CLASSIFICATION

FA590 STATISTICAL LEARNING IN FINANCE

Dr. Zonghao Yang

Stevens Institute of Technology

Previous Lecture

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

- Linear regression: Parameter estimation, hypothesis testing, model interpretation, and prediction
- Model performance metrics for regression: Mean squared error and (adjusted) R-squared
- Regularization: Lasso (L1 penalty), ridge (L2 penalty), elastic net (L1 and L2 penalties)
- Financial applications
  - Time-series, cross-section, and panel regressions
  - Model trend and seasonality

Learning Objectives

- Understand and apply logistic regression for binary classification
- Generalize logistic regression to multiclass classification problems
- Evaluate classification model performance using metrics such as confusion matrix, precision, recall, F1 score, and ROC-AUC
- Articulate and compare techniques for handling imbalanced labels in classification tasks

Part I

Logistic Regression

Motivation Examples

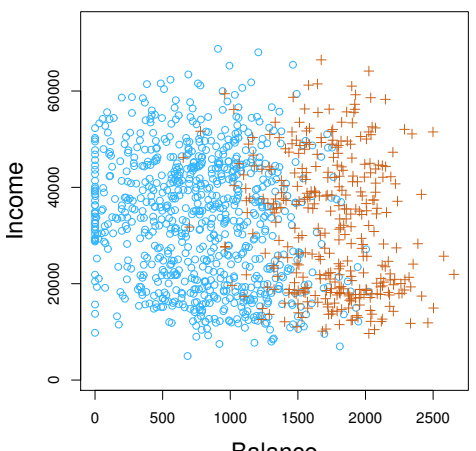
- Spam: An email is spam or not spam
- Fraud detection: A transaction is fraudulent or not
- Credit risk assessment: If banks issue a loan, they need to assess the credit risk of the borrower, i.e., whether the borrower will default on the loan
- Financial news classification: Market news, economic policy, corporate earnings, and mergers & acquisitions (M&A)

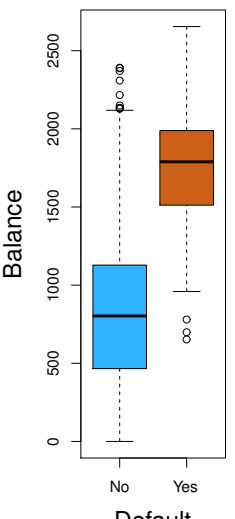
- The prediction target is a qualitative variable

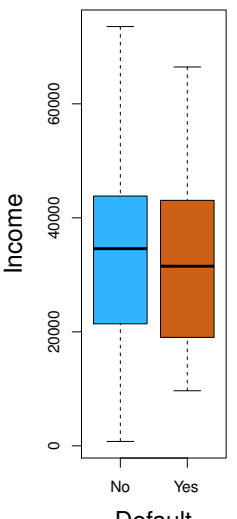
Classification Problems

- Qualitative variables take values in an unordered set $\mathcal{C}$
  - Binary class: Spam or not, fraud or not, default or not
  - Multiclass: News themes
- Given a feature vector X and a qualitative response Y
- The classification task is to build a function $f(X)$ that takes the feature vector X as input and predicts its value for Y; i.e., $y = f(X) \in \mathcal{C}$
- Often we are more interested in estimating the probability that X belongs to each category in $\mathcal{C}$
  - The probability that a borrower will default, rather than whether a loan is in default or not

Example: Default





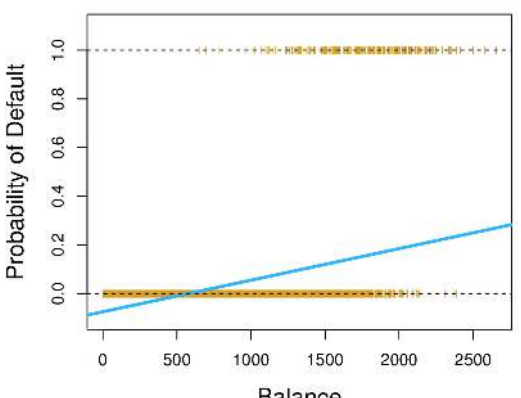


Linear Regression for binary classification?

