A Big Data and Analysis Project Report on

Pharmaceutical Business Sales Analysis

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELORS OF TECHNOLOGY

In Computers Science and Engineering

2020-2021

By

Aashish Bora 179301005

Abhinay Bhatt 179301009

Anugrah Rastogi- 179301037

MANIPAL UNIVERSITY
JAIPUR

Under the guidance of

Mrs. Bali Devi

Department of Computer Science and Engineering

School of Computing and Information Technology

Manipal University Jaipur

Jaipur, Rajasthan

CERTIFICATE

This is to certify that the project entitled " **Pharmaceutical Business Sales Analysis**" is a bona fide work carried out as part of the course ***Big data Analytics,CS1701***, under my guidance by ***Aashish Bora, Abhinay Bhatt, Anugrah Rastogi***, students of Bachelor Of Technology (B.Tech.) in Computer Science & Engineering (CSE) at the Department of Computer Science & Engineering , Manipal University Jaipur, during the academic semester *VII of year 2020-21.*

Place: Manipal University Jaipur
Date: 23 Jan, 2021 Signature of the Instructor (s)

DECLARATION

I hereby declare that the project entitled "**Pharmaceutical Business Sales Analysis**" submitted as part of the partial course requirements for the course Big Data and Analytics Lab for the award of the degree of Bachelor of Technology in Computer Science & Engineering at Manipal University Jaipur in the semester during academic year 2020-21, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

Signature of the Student:

Place: Manipal University Jaipur

Date: 23 January, 2021

Abstract

This application-based project has the objective of doing sales analysis on a variety of pharmaceutical data using various data visualization tools like matplotlib library in python and tableau. This project also uses various machine learning models to analyze sales data and make future predictions based on the historical data collected. At last, it answers a particular set of questions proposed by the company which would help the company in making decisions regarding their future plans.

INDEX

2. Introduction

Analysing sales data and predicting future sales based on historical data is a very common data science task. One of the problems of pharmaceutical distribution companies (PDCs) is how to control inventory levels in order to prevent costs of excessive inventory and to prevent losing customers due to drug shortage. Consequently, the purpose of this study is to propose a novel method to forecast sales of PDCs.

Actually, PDCs are facing several challenges, including huge amount of inventory, increased competition, and tough regulations that limit advertising. They have to meet their customers' needs by delivering the right amount of medicines to the right place and at the right time. In PDCs, both shortage and surplus of goods can lead to loss of income for these companies. Accordingly, one of the problems in PDCs is how much quantity of each drug should be kept in the inventory.

Problem Definition

There are specific problems that we will be answering in this project –

1. On which day of the week is the second drug (M01AE) most often sold?
2. Which three drugs have the highest sales in January 2015, July 2016, September 2017.

3. Which drug has sold most often on Mondays in 2017?

4. Draw a sales analysis graph of the drug N05C for the year 2017 monthly?

5. Which drug has the least sales in the last 5 years?

6. Which was the drug sold most in the 2 week of march 2017?

7. What medicine sales may be in January 2020? (Our data set only contains information about sales from January 2014 to October 2019)

Importance and benefits of sales analysis

Sales data analytics examines sales reports to evaluate how your company is performing against its goals. Here's why you need to integrate it into your sales operations.

- **Make data-driven decisions instead of relying on gut instinct** - Effective and regular sales analysis unveils how your sales plan is panning out and measures the performance of every individual rep on your team in real-time.
- **Find your most profitable customers -** Your sales reps should spend the majority of their time engaging with high-quality leads that add value to your company. So it's invaluable to identify the characteristics of customers that spend the most money on your products and remain loyal to your company.
- **Get awareness on the market trends** - Are you preparing to launch a new product? Are you planning your future course of actions in terms of stocking inventory, rolling out schemes, and modifying your manufacturing process (if applicable)? A sales analysis report identifies market opportunities and trends to support these efforts.
- **Serve your customers better -** If you can nail down why a deal closed, you can keep your customers happy and forge deeper relationships. Once you understand their needs better and your brand develops goodwill, you can also upsell and cross-sell to these existing customers.
- **Expand your market reach -** Sales data analysis and interpretation will also fetch intel on your non-customers. The information is invaluable for

sharpening your sales pitches and personalizing your future marketing activities to potentially find new customers.

MOTIVATION

Business data analytics has become an important part in a computer engineer's life. And as data is increasing, the demand for ability to extract useful information from the existing data is also increasing. Pharmaceutical Industry has boomed in India due to the various reasons like clinical research, research and development related to various vaccines, etc. Various multinational pharmaceutical corporations are outsourcing their research and development activities to India, giving this industry a rise like never before and due to this the analysis of the sales of these pharmaceutical sales has become very important to stay in the market.

3. Literature Review

Neda Khalil Zadeh et al. (2014), This research introduced a novel method of grouping products to make use of group members' past sales data for each other in sales prediction and increase the accuracy of the prediction. The introduced scheme out performed two previously known methods of (1) ARIMA modeling and (2) building ANNs by just using each drug's own past records. The empirical contribution of this research can be considered from two different point of views. Firstly, are al problem and an actual company (Pakhsh Hejrat Co.) with its genuine sales data were chosen, and sales prediction results were compared with unseen sales data of this company, and acceptable outcomes were achieved. Secondly, 8 managers and experts of 4 main and famous PDCs (Pakhsh Hejrat, Armaghan Daroo, Pakhsh Ferdows, and Daroo Pakhsh) that stand for morethan70%oftotalsalesofmedicinestodrugstoresand hospitals in Iran were interviewed. All managers and experts were attracted by the introduced method of finding group members and sales forecasting. They strongly believed that finding groups of products with similar changes in their past sales records would also help managers to optimize ordering, storage, transport, and delivery of products. Accordingly, empirical supports were attained for the proposed methodology.

Xiaodan Yu et al. (2013), the overall objective of this study is to accurately predict newspaper/magazine sales. This paper presented how SVR method can be used in sale forecasting. The experiment showed that SVR is a superior method in this kind of task. Future research could compare the performance of SVR with the performance of other data mining methods on forecasting sales. Further, more research is needed in selecting appropriate explanatory variables with theoretical bases.

Ibtissem Ben Othman et al. (2009), In this work, they intend to evaluate with a new criterion the classification stability between neural networks and some statistical classifiers based on the optimization Fischer criterion or the maximization of Patrick-Fischer distance orthogonal estimator. The stability comparison is performed by the error rate probability densities estimation which

is valorized by the performed kernel-diffeomorphism Plug-in algorithm. The results obtained show that the statistical approaches are more stable compared to the neural networks.

4. Member Contribution

The group consists of 3 members namely Aashish Bora, Abhinay Bhatt and Anugrah Rastogi.

- Aashish is responsible for the fitting the machine learning model into the dataset and selecting the best model to use for the predictions.
- Abhinay is responsible for doing the data analysis-using tableau and answering the following questions by developing machine learning models.
- Anugrah is responsible for doing the data analysis and visualization using python and cleaning the data for outliners.
- Apart From this, the three members will collaborate with each other to find the answers to the specific problems detailed earlier in the synopsis.

5. Dataset Description

The dataset is taken from Kaggle. It is built from the initial dataset consisted of 600000 transactional data collected in 6 years (period 2014-2019), indicating date and time of sale, pharmaceutical drug brand name and sold quantity, exported from Point-of-Sale system in the individual pharmacy. Selected group of drugs from the dataset (57 drugs) is classified to the following Anatomical Therapeutic Chemical (ATC) Classification System categories:

- M01AB - Anti-inflammatory and antirheumatic products, non-steroids, Acetic acid derivatives and related substances
- M01AE - Anti-inflammatory and antirheumatic products, non-steroids, Propionic acid derivatives

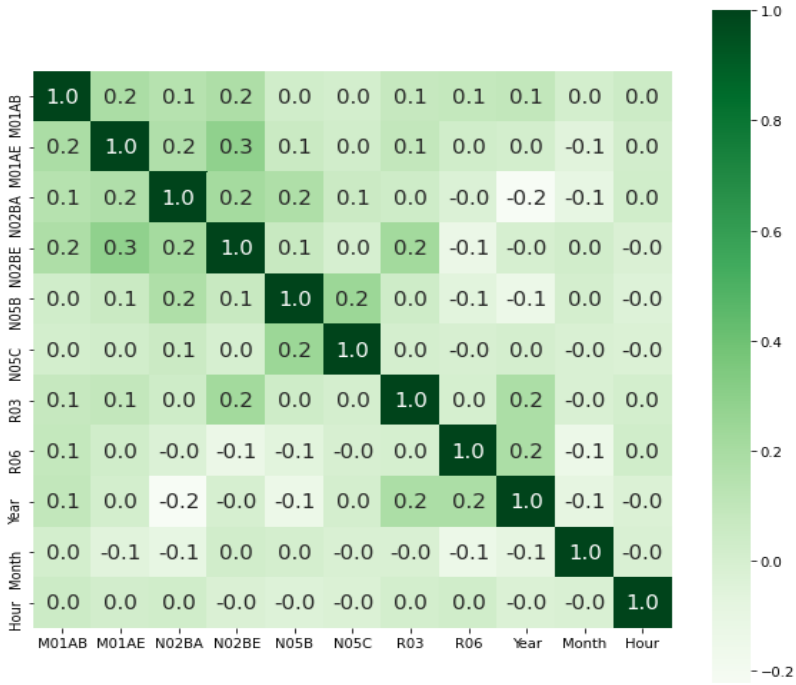
- N02BA - Other analgesics and antipyretics, Salicylic acid and derivatives
- N02BE/B - Other analgesics and antipyretics, Pyrazolones and Anilides
- N05B - Psycholeptics drugs, Anxiolytic drugs
- N05C - Psycholeptics drugs, Hypnotics and sedatives drugs
- R03 - Drugs for obstructive airway diseases
- R06 - Antihistamines for systemic use Sales data are resampled to the hourly, daily, weekly and monthly periods. Data is already pre-processed, where processing included outlier detection and treatment and missing data imputation.

```
1 df
```

