

Contents

[**Introduction**](about:blank)

[**Data Processing**](about:blank)

[**Target Variables**](about:blank)

[**Predictive Variables**](about:blank)

[**Pre-modeling**](about:blank)

[**Modeling**](about:blank)

[**Algorithmic Solution Results**](about:blank)

[**Deliverables**](about:blank)

Introduction

* The Online Retail II data set, which gathered actual online retail data in two years, is from the UCI machine learning repository. The source of the data set is Dr. Daqing Chen, Course Director: MSc Data Science. chend **'@'** lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK.
* The Online Retail data set contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011. The company mainly sells unique all-occasion giftware. Many customers of the company are wholesalers.
* Total sales will be the target variable and predicting variable will be time series.
* Time series analysis model (MA, AR, ARIMA, etc.) and machine learning model will be implemented to analyze and forecast future sales.

The purpose of this document is to provide a detailed technical overview of

1. Data collecting and cleaning
2. Data inclusions and exclusions
3. Predictive variable creation process
4. Target variable definition
5. Iterative model building process
6. Metric evaluation process

Diagram

Description automatically generated

Data Processing

**Data Overview**

We are doing Forecasting on UCI Online Retail Data Set. This dataset contains all the transaction information occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

**Data Extraction Process**

The data is downloaded from UCI Machine Learning Repository, Online Retail Data.

**Data Diagnostics**

A series of quality checks are performed on the data extract provide. These checks included:

* Attribute information
* Number of records
* Duplicate records if any
* Missing values in relevant fields

Attribution information:

1. InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
2. StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
3. Description: Product (item) name. Nominal.
4. Quantity: The quantities of each product (item) per transaction. Numeric.
5. InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction.
6. UnitPrice: Unit price. Numeric, Product price per unit in sterling.
7. CustomerID: Customer number. Nominal, a 5-digit integral unique number.
8. Country: Country name. Nominal, the name of the country where each customer resides.

The number of records:

1. Year 2009-2010: 525461
2. Year 2010-2011: 541910

Duplicate records if any: There are no duplicate records.

Missing values in relevant fields:

Year 2009-2010

| Attributes | Number of missing values |
| --- | --- |
| Invoice | 0 |
| StockCode | 0 |
| Description | 2928 |
| Quantity | 0 |
| InvoiceDate | 0 |
| Price | 0 |
| Customer ID | 107927 |
| Country | 0 |

Year 2010-2011

| Attributes | Number of missing values |
| --- | --- |
| Invoice | 0 |
| StockCode | 0 |
| Description | 1454 |
| Quantity | 0 |
| InvoiceDate | 0 |
| Price | 0 |
| Customer ID | 135080 |
| Country | 0 |

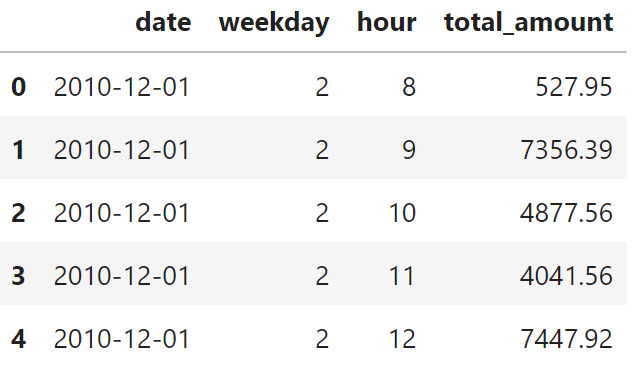
**Modeling Data Creation**

For each of the customers, the modeling data are collected and processed using the following steps. The below also addresses the feature engineering section.

* Holidays: create features using time lags from previous 3 days to future 2 days [-3, 2]
* Time related features: month, year, week are created using datetime

Target Variables

Total\_Amount: quantity times price for each product which is the sales of the day



Predictive Variables

A predictive variable is a variable used in algorithmic solutions to predict the target variable. During our analysis, we categorized predictive variables into two categories:

1. Direct variable – These variables were directly from the dataset that was provided by direct customers
2. Derived variable – These variables were created by manipulating the direct variables

**Variable List**

1. Weather: Temperature, Seasons data.
2. Inventory: customer’s inventory.
3. Holiday: US national holidays. Holiday has effect on people’s purchasing patterns, for example, Black Friday, Christmas, etc. It is partly related to the promotion.
4. Features of time: Year, Month, Week. Features of time are critical in the modeling because they capture seasonality in shipments. For example, the shipment of ice cream increases during summer times.
5. Quantity: How many weeks the product was sold

Pre-modeling

* Before modeling, some analysis techniques were used to discover the seasonality and trend of the original dataset.
* Seasonal decomposition for the hourly sales dataset and daily sales dataset can help us pick the dataset for modeling.
* The pattern and the cut-offs in ACF and PACF can help choose parameters for models such as ARIMA and SARIMAX.

**Training & Testing Split**

We split the dataset into three parts - training, validation, and testing. The ratio of each part is 8:1:1.



As displayed in the figure above, 80% of the data was used to train the model, and about 10% data was used for validation during the hyperparameter tuning and model selection process. Another 10% of the data was used as Testing data to calculate the error of the best model. The Testing data will be split into three parts and evaluated each part of the data.

Modeling

**Introduction**

After the candidate predictive variables were finalized in the pre-modeling step, the team performed the necessary data mining and statistical iterations to develop and build the Algorithmic Solutions.

1. SARIMAX

* Seasonal Auto-regressive Integrated Moving Average with eXogenous factors, is an extension of the ARIMA class of models.
* SARIMA has the capability of dealing with seasonality and handling exogenous variables.

