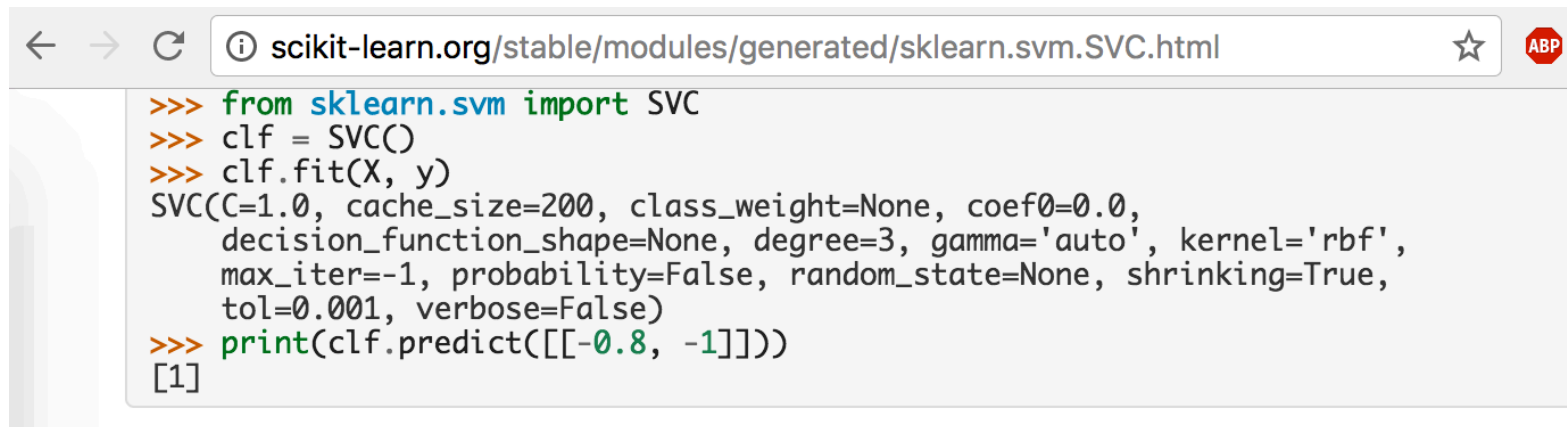


# Coding Up The SVM

- ex1



```
< > ↻ ⓘ scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html ☆ ABP  
>>> from sklearn.svm import SVC  
>>> clf = SVC()  
>>> clf.fit(X, y)  
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)  
>>> print(clf.predict([[ -0.8, -1]]))  
[1]
```

SVM *vs.* Naïve Bayes

## Ex2

Use SVM to classify emails by author

- SVM Author ID Accuracy
- SVM Author ID Timing

# Ex2

## Use SVM to classify emails by author

- A Smaller Training Set

*features\_train = features\_train[:len(features\_train)/100]*

*labels\_train = labels\_train[:len(labels\_train)/100]*

- Speed-Accuracy Tradeoff
- Deploy An RBF Kernel
- Optimize C Parameter (e. g. 10, 100, 1000, and 10000)

## Ex2

Use SVM to classify emails by author

- Use the full training set with optimized  $C$  for the RBF kernel
- Optimized RBF vs. Linear SVM?

## Ex2

Use SVM to classify emails by author

- Extract Predictions From An SVM (Use the RBF kernel,  $C=10000$ , and 1% of the training set)  
*answer=predictions[100]*
- How Many Chris Emails Predicted? (Use the RBF kernel,  $C=10000.$ , and the full training set)