Datetime features to capture seasonality

Seasonality features

Datetime features

- Seasonality is often driven by the calendar date and time.

- Examples:
  - Traffic patterns have **daily seasonality** related to the **hour of the day**.
  - Retail sales have **weekly seasonality** related to the **day of the week**.
  - Air passengers have a **yearly seasonality** related to the **month or week of the year**.

- Extracting features from the date and time can therefore help capture seasonality.

Datetime features

Time index	Electricity demand	Hour of day	Day of week	Month of year
2020-02-12 01:00:00	23	1	2	2
2020-02-12 02:00:00	30	2	2	2
2020-02-12 03:00:00	35	3	2	2
2020-02-12 04:00:00	30	4	2	2

Daily seasonality

Weekly seasonality

Yearly seasonality

Implementation in sktime

DateTimeFeatures

```
class DateTimeFeatures(ts_freq=None, feature_scope='minimal',
manual_selection=None, keep_original_columns=False)
```

[source]

DateTime feature extraction for use in e.g. tree based models.

DateTimeFeatures uses a date index column and generates date features identifying e.g. year, week of the year, day of the week.

Implementation in sktime

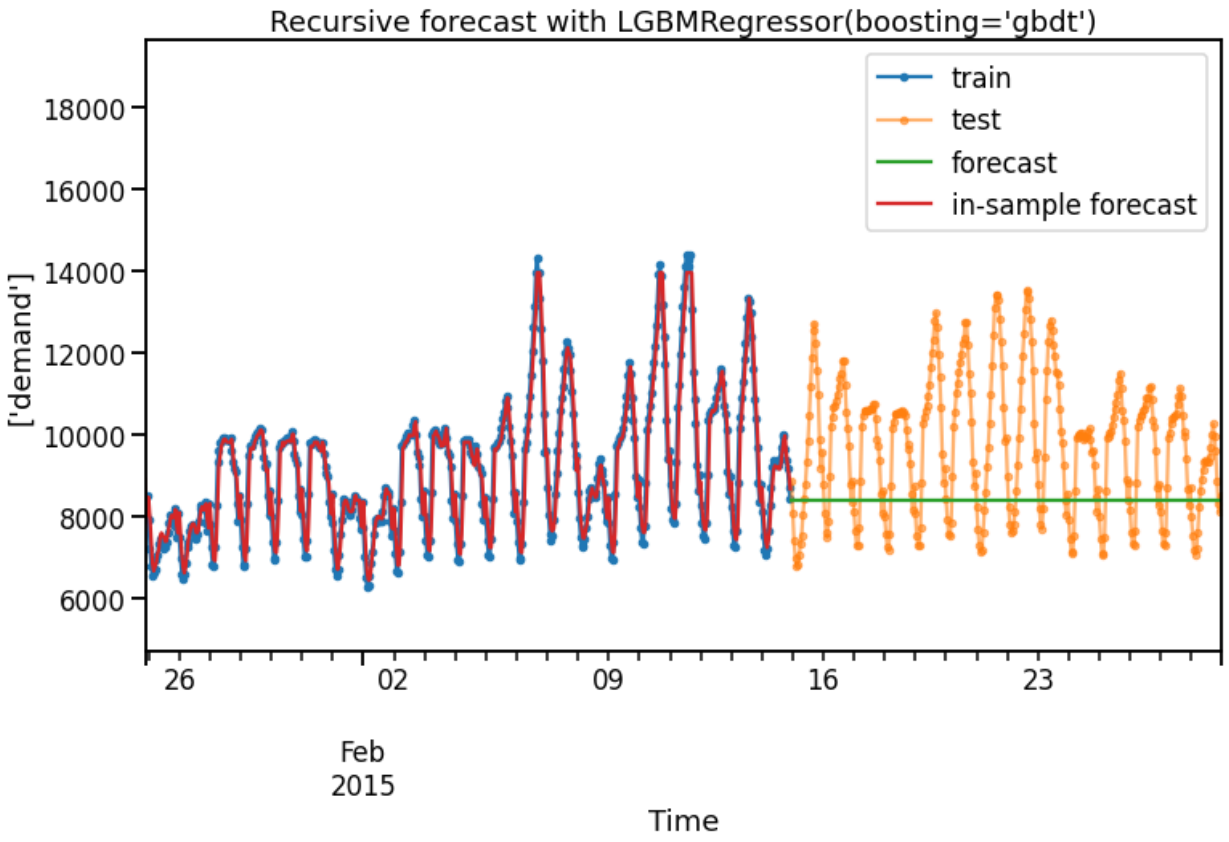
```
# Specify which datetime features to create
datetime_features = [
                        "year",
                        "month_of_year",
                        "week_of_year",
                        "day_of_year",
                        "day_of_week",
                        "hour_of_day",
                        "is_weekend",
                    ]

# Create the DateTimeFeatures transformer
transformer = DateTimeFeatures(
                               manual_selection=datetime_features, # Select which features to
                                                                  # create.
                               keep_original_columns=True, # Flag if we want to keep columns
                                                           # in dataframe passed to `transform`.
                              )

# Fit and transform to create our features
result = transformer.fit_transform(data)
result
```

