Logistic Regression

Machine Learning - Section 3.1.4

Marc Omar Haddad

Published: 22 February, 2020

Updated: 23 February, 2020

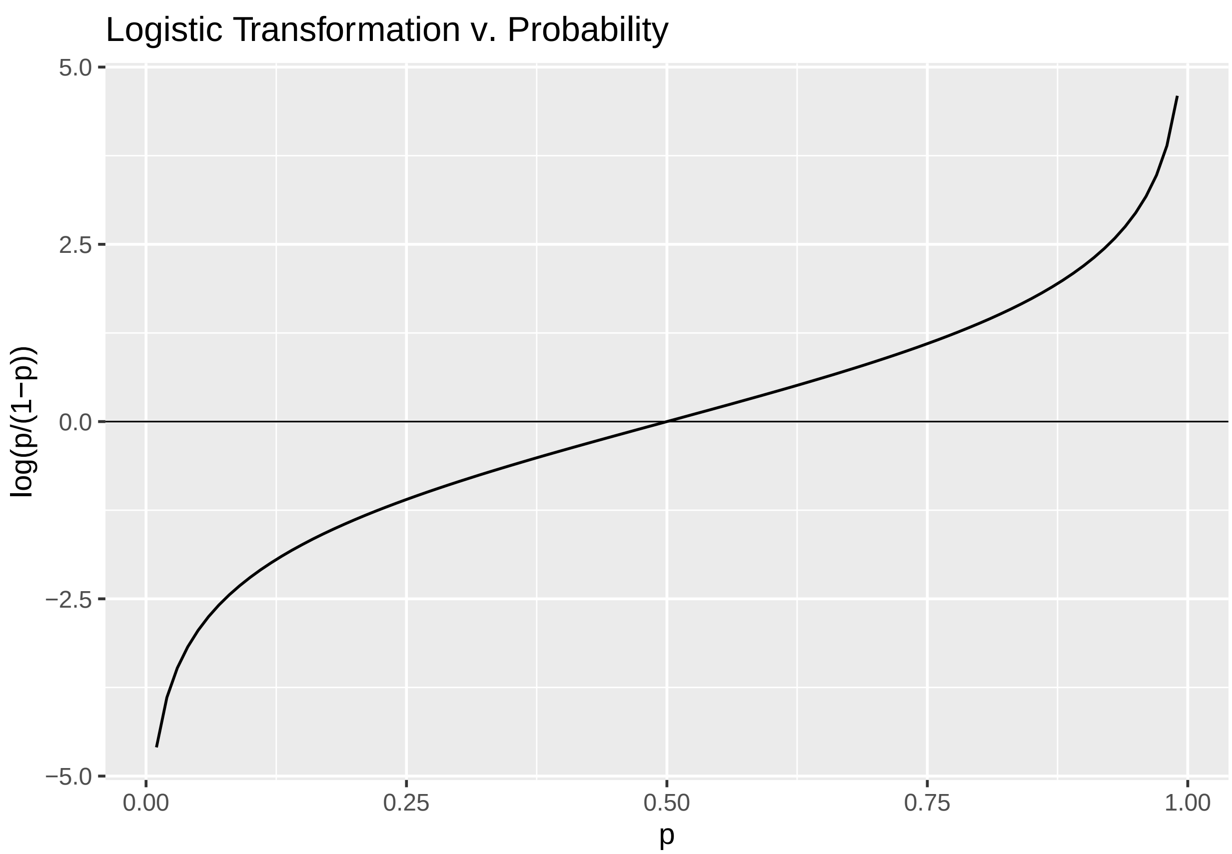
Logistic Regression is an extension of linear regression that **ensures our estimated Conditional Probabilities are between** 0 **and** 1.

Logistic Regression requires a **logistic transformation** of our outcomes:

$$g(p) = \log \frac{p}{1-p}$$

Logistic transformation converts *Probabilities* to *Log Odds*. **Log Odds** tell us how much more likely an event will happen versus *not* happen. For example: If $p = 0.5$ the odds are $1:1$.

We use logistic transformation because it transforms probabilities to be symmetric around 0.



To fit this model we use the **Maximum Likelihood Estimate (MLE)**. The `R` function `glm()` (which stands for "Generalized Linear Models") allows us to fit a logistic regression model.

