Решение задачи предсказания уровня воды в реке Амур

AI Jorney 2020

Мамаев Александр

Картинок не много, но они есть

Преобразование таргета

- Обычно в таких задачах модели напрямую используются для предсказания значений уровней на следующие 10 дней
- Мы строим либо одну модель на все 10 дней, или же 10 моделей на каждый день
- Я решил поступить несколько иначе, и упростить задачу

Преобразование таргета

- Поскольку уровень воды в реке в течение 10 дней не имеет большой дисперсии, то есть вряд ли может резко упасть в середине предсказательного периода, а потом вырасти (а если и может, то это маловероятно)

- Следовательно, для предсказываемых 10 дней можно выделить линейный тренд, в какую сторону и как сильно будет расти уровень воды

Преобразование таргета

- Тренд определяется простой прямой относительно дня наблюдения

$$K*X + B$$

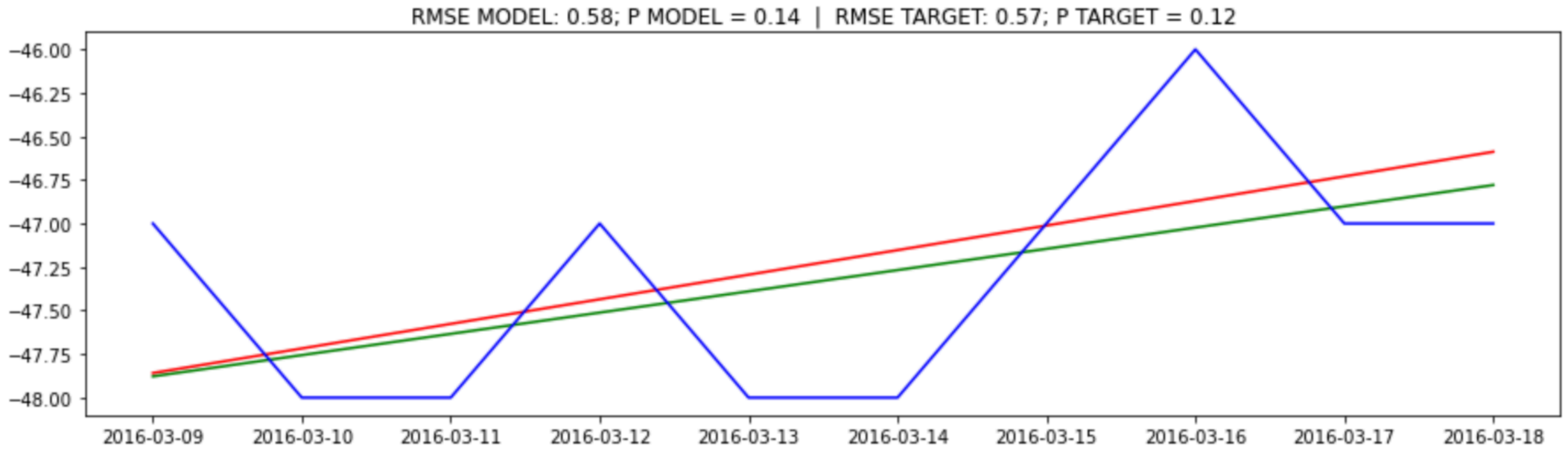
- За X можно принять номер дня относительно предсказания
- B - последнее наблюдаемое нами значение
- В таком случае именно параметр K (степень наклона прямой) и должна определить модель

Преобразование таргета

- Для получения таргетов я брал для каждой записи в таблице значения следующих 10 дней
- Вычитал значение первого дня
- Обучал линейную модель без параметра смещения (bias)
- Далее, просто брал коэффициент наклона
- Данный коэффициент записывался в качестве таргета

Преобразование таргета

Пример выделенного тренда



В данном примере:
зеленый - выделенный тренд на обучающей выборке (таргет)
красный - предсказанный моделью

Преобразование таргета

Плюсы подхода

- На практике нас не так сильно интересует точное значение уровней
- Более важен тренд в котором будет двигаться уровень воды
- Значение коэффициента тренда достаточно легко можно оценить и понять на сколько сильно и как быстро уровень будет подниматься
- Оценку а так же вероятность затопления можно будет совершать с помощью базового статистического анализа

Используемые признаки

В качестве входных данных использовалась следующая информация:

- Прошлые уровни воды на станции
- Погодные данные с метео-станций
- Прогноз погоды на следующие несколько дней
- Информация об уровнях воды на ближайших станциях выше по течению
- Месяц, день
- Идентификатор станции

Используемые признаки

Погодные данные

Поскольку вокруг станции располагается несколько метеостанций и не всегда они есть в непосредственной близости со станцией измерения, я сделал агрегацию информации с четырех ближайших метео-станций

Все признаки суммировались с некоторым коэффициентом, который брался относительно функции softmax по расстояниям до этих станций.

