Principal

Bharath Variar, 2019B5A70930H
Kanika Gandhi, 2019B5A71080H
Karan Moza, 2019B4A71372H

Assigment-2 Report
CS F320: Foundations of Data Science
Department of Computer Science and Information
BITS Pilani, Hyderabad Campus
November2022

Contents

1 Question 2A

1.1 Pearson Correlation Coefficient

Pearson Correlation Coefficient between 2 sets of data x and y is calculated as:

$$r = \frac{Cov(x, y)}{\sigma_x \sigma_y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{1}$$

The Pearson Correlation coefficient is a measure of how each dataset changes with respect to the other. The value of r $\in$ [-1, 1]. If the magnitude of r is a higher value, there is strong linear relationship between x and y.
For r = 0, there is no correlation between x and y. We have calculated Pearson Correlation Coefficient between y and each of the 26 features. For linear regression using n features, we have chosen the top n features with the highest magnitude of the coefficient.

Figure 1: Training Error after feature selection

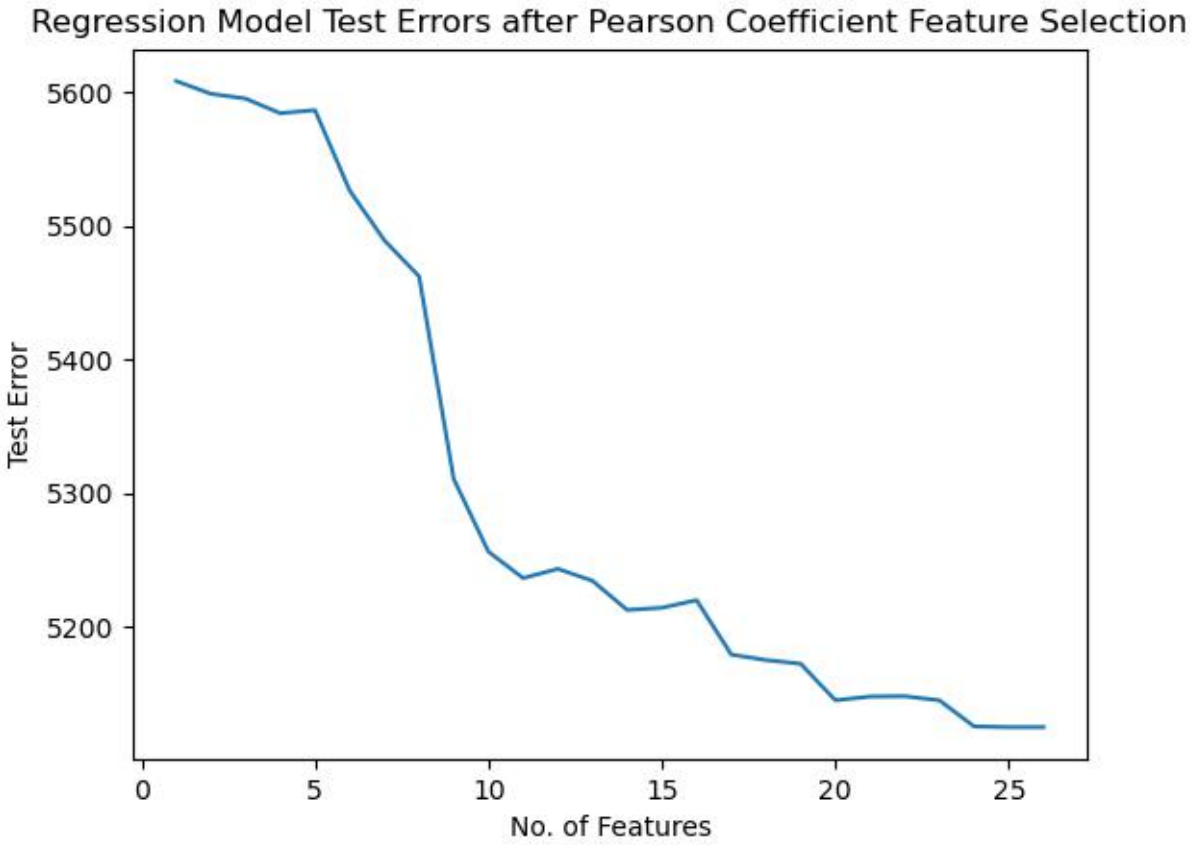

Figure 2: Testing Error after feature selection

As a result, we get the following testing and training errors for each value of n:

