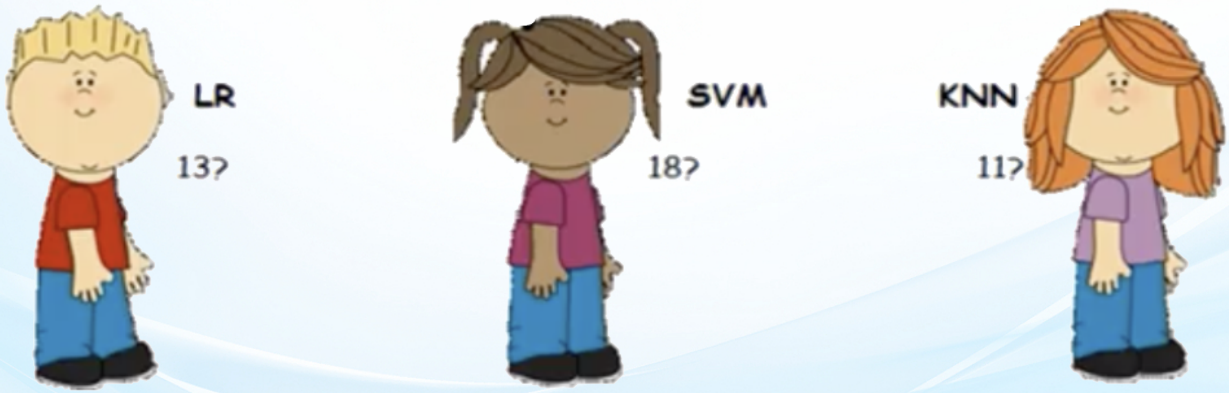
Машинное обучение

DS-поток

Лекция 8

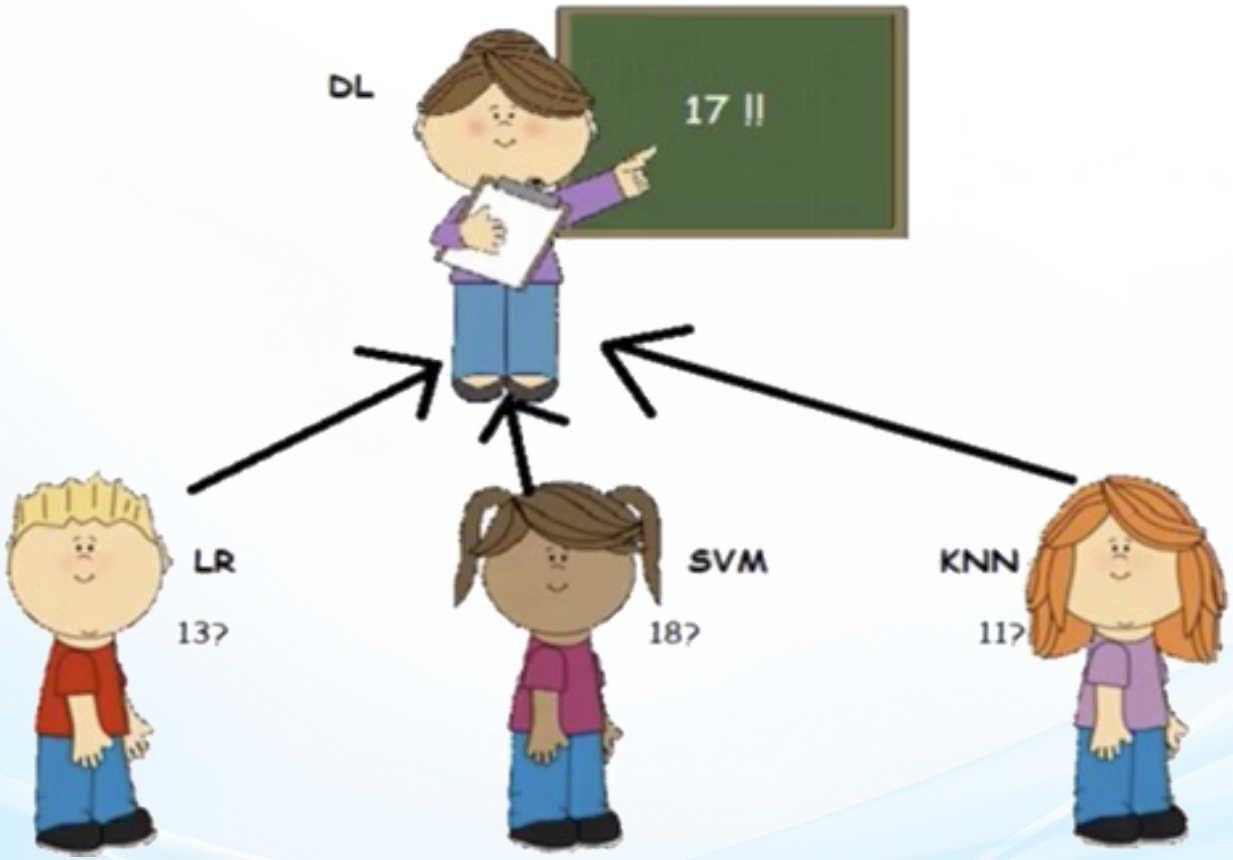
Стекинг и блендинг

Основная идея



Обучим независимо T базовых моделей $b_1(x), \ldots, b_T(x)$.

Основная идея



Будем использовать предсказания моделей как новые признаки.

Обучим на них мета-модель $\widehat{y}(x)$.

Обучение

Как обучать?

Самый простой вариант — обучить мета-модель на этой же выборке:

$$\sum_{i=1}^{n} \mathcal{L}\Big(y_i,\ \widehat{y}(b_1(x_i), \ldots, b_T(x_i))\Big) \to \min_{\widehat{y}}.$$

В чем тут проблема?

В предсказаниях базовых моделей уже неявным образом зашита информация об откликах, ведь они использовались при их обучении.

Тогда $\widehat{y}(x)$ будет отдавать предпочтение переобученнным $b_j(x)$, т.к. по их прогнозам лучше восстанавливаются истинные ответы.

Но такая мета-модель будет иметь очень низкое качество на тесте.

Обучение

Будем обучать базовые модели и мета-модель на разных выборках.

Вариант 1

- Разделим обучающие данные на 2 части.
- На первой части обучаем все базовые модели.
- Делаем предсказания для второй части.
- Используя ответы на второй части как признаки, обучаем мета-модель.

Плюсы:

Просто для понимания и реализации.

Минусы:

Из-за разделения на 2 части меньше данных для обучения.

Обучение

Вариант 2

- Разделим обучающую выборку на K блоков $X_1, \ldots, X_K$.
