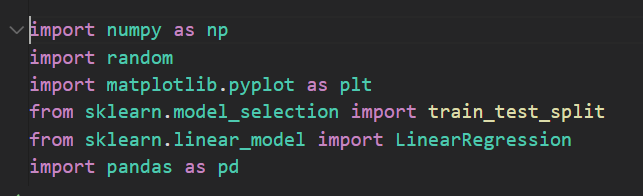
Долгинцев Степан КТСО-03-20

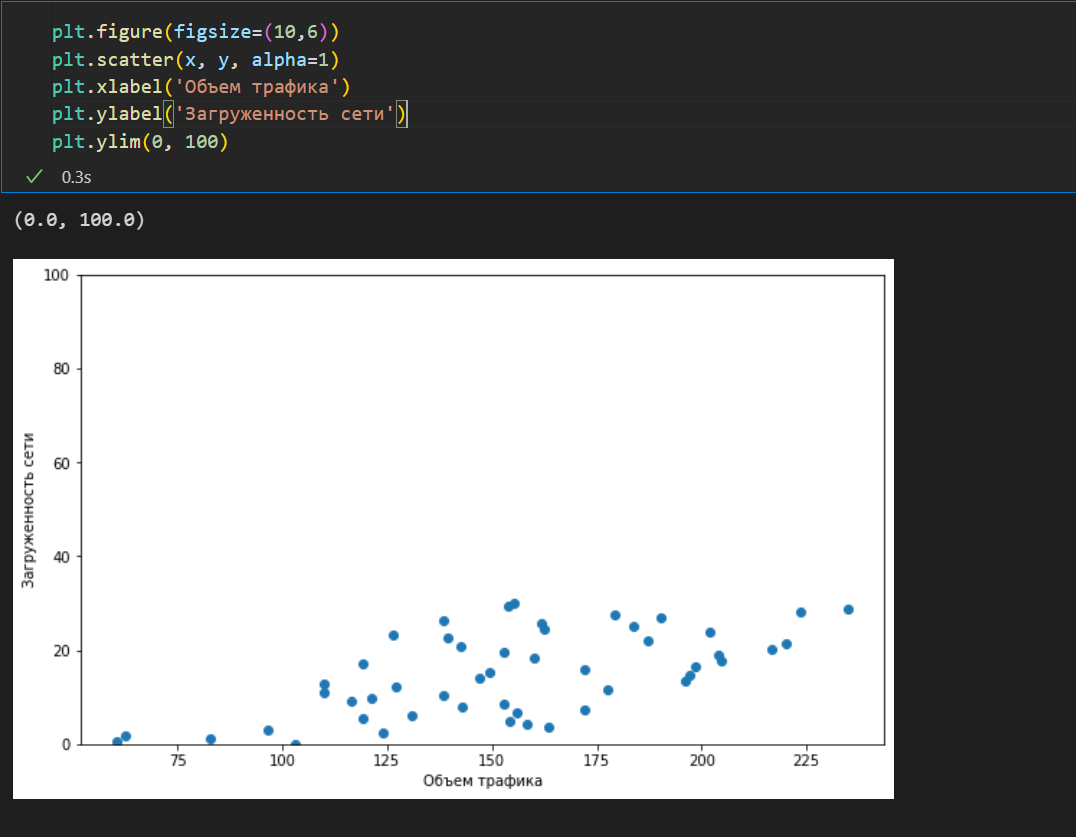


Подключаем библиотеки.