Number of Components	Train MSE	Test MSE
1	5231.579	5608.494
2	5227.19	5598.944
3	5226.136	5595.36
4	5218.753	5584.401
5	5213.327	5586.691
6	5181.597	5526.516
7	5148.425	5489.185
8	5117.057	5462.249
9	4959.1	5310.35
10	4897.697	5256.035
11	4877.113	5236.265
12	4870.357	5243.169
13	4854.419	5234.243
14	4843.046	5212.385
15	4843.853	5214.069
16	4844.941	5219.727
17	4825.52	5178.999
18	4803.237	5174.887
19	4802.579	5172.124
20	4794.23	5144.745
21	4785.893	5147.564
22	4784.476	5147.803
23	4784.147	5144.723
24	4763.616	5125.158
25	4763.604	5124.65
26	4763.604	5124.643

Table 1: Error Table for PCC

From this, we observe that minimum training error is for 25 features and minimum testing error is for 26 features.
Therefore, we select the model with 26 features as the optimal model.

1.2 Principal-Component Analysis

Principal Component Analysis (PCA) is used to reduce the number of features used for linear regression, i.e., dimensionality reduction. This is done by projecting the datapoints onto a lower number of dimensions. These dimensions are selected such that they capture the maximum possible variance of the original data. The n^{th} principal

component corresponds to projection onto n-dimensions. The training error, testing error and percentage of variance captured after linear regression using n transformed features is:

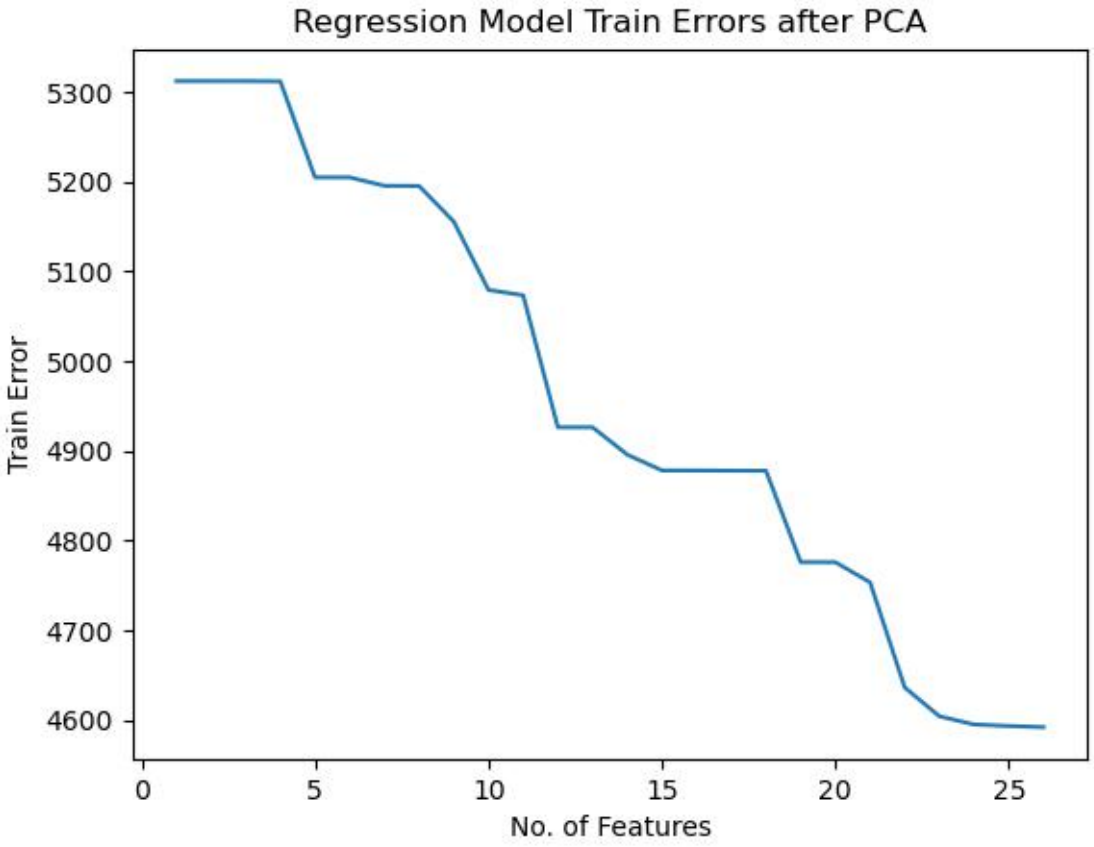

Figure 3: Training Error after PCA

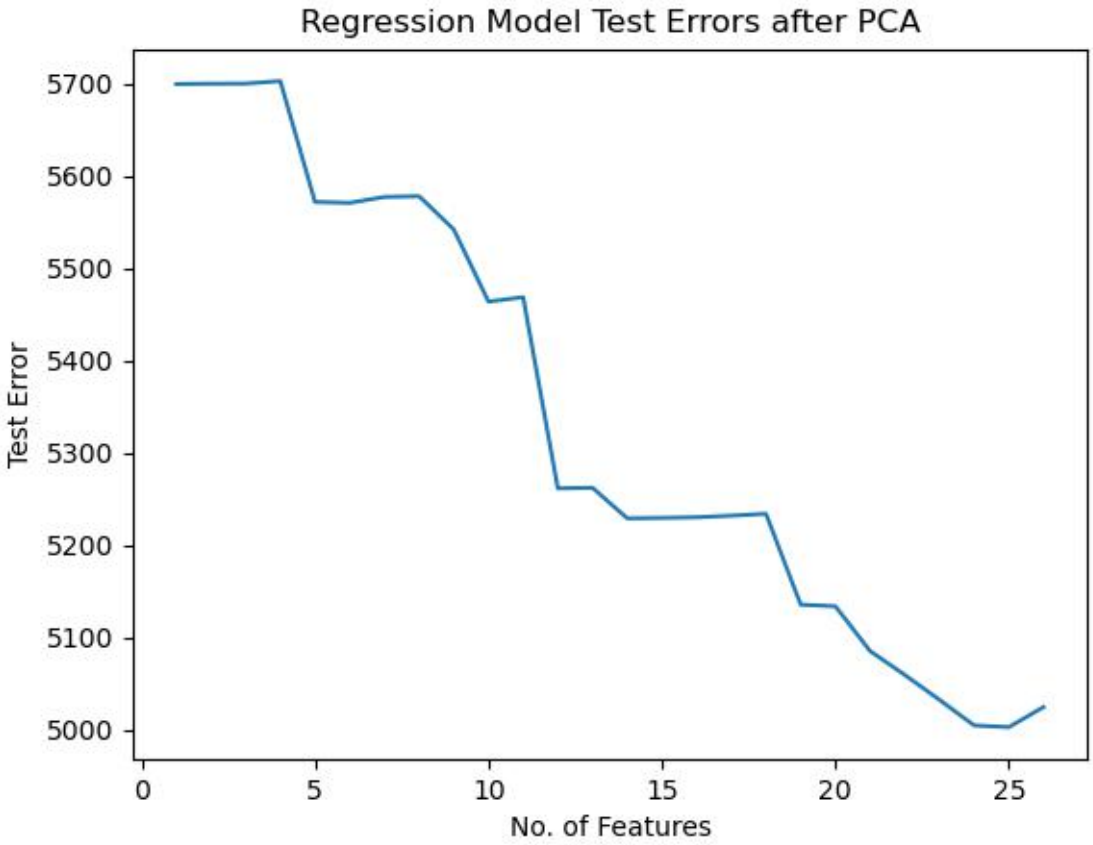

Figure 4: Testing Error after PCA

Number of Components	Train MSE	Test MSE	Variance %
1	5312.286	5700.099	54.87311
2	5312.273	5700.466	74.15442
3	5312.258	5700.618	81.50752
4	5311.972	5703.322	87.82324
5	5204.987	5572.232	92.5398
6	5204.941	5571.238	95.68014
7	5195.553	5577.596	97.91332
8	5195.35	5578.423	98.44902
9	5155.67	5542.46	98.81189
10	5079.481	5464.112	99.16108
11	5073.467	5468.838	99.34855
12	4926.398	5261.52	99.50793
13	4926.374	5262.16	99.63348
14	4895.873	5229.054	99.75331
15	4878.227	5229.477	99.81443
16	4878.151	5230.179	99.8573
17	4877.965	5231.898	99.89077
18	4877.902	5233.965	99.92015
19	4776.071	5135.389	99.94289
20	4775.927	5133.711	99.96188
21	4753.472	5085.099	99.9757
22	4636.199	5059.168	99.98575
23	4604.103	5032.138	99.99205
24	4595.01	5004.424	99.99664
25	4593.35	5002.51	100
26	4592.034	5024.306	100

