

Disclaimer: Never had to write a write-up like this for code — so this is my best attempt.

Database Used: Iris Dataset

Classifier Used:

Scikit-Learn: SVM implementations from Scikit-learn was used. SVC with 'poly' kernel was used. Support vector machines do not directly provide estimates, but go through a 5-fold cross-validation for multi-class classification.

kNN: k-Nearest Neighbors classifier was made. First, the data was split. kNN works off of the 'distance' between data points, so first thing to do is to get that distance. Then, in the case of the iris dataset, we see how many of each type there are in the list of closest neighbors. Then the type with the most is the type that we classify our point as the neighbor.

How the Best Model Parameters Were Chosen:

Scikit-Learn: Working with a polynomial kernel, the best way to get a significant difference was to change the degree of the polynomial. As described below, the parameters that would have had an impact on the 'poly' kernel were tweaked, mostly through trial and error after learning what each one does.

kNN: There is competition between each of the model elements (data instances) in order to make a predictive decision and makes a model at the very end when it tells the prediction. In the code, k is optimized by seeing which value produces the most accurate predictions, and producing a number for k.

What the Parameters Were:

Scikit-Learn:

These are descriptions of the parameters. Tweaks can be found in the code.

- gamma: $1/n_{\text{features}}$ used when 'auto'
- max_iter: -1 means no limit
- probability: Probability estimates
- shrinking: Shrinking heuristic -- avoid heuristic so you can focus on accuracy
- tol: Tolerance for stopping
- degree: Degree of the polynomial function
- random_state: Random number generator to use when shuffling the data for probability estimation

Best Model Comparison (SVM vs. KNN):

- SVM Prediction Accuracy w/ Parameter Changes = 100.0%
- kNN Results when k=9: 95.65217391304348%

The SVM prediction vs. the kNN evaluated on the test set resulted in some interesting results, it did better. My thinking is because the iris set is designed to be linearly separable, the SVM is more likely to be more accurate, but the kNN model is something that is easily understandable but very powerful in application.