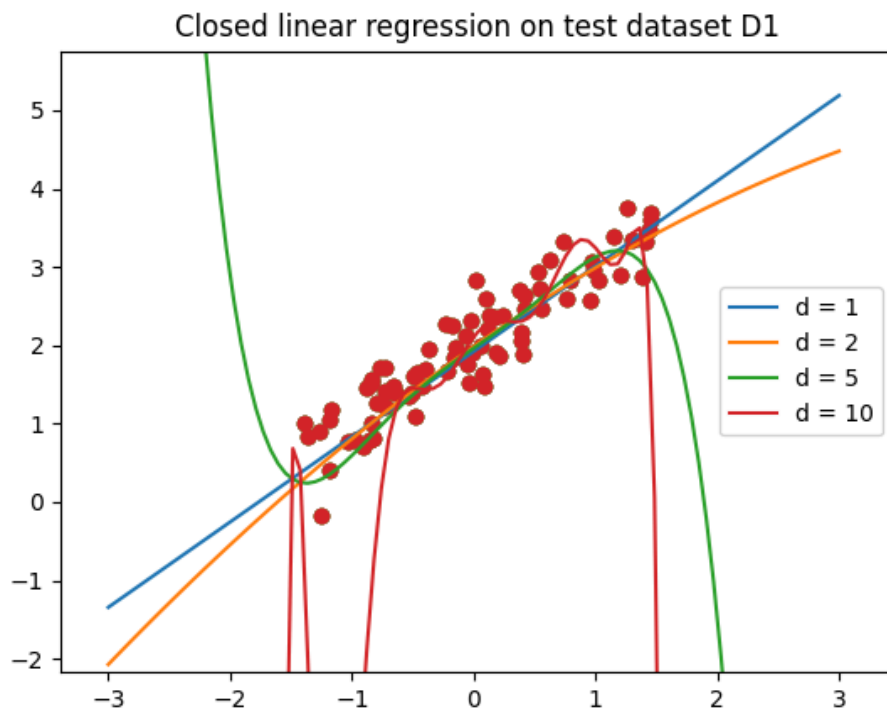
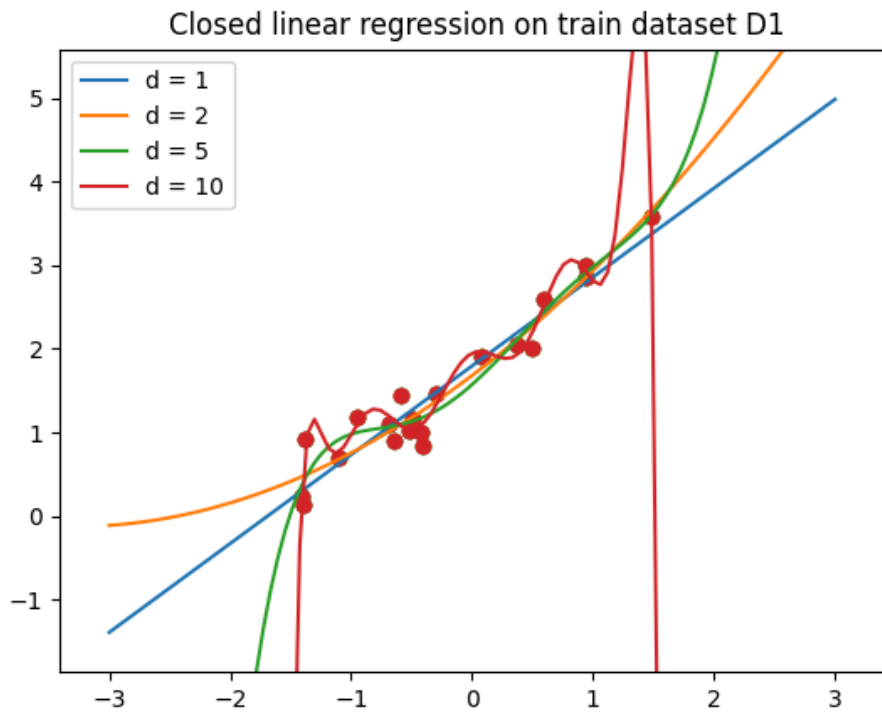
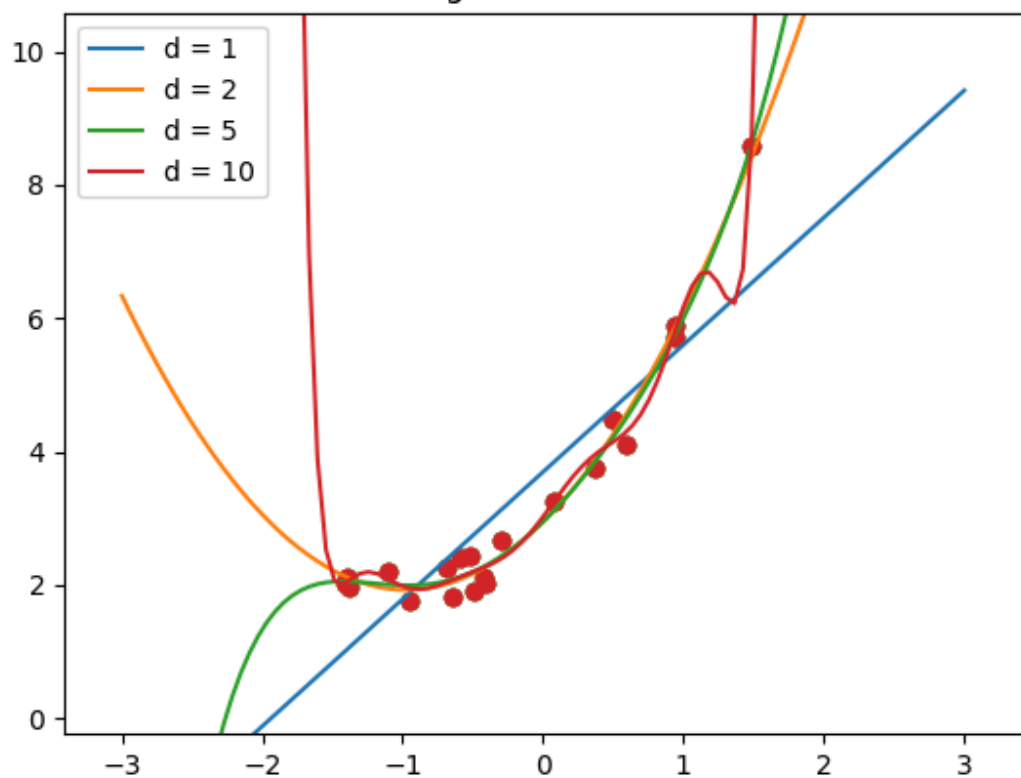


