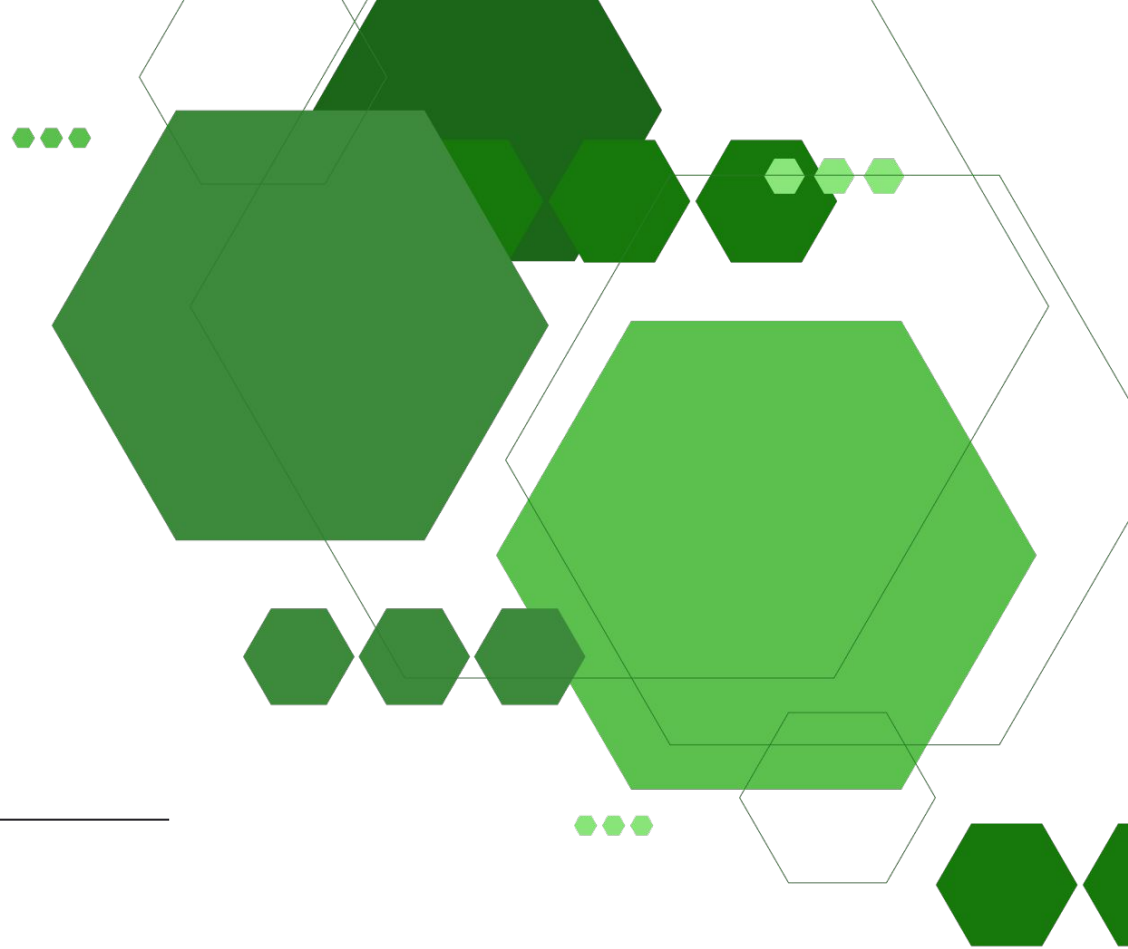
SkillFactory

Вебинар: Линейная регрессия

Виктория Тюфякова

Data scientist, ментор курса DST



Знакомство:

VICDS - Victoria Data Science

- Магистр информационных технологий КубГУ
- **Data Scientist -** Аналитик данных
- Ментор онлайн курса по **DS**
- Автор канала **@vicds** в телеграмм
- Блог в инстаграм **@vicdscience**

Структура вебинара:

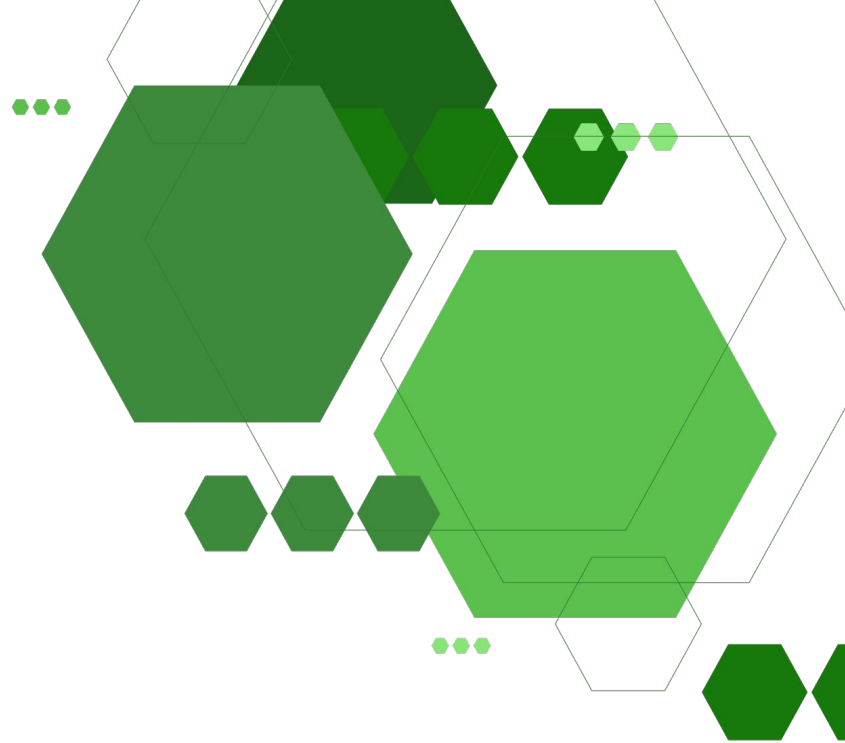
План (~90 минут)

•Повторение и углубление теории модуля ML-4 (30 минут)

