10 Руководство программиста

10.1 Аннотация

Настоящий раздел содержит назначение, условия применения, характерные особенности и описание возможностей программного обеспечения «WhoseCppCode». Предъявлены требования к квалификации программиста. Указана последовательность действий программиста, обеспечивающих установку, настройку, загрузку, запуск, выполнение и завершение программы, приведено описание входных и выходных данных, а также сообщения, выводимые программой, и соответствующие им действия.

10.2 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.

10.3 Уровень подготовки программиста

Программист, использующий ПО «WhoseCppCode», должен иметь опыт работы с ОС Linux, базовые навыки программирования, а также обладать следующими знаниями:

– знать соответствующую предметную область;

– понимать основы машинного обучения, построения и оценки моделей классифи-кации.

Квалификация программиста должна позволять осуществлять установку и настройку программной среды для работы системы, а также сбор и анализ данных.

10.4 Условия применения программы

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

– ОС Linux (тестирование программы производилось на ОС Ubuntu 17.04) с доступом к глобальной сети Интернет;

– программная среда с установленными зависимостями (библиотеками);

– веб-обозреватель.

Инструкция по установке и настройке программной среды приведена в разделе 10.6 настоящего документа.

Требования к характеристикам автоматизированного рабочего места:

– 2,00 Гб ОЗУ или выше;

– CD-ROM;

– 1 Гб на жестком диске.

10.5 Характеристика программы

ПО «WhoseCppCode» состоит из двух основных частей:

– программного модуля, реализующего все необходимые функции для сбора, анализа и обработки данных, а также построения модели классификации авторов исходного кода

– программного интерфеса на основе технологии Jupyter Notebook, предназначенного для визуализации полученных в ходе классификации результатов, сбора необходимых данных с ресурса GitHub, построения матрицы объектов-признаков на основе входных данных, проведения вычислительных экспериментов.

Интерфейс основан на веб-технологиях, может использоваться для демонстрации возможностей программ на языке Python. ПО «WhoseCppCode» использует ряд сторонних библиотек, реализующих алгоритмы классификации, используемых в процессе классификации авторов исходного кода, обеспечивающих работу интерфейса, а также визуализацию полученных результатов. Подробное описание особенностей программы приведено в пояснительной записке к ПО «WhoseCppCode».

Основной модуль программы «WhoseCppCode» может быть использован отдельно от Jupyter Notebook при разработке различного рода программ, систем и интерфейсов лицами, заинтересованными в задаче классификации программистов.

Время выполнения программы зависит от вычислительных мощностей автоматизированного рабочего места, количества данных, подвергаемых обработке, а также количества циклов работы классификатора (число итераций вычислительных экспериментов), задаваемых пользователем.

10.6 Инструкция по установке и настройке программной среды

Ниже приведена последовательность терминальных команд по установке и настройке ПО «WhoseCppCode» в ОС Ubuntu:

1) Перейти в директорию проекта:

$ cd whose_cpp_code

2) Установить Python версии 3.6:

$ sudo apt-get install python3.6

3) Создать виртуальную оболочку, в которую будут установлены необходимые зависимости проекта:

$ virtualenv --python=python3.6

4) Активировать виртуальную оболочку:

$ source whose_cpp_code/bin/activate

5) Установить необходимые зависимости:

$ pip install -r requirements.txt

6) Запустить локальный сервер Jupyter Notebook:

$ jupyter notebook

После запуска локального сервера в веб-обозревателе автоматически открывается страница проекта. Необходимо запустить файл user_notebook.ipynb (рис. 10.1).

