

Time Forecasting System

Major project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering.

By:

MANIK SAYAL
(181474)

UNDER THE SUPERVISION

Of

DR. PARDEEP KUMAR

(Associate Professor)

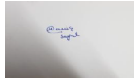


Department of Computer Science & Engineering and
Information Technology, Jaypee University of Information
Technology, Waknaghat , 173234, Himachal Pradesh,
INDIA.

Candidate Declaration

I hereby declare that the work presented in this report titled "Time Series Forecasting System" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology, submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of my own work carried out over a period of six months beginning in August of this year. Under the supervision of Dr. Pardeep Kumar (Associate Professor).

The report's contents have not been submitted for the granting of any other degree or certificate.



(Student Signature)

Student Name: Manik Sayal

Rollno. : 181474

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



(Supervisor Signature)

Supervisor: Dr. Pardeep Kumar

Designation: Associate Professor

Department names: CSE and IT department

Dated: 26/05/2022

ACKNOWLEDGMENT

To begin, I would like to offer my heartfelt gratitude and appreciation to Almighty God for His wonderful grace, which has enabled us to successfully finish the project work.

I am really grateful and desire to express my heartfelt gratitude to my supervisor, Dr. Pardeep Kumar, Associate Professor, Department of CSE, Jaypee University of Information Technology, Wakhnaghat. My supervisor's deep knowledge and genuine interest in the subject of machine learning are essential for carrying out this research. His unending patience, intellectual direction, consistent encouragement, persistent and vigorous supervision, constructive criticism, helpful counsel, and reading many poor versions and correcting them at all stages, enabled us to accomplish this project.

I would like to convey my heartfelt thanks to Dr. Pardeep Kumar, Department of CSE, for his generous assistance in completing my research.

I would also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unusual scenario, I would like to thank the many staff members, both teaching and non-teaching, who have provided their convenient assistance and helped my project.

Finally, I must express my gratitude for my parents' unwavering support and patience.



Manik Sayal

(181474)

Table Of Content

Title Page		
Certificate.....	(i)	
Acknowledgement.....	(ii)	
Table of Contents...	(iii)	
List of Figures.....	(iv)	
Abstract.....	(v)	
1. Chapter-1	Introduction	1
2. Chapter-2	Literature Survey	8
3. Chapter-3	System Development	13
4. Chapter-4	Performance Analysis	25
5. Chapter-5	Conclusion	39
References		43

List Of Figures:

1. ARIMA Graph
2. SARIMA Graph
3. Prophet Graph
4. Holt Winters' Graph
5. Data Set Image
6. Top 10 clients
7. Bottom 10 Clients
8. Outlier treatment Images
9. Flask UI

Abstract

Time series analysis is a statistical paradigm that is suitable for a variety of continuous research designs. Sole investigators or experimental divisions are more often assessed continually at periodic intervals over a large number of points in such systems. Time Series analytics may very well be conceived of as an instance of something like a longitudinal design. A time series analysis can assist us in addressing the underlying dynamical mechanism, the variation through time, or analyzing the repercussions of a scheduled or unscheduled intervention. We will describe time series analytics as it is often used in psychological research, outlining both the technique's prior record and future trends. Time series designs are becoming a more acceptable method for discovering significant psychological aspects as information system technology progresses.

Time series forecasting is a data science strategy which is widely used in business, finance, supply chain management, manufacturing, and inventory planning. Numerous projection challenges have a time component and hence need time series extrapolation for forecasting. Time series forecasting is indeed a key field of machine learning (ML) that may be regarded as a supervised learning technique. It may be used with ML approaches such as Regression, Neural Networks, Support Vector Machines, Random Forests, and XGBoost. Forecasting is the process of employing models fit on previous data to anticipate future discoveries.

Forecasting or anticipating potential value throughout time is referred to as time series forecasting. It entails devising simulations based on historical data and using them to discover information and plan the future business strategies. Based on what has already occurred, the potential is projected or anticipated. Time series conveys time order correlation among samples. This interdependence is both a limitation and a component that offers an extra source of information.

A forecasting time series begins with a prior time series. Forecasters study existing data for time decomposition characteristics such as trends, seasonal patterns, repetitive patterns, and predictability. Many departments inside firms, such as marketing, finance, and sales, employ some sort of time series forecasting to estimate likely technical costs and customer demand. Time series data models may take numerous shapes and reflect many stochastic processes.

Introduction

1.1 Problem Statement

We were expected to forecast the revenue generated by the customers, for the company. We were given the records of 1495 customers, along with their go-live date, Biller Type, Industry Type Fees and revenue generated from those customers from January 2012 to September 2020. We wanted to forecast the future earnings from those customers. The important thing to note is that we were given the data on a monthly basis. So the problem here is best handled using a time series approach as we were provided a time component in the data, but just forecasting the revenue generated by the customers was not our ultimate goal. We wanted to create an API for the task in hand as well as any other time series problems coming our way in the future. That is, we wanted to automate the whole process and keep it as simple as possible.

When you are presented with a new time series forecasting problem, there are many things to consider. The choice that you make directly impacts each step of the project, from the design of a test harness to evaluate forecast models to the fundamental difficulty of the forecast problem that you are working on.

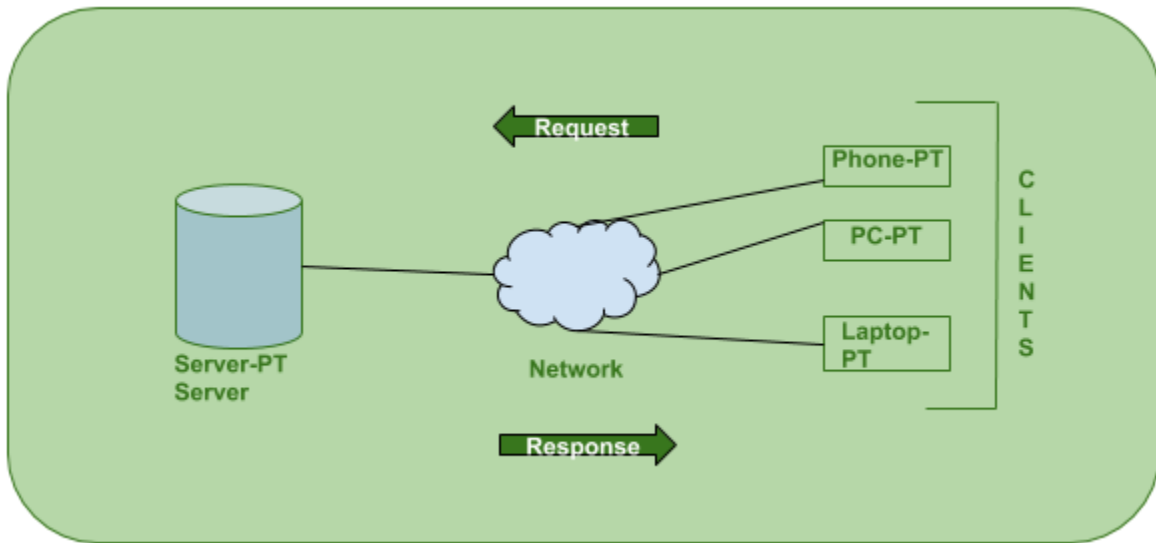
It is possible to very quickly narrow down the options by working through a series of questions about your time series forecasting problem. By considering a few themes and questions within each theme, you narrow down the type of problem, test harness, and even choice of algorithms for your project.

1.2 Methodology

The whole project works in two phases:

Phase 1: Client Server connection

The user will connect to the server and pass the required information to the server using the http protocol.



Phase 2: Hyperparameter tuning on selected models

After getting the information server, it will compute the results on all the models generated, and after selecting the best model, the user will be served with the predictions of the best model.

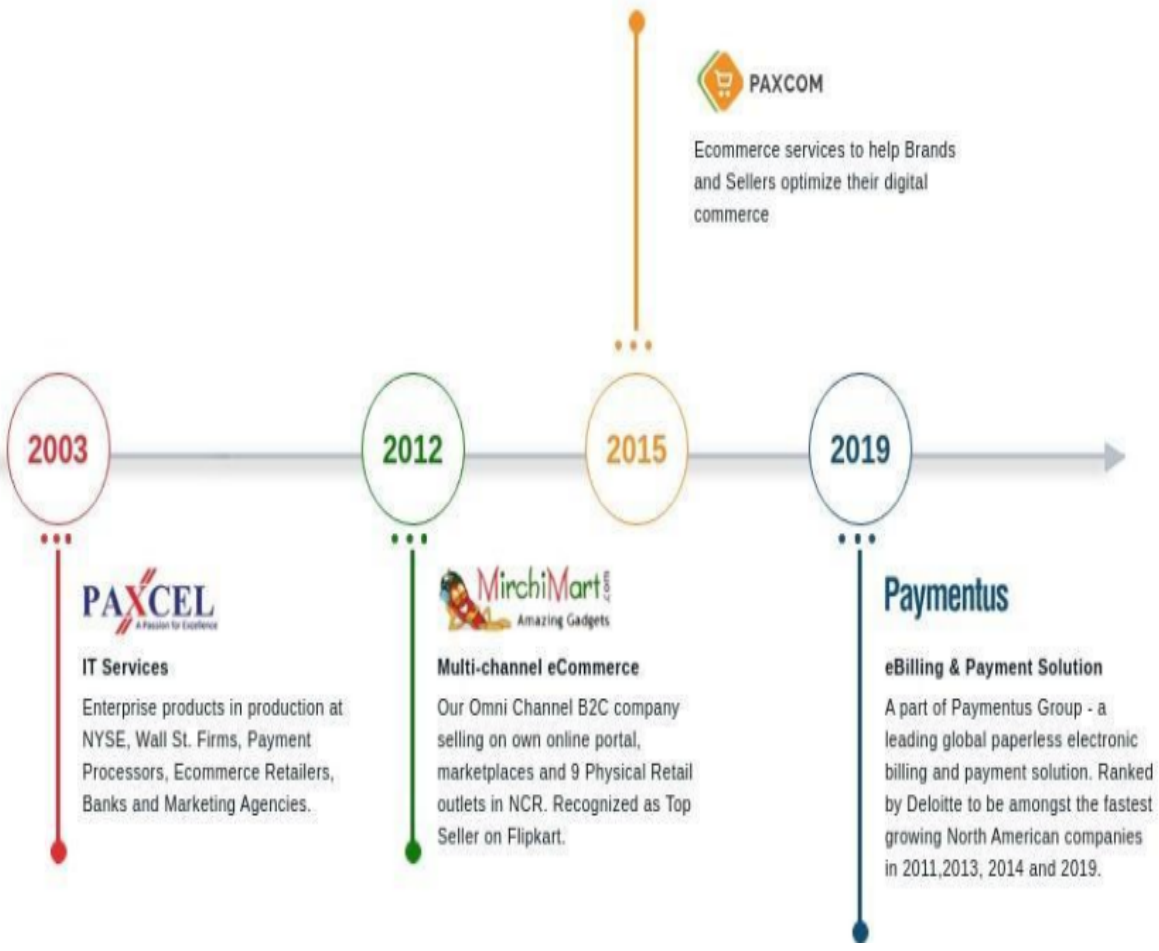


1.3 Organization

Paxcom:

