import pandas as pd import numpy as np import matplotlib.pyplot as plt import datetime import warnings warnings.filterwarnings('ignore') print('numpy version:', np. version) print('matplotlib version:', pd. version) #read csv file df1 = pd.read csv('C:/Users/Bilal Aktas/Desktop/Git Lab/Fontys/Fontys Subjects/Semester 4/Applied Data Scien df2 = pd.read csv('C:/Users/Bilal Aktas/Desktop/Git Lab/Fontys/Fontys Subjects/Semester 4/Applied Data Scien #df = pd.read csv ('C:/Users/Bilal Aktas/Desktop/CleanedDataset.csv') numpy version: 1.19.2 matplotlib version: 1.1.5 #add columns df1.columns = ['Delivery date', 'Delivery time', 'Pharmacy number', 'Pharmacy Postcode (2)', 'Year of birth' 'Gender', 'CNK', 'Product name', 'ATC code', 'Units', 'Price', 'Contribution'] df2.columns = ['Delivery date', 'Delivery time', 'Pharmacy number', 'Pharmacy Postcode (2)', 'Year of birth' 'Gender', 'CNK', 'Product name', 'ATC code', 'Units', 'Price', 'Contribution'] #check dataframe dfl.head(10) **Pharmacy** Year **Delivery Delivery** ATC Pharmacy **Postcode CNK Units Price Contribution** of Gender **Product name** date time number code (2) birth AMOXICLAV SANDOZ 500MG/125 MG COMP **0** 01/01/2017 00:00 7341765 21 1924 1 1715119 J01CR02 30 14.81 3.47 1 01/01/2017 5520523 WACHTHONORARIUM 0.00 00:00 7341765 21 1922 4.90 ZALDIAR 37,5 MG/325 **2** 01/01/2017 00:00 7341765 21 1925 1 1799931 N02AJ13 20 9.26 3.62 MG FILMOMH TABL 20 VASEXTEN CAPS BLIST **3** 01/01/2017 00:00 8272695 1932 2 1719400 C08CA12 19.22 4.98 28 X 10 MG 8272695 **4** 01/01/2017 00:00 1933 2 5520523 WACHTHONORARIUM 4.90 0.00 16 AACIDEXAM 5MG/ML **5** 01/01/2017 9111423 1931 1 1750132 H02AB02 6.15 0.39 OPL INJ FL INJ 1 X 1ML AACIDEXAM 5MG/ML **6** 01/01/2017 00:00 8272695 1933 1 1750132 H02AB02 6.15 0.39 OPL INJ FL INJ 1 X 1ML AMOXICLAV SANDOZ **7** 01/01/2017 8272695 1935 2 1715127 875 MG/125 MG COMP J01CR02 20 15.18 3.58 8 01/01/2017 00:00 8272695 1933 2 5520523 WACHTHONORARIUM 4.90 0.00 01/01/2017 00:00 8272695 1940 5520523 WACHTHONORARIUM 4.90 0.00 #gives information of dataframe df1.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 22705348 entries, 0 to 22705347 Data columns (total 12 columns): # Column Dtvpe 0 Delivery date object Delivery time object Pharmacy number int64 Pharmacy Postcode (2) int64 4 Year of birth int64 int64 5 Gender CNK int64 Product name object object 8 ATC code int64 9 Units 10 Price float64 float64 11 Contribution dtypes: float64(2), int64(6), object(4) memory usage: 2.0+ GB df2.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 5072146 entries, 0 to 5072145 Data columns (total 12 columns): # Column Dtype O Delivery date object 1 Delivery time object 2 Pharmacy number int64 1 Delivery time 3 Pharmacy Postcode (2) int64 Year of birth int64 4 int64 Gender CNK int64 Product name 7 object 8 ATC code object 9 Units int64 10 Price float64 11 Contribution float64 dtypes: float64(2), int64(6), object(4) memory usage: 464.4+ MB #sums all the null values of the columns in the dataframe df1.isnull().sum() Out[81]: Delivery date 4941181 Delivery time Pharmacy number 0 Pharmacy Postcode (2) Year of birth 0 Gender 0 0 CNK Product name ATC code Units 0 0 Price Contribution dtype: int64 df2.isnull().sum() Out[82]: Delivery date 0 972059 Delivery time Pharmacy number 0 Pharmacy Postcode (2) 0 Year of birth 0 0 Gender CNK Product name 0 ATC code Units 0 Price 0 Contribution dtype: int64 df1.isnull().any() Out[83]: Delivery date False Delivery time True Pharmacy number False Pharmacy Postcode (2) False Year of birth False Gender False False CNK False Product name ATC code False Units False Price False Contribution False dtype: bool In [84]: #filling na values in delivery time column with 00:00 TimeNull = df1['Delivery time'] TimeNull.fillna('00:00',inplace=True) TimeNull = df2['Delivery time'] TimeNull.fillna('00:00',inplace=True) #dropping outliers in gender column df1 = df1[df1['Gender'] != 0] df1 = df1[df1['Gender'] != 3] df2 = df2[df2['Gender'] != 0] df2 = df2[df2['Gender'] != 3] df1["Gender"].value counts(sort=True) Out[88]: 2 13061752 9565378 Name: Gender, dtype: int64 df2["Gender"].value counts(sort=True) Out[89]: 2 2856183 2203711 Name: Gender, dtype: int64 #transform into datetime df1['Delivery date'] = pd.to datetime(df1['Delivery date'], infer datetime format=True) df2['Delivery date'] = pd.to_datetime(df2['Delivery date'], infer_datetime_format=True) #checking the value which date was both d-m-Y and m-d-Y rslt df = df1[df1['Delivery date'] > '2017-01-30'] print('\nResult dataframe :\n', rslt df) Result dataframe : Delivery date Delivery time Pharmacy number Pharmacy Postcode (2) \ 2018-01-01 00:00 7981002 30 3103 7729509 30 7729509 3104 30 2018-01-01 7981002 3105 30 2018-01-01 7729509 3106 30 22705343 2019-12-31 23:56 7068483 23:56 7068483 90 23:56 7068483 22705344 2019-12-31 90 2019-12-31 2019-12-31 23:56 23:56 22705345 7068483 90 22705346 7068483 90 22705347 2019-12-31 23:56 7068483 90 Year of birth Gender CNK \ 3102 1926 2 5520465

