Streaming Data Management and Time Series Analysis Project

University of Milano-Bicocca

Ismail AHOUARI - 896440 (i.ahouari@campus.unimib.it)

A.A 2023/24

Introduction

Streaming data management refers to the process of ingesting, processing, and analyzing continuously generated data streams in real-time. This type of data typically comes from various sources such as sensors, social media feeds, website clickstreams, financial transactions, and IoT devices.

Time series analysis, on the other hand, is a statistical technique used to analyze data points collected or recorded over a period, where each data point is associated with a timestamp. Time series data is sequential and can exhibit patterns, trends, seasonality, and irregular fluctuations over time.

In practice, streaming data management and time series analysis often go hand in hand, especially in applications like financial market analysis, IoT monitoring, predictive maintenance, and real-time event detection, where timely analysis of streaming time series data is crucial for decision-making.

Objective

The objective of this project is to generate forecasts for the six months after the available data. Specifically, employing conventional time series modeling techniques, the aim is to forecast the period spanning from January 4, 2007, to March 31, 2015, encompassing a total of NUMBER observations.

Initially, linear models such as ARIMA and UCM will be applied, followed by a non-linear typological approach utilizing Machine Learning techniques, **Random Forest**, **Decision Tree**, and **SVR**. These models were chosen for their ability to capture complex patterns and relationships in the data, complementing the traditional approaches.

The performance of the models will be evaluated and compared using the mean absolute error (MAE).

Pre-processing

The dataset consists of 2 columns: the date, which runs from January 1, 2017 to November 30, 2017 in the format dd/mm/yyyy day/month/year format); the column y, which contains information on the total amount of ave_days consumed in a given time period.

First, a brief pre-processing was carried out: the presence of possible duplicates was checked and values missing in the historical series and, having found none, continued with the next phase.

In the data preprocessing phase, I extracted additional features from the 'date' attribute to enrich the dataset and potentially improve the predictive performance of the models. These additional features include: weekday,dayofweek, quarter,dayofmonth, month, year, ve_days.

We are handling outliers by replacing them with more representative values (i.e., the mean of the corresponding week) instead of outright removing them. This helps in preserving the overall distribution of the data while mitigating the impact of extreme values on the analysis. Additionally, replacing outliers with weekly means leverages the temporal aspect of the data, considering that data points within the same week may exhibit similar characteristics or trends.

Exploratory and Series Analysis

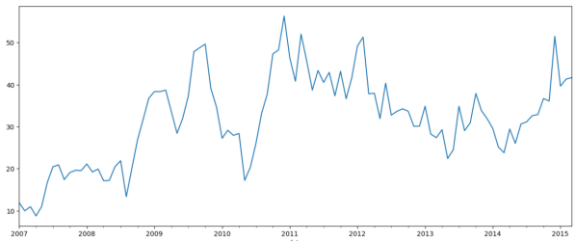
This report presents an exploratory data analysis (EDA) of the 'ave_days' time series data. The analysis aims to understand the temporal characteristics, identify trends, and assess the stationarity of the dataset.

Overall, the summary statistics suggest that the 'ave_days' variable exhibits a relatively symmetric distribution, with the majority of values clustered around the mean and median. However, there is variability in the data, as evidenced by the difference between the 1st and 3rd quartiles and the presence of outliers at the maximum value.

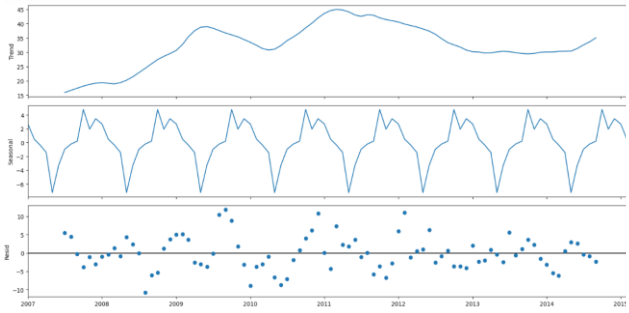
```
(3009, 9)
      statistic      value
0       Minimum   0.000000
1  1st Quartile  15.068966
2        Median  31.218638
3          Mean  31.782398
4  3rd Quartile  44.665615
5       Maximum  97.186992
```

To gain deeper insights into the underlying trends and patterns within the time series data, we employed a resampling and aggregation technique.

This technique allows us to transform the data into a more manageable form while preserving its temporal characteristics.



