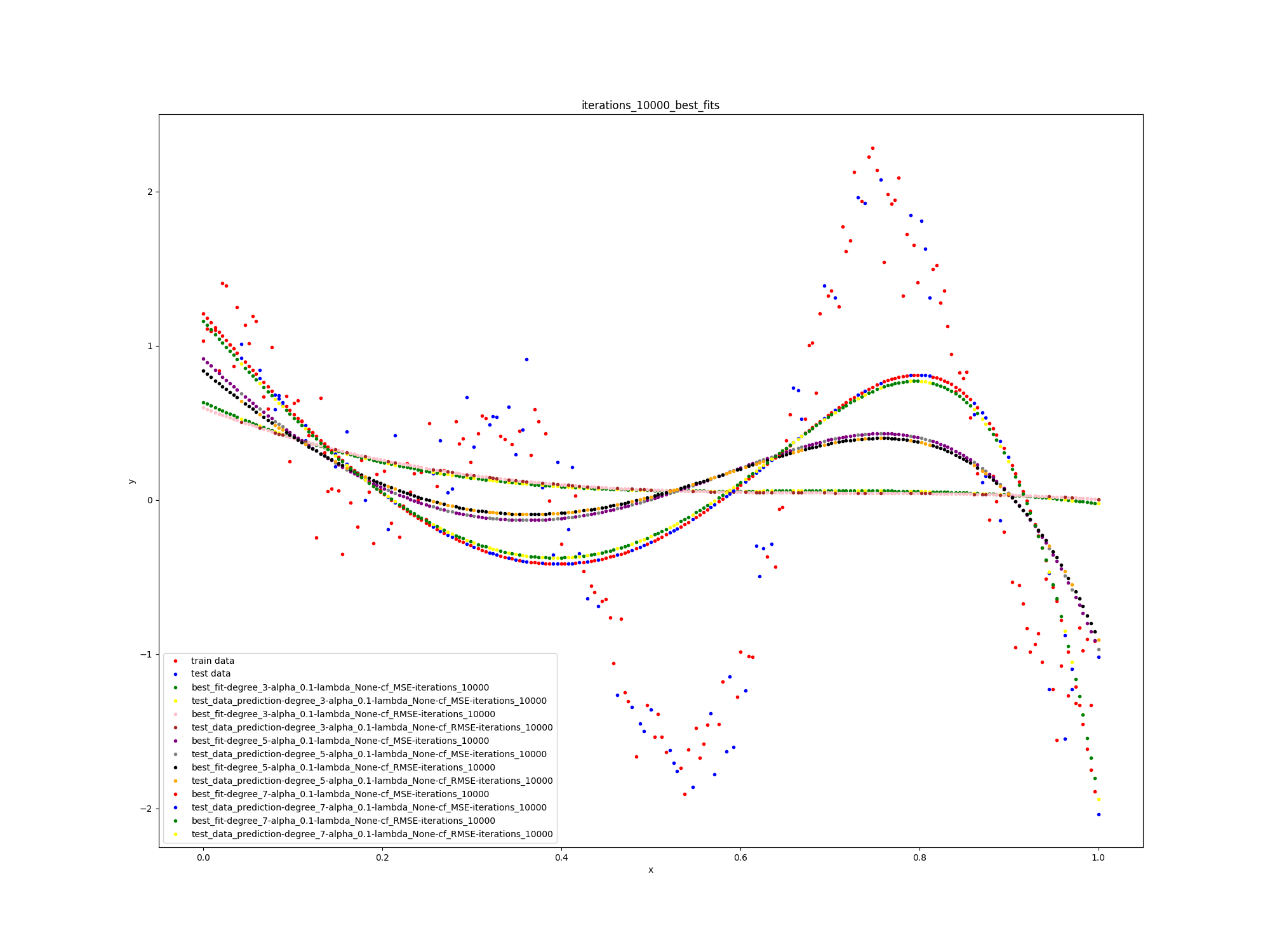
best fits for combinations of degree and cost function:

file names:

iterations\_1000\_best\_fits.png

iterations\_10000\_best\_fits.png



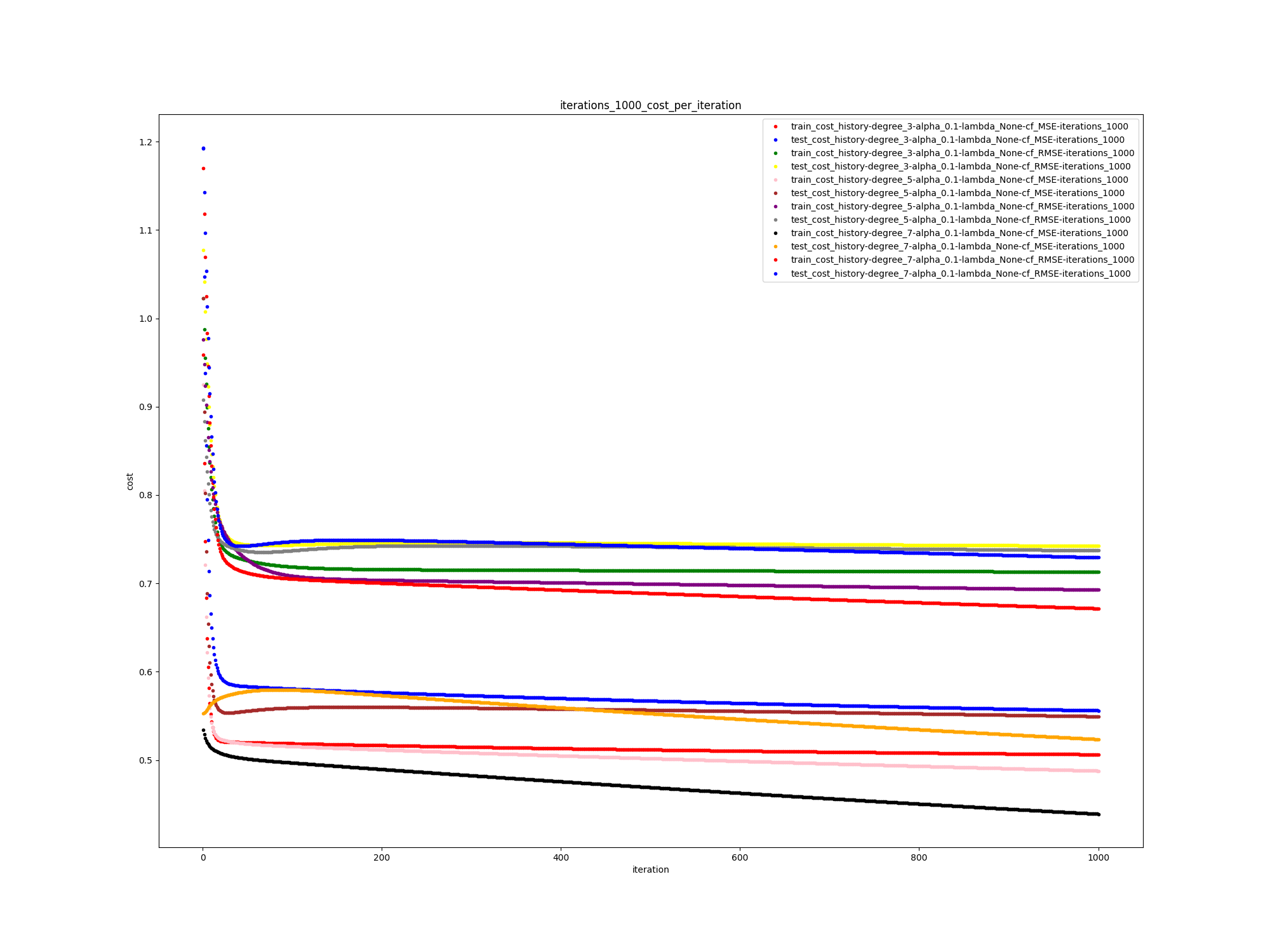


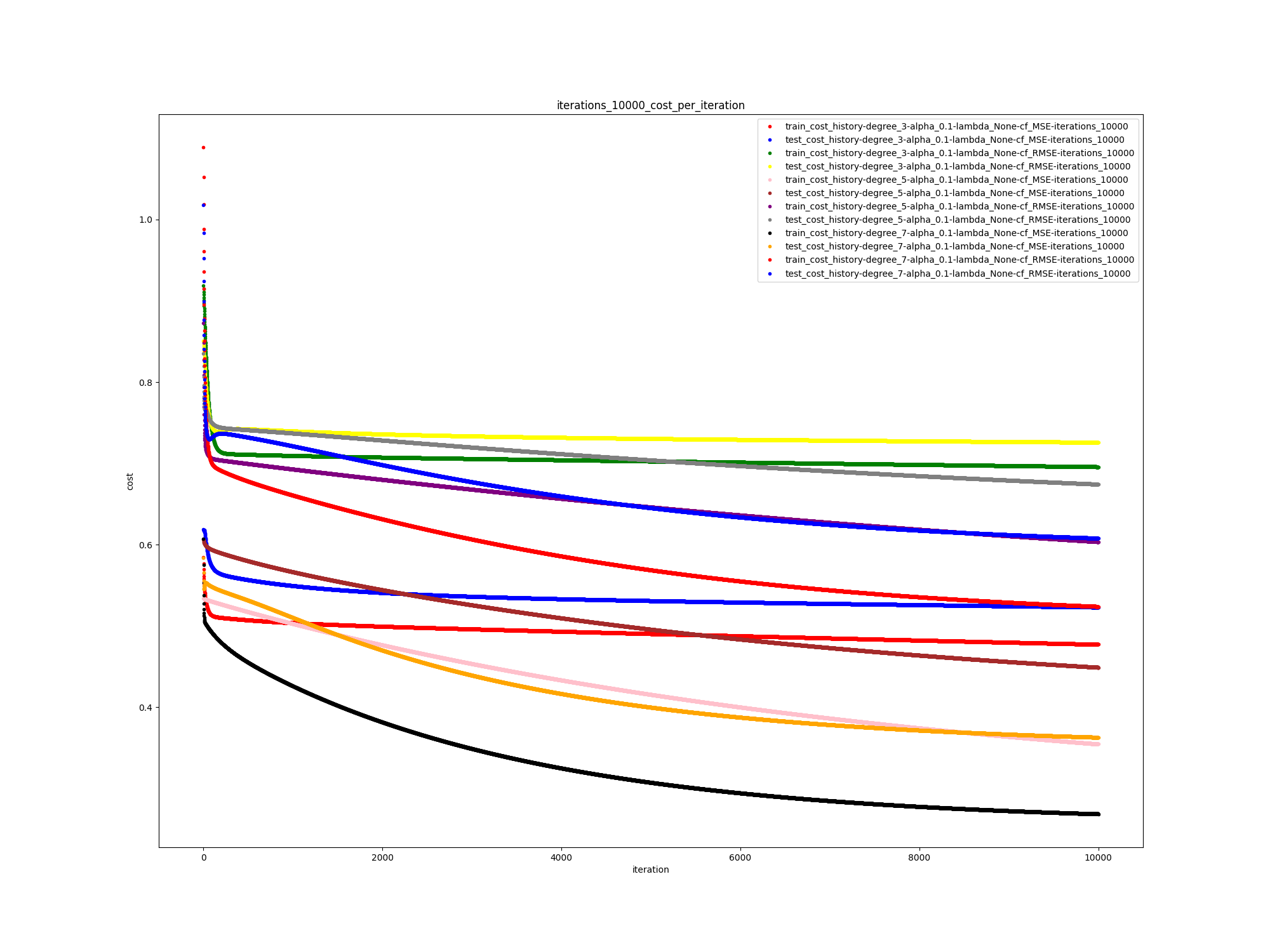
training and testing costs for combinations of degree and cost function:

file names:

iterations\_1000\_cost\_per\_iteration.png

iterations\_10000\_cost\_per\_iteration.png



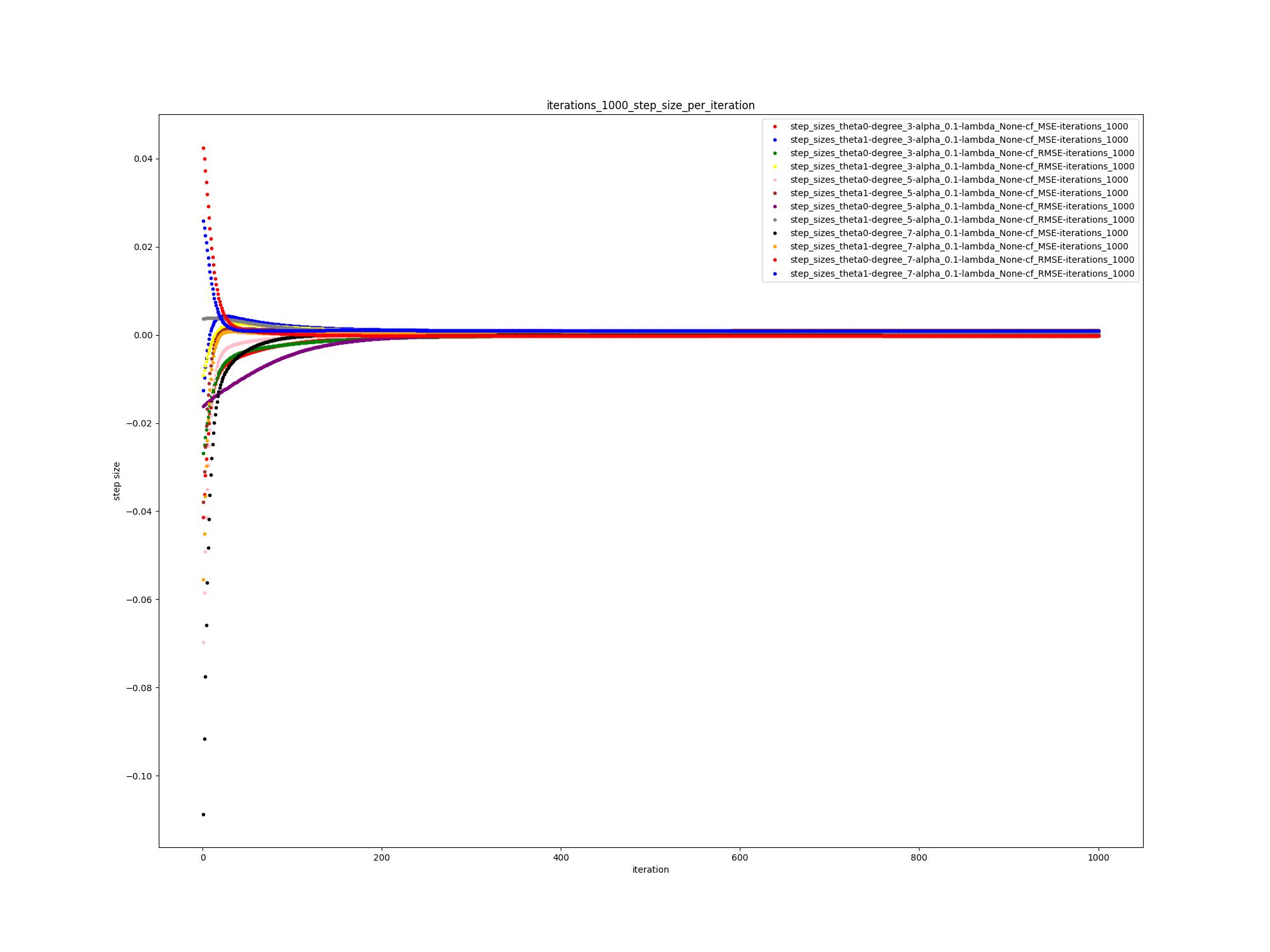


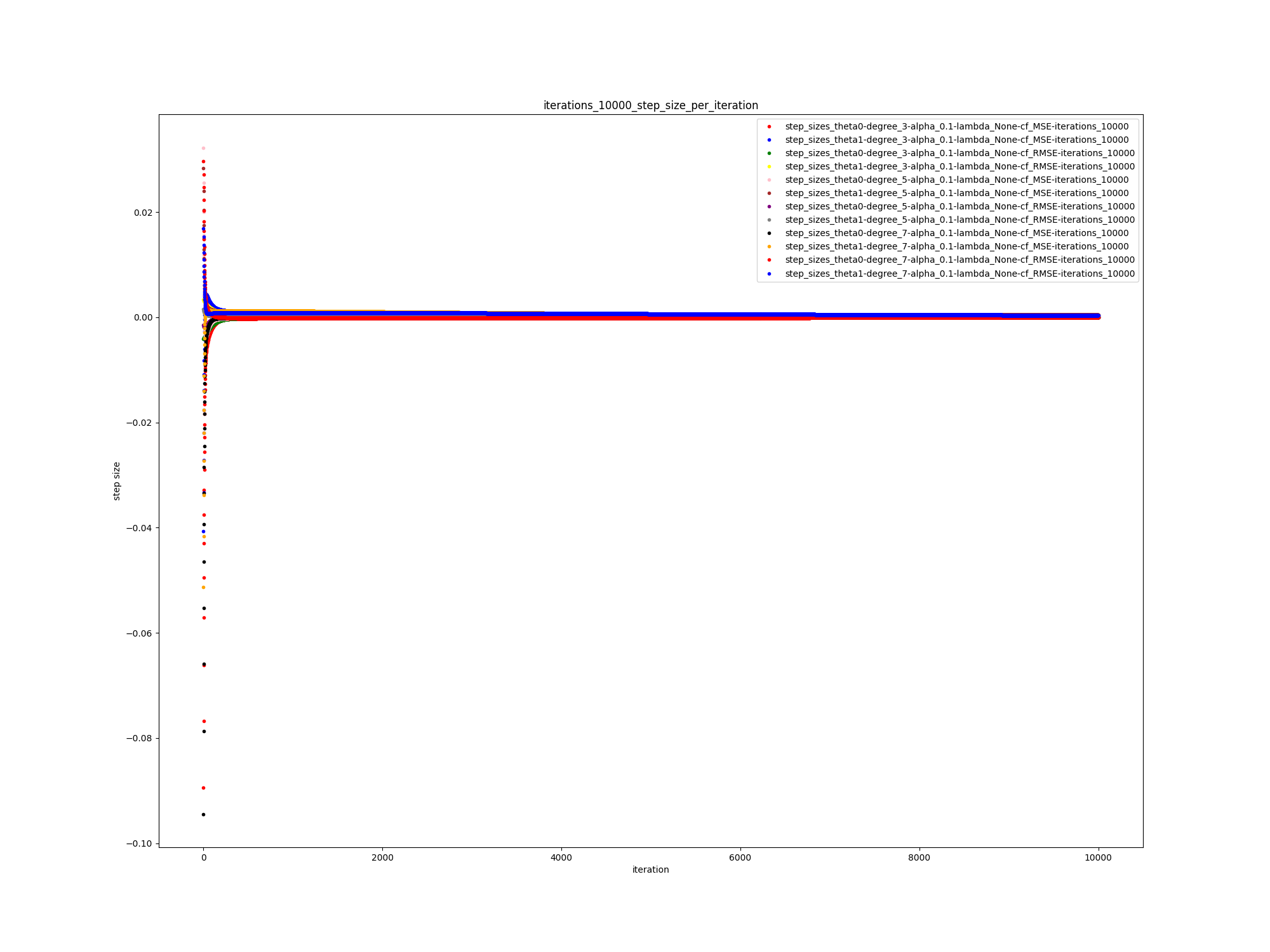
step sizes for combinations of degree and cost function:

file names:

iterations\_1000\_step\_size\_per\_iteration.png

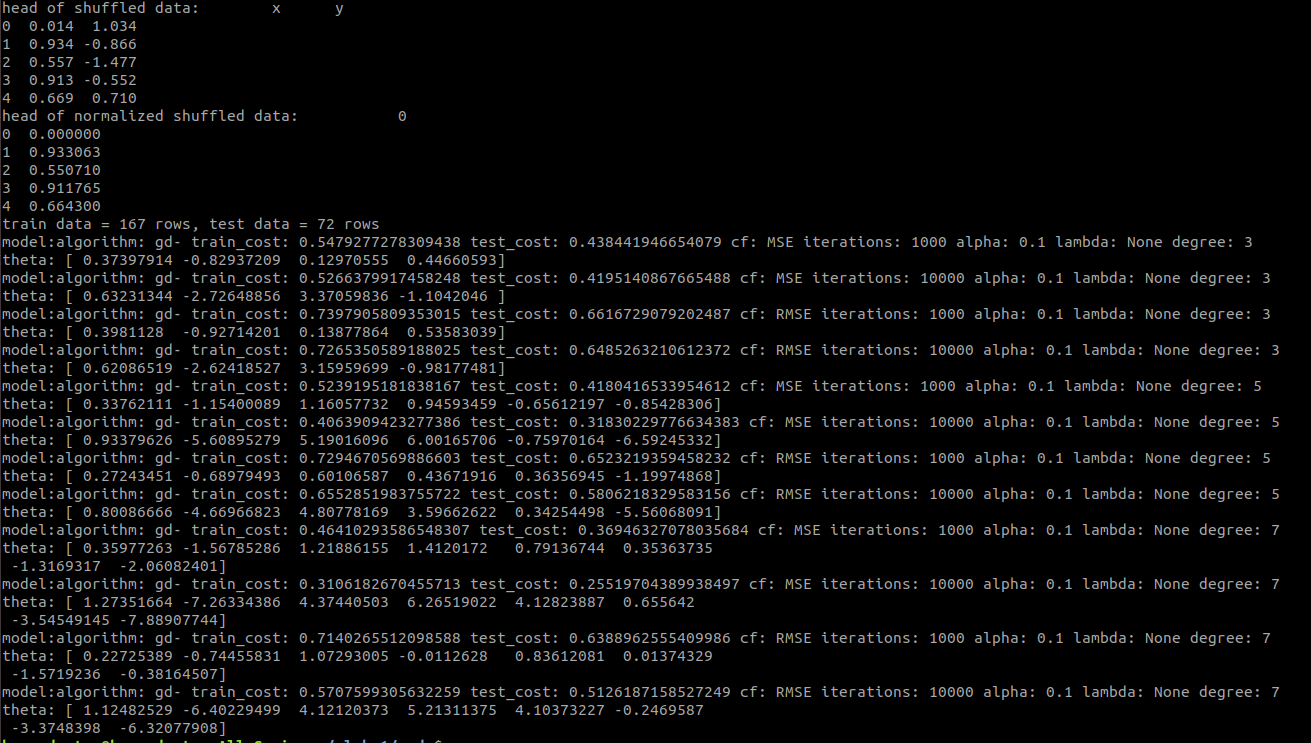
iterations\_10000\_step\_size\_per\_iteration.png





final results:

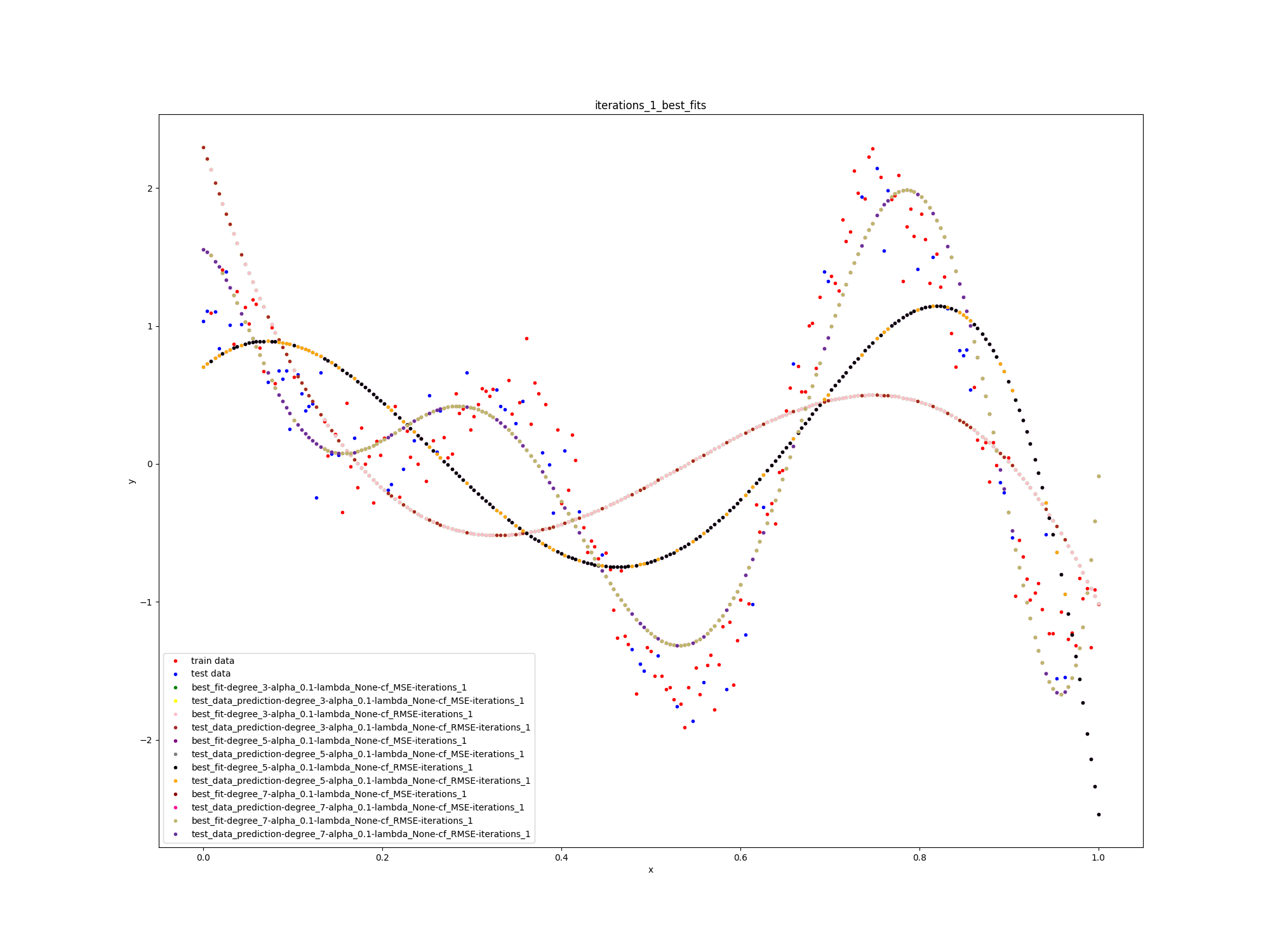
All models plus other figures are available in the folder q1-section\_b-outputs