date_time	demand	year	month_of_year	week_of_year	day_of_year	day_of_week	hour_of_day	is_weekend
2002-01-01 00:00:00	6919.366092	2002	1	1	1	1	0	0

Example: Electricity demand



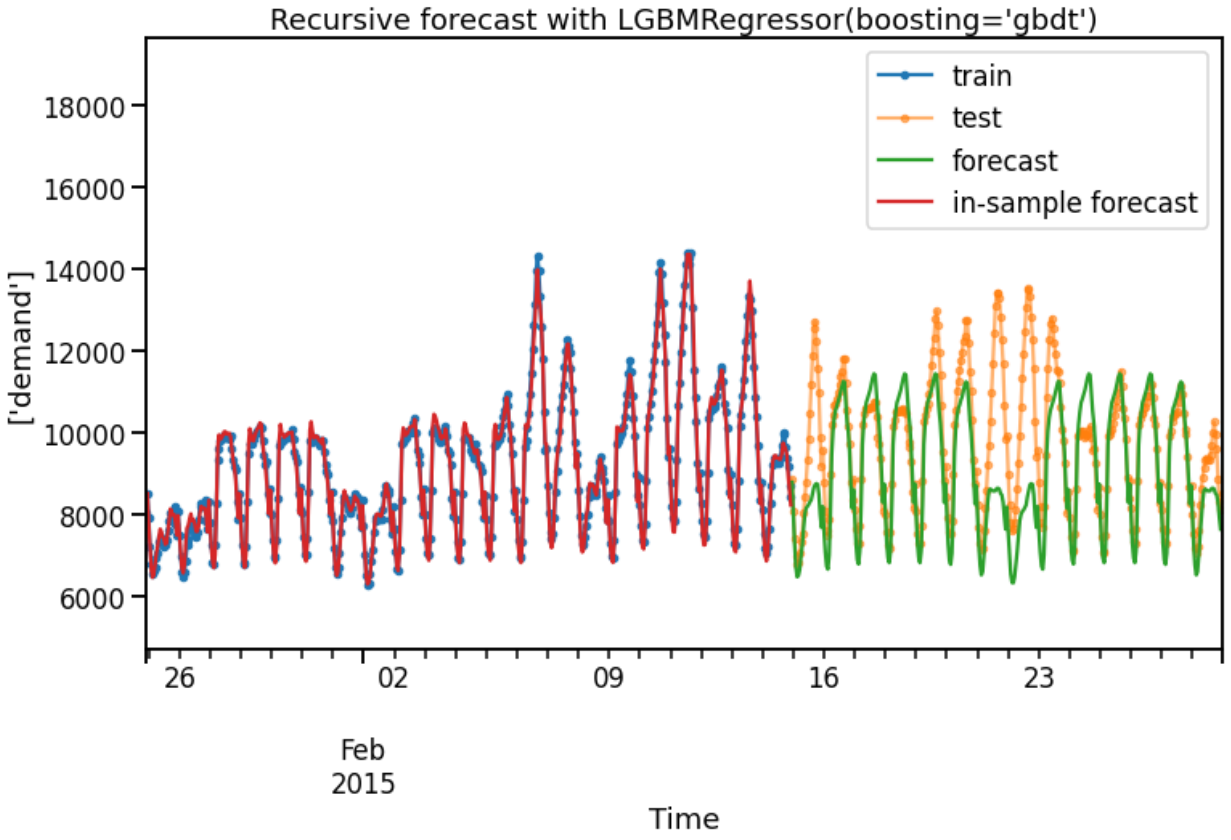
Features

- Trend feature: t,
- Lag of 1 hour: y_{t-1}

Model

- LightGBM

Example: Electricity demand