The resulting resampled and aggregated dataset provides valuable insights into the average consumption or activity level on a monthly basis, allowing us to identify long-term trends, seasonality, and other temporal patterns that may exist within the data.



Trend: This shows the underlying long-term trend of the data, removing any seasonality or noise.
Seasonal: This captures the recurring seasonal patterns in the data, like weekly or yearly cycles.
Residual: This represents the remaining fluctuations in the data after accounting for trend and seasonality.

ARIMA

The ARIMA methodology was proposed by Box and Jenkins (1976) and it is now a quite popular tool in economic forecasting. The basic idea is that a stationary time series can be modeled as having both autoregressive (AR) and moving average (MA) components. Non-stationary integrated series can also be handled in the ARIMA framework, but it has to be reduced to stationary beforehand by the difference in the data.

ARIMA uses an auto-regressive,(AR) component to capture the relationship between a value in the series and previous values, moving average (MA) to handle the presence of random prediction errors, and integrated differentiation (I) to deal with the presence of non-stationary trends in the series.

The dataset was divided into two parts: the train set, which runs from **2007-01-04** until **2013-03-30**, and the test set, which runs from **2013-03-31** until **2015-03-31**.

Our approach to tackle this forecast. First, we identified seasonality through ACF/PACF plots, and then we explored various ARIMA configurations.

We initially applied the extracted seasonal parameter, followed by experimenting with random p and q values. Additionally, we applied grid search for hyper-parameter optimization. By comparing each model's performance, we evaluated diagnostic results like residuals, goodness-of-fit, and MAE (Mean Absolute Error). This approach ensured we selected the ARIMA model that effectively captured the time series characteristics and delivered the most accurate forecasts.

The Parameters we examined are as follows:

- **ARIMA(0, 1, 0)(0, 1, 0)[24]**

The resulting model had an **AIC** of **9683** and an **MAE** of **45**.
This, together with the visual inspection of the Normal Q-Q plot, which shows a lack of normality in the residuals, indicates that this parameterization is not ideal.
As a result, further investigation of various p and q values is required to select a model that better represents the underlying structure of the data and gives more accurate results.

- **ARIMA(0, 1, 0)(0, 1, 0)[7]**

While this model outperforms the prior attempt with an **AIC** of **8184** and an **MAE** of **34**, some significant difficulties remain.

Despite the decreased **AIC** and MA**E**, normality in the residuals remains lacking, as evidenced by the lack of agreement between the data and the normal distribution in the Q-Q plot. Additionally, the residuals over time plot suggest seasonality, which is verified by the correlogram plot. These considerations show that the model requires further modification to produce more accurate and statistically sound findings.

- **ARIMA(0, 1, 0)(1, 1, 1)[24]**

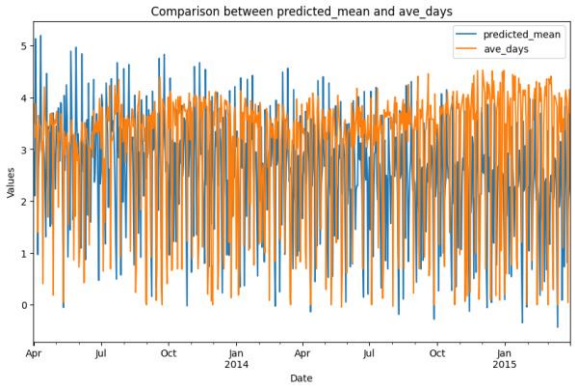
This model further reduces the **AIC** score (**8180**) and MAE (**2**) compared to the previous option, it

still requires refinements to address persistent issues.

Similar to the **ARIMA(0, 1, 0)(0, 1, 0)[7]** model, it has non-normal residuals as shown in the Q-Q plot, as well as probable seasonality as shown by the residuals over time plot and validated by the correlogram. The decreasing error metrics indicate promise, but addressing normalcy and seasonality difficulties is critical to generating statistically valid and reliable results.

- **ARIMA(6, 1, 5)(0, 1, 0)[24]**

Proceeding with grid-search, this model has a substantially lower **AIC** (**7322)** and **MAE** (**1**) than prior attempts, indicating better fit and accuracy. Furthermore, visual inspection of the Q-Q plot indicates that the residuals are normal, which is a critical requirement for a more sound statistical measurement. The lack of seasonality trends in the residuals plot and correlogram supports this model's appropriateness. While additional testing is recommended, the **ARIMA(6, 1, 5)(0, 1, 0)[24]** model appears to be the best option based on the current findings from the AIC, MAE, and diagnostic plots.
This model stands out as the current top performer for accurate and statistically reliable forecasting.