We are a team of 200+Ecommerce enthusiasts, passionate about using technology to simplify Digital Commerce and Payments for brands across the globe. Paxcom is a part of the Paymentus group -a leading

Global paperless electronic billing and payment solution providers



speaking. That characterized way empowers the making of a uniform code base over an association, without agonizing over characterizing certain principles.

Angular as of now accompanies these principles out of the crate. Tailing them doesn't just build the consistency and hence the nature of the code. It makes your application more obvious, as well. That is, in the event that you are as of now acquainted with angular. This exacting methodology likewise proves to be useful across collaboration fringes. It empowers new engineers to incorporate into another group rapidly, as a result of the high commonality with the code.

What I need to state is, you should follow the angular structure rules to capitalize on the angular system. It will make your life significantly simpler, when coming into new undertakings and will expand the nature of your code consequently.

A typical activity, when learning angular is to put everything into the application module. Obviously, that your application will advance into a total chaos. Modules are there which is as it should be!

Modules help to sort out your code into littler groups to make discovering things simpler. Yet, they are not just a corrective thing. With the assistance of lethargic stacked modules, you can likewise build the client experience by just downloading the pieces of the application, that are required at that point.

Angular is written in typescript by Google development team and is maintained by them regularly. It's first initial release was in beginning of 2016. Very first version was called as Angular-JS so in future updates to make it distinguish from other they named it to just Angular and removed the JS. 11 stable versions has been released by the google team along with the community support as of date 11 Nov 2020. To do material design in angular, it provides angular material library.

Paymentus:

Paymentus is a North Carolina based software company providing complete billing solutions.

I worked at Paymentus India Pvt Ltd at Mohali branch. I worked with the Hybrid-Billers Team and was successfully able to understand their working structure and pattern. I also learned soft skills like communicating within a corporate firm and working with a team. I was successfully able to understand various coding paradigms used by a company to build its application. I got a thorough understanding of some of the company's existing products and some of the upcoming products.

The working culture of the company is great. I thoroughly enjoyed myself working there. Paymentus has a typical blend of work and fun. Although I didn't get to spend much time in the company office and started working from home after the coronavirus pandemic lockdown, I really enjoyed the weekend football and various parties at the company. And even after the lockdown the communication between me and my team was good through various meeting platforms like skype and google meets.

There are numerous things I love about my internship - the experience/information I've acquired, the balance between fun and serious activities climate, my director and colleagues, and surprisingly the gathering of people I exercise with on my mid-day breaks. In any case, in the event that I needed to put it on a certain something, what I like most is, at last, the way that this chance truly opened the entryway into my profession. The most stunning thing about this temporary position experience is that it truly overcame any barrier between learning things and really applying a portion of this information into a reality, AND getting paid for it! The way that I presently have an entry-level position straightforwardly identified with what I concentrate on it makes me study more diligently and makes me need to sort out how I can apply a greater amount of what I'm realizing in an internship. It's interesting, in light of the fact that things I've learned in college help me at work

LITERATURE SURVEY

What is Time Series?

Time-series data is data that has data points that have been cataloged or spliced in time-based order. Time-series data includes the height of the ocean tides, the number of sunspots, and even the daily closing time of stock trades.

Time-series forecasting is one way of formulating a model for predicting future values using current and past time series data. The Time-series is based on four principles:

Trend: an increase or decrease of data; could be linear or nonlinear (logistics growth).

Seasonality: a characteristic of time-series data when it experiences regular and predictable movement after a fixed period of time.

Cyclic: a cyclic pattern exists when the data experience rises or falls (regular or periodic fluctuations in data).

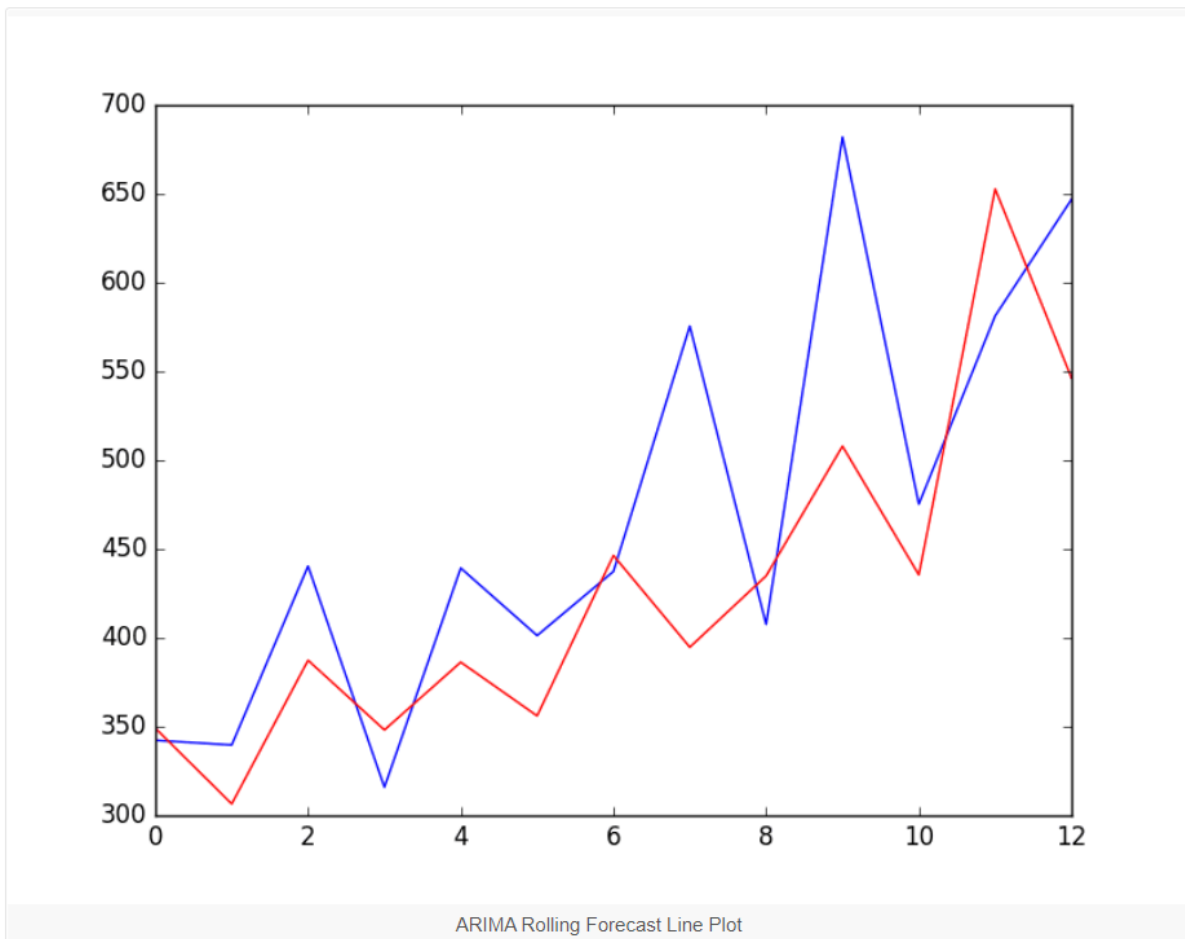
Irregularity: the residual of time series after the trend-cycle and seasonal component are removed.

Models Used:

ARIMA:

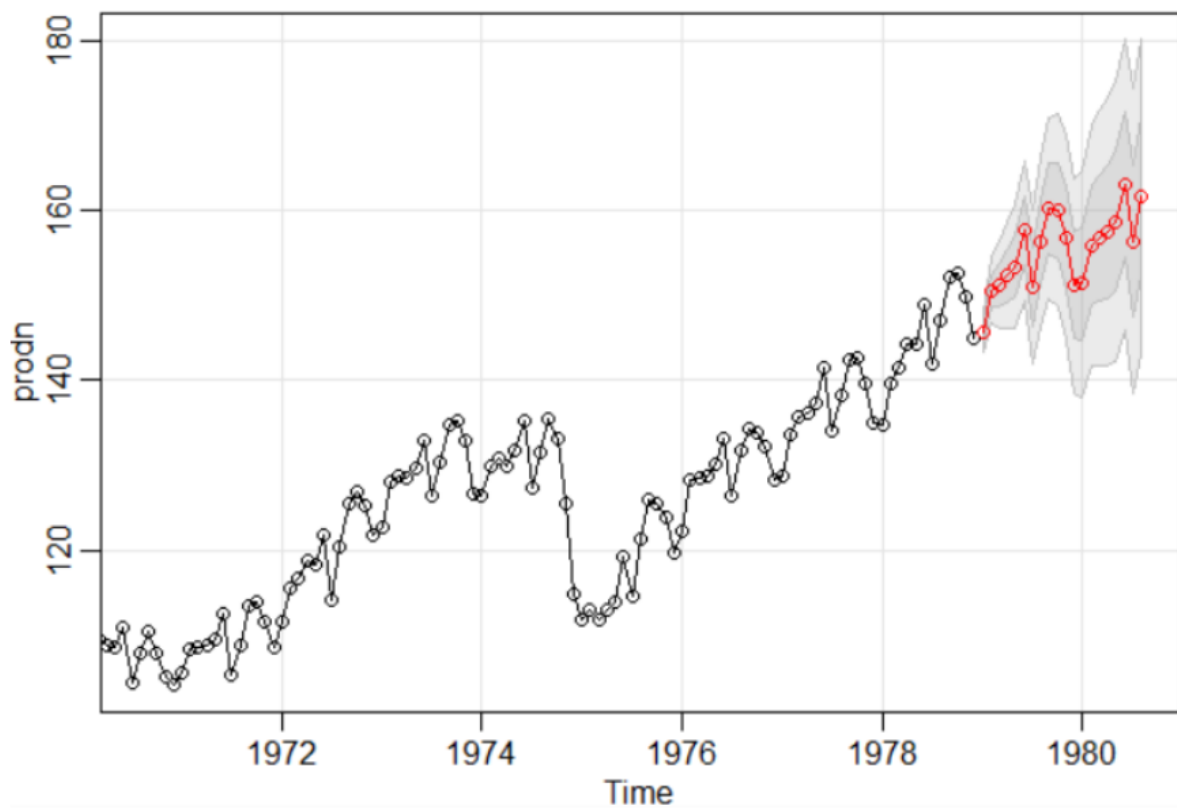
ARIMA is an acronym that stands for autoregressive integrated moving average. It is a statistical analysis approach that tends to make use of time series data to decipher the data set or identify patterns.

Autoregressive statistical models anticipate future values based on prior values. For example, an ARIMA model may try to forecast a company's profitability based on previous periods or to forecast a stock's future price based on historical data.



SARIMA:

Premised on the idea of seasonal trends, a seasonal autoregressive integrated moving average (SARIMA) model derives from an ARIMA model. Seasonal impacts are consistently observed in time series data. Consider the average temperature measured in a four-season area. On an annual basis, there will be a seasonal influence, and the temperature in this season will undoubtedly have a significant association with the temperature observed last year in the same season.

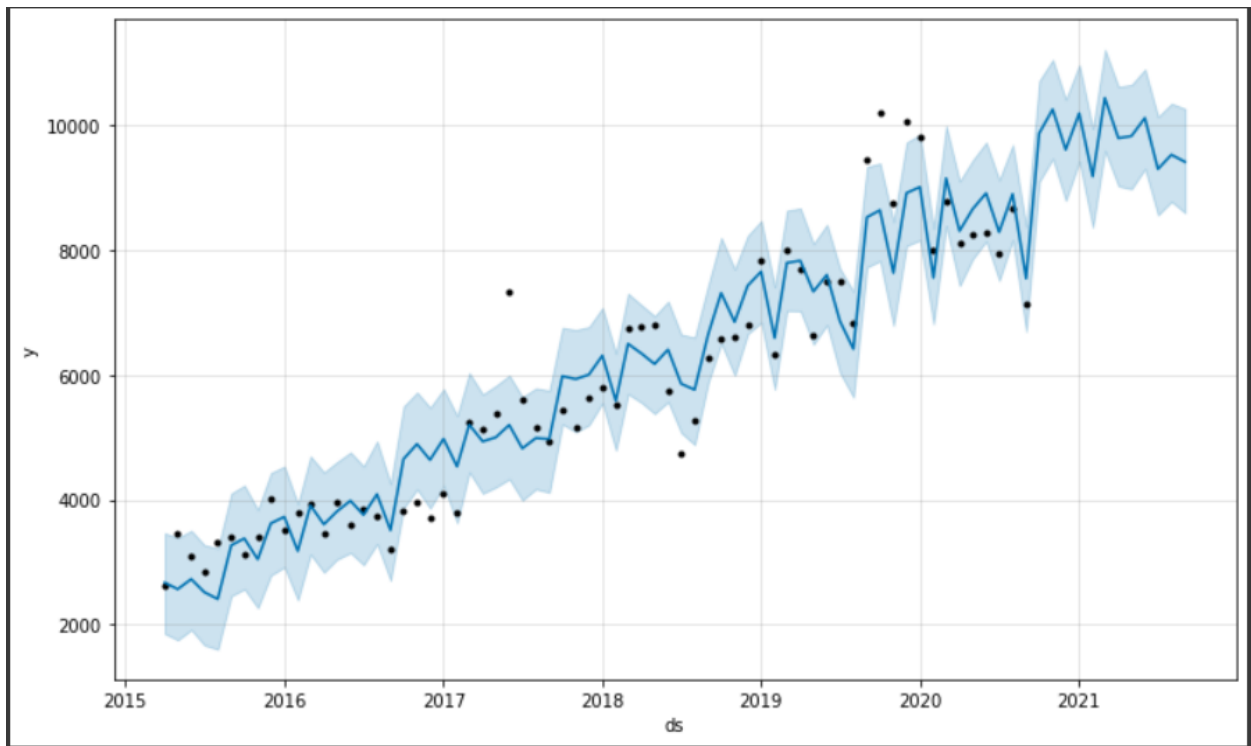


FBProphet:

FBProphet employs a decomposable time series model featuring three major components: seasonal, trends, holidays or events of influence, and error, which are integrated in the following equation:

$$f(x) = g(x) + s(x) + h(x) + e(t)$$

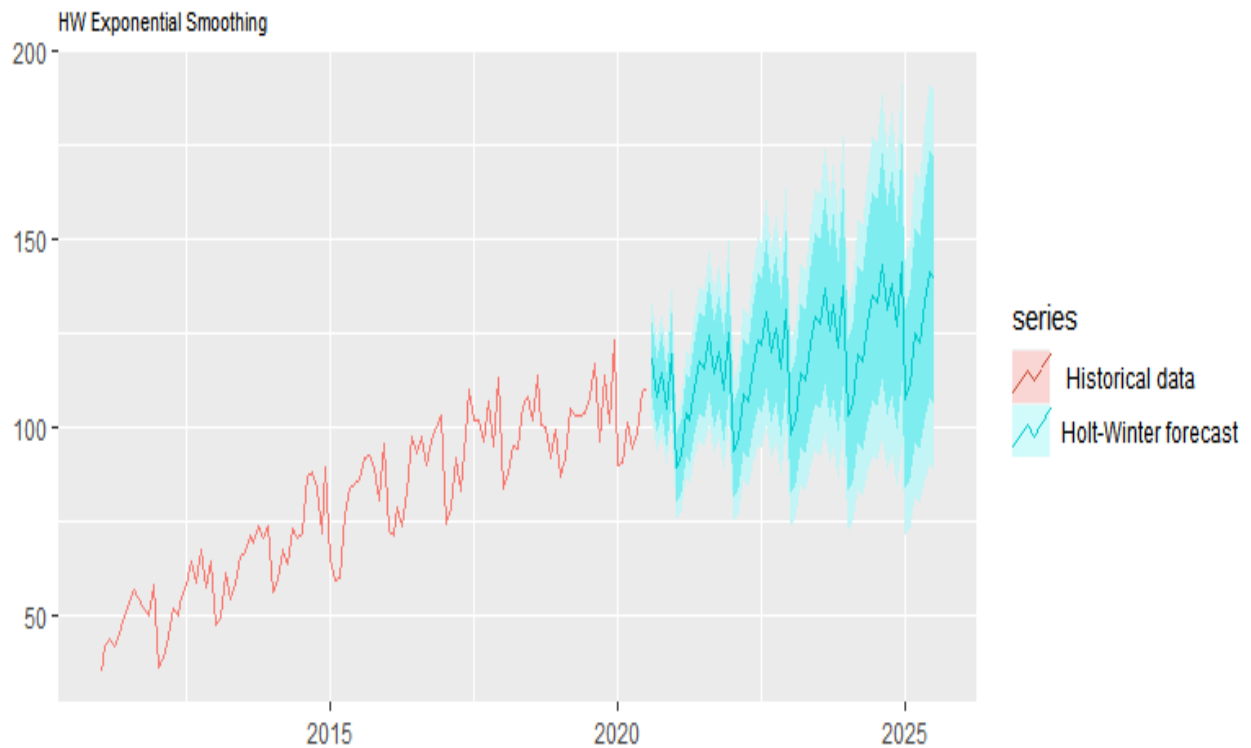
FBProphet employs time as a regressor and attempts to fit many linear and nonlinear time functions as components. FBProphet will fit the data using a linear model by default, but it can be modified to a nonlinear model (logistics growth) using its parameters.



Holt Winters' Triple Exponential Smoothing:

Simple Exponential Smoothing: As the name implies, simple exponential smoothing is used for forecasts when the data set has no characteristics or seasonality.

Holt's smoothing approach: Holt's smoothing technique, also known as linear exponential smoothing, is a well-known smoothing model for anticipating trending data. Winter's Smoothing Method: The Winter's Smoothing Method incorporates seasonality into the prediction along with the trend.



System Development

DataSet :

The dataset contained the revenue generated by clients for a particular organization. Our goal is to forecast the revenue generated by them in the future, or in simple terms, which clients will generate more revenue in the future.

Other than the revenue column, our dataset contained columns such as:

1. ClientCode
2. TLA
3. Biller Information
4. Industry Type
5. Fee Type
6. GoLiveDate
7. State

Fig. 0 shows EDA on the Data Set.

