HANDOUT 4 - WEEK 4 READING

Regression and Model Metrics

CIS 350: AI for Cybersecurity - Smart Defense for the Digital Business

Colorado State University - Computer Information Systems

How to use this reading

This handout supports Week 4. It explains regression (predicting a number), how it differs from classification, and - most importantly - how we JUDGE a model: the confusion matrix, false positives vs false negatives, precision and recall, and the alert-threshold trade-off. The figures and code match Lab 4 and the lecture; the review questions map to Quiz 4 and prepare you for Exam 1.

Estimated reading time: 20-25 minutes. No prior coding background needed.

1. From Categories to Numbers: Regression

Last week's classifiers predicted a category (spam or ham). Regression predicts a NUMBER on a scale - a risk score from 0 to 100, an expected dollar loss, or the days until a system needs patching. The setup is the same supervised learning as before (labeled examples with features), but the label is now numeric. A number is powerful because it lets you RANK and PRIORITIZE - an analyst can work the ten riskiest logins first, which a plain yes/no label can't tell you.

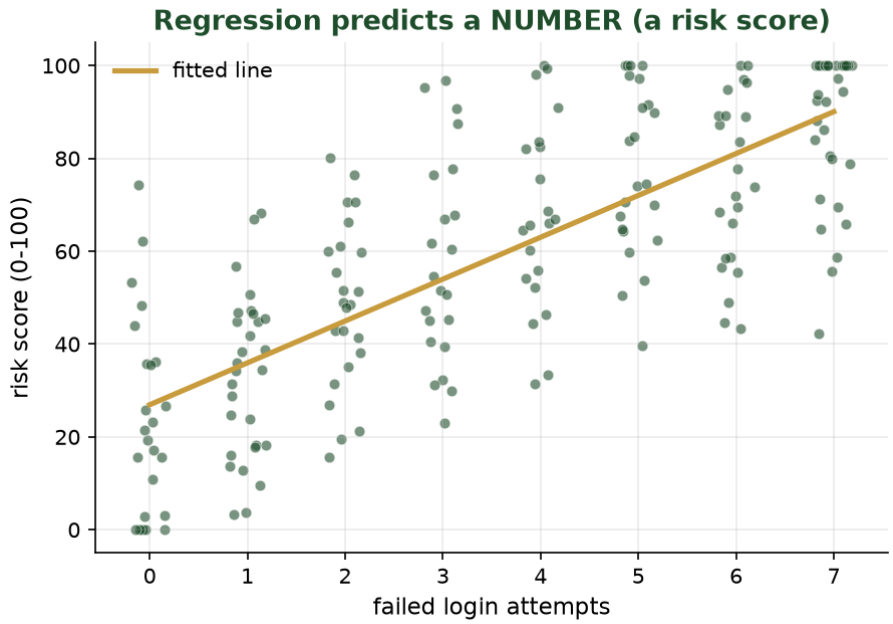

Figure 1. Regression fits a line (or curve) through the data. More failed logins -> a higher predicted risk score; you read a NUMBER off the line.

2. Regression vs. Classification

- Classification: the answer is a category ('is this phishing?'). Judged with accuracy, precision, recall.
- Regression: the answer is a number ('how risky, 0-100?'). Judged with error size and R-squared.
- Same workflow in code: create -> fit -> predict; only the model class changes.
- Rule of thumb: if you can list the answers as categories it's classification; if it's a quantity, regression.

Judging a regressor

Two ideas: (1) error - how far predictions land from the truth (smaller is better); (2) R-squared - a 0-to-1 summary where 1.0 is perfect and 0 is no better than guessing the average. A risk model doesn't have to be perfect; it has to RANK usefully - putting the riskiest cases on top.

3. Judging a Classifier: the Confusion Matrix

Accuracy - the percent of predictions that are correct - is a fine start, but it treats every mistake as equal. In security they are not: a missed attack and a false alarm have very different costs. The confusion matrix

breaks results into four outcomes so we can see WHAT KIND of mistakes a model makes.

- True Positive (TP): flagged spam, and it was spam - a correct catch.
- True Negative (TN): passed a real email, and it was legit - a correct pass.
- False Positive (FP): flagged a real email as spam - a FALSE ALARM.
- False Negative (FN): let spam through as legit - a MISSED ATTACK.

