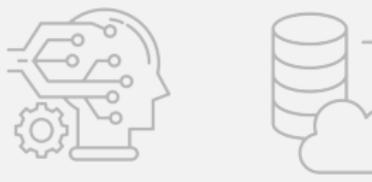
Контролируемое обучение (обучение с учителем)

Сценарий разработки, валидации и внедрения модели ML



Описание задачи

что хотим сделать

Подготовка данных

какие признаки объектов доступны

Предобработка данных

подготовка признаков для решения задачи

Выбор алгоритма

обучение и выбор лучшего алгоритма

Оценка алгоритма

вывод о качестве по метрикам

Валидация модели

независимая, детальная проверка качества алгоритма

Внедрение в пром

включение в промышленные бизнес процессы

Мониторинг качества

контроль качества работы решения в ПРОМ

Классическое Обучение

Данные заранее категоризированы или численные

Данные никак не размечены

С учителем

Предсказать категорию

Предсказать значение

Классификация

«Разложи носки по цвету»

Регрессия

«Разложи галстуки по длине»

Без учителя

Разделить по схожести

Найти зависимости

Выявить последовательности

Кластеризация

«Разложи похожие вещи по кучкам»

Ассоциация

«Найди какие вещи я часто ношу вместе»

Уменьшение Размерности (обобщение)

«Собери из вещей лучшие наряды»

Контролируемое обучение

- Контролируемое обучение (обучение с учителем, supervised learning) – это метод машинного обучения, при котором модель обучается на размеченных данных.
- Размеченные данные – это данные с присвоенной выходной информацией, т.е. у наблюдений помимо входных параметров (независимых переменных) есть выходной параметр/показатель (зависимая/целевая переменная).

Задача обучения с учителем

Необходимо найти закономерности в имеющихся прецедентах и обобщить на объекты, для которых ответы неизвестны.

Имеются:

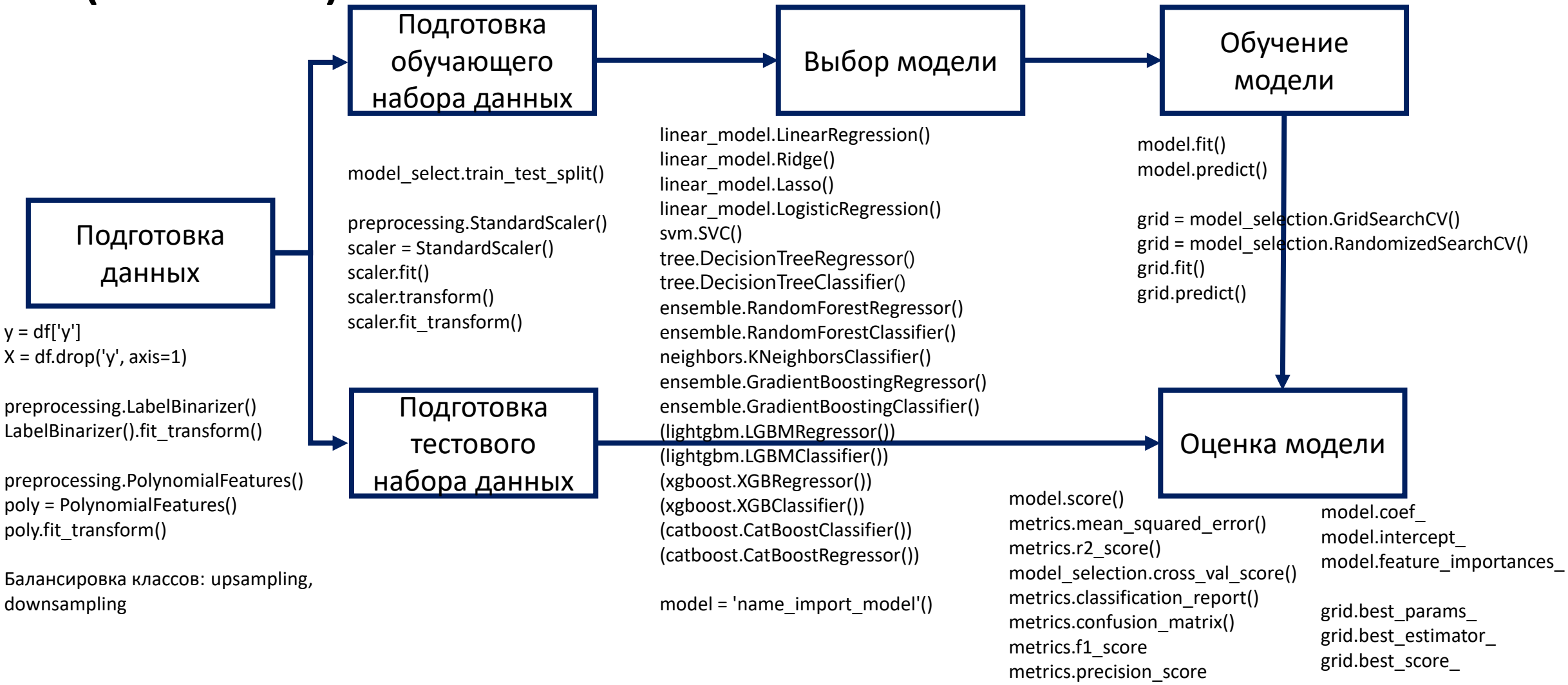
- Множество объектов (ситуаций) X со своими признаковыми описаниями.
- Множество возможных ответов (откликов, реакций) Y.
- Между X и Y существует зависимость a: X → Y, известная на конечной выборке прецедентов (исторических данных) (x_i, y_i) – парах "объект-ответ".
- Множество прецедентов называется обучающей выборкой X_train.

На основе имеющихся прецедентов необходимо построить алгоритм a: X → Y , способный построить достаточно точный ответ для любого допустимого x_i из X.

Алгоритмы/модели контролируемого обучения

- Классификация – выходная переменная категоризирована.
- Регрессия – выходная переменная является числовой величиной.

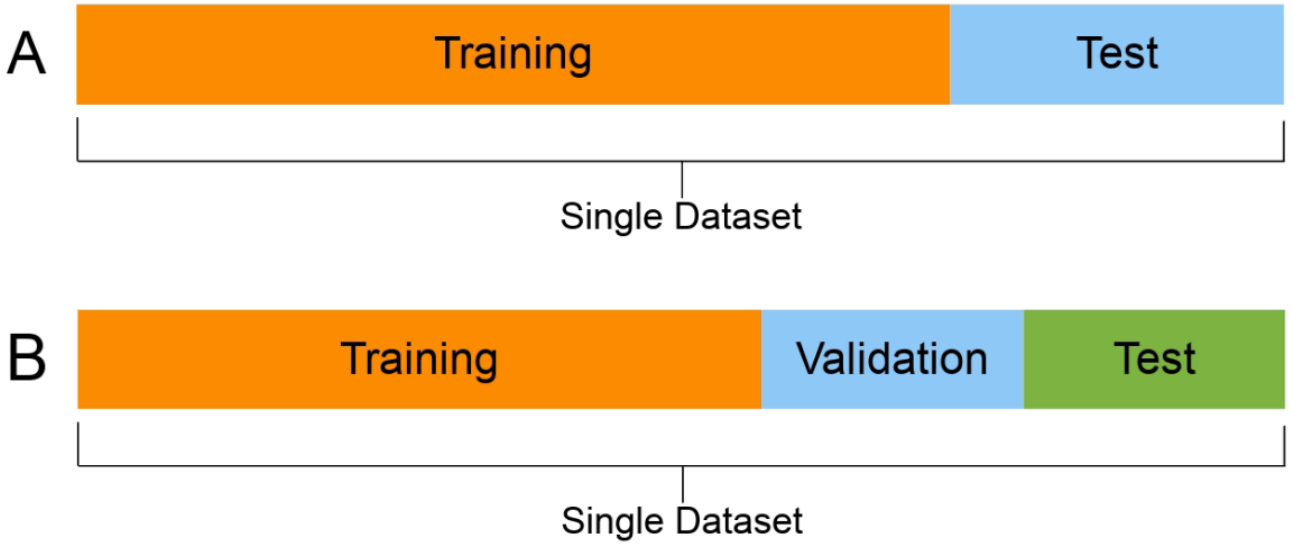
Процесс контролируемого обучения (sklearn)

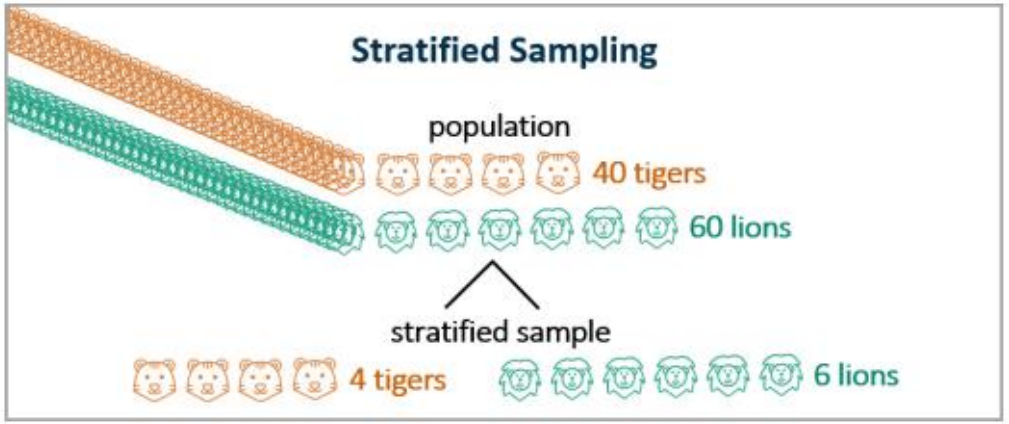


Оценка модели (качества предсказания)

В задачах supervised learning принято делить выборку на 2 (A) или 3 (B) непересекающиеся части. Каждая выборка должна быть репрезентативна!

- Обучающая (training sample). На ней происходит обучение модели.
- Валидационная (validation sample). На ней считают метрики качества, а по ним уже подбирают гиперпараметры. Валидационную выборку используют не всегда.
- Тестовая (test sample). По ней оценивают качество обученной модели.



