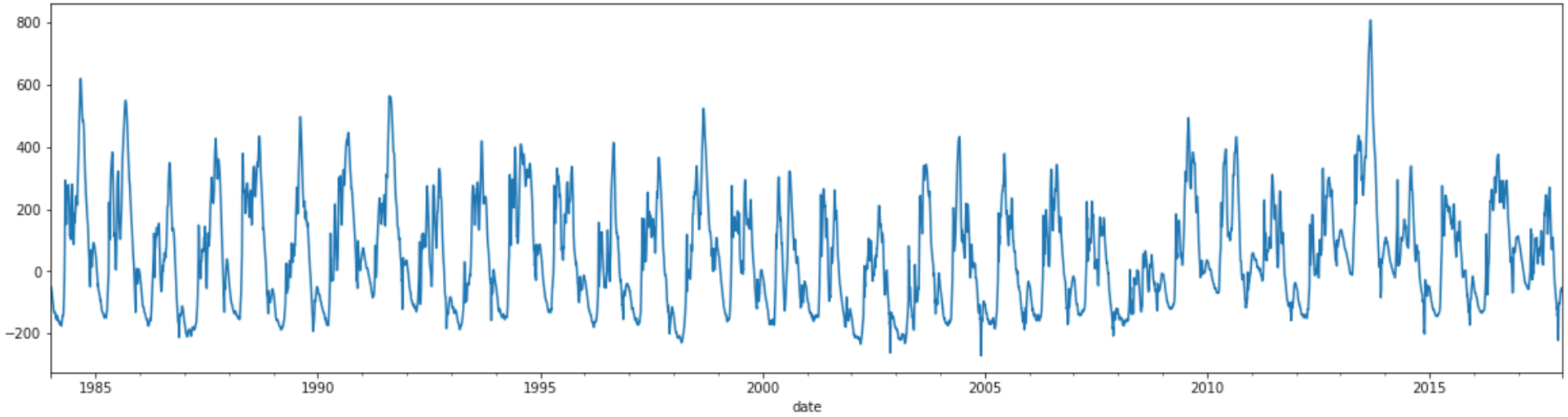
LGBM для прогнозирования паводков на р. Амур

Данные об уровнях

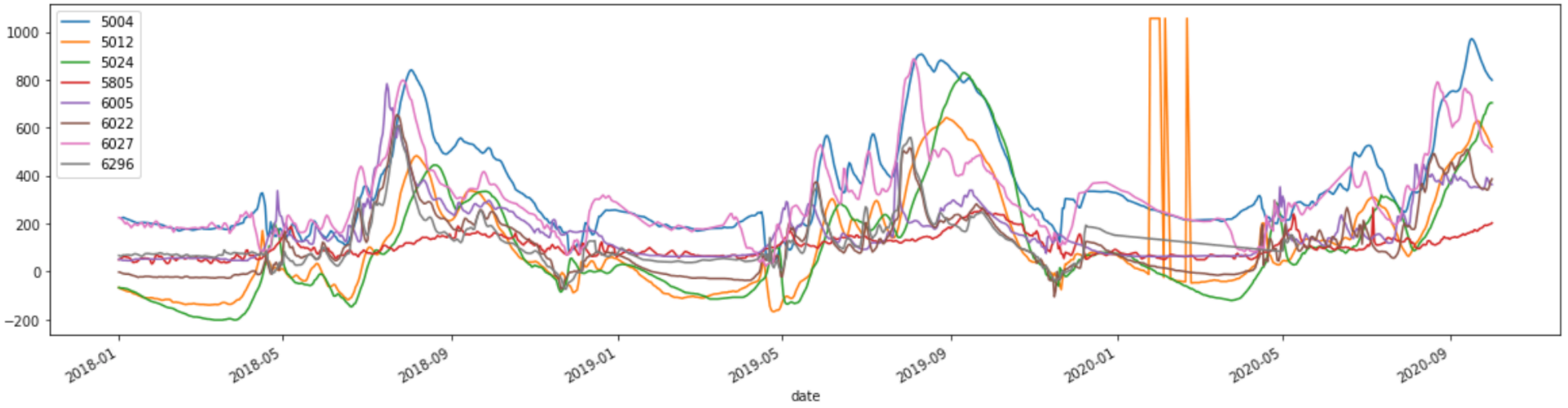
- Ряд уровней воды для гидрологического поста номер 5012 за период 1984.01.01 – 2017.12.31:



