

УДК 00.004.896

МЕТОДИКА ОЦЕНКИ ТОЧНОСТИ АЛГОРИТМОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА ОСНОВАНИИ КРИВЫХ RECEIVER OPERATING CHARACTERISTIC

Пылов П. А., студент группы ИТб-162, IV курс

Протодьяконов А.В.,¹ к.т.н., доцент

Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Библиотека scikit-learn определяет набор функций для создания визуализаций для машинного обучения. Ключевые особенности – возможность быстрого построения графиков и визуальных корректировок без пересчета данных. В этой статье продемонстрируем, как использовать библиотеку для визуализации, сравнивая кривые ROC (receiver operating characteristic).

Первоначально загрузим данные и обучим модель машинного обучения. Моделью будет выступать двоичный классификатор. Представив задачу как двоичную классификацию, обучаем классификатор опорных векторов на наборе тренировочных данных (рисунок 1 отображает состав программного кода, требующегося для обучения классификатора)

```
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import plot_roc_curve
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split

X, y = load_dataset(return_X_y=True)
y = y == 2

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(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
     decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
     max_iter=-1, probability=False, random_state=42, shrinking=True, tol=0.001,
     verbose=False)
```

Рисунок 1. Обучение модели опорных векторов.

Приведенный рисунок (рисунок 1) отображает следующую последовательность действий: загрузка данных из датасета, присвоение значений мето-

¹ Научный руководитель

дом `train_test_split` (разделение датасета на тренировочные и тестовые данные) в обозначенные переменные, передача параметра алгоритму опорных векторов и обучение модели.

Теперь приступим к построению кривой ROC через метод `plot_roc_curve`. Объект кривой, получаемый в результате построения кривой ROC, впоследствии может быть использован в аналитике будущих графиков и кривых (рисунок 2 отображает программный код для получения кривой и кривую ROC)

```
svc_disp = plot_roc_curve(svc, X_test, y_test)
plt.show()
```

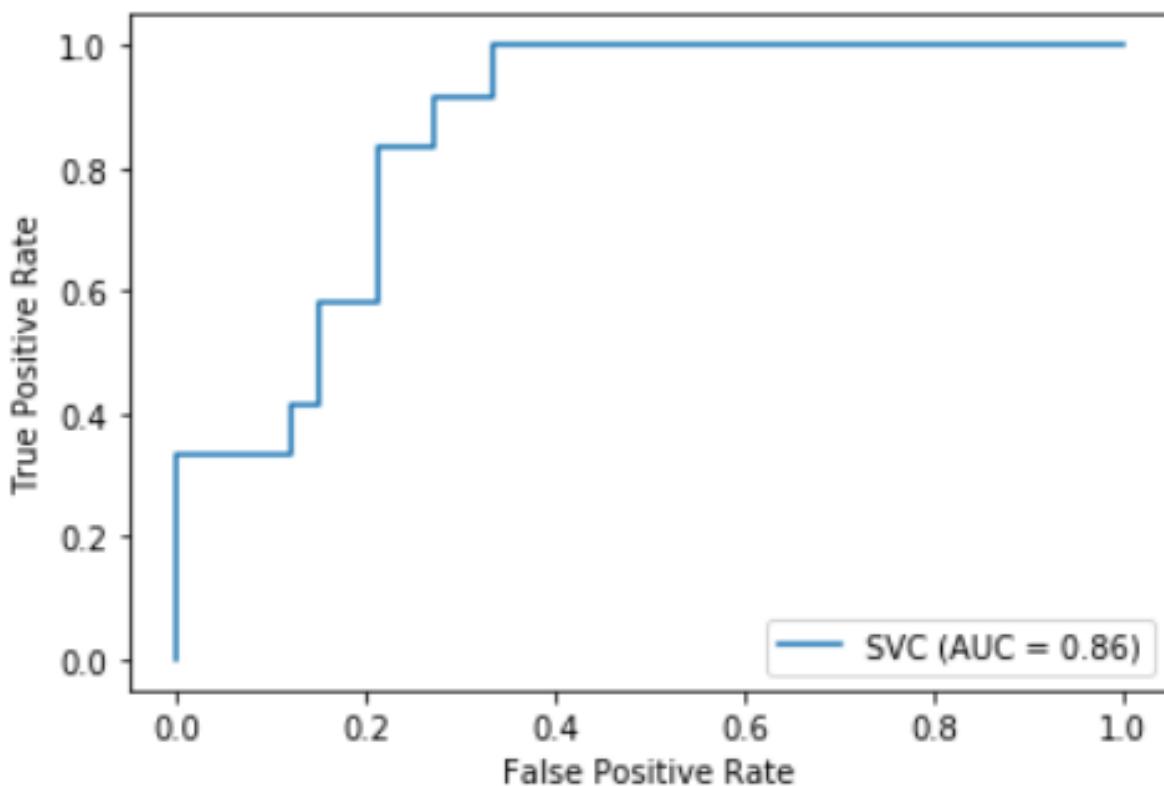


Рисунок 2. Получение кривой ROC.