- Suppose the default classification task

$$Y = \begin{cases} 1 & \text{if default} \\ 0 & \text{if not default} \end{cases}$$

- The regression approach: Regress Y on X, and classify Y as 1 if $\hat{Y} > 0.5$
- The predicted values include probabilities less than zero or bigger than one



Linear Regression for Multiclass Classification?

- Suppose a response variable with more than two possible values
  - Image classification

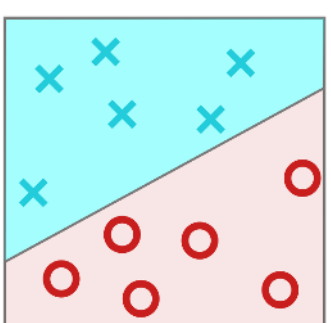
$$Y = \begin{cases} 1 & \text{dogs} \\ 2 & \text{cats} \\ 3 & \text{people} \end{cases}$$

- The encoding of qualitative responses implies an ordering on the outcomes
  - The difference between dogs and cats is the same as between cats and people
  - Not natural as a regression target
- Linear regression cannot accommodate this scenario
- **We need a new model for classification**

Classification vs. Regression

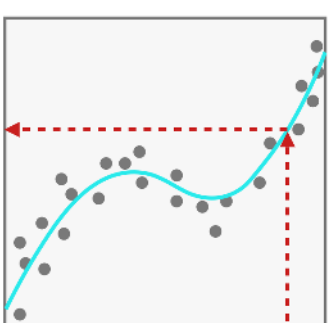
- Most statistical learning applications are either classifications or regressions
- There is generally no ordering relationship between the classes (nominal data), but regression targets can be ordered
- In other words: Prediction of nominal data requires classifiers, rather than regressors

Classification Groups observations into "classes"



Here, the line classifies the observations into X's and O's

Regression predicts a numeric value



Here, the fitted line provides a predicted output, if we give it an input

Logistic Regression

- Let $Y \in \{0, 1\}$ and $p(X) = Pr(Y = 1|X)$; The odds of the event $Y = 1$ is:

$$\text{odds}(p(X)) := \frac{p(X)}{1 - p(X)}$$

- Logistic regression models the log-odds or logit with a linear function:

$$\log(\text{odds}) = \log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- We can then solve for $p(X)$ algebraically:

$$p(X) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Translate the probability prediction to a binary predictor:

$$f(X) = \text{sign}(2p(X) - 1)$$

Logistic Regression vs. Linear Regression

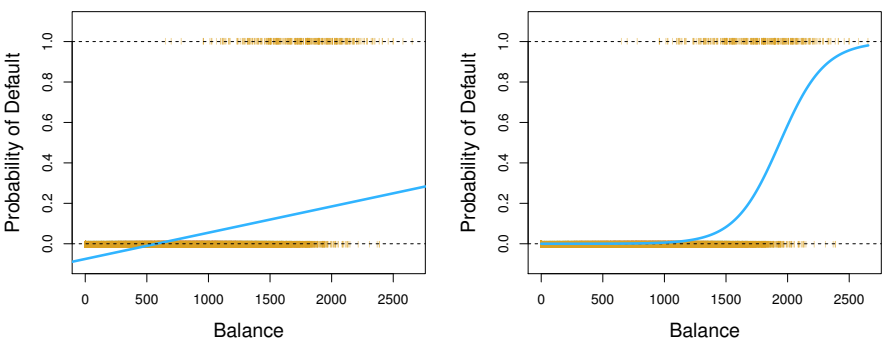


Figure. Linear Regression (Left) vs. Logistic Regression (Right)

Likelihood Function

- Given samples $S = (x_1, y_1), \ldots, (x_n, y_n)$, where $y_i \in \{0, 1\}$ is binary
- For sample i, the model's predicted probability of the observed outcome is:

$$P(Y_i = y_i) = \begin{cases} p(x_i) & \text{if } y_i = 1 \\ 1 - p(x_i) & \text{if } y_i = 0 \end{cases}$$

- Alternatively, a compact way to expressing the above function is

$$P(Y_i = y_i) = p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1 - y_i}$$

- The likelihood function: The total probability of observing the exact outcomes in S given the model's predictions $p(X)$

$$\mathcal{L}(\beta \mid S) = \prod_{i=1}^{n} p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1 - y_i}$$

