# CS 6375
# ASSIGNMENT __1__

Names of students in your group:

Sennan Liu

Number of free late days used: _____0_____

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

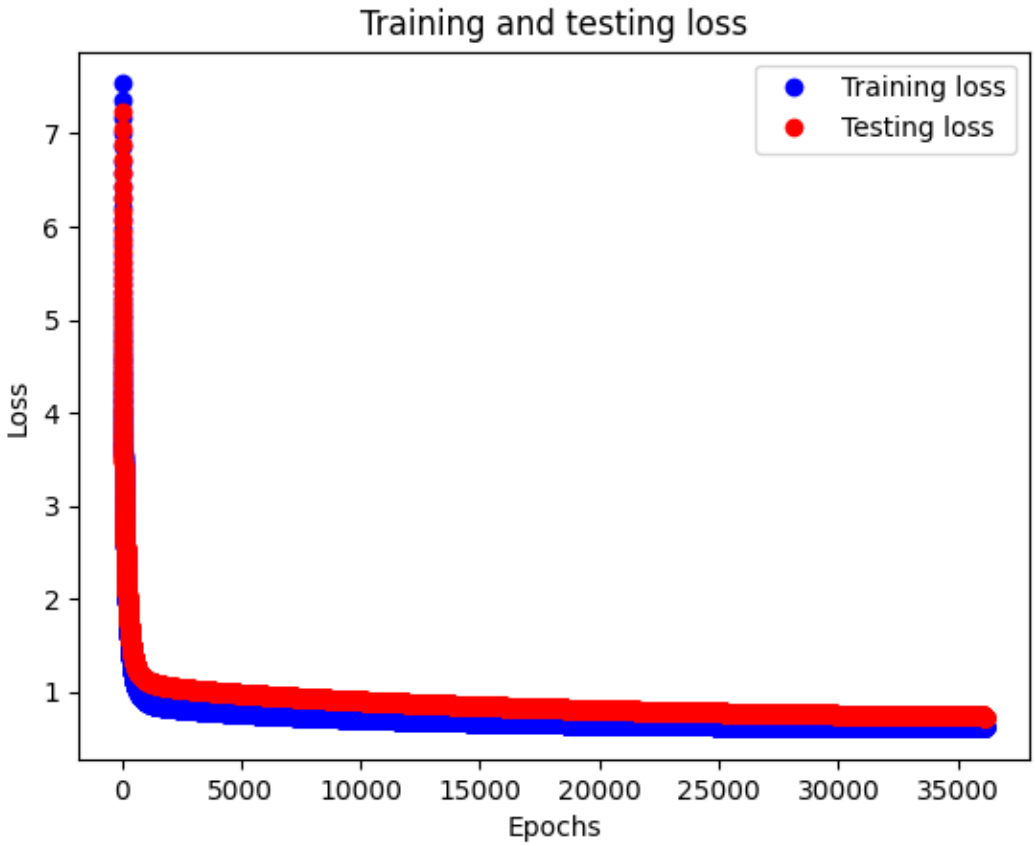
Python packages:
Numpy https://numpy.org/
Pandas https://pandas.pydata.org/
scikit-learn  https://scikit-learn.org/stable/

