Лекция 6

Регрессия

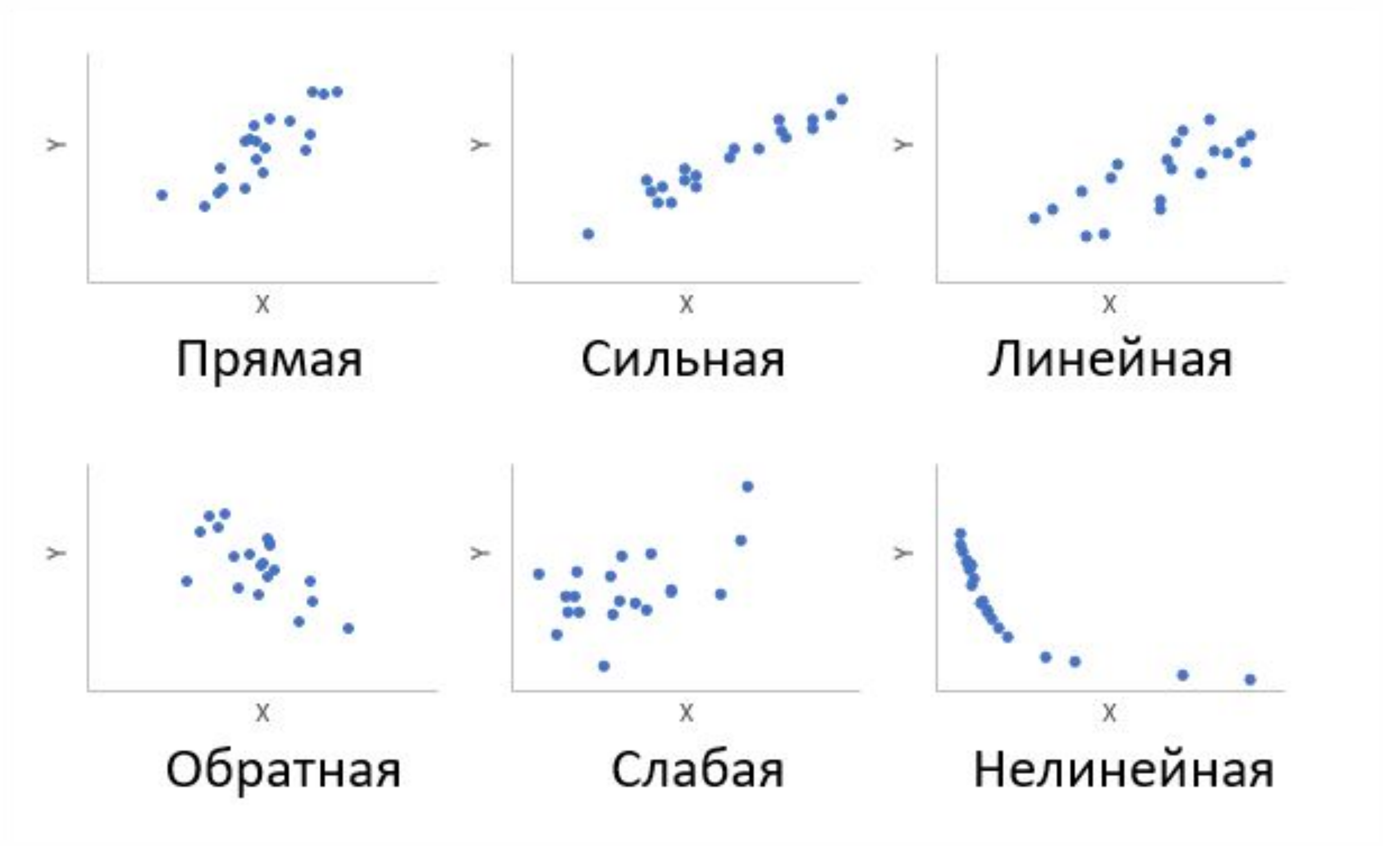
Корреляция

- Корреляция - статистическая взаимосвязь между двумя случайными величинами, такая, что изменение одной величины ведет к систематическому изменению другой.

