Linear Regression and Regularization

Applied Machine Learning for Educational Data Science

true

10/12/2021

Contents

[Updated: Thu, Oct 14, 2021 - 12:16:38]

In the machine learning literature, the prediction algorithms are classified into two main categories: *supervised* and *unsupervised*. Supervised algorithms are being used when the dataset has an actual outcome of interest to predict (labels), and the goal is to build the "best" model predicting the outcome of interest that exists in the data. On the other side, unsupervised algorithms are being used when the dataset doesn't have an outcome of interest, and the goal is typically to identify similar groups of observations (rows of data) or similar groups of variables (columns of data) in data. In this course, we plan to cover a number of *supervised* algorithms. Linear regression is one of the simplest approach among supervised algorithms, and also one of the easiest to interpret.

Linear Regression

Model Description

In most general terms, the linear regression model with P predictors (X_1,X_2,X_3,. . . ,X_p) to predict an outcome (Y) can be written as the following:

$$Y = \beta_0 + \sum_{p=1}^{P} \beta_p X_p + \epsilon.$$

In this model, Y represents the observed value for the outcome for an observation, X_p represents the observed value of the p^{th} variable for the same observation, and β_p is the associated model parameter for the p^{th} variable. ϵ is the model error (residual) for the observation.

This model includes only the main effects of each predictor and can be easily extended by including a quadratic or higher-order polynomial terms for all (or a specific subset of) predictors. For instance, the model below includes all first-order, second-order, and third-order polynomial terms for all predictors.

$$Y = \beta_0 + \sum_{p=1}^{P} \beta_p X_p + \sum_{k=1}^{P} \beta_{k+P} X_k^2 + \sum_{m=1}^{P} \beta_{m+2P} X_m^3 + \epsilon.$$

The simple first-order, second-order, and third-order polynomial terms can also be replaced by corresponding terms obtained from B-splines or natural splines.

Sometimes, the effect of predictor variables on the outcome variable are not additive, and the effect of one predictor on the response variable can depend on the levels of another predictor. These non-additive effects are also called interaction effects. The interaction effects can also be a first-order interaction (interaction between two variables, e.g., $X_1 * X_2$), second-order interaction ($X_1 * X_2 * X_3$), or higher orders. It is also possible to add the interaction effects to the model. For instance, the model below also adds the first-order interactions.

$$Y = \beta_0 + \sum_{p=1}^{P} \beta_p X_p + \sum_{k=1}^{P} \beta_{k+P} X_k^2 + \sum_{m=1}^{P} \beta_{m+2P} X_m^3 + \sum_{i=1}^{P} \sum_{j=i+1}^{P} \beta_{i,j} X_i X_j + \epsilon.$$

If you are not comfortable or confused with notational representation, below is an example for different models you can write with 5 predictors (X_1, X_2, X_3).

A model with only main-effects:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon.$$

A model with polynomial terms up to the 3rd degree added:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1^2 + \beta_5 X_2^2 + \beta_6 X_3^2 + \beta_7 X_1^3 + \beta_8 X_2^3 + \beta_9 X_3^3$$