- Для всех $k \in [1, K]$ и для всех $t \in [1, T]$:

  Обучим модель b_t на всех блоках, кроме k-го.

  Обозначим эту модель b_t^{-k}
- Обучаем мета-модель по следующему функционалу:

$$\sum_{k=1}^{K} \sum_{(x_i, Y_i) \in X_k} \mathcal{L}\left(Y_i,\ \widehat{y}(b_1^{-k}(x_i), \ldots, b_T^{-k}(x_i))\right) \longrightarrow \min_{\widehat{y}}$$

Смысл:

При обучении мета-модели на x_i используются базовые модели, которые не видели этот объект при обучении.

$\Rightarrow$ Мета-модель не переобучается на прогнозах базовых моделей.

Замечания

- Разнообразие базовых моделей очень важно.
- Если данные зависят от времени:

  При обучении мета-модели на x_i должны использоваться базовые модели, обучавшиеся на более ранних объектах.
- Мета-модель не обязательно должна быть сложной.
- Базовые модели должны быть достаточно сложными.

  Т.к. они должны сделать глубокий анализ признаков.

Стекинг: категориальные признаки

Как работать с категориальными признаками?

Самый простой вариант — one-hot encoding.

Получим большую размерность признакового пространства.

- Случайный лес будет обучаться долго.
- Градиентный бустинг покажет плохие результаты
  Базовые деревья небольшой глубины, например 4,
  позволяют учитывать лишь зависимость целевой переменной
  от наборов из 4-х признаков.

Решение — стекинг.

