**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

**Курсовая работа**

по курсу «Методы, средства и технологии мультимедиа»

Студент: Жерлыгин М.А.

Группа: М8О-408Б-18

Преподаватель: Б. В. Вишняков.

Оценка:

Москва, 2021

**Постановка задачи:**  Выбрать задачу (классификация или регрессия), датасет и метрику качества. Выбранные данные необходимо визуализировать и проанализировать. После этого выполнить препроцессинг. Затем реализовать алгоритм линейной регрессии, проверить качество обучения, сравнить с моделью из sklearn.

**Вариант:** Линейная регрессия.

**Входные данные**

Датасет представляет из себя информацию о работниках в некоторой IT компании “X”. Он составлен специально для задачи линейной регрессии. Датасет содержит следующие признаки:

* Возраст
* Возможность командировок
* Отдел
* Удаленность офиса от дома
* Количество образований
* Специальность образования
* Пол
* Уровень квалификации
* Роль
* Семейный статус
* Ежемесячная зарплата
* Количество прошлых рабочих мест
* Старше 18 лет (да/нет)
* Повышение зарплаты в процентах
* Стандартное количество часов работы
* Уровень опциона на акции
* Стаж работы
* Количество пройденных курсов за прошлый год
* Количество лет, отработанных в компании
* Количество лет с прошлого повышения
* Количество лет с текущим менеджером

**Описание**

**Линейная регрессия** — модель зависимости переменной x от одной или нескольких других переменных (факторов, регрессоров, независимых переменных) с линейной функцией зависимости.

Линейная регрессия относится к задаче определения «линии наилучшего соответствия» через набор точек данных и стала простым предшественником нелинейных методов, которые используют для обучения нейронных сетей. В этой статье покажем вам примеры линейной регрессии.

**Градиентный спуск** — метод численной оптимизации, который может быть использован во многих алгоритмах, где требуется найти экстремум функции — нейронные сети, SVM, k-средних, регрессии. Однако проще его воспринять в чистом виде (и проще модифицировать).

**Регуляризация** — метод добавления некоторых дополнительных ограничений к условию с целью решить некорректно поставленную задачу или предотвратить переобучение. Чаще всего эта информация имеет вид штрафа за сложность модели.

**Регуляризация L2** – это тип линейной регрессии, который позволяет регуляризовать модель, основан на выборе как можно меньших значений веса. Другими словами, регуляризация ограничивает модель, уменьшая влияние входов на выход, тем самым модель становится регуляризованной и избегает переоснащения этими ограничениями.

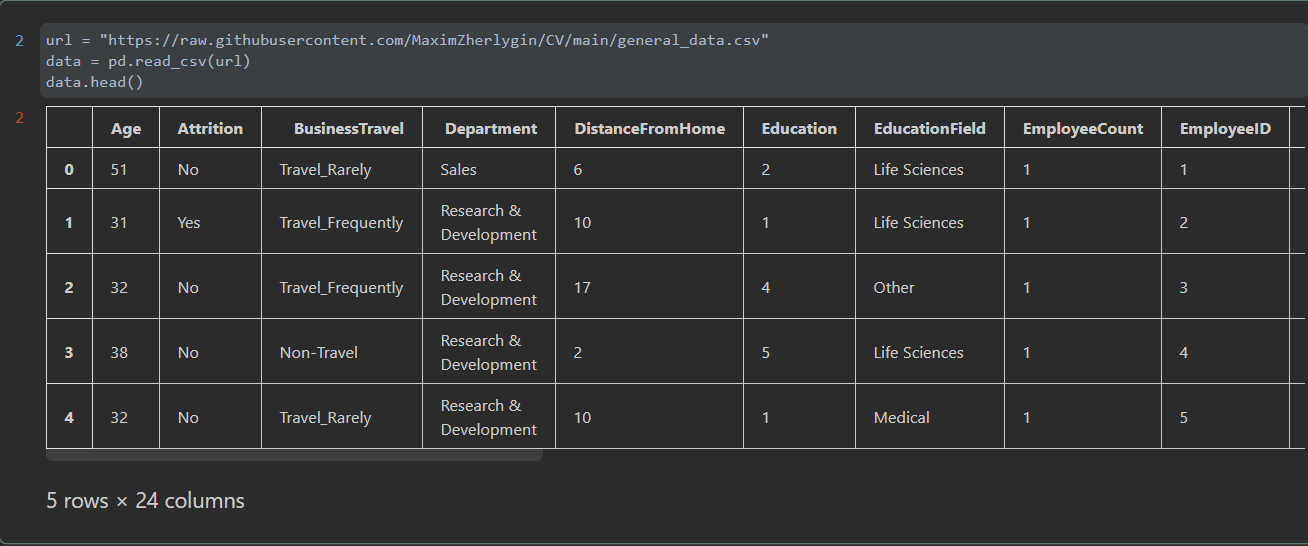
**Регуляризация L1** – оператор наименьшей абсолютной усадки и выбора (ЛАССО), при этом методе веса некоторых значений принимаются равными модулю числа, предполагая, что они не влияют на результат.

