Data1:

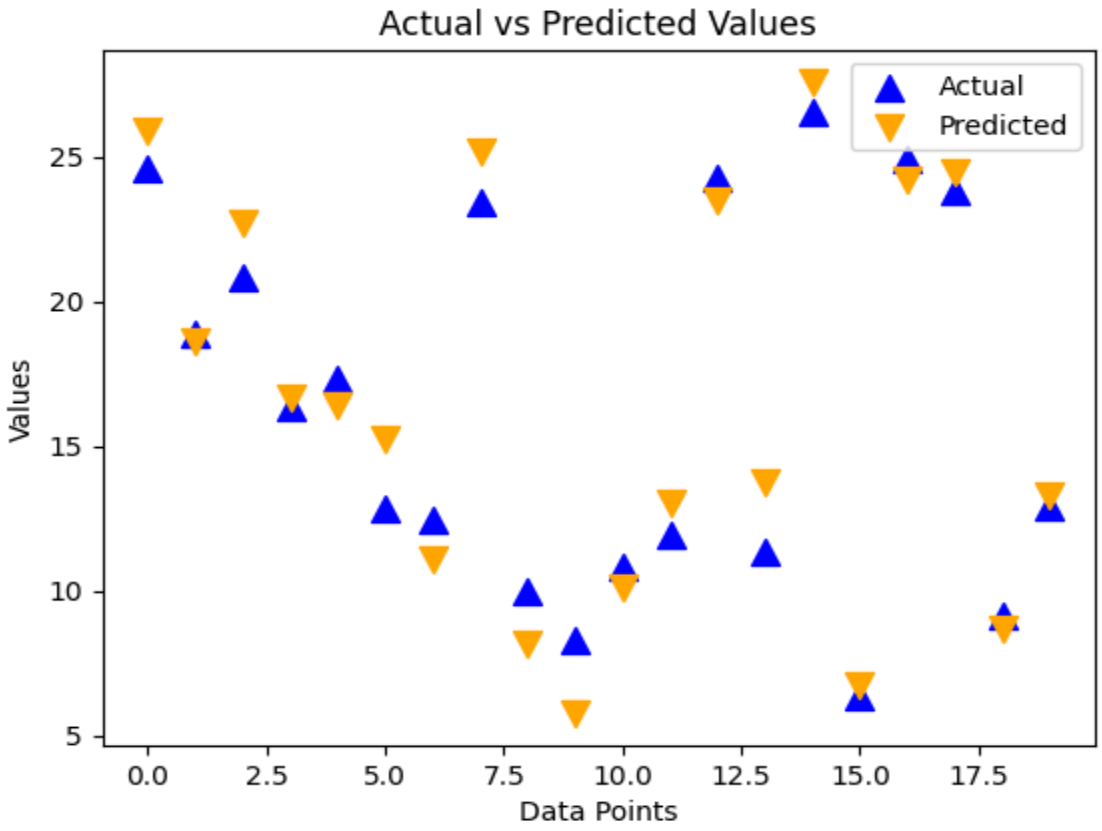
This is a 2-D data, so we can easily plot and visualize. Clearly, after plotting I got that this is following the normal case of Linear Regression. However I tried all the versions and here are the different error terms obtained.

I am showing here the error terms for my own written code and scikit-learn code.

Functions applied	Mean Squared	Mean Absolute	Root Mean Square	R-Squared
multiLinearReg @ mycode	1.84772271	1.15253739	0.30395088	0.95448323
multiLinearReg @ scikit-learn	1.84772271	1.15253739	0.30395088	0.95448323
expoLinearReg @ mycode	2.48731402	1.34586147	0.30395088	0.95448323
expoLinearReg @ scikit-learn	2.48731402	1.34586147	0.35265522	0.93866524
polyLinearReg @ mycode	2.3468628	1.12519248	0.30160663	0.95489479
polyLinearReg @ scikit-learn	1.84772271	1.15253739	0.30395088	0.95448323

Hence I am choosing multiLinearReg for 1 dimension(Basically normal Linear Regression) as the best model for this dataset because the model's error and scikit-learn's error are the same(that's the distinguishing feature here).