Базовые модели обрабатывают большое признаковое пространство.

Градиентый бустинг (= мета-модель) обрабатывает небольшое число признаков от базовых моделей.

Блендинг

Блендинг — частный случай стекинга.

Мета-модель является линейной:

$$\widehat{y}(x) = \sum_{t=1}^{T} w_t b_t(x)$$

Есть несколько вариантов весов:

- Веса находятся с помощью некоторой линейной модели.
- Веса берутся равными: $w_1 = \cdots = w_T = 1/T$.
- Веса подбираются по кросс-валидации.
- Веса берутся пропорционально качеству модели.

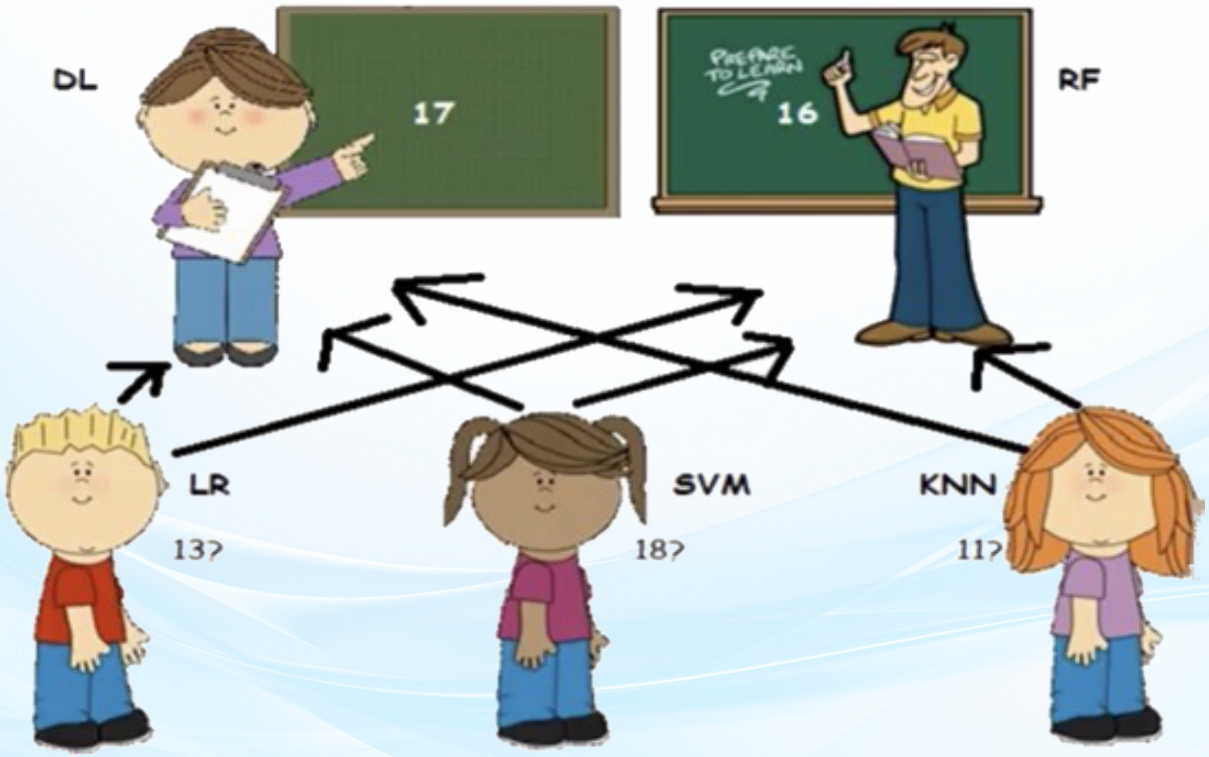
  Пусть b_1 имеет качество 0.967 AUC на валидации.

  b_2 имеет качество 0.943 AUC на валидации.