- По графику видна годовая сезонность

Выбросы

- На графиках за период 2018.01.01 – 2020.10.01 встречаются выбросы

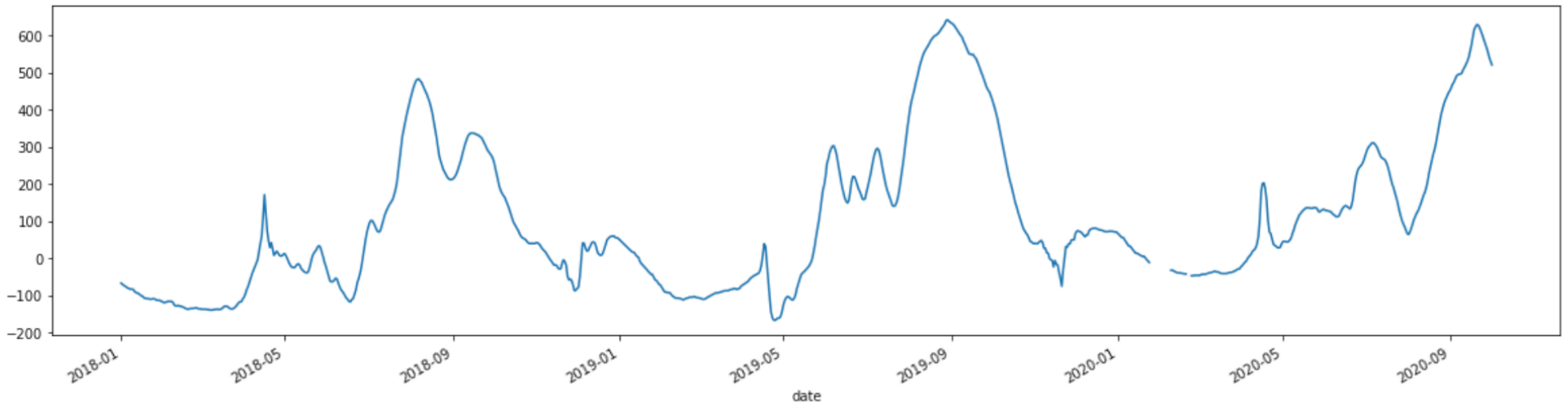


Удаление выбросов

In [12]:
```
nf.loc[(nf['stage_max'] > nf['stage_max'].\
          quantile(0.99))&(nf['station_id'] == 5012), 'stage_max'] = np.nan
```

In [13]:
```
sf = nf[nf['station_id'] == 5012]
sf['stage_max'].plot(figsize=(20,5))
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22e9af2710>



Метеоданные

- Очевидно, что данные о погоде должны сильно влиять на целевую переменную. Так как предсказание будет делаться на *k* дней вперед, я буду использовать данные о погоде *k* дней назад.
- Буду использовать данные о
  - температуре воздуха
  - температуре почвы
  - влажности
  - количестве осадков

Метеоданные

- Данные о температуре делю по времени суток и усредняю по дате. Получаю таблицу:

station_id	date	day_temperature_air	night_temperature_air	day_temperature_ground	night_temperature_ground
5004	1985-01-01	-17.900	-24.400	-19.950	-27.275
	1985-01-02	-16.775	-26.250	-19.725	-29.375
	1985-01-03	-18.975	-24.650	-19.900	-26.900
	1985-01-04	-19.150	-26.050	-19.900	-28.175
	1985-01-05	-21.200	-27.675	-21.775	-28.175

- Влажность и количество осадков просто усредняю по дате.

Агрегирование метеоданных

- Далее реализована функция агрегирующая метеоданные, где количества дней и агрегирующие функции задаются как параметры, значения которых подбираются экспериментально.

In [23]:

