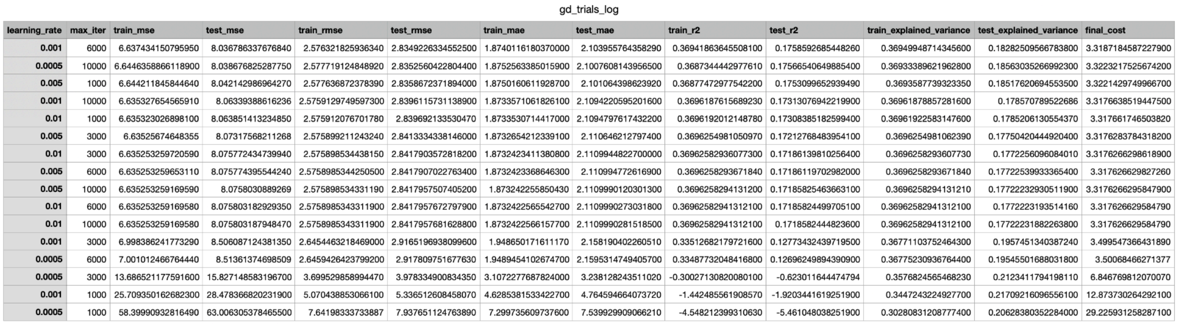
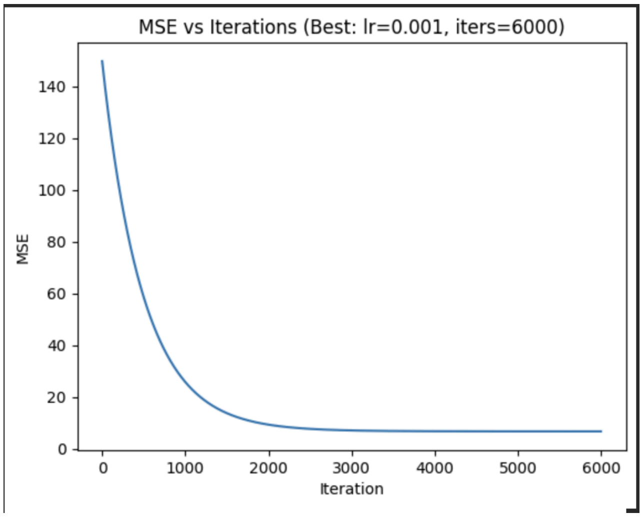
Part 1

Log File with different hyperparameters tuned (learning rate, max iterations):