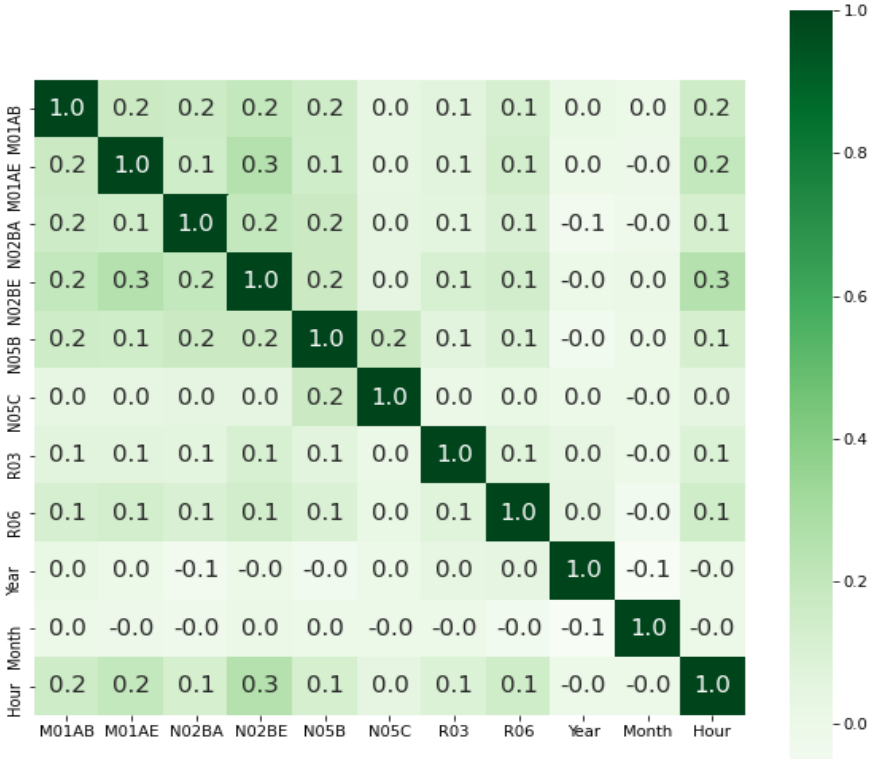
	datum	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Year	Month	Hour	Weekday Name
0	1/2/2014	0.00	3.670	3.40	32.40	7.0	0.0	0.0	2.00	2014	1	248	Thursday
1	1/3/2014	8.00	4.000	4.40	50.60	16.0	0.0	20.0	4.00	2014	1	276	Friday
2	1/4/2014	2.00	1.000	6.50	61.85	10.0	0.0	9.0	1.00	2014	1	276	Saturday
3	1/5/2014	4.00	3.000	7.00	41.10	8.0	0.0	3.0	0.00	2014	1	276	Sunday
4	1/6/2014	5.00	1.000	4.50	21.70	16.0	2.0	6.0	2.00	2014	1	276	Monday
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2101	10/4/2019	7.34	5.683	2.25	22.45	13.0	0.0	1.0	1.00	2019	10	276	Friday
2102	10/5/2019	3.84	5.010	6.00	25.40	7.0	0.0	0.0	0.33	2019	10	276	Saturday
2103	10/6/2019	4.00	11.690	2.00	34.60	6.0	0.0	5.0	4.20	2019	10	276	Sunday
2104	10/7/2019	7.34	4.507	3.00	50.80	6.0	0.0	10.0	1.00	2019	10	276	Monday
2105	10/8/2019	0.33	1.730	0.50	44.30	20.0	2.0	2.0	0.00	2019	10	190	Tuesday

2106 rows × 13 columns

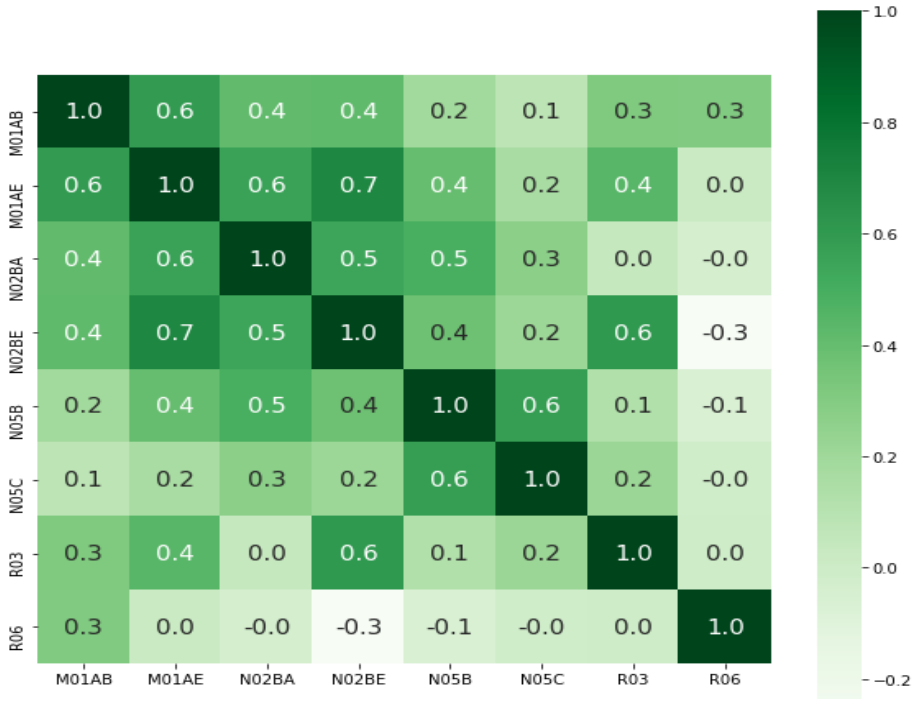
Sales daily data



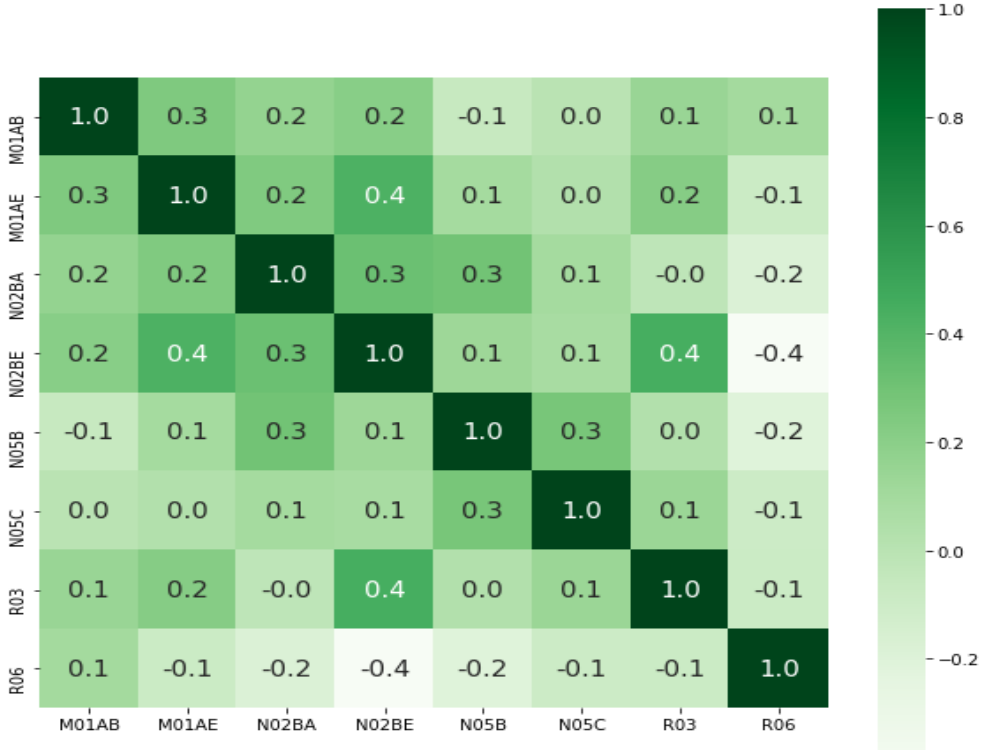
Correlation Matrix(Sales daily)



Correlation Matrix(Sales Hourly)

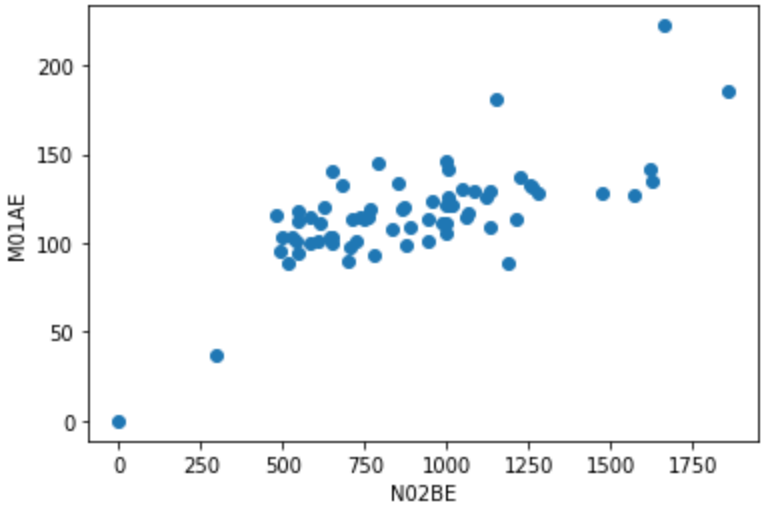


Correlation Matrix(Sales Monthly)



Correlation Matrix(Sales Weekly)

```
[ ]  1  plt.scatter(df1['N02BE'], df1['M01AE'])
     2  plt.xlabel("N02BE")
     3  plt.ylabel("M01AE")
     4  plt.show()
```



Scatter plot between most correlated drugs

6. Method Description

The methodology of the project is divided into 2 parts-

1. The first one is using various data visualization tools to visualize the raw data and convert it to useful information. The first visualization tool used is matplotlib.pyplot and seaborn from python libraries. These are popular plotting libraries from python and provide object oriented API for embedding plots into applications using general –purpose GUI like Tkinter. It is used to plot various graphs based on the data to better analyze the data. Apart from this another visualization tool called tableau is also used to analyze the data. It is a visual analytics platform transforming the way to use data to solve problems. It is used to visually depict the analysis of data we analyze and it is easy to decipher the questions we have regarding the data.

