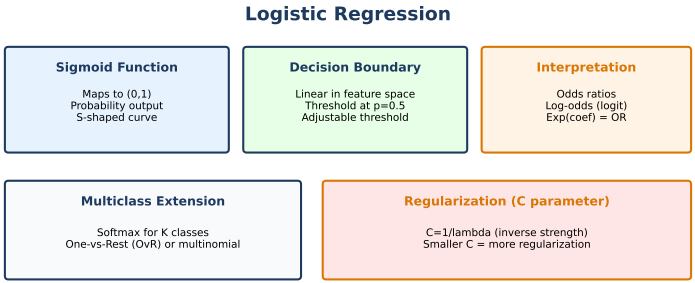
Lesson 25 Summary: Logistic Regression

Data Science with Python – Key Concepts

Data Science Program

Logistic Regression Overview



LogisticRegression(C=1.0).fit(X, y) | model.predict_proba(X)

Logistic regression is the go-to model for binary classification

The Sigmoid Function

Transforms linear combination to probability:

- **Formula:** $\sigma(z) = \frac{1}{1+e^{-z}}$
- **Output:** Always between 0 and 1
- **Input:** $z = w^T x + b$ (linear combination)

Sigmoid maps any real number to a probability

Decision Boundary

Separating classes:

- **Threshold:** Default at probability = 0.5
- **Geometry:** Linear boundary in feature space
- **Flexibility:** Can adjust threshold based on costs

If P(y=1—x) > threshold, predict class 1

The decision boundary is where $w^T x + b = 0$

Interpreting Coefficients

Odds ratio interpretation:

- **Coefficient:** Change in log-odds per unit x
- **Odds ratio:** $e^{coefficient}$
- **OR $>$ 1:** Feature increases positive class odds

Odds ratios are multiplicative effect on odds

Probability Outputs

Getting class probabilities:

- **predict():** Returns class labels (0 or 1)
- **predict_proba():** Returns probability array
- **Use case:** Ranking, threshold tuning

Probabilities are more useful than binary predictions

Regularization (C parameter)

Controlling complexity:

- **C:** Inverse of regularization strength
- **High C:** Less regularization, may overfit
- **Low C:** More regularization, simpler model

Default C=1.0 works well for most cases

Multiclass Classification

Extending beyond binary:

- **One-vs-Rest (OvR):** K binary classifiers
- **Multinomial:** Softmax for K classes simultaneously
- **sklearn default:** OvR for small data, multinomial for large

---

Set multi_class parameter explicitly for control

Feature Scaling

Important for convergence:

- **Problem:** Different scales slow optimization
- **Solution:** StandardScaler before fitting
- **Benefit:** Faster training, better convergence

Always scale features for logistic regression

Finance Application

Credit default prediction:

- **Target:** Default (1) vs No default (0)
- **Features:** Debt ratio, income, history
- **Output:** Probability of default score

Probability calibration is crucial for risk scoring

Quick Reference

Essential Commands:

Task	Code
Fit model	`LogisticRegression().fit(X, y)`
Predict class	`model.predict(X)`
Get probabilities	`model.predict_proba(X)`
Coefficients	`model.coef_, model.intercept_`
Regularization	`LogisticRegression(C=0.1)`

Logistic regression is interpretable and efficient