Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Кафедра информатики

Лабораторная работа №10
«Градиентный бустинг»

Выполнил: Чёрный Родион Павлович
магистрант кафедры информатики
группа №858642

Проверил: доцент кафедры информатики
Стержанов Максим Валерьевич

Минск 2019

Постановка задачи

Для выполнения необходимо использовать набор данных boston из библиотеки sklearn

https://scikit-learn.org/stable/datasets/index.html#boston-dataset

1. Загрузите данные с помощью библиотеки sklearn.
2. Разделите выборку на обучающую (75%) и контрольную (25%).
3. Заведите массив для объектов DecisionTreeRegressor (они будут использоваться в качестве базовых алгоритмов) и для вещественных чисел (коэффициенты перед базовыми алгоритмами).
4. В цикле обучите последовательно 50 решающих деревьев с параметрами max_depth=5 и random_state=42 (остальные параметры - по умолчанию). Каждое дерево должно обучаться на одном и том же множестве объектов, но ответы, которые учится прогнозировать дерево, будут меняться в соответствие с отклонением истинных значений от предсказанных.
5. Попробуйте всегда брать коэффициент равным 0.9. Обычно оправдано выбирать коэффициент значительно меньшим - порядка 0.05 или 0.1, но на стандартном наборе данных будет всего 50 деревьев, возьмите для начала шаг побольше.
6. В процессе реализации обучения вам потребуется функция, которая будет вычислять прогноз построенной на данный момент композиции деревьев на выборке X. Реализуйте ее. Эта же функция поможет вам получить прогноз на контрольной выборке и оценить качество работы вашего алгоритма с помощью mean_squared_error в sklearn.metrics.
7. Попробуйте уменьшать вес перед каждым алгоритмом с каждой следующей итерацией по формуле 0.9 / (1.0 + i), где i - номер итерации (от 0 до 49). Какое получилось качество на контрольной выборке?
8. Исследуйте, переобучается ли градиентный бустинг с ростом числа итераций, а также с ростом глубины деревьев. Постройте графики. Какие выводы можно сделать?
9. Сравните качество, получаемое с помощью градиентного бустинга с качеством работы линейной регрессии. Для этого обучите LinearRegression из sklearn.linear_model (с параметрами по умолчанию) на обучающей выборке и оцените для прогнозов полученного алгоритма на тестовой выборке RMSE.

Описание реализации

1. Загрузим набор данных boston из библиотеки sklearn:

```
boston = load_boston()
x = boston['data']
y = boston['target']
```

2. Разделим выборку на обучающую и контрольную, используя метод train_test_split из пакета sklearn.model_selection:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

3. Заведем список для объектов DecisionTreeRegressor и список из 50 вещественных чисел - коэффициенты перед алгоритмами, положим их всех равными 0.9

```
m = 50
regressors = []
coefficients = [0.9] * m
```

4. Обучим последовательно 50 деревьев, каждое последующее будет учиться на ошибках предыдущего.

```
def train_predictors(coefficients, x_train, y_train):
    yi = y_train
    predictors = []
    for i in range(m):
        tree = DecisionTreeRegressor(max_depth=5, random_state=42)
        predictors.append(tree)
        tree.fit(x_train, yi)
        prediction = coefficients[i] * tree.predict(x_train)
        e = yi - prediction
        yi = e
    return predictors
```

5, 6. В результате обучения классификаторов получим значение функции ошибки (на валидационном наборе данных) - 13.943. Значения коэффициентов всегда брали равным 0.9

7. Реализуем функцию оценки качества прогнозирования модели:

```
def cost(predictors, coefficients, x, y):
    prediction = predict(predictors, coefficients, x)
    return mean_squared_error(y, prediction)
```

Попробуем уменьшать значения коэффициентов после каждой итерации. В результате получим значение функции ошибки равным 11.3051. Результат получился лучше, чем при постоянном значении коэффициентов.