2. The second part involves the various machine learning models we will use to make future predictions through our collected data. The machine learning

models involve Linear Regression, Polynomial Regression and simple vector machines. Apart from this, to get better results we can use data science techniques like ensemble learning or neural networks.

Algorithms / Techniques / Softwares

- Google Colab
- Tableau
- Jupyter Notebook
- Python
- Numpy
- Pandas
- Matplotlib
- Seaborn
- Linear Regression
- Polynomial Regression
- Simple Vector Regression

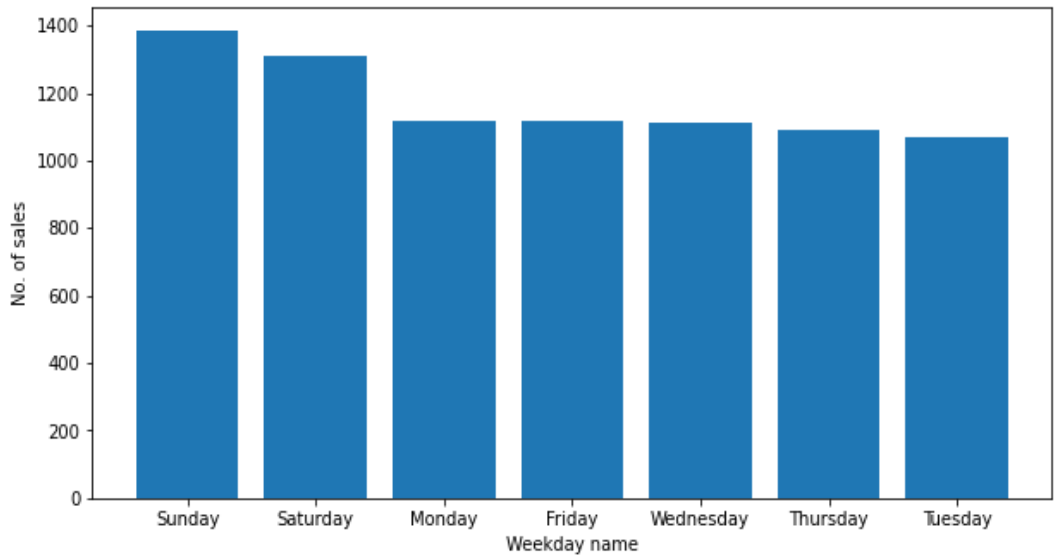
7. Results And Conclusions

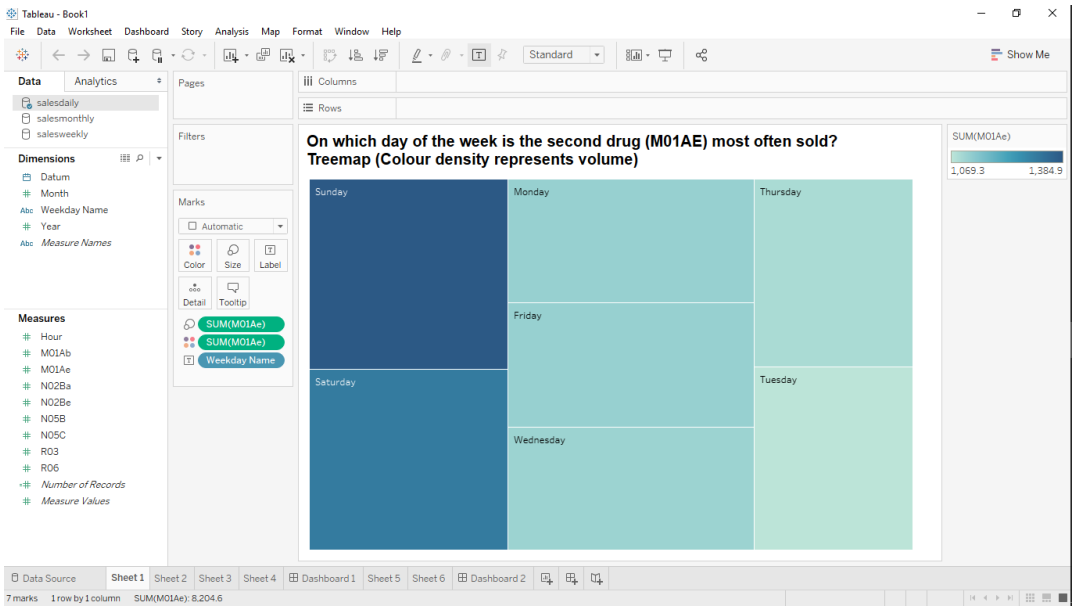
a. Results

1. On which day of the week is the second drug (M01AE) most often sold?

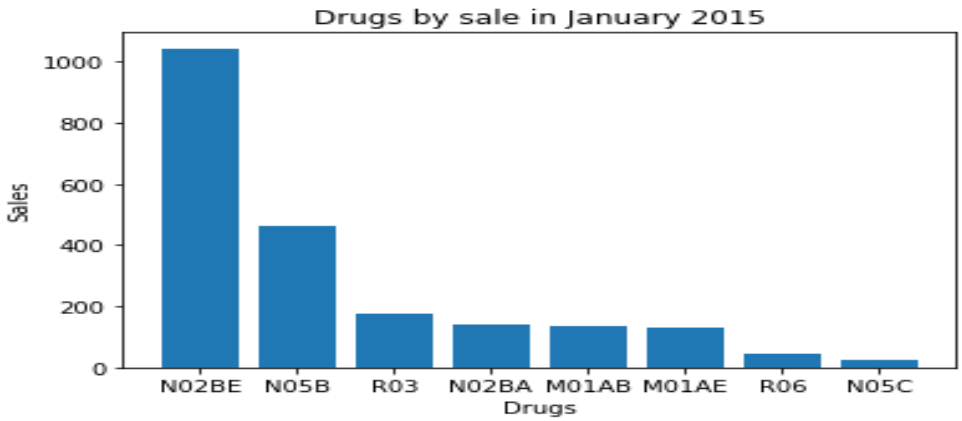
```
The second drug, M01AE, was most often sold on Sunday
with the volume of 1384.94
```

```
1  fig = plt.figure(figsize = (10, 5))
2  plt.bar(result['Weekday Name'], result['M01AE'])
3  plt.xlabel("Weekday name")
4  plt.ylabel("No. of sales")
5  plt.show()
```

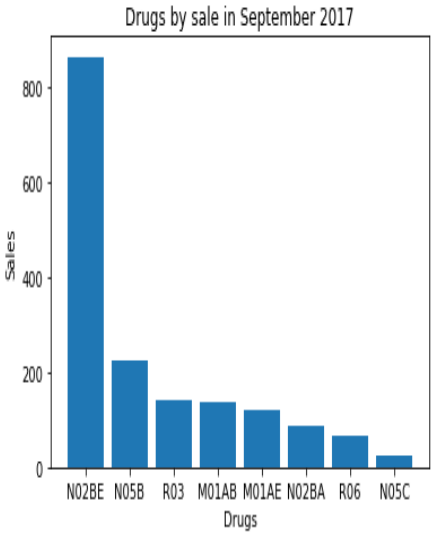




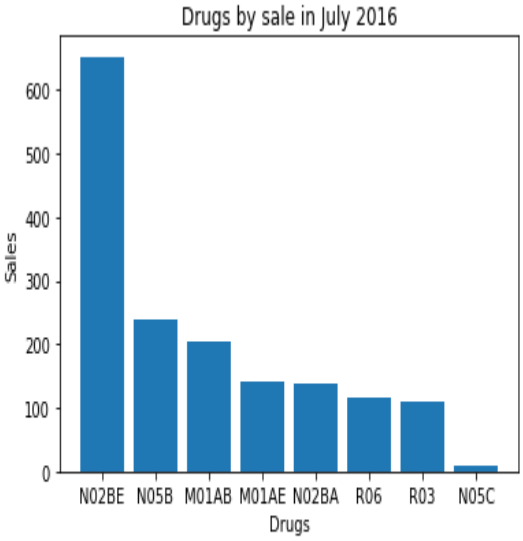
2. Which three drugs have the highest sales in January 2015, July 2016, September 2017.



