Lag features

Lag features

Motivation for lag features

- We want to predict future values of the target.
- Past values of the target are likely to be predictive.
- Past values of a feature could also be predictive (e.g., the sales on a day is related to advertising (ad) spend on prior days).

	target	feature
Date	**Sales**	**Ad spend**
2020-02-12	23	100
2020-02-13	30	120
2020-02-14	35	90
2020-02-15	30	80
2020-02-16	?	100

Lag features

- A lag feature is the value of the target or feature k period(s) in the past: x_{t-k}.
- k is the lag, set by the user.

	target	feature
Date	**Sales**	**Ad spend**
2020-02-12	23	100
2020-02-13	30	120
2020-02-14	35	90
2020-02-15	30	80
2020-02-16	?	100

Lag features

- A lag feature is the value of the target or feature k period(s) in the past: x_{t-k}.
- k is the lag, set by the user.

target features

	Date	Sales	Sales Lag 1	Ad spend
	2020-02-12	23	NaN	100
	2020-02-13	30	23	120
t - 2	2020-02-14	35	30	90
t - 1	2020-02-15	30	35	80
t	2020-02-16	?	30	100

k = 1

Lag features

- A lag feature is the value of the target or feature k period(s) in the past: x_{t-k}.
- k is the lag, set by the user.

target features

	Date	Sales	Sales Lag 1	Ad spend
	2020-02-12	23	NaN	100
	2020-02-13	30	23	120
t - 2	2020-02-14	35	30	90
t - 1	2020-02-15	30	35	80
t	2020-02-16	?	30	100

k = 1

Lag features

- A lag feature is the value of the target or feature k period(s) in the past: x_{t-k}.
- k is the lag, set by the user.

target features

	Date	Sales	Sales Lag 2	Ad spend
	2020-02-12	23	NaN	100
	2020-02-13	30	NaN	120
t - 2	2020-02-14	35	23	90
t - 1	2020-02-15	30	30	80
t	2020-02-16	?	35	100

k = 2

Lag features

	target	lag features **from target**		original features	lag features **from original features**	
Date	**Sales**	**Sales Lag 1**	**Sales Lag 3**	**Ad spend**	**Ad spend Lag 1**	**Ad spend Lag 2**
2020-02-12	23	NaN	NaN	100	NaN	NaN
2020-02-13	30	23	NaN	120	100	NaN
2020-02-14	35	30	NaN	90	120	100
2020-02-15	30	35	23	80	90	120
2020-02-16	?	30	30	100	80	90

We can create multiple lag features with different lags from the target and features.

Problem: Which lags to use? How many lag features to create?

Lag features

	target		lag features **from target**		original features	lag features **from original features**	
Date	**Sales**		**Sales Lag 1**	**Sales Lag 3**	**Ad spend**	**Ad spend Lag 1**	**Ad spend Lag 2**
2020-02-12	23		NaN	NaN	100	NaN	NaN
2020-02-13	30		23	NaN	120	100	NaN
2020-02-14	35		30	NaN	90	120	100
2020-02-15	30		35	23	80	90	120
2020-02-16	?		30	30	100	80	90

We can create multiple lag features with different lags from the target and features.

Problem: Which lags to use? How many lag features to create?

Lag features

	target		lag features **from target**		original features	lag features **from original features**	
Date	**Sales**		**Sales Lag 1**	**Sales Lag 3**	**Ad spend**	**Ad spend Lag 1**	**Ad spend Lag 2**
2020-02-12	23		NaN	NaN	100	NaN	NaN
2020-02-13	30		23	NaN	120	100	NaN
2020-02-14	35		30	NaN	90	120	100
2020-02-15	30		35	23	80	90	120
2020-02-16	?		30	30	100	80	90

We can create multiple lag features with different lags from the target and features.

Problem: Which lags to use? How many lag features to create?

Lag features

	target	lag features **from target**		original features	lag features **from original features**	
Date	**Sales**	**Sales Lag 1**	**Sales Lag 3**	**Ad spend**	**Ad spend Lag 1**	**Ad spend Lag 2**
2020-02-12	23	NaN	NaN	100	NaN	NaN
2020-02-13	30	23	NaN	120	100	NaN
2020-02-14	35	30	NaN	90	120	100
2020-02-15	30	35	23	80	90	120
2020-02-16	?	30	30	100	80	90

We can create multiple lag features with different lags from the target and features.

Problem: Which lags to use? How many lag features to create?

Lag features

	target	lag features **from target**		original features	lag features **from original features**	
Date	**Sales**	**Sales Lag 1**	**Sales Lag 3**	**Ad spend**	**Ad spend Lag 1**	**Ad spend Lag 2**
2020-02-12	23	NaN	NaN	100	NaN	NaN
2020-02-13	30	23	NaN	120	100	NaN
2020-02-14	35	30	NaN	90	120	100
2020-02-15	30	35	23	80	90	120
2020-02-16	?	30	30	100	80	90

We can create multiple lag features with different lags from the target and features.

