Risk Modeling and Classification

EN5422/EV4238 | Fall 2023

w05_classification_2.pdf

(Week 5 – 2/2)

Contents

1 Evaluating Binary Risk Models

1.1 Common Binary Loss Function

- Suppose we are going to predict a binary outcome $Y \in \{0,1\}$ with $\hat{p}(x) \in \{0,1\}$.
    - Call $\hat{p}(x)$ *the risk score*
- **Brier Score / Squared Error**

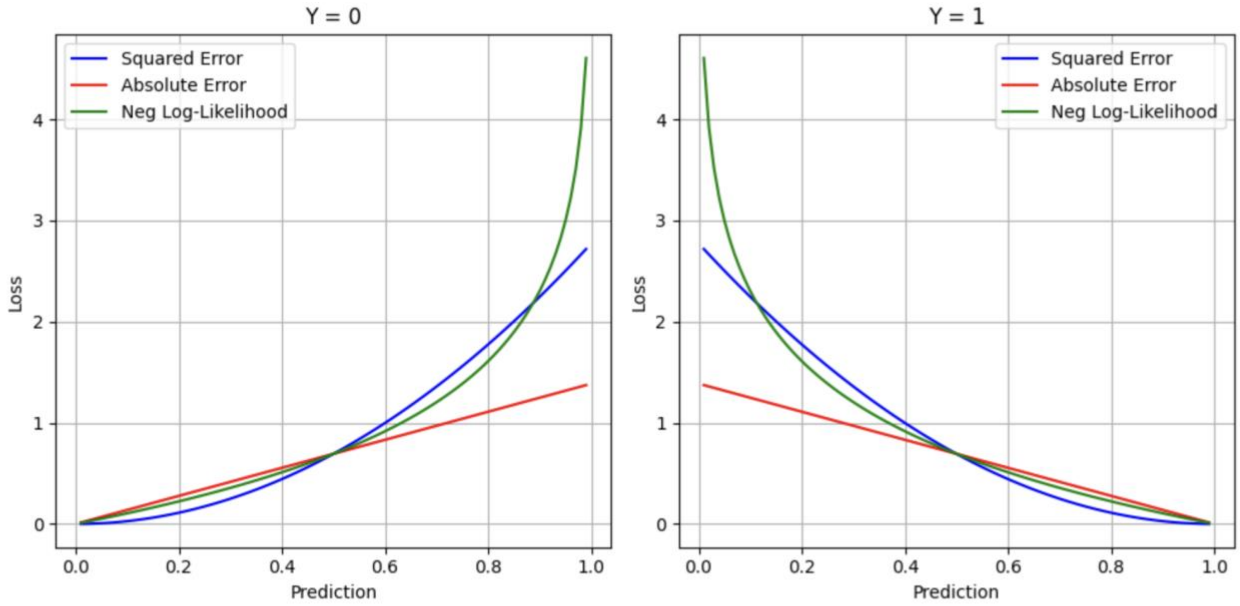
$$L(y,\hat{p}) = (y-\hat{p})^2 = \begin{cases} (1-\hat{p})^2 & y=1 \\ \hat{p}^2 & y=0 \end{cases}$$

- **Absolute Error**

$$L(y,\hat{p}) = |y-\hat{p}| = \begin{cases} 1-\hat{p} & y=1 \\ \hat{p} & y=0 \end{cases}$$

- **Bernoulli negative log-likelihood (Log-Loss)**
    - This is the loss function for Logistic Regression

$$L(y,\hat{p}) = -\{y\log\hat{p} + (1-y)\log(1-\hat{p})$$
$$= \begin{cases} -\log\hat{p} & y=1 \\ -\log(1-\hat{p}) & y=0 \end{cases}$$



1.2 Model Comparison

```
# Evaluation function
def evaluate(p_hat, y):
    return {
        'mn_log_loss': -log_loss(y, p_hat),
        'mse': mean_squared_error(y, p_hat),
        'mae': mean_absolute_error(y, p_hat)
    }

# Train/test split
```

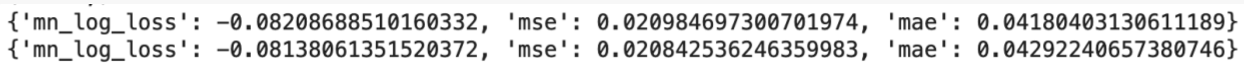
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=2000,
random_state=321)

# Fit logistic regression on training data
lm = LogisticRegression(max_iter=5000)
lm.fit(X_train, y_train)
p_hat_lm = lm.predict_proba(X_test)[:, 1]

print(evaluate(p_hat_lm, y_test))