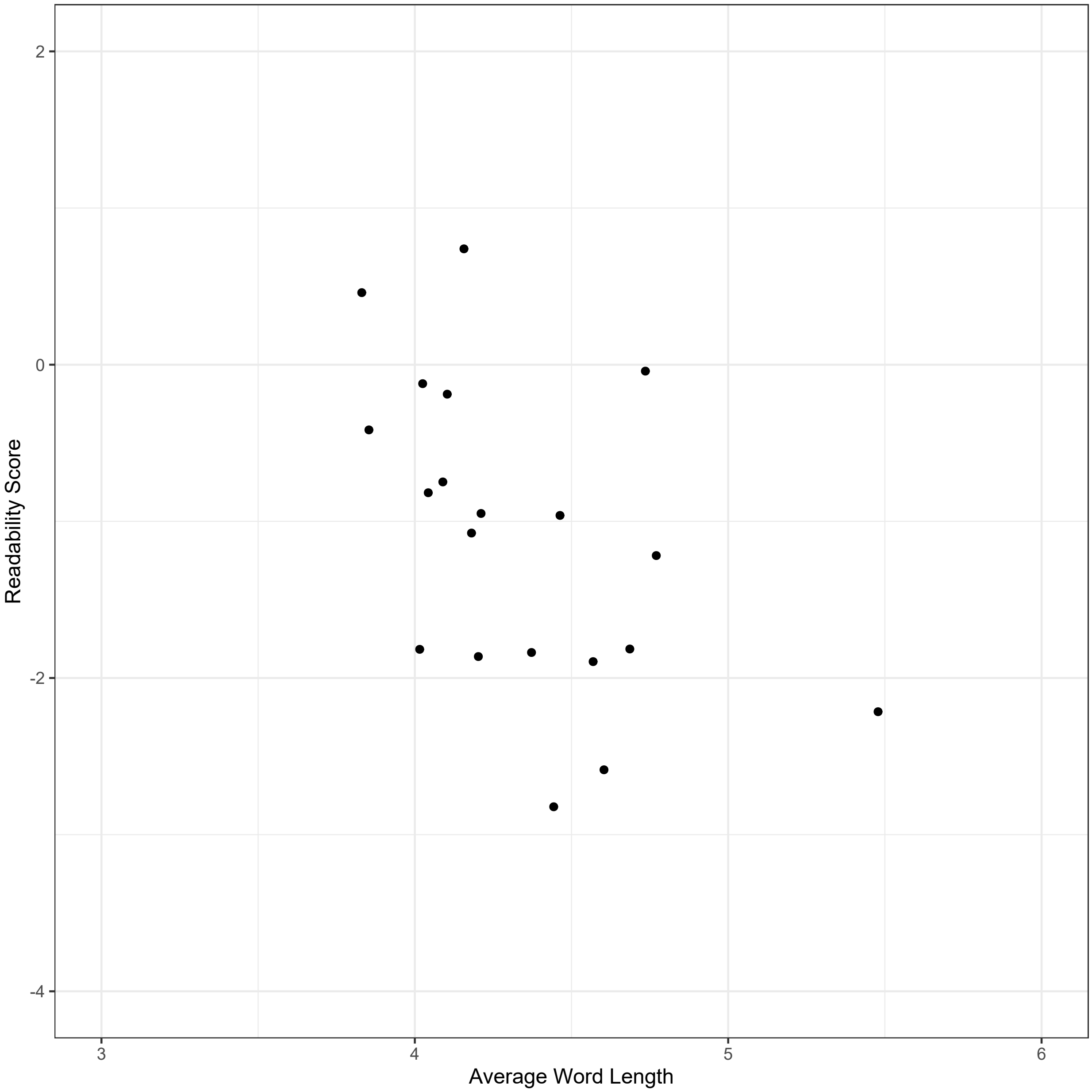
A model with both interaction terms and polynomial terms up to the 3rd degree added:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1^2 + \beta_5 X_2^2 + \beta_6 X_3^2 + \beta_7 X_1^3 + \beta_8 X_2^3 + \beta_9 X_3^3 + \beta_{1,2} X_1 X_2 + \beta_{1,3} X_1 X_3 + \beta_{2,3} X_2 X_3 + \epsilon$$

Model Estimation

Suppose that we would like to predict the target readability score for a given text from average word length in the text. Below is a scatterplot to show the relationship between these two variables for a random sample of 20 observations. There seems to be a moderate negative correlation. So, we can tell that the higher the average word length is in a given text, the lower the readability score (more difficult to read).

```
readability_sub <- read.csv('https://raw.githubusercontent.com/uo-datasci-specialization/c4-ml-fall-202
```