- Если в наборе данных наблюдается корреляция, то можно говорить о проведении регрессии для получения модели зависимости.

- Наличие корреляции можно оценивать визуально или расчетом специальных коэффициентов, отражающих степень корреляции

Виды корреляции



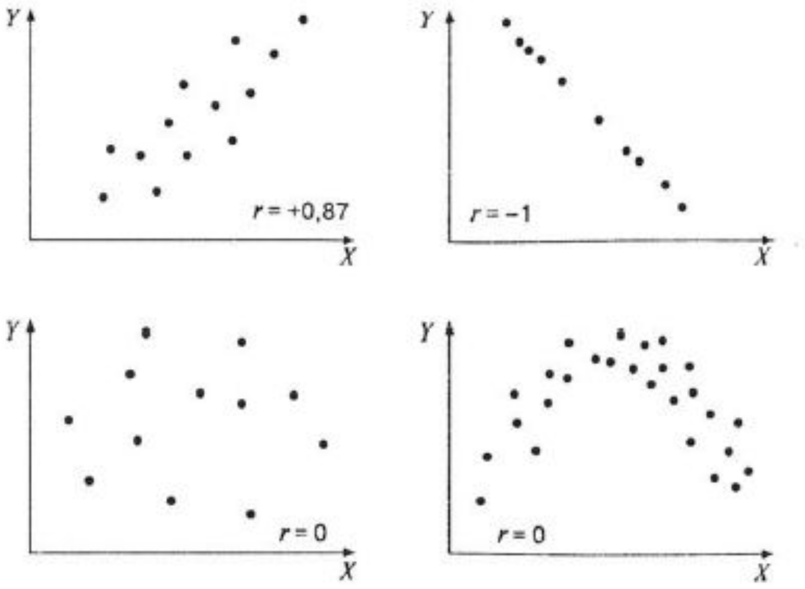
Коэффициент корреляции r

- Коэффициент корреляции r - также коэффициент линейной корреляции
- Принимает значение от -1 до 1.
- Если r равен 1 или -1, то наблюдается полная линейная зависимость
- Если r равен 0, то отсутствует какая либо линейная зависимость

$$r_{XY} = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\sum(X-\overline{X})^2\sum(Y-\overline{Y})^2}} = \frac{Cov(X,Y)}{\sigma_X\sigma_Y}$$

Значения коэффициента корреляции

- Если r положительный, то наблюдается прямая зависимость.
- Если r отрицательный, то наблюдается обратная зависимость.
- r скорее всего не отобразит наличие нелинейной зависимости.



В SKlearn sklearn.feature_selection.r_regression

Регрессия

- Задача регрессии - является задачей обучения с учителем.
- Цель регрессии выявить вид зависимости между независимыми переменными X (*предикторы*) и зависимой переменной Y (*отклик*) .

- Примеры задач регрессии:
    - Кредитный скоринг - по анкете оценить величину кредитного лимита
    - Оценка стоимости недвижимости
    - Прогноз свойств химических соединений
    - Оценка экологической обстановки

