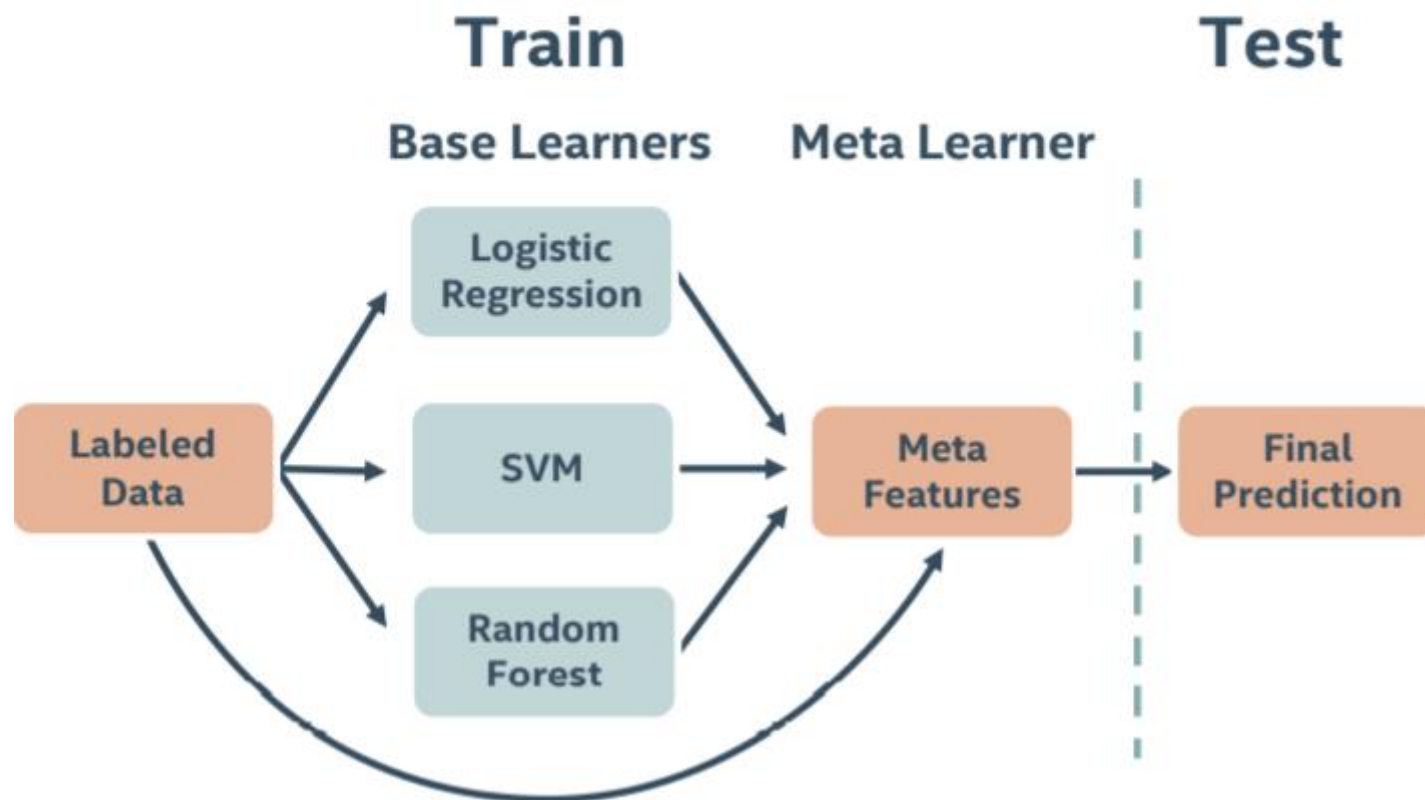


## 3.3 Stacking: объединение разнородных моделей

# Stacking: объединение разнородных моделей

- Объединяем модели любого вида для создания метамодели
- Стекинг применяется к моделям, построенным с помощью различных алгоритмов обучения
- Как бэггинг, но не ограничивается деревьями решений
- Выходы базовых моделей образуют входы мета модели, может рекомбинировать с данными



# Stacking

Идея стекинга состоит в том, чтобы выучить нескольких разных слабых учеников и **объединить их, обучив метамоделю** для вывода предсказаний, основанных на множественных предсказаниях, возвращаемых этими слабыми моделями.

Итак, нам нужно определить две вещи для построения нашей модели стека:  $L$  учеников, которых мы хотим обучить, и метамоделю, которая их объединяет.

