Classification with KNN (Answer Key)

DATA 3301 Introduction to Data Science Lecture Notes 16

Name: _____ Date: _____

Key terms (define them in your own words)

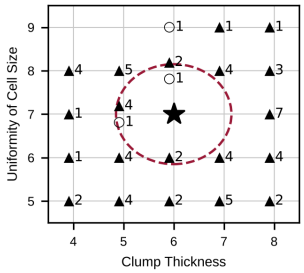
classification	supervised learning where the target is a category (a class), not a number
class probability	the share of the K neighbors' votes that went to a class
confusion matrix	a table counting predicted class against actual class
false positive	predicted the positive class, but the row is actually negative (a false alarm)
false negative	predicted negative, but the row is actually positive (a miss)
precision	of everything predicted in a class, the fraction that really is
recall	of everything actually in a class, the fraction the model found
F1 (F-measure)	the harmonic mean of precision and recall; high only when both are high

1. Regression vs. classification

- **Regression** predicts a numeric target. **Classification** predicts a categorical target, called a class.
- Breast biopsies: 9 features, each scored from 1 to 10. Target `Class`: 0 = benign, 1 = malignant.
- 683 biopsies: 444 benign and 239 malignant. The classes are not balanced.

2. KNN for classification

- Steps 1 and 2 are the same as regression: pick K, find the K nearest neighbors.
- Step 3: pick the majority class. With 3+ classes, the class with the most votes is the plurality.



- ▲ = malignant, ○ = benign, number = how many biopsies. **Circle** the 10 nearest neighbors of the star (6, 7); all are 1 unit away.
- Votes: 8 malignant, 2 benign → predict malignant.
- P(malignant) = 0.8, P(benign) = 0.2.

```
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X, y)
knn.predict_proba(new)   # [[0.2, 0.8]]
```

- **In your own words:** why can't KNN classification average its neighbors like KNN regression does?

The target is a category. You can't average "benign" and "malignant," so instead the neighbors vote and the most common class wins.

3. The confusion matrix

actual \ predicted	benign	malignant
benign	426 true negative	18 false positive
malignant	14 false negative	225 true positive

- **Write** true/false positive/negative in each cell. Positive class = malignant.
- Rows = actual, columns = predicted. The diagonal holds the correct predictions.
- A false positive is also a Type I error; a false negative a Type II error.

4. Accuracy, and when it misleads

- **Accuracy** = correct predictions / all predictions = (426 + 225) / 683 = 0.953. Training accuracy, so too optimistic.
- **Majority class trap:** 95 benign, 5 malignant; a model that always says benign has accuracy 0.95 and finds 0 malignant tumors.

```
accuracy_score(y_true, y_pred)
```

5. Precision, recall and F1 (one class at a time)

precision = TP / (TP + FP) **recall = TP / (TP + FN)**

- Malignant: precision = 225 / (225 + 18) = 0.926; recall = 225 / (225 + 14) = 0.941.
- Precision uses the predicted column; recall uses the actual row.

- **F1** = 2 × precision × recall / (precision + recall) = **0.934**. It is the **harmonic** mean, pulled toward the smaller value.

```
precision_score(y_true, y_pred, pos_label=1)
```

- **Synthesize:** for a cancer screening test, which matters more for the malignant class, precision or recall? Why?

Recall. A false negative sends a patient with cancer home untreated; a false positive only costs a follow-up test. Recall counts the misses.

6. Evaluating and tuning a classifier

- Same tools as regression: train-test split, cross-validation, grid search. Only the **metrics** change.

K = 10, random_state = 100	accuracy	recall (mal.)	F1 (mal.)
training set	0.963	0.938	0.947
test set	**0.927**	**0.851**	0.889

```
cross_validate(knn, X, y, cv=10, scoring=["accuracy", "f1"],
               return_train_score=True)
```

- GridSearchCV over K = 1 to 15 and two metrics, 10 folds: best K = **7** (Euclidean), CV accuracy **0.952**.
- **Exit question:** test accuracy (0.927) looks fine. Which test number above should worry a doctor, and why?

Test recall for malignant, 0.851: on new biopsies the model misses about 15% of cancers. Accuracy hides this because most biopsies are benign.

