Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский технологический университет
«МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)
Кафедра Автоматизированных Систем Управления (АСУ)

ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ

По теме: “Домашняя работа №1”

Выполнил:
студент группы БИВТ-20-1
Янушка А.В.
Проверил:
доцент кафедры АСУ
Громов С.В.

МОСКВА 2022

Задачи работы:

1. Создать репозиторий на github
2. Залить туда jupyter тетрадку, которая воспроизводит любую картинку со страницы определенного сайта (http://scikit learn.org/stable/tutorial/statistical_inference/supervised_learning.html)

Используемые технологии:

✔ GitHub - веб-сервис для хостинга IT проектов и их совместной разработки, основанный на системе контроля версий Git.

Создатели называют GitHub “социальной сетью для разработчиков”, так как помимо размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых.

С помощью широких возможностей Git программисты могут объединять свои репозитории - GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева. Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок.

Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

Можно создавать приватные репозитории, которые будут видны только вам и выбранным вами людям.

Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.

Код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта.

Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.

На сайте есть pastebin-сервис gist.github.com для быстрой публикации

фрагментов кода.

✔ Jupyter - это среда разработки, где сразу можно видеть результат выполнения кода и его отдельных фрагментов. Отличие от традиционной среды разработки в том, что код можно разбить на куски и выполнять их в произвольном порядке. Jupyter notebook, помимо поддержки работы на Python, поддерживает работу с такими языками программирования, как Ruby, Perl, R, bash-скрипты, используя для этого специальные magic-команды.

Также, в Jupyter notebook есть вывод результата сразу после фрагмента кода. ✔

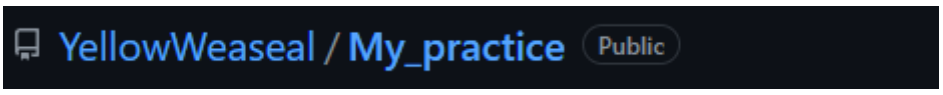
Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нем программ.

Описание решения задач и использованных подходов:

1. Создание репозитория на github:

Для создания нового репозитория на странице github перешёл во вкладку “Your repositories” и нажал кнопку “new”:

После чего ввел название репозитория “my_practice” и нажал кнопку “Create repository”. Таким образом был создан новый репозиторий “my_practice” куда я и залью jupyter notebook:



2. Запись кода для воспроизведения картинки с сайта:

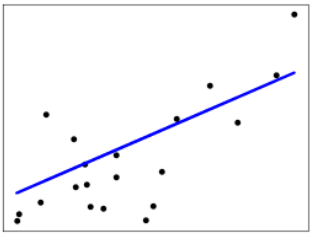
С сайта было выбрано данное изображение:

Linear regression

`LinearRegression`, in its simplest form, fits a linear model to the data set by adjusting a set of parameters in order to make the sum of the squared residuals of the model as small as possible.

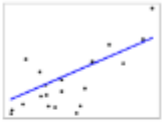
Linear models: $y = X\beta + \epsilon$

- X: data
- y: target variable
- β: Coefficients
- ϵ: Observation noise



И далее следовало написать код для его получения:

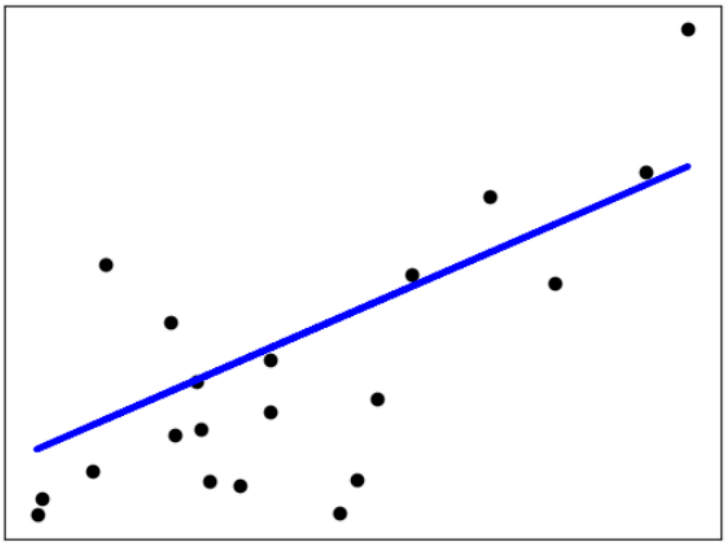
В первом случае, нам необходим url адрес изображения, который мы можем легко найти на самом сайте:



Rendered size: 320 × 240 px
Rendered aspect ratio: 4:3
Intrinsic size: 640 × 480 px
Intrinsic aspect ratio: 4:3
File size: 4.0 kB
Current source: https://scikit-learn.org/stable/_images/sphx_glr_plot_ols_001.png