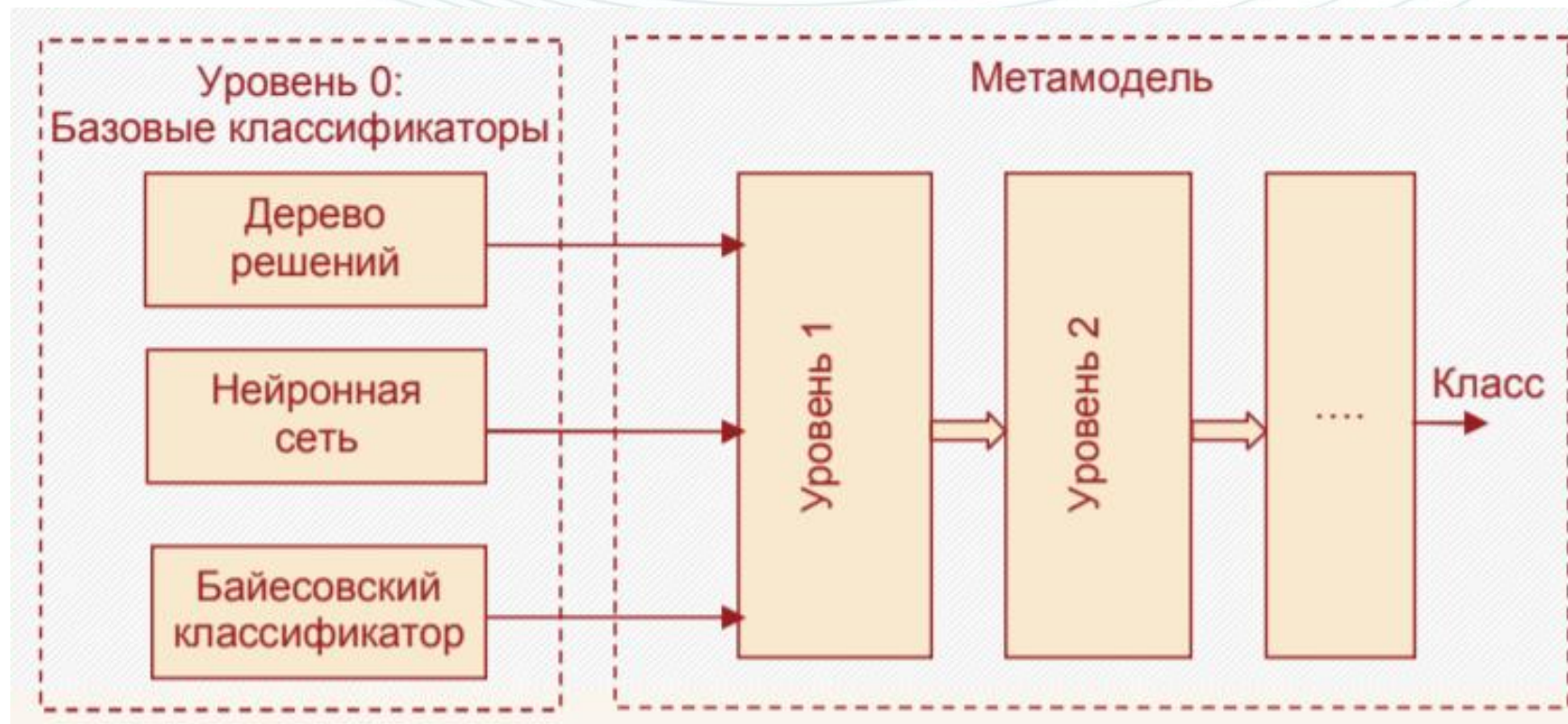
Например, для задачи классификации мы можем в качестве слабого ученика выбрать классификатор KNN, логистическую регрессию и SVM и принять решение обучить нейронную сеть в качестве метамоделю. Затем нейронная сеть примет в качестве входных данных результаты трех наших слабых учеников и научится давать окончательные прогнозы на их основе.