# Lab Assignment 7

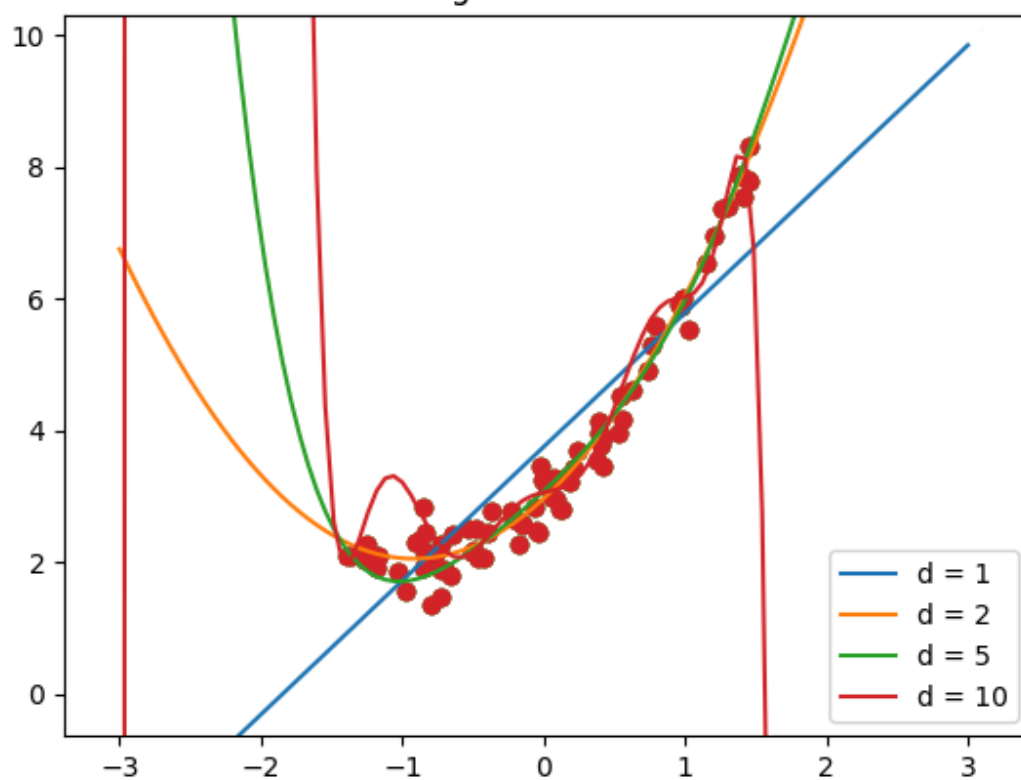
2) (c) Function obtained along with data points:



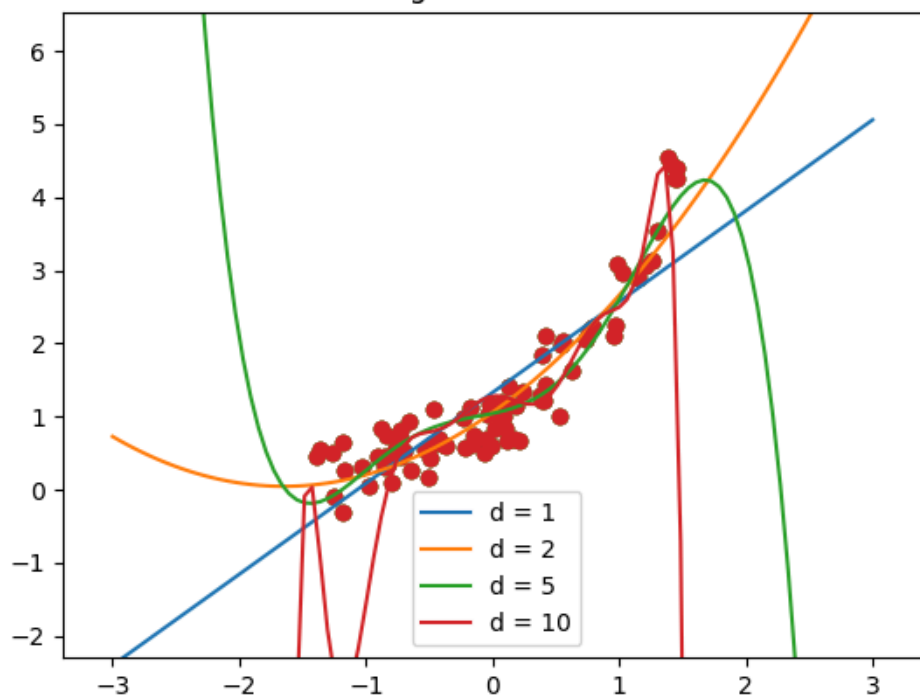
Closed linear regression on train dataset D2



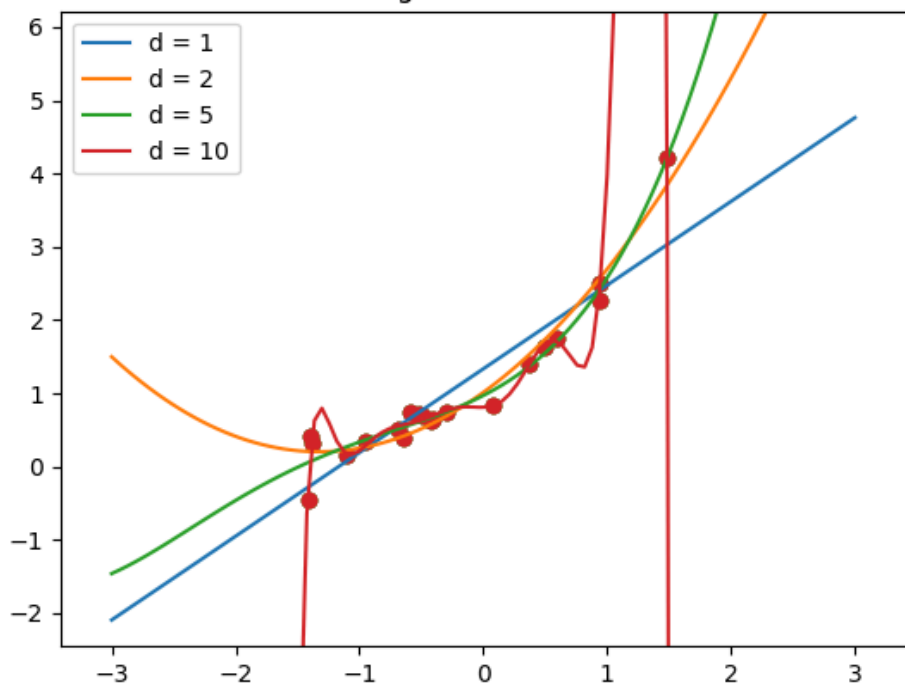
Closed linear regression on test dataset D2