 1927
 2
 5520465

 1930
 2
 708099

 1927
 2
 743294

 1929
 2
 743294

 3103 3104 3105 3106 1963 2 2139137 1962 2 22705343 1962 22705344 2 2139137 1961 2 2139137 22705345 2 2139137 1962 22705346 22705347 1962 2 2139137 Product name ATC code Units 3102 HONORARIUM PER WEEK PER RUSTOORDBEWONER ROB-RVT \cap 3103 HONORARIUM PER WEEK PER RUSTOORDBEWONER ROB-RVT 0 3104 GLUCOPHAGE COMP 1 X 850 MG A10BA02 3105 L THYROXINE CHRISTIAENS COMP 1 X 100 MCG H03AA01 3106 L THYROXINE CHRISTIAENS COMP 1 X 100 MCG H03AA01 0 22705343 FLOCATH QUICK SONDE INTERM.+AUTOLUB. CH14 20CM 30 0 22705344 FLOCATH QUICK SONDE INTERM.+AUTOLUB. CH14 20CM 30 0 22705345 FLOCATH QUICK SONDE INTERM.+AUTOLUB. CH14 20CM 30 0 22705346 FLOCATH QUICK SONDE INTERM.+AUTOLUB. CH14 20CM 30 0 22705347 FLOCATH QUICK SONDE INTERM.+AUTOLUB. CH14 20CM 30 Price Contribution 3102 3.09 0.00 3103 3.09 0.00 3104 0.03 0.00 3105 0.04 0.01 3106 0.04 0.01 . . . 22705343 81.81 22705344 81.81 0.00 22705345 81.81 0.00 22705346 81.81 0.00 22705347 81.81 0.00 [21969832 rows x 12 columns] df2.isnull().any() Out[93]: Delivery date False Delivery time Pharmacy number False Pharmacy Postcode (2) False Year of birth False Gender False CNK False False Product name ATC code False Units False Price False False Contribution dtype: bool In [94]: df1['Delivery date'].min() Out[94]: Timestamp('2017-01-01 00:00:00') df1['Delivery date'].max() Out[95]: Timestamp('2019-12-31 00:00:00') df2['Delivery date'].min() Out[96]: Timestamp('2020-01-01 00:00:00') df2['Delivery date'].max() Out[97]: Timestamp('2020-12-06 00:00:00') df = df1In [341... df['ATCShort'] = df['ATC code'].str[:3] df.head() **Pharmacy** Year **Delivery** Delivery Pharmacy **Postcode** of Gender **CNK Product name Units Price Contribution ATCShort** date time number code birth (2) AMOXICLAV SANDOZ 2017-0 J01CR02 J01 00:00 7341765 1924 500MG/125 MG 30 14.81 3.47 21 1 1715119 01-01 COMP 30 2017-00:00 0.00 1 7341765 21 1922 1 5520523 WACHTHONORARIUM 4.90 01-01 ZALDIAR 37,5 MG/325 2017-2 MG FILMOMH TABL N02 00:00 7341765 1925 1 1799931 N02AJ13 9.26 3.62 21 20 01-01 20 2017-VASEXTEN CAPS BLIST 2 1719400 C08CA12 3 00:00 1932 4.98 C08 8272695 16 28 19.22 01-01 28 X 10 MG 2017-2 5520523 WACHTHONORARIUM 4.90 0.00 00:00 8272695 1933 16 0 01-01 In [342.. df['Age'] = 2021 -df['Year of birth'] In [343... tempMed = ['A02', 'B01', 'C07', 'C03', 'A10'] df['ATCShort'].isin(tempMed) med filter = df[df['ATCShort'].isin(tempMed)] med_filter['ATC_ABB'] = med_filter['ATCShort'].map({'A02': 0, 'B01': 1, 'C07': 2,'C03':3,'A10':4}) **Particular ATC** A02 Drugs for acid related disorders df =df.loc[(df["ATCShort"]=="A02")] df = df.sort values('Delivery date') In [269.. df = df.groupby('Delivery date')['Units'].sum().reset index() df = df.set index('Delivery date') y= df['Units'].resample('MS').mean() import itertools import statsmodels.api as sm p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: (U, Ι, \cup) \times $(\cup$, ⊥, SARIMAX: (0, 1, 0) x (1, 0, 0, 12) In [274... for param in pdq: for param seasonal in seasonal pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param seasonal, results.aic)) except: continue $\mathtt{ARIMA}\,(0,\ 0,\ 0)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC:890.1893698252876}$ ARIMA(0, 0, 0) \times (0, 0, 1, 12)12 - AIC:565.4150495709108 ARIMA(0, 0, 0) \times (0, 1, 0, 12)12 - AIC:488.61552023538275 ARIMA(0, 0, 0) \times (0, 1, 1, 12)12 - AIC:226.91563704043097 $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:506.58439992497085$ ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:487.45073518958975 $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:244.91781409333925$ $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:227.4309002911438$ $\mathtt{ARIMA}\,(0,\ 0,\ 1)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!841.7531652617959$ ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:591.0522114625408 ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:544.2489265865637 $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC:247.82250916942868$ $ARIMA(0, 0, 1) \times (1, 1, 1, 12) 12 - AIC:206.23406439629193$ $\mathtt{ARIMA}\,(\mathtt{0,\ 1,\ 0})\,\mathtt{x}\,(\mathtt{0,\ 0,\ 0,\ 12})\,\mathtt{12\ -\ AIC:}715.5535027767473$ ARIMA(0, 1, 0) \times (0, 0, 1, 12)12 - AIC:450.6934871778215 ARIMA(0, 1, 0) \times (0, 1, 0, 12)12 - AIC:466.64894655619077 ARIMA(0, 1, 0) \times (0, 1, 1, 12)12 - AIC:209.5591478088914 ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:464.18008807307257 ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:452.32260504922726 ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:228.62534785418762 $ARIMA(0, 1, 0) \times (1, 1, 1, 12) 12 - AIC:210.86393485813971$ $\mathtt{ARIMA}\,(0,\ 1,\ 1)\,\mathtt{x}\,(0,\ 0,\ 0,\ 12)\,12\ -\ \mathtt{AIC:695.046720268938}$ $ARIMA(0, 1, 1) \times (0, 0, 1, 12) 12 - AIC:422.07021234262965$ $ARIMA(0, 1, 1) \times (0, 1, 0, 12) 12 - AIC:436.34996578505263$ $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:186.03117024629802$ ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:463.3246571002089 $ARIMA(0, 1, 1) \times (1, 0, 1, 12) 12 - AIC:422.70083961659304$ ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:226.08209352365705 $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:187.9893516183192$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!737.7769302128178$ $ARIMA(1, 0, 0) \times (0, 0, 1, 12) 12 - AIC:473.02317541914545$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:482.6830819161955$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:229.47228814400626$ ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:474.0272287464602 $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC:474.7747122747352$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:228.22177370530966$ $ARIMA(1, 0, 0) \times (1, 1, 1, 12) 12 - AIC:229.48731494588048$ $\mathtt{ARIMA}\,(1,\ 0,\ 1)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!727.0772397435351$ $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:443.13535438619806$ $\mathtt{ARIMA}\,(1,\ 0,\ 1)\,\mathtt{x}\,(0,\ 1,\ 0,\ 12)\,12\ -\ \mathtt{AIC:}464.1666298725106$ $ARIMA(1, 0, 1) \times (0, 1, 1, 12) 12 - AIC: 206.0205645972613$ ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:466.3082497166157ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:444.9078450877433ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:227.35806741441394 $ARIMA(1, 0, 1) \times (1, 1, 1, 12) 12 - AIC: 207.57566652733314$ $ARIMA(1, 1, 0) \times (0, 0, 12) 12 - AIC:716.779357006552$ $ARIMA(1, 1, 0) \times (0, 0, 1, 12) 12 - AIC:452.9650091339171$ $ARIMA(1, 1, 0) \times (0, 1, 0, 12) 12 - AIC:467.80699409582263$ $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:210.60037868282473$ ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:453.8014088435554 ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:454.6733451016035 $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:210.90829922949393$ $ARIMA(1, 1, 0) \times (1, 1, 1, 12) 12 - AIC:212.18960693488302$ $ARIMA(1, 1, 1) \times (0, 0, 0, 12) 12 - AIC:706.7035572614831$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:423.29342274244544$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:433.39131412280165$ $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:187.7711601683244$ 0, 0, 12)12 - AIC:444.13324370360453 ARIMA(1, $1, 1) \times (1,$ ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:424.5557390740716 $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC:207.6832936972972$ $ARIMA(1, 1, 1) \times (1, 1, 1, 12) 12 - AIC:189.4249761734752$ mod = sm.tsa.statespace.SARIMAX(y, order=(0, 1, 1), $seasonal_order=(0, 1, 1, 12),$ enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025 0.975] ma.L1 -0.8505 0.167 -5.087 0.000 -1.178 0.2420 0.324 0.746 0.456 -0.394-0.523 ma.S.L12 0.2420 0.323 sigma2 4.99e+07 1.59e-10 3.14e+17 0.746 0.456 -0.394 0.878 0.000 4.99e+07 pred = results.get_prediction(start=pd.to_datetime('2018-12-01'), dynamic=False) pred_ci = pred.conf_int() ax = y['2013':].plot(label='observed') pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill_between(pred_ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity', fontsize=15) ax.set_title ('The amount of Sales of medicines for acid related disorders',fontsize=20) plt.legend() plt.show() The amount of Sales of medicines for acid related disorders observed One-step ahead Forecast 100000 90000 Sales in Quantity 80000 70000 60000 Jul Jul Jul Jan 2017 Date y forecasted = pred.predicted mean y truth = y['2018-12-01':]# Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 29982136.72 The Root Mean Squared Error of our forecasts is 5475.59 In [279... pred uc = results.get forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred uc.predicted mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantity', fontsize=15) ax.set