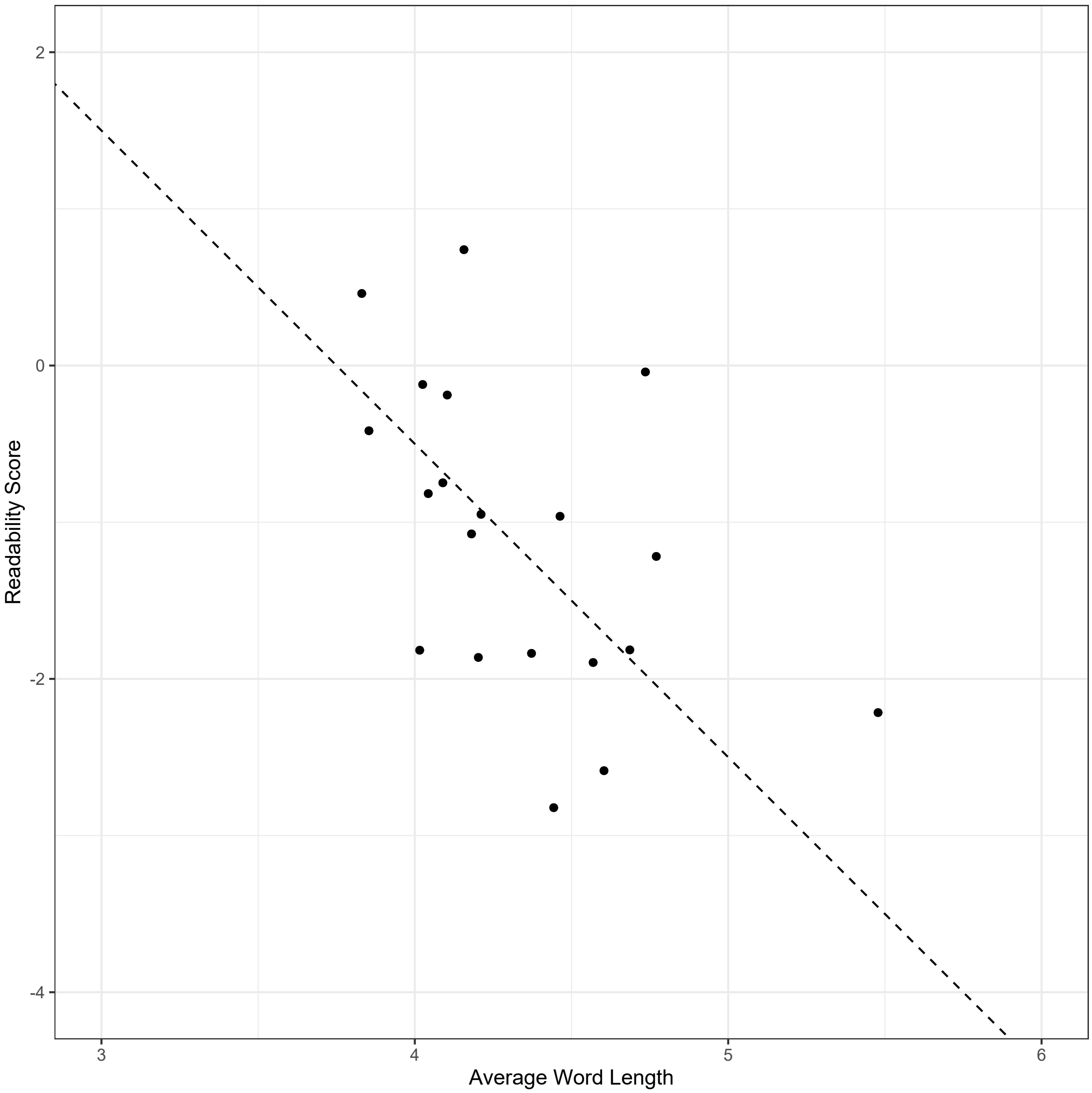


Let's consider a simple linear regression model such that the readability score is the outcome (Y) and average word length is the predictor$(X1)$. Our regression model would be

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

In this case, the set of coefficients, $\{\beta_0, \beta_1\}$, represents a linear line. We can come up with any set of $\{\beta_0, \beta_1\}$ coefficients and use it as our model. For instance, suppose I guesstimate that these coefficients are $\{\beta_0, \beta_1\}$ = {7.5,-2}. Then, my model would look like the following.