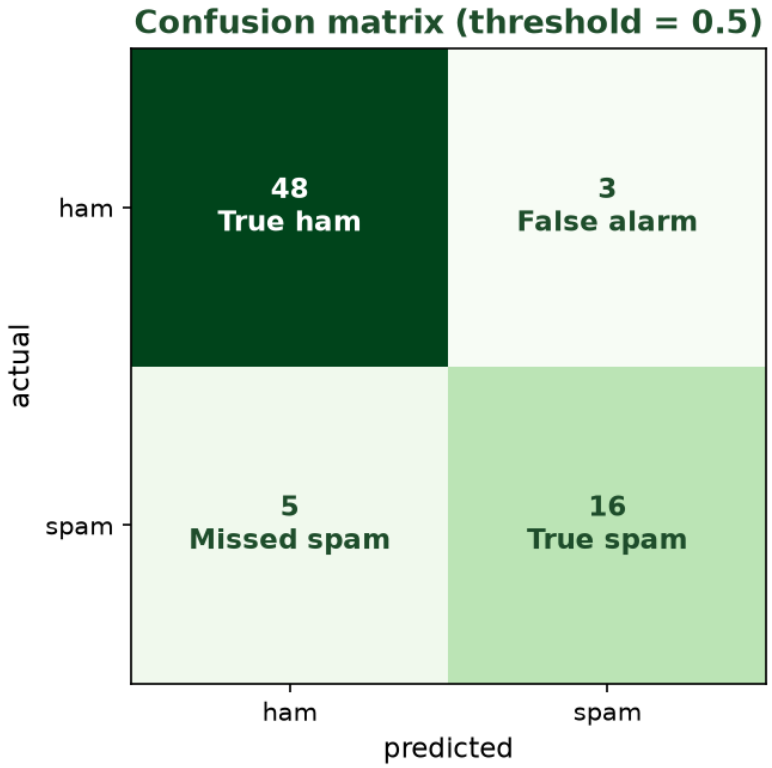

Figure 2. The confusion matrix: correct predictions on the diagonal, the two error types off it.

Which error is worse depends on the setting. Quarantining a real invoice (FP) hurts productivity and trust; letting ransomware through (FN) risks a breach. Security leaders decide which mistake they can least afford - and, as we'll see, there is usually a trade-off between the two.

4. Beyond Accuracy: Precision and Recall

When one class dominates - say 95% legitimate email - a model that always guesses 'legit' scores 95% accuracy while catching zero spam. To judge the class we care about, we use two measures:

- PRECISION: of the items we flagged, how many were actually positive? (high precision = few false alarms)
- RECALL: of all the real positives, how many did we catch? (high recall = few misses)
- There is tension: pushing one up often pulls the other down.
- Choose which to favor based on the cost of each error (e.g., recall for a threat scanner).

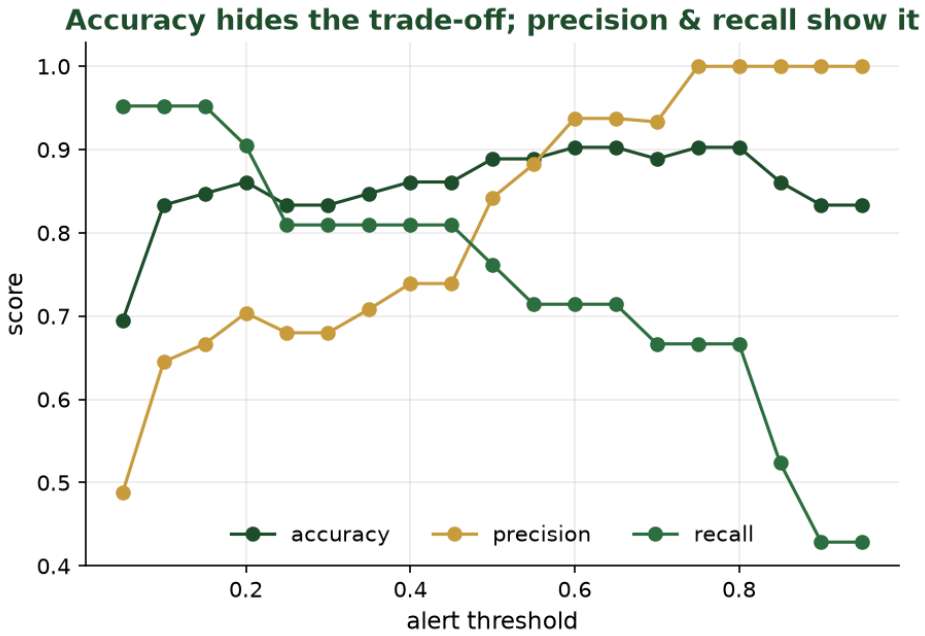

Figure 3. Accuracy looks flat and comfortable, while precision and recall move in opposite directions - the real trade-off accuracy hides.