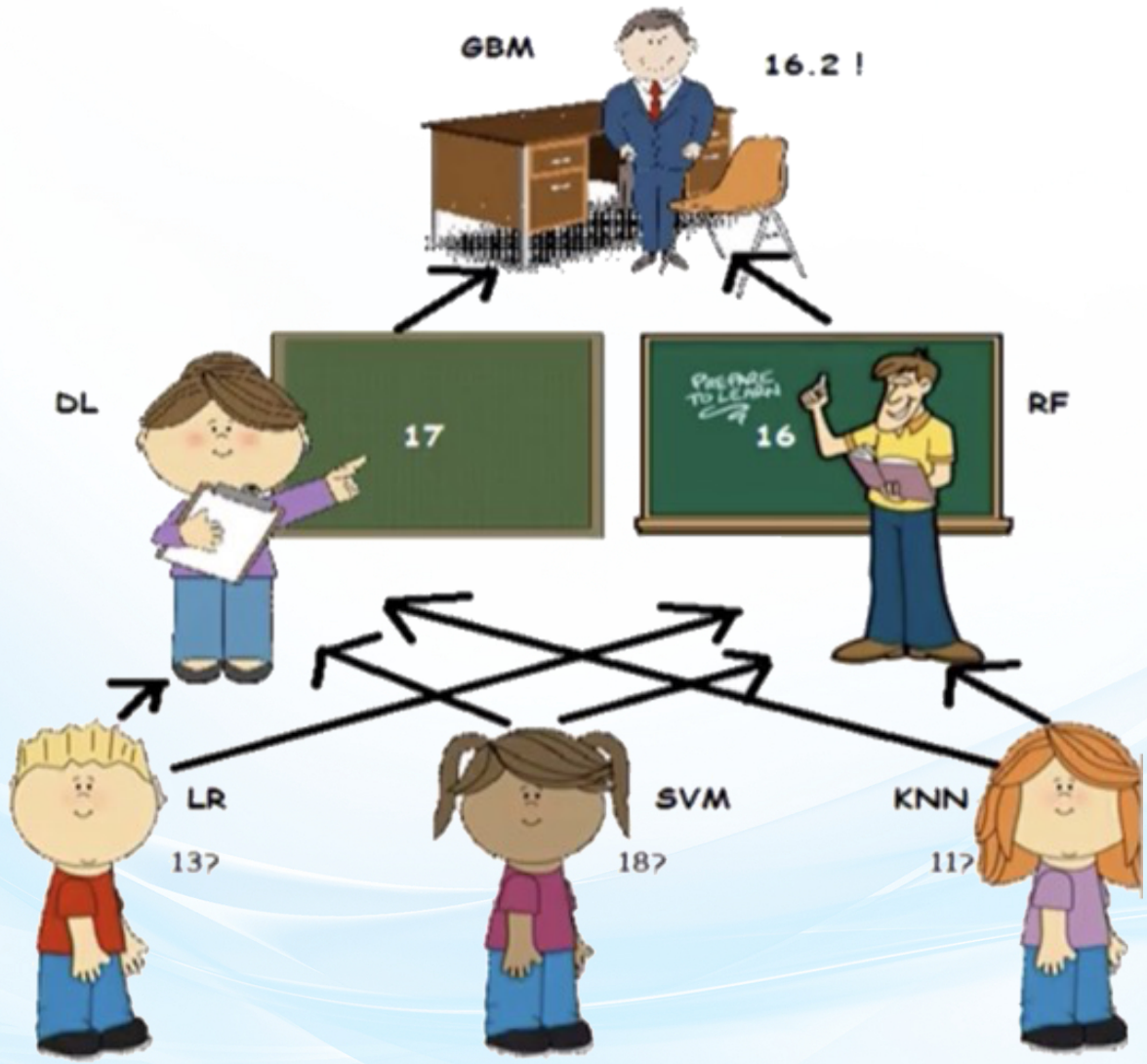
  Возьмем: $\widehat{y}(x) = (0.967 \cdot b_1(x) + 0.943 \cdot b_2(x)) \;/\; (0.967 + 0.943)$

Иногда даже блендинг с равными весами позволяет улучшить качество по сравнению с отдельными базовыми моделями.

StackNet: Основная идея



StackNet: Основная идея



StackNet — сеть, в которой каждая вершина является моделью.