Table 2: Error Table for PCA

From above graphs, we observe that minimum training error is for 26 features and minimum testing error is for 25 features.
Therefore, we select the model with 25 features as the optimal model. We can also see from figure 5 that the variance captured saturates after 8-9 components.

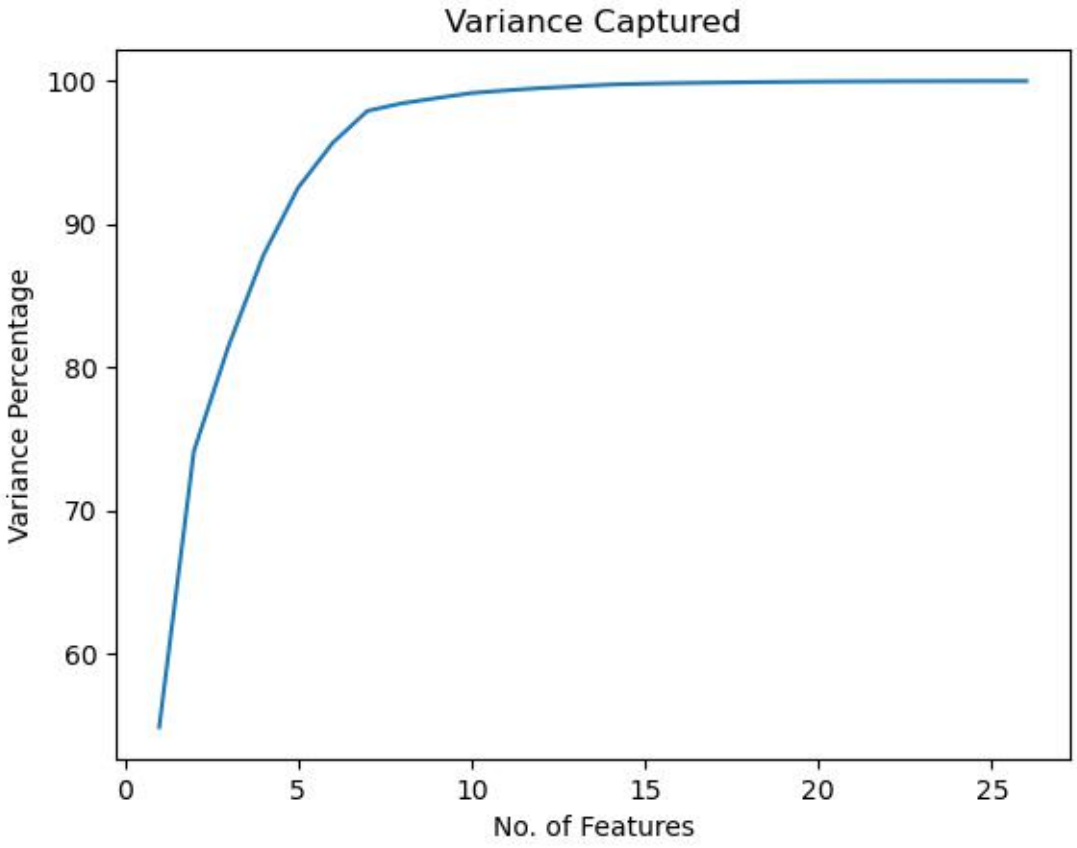

Figure 5: Variance Captured vs. Number of components

2 Question 2B

2.1 Greedy Forward Feature Selection

In greedy forward algorithm, the machine starts with choosing a single feature with the minimum loss and adds this feature as the primary component.
It then iterates the process for the rest of the features, algorithmically adding the feature that contributes the least loss.
The algorithm finally chooses that set of features, which has the least testing error, thus decreasing the number of components of the original dataset.