Log-Likelihood Function

- The likelihood function

$$\mathcal{L}(\beta \mid S) = \prod_{i=1}^{n} p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1-y_i}$$

- The log-likelihood function

$$\ell(\beta \mid S) = \log \mathcal{L}(\beta \mid S) = \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))$$

- An alternative expression, $-y_i \log p(x_i) - (1 - y_i) \log(1 - p(x_i))$, is also known as the **cross-entropy**

Parameter Estimation

$$\hat{\beta} = \arg\max_{\beta} \ell(\beta \mid S) = \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))$$

- Choose the parameter that maximizes the (log) likehood function
- No closed form solution
- Use iterative methods like **gradient descent** to find the solution
  - Define objective function: $g(\beta) = -\ell(\beta \mid S)$
  - Gradient descent: $\beta_{k+1} = \beta_k - \gamma_k \nabla g(\beta_k)$
- Just like for linear regression, we can add regularization terms in case some of the features are correlated

Making Predictions

- Predictions given the estimated parameters

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\cdots+\hat{\beta}_p x_p}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\cdots+\hat{\beta}_p x_p}}$$

- **Example** Fit a logistic regression of loan default vs. balance:

$$\hat{\beta}_0 = -10.6513, \quad \hat{\beta}_1 = 0.0055$$

  - What is our estimated probability of **default** for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

  - With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

EXERCISE

Let's do it again, using **student** as the predictor.

	Coefficient	**Std. Error**	**Z-statistic**	**P-value**
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004

Table. Estimation Results

- The probability of default if the borrower is a student

$$\hat{p}(\text{default} = \text{Yes}|\text{student} = \text{Yes}) = ?$$

- The probability of default if the borrower is not a student

$$\hat{p}(\text{default} = \text{Yes}|\text{student} = \text{No}) = ?$$

EXERCISE

Let's do it again, using **student** as the predictor.

	Coefficient	**Std. Error**	**Z-statistic**	**P-value**
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004

Table. Estimation Results

- The probability of default if the borrower is a student

$$\hat{p}(\text{default} = \text{Yes}|\text{student} = \text{Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1 + e^{-3.5041+0.4049\times 1}} = 0.0431$$

- The probability of default if the borrower is not a student

$$\hat{p}(\text{default} = \text{Yes}|\text{student} = \text{No}) = \frac{e^{-3.5041+0.4049\times 0}}{1 + e^{-3.5041+0.4049\times 0}} = 0.0292$$

Part II

Assessing Prediction Performance

Confusion Matrix

- **Confusion matrix** is a table that summarizes the performance of classification

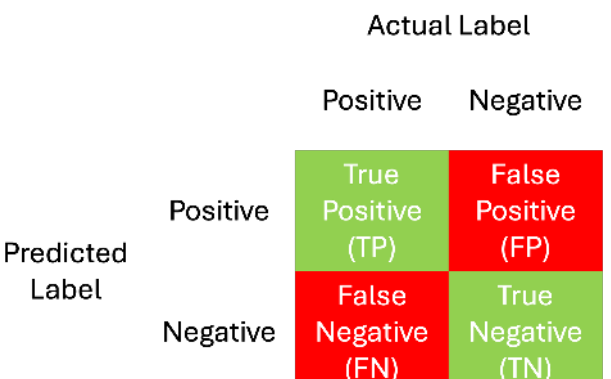


Figure. Confusion Matrix

- **Misclassification** Model classifies the data as a different class than it actually is

Accuracy

- The most common and intuitive metric is the accuracy score

$$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

- Accuracy score alone is not a good metric when the label imbalance exists
  - For example, if only 10% of the loans are charged off, a naïve classifier that predicts all loans not to default has an accuracy of 90%

Classification: Confusion Matrix (Loan Example)

- False positive: Fully paid loans that are predicted to be charged off
- False negative: Defaulted loans that are predicted to be fully paid
- False negative in unsecured loans is costly for banks
- What about cancer detection? Both FN and FP are costly