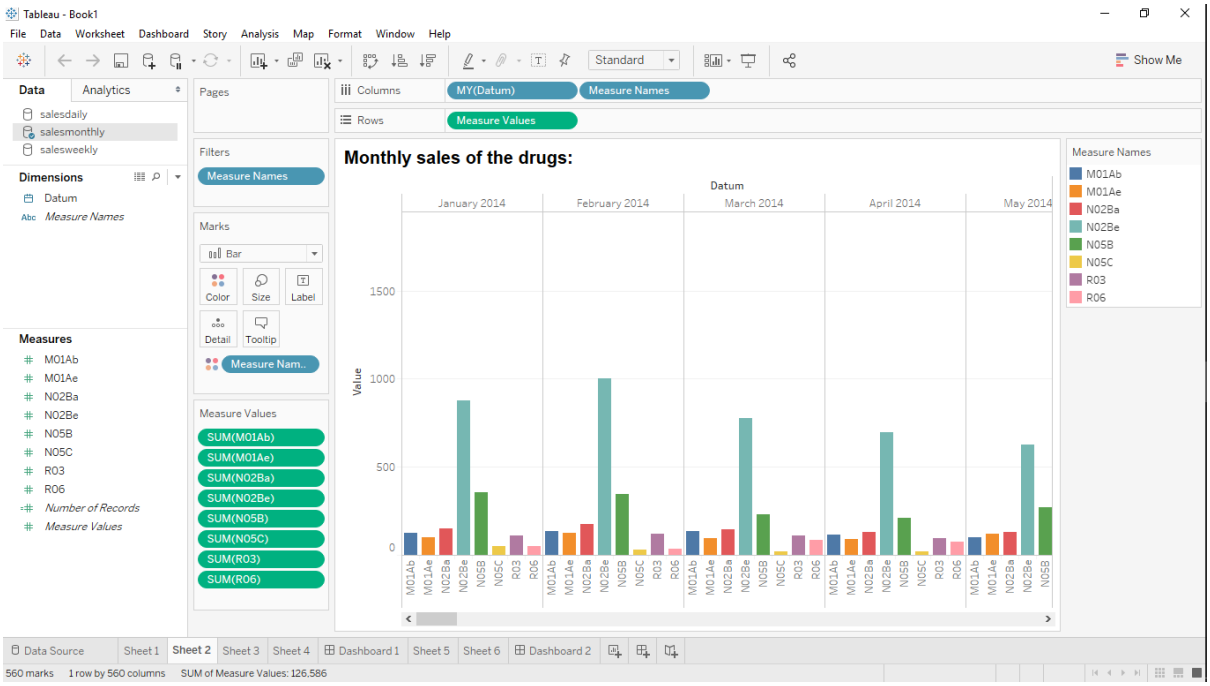

```
Top 3 drugs by sale in January 2015
 - Product: N02BE, Volume sold: 1044.24
 - Product: N05B, Volume sold: 463.0
 - Product: R03, Volume sold: 177.25
```




```
Top 3 drugs by sale in September 2017
 - Product: N02BE, Volume sold: 863.75
 - Product: N05B, Volume sold: 223.0
 - Product: R03, Volume sold: 139.0
```




```
Top 3 drugs by sale in July 2016
 - Product: N02BE, Volume sold: 652.36
 - Product: N05B, Volume sold: 240.0
 - Product: M01AB, Volume sold: 203.97
```

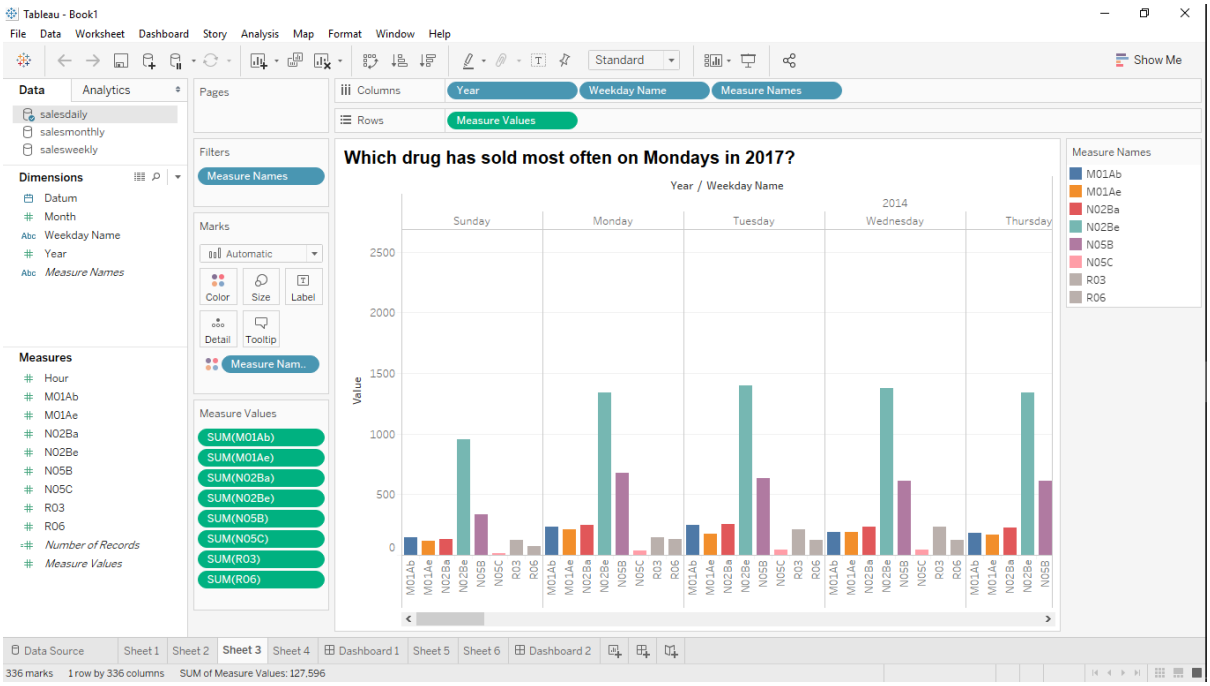


3. Which drug has sold most often on Mondays in 2017?

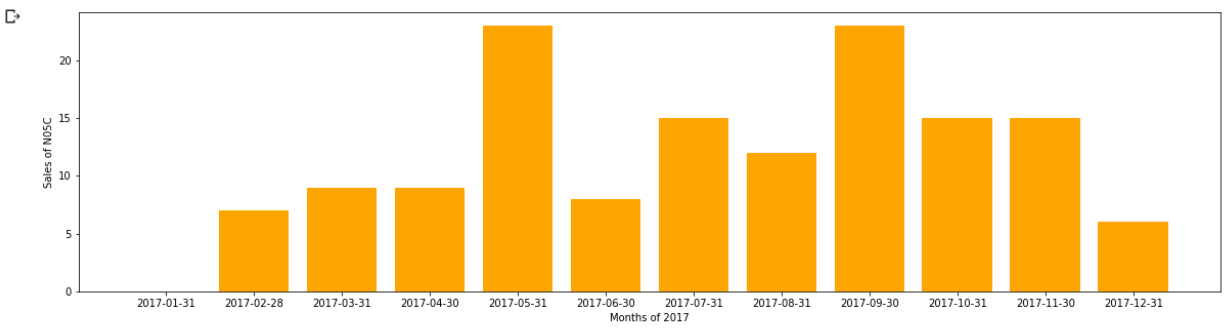
Displaying results

```
[ ]  1  for field in result.columns.values[0:1]:
     2      print('The drug most often sold on Mondays in 2017 is ' + str(field))
     3      print('with the volume of ' + str(round(result[field].iloc[0], 2)))
```

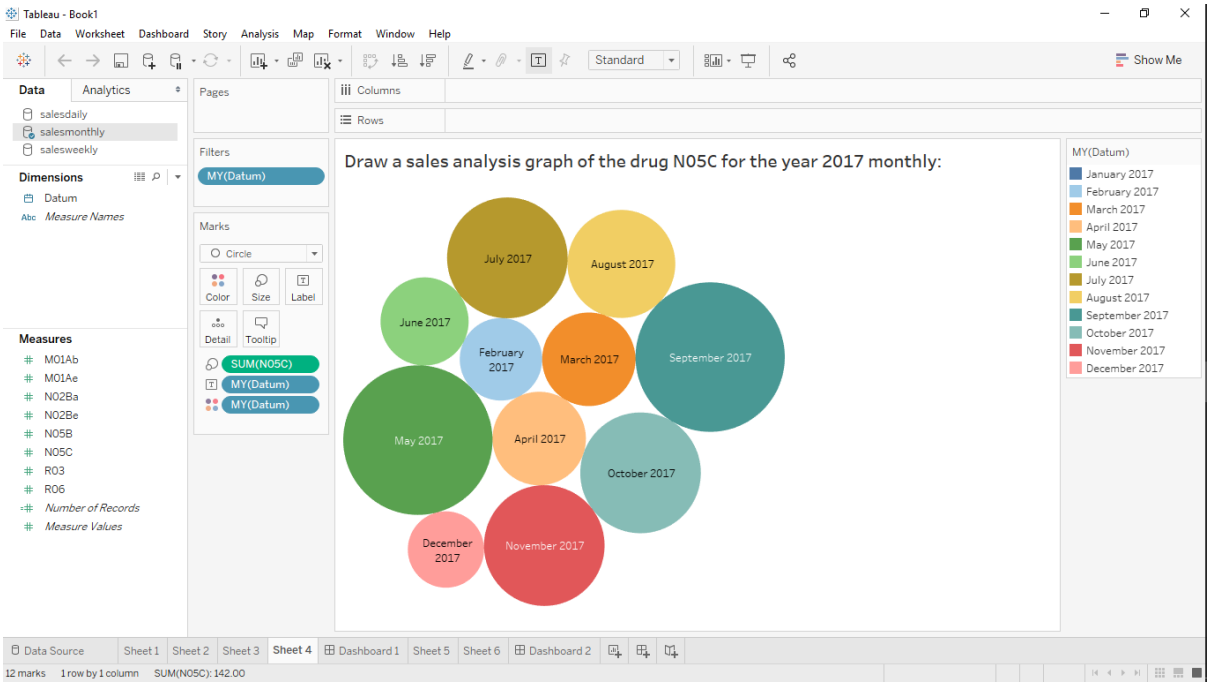
```
The drug most often sold on Mondays in 2017 is N02BE
with the volume of 1160.56
```



