from datetime import datetime

import seaborn as sns

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from time import time

%matplotlib inline

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import OrdinalEncoder

from sklearn.model\_selection import RandomizedSearchCV, GridSearchCV

from sklearn.model\_selection import cross\_val\_score

import warnings

pd.set\_option("display.max\_columns", 100)

from datetime import datetime

def fxn():

warnings.warn("deprecated", DeprecationWarning)

with warnings.catch\_warnings():

warnings.simplefilter("ignore")

fxn()

import scipy.stats as stats

from sklearn.base import clone

from sklearn.metrics import make\_scorer

from sklearn.metrics import mean\_squared\_log\_error

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import r2\_score

df=pd.read\_csv( 'train.csv' )

df.shape

df.info()

df.head()

col\_names=list(df.columns)

for index,col\_name in enumerate(col\_names):

print(index ," :" ,col\_name)

df['datetime'] = pd.to\_datetime(df['datetime'])

df['month'] = df['datetime'].dt.month

df['hour'] = df['datetime'].dt.hour

df['weekday'] = df['datetime'].dt.dayofweek

df["day"]=df["datetime"].dt.day

df["year"]=df["datetime"].dt.year

df['dayofweek'] = df['datetime'].dt.dayofweek

df['month\_start'] = df['datetime'].dt.is\_month\_start

df.head()

df['holiday'].unique()

df.isnull().sum().sort\_values(ascending=False)

df=df.rename(columns={col\_names[11]:"Number of rented bikes"})

df=df.drop(['datetime','atemp','casual','registered'],axis=1)

df.describe().T

features\_plot=['Number of rented bikes', 'temp', 'humidity',

'windspeed']

df[features\_plot].describe()

df.isna().sum().plot(kind="bar")

plt.show()

plt.figure(figsize=(5,4))

df.groupby('hour').sum()['Number of rented bikes'].plot.bar(color=['#fa6d1d','#0780cf']);

plt.ylabel("Number of rented bikes")

plt.title("Number of rented bikes per hour");

plt.figure(figsize=(5,4))

df.groupby('month').sum()['Number of rented bikes'].plot.pie();

plt.xlabel("month");

plt.ylabel("Number of rented bikes")

plt.title("Number of rented bikes per month");

sns.barplot(data=df, x='year', y=df['Number of rented bikes']);

df.groupby('humidity').mean()['Number of rented bikes'].plot()

df.groupby('temp').mean()['Number of rented bikes'].plot()

df.groupby('windspeed').mean()['Number of rented bikes'].plot()

fig = plt.figure(figsize=(18, 8))

axes = fig.add\_subplot(1, 3, 1)

sns.regplot(data=df, x='temp', y='Number of rented bikes',ax=axes)

axes.set(title='Reg Plot for Temperature vs. Number of rented bikes')

axes = fig.add\_subplot(1, 3, 2)

sns.regplot(data=df, x='humidity', y='Number of rented bikes',ax=axes, color='r')

axes.set(title='Reg Plot for Humidity vs. Number of rented bikes')

axes = fig.add\_subplot(1, 3, 3)

sns.regplot(data=df, x='windspeed', y='Number of rented bikes',ax=axes, color='g')

axes.set(title='Reg Plot for Windspeed vs. Number of rented bikes')

plt.show()