# Stacking: объединение разнородных моделей

- Выход базовых учащихся может быть объединен большинством голосов или взвешенным голосованием
- Стекинг пытается обучить каждый классификатор, используя алгоритм метаобучения, который позволяет обнаружить лучшую комбинацию выходов базовых моделей



- Для борьбы с переобучением и «утечкой» при обучении метамоделю необходимо дополнительно выделить часть доступных для обучения данных в виде тестового набора (hold-out data)
- Нужно помнить об увеличении сложности модели

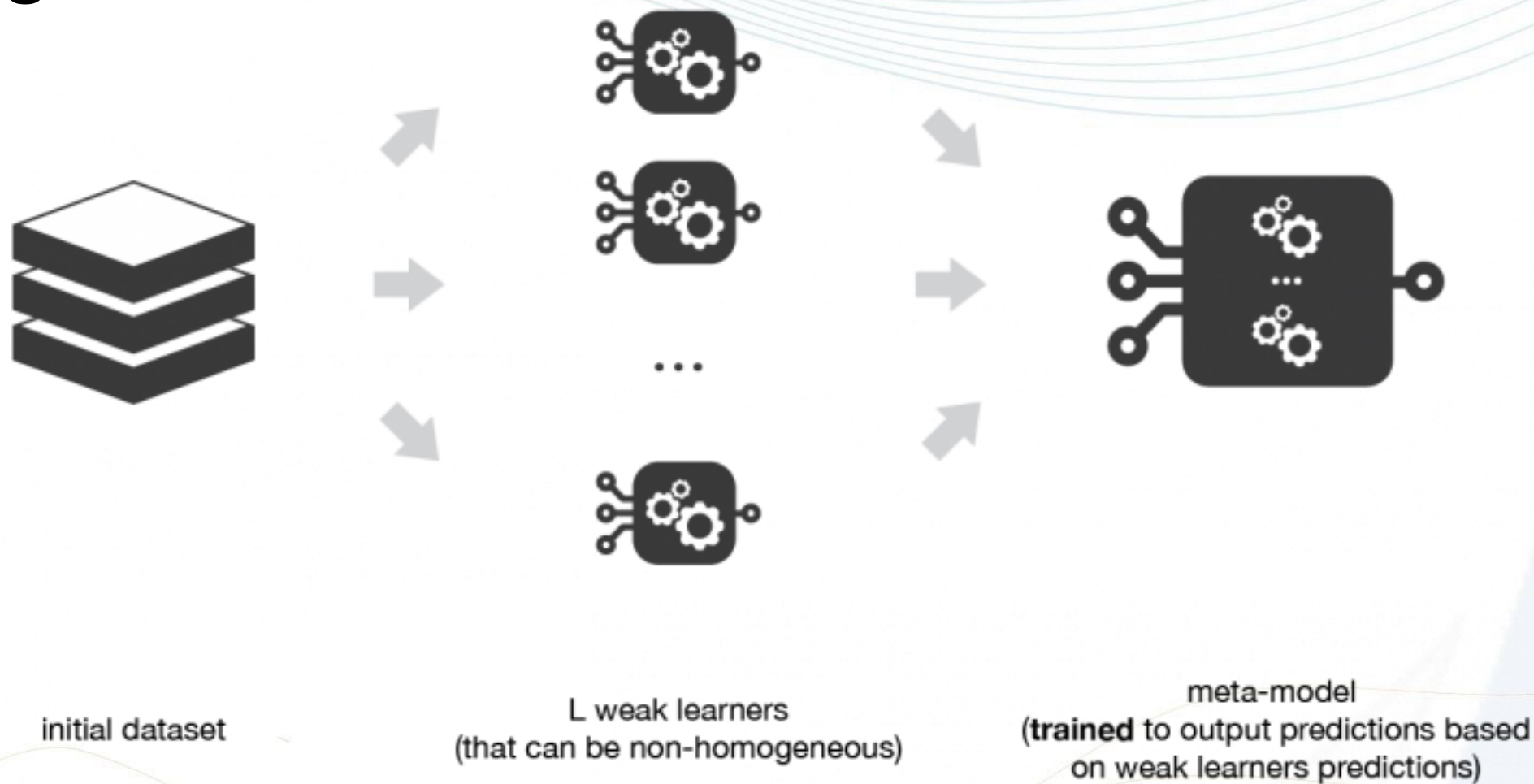
# Stacking

Итак, предположим, что мы хотим обучить стековый ансамбль, составленный из  $L$  слабых учеников. Затем мы должны выполнить следующие шаги:

1. разделить тренировочные данные на две части
2. выберите  $L$  слабых учеников и обучите их на данных первого **фолда (части)**
3. для каждого из  $L$  слабых учеников сделайте прогнозы для объектов из второго **фолда**
4. обучить метамоделю во второй раз, используя в качестве входных данных прогнозы, сделанные слабыми учениками



