Volos 2020

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

TIME-SERIES ANALYSIS

Diploma Thesis

Kalliopi Rantou

Supervisor

Assoc. Prof.

Dimitrios Katsaros

Volos 2020

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

TIME-SERIES ANALYSIS

Diploma Thesis

Kalliopi Rantou

Supervisor

Assoc. Prof.

Dimitrios Katsaros

Βόλος 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΛΥΣΗ ΧΡΟΝΟΣΕΙΡΩΝ

Διπλωματική Εργασία

Καλλιόπη Ράντου

Επιβλέπων

Αναπλ. Καθ.

Δημήτριος Κατσαρός

Dedicated to my family and friends, who believed in me and support me till the end. Thank you!

Volos 2020

Acknowledgements

First and foremost, I would like to thank my thesis supervisor associate professor Katsaros Dimitrios of the Department of Electrical and Computer Engineering of University of Thessaly. He supported me and helped me sort out my thoughts, organize my plans and delimit my time. He considered an important aid with the implementation of this thesis and so the completion of my studies. In addition, I would like to thank miss Tousidou Eleni, who belongs to the Laboratory Teaching Staff and the professor Tsoukalas Eleutherios, who cooperated with Assoc. Prof. Katsaros for the supervision of my thesis.

The most important role in achieving my goal have my parents, my sister and my beloved friends. I must express my gratitude to them, because their support and their ‘desire’ to succeed was the biggest motivation for me. I felt their love and this is what I needed those late nights.

Thank you all so much!

ΠΕΡΙΛΗΨΗ

Οι χρονοσειρές οριίζονται ως μία σειρά παρατηρήσεων, οι οποίες δειγματοληπτούνται σειριακά στον χρόνο. Συγκεκριμένα, αναφέρουμε κάποια παραδείγματα όπως οι μηνιαίες πωλήσεις ενός σαμπουάν, οι εβδομαδιαίες καταγραφές αυτοκινητιστικών ατυχημάτων, η ελάχιστη θερμοκρασία που υπολογίζεται καθημερινά ή οι παρατηρήσεις ανά ώρα ενός χημικού πειράματος. Αυτά τα παραδείγματα παρατηρούνται σε διάφορους τομείς, όπως είναι στην οικονομία, τη μηχανική, στις επιχειρήσεις και σε φυσικά φαινόμενα. Αυτό αποτέλεσε και το κύριο κίνητρο αυτής της διπλωματικής εργασίας, η εξευρεύνηση και η ανάλυση δηλαδή των χρονοσειρών, έτσι ώστε να εξαχθούν πολύτιμες πληροφορίες για μελέτη. Αυτές, καθώς και η συμπεριφορά που παρατηρείται στην εξέλιξη του χρόνου, θα εφαρμοστούν στο επιθυμητό πεδίο του αναλυτή. Τα μέσα στα οποία βασίστηκε η ανάλυση είναι το Prophet model για πρόβλεψη, καθώς και το Dynamic Time Warping για την μέτρηση της ομοιότητας ανάμεσα στις χρονοσειρές. Και οι δύο αλγόριθμοι καταγράφονται στην ελίτ των μεθόδων και παρέχουν ευελιξία στον αναλυτή καθώς και αξιόπιστα αποτελέσματα. Αναλυτικότερα, το Prophet model παράγει μία ολοκληρωμένη και εμπεριστατωμένη πρόβλεψη των δεδομέων που δίνονται ως είσοδο, και τα οποία μπορεί να εμφανίζουν έντονα χαρακτηριστικά που επηρεάζονται από την εποχικότητα. Το δεύτερο μοντέλο, DTW, υπολογίζει τη βέλτιστη απόσταση ανάμεσα στις δεδομένες χρονοσειρές. Είναι σημαντικό να αναφερθεί ότι το μήκος τους μπορεί να διαφέρει, εξασφαλίζοντας έτσι μία καλύτερη ανάλυση, που θα μπορούσε να χρησιμοποιηθεί τόσο στην εξόρυξη δεδομένων, όσο και στην μηχανική μάθηση κλπ. Οι λειτουργικότητες των αλγορίθμων θα παρουσιαστούν παρακάτω μέσα από εκτελέσεις με εισόδους δεδομένων.