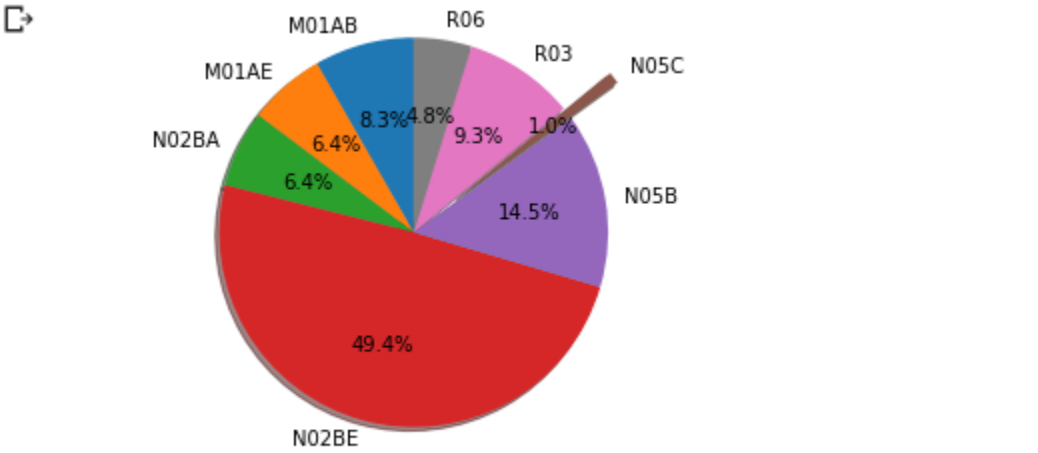
4. Draw a sales analysis graph of the drug N05C for the year 2017 monthly?



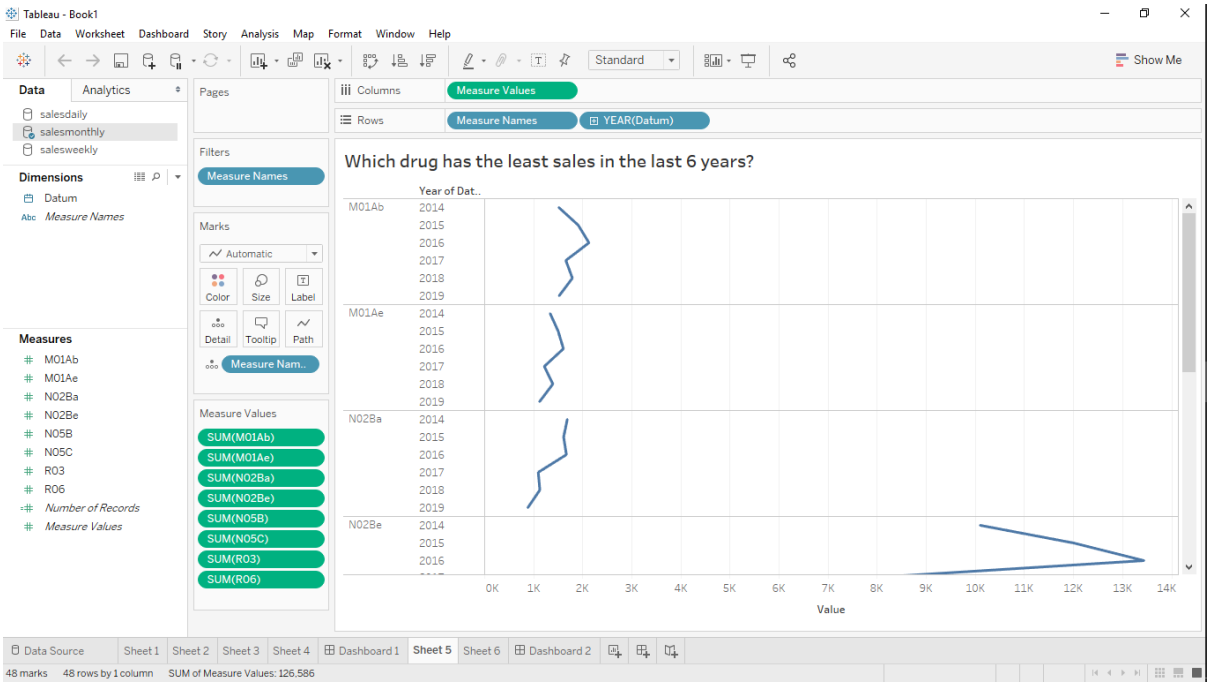
We see that sales of N05C in the year 2017 is maximum in the month of May and September



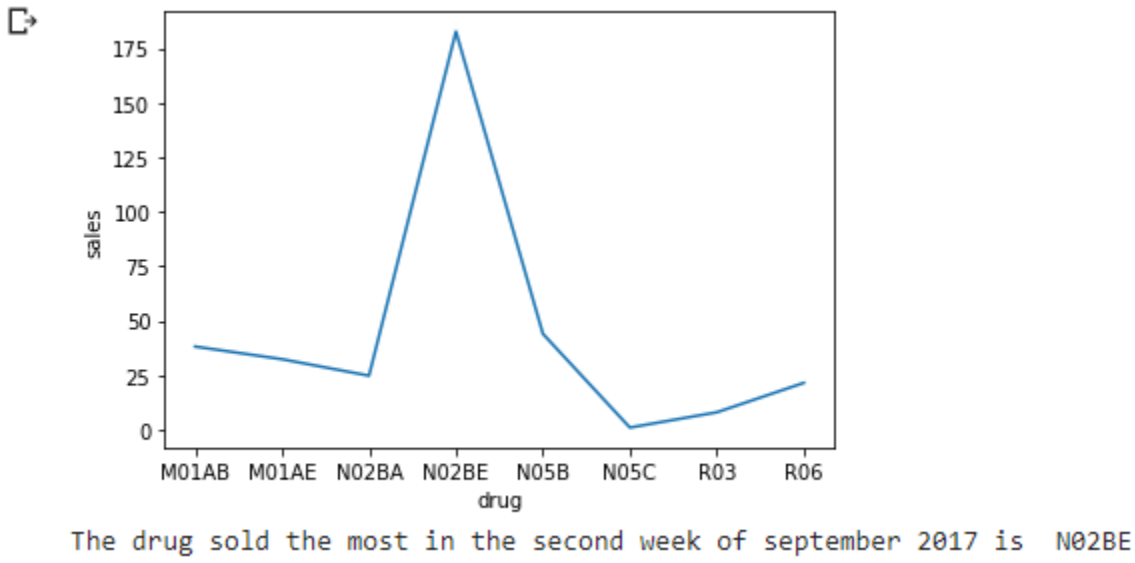
5. Which drug has the least sales in the last 6 years?

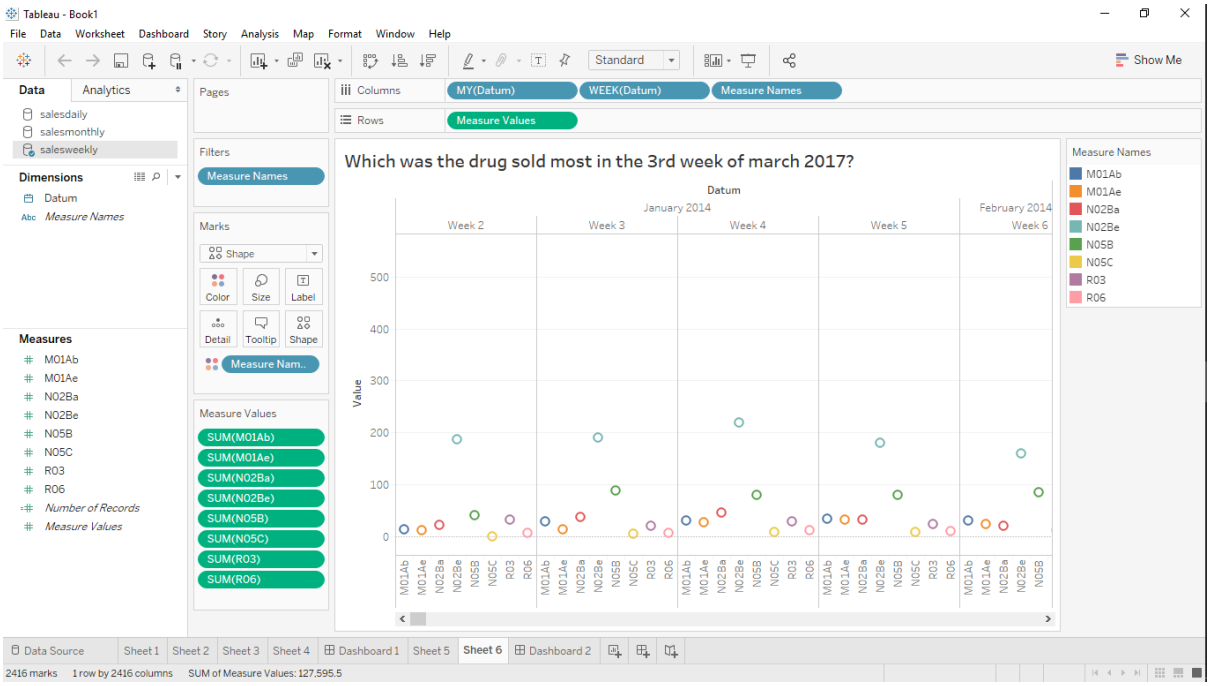



```
The drug which has the least sales in the last 5 years is N05C
```

