

## **Hierarchical Forecasting Using Machine Learning**

This MS Project report is submitted to the Department of Computer Science as partial fulfillment of Master of Science in Computer/Data Science degree

by

**Sehrish Muhammad**

Supervised by  
**Dr. Tariq Mahmood**  
Assistant Professor  
Department of Computer Science  
School of Mathematics and Computer Science (SMCS)  
Institute of Business Administration (IBA), Karachi

Fall 2021  
Institute of Business Administration (IBA), Karachi, Pakistan



## **Hierarchical Forecasting Using Machine Learning**

This MS Project report is submitted to the Department of Computer Science as partial fulfillment of Master of Science in Computer/Data Science degree

by

**Sehrish Muhammad**  
(ERP ID: 18150)

**Supervisor:**

**(Dr./Mr.) First name Last name**      Dr. Tariq Mahmood

**Designation**      Assistant Professor

School of Mathematics and Computer Science (SMCS)  
Institute of Business Administration (IBA), Karachi

Fall 2021  
Institute of Business Administration (IBA), Karachi, Pakistan

Copyright: **2021, Sehrish Muhammad**  
All Rights Reserved

## **Dedication**

This report is dedicated to my beloved family members, who have encouraged me with their undivided attention (devotion) to complete my work with true self-confidence.

## **Acknowledgment**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am grateful to Dr. Tariq for his continued belief in my abilities.

Furthermore, would like to express my gratitude towards my parents & member of office colleagues for their kind co-operation and encouragement which help me in the completion of this project.

I would also like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

# Table of Contents

Introduction	7
Core Functionalities	7
Design Methodology	8
Implementation Details	15
Impact	17
References	18

## **Introduction**

In this project we have developed an API for ingesting time-series data and generating baseline forecasts against it. To make forecasts, the user can choose from a list of models. The framework at the backend provides a basic skeleton for a codebase that may be customized to meet the demands of data scientists.

Because many types of data from different domains can exist, it's crucial to note that we've mostly concentrated on time-series data with hierarchies. This framework will not only be able to support commonly used forecasting libraries but will also support out-of-the-box methods that have been proposed/implemented to improve accuracy and performance.

## **Core Functionalities**

The solutions implemented are primarily focused on how to best manage business use-cases in which forecasts must be aggregated or disaggregated across multiple levels in data, which can have a direct impact on the final outcomes and major business decisions.

A single dataset was used to build this skeleton, however, the goal is that the framework will be able to offer results for any dataset that matches the requisite data format and structure.

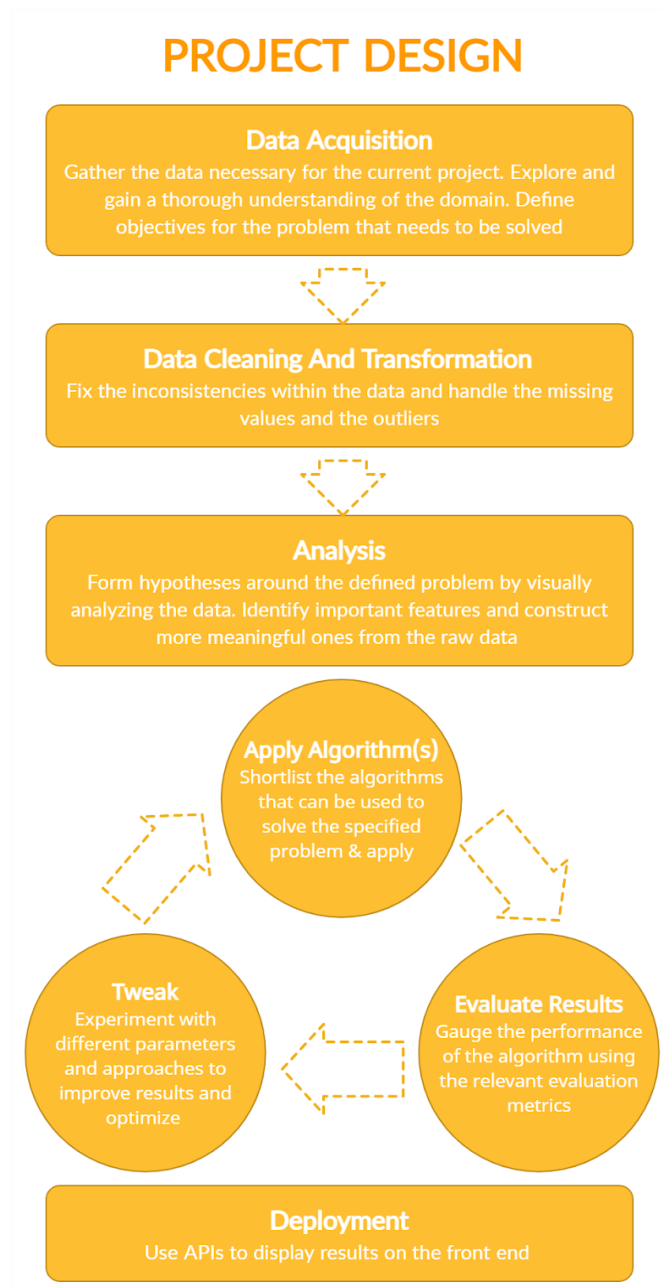
The user is not exposed to the complex workings taking place behind the scenes, he/she is only given an API endpoint through which the data is acquired and forecasts are generated.