# Fit lasso logistic regression with cross-validation
lasso = LogisticRegressionCV(penalty='l1', solver='liblinear', cv=10,
max_iter=5000)
lasso.fit(X_train, y_train)
p_hat_lasso = lasso.predict_proba(X_test)[:, 1]

print(evaluate(p_hat_lasso, y_test))
```



```
{'mn_log_loss': -0.08208688510160332, 'mse': 0.020984697300701974, 'mae': 0.04180403130611189}
{'mn_log_loss': -0.08138061351520372, 'mse': 0.020842536246359983, 'mae': 0.04292240657380746}
```

1.3 Calibration

A risk model is said to be *calibrated* if the predicted probabilities are equal to the true risk (probabilities).

$$\Pr(Y = 1 \mid \hat{p} = p) = p \qquad \text{for all } p$$

Note

Calibration in the context of a risk model refers to the idea that the predicted probabilities from a model should match the actual outcomes in the real world, especially over a large number of predictions.

Example:

Imagine you have a weather forecasting model that predicts the probability of rain tomorrow.

1. On 100 days, your model predicted a 70% chance of rain the next day.
2. If the model is well-calibrated, it would mean that out of those 100 days, it actually rained on approximately 70 of them.

So, the model's predicted probability (70% chance of rain) should be close to the true frequency of the event happening (rain on 70 out of 100 days).

In summary, a calibrated model provides trustworthy probabilities. If it says there's an X% chance of an event happening, then over many occurrences, that event should happen approximately X% of the time.

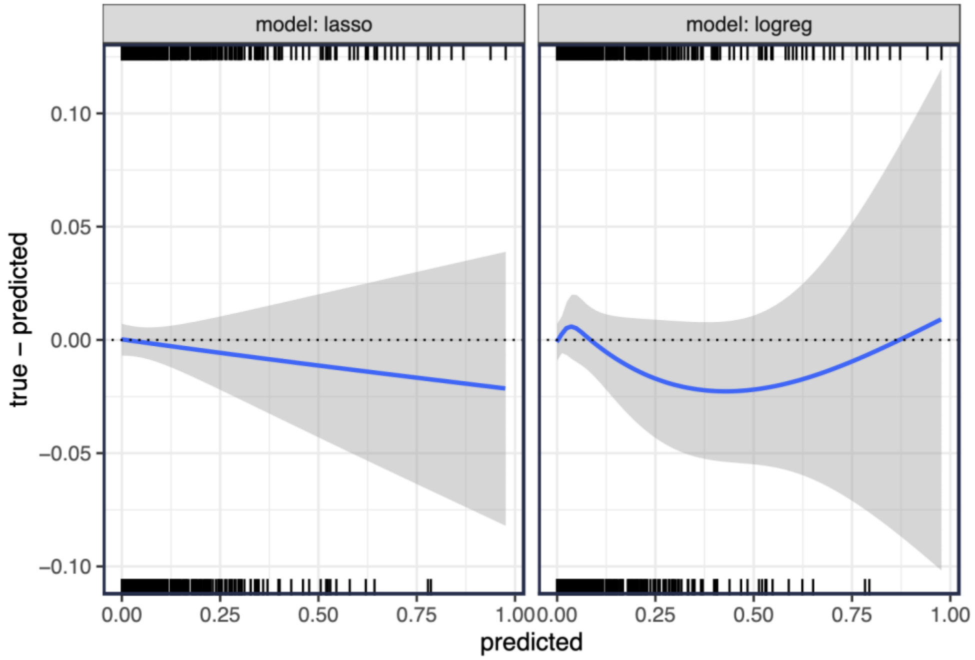
$\Pr(Y = 1 \mid \hat{p} = p) = p$ is the probability that the outcome Y is 1 (or the event happens) given that the predicted probability is $\hat{p}$. p is a specific probability value, such as 0.70 for 70%.

Interpretation: For a model to be calibrated, the conditional probability of the event happening (e.g., it raining) given a predicted probability should be equal to that predicted probability, across all possible predicted probabilities.

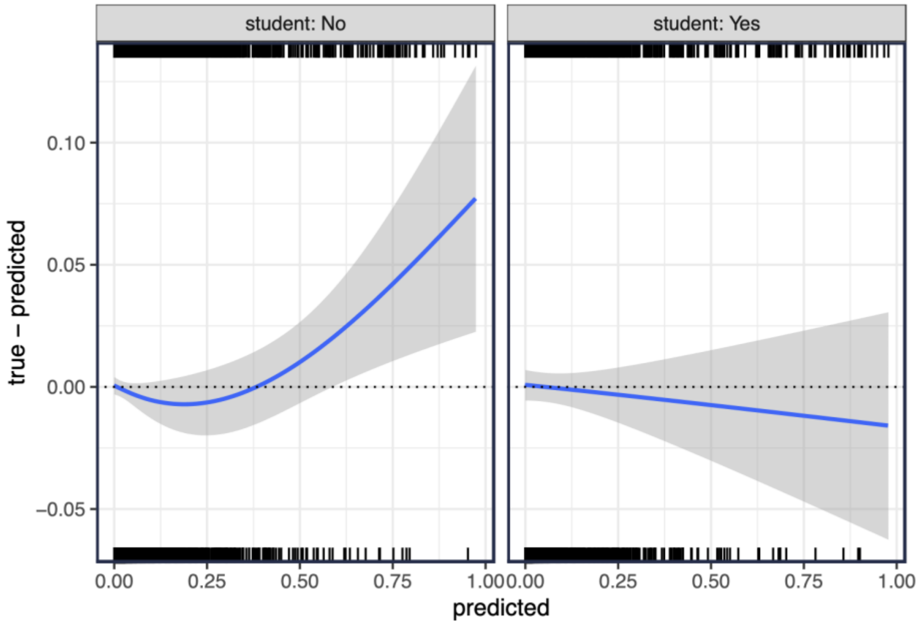
Using our earlier example:

If our weather model predicts a 70% chance of rain tomorrow ($\hat{p} = 0.70$), then the true probability of it raining tomorrow, based on many such predictions, should also be 70% ($\Pr(Y = 1 \mid \hat{p} = 0.7) = 0.70$).

In other words, whenever the model says there's a "p" chance of something happening, the actual chance of that event occurring should be "p". If this holds true for all values of "p", then the model is calibrated.



Calibration plots can be used to measure drift, fairness, and model/algorithmic bias. Consider comparing the predictive performance of our models for Students and Non-Students.



