

Конспект по теме "Переходим к регрессии"

Среднеквадратичное отклонение

Наиболее распространённая метрика качества в задаче регрессии — **среднеквадратичное отклонение**, или **среднеквадратичная ошибка, MSE**.

Чтобы получить среднеквадратичную ошибку, сначала вычисляется отклонение каждого объекта:

отклонение объекта = предсказание модели — правильный ответ

Среднеквадратичное отклонение рассчитывается по формуле:

$$MSE = \frac{\text{Сумма квадратов отклонений объекта}}{\text{Количество объектов}}$$

Разберём вычисления:

1. Отклонение объекта показывает, как сильно правильный ответ отличается от предсказания.
2. Возведение в квадрат избавляет от разницы между переоценкой и недооценкой.
3. Усреднение нужно, чтобы получить данные по всем объектам.

Чем меньше среднеквадратичное отклонение, тем лучше модель регрессии.

Расчёт MSE

Для расчёта среднеквадратичного отклонения, импортируйте из модуля *sklearn.metrics* функцию *mean_squared_error*:

```
from sklearn.metrics import mean_squared_error

mse = mean_squared_error(answers, predictions)
```

В результате вычисления *MSE*, мы получим число, единица измерения которого — квадрат исходной единицы измерения (например, «квадратные рубли»). Чтобы получить метрику качества в исходных единицах измерения, берут **корень от среднеквадратичной ошибки** — *RMSE* (*root mean squared error*):

```
rmse = mse ** 0.5
```

Дерево решений в регрессии

Дерево в задаче регрессии обучается так же, только предсказывает оно не класс, а число.

Решающее дерево для задачи регрессии называется *DecisionTreeRegressor* и находится в модуле *sklearn.tree*.

```
from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor(random_state=12345)
```

Случайный лес в регрессии

Случайный лес для регрессии не сильно меняется. Он обучает множество независимых деревьев, а потом принимает решение, усредняя их ответы:

```
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(random_state=12345, n_estimators=3)
```

Линейная регрессия

Линейная регрессия похожа на логистическую. Название пришло из линейной алгебры, которой будет посвящён отдельный курс. Из-за малого количества параметров линейная регрессия не склонна к переобучению.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
```