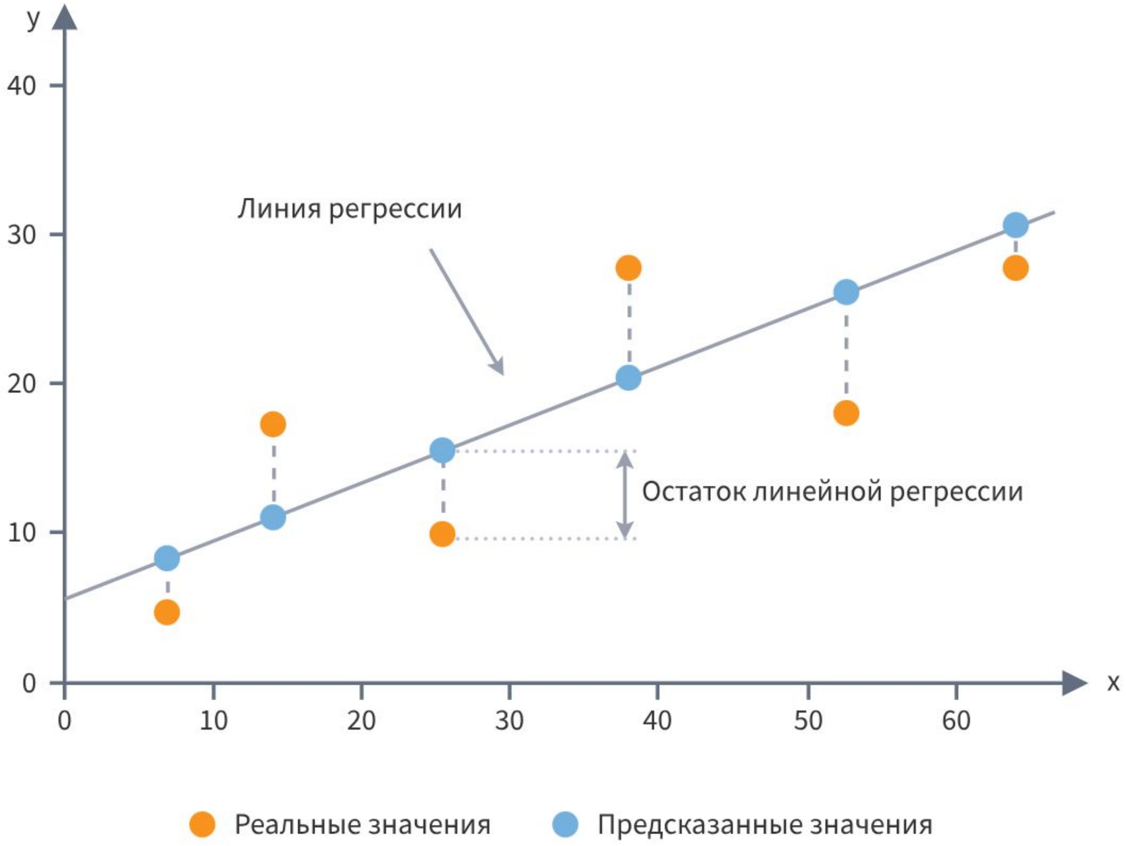
Линейная регрессия

- Линейная регрессия - вид регрессии, в котором производится поиск линейной зависимости между предикторами и откликом.
- Регрессия от одного предиктора имеет вид: $y = a \cdot x + b + \varepsilon$
- Где, *x* - предиктор, *y* - отклик, *a* и *b* - параметры регрессионной модели
- В общем виде имеет запись: $y = f(x, b) + \varepsilon$
- *f(x,b)* - регрессионная модель:

$$f(x, b) = b_0 + b_1 \cdot x_1 + ... + b_n \cdot + x_n =$$
$$b_0 + \sum_{i=1}^{n} b_i \cdot x_i = x^T \cdot b, (x_0 = 1)$$

Линейная регрессия - визуализация

$$\varepsilon = y - f(x, b)$$



Ошибка регрессионной модели

- Использовать остатки напрямую неудобно, так как они могут иметь отрицательные значения
- Как ошибку предсказания используем квадрат разницы:

$$err = (y - \widehat{y})^2 = (y - f(x, b))^2$$

- Для ошибки всей модели используем сумму квадратов откликов (SSE):

$$SSE = \sum_{i=1}^{N} (y_i - f(x_i, b))^2 = \sum_{i=1}^{N} err_i$$

- Также используют среднее по квадратам откликов (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i, b))^2 = \frac{SSE}{N}$$

Метод наименьших квадратов

- При обучении модели, основная задача, минимизировать ошибку.
- Известно, что в точке минимума производная функции равна 0
- Чтобы найти значения параметров при которых ошибка минимальна, необходимо найти частные производные по каждому параметру, и решить систему уравнений:

$$\begin{cases} \frac{\partial SSE}{b_0} = 0 \\ \frac{\partial SSE}{b_1} = 0 \\ \dots \\ \frac{\partial SSE}{b_n} = 0 \end{cases}$$

