

Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

## КУРСОВАЯ РАБОТА СТУДЕНТА 317 ГРУППЫ

« Логические корректоры в задаче восстановления регрессии »

Выполнил:

студент 3 курса 317 группы

*Байтеков Никита Вячеславович*

Научный руководитель:

д.ф-м.н., доцент

*Дюкова Елена Всеволодовна*

Москва, 2017

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Логические корректоры в задаче классификации</b>	<b>4</b>
2.1	Алгоритм MON . . . . .	4
2.2	Алгоритм MONS . . . . .	4
2.3	Генетический алгоритм . . . . .	6
<b>3</b>	<b>Сведение задачи восстановления регрессии к задаче классификации</b>	<b>8</b>
3.1	Алгоритм MONS-Reg . . . . .	8
3.2	Алгоритм кластеризации DM-DBSCAN . . . . .	8
3.3	Восстановление целевой переменной . . . . .	11
<b>4</b>	<b>Вычислительные эксперименты</b>	<b>11</b>
<b>5</b>	<b>Заключение</b>	<b>12</b>
	<b>Список литературы</b>	<b>12</b>

# 1 Введение

Исследуется множество объектов  $M$ , которые описываются набором целочисленных признаков  $\{x_1, \dots, x_n\}$ . Дано обучающее множество объектов  $T = \{S_1, \dots, S_m\}$  из  $M$ , такое, что для каждого обучающего объекта (прецедента) известно значение «ответа» из числового множества ответов  $Y$ . Требуется построить алгоритм распознавания  $A_T : M \rightarrow Y$ , ставящий в соответствие произвольному объекту из  $M$  ответ из  $Y$ . В случае, когда  $Y = \mathbb{R}$ , решается задача восстановления регрессии, а при  $Y = \{1, \dots, l\}, l \geq 2$  — задача классификации.

В работе рассматривается задача восстановления регрессии, для решения которой существует много методов. Чаще всего используется многомерная линейная регрессия, когда зависимость ответа от признаков предполагается линейной. Для решения задачи также используются адаптации методов классификации. Хорошо показывает себя классификация по методу ближайших соседей, который исходит из предположения, что у похожих объектов будут похожие ответы. Также рассматриваются вариации решающих деревьев, которые стабильно показывают высокое качество полученных решений.

Одной из современных моделей классификации является логический корректор [1] [2], в основе которого лежат идеи как логического, так и алгебраического подходов [3]. Основа классического логического подхода — использование корректных алгоритмов, базирующихся на построении корректных элементарных классификаторов. В алгебраическом подходе несколько распознающих алгоритмов объединяются в один для взаимной коррекции ошибок. Логические корректоры, хорошо зарекомендовавшие себя при решении прикладных задач, базируются на построении корректных наборов из произвольных элементарных классификаторов. В [6] разработан стохастический вариант логического корректора MONS. Целью данной работы является адаптация этого алгоритма к задаче восстановления регрессии.

Рассмотрим задачу классификации и введём основные определения. Пусть обучающее множество  $T$  разбито на  $l$  непересекающихся множеств  $\{K_1, \dots, K_l\}$ , называемых *классами*, и для каждого прецедента известно, к какому из классов он принадлежит. Обозначим множество прецедентов из  $T$ , принадлежащих некоторому классу  $K, K \in \{K_1, \dots, K_l\}$ , как  $T \cap K$ , а  $T \cap \bar{K}$  — как множество прецедентов из  $T$ , не

принадлежащих классу  $K$ . Пусть  $H = \{x_{j_1}, \dots, x_{j_r}\}$  — набор различных признаков,  $\sigma = \{\sigma_1, \dots, \sigma_r\}$  — набор допустимых значений этих признаков, где  $\sigma_q$  - допустимое значение признака  $x_{j_q}$ ,  $q = 1, \dots, r$ .