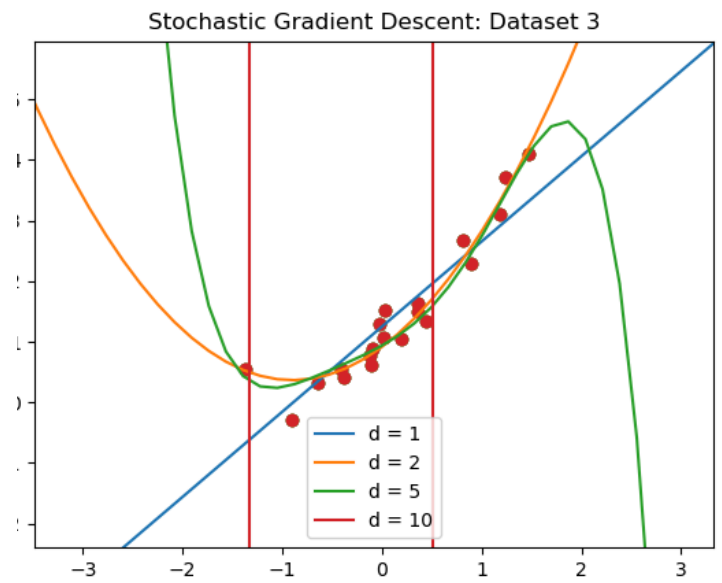
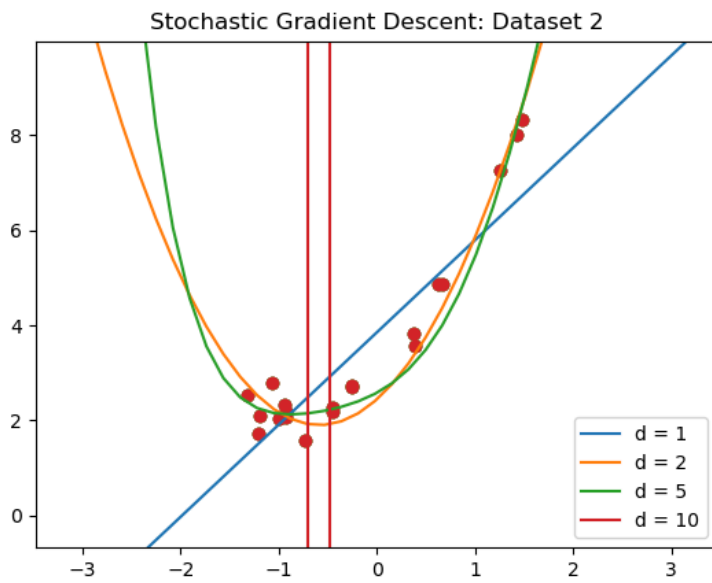
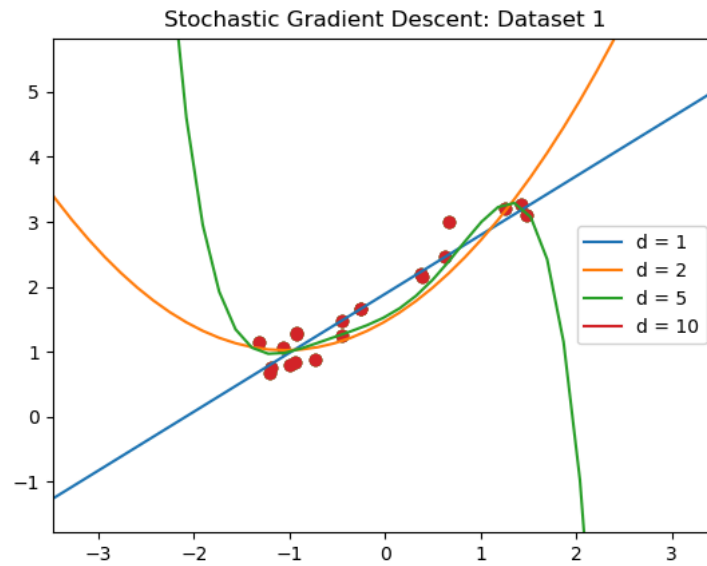
Closed linear regression on test dataset D3



