In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
import seaborn as sns
import torch


from catboost import CatBoostRegressor, Pool


sns.set(font_scale=1.5)
%matplotlib inline
```

In [ ]:

```
CALENDAR_DATA_PATH = "/kaggle/input/m5-forecasting-accuracy/calendar.csv"
SALES_DATA_PATH = "/kaggle/input/m5-forecasting-accuracy/sales_train_validatio
PRICES_DATA_PATH = "/kaggle/input/m5-forecasting-accuracy/sell_prices.csv"
```

In [ ]:

```
CALENDAR_DTYPES = {
    "event_name_1": "category",
    "event_name_2": "category",
    "event_type_1": "category",
    "event_type_2": "category",
    "weekday": "category",
    "wm_yr_wk": "int16",
    "wday": "int16",
    "month": "int16",
    "year": "int16",
    "snap_CA": "int16",
    "snap_TX": "int16",
    "snap_WI": "int16"
}
```

In [ ]:

```
PRICES_DTYPES = {
    "store_id": "category",
    "item_id": "category",
    "wm_yr_wk": "int16",
    "sell_price":"float32"
}
```

In [ ]:

```
num_columns = [f"d_{day}" for day in np.arange(1, 1914)]
categorical_columns = ['id', 'item_id', 'dept_id',
                       'store_id', 'cat_id', 'state_id']
SALES_DTYPES = {
    numcol : "int32" for numcol in num_columns
}
SALES_DTYPES.update({column: "category" for column in categorical_columns if
```

In [ ]:

```
calendar = pd.read_csv(CALENDAR_DATA_PATH,
                       parse_dates=["date"], date_parser = pd.to_datetime)
calendar = calendar.fillna('missin')
calendar = calendar.astype(CALENDAR_DTYPES)
```

In [ ]:

```
prices = pd.read_csv(PRICES_DATA_PATH, dtype=PRICES_DTYPES)
```

In [ ]:

```
sales = pd.read_csv(SALES_DATA_PATH, dtype=SALES_DTYPES)
```

In [ ]:

```
catcols = ['id', 'item_id', 'dept_id','store_id', 'cat_id', 'state_id']
```

In [ ]:

```
sales = pd.melt(sales,
             id_vars = catcols,
             value_vars = [col for col in sales.columns if col.startswith("d_'
             var_name = "d",
             value_name = "sales")
```

In [ ]:

```
sales = sales.merge(calendar, on= "d", copy = False)
```

In [ ]:

```
sales = sales.merge(prices, on = ["store_id", "item_id", "wm_yr_wk"], copy = F
```

In [ ]:

```
cat_feats = ['item_id', 'dept_id','store_id', 'cat_id', 'state_id'] + ["event_
useless_cols = ["id", "date", "sales","d", "wm_yr_wk", "weekday"]
train_cols = sales.columns[~sales.columns.isin(useless_cols)]
X_train = sales[train_cols]
y_train = sales["sales"]
```

In [ ]:

```
train_data = Pool(
    data=X_train,
    label=y_train,
    cat_features=cat_feats
)
```

In [ ]:

```
del calendar
del prices
del sales
del X_train
del y_train
```

In [ ]:

```
model = CatBoostRegressor(learning_rate=0.1, one_hot_max_size=1)
```

In [ ]:

```
model.fit(train_data)
```

In [ ]:

```

```