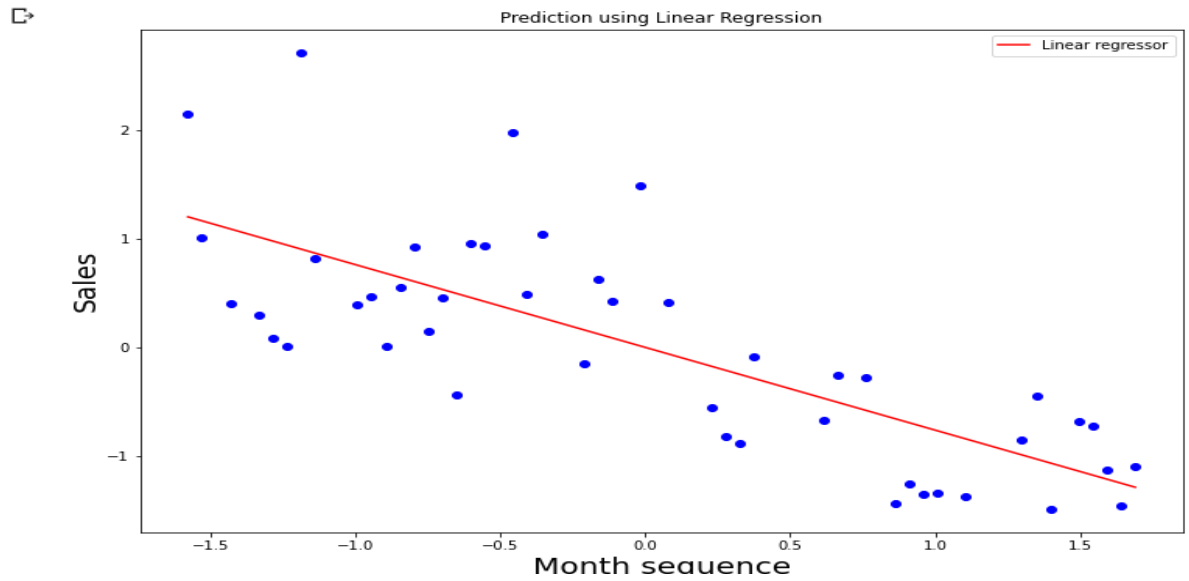


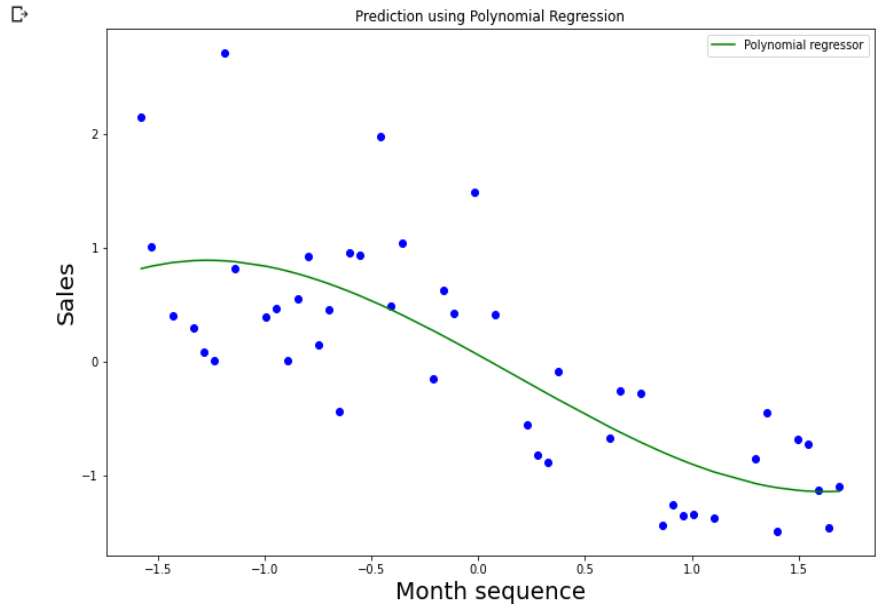
6. Which was the drug sold most in the 2 week of march 2017?

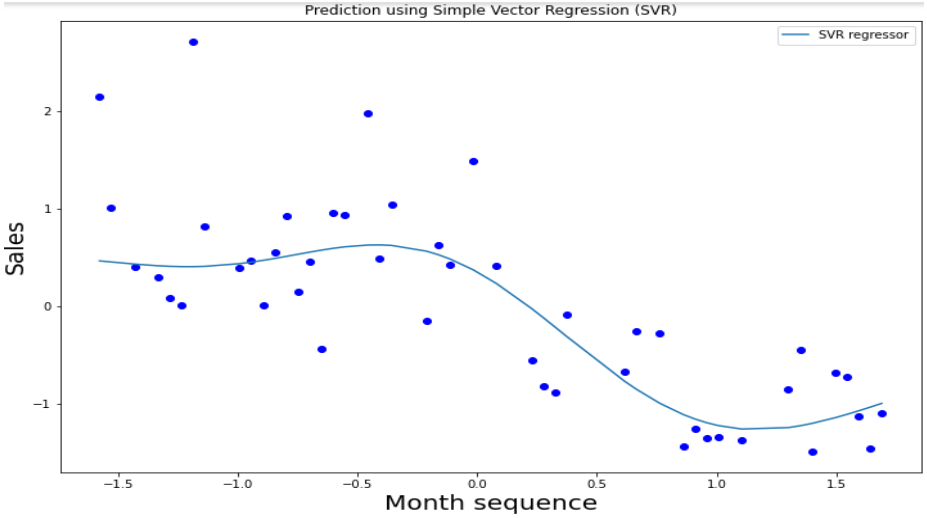




7. What medicine sales may be in January 2020? (Our data set only contains information about sales from January 2014 to October 2019)?








```
1  vRegressor.score(X_test, y_test)
```

```
0.837522368994241
```

```
1  regResults
```

	Linear	Polynomial	SVR	Voting Regressor
N02BA	86	93	121	98

b. Conclusion

Pharmaceutical sales analysis of these eight types of drugs was performed on daily, hourly, weekly and monthly data to identify the periods in which special sales and marketing campaigns could be implemented, except for N05B and N05C categories of drugs, which did not exhibit significant regularities. Sales analysis of pharmaceutical drugs is very useful and it helps in providing conclusions and recommendations to pharmacy. This helps the pharmacy to analyze the sales in order to maximize profit. We see that N02BE and M01AE drugs are highly co-related. This means that, if sales of N02BE increases then, sales of M01AE will also increase. As N02BE is an analgesic and antipyretics drug, M01AE is an Anti-inflammatory, and anti-rheumatic, both drugs are used together, that is why they are highly co-related. On analyzing the data it is seen that M01AE is mostly sold on Sunday's. Then N02BE is the most sold drug in the year 2015, 2016 and 2017. The N05C drug is the least sold drug in the last five years. Predicting the sales of the drugs using regression techniques like Linear Regression, Polynomial Regression and Support Vector Regression, it is seen that Support Vector Regression gives the best performance.

8. Future Work

1. To increase the size of the dataset.
2. Hyper parameter in Polynomial Regression and Support Vector Regression could be tuned to achieve higher results.
3. Time-Series Analysis could be used to get higher results.
4. LSTM, CNN LSTM and ConvLSTM could also be used to increase the performance of the model.

9. Reference

[1] A. Gupta, C. D. Maranas, and C. M. McDonald, “Mid-term supply chain planning under demand uncertainty: customer demand satisfaction and inventory management,” Computers andChemicalEngineering,vol.24,no.12,pp.2613–2621,2000.

[2] P. Doganis, A. Alexandridis, P. Patrinos, and H. Sarimveis, “Timeseriessalesforecastingforshortshelf-lifefoodproducts based on artificial neural networks and evolutionary computing,” Journal of Food Engineering, vol. 75, no. 2, pp. 196–204, 2006.

[3] G.E.P.BoxandG.M.Jenkins,TimeSeriesAnalysis:Forecasting and Control, Holden-Day, San Francisco, Calif, USA, 2nd edition,1976.

[4] L. Wang, H. Zou, J. Su, L. Li, and S. Chaudhry, “An ARIMAANN hybrid model for time series forecasting,” Systems Research and Behavioral Science, vol. 30, no. 3, pp. 244–259, 2013.

[5] P. G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,”Neurocomputing, vol. 50, pp. 159– 175,2003.

[6] F.Alpaslan,E.E˘grio˘glu,C.H.Alada˘g,andE.Tiring,“Anstatistical research on feed forward neural networks for forecasting timeseries,”American Journal of Intelligent Systems, vol. 2, no. 3,pp.21–25,2012.

[7]R.P.Diezand J.P.Fern´andez, Forecasting the Price of Energy in Spain’s Electricity Production Market, 9th Congress of Industrial Engineering,Gij´on,Spain,2005.

[8] T.Hill, L.Marquez, M.O’Connor, and W.Remus, “Neural networks models for time series forecasts,”Management Science, vol.42, no.7, pp.1082–1092, 1996.

[9] R. Carbonneau, K. Laframboise, and R. Vahidov, “Application of machine learning techniques for supply chain demand forecasting, ”European Journal of Operational Research, vol.184, no.3,pp.1140–1154,2008.

[10] F. L. Chen and T. Y. Ou, “A neural-network-based forecasting method for ordering perishable food in convenience stores,” in Proceedings of the 4th

International Conference on Natural Computation(ICNC'08),pp.250–254,IEEEComputerSociety, October2008.

[11] C. Frank, A. Garg, A. Raheja, and L. Sztandera, "Forecasting women's apparel sales using mathematical modeling," International Journal of Clothing Science and Technology, vol.15,no.2, pp.107–125,2003.

[12] L. Yu, S. Wang, and K. K. Lai, "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchangerates, "Computers and Operations Research, vol.32,no. 10,pp.2523–2541,2005.

[13] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," Decision Support Systems, vol. 46, no. 1, pp. 411–419, 2008.

[14] C.-W. Chu and G. P. Zhang, "A comparative study of linear and nonlinear models for aggregate retail sales forecasting," International Journal of Production Economics, vol.86,no.3,pp. 217–231,2003.