# Stacking



Стекинг состоит в обучении метамодел для получения результатов, основанных на результатах, полученных несколькими слабыми учениками нижнего уровня



# Stacking

На предыдущих этапах мы разбили датасет на две части, потому что прогнозы данных, которые использовались для обучения слабых учеников, **не имеют отношения к обучению метамоделей**. Таким образом, очевидным недостатком этого разделения нашего датасета на две части является то, что у нас есть только половина данных для обучения базовых моделей и половина данных для обучения метамоделей. Чтобы преодолеть это ограничение, мы можем следовать некоторому подходу «k-fold кросс-обучение» (аналогичному тому, что делается в k-fold кросс-валидации), таким образом все объекты могут быть использованы для обучения мета-модели: для любого объекта предсказание слабых учеников делается на примерах этих слабых учеников, обученных на k-1 фолдах, которые не содержат рассматриваемого объекта. Другими словами, он обучается по k-1 фолдам, чтобы делать прогнозы для оставшегося фолда для объектов в любых фолдах. Таким образом, мы можем создать соответствующие прогнозы для каждого объекта нашего датасета, а затем обучить нашу метамоделю всем этим прогнозам.



# Многоуровневый стекинг

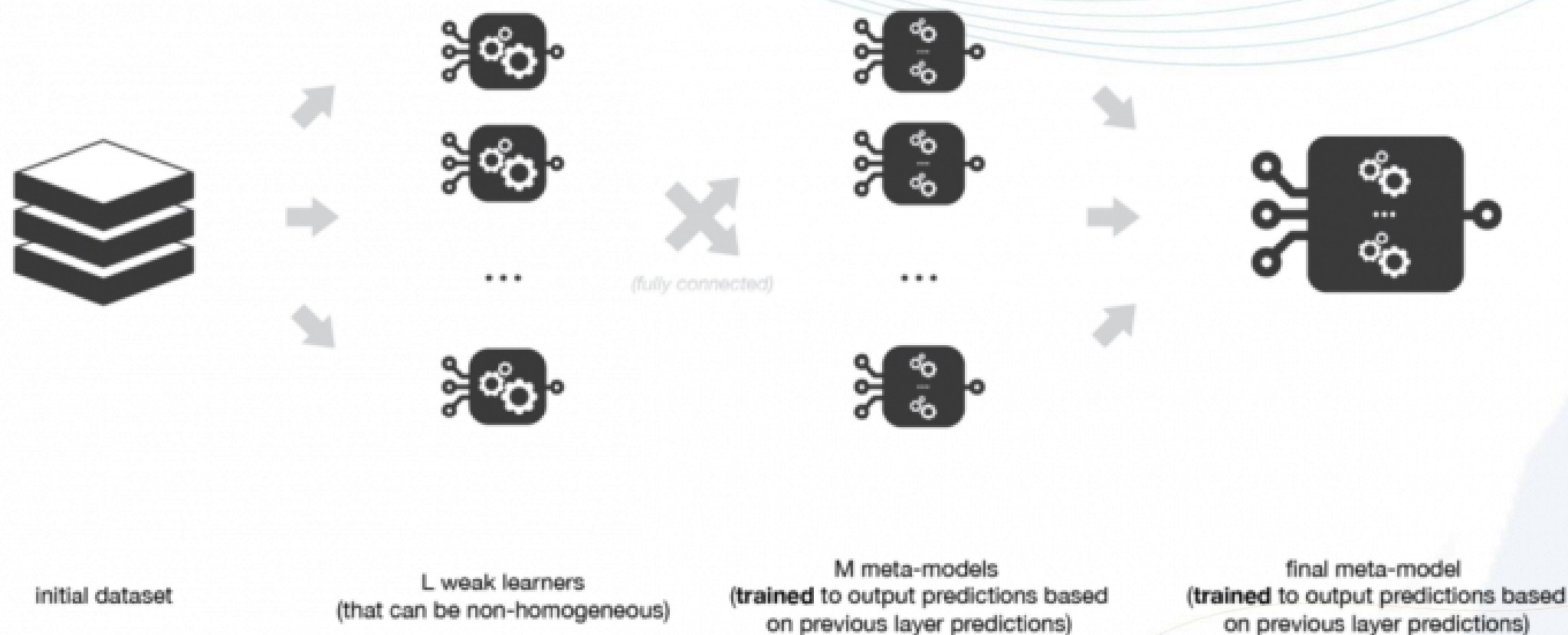
Возможное расширение стекинга — многоуровневый стекинг. Он заключается в выполнении **стекинга с несколькими слоями**. В качестве примера давайте рассмотрим стекинг в 3 уровня. На первом уровне (слое) мы подходим к  $L$  слабым ученикам, которые были выбраны. Затем на втором уровне вместо обучения одной метамоделі к прогнозам слабых моделей (как это было описано в предыдущем подразделе) мы обучаем  $M$  таких метамоделей. Наконец, на третьем уровне мы обучаем последнюю метамоделю, которая принимает в качестве входных данных прогнозы, возвращаемые  $M$  метамоделями предыдущего уровня.

