УДК 004.9

ПРИМЕНЕНИЕ БУСТИНГА НА ПРАКТИКЕ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

Чиков М.Ф., старший оператор научной роты, I курс
Военная академия связи имени Маршала Советского Союза С.М. Будённого, г. Санкт-Петербург

Для построения модели бустинга и оценки ее точности требуются программные методы библиотеки sklearn [1].

Следующим шагом необходимо загрузить набор данных в переменную датафрейма и представить его обобщенную структуру из первых пяти элементов при помощи метода head() [2, 3].

```
iris_data= pd.read_csv('Iris.csv')
iris_data.head(3)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa

```
iris_data.drop('Id', axis=1, inplace=True)
iris_data.head(3)
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

Рисунок 2 – Загрузка набора данных в переменную датафрейма

На 2 рисунке, кроме загрузки набора данных, представлено также удаление столбца Id. Этот столбец несёт в себе уникальное значение записи цветка ириса в общем наборе данных (иными словами, представляет собой тривиальный счётчик строк датасета) [4]. Если не удалить данный параметр, то он будет снижать обобщающую способность алгоритма [5, 6].

Исследуем набора данных на наличие пропущенных значений (рисунок 3). Пропуски не позволяют сформировать правильную обобщающую способность даже для одного алгоритма, а в модели бустинга их целый комплекс [7].

```
iris_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Рисунок 3 – Проверка набора данных на наличие пропусков

В наборе данных отсутствуют пропуски, поэтому можно переходить к следующему шагу построения программного решения.
Далее разделяем выборку набора данных на набор признаков (X) и столбец ответов (y):

```
X = iris_data[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']]
X.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
y = iris_data['Species']
y.head()
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

Рисунок 4 – Разделение признаков объектов (X) и целевого столбца прогнозирования (y)

Следующим шагом в разработке алгоритма бустинга является кодировка меток целевого столбца, то есть целевые классы цветка ириса (Iris-setosa, Iris-

versicolor, Iris-virginica) будут представлены классами 0, 1 и 2. Процесс кодировки меток представлен на рисунке 5 [8].

```
le=LabelEncoder()
y=le.fit_transform(y)
```

```
y
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int64)
```

Рисунок 5 – Кодировка метод целевого столбца

Поскольку набор данных состоит из 150 строк, то наиболее приемлемой величиной соотношения разбиения данных тренировочные/тестовые будет значение 70/30 [9] соответственно (рисунок 6).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Рисунок 6 – Разбиение набора данных на тренировочную и тестовую выборку

Теперь, когда всё необходимое подготовлено для программной реализации модели бустинга, напишем программный код для создания и обучения модели (рисунок 7).

```
abc = AdaBoostClassifier(n_estimators=50, learning_rate=1, random_state=0)
model1 = abc.fit(X_train, y_train)
y_pred = model1.predict(X_test)
```

Рисунок 7 – Создание и обучение модели бустинга

На рисунке 7 модель бустинга создается на основе 50 моделей решающих деревьев. Решающее дерево является стандартным алгоритмом в составе бустинга [10], а число 50 подобрано исходя из соответствия количеству экземпляров каждого целевого класса. Определим точность разработанной модели (рисунок 8).

```
print("Точность алгоритма бустинга на основе решающих деревьев составила:{:.3f} %"\
      .format(accuracy_score(y_test, y_pred)))
```

```
Точность алгоритма бустинга на основе решающих деревьев составила:0.933 %
```

Рисунок 8 – Точность разработанной модели бустинга на основе алгоритмов решающих деревьев

Отметим, что использовать можно не только решающие деревья, но и другие типы базовых алгоритмов. Для сравнения проверим точность модели бустинга на основе алгоритмов опорных векторов (задав аналогичные параметры количества алгоритмов в составе бустинга). Результат изображен на рисунке 9.

```
svc=SVC(probability=True, kernel='linear')
abc =AdaBoostClassifier(n_estimators=50, base_estimator=svc,learning_rate=1, random_state=0)
model2 = abc.fit(X_train, y_train)
y_pred = model2.predict(X_test)


print("Точность алгоритма бустинга на основе опорных векторов составила:{:.3f} %"\
      .format(accuracy_score(y_test, y_pred)))
```

```
Точность алгоритма бустинга на основе опорных векторов составила:0.911 %
```

Рисунок 9 – Точность модели бустинга на основе алгоритмов опорных векторов

Таким образом, по метрике точности accuracy, бустинг, демонстрирует высокие результаты точности (0,933 или 93,3%).

Данного значения точности достаточно для того, чтобы использовать разработанное решение в прикладных задачах (по общепринятой метрике точности, решение прикладных исследовательских задач должно быть больше или равняться величине 0,75 [11]).

Список литературы:

1. Свидетельство о государственной регистрации программы для ЭВМ № 2023680124 Российская Федерация. BrainPower : № 2023669010 : заявл. 16.09.2023 : опубл. 26.09.2023 / Р. В. Майтак. – EDN QXBJIM.
2. Математические и программные методы построения моделей глубокого обучения : Учебное пособие / А. В. Протодьяконов и др. – Вологда : Общество с ограниченной ответственностью "Издательство "Инфра-Инженерия", 2023. – 176 с. – ISBN 978-5-9729-1484-5. – EDN PZLUAH.
3. Свидетельство о государственной регистрации программы для ЭВМ № 2023680335 Российская Федерация. Maitak Intelligence Natural Language Processing Module : № 2023669704 : заявл. 27.09.2023 : опубл. 28.09.2023 / Р. В. Майтак.
4. Методы восстановления непараметрической регрессии в условиях несбалансированных данных / А. Д. Салычева и др. – Вологда : Общество с ограниченной ответственностью "Издательство "Инфра-Инженерия", 2024. – 192 с. – ISBN 978-5-9729-1856-0. – EDN AAJATW.

5. Свидетельство о государственной регистрации программы для ЭВМ № 2023684619 Российская Федерация. Efficient Network: № 2023684038: заявл. 14.11.2023: опубл. 16.11.2023 / П. А. Пылов.
6. Свидетельство о государственной регистрации программы для ЭВМ № 2023680070 Российская Федерация. Модернизированная модель DBSCAN для определения скрытых взаимосвязей : № 2023668841 : заявл. 13.09.2023 : опубл. 26.09.2023 / Р. В. Майтак. – EDN KQUUKF.
7. Асимптотический анализ поведения прикладных моделей машинного обучения : Учебное пособие / А. В. Протодьяконов и др. – Вологда : Общество с ограниченной ответственностью "Издательство "Инфра-Инженерия", 2023. – 144 с. – ISBN 978-5-9729-1455-5. – EDN APHQME.
8. Свидетельство о государственной регистрации программы для ЭВМ № 2023684621 Российская Федерация. Destructed Deep Random Forest: № 2023684050: заявл. 14.11.2023: опубл. 16.11.2023 / П. А. Пылов.
9. Свидетельство о государственной регистрации программы для ЭВМ № 2023684622 Российская Федерация. Mask Made AI: № 2023684042: заявл. 14.11.2023: опубл. 16.11.2023 / П. А. Пылов.
10. Свидетельство о государственной регистрации программы для ЭВМ № 2023680103 Российская Федерация. Cognitive Solution : № 2023669189 : заявл. 19.09.2023 : опубл. 26.09.2023 / Р. В. Майтак. – EDN QEMFJA.
11. Свидетельство о государственной регистрации программы для ЭВМ № 2023684624 Российская Федерация. Программа автоматического распознавания лиц в видеопотоке: № 2023684236: заявл. 15.11.2023: опубл. 16.11.2023 / П. А. Пылов.