При этом, если в одном из полей станции стоял NaN, то ее расстояние не попадало в расчет коэффициентов.

Используемые признаки

Погодные данные

Softmax был выбран, т.к. он дает гораздо больший вес тем станциям, которые на много ближе. Таким образом, если есть станция в непосредственной близости, то она получит коэффициент близкий к 0.99, а если у нас есть несколько станций на расстоянии 20км, то они получат примерно равные значения.

Используемые признаки

Погодные данные

Данные об осадках так же агрегировались, так например статус о погоде кодировался в onehot вектор с полями rain, snow, drizzle, fog, и shower.

Я просто сделал мэпинг этих кодов в соответсвующие колонки.

Далее относительно того, сколько наблюдений дождя было совершено, ставилось значение от 0 до 1, по сути это просто доля дождя (снега) за сутки

Используемые признаки

Погодные данные

Далее происходил подсчет статистик погоды для каждого момента времени, средняя температура, влажность за N дней, сумма осадков, количество дождливых дней и так далее.

Этот процесс происходил автоматически, поэтому тут много чего не опишешь.

```
X = pd.concat((trend_features_1, trend_features_2, trend_features_3,
               shift_features_1, shift_features_2, shift_features_3, shift_features_4,
               agg_features_1, agg_features_2,
               weather), axis=1)
```

Используемые признаки

Прогноз погоды

В качестве признака так же использовалась информация о количестве дождливых/снежных дней в будущем, а так же о средней температуры.

На тренировочном датасете я просто брал информацию из “будущего” и составлял по ним статистики.

Для составления прогноза используется сервис weatherapi.com (пришлось купить подписку)

Уровни воды по ближайшим станциям

Выделение станций

Ближайшие станции я просто определил по расстоянию на координатах.

Далее отсеял те, что находятся по долготе правее, так я выделил только те станции, которые находятся выше по течению

Уровни воды по ближайшим станциям

Выделение станций

Далее выделялись станции, которые были открыты достаточно давно, чтобы по ним было достаточно информации.

После чего брались три ближайшие станции из них выделялись тренды, и смещение трендов, аналогично как и с уровнем воды на рассчитываемой станции.

Так же в качестве фичи бралось расстояние до каждой из станций, чтобы модель могла как-то учитывать это в прогнозе.

(Чем меньше расстояние, тем меньший лаг надо брать, но эту гипотезу я проверить не успел)

Категориальные фичи

В качестве дополнительных фичей я добавил месяц и идентификатор станции в качестве категориальных фичей.

Т.к. Хоть у всех станций должны быть похожие физические процессы с точки зрения трендов движений уровней воды, но у каждой станции в свой месяц могут быть свои смещения.

Ну и день в месяце закинул, а почему бы и нет.

