Руководство разработчика к программе для построения модели множественной регрессии с Web реализацией пользовательского интерфейса

Разработчики:

Подчезерцев А.Е.

Солодянкин А.А.

Оглавление

[Структура исходных файлов 3](#_Toc515391998)

[Структура каталогов приложения 4](#_Toc515391999)

[Структура приложения 5](#_Toc515392000)

[Стандартные функции разработчика 6](#_Toc515392001)

[Функции скрипта reglib.py (входит в библиотеку разработчика) 6](#_Toc515392002)

[Функции скрипта weblib.py (входит в библиотеку разработчика) 6](#_Toc515392003)

[Функции скрипта index.py 7](#_Toc515392004)

[Программные требования 8](#_Toc515392005)

[Аппаратные требования 9](#_Toc515392006)

[Приложение 10](#_Toc515392007)

[Листинг скрипта reglib.py 10](#_Toc515392008)

[Листинг скрипта weblib.py 12](#_Toc515392009)

[Листинг скрипта index.py 14](#_Toc515392010)

# Структура исходных файлов

Исходные данные, загружаемые в программу, могут иметь любое имя и расширение и находиться в любом месте на компьютере. В первой строчке должны находиться названия столбцов, указывающих какие данные хранятся в нем. Названия не должны содержать пробелов и символов разделителей. В последующих строчках находятся сами данные, сепарированные разделителем. В каждой строчке количество данных должно совпадать с общим числом столбцов. Формат разделителя для дробных чисел – точка. Символ разделитель данных выбирает пользователь, рекомендуется ставить точку с запятой. Формат конца строки – любой – Unix, Mac или Windows, кодировка - Unicode.

После загрузки файла через форму данные помещаются в файл Work/Data/.temp, после чего файл проверяется и преобразуется с форматом разделителя запятая, конец строки – Windows и сохраняется с именем равным количеству секунд, прошедших с начала эпохи Unix, и расширением “.csv”.

# Структура каталогов приложения

Основной каталог решения – Work, внутри которого находятся следующие подкаталоги:

* Data – предобработанные и тестовые датасеты;
* Graphics – шаблоны страниц;
* Library – библиотека универсальных функций;
* Notes – руководства пользователя и разработчика;
* Output – выходные данные приложения;
* Scripts – скрипты для запуска приложения.

В каталоге Data автоматически сохраняются загруженные датасеты. Кроме того, предоставлено несколько тестовых для демонстрации работы.

# Структура приложения

Файлы reglib.py и weblib.py в директории Library содержит методы, которые в дальнейшем можно использовать в других проектах. В частности, первая библиотека позволяет загружать датасет с диска, обрабатывать файл, полученный от клиента, вычислять значение регрессии, а второй обрабатывает параметры web сервера.

Скрипты папки Scripts выполняют следующие функции:

* index.py – методы для обработки web запросов;
* run.py – запускает web сервер;
* settings.py – конфигурация сервера;
* urls.py – правила направления web запроса к необходимому обработчику;
* wsgi.py – интерфейс взаимодействия программы и web сервера.

Настройки сервера находятся в файле Scripts/settings.conf, который отвечает за IP адрес и порт прослушивания. Остальные параметры можно установить в файле settings.py.

# Стандартные функции разработчика

## Функции скрипта reglib.py (входит в библиотеку разработчика)

|  |  |
| --- | --- |
| Заголовок функции | Документация |
| **def** load\_dataset(name, sep=**','**, end=**None**) | *Загружает датасет с диска* **:param** *name: Имя файла* **:param** *sep: Формат разделителя столбцов, по умолчанию ','* **:param** *end: Формат разделителя строк, по умолчанию '\n'* **:return***: Датасет Автор: Подчезерцев А.Е.* |
| **def** handle\_uploaded\_file(f, sep=**';'**, end=**None**) | *Проверяет загруженный файл* **:param** *f: Структура файла* **:param** *sep: Разделитель символов* **:param** *end: Разделитель строк* **:return***: Адрес для перенаправления: имя загруженного файла в случае успеха или '/' в обратном Автор: Подчезерцев А.Е.* |
| **def** handle\_dataset(dataset, result, line, square) | *Вычисляет параметры регрессии* **:param** *dataset: Датасет* **:param** *result: Номер результирующей переменной* **:param** *line: Логический массив переменных, которые должны быть обработаны линейно* **:param** *square: Логический массив переменных, которые должны быть обработаны квадратично* **:return***: R квадрат и данные регрессии для переменных Автор: Солодянкин А.А.* |