StackNet: Обучение

Вариант 1

- Сделаем разбиение данных на train и valid.
- На train учим модели первого уровня,
  на valid они делают предсказания.
- Далее сделаем разбиение valid на mini_train, mini_valid.
- На mini_train учим модели второго уровня,
  на mini_valid они предсказывают.
- И так далее ...

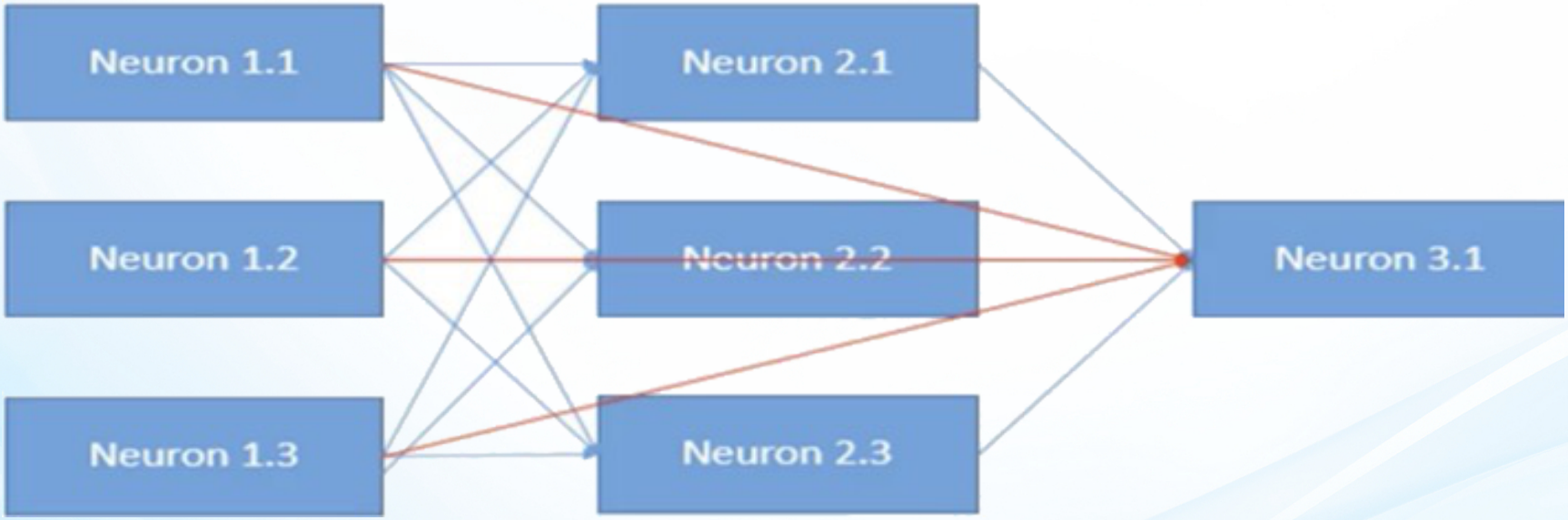
Минусы:

Очень мало данных для обучения каждой модели.

Вариант 2

Использовать K-fold-подобные схемы.

StackNet: Основная идея



Можно использовать не только выходы моделей предыдущего слоя, но выходы более ранних слоев.

StackNet: разнообразие

Разнообразие, основанное на моделях:

- 2-3 разных градиентных бустинга

  Lightgbm, xgboost, catboost или разная глубина

- 2-3 нейросети.

  Например, глубокая, средняя и маленькая

- 1-2 линейных модели.
- 1-2 Knn модели.
- 1-2 RandomForest.

StackNet: разнообразие

Разнообразие, основанное на данных:

- Препроцессинг категориальных признаков

  OneHot, Label, Mean encodings

- Препроцессинг вещественных признаков

  убирать ли выбросы и как, масштабирование, трансформации.

- Добавление разных признаков

  $x_1 \times x_2$,

  groupby,

  ...