$$Y_i = 7.5 - 2X_i + \epsilon_i.$$



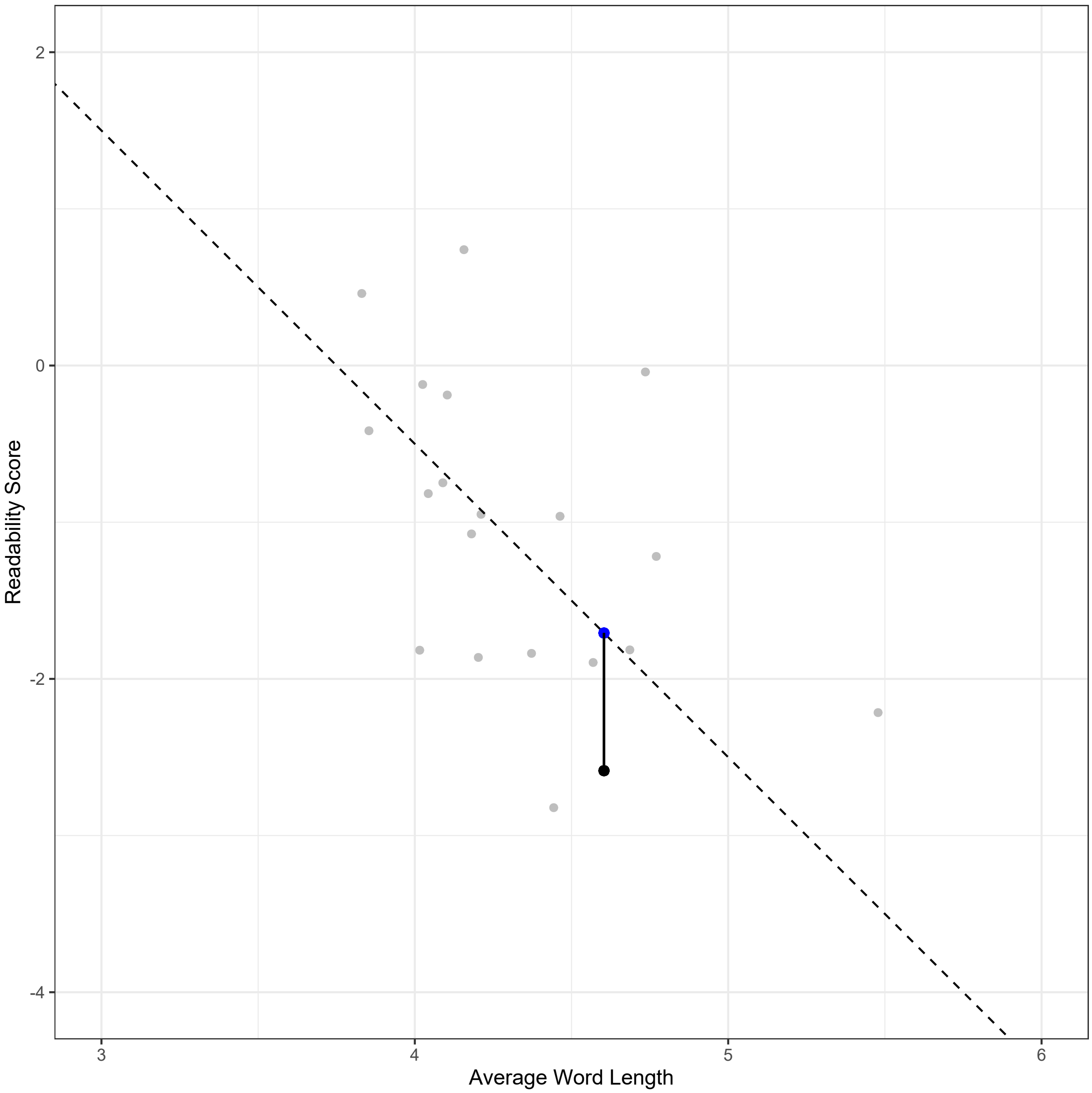
Using this model, I can predict the target readability score for all the observation in my dataset. For instance, the average word length is 4.604 for the first reading passage. Then, my prediction of readability score based on this model would be -1.708. On the other side, the observed value of the target score for this observation is -2.586. This discrepancy between the observed value and my model predicts is the model error (residual) for the first observation and captured in the ϵ term in the model.

$$Y_1 = 7.5 - 2X_1 + \epsilon_1.$$

$$\hat{Y_1} = 7.5 - 2 * 4.604 = -1.708$$

$$\epsilon_1 = -2.586 - (-1.708) = -0.878$$

We can visualize this in the plot. The black dot represents the observed data point, and the blue dot on the line represents the model prediction for a given X value. The vertical distance between these two data points is the model error for this particular observation.