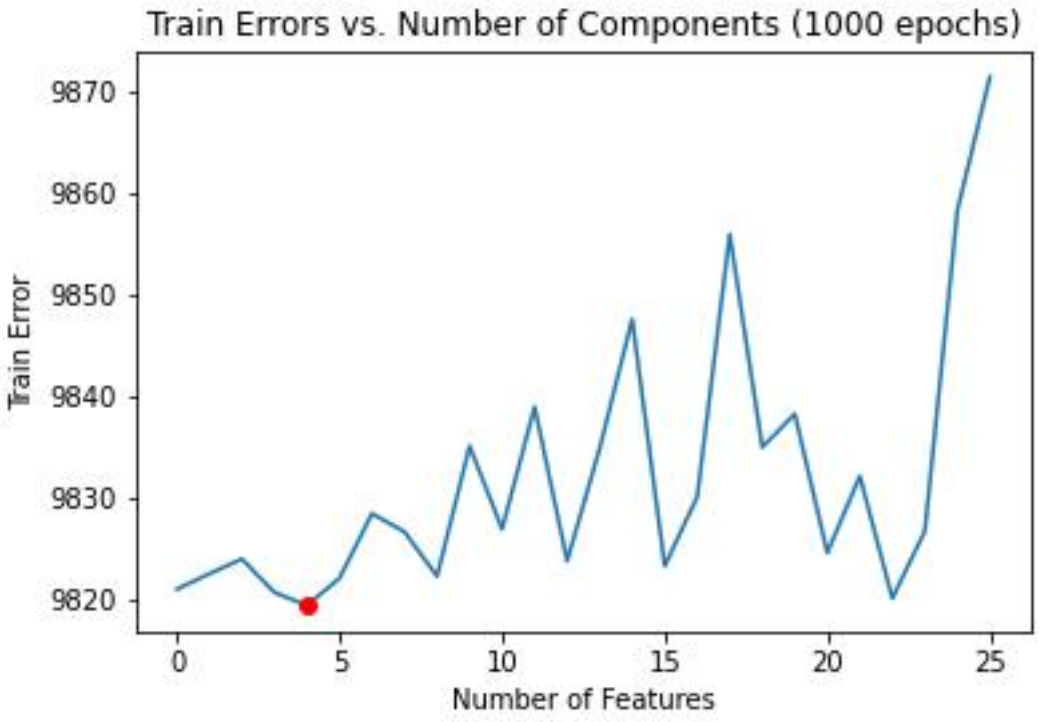

Figure 6: Training Error with greedy forward

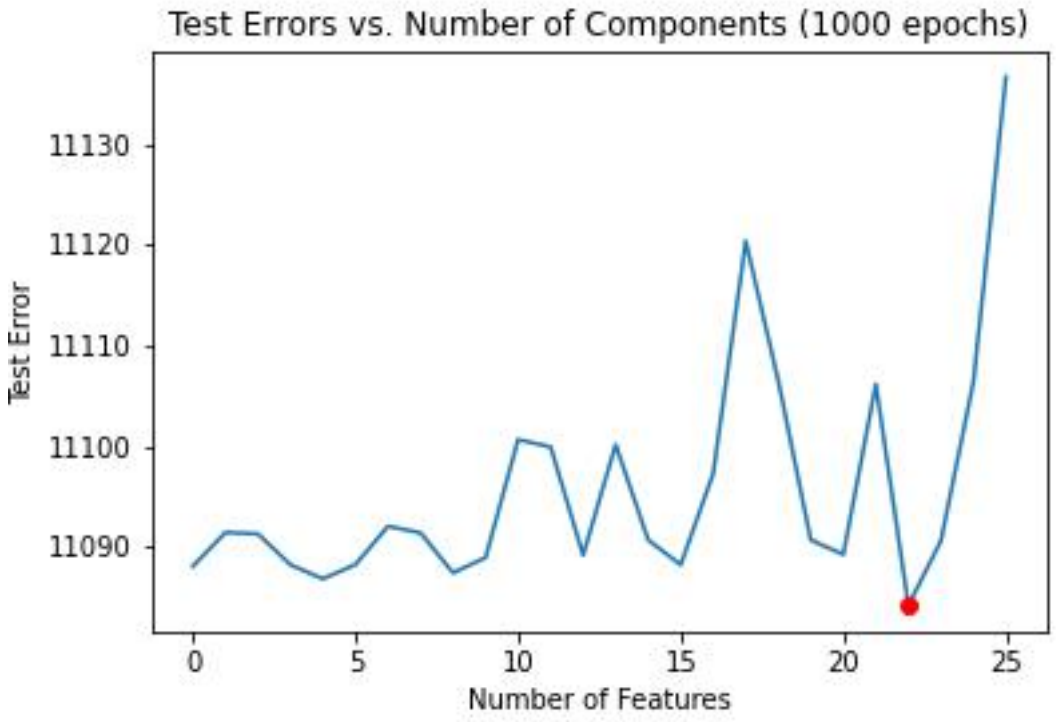

Figure 7: Testing Error with greedy forward

From figure 7, we can see that the minimum testing error (11106.15) occurs when 22 components are used.

Number of Components	Train Error	Test Error
1	9821.086	11088.06
2	9822.611	11091.41
3	9824.078	11091.27
4	9820.763	11088.2
5	9819.502	11086.81
6	9822.168	11088.23
7	9828.539	11092.03
8	9826.745	11091.37
9	9822.36	11087.4
10	9835.188	11088.92
11	9827.021	11100.63
12	9839.054	11099.92
13	9823.873	11089.16
14	9834.963	11100.14
15	9847.717	11090.67
16	9823.378	11088.2
17	9830.195	11097.27
18	9856.07	11120.37
19	9835.09	11106.48
20	9838.369	11090.67
21	9824.68	11089.23
22	9832.228	11106.15
23	9820.189	11084.18
24	9826.836	11090.56
25	9858.434	11106.35
26	9871.594	11136.67

Table 3: Error Table for Greedy Forward

In comparison to vanilla regression with all features, the model with 15 components has lesser testing and training errors.

2.2 Greedy Backward Feature Selection

The greedy backward algorithm on the other hand, starts with choosing all features and drops one feature at a time with the maximum loss and adds the rest of the features as the reduced parameters.
It then iterates the process for the rest of the features, algorithmically removing the

feature that contributes to the most loss.
The algorithm finally chooses that set of features, which has the least testing error, thus decreasing the number of components of the original dataset.