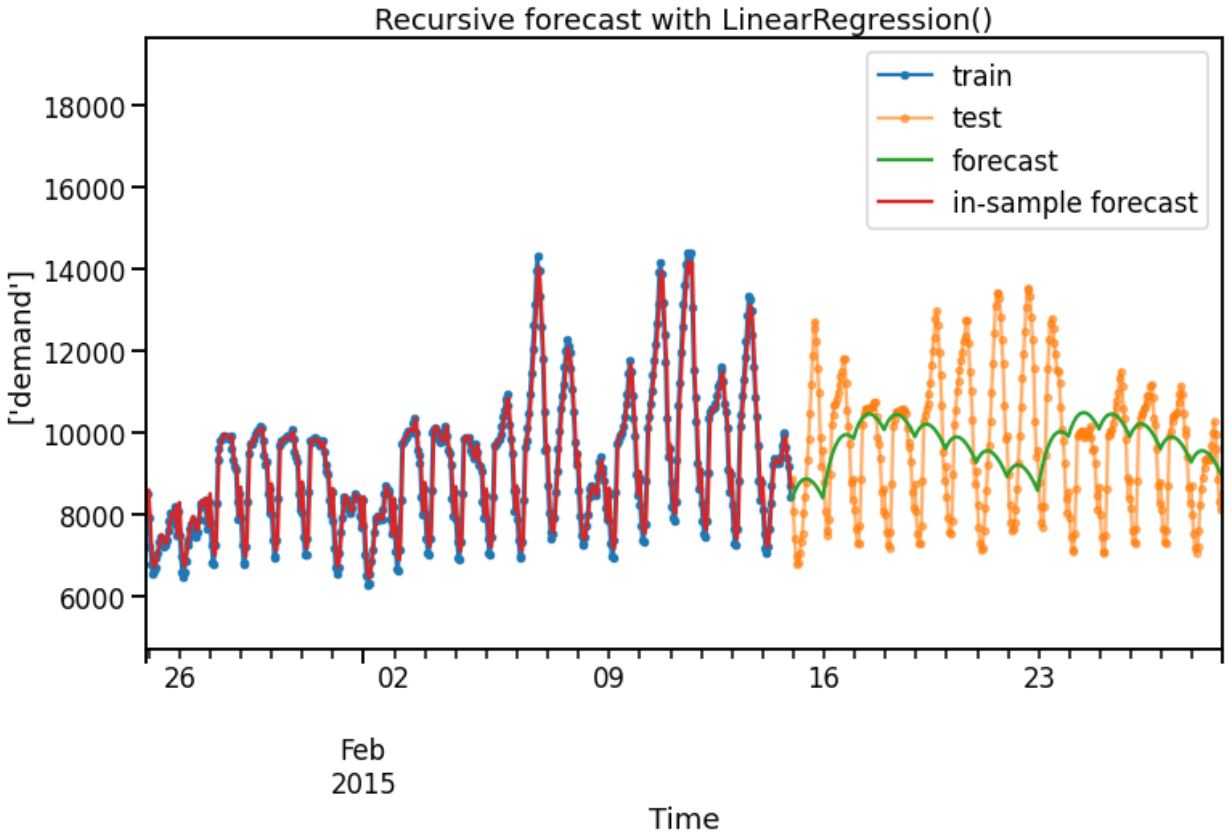
Features

- Trend feature: t,
- Lag of 1 hour: y_{t-1}
- **Hour of day**
- **Day of week**
- **Month of the year**

Model

- LightGBM

Example: Electricity demand



Features

- Trend feature: t,
- Lag of 1 hour: y_{t-1}
- **Hour of day**
- **Day of week**
- **Month of the year**

🤔

Model

- Linear regression