Problem: Which lags to use? How many lag features to create?

Lag feature implementation in Pandas

pandas.DataFrame.shift

DataFrame.shift(*periods=1, freq=None, axis=0, fill_value=NoDefault.no_default*)

Shift index by desired number of periods with an optional time *freq*. [source]

When *freq* is not passed, shift the index without realigning the data. If *freq* is passed (in this case, the index must be date or datetime, or it will raise a *NotImplementedError*), the index will be increased using the periods and the *freq*. *freq* can be inferred when specified as "infer" as long as either freq or inferred_freq attribute is set in the index.

Parameters: **periods** : ***int***

Number of periods to shift. Can be positive or negative.

freq : ***DateOffset, tseries.offsets, timedelta, or str, optional***

Offset to use from the tseries module or time rule (e.g. 'EOM'). If *freq* is specified then the index values are shifted but the data is not realigned. That is, use *freq* if you would like to extend the index when shifting and preserve the original data. If *freq* is specified as "infer" then it will be inferred from the freq or inferred_freq attributes of the index. If neither of those attributes exist, a ValueError is thrown.

axis : ***{0 or 'index', 1 or 'columns', None}, default None***

Shift direction.

Lag feature implementation in Pandas

Original time series

```
df.head()
```

ds	y
1992-01-01	146376
1992-02-01	147079
1992-03-01	159336
1992-04-01	163669
1992-05-01	170068

Lag of 2 months

```
df.shift(periods=2).head()
```

ds	y
1992-01-01	NaN
1992-02-01	NaN
1992-03-01	146376.0
1992-04-01	147079.0
1992-05-01	159336.0

Lag of 2 months

```
df.shift(periods=2, freq='MS').head()
```

ds	y
1992-03-01	146376
1992-04-01	147079
1992-05-01	159336
1992-06-01	163669
1992-07-01	170068

Lag feature implementation in Pandas

```
freq = '2MS'
df_[f"y_lag_{freq}"] = df_["y"].shift(freq=freq)
df_.head()
```

ds	y	y_lag_2MS
1992-01-01	146376	NaN
1992-02-01	147079	NaN
1992-03-01	159336	146376.0
1992-04-01	163669	147079.0
1992-05-01	170068	159336.0

Lag feature implementation in Feature-engine

LagFeatures

class feature_engine.timeseries.forecasting.**LagFeatures**(*variables=None, periods=1, freq=None, sort_index=True, missing_values='raise', drop_original=False*) [source]

LagFeatures adds lag features to the dataframe. A lag feature is a feature with information about a prior time step.

LagFeatures has the same functionality as pandas `shift()` with the exception that only one of `periods` or `freq` can be indicated at a time. LagFeatures builds on top of pandas `shift()` in that multiple lags can be created at the same time and the features with names will be concatenated to the original dataframe.

To be compatible with LagFeatures, the dataframe's index must have unique values and no NaN.

LagFeatures works only with numerical variables. You can pass a list of variables to lag. Alternatively, LagFeatures will automatically select and lag all numerical variables found in the training set.

More details in the User Guide.

Parameters:

variables: list, default=None

The list of numerical variables to transform. If None, the transformer will automatically find and select all numerical variables.

periods: int, list of ints, default=1

Number of periods to shift. Can be a positive integer or list of positive integers. If list, features will be created for each one of the periods in the list. If the parameter `freq` is specified, `periods` will be ignored.

freq: str, list of str, default=None

Offset to use from the tseries module or time rule. See parameter `freq` in pandas `shift()`. It is the same functionality. If freq is a list, lag features will be created for each one of the frequency values in the list. If freq is not None, then this parameter overrides the parameter `periods`.

Lag feature implementation in Feature-engine

```python
from feature_engine.timeseries.forecasting import LagFeatures
```

```python
lag_transformer = LagFeatures(variables=['y'], freq=['1MS', '2MS', '3MS'])
lag_transformer.fit_transform(df)
```

	y	y_lag_1MS	y_lag_2MS	y_lag_3MS
ds				
1992-01-01	146376	NaN	NaN	NaN
1992-02-01	147079	146376.0	NaN	NaN
1992-03-01	159336	147079.0	146376.0	NaN
1992-04-01	163669	159336.0	147079.0	146376.0
1992-05-01	170068	163669.0	159336.0	147079.0
...	...	...	...	...

Summary

Lag features are a way of using the past to predict the future.

Can lag the target or other features.

Which features to lag and by how much? 🧐