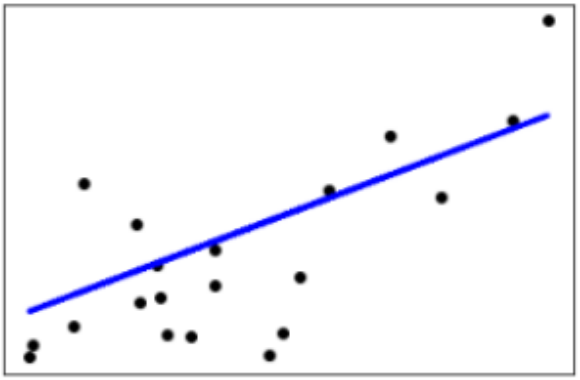
После чего был написан код, который выводил изображение с сайта по его адресу:

```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "https://scikit-learn.org/stable/_images/sphx_glr_plot_ols_001.png")
```



Во втором случае необходимо углубиться в код для построения точной копии изображения без знания его url:

```python
#импортируем модуль matplotlib.pyplot под именем plt
import matplotlib.pyplot as plt
#аналогично
import numpy as np
#из библиотеки sklearn импортируем модули datasets, linear_model
from sklearn import datasets, linear_model
#аналогично
from sklearn.metrics import mean_squared_error, r2_score

# загружаем набор данных по диабету
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

# Use only one feature
diabetes_X = diabetes_X[:, np.newaxis, 2]

# разделяем данные на обучающие/тестовые
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]

# создаем объект линейной регрессии
regr = linear_model.LinearRegression()

# обучение модели с помощью обучающего набора
regr.fit(diabetes_X_train, diabetes_y_train)

# Делаем прогнозы, используя тестовый набор
diabetes_y_pred = regr.predict(diabetes_X_test)

# строим точечный график
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
# строим линию среднего значения шириной 3
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)

#устанавливаем тики и метки оси x так, чтобы их не было видно
plt.xticks(())
#аналогично для оси y
plt.yticks(())

#отображаем все открытые фигуры
plt.show()
```



Далее объяснение написанного:

✔ *import* - инструкция, осуществляющая поиск модулей в списке путей, содержащемся в переменной sys.path (sys - встроенный модуль Python, который содержит переменные и методы, которые взаимодействуют с интерпритатором и также управляются им). Таким образом, когда используется модуль, Python запускает код из файла модуля. В коде использовались:

● *from Pack import Func* - делает *Func* доступной напрямую из *Pack*; ● *Import Pack as Name* - инструкция *as* переименовывает *Pack* в *Name* в пределах файла скрипта.

✔ *datasets.load_diabetes()* - загружает и возвращает набор данных о диабете; ✔ *numpy.newaxis* - используется для увеличения размерности существующего массива на одно измерение, при однократном использовании;

✔ *array1 = array2[:-number]* - *array1* сохраняет в себе все строки *array2* за вычетом последних *number* строк;

✔ *sklearn.linear_model.LinearRegression()* - создает объект линейной регрессии;

✔ Метод *".fit(x, y)"* метода *".LinearRegression()"* вычисляет оптимальные значения весов b_0 и b_1, используя существующие вход и выход (x и y) в качестве аргумента. Иными словами, совмещает модель;

✔ Применение *".predict()"* передаёт регрессор в качестве аргумента и получает соответствующий предсказанный ответ;

✔ *pyplot.scatter(x,y)* - строит точечную диаграмму y против x с возможностью изменения размера маркера и/или цвета;

✔ *pyplot.plot()* - строит x против y в виде линии и/или маркеров; ✔ *pyplot.xticks() / pyplot.yticks()* - получение или установка текущих местоположений тиков и меток оси X / оси Y;

✔ *pyplot.show()* - отображение всех открытых фигур.

3. Загрузка jupyter тетрадки с кодом на GitHub:

Далее я применил следующую последовательность действий, описывая с самого начала:

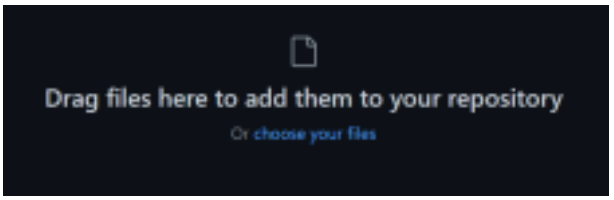
✔ Создать репозиторий на GitHub (в моем случае с именем "my_practice" уже был создан);

✔ Написание кода в jupyter тетрадке (ранее написан);

✔ Сохраняем написанный код в файле с расширением ".ipynb". В моем случае,

“1pr.ipynb”;