Пара  $(H, \sigma)$  называется *элементарным классификатором (эл.кл.)* Число  $r$  называется *рангом* эл.кл.  $(H, \sigma)$ . Обозначим через  $x_j(S)$  значение признака  $x_j$  на объекте  $S$  и через  $H(S)$  вектор значений признаков  $(x_{j_1}(S), \dots, x_{j_r}(S))$ . Говорят, что эл.кл.  $(H, \sigma)$  *выделяет* объект  $S$ , если  $H(S) = \sigma$ . Эл.кл.  $(H, \sigma)$  считается *корректным для класса*  $K$ , если не существует двух выделяемых эл.кл.  $(H, \sigma)$ , таких, что  $S' \in K, S'' \notin K$ .

Пусть имеется упорядоченный *набор эл.кл.*  $U = ((H_1, \sigma_1), \dots, (H_d, \sigma_d))$ . Набор  $U$  ставит в соответствие объекту  $S$  из  $M$  бинарный вектор  $U(S) = ([H_1(S) = \sigma_1], \dots, [H_d(S) = \sigma_d])$ , который называется *откликом* набора эл.кл.  $U$  на объекте  $S$  (здесь через  $[p]$  обозначается предикат, принимающий 1, если выражение  $p$  истинно, и 0 – в противном случае) Набор эл.кл.  $U$  называется *монотонно корректным для класса*  $K$ , где  $K \in \{K_1, \dots, K_l\}$ , если  $\forall(S', S''), S' \in K, S'' \notin K$  существует монотонная функция алгебры логики  $F_{U,K}: F_{U,K}(U(S')) \neq F_{U,K}(U(S''))$ . В таком случае, функция  $F_{U,K}$  называется *монотонно корректирующей* для  $U$ . Монотонно корректный набор  $U$  для класса  $K$  является *неприводимым*, если любой набор эл.кл., являющийся подмножеством  $U$ , не является монотонно корректным. Неприводимый набор для класса  $K$  является *минимальным*, если не существует монотонно корректных наборов для класса  $K$  меньшей мощности.

В настоящей работе разработан алгоритм MONS-Reg, решающий задачу восстановления регрессии на основе модификации стохастического логического корректора MONS. Проведено тестирование алгоритма MONS-Reg на реальных задачах.

## 2 Логические корректоры в задаче классификации

### 2.1 Алгоритм MON

Одним из первых логических корректоров, показывающих хорошие результаты, является алгоритм MON [5], который использовал монотонно корректные наборы, состоящие из эл.кл. ранга 1.

Процесс работы корректора MON разбивается на два этапа: обучение модели и распознавание. На этапе обучения для каждого класса  $K$ ,  $K \in \{K_1, \dots, K_l\}$ , распознающий алгоритм  $A$  строит семейство монотонно корректных наборов эл.кл.  $W_A(K)$ . При распознавании для каждого объекта  $S$  вычисляется оценка принадлежности этого объекта к каждому из классов  $K$  по формуле:

$$\Gamma_K(S) = \frac{1}{|W_A(K)|} \sum_{U \in W_A(K)} \frac{1}{|T \cap K|} \sum_{S' \in T \cap K} \delta_U(S, S').$$

где  $\delta_U(S, S')$  называется *функцией голосования* и определяется как:

$$\delta_U(S, S') = \begin{cases} 1, & \text{если } U(S) \succeq U(S'); \\ 0, & \text{иначе.} \end{cases}$$

Алгоритм относит неизвестный объект  $S$  к тому классу, для которого оценка принадлежности будет максимальной.

### 2.2 Алгоритм MONS

Алгоритм MONS является улучшенной моделью алгоритма MON, который использует т.н. *локальный базис* — множество эл.кл., на основе которого базовые алгоритмы строят наборы эл.кл. Использование локального базиса является компромиссом между желанием использовать в логическом корректоре эл.кл. ранга больше 1 и трудоёмкостью вычислений при поиске монотонно корректных наборов эл.кл. Формирование локального базиса само по себе является непростой задачей, и каждый алгоритм решает её по-разному. В нашем случае MONS использует стохастические методы, каждый раз формируя случайно небольшой локальный базис и на его основании составляя новые наборы эл.кл., которые добавляются к растущим семействам.