```
def agg_meteo(df,
              columns = ['day_temperature_air', 'night_temperature_air', 'day_temperature_ground',
                         'night_temperature_ground', 'humidity','precipitation_amount'],
              agg_days = [15, 15, 15, 15, 10, 60],
              agg_funcs = ['mean', 'mean', 'mean', 'mean', 'sum', 'sum'],
              shift = [10, 10, 10, 10, 10, 10]):
    res = pd.DataFrame(index=df.index)
    names = []
    for c, d, f, s in zip(columns, agg_days, agg_funcs, shift):
        name = '{}_{}_{}'.format(c, d, s)
        res[name] = df[c].rolling(d, min_periods=1).agg(f).shift(s)
        names.append('{}_{}_{}'.format(c, d, s))
    return res.reset_index(), names
```

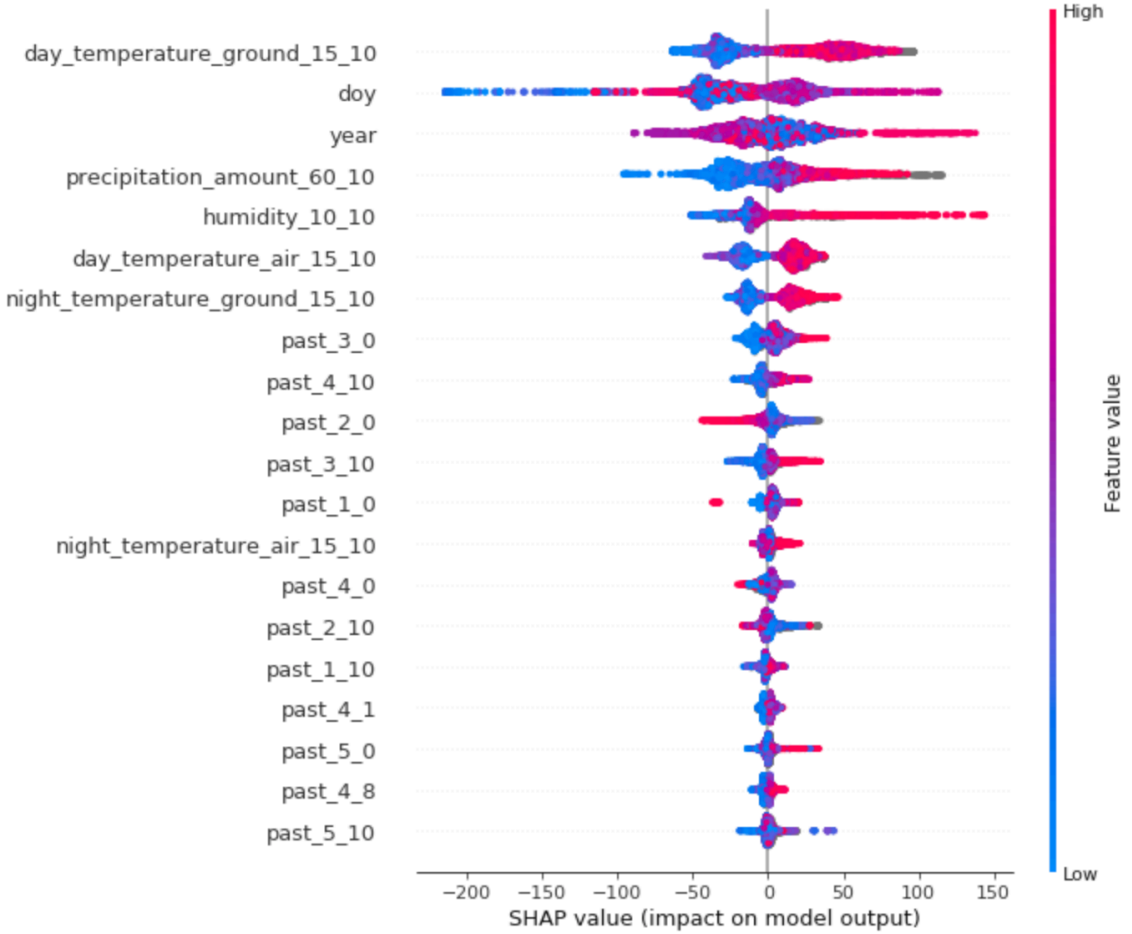
Другие признаки

- Как было сказано ранее, есть некая периодичность в значениях уровня, поэтому использую календарные признаки: (день в году, месяц, год) и признаки – значения целевой переменной в аналогичный период за прошедшие годы.
- И, конечно, значения уровня воды в предыдущие *n* дней

Важность признаков

- Обучив модель без признаков временного ряда (значений за предыдущие *n* дней) получил следующий график важности признаков

Из графика видно, что метеопризнаки довольно сильно влияют на целевое значение. Например,
