

11 Руководство пользователя

11.1 Аннотация

Настоящий раздел содержит назначение, условия применения и описание возможностей программного обеспечения «WhoseCppCode». Предъявлены требования к квалификации пользователя. Указана последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведены сообщения, выводимые программой.

11.2 Назначение программы

Программное обеспечение (ПО) «WhoseCppCode» предназначено для определения авторства программ на языке C/C++ по исходному коду и может быть использовано организациями, занимающимися решением вопросов информационной безопасности, лицензирования ПО, интеллектуальной собственности и расследования инцидентов, связанных с применением вредоносного ПО. Программа состоит из программного модуля, реализующего заявленный функционал, и интерфейса, предназначенного для визуализации ввода и вывода данных, а также удобной работы с возможностями основного модуля. При этом интерфейс не является обязательным, программа может быть использована в качестве модуля при разработке иных автоматизированных систем.

ПО «WhoseCppCode» предоставляет следующие возможности:

- обработка файлов исходного кода на языке C/C++;
- сбор данных с ресурса GitHub;
- построение модели классификации авторов программного обеспечения;
- формирование отчетности в форматах *.json и *.csv;
- визуализация результатов классификации.

Работа с ПО «WhoseCppCode» доступна всем пользователям с доступом к предварительно установленной и настроенной рабочей программной среде, реализованной на ПЭВМ, специально предназначенном сервере или с помощью средств виртуализации.

11.3 Уровень подготовки пользователя

Пользователь ПО «WhoseCppCode» должен иметь пользовательский опыт работы с ОС Linux, а также обладать следующими знаниями:

- знать соответствующую предметную область;
- понимать основы машинного обучения, построения и оценки моделей классификации.

Квалификация пользователя должна позволять осуществлять сбор и анализ данных.

11.4 Условия выполнения программы

Для работы с ПО «WhoseCppClassCode» необходимо следующее программное обеспечение:

- 64-разрядная ОС Linux (тестирование программы производилось на ОС Ubuntu 17.04) с доступом к глобальной сети Интернет;
- программная среда с установленными зависимостями (библиотеками);
- веб-обозреватель.

Требования к характеристикам автоматизированного рабочего места:

- 2,00 Гб ОЗУ или выше;
- CD-ROM;
- 1 Гб на жестком диске.

11.5 Выполнение программы

Для начала работы с программой необходимо открыть в веб-обозревателе адрес сервера JupyterNotebook, перейти в меню «Cell» и запустить выполнение ячеек командой «Run All», после чего дождаться загрузки всех ячеек (рис. 11.1).

