Reading: Interpret logistic regression models

Interpreting a logistic regression model involves examining coefficients and computing metrics. After you fit your logistic regression model to training data, you can access the coefficient estimates from the model using code in Python. You can then use those values to understand how the model makes predictions. This reading will show you an example of how to interpret coefficients from a logistic regression model, as well as things to consider when choosing metrics for model evaluation.

Coefficients from the model

To understand how a logistic regression model works, it is important to start with the equation that describes the relationship between the variables. That equation is also called the logit function.

The logit function

When the logit function is written in terms of the independent variables, it conveys the following: there is a linear relationship between each independent variable, X, and the logit of the probability that the dependent variable, Y, equals 1. The logit of that probability is the logarithm of the odds of that probability.

The equation for the logit function in binomial logistic regression is shown below. This involves the probability that Y equals 1, because 1 is the typical outcome of interest in binary classification, where the possible values of Y are 1 and 0.

$$logit(p) = log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 \quad where\ p = P(Y = 1)$$

Interpret coefficients

Imagine you have built a binomial logistic regression model for predicting emails as spam or non-spam. The dependent variable, Y, is whether an email is spam (1) or non-spam (0). The independent variable, X_1, is the message length. Assume that **clf** is the classifier you fitted to training data.

You can use the following code to access the coefficient β_1 estimated by the model:

```
clf.coef_
```

If the estimated β_1 is 0.186, for example, that means a one-unit increase in message length is associated with a 0.186 increase in the log odds of p. To interpret change in odds of Y as a percentage, you can exponentiate β_1, as follows.

$$e^{\beta_1}=e^{0.186}\approx 1.204$$

So, for every one-unit increase in message length, you can expect that the odds the email is spam increases by 1.204, or 20.4%.

Things to consider when choosing metrics

The next important step after examining the coefficients from a logistic regression model is evaluating the model through metrics. The most commonly used metrics include precision, recall, and accuracy. The following sections describe things to keep in mind when choosing between these.

When to use precision

Using precision as an evaluation metric is especially helpful in contexts where the cost of a false positive is quite high and much higher than the cost of a false negative. For example, in the context of email spam detection, a false positive (predicting a non-spam email as spam) would be more costly than a false negative (predicting a spam email as non-spam). A non-spam email that is misclassified could contain important information, such as project status updates from a vendor to a client or assignment deadline announcements from an instructor to a class of students.

When to use recall

Using recall as an evaluation metric is especially helpful in contexts where the cost of a false negative is quite high and much higher than the cost of a false positive. For example, in the context of fraud detection among credit card transactions, a false negative (predicting a fraudulent credit card charge as non-fraudulent) would be more costly than a false positive (predicting a non-fraudulent credit card charge as fraudulent). A fraudulent credit card charge that is misclassified could lead to the customer losing money, undetected.

When to use accuracy

It is helpful to use accuracy as an evaluation metric when you specifically want to know how much of the data at hand has been correctly categorized by the classifier. Another scenario to consider: accuracy is an appropriate metric to use when the data is balanced, in other words, when the data has a roughly equal number of positive examples and negative examples. Otherwise, accuracy can be biased. For example, imagine that 95% of a dataset contains positive examples, and the remaining 5% contains negative examples. Then you train a logistic regression classifier on this data and use this classifier predict on this data. If you get an accuracy of 95%, that does not necessarily indicate that this classifier is effective. Since there is a much larger proportion of positive examples than negative examples, the classifier may be biased towards the majority class (positive) and thus the accuracy metric in this context may not be meaningful. When the data you are working with is imbalanced, consider either transforming it to be balanced or using a different evaluation metric other than accuracy.

Key takeaways

- Examine the beta coefficients from a model to understand how the model predicts the dependent variable.
- When determining which metrics are meaningful for evaluating a logistic regression classifier, consider the context of the data involved, how the predictions will be used, and how impactful False Positives versus False Negatives are in that context.

Resources for more information

- LogisticRegression: Documentation for implementing Logistic Regression models using **sklearn** and accessing intercept and coefficients from a model

---