Изображение выглядит как текст

Автоматически созданное описание

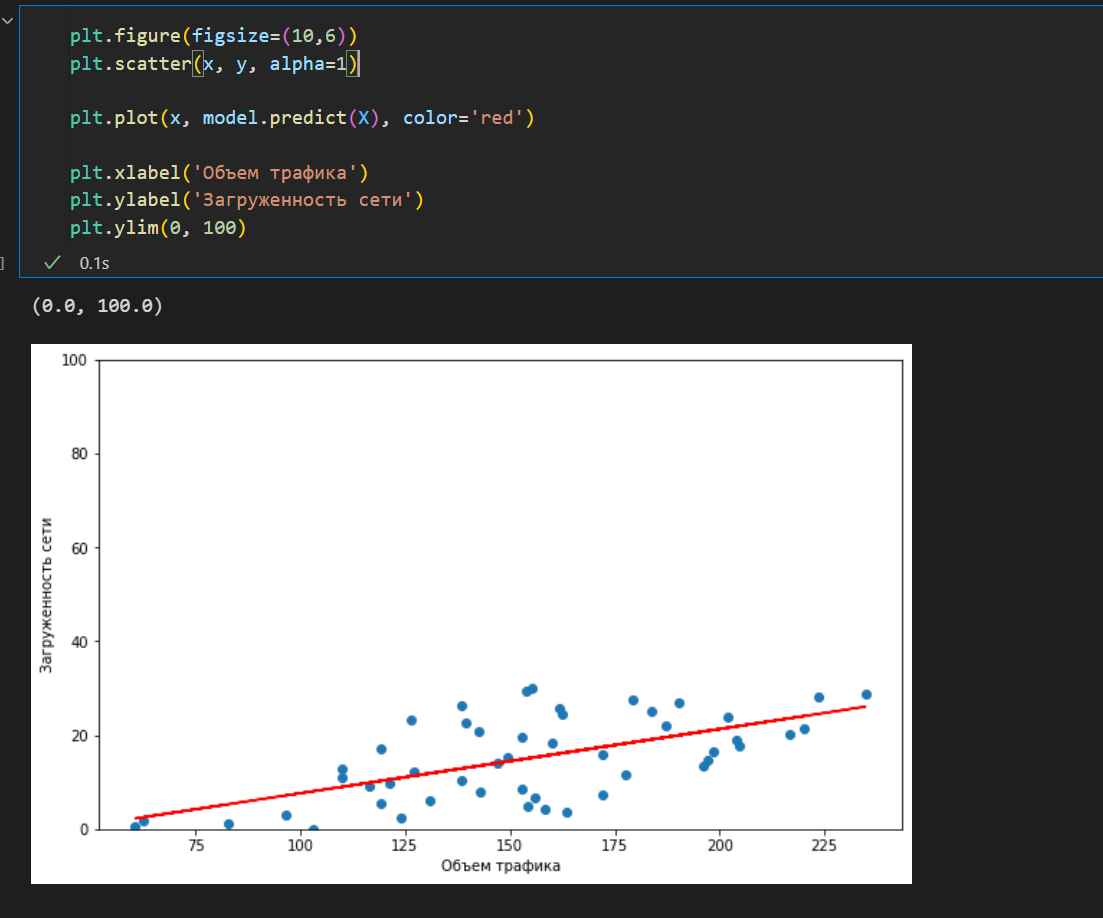
Загружаем датасет

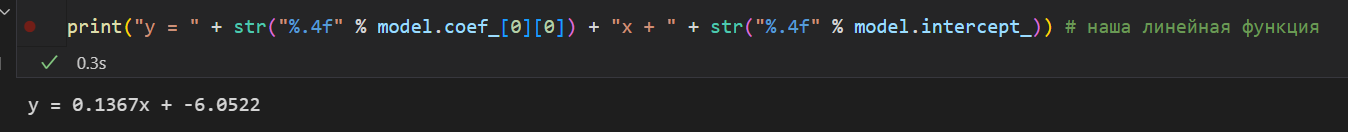


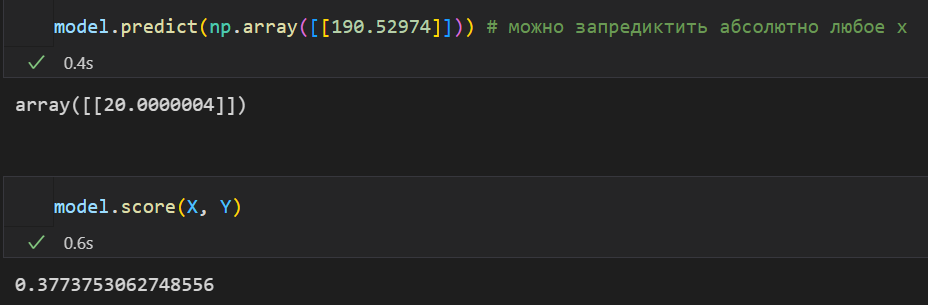
Изображение выглядит как текст

Автоматически созданное описание

1 способ с линейной регрессией из библиотеки sklearn. Coef\_ - это наш массив весов. Далее график этой прямой.