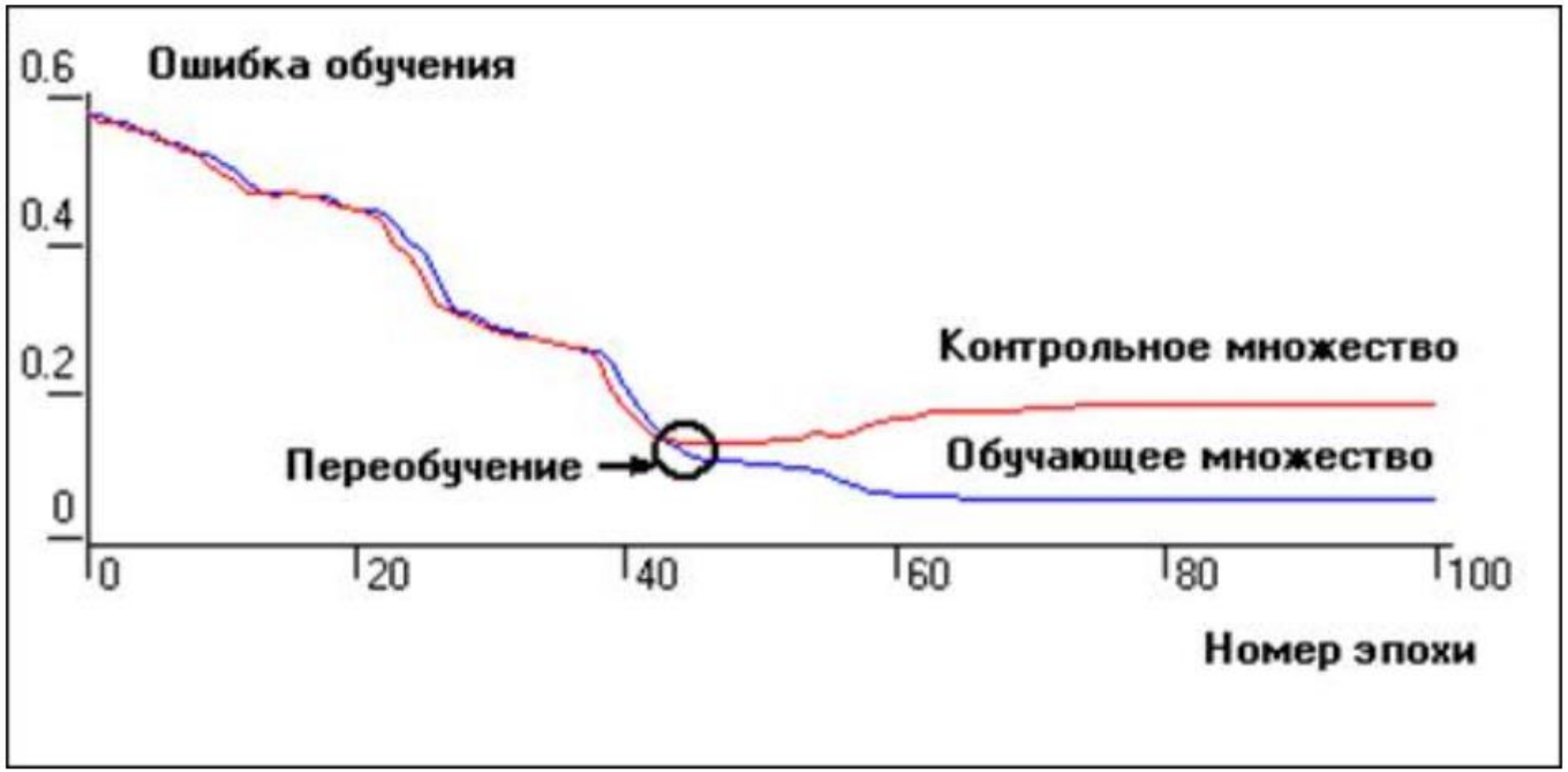


Стратификация — способ балансировки выборок в случае дисбаланса типов объектов.

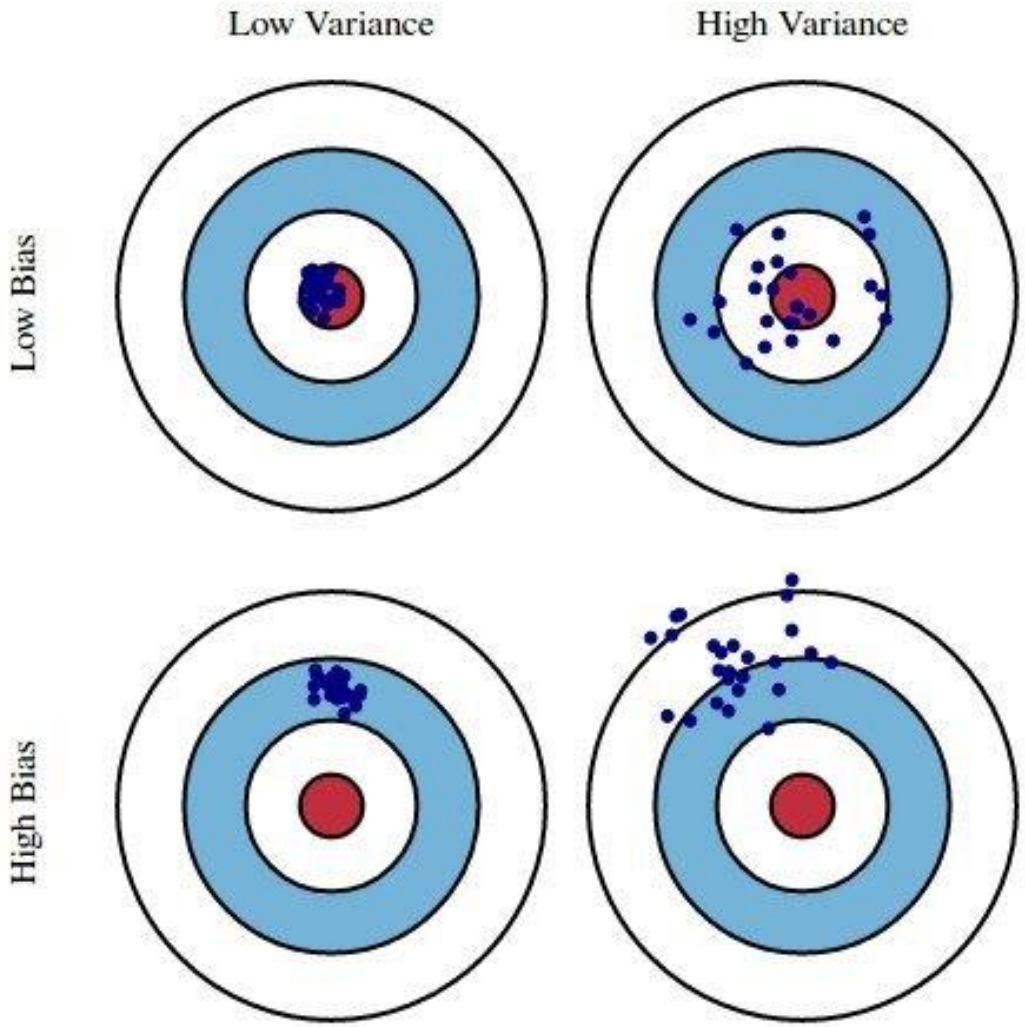
Кросс-валидация (CV)

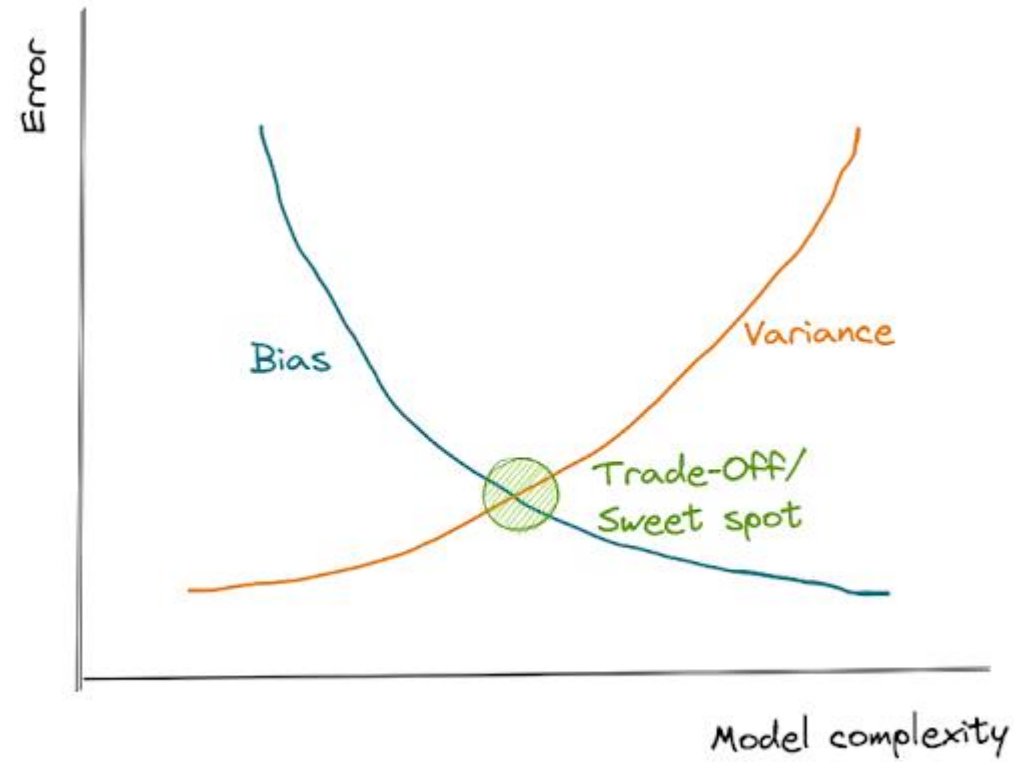
Переобучение

Явление, когда алгоритм хорошо приближает зависимость на обучающей выборке и только на ней, называется переобучением.