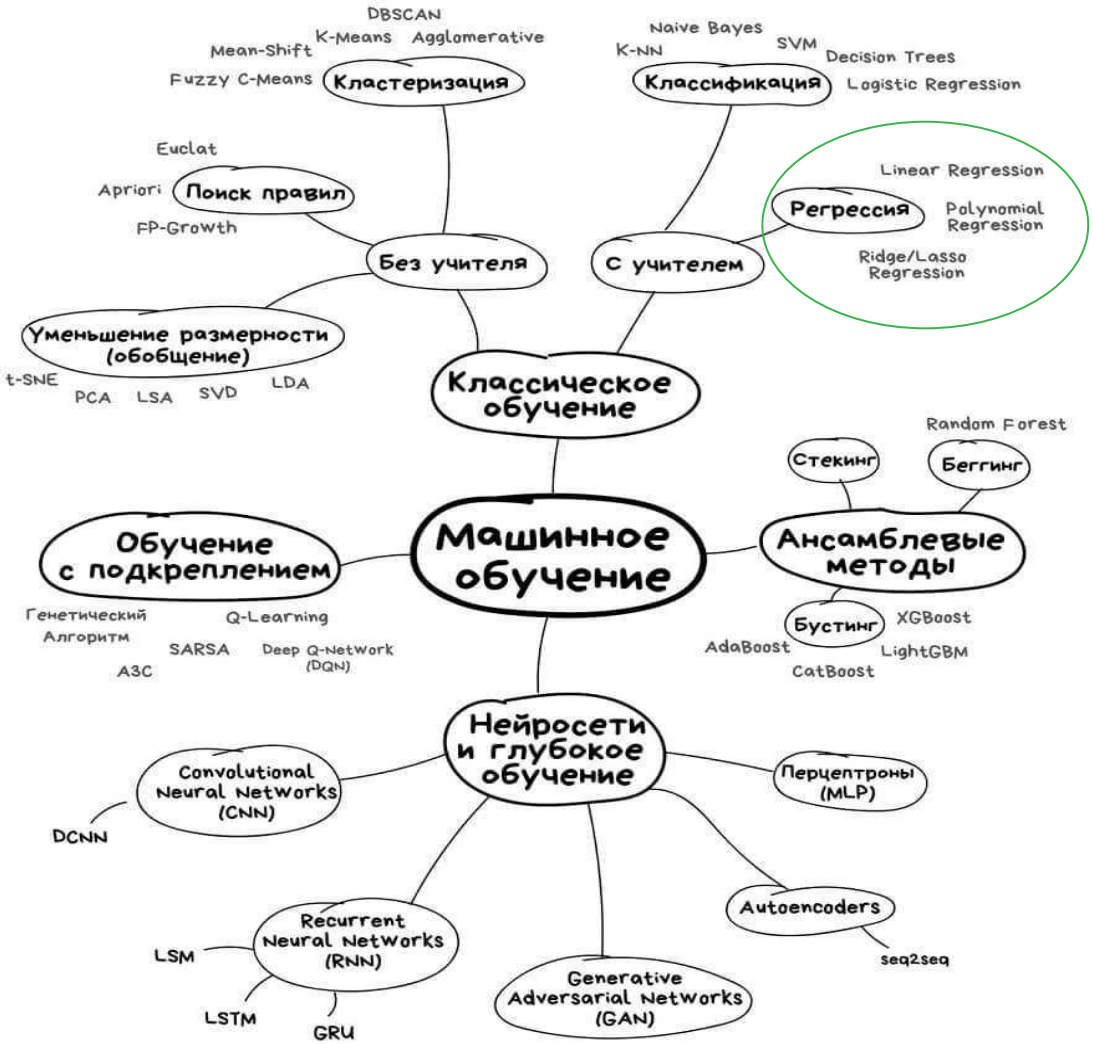
- Задачи регрессии
- Регуляризация
- Метрики регрессии
- Кросс-валидация

•Практическая работа (40 - 50 минут)

•Q&A сессия по теме вебинара (10 - 20 минут)



Классическое машинное обучение



Задача регрессии

Задача регрессии — нахождение зависимостей между определяющими переменными и целевой переменнои, если она является непрерывным числом.

Задача линейной регрессии — нахождение такои зависимости, если она линейная.

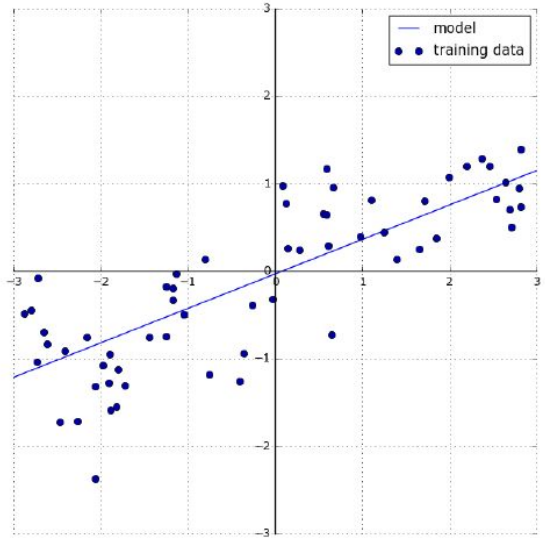
Цель регрессии состоит в том, чтобы спрогнозировать непрерывное число или вещественное число (float number).

Линейная регрессия

$$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j$$

Здесь **x** обозначают признаки (в данном примере число характеристик равно **j**) для отдельной точки данных, **w1-wj** – параметры модели, оцениваемые в ходе обучения, и **a(x)** прогноз, выдаваемый моделью.

Вес ***w*0** называется свободным коэффициентом или сдвигом (bias).