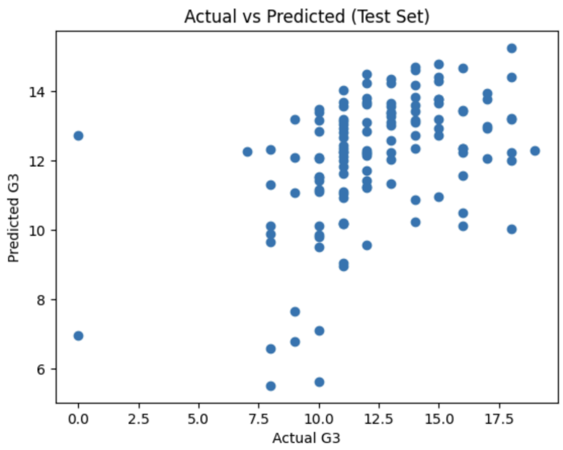


gd_trials_log

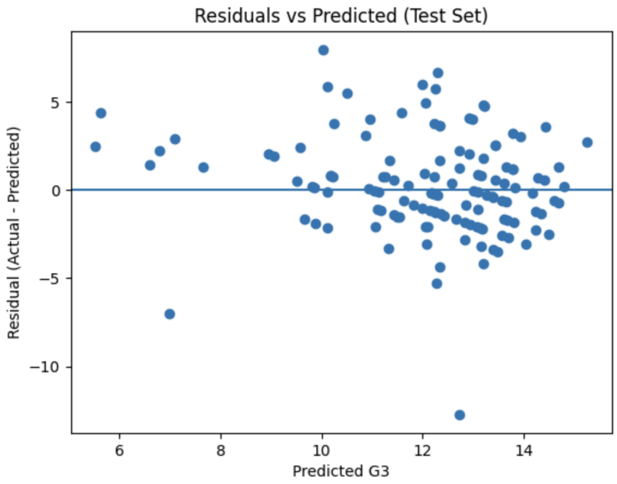
learning_rate	max_iter	train_mse	test_mse	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2	train_explained_variance	test_explained_variance	final_cost
0.001	6000	6.637434150795950	8.036786337676840	2.576321825936340	2.8349226334552500	1.8740116180370000	2.103955764358290	0.36941863645508100	0.1758592685448260	0.3694994871434560	0.1828250956678380	3.3187184587227900
0.0005	10000	6.6446358866118900	8.038676825287750	2.577719124848920	2.8352560422804400	1.8752563385015900	2.1007608143956500	0.3687344442977610	0.1756654064988540	0.36933389621962800	0.1856303526699230	3.3223217525674200
0.005	1000	6.644211845844640	8.042142986964270	2.577636872378390	2.8358672371894000	1.8750160611928700	2.101064398623920	0.36877472977542200	0.1753099652939490	0.3693587739323350	0.1851762069455350	3.3221429749966700
0.001	10000	6.635327654565910	8.06339388616236	2.5759129749597300	2.8396115731138900	1.8733571061826100	2.1094220595201600	0.3696187615689230	0.1731307694221990	0.36961878857281600	0.178570789522686	3.3176638519447500
0.01	1000	6.635323026898100	8.063851413234850	2.575912076701780	2.839692133530470	1.8733530714417000	2.1094797617432200	0.3696192012148780	0.1730838518259940	0.36961922583147600	0.1785206130554370	3.317661746503820
0.005	3000	6.63525674648355	8.07317568211268	2.575899211243240	2.8413334338146000	1.873265421233910	2.110646212797400	0.3696254981050970	0.1721276848395410	0.3696254981062390	0.1775042044492040	3.3176283784318200
0.01	3000	6.635253259720590	8.075772434739940	2.575898534438150	2.8417903572818200	1.8732423411380800	2.1109944822700000	0.36962582936077300	0.1718613981025640	0.3696258293607730	0.1772256096084010	3.3176266298618900
0.005	6000	6.635253259653110	8.075774395544240	2.5758985344250500	2.8417907022763400	1.8732423368646300	2.110994772616900	0.3696258293671840	0.1718611970298200	0.3696258293671840	0.1772253993336540	3.31762662982726
0.005	10000	6.635253259169590	8.0758030889269	2.575898534331190	2.8417957507405200	1.87324225585043	2.1109990120301300	0.3696258294131200	0.1718582546366310	0.3696258294131210	0.1772223293051190	3.3176266295847900
0.01	6000	6.635253259169580	8.075803182929350	2.5758985343311900	2.8417957672797900	1.8732422565542700	2.1109990273031800	0.36962582941312100	0.1718582449970510	0.36962582941312100	0.177222319351416	3.31762662958479
0.01	10000	6.635253259169580	8.075803187948470	2.5758985343311900	2.8417957681628800	1.8732422566157700	2.1109990281518500	0.36962582941312100	0.171858244482360	0.36962582941312100	0.1772223188226380	3.31762662958479
0.001	3000	6.998386241773290	8.506087124381350	2.6454463218469000	2.9165196938099600	1.94865017161117	2.158190402260510	0.33512682179721600	0.1277343243971950	0.36771103752464300	0.195745134038724	3.49954736643189
0.0005	6000	7.00101246676444	8.51361374698509	2.6459426423799200	2.917809751677630	1.9489454102674700	2.1595314749405700	0.33487732048416800	0.1269624989439090	0.36775230936764400	0.1954550168803180	3.50068466271377
0.0005	3000	13.686521177591600	15.827148583196700	3.69952985899447	3.97833490083435	3.1072277687824000	3.238128243511020	-0.30027130820080100	-0.623011644474794	0.357682456546823	0.212341179419811	6.84676981207007
0.001	1000	25.709350162682300	28.478366820231900	5.0704388530661	5.33651260845807	4.6285381533422700	4.764594664073720	-1.442485561908570	-1.9203441619251900	0.3447243224927700	0.2170921609655610	12.873730264292100
0.0005	1000	58.39990932816490	63.006305378465500	7.64198333733887	7.93765112476389	7.2997356097376	7.53992990906621	-4.548212399310630	-5.461048038251900	0.3028083120877740	0.2062838035228400	29.225931258287100