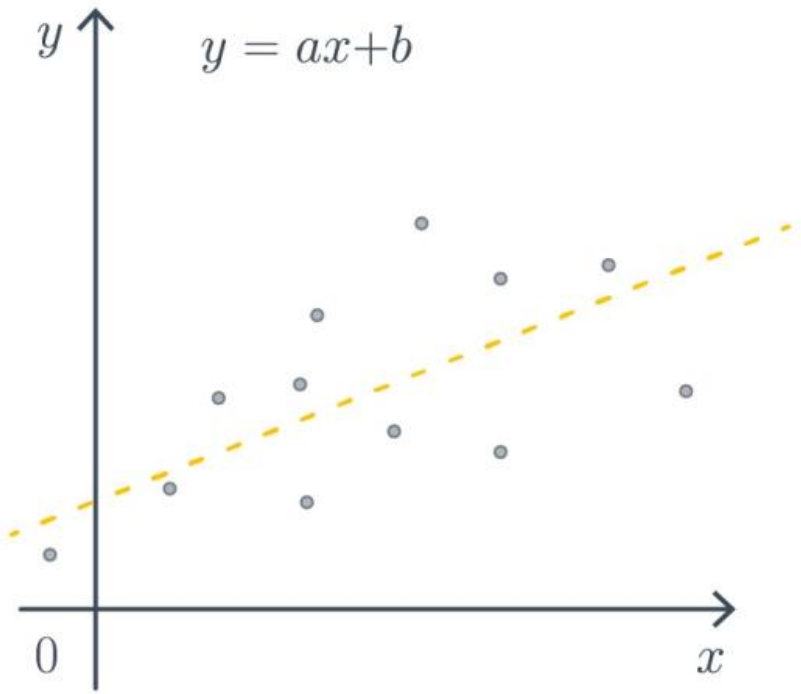
Проблема Bias-Variance (Смещение-Дисперсия)





Линейная модель и линейная классификация

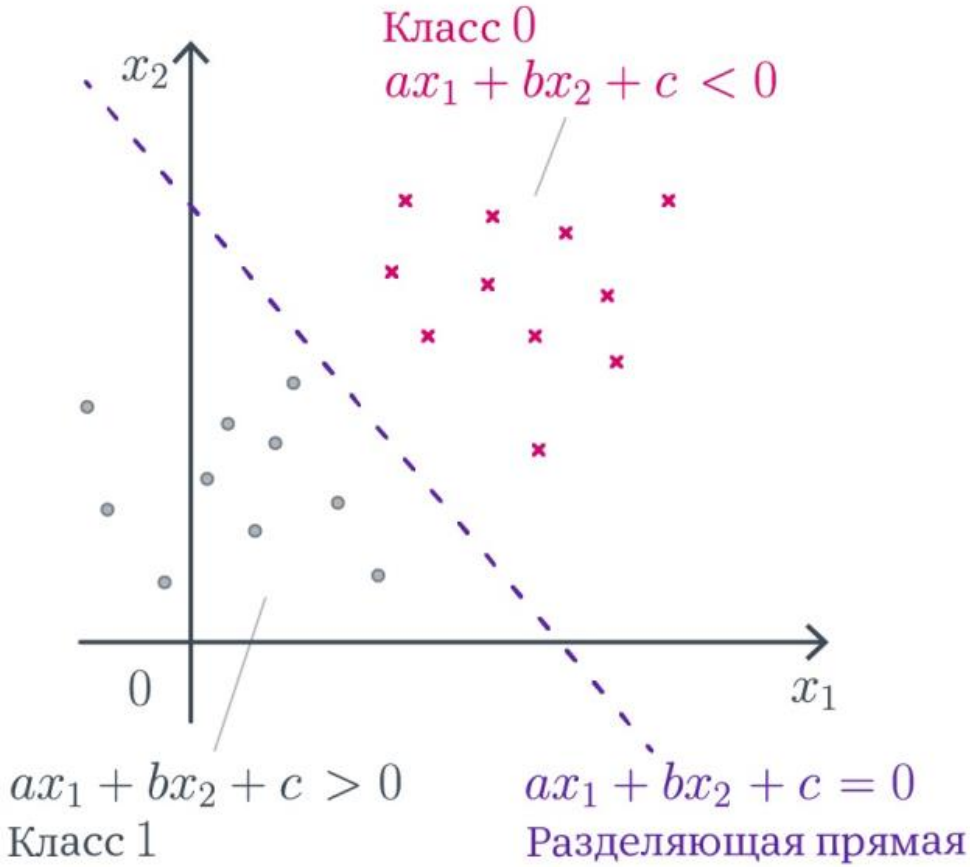
Регрессия



x — (единственный) признак

y — таргет

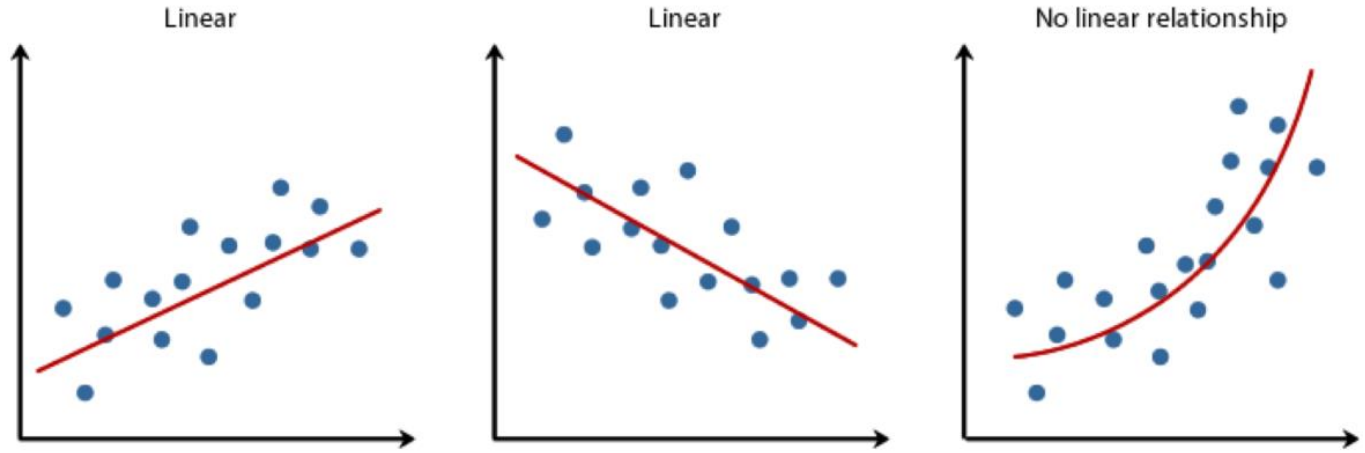
Классификация



x_1, x_2 — признаки

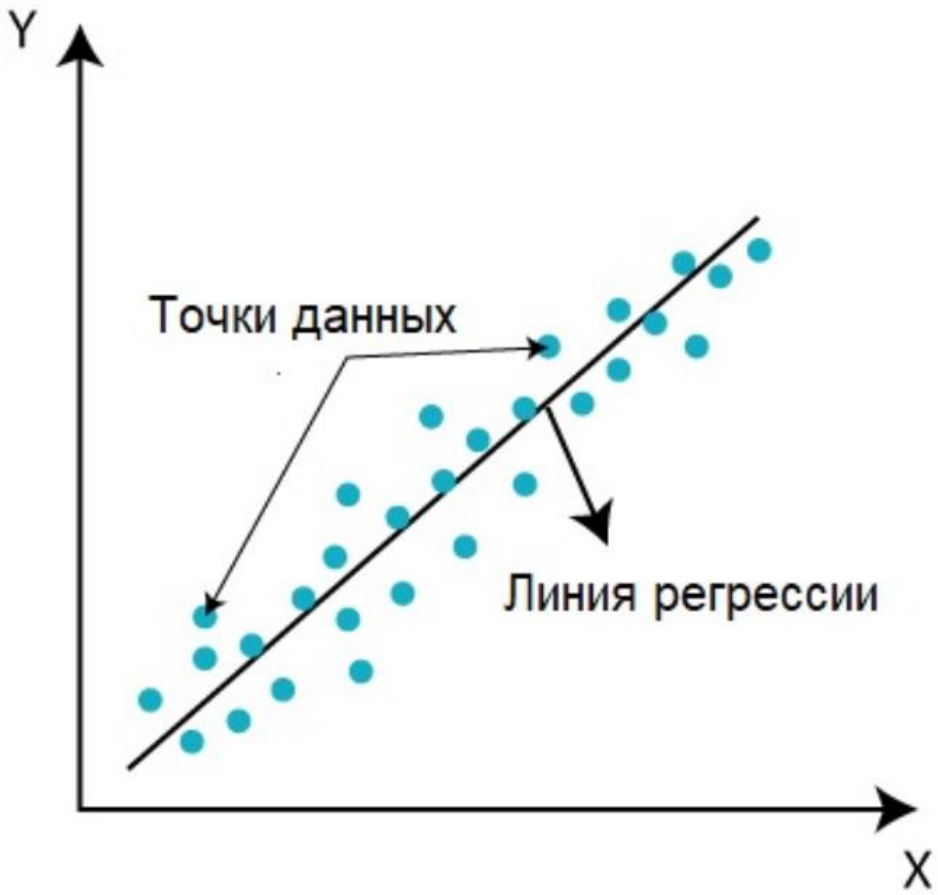
$$y = sgn\left(\sum_{i=1}^{n} \omega_i x_i + \omega_0\right)$$