ABSTRACT

A time series is a sequence of observations taken sequentially in time. Many sets of data are considered as time series. Specifically, a monthly number of sales of shampoo, a weekly series of the number of road accidents, the minimum daily temperatures or the hourly observations of a chemical experiment. Such examples can be found in fields of economics, engineering, business and natural science. That was the main motivation of the current thesis, exploring and analyzing time series in order to extract valuable information. This information and the behavior that appears through time will be applied on the desired field. The means on which the analysis based on are the Prophet model for forecasting and the Dynamic Time Warping (DTW) algorithm for measuring the similarity. Both of them, are recognized as state-of-the-art algorithms, which provide flexibility for the analyst and reliability on the results. Prophet model creates a complete and thorough prediction of the input data set, which may have strong seasonal effects, whereas DTW model calculates the optimal distance of two time series. It is important to mention that their length can be the same or differ, offering the facility for better analysis, which will be used in data mining, machine learning etc. Their function will be visualized below through executions of input data set.

Table of Contents

CHAPTER 1

Introduction to time-series

1.1 Introduction

Time series ideas appear to all activities. The data are collected from a real life thing, that we are interested in and analyzed by specific algorithms to create graphic and numeric output. The output of the analysis reveal more information about real life condition. It can be used to make informed predictions of future values or help for the classification and clustering, for data mining and pattern recognition.

The basic idea of the current thesis is the analysis of time series using state-of-the-art algorithms. Specifically, the main purpose is to familiarize with translating data into models, so as to make forecasts for a wide variety of fields, ranging from macroeconomics, to finance and marketing and through similarity measurement, to support machine learning and more.

The following chapters introduce this idea. Specifically, the second chapter defines the meaning of time-series and their characteristics of forecast and similarity, referring some of their applications. The third chapter is referred at the state-of-the-art algorithms that are available for forecasting and similarity, whereas the fourth sets the basic idea of the thesis, included the algorithms that have been used and pseudo-code. The fifth chapter documents the code. In other words, defines the requirements and restrictions of the program and analyzes the files and their functions. The next chapter concerns the executions, so as to visualize the theoretical approach based on the results. Finally, the thesis is completed by the conclusion and the future work that has been set as a goal.

1.2 Time-series

Time series are series of data points ordered in time.[1] Time is often the independent variable. They track the movement of chosen data points, such as daily closing value of the Dow Jones Industrial Average, over a specified period of time, with data points recorded at regular intervals.[2] There is not a minimum or maximum amount of time that must be included, allowing the data to be gathered in a way that provide all the necessary information to the investor or analyst who examines the current activity. Mathematically, time series can be represented by the function Ft = f(t), where t stands for values of time and Ft the corresponding value of the model.

Time series refer to any database which has a ‘time’ variable. Their use is important for data mining and predictive analysis, including forecasting, due to the fact that future predictions are based on past data records and also, that the trends observed in the past may recur in the future.

The analysis of time series consist the base for descriptive, explanation, forecasting and intervention analysis.[3] Specifically, the definition for them are:

- **Descriptive**: Identify patterns in correlated data, trends and seasonal variation, which will be described below.
- **Explanation**: understanding and modeling the data.
- **Forecasting**: prediction of short-term trends from previous patterns.
- **Intervention analysis**: how does a single event change the time series.

1.3 Stationarity

Stationarity is an important characteristic of time series. [4] A time series is stationary, when all statistical properties of that series remain unchanged by shifts in time. In other words, it has constant mean and variance, and covariance is independent of time. In technical terms, strict stationarity implies that the joint distribution of (y_t, …, y_{t-h}) depends only on the lag, h, and not on the time, t. Note that strict stationarity is not widely necessary in time series analysis. Often, stock prices are not a stationary process, since we might see a growing trend, or its volatility might increase over time, which means that variance is changing.

1.4 Plots and components

The graphs based on time series plot observed values on y-axis against time on x-axis. Through them, we can visualize the behavior and the patterns of data and so we can base a reliable model, for the analysis, on them. In addition, we can recognize the major components of the certain time series which are: [5]

1.4.1 Trend

Trend is the increasing or decreasing values in the series over a period of time. It is important to note that it persists over a long period of time. For example, the stock price. The plot below suggests the presence of trend (fig. 1).