The user is also prompted to pass a model name that will be run against the given dataset. The API triggers computation and displays the end result which includes the model name for the model that was run and error (RMSE) so the user can gauge how the model performed.



## Design Methodology

To build the framework we needed a dataset that could serve as a testing ground, for this purpose we chose the Kaggle dataset from Walmart. Design diagram can be viewed below:



## **Data acquisition and exploration**

Like any other data science use case, we went through the process of getting a good grasp of the dataset. This was not made part of the pipeline because during analysis approaches may vary a lot depending on the data you're working with. We did the analysis in a notebook which can be used for reference or future analysis exercises, idea was to create methods for plotting basic charts that can be re-used.

## **Data pre-processing**

For this particular use-case we generated benchmark forecasts on raw data itself. However, we do have a placeholder in our codebase where we may build on various data cleaning and preparation techniques like feature engineering & denoising, etc.

Some of the techniques we included are:

## **Feature Engineering**

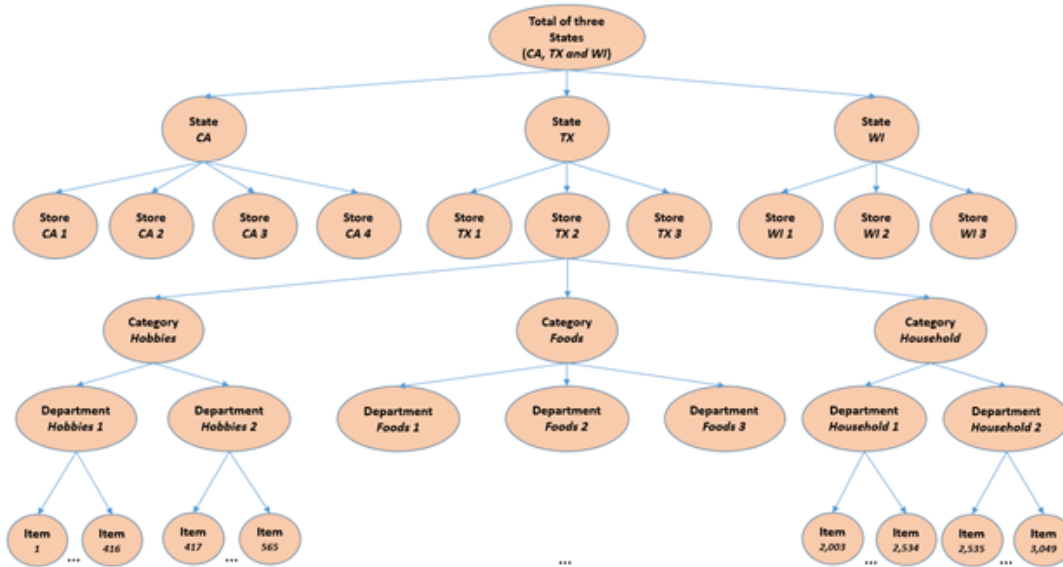
- Basic features like the week of the month, week of the year, day, year, month, is week flag, etc.
- Rolling means in the specified window
- Lags or shifts

