

НЕЧЕТКИЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧ КЛАССИФИКАЦИИ

Магеррамов И.М.

Аспирант 2 курса, ЧОУ ВО ЮУ (ИУБиП)

e-mail:imr.magerramow@yandex.ru

Александрова Т.

Аспирант 1 курса, ЧОУ ВО ЮУ (ИУБиП)

e-mail: anysya08@yandex.ru>

Аннотация: В статье рассматриваются вопросы классификации сущностей при принятии решений в коммерческих организациях, а также предлагается алгоритм и инструментарий для проведения анализа эффективности сотрудников в разрезе обработки клиентских обращений. Алгоритм основан на многопараметрической оптимизации исходного набора данных по нечетким векторам требований. В качестве инструментов используются возможности языков Python (библиотеки Tkinter, matplotlib, openpyxl), PHP (обработка данных и их классификация с помощью серверных методов на Yii 2.0), а также SQL (запросы к базе данных корпоративной CRM-системы). Для визуализации данных применяются Excel и PowerBI. В качестве методической базы используется аппарат нечеткой логики (функции принадлежности, их аргументы и коэффициенты). Используются парадигмы функционального и объектно-ориентированного программирования. В результате формируется набор данных для классификации персонала и принятия решений по каждому сотруднику.

Ключевые слова: классификация, принятие решений, нечеткость, оптимизация, функциональное программирование, объектно-ориентированное программирование, визуализация данных, хранение и обработка данных.

FUZZY APPROACHES TO SOLVING CLASSIFICATION PROBLEMS

Magerramow I.M.

Aleksandrova T,S,

Abstract: In the article are considered entities classification issues for the purposes of decision making in commercial companies and also there are offered algorithm and tools for employees efficiency analysis conducting in the context of clients' requests processing. The algorithm is based on multiparameter optimization of the initial dataset by fuzzy requirements vectors. As tools there're used Python (Tkinter, matplotlib and openpyxl libraries) PHP (data processing and classification via server methods based on Yii 2.0) and also SQL (corporate CRM database queries) languages abilities. For data visualizing there're used Excel and Power BI. The methodological base, used in this work, is the fuzzy logics terms. Also there're used functional and object oriented programming paradigms. As the result of the algorithm, there is being formed the dataset for staff classification and decision making by each of the employees.

Keywords: classification, decision making, fuzziness, optimization, functional programming, object oriented programming, data visualizing, data storage and processing.

В предыдущих работах рассматривался алгоритм многопараметрической оптимизации для сравнения оценок альтернатив решения с нечеткими векторами требований, основанный на подсчете взвешенного расстояния Хэмминга, детерминированной и неопределенной составляющих, а также его реализация на Python[3, с.158].

Рассмотрим практическое применение алгоритма по данным корпоративной CRM в части обработки клиентских обращений (заявок). Введем следующие ограничения и допущения:

- завершенной считается заявка, получившая результативный статус («Успех», «Для ознакомления», «Отказ», «Передано партнеру»);
- временем жизни (time-to-live) заявки считается время от ее добавления в систему до последнего переключения статуса менеджером;
- датой закрытия считается дата последнего переключения статуса заявки менеджером;
- в подсчетах времени жизни участвуют только завершенные заявки;
- временной горизонт – текущий и предыдущий годы;
- в выборках участвуют только активные записи.

CRM хранит данные в реляционной СУБД MySQL на InnoDB. Ее таблицы нормализованы до третьего порядка, что позволяет строить выборки с минимальным временем обработки. Для целей подсчета TTL необходимо использовать таблицы заявок, логов, сотрудников и пользователей. Общий вид SQL-запроса на получение дат создания, закрытия приведен на рисунке 1.

```
SELECT f_logs.line_id As zayavka, f_users.fio As manager, f_data130.f1680 As status, f_data130.add_time As add_date, Max(f_logs.date) As close_date FROM f_logs join f_data130 On (f_logs.line_id = f_data130.id) left join f_users On (f_data130.f1740 = f_users.id) where f_logs.table_id = 130 And f_logs.user_id In (Select f1400 From f_data46 where f555 In (2, 781) And f_data46.status = 0) And event = 'change_field' And text Like '%"Статус" таблицы "Заявки"%' And line_id In (Select id From f_data130 Where add_time > '2020' And add_time < '2022' And f1680 In ('Отказ', 'Для ознакомления', 'Успех', 'Передано партнеру') And status = 0) Group BY f_data130.id
```

