

Входные данные в формате дата\кол-во мест

Дата: год и месяц

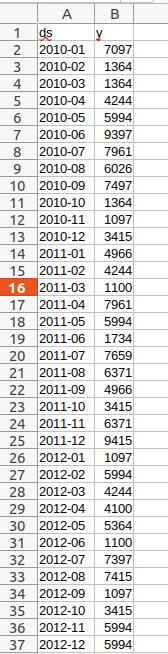
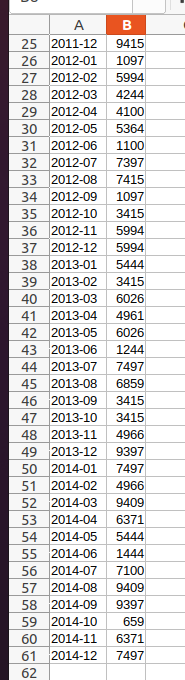
В данном случае у нас агрегированные данные (поэтому по оси y – тысячи)

По оси х месяца на весь 2014 год – от 0 до 11

В данном случае мы срезали с нашей выборки все данные за 2014 год и попытались на основе данных 2010-2013 года получить прогноз на 2014

Синяя ветка – исходные данные, оранжевая - прогноз

Данные в файле (с 2010 по 2014 год на каждый месяц введены данные):

from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from sklearn.metrics import mean\_absolute\_error

from matplotlib import pyplot

# load data

#path = 'newdata2.csv'

#path = 'ForProph.csv'

path = 'test2.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds']= to\_datetime(df['ds'])

#df['ds']= df['ds'].apply(lambda x: x.strftime('%Y-%B'))

# create test dataset, remove last 12 months

train = df.drop(df.index[-12:]) #здесь срезаем данные, чтобы они не попали в датафрейм

#print(train.tail())

# define the model

model = Prophet()

# fit the model

model.fit(train)

# define the period for which we want a prediction

future = list()

for v in range(1, 13):

date = '2014-%02d' % ( v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# calculate MAE between expected and predicted values for december

y\_true = df['y'][-12:].values #рассчитываем ошибку на данных за 12 месяцев

y\_pred = forecast['yhat'].values

mae = mean\_absolute\_error(y\_true, y\_pred)

print('MAE: %.3f' % mae)

# plot expected vs actual

pyplot.plot(y\_true, label='Actual')

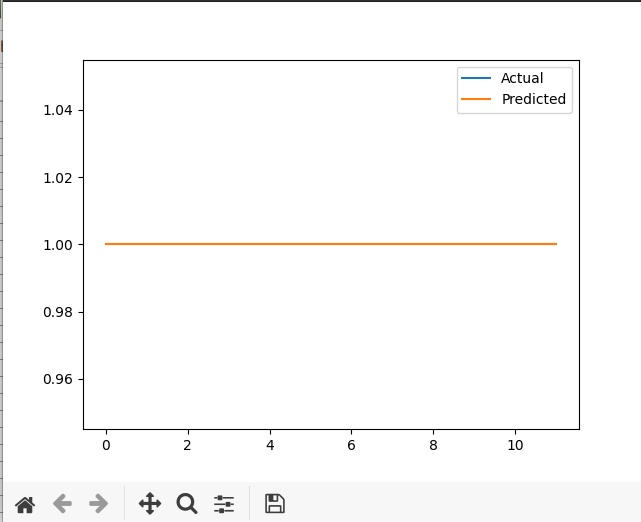
pyplot.plot(y\_pred, label='Predicted')

pyplot.legend()

pyplot.show()

предикт3.пай

\\--------------------------------------------------------------------------------------------------------------



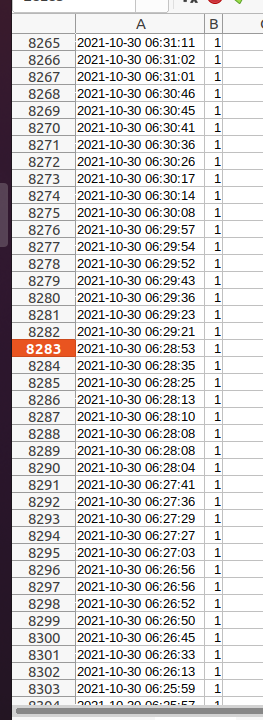
Входные данные в формате дата\кол-во мест