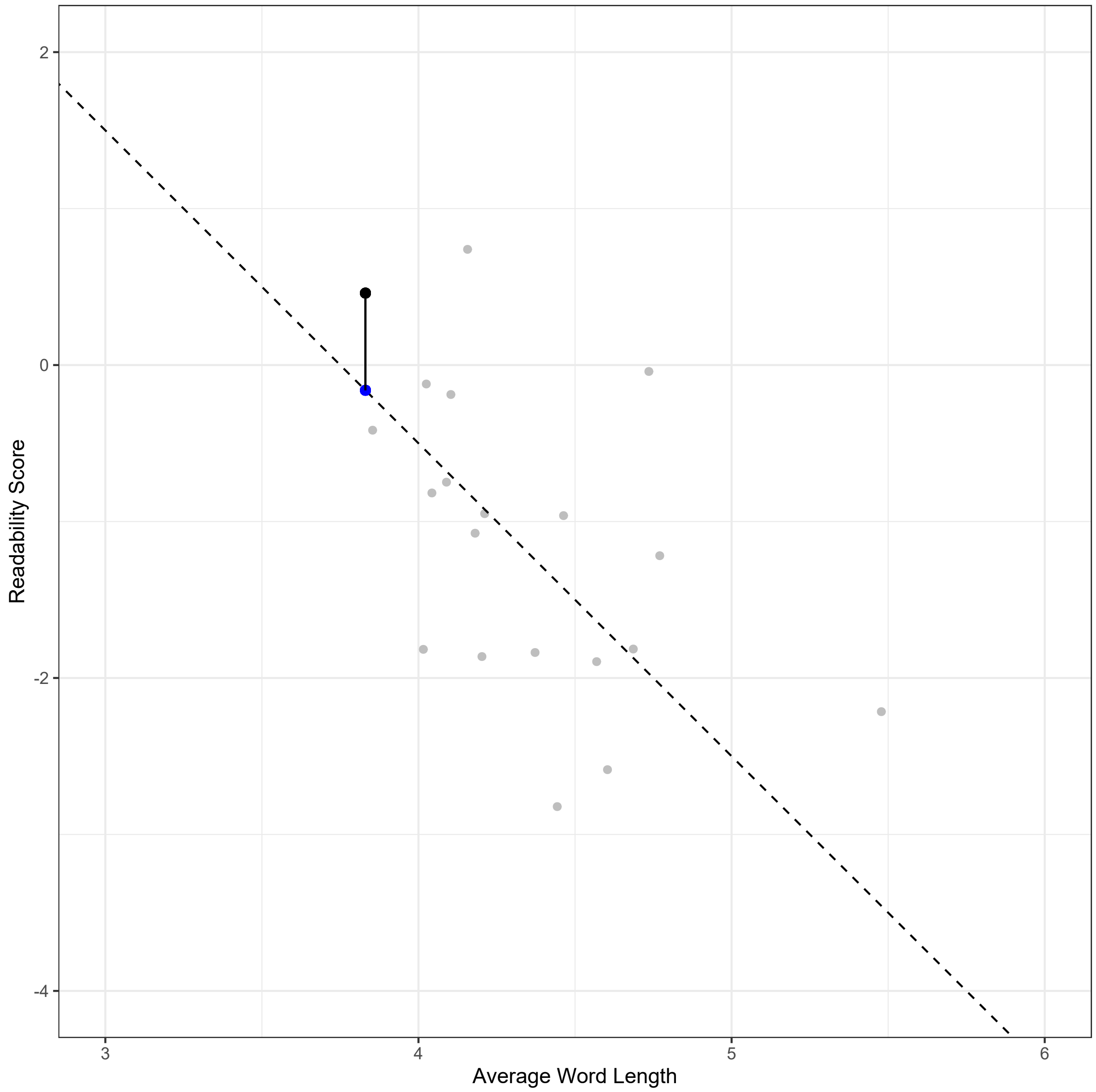


We can do the same experiment for the second observation. The average word length is 3.830 for the second reading passage. The model predicts a readability score of be -0.161. Observed value of the target score for this observation is 0.459. Therefore the model error for the second observation would be 0.62.