Рисунок 1 – SQL запрос на поиск ключевых дат по заявкам

Приведенный на рисунке запрос состоит из двух подзапросов:

- поиск активных менеджеров (сотрудники с категориями 2, 781 и активным статусом 0);

- поиск закрытых заявок в обозначенном временном горизонте (год от 2020 до 2022, статус заявки входит в закрывающие, а статус записи активен).

По таблице логов ищутся записи, имеющие тип события «изменение поля» («change_field»), относящиеся к таблице заявок (условие f_logs.table_id = 130), затрагивающие закрытые заявки из вложенного подзапроса, ведущиеся активными менеджерами. Также дополнительно указывается текстовая маска события («Статус» таблицы «Заявки» в условии like) и выполняется группировка по IDзаявки (f_data130.id).

Для того, чтобы в дальнейшем использовать этот запрос и сделать его динамически обновляемым от года и категорий сотрудников, создадим на сервере класс Queries и реализуем в нем публичные методы для вложенных подзапросов и самого запроса.

Метод для поиска IDактивных менеджеров в системе приведен на рисунке 2.

```

public function people($categories){
    if ((is_array($categories)) && (count($categories) > 0)){
        $query = new Query();
        $result = [];
        $managers = $query->select('f1400 AS id')
            ->from('f_data46')
            ->where(['f555' => $categories])
            ->andWhere(['=' , 'status', 0])
            ->all();
        foreach($managers as $human){
            array_push($result, $human['id']);
        }
        return $result;
    }
}

```

Рисунок 2 – Метод поиска активных менеджеров в системе

Метод принимает на входе массив IDкатегорий сотрудников. Предполагается, что заявки могут вести не только менеджеры, но и руководящий персонал. На выходе метод возвращает ассоциативный массив, состоящий из IDсотрудников [4].

На рисунке 3 приведен метод поиска IDзакрытых заявок.

```

public function datalist($year, $statuses, $people){
    if ((is_array($statuses)) && (count($statuses) > 0)
        && isset($year) && (is_array($people)) && (count($people) > 0)){
        $query = new Query();
        $datalist = $query->select(['id'])
            ->from('f_data130')
            ->where(['>', 'add_time', strval($year - 1)])
            ->andWhere(['<', 'add_time', strval($year + 1)])
            ->andWhere(['f1680' => $statuses])
            ->andWhere(['f1740' => $people])
            ->andWhere(['=' , 'status', 0])
            ->all();
        $result = [];
        foreach($datalist as $item){
            array_push($result, $item['id']);
        }
        return $result;
    }
}

```

Рисунок 3 – Метод поиска ID закрытых заявок

Принимая на входе год точки отсчета, а также массивы закрывающих статусов и IDсотрудников, метод возвращает список IDзакрытых заявок в ведении активных менеджеров[5,6], созданных в течение обозначенного выше временного горизонта. Метод поиска ключевых дат по закрытым заявкам приведен на рисунке 4.

```

public function speedstats($people, $datalist){
    if ((is_array($people) && (count($people) > 0) && (is_array($datalist)
    && (count($datalist) > 0))){
        $query = new Query();
        $data = $query->select(['f_logs.line_id AS zayavka',
        'f_users.fio AS manager',
        'f_data130.f1680 AS status',
        'f_data130.add_time AS add_date',
        'Max(f_logs.date) AS close_date'])
        ->from('f_logs')
        ->innerJoin('f_data130', 'f_logs.line_id = f_data130.id')
        ->leftJoin('f_users', 'f_data130.f1740 = f_users.id')
        ->where(['=', 'f_logs.table_id', 130])
        ->andWhere(['f_logs.user_id' => $people])
        ->andWhere(['=', 'f_logs.event', 'change_field'])
        ->andWhere(['like', 'f_logs.text', "Статус" таблицы "Заявки"])
        ->andWhere(['f_logs.line_id' => $datalist])
        ->groupBy('f_logs.line_id')
        ->all();
        return $data;
    }
}