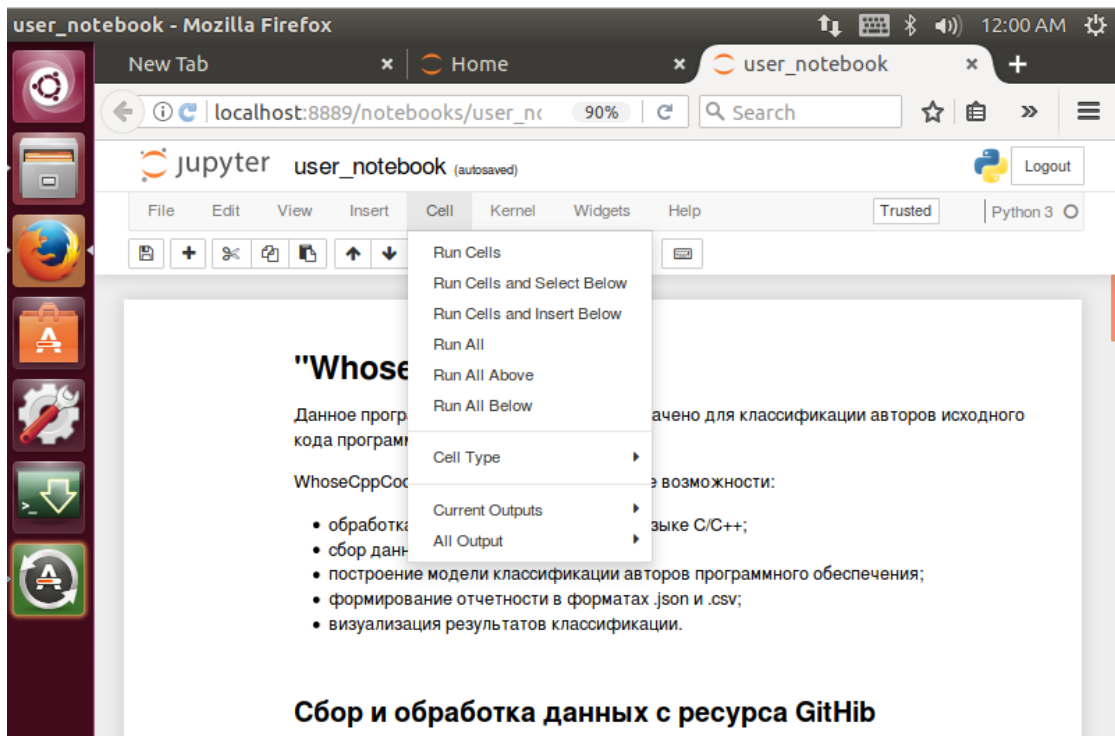


Рисунок 11.1 — Запуск программы

11.5.1 Сбор и обработка данных с ресурса GitHub

Форма, приведенная на рисунке 11.2, предназначена для сбора файлов исходного кода на языке C/C++ с веб-хостинга GitHub. При этом необходимо иметь аутентификационные данные, соответственно, пользователь должен быть зарегистрирован на сайте www.github.com. Помимо логина и пароля в форму через запятую вводится список пользователей, чьи файлы будут загружены и обработаны.

В процессе работы данной формы файлы указанных пользователей загружаются в корень проекта в директорию «data», все файлы конвертируются в кодировку UTF-8, удаляются пустые. Подобная обработка необходима для корректной дальнейшей работы с загруженными файлами.

▼ **Сбор и обработка данных с ресурса GitHub**

Введите список пользователей через запятую.

Удаление пустых файлов и изменение кодировки.

```
In [1]: from scrap_widget import display_scrapping_form
display_scrapping_form()
```

✕

Логин:	<input type="text" value="IvanovIvan"/>
Пароль:	<input type="text" value="duh79@I"/>
Найти:	<input type="text" value="paroj, Soyen, sipa, gavinandresen, theuni, luke-jr, ddunbar"/>

✓ Получить данные

Идет сбор файлов, пожалуйста, подождите...

Готово. Данные расположены в корне проекта в папке data.

Если данные не были загружены, проверьте правильность ввода логина, пароля, а также имен пользователей.

Рисунок 11.2 — Пример работы с формой «Сбор и обработка данных»

11.5..2 Вычисление матрицы объектов-признаков

На рисунке 11.3 приведен пример работы с формой, предназначенной для формирования на основе загруженных данных матрицы-объектов признаков, подаваемой в последствии на вход алгоритму классификации. В данном случае под объектами подразумеваются авторы программ на языке C/C++, каждому из которых соответствует вектор стилистических признаков, вычисленных на основе представленных файлов исходного кода.

Матрица объектов-признаков вычисляется отдельно во избежание повторения вычислений и для экономии времени, затрачиваемого на работу программы, поскольку процесс классификации предполагает несколько циклов вычислительных экспериментов, включающих обучение и тестирование классификатора.

▼ Матрица объектов-признаков

Формирование матрицы объектов-признаков для дальнейшей классификации. Объектами в данном случае являются авторы исходного кода, признаками - вычисленный для каждого автора на основе набора файлов исходного кода вектор значений признаков, характеризующих индивидуальный стиль разработчика.

Матрица объектов-признаков вычисляется отдельно во избежание повторения вычислений, а также сокращения времени, затрачиваемого на обучение и тестирование классификатора.

```
In [2]: from sample_matrix_widget import display_matrix_widget

# Путь к данным
path = './data/'
outpath = './data/matrices/'

display_matrix_widget(path, outpath)
```

✕ ☒ Получить матрицу

Готово.