Для поиска минимальных наборов эл.кл. используется следующий алгоритм. Пусть  $LB = \{P_{(H_1, \sigma_1)}, \dots, P_{(H_{N_K}, \sigma_{N_K})}\}$  – локальный базис из  $N_K$  эл.кл., составленный для некоторого фиксированного класса  $K \in \{K_1, \dots, K_l\}$ . Рассмотрим всевозможные пары объектов вида  $(S', S'')$  и сопоставим каждой паре бинарный вектор вида  $B(S', S'') = (b_1, \dots, b_{N_K})$ , где

$$b_i(S', S'') = \begin{cases} 1, & \text{если } P_{(H_i, \sigma_i)}(S') = 1 \text{ и } P_{(H_i, \sigma_i)}(S'') = 0; \\ 0, & \text{иначе.} \end{cases}$$

Составим булеву матрицу  $L_K$  из всевозможных строк вида  $B(S', S'')$ , где  $S' \in T \cap K$ , а  $S'' \in T \cap \bar{K}$ . Заметим, что каждый столбец  $L_K$  соответствует определённому эл.кл. из  $LB$ , так что набор таких столбцов задаёт набор эл.кл. Очевидно, что любой набор эл.кл.  $U_K$  является монотонно корректным тогда и только тогда, когда набор признаков  $H = \{H_1, \dots, H_{N_K}\}$  является *покрытием матрицы*  $L_K$  – таким набором столбцов, что любая строка на пересечениях с этими столбцами имеет хотя бы одну единицу. Более того, любому неприводимому (минимальному) набору эл.кл. однозначно соответствует неприводимое (минимальное) покрытие. Таким образом мы свели задачу к поиску неприводимого покрытия булевой матрицы, которой известен как приложение задачи дуализации [8]. Для решения используются приближённые алгоритмы поиска, такие как градиентный спуск или генетический алгоритм, так как задача дуализации относится к трудно решаемым перечислительным задачам, и полиномиальных алгоритмов для её решения не было открыто до сих пор.

При обучении алгоритма MONS исходное обучающее множество  $T$  (или просто обучающая выборка) разбивается на две части – усечённую обучающую выборку  $T'$  и валидационную выборку  $V$ . Зададим отступ  $M(S)$  объекта  $S$  как разность между оценками истинного и предсказанного классов. Тогда алгоритм остановится, когда прирост качества для всех объектов станет меньше фиксированного  $\varepsilon$ .

---

**Algorithm 1** Процедура обучения алгоритма MONS

---

**Вход:**  $T$  — обучающая выборка,  $max\_i$  — максимальное число итераций,  $\varepsilon$  — порог;

**Выход:**  $W_K, K \in \{K_1, \dots, K_l\}$  — семейства мон. корректных наборов;

```
1:  $V := V(T)$ ;    // случайное выделение выборки  $V \subset T$  в качестве валидационной
2:  $T' := T \setminus V$ ; // усечение обучающей выборки
3: для  $i = 1, \dots, max\_i$ 
4:   для всех  $K \in \{K_1, \dots, K_l\}$ 
5:      $LB := LB()$ ; // случайным образом сформировать локальный базис
6:      $W_K^i := GA(T', LB, K)$ ; // ГА возвращает семейство мон. наборов из  $LB$ 
7:      $W_K := W_K \cup W_K^i$ ; // добавляем полученные наборы в семейство  $W_K$ 
8:   для всех  $S \in V$ 
9:      $M_i(S) := \Gamma_{y(S)}(W_{y(S)}^i, S) - \max_{K \in \{K_1, \dots, K_l\} \setminus y(S)} \Gamma_K(W_K^i, S)$  // отступ
10:     $M_i^{avg}(S) := \frac{1}{i} \sum_{j=1}^i M_j(S)$ ; // средний отступ по предыдущим итерациям
11:    если  $\forall S \in V M_i^{avg}(S) - M_{i-1}^{avg}(S) < \varepsilon$  то
12:      выход
```