```

Рисунок 4 – – Метод поиска ключевых дат по заявкам

Таким образом, принимая на входе массивы IDсотрудников и IDзакрытых заявок, метод возвращает даты создания, закрытия и статус заявки.

```

public function actionSpeed($key, $platform){
    if (\Yii::$app->request->isGet &&
    $key === 'МОНИТОРИНГ' && $platform === 'МОНИТОРИНГ'){
        $cache = Yii::$app->cache;
        if (!$cache->get('speeds')){
            $year = intval(date('Y'));
            $statuses = ['Отказ', 'Для ознакомления', 'Успех', 'Передано партнеру'];
            $categories = [2, 781];
            $people = Queries::people($categories);
            $datalist = Queries::datalist($year, $statuses, $people);
            $speeds = Queries::speedstats($people, $datalist);
            $result = [];
            foreach($speeds as $item){
                $begin = date_create($item['add_date']);
                $end = date_create($item['close_date']);
                $period = date_diff($begin, $end);
                array_push($result, [
                    'id' => $item['zayavka'],
                    'manager' => $item['manager'],
                    'status' => $item['status'],
                    'add_time' => $item['add_date'],
                    'close_time' => $item['close_date'],
                    'duration' => $period->format('%a'),
                ]);
            }
        }
    }
}

```

Рисунок 5 – Метод сбора данных

Параметрически функция зависит от кеуи platform, которые скрыты из соображений безопасности. На выходе формируется массив заявок со всей ключевой информацией (в свойство linkкаждой строки передается ссылка на заявку в CRMдля удобства проверяющего). Для удобства обработки PowerВion отдается на вывод в JSON-формате[7,8]. Для сокращения объема запросов к БД CRMвыполняется кэширование выходного массива данных на 10 минут.

Библиографический список

1. Крамаров С.О., Храмов В.В.: Системно-инженерный подход к исследованиям сложных многомерных систем на основе мягких моделей // Интеллектуальные ресурсы – региональному развитию. – 2018. – Т.4, №1. – С.222-228.
2. Документация по Power BI. [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/power-bi/> (Дата обращения: 15.04.2021).
3. Магеррамов И.М. Моделирование принятия решений в условиях нечеткости требований // Интеллектуальные ресурсы – региональному развитию. – 2021. – №1. – С.157-163.
4. Yii Framework 2.0. API Documentation. YiiFramework. [Электронный ресурс]. – URL: <https://www.yiiframework.com/doc/api/2.0> (Дата обращения: 16.04.2021).
5. Храмов В.В., Гвоздев Д.С. Интеллектуальные информационные системы: интеллектуальный анализ данных: учебное пособие. – Ростов-на-Дону, 2012 – 98 с. – URL: <https://www.elibrary.ru/item.asp?id=32762296> (Дата обращения: 16.04.2021).
6. Храмов В.В. Моделирование на ЭВМ. – М.: МО РФ, 1992. – 98 с. – URL: <https://www.elibrary.ru/item.asp?id=34870614> (Дата обращения: 16.04.2021).
7. Храмов В.В. Теория информационных процессов и систем. Ростов-на-Дону: РГУПС, 2011.– 47 с. – URL: <https://www.elibrary.ru/item.asp?id=32764234> (Дата обращения: 16.04.2021).
8. Храмов В.В. Особенности мажоритарной обработки нечеткой информации // Спектральные методы обработки информации в научных исследованиях: Доклады I Всероссийской конференции (Спектр-2000). – 2000. – С. 136-138. – URL: <https://www.elibrary.ru/item.asp?id=32656899> (Дата обращения: 16.04.2021).