## Функции скрипта weblib.py (входит в библиотеку разработчика)

|  |  |
| --- | --- |
| Заголовок функции | Документация |
| **def** validate\_port(port) | *Проверяет является ли исходная строка валидным портом* **:param** *port: Строка для проверки* **:return***: Логический результат проверки Автор: Подчезерцев А.Е.* |
| **def** validate\_ip(ip) | *Проверяет является ли исходная строка валидным ip адресом* **:param** *ip: Строка для проверки* **:return***: Логический результат проверки Автор: Подчезерцев А.Е.* |
| **def** load\_config(path=**"Scripts/settings.conf"**) | *Загружает настройки с диска, если они с ошибкой, то применит стандартные* **:param** *path: Путь к файлу настроек* **:return***: ip адрес и порт, на котором необходимо запустить сервер Автор: Подчезерцев А.Е.* |

## Функции скрипта index.py

|  |  |
| --- | --- |
| Заголовок функции | Документация |
| **def** upload\_file(request) | *Обрабатывает web запрос загрузки файла* **:param** *request: Web-запрос* **:return***: Перенаправление при отправке файла, иначе страницу Автор: Солодянкин А.А.* |
| **def** calculate(request, num) | *Обрабатывает web запрос вычисления регрессии* **:param** *request: Web-запрос* **:param** *num: номер файла* **:return***: Данные регрессии при отправке запроса, иначе страницу Автор: Солодянкин А.А.* |

## 

# Программные требования

Для корректной работы приложения необходимо установить следующие библиотеки:

* Django – web framework;
* NumPy – математические функции;
* Pandas – обработка и анализ данных;
* Statsmodels – статистические исследования;

# Аппаратные требования

Любая система (32- или 64- битная/ARM и другие), на которую можно установить Python 3.5+ и необходимые библиотеки (https://www.python.org/downloads/)

# Приложение

## Листинг скрипта reglib.py

*# -\*- coding: utf-8 -\*-  
"""  
Стандартная библиотека для обработки данных  
  
Данный файл сожержит методы для обработки данных и датасетов  
  
Авторы: Подчезерцев А.Е.  
 Солодянкин А.А.  
"""***from** time **import** time  
  
**import** pandas **as** pd  
**from** statsmodels.formula.api **import** ols  
  
DATASET\_PATH = **"Data/"  
  
  
def** load\_dataset(name, sep=**','**, end=**None**):  
 *"""  
 Загружает датасет с диска* **:param** *name: Имя файла* **:param** *sep: Формат разделителя столбцов, по умолчанию ','* **:param** *end: Формат разделителя строк, по умолчанию '\n'* **:return***: Датасет  
 Автор: Подчезерцев А.Е.  
 """* **return** pd.read\_csv(name, sep=sep, lineterminator=end, encoding=**'utf-8'**)  
  
  
**def** handle\_uploaded\_file(f, sep=**';'**, end=**None**):  
 *"""  
 Проверяет загруженный файл* **:param** *f: Структура файла* **:param** *sep: Разделитель символов* **:param** *end: Разделитель строк* **:return***: Адрес для перенаправления: имя загруженного файла в случае успеха или '/' в обратном  
 Автор: Подчезерцев А.Е.  
 """* filename = str(round(time())) + **".csv"  
 if** end **is None**:  
 end = **'\n'  
 with** open(DATASET\_PATH + **".temp"**, **'wb'**) **as** destination:  
 **for** chunk **in** f.chunks():  
 destination.write(chunk)  
 **try**:  
 dataset = load\_dataset(DATASET\_PATH + **".temp"**, sep, end)  
 dataset.to\_csv(DATASET\_PATH + filename, encoding=**'utf-8'**)  
 **except** Exception:  
 **return "/"  
 return** filename  
  
  
**def** handle\_dataset(dataset, result, line, square):  
 *"""  
 Вычисляет параметры регрессии* **:param** *dataset: Датасет* **:param** *result: Номер результирующей переменной* **:param** *line: Логический массив переменных, которые должны быть обработаны линейно* **:param** *square: Логический массив переменных, которые должны быть обработаны квадратично* **:return***: R квадрат и данные регрессии для переменных  
 Автор: Солодянкин А.А.  
 """* statement = dataset.columns[result] + **" ~ "** variables = []  
 **for** i **in** range(len(line)):  
 **if** square[i]:  
 dataset = dataset.assign(cp=**lambda** x: x[dataset.columns[i]] \*\* 2)  
 dataset.rename(columns={**"cp"**: dataset.columns[i] + **"Квадрат"**}, inplace=**True**)  
 variables.append(dataset.columns[i] + **"Квадрат"**)  
 **if** line[i]:  
 variables.append(dataset.columns[i])  
 statement += **" + "**.join(variables)  
 model = ols(statement, dataset).fit()  
 **return** model.rsquared, [model.params.to\_dict(),  
 model.bse.to\_dict(),  
 model.tvalues.to\_dict(),  
 model.pvalues.to\_dict(), ]