[16] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: a review of applications," Expert Systems with Applications,vol.36,no.1,pp.2–17,2009.

[17] S.Goyal, "Artificial neural networks (ANNs) infoodscience—a review,"International Journal of Scientific World, vol.1, no.2, pp. 19–28,2013.

[18] R.J.Kuo,P.Wu,andC.P.Wang,"Anintelligentsalesforecasting system through integration of artificial neural networks and fuzzy neural networks with fuzzy weight elimination," Neural Networks,vol.15,no.7,pp.909–925,2002.

[19] M.AdyaandF.Collopy, "How effective are neural networks at forecasting and prediction? Are view and evaluation, "Journal of Forecasting,vol.17,no.5-6,pp.481–495,1998.

[20] G. Zhang, E. B. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," International Journal of Forecasting, vol.14,no.1,pp.35–62,1998.

[21] M. Nelson, T. Hill, B. Remus, and M. O'connor, "Can neural networks applied to time series forecasting learn seasonal patterns: anempirical

investigation,”inProceedingsofthe27th Annual Hawaii International Conferenceon System Sciences, pp. 649–655, Wailea, Hawaii, USA, January1994.

[22] Z.Tang,C.deAlmeida,andP.A.Fishwick,“Timeseriesforecasting using neural networks vs. Box-Jenkins methodology,” Simulation,vol.57,no.5,pp.303–310,1991.

Appendix

```
# Loading libraries & Setup
"""

from google.colab import files
file = files.upload()

# Commented out IPython magic to ensure Python compatibility.
# Pandas - Data manipulation and analysis library
import pandas as pd
# NumPy - mathematical functions on multi-dimensional arrays and matrices
import numpy as np
# Matplotlib - plotting library to create graphs and charts
import matplotlib.pyplot as plt
# Re - regular expression module for Python
import re
# Calendar - Python functions related to the calendar
import calendar

# Manipulating dates and times for Python
from datetime import datetime

# Scikit-learn algorithms and functions
```

```
from sklearn import linear_model
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PolynomialFeatures
from sklearn.svm import SVR
from sklearn.ensemble import VotingRegressor
#from sklearn import metrics
from sklearn.metrics import mean_squared_error,r2_score
import seaborn as sns
# Settings for Matplotlib graphs and charts
from pylab import rcParams
rcParams['figure.figsize'] = 12, 8

# Display Matplotlib output inline
# %matplotlib inline

# Additional configuration
np.set_printoptions(precision=2)

# dir(metrics)

"""# Sales Analysis

Correlation Matrix
"""

# sales daily
df = pd.read_csv("salesdaily.csv")
```

```
corr = df.corr()
corr.shape

plt.figure(figsize=(10,10))
sns.heatmap(corr, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':15}, cmap='Greens')

# sales hourly
df1 = pd.read_csv("saleshourly.csv")

corr = df1.corr()
corr.shape

plt.figure(figsize=(10,10))
sns.heatmap(corr, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':15}, cmap='Greens')

# sales monthly
df1 = pd.read_csv("salesmonthly.csv")

corr = df1.corr()
corr.shape

plt.figure(figsize=(10,10))
sns.heatmap(corr, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':15}, cmap='Greens')

plt.scatter(df1['N02BE'], df1['M01AE'])
plt.xlabel("N02BE")
plt.ylabel("M01AE")
plt.show()
```

"""Coffecient of Correlation between N02BE and M01AE is 0.7

This means that both are highly co-related i.e if sales of N02BE increases then sales of M01AE will also increase

As N02BE is an analgesics and antipyretics drug and M01AE is an Anti-inflammatory and antirheumatic. As both drugs are used together, that is why they are highly co-related.

"""

```
# sales weekly
df1 = pd.read_csv("salesweekly.csv")
corr = df1.corr()
corr.shape
plt.figure(figsize=(10,10))
sns.heatmap(corr, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':15}, cmap='Greens')
```

"""# On which day of the week is the second drug (M01AE) most often sold?

Loading our sales daily data set from csv file using Pandas.

"""

```
df = pd.read_csv("salesdaily.csv")
```

"""Grouping the second drug sales by weekday name."""

```
df = df[['M01AE', 'Weekday Name']]
result = df.groupby(['Weekday Name'], as_index=False).sum().sort_values('M01AE', ascending=False)
```