# Training logs:

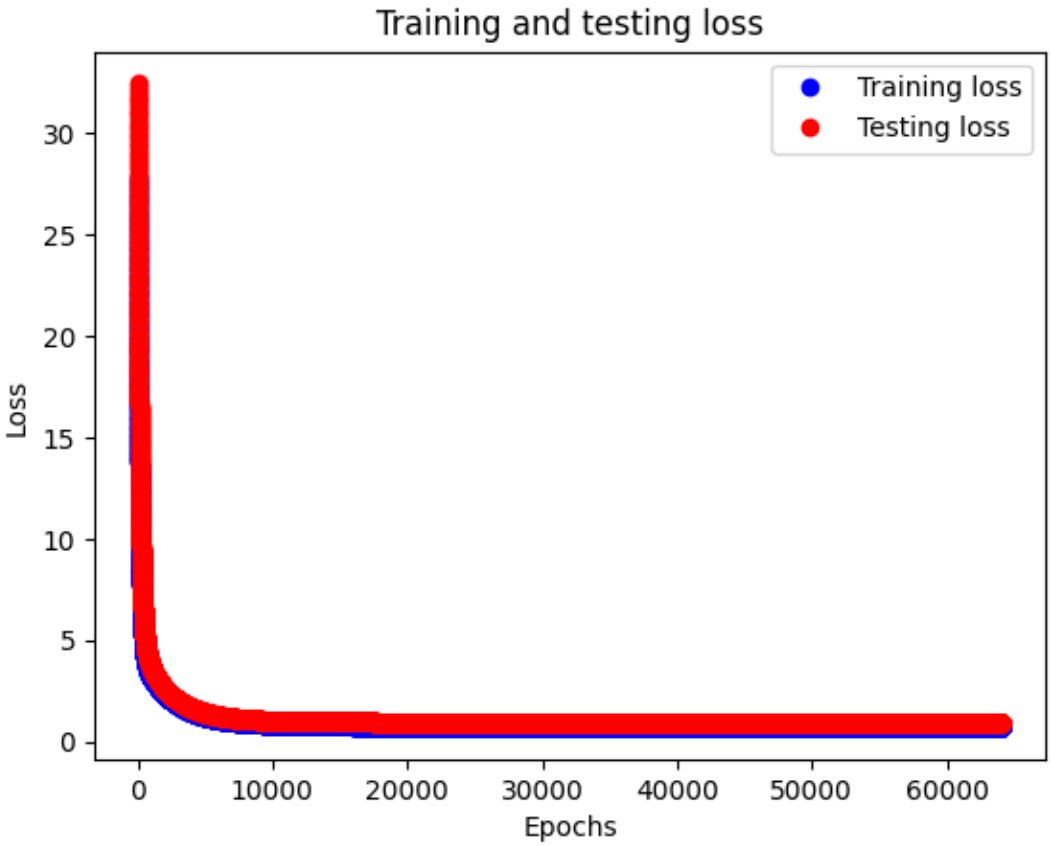
Results using scikit-learn SGDRegressor:
{"max_iter": 1000000, "penalty": "l2", "learning_rate": 9e-05, "verbose": 1, "tol": 1e-06, "alpha": 0.0001, "log_path": "result.log", "mse": 0.6966374322267272}

Results using homemade regressor
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 0.0001, "learning_rate": 9e-05, "loss_graph_path": "homemade_graph.png", "log_path": "result.log", "mse": 1.1876982164305565}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 1e-05, "learning_rate": 0.0001, "loss_graph_path": "1e-04", "log_path": "result.log", "mse": 1.0285168256048542}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 1e-05, "learning_rate": 0.001, "loss_graph_path": "1e-03", "log_path": "result.log", "mse": 0.8978308671845259}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 1e-05, "learning_rate": 0.01, "loss_graph_path": "1e-02", "log_path": "result.log", "mse": 0.7382917205261281}
{"n_iter": 1000000, "tolerance": 0.1, "penalty": "l2", "alpha": 1e-05, "learning_rate": 9e-05, "loss_graph_path": "homemade_lr-1e-01_alpha_1e-04.png", "log_path": "result.log", "mse": 51.09549387985032}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 0.0001, "learning_rate": 0.01, "loss_graph_path": "homemade_lr-1e-02.png", "log_path": "result.log", "mse": 0.7385730441016103}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 0.001, "learning_rate": 0.01, "loss_graph_path": "homemade_alpha_1e-03.png", "log_path": "result.log", "mse": 0.7373508180633499}
{"n_iter": 1000000, "tolerance": 1e-06, "penalty": "l2", "alpha": 0.01, "learning_rate": 0.01, "loss_graph_path": "homemade_alpha_1e-02.png", "log_path": "result.log", "mse": 0.7386764288668501}