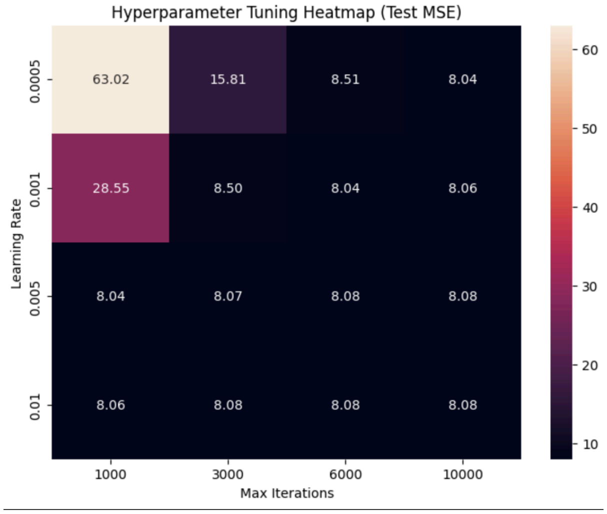
MSE vs Iterations (Best: lr=0.001, iters=6000)



Actual vs Predicted (Test Set)



Residuals vs Predicted (Test Set)



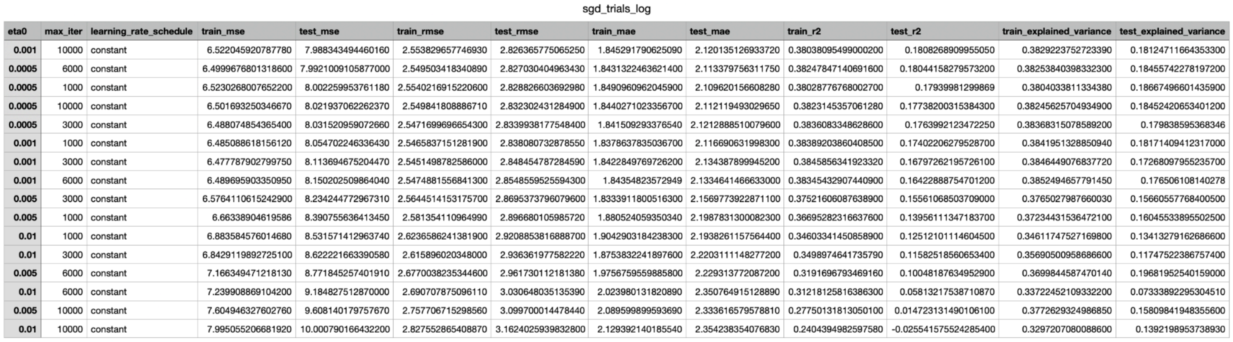
Hyperparameter Tuning Heatmap (Test MSE)

Are you satisfied that you have found the best solution?

I am reasonably satisfied with the solution but not completely certain. This is because we tested using a combination of 4 learning rates and 4 iterations and the best MSE we got was 8.036, however there could've been a better parameter that we did not explore while hyperparameter tuning. Conducting a better hyperparameter search with more granular data or using various optimization techniques like grid search or randomized search could possibly help yield better results. Also we did not export with feature selection. There could have been features in the dataset that might not have contributed meaningfully to the prediction and removing these could have potentially improved the accuracy. Also while applying the linear regression function I realized that the gradient descent successfully converged which means the cost stopped decreasing but however this could just be the local minimum and not necessarily the global minimum.