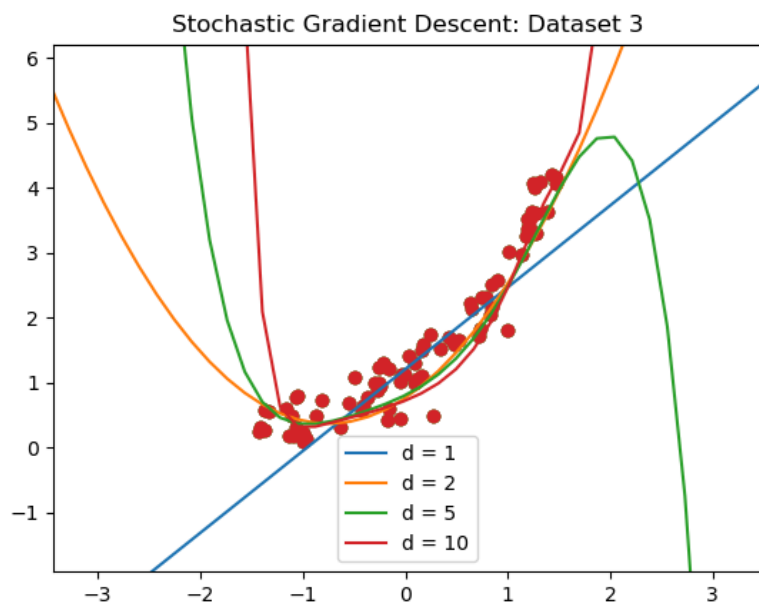
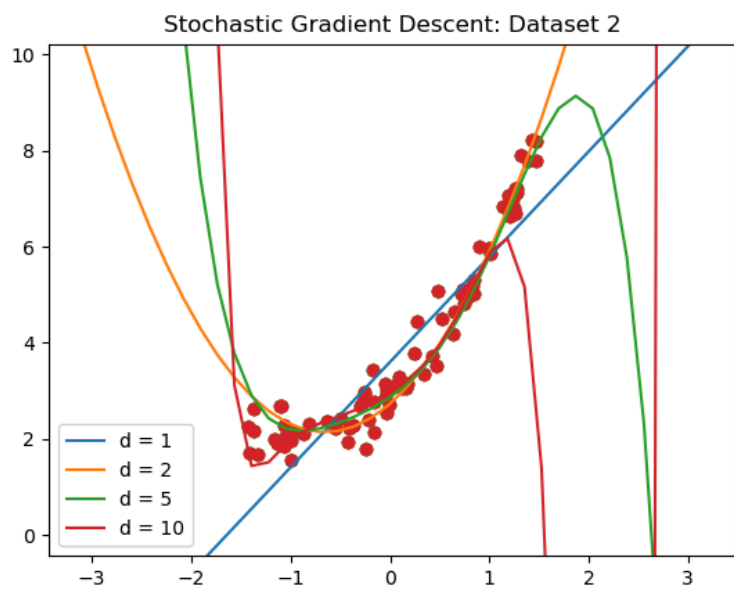
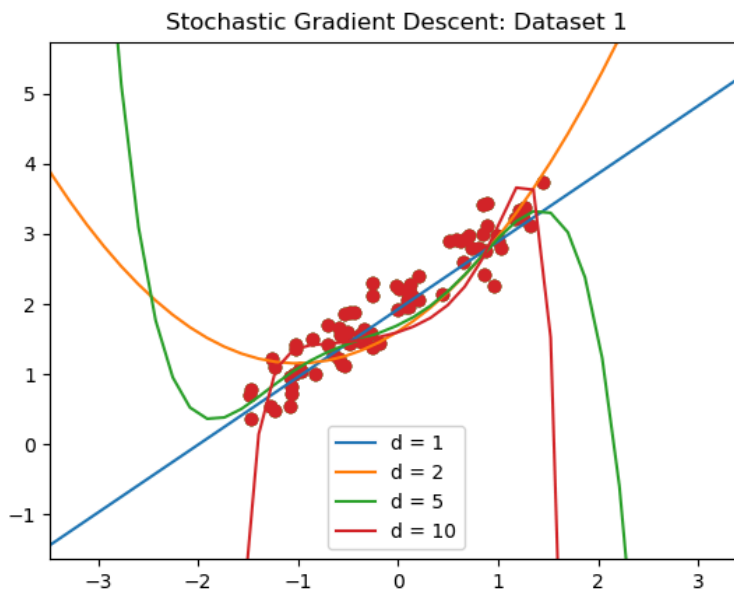
Closed linear regression on train dataset D3



### 3) (a) Function obtained from SGD along with train dataset



(b) Function obtained from SGD along with test dataset



We also observe that the co-efficient vector obtained from the closed-form regression and ridge regression are almost the same.