Рисунок 11.3 — Пример работы с формой «Матрица объектов-признаков»

11.5.3 Классификация

Основной функцией пользовательского интерфейса ПО «WhoseCppClassCode» является демонстрация процесса классификации авторов исходного кода программ на языке C/C++ (рис. 11.4) на заранее сформированных наборах данных:

- «students» — лабораторные работы студентов ТУСУР кафедры КИБЭВС 1-го курса обучения по дисциплине «Основы программирования»;
- «Google Code Jam 2016» — работы участников ежегодной олимпиады по программированию от компании Google;
- «GitHub» — данные с веб-хостинга www.github.com.

Существует возможность классификации пользовательского набора данных. Для этого необходимо сформировать и обработать данные при помощи форм, описанных в разделах 11.5.1 и 11.5.2, после чего в выпадающем меню «Данные» выбрать опцию «user_data» (рис. 11.5). При этом на вход классификатору будет подаваться сформированная пользователем матрица объектов-признаков, расположенная в корне проекта в директории data/matrices.

Все наборы данных, а также список GitHub-пользователей, чьи программы подвергались классификации, прилагаются к ПО «WhoseCppClassCode».

▼ Классификация

Построение модели классификации, ее обучение на выбранном наборе данных, визуализация результатов.

Отчеты по результатам работы программы в форматах .json и .csv располагаются в директории results в корне проекта.

Ввод:

- **Циклов** - количество итераций эксперимента
- **Данные** - данные для классификации:
 - students - лабораторные работы студентов каф. КИБЭВС по дисциплине "Основы программирования"
 - Google Code Jam 2016 - работы участников ежегодной олимпиады по программированию от компании Google
 - GitHub - данные с веб-хостинга GitHub
- **Алгоритм** - алгоритм классификации

```
In [3]: from main_widget import display_main_form  
display_main_form()
```

× Циклов:

Данные:

Алгоритм:

✓ Классифицировать

Рисунок 11.4 — Пример работы с формой «Классификация»

В форме «Классификация» существует возможность выбора числа циклов (количество итераций вычислительных экспериментов), набора данных, на которых будет производиться процесс классификации (рис. 11.5), алгоритма классификации (рис. 11.6).

jupyter user_notebook (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

• Данные - данные для классификации:

- students - лабораторные работы студентов каф. КИБЭВС по дисциплине "Основы программирования"
- Google Code Jam 2016 - работы участников ежегодной олимпиады по программированию от компании Google
- GitHub - данные с веб-хостинга GitHub

• Алгоритм - алгоритм классификации

```
In [3]: from main_widget import display_main_form  
display_main_form()
```

× Циклов:

Данные:

Алгоритм:

✓ Классифицировать

Рисунок 11.5 — Выбор набора данных для дальнейшей классификации

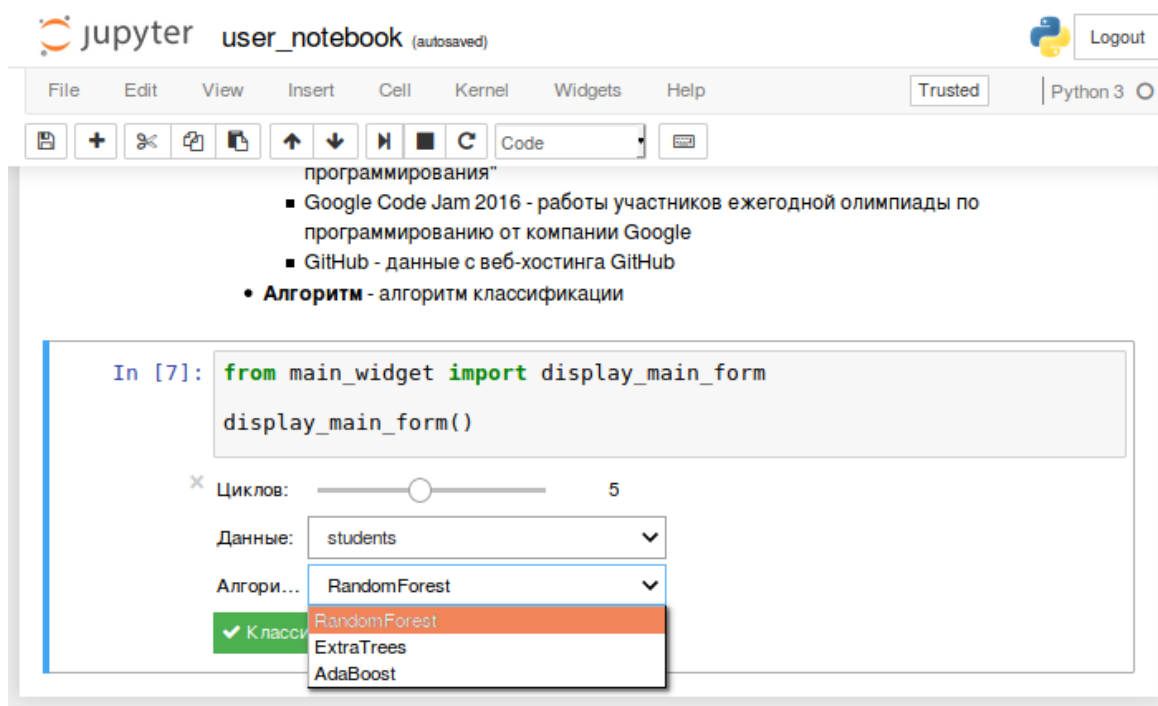


Рисунок 11.6 — Выбор алгоритма классификации

На рисунках 11.7, 11.8 и 11.9 представлен вывод результатов работы программы. Полученные диаграммы можно экспортировать в формате *.png с помощью команды «Download plot as png», как показано на рисунке 4.10.