title ('The amount of Sales of medicines for acid related disorders', fontsize=20) plt.show() The amount of Sales of medicines for acid related disorders Forecast 120000 110000 100000 Sales in Quantity 90000 80000 70000 60000 2019 2017 2023 2025 2027 Date **B01** Antithrombotic agents df =df.loc[(df["ATCShort"]=="B01")] df = df.sort_values('Delivery date') df = df.groupby('Delivery date')['Units'].sum().reset index() df = df.set index('Delivery date') y = df['Units'].resample('MS').mean() In [304... import itertools import statsmodels.api as sm p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param seasonal in seasonal pdq: try: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param seasonal, results.aic)) except: $ARIMA(0, 0, 0) \times (0, 0, 0, 12) 12 - AIC:896.0673609781238$ $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:568.6654466537661$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:478.3529179086874$ ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:479.04756132240084 $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:245.8690452705066$ $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:226.4879552057049$ $ARIMA(0, 0, 1) \times (0, 0, 0, 12) 12 - AIC:848.0400346971867$ $ARIMA(0, 0, 1) \times (0, 0, 1, 12) 12 - AIC:548.4034664776593$ $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC:457.7757570223302$ $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC:208.5102130248407$ ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:634.7711509810224ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:547.8280583968041ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:247.43192741220003 ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:206.56347696394124ARIMA(0, 1, 0)x(0, 0, 12)12 - AIC:709.5442319264256 $ARIMA(0, 1, 0) \times (0, 0, 1, 12) 12 - AIC:454.49972952020835$ $ARIMA(0, 1, 0) \times (0, 1, 0, 12) 12 - AIC:465.62396026115755$ ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:456.03838547024515 ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:234.30005242599958 ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:214.01700140053637 $ARIMA(0, 1, 1) \times (0, 0, 0, 12) 12 - AIC:686.3209738238764$ $ARIMA(0, 1, 1) \times (0, 0, 1, 12) 12 - AIC:430.5355982991507$ $ARIMA(0, 1, 1) \times (0, 1, 0, 12) 12 - AIC:435.5994441320641$ $\mathtt{ARIMA}\,(0,\ 1,\ 1)\,\mathtt{x}\,(0,\ 1,\ 1,\ 12)\,12\ -\ \mathtt{AIC:}188.41671152161072$ ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:471.2676780018432ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:430.4224939204161 ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:228.70057981327236 $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:190.3689878798538$ $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:731.7769369698151$ $ARIMA(1, 0, 0) \times (0, 0, 1, 12) 12 - AIC:476.5042671635142$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:477.8034993625623$ ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:478.16495985346234 ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:226.50579311846178 $ARIMA(1, 0, 0) \times (1, 1, 1, 12) 12 - AIC:228.0539840261546$ $ARIMA(1, 0, 1) \times (0, 0, 12) 12 - AIC:707.4052606846315$ $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:449.82215510371134$ $ARIMA(1, 0, 1) \times (0, 1, 0, 12) 12 - AIC:459.491487195816$ $ARIMA(1, 0, 1) \times (0, 1, 1, 12) 12 - AIC:210.42314935968972$ ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:473.24404000887415ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:451.7042012051508ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:227.5722840197307 $ARIMA(1, 0, 1) \times (1, 1, 1, 12) 12 - AIC:208.43870580008758$ $ARIMA(1, 1, 0) \times (0, 0, 12) 12 - AIC:710.8231961401093$ $ARIMA(1, 1, 0) \times (0, 0, 1, 12) 12 - AIC:455.6892442938954$ $ARIMA(1, 1, 0) \times (0, 1, 0, 12) 12 - AIC:465.08971494963373$ $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:211.66134633284753$ $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:211.91953967277385$ ARIMA(1, 1, 0)x(1, 1, 1, 12)12 - AIC:205.58818018035376 $ARIMA(1, 1, 1) \times (0, 0, 0, 12) 12 - AIC:687.7014702973195$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:430.85220007050026$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:437.51195840380274$ $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:189.271549598417$ ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:452.6138692467982 $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:432.30093236386176$ ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:209.4832369755357 ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:183.04272450065957 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1), seasonal order=(1, 1, 1, 12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025 0.975]ar.L1 -0.6986 0.407 -1.716 0.086 -1.497 0.099 0.191 ma.L1 -0.3908 -2.045 0.041 -0.765 -0.016 -7.131 -1.3252 ar.S.L12 0.000 -1.689-0.961 ma.S.L12 1.9238 0.309 6.236 0.000 1.319 2.528 6.733e+06 1.91e-08 3.53e+14 sigma2 0.000 6.73e+06 6.73e+06 ______ pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred_ci = pred.conf int() ax = y['2013':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantities', fontsize=15) ax.set title ('The amount of Sales of Antithrombotic agents', fontsize=20) plt.legend() plt.show() The amount of Sales of Antithrombotic agents 140000 observed One-step ahead Forecast 130000 120000 Sales in Quantities 110000 100000 90000 80000 70000 Jul Jan 2018 Jan 2019 Jan 2017 y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y forecasted - y truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 192690511.41 The Root Mean Squared Error of our forecasts is 13881.3 In [309... pred uc = results.get forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title ('The amount of Sales of Antithrombotic agents', fontsize=20) plt.legend() plt.show() The amount of Sales of Antithrombotic agents 150000 100000 50000 Sales in Quantities -50000 -100000 -150000 Date C07 Beta blocking agents In [314... df =df.loc[(df["ATCShort"]=="C07")] df = df.sort_values('Delivery date') df = df.groupby('Delivery date')['Units'].sum().reset_index() df = df.set_index('Delivery date') y = df['Units'].resample('MS').mean() import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ import warnings warnings.filterwarnings("ignore") for param in pdq: for param_seasonal in seasonal_pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal_order=param_seasonal, enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic)) except: continue ARIMA(0, 0, 0)x(0, 0, 12)12 - AIC:893.7657990697556 $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:566.9994057938942$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:473.76891702117746$ $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC:225.62999289287296$ $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:493.8050736441952$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:475.3502768095257$ ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:244.8979015169467 $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:225.877364984478$ ARIMA(0, 0, 1)x(0, 0, 12)12 - AIC:845.7127939761153ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:546.9131193179551ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:452.8152053316322 $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC:208.26409353966244$ ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:632.860111402267 $ARIMA(0, 0, 1) \times (1, 0, 1, 12) 12 - AIC:546.3098524005709$ $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC:246.8556011544573$ ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:450.6457967736724 $ARIMA(0, 1, 0) \times (0, 1, 0, 12) 12 - AIC:461.1928031330283$ $ARIMA(0, 1, 0) \times (0, 1, 1, 12) 12 - AIC:214.17303780540846$ $ARIMA(0, 1, 0) \times (1, 0, 0, 12) 12 - AIC:470.3859447084332$ $ARIMA(0, 1, 0) \times (1, 0, 1, 12) 12 - AIC:452.2409062206534$ $ARIMA(0, 1, 0) \times (1, 1, 0, 12) 12 - AIC:234.29254626548172$ ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:426.8931345319276 ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:433.53384035056484 $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:186.88079986495342$ $ARIMA(0, 1, 1) \times (1, 0, 0, 12) 12 - AIC:467.73011390437256$ $ARIMA(0, 1, 1) \times (1, 0, 1, 12) 12 - AIC:428.0375276889837$ $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC: 228.21758668577272$ ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:472.4266158842965 $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:473.39068841014046$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:227.74140951588006$ $ARIMA(1, 0, 0) \times (1, 0, 0, 12) 12 - AIC:473.60405277153205$ $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC:474.0884500709542$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:227.3698154964428$