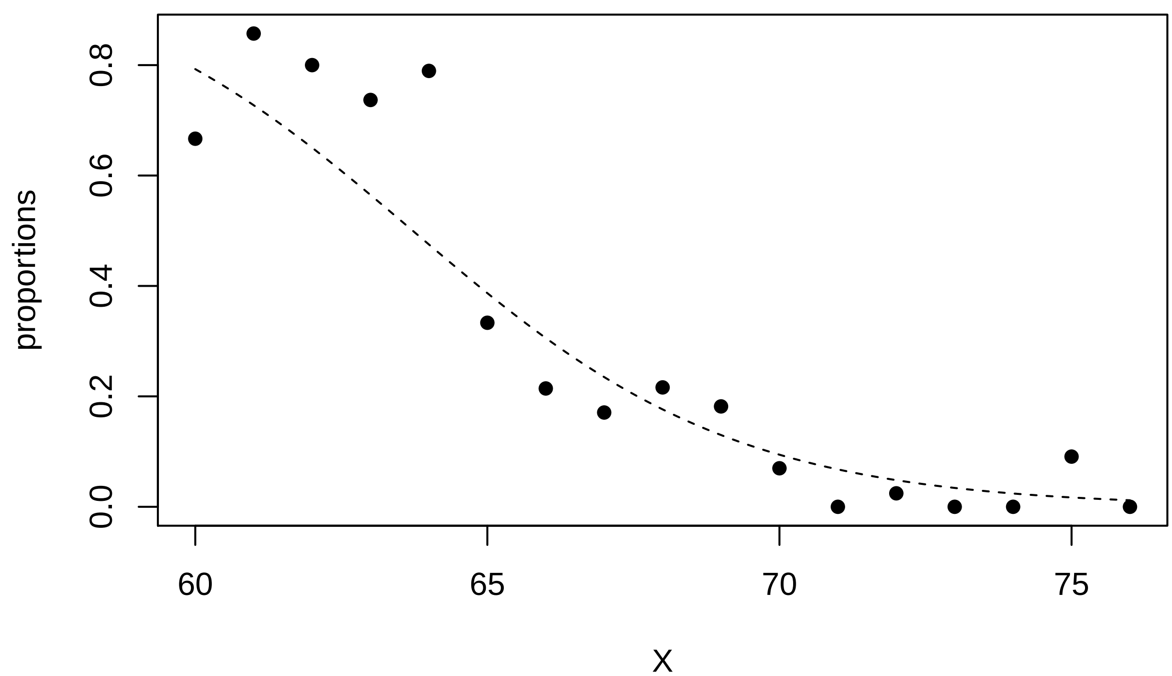
```
# The following fits a logistic regression model to our data.
glm_fit = train_set %>%
  mutate(y = as.numeric(sex == "Female")) %>%
  glm(y ~ height, data = ., family = "binomial") # Note: 'family' is required for glm().
```

Since `glm()` is more generalized than `lm()`, we are required to specify our desired model through the `family` parameter.

Similarly to our linear regression model we can obtain predictions with the `predict()` function.

```
p_hat_logit = predict(glm_fit, newdata = test_set, type = "response")
# 'type' param must be set to 'response' if we want conditional probs.
```

We can see how well our new model `p_hat_logit` fits by plotting it against our actual Conditional Probabilities (i.e. the proportions of `female` to `male` in our data set for each rounded height).



As we can clearly see, this is a much better fit than our previous linear curve (see section 3.1.3 notes).

Now let's assess our trained algorithm by comparing the predictions to our `test_set`.

```
y_hat_logit = ifelse(p_hat_logit > 0.5, "Female", "Male") %>% factor()

confusionMatrix(y_hat_logit, test_set$sex)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Female Male
##     Female     31   19
##     Male       87  389
##
##                Accuracy : 0.7985
##                  95% CI : (0.7616, 0.832)
##     No Information Rate : 0.7757
##     P-Value [Acc > NIR] : 0.1138
##
##                   Kappa : 0.2718
##
##  Mcnemar's Test P-Value : 7.635e-11
##
##             Sensitivity : 0.26271
##             Specificity : 0.95343
##          Pos Pred Value : 0.62000
##          Neg Pred Value : 0.81723
##              Prevalence : 0.22433
##          Detection Rate : 0.05894
##    Detection Prevalence : 0.09506
##       Balanced Accuracy : 0.60807
##
##        'Positive' Class : Female
##
```

Though our new model accuracy (0.7985) is slightly higher than our previously obtained accuracy of 0.7852, our new model does not improve much upon our previous linear model. This is due to the fact that our **Decision Rule**— predicting `female` if our estimated conditional probability is greater than 0.5— results in similar prediction regions. This is illustrated in the graph below.

