COMPARISON OF CLASSIFIERS BASED ON THE DECISION TREE

Федоров Д.П.
Науковий керівник – д.т.н., проф. Кіріченко Л.О.
Харківський національний університет радіоелектроніки
61166, Харків, просп. Науки, 14, каф. прикладної математики,
тел. (057) 702-14-36, e-mail: demis.fedorov@nure.ua

The main purpose of this work is to compare classifiers. Random Forest and XGBoost are two popular machine learning algorithms. In this paper, we looked at how they work, compared their features, and obtained accurate results from their robots.

Однією з найпоширеніших завдань в інтелектуальному аналізі даних є класифікація даних, тобто побудова алгоритму класифікації. Загальна структура класифікаторів повторює принципи навчання з учителем: на вхід подається масив даних, що представляє собою набір випадків. Випадок має логічну структуру описує один зі станів. Алгоритм вибудовує свою логіку на основі навчання по цих зразках і подає на вихід класифікатор – механізм, здатний аналізувати набор даних і прогнозувати клас кожного зразка.

1. Розуміння бізнесу та даних. Метою нашої роботи є розробка математичної моделі класифікатора, який дозволяє найкращим способом встановити розподіл вхідних даних та встановити приналежність до класу.

У нашому випадку, базуючись на наборі даних італійського вина, дерево використовується для класифікації різних вин на основі вмісту алкоголю та ступеня розведення.

2. Вилучення ознак. Проблемою при побудові моделі класифікатора з якою ми зіткнулися це вилучення важливих ознак із набору даних для детального опису даних щоб далі добре навчити моделі.

Типовими ознаками можуть бути попередні значення ряду, мінімальні або максимальні значення в межах деякого вікна, стандартне відхилення і середнє, і так далі. Ознак можна придумувати нескінченно багато і нескінченно довго, але нам потрібен автоматизований алгоритм для цього. Для таких цілей ми використовуємо бібліотеку Tsfresh.

Ознаки можуть бути звичні – ті ж середні, максимальні і мінімальні значення. Далі поділили дані на навчальний та тестовий набори у відношенні 85:15 відповідно, через сильну схожість класів, тож 15 % всіх зразків будуть тестовими .

3. Моделювання та оцінка. Для розв’язання даної задачі нами були обрані ансамблеві класифікатори Random Forest та XGBoost .

Random Forest та XGBoost - два популярні алгоритми дерева рішень для машинного навчання.

Навчання на дереві рішень - це поширений тип алгоритму машинного навчання. Однією з переваг дерев рішень перед іншими алгоритмами

машинного навчання є те, наскільки легко вони спрощують візуалізацію даних. У той же час вони пропонують значну універсальність: їх можна використовувати для побудови як класифікаційних, так і регресійних прогнозних моделей.

Реалізація Random Forest є методом Bagging, який має кілька особливостей, зокрема, використовує всередині себе ансамбль тільки регресійних або класифікуючих дерев рішень і крім випадкового вибору об'єктів, також проводиться випадковим вибір ознак.

Для побудови бустингу ми обрали реалізацію з бібліотеки XGBoost, а точніше класифікатор XGBClassifier. Існують багато відомих реалізацій бустінгу, але обрана є найбільш відомою, має продуктивну модель та перевагу у швидкодії над іншими.

Для порівняння результатів ми виводимо точність класифікаторів на рисунку 1. Бачимо, що їх точність співпадає при невеликому набору даних.

```
test_score = RF_clf.score(test_x, test_y)
test_score
```

```
0.9629629629629629
```

(а) Точність алгоритму Random Forest

```
ypred = bst.predict(dtest, ntree_limit=bst.best_ntree_limit)
accuracy_score(test_y,ypred)
```

```
0.9629629629629629
```

(б) Точність алгоритму XGBClassifier

Рисунок 1 – Точність моделей класифікаторів

4. Результати.

Підводячи підсумок маємо два ансамблеві прийоми, які можуть зміцнити моделі на основі дерев рішень. Використання випадкового лісу генерує багато дерев, кожне з яких має листя однакової ваги в рамках моделі, щоб отримати вищу точність. З іншого боку, Gradient Boosting вводить зважування листя, щоб виправити тих, хто не покращує передбачуваність моделі. Обидва алгоритми дерева рішень, як правило, зменшують дисперсію, тоді як посилення також покращує зміщення.

Список використаних джерел:

1. Aurelien G. Hands-On Machine Learning with Scikit-Learn & TensorFlow : «O`REILLY Inc».

2. Breiman L. Bagging predictors in Machine Learning.