## Листинг скрипта weblib.py

*# -\*- coding: utf-8 -\*-  
"""  
Библиотека функций загрузки и проверки настроек  
  
Автор: Подчезерцев А.Е.  
"""***def** validate\_port(port):  
 *"""  
 Проверяет является ли исходная строка валидным портом* **:param** *port: Строка для проверки* **:return***: Логический результат проверки  
 Автор: Подчезерцев А.Е.  
 """* **if** port.isdecimal():  
 port = int(port)  
 **return** 1 < port < 65536  
 **else**:  
 **return False  
  
  
def** validate\_ip(ip):  
 *"""  
 Проверяет является ли исходная строка валидным ip адресом* **:param** *ip: Строка для проверки* **:return***: Логический результат проверки  
 Автор: Подчезерцев А.Е.  
 """* byte = ip.split(**"."**)  
 **if** len(byte) == 4:  
 **for** b **in** byte:  
 **if** b.isdecimal():  
 b = int(b)  
 **if not** 0 <= b <= 255:  
 **break  
 else**:  
 **break  
 else**:  
 **return True  
 return False  
  
  
def** load\_config(path=**"Scripts/settings.conf"**):  
 *"""  
 Загружает настройки с диска, если они с ошибкой, то применит стандартные* **:param** *path: Путь к файлу настроек* **:return***: ip адрес и порт, на котором необходимо запустить сервер  
 Автор: Подчезерцев А.Е.  
 """* **with** open(path, **"r"**) **as** f:  
 **for** line **in** f.readlines():  
 **try**:  
 ip, port = line.strip().split(**":"**)  
 **if** validate\_ip(ip) **and** validate\_port(port):  
 **return ":"**.join((ip, port))  
 **except** ValueError:  
 **pass  
 except** Exception **as** e:  
 **raise** e  
 **return "0.0.0.0:8080"**

## Листинг скрипта index.py

*# -\*- coding: utf-8 -\*-  
"""  
Библиотека функций обработчиков web запросов  
  
Данный файл сожержит методы для обработки данных и запросов web сервера  
  
Автор: Солодянкин А.А.  
"""***from** django.http **import** JsonResponse  
**from** django.shortcuts **import** render, HttpResponseRedirect  
**from** django.views.decorators.csrf **import** csrf\_exempt  
  
**from** Library.reglib **import** \*  
  
  
@csrf\_exempt  
**def** upload\_file(request):  
 *"""  
 Обрабатывает web запрос загрузки файла* **:param** *request: Web-запрос* **:return***: Перенаправление при отправке файла, иначе страницу  
 Автор: Солодянкин А.А.  
 """* **if** request.method == **'POST'**:  
 end = **None  
 if** request.POST[**'end'**] == **'LF'**:  
 end = **'\n'  
 elif** request.POST[**'end'**] == **'CR'**:  
 end = **'\r'  
 elif** request.POST[**'end'**] == **'CRLF'**:  
 end = **'\r\n'** file = handle\_uploaded\_file(request.FILES[**'file'**], request.POST[**'sep'**], end)  
 **return** HttpResponseRedirect(file)  
 **return** render(request, **'index.html'**)  
  
  
@csrf\_exempt  
**def** calculate(request, num):  
 *"""  
 Обрабатывает web запрос вычисления регрессии* **:param** *request: Web-запрос* **:param** *num: номер файла* **:return***: Данные регрессии при отправке запроса, иначе страницу  
 Автор: Солодянкин А.А.  
 """* dataset = load\_dataset(DATASET\_PATH + str(num) + **".csv"**)  
 **if** request.method == **"POST"**:  
 result = **None** line = [**False for** \_ **in** range(len(dataset.columns))]  
 square = [**False for** \_ **in** range(len(dataset.columns))]  
 **for** key, value **in** request.POST.items():  
 **if** key == **'result'**:  
 result = dataset.columns.get\_indexer\_for([value])[0]  
 **elif** key.startswith(**"variable-line-"**):  
 line[dataset.columns.get\_indexer\_for([value])[0]] = **True  
 elif** key.startswith(**"variable-square-"**):  
 square[dataset.columns.get\_indexer\_for([value])[0]] = **True  
 if** result **is None or** (**True not in** line **and True not in** square):  
 **return** JsonResponse({**'status'**: **False**})  
 r, k = handle\_dataset(dataset, result, line, square)  
 **for** col **in** k:  
 **for** key, value **in** col.items():  
 **if** value == float(**"inf"**):  
 col[key] = **"inf"  
 elif** value != value:  
 col[key] = **"nan"  
 return** JsonResponse({**'status'**: **True**, **'r'**: r, **'k'**: k})  
  
 **return** render(  
 request,  
 **'calculate.html'**,  
 {**'num'**: num, **'dataset'**: dataset, **'head'**: dataset.columns.tolist()[1:]}  
 )