Линейные и нелинейные зависимости

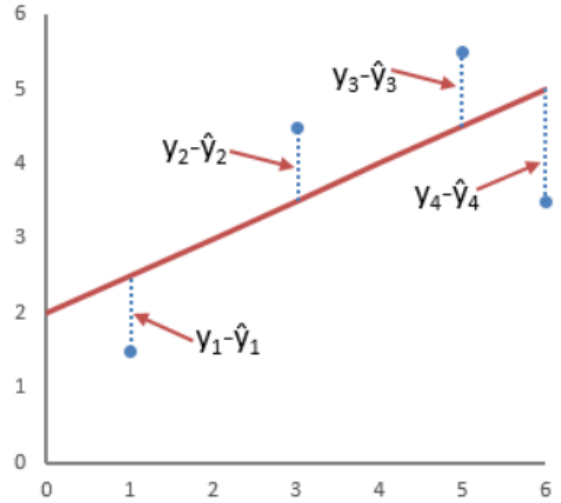


No.	Machine Learning Model	Category
1	Linear Regression (LR)	Linear
2	Linear Discriminant Analysis (LDA)	Linear
3	Support Vector Machine (SVM)	Linear
4	Quadratic Discriminant Analysis (QDA)	Non-linear
5	Random Forest (RF)	Non-linear
6	K-Nearest Neighbors (KNN)	Non-linear
7	Nearest Centroid	Linear
8	Naive Bayes	Linear
9	Perceptron	Linear
10	Decision Tree (DT)	Non-linear
11	Dummy	Non-linear
12	Neural Networks	Non-linear

Линейная регрессия



$$y = w_1x_1 + w_2x_2 + \ldots + w_kx_k + b,$$



где y - целевая переменная,
x_i – i-й признак объекта x,
w_i - вес i-го признака,
b - свободный член

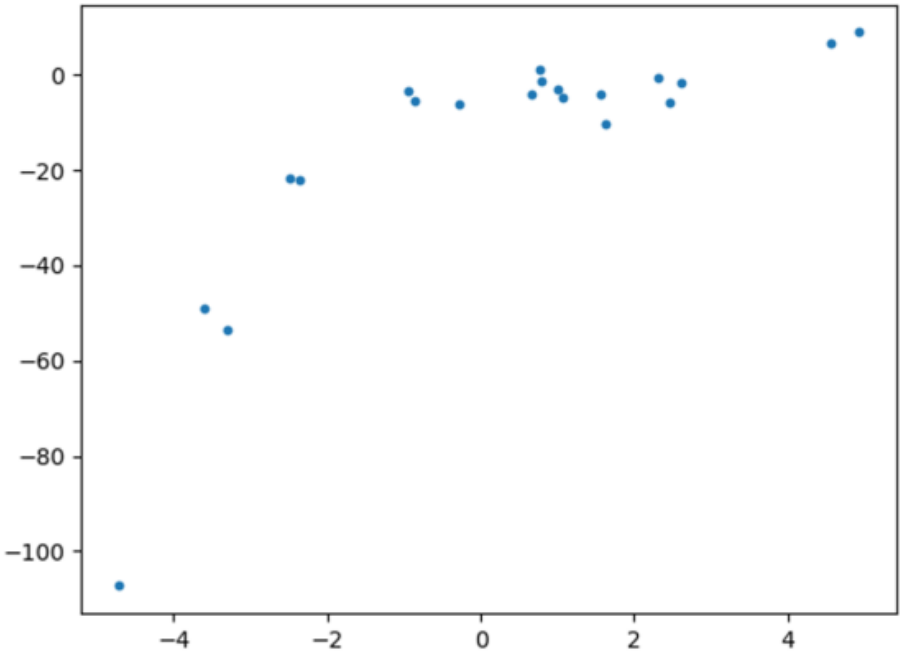
$$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \qquad \boldsymbol{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} = \sqrt{MSE}$$

$$SS_{total} = SUM\ (y_i - y_{avg})^2 \qquad R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

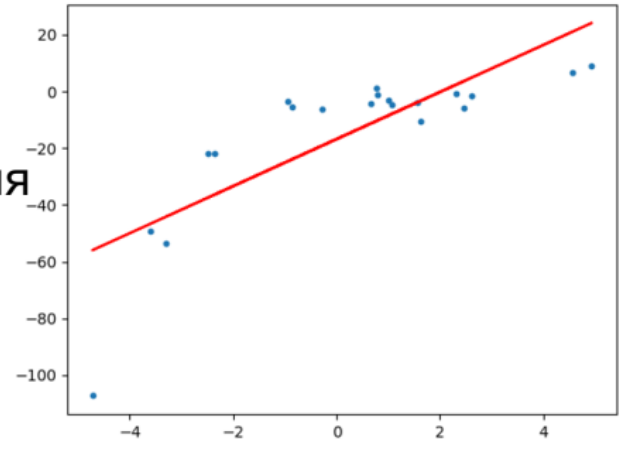
Виды регрессий

Исходные данные



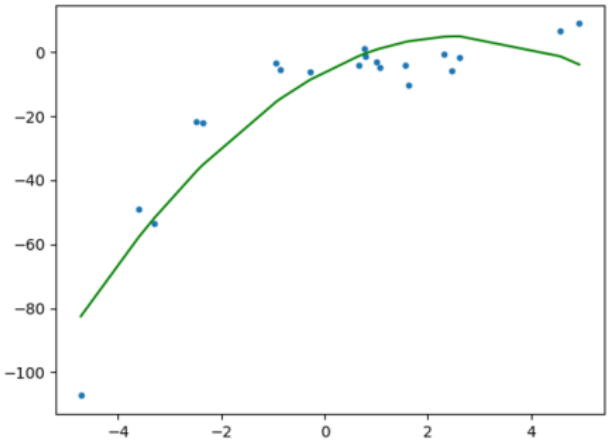
Линейная регрессия

$$Y = \theta_0 + \theta_1 x$$



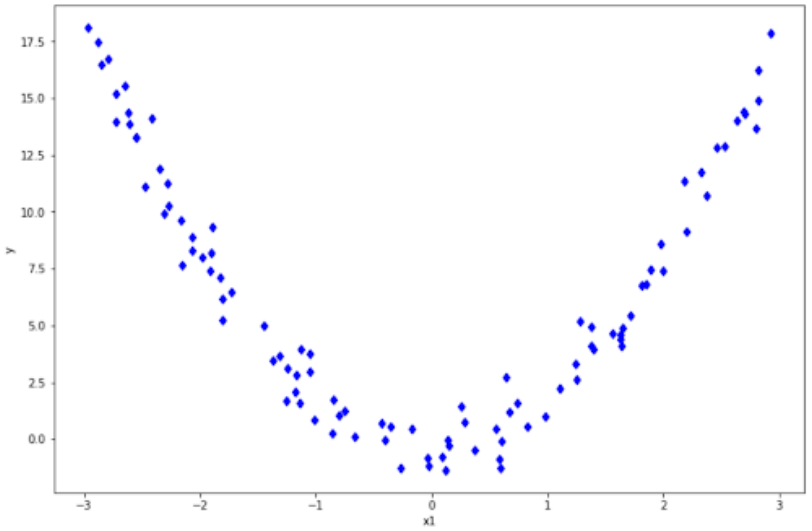
Полиномиальная регрессия

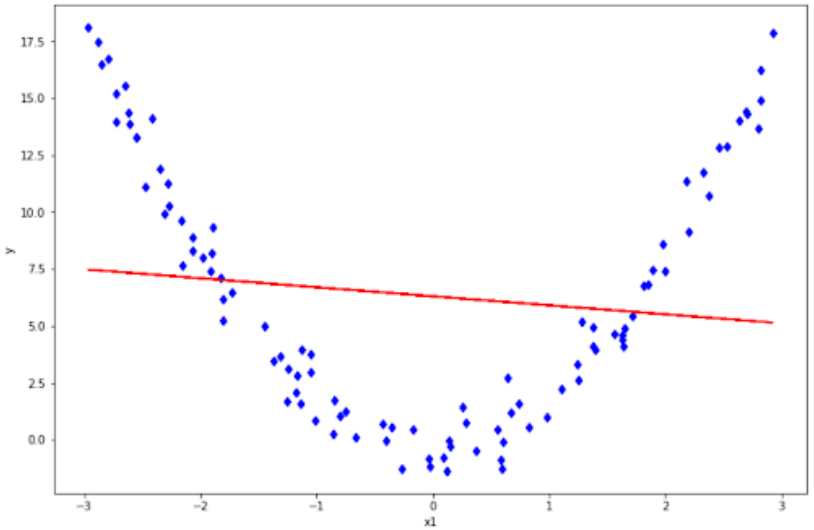
$$Y = \theta_0 + \theta_1 x + \theta_2 x^2$$

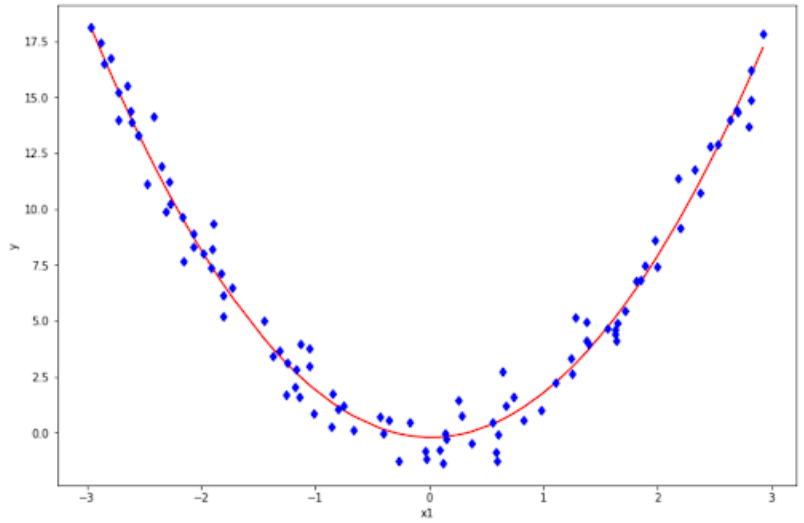


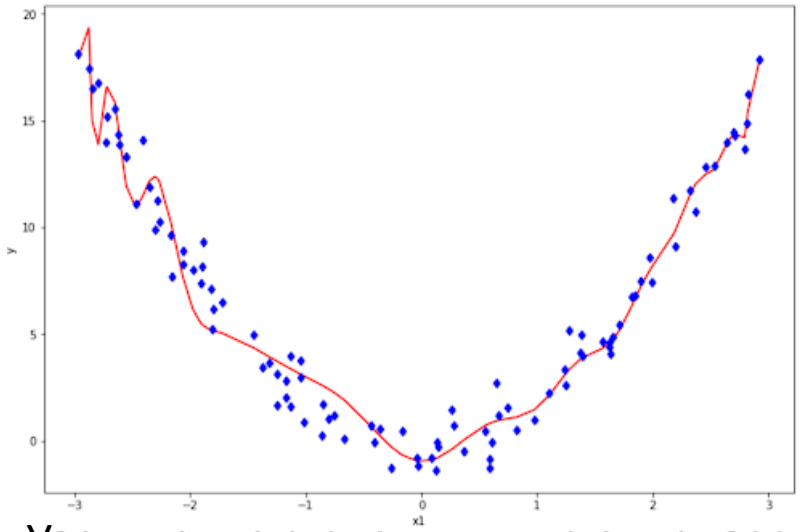
Полиномиальная регрессия означает приближение данных (x_i, y_i) полиномом k-й степени.