С практической точки зрения, обратите внимание, что для каждой метамоделі различных уровней многоуровневой модели стекового ансамбля мы должны выбрать алгоритм обучения, который может быть почти любым, который мы хотим (даже алгоритмы, уже используемые на более низких уровнях). Мы также можем упомянуть, что **добавление уровней может быть требовательным к данным** (если k-fold-подобная методика не используется, тогда требуется больше данных) **или требовательным к времени** (если используется k-fold-подобная техника и должны быть обучены много моделей).





# Многоуровневый стекинг



Многоуровневый стекинг предполагает несколько уровней стекинга: некоторые метамодели обучаются на выходных данных, возвращаемых метамоделями более низкого уровня, и так далее. Здесь мы представили 3-х слойную модель стекинга.



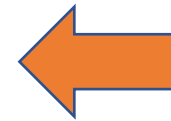
# VotingClassifier: синтаксис

Импортируем класс, содержащий метод классификации

```
from sklearn.ensemble import VotingClassifier
```

Создадим экземпляр класса

```
VC = VotingClassifier(estimator_list, voting='hard',  
                      weights=estimator_weight_list)
```



список базовых  
моделей и как их  
комбинировать

Обучим модель на обучающей выборке, а затем прогнозируем ожидаемое значение на тестовой выборке

```
VC = V.fit(X_train, y_train)  
y_predict = VC.predict(X_test)
```

Настройка ДОПОЛНИТЕЛЬНОГО УРОВНЯ методом перекрестной проверки с hold-out set. Используйте **VotingRegressor** для регрессии



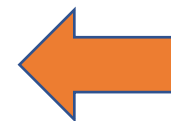
# StackingRegressor: синтаксис

Импортируем класс, содержащий метод классификации

```
from sklearn.ensemble import GradientBoostingRegressor  
from sklearn.ensemble import StackingRegressor
```

Создадим экземпляр класса

```
reg = StackingRegressor(  
    estimators=estimators,  
    final_estimator=GradientBoostingRegressor(random_state=42))
```



список базовых  
моделей и как их  
обучать на следующем  
уровне

Обучим модель на обучающей выборке, а затем прогнозируем  
ожидаемое значение на тестовой выборке

```
reg.fit(X_train, y_train)
```

```
y_pred = reg.predict(X_test)
```

Настройка ДОПОЛНИТЕЛЬНОГО УРОВНЯ методом перекрестной проверки с  
hold-out set. Используйте **StackingClassifier** для классификации





# Итоги

- ❑ Ансамблевые методы — это парадигма машинного обучения, в которой несколько моделей (часто называемых слабыми учениками или базовыми моделями) обучаются для решения одной и той же проблемы и объединяются для повышения производительности.
- ❑ Основная гипотеза состоит в том, что, если мы правильно объединим слабых учеников, мы сможем получить более точные и/или надежные модели
- ❑ В методах бэггинга несколько экземпляров одной и той же базовой модели обучаются параллельно (независимо друг от друга) на разных бутстрэп выборках, а затем агрегируются некоторым процессом «усреднения»
- ❑ Вид операции усреднения, выполняемый над моделями, обученными на (почти) независимых одинаково распределенных выборках в методах бэггинга, в основном позволяет нам получить ансамблевую модель с меньшим разбросом, чем ее компоненты: поэтому базовые модели с низким смещением, но высоким разбросом хорошо приспособлены для бэггинга.