A comparison between the original series and the predictions provided by the improved Arima **ARIMA(6, 1, 5)(0, 1, 0)[24]** can be seen in the above figure.

During our exploration, we found additional ARIMA models with lower AIC and MAE scores (**p=6, q=5**). However, they failed in essential key areas. Comparing these models to the **ARIMA(6, 1, 5)(0, 1, 0)[24]** model, the latter showed more accuracy on the test set and also in diagnostic tests. This emphasizes how critical it is to prioritize overall model performance above and beyond measurements such as AIC and MAE since real-world accuracy and diagnostic evaluations are essential for choosing a strong and trustworthy forecasting model.

UCM

Within the realm of time series analysis, unobserved components models (UCMs) offer a powerful tool for both understanding and forecasting univariate data. These models, also known as structural models, unpack the intricacies of time series by decomposing them into distinct components: trend, seasonality, cycles, and even the influence of external variables. Each component is carefully chosen to encapsulate the key characteristics that drive the behavior of the series, essentially capturing the "stylized facts" that define its evolution.

Interestingly, UCMs bridge the gap between the flexibility of ARIMA models and the interpretability of smoothing models. ARIMA models boast impressive forecasting capabilities but can sometimes lack transparency in their inner workings. Conversely, smoothing models prioritize clarity but may struggle to capture intricate patterns. UCMs strike a balance, offering interpretability through clearly defined components while maintaining the ability to model complex dynamics, akin to ARIMA.

It is important to note that the train-test split was not modified for this iteration. Maintaining the same **80**/**20** split.

As for the UCM part, different models were tested again to obtain the one that best fit the

data. Thus, the level and seasonal parameters were considered.

The level parameter that models the trend component, was initially considered. Considering daily seasonality, the most evident type of seasonality within the available data, different types of models were tested by varying the level parameter between the following values: No trend, Local level, Random walk, Deterministic trend, Local linear deterministic trend, Random walk with drift, Local linear trend, Smooth trend. Taking MAPE and secondarily AIC as metrics, the type of level that yielded the best results was **rwdrift**, which is a random walk with drift. It is possible to observe the results obtained in Table below.

Level	AIC	MAPE
llevel	8973.66	0.46
rwdrift	9343.82	0.43
ntrend	11832.11	1.47
Rwalk	9350.27	0.51
lldtrend	8971.04	0.38
lltrend	10044.68	57.77
strend	10042.50	57.78
rtrend	10287.24	98.34

Table: Modeling Component of UCM models

Next regarding the seasonal component, making several attempts, it was decided to opt for 3 sine waves of period 24 and 2 sine waves of period 168 to model daily seasonality.

Thus considering the final model, the best results were obtained with a level equal to random walk with drift and the seasonal part modeled by 3 sine waves of period 24 and 2 sine waves of period 168.

The trained model recorded on the test set a MAPE value of 11.02 %. The result from the graphical point of view of fitting the model on the testing data can be observed next figure

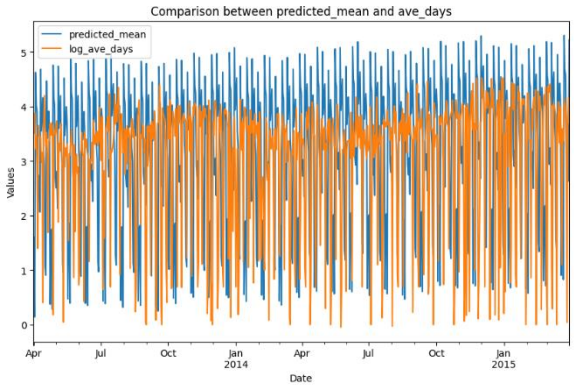


Figure: Prediction on the UCM test set

Machine Learning

The machine learning process involved several key steps, beginning with the definition of the predictive problem at hand. Our objective was to develop models capable of predicting the *'ave_days'* variable based on a set of features extracted from the dataset, including temporal attributes such as *dayofweek, quarter, dayofmonth, month, year, Day_of_year, Weekend, and season.*

Following the problem definition, the dataset underwent preprocessing to clean and prepare the data for modeling. This included extracting additional features from the 'date' attribute to enrich the dataset with temporal information. Subsequently, the dataset was split into training and test sets, with **2156** instances allocated to the training set and **539** instances to the test set.