Разница между L1 и L2:

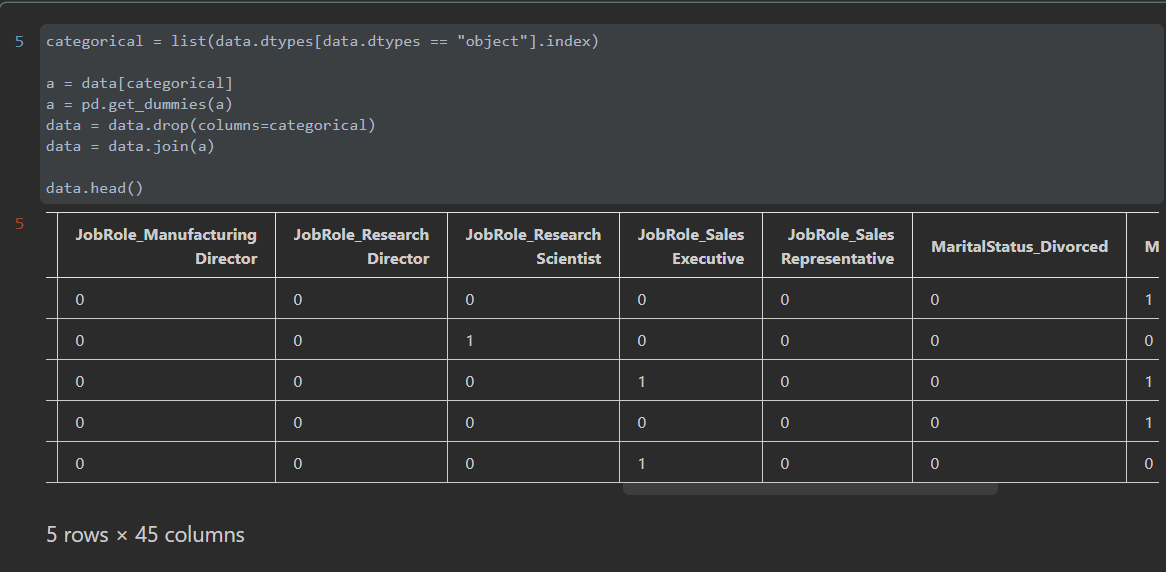
При L2-регуляризации дополнительный член является квадратичной функцией, при L1-регуляризации – модулем. Что здесь действительно важно – производная функции. Производная, конечно, является ключевой, поскольку градиентный спуск в основном движется в направлении производной.При квадратичном члене чем ближе вы находитесь к нулю, тем меньшей становится ваша производная, пока также не приблизится к нулю. Поэтому при L2-регуляризации когда ваша величина w уже мала, дальнейший градиентный спуск уже её сильно не изменит. В случае модуля производная является константой с абсолютной величиной, равной единице. Формально в нуле она не определена, но мы считаем её также равной нулю.Поэтому при L1-регуляризации градиентный спуск будет стремиться к нулю с постоянной скоростью, а достигнув его, там и останется. Вследствие этого L2-регуляризация способствует малой величине весовых коэффициентов, а L1-регуляризация способствует их равенству нулю, тем самым провоцируя разрежённость.