	January 2012	February 2012	March 2012	April 2012	May 2012	June 2012	July 2012	August 2012	September 2012	October 2012
count	263.000000	265.000000	277.000000	274.000000	283.000000	295.000000	303.000000	315.000000	315.000000	324.000000
mean	1721.299468	1850.821283	1831.912708	1739.297664	1821.790565	1682.790000	1725.523168	1736.698508	1653.611746	1857.179198
std	3365.579599	3473.147200	3512.993595	3161.797684	3431.775209	3293.017377	3367.650489	3523.368459	3212.713813	3567.324746
min	-455.920000	-138.960000	-123.390000	-142.410000	-138.770000	-2579.400000	-813.600000	-4668.740000	-1102.350000	-1711.270000
25%	74.375000	91.420000	77.700000	89.700000	81.085000	67.205000	60.905000	53.220000	51.750000	62.777500
50%	387.660000	543.540000	455.730000	438.585000	516.210000	429.470000	394.440000	399.860000	390.220000	380.135000
75%	1544.980000	1655.970000	1625.940000	1731.452500	1661.550000	1601.270000	1573.965000	1733.250000	1537.130000	1940.427500
max	21926.550000	23478.200000	24386.320000	21641.390000	23631.720000	24687.200000	22651.800000	24852.550000	23855.620000	26711.130000

Fig 1 shows the top 10 clients from our dataset.

```
client27->3896565.82  
client7->4558902.94  
client3->4812388.21  
client12->6483359.09  
client11->7145958.62  
client0->11397625.89  
client2->13957048.64  
client5->16027439.01  
client388->16934313.98  
client1->26247815.85
```

Fig 2 shows the bottom 10 clients from our dataset.

```
client1496->-4705977.31  
client1499->-607207.64  
client1489->-587200.72  
client164->-583848.80  
client1492->-384982.41  
client502->-257402.46  
client1497->-249673.03  
client31->-242116.50  
client193->-220912.89  
client45->-189914.95
```

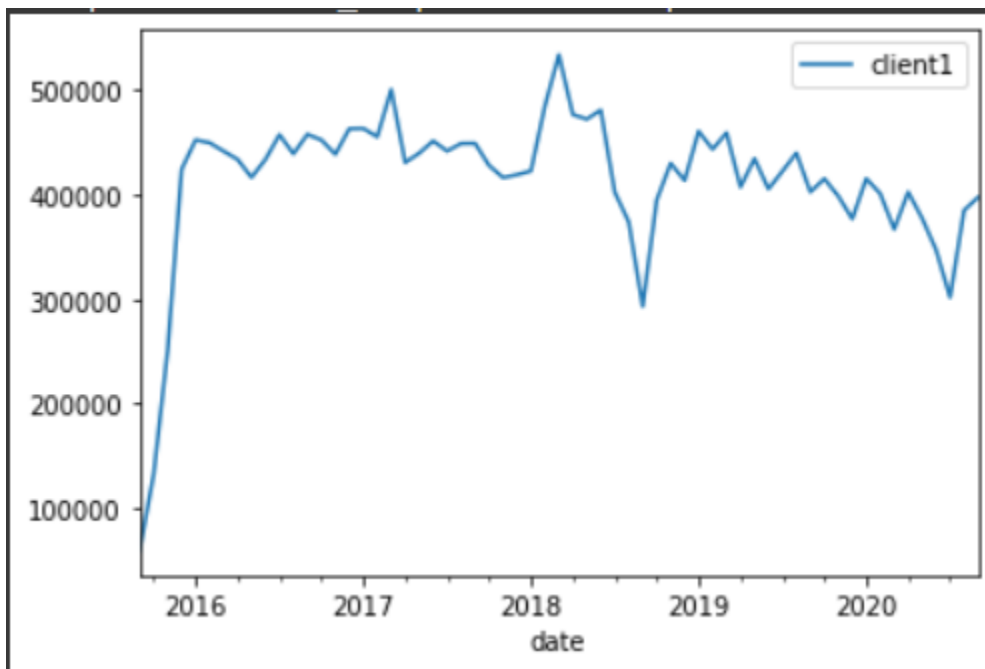
The dataset was non-time indexed data and had missing values and outliers in it.

Fig. 1 shows the first five values and missing values in the dataframe.

	ClientCode	TLA	Biller	Industry	FeeType	GoLiveDate	State	January 2012	February 2012	March 2012	April 2012	May 2012	June 2012	July 2012	August 2012
0	8067656700	CAC	Credit Acceptance	Financial Services	Convenience Fee	2/1/2018	Michigan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	8067808373	CPSI	Consumer Portfolio Services	Financial Services	Hybrid	9/1/2015	California	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	8067697773	CEMI	Consumers Energy	Utilities	Absorbed	2/1/2017	Michigan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	8067696600	CEB	Constellation Utility	Utilities	Absorbed	7/1/2017	Wisconsin	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	8079716583	OGAS	ONE Gas	Utilities	Hybrid	12/1/2019	Texas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows x 112 columns

Fig 2 shows the outliers in the data set.

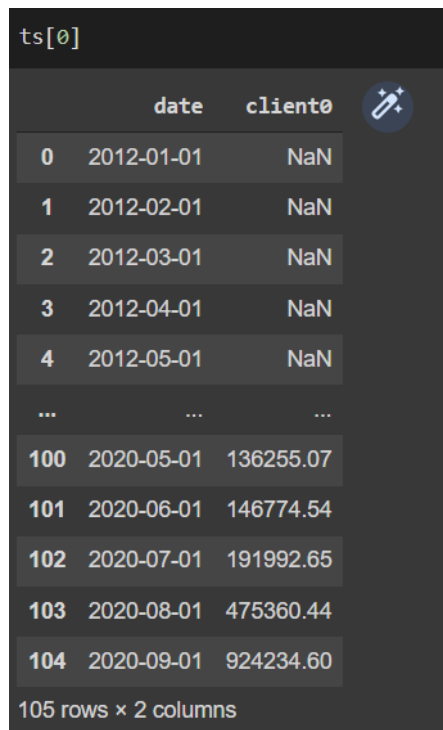


Converting to Time Series Data:

Steps for conversion:

1. Convert the column names from the range of Jan 2012 to Sept 2022 into the form of datetime, like 2012-01-01 to 2020-09-01 on a monthly basis.
2. Since the data didn't have all the client codes, we created our own column containing numbers from 0 to 1499, representing all the clients.
3. Extracting every client's individual time series, converting it to a dataframe with a column named as "client no." and placing it in a dictionary (ts), also removed clients based on the last 5 months of data.

Figure 3 depicts the first client dataframe.



The image shows a screenshot of a Jupyter Notebook interface. At the top, the text 'ts[0]' is displayed. Below it is a pandas DataFrame with two columns: 'date' and 'client0'. The 'date' column contains dates from 2012-01-01 to 2020-09-01 in monthly increments. The 'client0' column contains numerical values, with the first five rows showing 'NaN' and subsequent rows showing values like 136255.07, 146774.54, 191992.65, 475360.44, and 924234.60. The DataFrame is shown with 105 rows and 2 columns.

	date	client0
0	2012-01-01	NaN
1	2012-02-01	NaN
2	2012-03-01	NaN
3	2012-04-01	NaN
4	2012-05-01	NaN
...
100	2020-05-01	136255.07
101	2020-06-01	146774.54
102	2020-07-01	191992.65
103	2020-08-01	475360.44
104	2020-09-01	924234.60

105 rows × 2 columns

Preprocessing for Outlier Detection:

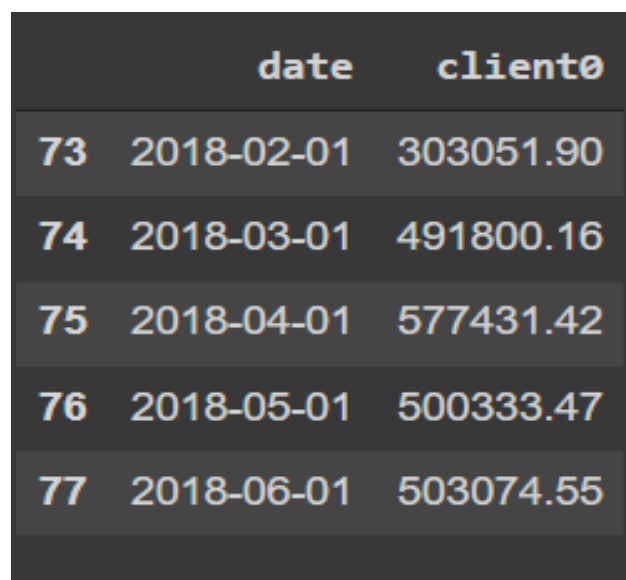
Every time series had a different starting point, but our dataframe was started from 2012-01-01 for every series, so to handle those missing values, we dropped the rows till we encountered the first billing month.

After removing the first NaN values from the time series, we would like to handle the missing values that were in between the new time series. For that, we used the interpolation approach to fill in those missing values.

What is interpolation?

Interpolation is typically utilized when engaging with time-series data because we want to substitute incomplete data with the prior one or two values in time-series data. For example, if it's temperature, we'd rather populate today's temperature with the mean of the past two days than the mean of the month. Moving averages can also be calculated via interpolation.

Fig 4 shows Client 0 after preprocessing.



	date	client0
73	2018-02-01	303051.90
74	2018-03-01	491800.16
75	2018-04-01	577431.42
76	2018-05-01	500333.47
77	2018-06-01	503074.55

Outlier Detection:

There are many methods for outlier detection in time series analysis. Here we used two of them:

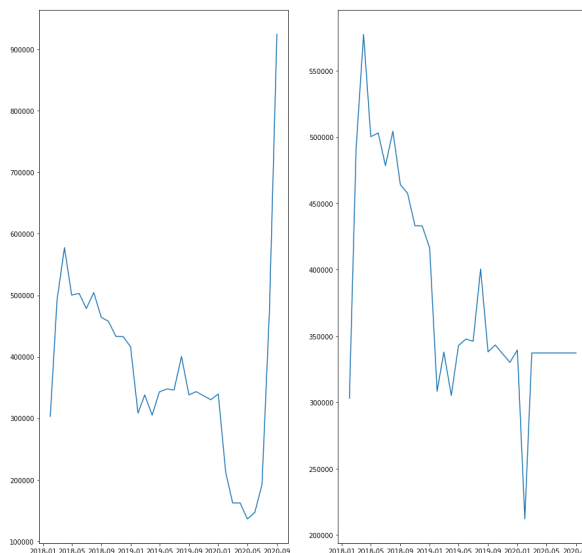
1. Hampel Filter
2. Outlier Detection

Hampel Filter:

The Hampel filter is utilised to find oddities in time series data. It is primarily a scrolling window with a parameterizable size. Each interpretation will be analyzed towards the Median Absolute Deviation for each window (MAD). If the observation exceeds the MAD by n fold, it will be labeled an abnormality (the parameter n is also parameterizable).