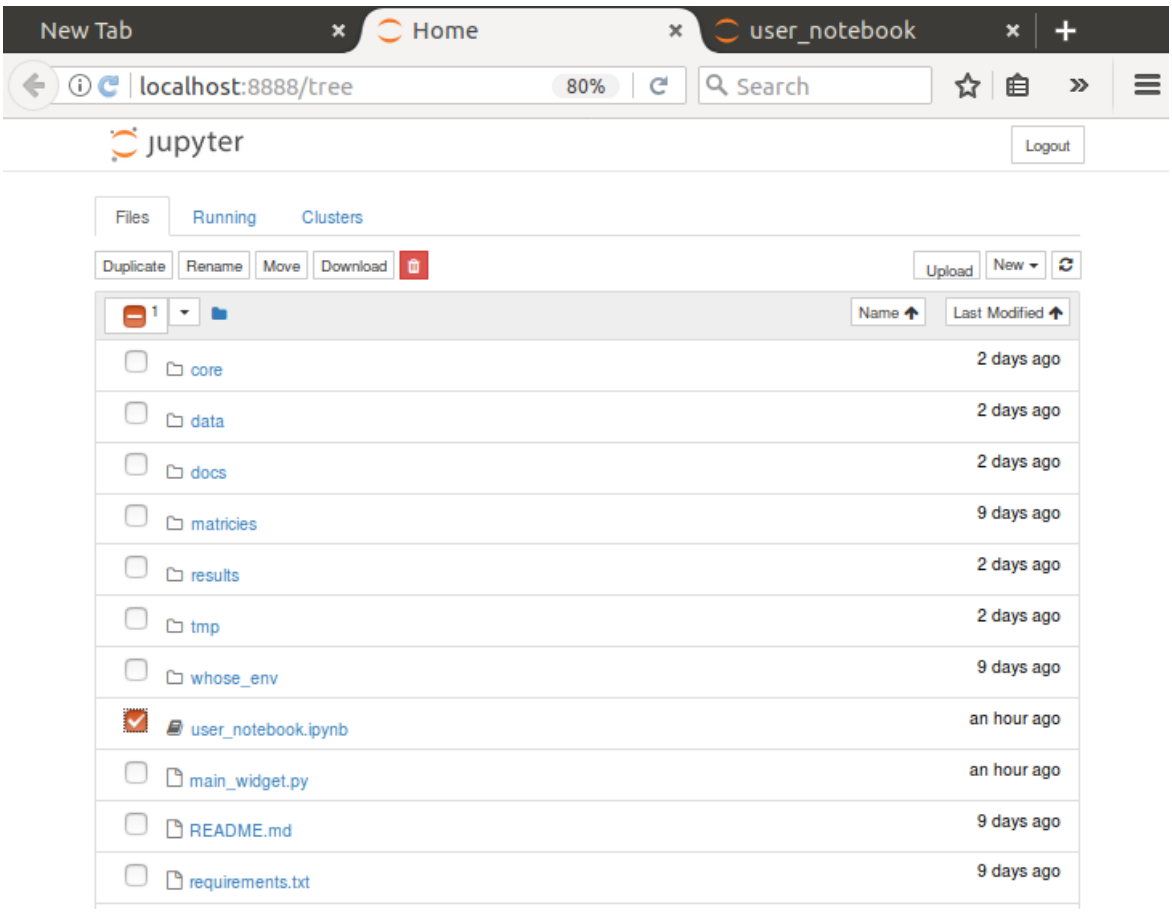

Рисунок 10.1 — Главное окно Jupyter Notebook

10.7 Обращение к программе

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. 10.2).

