Train In Data

Calibrating a Classifier

Imbalance data techniques and probabilities

Over-sampling, **under-sampling** and **cost-sensitive learning distort** the relationship between the returned probabilities and the fraction of positive observations.

To convey likelihood, we need calibrated probabilities.

Calibrating a Classifier

Mapping model predictions to posterior probabilities:

$$f_{calib}(s(x)) \approx p(y)$$

- ***fcalib*** is a calibration function
- **s(x)** is the score returned by a model: probability or the decision function (eg SVMs)
- **p(y)** is the posterior probability

Calibrating a Classifier

Mapping model predictions to posterior probabilities:

$$f_{calib}(s(x)) \approx p(y)$$

- Platt Scaling
- Isotonic Regression

Which data should we use?

- To get good posterior probabilities we should not calibrate our classifiers on the train set.

- Perform the calibration on the test set

  - When possible keep more than 1 hold out sample.

- If little data, then we can do cross-validation.

Platt Scaling

Mapping model predictions to posterior probabilities:

$$f_{calib}(s(x)) \approx p(y)$$

- Logistic Regression to regress the classifier scores to real likelihood (fraction of positives)

$$f_{platt} = \frac{1}{1 + exp(-ws(x) - b)}$$

Isotonic Regression

Mapping model predictions to posterior probabilities:

$$f_{calib}(s(x)) \approx p(y)$$

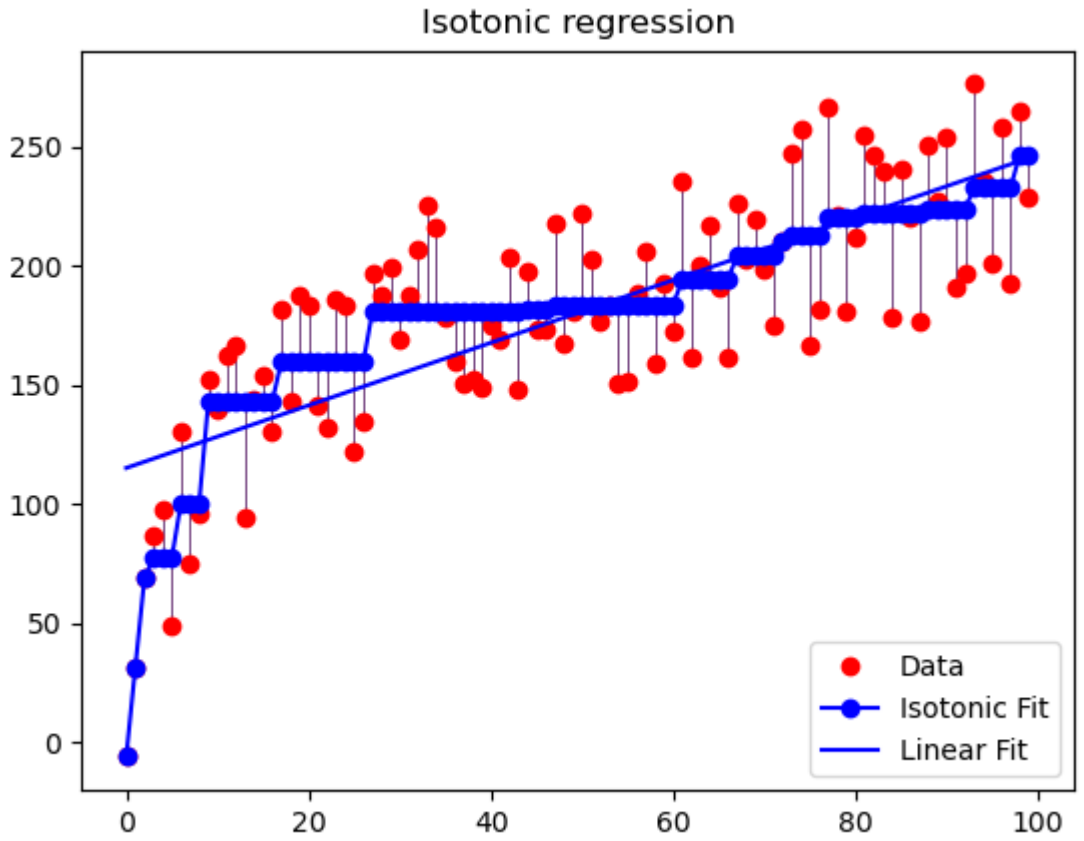
- **fcalib** can be any function
- Only restriction is that it is monotonic

Isotonic Regression

Learns arbitrary monotonically increasing step functions.

Groups data into constant parts, steps

Minimises the estimation of y respect of y



https://scikit-learn.org/stable/modules/isotonic.html#isotonic

Isotonic Regression

In the sklearn implementation ➔ scipy.interpolate.interp1d

https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html

Interpolation is a convenient method to create a function based on fixed data points, which can be evaluated anywhere within the domain defined by the given data using linear interpolation

Calibrating Probability with sklearn

```
# Isotonic calibration

clf_isotonic = CalibratedClassifierCV(rf, cv=5, method='isotonic')
clf_isotonic.fit(X_test, y_test)
prob_isotonic = clf_isotonic.predict_proba(X_test)[:, 1]

# Gaussian Naive-Bayes with sigmoid calibration
clf_sigmoid = CalibratedClassifierCV(rf, cv=5, method='sigmoid')
clf_sigmoid.fit(X_test, y_test)
prob_sigmoid = clf_sigmoid.predict_proba(X_test)[:, 1]
```

Train In Data

THANK YOU

www.trainindata.com