Полиномиальная регрессия

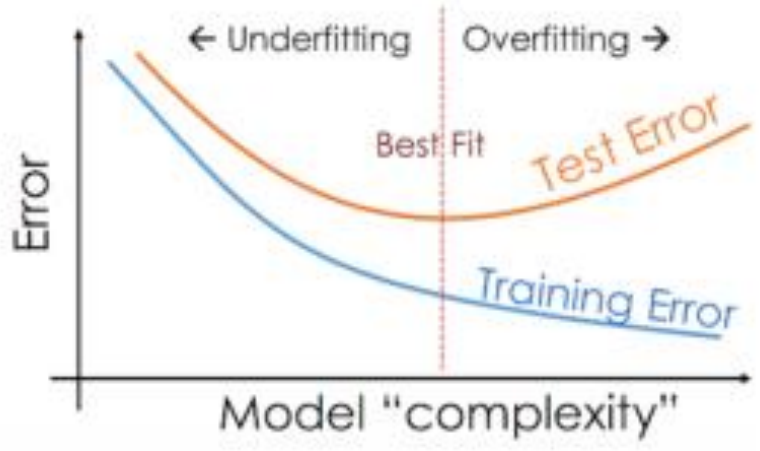








Линейная регрессия плохо описывает данные.
Явление - недообучение.



Увеличение степени лучше описывает текущие данные, но не новые.
Явление - переобучение.

Регуляризация – борьба с переобучением

Регуляризация - введение дополнительного ограничения на размер весов w_i

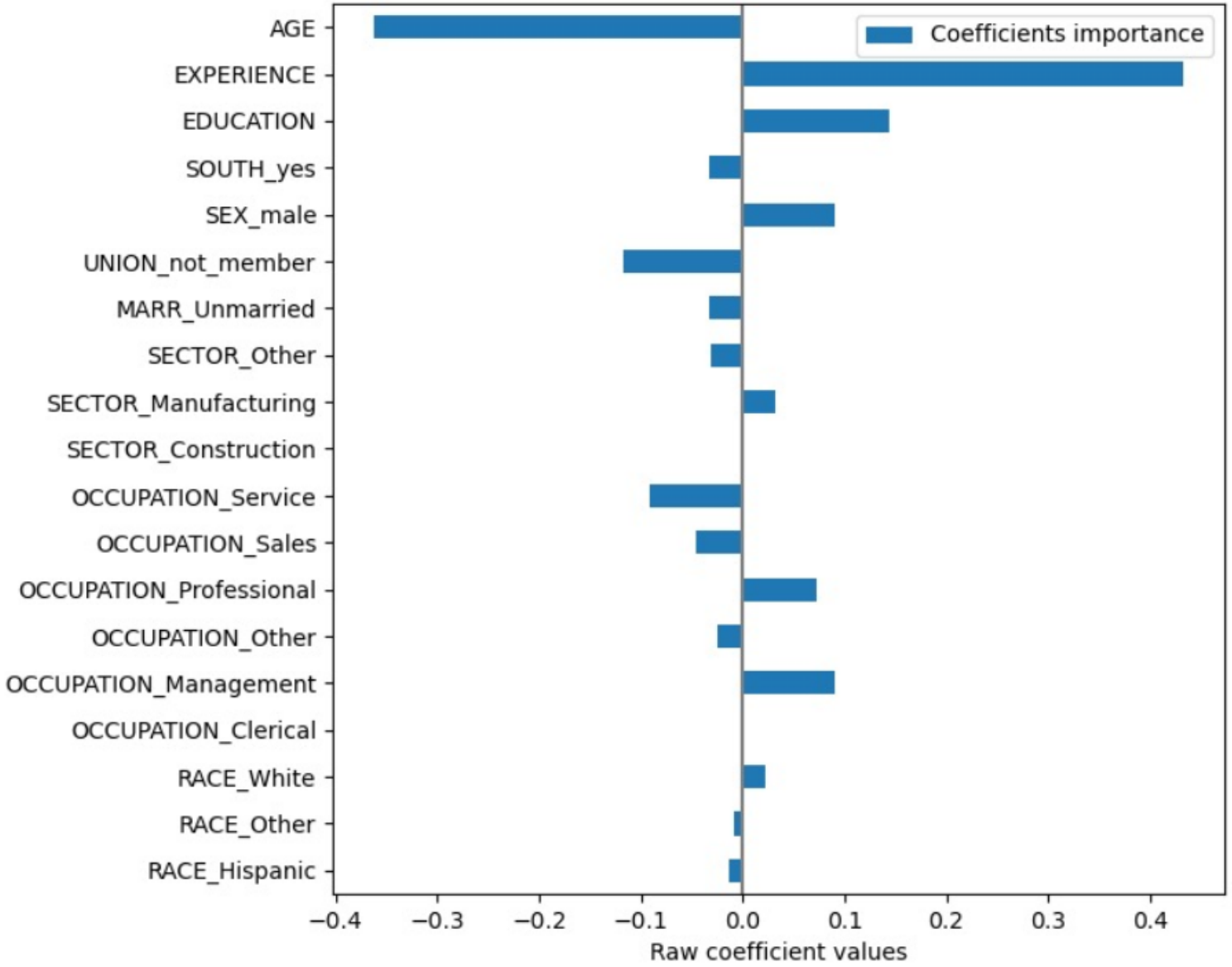
L1 (Lasso) $\|w\|_1 = \lambda_1 \sum_{j=1}^{d} |wj|$

L2 (Ridge) $\|w\|^2 = \lambda_2 \sum_{j=1}^{d} w_j^2$

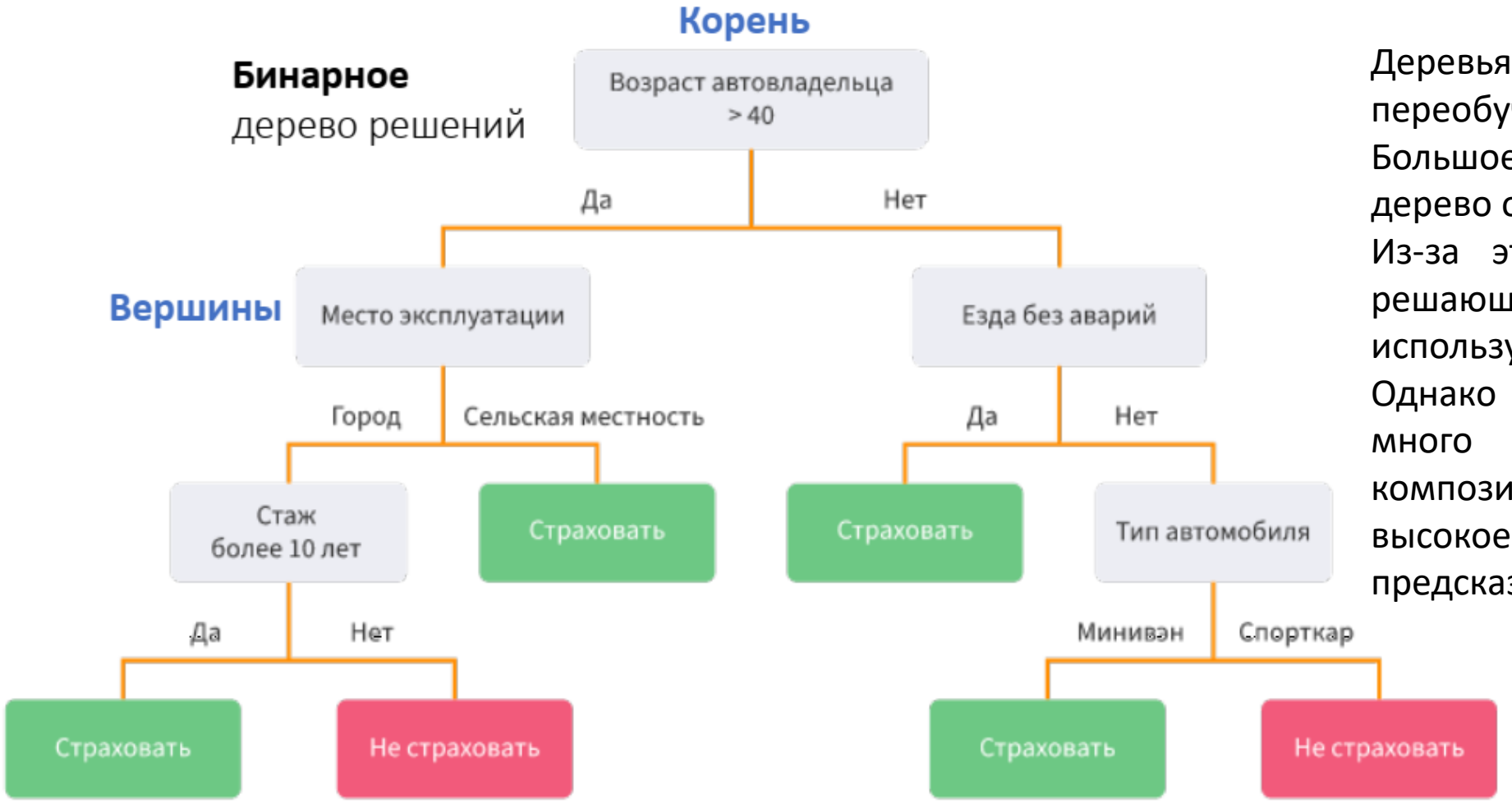
ElasticNet $\|w\|_3 = \lambda_1 \sum_{j=1}^{d} |wj| + \lambda_2 \sum_{j=1}^{d} w_j^2$