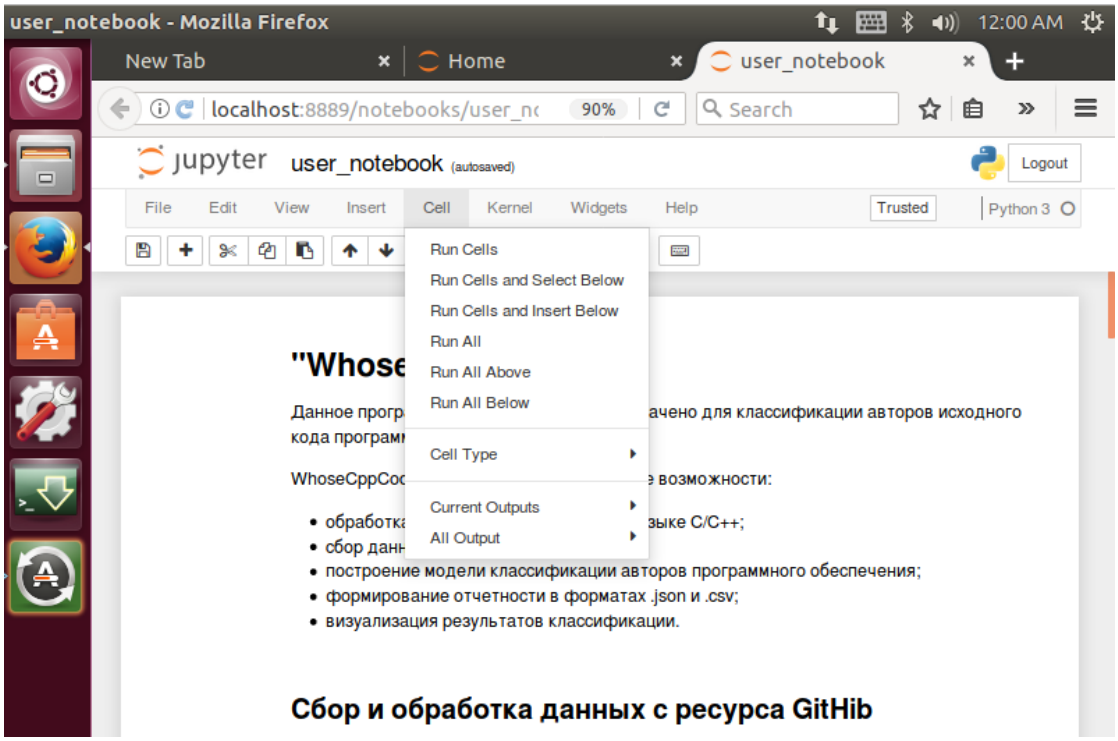

Рисунок 10.2 — Запуск программы

10.7.1 Сбор и обработка данных с ресурса GitHub

Форма, приведенная на рисунке 10.3, 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.

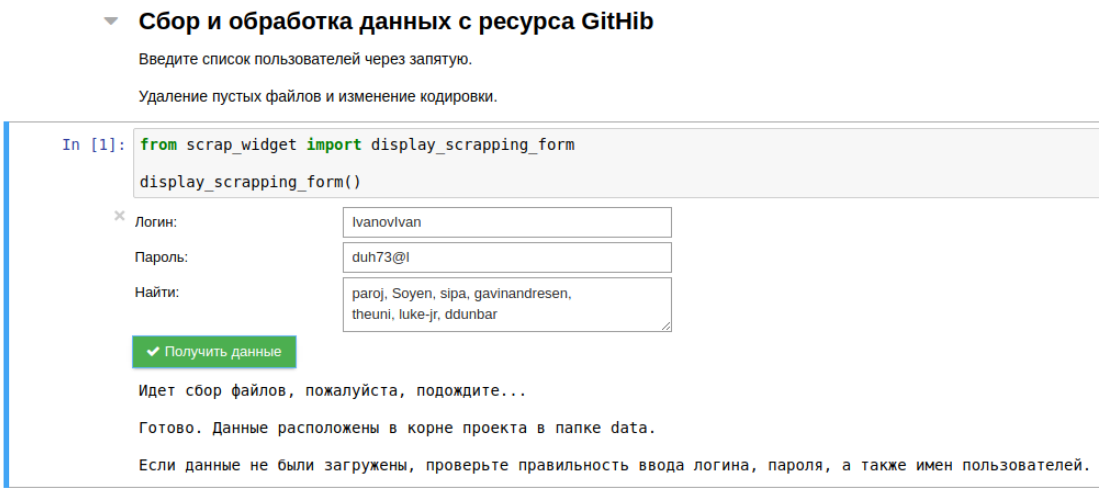

Рисунок 10.3 — Пример работы с формой «Сбор и обработка данных»

10.7.2 Вычисление матрицы объектов-признаков

На рисунке 10.4 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.