1.4.2 Seasonality

Seasonality describes a regular pattern for a specific period of time, such as week or month etc. Small alterations are occurring due to to seasonal factors (e.g., the quarter of the year, the month, or day of the week) and are always on a fixed and known period. Specifically, there is a precise amount of time between the peaks and bottoms of the data.

An example of a time series with seasonality is retail sales, which often increase between September to December and will decrease between January and February. The presence of seasonality is shown on the plot below (fig. 1).

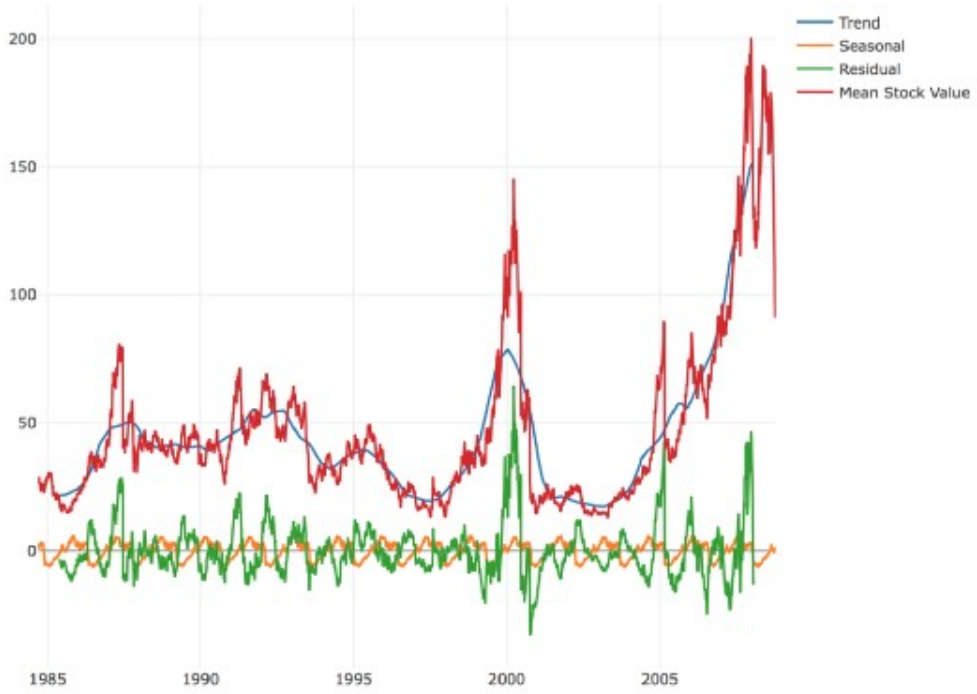

Fig. 1 Trend and seasonality in Apple Stock Prices

The red line plots the data from apple stock prices. Blue line indicates the trend, whereas the orange the seasonality (yearly) about them.

1.4.3 Cyclicity

A cyclic pattern exists when data rises and falls that are not of fixed period. The duration of these fluctuations is usually of at least 2 years and is caused by circumstances, which repeat in irregular intervals. For example, the stock market tends to cycle between periods of high and low values, but there is no set amount of time between those fluctuations.

1.4.4 Irregularity

Irregular component, also referred as White noise, refers to variations which occur due to unpredictable factors and results from short term fluctuations. It ,also, does not repeat in particular patterns.

These components can help guide the testing and estimation methods, that will be used during time series modeling and analysis. As Shumay and Stoffer mentioned, “the first step in any time series investigation always involves careful scrutiny of the recorded data plotted over time. This scrutiny often suggests the method of analysis as well as statistics that will be of use in summarizing the information in the data.”

1.5 Applications

There are plenty of uses of time series. Nominally, some of them are statistics, signal processing, pattern recognition, financial issues, weather forecasting, earthquake prediction, economics (unemployment rates) , social sciences (migration data), epidemiology (disease rates) and the physical science (global temperatures). In general, there is a large domain of applied science and engineering ,which includes these measurements.