как и ожидалось, признак `precipitation_amount_60_10` - накопленные осадки за 60 дней, 10 дней назад при больших значениях дает положительный вклад в целевое значение, а при малых - отрицательный. Тот же эффект виден для `humidity_10_10` - накопленная влажность воздуха за 10 дней, 10 дней назад.



Обучение

- Для каждого гидрологического поста я решил обучать отдельную модель
- В качестве тестового периода я беру: 2019-11-01 – 2019-11-10
- На фолды для кросс-валидации разбиваю следующим образом:
  - test: *year*-11-01 – *year*-11-10
  - train: все данные до test

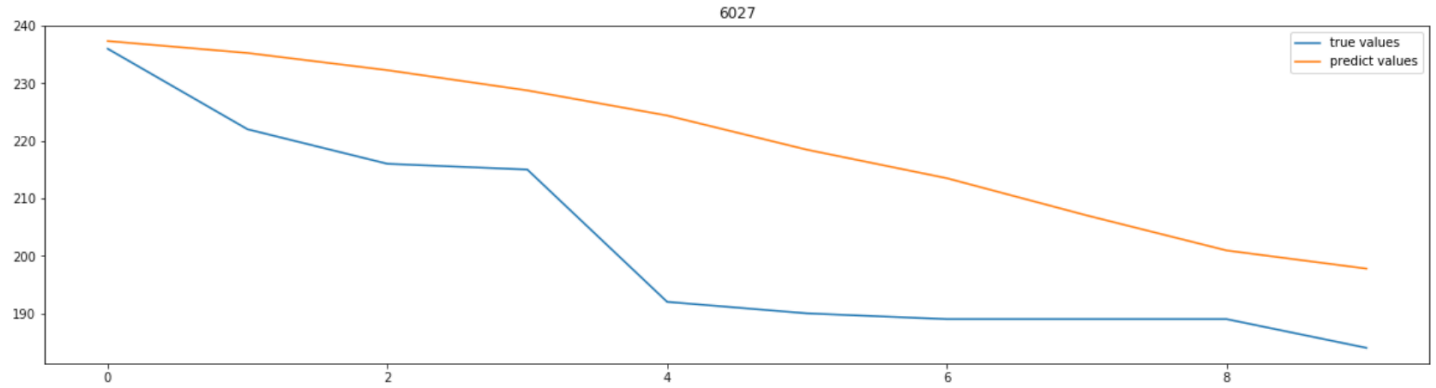
  Где *year* из [2014, 2015, 2016, 2017, 2018]
- Далее использую gridsearch для подбора трех параметров LGBM:
  - ‘n_estimators’
  - ‘learning_rate’
  - ‘num_leaves’
- При подборе гиперпараметров я минимизирую усредненную по фолдам ***MAE***

Результаты

Значения метрик вышли не очень хорошими, но, во-первых, в модели не было метеоданных за 2018-2020 гг., что сильно ухудшило метрики, во-вторых, по графикам видно, что модель улавливает основную тенденцию.

Для улучшения метрик можно попробовать подключить свежие метеоданные, а также, возможно, использовать метеоданные из других источников.

```
6027  MAE:  17.37000850318993
```




```
5004  MAE:  23.245771608120506
```