Дата: год, месяц, день, час, минуты, секунды

Кол-во мест: везде 1

Файл: 11\_10-30\_10 без обработки.csv (в коде переименован на newdata2.csv)

По оси х часы с 0 до 11 на день 2021-11-10 (прогноз на 12 часов). По оси y кол-во мест (в нашем случае стабильно 1)



from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from sklearn.metrics import mean\_absolute\_error

from matplotlib import pyplot

# load data

path = 'newdata2.csv'

#path = 'ForProph.csv'

#path = 'test2.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds']= to\_datetime(df['ds'])

#df['ds']= df['ds'].apply(lambda x: x.strftime('%Y-%B'))

# create test dataset, remove last 12 months

train = df.drop(df.index[-12:])

#print(train.tail())

# define the model

model = Prophet()

# fit the model

model.fit(train)

# define the period for which we want a prediction

future = list()

for v in range(1, 13):

date = '2021-11-10 %02d:00:00' % (v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# calculate MAE between expected and predicted values for december

y\_true = df['y'][-12:].values

y\_pred = forecast['yhat'].values

mae = mean\_absolute\_error(y\_true, y\_pred)

print('MAE: %.3f' % mae)

# plot expected vs actual

pyplot.plot(y\_true, label='Actual')

pyplot.plot(y\_pred, label='Predicted')

pyplot.legend()

pyplot.show()

предикт3

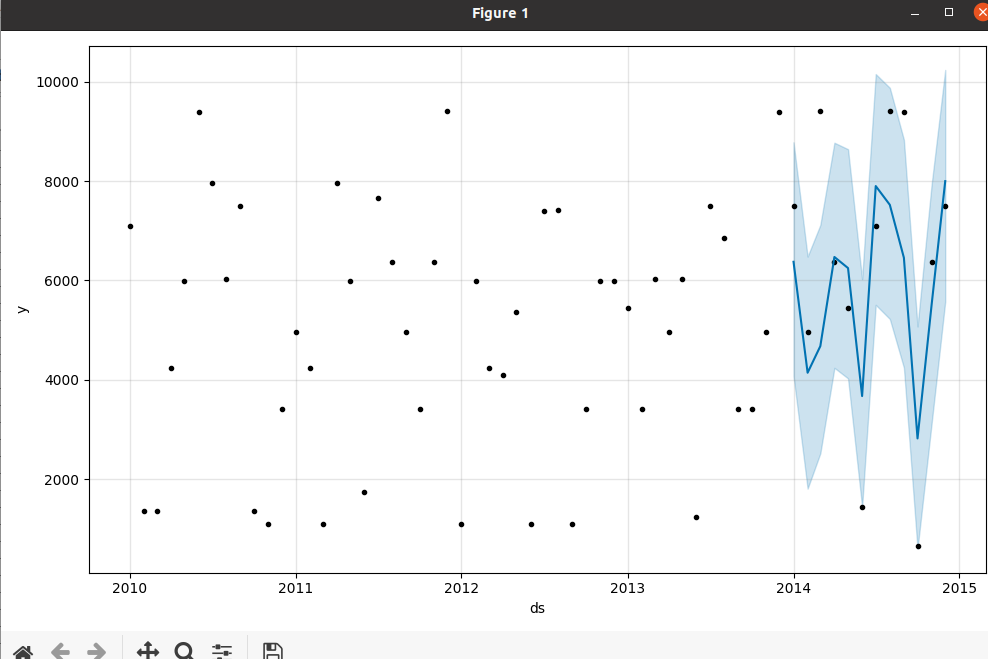
\\-----------------------------------------------------------------------------------------------------