1.6 Forecast

Time series analysis consists of methods which analyze the data points, so as to extract information and statistics and, also, recognize characteristics of the data.[6] The forecasting of time series refers to the use of a model for predicting future values, based on previous data. More specifically, forecasting is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts. The goal is to go beyond, knowing what has happened to providing a best assessment of what will happen in the future. A company needs to develop a forecasting system that involves several approaches to predict uncertain events.

1.7 Steps of forecast

A forecasting system usually involves five basic steps. [7]

1.7.1 Definition of the problem.

The definition of a problem consists the most difficult part of forecasting. It is necessary to understand the way that the forecasts will be used, who requires the forecasts, and how the forecasting function fits within the organization, which requires the prediction. The analyst needs to schedule the collection of data, maintain databases and ,finally, use the forecasts for the analysis.

1.7.2 Gathering information.

There are many kind of information that can be gathered, such as statistical data, accumulated expertise of analysts. However, it is quite difficult to obtain enough data so as to fit well a model. Often, old data are not so useful due to changes that has been occurred.

1.7.3 Preliminary analysis.

Plotting the data helps extracting the necessary information about significant trends, seasonality, or business cycles. Observations that are very different from the majority of the observations in the time series, also referred as outliers, will be revealed. They may be errors, or they may simply be unusual.

1.7.4 Choosing and fitting models.

The ideal model for the forecasting depends on the availability of data, the strength of relationships between the variable to be predicted and any depending variables, and the goal that has been set for the forecast. For getting the best analysis, it is common to compare more than two potential models. Each model consists of certain assumptions and involves parameters that have to be estimated by the given data.

1.7.5 Using and evaluating a forecasting model.

After selecting the model and setting the parameters, forecast is made on the selected time series. The estimation about the future values will be used for the initial purpose.

Note that we cannot forecast everything. There must be a set of conditions to be the base of the prediction. Random events, such as lottery, incomplete information about data and its relations are not acceptable for the forecast.

1.8 Applications of forecasting

Forecasting is a method, which bases on past and present data for predicting future behavior. As a result, this method has a significant role for both organizations and companies. It could support the development of business and so lead to success, or to failure when the produced data are inaccurate.[8] Specifically, planning and setting new goals is a major part of a business. For example, in a sales business forecasting will help in analyzing consumer's behavior based on past purchases or responses, predicting potential product demand and reducing the risk of a company's decisions .

Generally, businesses are turning to predictive analysis to help solve difficult problems and uncover new opportunities. Some more uses include, the detection of fraud in the base of the company. Gathering the past data and combining them can make more accurate the pattern detection and so prevent almost every criminal action. Outbalancing other competitors is one more use. Prediction provides valuable information that highlight the advantages over other companies. Based on these, the effectiveness of the company will grow and more profitable targets will be set. To sum up, forecast can help undertake critical decisions on how to allocate resources and set overhead levels within a business.

Beyond businesses, some other applications of forecasting are: [9]

- Earthquake prediction. It refers to the definition of parameters for the next earthquake, including time, longitude and latitude and the frequency.
- Sales forecasting. It helps improving the planning of a company, short-term or long-term.
- Political forecasting. It is very useful for showing the election outcome and giving the necessary information to every party.
- Economic forecasting. It refers to the process for making predictions for the economy. Not only businesses, but also governments use it for setting their goals and strategies for the next periods of time.

- Weather forecasting. Warnings about the weather are important for the every day life, but mostly for properties of people.
- Unemployment forecasting for a country.

1.9 Similarity

Similarity is defined as the distance between various data points. It reflects the strength of relationship among the data items and sets how similar or different these patterns are. The distance metric, which produced by a specific metric function, consists the base in clustering techniques.

Clustering bases on similarity measures to group data points, which have a small distance, together. The form of clusters has been defined so as objects in the same cluster have the minimum distance value, while the distance among any other object of other clusters has the maximum. The importance of this method is, not only to group similar data (fig. 2), but also data that differ in a larger degree (fig. 3) .