MSE + ||w|| → min

Важность признаков



Дерево решений

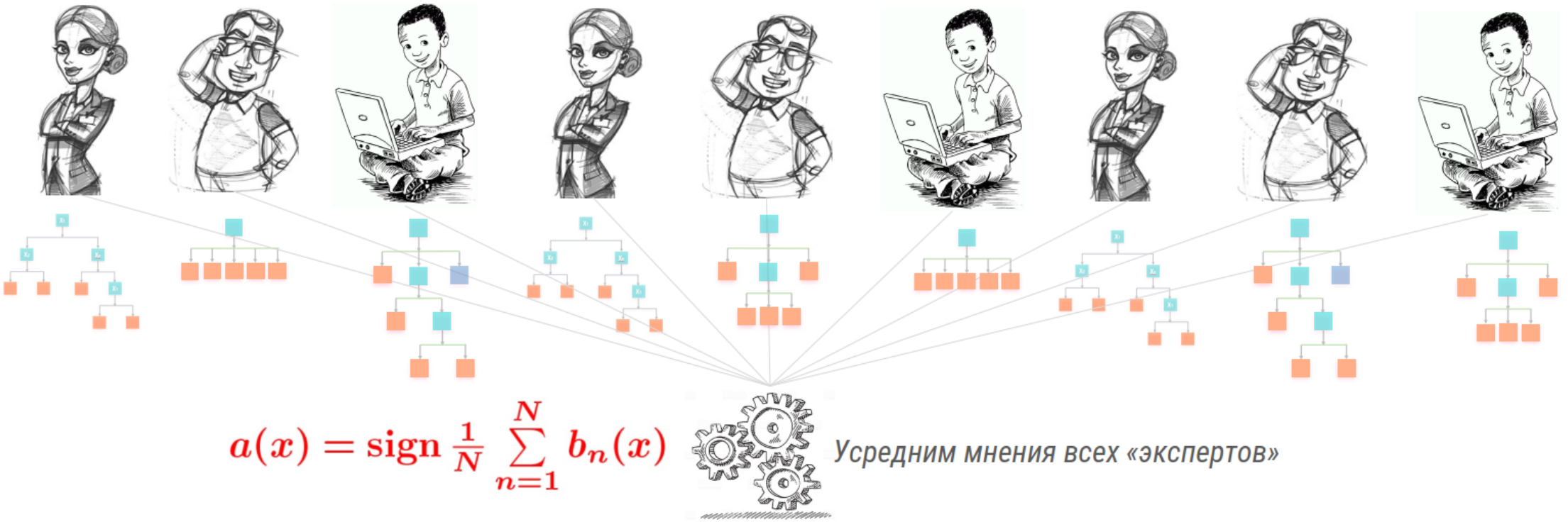


Лист: *среднее (регрессия) или класс (классификация)*

Деревья сильно склонны к переобучению.
Большое качественное дерево обучать сложно.
Из-за этого сами по себе решающие деревья редко используются на практике.
Однако можно объединять много деревьев в композиции, которые дают высокое качество предсказаний.

Композиции моделей

Спросим мнение каждого эксперта по отдельности



$$a(x) = \text{sign}\, \frac{1}{N} \sum_{n=1}^{N} b_n(x)$$

Усредним мнения всех «экспертов»

Получим наиболее точный ответ

Композиции моделей

Композиции моделей: бэггинг, бустинг, стекинг.

Бэггинг:

Набирается m случайных объектов с повторением и на них обучается модель. Эти действия повторяются N раз. Затем усредняются ответы всех полученных моделей.

Случайный лес основан на данном подходе и представляет собой композицию деревьев.

Бустинг (градиентный):

Обучаются N простых моделей $b_n(x)$ с обучаемый весом (вкладом) каждой модели γ_n, ответ композиции a_N на объекте x определяется по формуле: $a_N(x) = \sum_{n=0}^{N} \gamma_n b_n(x)$. Каждая следующая модель обучается так, чтобы она уменьшала ошибку всех уже построенных.

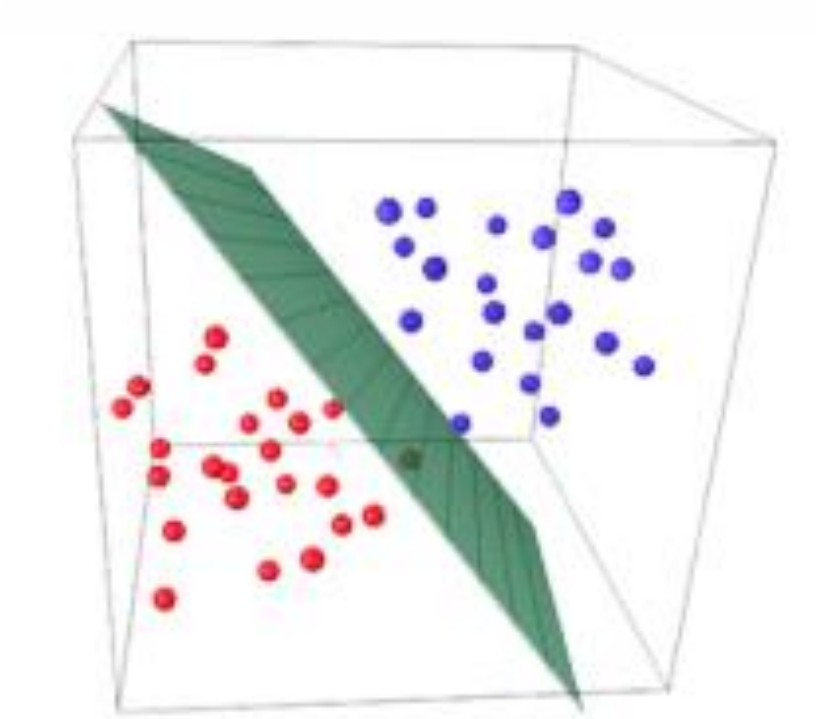
Стекинг:

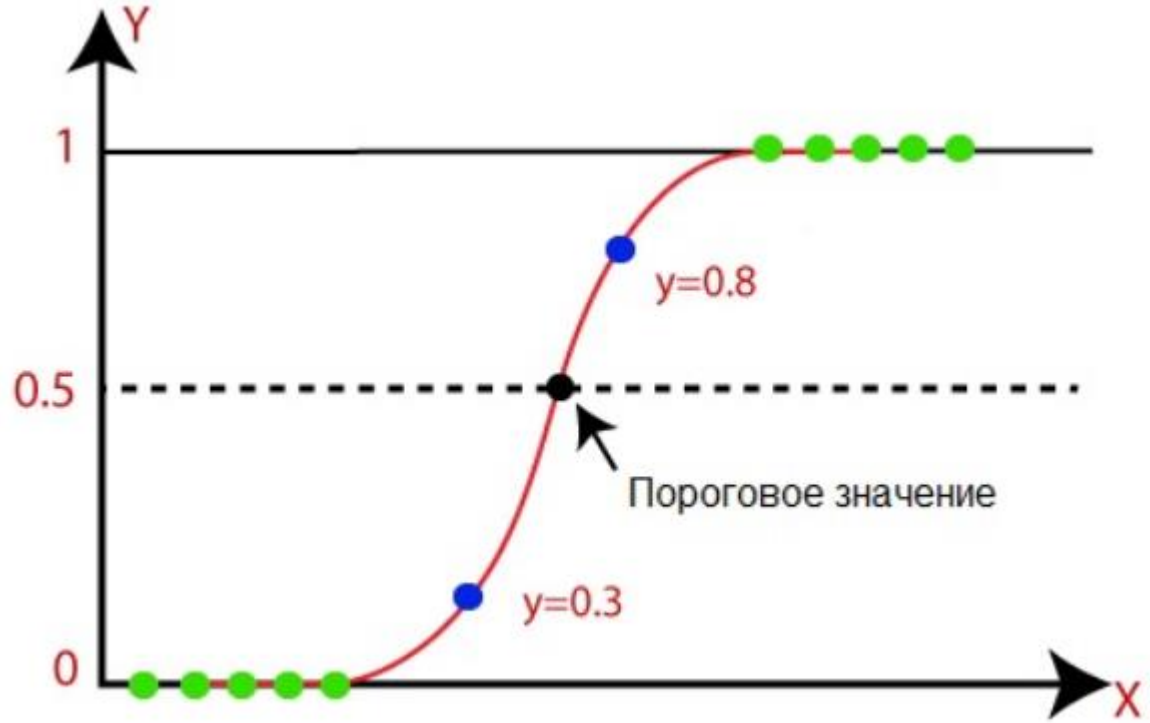
В начале обучаются базовые модели, а потом на их выходах обучается еще одна модель.