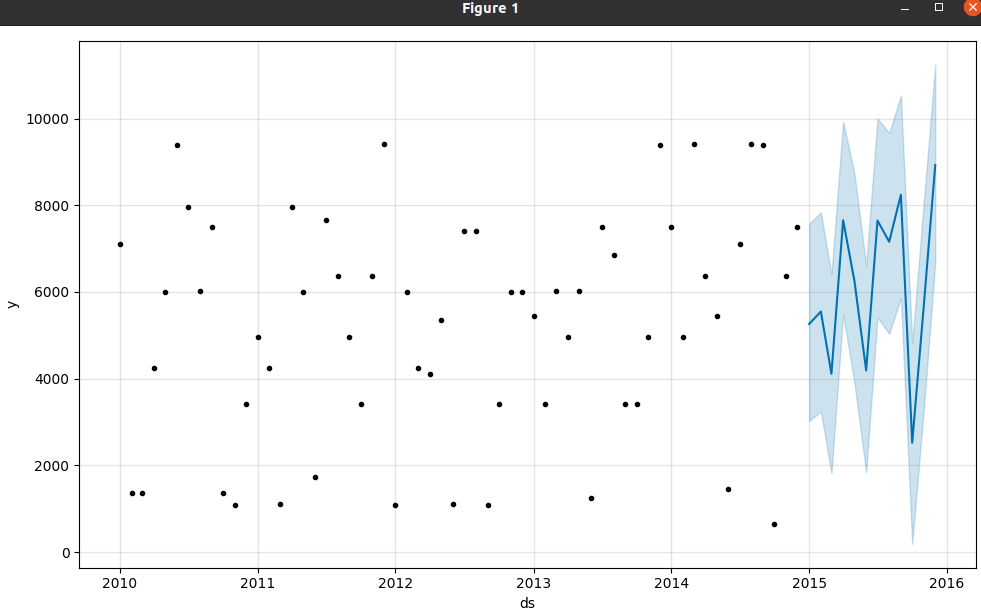
Входные данные в формате дата\кол-во мест

Дата: год, месяц

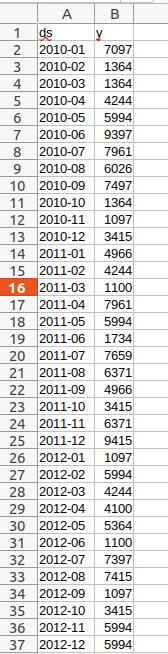
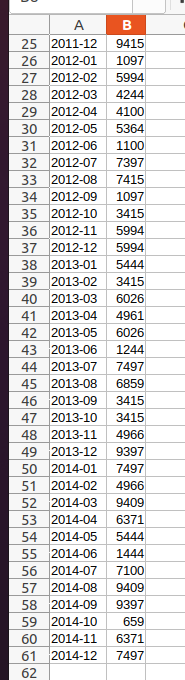
В данном случае у нас агрегированные данные (поэтому по оси y – тысячи)

По оси х все наши года (с 2010 по 2014 включительно)

Прогноз на 2014 год



Прогноз на 2015 год (на основе данных за 2010-2014 года)

# make an out-of-sample forecast

from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from matplotlib import pyplot

import psycopg2

from sklearn.metrics import mean\_absolute\_error

con = psycopg2.connect(

database="mybd",

user="doctor", # doctor

password="6675", # 6675

host="127.0.0.1",

port="5432"

)

print("Database opened successfully")

cur = con.cursor()

# load data

path = 'test2.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds'] = to\_datetime(df['ds'])

# define the model

model = Prophet()

# fit the model

model.fit(df)

# define the period for which we want a prediction

future = list()

#for i in range(2010, 2015):

for v in range(1, 13):

#for i in range(1, 24):

date = '2015-%02d' % ( v)

#date = '2021-11-10 %02d:00:00' % (v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# summarize the forecast

print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())