---

## 2.3 Генетический алгоритм

На каждом шаге генетического алгоритма обновляется множество приближённых решений задачи, называемое *популяцией* в ходе итераций, называемых *поколениями*. Элементы популяции называются *особями*. Качество найденного приближенного решения характеризуется *функцией приспособленности*. При развитии популяции можно придерживаться различных подходов, к примеру, стратегия частичной замены подразумевает, что часть популяции переходит в следующее поколение без изменений. При таком подходе популяция постепенно избавляется от «наименее приспособленных» особей, но для этого необходимо поддерживать разнообразие особей, иначе генетический алгоритм преждевременно сойдется, попав в локальный минимум. Получение новых особей происходит с помощью операторов скрещивания и мутации, которые имеют явные параллели со своими биологическими аналогами.

В данном случае особями популяции являются покрытия булевой матрицы, кодирующие соответственные наборы эл.кл. Функция приспособленности для набора

эл.кл.  $U$  задаётся следующей формулой:

$$\begin{cases} f(U_i) = \tau(U_i) - \min_{j \in \{1, \dots, N\}} \tau(U_j) + 1, \\ \tau(U_i) = \frac{1}{|T' \cap K|} \sum_{S \in T' \cap K} \frac{1}{|V \cap K|} \sum_{S' \in V \cap K} \delta_{U_i}(S, S'), \end{cases}$$

где  $T'$  - усеченная обучающая, а  $V$  - валидационная выборки. Функцию  $f(U_i)$  требуется максимизировать — это важно, так как при разных функциях приспособленности могут возникнуть задача как максимизации, так и минимизации. *Начальная популяция* формируется из случайных наборов столбцов, которые, если понадобится, дополняются до покрытий. Если случайно сгенерированная особь уже присутствует в начальной популяции, то ничего не добавляется. *Выбор подпопуляции для скрещивания* осуществляется с помощью метода рулетки, когда вероятность выбора зависит от значения функции приспособленности у особи:

$$p_i = \frac{1/f_i}{\sum_{j=1}^{N_P} 1/f_j}$$

На каждой итерации для дальнейшего скрещивания выбирается ровно одна пара родителей, так как в этом случае лучшие особи остаются в популяции и сразу же могут участвовать в следующих скрещиваниях (метод *последовательной замены*). Новая особь получается с помощью взвешенного однородного кроссовера, когда значение каждой хромосомы копируется в набор потомка из набора одного из родителей с вероятностью, пропорциональной их значениям функции приспособленности. В алгоритме также используется особый оператор мутации, который повышает вероятность мутации с течением времени по следующей формуле:

$$k(t) = k_0 \left( 1 - \frac{1}{C \cdot t + 1} \right),$$

где  $k(t)$  - количество мутируемых хромосом на шаге  $t$ ,  $k_0$  - количество мутируемых хромосом на последнем шаге алгоритма, коэффициент  $C$  отражает скорость стремления  $k(t)$  к  $k_0$ . Затем восстанавливается допустимость решения, так как, вообще говоря, полученный набор столбцов не является неприводимым покрытием. На последнем этапе одна из наименее приспособленных особей замещается потомком.



## 3 Сведение задачи восстановления регрессии к задаче классификации

### 3.1 Алгоритм MONS-Reg

В данной работе была разработана модель *MONS-Reg*, которая решает задачу восстановления регрессии и является модификацией стохастического алгоритма классификации MONS. На первом этапе данные разбиваются по признаковому описанию на несколько кластеров с помощью алгоритма кластеризации DM-DBSCAN. На втором этапе применяется стандартный MONS для решения возникшей задачи классификации. На третьем этапе восстанавливается значение целевой переменной по полученной метке класса. Можно исходить из предположения, что у похожих объектов близкие значения целевой переменной, либо не допускать подобного. В первом случае логично разбить объекты (точки в пространстве признаков) по их признаковым описаниям на несколько кластеров, каждому из которых будет присвоена своя метка класса, и свести задачу к предыдущей. Во втором случае мы можем разбить на классы исходя только из значения самой целевой переменной.