Финальный дотаяет

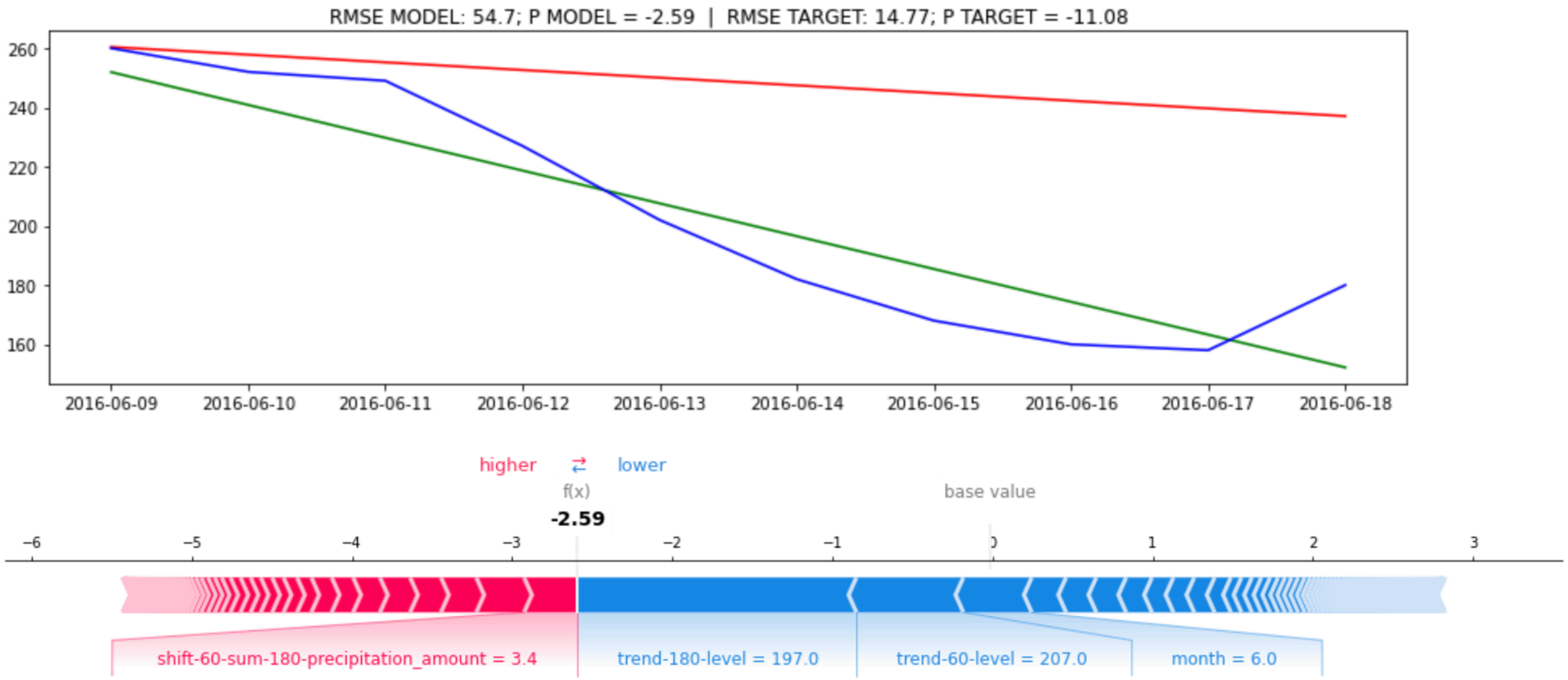
Обучение происходило на станциях которые были выбраны в качестве таргета + произвольные 30 станций (ну сколько успел посчитать, столько и успел)

Идея была в том, что добавлением других станций мы дадим модели больше понимания об устройстве физического процесса движения уровней воды.

Модель

- Для предсказания используется модель **CatBoost**
- Параметры подбирались вручную на основе метрики **RMSE**
- Так же происходила валидация фичей на основе библиотеки shap
- Считались уровни отклонений модели за несколько дней
- (картинки далее)