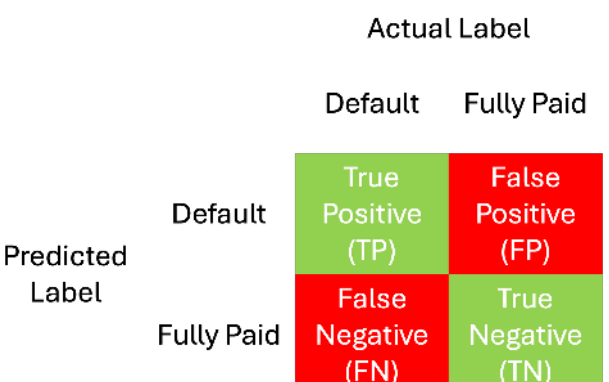


Figure. Confusion Matrix

Precision, Recall, and F1 Score

- **Precision** and **recall** indicate the occurrence of false positives and false negatives, respectively

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$$

- **F1 score** Weighted average (Harmonic mean) of precision and recall

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Classification Threshold

- Making a classification

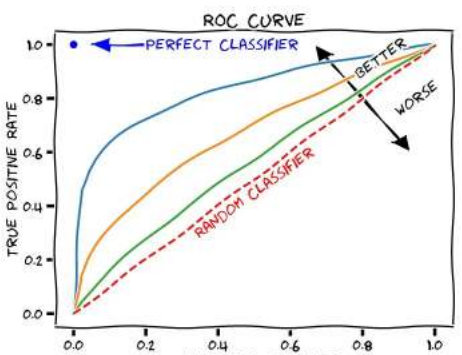
$$f(X) = \text{sign}(2p(X) - 1) = \begin{cases} 1 & p(X) > 0.5 \\ 0 & p(X) \leq 0.5 \end{cases}$$

- As we can see, 0.5 is a **threshold**
- The threshold is a hyperparameter; let's denote it as ρ
- Does ρ have to be 0.5? No
- As ρ changes between 0 and 1, the evaluation scores also change
  - **False-positive rate** $\text{FPR} = \frac{\sum \text{FP}}{\sum \text{TN+FP}}$
  - **True-positive rate** $\text{TPR} = \frac{\sum \text{TP}}{\sum \text{TP+FN}}$

ROC Curve

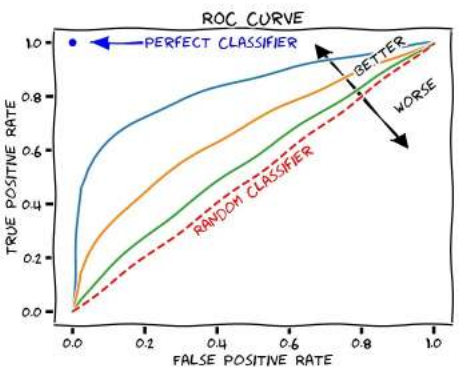
$$\rho \in [0, 1], \quad \text{FPR} = \frac{\sum \text{FP}}{\sum \text{TN} + \text{FP}}, \quad \text{TPR} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$$

- If $\rho = 0$, then FPR=1 and TPR=1
- If $\rho = 1$, then FPR=0 and TPR=0
- For $\rho \in (0, 1)$ there is a **trade-off** between FPR and TPR.
- The trade-off can be visualized by plotting the FPR against the TPR
- **Receiver operating characteristic (ROC) curve**



AUC Scores

- **Area Under the Curve** AUC score
  - AUC=1: Perfect classification
  - AUC=0.5: Random guess
  - Random guess is a naïve but often very useful benchmark
- Threshold independent: AUC measures how well the model can distinguish between the classes across all possible classification thresholds



Selection of Evaluation Metrics

- Precision: When the cost of false positive is high
  - Example: Email spam detection. The email user might lose important emails if the precision is not high for the spam detection model.
- Recall: When the cost of false negative is high
  - Example: Loans
- F1 score and AUC score are two useful balance between precision and recall
  - Example: Disease detection

- Select the best threshold ρ that optimizes the desired evaluation metric

Example 1: Loan Default Prediction

Table. Confusion Matrix

	Actual Default	**Actual No Default**
Predicted Default	20	15
Predicted No Default	10	55

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{75}{100} = 0.75$
  - Precision: $\frac{TP}{TP+FP} = \frac{20}{35} \approx 0.571$
  - Recall: $\frac{TP}{TP+FN} = \frac{20}{30} \approx 0.667$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.615$

Example 1: Loan Default Prediction

Table. Confusion Matrix

	Actual Default	**Actual No Default**
Predicted Default	20	15
Predicted No Default	10	55