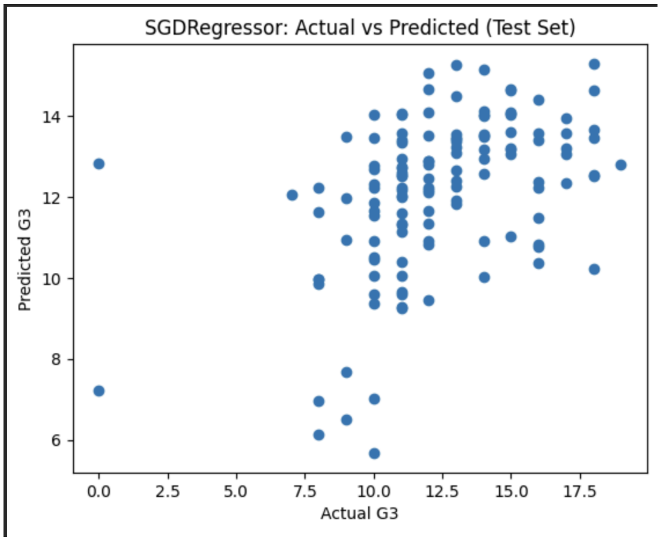
Part 2

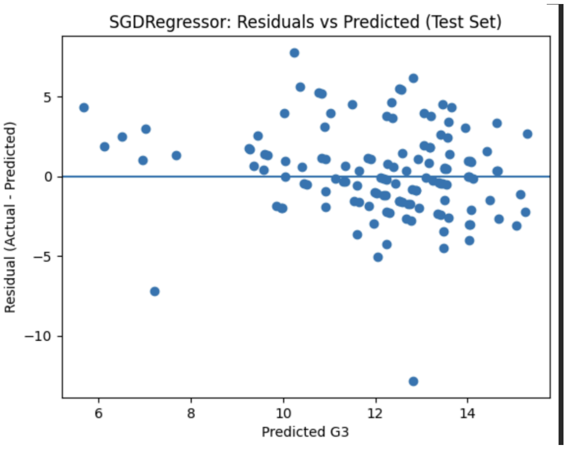
Log File with different hyperparameters tuned (eta0, max iterations):