 $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:446.756877234146$ ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:454.7270733701521 ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:210.25238377195882 $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:469.7563036919878$ $ARIMA(1, 0, 1) \times (1, 0, 1, 12) 12 - AIC:448.75085602227074$ $ARIMA(1, 0, 1) \times (1, 1, 0, 12) 12 - AIC:228.03425758485594$ ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:208.79044864850636 $\mathtt{ARIMA}\,(1,\ 1,\ 0)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC:}705.1820583733651$ ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:212.37186507113353 ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:453.0345104971078 ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:454.10060819236816 ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:212.52846271766228 $ARIMA(1, 1, 0) \times (1, 1, 1, 12) 12 - AIC:213.46970463546938$ $ARIMA(1, 1, 1) \times (0, 0, 12) 12 - AIC:679.3390698276093$ ARIMA(1, 1, 1)x(0, 1, 1, 12)12 - AIC:187.76130984511948 $ARIMA(1, 1, 1) \times (1, 0, 0, 12) 12 - AIC:449.4252469942211$ $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:429.66626308983996$ $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC:209.46733399213454$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:189.47440287714772 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(0,1,1), seasonal order=(1, 1, 1, 12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025 ma.L1 0.554 -1.771 0.076 -2.066 0.327 0.819 0.413 -0.373-0.9806 0.104 0.819 -2.466 -0.373 ar.S.L12 0.2675 0.327 0.413 0.234 -0.5763 -1.034 -0.118 ma.S.L12 0.014 1.668e+07 2.86e-08 5.83e+14 0.000 sigma2 1.67e+07 1.67e+07 pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2016':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred ci.iloc[:, 0], pred ci.iloc[:, 1], color='k', alpha=.2) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title ('The amounf of Sales of Beta blocking agents', fontsize=20) plt.legend() plt.show() The amounf of Sales of Beta blocking agents observed 105000 One-step ahead Forecast 100000 95000 ales in Quantities 90000 85000 80000 S 75000 70000 65000 Jan 2017 Jan 2018 Jan 2019 Date y forecasted = pred.predicted mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 49805633.53 The Root Mean Squared Error of our forecasts is 7057.31 In [324... pred_uc = results.get_forecast(steps=100) pred_ci = pred_uc.conf_int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Prices', fontsize=15) ax.set_title('Sales of beta blocking agents',fontsize=20) plt.legend() plt.show() Sales of beta blocking agents Observed Forecast 110000 100000 70000 2019 2017 2021 2023 2025 2027 Date C03 Diuretic drugs df =df.loc[(df["ATCShort"]=="C03")] df = df.sort values('Delivery date') df = df.groupby('Delivery date')['Units'].sum().reset index() df = df.set index('Delivery date') y = df['Units'].resample('MS').mean() In [334... import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param seasonal in seasonal pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param seasonal, results.aic)) except: continue $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:413.29494916845033$ ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:198.83621505109795 $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:433.51254731197173$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:416.43842215040246$ $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:216.190040133187$ $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:200.74934284785428$ ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:393.0915829915456 ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:181.68812335581964 ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:549.2044149267457ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:501.5676612008901 $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC:213.91965413124294$ ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:402.04129408082207 ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:185.64450933436035 $ARIMA(0, 1, 0) \times (1, 0, 0, 12) 12 - AIC:407.303395901243$ $ARIMA(0, 1, 0) \times (1, 0, 1, 12) 12 - AIC:391.99722390662043$ $ARIMA(0, 1, 0) \times (1, 1, 0, 12) 12 - AIC:203.82581222355978$ $ARIMA(0, 1, 0) \times (1, 1, 1, 12) 12 - AIC:187.2789225860381$ ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:374.483502319514 $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:167.04650770905948$ ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:401.6726033101406 $ARIMA(0, 1, 1) \times (1, 0, 1, 12) 12 - AIC:369.0153705357959$ $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC:202.05732117294158$ $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:164.74006624374232$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:412.5162959991798$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:200.66638721604014$ $ARIMA(1, 0, 0) \times (1, 0, 0, 12) 12 - AIC:410.68835281067095$ $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC: 412.23946887704665$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(1,\ 1,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!201.1370181075443$ $ARIMA(1, 0, 0) \times (1, 1, 1, 12) 12 - AIC:202.5751685830648$ $ARIMA(1, 0, 1) \times (0, 0, 12) 12 - AIC:609.7074490674499$ $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:388.4927360432203$ ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:395.0899552194534 $ARIMA(1, 0, 1) \times (0, 1, 1, 12) 12 - AIC:183.18689854143673$ $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:406.1726190262535$ $ARIMA(1, 0, 1) \times (1, 0, 1, 12) 12 - AIC:390.31359070989157$ $\mathtt{ARIMA}\,(1,\ 0,\ 1)\,\mathtt{x}\,(1,\ 1,\ 0,\ 12)\,12\ -\ \mathtt{AIC:}198.81224234394838$ $\texttt{ARIMA(1, 0, 1)} \times (1, 1, 1, 12) \\ \texttt{12 - AIC:183.6749404043551}$ $ARIMA(1, 1, 0) \times (0, 0, 12) 12 - AIC:614.2154297716693$ ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:391.9111844011455ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:403.772099012935 ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:187.61298602790924 $ARIMA(1, 1, 0) \times (1, 0, 0, 12) 12 - AIC:392.637097951678$ $ARIMA(1, 1, 0) \times (1, 0, 1, 12) 12 - AIC:393.9035899380464$ $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:187.84133663477695$ $ARIMA(1, 1, 0) \times (1, 1, 1, 12) 12 - AIC:189.36321435029024$ $ARIMA(1, 1, 1) \times (0, 0, 0, 12) 12 - AIC:589.2197187895413$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:369.07898396799953$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:388.90338055691234$ ARIMA(1, 1, 1)x(0, 1, 1, 12)12 - AIC:170.31220054220464 ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:386.1317846126396 $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:370.96334451237215$ $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC:186.34285944785094$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:170.40613693946813 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(0,0,1), seasonal order=(0,1,1,12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) z P>|z| [0.025 0.975]coef std err 1.1567 0.232 4.988 0.000 0.702 0.1931 0.237 0.851 0.205 -0.252 ma.S.L12 0.1931 0.227 0.851 0.395 0.638 0.322 -2.37e+06 sigma2 2.42e+06 2.45e+06 7.21e+06 0.990 pred = results.get_prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2016':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred ci.iloc[:, 0], pred ci.iloc[:, 1], color='k', alpha=.2) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title('The amount of Sales of Diuretic drugs',fontsize=20) plt.legend() plt.show() The amount of Sales of Diuretic drugs observed 26000 One-step ahead Forecast 24000 22000 Sales in Quantii 20000 18000 16000 14000 Jul Jan 2018 Jul Jan 2019 Jul Jan 2017 Date In [338... y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 1914298.97 The Root Mean Squared Error of our forecasts is 1383.58 In [339... pred uc = results.get forecast(steps=100) pred ci = pred uc.conf int() = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred uc.predicted mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title('The amount of Sales of Diuretic drugs',fontsize=20) plt.legend() plt.show() The amount of Sales of Diuretic drugs 23000 22000 21000 Sales in Quantities 19000 18000 17000 2021 2027 2017 2019 2023 Date A10 Drugs used in diabetes In [344... df =df.loc[(df["ATCShort"]=="A10")] In [345... df = df.sort values('Delivery date') In [346... df = df.groupby('Delivery date')['Units'].count().reset index() In [347... df = df.set index('Delivery date') In [348... y = df['Units'].resample('MS').mean() In [349... import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: (0, 0, 1) x (0, 0, 1, 12) SARIMAX: (0, 0, 1) x (0, 1, 0, 12) SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param_seasonal in seasonal_pdq: try: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal_order=param_seasonal, enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic)) except: continue $ARIMA(0, 0, 0) \times (0, 0, 12) 12 - AIC:595.3166395819993$ $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:375.3642262037889$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:278.7045915889768$ $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC:129.10159742844596$ $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:287.6586652973614$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:276.9969746980637$ $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:140.78081903319753$ ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:129.98818294230415 ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:537.1083462288468 $ARIMA(0, 0, 1) \times (0, 0, 1, 12) 12 - AIC:335.3420725080202$ $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC: 265.00605269363604$ $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC:119.45662258544519$ $\mathtt{ARIMA}\,(0,\ 0,\ 1)\, \mathtt{x}\,(1,\ 0,\ 0,\ 12)\, 12\ -\ \mathtt{AIC}\!:\!287.9056120513402$ $ARIMA(0, 0, 1) \times (1, 0, 1, 12) 12 - AIC: 264.5679085822349$ ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:142.72012738624628ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:120.7599438337709 ARIMA(0, 1, 0)x(0, 0, 12)12 - AIC:409.52674822024994ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:263.4416432193273 $ARIMA(0, 1, 0) \times (0, 1, 0, 12) 12 - AIC:271.0561141690568$ $ARIMA(0, 1, 0) \times (0, 1, 1, 12) 12 - AIC:126.71349340060858$ $ARIMA(0, 1, 0) \times (1, 0, 0, 12) 12 - AIC:272.84222025662604$ $ARIMA(0, 1, 0) \times (1, 0, 1, 12) 12 - AIC: 263.71804681092124$ $ARIMA(0, 1, 0) \times (1, 1, 0, 12) 12 - AIC:137.9840495272948$ ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:128.70679791380942 ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:393.6943827977473 $ARIMA(0, 1, 1) \times (0, 0, 1, 12) 12 - AIC: 242.96561138049708$ ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:255.53676619094645 ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:106.54716832823716 $\mathtt{ARIMA}\,(0,\ 1,\ 1)\, \mathtt{x}\,(1,\ 0,\ 0,\ 12)\, 12\,\,-\,\,\mathtt{AIC}\!:\!264.4592820421401$ ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:108.54577666380314 $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:422.9173425823452$ $ARIMA(1, 0, 0) \times (0, 0, 1, 12) 12 - AIC:276.3974749990728$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC: 276.22564622594325$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:131.04133945236077$ $ARIMA(1, 0, 0) \times (1, 0, 0, 12) 12 - AIC:274.8276026416589$ $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC:276.4473782964738$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:131.02360182428973$ ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:131.8804488527305ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:403.9220571772916 $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:253.0586239941553$ $ARIMA(1, 0, 1) \times (0, 1, 0, 12) 12 - AIC: 266.61699359160207$ ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:121.39493757649406 $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:264.55058932935736$ $ARIMA(1, 0, 1) \times (1, 0, 1, 12) 12 - AIC:253.5952334779744$ ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:132.64490125538083ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:122.75980353819351 ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:410.74493029156 $ARIMA(1, 1, 0) \times (0, 0, 1, 12) 12 - AIC: 263.2253722055204$ $ARIMA(1, 1, 0) \times (0, 1, 0, 12) 12 - AIC:270.68969569201295$ $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:124.65130013158739$ $ARIMA(1, 1, 0) \times (1, 0, 0, 12) 12 - AIC:262.23832064654715$ ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:263.03136784922435 $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC: 124.65167166224192$ ARIMA(1, 1, 0)x(1, 1, 1, 12)12 - AIC:126.6296918887673 $ARIMA(1, 1, 1) \times (0, 0, 0, 12) 12 - AIC:392.93619587926037$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:244.8675274082683$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:256.18586986809476$ $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:105.53219262047611$ $ARIMA(1, 1, 1) \times (1, 0, 0, 12) 12 - AIC:255.99790853669452$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:107.52849254550925 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1), $seasonal_order=(1,1,1,12)$, enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025 0.975] ar.L1 -0.4786 0.311 -1.539 0.124 -1.088 0.131 ma.L1 -1.0000 0.923 -1.084 0.279 -2.809 0.200 0.308