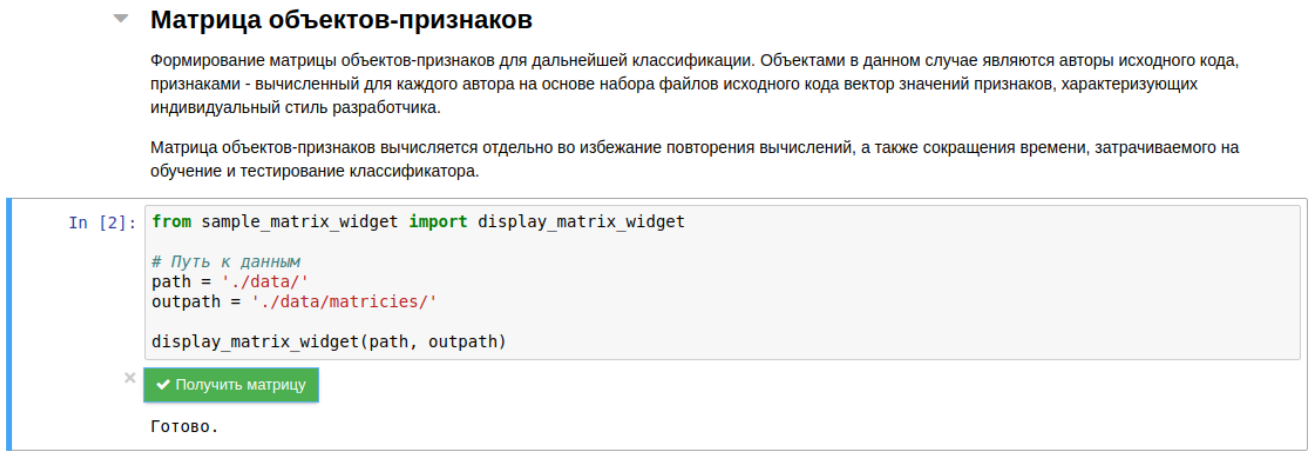

Рисунок 10.4 — Пример работы с формой «Матрица объектов-признаков»

10.7.3 Классификация, входные данные и вывод программы

Основной функцией пользовательского интерфейса ПО «WhoseCppCode» является демонстрация процесса классификации авторов исходного кода программ на языке C/C++ (рис. 6.4) 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 программи-рованию 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 10.7.1 и 10.7.2, после чего в выпадающем меню «Данные» выбрать опцию «user_data» (рис. 10.5). 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 «WhoseCppCode».