It also allocates the quantity of the mean absolute deviation to the abnormality.

Fig 5 shows the client 0 before and after outlier detection.



InterQuartile Range(IQR):

The interquartile range is an evaluation of where a data set's "middle fifty" is sited. Whereas a range is a metric of where the upper and lower limits of a set are, an interquartile range is a representation of where the large percentage of the readings are. That is why, when posting stuff like academic results or SAT scores, it is recommended above most other measures of dispersion.

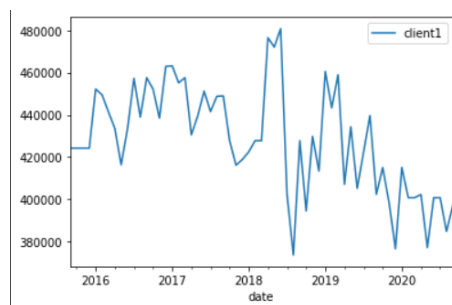
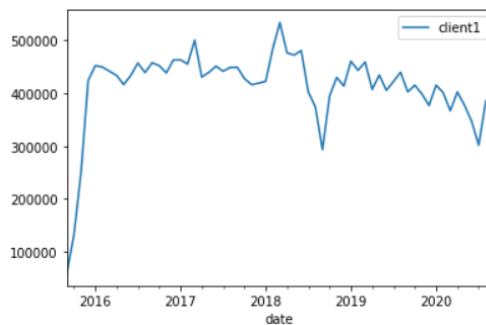
The interquartile range is computed by deducting the first quartile from the third quartile

$$\text{IQR} = \text{Q3} - \text{Q1}$$

$$\text{Upper Limit} = \text{Q3} + 1.5 * \text{IQR}$$

$$\text{Lower Limit} = \text{Q1} - 1.5 * \text{IQR}$$

Fig. 6 shows client 1 before and after the IQR method.



Clustering Methods Used:

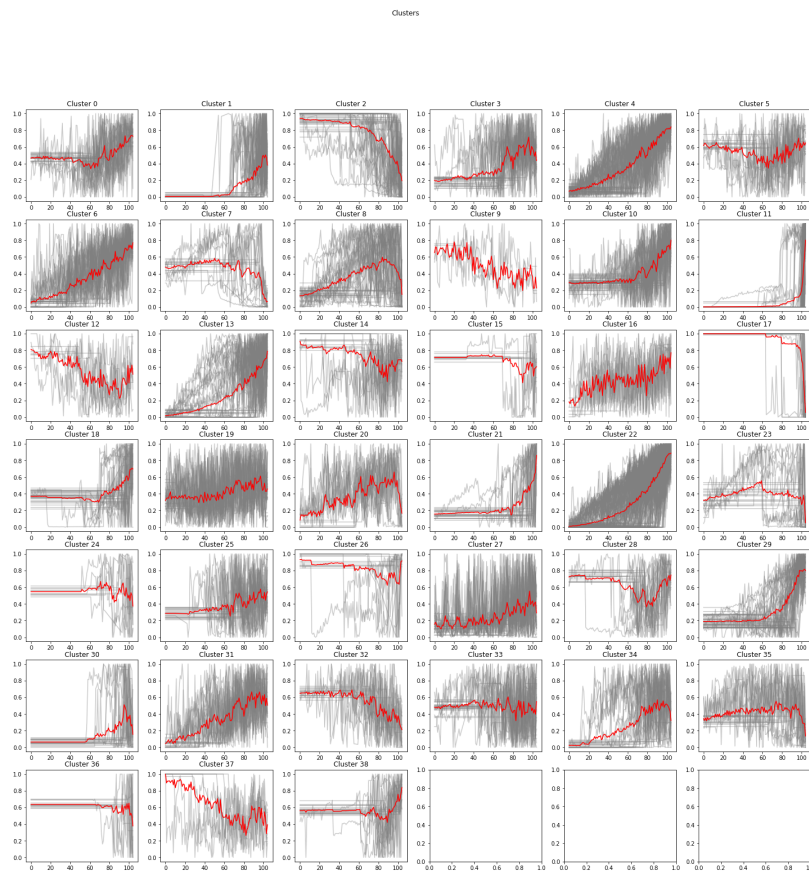
1. K-means
2. Self-organizing maps
3. Grouping the data based on industry type and then using a clustering algorithm on it

K means clustering:

K-means Clustering is a process for assembling n entries into k clusters, for each data point referring to the region with the least mean (cluster centroid). It is one of the most common and fundamental clustering techniques, and it may be shown as Voronoi cells.

We created 38 clusters using k means.

Fig. 8 shows the clusters created using K Means.

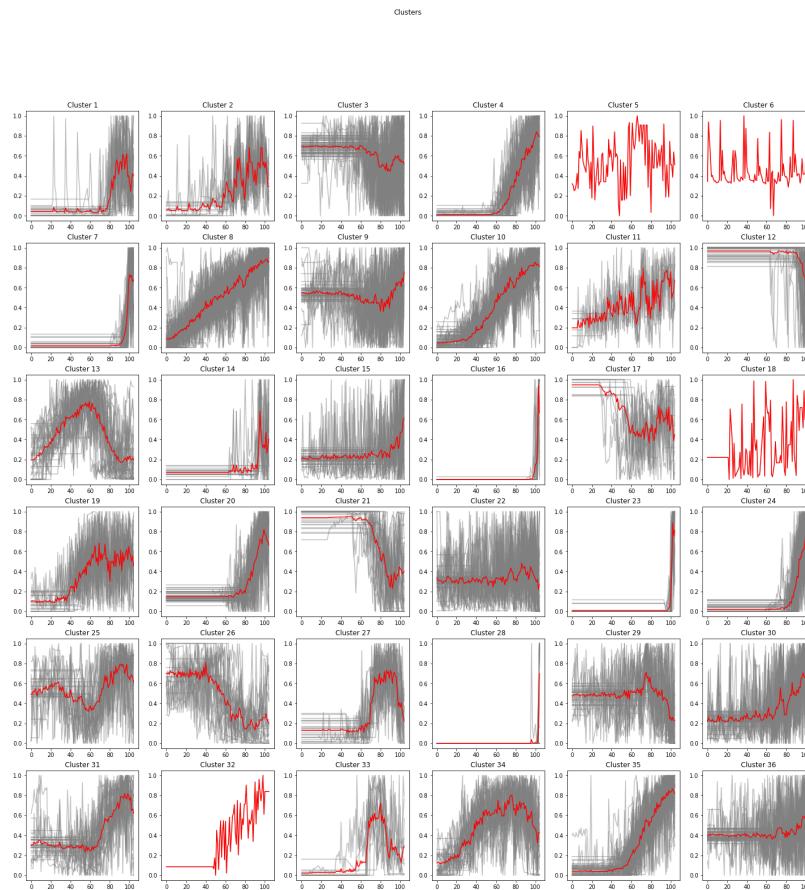


Self Organizing Maps:

Self-organizing maps are a sort of neural network that is trained via unsupervised learning to generate a map, which is a low-dimensional representation of the training samples' input space.

Furthermore, self-organizing maps differ from other artificial neural networks in that they use competitive (or cooperative) learning rather than error-correction learning (such as backpropagation with gradient descent), and they use a neighborhood function to preserve the topological properties of the input space.

Fig. 9 shows clusters created by SOM.



Grouping the data based on industry type:

In this method, we first group the data according to the industry type they belong to, and after that, we do the clustering for individual groups.

Groups:

1. ***Accountants***
2. ***County And Municipal Taxes***
3. ***City Services***
4. ***Financial Services***
5. ***Government Services***
6. ***Health Care***
7. ***Insurance***
8. ***Miscellaneous***
9. ***Telecommunications***
10. ***Transportation***
11. ***Utilities***

Without Clustering Approach:

The core concept behind this mechanism is to establish a time series for each client because the clustering algorithm is ineffective due to the limited amount of data.

So we generated fbprophet, ARIMA, SARIMA, and Holt's winter exponential model for each client, and this is the final strategy we will employ because it provides a better match to the data than the clustering approach.

Prophet Model:

FBProphet employs a decomposable time series model featuring three major components: seasonal, trends, holidays or events of influence, and error, which are integrated in the following equation:

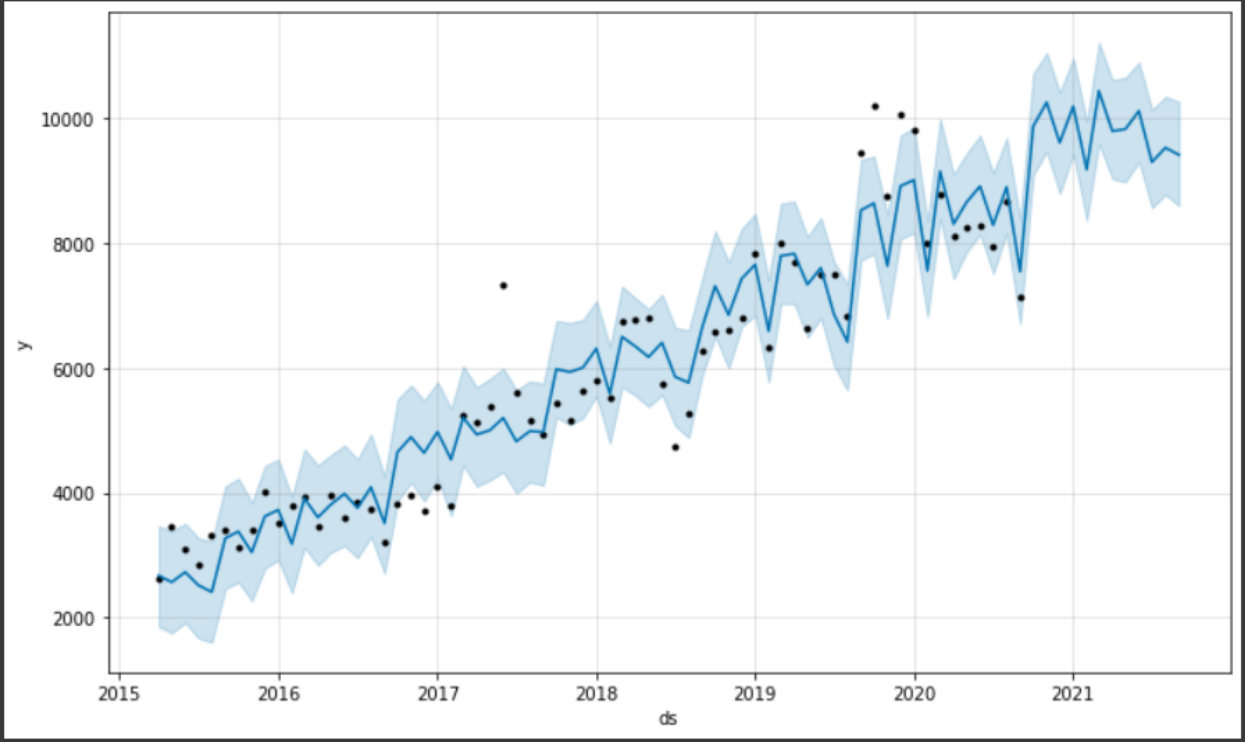
$$f(x) = g(x) + s(x) + h(x) + e(t)$$

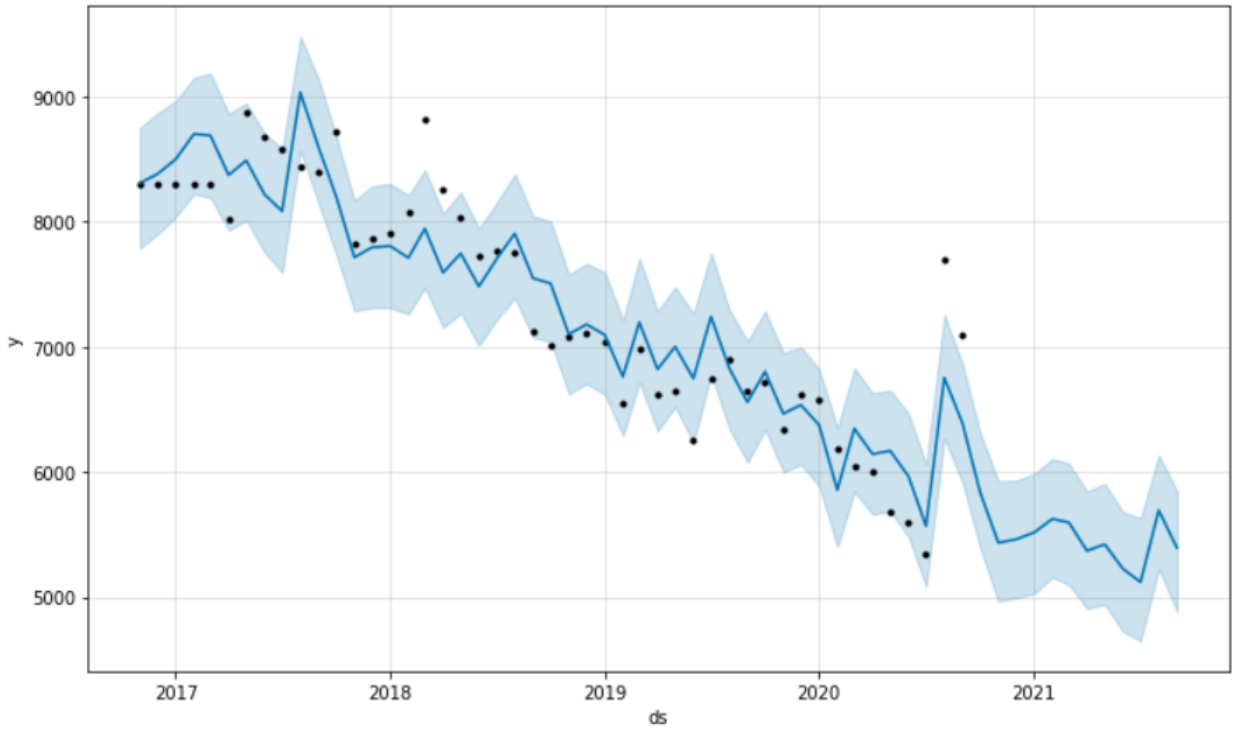
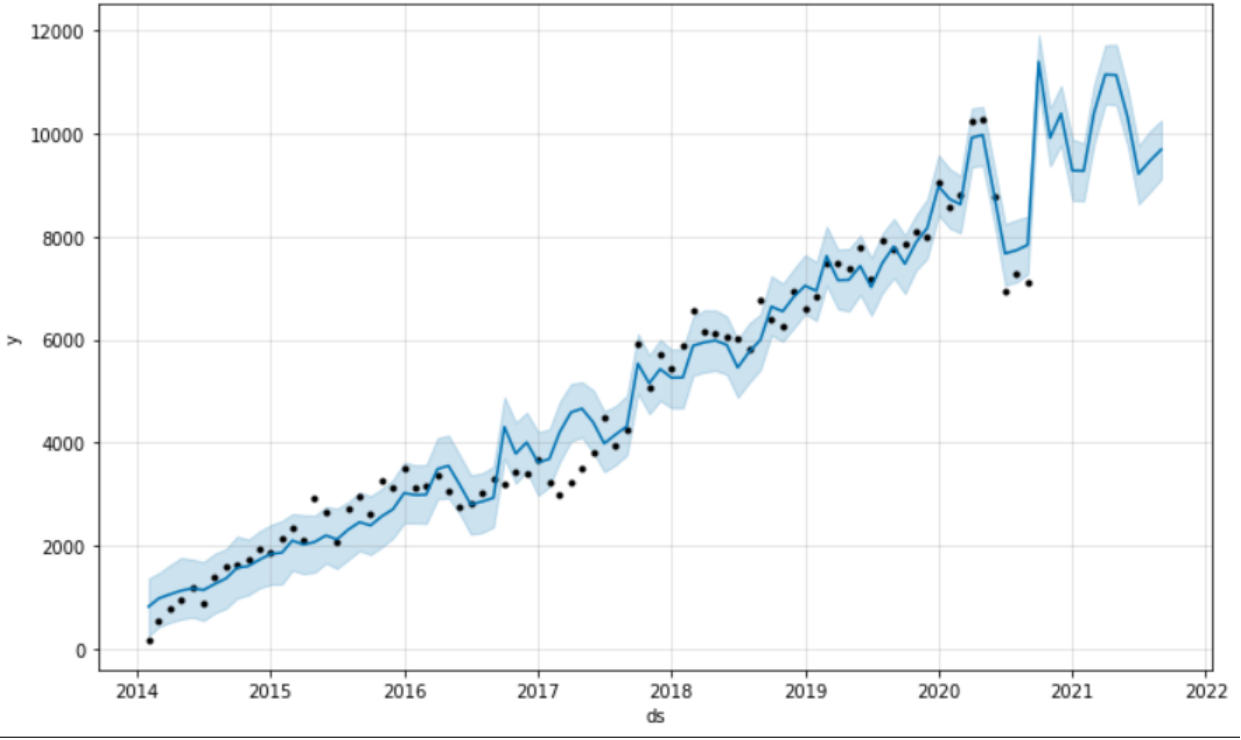
FBProphet employs time as a regressor and attempts to fit many linear and nonlinear time functions as components. FBProphet will fit the data using a linear model by default, but it can be modified to a nonlinear model (logistics growth) using its parameters.

Tuned parameters:

1. n_changepoints
2. change_point_prior scale

Fig. 10: Prophet model results





ARIMA Model:

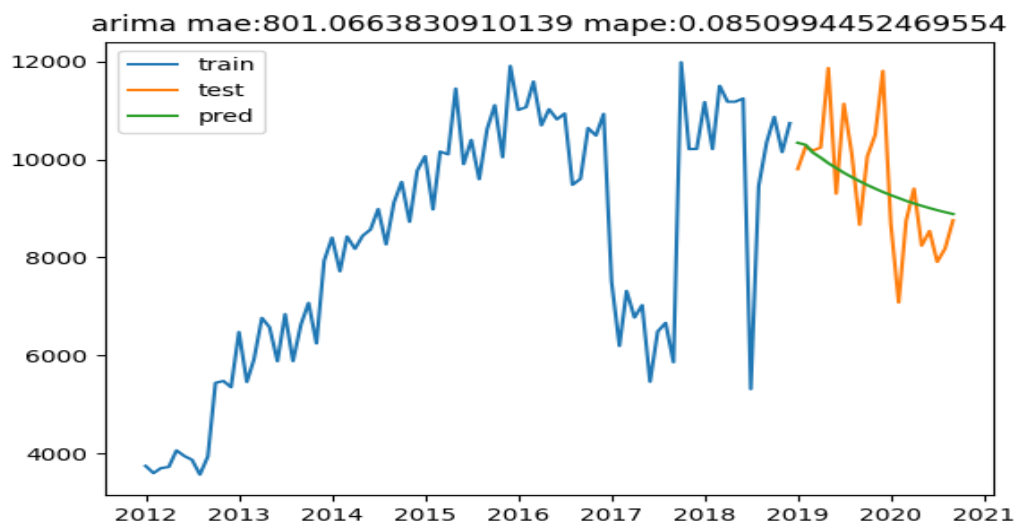
ARIMA is an acronym that stands for autoregressive integrated moving average. It is a statistical analysis approach that tends to make use of time series data to decipher the data set or identify patterns.

Autoregressive statistical models anticipate future values based on prior values. For example, an ARIMA model may try to forecast a company's profitability based on previous periods or to forecast a stock's future price based on historical data.