We then selected and evaluated three different machine-learning models:
Random Forest , **Decision Tree**, and **SVR**.

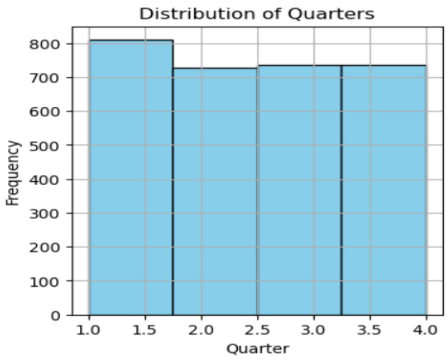
Each model was trained using the training dataset and evaluated using evaluation metrics

such as:

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) on the test dataset.

The results of the evaluation revealed that the Random Forest model outperformed the other models, demonstrating lower error metrics and higher predictive accuracy. This suggests that the Random Forest model is the most suitable for predicting the *'ave_days*' variable in our dataset.

Overall, the machine learning process involved iterative steps of data preprocessing, model selection, training, and evaluation to develop and assess predictive models capable of accurately predicting the target variable.



The code creates a visually informative histogram that illustrates the distribution of quarters, providing a clear representation of the frequency of occurrences for each quarter.

SVR

Support Vector Regression (SVR) is a machine learning algorithm used for regression tasks. It works by finding the optimal hyperplane that best fits the training data while minimizing prediction errors.

SVR is particularly useful for datasets with complex relationships and non-linear patterns, as it can capture these intricacies effectively. However, SVR may be sensitive to its hyper-parameters and may require careful tuning to achieve optimal performance.

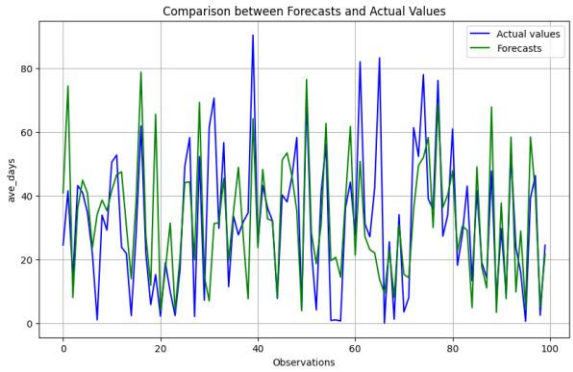
Applying this type of model after standardizing the input data, predictions were made on the validation set returning an MAE of **19.01** Different approaches were then tried to see if better performance could be achieved: I focused, above all, on decision tree approaches, and Random forests.

The performance metrics obtained from the evaluation of the SVR model indicate that it struggles to accurately predict the target variable compared to the other models evaluated. The relatively high RMSE, MAE, and MAPE values suggest that SVR may not effectively capture the underlying patterns and relationships present in the dataset. This could be due to challenges in finding the optimal hyper-parameters for the SVR model or limitations in its ability to handle the complexity of the data.

Random Forest

The Random Forest model played a pivotal role in our predictive analysis, offering valuable insights and accurate forecasts for the 'ave_days' variable. Here's a detailed examination and commentary on the relevance and performance of the Random Forest model

Random Forest emerged as a compelling choice for our predictive modeling task due to its ensemble learning nature and robustness to overfitting. In our dataset, which comprises various temporal attributes and features extracted from the 'date' attribute, Random Forest's ability to handle non-linear relationships and capture complex interactions proved highly relevant.



In particular, when applying Grid Search to Random Forest, an **MAE of `11.03`** is achieved on the validation set.

These algorithms, including Random Forest, are well-suited for consumption forecasting due to their ability to model intricate relationships among multiple variables and effectively handle datasets with numerous variables or heterogeneous data. Additionally, their high flexibility allows for adaptation to various scenarios and datasets

Decision Tree

Decision Trees are well-known for their simplicity and interpretability, making them an attractive choice for tasks where understanding the decision-making process is important. However, Decision Trees can be susceptible to overfitting and may struggle with capturing complex, non-linear relationships present in the data. In this case, the DecisionTree achieves an MAE of **14.41**, which suggests decent performance but falls short of the Random Forest model. While Decision Trees provide valuable insights into feature importance and decision rules, their limitations in handling complex relationships might hinder their predictive power compared to other models like Random Forests. It's important to consider these trade-offs when choosing the appropriate model for your specific needs.