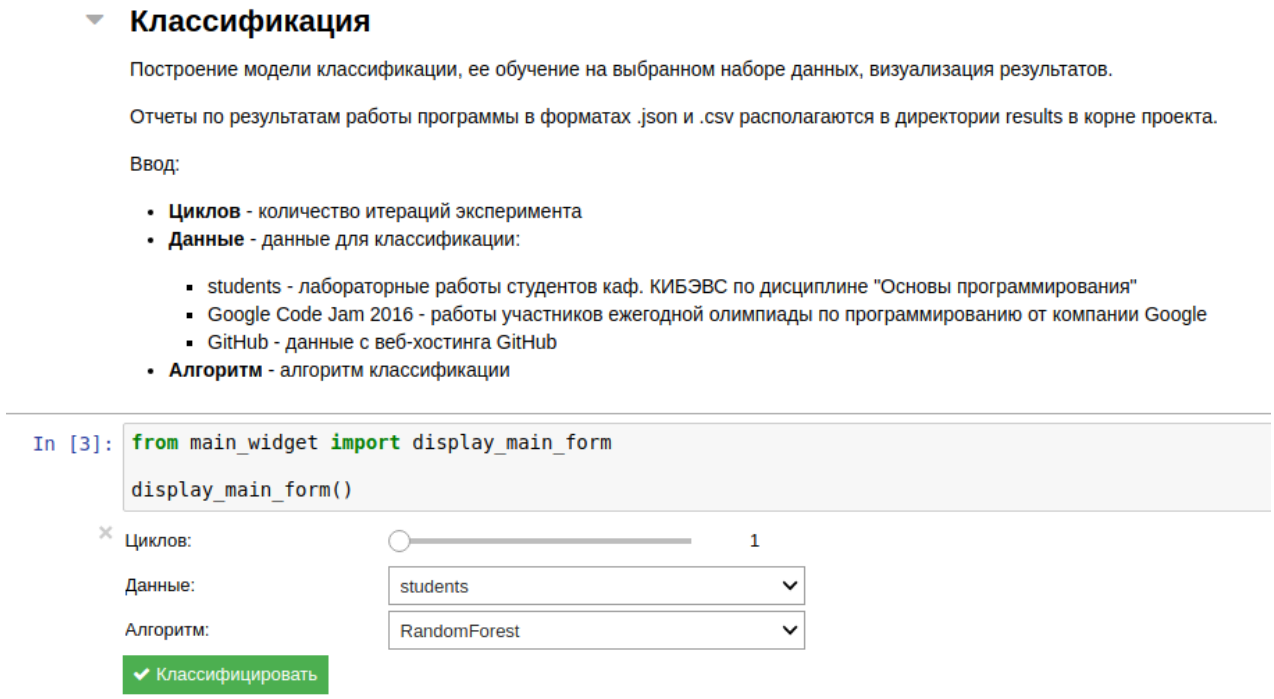

Рисунок 10.5 — Пример работы с формой «Классификация»

В форме «Классификация» существует возможность выбора числа циклов (количество итераций вычислительных экспериментов), набора данных, на которых будет производиться процесс классификации (рис. 10.6), алгоритма классификации (рис. 10.7).