**Ход работы**

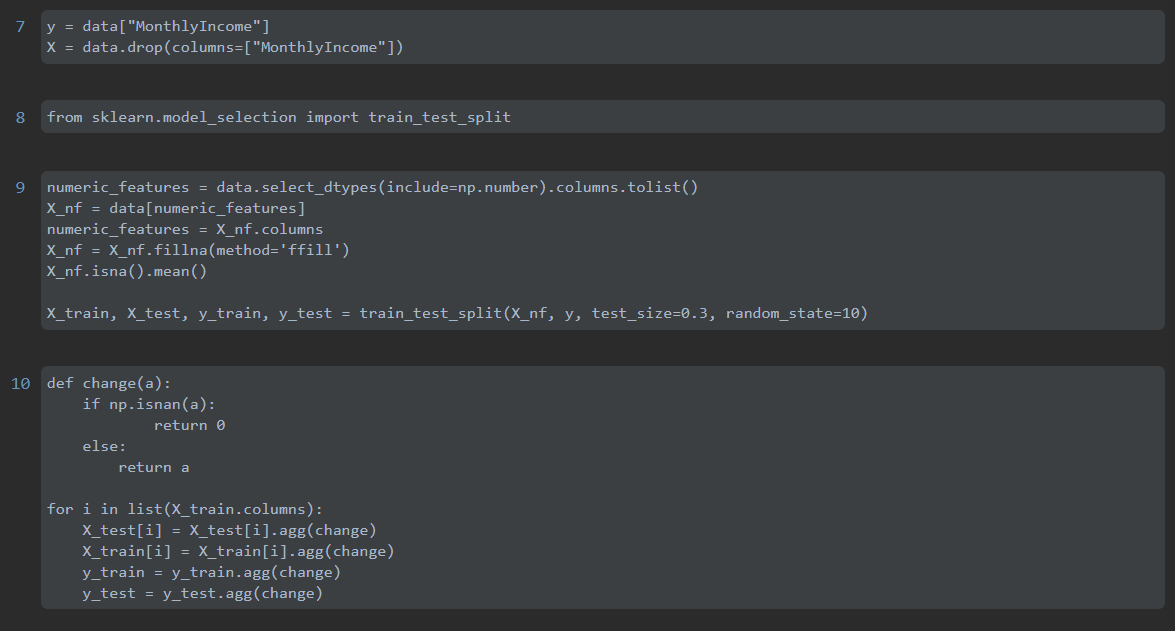
Загружаем датасет и изучаем его данные:

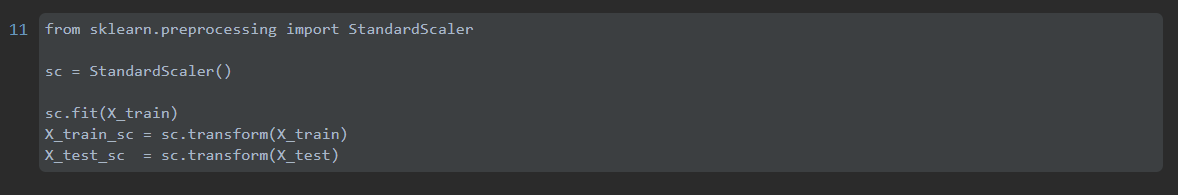


Производим One Hot Encoding на категориальных признаках:

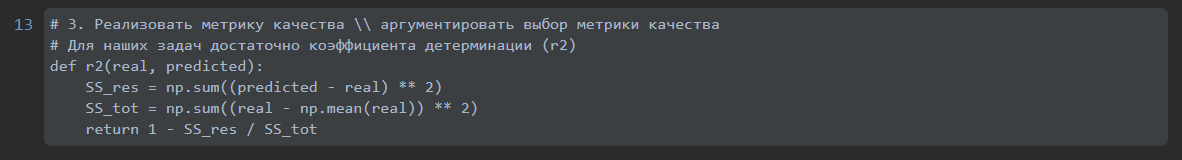


Далее мы разделяем данные на обучающую и тестовую выборку, а также нормализуем X-данные с помощью StandardScaler.