## Analysis

Because we intended the framework to be focused on facilitating data conversions and multiple forecasting methodologies, we did the analysis outside of the pipeline

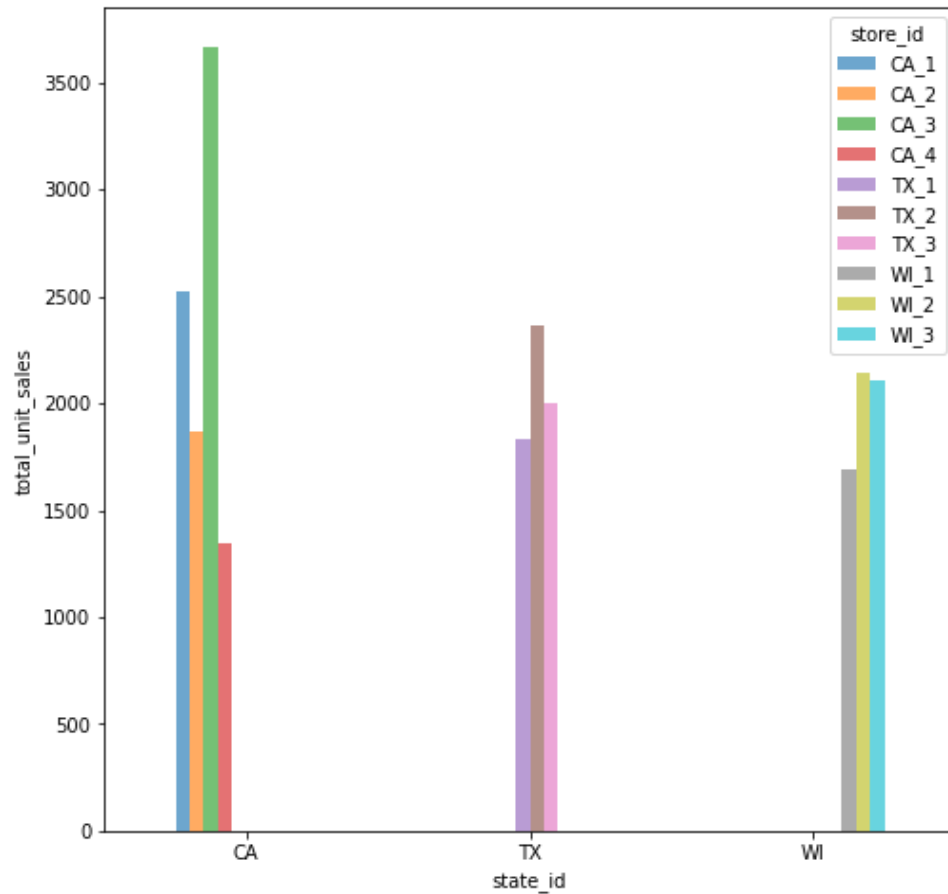
### Short Summary of the analysis

1. From this exercise, we could see that the dataset follows the hierarchy given below:



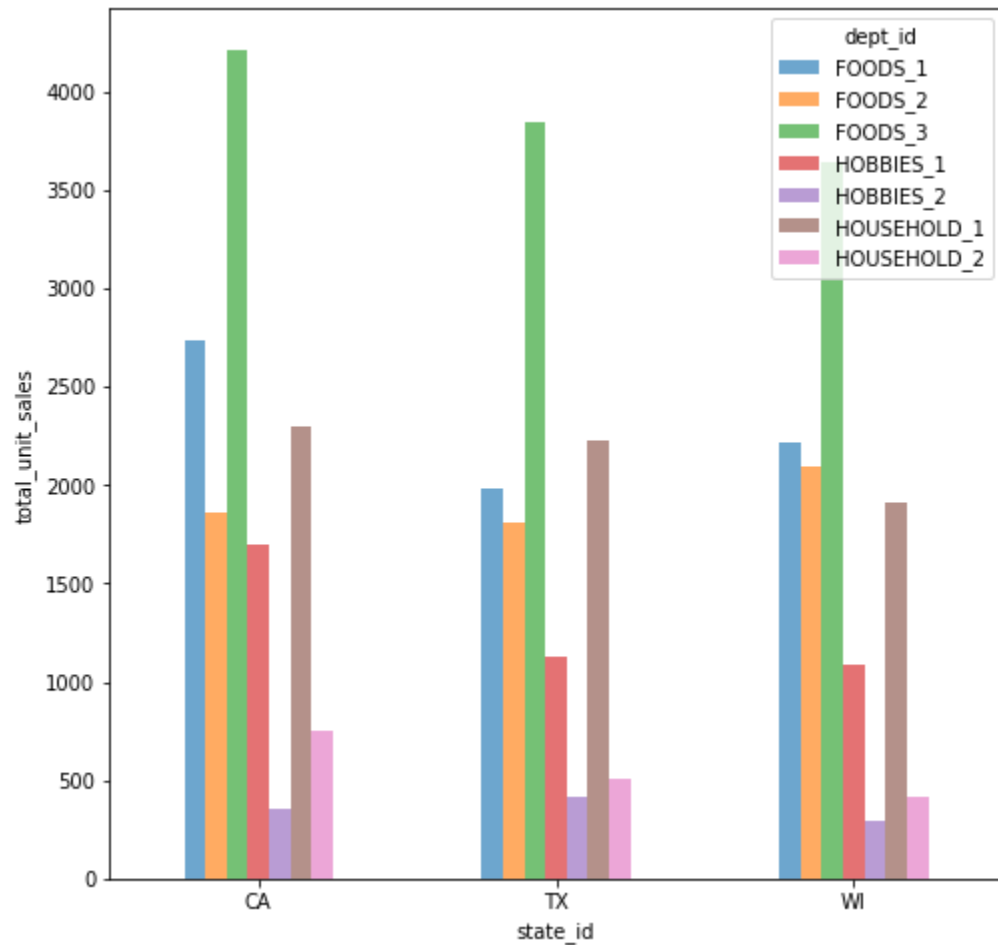
2. There are no Nans in the sales dataset
3. There are around 5 years of data points available for training
4. California stores have the highest variance and mean sales among all the stores in dataset

Another important observation was that there was a lot of variance in sales across different stores within the states



5. We also saw a dip in sales at the end of each year which can be explained by the year-end holidays (Christmas and New Year)
6. FOODS are the most common category, followed by HOUSEHOLD which is still quite a bit above HOBBIES. The number of HOUSEHOLD rows is closer to the number of FOODS rows than the corresponding sales figures, indicating that

more FOODS units are sold than HOUSEHOLD ones



The analysis was done using simple statistics and by going through visualizations

## Modeling

For the modeling, we have currently implemented the training, testing and validation steps.

1. Naive: A naive forecast involves using the previous observation directly as the forecast without any change. It is often called the persistence forecast as the prior observation is persisted.
2. Moving Average: The moving average is a statistical method used for forecasting long-term trends. The technique represents taking an average of a set of numbers in a given range while moving the range.

3. Holt Winters: The Holt-Winters forecasting method applies a triple exponential smoothing for level, trend and seasonal components. It is defined by its three order parameters, alpha, beta, gamma.
  - a. Alpha specifies the coefficient for the level smoothing
  - b. Beta specifies the coefficient for the trend smoothing
  - c. Gamma specifies the coefficient for the seasonal smoothing

There is also a parameter for the type of seasonality:

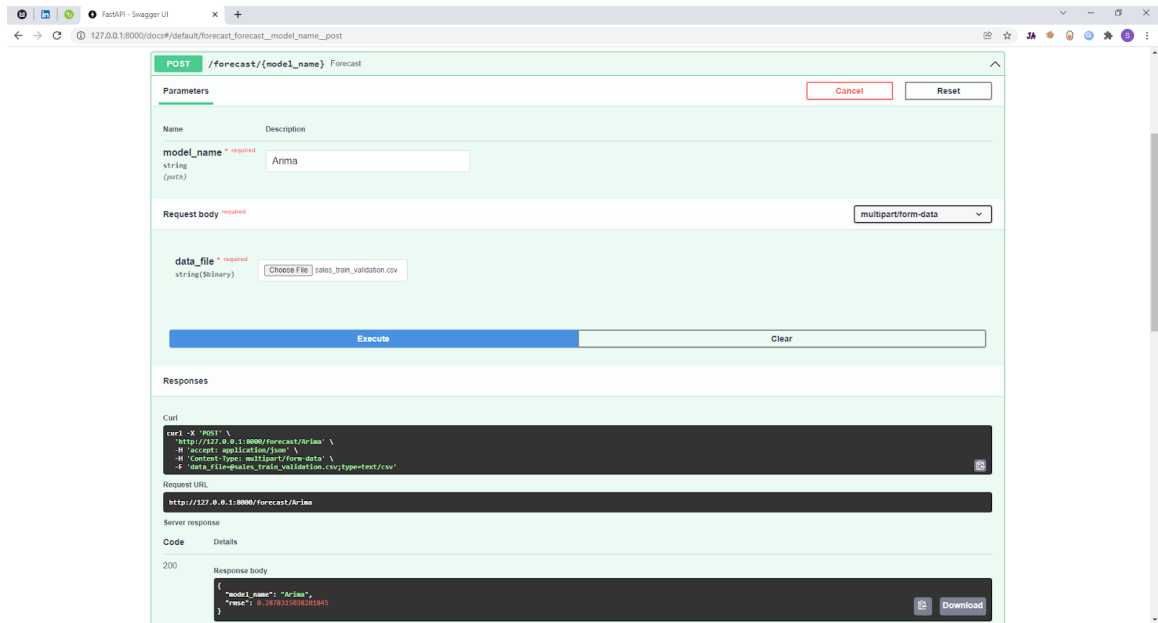
- a. Additive seasonality, where each season changes by a constant number
  - b. Multiplicative seasonality, where each season changes by a factor
4. Exponential Smoothing: This method produces forecasts that are weighted averages of past observations where the weights of older observations exponentially decrease. Forms of exponential smoothing extend the analysis to model data with trends and seasonal components.
5. Arima: The arima forecasting, short for ‘Auto Regressive Integrated Moving Average’ is actually a class of models that ‘explains’ a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.
6. Prophet: Facebook developed an open sourcing Prophet, a forecasting tool available in both Python and R. It provides intuitive parameters which are easy to tune. Even someone who lacks deep expertise in time-series forecasting models can use this to generate meaningful predictions for a variety of problems in business scenarios.
7. LightGBM with scikit-hts for reconciliation: LightGBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.

For initial testing purposes we have used the models in their vanilla form i.e. with their default hyperparameters

## API

We created an API called forecast, users can utilize it to trigger training on the provided data set. The model name is an important input that is used by the backend to determine which model to run.

In the response, it returns the model name and the RMSE calculated on actual and forecasted values.



For creating the API FastAPI was used.

## FastAPI

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints.

The key features include

1. Fast: Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
2. Fast to code: Increase the speed to develop features by about 200% to 300%. \*
3. Fewer bugs: Reduce about 40% of human (developer) induced errors. \*
4. Intuitive: Great editor support. Completion everywhere. Less time debugging.
5. Easy: Designed to be easy to use and learn. Less time reading docs.

6. **Short:** Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
7. **Robust:** Get production-ready code. With automatic interactive documentation.
8. **Standards-based:** Based on (and fully compatible with) the open standards for APIs: OpenAPI (previously known as Swagger) and JSON Schema.