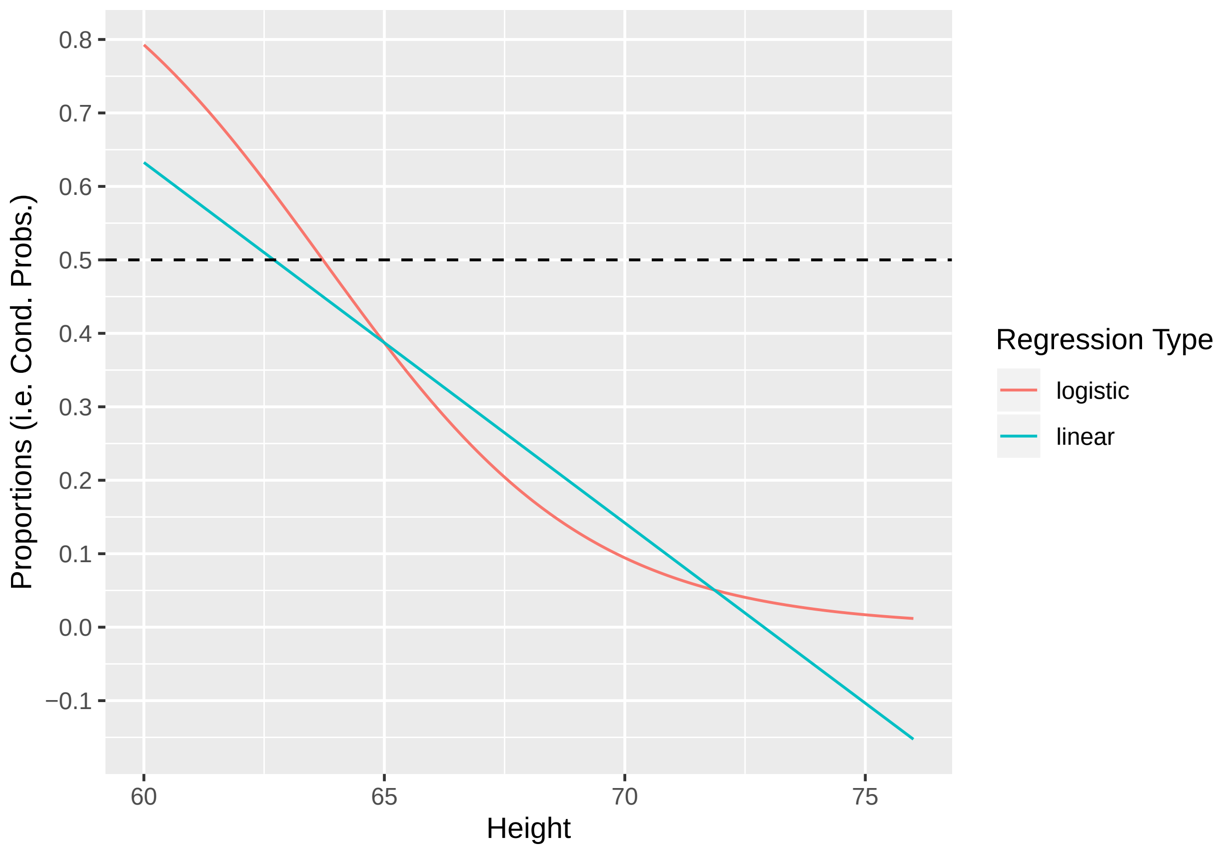
```r
height_seq2 = seq(60, 76, 0.01) # Define new height_seq w/ small intervals.

# Re-predict cond. probs. for each reg. type w/ new height_seq.
p_hat_logit2 = predict(glm_fit2, list(height = height_seq2), type = "response")
p_hat2 = predict(lm_fit, list(height = height_seq2))

df_wide = data.frame(height = height_seq2, logistic = p_hat_logit2, linear = p_hat2)

# Convert from wide-form to long-form data for ggplot.
df_long = gather(df_wide, reg_type, reg_val, logistic:linear, factor_key = TRUE)

df_long %>% ggplot(aes(height, reg_val, color = reg_type)) +
  geom_line() +
  geom_hline(yintercept = 0.5, lty = 2) +
  labs(y = "Proportions (i.e. Cond. Probs.)", x = "Height", color = "Regression Type") +
  scale_y_continuous(breaks = seq(-0.2, 0.8, 0.1))
```



Both types of regression provide an **estimate for the Conditional Expectation**; in binary data, this conditional expectation is **equivalent to Conditional Probability**.

It must be noted, however, that both logistic and linear regression are not the best approaches in more complex Machine Learning algorithms due to their rigidity. More appropriate techniques (to be learned in later sections) allow for more flexibility and are primarily **approaches to *estimating* Conditional Probabilities and/or Conditional Expectations**.