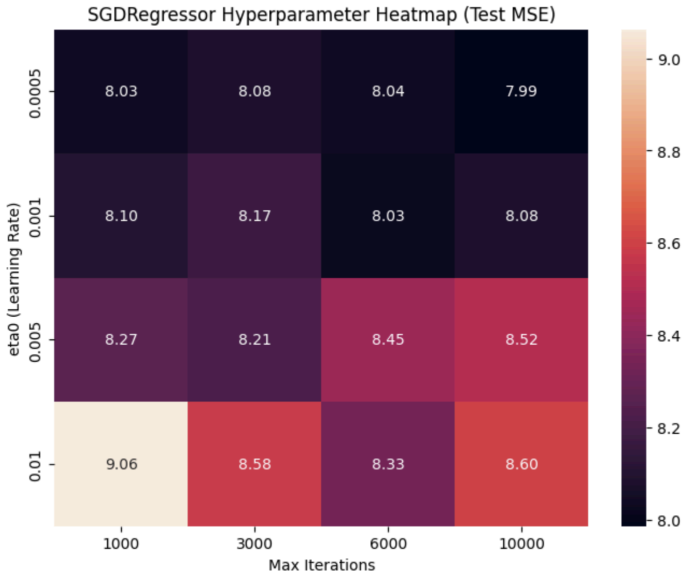


sgd_trials_log

eta0	max_iter	learning_rate_schedule	train_mse	test_mse	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2	train_explained_variance	test_explained_variance
0.001	10000	constant	6.522045920787780	7.988343494460160	2.553829657746930	2.826365775065250	1.845291790625090	2.120135126933720	0.38038095499000200	0.1808268909955050	0.3829223752723390	0.18124711664353300
0.0005	6000	constant	6.4999676801318600	7.9921009105877000	2.549503418340890	2.827030404963430	1.8431322463621400	2.113379756311750	0.38247847140691600	0.18044158279573200	0.38253840398332300	0.18455742278197200
0.0005	1000	constant	6.5230268007652200	8.002259953761180	2.5540216915220600	2.828826603692980	1.8490960962045900	2.109620156608280	0.38028776768002700	0.17939981299869	0.3804033811334380	0.18667496601435900
0.0005	10000	constant	6.501693250346670	8.021937062262370	2.549841808886710	2.832302431284900	1.8440271023356700	2.112119493029650	0.3823145357061280	0.17738200315384300	0.38245625704934900	0.18452420653401200
0.0005	3000	constant	6.488074854365400	8.031520959072660	2.5471699696654300	2.8339938177548400	1.841509293376540	2.1212888510079600	0.3836083348628600	0.1763992123472250	0.38368315078589200	0.179838595368346
0.001	1000	constant	6.485088618156120	8.054702246336430	2.5465837151281900	2.838080732878550	1.8378637835036700	2.116690631998300	0.38389203860408500	0.17402206279528700	0.3841951328850940	0.18171409412317000
0.001	3000	constant	6.477787902799750	8.113694675204470	2.5451498782586000	2.848454787284590	1.8422849769726200	2.134387899945200	0.3845856341923320	0.16797262195726100	0.3846449076837720	0.17268097955235700
0.001	6000	constant	6.489695903350950	8.150202509864040	2.5474881556841300	2.8548559525594300	1.84354823572949	2.1334641466633000	0.38345432907440900	0.16422888754701200	0.3852494657791450	0.176506108140278
0.005	3000	constant	6.5764110615242900	8.234244772967310	2.5644514153175700	2.8695373796079600	1.8333911800516300	2.1569773922871100	0.37521606087638900	0.15561068503709000	0.3765027987660030	0.15660557768400500
0.005	1000	constant	6.66338904619586	8.390755636413450	2.581354110964990	2.896680105985720	1.880524059350340	2.1987831300082300	0.36695282316637600	0.13956111347183700	0.37234431536472100	0.16045533895502500
0.01	1000	constant	6.883584576014680	8.531571412963740	2.6236586241381900	2.9208853816888700	1.9042903184238300	2.1938261157564400	0.34603341450858900	0.12512101114604500	0.34611747527169800	0.13413279162686600
0.01	3000	constant	6.8429119892725100	8.622221663390580	2.615896020348000	2.936361977582220	1.8753832241897600	2.2203111148277200	0.3498974641735790	0.11582518560653400	0.35690500958686600	0.11747522386757400
0.005	6000	constant	7.166349471218130	8.771845257401910	2.6770038235344600	2.961730112181380	1.9756759559885800	2.229313772087200	0.3191696793469160	0.10048187634952900	0.3699844587470140	0.19681952540159000
0.01	6000	constant	7.239908869104200	9.184827512870000	2.690707875096110	3.030648035135390	2.023980131820890	2.350764915128890	0.31218125816386300	0.05813217538710870	0.33722452109332200	0.07333892295304510
0.005	10000	constant	7.604946327602760	9.608140179757670	2.757706715298560	3.099700014478440	2.089599899593690	2.333616579578810	0.27750131813050100	0.014723131490106100	0.3772629324986850	0.15809841948355600
0.01	10000	constant	7.995055206681920	10.000790166432200	2.827552865408870	3.1624025939832800	2.129392140185540	2.354238354076830	0.2404394982597580	-0.025541575524285400	0.3297207080088600	0.1392198953738930







SGDRegressor Hyperparameter Heatmap (Test MSE)

eta0 (Learning Rate)	1000	3000	6000	10000
0.0005	8.03	8.08	8.04	7.99
0.001	8.10	8.17	8.03	8.08
0.005	8.27	8.21	8.45	8.52
0.01	9.06	8.58	8.33	8.60

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

Although the package produced a strong solution with a test MSE of 7.99, I am not fully satisfied that the package has found the best solution. The hyperparameter tuning that we did for our model was limited to a small array of values and we only focused on two parameters to tune (eta0 and max iterations). A more exhaustive approach to doing hyperparameter tuning like trying iteration counts or adjusting the regularization setting and adding a wider range of values to try for each parameter would make me more confident that the package found the best solution. Some ways to check if the package has found the best solution is by expanding the hyperparameter search as already discussed, checking convergence behavior at the final iteration (if the loss has not flattened, then the model has not converged), and use cross-validation instead of a single train/test split. These checks would reinforce that we have found the global minimum for the loss function for the dataset.