$$Y_2 = 7.5 - 2X_2 + \epsilon_2.$$

$$\hat{Y_2} = 7.5 - 2 * 3.830 = -0.161$$

$$\epsilon_1 = 0.459 - (-0.161) = 0.62$$



Using a similar approach, we can calculate the model error for every single observation.

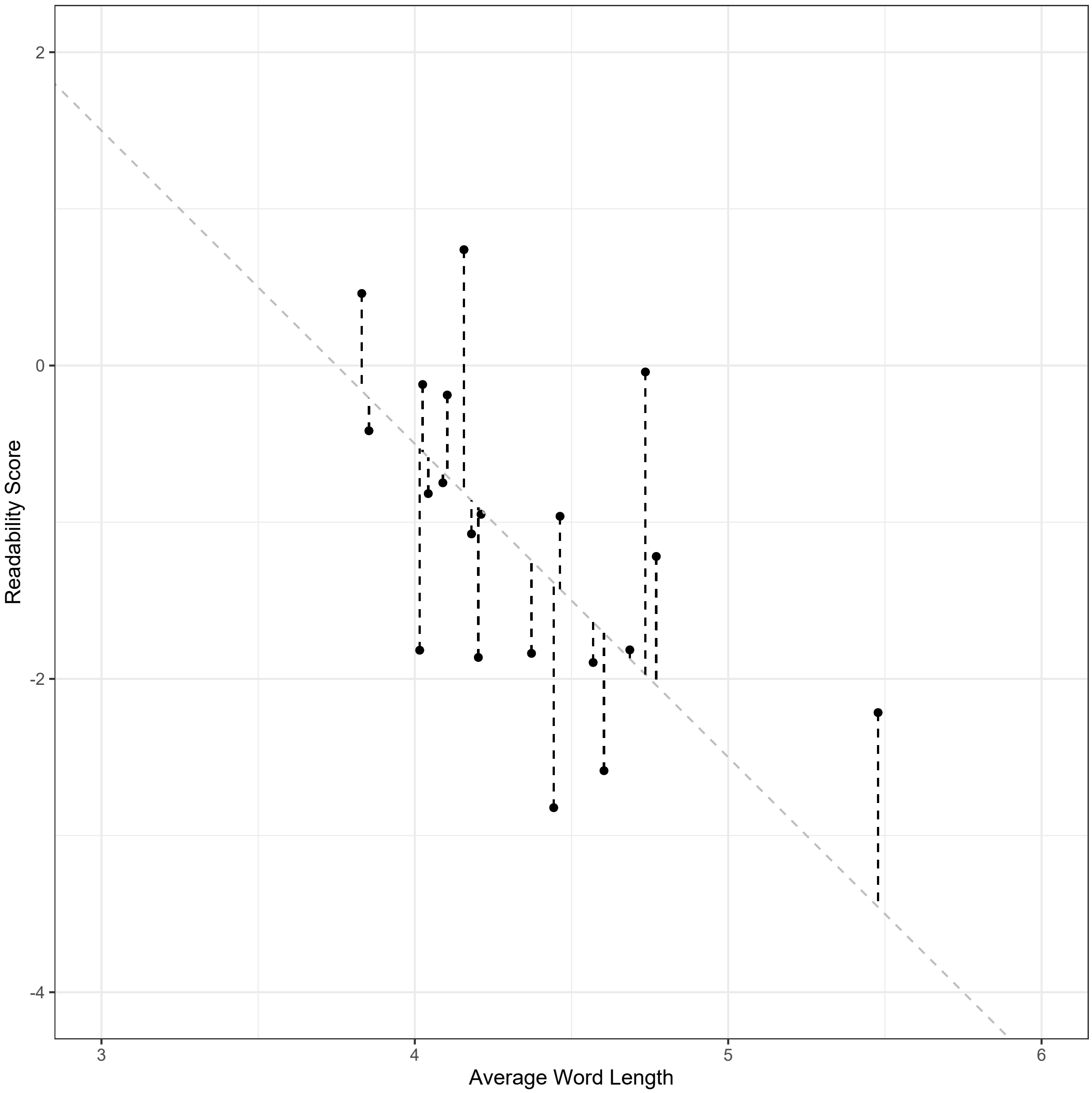
```
d <-  readability_sub[,c('mean.wl','target')]

d$predicted <- d$mean.wl*-2 + 7.5
d$error     <- d$target - d$predicted

d
```

```
    mean.wl      target  predicted       error
1  4.603659 -2.58590836 -1.7073171 -0.87859129
2  3.830688  0.45993224 -0.1613757  0.62130790
3  4.180851 -1.07470758 -0.8617021 -0.21300545
```

```
4  4.015544 -1.81700402 -0.5310881 -1.28591594
5  4.686047 -1.81491744 -1.8720930  0.05717559
6  4.211340 -0.94968236 -0.9226804 -0.02700194
7  4.025000 -0.12103065 -0.5500000  0.42896935
8  4.443182 -2.82200582 -1.3863636 -1.43564218
9  4.089385 -0.74845172 -0.6787709 -0.06968077
10 4.156757  0.73948755 -0.8135135  1.55300107
11 4.463277 -0.96218937 -1.4265537  0.46436430
12 5.478261 -2.21514888 -3.4565217  1.24137286
13 4.770492 -1.21845136 -2.0409836  0.82253224
14 4.568966 -1.89544351 -1.6379310 -0.25751247
15 4.735751 -0.04101056 -1.9715026  1.93049203
16 4.372340 -1.83716516 -1.2446809 -0.59248431
17 4.103448 -0.18818586 -0.7068966  0.51871069
18 4.042857 -0.81739314 -0.5857143 -0.23167886
19 4.202703 -1.86307557 -0.9054054 -0.95767016
20 3.853535 -0.41630158 -0.2070707 -0.20923088
```



