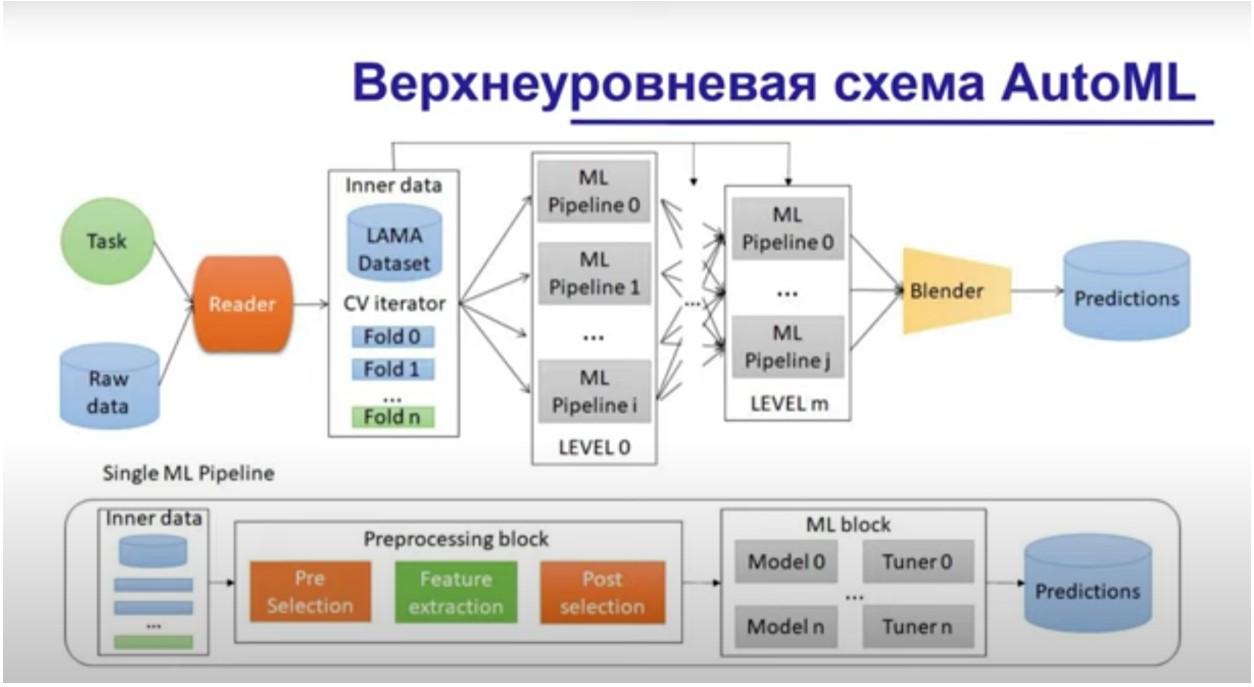
LightAutoML model для задачи регрессии

Данная модель в качестве пайплайна для задачи регрессии использует ансамбль линейной модели, катбуста, light GBM, а так же fine-tuned версии данных бустингов. Все эти модели работают вместе с помощью технологии блендинга.



Если отвечать на вопрос почему именно эта модель была выбрана для маленьких данных, то стоит понять, что одна из главных проблем маленьких данных - переобучение. Т.к. данных не хватает многие модели очень легко переобучаются и не могут показать удовлетворяющей результат на тестовой выборке.

Рассмотрим способы борьбы с переобучением:

1) Регуляризация и оптимизация гиперпараметров. С помощью подбора параметров можно добиться явно снижения переобучения. Особенно это актуально для бустингов, где можно искусственно контролировать “силу” модели, регулируя максимальную глубину дерева решений, а также количество подобных деревьев. В LightAutoML также используются Экспертные системы для подбора параметров, например, как их как: learning_rate, num_leaves, colsample_by_tree. Подобные параметры могут быть далеки от идеальных, но в совокупности с другими методами, дают неплохой результат, а главное данные параметры были проверены экспертами, которые понимали, как работать с переобучением. Если позволяет время обучения, то используется более тонкая настройка экспертных параметров с помощью TPE (Optuna). На каждой итерации:

-приближается зависимость целевой метрики от гиперпараметров суррогатной функцией

-выбирается новый набор гиперпараметров, оптимизирующий суррогатную функцию

- обучается для него модель, запоминается значение целевой метрики и происходит переход к новой итерации

Для настройки линейной модели используются методы Жадного поиска (перебор параметров по сетке).

2) Cross-validation.