5. The Threshold Trade-off and Analyst Fatigue

Most classifiers output a probability (e.g., '0.82 likely spam'), and WE choose the cutoff. Lower the threshold and you flag more - catching more attacks (higher recall) but raising more false alarms. Raise it and you get fewer false alarms (higher precision) but miss more. The threshold is a business decision about which error you can least afford.

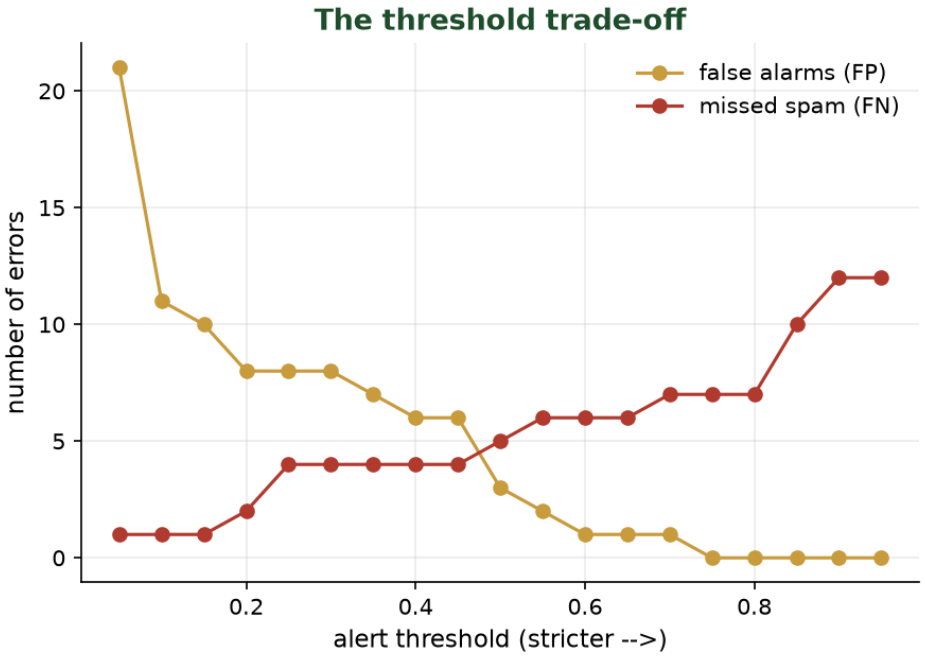

Figure 4. As the threshold gets stricter, false alarms fall but missed attacks rise. You cannot minimize both at once.

Analyst fatigue

Set the threshold too low and analysts drown in false alarms - then start ignoring alerts, including the real ones. Alert fatigue is a leading reason real breaches get missed. The fix is not just a better model but a well-chosen threshold, good triage, and ranking alerts by risk score (that's where regression helps).

6. Evaluating a Model in Code

Every metric in this handout is a one-line call in scikit-learn - exactly what you use in Lab 4:

```
from sklearn.metrics import (accuracy_score, confusion_matrix,
                             precision_score, recall_score)

preds = model.predict(X_test)
accuracy_score(y_test, preds)                    # overall correctness
confusion_matrix(y_test, preds)                  # [[TN, FP], [FN, TP]]
precision_score(y_test, preds, pos_label='spam')   # of flagged, how many right
recall_score(y_test, preds, pos_label='spam')      # of real spam, how many caught

prob = model.predict_proba(X_test)[:, 1]         # probability of 'spam'
strict = (prob >= 0.8)   # raise the threshold: fewer false alarms, more misses
```

Key Terms

- Regression - predicting a number; Classification - predicting a category.
- R-squared - 0-to-1 summary of regression fit (1.0 = perfect).
- Confusion matrix - a breakdown of TP, TN, FP, FN.
- False positive (false alarm) / False negative (missed attack).
- Precision - of flagged, how many right; Recall - of real positives, how many caught.
- Threshold - the probability cutoff for flagging; a business choice.
- Analyst fatigue - ignoring alerts due to too many false alarms.

Review Questions (self-check for Quiz 4, Lab 4, and Exam 1)

- Give one security task that is regression and one that is classification.
- Define false positive and false negative for a spam filter.
- Why can a 95%-accurate model still be a poor security tool?
- What is the difference between precision and recall?
- If you raise the alert threshold, what happens to false alarms and missed attacks?
- For a hospital's phishing filter, would you favor precision or recall - and why?

Remember: AI tools are not permitted on quizzes and labs in this course.

References

- NIST, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023.
- CISA, Phishing Guidance: Stopping the Attack Cycle at Phase One, 2023.
- OWASP, Machine Learning Security Top 10.
- IBM, Cost of a Data Breach Report (annual).
- Verizon, Data Breach Investigations Report (DBIR), annual.
- scikit-learn User Guide: Model evaluation (metrics and scoring).