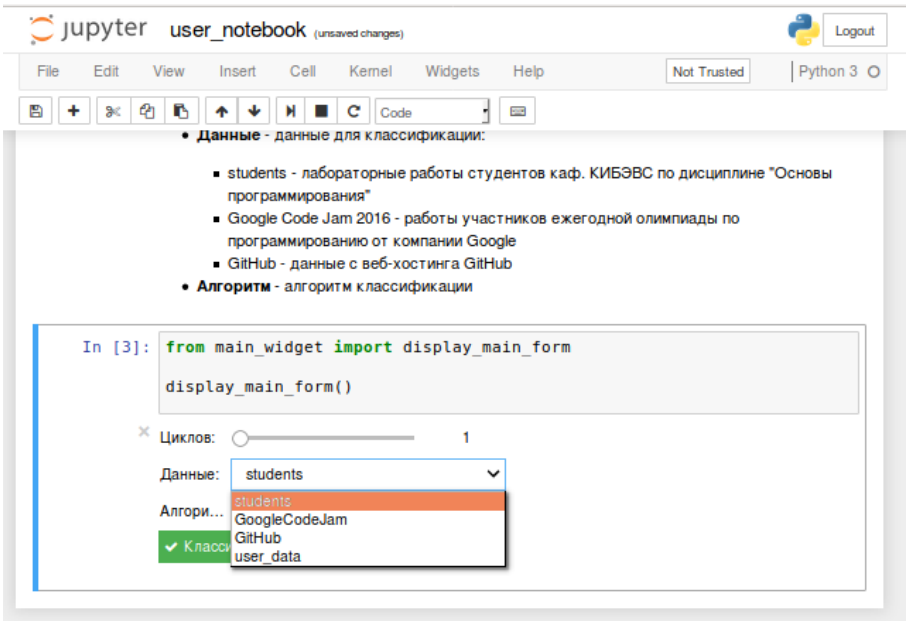

Рисунок 10.6 — Выбор набора данных для дальнейшей классификации

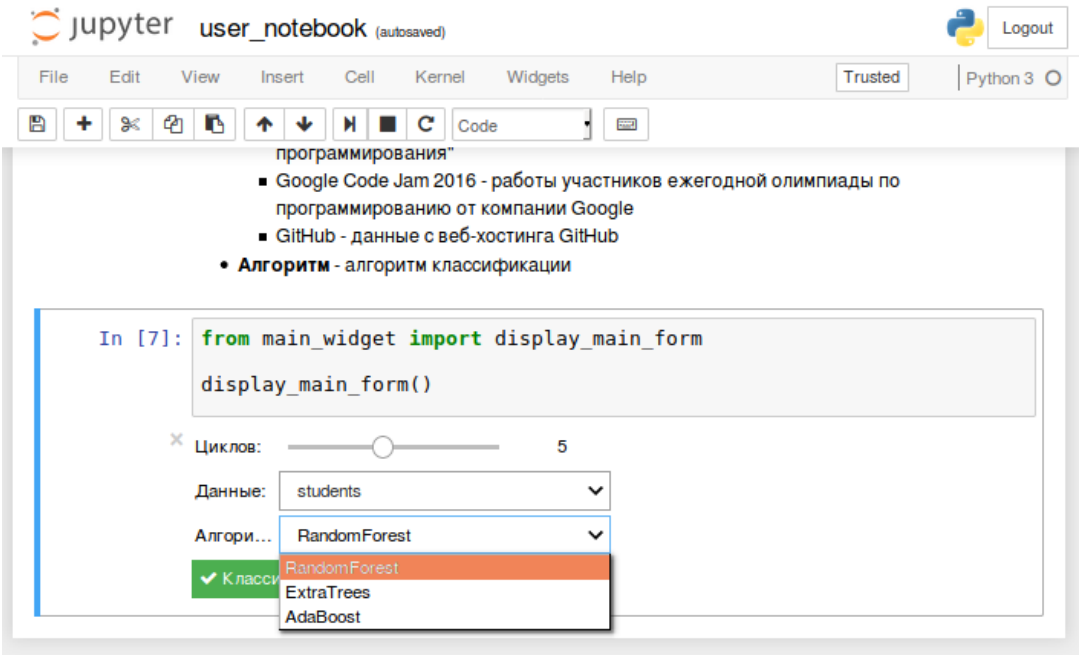

Рисунок 10.7 — Выбор алгоритма классификации

На рисунках 10.8, 10.9 и 10.10 представлен вывод результатов работы программы. Полученные диаграммы можно экспортировать в формате *.png с помощью команды «Download plot as png», как показано на рисунке 10.11.