## Implementation Details

### 1. Programming Language

Python 3.6.9

### 2. Libraries

fastapi - a web framework for building APIs with python 3.6+

uvicorn - a lightweight server/application interface for running the API

python-multipart - a streaming multipart parser for python (used to allow file upload via API)

pandas - a software library for data manipulation and analysis

numpy - a library to support large, multi-dimensional arrays and matrices, along with providing a large collection of high-level mathematical functions to operate on these arrays

tqdm - a library to build progress bars within your application where needed (in our use-case we use it to show model training progress)

ipywidgets - a library to facilitate interactivity

pystan - a package for bayesian inference

fbprophet - a powerful time series analysis package released by Core Data Science Team at Facebook

lightgbm - a distributed gradient boosting framework based on decision tree algorithms and used for ranking, classification, and other machine learning tasks



scikit-hts - a package that provides a python implementation of general hierarchical time series modeling

### 3. Specifications

The model runs were tested on Google Colab's default allocated machine without GPUs which has 12 GB available and can be increased to high-runtime when needed. Data was uploaded to personal google drive and the drive was mounted on the notebook.

### 4. Details of dataset

Kaggle dataset from Walmart is hierarchical sales data from Walmart, the world's largest company by revenue, used to forecast daily sales for the next 28 days. The data covers stores in three US states (California, Texas, and Wisconsin) and includes item level, department, product categories, and store details.

In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust dataset can be used to work on and improve forecasting accuracy. The data includes

- **calendar.csv** Contains information about the dates on which the products are sold
- **sales\_train\_validation.csv** Contains the historical daily unit sales data per product and store [d\_1 - d\_1913]
- **sample\_submission.csv** The correct format for submissions. Reference the Evaluation tab for more info
- **sell\_prices.csv** Contains information about the price of the products sold per store and date
- **sales\_train\_evaluation.csv** Includes sales [d\_1 - d\_1941] (labels used for the Public leaderboard)

## Impact

This tool can be used to generate swift forecasts on hierarchical timeseries data. The results would provide a baseline over which more work can be done.

There hasn't been a demo to the industry yet, and the product still needs a few tweaks before we can consider it ready.


### **Some polishes that need to be made immediately include:**

1. Enable the API to output more evaluation metrics for better comparison between the models
2. Include functionality to show results against all models for comparison in one go
3. Use categorical features as well as other available external regressors as well in the training (in the long run this could be a cool feature in itself, as per the domain we can ask the app to collect data from the web, like in delivery service data we could look up the weather which surely affects the services)

### **Some enhancements that can be done over time:**

4. Just as new approaches are made available in the market or are worked upon by anyone in the team, they can be made a part of this tool to ensure optimal results
5. User experience can be enhanced by providing more customizability within the app (like choosing the forecast horizon or overriding the hyperparameters)
6. We also have only one API at the moment that is handling training and testing, for performance and readability we can split it into multiple modules, each module would address each singular component in our project design (like separate APIs for pre-processing & feature engineering, training, and validation)

## References

1. PAPARAJU, TARUN. 2019. "M5 Competition: EDA + Models  | Kaggle." <https://www.kaggle.com/tarunpaparaju/m5-competition-eda-models>
2. Carlo Mazzaferro. 2019. "scikit-hts." [scikit-hts-examples/M5.ipynb at master](#)
3. Winegeart, Karsten. 2021. "How to Deploy a Machine Learning Model with FastAPI, Docker and Github Actions." Towards Data Science. <https://towardsdatascience.com/how-to-deploy-a-machine-learning-model-with-fastapi-docker-and-github-actions-13374cbd638a>
4. Green, Eric P. 2021. "Turn a Pandas DataFrame into an API." <https://dev.to/ericpgreen/turn-a-pandas-dataframe-into-an-api-57pk>
5. Rooney, Collin. 2017. "htsprophet." <https://github.com/CollinRooney12/htsprophet>
6. [Time Series Prediction Tutorial with EDA | Kaggle](#)
7. DATAI. 2019. "Time Series Prediction." <https://www.kaggle.com/kanncaa1/time-series-prediction-tutorial-with-eda>
8. Prabhakaran, Selva. 2019. "Time Series Analysis in Python - A Comprehensive Guide with Examples - ML+." Machine Learning Plus. <https://www.machinelearningplus.com/time-series/time-series-analysis-python/>
9. Hyndman, Rob J. 2019. "Fast forecast reconciliation using linear models." Rob J Hyndman. <https://robjhyndman.com/papers/lhf.pdf>
10. Spiliotis, Evangelos. 2020. "Hierarchical forecast reconciliation with machine learning." <https://arxiv.org/pdf/2006.02043.pdf>
11. guide, step. 2021. "Optimal Forecast Reconciliation for Hierarchical Time Series | by Jiahao Weng." Towards Data Science. <https://towardsdatascience.com/optimal-forecast-reconciliation-for-hierarchical-time-series-ea892ca105a9>
12. [https://colab.research.google.com/drive/1v\\_glauK3k4\\_gy4XRPgxeRu6bjjAMtM?usp=sharing](https://colab.research.google.com/drive/1v_glauK3k4_gy4XRPgxeRu6bjjAMtM?usp=sharing)
13. Wahome, Ronald. 2018. "Cleaning Financial Time Series data with Python." <https://towardsdatascience.com/cleaning-financial-time-series-data-with-python-f30a3ed580b7>
14. Wasike, Bravin. 2021. "Creating a Machine Learning App using FastAPI and Deploying it Using Kubernetes." <https://www.section.io/engineering-education/how-to-create-a-machine-learning-app-using-the-fastapi-and-deploying-it-to-the-kubernetes-cluster/>
15. Cappello, Nicholas, and Daniel J. TOTH. 2021. "Exploratory data analysis (EDA) of non-seasonal time series." Medium.