 -0.0009
 0.158
 -0.006
 0.995
 -0.310

 -0.0907
 0.246
 -0.369
 0.712
 -0.573

 2797.9656
 0.000
 8.48e+06
 0.000
 2797.965

 ar.S.L12 ma.S.L12 -0.0907 sigma2 2797.9656 0.391 2797.966 pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2016':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 4)) ax.fill between (pred ci.index, pred ci.iloc[:, 0], pred ci.iloc[:, 1], color='k', alpha=.2) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title('The amount of Sales of medicines used in Diabets',fontsize=20) plt.legend() plt.show() The amount of Sales of medicines used in Diabets 1400 observed One-step ahead Forecast Sales in Quantities 1300 1200 1100 1000 Jul Jul Jan 2017 Date y forecasted = pred.predicted mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 3422.12 The Root Mean Squared Error of our forecasts is 58.5 pred uc = results.get forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred uc.predicted mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title('The amount of Sales of medicines used in Diabets',fontsize=20) plt.legend() plt.show() The amount of Sales of medicines used in Diabets Observed 1700 1600 1500 Sales in Quantities 1200 1000 2017 2019 2021 2025 2027 2023 Date **Specific location Forecasting Brussels** df=df1 df =df.loc[(df["Pharmacy Postcode (2)"] >=10) & (df["Pharmacy Postcode (2)"] <=12)] df = df.sort_values('Delivery date') df = df.groupby('Delivery date')['Units'].sum().reset_index() df = df.set index('Delivery date') In [364... y= df['Units'].resample('MS').mean() import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: (0, 1, 0) x (0, 1, 1, 12) SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param seasonal in seasonal pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic)) except: continue $ARIMA(0, 0, 0) \times (0, 0, 12) 12 - AIC:936.609269941616$ $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:596.3350103543928$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:513.9417984384806$ $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC: 246.24601732400745$ $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:524.7381282087367$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:506.8158855071904$ $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:263.1502353669254$ $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:240.46096898838636$ ARIMA(0, 0, 1)x(0, 0, 12)12 - AIC:887.7972151556331 ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:574.3883877329573 $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC:490.4632266732547$ $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC: 225.51880767226015$ $ARIMA(0, 0, 1) \times (1, 0, 0, 12) 12 - AIC:622.9602293027407$ $ARIMA(0, 0, 1) \times (1, 0, 1, 12) 12 - AIC:573.5379569355531$ $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC: 266.6652884461187$ $ARIMA(0, 0, 1) \times (1, 1, 1, 12) 12 - AIC: 223.28176639875585$ ARIMA(0, 1, 0)x(0, 0, 12)12 - AIC:747.351831975033ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:481.0599875782106ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:492.7843906309351 $ARIMA(0, 1, 0) \times (0, 1, 1, 12) 12 - AIC:223.14494440494929$ ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:501.1099198967584 $ARIMA(0, 1, 0) \times (1, 0, 1, 12) 12 - AIC:482.88119676198596$ $ARIMA(0, 1, 0) \times (1, 1, 0, 12) 12 - AIC:248.2142199427712$ ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:225.12666831201955 ARIMA(0, 1, 1)x(0, 0, 12)12 - AIC:721.945964468431ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:453.1340526565759ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:460.61171157929147 $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:199.2442142388339$ $\mathtt{ARIMA}\,(\mathtt{0,\ 1,\ 1})\,\mathtt{x}\,(\mathtt{1,\ 0,\ 0,\ 12})\,\mathtt{12\ -\ AIC:}\,\mathtt{494.8964638009092}$ ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:453.9969247707715 $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC:242.15644997271372$ $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:201.15005100367802$ $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:772.377872068088$ $ARIMA(1, 0, 0) \times (0, 0, 1, 12) 12 - AIC:504.312074709814$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:510.1117810548759$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:246.38949151470553$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(1,\ 0,\ 0,\ 12)\,12\ -\ \mathtt{AIC:}504.27114277552243$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(1,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC:}506.0908536127075$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:243.94857222156017$ $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:474.00538970410304$ ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:477.5933143453971ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:221.10890716485852ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:496.75930770945575ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:475.6660453929229 $ARIMA(1, 0, 1) \times (1, 1, 0, 12) 12 - AIC:242.23839696645697$ ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:479.44044704826536 ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:487.36579301069264 $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:222.10314337553055$ $ARIMA(1, 1, 0) \times (1, 0, 0, 12) 12 - AIC:478.77684152127813$ ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:480.77595166450294 $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:222.25209782003267$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:454.7440210587142$ ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:461.1759735218664 $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:198.89141765447715$ $ARIMA(1, 1, 1) \times (1, 0, 0, 12) 12 - AIC:474.0765212004487$ $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:454.89131786406256$ $\mathtt{ARIMA}\,(1,\ 1,\ 1)\,\mathtt{x}\,(1,\ 1,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!220.4444545739845$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:200.65231816793207 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1), seasonal order=(0, 1, 1, 12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) ______ coef std err z P>|z| [0.025 0.975] ar.Ll -0.6163 1.156 -0.533 0.594 -2.881 1.649 ma.Ll -0.8773 0.122 -7.183 0.000 -1.117 -0.638 ma.S.Ll2 0.1937 0.330 0.587 0.557 -0.453 0.841 sigma2 1.763e+08 2.11e-09 8.37e+16 0.000 1.76e+08 1.76e+08 ______ pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2013':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantities ',fontsize=15) ax.set title('The amount of Sales in Brussels area',fontsize=20) plt.legend() plt.show() The amount of Sales in Brussels area observed 220000 One-step ahead Forecast 200000 Sales in Quantities 180000 160000 140000 120000 Jul Jul Jan 2017 Jul Date y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 230157023.29 The Root Mean Squared Error of our forecasts is 15170.93 pred_uc = results.get_forecast(steps=100) pred_ci = pred_uc.conf_int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantities ',fontsize=15) ax.set_title('The amount of Sales in Brussels area',fontsize=20) plt.legend() plt.show() The amount of Sales in Brussels area 275000 250000 225000 Sales in Quantities 200000 175000 150000 125000 2019 **Antwerp** df=df1 df =df.loc[(df["Pharmacy Postcode (2)"] >=20) & (df["Pharmacy Postcode (2)"] <=29)]</pre> df = df.sort values('Delivery date') In [374.. df = df.groupby('Delivery date')['Units'].sum().reset_index() df = df.set index('Delivery date') y= df['Units'].resample('MS').mean() import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: (0, 0, 1) x (0, 1, 0, 12) SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param seasonal in seasonal pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param seasonal, results.aic)) except: continue $ARIMA(0, 0, 0) \times (0, 0, 0, 12) 12 - AIC:987.9327666777162$ $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:628.9054542433828$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:536.6746493538152$ $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC:256.2889215834581$ $\mathtt{ARIMA}\,(\mathtt{0,\ 0,\ 0})\,\mathtt{x}\,(\mathtt{1,\ 0,\ 0,\ 12})\,\mathtt{12\ -\ AIC:}561.9448977522754$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:538.5645721469591$ ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:936.9424918910443

 $ARIMA(0, 1, 0) \times (1, 1, 0, 12) 12 - AIC:263.67231499457785$ $ARIMA(0, 1, 0) \times (1, 1, 1, 12) 12 - AIC:239.62320554302983$ $\mathtt{ARIMA}\,(0,\ 1,\ 1)\,\mathtt{x}\,(0,\ 0,\ 0,\ 12)\,12\ -\ \mathtt{AIC:}774.8224200702249$ ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:211.67457591287248 ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:529.5043854538135ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:482.37433369957336 ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:257.95589924166086 $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:213.47795557879923$ $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:822.3474826826427$ ARIMA(1, 0, 0)x(0, 1, 1, 12)12 - AIC:258.0792898401229 ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:534.5212656269248 $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC:535.2188207652418$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:257.91697494765134$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(1,\ 1,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!257.5307068144384$ $ARIMA(1, 0, 1) \times (0, 0, 12) 12 - AIC:799.3809969833626$ ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:237.28176197544235 $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:533.6310940302528$ $ARIMA(1, 0, 1) \times (1, 0, 1, 12) 12 - AIC:506.4423499857549$ ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:258.9797259726727 $ARIMA(1, 0, 1) \times (1, 1, 1, 12) 12 - AIC:233.11745836321595$ ARIMA(1, 1, 0)x(0, 0, 12)12 - AIC:799.1185956209415ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:239.4368368946162 ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:511.72313296728237 $ARIMA(1, 1, 0) \times (1, 0, 1, 12) 12 - AIC:512.771685228397$ $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:239.67795425880013$ $\mathtt{ARIMA}\,(1,\ 1,\ 0)\,\mathtt{x}\,(1,\ 1,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!240.05357741174936$ $\mathtt{ARIMA}\,(1,\ 1,\ 1)\,\mathtt{x}\,(0,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!762.9892453308491$ ARIMA(1, 1, 1)x(0, 1, 1, 12)12 - AIC:212.85583963510814 ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:507.6668530597765 $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:484.30869061835455$ $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC: 236.83919649587452$ ARIMA(1, 1, 1)x(1, 1, 12)12 - AIC:214.72253856904229 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1), seasonal order=(0, 1, 1, 12), enforce_stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025 0.975] ______
 ar.L1
 -0.3816
 0.732
 -0.521
 0.602
 -1.817
 1.054