Логистическая регрессия

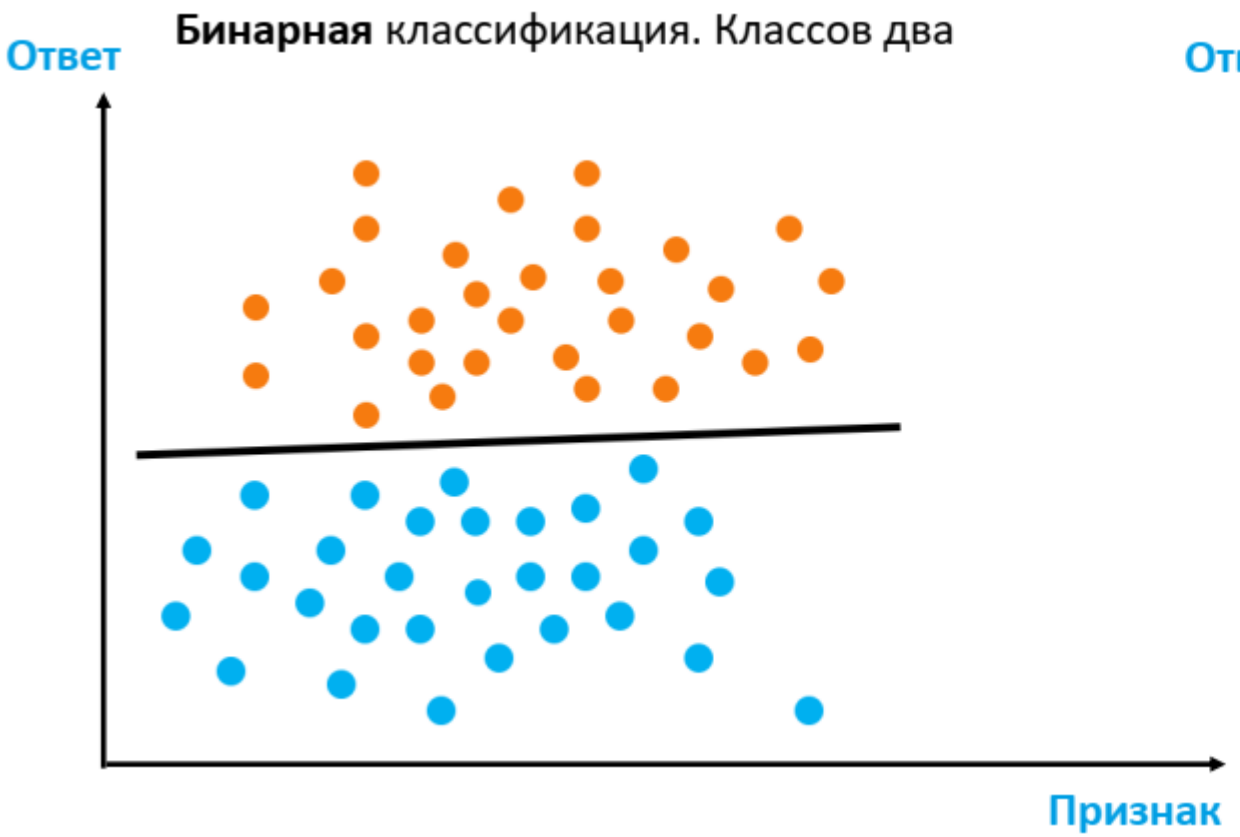
В логистической регрессии $\hat{y} \in [0, 1]$.

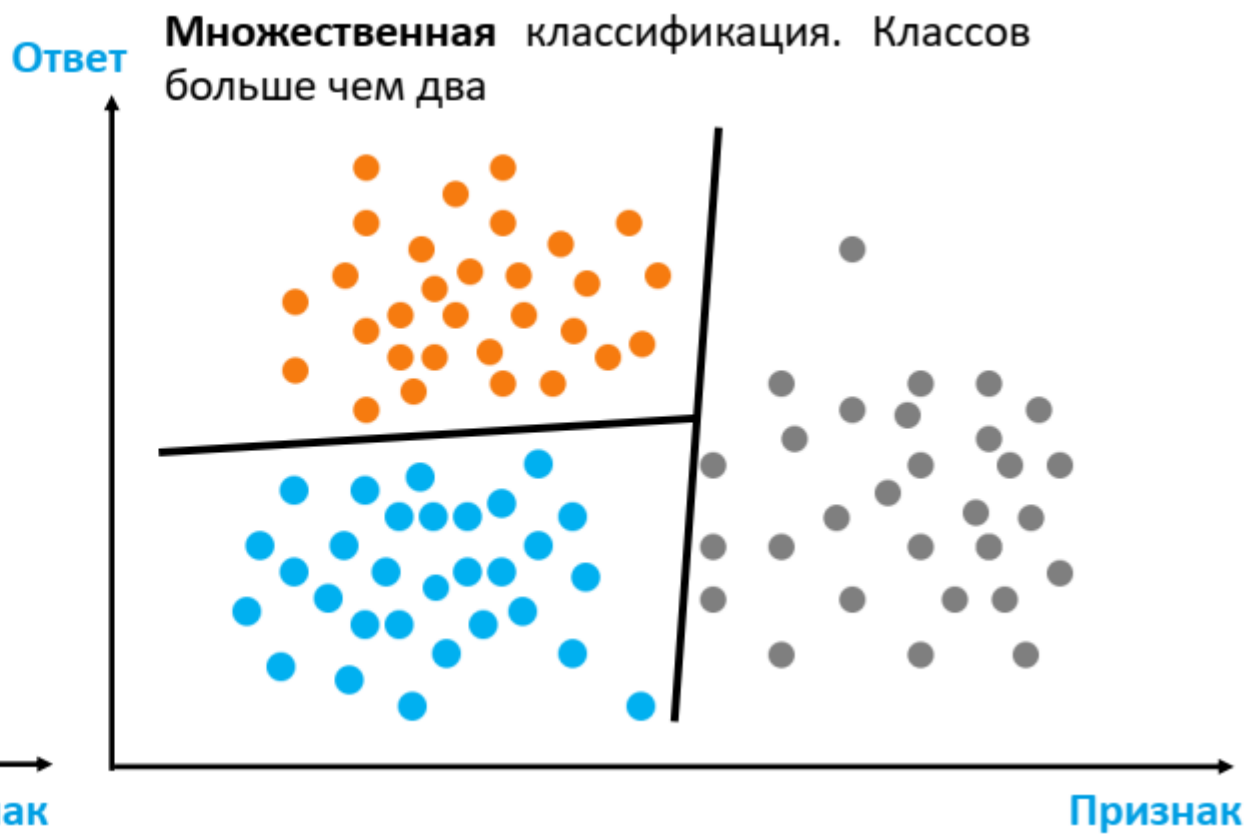
$\hat{y} = \frac{1}{1+e^{-(w_1x_1 + w_2x_2 + ... + wkxk + b)}}$ - уравнение логистической регрессии.





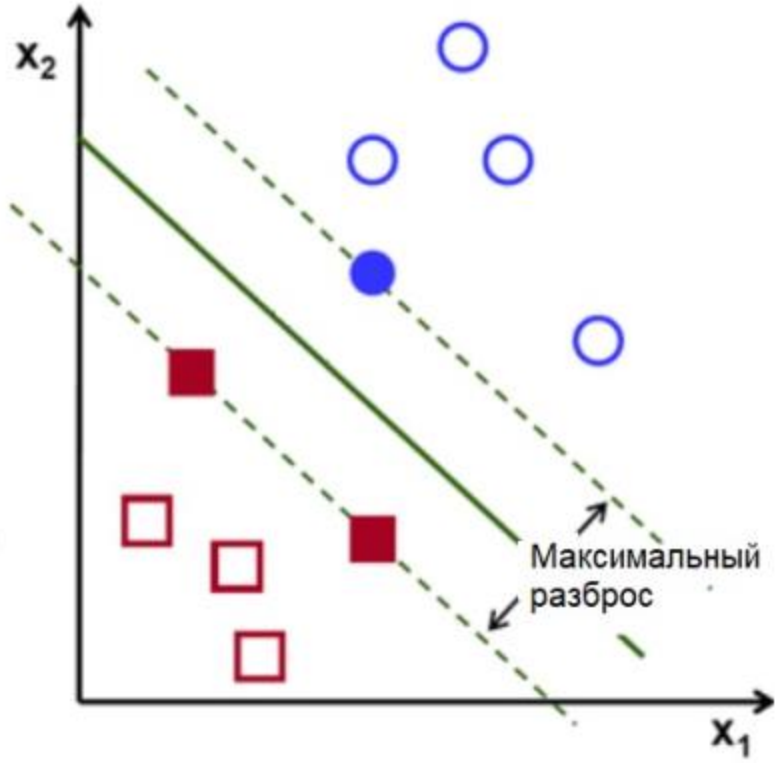
Классификация





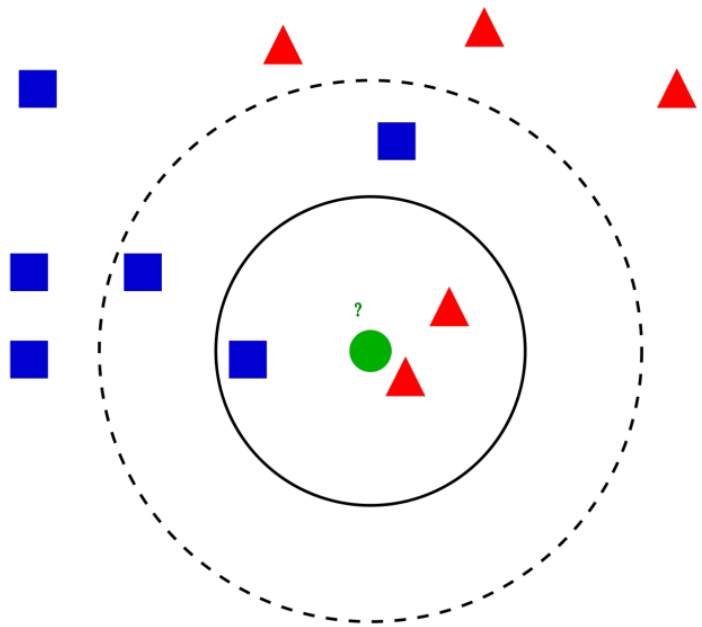
Метод опорных векторов (SVM)