### 3.2 Алгоритм кластеризации DM-DBSCAN

Для разбиения данных на кластеры можно использовать разные алгоритмы, но важно, чтобы алгоритм умел самостоятельно определять количество кластеров, присутствующих в выборке, отфильтровывал выбросы, умел работать с нелинейной геометрией данных а также был быстрым и простым в реализации. Таким параметрам удовлетворяет алгоритм DM-DBSCAN[7] — динамический плотностной алгоритм кластеризации пространственных данных с присутствием шума. Он, как и его предшественник DBSCAN, основывается на предположении, что внутри каждого кластера наблюдается типичная плотность объектов, которая заметно выше плотности снаружи, в то время как шумовые объекты разрежены сильнее, чем информативные.

Согласно алгоритму, объекты разбиваются на три типа: «ядровые», «граничные» и «шумовые». Для каждой точки в заранее заданном радиусе  $\epsilon$  считается число соседей — если оно больше некоторого порога, то она находится в середине предпола-

гаемого кластера и называется *ядровой*. В таком случае ей присваивается некоторая метка класса, и процесс рекурсивно повторяется для всех её соседей, которые ещё не были обработаны. Если же соседей меньше, чем нужно, но в их числе есть хоть один ядровой объект, то текущий объект является *граничным*. После первоначальной разметки(когда обрабатываются только ядровые точки) граничным объектам присваиваются метки ближайших ядровых. В противном случае, если в окрестности слишком мало соседей, а также нет ядровых точек, то объект помечается как *шумовой*. Стоит отметить, что у алгоритма DBSCAN два входных параметра: радиус распознавания соседей  $\varepsilon$  и порог для определения ядровых точек  $n_{min}$ . Но так как плотность получается фиксированной, то DBSCAN не умеет обрабатывать кластеры с различной плотностью. DM-DBSCAN лишён этих недостатков, так как оценивает уровни плотности каждого из кластеров по графику кривой расстояний до k-го ближайшего соседа.

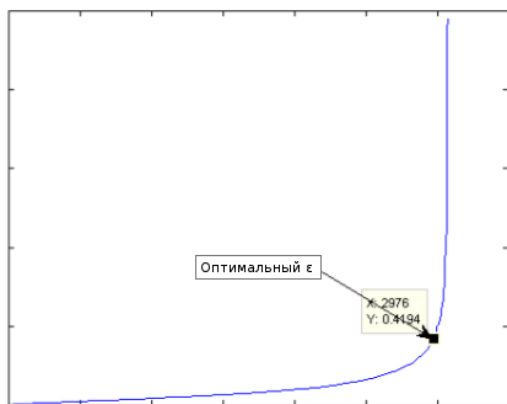


Рис. 1: Кривая k-расстояний для одного уровня плотности

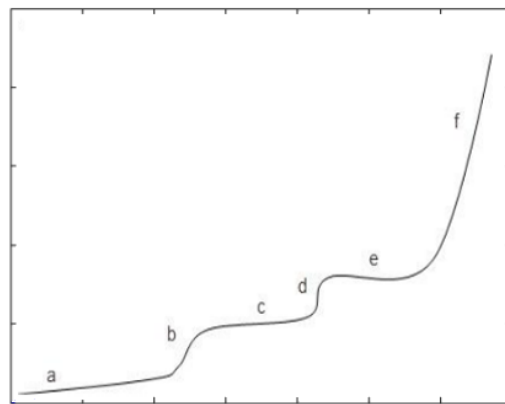


Рис. 2: Кривая k-расстояний для нескольких уровней плотности

На графиках горизонтальные участки соответствуют уровням плотности в данных, вертикальные участки соответствуют уровням шумовых точек. Для определения оптимальных значений используются точки перемены знака второй производной графика, которая вычисляется по стандартной разностной схеме.