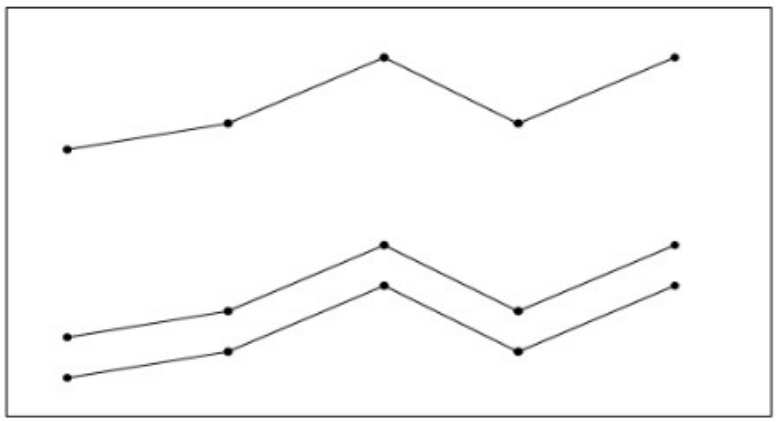

Fig. 2 Clustering of similar data

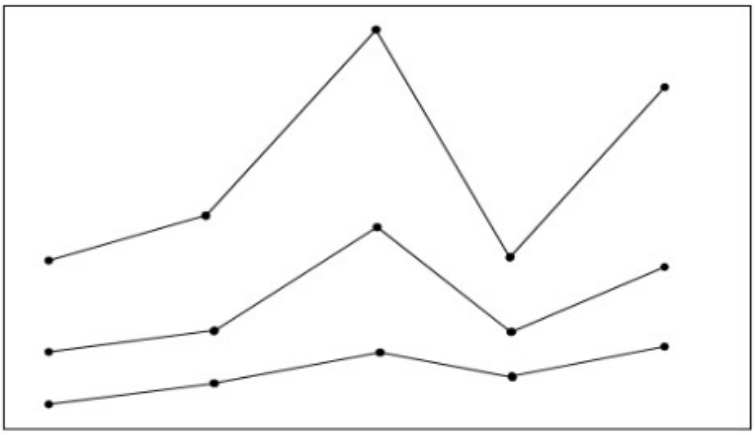

Fig. 3 Clustering of data that differ

1.10 Similarity measures

There are many similarity measures that can be used for clustering. The most common are:[10]

- **Euclidean distance**. It refers to the straight line between two points in Euclidean space and determines the root of square differences between the coordinates of a pair of objects.
- **Manhattan distance**. It calculates the absolute differences between coordinates of pair of data objects. In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2), it is $|x_1 - x_2| + |y_1 - y_2|$.
- **Jaccard distance**. It is defined as the size of the intersection divided by the size of the union of the sample sets.
- **Dynamic time warping.** It calculates the similarity between sequences, which may differ in length, using an optimal match with certain restrictions. [11]

1.11 Applications of similarity

As mentioned above, similarity is defined as the base of clustering technique. Below, some applications of clustering will be cited: [12]

- **Spamming**. K-means clustering is the effective method for identifying spam emails. The way that it works is simple. After collecting all the different sections of an email (header, sender, content), it organizes them and classifies them to identify the spams. As a result, personal data of users are more protected.
- **Marketing**. Especially, in this era personalized marketing is defined as the major policy of all businesses. Clustering algorithms are able to group together people with similar traits and likelihood to purchase a specific product.
- **Cyber profiling criminals**. It refers to the process of collecting information from individuals so as to analyze their behavior and other relations. That will help to investigate criminal profiles that may relate with a crime.

CHAPTER 2

State-of-the-art algorithms

2.1 Forecasting methods

Time series forecasting is a considerable phenomenon and extremely useful for almost every science. As mentioned above, there are many applications, where they can be applied. For this reason, there are plenty of techniques for modeling time series and then predict their evolution through time. Analysts are responsible for choosing the appropriate model, based on their objectives, fitting the training set and make the predictions based on the training set. In this section, we will analyze some of the most common techniques for time series forecasting. [13] The programming language on which they are developed are python and R.