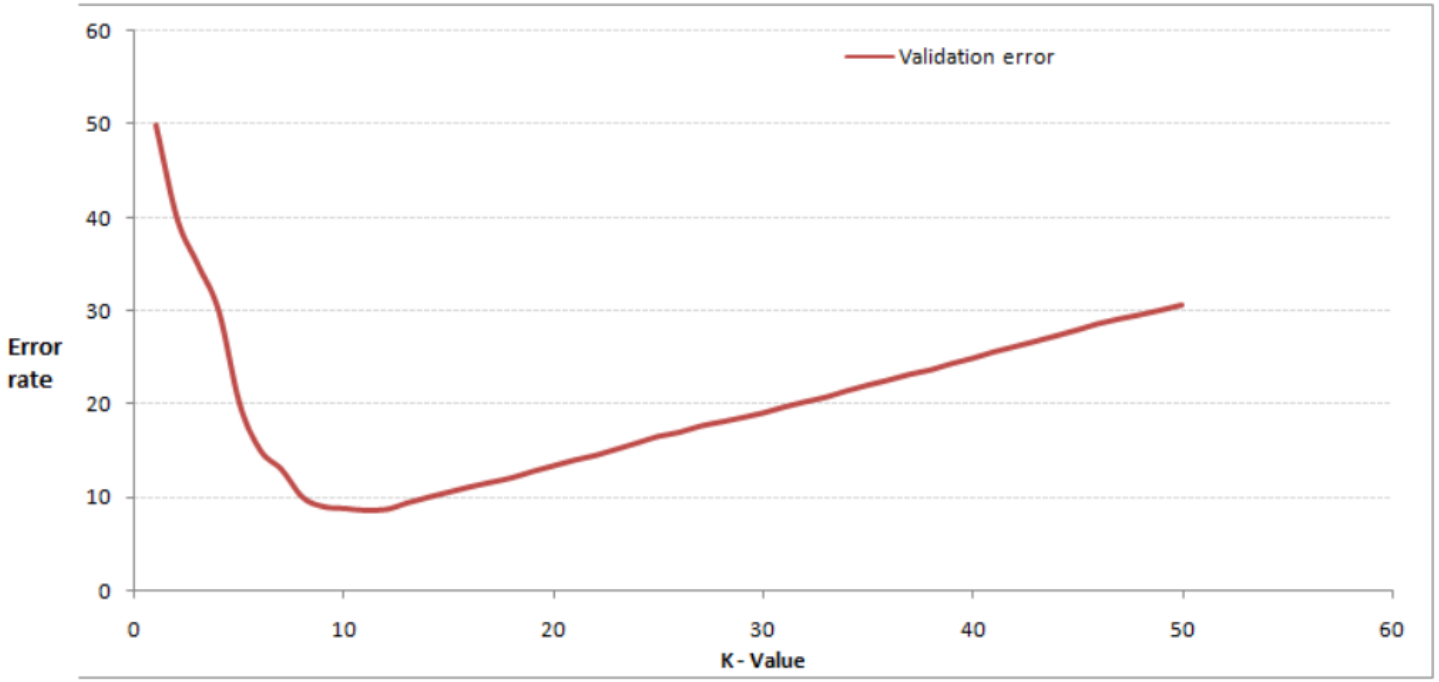
SVM - классификатор, который пытается построить такую линию, чтобы самым точным образом разделить между собой разные типы объектов.



Метод k-ближайших соседей

Будем относить к одному классу объекты, расстояние между которыми минимально.





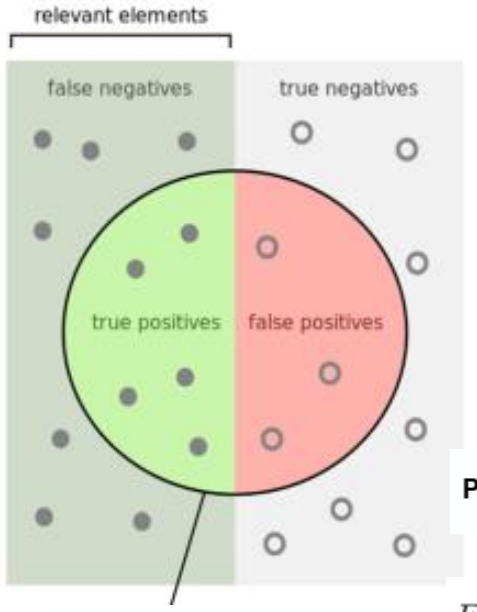
Матрица ошибок (метрики качества модели классификации)

Матрица ошибок отражает количество ошибочно/верно определенных наблюдений.

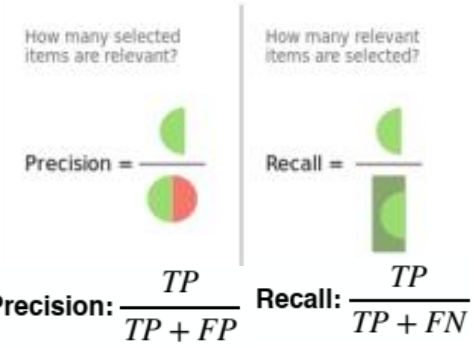
По матрице ошибок можно вычислить такие метрики, как Accuracy, Precision, Recall, F1.

МАТРИЦА ОШИБОК

	$y = 1$ (Фактический класс)	$y = 0$ (Фактический класс)
$\hat{y} = 1$ (Прогнозный класс)	True Positive (TP)	Type 1 Error *Ложная тревога*
$\hat{y} = 0$ (Прогнозный класс)	Type 2 error *Пропуск цели*	True Negative (TN)



Accuracy: $\frac{TP + TN}{n}$



Precision: $\frac{TP}{TP + FP}$ **Recall:** $\frac{TP}{TP + FN}$

$$F_\beta = (\beta^2 + 1)\frac{Recall \cdot Precision}{Recall + \beta^2 Precision}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Метрики качества

Accuracy - доля верно определенных наблюдений.

Неинформативна в случае несбалансированности классов. Не учитывает цены разных типов ошибок. Из-за этого редко используется на практике.

Precision (точность прогноза) - отражает точность классификатора в определении единиц.

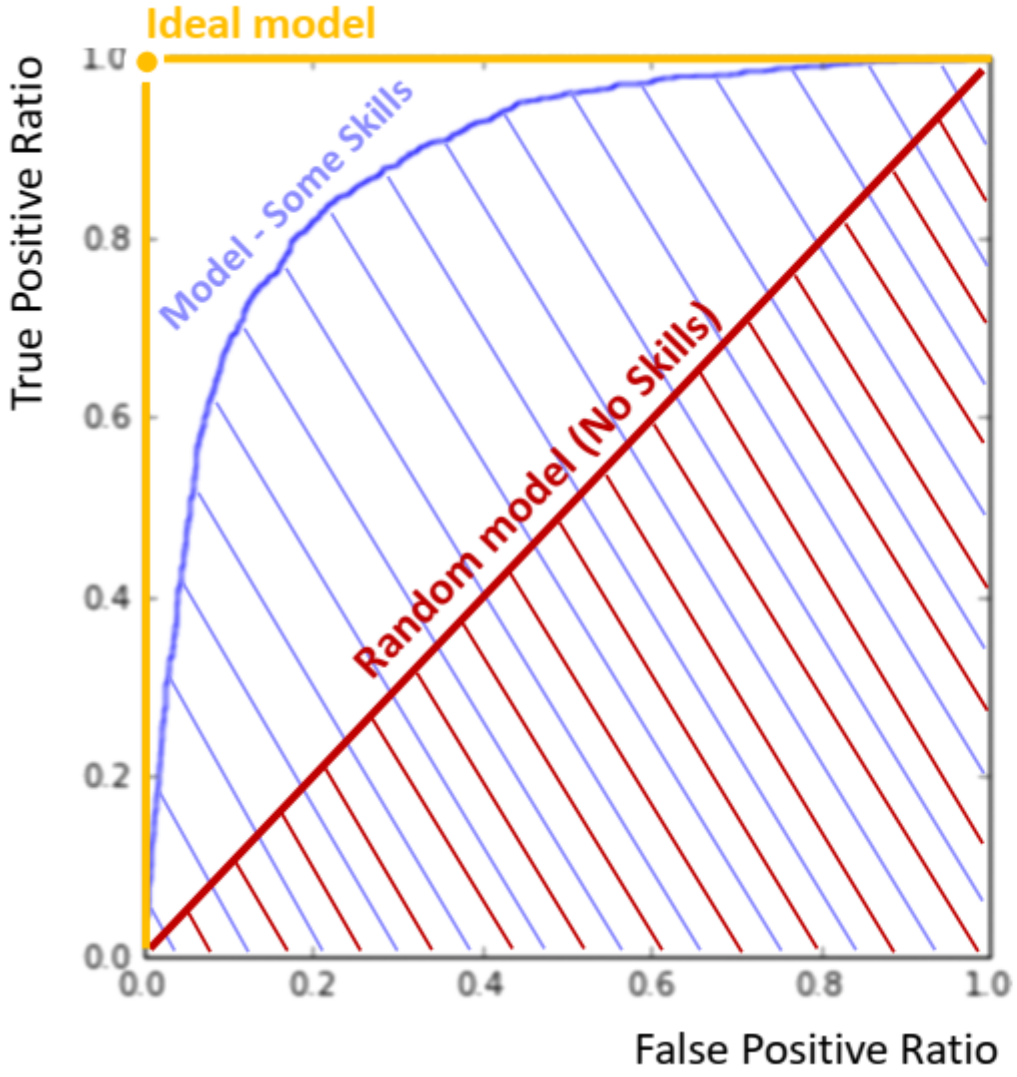
Используется для минимизации ложных срабатываний (ложных тревог), т.е. когда важно не ошибиться в прогнозе. И если модель отнесла объект к классу «1» то это действительно так.

Recall (полнота прогноза) - показывает процент фактических «единиц», определяемых моделью.

Используется для минимизации пропуска цели, т.е. когда важно определить все объекты истинного класса «1».

F1 (гармоническое среднее) - описывает качество модели в целом, в равной мере учитывая Precision и Recall. Чем ближе метрика к 1, тем модель лучше.

ROC-AUC (кривая ошибок)



МАТРИЦА ОШИБОК

	$y = 1$ (Фактический класс)	$y = 0$ (Фактический класс)
$\hat{y} = 1$ (Прогнозный класс)	True Positive Ratio	False Positive Ratio
$\hat{y} = 0$ (Прогнозный класс)	False Negative Ratio	True Negative Ratio

$$ROC = \frac{\text{площадь закрашенной фигуры}}{\text{площадь идеальной модели}}$$

$ROC = [0, 1]$

$ROC = 1$ – идеальная модель

$ROC = 0{,}5$ – соответствует случайному гаданию

$ROC < 0{,}5$ – классификатор действует с точностью до наоборот

$ROC = 0$ – полная противополождность идеальной модели

Использование Scikit-learn (sklearn) для линейной регрессии

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
model.score(X_test, y_test) #score для линейной регрессии определяет R^2
r2_score(y_test, y_pred) #R^2 – коэффициент детерминации
mean_squared_error(y_test, y_pred) ** 0.5
```