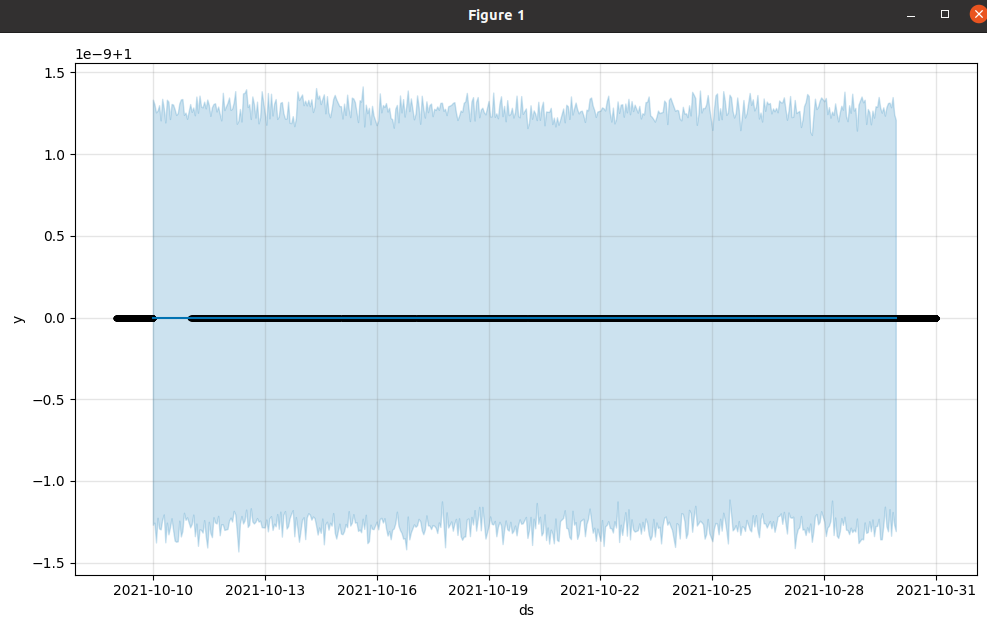
# plot forecast

model.plot(forecast)

pyplot.show()

проф

\\-----------------------------------------------------------------------------------------------------------------------

Входные данные в формате дата\кол-во мест

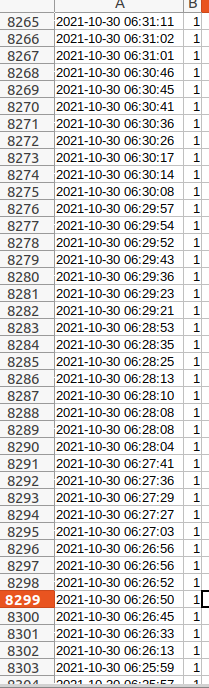
Дата: год, месяц, день, час, минуты, секунды

Кол-во мест: везде 1

Файл: 11\_10-30\_10 без обработки.csv (в коде переименован на newdata2.csv)

По оси х прогноз на 2021-10-10 – 2021-10-30

По оси y кол-во мест (в нашем случае стабильно 1)



# make an out-of-sample forecast

from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from matplotlib import pyplot

import psycopg2

from sklearn.metrics import mean\_absolute\_error

con = psycopg2.connect(

database="mybd",

user="doctor", # doctor

password="6675", # 6675

host="127.0.0.1",

port="5432"

)

print("Database opened successfully")

cur = con.cursor()

# load data

path = 'newdata2.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds'] = to\_datetime(df['ds'])

# define the model

model = Prophet(yearly\_seasonality=False,

weekly\_seasonality=True,

daily\_seasonality=True,

holidays\_prior\_scale=35)

# fit the model

model.fit(df)

# define the period for which we want a prediction

future = list()

for x in range(10, 30):

for v in range(0, 23):

date = '2021-10-%02d %02d:00:00' % (x, v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# summarize the forecast

print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())