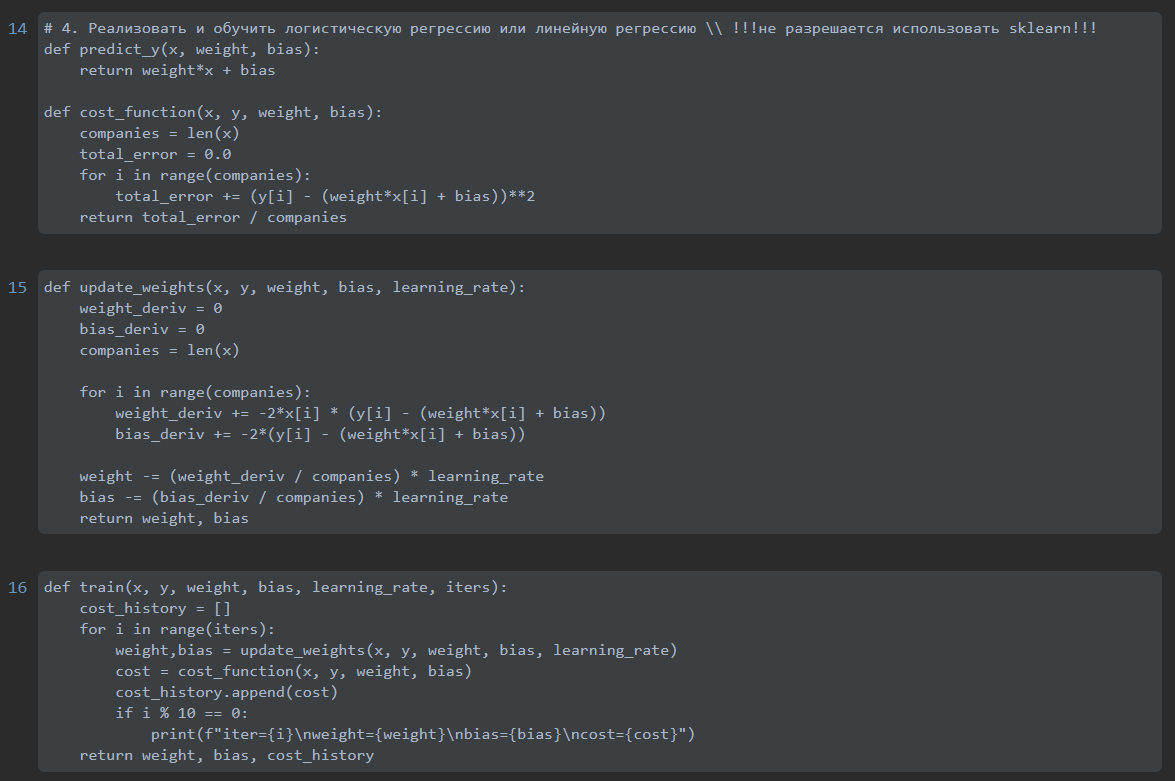




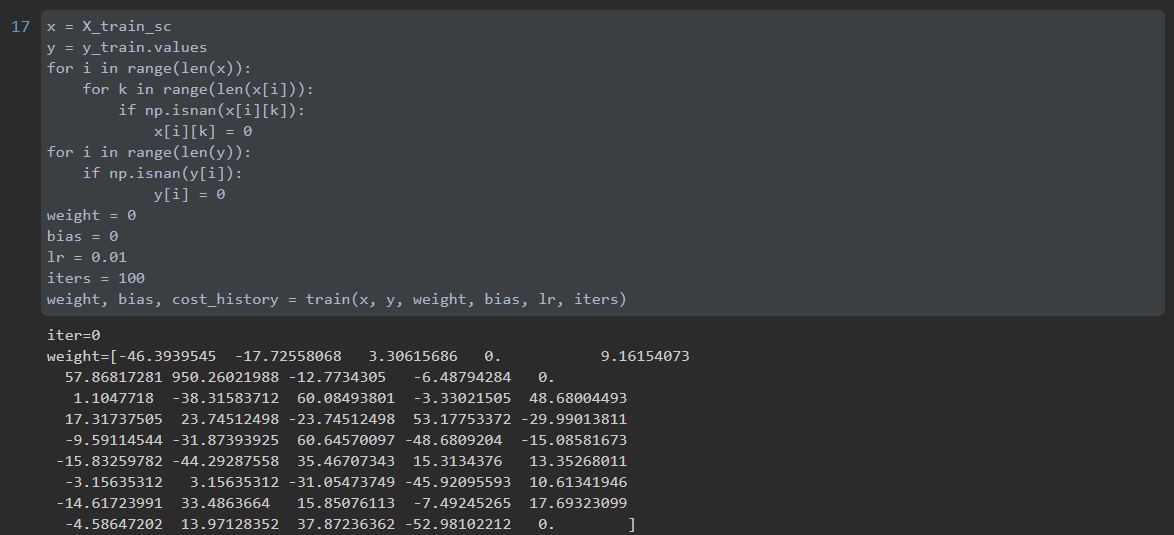
Реализуем метрику качества:

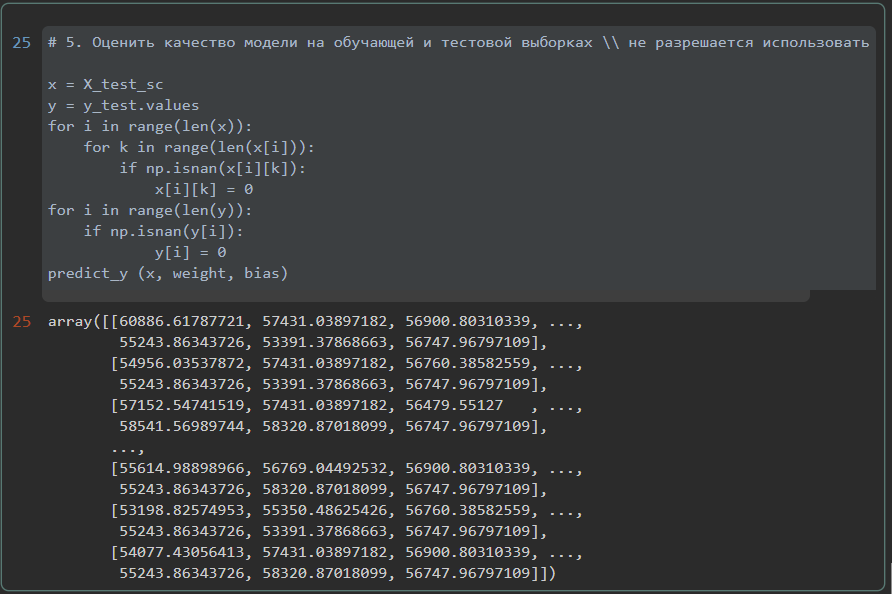


Реализуем линейную регрессию:

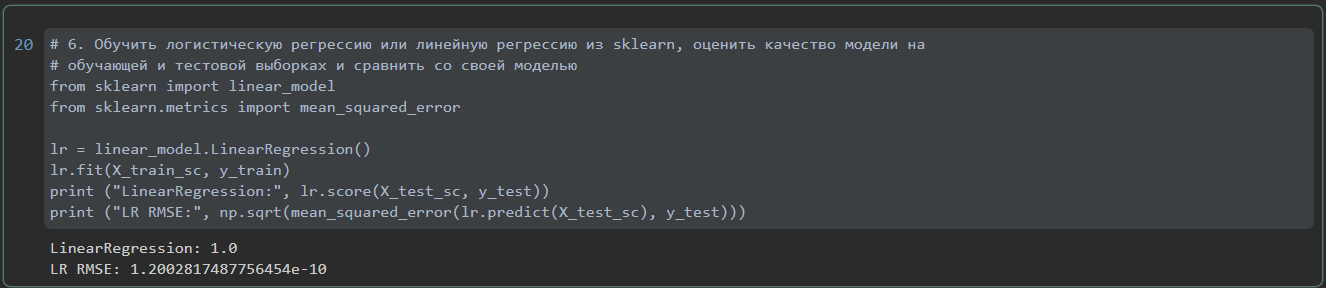


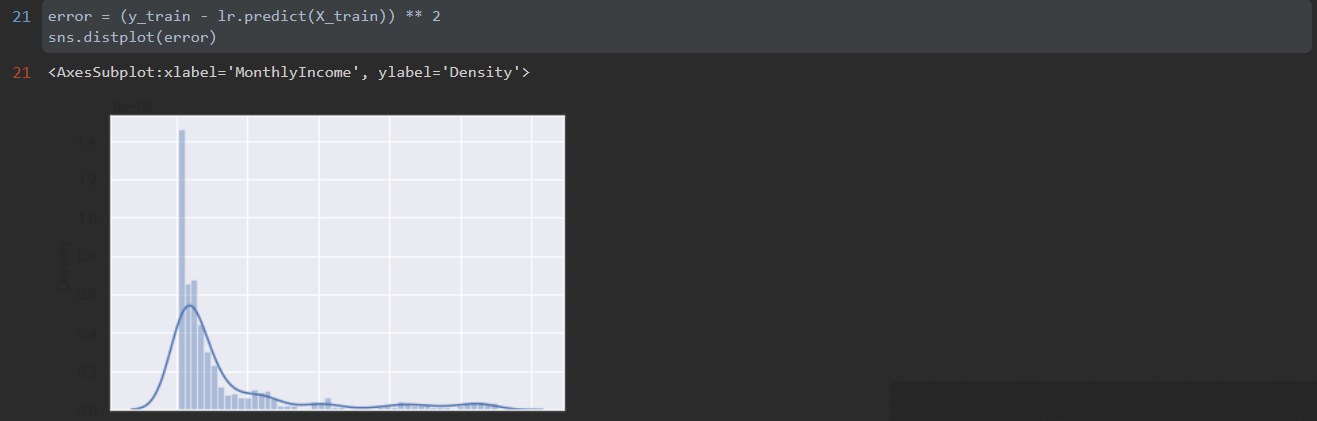
Обучим линейную регрессию:

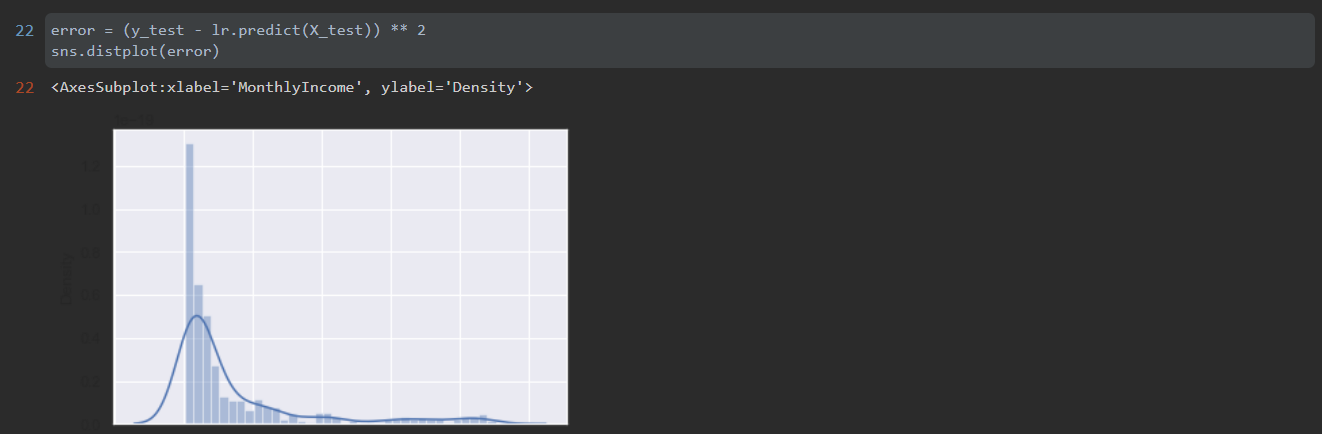




Обучаем логистическую регрессию или линейную регрессию из sklearn, оцениваем качество модели на обучающей и тестовой выборках:







**Вывод**

Познакомился с Линейной регрессией, которая является простым, но мощным механизмом для аппроксимации линейных зависимостей.

Модель является предшественником нелинейных методов, которые используют для обучения нейронных сетей.