2.1.1 Naïve approach

Naïve model is often used as benchmark model and is considered as the most cost-effective model for forecasting. Its forecasts are equal to the last observed value and it is suitable for economic and financial time series, with difficult patterns for prediction. The notation of naïve approach in time series is: $\hat{Y}(t+h|t) = Y(t)$'

The following plot shows the prediction that have occurred for 2007 year, for the electrical equipment manufacturing. (fig. 4)

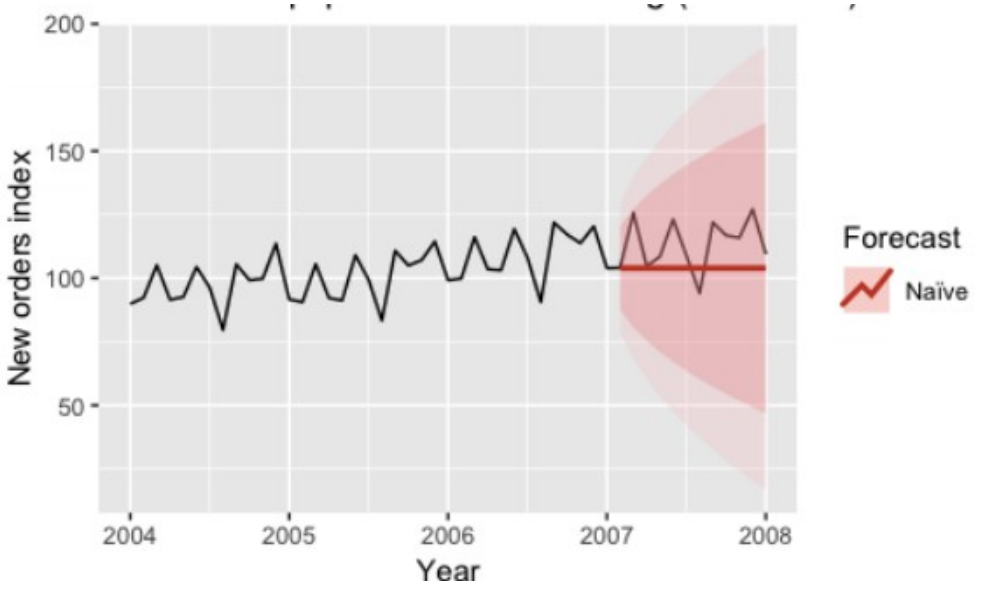

Fig. 4 Naïve approach for electrical equipment manufacturing

2.1.2 Exponential smoothing

As the name of the model reveals, exponential smoothing technique smooths time series, using an exponential smoothing window. It is working as a low-pass filter, which tapering high frequency noise. Mathematically, when a data sequence multiplied with the window function, the result is the common part that is overlapped. The forecasts equal to past observations that weighted equally and weights are decreasing exponentially over time using exponential functions.

At the beginning of time t = 0, the simplest form of exponential smoothing is given by the equations that shows the picture below (fig. 5).

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)s_{t-1},\ t > 0$$

where α is the *smoothing factor*, and $0 < \alpha < 1$.

Fig. 5 Exponential smoothing equations

The following plot shows the prediction that have occurred for 2007 year, for the same data set of electrical equipment manufacturing. (fig. 6)

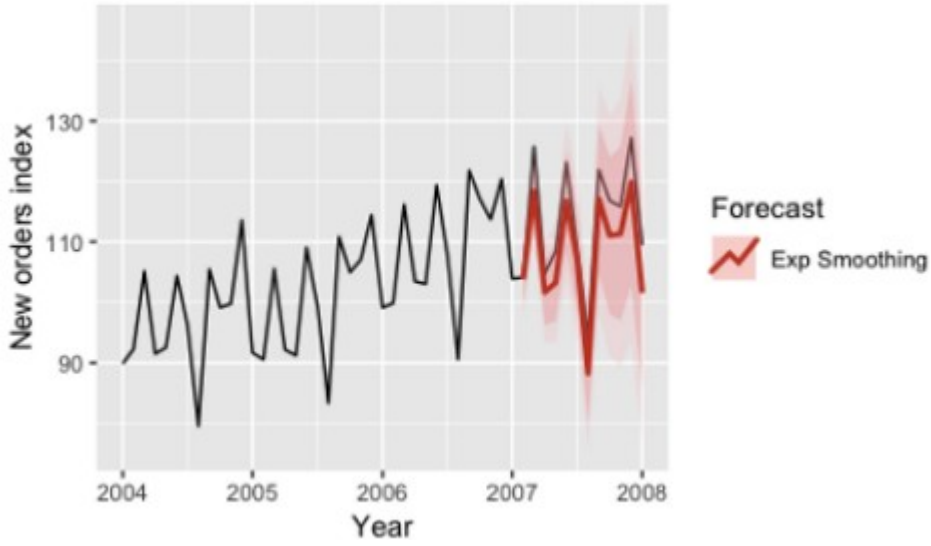

Fig. 6 Exponential smoothing for electrical equipment manufacturing

2.1.3 ARIMA-SARIMA

ARIMA model aims to describe the auto-correlation in the data and often is applied on data. The name stands for Auto-Regressive Integrated Moving Average.