Метод наименьших квадратов (матричная форма)

- В матричной форме линейная регрессия $y = X \cdot b + \varepsilon$
- Предсказанные значения $\widehat{y} = X \cdot b$
- Ошибки $err = y - \widehat{y} = y - X \cdot b$
- SSE $err^T \cdot err = (y - X \cdot b)^T (y - X \cdot b)$
- Дифференцируя по b можно получить систему $(X^T \cdot X)b = X^T \cdot y$
- Параметры модели можно вычислить $b = (X^T \cdot X)^{-1} X^T \cdot y$

Метод наименьших квадратов для 1 предиктора (1)

- Ошибка $SSE = \sum_{i=1}^{N} (a \cdot x_i + b - y_i)^2$
- Частная производная по a $\frac{\partial SSE}{\partial a} = 2 \sum x_i \left(a \cdot x_i + b - y_i\right) \Rightarrow a = \frac{\sum x_i \cdot y_i - b \sum x_i}{\sum x_i^2}$
- Частная производная по b $\frac{\partial SSE}{\partial b} = 2 \sum \left(a \cdot x_i + b - y_i\right) \Rightarrow b = \frac{\sum y_i - a \sum x_i}{N}$
- Сделаем замену $x' = x - \overline{x}, y' = y - \overline{y}$
- Рассмотрим уравнение $y' = a' \cdot x' + b'$

Метод наименьших квадратов для 1 предиктора (2)

- Так как, средние значения равны 0, то $b' = 0$
- Получаем значение $a' = \frac{\sum x'_i \cdot y'_i}{\sum x'^2_i}$
- Из центрирования y и уравнения с заменой получаем $y = a' \cdot x' + \overline{y}$
- Вернем замену a’ и x’ $y = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sum(x_i-\overline{x})^2} \cdot x + \overline{y} - \frac{\sum(x-\overline{x})(y-\overline{y})}{\sum(x_i-\overline{x})^2} \cdot \overline{x}$
- Откуда $a = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sum(x_i-\overline{x})^2}, b = \overline{y} - a\overline{x}$

Оценка качества линейной регрессии

- Средний квадрат ошибки MSE - *sklearn.metrics.mean_squared_error*
- Средняя абсолютная ошибка MAE - *sklearn.metrics.mean_absolute_error*

$$MAE = 1/N \cdot \sum |y - \widehat{y}|$$

- Средняя абсолютная ошибка измеряется в тех же величинах, что и отклик
- Процентное отклонение - *sklearn.metrics.mean_absolute_percentage_error*

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - \widehat{y}|}{|y_i|}$$

- Коэффициент детерминации R^2 - *sklearn.metrics.r2_score*

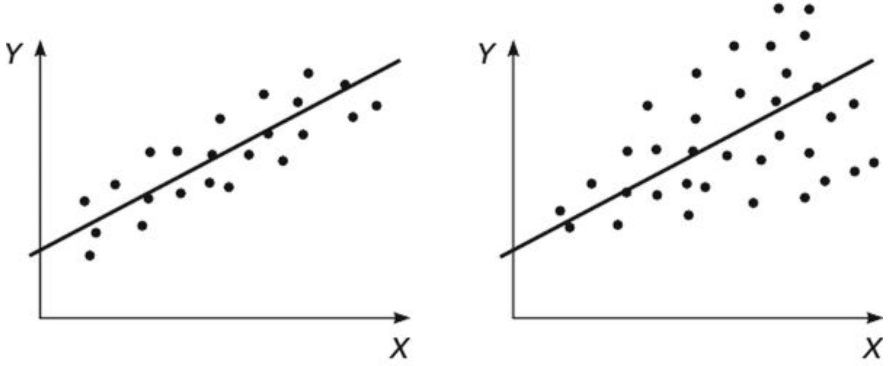
$$R^2 = 1 - \frac{SSE}{SST}, SST = \sum_{i=1}^{N} (y_i - \overline{y})^2$$