4) For Dataset1:

```
$ python -u "d:\assignments\Machine learning\lab7\Ques1-4.py"
Enter space separated values of a, b, c: 1 2 3
Train least square error for d = 1 and lambda = 1: [1.31090448]
Test least square error for d = 1 and lambda = 1: [5.88967934]
Train least square error for d = 2 and lambda = 1: [1.46800759]
Test least square error for d = 2 and lambda = 1: [6.02454465]
Train least square error for d = 5 and lambda = 1: [1.46822683]
Test least square error for d = 5 and lambda = 1: [11.5446899]
Train least square error for d = 10 and lambda = 1000.0: [55.0354434]
Test least square error for d = 10 and lambda = 1000.0: [804.62480833]
Best d for dataset: 1
```

For Dataset2:

```
Train least square error for d = 1 and lambda = 0.0001: [723.25145464]
Test least square error for d = 1 and lambda = 0.0001: [5513.39133892]
Train least square error for d = 2 and lambda = 0.1: [1.8890428]
Test least square error for d = 2 and lambda = 0.1: [4.83989345]
Train least square error for d = 5 and lambda = 1: [2.99243117]
Test least square error for d = 5 and lambda = 1: [24.34636872]
Train least square error for d = 10 and lambda = 0.1: [1.04698573]
Test least square error for d = 10 and lambda = 0.1: [1182.97065487]
Best d for dataset: 2
```

For Dataset3:

```
Train least square error for d = 1 and lambda = 100.0: [9921.3546246]
Test least square error for d = 1 and lambda = 100.0: [36283.27217957]
Train least square error for d = 2 and lambda = 0.0001: [2178.1293627]
Test least square error for d = 2 and lambda = 0.0001: [10627.74498419]
Train least square error for d = 5 and lambda = 100.0: [37.99558131]
Test least square error for d = 5 and lambda = 100.0: [327.45019807]
Train least square error for d = 10 and lambda = 1000.0: [43.32101086]
Test least square error for d = 10 and lambda = 1000.0: [544.00860577]
Best d for dataset: 5
```

5) We observe that for Red wine, we have best  $d = 4$ , and for White wine, best  $d = 6$

```
• $ python -u "d:\assignments\Machine learning\lab7\wine-datasets.py"
RED WINE DATASET PREDICTION

Test least square error for d = 1 and lambda = 0.01: [[277.31021668]]
Test least square error for d = 2 and lambda = 1: [[270.00185602]]
Test least square error for d = 3 and lambda = 0.1: [[264.39135347]]
Test least square error for d = 4 and lambda = 100.0: [[255.89874025]]
Test least square error for d = 5 and lambda = 10.0: [[290.05910368]]
Test least square error for d = 6 and lambda = 1: [[271.41638228]]
Test least square error for d = 7 and lambda = 1000.0: [[274.83362336]]
Test least square error for d = 8 and lambda = 10000.0: [[74518.64022436]]
Test least square error for d = 9 and lambda = 10000.0: [[498118.06748657]]
Test least square error for d = 10 and lambda = 10000.0: [[11012795.16968675]]

Best d: 4
Best lambda: 100.0

WHITE WINE DATASET PREDICTION

Test least square error for d = 1 and lambda = 0.0001: [[1099.72590213]]
Test least square error for d = 2 and lambda = 0.0001: [[1331.4094146]]
Test least square error for d = 3 and lambda = 10000.0: [[1165.91208843]]
Test least square error for d = 4 and lambda = 1: [[331745.89252642]]
Test least square error for d = 5 and lambda = 100.0: [[1228.87906362]]
Test least square error for d = 6 and lambda = 10.0: [[1045.8060289]]
Test least square error for d = 7 and lambda = 100.0: [[1.91853413e+08]]
Test least square error for d = 8 and lambda = 100.0: [[3.26021304e+09]]
Test least square error for d = 9 and lambda = 10000.0: [[2.50508837e+11]]
Test least square error for d = 10 and lambda = 10000.0: [[4969291.28242735]]

Best d: 6
Best lambda: 10.0
```