**Руководство разработчика**

# Требования к характеристикам компьютера и операционной системы

# Версия языка и библиотек

**Язык:**

Python 3.7

**Библиотеки:**

Numpy 1.16.4

Tkinter 8.6

Pandas 0.23.4

Matplotlib 3.1.0

# Архитектура приложения

# Описание структуры каталогов

**Work** <- основной каталог

**Data** – содержит базу данных

**Graphics** – содержит копии графических отчетов

**Library** – содержит библиотеку стандартных(универсальных) функций, разработанных бригадой, которые могут использоваться для создания других приложений

**Notes** – содержит документацию, в нем размещается Руководство пользователя и разработчика

**Output** – содержит копии текстовых отчетов

**Scripts** – содержит специализированный модуль и файл с определением параметров настройки приложения

# Структура приложения

Приложение состоит из 11 библиотек и главного скрипта.

Библиотеки (находятся в папке Library):

* main0.py — скрипт стартовой точкой программы - ее главном окном; также содержит функции удаления записей, классы главного окна приложения;
* adding\_window.py – содержит набор функций для добавления записей и прогнозирования отдельных полей при помощи алгоритмов линейной регрессии и случайного леса, класс всплывающего при добавлении окна с интерфейсом;
* changing\_window.py – содержит набор функций для редактирования определённого поля записи, а также класс всплывающего при редактировании окна с интерфейсом;
* searching\_window.py –содержит набор функций для поиска полей записей в определённом диапазоне, а также класс всплывающего при поиске окна с интерфейсом;
* sorting\_window.py – содержит набор функций для сортировки записей по определённому полю, а также класс всплывающего при сортировке окна с интерфейсом;
* backend.py – содержит функции обработки базы данных, без интегрирования в интерфейс приложения;
* static\_report\_window.py – скрипт, отображающий текстовый отчёт;
* text\_report\_window.py – скрипт, отображающий текстовый статистический отчёт;
* wiscker.py – скрипт, отображающий диаграмму Бокса-Вискера;
* grad.py – скрипт, отображающий диаграмму рассеивания;
* writing\_file\_window.py – скрипт, преобразующий базу данных в csv-файл;

Скрипт:

* main.pyw – этот скрипт является стартовой точкой программы - ее главном окном

backend.py

|  |  |  |  |
| --- | --- | --- | --- |
| Название функции | Входные параметры | Выходные параметры | Описание |
| make\_DataFrame | DataFrame | DataFrame, состоящий из одной строки | Пользователь вводит значения столбцов по очереди, а если хочет, чтобы 2 последних столбца спрогнозировались автоматически, то выбирает данную опцию. |
| appending | DataFrame | DataFrame с добавленной строкой | В функции вызывается make\_DataFrame, после чего ее выходной параметр добавляется в главный DataFrame |
| class\_balancing | DataFrame | DataFrame с уравновешенным число классов | Считается число значений у разных классов и берется максимум из них. К классам добавляется столько элементов, сколько не хватает до максимального значения |
| classificator | DataFrame, вектор признаков | Предсказание класса | После балансировки классов происходит обучение модели классификации и после выводится предсказания для входного вектора |
| regression\_weights | Матрица признаков, вектор ответов | Веса регрессии | Считаются веса линейной регрессии |
| predict | Матрица признаков, веса регресси | предсказание |  |
| score | Массив предсказанных значений, массив реальных значений | Точность предсказаний | Суммируется число верных соответствий и делится на количество элементов |
| deleting\_by\_numbers | DataFrame | DataFrame без выбранных элементов | Удаляются строки, номера которых введет пользователь |
| deleting\_by\_range | DataFrame | DataFrame без выбранных элементов | Удаляются строки в диапазоне от начального до конечного значения, введенного пользователем |
| sort\_descending / sort\_ascending | DataFrame | Отсортированный DataFrame | Сортировка по атрибуту, введенному пользователем. Сортировка по убыванию / возрастанию |
| Searching | DataFrame | Под - DataFrame | Пользователь вводит диапазон значений и имена столбцов |
| change | DataFrame | DataFrame с измененным значением | Вводится номер строки и имя столбца, а также новое значение |
| writing\_to\_csv | DataFrame | None | Запись DataFrame в csv файл |

# Технические требования

ПК под управлением Windows XP и выше, ИЛИ MacOS Leopard и выше, ИЛИ Linux. Также требуется минимум 1,5 ГГц тактовой частоты процессора и разрешение экрана 1280x720 и выше. На ПК должен быть установлен дистрибутив Python 3 (Anaconda или другой)