**c) solve the previous problem with the normal equation.**

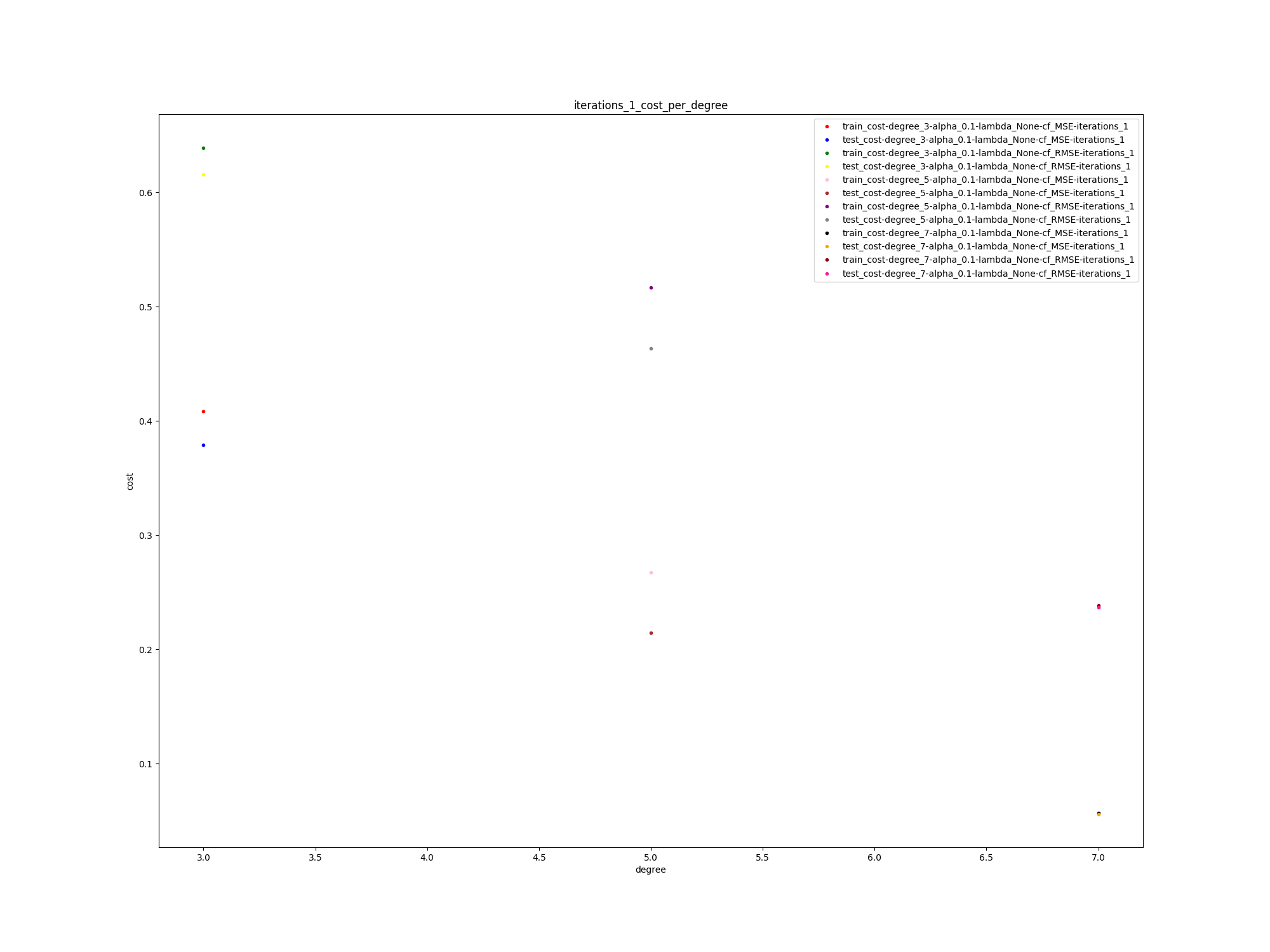
best fits:

file name: iterations\_1\_best\_fits.png

note: assume there is no **alpha** and **iterations** in the figure.

training and testing costs:

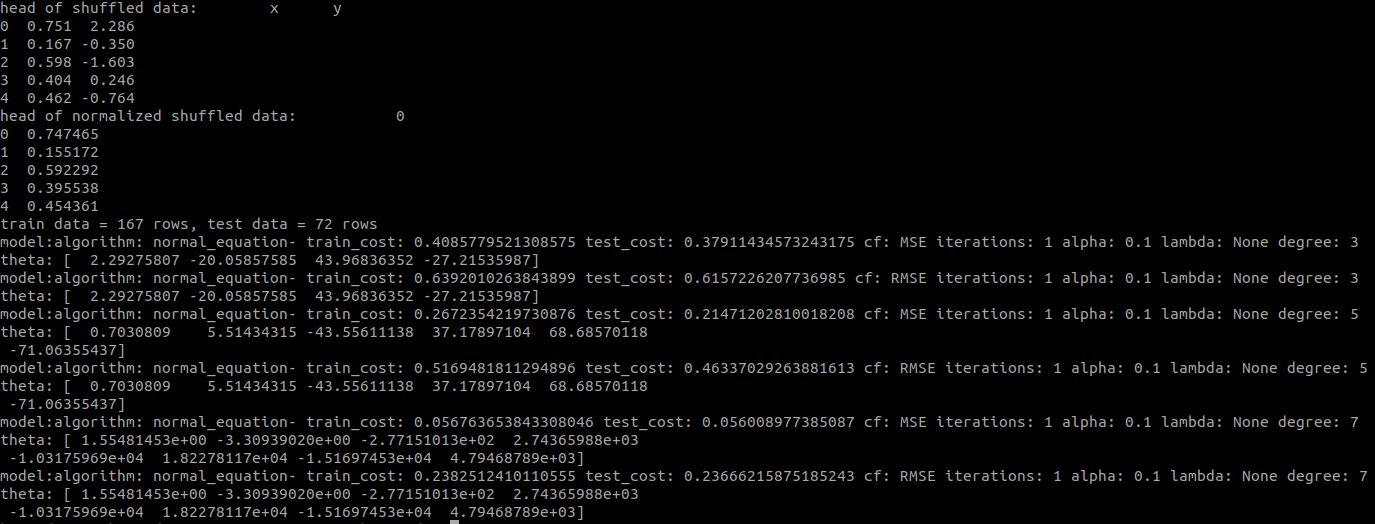
file name: iterations\_1\_cost\_per\_degree.png

note: assume there is no **alpha** and **iterations** in the figure.

final results:

All models plus other figures are available in the folder q1-section\_c-outputs

note: assume there is no **alpha** and **iterations** in the figure.



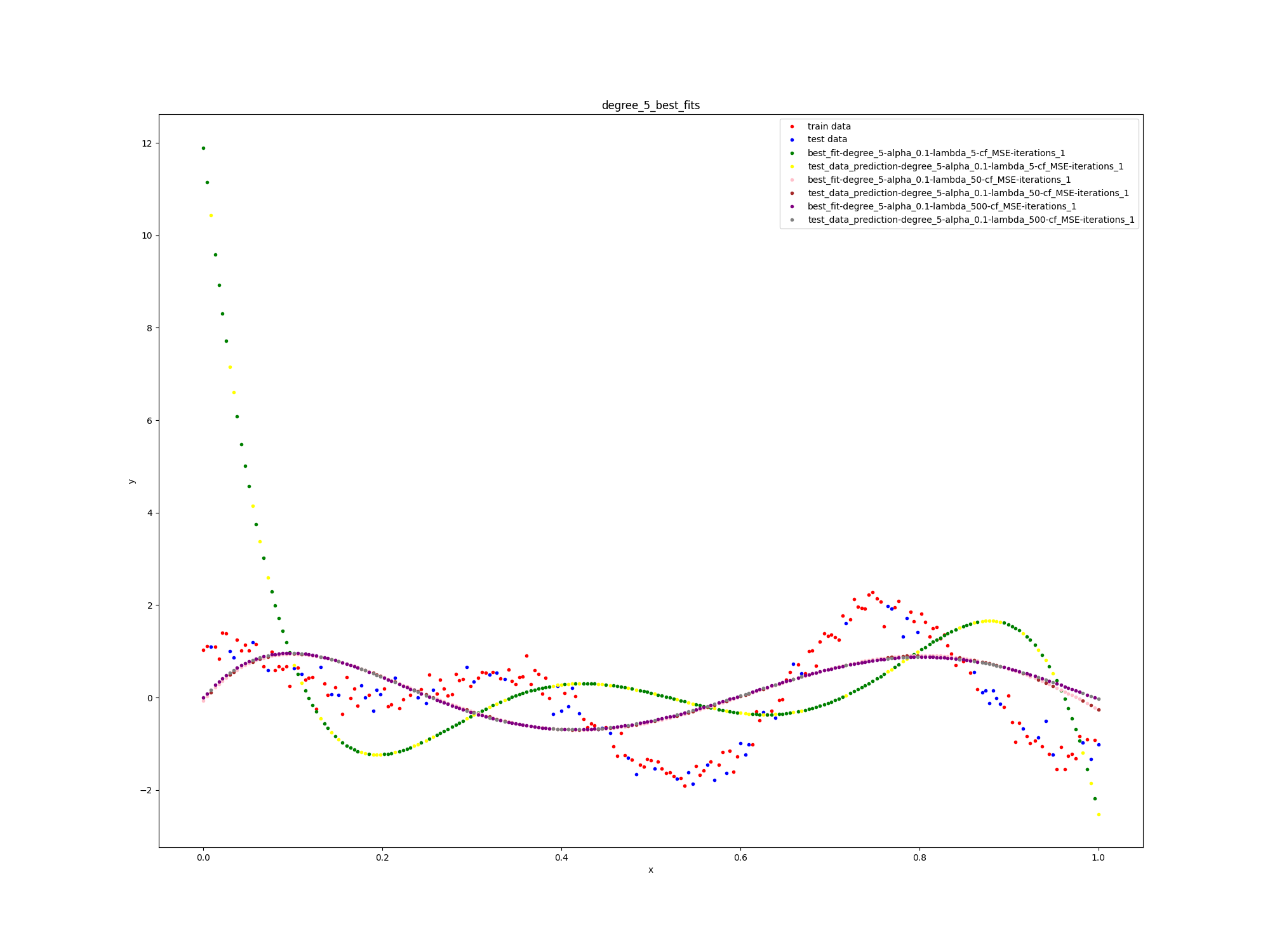
Absolutely, the normal equation algorithm has performed far better than the gradient descent algorithm since it calculates the global minimum whereas in gradient descent we move forward to the local minimum (in linear regression, global minimum) step by step and it's possible that we never reach the minimum point because of the too large step size or too low step size, or too few iterations.

**d- use lambda = 5, 50, and 500 and resolve the previous question (section c).**

best fits:

file name: degree\_5\_best\_fits.png

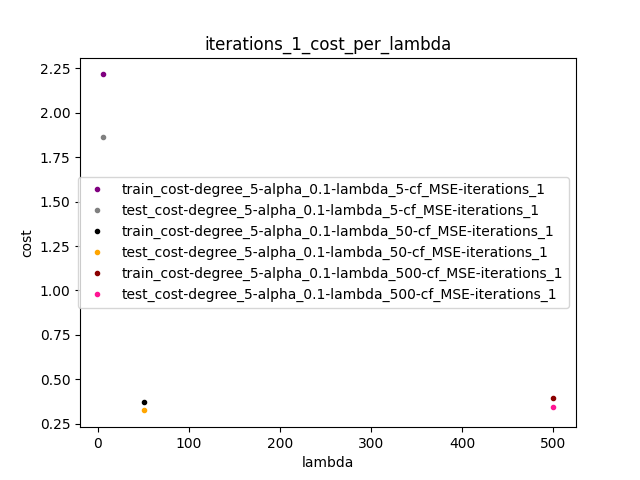
note: assume there is no **alpha** and **iterations** in the figure.



training and testing cost per lambda:

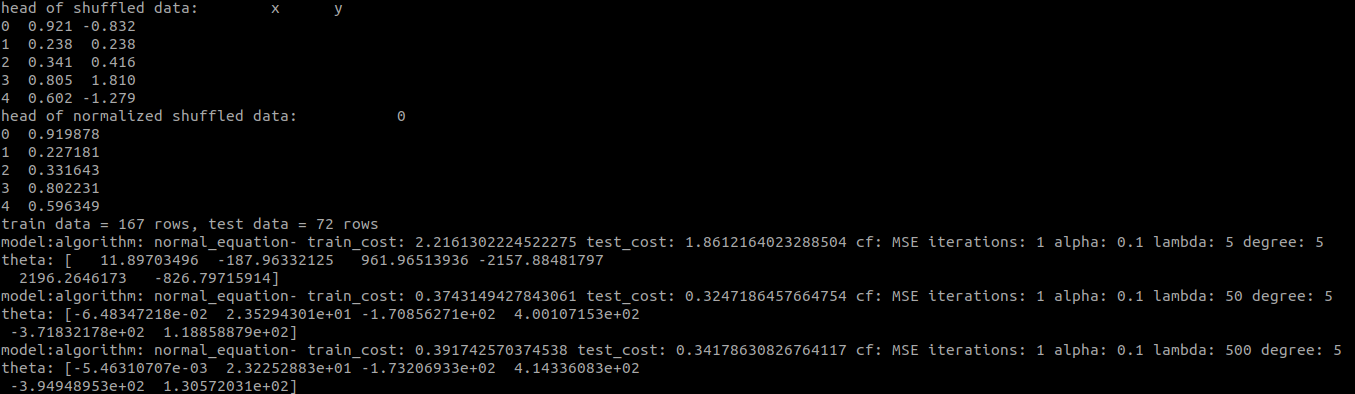
file name: iterations\_1\_cost\_per\_lambda.png

note: assume there is no **alpha** and **iterations** in the figure.



final results:

All models plus other figures are available in the folder q1-section\_d-outputs

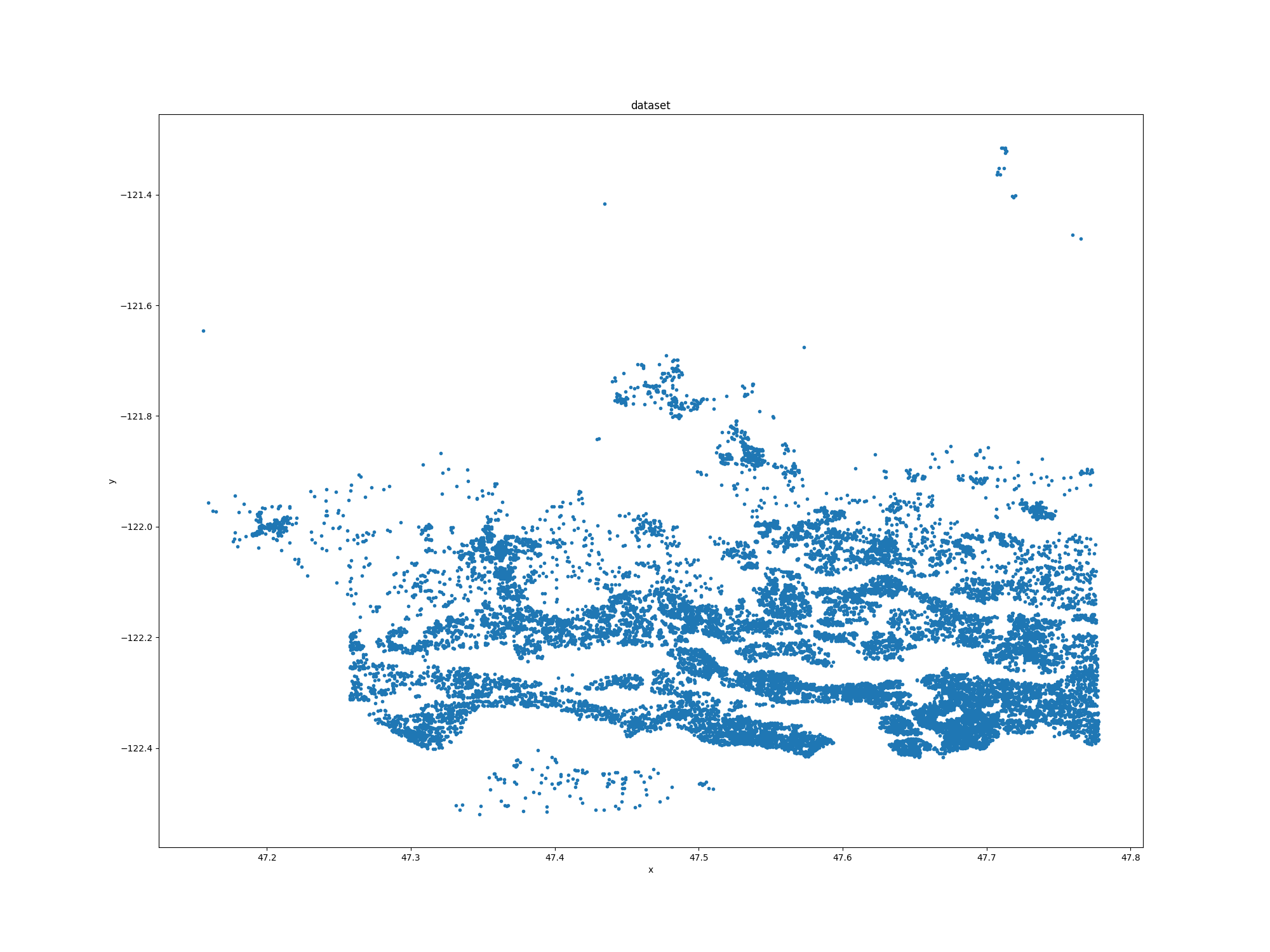
note: assume there is no **alpha** and **iterations** in the figure.

We can conclude that increasing the absolute value of lambda decreases the values of theta since it prevents the theta vector to get large values because the cost function returns a high cost with big elements in theta vector which is not desired.

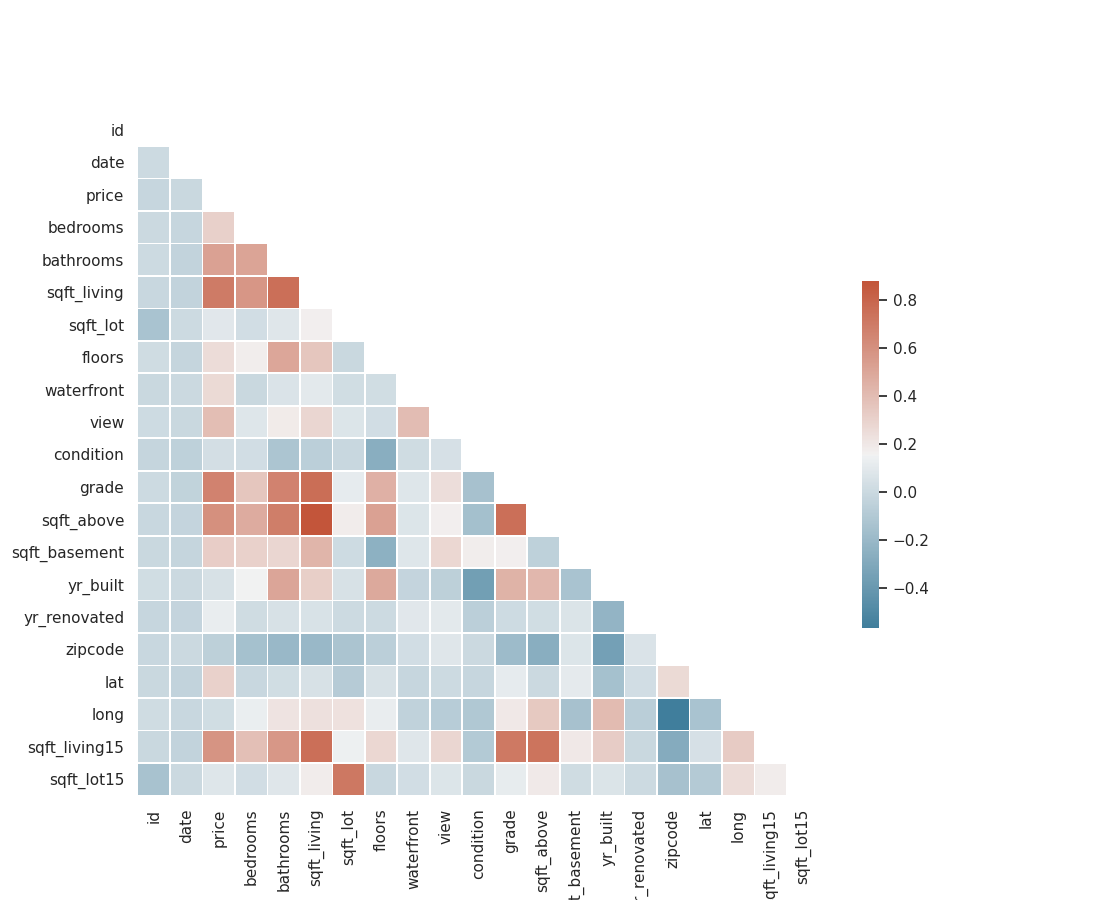
2- problem: house price prediction

**a- plot the data with lat and long attributes.**

x = lat, y = long



**b- plot correlation hit map between dataset features.**



**c- is it possible to remove one or more of the features? is so, why?**

yes, we can delete **<id>** and **<date>** columns since they don't provide any useful information and insight, in other words, their **gain ratio** is too low.

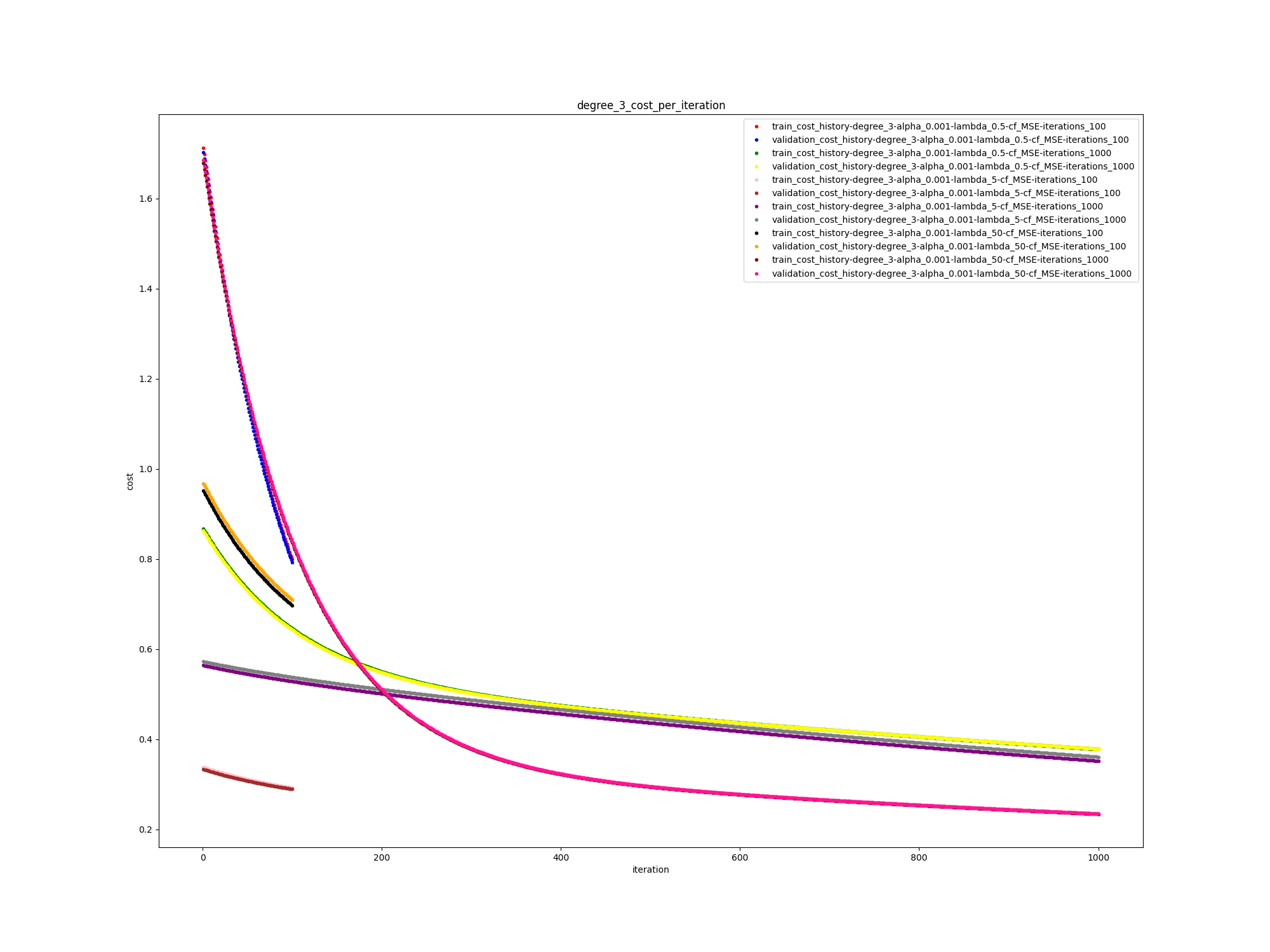
also, we can remove **<sqft\_above>** or **<sqft\_living>** because of the too high correlation, in other words, they provide the same information for the problem.

**d- use gradient descent algorithm to fit a regression model. choose the best model as far as possible. once use all of the features and again remove selected features in section c then try again.**

to solve this problem, 50% of data is used to train the models, 30% is used for validation and the rest is used for testing. the dataset is shuffled and normalized before training.

