From the given code snippet for generate regression data present in generate\_regression\_data.py, We have used it to generate 6 test datasets, that we believe cover all the edge cases, the model has to evaluate its performance. We have supported each dataset with what we are expecting from it.   
  
  
**Features and its meaning being used to construct this data:**

m: Regression coefficients (a vector).

b: Intercept.

rnge: The range for randomly generated X values.

N: Number of samples.

scale: Standard deviation of Gaussian noise.

seed: Random seed for reproducibility.

**Data1.csv:**

python generate\_regression\_data.py -N 5000 -m 1.0 2.0 2.0 -b 0.5 -scale 0.1 -rnge -1 1 -seed 42 -output\_file test\_data/collinear\_data.csv

* Test: Handling perfectly correlated features
* Number of Samples: 5,000
* Features: Highly correlated
* Expect:
  + LASSO should select a subset of correlated features
  + Demonstrate feature selection capability
  + Reduce feature dimensionality

**Data2.csv:**

python generate\_regression\_data.py -N 10000 -m 1.0 1.0 0.1 -b 0.3 -scale 0.5 -rnge -2 2 -seed 123 -output\_file test\_data/noisy\_collinear.csv

* Test: Robustness to noise in correlated features
* Number of Samples: 10,000
* Features: Correlated with added noise
* Expect:
  + Stable coefficient selection
  + Resilience to small variations
  + Effective noise filtering

**Data3.csv:**

python generate\_regression\_data.py -N 7500 -m 2.0 0.0 0.0 1.5 0.0 0.0 -b 0.2 -scale 0.2 -rnge -3 3 -seed 256 -output\_file test\_data/sparse\_true.csv

* Test: Correct sparse feature selection
* Number of Samples: 7,500
* Features: Mostly zero coefficients
* Expect:
  + Accurately identify non-zero features
  + Zero out irrelevant features
  + Demonstrate sparsity property

**Data4.csv:**

python generate\_regression\_data.py -N 5000 -m 1.0 1.5 2.0 3.0 4.0 5.0 6.0 7.0 0.0 0.0 0.0 0.0 -b 0.1 -scale 0.3 -rnge -5 5 -seed 789 -output\_file test\_data/data4.csv

* Test: Complex high-dimensional sparse feature selection
* Number of Samples: 5,000
* Features: 12 features
* Expect:
  + Effectively identify most important features
  + Sparsity in feature selection
  + Robust performance in high-dimensional space

**Data5.csv:**

python generate\_regression\_data.py -N 6000 -m 2.0 1.5 1.0 0.5 0.3 0.2 0.1 0.05 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -b 0.2 -scale 0.4 -rnge -3 3 -seed 456 -output\_file test\_data/data5.csv

* Test: Robustness in noisy, feature-rich environment
* Number of Samples: 6,000
* Features: 16 features
* Expect:
  + Noise filtering
  + Stable coefficient estimation
  + Gradual feature importance ranking

**Data6.csv:**

python generate\_regression\_data.py -N 7500 -m 10.0 5.0 2.0 1.0 0.5 0.1 0.05 0.01 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -b 0.5 -scale 0.5 -rnge -10 10 -seed 999 -output\_file test\_data/data6.csv

* Test: Handling extreme coefficient scales and sparsity
* Number of Samples: 7,500
* Features: 20 features
* Expect:
  + Accurate handling of widely varying coefficient magnitudes
  + Extreme sparsity
  + Robust regularization