- R^2 изменяется от 0 до 1, и равен 1 если модель хорошо приближает
- Также R^2 можно вычислить через объясненную сумму квадратов:

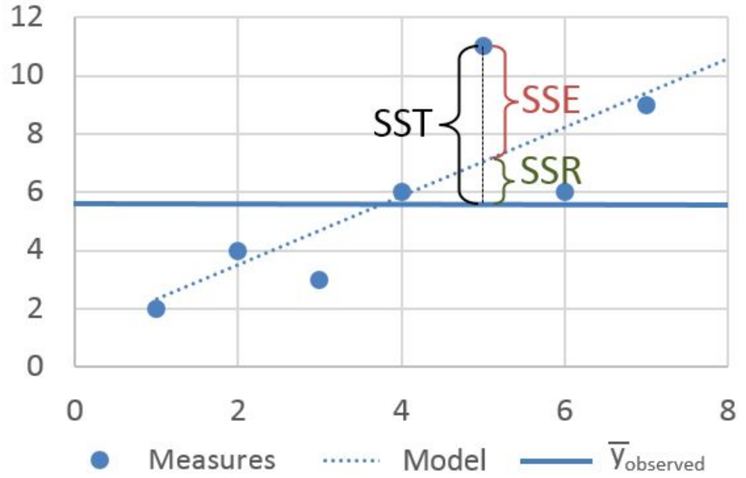
$$SSR = \sum_{i=1}^{N} \left(\widehat{y_i} - \overline{y}\right), SST = SSE + SSR \quad R^2 = 1 - \frac{SSE}{SST} = \frac{SST - SSE}{SST} = \frac{SSR}{SST}$$

Анализ остатков

- Чтобы говорить о том, что регрессия подходит для описания данных, остатки должны быть:
    - С нормальным распределением с 0 мат. ожиданием
    - Постоянной дисперсия - гомоскедастичность
    - Отсутствует линия тренда



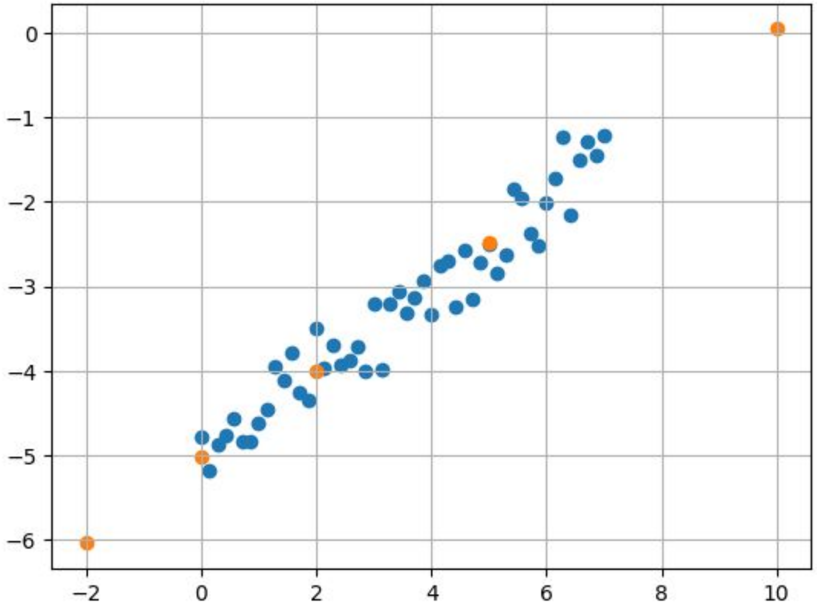
Пример гомоскедастичности и гетероскедастичности



Соотношение SST, SSE, SSR

Линейная регрессия в SKLearn

```python
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(xp.reshape(-1,1), yp.reshape(-1,1))
print(lin_reg.coef_) #пересечение
print(lin_reg.intercept_) #свободный член
yp_pred = lin_reg.predict(xp.reshape(-1,1))
xp_new = np.array([-2, 0, 2, 5, 10]).reshape(-1,1)
yp_new_pred = lin_reg.predict(xp_new)
```