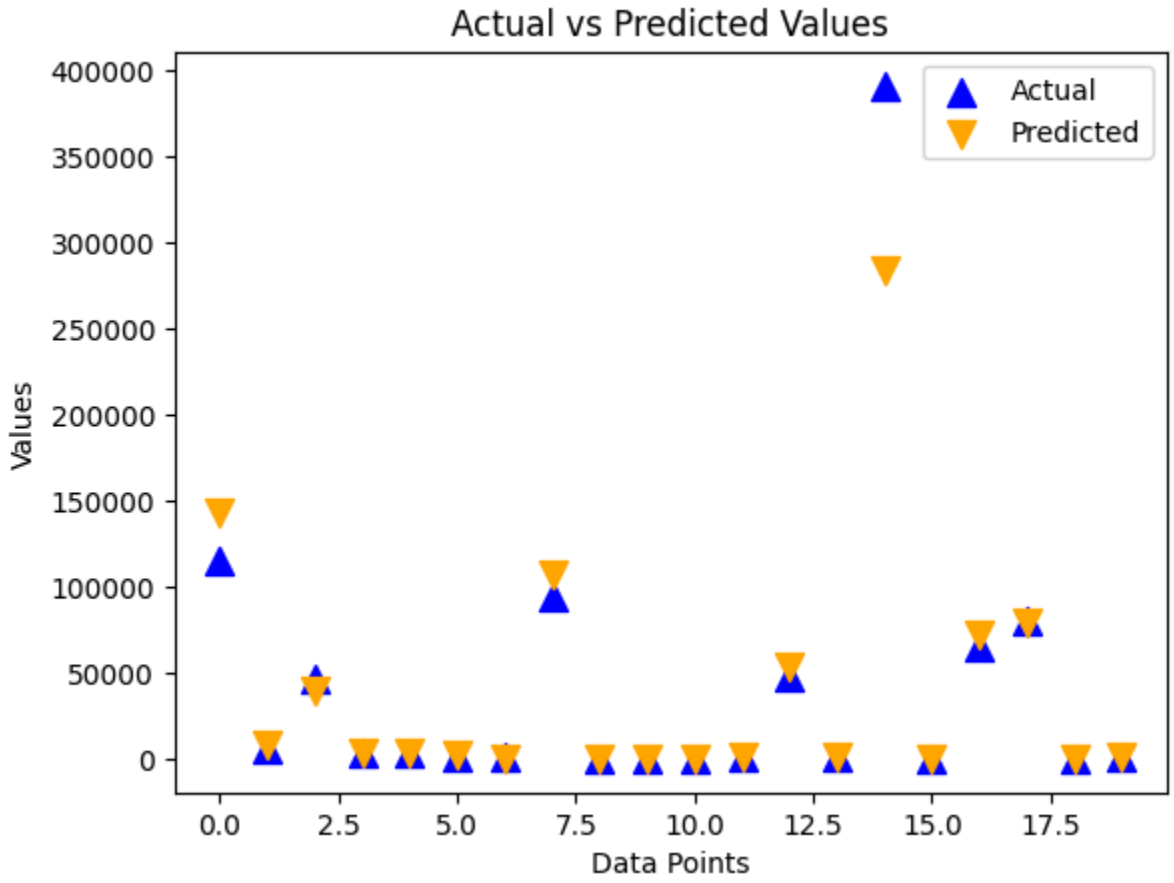
Plotting the data points as what obtained from model v/s actual.(For the multiLinReg)



Data2:

This looks exponential after plotting, so I am going to try the expoLinearReg.

Functions applied	Mean Squared	Mean Absolute	Root Mean Square	R-Squared
expoLinearReg @ mycode	6*10^8	8537.51124713	5578.35228999	0.9188804
expoLinearReg @ scikit-learn	6*10^8	8537.51124713	5578.35228999	0.9188804



Data3:

This doesn't give any clear perspective due to plot. Here I tried different combinations and got a correct function that has better fitting.

Functions applied	Mean Squared	Mean Absolute	Root Mean Square	R-Squared
multiLinearReg @ mycode	0.11112138	0.28848612	0.07453904	0.11860306
expoLinearReg @ mycode	0.19883427	0.39251686	0.09970814	-0.12952174

Here MSE, MAE, RMSE are much better than comparing other data sets. But then also I am not considering that R-Square is very bad. Also plot does not support these two methods. I tried a Polynomial version of Linear Regression over different degrees. The error terms obtained are following:

polyLinReg with different degree	Mean Squared	Mean Absolute	Root Mean Square	R-Squared
Degree 2	0.14770762	0.42634528	0.07660738	0.21769096
Degree 3	0.12634305	0.34208411	0.06387378	0.20166903

I also tried some higher terms but it didn't work. Their R-Square error was worse than these obtained now.
For this data set, I couldn't get which is suitable.

Data4 :

This has 4-dimensions. I applied various functions and the error results are as follows.
This has high accuracy while using multivariate linear regression.

Functions applied	Mean Squared	Mean Absolute	Root Mean Square	R-Squared
multiLinearReg @ mycode	32.12918681	5.10251859	1.26746177	0.98580599

Plot for multivariate Linear Regression.