# plot forecast

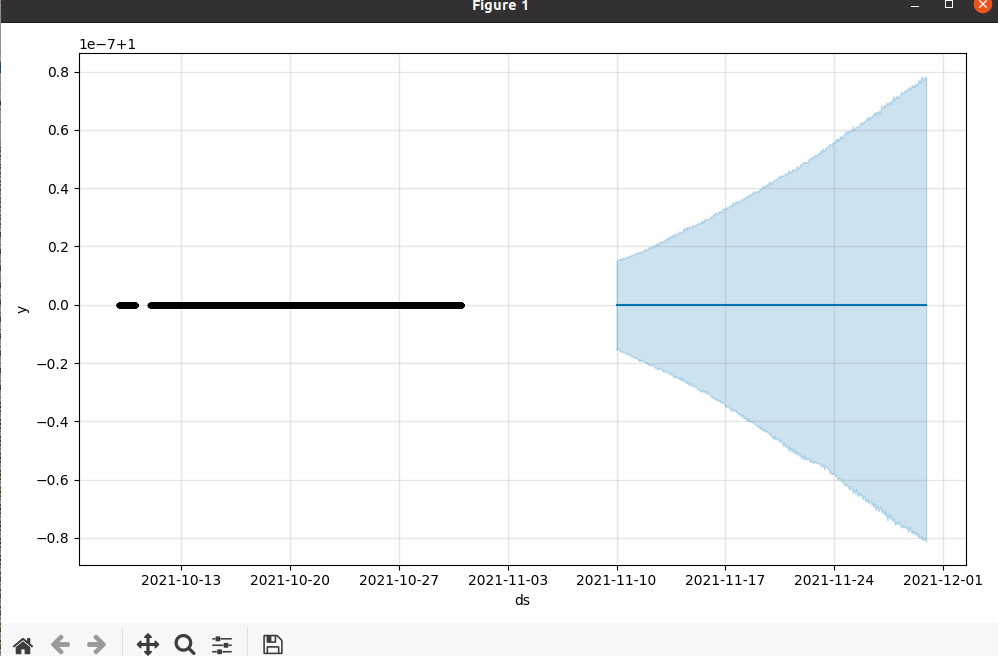
model.plot(forecast)

pyplot.show()

проф2

\\--------------------------------------------------------------------------------------------------------------------

Все как и в прошлом случае, но прогноз на будущие дни (2021-11-10 – 2021-11-30)



# make an out-of-sample forecast

from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from matplotlib import pyplot

import psycopg2

from sklearn.metrics import mean\_absolute\_error

con = psycopg2.connect(

database="mybd",

user="doctor", # doctor

password="6675", # 6675

host="127.0.0.1",

port="5432"

)

print("Database opened successfully")

cur = con.cursor()

# load data

path = 'newdata2.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds'] = to\_datetime(df['ds'])

# define the model

model = Prophet(yearly\_seasonality=False,

weekly\_seasonality=True,

daily\_seasonality=True,

holidays\_prior\_scale=35)

# fit the model

model.fit(df)

# define the period for which we want a prediction

future = list()

for x in range(10, 30):

for v in range(0, 23):

# for i in range(0, 59):

#date = '2021-11-%02d %02d:%02d:00' % (x, v, i)

date = '2021-11-%02d %02d:00:00' % (x, v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# summarize the forecast

print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())

# plot forecast

model.plot(forecast)

pyplot.show()

проф2

\\-----------------------------------------------------------------------------------------------------------------

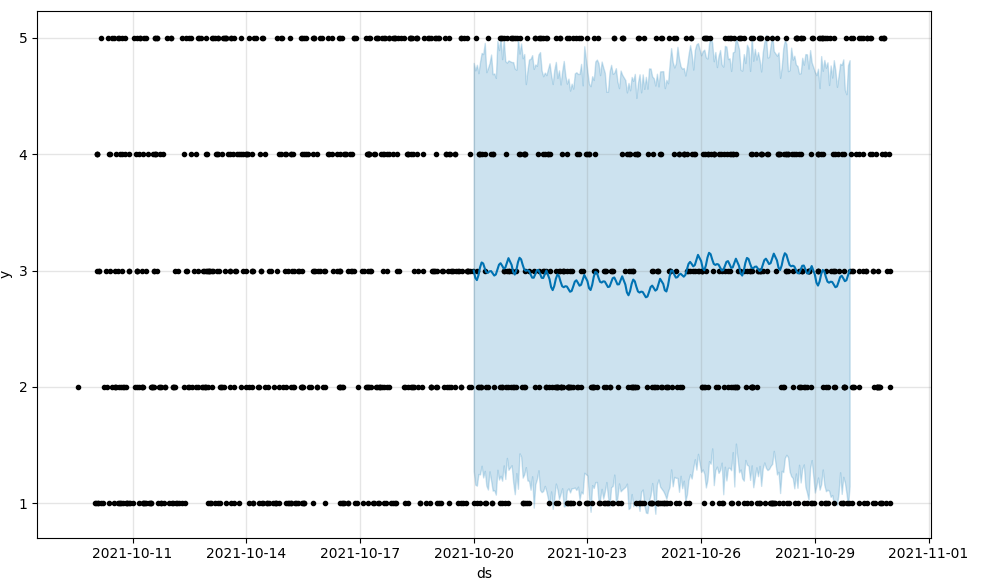
Файл ForProph.csv. Там храним данные из БД, где ориджин =1. Изначально было добавлено по 500 записей на каждый день с 2021-10-10 по 2021-10-30. Так как данные генерировались рандомно, прогноз представляет собой шум.