Валидация

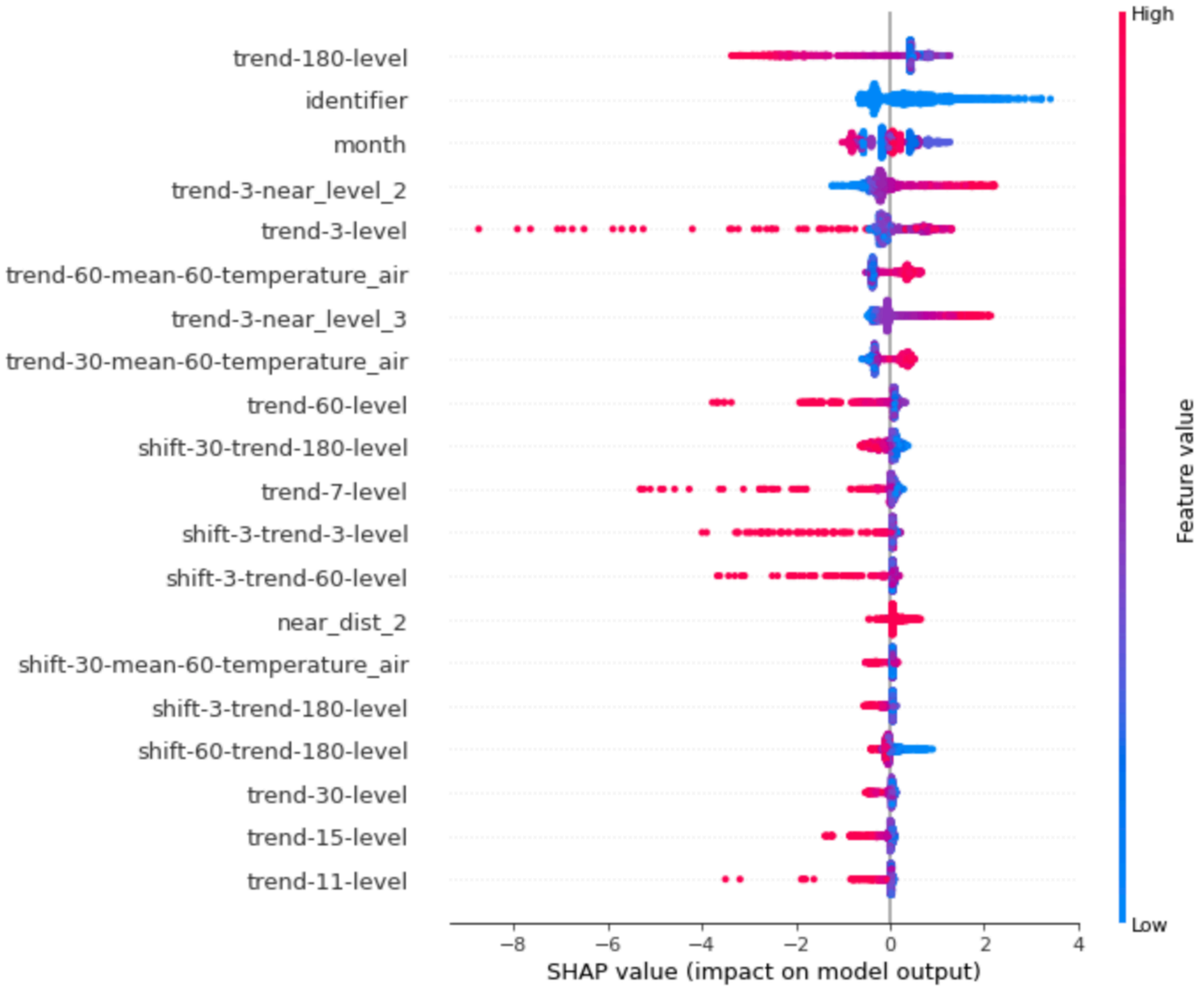

RMSE MODEL: 54.7; P MODEL = -2.59 | RMSE TARGET: 14.77; P TARGET = -11.08
260
240
220
200
180
160
2016-06-09
2016-06-10
2016-06-11
2016-06-12
2016-06-13
2016-06-14
2016-06-15
2016-06-16
2016-06-17
2016-06-18
higher
lower
f(x)
-2.59
base value
−6
−5
−4
−3
−2
−1
0
1
2
3
shift-60-sum-180-precipitation_amount = 3.4
trend-180-level = 197.0
trend-60-level = 207.0
month = 6.0

