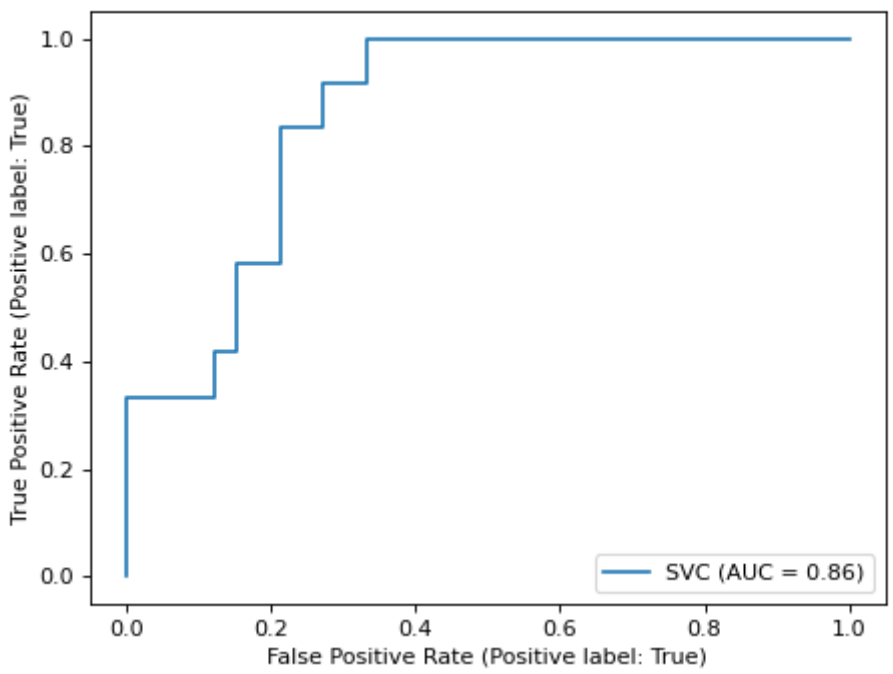
5. Visualizations

Scikit-learn defines a simple API for creating visualizations for machine learning. The key feature of this API is to allow for quick plotting and visual adjustments without recalculation. We provide `Display` classes that expose two methods for creating plots: `from_estimator` and `from_predictions`. The `from_estimator` method will take a fitted estimator and some data (`X` and `y`) and create a `Display` object. Sometimes, we would like to only compute the predictions once and one should use `from_predictions` instead. In the following example, we plot a ROC curve for a fitted support vector machine:

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import RocCurveDisplay
from sklearn.datasets import load_wine

X, y = load_wine(return_X_y=True)
y = y == 2  # make binary
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
svc = SVC(random_state=42)
svc.fit(X_train, y_train)

svc_disp = RocCurveDisplay.from_estimator(svc, X_test, y_test)
```

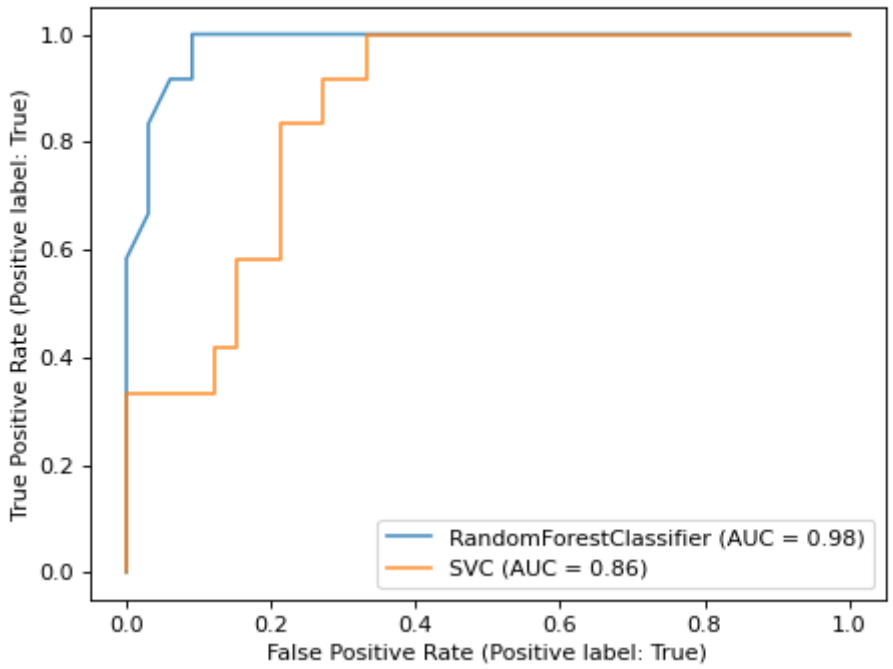


The returned `svc_disp` object allows us to continue using the already computed ROC curve for SVC in future plots. In this case, the `svc_disp` is a `RocCurveDisplay` that stores the computed values as attributes called `roc_auc`, `fpr`, and `tpr`. Be aware that we could get the predictions from the support vector machine and then use `from_predictions` instead of `from_estimator`. Next, we train a random forest classifier and plot the previously computed roc curve again by using the `plot` method of the `Display` object.

```
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n_estimators=10, random_state=42)
rfc.fit(X_train, y_train)

ax = plt.gca()
rfc_disp = RocCurveDisplay.from_estimator(rfc, X_test, y_test, ax=ax, alpha=0.8)
svc_disp.plot(ax=ax, alpha=0.8)
```



Notice that we pass `alpha=0.8` to the plot functions to adjust the alpha values of the curves.

Examples:

- ROC Curve with Visualization API
- Advanced Plotting With Partial Dependence
- Visualizations with Display Objects
- Comparison of Calibration of Classifiers

5.1. Available Plotting Utilities

5.1.1. Display Objects

`calibration.CalibrationDisplay`(prob_true, ...)	Calibration curve (also known as reliability diagram) visualization.
`inspection.PartialDependenceDisplay`(...[, ...])	Partial Dependence Plot (PDP).
`inspection.DecisionBoundaryDisplay`(*, xx0, ...)	Decisions boundary visualization.
`metrics.ConfusionMatrixDisplay`(...[, ...])	Confusion Matrix visualization.
`metrics.DetCurveDisplay`(*, fpr, fnr[, ...])	DET curve visualization.
`metrics.PrecisionRecallDisplay`(precision, ...)	Precision Recall visualization.
`metrics.PredictionErrorDisplay`(*, y_true, y_pred)	Visualization of the prediction error of a regression model.
`metrics.RocCurveDisplay`(*, fpr, tpr[, ...])	ROC Curve visualization.
`model_selection.LearningCurveDisplay`(*, ...)	Learning Curve visualization.