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{75}{100} = 0.75$
  - Precision: $\frac{TP}{TP+FP} = \frac{20}{35} \approx 0.571$
  - Recall: $\frac{TP}{TP+FN} = \frac{20}{30} \approx 0.667$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.615$
- Discussion
  - Accuracy is high, but precision is lower due to false positives
  - For the bank, false positives might mean creditworthy customers are denied loans

Example 2: Credit Card Fraud Detection

Table. Confusion Matrix

	Actual Fraud	**Actual No Fraud**
Predicted Fraud	5	2
Predicted No Fraud	25	68

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{73}{100} = 0.73$
  - Precision: $\frac{TP}{TP+FP} = \frac{5}{7} \approx 0.714$
  - Recall: $\frac{TP}{TP+FN} = \frac{5}{30} \approx 0.167$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.27$

EXAMPLE 2: CREDIT CARD FRAUD DETECTION

Table. Confusion Matrix

	Actual Fraud	**Actual No Fraud**
Predicted Fraud	5	2
Predicted No Fraud	25	68

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{73}{100} = 0.73$
  - Precision: $\frac{TP}{TP+FP} = \frac{5}{7} \approx 0.714$
  - Recall: $\frac{TP}{TP+FN} = \frac{5}{30} \approx 0.167$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.27$
- Discussion
  - Low recall: The model misses many actual fraud cases
  - High precision suggests that when fraud is predicted, it's usually correct, but missing fraud could be very costly

Example 3: Spam Email Classification

Table. Confusion Matrix

	Actual Spam	**Actual Not Spam**
Predicted Spam	30	20
Predicted Not Spam	5	45

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{75}{100} = 0.75$
  - Precision: $\frac{TP}{TP+FP} = \frac{30}{50} = 0.60$
  - Recall: $\frac{TP}{TP+FN} = \frac{30}{35} \approx 0.857$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.705$

EXAMPLE 3: SPAM EMAIL CLASSIFICATION

Table. Confusion Matrix

	Actual Spam	**Actual Not Spam**
Predicted Spam	30	20
Predicted Not Spam	5	45

- Evaluation metrics
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN} = \frac{75}{100} = 0.75$
  - Precision: $\frac{TP}{TP+FP} = \frac{30}{50} = 0.60$
  - Recall: $\frac{TP}{TP+FN} = \frac{30}{35} \approx 0.857$
  - F1 Score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.705$
- Discussion
  - High recall but lower precision: Many spam emails are caught, but some legitimate emails are falsely marked as spam
  - Consideration: A good balance between catching spam and avoiding false positives is crucial for user experience

Part III

Generalization of Logistic Regression

Multiclass Logistic Regression

- Suppose there are $K > 2$ classes. We model the probability

$$p(Y = k \mid X) = \frac{e^{\beta_{0k} + \beta_{1k} X_1 + \cdots + \beta_{pk} X_p}}{1 + \sum_{\ell=1}^{K-1} e^{\beta_{0\ell} + \beta_{1\ell} X_1 + \cdots + \beta_{p\ell} X_p}} \qquad k = 1, \ldots, K-1$$

$$p(Y = K \mid X) = 1 - \sum_{k=1}^{K-1} p(Y = k|X) = \frac{1}{1 + \sum_{\ell=1}^{K-1} e^{\beta_{0\ell} + \beta_{1\ell} X_1 + \cdots + \beta_{p\ell} X_p}}$$

- The log-odds between the Kth class and any other classes is linear in the features

$$\log\left(\frac{p(Y = k \mid X = x)}{p(Y = K \mid X = x)}\right) = \beta_{k0} + \beta_{k1} x_1 + \cdots + \beta_{kp} x_p$$

- Parameter estimation

$$\hat{\beta} = \arg\max_{\beta} \log \mathcal{L}(\beta) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log\left(p(Y_i = k \mid X_i)\right)$$