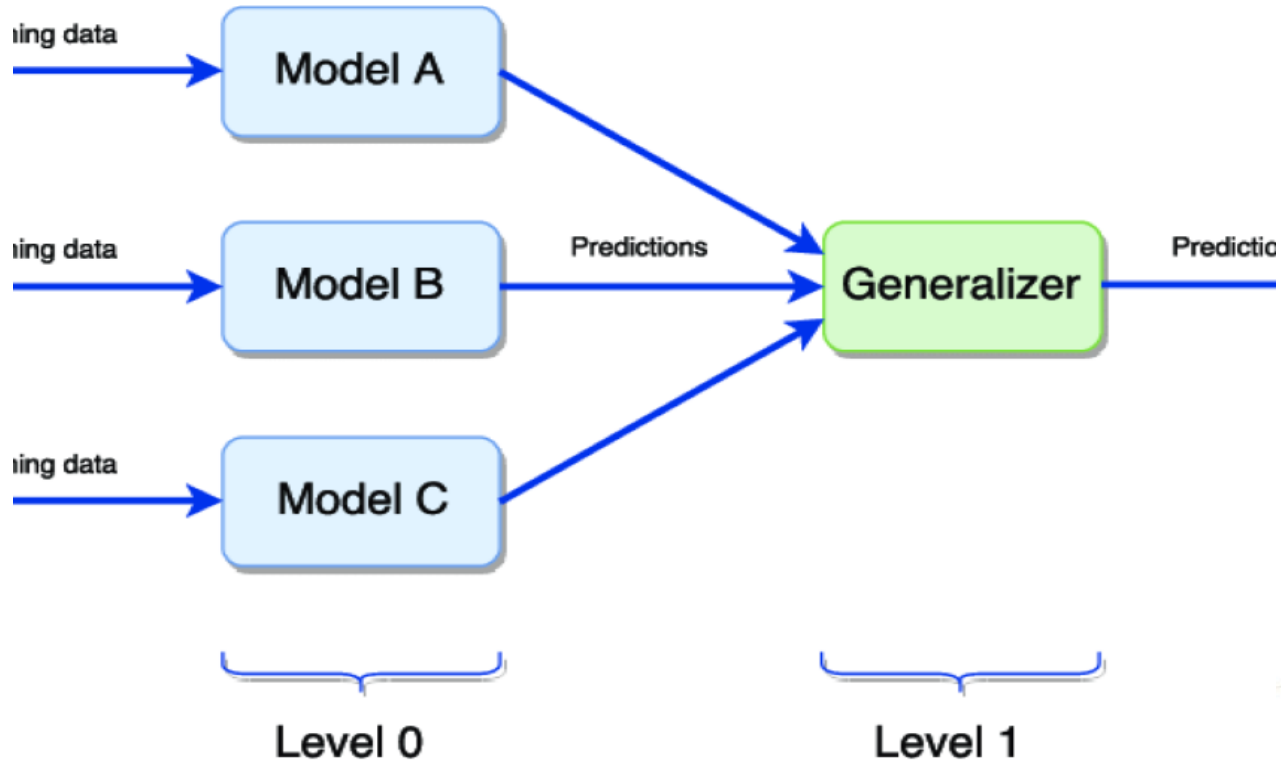


# Итоги

- ❑ В методах бустинга несколько экземпляров одной и той же базовой модели обучаются последовательно, так что на каждой итерации способ обучения текущего слабого ученика зависит от предыдущих слабых учеников и, в особенности, от того, как они работают с данными
- ❑ Эта итеративная стратегия обучения, используемая в методах бустинга, которая адаптируется к недостаткам предыдущих моделей для обучения текущей, в основном позволяет нам получить модель ансамбля с более низким смещением, чем ее компоненты: именно поэтому слабые ученики с низким разбросом, но высоким смещением хорошо приспособлены для бустинга
- ❑ в методах стекинга разные слабые ученики подбираются независимо друг от друга, и в дополнение к этому обучается метамодел для прогнозирования результатов на основе результатов, получаемых базовыми моделями



# Пример: Повысьте оценку своей прогнозной модели с помощью стекового регрессора



Ансамблевое обучение - это метод, широко используемый практиками машинного обучения, который объединяет навыки разных моделей для прогнозирования на основе заданных данных. Мы используем это, чтобы объединить лучшие из нескольких алгоритмов, которые могут давать более стабильные прогнозы с очень небольшой дисперсией, чем то, что мы получаем с одним регрессором.

**Уровень 0** - обучение разных моделей на одном наборе данных с последующим построением прогнозов.

**Уровень 1.** Обобщите прогнозы, сделанные разными моделями, чтобы получить окончательный результат. Наиболее распространенный способ обобщения - это взятие среднего значения всех прогнозов модели уровня 0 для получения окончательного результата.