1. GLM

* Because GLM cannot handle date-time type data, quarter, month, day in the month, and weekday are extracted from the date. Also, the holiday days in the UK From 2009 to 2011 are marked for tracking the holiday season trend. Daily weather data collected by the weather station at Heathrow Airport are also implemented.
* Among OLS, Poisson, Gaussian, and Gamma regression in the GLM model, the Poisson regression yields the best result with MAPE = 0.324.

1. Prophet[[1]](#footnote-0)

* Prophet is a procedure for forecasting time series based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series with substantial seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend and typically handles outliers well.

1. Neural Prophet[[2]](#footnote-1)

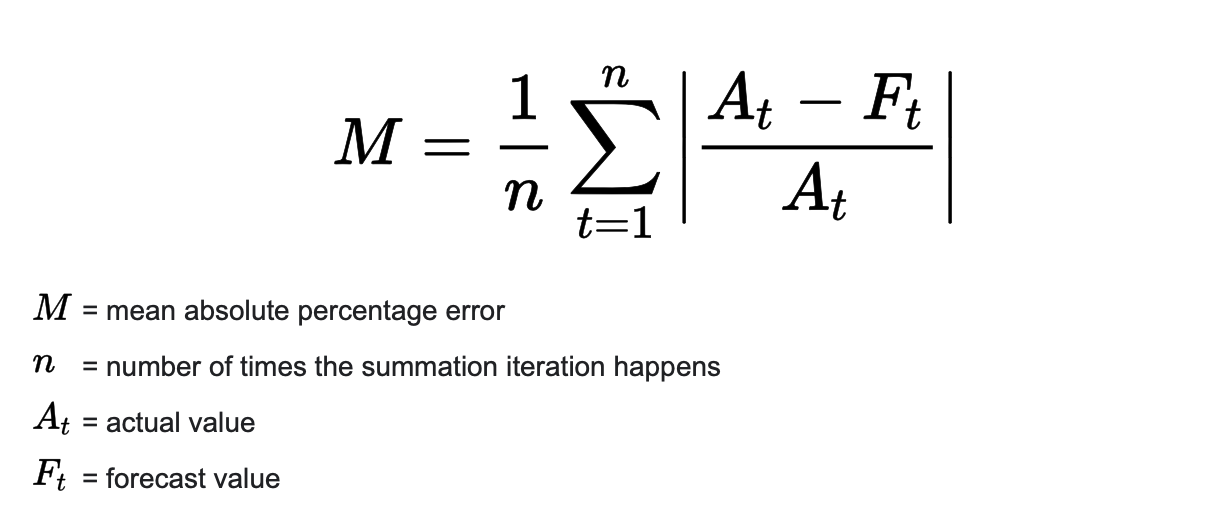
* Neural Prophet is an easy-to-learn framework for interpretable time series forecasting. Neural Prophet is built on PyTorch and combines Neural Network and traditional time-series algorithms, inspired by Facebook Prophet and AR-Net.

**Hyperparameters Selection of Algorithms**

* SARIMAX
  + AR parameters
  + Differences
  + MA parameters
  + Seasonal component of the model for the AR parameters
  + Seasonal component of the model for the Differences
  + Seasonal component of the model for the MA parameters
* GLM
  + Different parameters combinations choosing
  + Link Function
* Prophet
  + change\_prior\_scale
  + change\_point\_range
  + holidays\_prior\_scale
  + interval\_width
  + seasonality\_prior\_scale
  + uncertainty\_samples
* Neural Prophet
  + batch\_size
  + changepoints\_range
  + daily\_seasonality
  + d\_hidden
  + epochs
  + growth
  + impute\_missing
  + learning\_rate
  + loss\_func
  + normalize
  + num\_hidden\_layers
  + n\_change\_points
  + n\_forecasts
  + optimizer
  + seasonality\_mode
  + weekly\_seasonality
  + yearly\_seasonality

**Evaluation Metric**

One metric was adopted in model evaluation: Mean Absolute Percentage Error



Mean Absolute Error captures the percentage error between the actual and predicted value, which would be a good choice when considering a time-series model.

| **Model** | **SARIMAX** | **GLM** | **Prophet** | **Neural Prophet** |
| --- | --- | --- | --- | --- |
| **Test1** | 0.30438 | 0.43457 | 0.31184 | 0.24937 |
| **Test2** | 0.32932 | 0.26476 | 0.19802 | 0.18288 |
| **Test3** | 0.44467 | 0.27447 | 0.22921 | 0.16387 |
| **Average MAPE** | 0.35946 | 0.32419 | 0.24606 | 0.19807 |

**Algorithmic Solution Finalization**

The following procedure was used to determine the Sales predictions. First, for each product, we run the four models with configured hyperparameters on training data. The model with the highest forecast accuracy is chosen. And then, we train on the whole dataset using the selected model and make predictions of testing data.

From the evaluation metric, Neural Prophet is the best model so far. The Neural Prophet integrates the Prophet and Neural Network which brings the power for handling all seasonal data and error in data.

Deliverables

**Results**

# <https://github.com/yichuang25/Forecasting-Project-Sales/tree/main/Analysis%26Modeling>

**Model**

# <https://github.com/yichuang25/Forecasting-Project-Sales/tree/main/Models>

**Datasets**

<https://github.com/yichuang25/Forecasting-Project-Sales/tree/main/Data>

**GitHub**

<https://github.com/yichuang25/Forecasting-Project-Sales>

To access the deliverables, email us for access.

# Team Information

Haolong Liu - hl3614@columbia.edu

Yichen Huang - yichen.huang@columbia.edu

Taichen Zhou - tz2555@columbia.edu

1. https://facebook.github.io/prophet/ [↑](#footnote-ref-0)
2. https://neuralprophet.com/contents.html# [↑](#footnote-ref-1)