---

**Algorithm 2** Алгоритм DM-DBSCAN

---

**Вход:**  $T$  — множество данных из  $n$  точек,  $n_{min}$  — минимальное число соседей;

**Выход:**  $Y$  — вектор разметки данных по кластерам;

```
1:  $D = dist\_mat(T)$ ; // матрицы расстояний по выбранной метрике
2:  $D^d = k\_deriv(D)$ ; // вторых производных графика  $k$ -расстояний
3:  $E = \emptyset, N = \emptyset$ ; // инициализация множеств порогов плотностей и шумов
4: для  $i = 1, \dots, n - 1$ 
5:   если  $D_i^d < 0$  и  $D_{i+1}^d > 0$  и  $D_i^d - D_{i+1}^d < \delta$  то
6:      $E = E \cup \{\frac{D_i^d + D_{i+1}^d}{2}\}$  // добавляем в множество порогов плотностей
7:   иначе если  $D_i^d > 0$  и  $D_{i+1}^d < 0$  и  $D_i^d - D_{i+1}^d < \delta$  то
8:      $N = N \cup \{\frac{D_i^d + D_{i+1}^d}{2}\}$  // или добавляем в множество порогов шумов
9:    $PQ = \emptyset, C = 0$ ; // очередь обработки и переменная для меток класса
10:   $PD = \emptyset, BD = \emptyset$ ; // множества обработанных и граничных точек
11:  для  $i = 1, \dots, n$ 
12:    если  $type(T_i, E, N) = \text{«ядровая точка»}$  то
13:       $Y_{T_i} = C$ ; // присвоим новую метку класса
14:       $PD = PD \cup \{T_i\}$ ; // помечаем точку как обработанную
15:       $PQ = PQ \cup (neighbors(T_i, E) \setminus PD)$ ; // добавляем в  $PQ$  всех соседей  $T_i$ 
16:      пока  $PQ \neq \emptyset$ 
17:        если  $PQ_1 \notin PD$  и  $type(T_i, E, N) = \text{«ядровая точка»}$  то
18:           $Y_{PQ_1} = C$ ; // присвоим новую метку класса
19:           $PQ = PQ \cup (neighbors(PQ_1, E) \setminus PD)$ ; // добавляем соседей  $PQ_1$ 
20:        иначе если  $PQ_1 \notin PD$  и  $type(T_i, E, N) = \text{«граничная точка»}$  то
21:           $BD = BD \cup \{T_i\}$ ;
22:           $PQ = PQ \setminus \{PQ_1\}$ ;
23:           $C = C + 1$ ;
24:        иначе если  $type(T_i, E, N) = \text{«граничная точка»}$  то
25:           $BD = BD \cup \{T_i\}$ ;
26:        иначе если  $type(T_i, E, N) = \text{«шум»}$  то
27:           $Y_i = -1$ ; // шум помечается отдельной меткой
28:  для  $i = 1, \dots, |BD|$ 
29:     $Y_{BD_i} = label(closest\_core(BD_i, D))$ ;
```

---

### 3.3 Восстановление целевой переменной

После разбиения на кластеры используется алгоритм MONS для решения уже знакомой задачи классификации, настраивая модель. При распознавании нового объекта алгоритм MONS в качестве ответа выдаст один из кластеров, к которому объект, скорее всего, принадлежит. Найдём значение целевой функции объекта, основываясь на значениях объектов из данного кластера. Есть несколько вариантов поведения, например брать среднее значение или медиану всех значений целевой переменной в классе или у некоторых объектов в нашем классе, которые наиболее похожи на тестируемый — можно использовать разные метрики, в зависимости от типа данных.

## 4 Вычислительные эксперименты

Алгоритм *MONS-Reg* сравнивался с 4 основными моделями, решающими задачу восстановления регрессии: линейной регрессией (далее обозначаемой как *Linear*), градиентным бустингом (*GB*), методом ближайших соседей (*kNN*) и многослойным персептроном (*MLP*). Реализация данных моделей была взята из библиотеки машинного обучения *scikit-learn*. Использовались стандартные параметры. Для тестирования было выбрано 8 прикладных задач из ресурса UCI. Для оценки качества работы применялся скользящий контроль на 10 разбиениях по функционалу качества MSE по следующей формуле:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

где  $y$  - вектор истинных значений ответов для объектов проверочной выборки,  $\hat{y}$  - вектор предсказанных моделью ответов,  $n$  - размер выборки.