"""Taking the weekday name with most sales and the volume of sales from the result"""

```
resultDay = result.iloc[0,0]
resultValue = round(result.iloc[0,1], 2)
```

```
"""Printing the result"""

print('The second drug, M01AE, was most often sold on ' + str(resultDay))

print('with the volume of ' + str(resultValue))

fig = plt.figure(figsize = (10, 5))

plt.bar(result['Weekday Name'], result['M01AE'])

plt.xlabel("Weekday name")

plt.ylabel("No. of sales")

plt.show()

"""# Which three drugs have the highest sales in Jan 2015, Jul 2016, Sep 2017

Loading monthly sales data set from csv file using Pandas.

"""

df = pd.read_csv("salesmonthly.csv")

"""Let's look at the data."""

df.head()

"""Because we will be repeating the same calculations for different months and years it is a good idea to write a function"""

def top3byMonth(month, year):

    """

    given a month and a year

    find top 3 drugs sold

    """
```

```
    month = str(month) if (month > 9) else '0'+str(month)
    year = str(year)
    # filter by date
    sales = df.loc[df['datum'].str.contains('^'+year+'\-'+month+'', flags=re.I, regex=True)]
    # reset index
    sales = sales.reset_index()
    # filter relevant columns
    topSales = sales[['M01AB', 'M01AE', 'N02BA', 'N02BE', 'N05B', 'N05C', 'R03', 'R06']]
    # sort values horizontally
    topSales = topSales.sort_values(by=0, ascending=False, axis=1)

    # plotting the bar graph
    drugs = topSales.columns.values
    sales = []
    for field in topSales.columns.values:
        sales.append(topSales[field].iloc[0])
    plt.bar(drugs, sales)
    plt.xlabel("Drugs")
    plt.ylabel("Sales")
    plt.title("Drugs by sale in "+calendar.month_name[int(month)]+' '+year)
    plt.show()

    # print results
    print('Top 3 drugs by sale in '+calendar.month_name[int(month)]+' '+year)
    for field in topSales.columns.values[0:3]:
        print(' - Product: ' + str(field) + ', Volume sold: ' + str(round(topSales[field].iloc[0], 2)))
    print("\n")

"""We are now calling the function for different months and years and printing results"""
```

```
# top3 drugs by sale in January 2015

top3byMonth(1, 2015)


# top3 drugs by sale in July 2016

top3byMonth(7, 2016)


# top3 drugs by sale in September 2017

top3byMonth(9, 2017)


"""# Which drug has sold most often on Mondays in 2017?


Loading our sales daily data set from csv file using Pandas.

"""


df = pd.read_csv("salesdaily.csv")


"""Filtering out from the data everything else apart from yar 2017 and Monday"""


df = df.loc[df['datum'].str.contains('2017', flags=re.I, regex=True) & (df['Weekday Name'] == 'Monday')]


"""Groupping by weekday name and summarising"""


df = df.groupby(['Weekday Name'], as_index=False).sum()


"""Filtering only relevant columns and sorting values of most sold drugs horizontally to achieve the most often sold drug on the left"""


df = df[['M01AB', 'M01AE', 'N02BA', 'N02BE', 'N05B', 'N05C', 'R03', 'R06']]

result = df.sort_values(by=0, ascending=False, axis=1)


"""Displaying results"""
```

```
for field in result.columns.values[0:1]:
    print('The drug most often sold on Mondays in 2017 is ' + str(field))
    print('with the volume of ' + str(round(result[field].iloc[0], 2)))

"""# The sales analysis graph of the drug N05C for year 2017 monthly"""

df = pd.read_csv("salesmonthly.csv")

sales = df.loc[df['datum'].str.contains('2017')]
sales

fig = plt.figure(figsize = (20, 5))
plt.bar(sales['datum'], sales['N05C'], color='orange')
plt.xlabel("Months of 2017")
plt.ylabel("Sales of N05C")
plt.show()

"""We see that sales of N05C in the year 2017 is maximum in the month of May and September

# Which drug has the least sales in the last 5 years?
"""

df = pd.read_csv("salesmonthly.csv")

df.head()

sales = {}
sales["M01AB"] = 0
sales["M01AE"] = 0
```

```
sales["N02BA"] = 0
sales["N02BE"] = 0
sales["N05B"] = 0
sales["N05C"] = 0
sales["R03"] = 0
sales["R06"] = 0

for index,row in df.iterrows():
  sales["M01AB"] += row["M01AB"]
  sales["M01AE"] += row["M01AE"]
  sales["N02BA"] += row["N02BA"]
  sales["N02BE"] +=row["N02BE"]
  sales["N05B"] += row["N05B"]
  sales["N05C"]+= row["N05C"]
  sales["R03"]+= row["R03"]
  sales["R06"] +=row["R06"]

drugs = list(sales.keys())
money = list(sales.values())
#plt.plot(drugs,money)
# plt.xlabel("medicine")
# plt.ylabel("sales")
# plt.show()
explode = (0, 0, 0, 0,0,0.3,0,0)
fig,ax1 = plt.subplots()
ax1.pie(money,explode = explode ,labels=drugs, autopct='%1.1f%%', shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
#plt.figure(figsize = (100,100))
plt.show()
print(" The drug which has the least sales in the last 5 years is N05C")
```

```
"""# . Which was the drug sold most in the 2 week of march 2017?"""

df = pd.read_csv("salesweekly.csv")

df.head()

# type(df['datum'])

a = df.loc[df['datum'] == '9/10/2017']

num = []
for (columnName, columnData) in a.iteritems():
  num.append(columnData.values)
num.pop(0)
ind =  num.index(max(num))
plt.plot(list(a.columns[1:]),list(a.values[0][1:]))
plt.xlabel("drug")
plt.ylabel("sales")
plt.show()
print("The drug sold the most in the second week of september 2017 is ",a.columns[ind+1])

"""# Sales of R06 for the year 2014-2019"""

month = pd.read_csv("salesmonthly.csv")

plt.plot(month['R06'])
plt.xlabel("Months")
plt.ylabel("Sale")
plt.title("Sales of R06 from 2014-2019")
```

```
plt.show()


plt.plot(month['R06'][0:12])
plt.xlabel("Months")
plt.ylabel("Sale")
plt.title("Sales of R06 for the year 2014")
plt.show()

"""From the above graphs we see that R06(Antihistamines) has the maximum sale during the months of May-July

# Predicting the sales of N02BA in January 2020

Defining the scattering function that will display scattered sales data on the chart
"""

def scatterData(X_train, y_train, X_test, y_test, title):
    plt.title('Prediction using ' + title)
    plt.xlabel('Month sequence', fontsize=20)
    plt.ylabel('Sales', fontsize=20)

    # Use Matplotlib Scatter Plot
    plt.scatter(X_train, y_train, color='blue')
    # plt.scatter(X_test, y_test, color='cyan', label='Testing observation points')

"""Defining predict sales and display Linear Regression model function"""

def predictLinearRegression(X_train, y_train, X_test, y_test):

    y_train = y_train.reshape(-1, 1)
    y_test = y_test.reshape(-1, 1)
```

```
    scatterData(X_train, y_train, X_test, y_test, 'Linear Regression')

    reg = linear_model.LinearRegression()
    reg.fit(X_train, y_train)
    plt.plot(X_train, reg.predict(X_train), color='red', label='Linear regressor')
    plt.legend()
    plt.show()

    # LINEAR REGRESSION - Predict/Test model
    y_pred = reg.predict(X_test)

    # LINEAR REGRESSION - Predict for January 2020
    linear_predict = reg.predict([[predictFor]])
    # linear_predict = reg.predict([[predictFor]])[0]

    # LINEAR REGRESSION - Accuracy
    accuracy = reg.score(X_train, y_train)
    #r2 = r2_score(y_test, y_pred)
    #r2_adjusted = 1-(1-r2_score(y_test, y_pred))*((len(X_test)-1)/(len(X_test)-len(X_test[0])-1))

    # LINEAR REGRESSION - Error
    # error = round(np.mean((y_predict_linear-y_test)**2), 2)

    # Results
    print('Linear Regression: ' + str(linear_predict) + ' (Accuracy: ' + str(round(accuracy*100)) + '%)')

    return {'regressor':reg, 'values':linear_predict}

"""Defining predict sales and display Polynomial Regression model function"""
```

```
def predictPolynomialRegression(X_train, y_train, X_test, y_test):

    y_train = y_train.reshape(-1, 1)
    y_test = y_test.reshape(-1, 1)

    scatterData(X_train, y_train, X_test, y_test, 'Polynomial Regression')

    poly_reg = PolynomialFeatures(degree = 3)
    X_poly = poly_reg.fit_transform(X_train)
    poly_reg_model = linear_model.LinearRegression()
    poly_reg_model.fit(X_poly, y_train)
    plt.plot(X_train, poly_reg_model.predict(poly_reg.fit_transform(X_train)), color='green', label='Polynomial regressor')
    plt.legend()
    plt.show()

    # Polynomial Regression - Predict/Test model
    y_predict_polynomial = poly_reg_model.predict(X_poly)

    # Polynomial Regression - Predict for January 2020
    polynomial_predict = poly_reg_model.predict(poly_reg.fit_transform([[predictFor]]))

    # Polynomial Regression - Accuracy
    # X_poly_test = poly_reg.fit_transform(X_test)
    accuracy = poly_reg_model.score(X_poly, y_train)

    # Polynomial Regression - Error
    # error = round(np.mean((y_predict_polynomial-y_train)**2), 2)

    # Result
```

```
    print('Polynomial Regression: ' + str(polynomial_predict) + ' (Accuracy: ' + str(round(accuracy*100)) + '%)')
    return {'regressor':poly_reg_model, 'values':polynomial_predict}

"""


Defining predict sales and display Support Vector Regression (SVR) function"""

def predictSVR(X_train, y_train, X_test, y_test):

    y_train = y_train.reshape(-1, 1)
    y_test = y_test.reshape(-1, 1)

    scatterData(X_train, y_train, X_test, y_test, 'Simple Vector Regression (SVR)')

    svr_regressor = SVR(kernel='rbf', gamma='auto')
    svr_regressor.fit(X_train, y_train.ravel())

    # plt.scatter(X_train, y_train, color='red', label='Actual observation points')
    plt.plot(X_train, svr_regressor.predict(X_train), label='SVR regressor')
    plt.legend()
    plt.show()

    # Simple Vector Regression (SVR) - Predict/Test model
    y_predict_svr = svr_regressor.predict(X_test)

    # Simple Vector Regression (SVR) - Predict for January 2020
    svr_predict = svr_regressor.predict([[predictFor]])

    # Simple Vector Regression (SVR) - Accuracy
```

```
    accuracy = svr_regressor.score(X_train, y_train)

    # Simple Vector Regression (SVR) - Error
    # error = round(np.mean((y_predict_svr-y_train)**2), 2)

    # Result
    print('Simple Vector Regression (SVR): ' + str(svr_predict) + ' (Accuracy: ' + str(round(accuracy*100)) + '%)')
    return {'regressor':svr_regressor, 'values':svr_predict}

"""We are defining a product that we will be predicting the January 2020 sales for.
We can change it to a differnt one and use the same calculations for a different product.
"""

product = 'N02BA'

"""For storing all regression results"""

regResults = pd.DataFrame(columns=('Linear', 'Polynomial', 'SVR', 'Voting Regressor'), index=[product])

"""To display a larger graph than a default with specify some additional parameters for Matplotlib library."""

rcParams['figure.figsize'] = 12, 8

"""We will be using monthly data for our predictions"""

df = pd.read_csv("salesmonthly.csv")

"""We will use monthly sales data from 2017, 2018, 2019. We could also use just 2019 for that."""

df = df.loc[df['datum'].str.contains("2014") | df['datum'].str.contains("2015") | df['datum'].str.contains("2016") |
df['datum'].str.contains("2017") | df['datum'].str.contains("2018") | df['datum'].str.contains("2019")]
```

```
df = df.reset_index()

"""It is always a good practice to look at the data often"""

df

"""We are adding a sequence number for each month as an independent variable"""

df['datumNumber'] = 1
for index, row in df.iterrows():
    df.loc[index, 'datumNumber'] = index+1

"""Removing the first and the last incompleted record from Pandas Data Frame"""

# the first and the last available month is quite low which may indicate that it might be incomplete
# and skewing results so we're dropping it
df.drop(df.head(1).index,inplace=True)
df.drop(df.tail(1).index,inplace=True)

"""Cleaning up any rows with the product value = 0."""

df = df[df[product] != 0]

"""Let's look at the data again."""

df.head()

"""What value we predict for? January 2020. Because we have data until August 2019 we're predicting for 5 months ahead"""

predictFor = len(df)+5
```

```
print('Predictions for the product ' + str(product) + ' sales in January 2020')

"""For storing regression results."""

regValues = {}

"""Preparing training and testing data by using train_test_split function. 70% for training and 30% for testing."""

dfSplit = df[['datumNumber', product]]

# We are going to keep 30% of the dataset in test dataset
train, test = train_test_split(dfSplit, test_size=0.3, random_state=0)

trainSorted = train.sort_values('datumNumber', ascending=True)
testSorted = test.sort_values('datumNumber', ascending=True)

X_train = trainSorted[['datumNumber']].values
y_train = trainSorted[product].values
X_test = testSorted[['datumNumber']].values
y_test = testSorted[product].values

dfSplit.head()

"""Performing feature scaling. Scaling the feature will improve the performance of the model."""

scale_X = StandardScaler()
scale_y = StandardScaler()

X_train = scale_X.fit_transform(X_train)
y_train = scale_y.fit_transform(y_train.reshape(-1, 1))
```

```
X_test = scale_X.fit_transform(X_test)
y_test = scale_y.fit_transform(y_test.reshape(-1, 1))

"""Performing and saving results for Linear Regression"""

# LINEAR REGRESSION
linearResult = predictLinearRegression(X_train, y_train, X_test, y_test)
reg = linearResult['regressor']
regValues['Linear'] = round(linearResult['values'][0][0])

"""Performing and saving results for Polynomial Regression"""

# POLYNOMIAL REGRESSION
polynomialResult = predictPolynomialRegression(X_train, y_train, X_test, y_test)
polynomial_regressor = polynomialResult['regressor']
regValues['Polynomial'] = round(polynomialResult['values'][0][0])

"""Performing and saving results for Simple Vector Regression (SVR)"""

# SIMPLE VECTOR REGRESSION (SVR)
svrResult = predictSVR(X_train, y_train, X_test, y_test)
svr_regressor = svrResult['regressor']
regValues['SVR'] = round(svrResult['values'][0])

"""Voting Regressor"""

vRegressor = VotingRegressor(estimators=[('reg', reg), ('polynomial_regressor', polynomial_regressor), ('svr_regressor', svr_regressor)])

vRegressorRes = vRegressor.fit(X_train, y_train.ravel())
```

```
# VotingRegressor - Predict for January 2020
vRegressor_predict = vRegressor.predict([[predictFor]])
regValues['Voting Regressor'] = round(vRegressor_predict[0])
print('Voting Regressor January 2020 predicted value: ' + str(round(vRegressor_predict[0])))
regResults.loc[product] = regValues

vRegressor.score(X_test, y_test)

"""Displaying all results"""

regResults
```