

```

import pandas as pd
import numpy as np
import lightgbm as lgb
import copy
import os
import glob
import pickle

class predict_model():
    def __init__(self, model_dir, mode):
        self.mode = mode

        # Load Model
        self.map_df = pd.read_csv(os.path.join(model_dir, "map_df.csv"))
        self.map_dict = {}
        for i in range(len(self.map_df)):
            self.map_dict[self.map_df.loc[i, "time"]] =
self.map_df.loc[i, "time_TE"]

        model_path = glob.glob(os.path.join(model_dir, "*.pkl"))
        self.models = []
        for p in model_path:
            self.models.append(pickle.load(open(p, 'rb')))

    def _preprocess(self, input):
        df = copy.deepcopy(input)
        # add weekday
        df["weekday"] = df["Timestamp"].dt.weekday
        df["Temp"] = df["Temp"].astype("float")
        df["Elec"] = df["Elec"].astype("float")

        # Add shifted Elec & Temp
        for i in range(1, 10):
            df["Elec_shift%i"%(i)] = df["Elec"].shift(i)
            df["Temp_shift%i"%(i)] = df["Temp"].shift(i)
        df["time"] = df["Timestamp"].dt.time
        df["month"] = df["Timestamp"].dt.month

        # Target Encoding

```

```
df["time_TE"] = df["time"].astype("str").replace(self.map_dict)
df = df.drop(["time", "Timestamp", "month"], axis=1)
df = df.fillna(method='bfill')
return df
```

```
def predict(self, input_):
    input = copy.deepcopy(input_)
    df = self._preprocess(input_)
    pred = np.array([m.predict(df) for m in self.models])
    pred = np.mean(pred, axis=0)
    input[self.mode] = pred
    return input
```