Интерпретация коэффициентов

- Коэффициенты линейной регрессии показывают в какую сторону и с какой силой предикаты влияют на отклик.
- Знак показывает в какую сторону предикат влияет на отклик
- Абсолютная величина показывает с какой силой влияет предикат на отклик

- Например, получено уравнение линейной регрессии:

  *Цена жилья = 800 + 100 * площадь - 50 * расстояние до метро + 2 * кол-во парков*

- Интерпретировать коэффициенты проще, если все признаки нормированы и имеют один порядок

Проблемы линейной регрессии

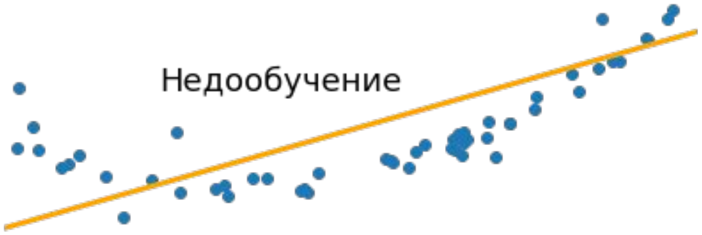
- Наличие выбросов - могут сильно сместить расположение линии

- Наличие мультиколлинеарности - наличие корреляции между предикторами. Для проверки можно делать каждый признак целевым, и проводить линейную регрессию и оценивать коэффициент детерминации (см. VIF)

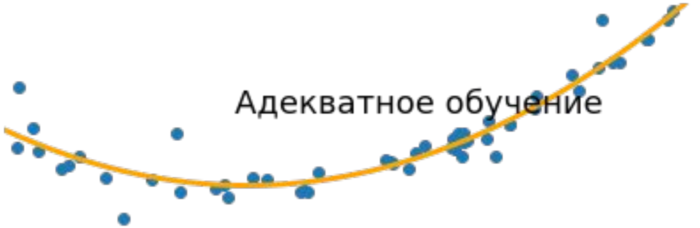
- Автокорреляция остатков - если целевой признак зависит от самого себя

Проблемы обучения с учителем

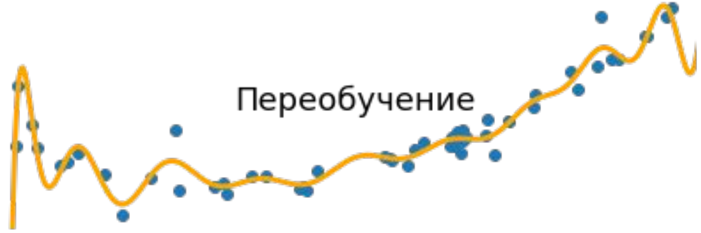
- Недообучение модели - ситуация, когда модель дает плохие предсказания на данных, на которых обучалась



- Переобучение модели - ситуация, когда модель дает хорошие результаты только на данных, на которых обучалась



- Обобщение - свойство модели корректно обрабатывать данные, которые она никогда не видела



Проверка переобучения

- Для выявления переобучения необходимо набор данных разделить на обучающую и тестовую выборку (обычно соотношение 70:30 или 80:20)

- Далее обучаем на обучающей выборке, проверяем на тестовой. Метрики на обучающей и тестовой выборке должны быть близкими. *Результат может зависеть от того, как разбили выборку.*

- Разбить можно вручную или с помощью *sklearn.model_selection.train_test_split*
- Параметры train_test_split:
  - test_size/train_size - целое число определяет кол-во наблюдений в подвыборке, вещественное число от 0 до 1 определяет долю.
  - shuffle (default = True) - перемешать выборку перед разделением

Смещение и дисперсия модели

- Смещение (bias) модели - показывает, насколько в среднем предсказания модели ошибаются относительно истинных значения. Высокое смещение говори о недообучении модели

- Дисперсия (variance) модели - показывает, насколько предсказания модели изменятся, если обучать ее на других данных внутри того же распределения. Высокая дисперсия говорит о переобучении модели.