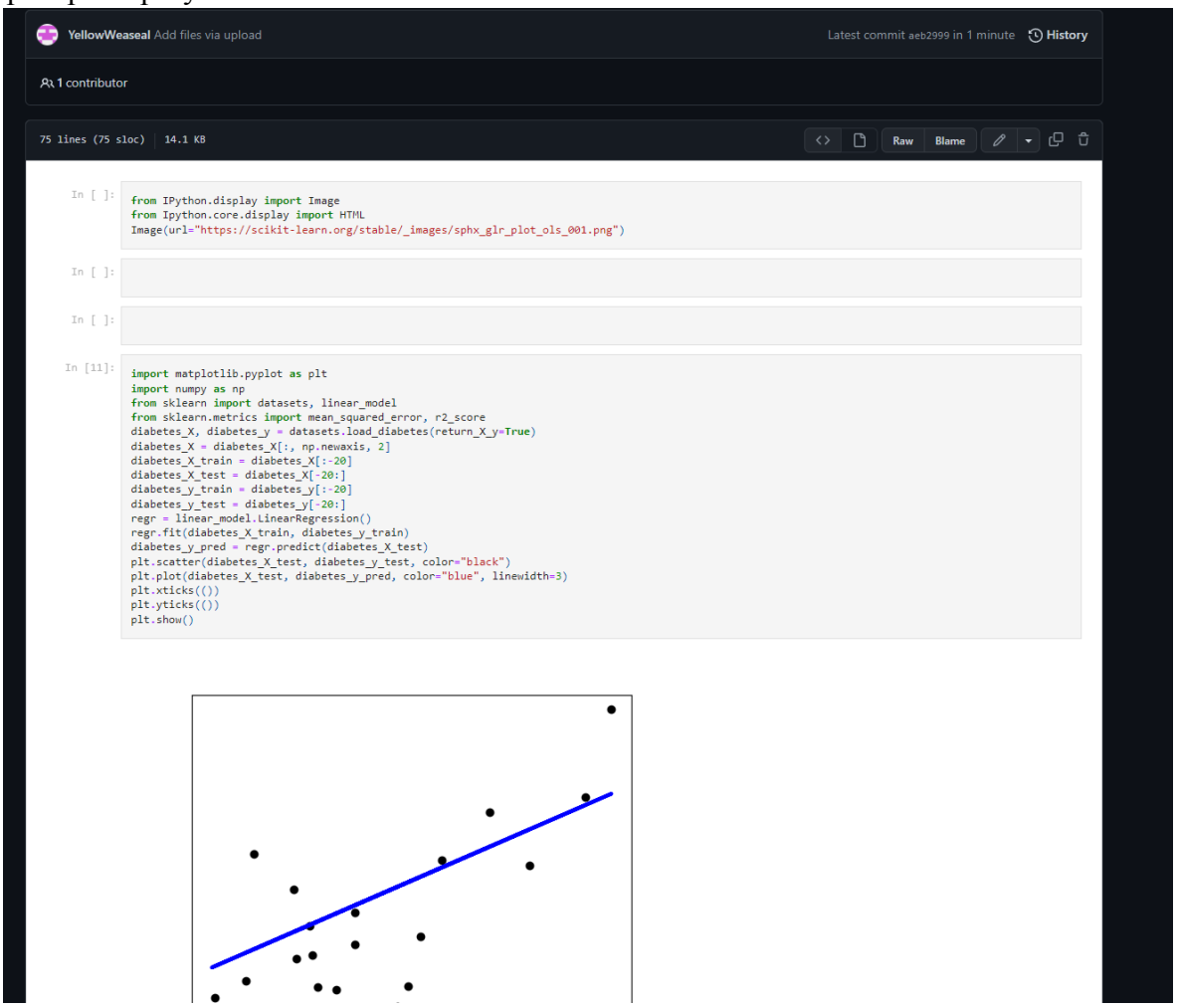
✔ Заходим на GitHub в репозиторий для проекта и нажимаем “Добавить файл”:



Загружаем “1pr.ipynb”;

✔ Коммитим;

✔ Проверяем результат:



4. Приложение: код программы:

import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear_model

from sklearn.metrics import mean_squared_error, r2_score

diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

```
diabetes_X = diabetes_X[:, np.newaxis, 2]
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
regr = linear_model.LinearRegression()
regr.fit(diabetes_X_train, diabetes_y_train)
diabetes_y_pred = regr.predict(diabetes_X_test)
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)
plt.xticks(())
plt.yticks(())
plt.show()
```

Заключение:

В ходе выполнения данного задания были поставлены и достигнуты задачи. Я вспомнил что такое GitHub и как с ним работать. Узнал про программное обеспечение jupyter и как с ним работать. Получил практические навыки пользования системой контроля версий.