- <https://medium.com/analytics-vidhya/exploratory-data-analysis-eda-of-non-seasonal-time-series-51923db4006e>
16. Raman, Jayashree. 2019. "TimeSeries\_EDA." [https://github.com/jayashreeraman/TimeSeries\\_EDA](https://github.com/jayashreeraman/TimeSeries_EDA)
  17. Gulati, Himani. n.d. "Time Series Analysis — Data Exploration and Visualization. | by Himani Gulati." Jovian — Data Science and Machine Learning. Accessed January 17, 2022. <https://blog.jovian.ai/time-series-analysis-data-exploration-and-visualization-9dbe5cbb8d>
  18. Rooney, Collin. 2022. "Hierarchical Time Series Forecasting using Prophet." PythonRepo. <https://pythonrepo.com/repo/CollinRooney12-htsprophet-python-machine-learning>
  19. Teoh, Eugene. 2022. "ARIMA for Hierarchical Time Series Forecasting." <https://datapane.com/u/eugene/reports/9Armyrk/arma-for-hierarchical-time-series-forecasting/>
  20. Mazzaferro, Carlo. 2019. "scikit-hts." <https://github.com/carlomazzaferro/scikit-hts>
  21. Banerjee, Shamik. 2020. "Predicting Walmart sales – A Solution to Kaggle M5 Forecasting Accuracy Competition." Shamik Banerjee. <https://iamshamikb.wordpress.com/2020/11/22/a-solution-to-kaggle-m5-forecasting-accuracy-competition-2/>
  22. Bracht, Fabio. 2020. "Back to (predict) the future - Interactive M5 EDA." Kaggle. <https://www.kaggle.com/headsortails/back-to-predict-the-future-interactive-m5-eda>
  23. X, Graham F. 2021. "Case Study on M5 Forecasting - Accuracy | Kaggle Competition." <https://medium.com/analytics-vidhya/case-study-on-m5-forecasting-accuracy-kaggle-competition-893d7e124b54>
  24. Kumar, Bharanish. 2021. "M5 Forecasting - Accuracy." <https://medium.com/analytics-vidhya/m5-forecasting-accuracy-ad0d01a79b8e>
  25. Street, Jamie, and Jane W. Liu. 2020. "M5 Forecasting- Accuracy. Forecasting is done using Xgboost... | by Jaswanth Badvelu." Towards Data Science. <https://towardsdatascience.com/m5-forecasting-accuracy-24d7f42130de>
  26. Nicault, Christophe. 2021. "M5 Forecasting Accuracy Competition." [https://www.christophenicault.com/post/m5\\_forecasting\\_accuracy/](https://www.christophenicault.com/post/m5_forecasting_accuracy/)
  27. Lu, Hanson. 2020. "M5 Forecasting-Accuracy: Time Series forecasting using Walmart sales data." Medium.

<https://medium.com/analytics-vidhya/m5-forecasting-accuracy-time-series-forecasting-using-walmart-sales-data-374765d3f1f7>