Валидация

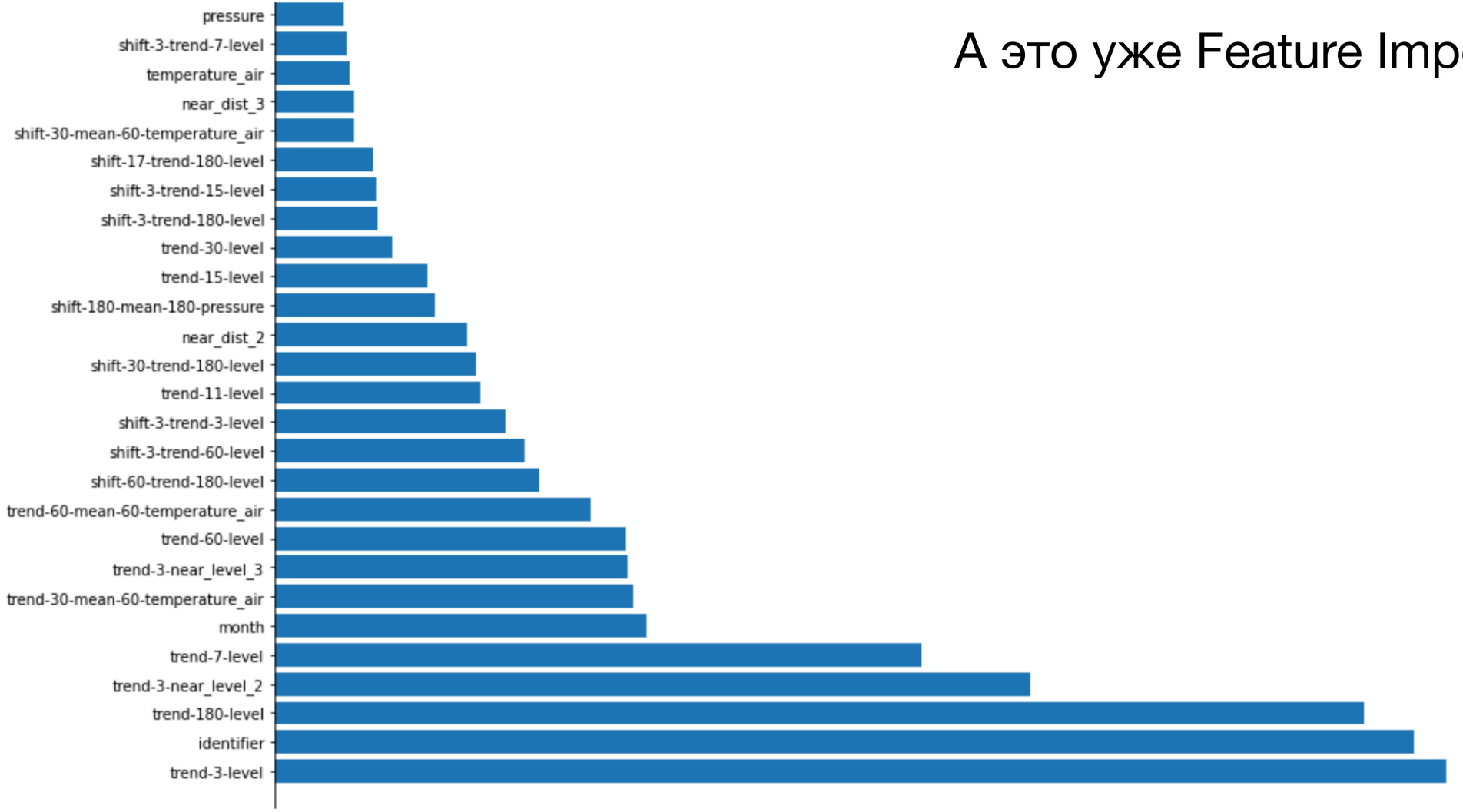
```
------------------------------------------------

--- MODEL VALIDATION REPORT FOR STANTION 6005 ---

Model RMSE: 59.38068091544269
Target RMSE: 28.63146552879636

Deviations rate:
>10 cm -> 53.0%
>15 cm -> 43.0%
>50 cm -> 18.0%
>60 cm -> 14.0%
>70 cm -> 12.0%
>80 cm -> 9.0%
>90 cm -> 8.0%
>100 cm -> 7.0%
>150 cm -> 4.0%
>170 cm -> 3.0%
>200 cm -> 2.0%
>300 cm -> 1.0%
>500 cm -> 0.0%
>1000 cm -> 0.0%
```

- Target RMSE - значение RMSE, если бы модель предсказала тренд идеально
- Ну и уровни отклонений воды в течение нескольких лет с предсказанными трендами

Валидация


trend-180-level
identifier
month
trend-3-near_level_2
trend-3-level
trend-60-mean-60-temperature_air
trend-3-near_level_3
trend-30-mean-60-temperature_air
trend-60-level
shift-30-trend-180-level
trend-7-level
shift-3-trend-3-level
shift-3-trend-60-level
near_dist_2
shift-30-mean-60-temperature_air
shift-3-trend-180-level
shift-60-trend-180-level
trend-30-level
trend-15-level
trend-11-level
−8
−6
−4
−2
0
2
4
SHAP value (impact on model output)
High
Feature value
Low

Валидация



А это уже Feature Importance модели

Финалочка

Или что еще можно было сделать

- Добавить предсказания модели ARIMA в качестве фичей
- Лучше пофильтровать фичи и лучше определить значения оптимальных лагов
- Попробовать подобрать более оптимальные параметры бустинга
- Как-то лучше использовать информацию о снеге и его таянии (данные о снеге были неоч)

Спасибо

Вы классные :)