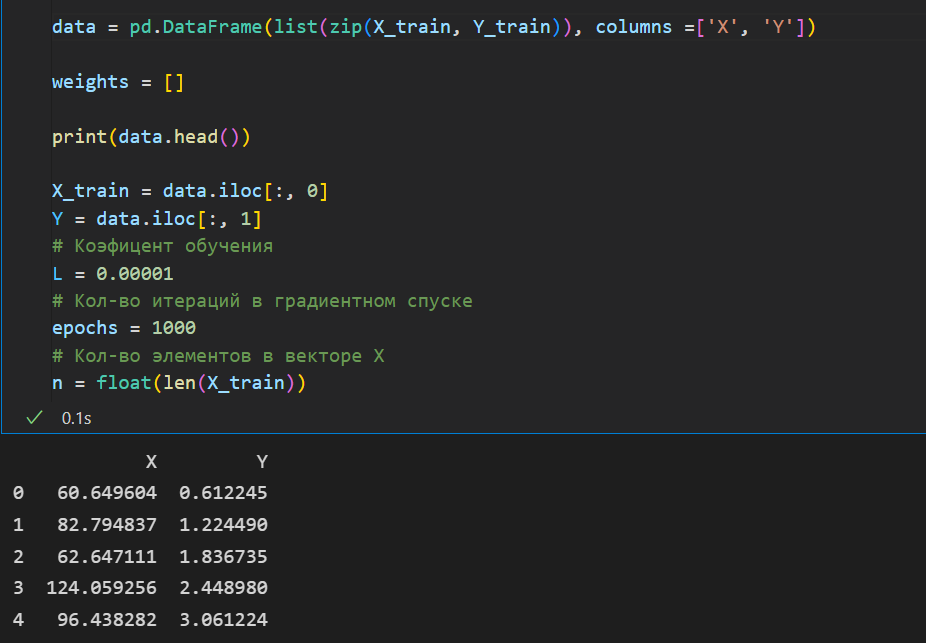


Score – определяет точность и она относительно плохая получилась в нашем случае.

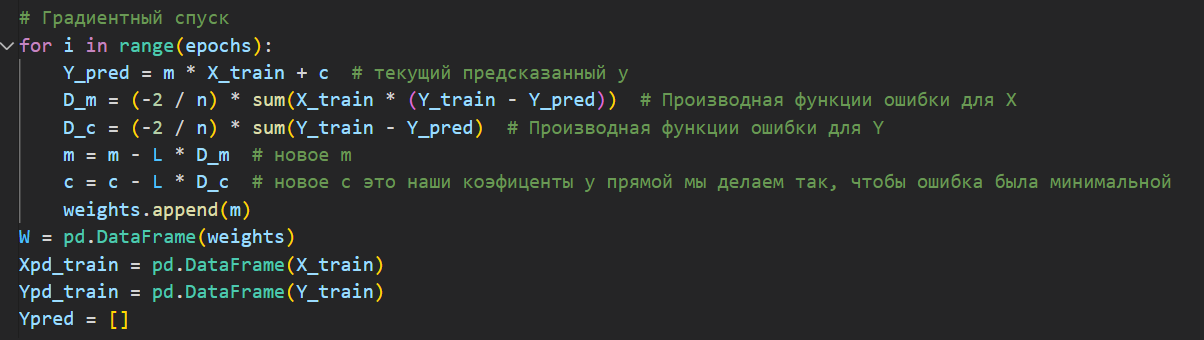
2ой способ. Своя реализация. Сплитим данные между тренировочными и тестовыми. В идеале еще добавить валидационные.

Изображение выглядит как текст

Автоматически созданное описание



Градиентный спуск – нужен для того, чтобы найти локальный минимум. Почему локальный и можно ли найти глобальный минимум у нашей функции ошибки? Глобальный минимум найти очень тяжело у сложных функций и это будет стоить больших вычислительных сил. Градиентный спуск это просто уменьшение функции при помощи вычитания градиента этой функции, так как выход градиента это просто направление возрастания функции.



Изображение выглядит как текст

Автоматически созданное описание

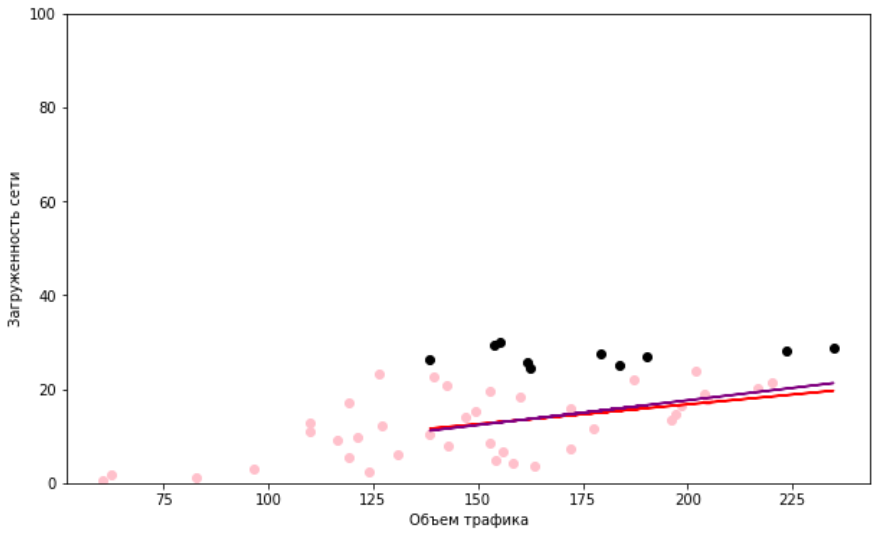
Проверяем на тех же тренировочных данных нашу при помощи sklearn.

Изображение выглядит как текст

Автоматически созданное описание

Чертим. Изображение выглядит как текст

Автоматически созданное описание



Черные точки – это тестовые данные. Розовые – тренировочные данные. Красная линия сделана по собственной модели, фиолетовая по sklearn.