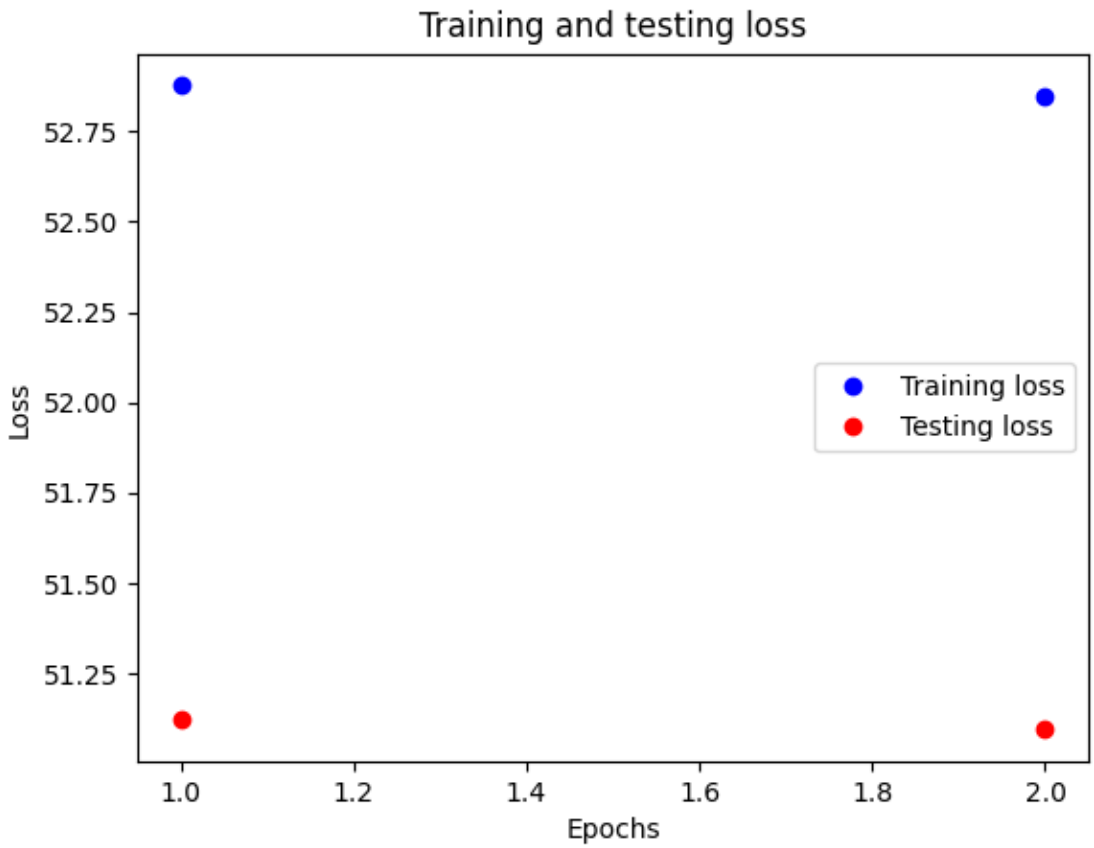
# Some of the plots:



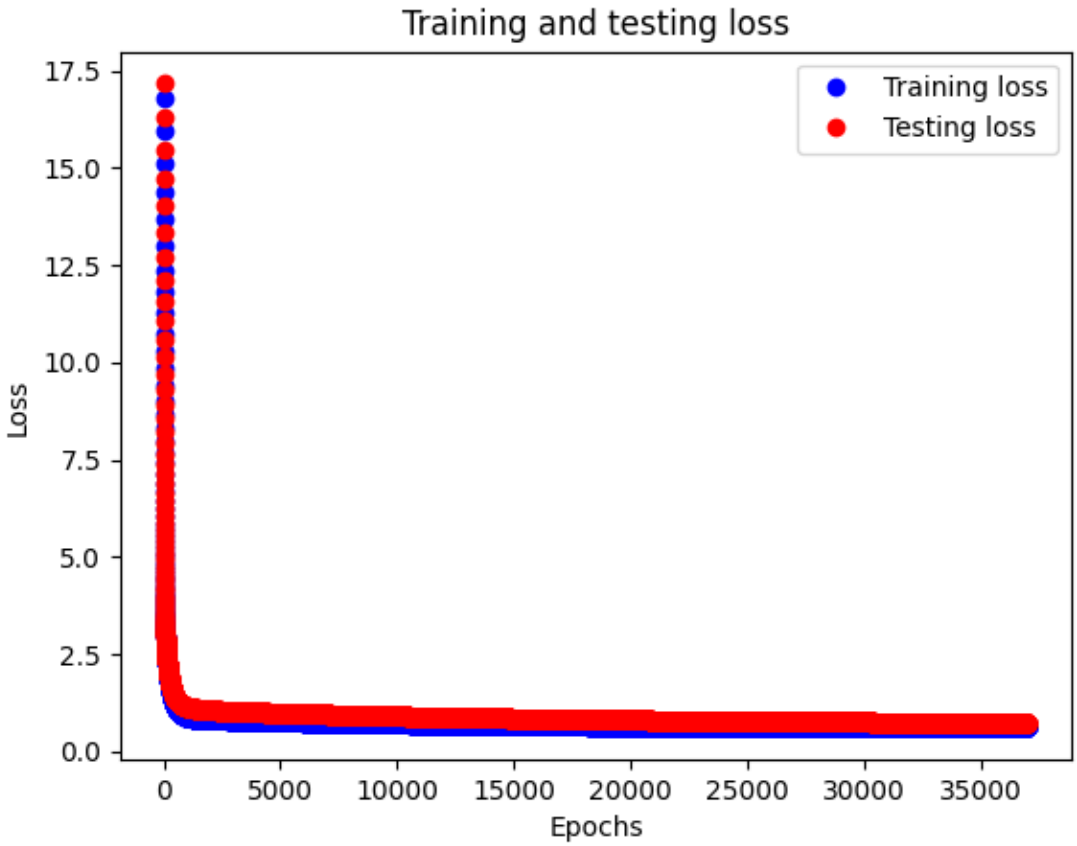

Mean Square Error of homemade gradient decent with a Learning rate 1e-02 and alpha of 1e-04



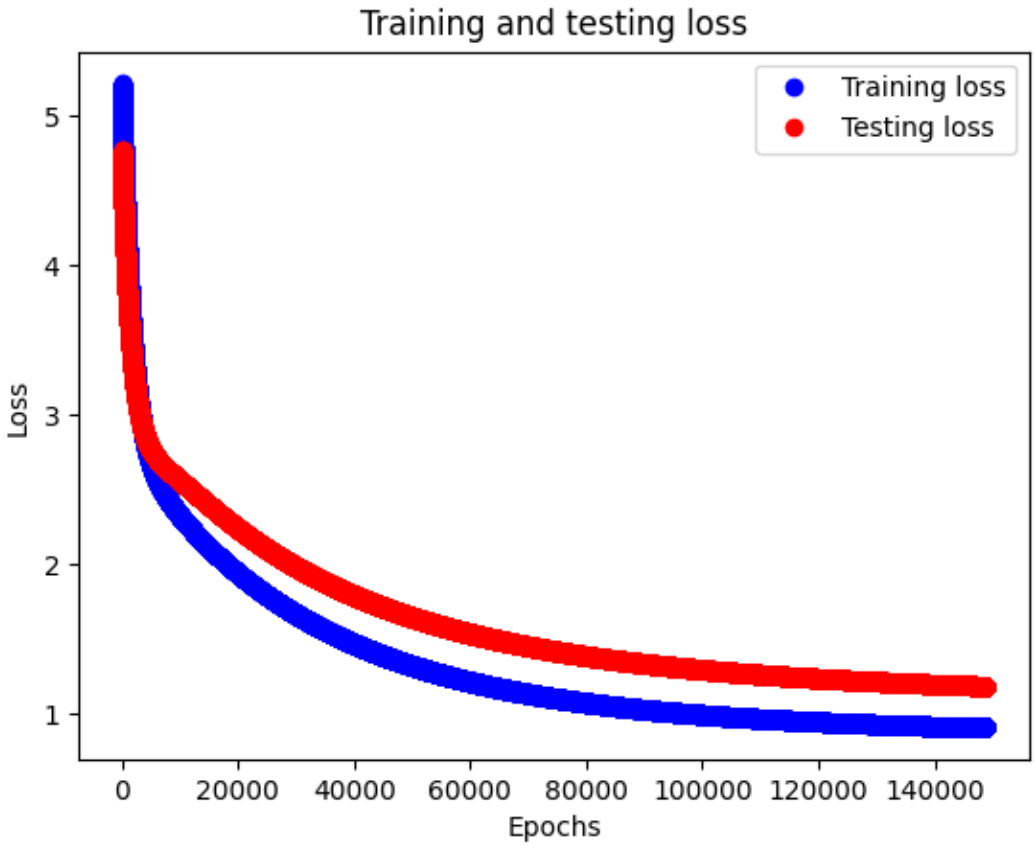

Mean Square Error of homemade gradient decent with a Learning rate 1e-03 and alpha of 1e-05




Mean Square Error of homemade gradient decent with a Learning rate 1e-01 and alpha of 1e-04




Mean Square Error of homemade gradient decent with a Learning rate 1e-02 and alpha of 1e-02




Mean Square Error of homemade gradient decent with a Learning rate 1e-06 and alpha of 1e-04

# Answer for the reminder question

After tuning on the learning rate and alpha value for a l2 regularization, I was able to significantly reduces the mean square error of the model on testing set from roughly 1.2+ to 0.73. Other tricks may be used to obtain better result (for example, a more sophisticated optimizer or a better parameter initialization may be more helpful), but for a homemade model when only tuning on learning rate and regulzaizaion, I am satisfied with the result.