"id", "date", "zipcode", and "sqft\_above" columns are removed in the second phase.

alpha = 0.001, degree = 3



train data = 10806 rows, test data = 6483 rows

model:algorithm: gd- train\_cost: 0.7988139709294926 validation\_cost: 0.7918911956363699 cf: MSE iterations: 100 alpha: 0.001 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: 0.37766068717820905 validation\_cost: 0.37830270332160826 cf: MSE iterations: 1000 alpha: 0.001 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: 0.29217469371274457 validation\_cost: 0.2892469839148515 cf: MSE iterations: 100 alpha: 0.001 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: 0.3516472889746067 validation\_cost: 0.36053020040800193 cf: MSE iterations: 1000 alpha: 0.001 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: 0.6970221952832046 validation\_cost: 0.7092553827426626 cf: MSE iterations: 100 alpha: 0.001 lambda: 50 degree: 3

model:algorithm: gd- train\_cost: 0.23396173996006375 validation\_cost: 0.2344914978529544 cf: MSE iterations: 1000 alpha: 0.001 lambda: 50 degree: 3

best model with alpha = 0.001, degree = 3:

model:algorithm: gd- train\_cost: 0.23396173996006375 validation\_cost: 0.2344914978529544 cf: MSE iterations: 1000 alpha: 0.001 lambda: 50 degree: 3

test cost on best model:

0.7863967153026591

alpha = 0.1, degree = 3

model:algorithm: gd- train\_cost: 0.06095081335593591 validation\_cost: 0.062117364375011724 cf: MSE iterations: 100 alpha: 0.1 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: 0.004684560895313374 validation\_cost: 0.00467537572410383 cf: MSE iterations: 1000 alpha: 0.1 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: 0.03560847603012767 validation\_cost: 0.03599898395854619 cf: MSE iterations: 100 alpha: 0.1 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: 0.004027994004611019 validation\_cost: 0.004105465695322559 cf: MSE iterations: 1000 alpha: 0.1 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: 0.036665490985332796 validation\_cost: 0.037637718028954964 cf: MSE iterations: 100 alpha: 0.1 lambda: 50 degree: 3

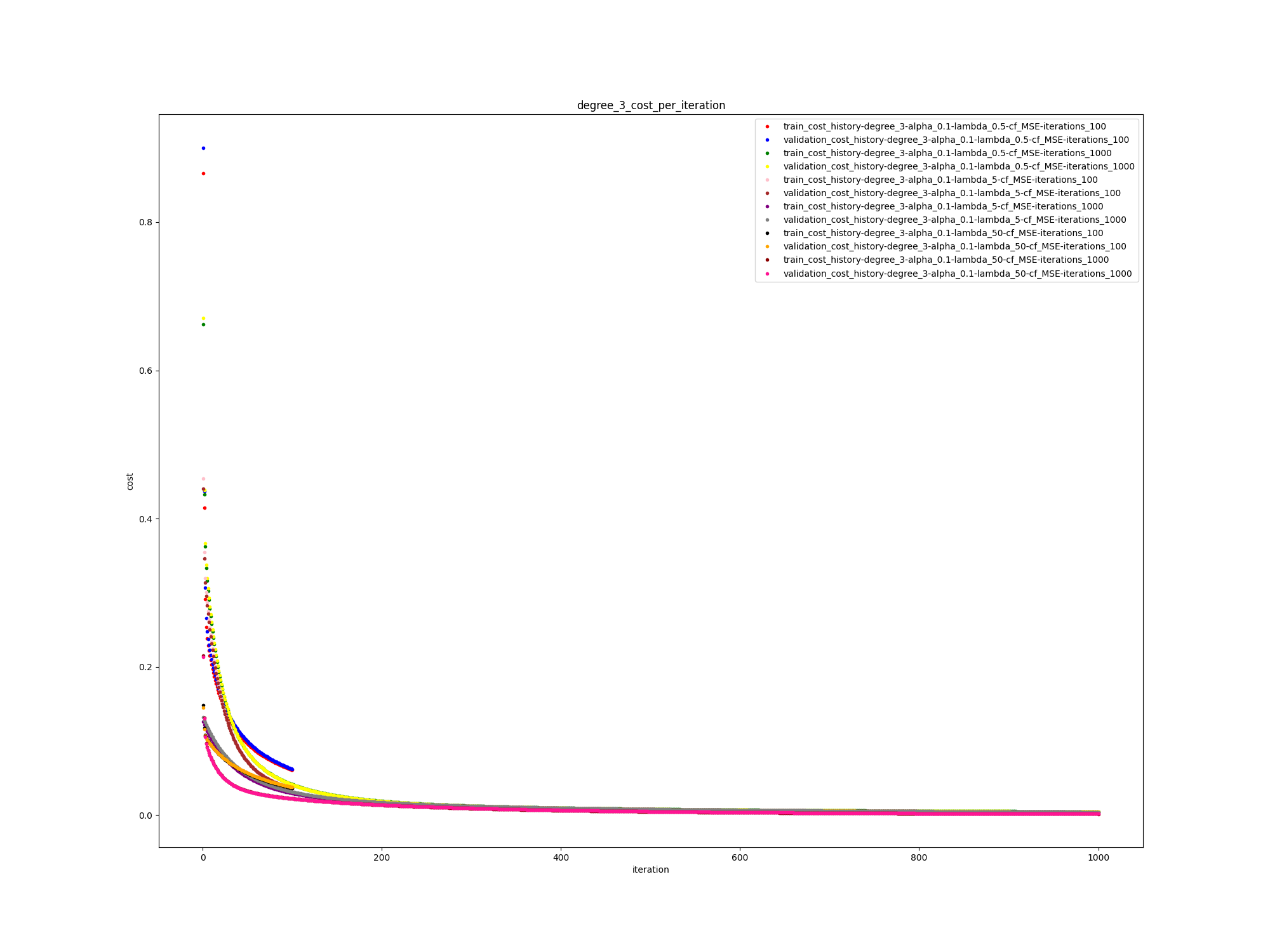
model:algorithm: gd- train\_cost: 0.0016206050611930491 validation\_cost: 0.0016903803945552893 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 3

best model with alpha = 0.1, degree = 3:

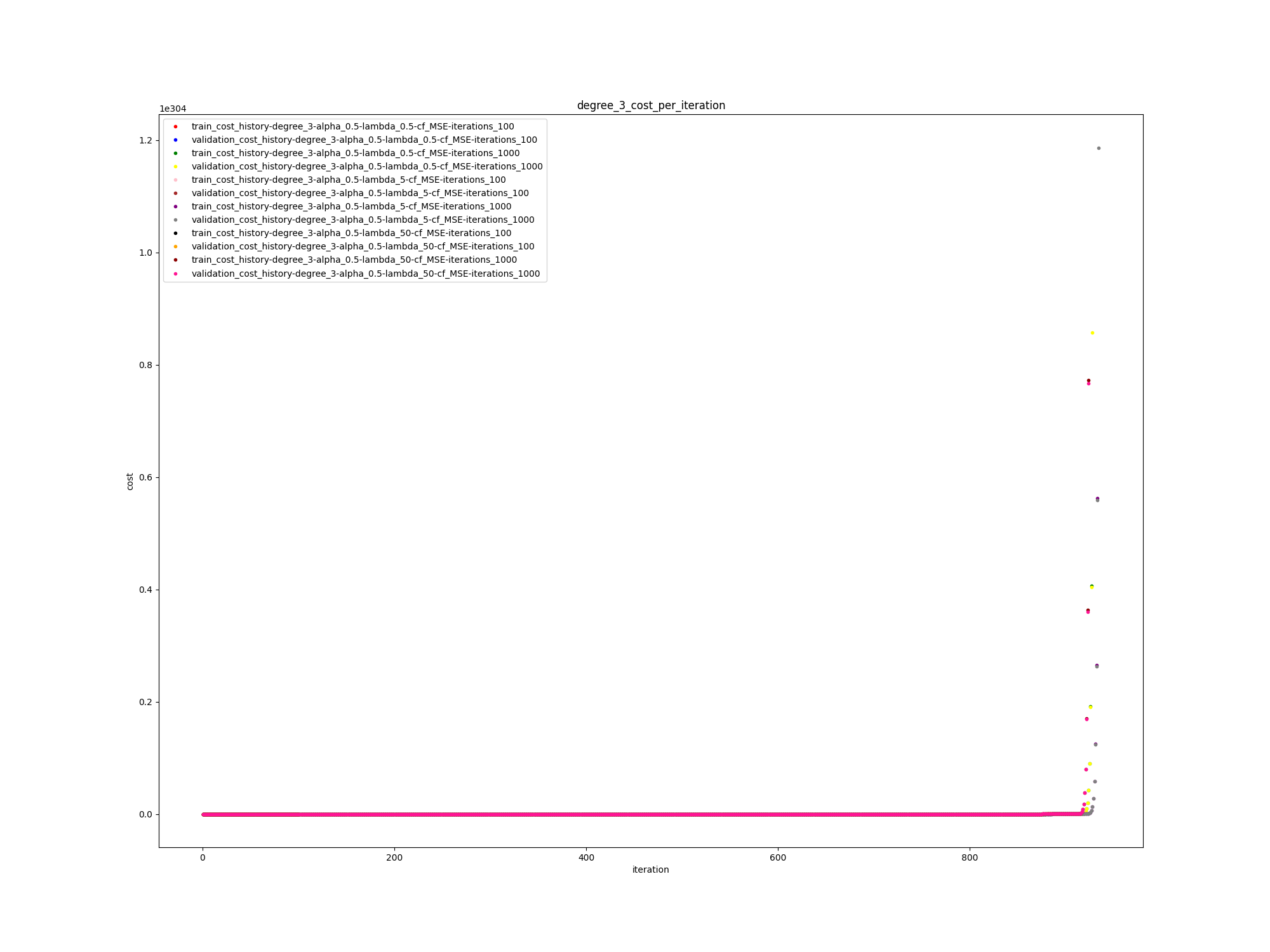
model:algorithm: gd- train\_cost: 0.0016206050611930491 validation\_cost: 0.0016903803945552893 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 3

test cost on best model:

0.0018371923705064145



alpha = 0.5, degree = 3



model:algorithm: gd- train\_cost: 6.827099424427372e+31 validation\_cost: 6.7832743397032205e+31 cf: MSE iterations: 100 alpha: 0.5 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: inf validation\_cost: inf cf: MSE iterations: 1000 alpha: 0.5 lambda: 0.5 degree: 3

model:algorithm: gd- train\_cost: 9.209280188501032e+32 validation\_cost: 9.150162185191889e+32 cf: MSE iterations: 100 alpha: 0.5 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: inf validation\_cost: inf cf: MSE iterations: 1000 alpha: 0.5 lambda: 5 degree: 3

model:algorithm: gd- train\_cost: 1.07020384495143e+33 validation\_cost: 1.0633325930243435e+33 cf: MSE iterations: 100 alpha: 0.5 lambda: 50 degree: 3

model:algorithm: gd- train\_cost: inf validation\_cost: inf cf: MSE iterations: 1000 alpha: 0.5 lambda: 50 degree: 3

best model with alpha = 0.5, degree = 3:

model:algorithm: gd- train\_cost: 6.827099424427372e+31 validation\_cost: 6.7832743397032205e+31 cf: MSE iterations: 100 alpha: 0.5 lambda: 0.5 degree: 3

test cost on best model:

6.809610365025575e+31

alpha = 0.001, degree = 2

model:algorithm: gd- train\_cost: 0.7916973908978685 validation\_cost: 0.7910479972846826 cf: MSE iterations: 100 alpha: 0.001 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 0.11533543420865726 validation\_cost: 0.1159030780419314 cf: MSE iterations: 1000 alpha: 0.001 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 1.2269232037067852 validation\_cost: 1.2114924788245145 cf: MSE iterations: 100 alpha: 0.001 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 0.10835128463638545 validation\_cost: 0.10800229491411835 cf: MSE iterations: 1000 alpha: 0.001 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 0.36180105700777787 validation\_cost: 0.3468368296982039 cf: MSE iterations: 100 alpha: 0.001 lambda: 50 degree: 2

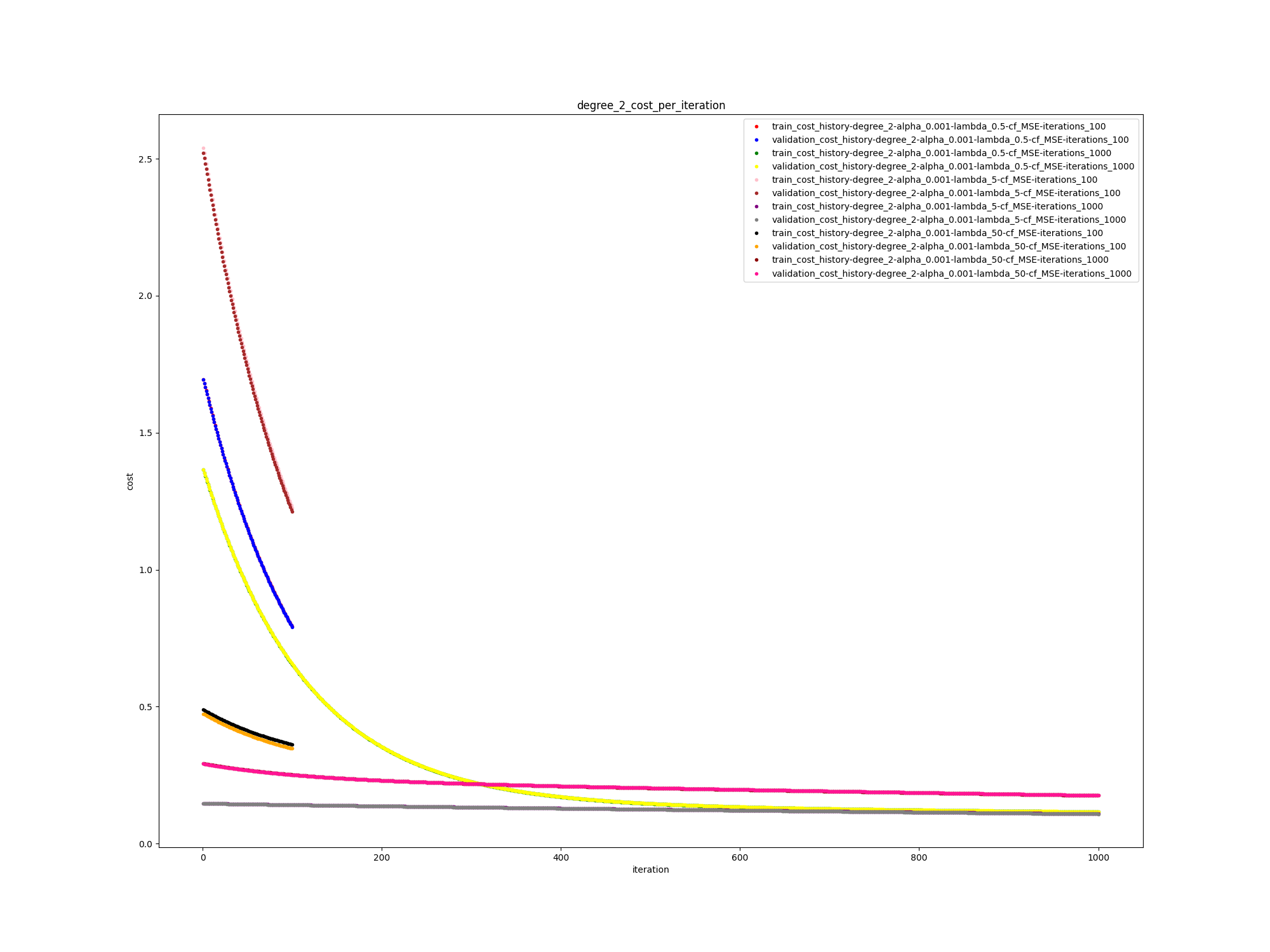
model:algorithm: gd- train\_cost: 0.17555705340715014 validation\_cost: 0.17573936021805497 cf: MSE iterations: 1000 alpha: 0.001 lambda: 50 degree: 2

best model with alpha = 0.001, degree = 2:

model:algorithm: gd- train\_cost: 0.10835128463638545 validation\_cost: 0.10800229491411835 cf: MSE iterations: 1000 alpha: 0.001 lambda: 5 degree: 2

test cost on best model:

0.10757126258459974

alpha = 0.1, degree = 2

model:algorithm: gd- train\_cost: 0.0230507417345265 validation\_cost: 0.023242313006507205 cf: MSE iterations: 100 alpha: 0.1 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 0.0026066893421623563 validation\_cost: 0.0025737149020751905 cf: MSE iterations: 1000 alpha: 0.1 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 0.03841893719979733 validation\_cost: 0.03704448556074236 cf: MSE iterations: 100 alpha: 0.1 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 0.005083463347699982 validation\_cost: 0.005306261259628261 cf: MSE iterations: 1000 alpha: 0.1 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 0.026165784564623304 validation\_cost: 0.025227154121630836 cf: MSE iterations: 100 alpha: 0.1 lambda: 50 degree: 2

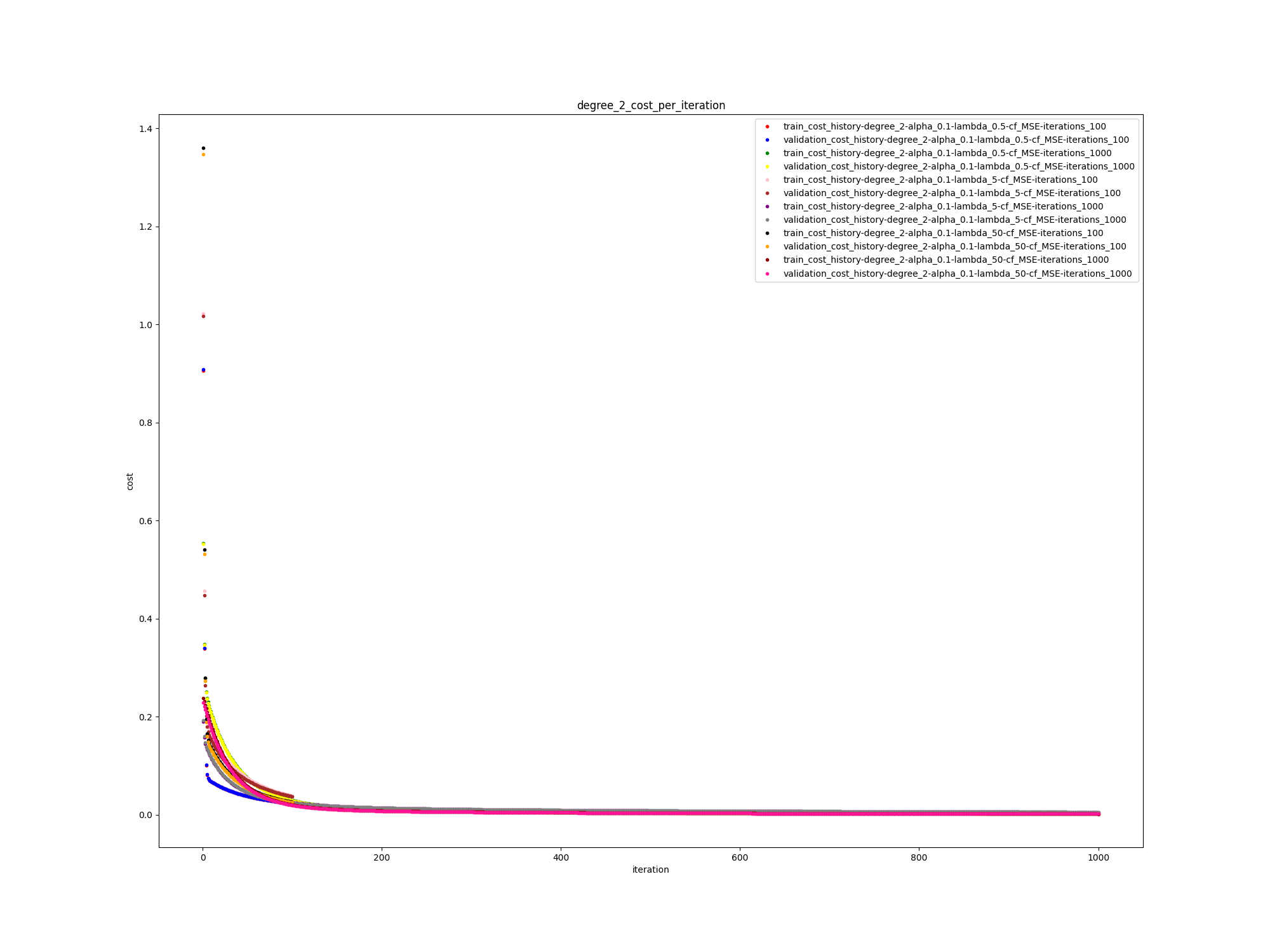
model:algorithm: gd- train\_cost: 0.0015756286098117438 validation\_cost: 0.001579416726282 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 2

best model with alpha = 0.1, degree = 2:

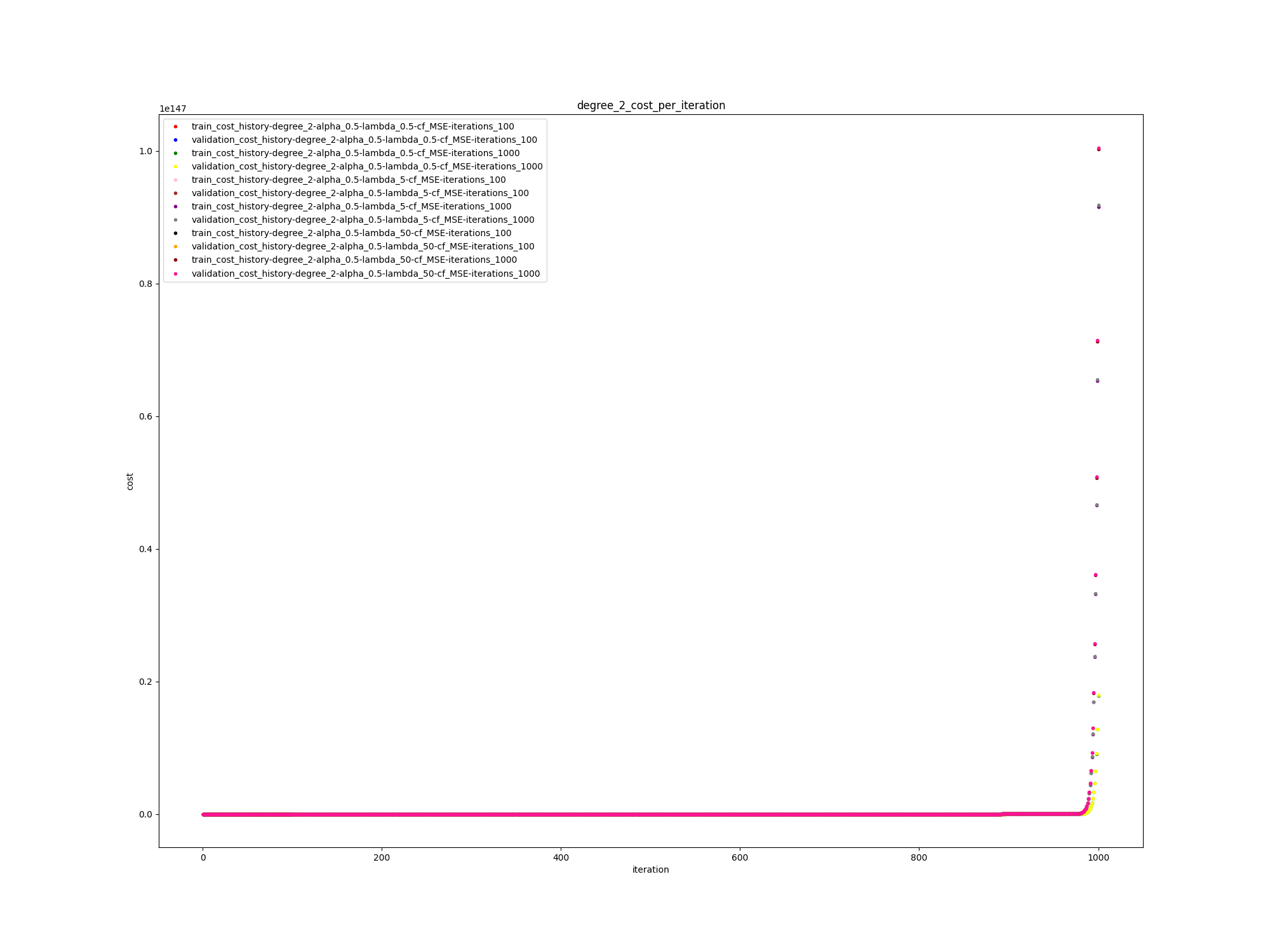
model:algorithm: gd- train\_cost: 0.0015756286098117438 validation\_cost: 0.001579416726282 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 2

test cost on best model:

0.0014896222045468472



alpha = 0.5, degree = 2



model:algorithm: gd- train\_cost: 108175360650557.45 validation\_cost: 108480788008486.36 cf: MSE iterations: 100 alpha: 0.5 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 1.786589303575044e+146 validation\_cost: 1.791633642531719e+146 cf: MSE iterations: 1000 alpha: 0.5 lambda: 0.5 degree: 2

model:algorithm: gd- train\_cost: 449528994880931.5 validation\_cost: 450798249689150.7 cf: MSE iterations: 100 alpha: 0.5 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 9.157645538652459e+146 validation\_cost: 9.183502347885044e+146 cf: MSE iterations: 1000 alpha: 0.5 lambda: 5 degree: 2

model:algorithm: gd- train\_cost: 207977166870282.66 validation\_cost: 208564551251089.56 cf: MSE iterations: 100 alpha: 0.5 lambda: 50 degree: 2

model:algorithm: gd- train\_cost: 1.0022153425693e+147 validation\_cost: 1.0050458726078732e+147 cf: MSE iterations: 1000 alpha: 0.5 lambda: 50 degree: 2

best model with alpha = 0.5, degree = 2:

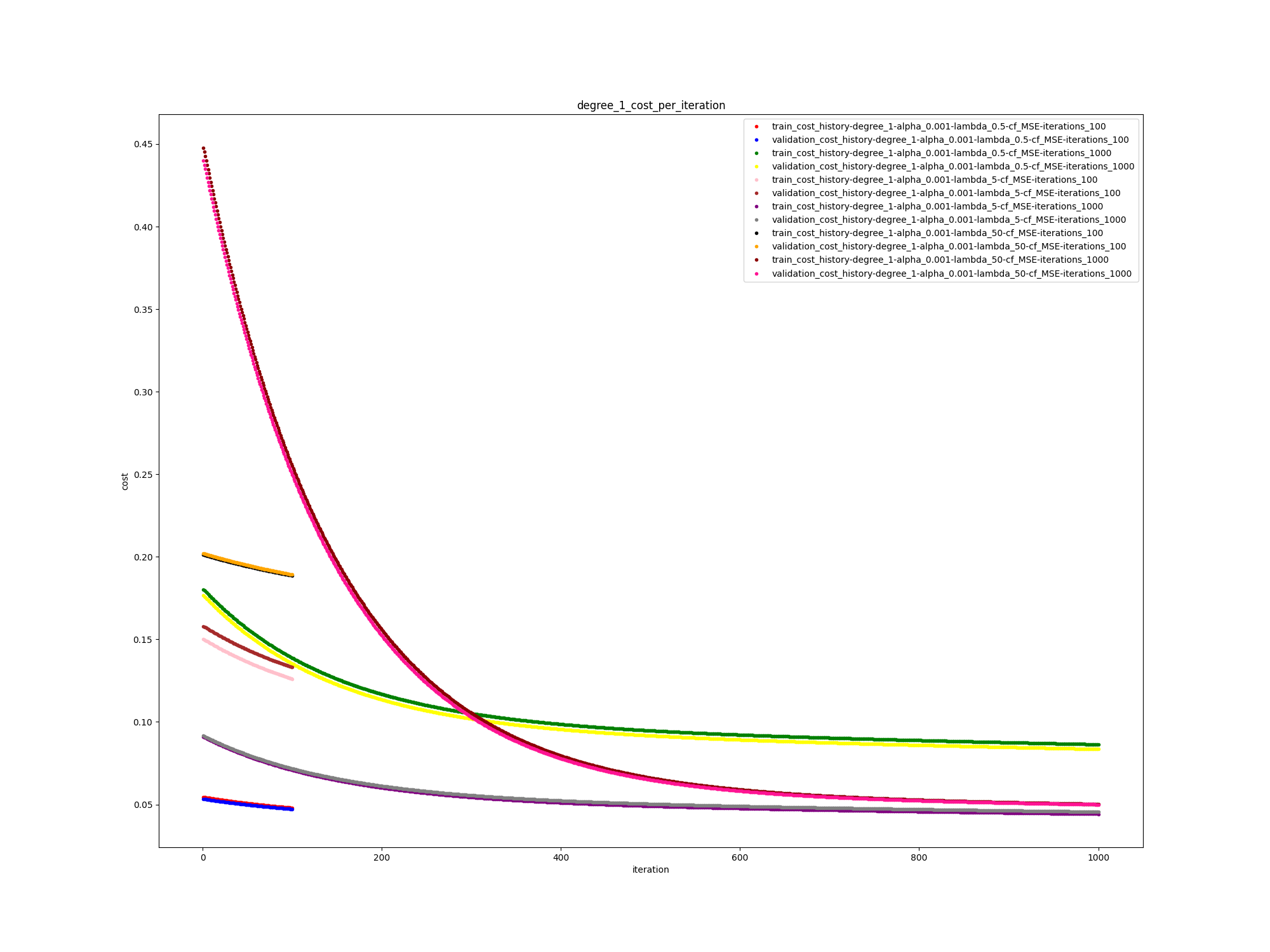
model:algorithm: gd- train\_cost: 108175360650557.45 validation\_cost: 108480788008486.36 cf: MSE iterations: 100 alpha: 0.5 lambda: 0.5 degree: 2

test cost on best model:

108637630449210.2

**the rest of the results are summarized:**

alpha = 0.001, degree = 1



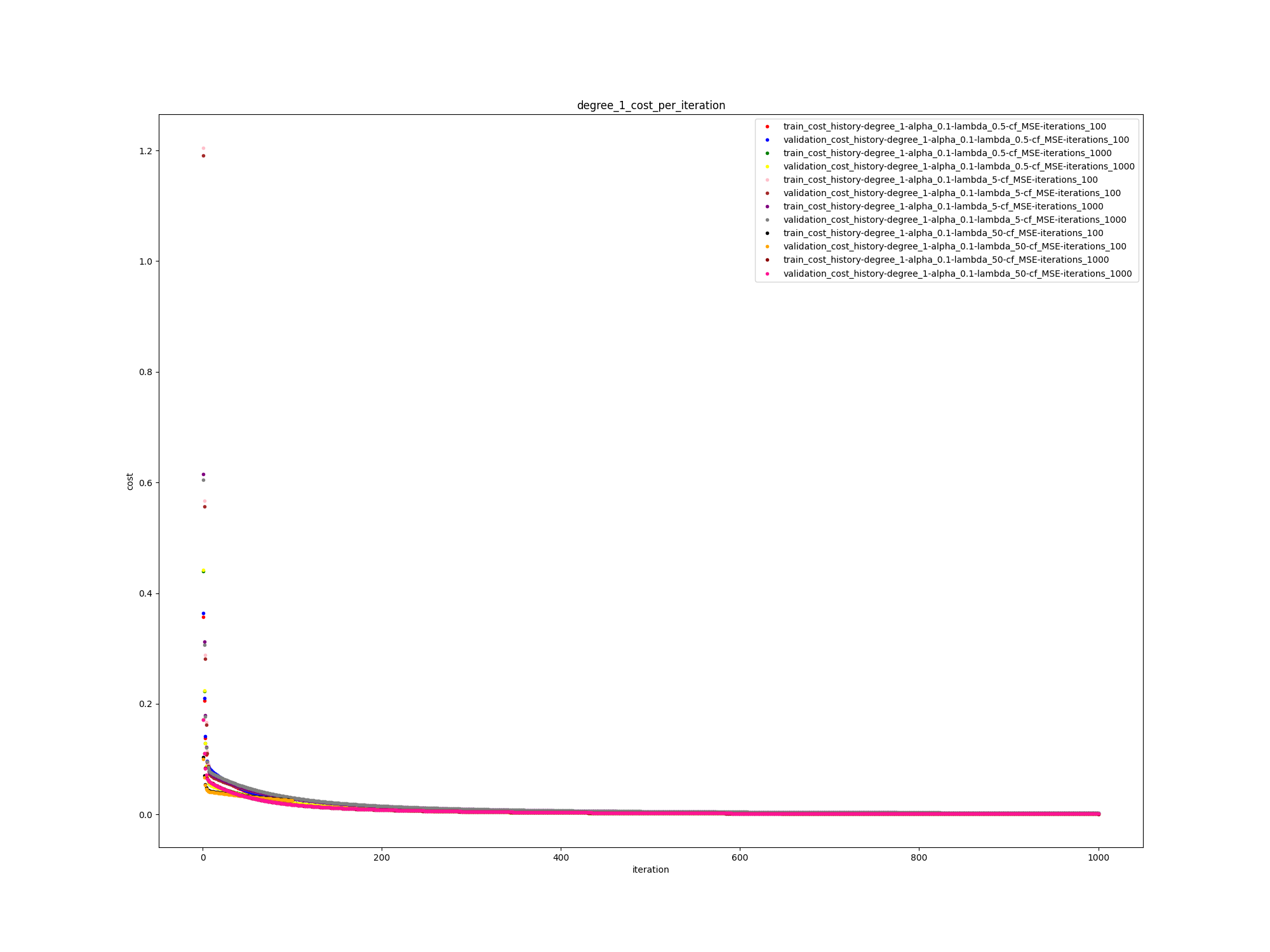
best model:

model:algorithm: gd- train\_cost: 0.04424991968911361 validation\_cost: 0.0455508635709605 cf: MSE iterations: 1000 alpha: 0.001 lambda: 5 degree: 1

test cost on best model:

0.045206349979736805

alpha = 0.1, degree = 1



best model:

model:algorithm: gd- train\_cost: 0.000959616604664554 validation\_cost: 0.0009649598374774853 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 1

test cost on best model:

0.0010472252912346388

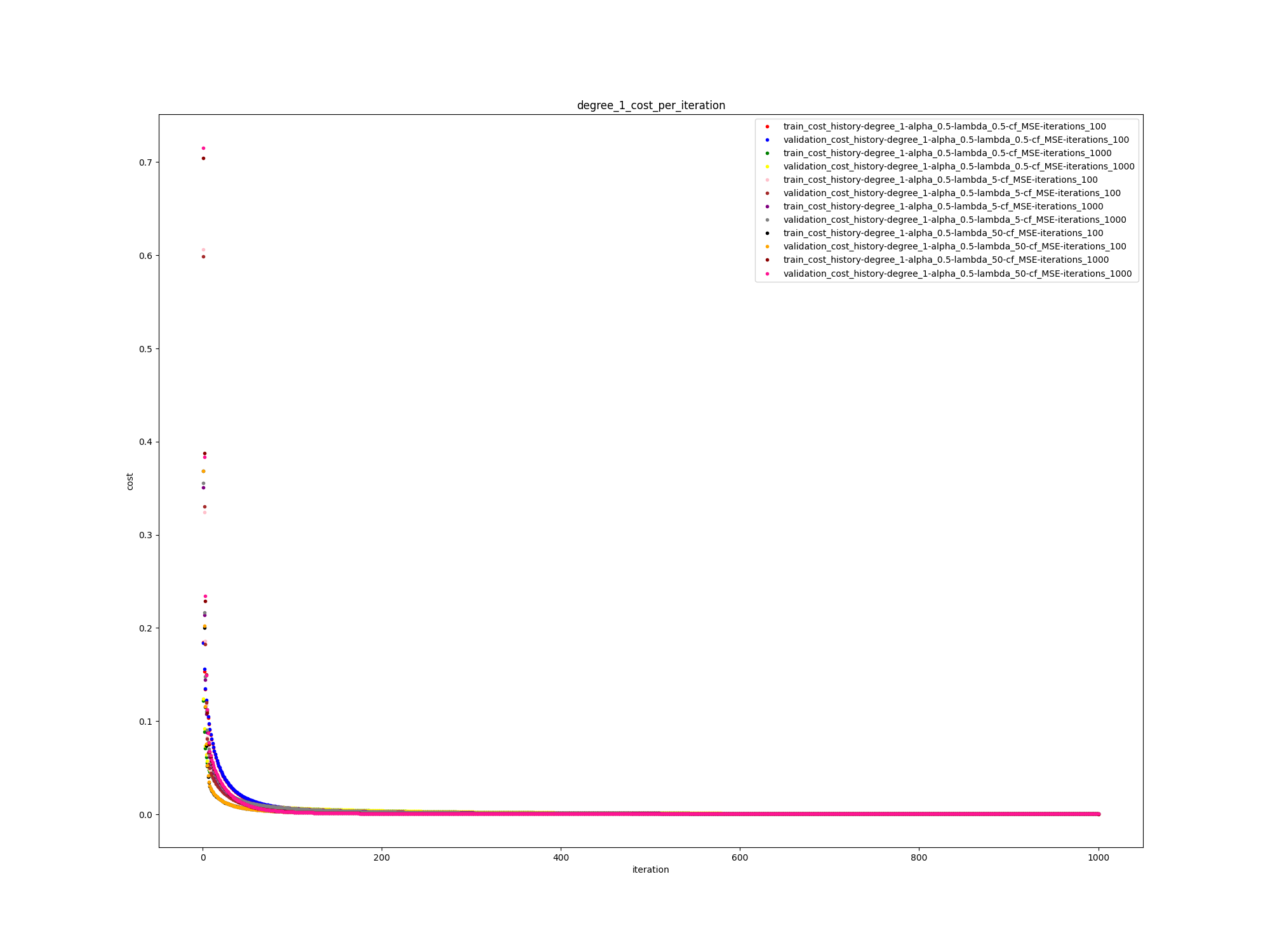
alpha = 0.5, degree = 1

best model:

model:algorithm: gd- train\_cost: 0.0003738714672594293 validation\_cost: 0.0003760298354493929 cf: MSE iterations: 1000 alpha: 0.5 lambda: 50 degree: 1

test cost on best model:

0.00041495426735845524

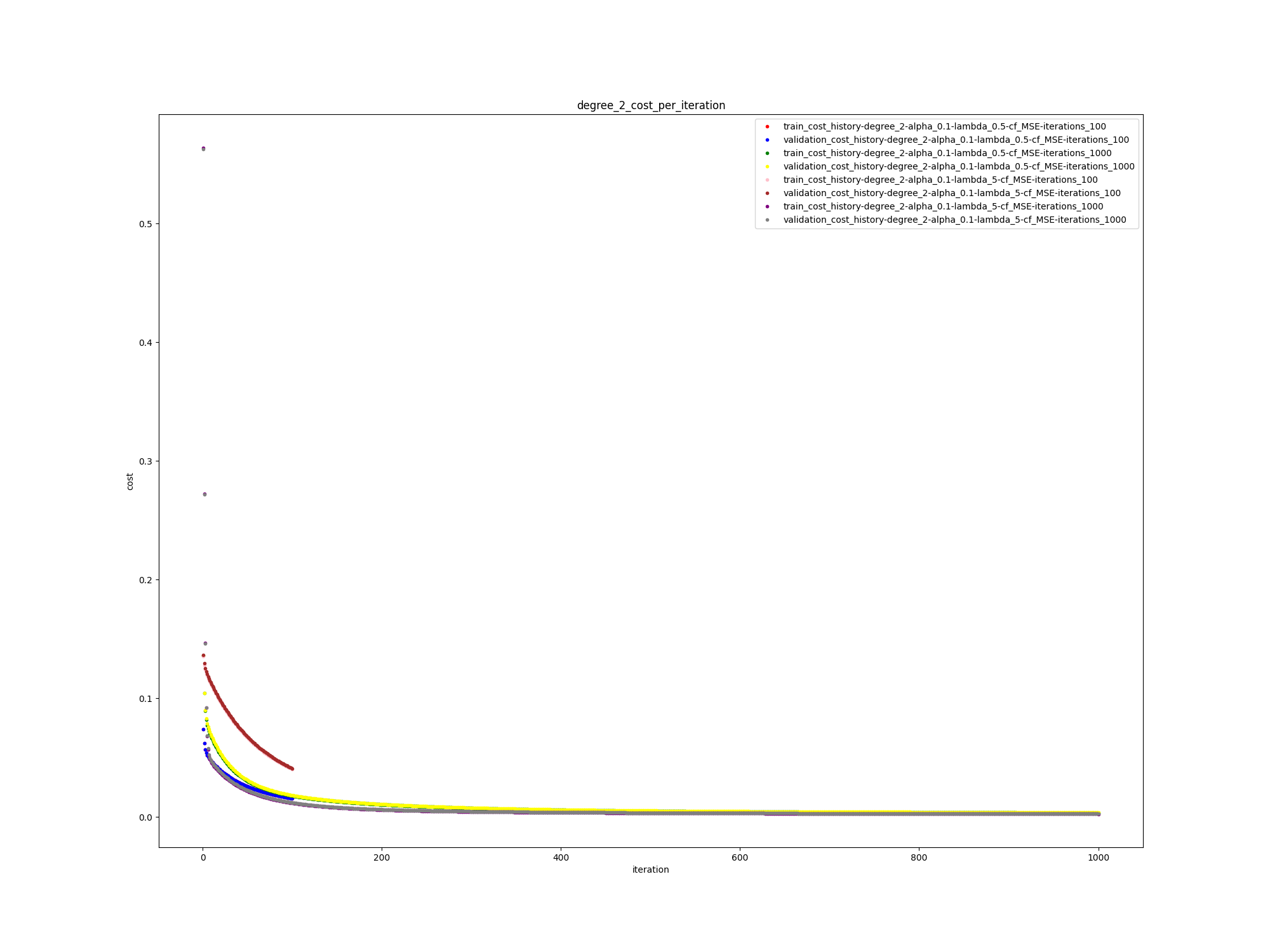


**best model without removing any features:**

**alpha = 0.1, degree = 3:**

**model:algorithm: gd- train\_cost: 0.0016206050611930491 validation\_cost: 0.0016903803945552893 cf: MSE iterations: 1000 alpha: 0.1 lambda: 50 degree: 3**

