Time Series Forecasting for Greykite and Neural Prophet Project Overview

Business Overview

A sequence of data points collected often at a constant time interval of a given entity is known as time series. The measure question asked by any business is "how did the past influence the future?" . Forecasting is a process by which the future observation is estimated by using the historical data. A statistical method which is used to analyse the data taken over time to forecast the future is known as time series forecasting. It is used to analyse time based patterns in data and hence to determine a good model to forecast the future behaviour. Basically time series connect the past, present and future.
From Supply chain, Stocks to weather, biomedical monitoring forecasting is used everywhere. There are two main use cases in forecasting. First being store sales prediction to manage inventory demand and plan ahead accordingly. Second is ride hailing demand for pricing and supply chain management. One of the importance of forecasting is if a holiday comes over how one should plan store sales to get maximum sales hence the profit. In this project we will use Walmart store sales data to predict store sales. Greykite, a python library developed by Linkedin and the famous Neural Prophet model developed by Facebook is used to forecast the demand.

Aim

To predict future demand/sales using historical data and other related features.

Dataset

https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

Data Description

The dataset used is Walmart store sales data. Walmart is an American multinational retail corporation which operates a chain of hypermarkets, department stores and Grocery stores. The dataset provided is historical sales data for 45 Walmart stores located in different regions. Each store contains many departments. Four different datasets are being provided which are discussed below.

1. Stores.csv: This file contains information about the 45 stores, indicating the type and size of store.
2. Train.csv: This is the historical training data, which covers 2010-02-05 to 2012-11-01.
3. Test.csv: This file is identical to train.csv, except the weekly sales which have to be predicted.
4. Features.csv: This file contains additional data related to the store, department, and regional activity for the given dates.

The basic information about the features available in data is as follows.

- Store - the store number
- Date - the week
- Dept - the department number
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region
- MarkDown1-5 - anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- CPI - the consumer price index
- Unemployment - the unemployment rate
- IsHoliday - whether the week is a special holiday week
- Weekly_Sales - sales for the given department in the given store

Tech Stack

- ➔ Language : Python
- ➔ Libraries : greykite, neural prophet, sci-kit learn, pandas, pandas_profiling, matplotlib, datetime, plotly, seaborn, numpy

Approach

1. Exploratory data analysis (EDA)
    a. Inference about features
    b. Data visualization using pandas profiling

2. Data cleaning (outlier/missing values)
    a. Missing value imputation
    b. Outlier detection

3. Feature Engineering
    a. Extracting day, month and year from date
    b. Mapping

4. Time series component analysis
    a. Trend
    b. Seasonality

5. Model building on training data
    a. Silverkite
    b. Neural Prophet

6. Model validation
    a. Mean Absolute Percent Error
    b. RMSE

7. Forecasting using trained models

Modular code overview

```
input
  |_ features
  |_ stores
  |_ test
  |_ train

lib
  |_ Forecasting_with_Prophet_Greykite .ipynb


output
  |_ prophet_model.pkl

src
  |_engine.py
  |_ ML_pipeline
            |_ Deploy.py
            |_ Preprocess.py
            |_ Train_Model.py
            |_ Utils.py

requirements.txt
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib
5. requirements.txt

1. The input folder contains all the data that we have for analysis. In our case, it will contain four csv files which are
   a. features.csv
   b. stores.csv
   c. test.csv
   d. train.csv

2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further contains the following.
   a. ML_pipeline
   b. engine.py
   c. server.py

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file

3. The output folder contains the models that we trained for this data. This model can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
   Note: This model is built over a chunk of data. One can obtain the model for the entire data by running engine.py and by taking the entire data to train the models.

4. The lib folder is a reference folder. It contains the original ipython notebook that we saw in the videos.

5. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command pip **install** -r **requirements**.**txt**
   Note: For installing the neuralprophet and greykite libraries kindly refer to the document “Steps to install Neuralprophet and Greykite libraries” .

Project Takeaways

1. Understanding the Business context and objective
2. Data Cleaning
3. Inference about data
4. Feature Engineering
5. Importing the dataset and importing libraries
6. Main components of time series
7. Trend Analysis
8. Seasonality Analysis
9. Lagged regressors
10. Future Regressors
11. Understanding Greykite library
12. Building Greykite model
13. Understanding Neural Prophet library
14. Building Neural Prophet model
15. Understanding AR net model
16. Model Predictions
17. Model Evaluation
18. Greykite and Neural Prophet model comparison
19. Flask Deployment