Ниже в таблице 1 приведены результаты работы перечисленных моделей на соответствующих задачах:

Таблица 1: Оценка MSE

Данные	Размер ( $m \times n$ )	MONS-Reg	Linear	kNN	MLP	GB
Servo	$167 \times 4$	0.95662	1.37443	0.94863	0.38500	<b>0.18757</b>
Computers Hardware	$209 \times 5$	<b>3908.57</b>	8128.60	8409.37	8869.31	4285.30
Yachts	$308 \times 6$	248.166	85.5714	137.483	13.5737	<b>0.64404</b>
Concrete Slump	$103 \times 7$	25.5992	8.12311	22.8047	<b>6.92463</b>	9.47000
Breast Cancer	$198 \times 31$	0.12037	0.15615	<b>0.11056</b>	0.23432	0.13707
Breast Cancer(wdbc)	$699 \times 8$	0.09354	0.06418	0.06032	0.16360	<b>0.03705</b>
Breast Cancer(wdbc)	$569 \times 29$	<b>0.18724</b>	0.19486	0.20970	0.38867	0.19619
Forest Fire	$517 \times 12$	5157.95	<b>4168.22</b>	4746.72	21395.5	5640.77

Из таблицы видно, что *MONS-Reg* превосходит другие алгоритмы восстановления регрессии на двух задачах из списка и показывает схожее качество на остальных. Из минусов можно отметить большее время работы по сравнению с остальными алгоритмами, так как модель является более сложной и таблицы видно, что модель способна соперничать с ведущими моделями, а порой и превосходить их на отдельных задачах.

## 5 Заключение

В данной работе была рассмотрена задача восстановления регрессии. Предложен новый подход к решению, основанный на использовании модели логического корректора. Реализованы стохастическая модификация логического корректора алгоритм MONS и динамический плотностной алгоритм кластеризации DM-DBSCAN. Поставленная задача решена на основе ансамбля алгоритмов. Приведены результаты тестирования алгоритма на прикладных задачах.

## Список литературы

- [1] *Дюкова Е.В., Журавлёв Ю.И., Прокофьев П.А.* Логические корректоры в задаче классификации по прецедентам // Ж. вычисл. матем. и матем. физики, 2017, 57:11 С.141-162
- [2] *Djukova E. V., Zhuravlev Yu. I., Sotnezov R. M.* Construction of an Ensemble of Logical Correctors on the Basis of Elementary Classifiers // Pattern Recognition and Image Analysis, 2011, Vol. 21, No4, pp. 599—605.
- [3] *Журавлёв Ю.И.* Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики, 1978, вып. 33, С.5–68
- [4] *Дюкова Е.В., Журавлёв Ю.И., Рудаков К.В.* Об алгебро-логическом синтезе корректных процедур распознавания на базе элементарных алгоритмов // Ж. вычисл. матем. и матем. физики, 1996, 36:8 215–223
- [5] *Sotnezov R. M.* Genetic Algorithms for Problems of Logical Data Analysis in Discrete Optimization and Image Recognition // Pattern Recognition and Image Analysis, 2009, Vol. 19, No. 3, 469–477
- [6] *Любимцева М.М.* Логические корректоры в задачах распознавания// Тезисы лучших дипломных работ факультета ВМК МГУ, 2014. С.47-49.
- [7] *Ashour W. M., Elbatta M. T. H.* A Dynamic Method for Discovering Density Varied Clusters // International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.6, No. 1, 2013
- [8] *Дюкова Е.В.* Об асимптотически оптимальном алгоритме построения тупиковых тестов // ДАН СССР. 1977. Т. 233. № 4. С. 527-530.