# Import the needed libraries   
## Attention:For the installation of Prophet please follow the instructions written here: https://stackoverflow.com/questions/53178281/installing-fbprophet-python-on-windows-10  
import pandas as pd  
import numpy as np  
from fbprophet import Prophet  
import matplotlib.pyplot as plt

# Read in the given data and set the index column to the date column and parse the dates  
df=pd.read\_csv("../02\_data\_acquisition\_understanding/01\_data\_source/sickness\_table.csv", index\_col="date", parse\_dates=True )   
df["drivers\_atwork"]=df["n\_duty"]+df["sby\_need"]  
df.index.freq="D" #Set the frequence to Daily  
df.index

DatetimeIndex(['2016-04-01', '2016-04-02', '2016-04-03', '2016-04-04',  
 '2016-04-05', '2016-04-06', '2016-04-07', '2016-04-08',  
 '2016-04-09', '2016-04-10',  
 ...  
 '2019-05-18', '2019-05-19', '2019-05-20', '2019-05-21',  
 '2019-05-22', '2019-05-23', '2019-05-24', '2019-05-25',  
 '2019-05-26', '2019-05-27'],  
 dtype='datetime64[ns]', name='date', length=1152, freq='D')

# Prepare the Data for Prophet  
# Create the column "date"  
df["date"]=df.index  
# Create new DataFrames with the column "date"  
df\_new=pd.DataFrame(df["date"])  
# Create the column y="calls"  
df\_new["y"]=df["calls"]  
# Delete the index  
df\_new.reset\_index(drop=True, inplace=True)  
# Rename the columns to the Prophet specifics  
df\_new.rename(columns={"date": "ds", "y": "y"}, inplace=True)  
# Show the head of the DataFrame  
df\_new.head()

ds

y

0

2016-04-01

8154.0

1

2016-04-02

8526.0

2

2016-04-03

8088.0

3

2016-04-04

7044.0

4

2016-04-05

7236.0

#Define the number of days the model should predict the workforce plan  
numberofdaysforprediction=31  
  
#Create the Prohpet-Model,with an interval\_width of 95% (95% of the train values are in the predicted range)   
m = Prophet(interval\_width=0.975, daily\_seasonality=True,seasonality\_mode="multiplicative")  
# Train the Prophet-Model  
m.fit(df\_new)  
# Create future datafame with xx days (periods & freq) as basis for prediction  
future = m.make\_future\_dataframe(periods=numberofdaysforprediction,freq='D')  
# Make predictions for the created future dataframe  
forecast = m.predict(future)  
#plot the predictions   
m.plot(forecast, figsize=(20, 6));  
forecast.set\_index(forecast["ds"], inplace=True)

png

png

# Generating the prdiction of drivers on duty and drivers on standby for the next 31 days  
x\_Value=4.82  
df\_predresult=pd.DataFrame(forecast["ds"].iloc[len(forecast)-numberofdaysforprediction:])  
df\_predresult["pred\_n\_duty"]=forecast["yhat"].iloc[len(forecast)-numberofdaysforprediction:]/x\_Value  
df\_predresult["pred\_n\_sby"]=forecast["yhat\_upper"].iloc[len(forecast)-numberofdaysforprediction:]/x\_Value-forecast["yhat"].iloc[len(forecast)-numberofdaysforprediction:]/x\_Value  
  
#Round the values and cast to intger:   
df\_predresult["pred\_n\_duty"]=df\_predresult["pred\_n\_duty"].apply(lambda x: np.around(x))  
df\_predresult["pred\_n\_duty"]=df\_predresult["pred\_n\_duty"].apply(lambda x: int(x))  
df\_predresult["pred\_n\_sby"]=df\_predresult["pred\_n\_sby"].apply(lambda x: np.around(x))  
df\_predresult["pred\_n\_sby"]=df\_predresult["pred\_n\_sby"].apply(lambda x: int(x))  
#drop the column "ds" since the index is already the date   
df\_predresult.drop("ds", inplace=True, axis=1)

# Show the predictions for the next 31 days  
df\_predresult

pred\_n\_duty

pred\_n\_sby

ds

2019-05-28

2071

437

2019-05-29

2052

418

2019-05-30

2025

445

2019-05-31

2007

440

2019-06-01

1975

455

2019-06-02

1880

438

2019-06-03

2104

439

2019-06-04

2109

439

2019-06-05

2091

425

2019-06-06

2063

444

2019-06-07

2045

410

2019-06-08

2010

438

2019-06-09

1913

428

2019-06-10

2134

434

2019-06-11

2136

426

2019-06-12

2113

426

2019-06-13

2081

444

2019-06-14

2059

419

2019-06-15

2020

499

2019-06-16

1918

443

2019-06-17

2135

439

2019-06-18

2134

446

2019-06-19

2108

421

2019-06-20

2073

429

2019-06-21

2048

430

2019-06-22

2007

426

2019-06-23

1904

419

2019-06-24

2121

442

2019-06-25

2120

451

2019-06-26

2095

443

2019-06-27

2061

436

# Create an extract as csv --> the separator is a Semikolon and the decimals is a comma  
df\_predresult.to\_csv("Workforceplaning.csv", decimal=",",sep=";")