While it is helpful to see the model error for every single observation, we will need to aggregate them in some way to form an overall measure for the total amount of error for this model. Some alternatives for aggregating these individual errors could be using

a. the sum of the residuals (SR),
b. the sum of absolute value of residuals (SAR), or
c. the sum of squared residuals (SSR)

Among these alternatives, (a) is not a useful aggregation as the positive residuals and negative residuals will cancel each other and may misrepresent the total amount of error for all observations. Both (b) and (c) are plausible alternatives and can be used. On the other hand, (b) is less desirable because the absolute values are mathematically more difficult to deal with (ask a calculus professor!). So, (c) seems to be a good way of aggregating the total amount of error, it is mathematically easy to work with. We can show (c) in a mathematical notation as the following.

$$SSR = \sum_{i=1}^{N}(Y_i - (\beta_0 + \beta_1 X_i))^2$$

$$SSR = \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

$$SSR = \sum_{i=1}^{N}\epsilon_i^2$$

For our model, the sum of squared residuals would be 15.384.

```
sum(d$error^2)
```

```
[1] 15.38364
```

Now, how do we know that the set of coefficients we guesstimate ,$\{\beta_0, \beta_1\}$ = {7.5,-2}, is a good model? Is there any other set of coefficients that would provide less error than this model? The only way of knowing this is to try a bunch of different models and see if we can find a better one that gives us better predictions (smaller residuals). But, there is literally infinite number of set of $\{\beta_0, \beta_1\}$ coefficients, so which ones we should try?

Matrix Approach to Linear Regression

Model Interpretation

Model Evaluation

Linear Regression with Regularization

Ridge Penalty

Lasso Penalty

Elastic Net

Wrapping up: Predicting Readability Scores