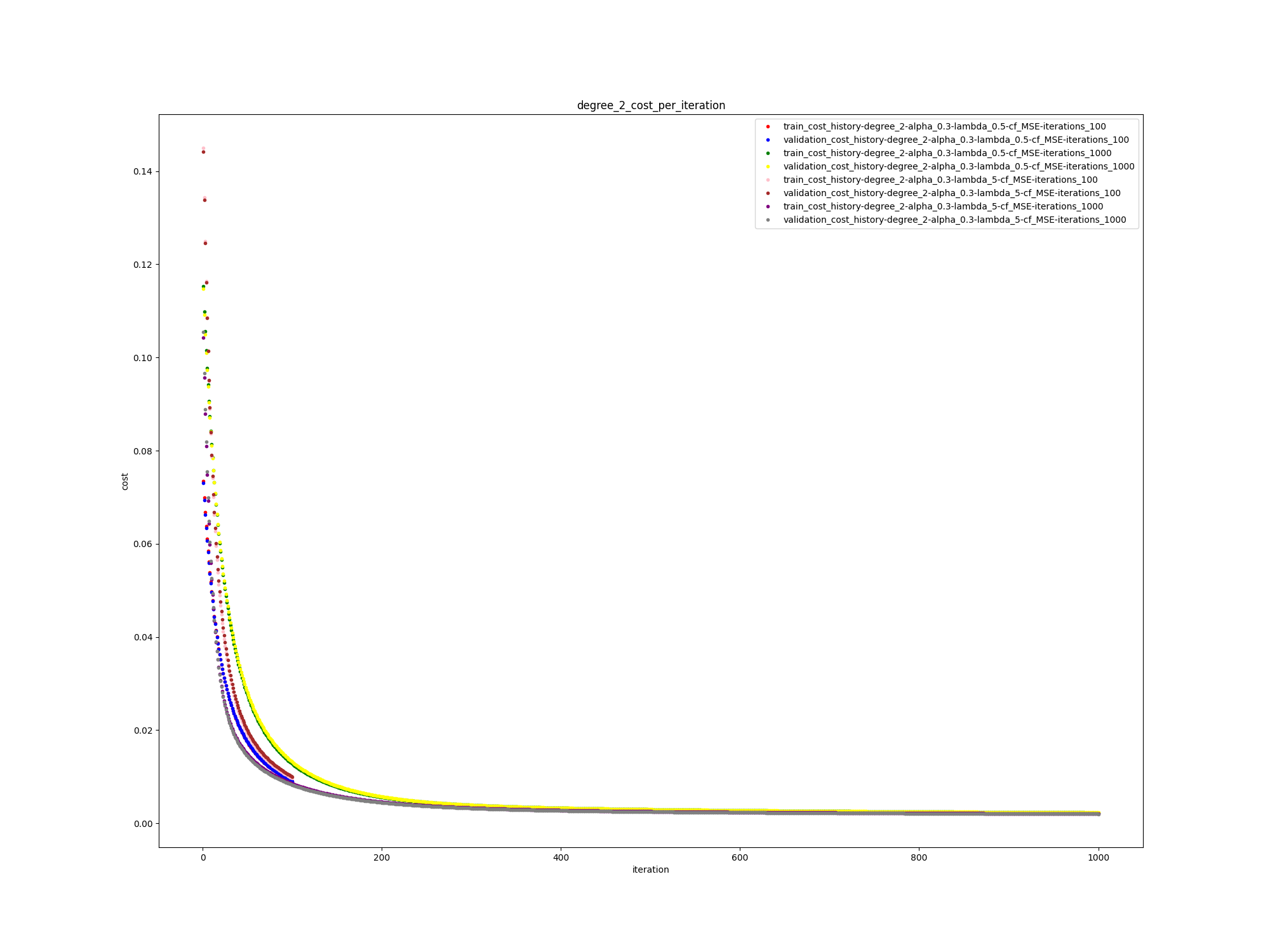
**removing features:**

best model:

model:algorithm: gd- train\_cost: 0.002664300188894608 validation\_cost: 0.0027390606569459774 cf: MSE iterations: 1000 alpha: 0.1 lambda: 5 degree: 2

test cost on best model:

0.002837981302535363

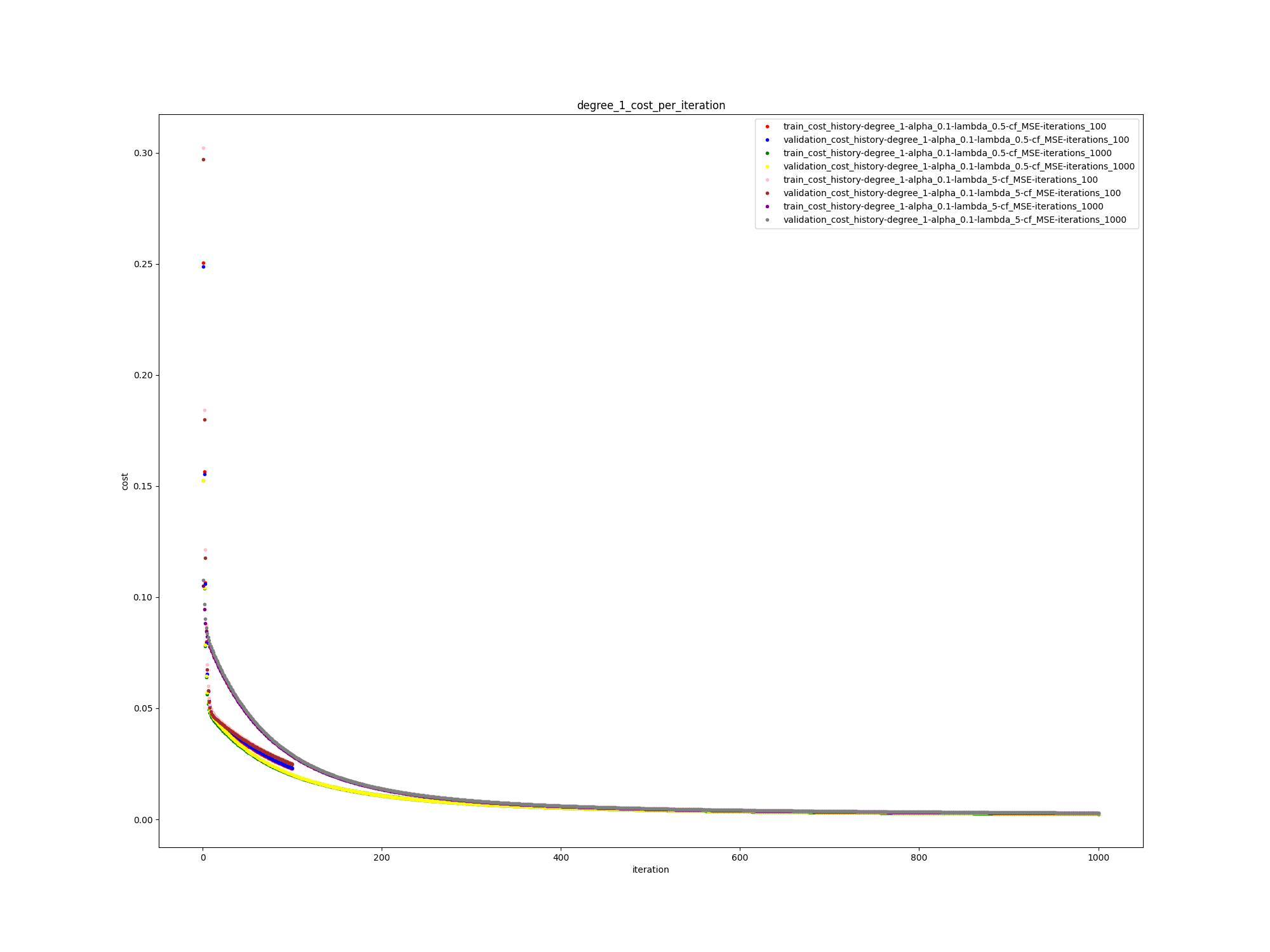


best model:

model:algorithm: gd- train\_cost: 0.0020571761715198838 validation\_cost: 0.0020057087610222952 cf: MSE iterations: 1000 alpha: 0.3 lambda: 5 degree: 2

test cost on best model:

0.002012497380810018

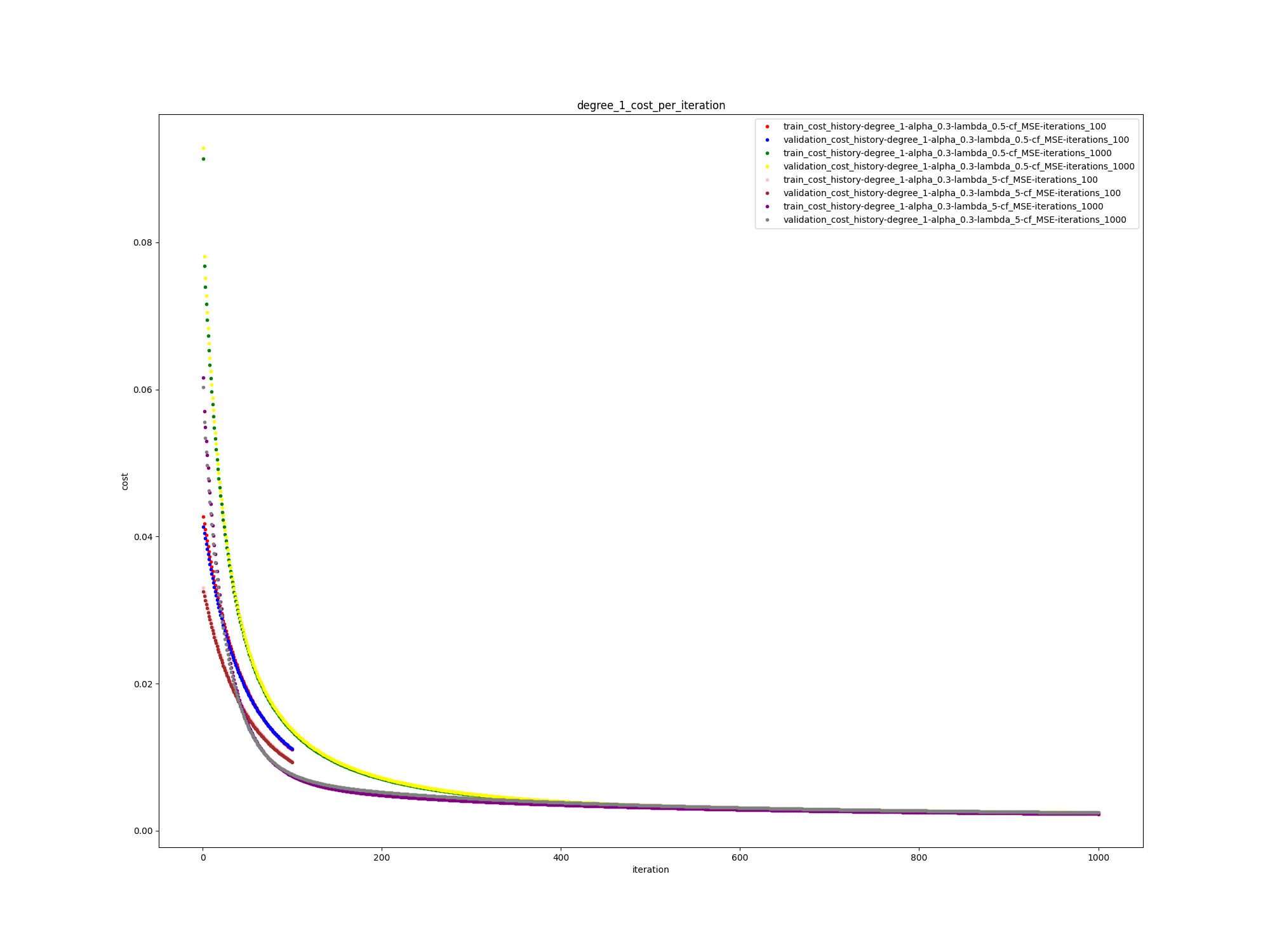


best model:

model:algorithm: gd- train\_cost: 0.002451082694009735 validation\_cost: 0.002510112234240649 cf: MSE iterations: 1000 alpha: 0.1 lambda: 0.5 degree: 1

test cost on best model:

0.0024204517827956474



best model:

model:algorithm: gd- train\_cost: 0.0023684684179785223 validation\_cost: 0.0024095358784667694 cf: MSE iterations: 1000 alpha: 0.3 lambda: 0.5 degree: 1

test cost on best model:

0.002401802215155933

**best model by removing features:**

**model:algorithm: gd- train\_cost: 0.0020571761715198838 validation\_cost: 0.0020057087610222952 cf: MSE iterations: 1000 alpha: 0.3 lambda: 5 degree: 2**

**test cost on best model:**

**0.002012497380810018**

**c- solve previous section with normal equation**