Построим простейшую модель зависимости веса от роста человека:

$$a(x) = w0 + w1 * x$$

$$x = 167\text{см}, w1 = 0.33, w0 = 2$$

$$a(54) = 0.33 * 167 + 2 = 57\text{кг}$$

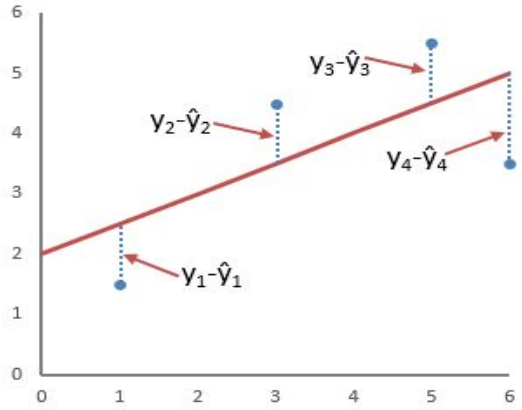
Обучение линейной регрессии

Чаще всего линейная регрессия обучается с использованием среднеквадратичной ошибки.

В этом случае получаем задачу оптимизации: $$\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\langle w, x_i\rangle - y_i\right)^2 \to \min_w$$

Метод наименьших квадратов (МНК)

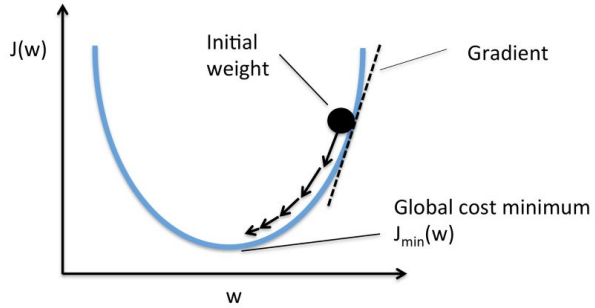
$$\vec{w} = \left(X^T X\right)^{-1} X^T \vec{y}$$



Градиентный спуск

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q\left(w^{(k-1)}\right)$$

Основное свойство антиградиента — он указывает в сторону наискорейшего убывания функции в данной точке.



Метрики для задач регрессии

MAE (*Mean Absolute Error)* среднее арифметическое модуля отклонения предсказанного значения от реального.

$$\mathrm{MAE} = \frac{1}{m}\sum_{i-1}^{m}|a_i - y_i|$$

MSE (*Mean Squared Error*)

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}|a_i - y_i|^2$$

RMSE (*Root Mean Squared Error*)

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}|a_i - y_i|^2}$$

MAPE (*Mean Absolute Percent Error*)

$$\mathrm{MAPE} = 100\% \cdot \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{|y_i|}$$

R-squared (коэффициент детерминации): Измеряет долю дисперсии, объясненную моделью.

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^{\ell}(a(x_i) - y_i)^2}{\sum_{i=1}^{\ell}(y_i - \bar{y})^2}$$

Универсальная мера зависимости одной случайной величины от множества других

Разложение ошибки на смещение и разброс (Bias-variance)

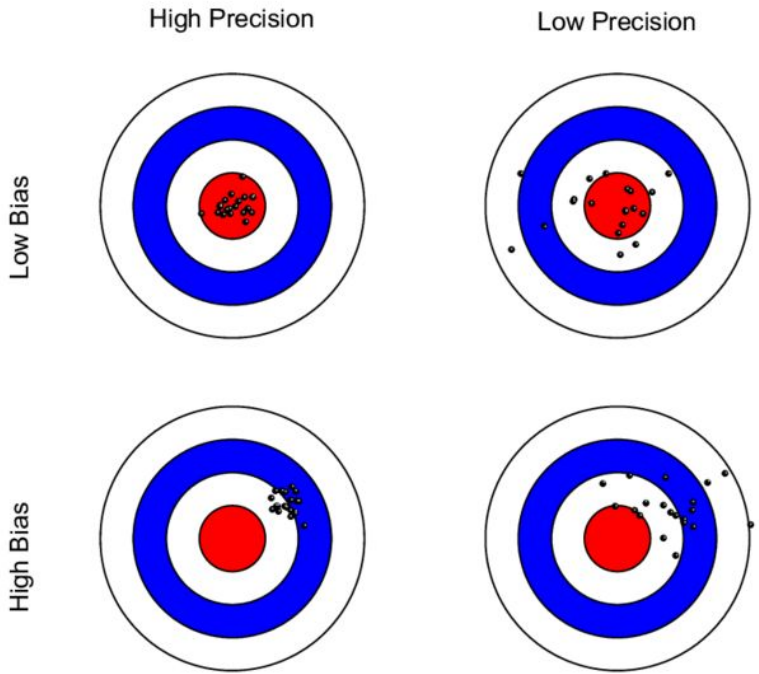
Ошибка прогноза любой модели вида $y=f(x\vec{})+\epsilon$ складывается из:

$$\mathrm{Err}\left(\vec{x}\right) = \mathrm{Bias}\left(\hat{f}\right)^2 + \mathrm{Var}\left(\hat{f}\right) + \sigma^2$$

- квадрата смещения: Bias– средняя ошибка по всевозможным наборам данных;
- дисперсии: Var – вариативность ошибки, то, на сколько ошибка будет отличаться, если обучать модель на разных наборах данных;
- неустранимой ошибки: $\sigma 2$.

При увеличении сложности модели увеличивается дисперсия (разброс) оценки, но уменьшается смещение.

Если же модель слабая, то она не в состоянии найти закономерность, в результате получаем что-то другое, смещенное относительно правильного решения.



Регуляризация

Регуляризация означает явное ограничение модели для предотвращения переобучения.

$$Q_\alpha(w) = Q(w) + \alpha R(w)$$

Коэффициент **α** называется параметром регуляризации и контролирует баланс между подгонкой под обучающую выборку и штрафом за излишнюю сложность. Значение параметра подбирается перебором под каждую задачу.

Наиболее распространенными являются L1 и L2-регуляризаторы:

L1 $$R(w) = \|w\|_1 = \sum_{i=1}^{d} |w_i|$$ Lasso-регуляризация

L2 $$R(w) = \|w\|_2 = \sum_{i=1}^{d} w_i^2$$ Ridge-регуляризация

Гребневая регрессия

В гребневой регрессии коэффициенты **(w)** выбираются не только с точки зрения того, насколько хорошо они позволяют предсказывать на обучающих данных, но они еще подгоняются в соответствии с дополнительным ограничением - все элементы **w** должны быть близки к нулю.

Это означает, что каждый признак должен иметь как можно меньшее влияние на результат (то есть каждый признак должен иметь небольшой регрессионный коэффициент) и в то же время он должен по-прежнему обладать хорошей прогнозной силой.
Регуляризация, использующаяся в гребневой регрессии, известна как L2 регуляризация.

sklearn.linear_model.Ridge

class `sklearn.linear_model.` **`Ridge`**(*alpha=1.0*, ***, *fit_intercept=True*, *normalize=False*, *copy_X=True*, *max_iter=None*, *tol=0.001*, *solver='auto'*, *random_state=None*) [source]

Linear least squares with l2 regularization.

Minimizes the objective function:

```
||y - Xw||^2_2 + alpha * ||w||^2_2
```

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e., when y is a 2d-array of shape (n_samples, n_targets)).

Лассо регрессия

Как и гребневая регрессия, лассо также сжимает коэффициенты до близких к нулю значений, но несколько иным способом, называемым L1 регуляризацией.
Результат L1 регуляризации заключается в том, что при использовании лассо некоторые коэффициенты становятся равны точно нулю.
Получается, что некоторые признаки полностью исключаются из модели.
Это можно рассматривать как один из видов автоматического отбора признаков.

sklearn.linear_model.Lasso

class `sklearn.linear_model.` **Lasso**(*alpha=1.0, *, fit_intercept=True, normalize=False, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Linear Model trained with L1 prior as regularizer (aka the Lasso)