PREDICTION

- For a new observation, the estimated parameters can be used to compute the predicted probabilities for each class,

$$\hat{p}(Y = k \mid X) = \frac{e^{\hat{\beta}_{0k} + \hat{\beta}_{1k} X_1 + \cdots + \hat{\beta}_{pk} X_p}}{1 + \sum_{\ell=1}^{K-1} e^{\hat{\beta}_{0\ell} + \hat{\beta}_{1\ell} X_1 + \cdots + \hat{\beta}_{p\ell} X_p}} \qquad k = 1, \ldots, K-1$$

$$\hat{p}(Y = K \mid X) = \frac{1}{1 + \sum_{\ell=1}^{K-1} e^{\hat{\beta}_{0\ell} + \hat{\beta}_{1\ell} X_1 + \cdots + \hat{\beta}_{p\ell} X_p}}$$

- The class with the highest probability is chosen as the predicted class

$$\hat{Y} = \underset{k \in \{1,\ldots,K\}}{\arg\max}\, \hat{p}(Y = k \mid X)$$

Evaluation

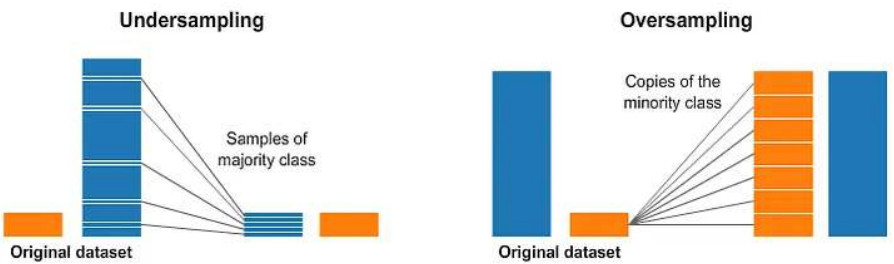
- Confusion matrix is still an intuitive and reliable evaluation to multiclass classification
- We can also reformulate the multiclass classification as multiple binary classification problems in evaluation
  - One class vs. else
  - Accuracy, precision, recall, f1, auc scores can be applied

Imbalanced Data

- **Imbalanced data** A situation where the classes are not represented equally, with one class having significantly more instances than the other(s), which can lead to biased model performance
  - Prevalent in financial applications, such as credit card fraud transactions and loan default
- There are several techniques for dealing with unbalanced data
  - Use of proper evaluation metrics based on the use cases
  - Threshold tuning
  - Resampling
  - Class weighting
  - Ensemble methods, such as bagging and boosting (later in this course)

RESAMPLING

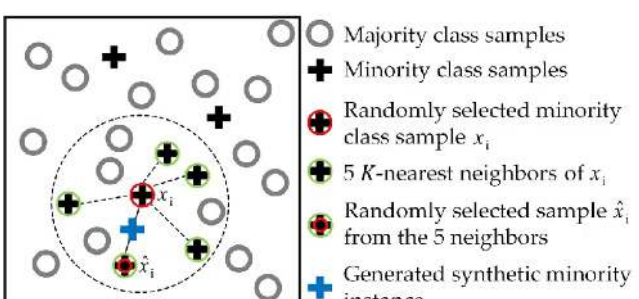
- **Undersampling** and **oversampling**



- More sophisticated resampling methods exist which take into consideration the features
- **Synthetic Minority Oversampling TEchnique (SMOTE)** generates synthetic instances for the minority class, rather than merely duplicating existing samples

SMOTE

1. Select a random example from the minority class
2. Find the k nearest neighbors for that example (typically $k = 5$)
3. A randomly selected neighbor (minority class) is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space
4. Step 1 - 3 is repeated until the desired number of samples is reached



Class Weighting

- Class weighting is implemented by modifying the loss functions to give more weight to the minority class
- Standard loss function

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

- Loss function with class weights

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \cdot [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

where w_{y_i} is the weight assigned to the class y_i; larger if y_i is the minority class

COMMENTS

- Resampling and class weighting both address imbalanced issues, but through different mechanisms
  - Resampling directly alters the data, while class weighting keeps the data unchanged
  - Class weighting adjusts the algorithm by modifying class importance
- Data synthetic approaches, such as SMOTE, have the risk of generating noisy and unrealistic data

SMOTE Risk: Generating Unrealistic Data

- Detecting a rare disease from two biomarkers, where the minority class (diseased) has a non-linear crescent shape
- SMOTE works by linear interpolation. It draws a line between two minority samples (Patient 1 and Patient 2) to create a new one
- The new "Synthetic Patient" is created in a biologically impossible region
- This noisy, unrealistic data can confuse the model and lead to a less accurate decision boundary