1.3.1 Estimating Calibration

To measure mis-calibration, we can treat the predictions as features and use the predictions as an offset. E.g., to check for linear deviation

$$\operatorname{logit} p(x) = \beta_0 + \beta_1 \hat{p}(x) + \operatorname{logit} \hat{p}(x)$$

fit on a hold-out set, and check how far β_0 and β_1 are from 0.

We will revisit calibration when we cover Support Vector Machines (SVM) to convert a generic score to a probability.

Some additional resources:

- Platt Scaling
- Isotonic Regression
- Calibration: the Achilles heel of predictive analytics
  - Article on clinical decision-making
- The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning

1.4 Area Under the ROC curve (AUC or AUROC or C-Statistic)

The AUROC of a risk model is: the probability that the model will rank a randomly chosen positive example ($Y = 1$) higher than a randomly chosen negative example ($Y = 0$) i.e.,

$$\mathrm{AUROC} = \Pr\big(\hat{p}(x_1) > \hat{p}(x_0)\big)$$

• where x_k is a randomly chosen example from class $Y = k$.

To estimate the AUROC you will fit a model to training data and make predictions on hold-out (test) data with known labels. Hopefully the model will assign large probabilities to the outcome of interest ($Y = 1$) and low probabilities to the other class. Then compare the probabilities between all pairs of observations where one comes from the $Y = 1$ set and the other from the $Y = 0$ set. The AUROC is the proportion of the pairs where the estimated probability for the outcome of interest is larger than the probability for the other outcome. The extra term is to handle ties in predicted probability.

$$\widehat{\mathrm{AUROC}} = \frac{1}{n_1 n_0} \sum_{i:y_i=1} \sum_{j:y_j=0} \left(\mathbb{I}\big(\hat{p}_i > \hat{p}_j\big) + \frac{1}{2}\mathbb{I}\big(\hat{p}_i = \hat{p}_j\big) \right)$$

The AUROC assesses the discrimination ability of the model. It gives a different assessment on model performance from calibration. Notice that the AUROC is the same for any monotonic transformation of the estimated probabilities. E.g., we can use $\hat{p}$ or $\log \hat{p}$ or $\mathrm{logit}(\hat{p})$ or $\frac{\hat{p}}{10}$ and still get the same AUROC. But calibration assesses how closely the estimated probabilities match the actual probabilities as well s helping to identify the regions in feature space where the predictions are poor.

- A helpful discussion on AUROC: https://stats.stackexchange.com/questions/145566/how-to-calculate-area-under-the-curve-auc-or-the-c-statistic-by-hand
- We will say more about ROC curves later in the notes.