

Harrison Lloyd
CSCI 485
Prof. Shen
2/23/26

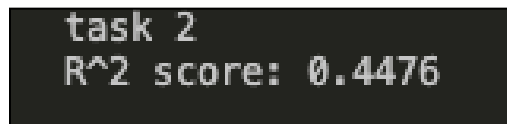
Homework 2

Task 1:

For this assignment, I loaded in the Diabetes dataset using `sklearn.datasets.load_diabetes()`. I found that there were 442 rows and 10 features, such as BMI, S1, S5, AGE, SEX, etc. As instructed, I split the dataset 80/20, 80% for training and 20% for testing. This means that there were 353 training samples and 89 testing samples.

Task 2:

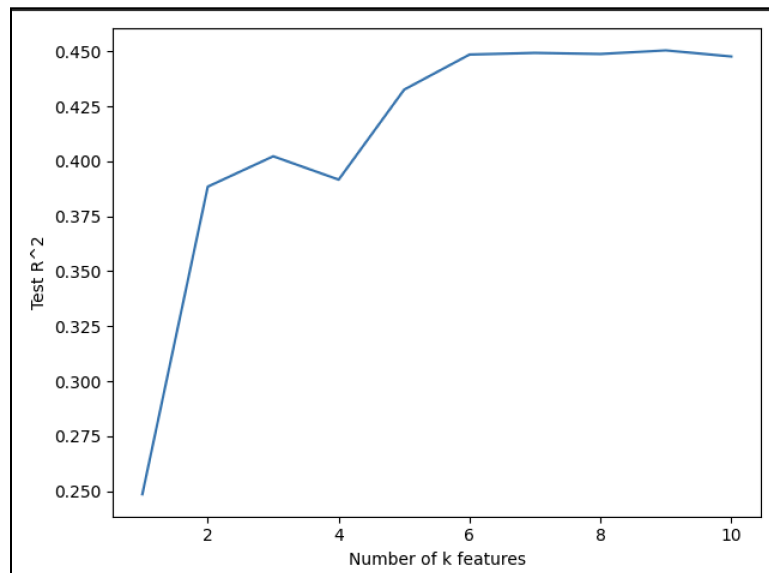
Before doing feature selection, I trained a standard Linear Regression model using all 10 of the features. Afterward, I evaluated the performance of the test set using the R^2 score. I got an R^2 score of about 0.44, however, it may be slightly different if it were to run again. Here is an image below of the output I got in the terminal.

A terminal window with a dark background and light green text. The text reads "task 2" on the first line and "R^2 score: 0.4476" on the second line.

```
task 2
R^2 score: 0.4476
```

Task 3:

I used an RFE with LinearRegression as the base estimator. RFE repeatedly fits a linear regression model and ranks features based on their importance, then removes the least important feature(s) to create a subset. I ran RFE starting with $K=10$ (all the features) and repeated it till we were left with a single feature, $K=1$. Each step represents removing 1 additional feature. For each K value, I trained a linear regression model using only the selected features and then evaluated it on the test set with R^2 . In addition, I recorded the model coefficients at each k so I could see how the feature changes as they get removed. I also plotted the R^2 score against the number of retained features, which can be seen below. In the graph, you can see that with only 1 feature, the performance is low. There's a huge jump from 1 feature to 2, and it continued to climb until around the 6th feature, where it leveled off. In order to choose the optimal amount of features, I used a threshold of 0.01, as it was the example in the assignment. I selected the largest k where the improvement was still at least 0.01, since improvements smaller than that likely meant the returns would be small. This left me with about 6 features (sex, bmi, bp, s1, s2, s5)



```
Task 3.5
threshold=0.01
chosen k=6 with test R^2=0.4485
the 6 features at k: ['sex', 'bmi', 'bp', 's1', 's2', 's5']
```

Task 4:

I created a table showing the coefficients of each feature at each iteration of RFE in the file `rfe_coefficients_by_k.csv`. The 3 most important features are S5, S1, and S2. S5 has the strongest positive effect, as its value is a little more than 874, so it has a higher predicted 'disease progression.' S1 has the strongest negative effect, so a higher S1 could indicate a lower predicted progression. S2, like S5, is strongly positive, meaning it contributes to an increase in predicting disease progression. The initial ranking by coefficient size mostly matched what RFE selected. The biggest coefficients (S1, S5, BMI, S2, BP) were all part of the final set. Sex managed to stay in, even though it had a smaller coefficient, so it may impact something that the other features don't. On the other hand, the other 4 features, S3, S4, S6, and age, were eliminated first, which likely showed that they either had weak predictive power or were redundant with the more important features.

Task 5:

I learned that not all features are equally important, and that it doesn't always hurt performance if you drop some of the least affected features. I can also see that the model

performed just as well with 6 features as it did with 10, meaning 4 of them basically didn't add much value. Both RFE and LASSO compare in that they eliminate weak features, although the way they do it is different. RFE removes the features one at a time, while LASSO shrinks the coefficients with regularization and eliminates the features at once. RFE gives you a little more control over how many features to keep, since you decide exactly how many you want. Based on the selected features, I can see that diabetes progression is mainly related to health since the strongest predictors were blood measurements and BMI. Heart health likely is a factor, too, since blood pressure was a significant measure of disease progression. One surprising thing to me was that age got eliminated first, but I guess that could be explained by the blood measurements already.

ChatGPT Citation

I asked ChatGPT, “How do I identify the optimal number of features using a threshold for significant R^2 improvement in RFE?” and it responded with this:

1. **Pick a threshold** (like 0.01) for what you'll count as a “meaningful” R^2 improvement.
2. **Sort the k values in increasing order** (so you're comparing $k - 1 \rightarrow k$).
3. **Loop from the second k to the last** (start at index 1), so you always have a previous k to compare to.
4. Inside the loop, set **current k** and **previous k**.
5. Compute the **marginal improvement**: $\Delta R^2 = R^2(\text{current}) - R^2(\text{previous})$.
6. Store that delta (like in a dictionary keyed by current k), then later use it to pick the “optimal” k based on your threshold rule.

I also told it to generate the README, which it did, so the content in the README is ChatGPT made and easy to follow.