Конспект по теме "Прогнозирование временных рядов"

Задача прогнозирования

Задача **прогнозирования временного ряда** состоит в построении модели, которая по историческим данным предскажет будущие значения временного ряда. Промежуток времени в будущем, на который строится прогноз, называется **горизонтом прогнозирования** (англ. *horizon forecast*). В задачах этой темы он будет равен одному шагу.

Если значения временного ряда, или функция ***x(t)***, где t — время, — это числа, то перед вами задача регрессии для временных рядов; если категории — задача классификации.

По исходным данным создадим обучающую (*train*) и тестовую (*test*) выборки. Перемешивать выборки в задаче прогнозирования временного ряда нельзя. Данные обучающей выборки должны предшествовать данным тестовой. Иначе тестирование модели будет некорректным: модель не должна обучаться на данных из будущего. Функция *train_test_split()* из модуля *sklearn.model_selection* по умолчанию перемешивает данные. Поэтому укажем аргумент **shuffle** (англ. «перетасовывать») равным *False*, чтобы разделить данные корректно:

```
import pandas as pd
from sklearn.model_selection import train_test_split

train, test = train_test_split(data, shuffle=False, test_size=0.2)
```

Качество прогноза

Чтобы проверять качество моделей в наших задачах, возьмём метрику *MAE*. Её можно легко интерпретировать.

Спрогнозировать временные ряды без обучения можно двумя способами:

1. Все значения тестовой выборки предсказываются одним и тем же числом (константой). Для метрики *MAE* — это медиана.
2. Новое значение *x(t)* прогнозируется предыдущим значением ряда, то есть *x(t-1)*. Этот способ не зависит от метрики.

Создание признаков

1. Календарные признаки (англ. *calendar features*)

Во многих данных тренды и сезонность привязаны к конкретной дате. Тип *datetime64* в Pandas уже содержит нужную информацию, осталось лишь представить её как отдельные столбцы:

```
# признак, в котором хранится год как число
data['year'] = data.index.year

# признак, в котором хранится день недели как число
data['dayofweek'] = data.index.dayofweek
```

2. «Отстающие значения» (англ. *lag features*)

Предыдущие значения временного ряда подскажут, будет ли функция *x(t)* расти или уменьшаться. Получим отстающие значения знакомой функцией *shift()*:

```
data['lag_1'] = data['target'].shift(1)
data['lag_2'] = data['target'].shift(2)
data['lag_3'] = data['target'].shift(3)
```

Для первых дат есть не все отстающие значения, поэтому в этих строках стоят *NaN*.

3. Скользящее среднее

Скользящее среднее как признак задаёт общий тренд временного ряда. Повторим, как его вычислять:

```
data['rolling_mean'] = data['target'].rolling(5).mean()
```

Скользящее среднее в моменте *t* учитывает текущее значение ряда *x(t)*. Это некорректно: целевой признак «убежал» в признаки. Вычисление скользящего среднего не должно включать в себя текущее значение ряда.