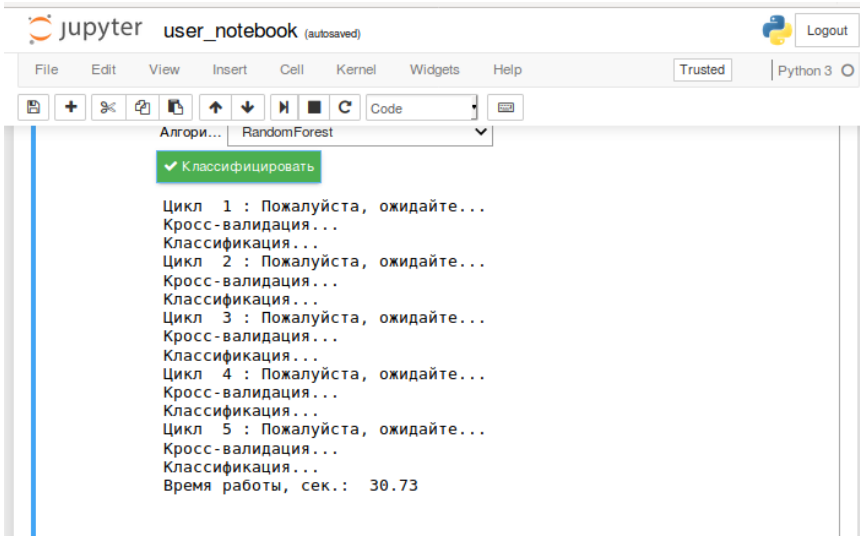

Рисунок 10.8 — Вывод программы в процессе классификации

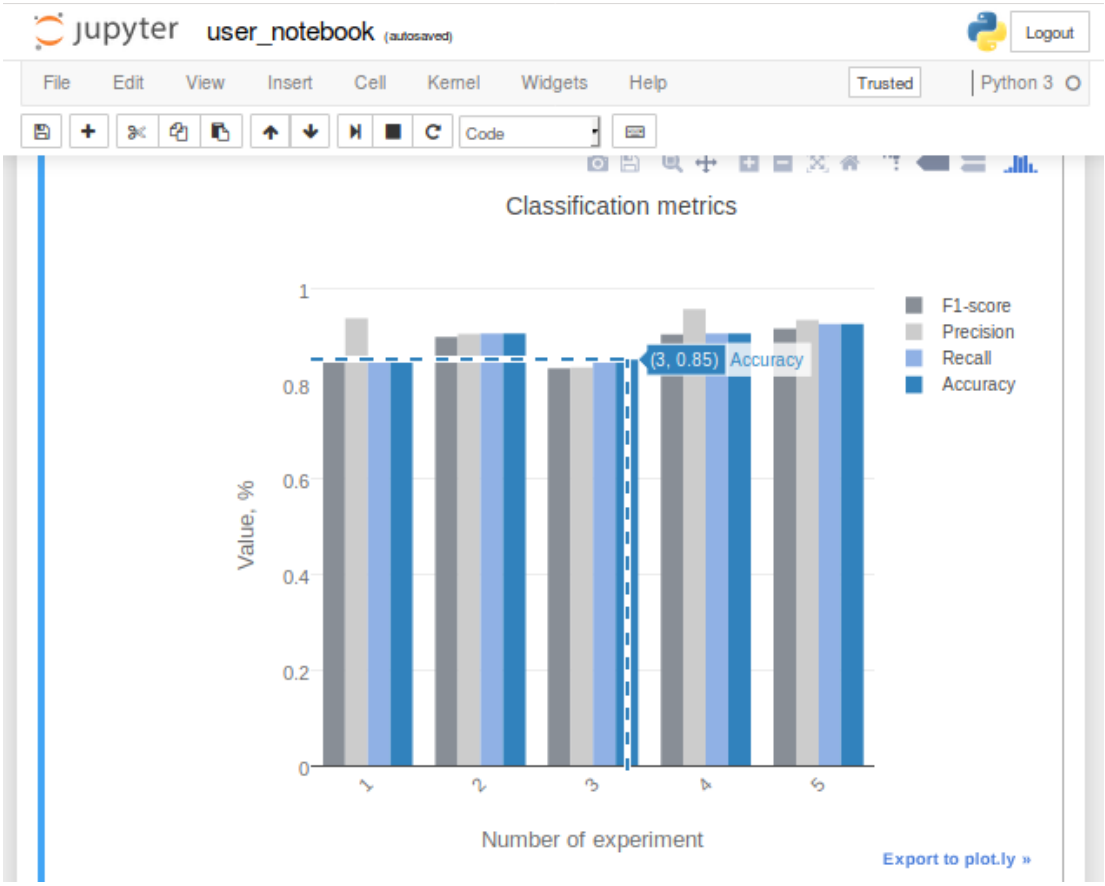

Рисунок 10.9 — Диаграмма значений метрик классификации

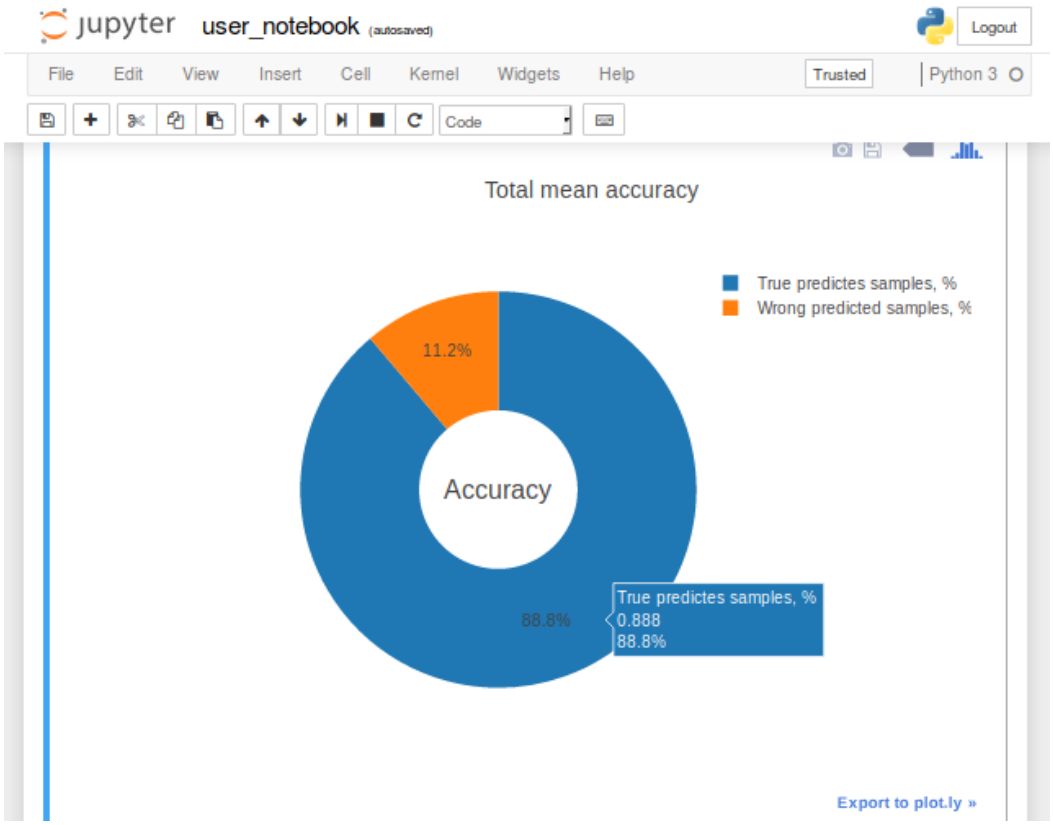

Рисунок 10.10 — Диаграмма значения средней точности классификации

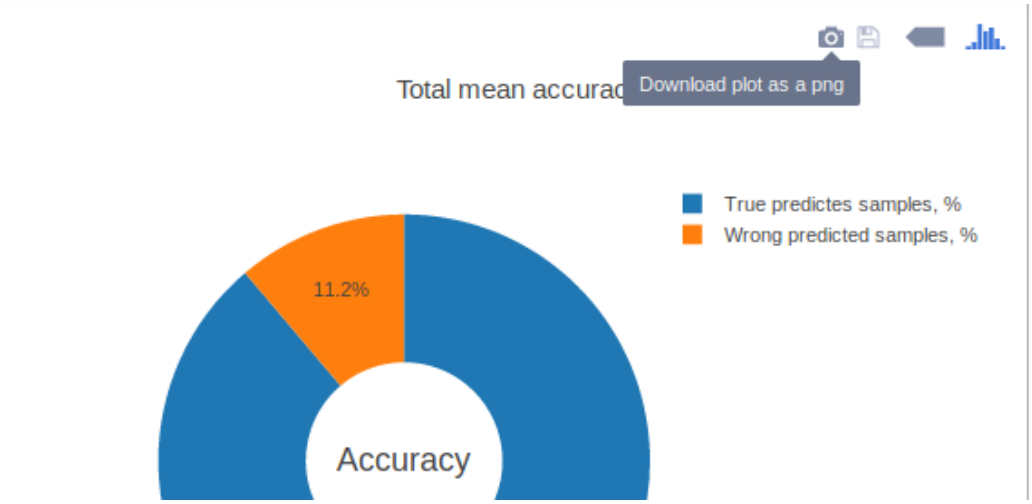

Рисунок 10.11 — Сохранение диаграммы в формате *.png

10.8 Сообщения

Выводимые сообщения и соответствующие им действия:

– «Пожалуйста, введите данные для аутентификации на сайте www.github.com». Запол-нены 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-призна-ков.

– «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.