Tuned parameters:

1. **p**: the number of lag observations in the model; also known as the lag order
2. **d**: the number of times that the raw observations are differenced; also known as the degree of differencing
3. **q**: the size of the moving average window; also known as the order of the moving average.

Fig. 11. ARIMA model results for one customer



Holt Winters' Model:

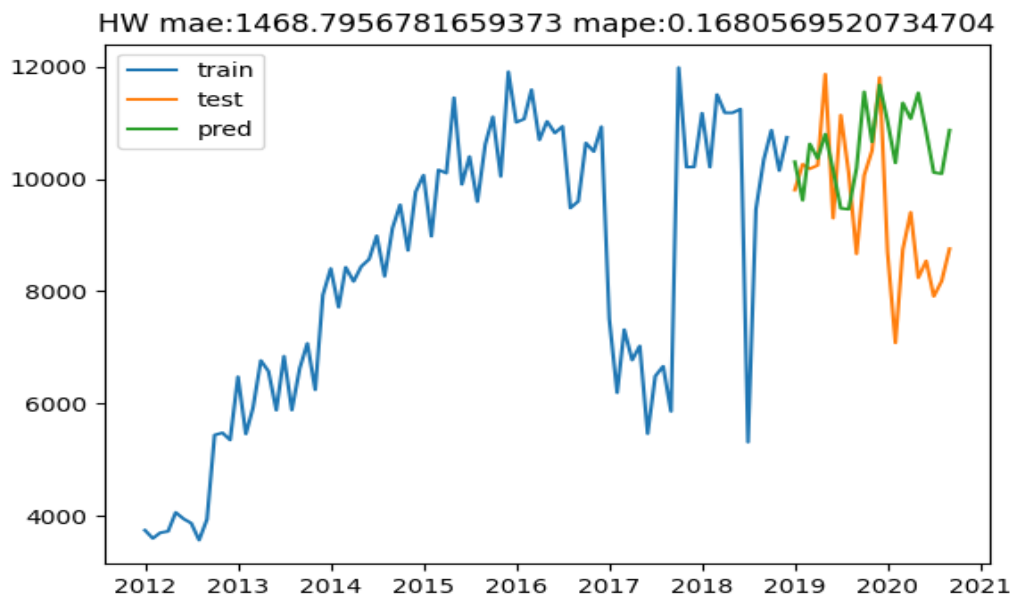
Simple Exponential Smoothing: As the name implies, simple exponential smoothing is used for forecasts when the data set has no characteristics or seasonality.

Holt's smoothing approach: Holt's smoothing technique, also known as linear exponential smoothing, is a well-known smoothing model for anticipating trending data. Winter's Smoothing Method: The Winter's Smoothing Method incorporates seasonality into the prediction along with the trend.

Tuned parameters:

1. Trend
2. Seasonal_periods

Fig.12: Holt Winters' Model results for one customer



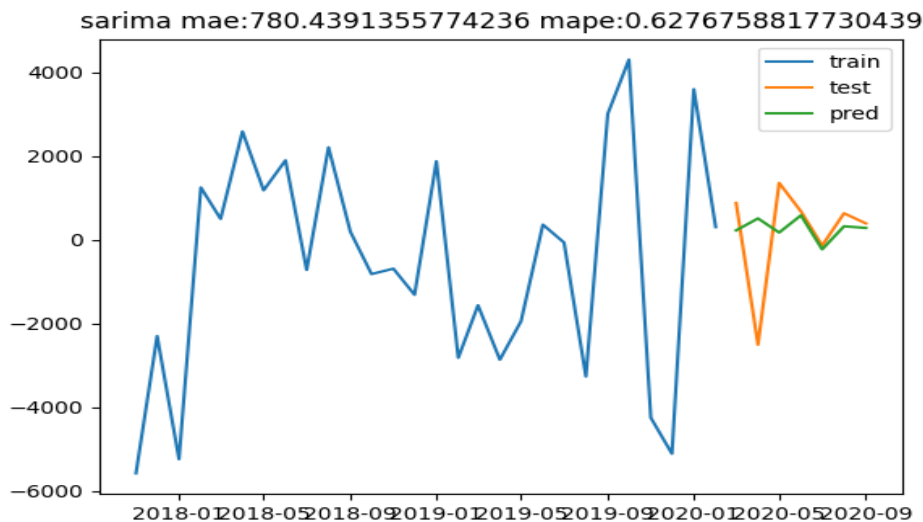
SARIMA:

Premised on the idea of seasonal trends, a seasonal autoregressive integrated moving average (SARIMA) model derives from an ARIMA model. Seasonal impacts are consistently observed in time series data. Consider the average temperature measured in a four-season area. On an annual basis, there will be a seasonal influence, and the temperature in this season will undoubtedly have a significant association with the temperature observed last year in the same season.

Tuned parameters:

1. **p**: Trend autoregression order.
2. **d**: Trend difference order.
3. **q**: Trend moving average order.
4. **P**: Seasonal autoregressive order.
5. **D**: Seasonal difference order.
6. **Q**: Seasonal moving average.
7. **m**: The number of time steps for a single seasonal period.

Fig.12: SARIMA model results for one customer.

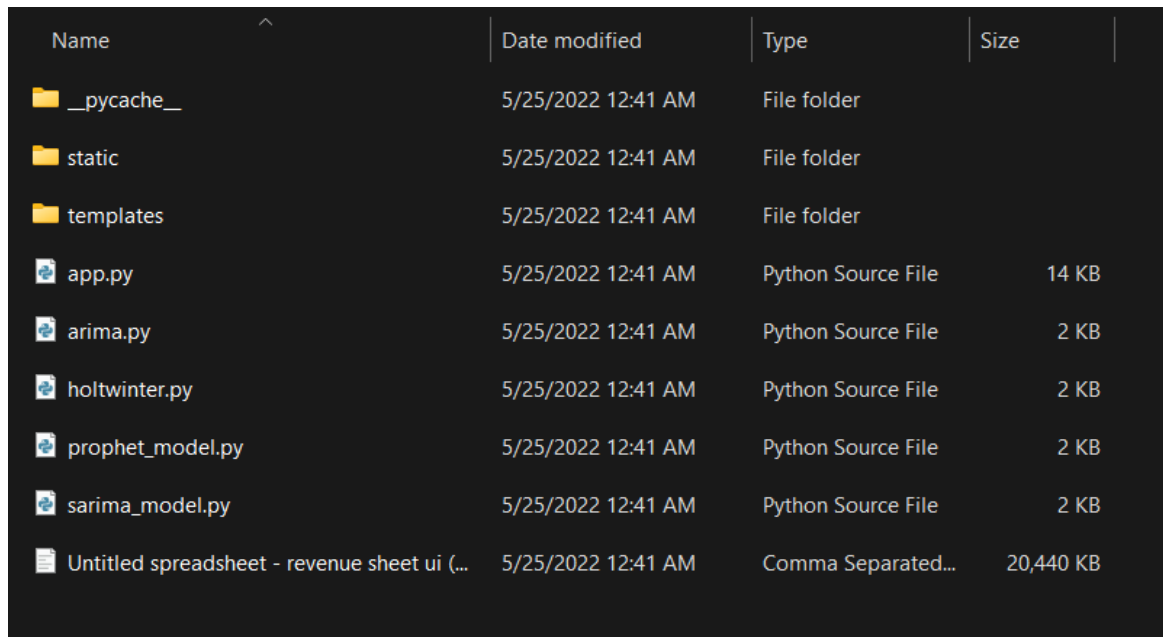


Flask Development:

Flask is a Python-based web application framework. It was created by Armin Ronacher, who heads the Pocco worldwide network of Python aficionados. Flask is built with the Werkzeug WSGI toolkit and the Jinja2 template engine. Both are Pocco initiatives.

I chose Flask as the backend framework as it is lightweight, easy to use, and is open source.

Fig. flask structure



The image shows a file explorer window with a dark theme. It displays a list of files and folders for a Flask project. The columns are 'Name', 'Date modified', 'Type', and 'Size'. The files listed are: __pycache__ (File folder), static (File folder), templates (File folder), app.py (Python Source File, 14 KB), arima.py (Python Source File, 2 KB), holtwinter.py (Python Source File, 2 KB), prophet_model.py (Python Source File, 2 KB), sarima_model.py (Python Source File, 2 KB), and Untitled spreadsheet - revenue sheet ui (... (Comma Separated..., 20,440 KB).

Name	Date modified	Type	Size
__pycache__	5/25/2022 12:41 AM	File folder	
static	5/25/2022 12:41 AM	File folder	
templates	5/25/2022 12:41 AM	File folder	
app.py	5/25/2022 12:41 AM	Python Source File	14 KB
arima.py	5/25/2022 12:41 AM	Python Source File	2 KB
holtwinter.py	5/25/2022 12:41 AM	Python Source File	2 KB
prophet_model.py	5/25/2022 12:41 AM	Python Source File	2 KB
sarima_model.py	5/25/2022 12:41 AM	Python Source File	2 KB
Untitled spreadsheet - revenue sheet ui (...	5/25/2022 12:41 AM	Comma Separated...	20,440 KB

The API has three decorator functions:

1. **helloworld()**: this function renders our index template.
 - a. Input: NA
 - b. Working: Renders the index.html template from our templates folder.
 - c. Output: index.html