 ma.L1
 -0.6502
 0.321
 -2.026
 0.043
 -1.279
 -0.021

 ma.S.L12
 -0.4161
 0.583
 -0.713
 0.476
 -1.559
 0.727

 sigma2
 7.796e+08
 2.54e-10
 3.06e+18
 0.000
 7.8e+08
 7.8e+08
 pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2013':].plot(label='observed') pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred ci.iloc[:, 0], pred ci.iloc[:, 1], color='k', alpha=.2) ax.set xlabel('Date', fontsize=15) ax.set ylabel('Sales in Quantities', fontsize=15) ax.set title('The amount of Sales in Antwerp area',fontsize=20) plt.legend() plt.show() The amount of Sales in Antwerp area observed One-step ahead Forecast 450000 Sales in Quantities 400000 350000 300000 250000 Jul Jul Date y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 1977781571.65 The Root Mean Squared Error of our forecasts is 44472.26 pred_uc = results.get_forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantities', fontsize=15) ax.set_title('The amount of Sales in Antwerp area', fontsize=20) plt.legend() plt.show() The amount of Sales in Antwerp area Observed Forecast 600000 Sales in Quantiti 500000 400000 2021 2019 2023 2025 2027 Luxembourg Area df=df1 In [384... df =df.loc[(df["Pharmacy Postcode (2)"] >=66) & (df["Pharmacy Postcode (2)"] <=69)]</pre> df = df.sort values('Delivery date') df = df.groupby('Delivery date')['Units'].count().reset index() df = df.set index('Delivery date') y= df['Units'].resample('MS').mean() In [389... import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: (0, 1, 0) x (0, 1, 1, 12) SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param_seasonal in seasonal_pdq: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic)) continue ARIMA(0, 0, 0)x(0, 0, 12)12 - AIC:492.85359235522037 $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:312.88461675307315$ ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:216.39880006994744 $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC:94.45584037543654$ $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC: 225.30003597410087$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:211.9394914759077$ $ARIMA(0, 0, 0) \times (1, 1, 0, 12) 12 - AIC:101.94805955694162$ ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:96.45590779616711ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:439.6066960294487ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:277.22235985466875 $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC:200.89057576028603$ $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC:81.84434389502306$ $ARIMA(0, 0, 1) \times (1, 0, 0, 12) 12 - AIC:223.03231167225763$ $ARIMA(0, 0, 1) \times (1, 0, 1, 12) 12 - AIC:198.20431607281134$ ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:103.92983091406686 $ARIMA(0, 0, 1) \times (1, 1, 1, 12) 12 - AIC:87.11188967626205$ ARIMA(0, 1, 0)x(0, 0, 12)12 - AIC:334.0998746268496ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:211.15757421021286 ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:206.80447208884746 $ARIMA(0, 1, 0) \times (0, 1, 1, 12) 12 - AIC: 93.69479530287465$ $ARIMA(0, 1, 0) \times (1, 0, 0, 12) 12 - AIC:210.6211613650527$ $ARIMA(0, 1, 0) \times (1, 0, 1, 12) 12 - AIC: 202.96746170751769$ ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:102.27006167993706ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:95.68953966645822 $ARIMA(0, 1, 1) \times (0, 0, 12) 12 - AIC:325.58708804483297$ $ARIMA(0, 1, 1) \times (0, 0, 1, 12) 12 - AIC:199.3966978599076$ ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:194.60646448284913 $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:84.76284795496156$ $ARIMA(0, 1, 1) \times (1, 0, 0, 12) 12 - AIC:206.9775508704848$ $ARIMA(0, 1, 1) \times (1, 0, 1, 12) 12 - AIC:191.3732125004161$ $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC:99.1264183114885$ ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:84.98192919031676 $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:345.83794554580476$ ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:221.94581909817003 $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:210.49212815917238$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC: 96.44814108287478$ $ARIMA(1, 0, 0) \times (1, 0, 0, 12) 12 - AIC:212.55990898962082$ $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(1,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!213.1106913535442$ $ARIMA(1, 0, 0) \times (1, 1, 0, 12) 12 - AIC:96.4481145591235$ ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:98.4410112832084 $ARIMA(1, 0, 1) \times (0, 0, 12) 12 - AIC:337.16271514295374$ ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:208.66498446820944 $ARIMA(1, 0, 1) \times (0, 1, 0, 12) 12 - AIC:200.23838053714476$ $ARIMA(1, 0, 1) \times (0, 1, 1, 12) 12 - AIC:83.70556928654501$ $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:205.14067600611352$ $ARIMA(1, 0, 1) \times (1, 0, 1, 12) 12 - AIC:199.4838644253359$ $ARIMA(1, 0, 1) \times (1, 1, 0, 12) 12 - AIC:97.7458589693725$ $ARIMA(1, 0, 1) \times (1, 1, 1, 12) 12 - AIC:87.76772471876306$ ARIMA(1, 1, 0)x(0, 0, 12)12 - AIC:335.64900514082933 ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:211.56131961536684 $ARIMA(1, 1, 0) \times (0, 1, 0, 12) 12 - AIC:204.64048115103134$ $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC: 93.42885722234678$ $ARIMA(1, 1, 0) \times (1, 0, 0, 12) 12 - AIC:200.23984834005734$ $ARIMA(1, 1, 0) \times (1, 0, 1, 12) 12 - AIC:201.54880899281085$ $\mathtt{ARIMA}\,(1,\ 1,\ 0)\, \mathtt{x}\,(1,\ 1,\ 0,\ 12)\, 12\ -\ \mathtt{AIC:}93.42915182992924$ ARIMA(1, 1, 0)x(1, 1, 1, 12)12 - AIC:95.40174921300891ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:320.1131786490133 $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:199.7623405845217$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:196.0026513229144$ $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:86.56649832112491$ $ARIMA(1, 1, 1) \times (1, 0, 0, 12) 12 - AIC:200.7765253524011$ $ARIMA(1, 1, 1) \times (1, 0, 1, 12) 12 - AIC:192.4568847522556$ $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC:92.72924757293033$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:86.94300342036125 import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(0, 0, 1), seasonal order=(0, 1, 1, 12), enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print(results.summary().tables[1]) std err P>|z| [0.025 coef -0.5084 0.193 -2.629 0.009 -0.888 -0.000 -1.0000 4757.255 1.000 -9325.048 ma.S.L12 1.000 -7.11e+05 pred = results.get_prediction(start=pd.to_datetime('2018-12-01'), dynamic=False) pred_ci = pred.conf_int() ax = y['2013':].plot(label='observed') pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill_between(pred_ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set_title('The amount of Sales in Luxembourg area',fontsize=20) plt.legend() plt.show() The amount of Sales in Luxembourg area 360 observed One-step ahead Forecast 340 320 300 Sales in Quantity 280 260 240 220 200 Jul Date y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 328.95 The Root Mean Squared Error of our forecasts is 18.14 In [394... pred_uc = results.get_forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set title('The amount of Sales in Luxembourg area', fontsize=20) plt.legend() plt.show() The amount of Sales in Luxembourg area Observed Forecast 320 300 Sales in Quantity 240 200 2019 2025 2027 2021 2023 2017 Date Flemish Region df=df1 df =df.loc[(df["Pharmacy Postcode (2)"] >=30) & (df["Pharmacy Postcode (2)"] <=34)]</pre> df = df.sort_values('Delivery date') In [254... df = df.groupby('Delivery date')['Units'].count().reset_index() df = df.set index('Delivery date') y= df['Units'].resample('MS').mean() import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param seasonal in seasonal pdq: try: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal_order=param_seasonal, enforce_stationarity=False, enforce_invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic)) except: continue ARIMA(0, 0, 0)x(0, 0, 12)12 - AIC:618.0467478488881 $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:383.61793411719333$ $ARIMA(0, 0, 0) \times (0, 1, 0, 12) 12 - AIC:308.20244665590764$ $ARIMA(0, 0, 0) \times (0, 1, 1, 12) 12 - AIC:152.76189398629234$ $ARIMA(0, 0, 0) \times (1, 0, 0, 12) 12 - AIC:317.25912955391726$ $ARIMA(0, 0, 0) \times (1, 0, 1, 12) 12 - AIC:300.62310740221625$ ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:558.6698978946073 ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:345.45691703085305 $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC:294.33410366647564$ $ARIMA(0, 0, 1) \times (0, 1, 1, 12) 12 - AIC:139.71605729907606$ $\mathtt{ARIMA}\,(\mathtt{0,\ 0,\ 1})\,\mathtt{x}\,(\mathtt{1,\ 0,\ 0,\ 12})\,\mathtt{12\ -\ AIC:316.9938793879542}$ $ARIMA(0, 0, 1) \times (1, 0, 1, 12) 12 - AIC:288.91959156507266$ $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC:158.92746024013178$ ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:133.3717713283507ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:431.00193167463ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:284.1404468497115 ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:290.4947800796603 $ARIMA(0, 1, 0) \times (0, 1, 1, 12) 12 - AIC:133.65589360963673$ ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:295.39092961384006ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:283.1734135242368ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:420.12176885408263 ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:273.0952364445121 $ARIMA(0, 1, 1) \times (0, 1, 0, 12) 12 - AIC: 272.36563896604946$ $ARIMA(0, 1, 1) \times (0, 1, 1, 12) 12 - AIC:118.60571971407249$ $\mathtt{ARIMA}\,(\mathtt{0,\ 1,\ 1})\,\mathtt{x}\,(\mathtt{1,\ 0,\ 0,\ 12})\,\mathtt{12}\,\mathtt{-AIC:293.4177029979912}$ $ARIMA(0, 1, 1) \times (1, 0, 1, 12) 12 - AIC:267.6416238938777$ $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC:144.00375139171186$ $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:119.24181915801232$ $ARIMA(1, 0, 0) \times (0, 0, 12) 12 - AIC:445.13178946751236$ ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:297.29452317988495 $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:302.7201678024095$ $ARIMA(1, 0, 0) \times (0, 1, 1, 12) 12 - AIC:148.0228346582823$ $ARIMA(1, 0, 0) \times (1, 0, 0, 12) 12 - AIC:296.73517976343027$ $ARIMA(1, 0, 0) \times (1, 0, 1, 12) 12 - AIC:295.9168990575221$ ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:434.1854101069826 ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:286.41292385406007 $ARIMA(1, 0, 1) \times (0, 1, 0, 12) 12 - AIC:285.1609878532184$ $ARIMA(1, 0, 1) \times (0, 1, 1, 12) 12 - AIC:131.8743327257215$ $ARIMA(1, 0, 1) \times (1, 0, 0, 12) 12 - AIC:293.5499430529538$ $\mathtt{ARIMA}\,(1,\ 0,\ 1)\,\mathtt{x}\,(1,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!279.72489275583126$ $\mathtt{ARIMA}\,(1,\ 0,\ 1)\,\mathtt{x}\,(1,\ 1,\ 0,\ 12)\,12\ -\ \mathtt{AIC}\!:\!146.7149937310033$ ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:284.8494728591838 ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:282.7279507835501 $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:131.93213616461588$ $ARIMA(1, 1, 0) \times (1, 0, 0, 12) 12 - AIC:280.42975706952785$ ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:280.86644806891866ARIMA(1, 1, 0) \times (1, 1, 0, 12)12 - AIC:131.0108288450655 ARIMA(1, 1, 0) \times (1, 1, 1, 12)12 - AIC:132.98075812886304 ARIMA(1, 1, 1) \times (0, 0, 0, 12)12 - AIC:419.7608089571791 $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:274.4739215674124$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:272.0274139749728$ $ARIMA(1, 1, 1) \times (0, 1, 1, 12) 12 - AIC:118.58650137364342$ $ARIMA(1, 1, 1) \times (1, 0, 0, 12) 12 - AIC:282.75746656394796$ $\mathtt{ARIMA}\,(1,\ 1,\ 1)\,\mathtt{x}\,(1,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!269.616822820223$ $ARIMA(1, 1, 1) \times (1, 1, 0, 12) 12 - AIC:132.32511341186915$ ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:119.9625231246365import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1), seasonal order=(0, 1, 1, 12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) coef std err z P>|z| [0.025]0.975] -0.6240 0.658 -0.948 0.343 -1.914 -0.3422-0.398 0.691 0.860 -2.027 ma.L1 -1.0011 -1103.926 ma.S.L12 562.727 -0.002 0.999 sigma2 8512.8769 4.79e+06 0.002 0.999 -9.38e+06 pred = results.get prediction(start=pd.to datetime('2018-12-01'), dynamic=False) pred ci = pred.conf int() ax = y['2013':].plot(label='observed') pred.predicted mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill between (pred ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set title('The amount of Sales in Flemish area',fontsize=20) plt.legend() plt.show() The amount of Sales in Flemish area observed One-step ahead Forecast 2000 1800 Sales in Quantity 1600 1400 1200 Jul Date y forecasted = pred.predicted mean y truth = y['2018-12-01':]# Compute the mean squared error mse = ((y forecasted - y truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 15729.05 The Root Mean Squared Error of our forecasts is 125.42 pred_uc = results.get_forecast(steps=100) pred ci = pred uc.conf int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set_title('The amount of Sales in Flemish area',fontsize=20) plt.legend() plt.show() The amount of Sales in Flemish area Observed Forecast 1250 Sales in Quantity 750 500 250 2017 2019 2021 Date **Wallon Region** df=df1 df =df.loc[(df["Pharmacy Postcode (2)"] >=13) & (df["Pharmacy Postcode (2)"] <=14)]</pre> df = df.sort values('Delivery date') In [240... df = df.groupby('Delivery date')['Units'].count().reset index() In [241... df = df.set_index('Delivery date') In [243... y= df['Units'].resample('MS').mean() In [244... import itertools p = d = q = range(0, 2)# Generate all different combinations of p, q and q triplets pdq = list(itertools.product(p, d, q)) # Generate all different combinations of seasonal p, q and q triplets $seasonal_pdq = [(x[0], x[1], x[2], 12)$ for x in list(itertools.product(p, d, q))] print('Examples of parameter combinations for Seasonal ARIMA...') print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1])) print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3])) print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4])) Examples of parameter combinations for Seasonal ARIMA... SARIMAX: $(0, 0, 1) \times (0, 0, 1, 12)$ SARIMAX: $(0, 0, 1) \times (0, 1, 0, 12)$ SARIMAX: $(0, 1, 0) \times (0, 1, 1, 12)$ SARIMAX: $(0, 1, 0) \times (1, 0, 0, 12)$ for param in pdq: for param_seasonal in seasonal_pdq: try: mod = sm.tsa.statespace.SARIMAX(y, order=param, seasonal order=param seasonal, enforce stationarity=False, enforce invertibility=False) results = mod.fit() print('ARIMA{}x{}12 - AIC:{}'.format(param, param seasonal, results.aic)) except: continue $ARIMA(0, 0, 0) \times (0, 0, 12) 12 - AIC:516.1988627719727$ $ARIMA(0, 0, 0) \times (0, 0, 1, 12) 12 - AIC:331.1491553757075$ ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:252.15294604286257ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:242.5535833632053 ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:120.35135240636349 $ARIMA(0, 0, 0) \times (1, 1, 1, 12) 12 - AIC:111.23757523426416$ ARIMA(0, 0, 1)x(0, 0, 12)12 - AIC:463.3096126184648 $ARIMA(0, 0, 1) \times (0, 0, 1, 12) 12 - AIC:293.2075842532061$ $ARIMA(0, 0, 1) \times (0, 1, 0, 12) 12 - AIC:231.2857368087276$ ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:231.39896238621657 $ARIMA(0, 0, 1) \times (1, 1, 0, 12) 12 - AIC: 116.20740174479042$ $ARIMA(0, 0, 1) \times (1, 1, 1, 12) 12 - AIC:96.75200412045996$ ARIMA(0, 1, 0)x(0, 0, 12)12 - AIC:355.53528705676075 $\mathtt{ARIMA}\,(0,\ 1,\ 0)\,\mathtt{x}\,(0,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!224.3314559665851$ $ARIMA(0, 1, 0) \times (0, 1, 0, 12) 12 - AIC:231.7499728498748$ ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:224.66329543336326 ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:101.25304686789065 ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:95.1883500696212ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:345.72302595260993 $ARIMA(0, 1, 1) \times (0, 0, 1, 12) 12 - AIC:212.2836205908636$ $ARIMA(0, 1, 1) \times (0, 1, 0, 12) 12 - AIC:223.35905830389603$ ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:213.33359487220878 $ARIMA(0, 1, 1) \times (1, 1, 0, 12) 12 - AIC: 98.94946264502512$ $ARIMA(0, 1, 1) \times (1, 1, 1, 12) 12 - AIC:85.0275215482829$ ARIMA(1, 0, 0)x(0, 0, 12)12 - AIC:367.898301232108 $\mathtt{ARIMA}\,(1,\ 0,\ 0)\,\mathtt{x}\,(0,\ 0,\ 1,\ 12)\,12\ -\ \mathtt{AIC}\!:\!235.48255876494173$ $ARIMA(1, 0, 0) \times (0, 1, 0, 12) 12 - AIC:239.9269933880727$ ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:235.7109330393729 ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:101.46497747993659 $ARIMA(1, 0, 0) \times (1, 1, 1, 12) 12 - AIC:103.1764109142354$ ARIMA(1, 0, 1)x(0, 0, 12)12 - AIC:354.79766640189433 $ARIMA(1, 0, 1) \times (0, 0, 1, 12) 12 - AIC:220.36979558779973$ $ARIMA(1, 0, 1) \times (0, 1, 0, 12) 12 - AIC:231.873710283378$ ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:222.0002069940624 ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:98.87388910001862 ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:93.02079294496627 ARIMA(1, 1, 0)x(0, 0, 12)12 - AIC:357.12297694801043 $ARIMA(1, 1, 0) \times (0, 0, 1, 12) 12 - AIC:225.79824643476277$ $ARIMA(1, 1, 0) \times (0, 1, 0, 12) 12 - AIC:233.20354858800076$ $ARIMA(1, 1, 0) \times (0, 1, 1, 12) 12 - AIC:94.03613838284497$ ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:229.65356630941596 ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:226.3354852220287 $ARIMA(1, 1, 0) \times (1, 1, 0, 12) 12 - AIC:94.03610995085715$ $ARIMA(1, 1, 0) \times (1, 1, 1, 12) 12 - AIC:96.03601958268845$ $ARIMA(1, 1, 1) \times (0, 0, 0, 12) 12 - AIC:343.35446343455584$ $ARIMA(1, 1, 1) \times (0, 0, 1, 12) 12 - AIC:212.8334550721066$ $ARIMA(1, 1, 1) \times (0, 1, 0, 12) 12 - AIC:220.9211640458189$ ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:212.7587207604334 ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:92.87142495444078 $ARIMA(1, 1, 1) \times (1, 1, 1, 12) 12 - AIC:86.88520787615914$ In [247... import statsmodels.api as sm mod = sm.tsa.statespace.SARIMAX(y, order=(0, 1, 1), seasonal order=(0, 1, 1, 12), enforce stationarity=False, enforce invertibility=False) results = mod.fit() print(results.summary().tables[1]) std err P>|z| coef 0.050 0.335 -1.964 -0.001 ma.L1 -0.6579 -1.315ma.S.L12 -0.2066 0.151 -1.366 0.172 0.090 -0.503 sigma2 350.4579 1.683 0.092 -57.691 In [248... pred = results.get_prediction(start=pd.to_datetime('2018-12-01'), dynamic=False) pred_ci = pred.conf_int() ax = y['2013':].plot(label='observed') pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7)) ax.fill_between(pred_ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1], color='k', alpha=.2) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set_title('The amount of Sales in Wallon area',fontsize=20) plt.legend() plt.show() The amount of Sales in Wallon area 500 observed One-step ahead Forecast 450 Sales in Quantity 400 350 300 250 Date In [249... y_forecasted = pred.predicted_mean $y_{truth} = y['2018-12-01':]$ # Compute the mean squared error mse = ((y_forecasted - y_truth) ** 2).mean() print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2))) print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 247.64 The Root Mean Squared Error of our forecasts is 15.74 pred_uc = results.get_forecast(steps=100) pred_ci = pred_uc.conf_int() ax = y.plot(label='Observed', figsize=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_xlabel('Date', fontsize=15) ax.set_ylabel('Sales in Quantity ',fontsize=15) ax.set_title('The amount of Sales in Wallon area', fontsize=20) plt.legend() plt.show() The amount of Sales in Wallon area Observed Forecast 600 Sales in Quantity Date