8. Исследуем, переобучается ли градиентный бустинг с ростом числа итераций, а также с ростом глубины деревьев. График изменения качества предсказаний в зависимости от роста числа итераций представлен на рисунке 1. График зависимости качества предсказаний от глубины деревьев представлен на рисунке 2.

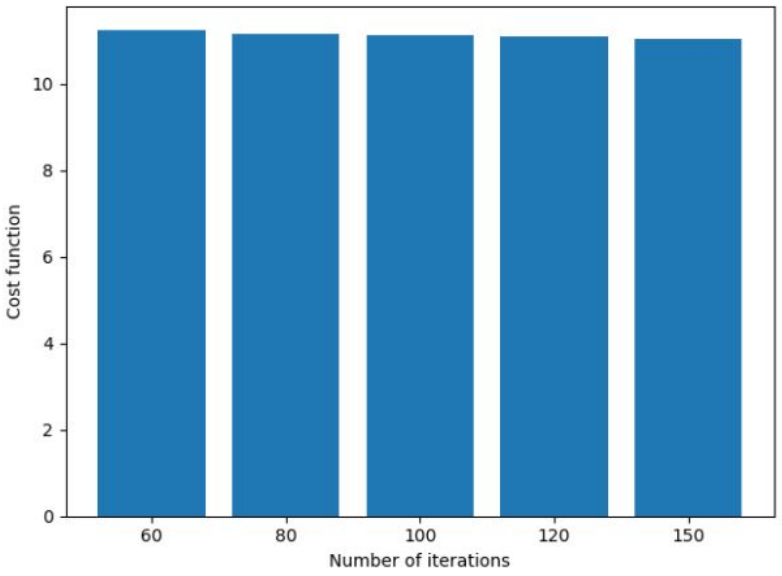

Рисунок 1 - Зависимость значения ошибки предсказаний от числа итераций

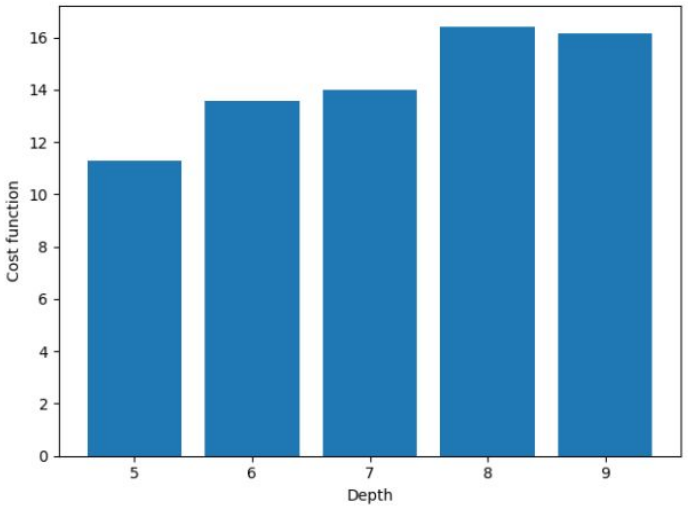

Рисунок 2 - Зависимость значения ошибки от глубины деревьев

Как видим из графиков, при росте числа итераций (количества деревьев) мы получаем лучшее качество предсказаний на контрольной выборке. А вот

при росте глубины деревьев происходит переобучение и качество на контрольной выборке падает.

9. Сравним качество, получаемое с помощью градиентного бустинга с качеством работы линейной регрессии. Обучим класс LinearRegression из sklearn.linear_model с параметрами по умолчанию на обучающей выборке и проверим качество модели на контрольной выборке.

```
lin_reg = LinearRegression()
lin_reg.fit(x_train, y_train)
error = mean_squared_error(y_test, lin_reg.predict(x_test))
```

В результате получим значение ошибки предсказаний 15.895, что немного хуже чем используя алгоритм градиентного бустинга.