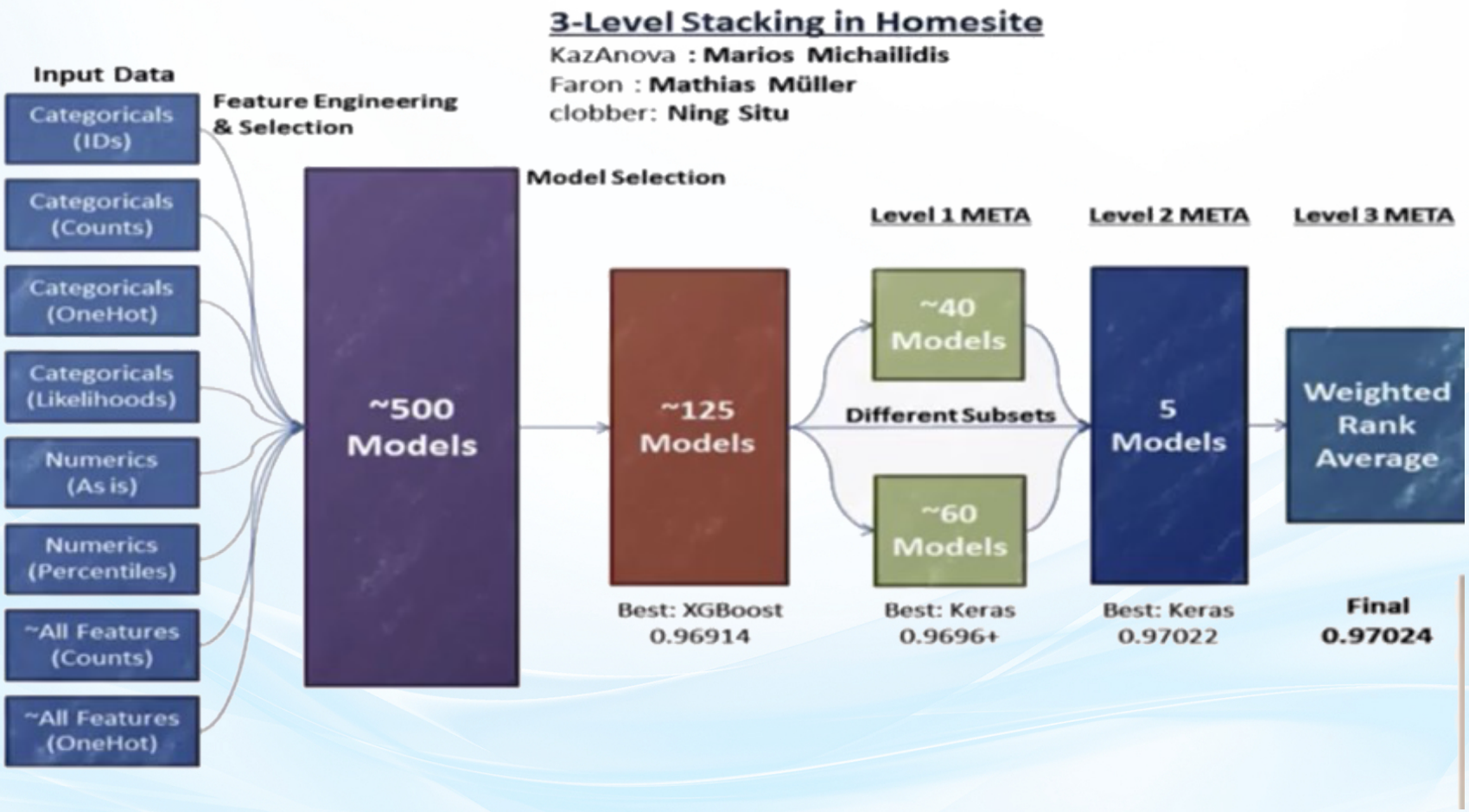
Рекомендации

- На каждые 7.5 модели в пред. слое 1 модель в следующем.
- Аккуратно с target leakage.

  Пусть модель b слоя m обучается на предсказаниях слоя $m-1$ для объектов X_b.
  Важно, чтобы слой $m-1$ не обучался на X_b.
- Можно использовать модель классификации для задачи регрессии и наоборот.

  Например, при предсказании возраста одна из моделей может предсказать переменную $\mathsf{I}(\text{age} > 50)$

Пример: StackNet




ВСЁ!