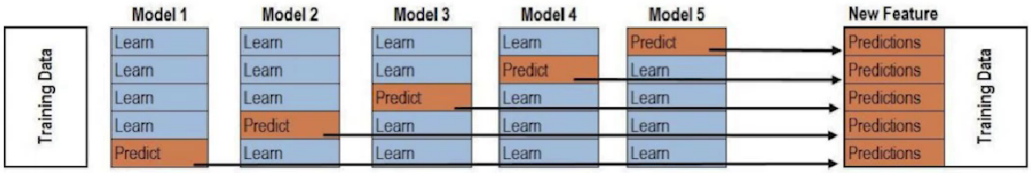
Cross-validation (CV) - это процесс получения множества подмножеств из обучающих данных и обучения модели на каждом подмножестве. Идея заключается в том, что модели может "повезти" и она будет иметь большую точность с одним подмножеством, но при использовании многих подмножеств модель не будет достигать такой высокой точности каждый раз. При составлении CV вы предоставляете набор данных, не прошедших проверку, указываете свои CV folds (количество подмножеств), и автоматизированный ML обучает вашу модель и настраивает гиперпараметры, чтобы свести к минимуму ошибки в вашем наборе проверки. Один CV fold может быть переобучен, но использование многих из них снижает вероятность того, что ваша конечная модель будет переобучена. К сожалению данный процесс приводит к увлечению времени работы алгоритма, но при маленьких данных это не страшно. В LightAutoML это реализовано подобным образом:

Внутренняя валидация

Out-of-fold (OOF) предсказания

- Не переобученные предсказания всей обучающей выборки для моделей следующего уровня
- Предсказание для новых данных (тест) – усреднение по всем K-Fold моделям

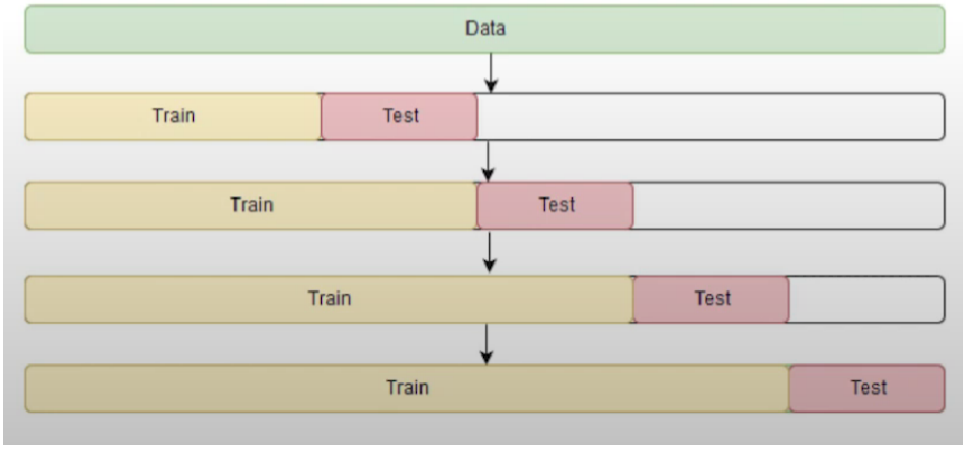
Тюнинг гиперпараметров



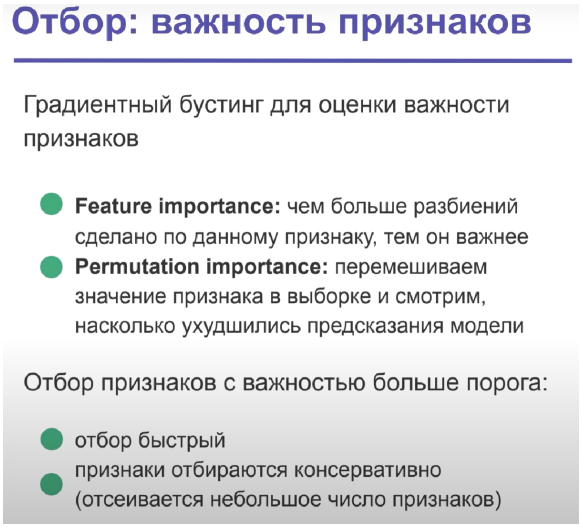
Для обучающей выборки генерируются Out-of-fold предсказания, для того чтобы генерировать не переобученные предсказания для последующего их использования в стекинге. Т.е. для каждого fold-а из обучающей выборки мы обучаемся на всех кроме одного, на котором и делаем предсказания (чтобы оно не было переобученно), после чего собираются предсказания всех K-fold моделей и усредняются.

Также в модели реализованы другие способы кросс-валидации, например holdout (отложенная выборка) - применяется, когда возможно изменение распределения данных.

Возможна и кастомная кросс-валидация, например Time series split:



Также LAMA умеет очень хорошо работать с отбором фичей (features), а также их автоматической типизацией, и модель сама может понять, что лучше, чтобы фича была категориальной или численной. Что на мой взгляд важно при работе с маленькими данными, т.к. выделение нужных фичей помогает хорошо приближать прогнозируемое значение к реальному, при этом сильно не переобучаясь.



Если оценивать результаты работы LAMA, то есть данные, на которых модель справилась отлично, но также имеются бандлы, на которых метрика не

соответствует оптимальной.

Использовались две типа моделей TabularAutoML, TabularUtilizedAutoML. Второй вид моделей в теории должен лучше показывать себя для маленьких данным, т.к. важна очень начальная настройка параметров (random_seed для CV и т.д.) и этим данная модель и занимается: обучает много разных Tabular моделей с различными параметрами и выбирает из них лучшую, но заметного прироста эта не принесло. Стандартная TabularAutoML на многих профессиях показала себя лучше.