PAXCOM Forecasting System

Upload File

Choose File 8080676688.csv

Phone_number

Date Column

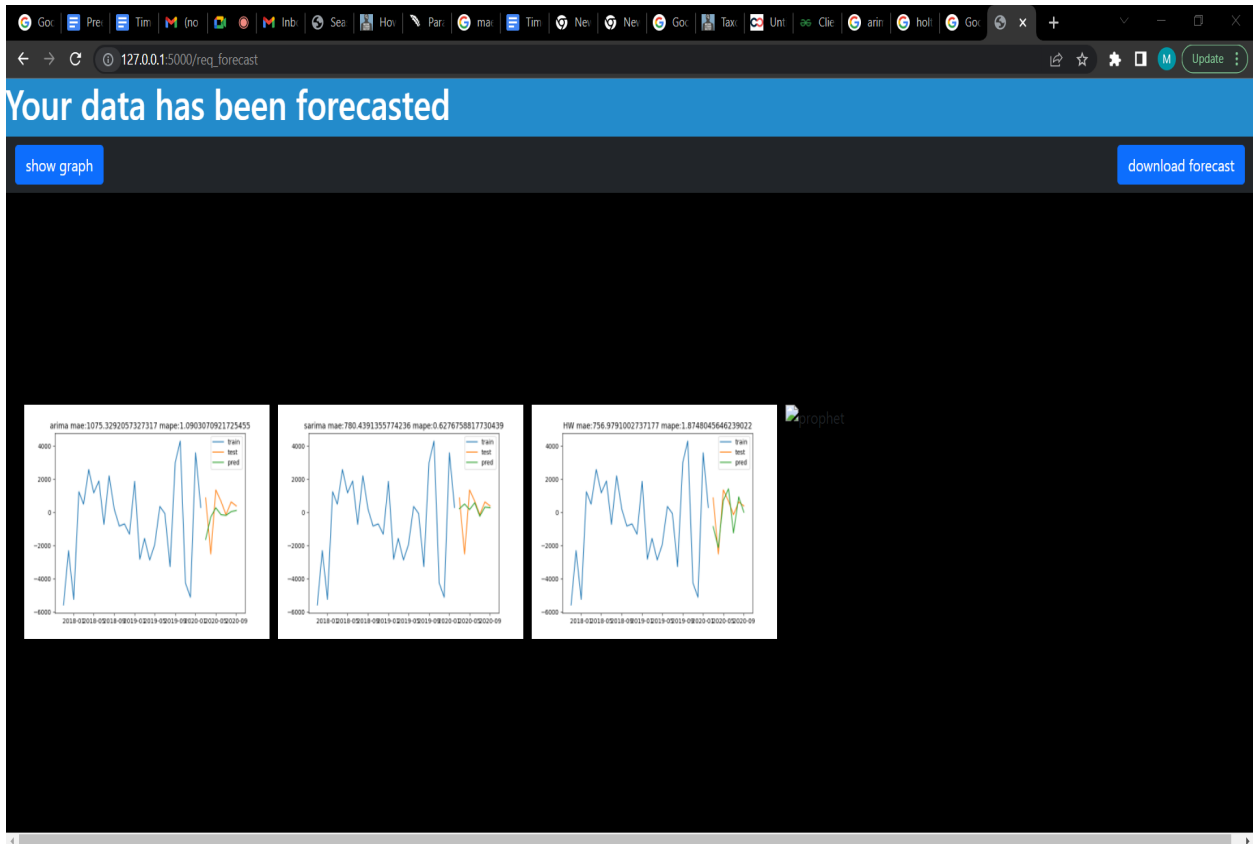
Observation Column name

Select Algorithms

ARIMA SARIMA
 Prophet Holt winter






index.html

2. **forecast()**: this function calls in all the models' functions and handles all the preprocessing for the forecasting.
 - a. Input: Form elements.
 - b. Working: Preprocessing, model training, and forecasting.
 - c. Output: Forecast.csv, graphs of all the models.



Options.html

3. **downloadforecast()**: this function serves the zip folder containing forecast.csv, test.csv, and graphs of all the selected models.
 - a. Input: NA
 - b. Working: Serves the folder to the client.
 - c. Output: timestamped zip folder

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
 arima_graph.png	PNG File	43 KB	No	45 KB	6%	5/25/2022 1:31 AM
 forecast.csv	Comma Separated Values ...	1 KB	No	1 KB	44%	5/25/2022 1:33 AM
 hw_graph.png	PNG File	44 KB	No	47 KB	6%	5/25/2022 1:32 AM
 sarima_graph.png	PNG File	43 KB	No	46 KB	6%	5/25/2022 1:32 AM
 test.csv	Comma Separated Values ...	1 KB	No	1 KB	61%	5/25/2022 1:33 AM

Forecast.zip folder

Contents of Forecast.zip folder:

1. Forecast.csv:

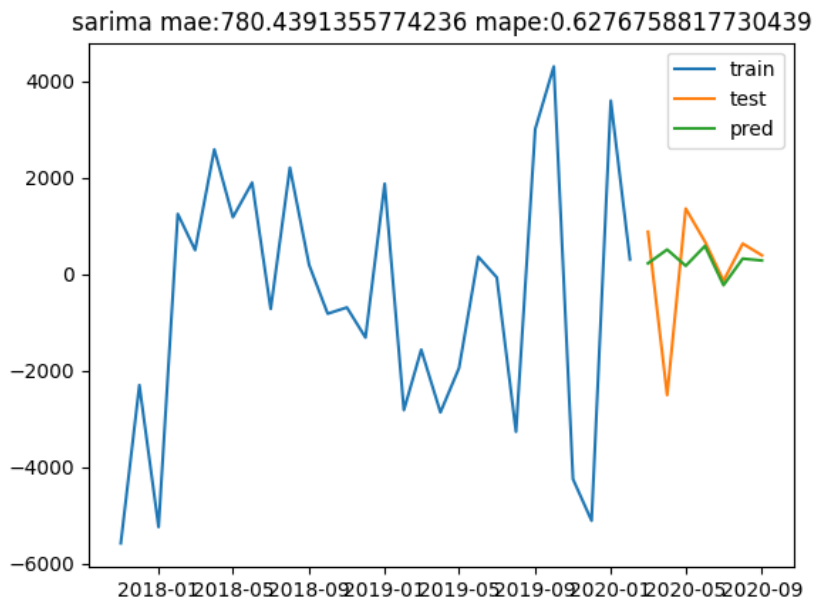
A	B
	revenue
2020-10-01	-570.0516926
2020-11-01	-1999.510172
2020-12-01	-2041.620445
2021-01-01	1379.435941
2021-02-01	-1808.023856
2021-03-01	-807.4881644

+ ☰ forecast ▼

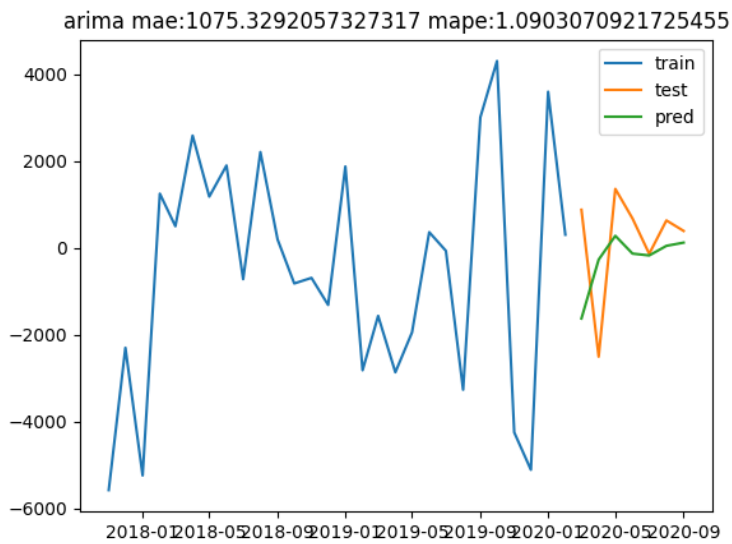
2. Test.csv:

A	B	C	D	
	revenue	original	Mean Absolute Error	
2020-03-01	-857.9281916	877.6087742	756.9791003	
2020-04-01	-2119.056121	-2509.436914	756.9791003	
2020-05-01	721.6385411	1358.770337	756.9791003	
2020-06-01	1426.524873	674.8175879	756.9791003	
2020-07-01	-1237.288359	-138.2651613	756.9791003	
2020-08-01	936.4968678	631.7742545	756.9791003	
2020-09-01	9.091175986	389.442227	756.9791003	

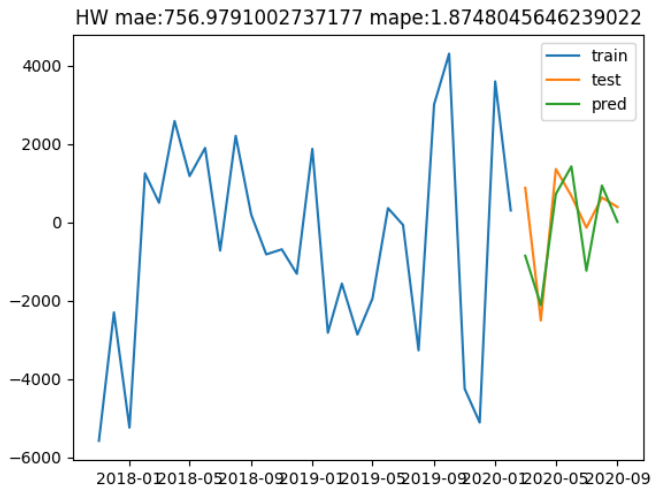
3. SARIMA graph:



4. ARIMA graph:



5. Holt Winters' graph:



Conclusion:

Without clustering approach, it works better most of the time, but the data still has some time series that have very low data points. For such short-term series, forecasting will not be that efficient, and API makes it very flexible to use.

References

ARIMA: https://www.researchgate.net/publication/328633706_Forecasting_of_demand_using_ARIMA_model

SARIMA: <https://ieeexplore.ieee.org/document/6676239>

FBProphet: <https://facebook.github.io/prophet/>

Flask: <https://flask.palletsprojects.com/en/2.1.x/>

K Means: <https://ieeexplore.ieee.org/document/5453745>

SOM: <https://ieeexplore.ieee.org/document/58325>

HW:

https://www.researchgate.net/publication/311945797_Revenue_Forecasting_using_Holt-Winters_Exponential_Smoothing

Kaggle: <https://www.kaggle.com/>

Analytics Vidhya:

<https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/#:~:text=Synopsis%20of%20Time%20Series%20Analysis&text=A%20time%20series%20is%20an,variable%20to%20predict%20the%20results.>