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1
```

Technically the Lasso model is optimizing the same objective function as the Elastic Net with `l1_ratio=1.0` (no L2 penalty).

Модель регрессии с двумя регуляризаторами L1 и L2 в пропорции

sklearn.linear_model.ElasticNet

class `sklearn.linear_model.` **`ElasticNet`**(*alpha=1.0, *, l1_ratio=0.5, fit_intercept=True, normalize=False, precompute=False, max_iter=1000, copy_X=True, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Linear regression with combined L1 and L2 priors as regularizer.

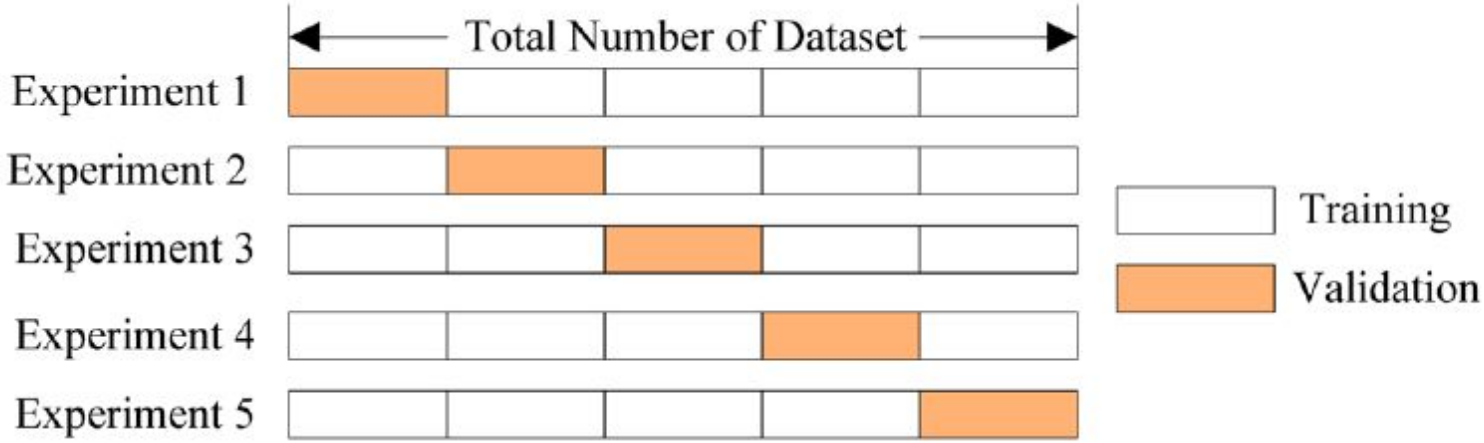
Minimizes the objective function:

```
1 / (2 * n_samples) * ||y - Xw||^2_2
+ alpha * l1_ratio * ||w||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||w||^2_2
```

Преимущества линейной регрессии

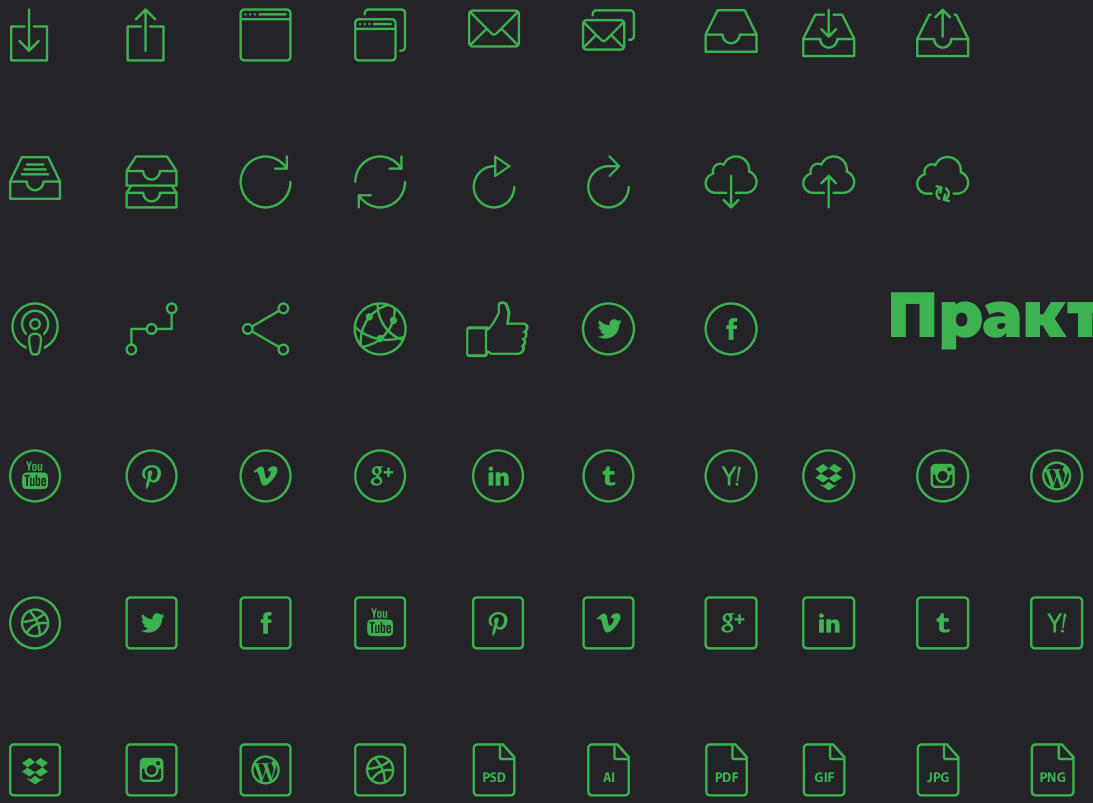
- Линейные модели очень быстро обучаются, а также быстро прогнозируют.
- Они масштабируются на очень большие наборы данных, а также хорошо работают с разреженными данными.
- Как правило, линейные модели хорошо работают, когда количество признаков превышает количество наблюдений.
- Кроме того, они часто используются на очень больших наборах данных, просто потому, что не представляется возможным обучить другие модели.
- Также у них мало параметров, благодаря чему удается контролировать риск переобучения и использовать их для работы с зашумленными данными и с небольшими выборками.
- Подходят для поиска простых взаимосвязей в данных.

Кросс - валидация



Важно помнить, что кросс-валидация не является способом построения модели, которую можно применить к новым данным. Перекрестная проверка не возвращает модель.

При вызове cross_val_score строится несколько внутренних моделей, однако цель перекрестной проверки заключается только в том, чтобы оценить обобщающую способность данного алгоритма, обучив на определенном наборе данных.

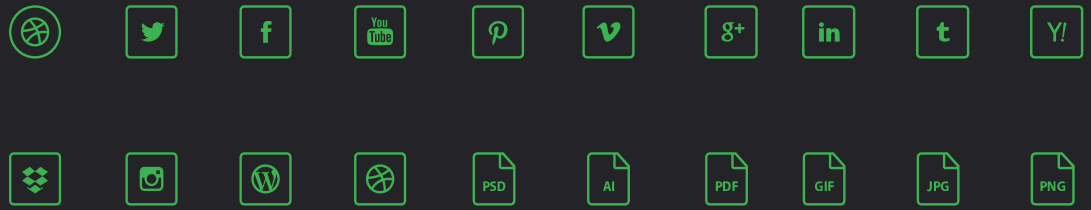


Практическая часть

Полезные источники

1. Андреас Мюллер, Сара Гвидо - Введение в машинное обучение с помощью Python;
2. Cross-Validation in Machine Learning: How to Do It Right:

https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right



Обратная связь

пройдите по QR-коду или ссылке
заполнение займет 2-3 минуты :)

ссылка на анкету

Укажите в соответствующих полях анкеты:
Ведущая вебинара: Виктория
Тема вебинара: Регрессия