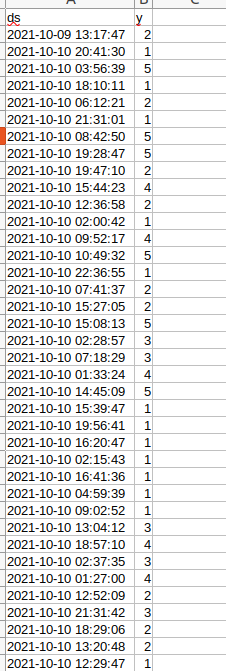
Входные данные в формате дата\кол-во мест

Дата: год, месяц, день, час,минуты,секунды

По оси х даты с 2021-10-10 по 2021-10-30

Прогноз проводим на 2021-10-20 – 2021-10-30 (на 10 дней).





# make an out-of-sample forecast

from pandas import read\_csv

from pandas import to\_datetime

from pandas import DataFrame

from fbprophet import Prophet

from matplotlib import pyplot

import psycopg2

from sklearn.metrics import mean\_absolute\_error

con = psycopg2.connect(

database="mybd",

user="doctor", # doctor

password="6675", # 6675

host="127.0.0.1",

port="5432"

)

print("Database opened successfully")

cur = con.cursor()

# load data

#path = 'newdata2.csv'

path = 'ForProph.csv'

df = read\_csv(path, header=0)

# prepare expected column names

df.columns = ['ds', 'y']

df['ds'] = to\_datetime(df['ds'])

# define the model

model = Prophet()

# fit the model

model.fit(df)

# define the period for which we want a prediction

future = list()

for x in range(20, 30):

for v in range(0, 23):

# for i in range(0, 59):

#date = '2021-11-%02d %02d:%02d:00' % (x, v, i)

date = '2021-10-%02d %02d:00:00' % (x, v)

future.append([date])

future = DataFrame(future)

future.columns = ['ds']

future['ds'] = to\_datetime(future['ds'])

# use the model to make a forecast

forecast = model.predict(future)

# summarize the forecast

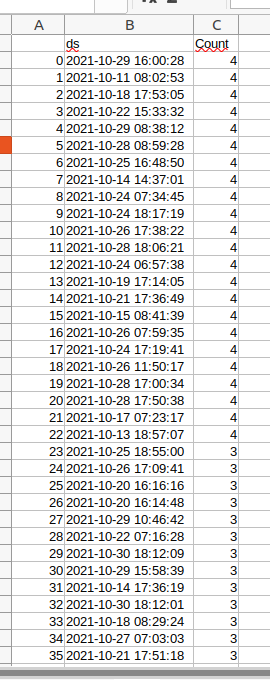
print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())

# plot forecast

model.plot(forecast)

pyplot.show()

Интенсивность заявок: даты из файла newdata2.csv (11\_10-30\_10 без обработки.csv)



import pandas

from pandas import read\_csv

paths = ['newdata2.csv']

path = paths[0]

df = read\_csv(path, header=0)

# print((df["TimeRequest"].unique()))

# df=pandas.DataFrame()

df['ds'] = pandas.to\_datetime(df['ds'])

dt = df.sort\_values(by='ds')

df = df.reset\_index(drop=True)

df.index = df.index + 1

print((dt["ds"]))

out = df['ds'].value\_counts()

#print(out)

df.rename(columns={'': 'ds', 'ds': 'Count'}, inplace=True)

#out.columns = ['TimeRequest', "count"]

#print(type(out))

out = pandas.DataFrame({'ds':out.index, 'Count':out.values})

out.to\_csv("Out.csv", sep=",")