General Sales in Quantities

df = df.sort values('Delivery date')

df = df.set index('Delivery date')

y= df['Units'].resample('MS').mean()

import itertools

p = d = q = range(0, 2)

df = df.groupby('Delivery date')['Units'].sum().reset index()

Generate all different combinations of p, q and q triplets

df=df1

In [399..

In [400...

### Date Pages Pa	ARIMA(0, 1, ARIMA(1, 0,	1)x(0, 1, 0, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(1, 0, 0, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 1, 12)12 - 1)x(1, 1, 1, 12)12 - 0)x(0, 0, 0, 12)12 - 0)x(0, 1, 0, 12)12 - 0)x(0, 1, 0, 12)12 - 0)x(1, 0, 0, 12)12 - 0)x(1, 0, 1, 12)12 - 0)x(1, 1, 0, 12)12 - 0)x(1, 1, 0, 12)12 - 1)x(0, 0, 0, 12)12 - 1)x(0, 0, 1, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(1, 0, 0, 12)12 - 1)x(1, 0, 0, 12)12 - 1)x(1, 0, 1, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 1, 12)12 - 1)x(1, 1, 1, 12)12 - 0)x(0, 0, 0, 12)12 - 0)x(0, 1, 1, 12)12 - 0)x(0, 1, 1, 12)12 - 0)x(1, 0, 0, 12)12 - 0)x(1, 0, 0, 12)12 - 0)x(1, 0, 1, 12)12 - 0)x(1, 0, 1, 12)12 - 0)x(1, 0, 1, 12)12 - 0)x(1, 1, 0, 12)12 - 0)x(1, 1, 1, 12)12 -	AIC:256.35644529103 AIC:709.73571408253 AIC:653.07185564104 AIC:304.58918886697 AIC:254.06123774659 AIC:866.75704575949 AIC:553.93127047344 AIC:568.80891799830 AIC:259.71218128371 AIC:577.19512719273 AIC:555.56921811604 AIC:284.92198180047 AIC:259.20636458447 AIC:259.20636458447 AIC:842.42975889997 AIC:556.31568391564 AIC:531.81956247764 AIC:229.90928586068 AIC:575.82067281990 AIC:525.07228103251 AIC:279.47738711416 AIC:231.87180469232 AIC:893.76174017252 AIC:584.16790828604 AIC:584.16790828604 AIC:280.83369134472 AIC:581.64616488345 AIC:582.29559060855 AIC:279.93149788281	776 61 165 154 178 22 8 925 86 14 91 774 64 9 07 15 69 61 1054 13 45 174 8 105 67 25 175		
### Section of Symphosis Section	ARIMA(1, 0, ARIMA(1, 1, ARIMA(1)x(0, 0, 0, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(0, 1, 0, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(1, 0, 0, 12)12 - 1)x(1, 0, 1, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 1, 12)12 - 1)x(1, 1, 1, 12)12 - 0)x(0, 0, 0, 12)12 - 0)x(0, 0, 1, 12)12 - 0)x(0, 1, 0, 12)12 - 0)x(0, 1, 1, 12)12 - 0)x(1, 0, 0, 12)12 - 0)x(1, 0, 0, 12)12 - 0)x(1, 1, 1, 12)12 - 0)x(1, 1, 1, 12)12 - 1)x(0, 0, 0, 12)12 - 1)x(0, 0, 1, 12)12 - 1)x(0, 1, 1, 12)12 - 1)x(1, 0, 0, 12)12 - 1)x(1, 0, 1, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 0, 12)12 - 1)x(1, 1, 1, 12)12 -	AIC:869.08081448674 AIC:549.22266312485 AIC:561.02325043964 AIC:254.95457976706 AIC:577.92002111043 AIC:550.90121638426 AIC:281.27138426864 AIC:255.78137987258 AIC:868.37334285652 AIC:555.79627167735 AIC:566.03149648485 AIC:258.39378999344 AIC:556.51572665802 AIC:557.64836284886 AIC:258.68504943049 AIC:259.14844345925 AIC:842.57004523155 AIC:525.08706507520 AIC:533.79659323035 AIC:230.78427196473 AIC:552.63970017615 AIC:557.00415295306 AIC:255.99100760234 AIC:232.42160780804	56 64 87 38 99 55 8883 07 78 25 17 61 33 305 08 195 16 12 98 87 21 99 87 21 66 66		
The amount Sales	print(resul ===================================	ts.summary().tables[coef std err -0.4525 0.872 -0.7543 0.537 -0.1154 0.944 6.185e+09 6.45e-11 lts.get_prediction(s red.conf_int() 3':].plot(label='obs ted_mean.plot(ax=ax, ween(pred_ci.index, oc[:, 0], oc[:, 1], color='k', el('Date', fontsize=2 el('Sales in Quantit e('The amount Sales'	enforce_invertibili 1]) =================================	ty=False) z [0.025	0.975] 1.256 0.299 1.734 5.19e+09 ======	, 7))
Date Date Date Date Date Date Date Date Date Date Date Date Date Date Date Date J. forecasted = pred.predicted_mean y_truth = y('2010-12-01') f. Compute the sean squared error mae = ('y, forecasted = y_truth) ** 21.mean() print('The Mean Squared Error of our forecasts is ()'.format(round(mp.sqrt(mse), 2))) The Mean Squared Error of our forecasts is 468338(465.27) The Root Mean Squared Error of our forecasts is 68479.05 Dred_uc = results.get_forecast(steps=100) pred_oi = pred_uc.comi_int() ax = y.pla(label='Observed', figs!ze=(20, 10),color='r',linewidth=6) pred_uc.predicted_mean.plot(ax=ax, label='Forecast',color='dodgerblue',linewidth=6) ax.set_vlabel('Date',fointsize=20) ax.set_vlabel('Date',fointsize=20) plt.legend() plt.shew() The amount Sales The amount Sales The amount Sales Date Date Date Date Date	le6		The ar	mount Sales		observed One-step ahead Forec
13 - September 10 - S	y_forecaste y_truth = y # Compute t mse = ((y_f print('The print('The print('The print ('The print ('T	d = pred.predicted_m ['2018-12-01':] he mean squared erro orecasted - y_truth) Mean Squared Error o Root Mean Squared Er ared Error of our for an Squared Error of our esults.get_forecast(red_uc.conf_int() (label='Observed', f dicted_mean.plot(ax= el('Date',fontsize=2 el('Sales in Quantit e('The Amount of Sal	ean r ** 2).mean() f our forecasts is { ror of our forecasts recasts is 468938046 ur forecasts is 6847 steps=100) igsize=(20, 10),colo ax, label='Forecast' 0) ies ',fontsize=15)	<pre>Date }'.format(round(mse is {}'.format(round) f9.27 9.05 r='r',linewidth=6)</pre>	e, 2))) id(np.sqrt(mse), 2	
Date	13 - 12 - 12 - 10 - 0.8 -	W/\^	M	\		
	2017	2019	2021	Date	2025	2027