Logistic regression: Takeaways

Syntax

- Define the logistic function:

```
def logistic(x):
    """
    np.exp(x) raises x to the exponential power e^x. e ~= 2.71828
    """
return np.exp(x)  / (1 + np.exp(x))
```

- Instantiate a logistic regression model:

```
from sklearn.linear_model import LogisticRegression
linear_model = LogisticRegression()
```

- Train a logistic regression model:

```
logistic_model.fit(admissions[["gpa"]], admissions["admit"])
```

- Return predicted probabilities for a column:

```
pred_probs = logistic_model.predict_proba(admission[["gpa"]])
```

Concepts

- In classification, our target column has a finite set of possible values, which represent different categories for a row.
- In binary classification, there are only two options for values:
  - `0` for the False condition.
  - `1` for the True condition.
- Categorical values are used to represent different options or categories. Classification focuses on estimating the relationship between the independent variables and the dependent categorical variable.
- One technique of classification is called **logistic regression**. While a linear regression model outputs a real number as the label, a logistic regression model outputs a probability value.
- The logistic function is a version of the linear function that is adapted for classification. Mathematically, the logistic function is represented as the following:

  $$\sigma(t) = \frac{e^t}{1+e^t}$$

  where e^t is the exponential transformation to transform all values to be positive, and $\frac{t}{1+t}$ is the normalization transformation to transform all values between `0` and `1` .

Resources

- [Documentation for the LogisticRegression class]
- [Documentation for the predict_proba method]