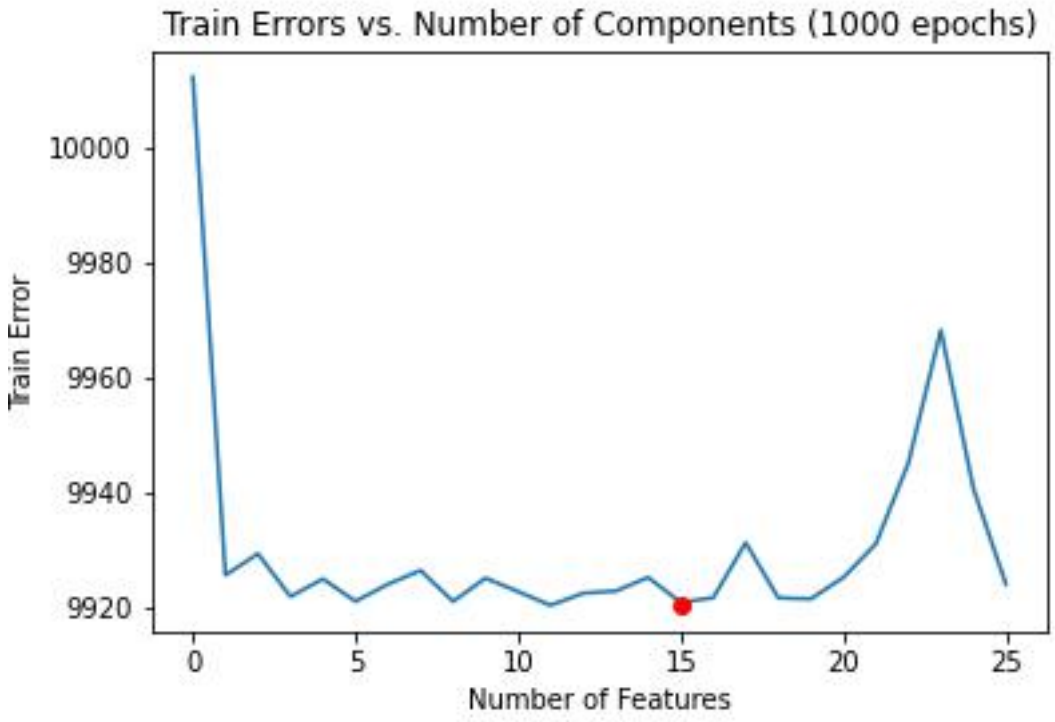

Figure 8: Training Error with greedy backward

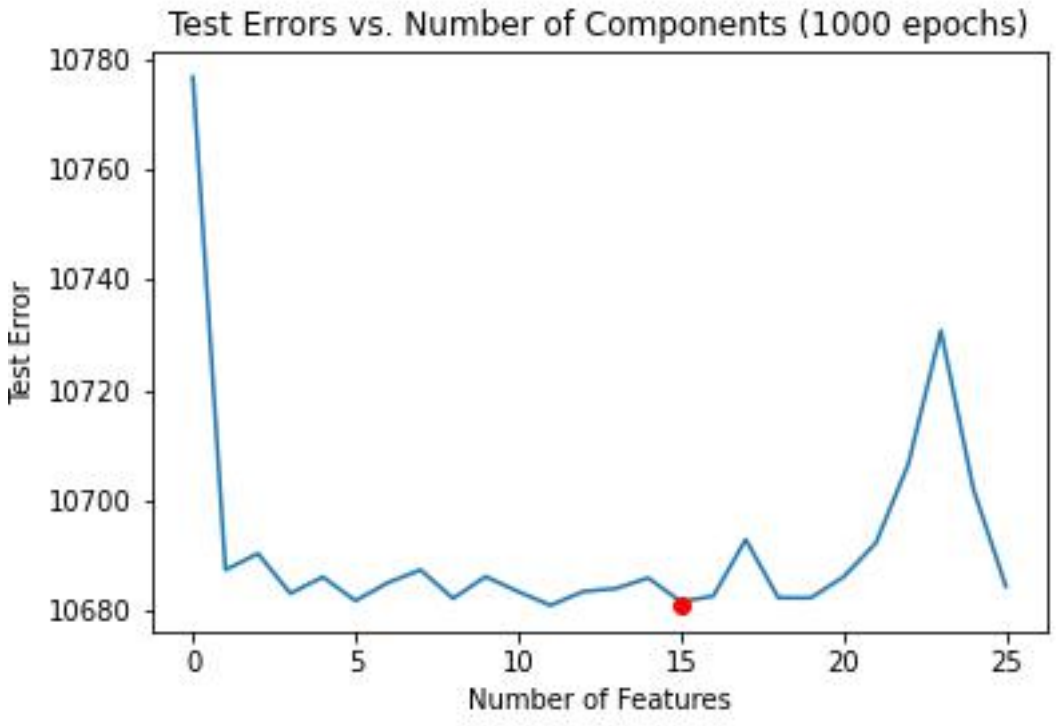

Figure 9: Testing Error with greedy backward

From figure 9, we can see that the minimum testing error (11090.67) occurs when 15 components are used.

Number of Components	Train Error	Test Error
1	9821.086	11088.06
2	9822.611	11091.41
3	9824.078	11091.27
4	9820.763	11088.2
5	9819.502	11086.81
6	9822.168	11088.23
7	9828.539	11092.03
8	9826.745	11091.37
9	9822.36	11087.4
10	9835.188	11088.92
11	9827.021	11100.63
12	9839.054	11099.92
13	9823.873	11089.16
14	9834.963	11100.14
15	9847.717	11090.67
16	9823.378	11088.2
17	9830.195	11097.27
18	9856.07	11120.37
19	9835.09	11106.48
20	9838.369	11090.67
21	9824.68	11089.23
22	9832.228	11106.15
23	9820.189	11084.18
24	9826.836	11090.56
25	9858.434	11106.35
26	9871.594	11136.67

Table 4: Error Table for Greedy Backward

In comparison to vanilla regression with all features, the model with 15 components has lesser testing and training errors.

```
print(
    f"Regression with all 26 features has: \nTraining error = {train_global[0]}\nTesting error = {test_global[0]}"
)
```

```
Regression with all 26 features has:
Training error = 10012.562258432208
Testing error = 10776.995787529328
```

Figure 10: Comparison with vanilla linear regression