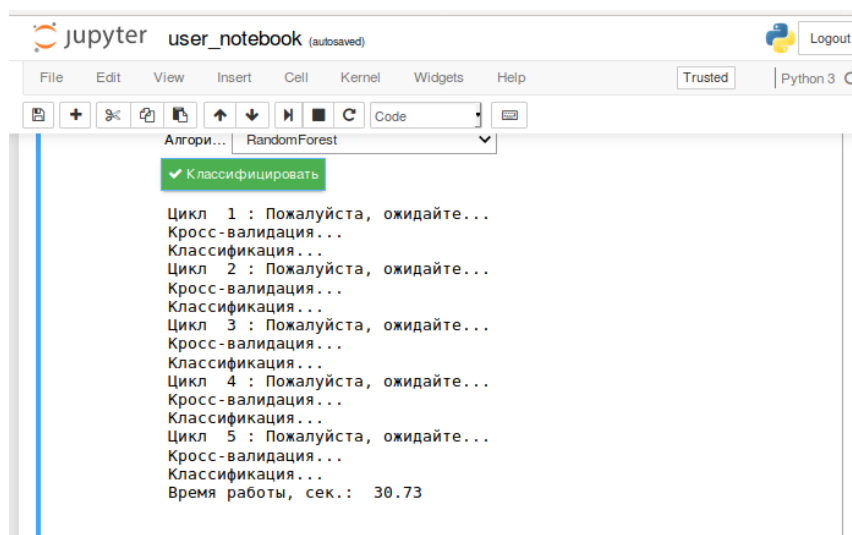


Рисунок 11.7 — Вывод программы в процессе классификации

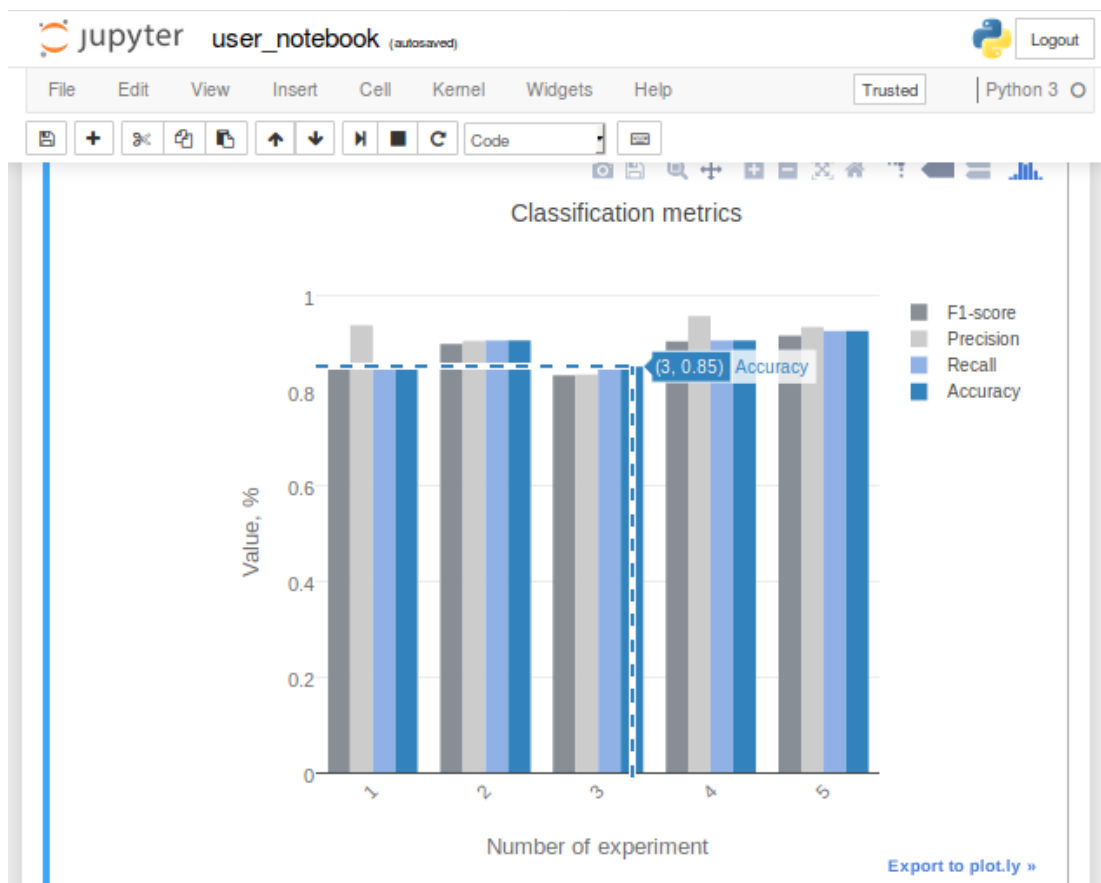


Рисунок 11.8 — Диаграмма значений метрик классификации

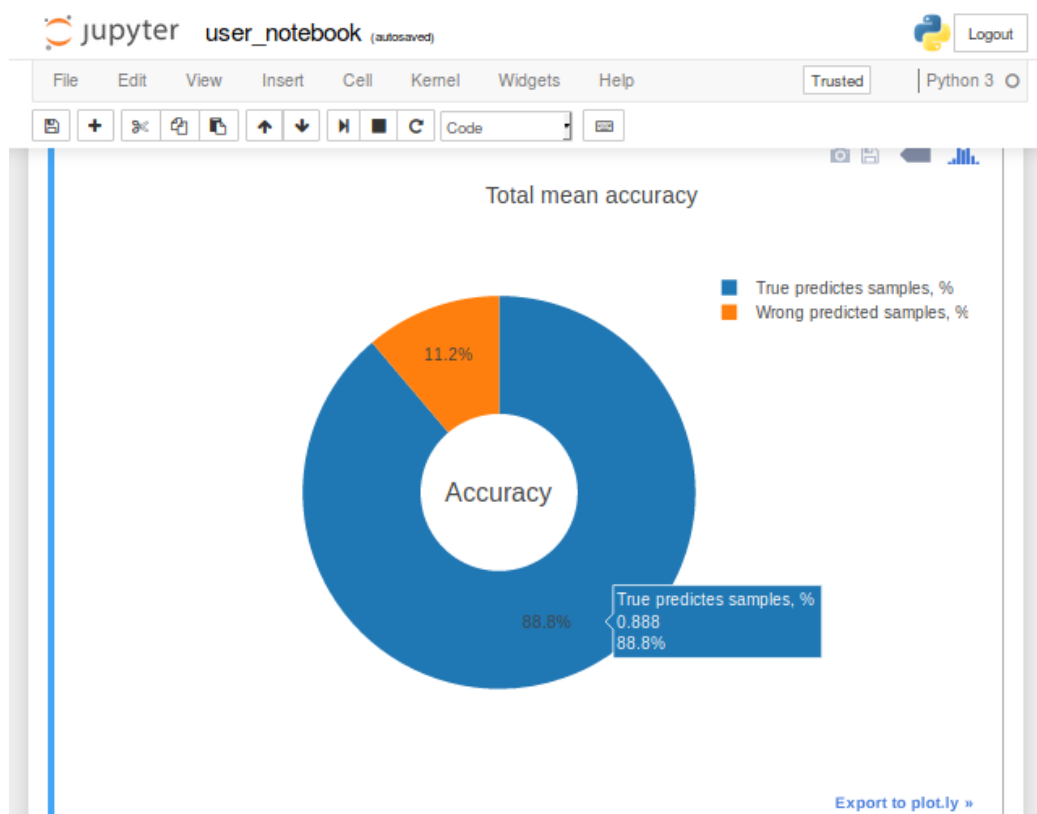


Рисунок 11.9 — Диаграмма значения средней точности классификации

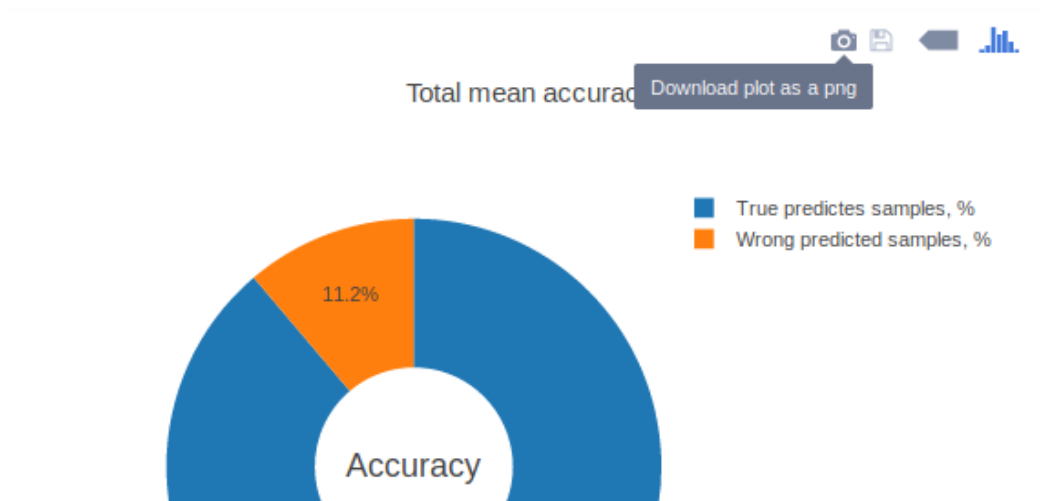


Рисунок 11.10 — Сохранение диаграммы в формате *.png

11.6 Сообщения пользователю

Выводимые сообщения и соответствующие действия пользователя:

— «Пожалуйста, введите данные для аутентификации на сайте www.github.com». Заполнены не все поля формы для сбора данных, необходимо ввести логин и пароль GitHub-пользователя.

— «Идет сбор файлов, пожалуйста, подождите...». Сообщение о запуске процесса сбора и обработки данных с веб-хостинга www.github.com.

— «Готово. Данные расположены в корне проекта в папке data. Если данные не были загружены, проверьте правильность ввода логина, пароля, а также имен пользователей.». Завершение сбора и обработки данных с веб-хостинга www.github.com. При отсутствии результатов работы программы в указанных директориях, следует проверить корректность ввода логина и пароля для аутентификации на сайте www.github.com, а также имен GitHub-пользователей, чьи файлы будут подвержены сбору и обработке.

— «Готово». Завершение процесса преобразования данных в матрицу объектов-признаков.

— «Цикл 1: Пожалуйста, ожидайте... Кросс-валидация... Классификация...». Вывод во время процесса классификации для контроля за выполнением программы.

— «Время работы, сек.: 9.42». Классификация успешно завершена, вывод суммарного времени работы в секундах.

Для повторного запуска выполнения любой из форм программ, необходимо нажать на нужную ячейку щелчком мыши, после чего — комбинацию клавиш «Ctrl + Enter» и продолжить работу с формой. При сбое в работе программы необходимо перезапустить все ячейки, перейдя в меню «Cell Run — All», дождаться загрузки ячеек. Если программа по-прежнему работает некорректно, необходимо обратиться к программисту, ответственному за установку и настройку программы.