График имеет особые названия осей:

- False Positive Rate – доля ложных положительных классификаций. Позволяет описать сколько из общего числа негативных существующих значений предсказаны были как ложно позитивные (иными словами, что модель нашла ошибку в данных общего датасета, но в реальности этой ошибки не существует – примером может служить ложный результат анализов);
- True Positive Rate – доля верных положительных классификаций. То есть то, на сколько в полном объеме алгоритм определяет правильные решения из всех возможных.

Добавим дополнительный алгоритм – алгоритм машинного обучения случайного леса; этот алгоритм будет обучен аналогичным образом на том же самом наборе данных. Представим характеристику кривой ROC алгоритма и проанализируем результаты (рисунок 3)

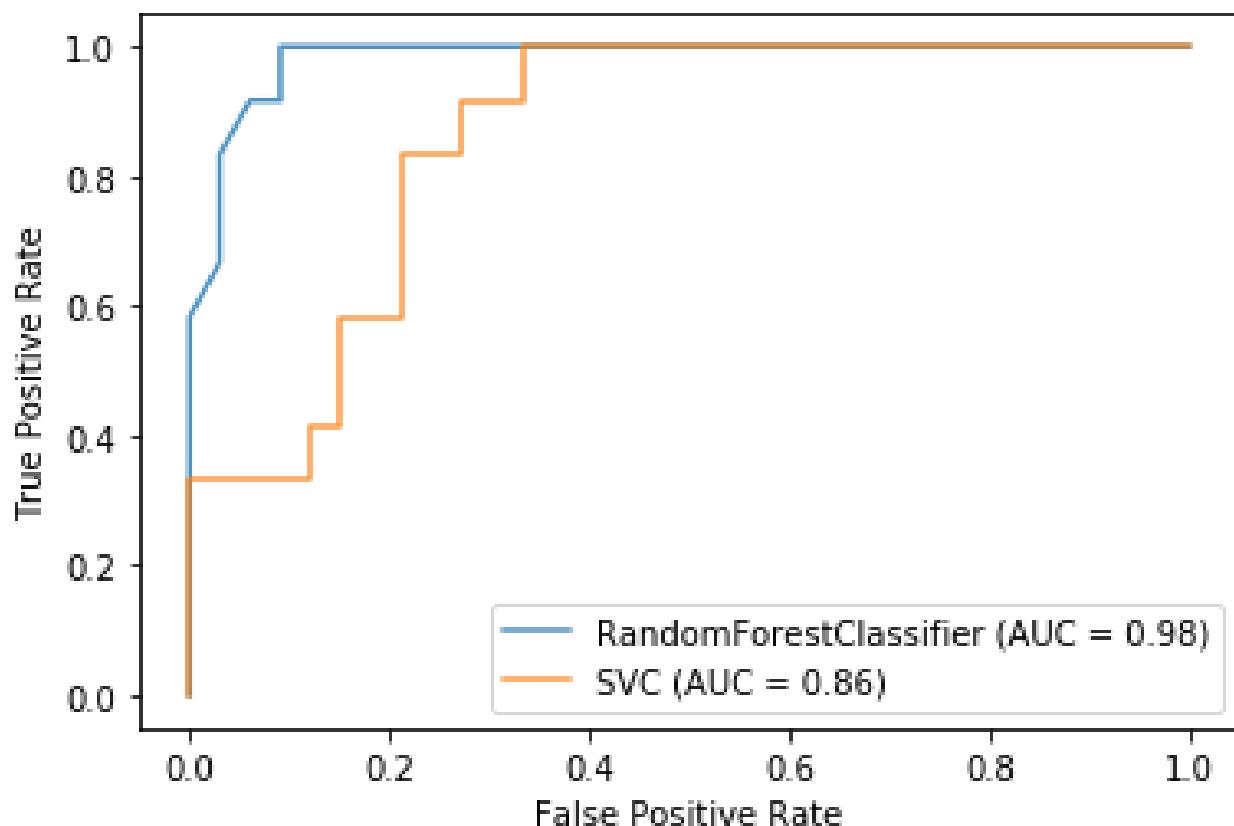


Рисунок 3. Кривые ROC алгоритмов случайного леса и опорных векторов.

Логически следует, что идеальным вариантом служит такая кривая ROC, которая проходила бы через три координатных точки – это начало координат (0,0), точка (0,1) и (1,1). Такой вариант практически невозможно реализовать, но достичь наиболее лучшего для требуемой задачи ответа поможет ROC кривая, благодаря которой появляется возможность визуальной оценки точности алгоритмов машинного обучения. Так, на предыдущем рисунке (рисунок 3) этим алгоритмом стал случайный лес, но в разных задачах более оптимальными могут стать совершенно другие алгоритмы, что ещё раз доказывает важность оценки алгоритмов на основе кривых ROC.

Список литературы:

1. M. Narasimha Murty, V. Susheela Devy. Introduction to pattern recognition and machine learning, IISc press, New Jersey – London, 2015
2. S. Theodoridis, A. Pikrakis. Introduction to Pattern Recognition: A MATLAB Approach, Elsevier press, 2010
3. Christopher Bishop. Pattern Recognition and Machine Learning, Plenum press, New York – London, 1971
4. A. Geron. Hands-on Machine learning with Scikit-Learn, Keras, and TensorFlow, OREILLY Sebastopol, California – USA, 2019
5. P. Bruce, A. Bruce. Practical statistics for data scientists, OREILLY Sebastopol, California – USA, 2017