The AutoRegressive model represents the predictions about the linear combination of past values of the variable. The Moving Average model, represents the forecasts correspond to a linear combination of errors that has been occurred from previous forecasts in the past. There two approaches are combined, so as the model to fit better the data. Finally, the 'I' indicates that data values have been replaced, with the difference between their values and the previous values.

The extension of ARIMA, SARIMA, has an extra parameter by adding a linear combination of seasonal past values and/or forecast errors.

The ARIMA forecasting equation for a stationary time series is a linear (i.e., regression-type) equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors. Specifically,

Predicted value of Y = a constant and/or a weighted sum of one or more recent values of Y and/or a weighted sum of one or more recent values of the errors.

A nonseasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

- **p** is the number of auto-regressive terms,
- **d** is the number of nonseasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation.

Mathematically, in terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

The plots below show the forecasts using ARIMA and SARIMA models for seasonally adjusted time series (fig. 6), (fig. 7).

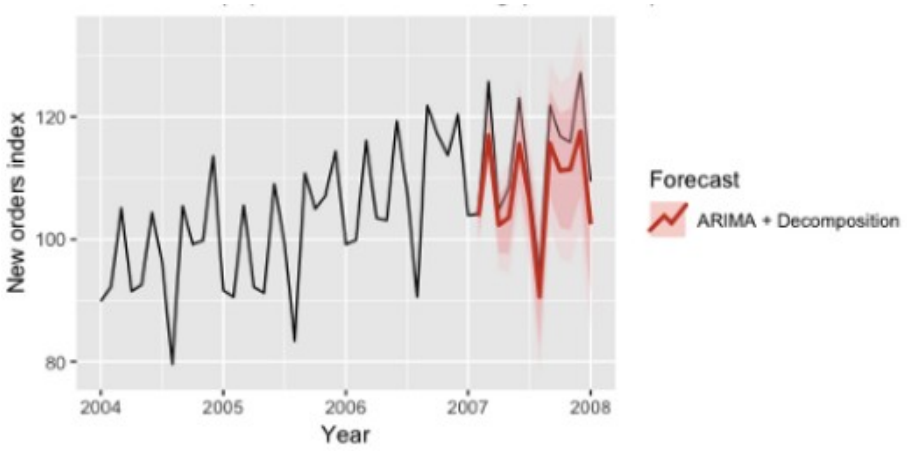

Fig. 6 ARIMA model for electrical equipment manufacturing

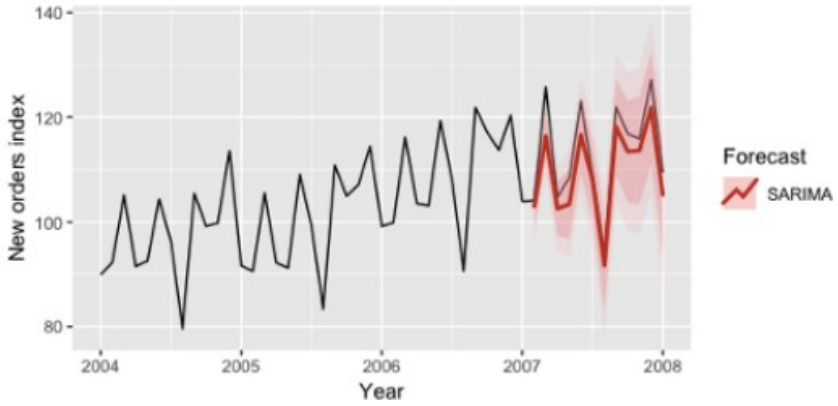

Fig. 7 SARIMA model for electrical equipment manufacturing

2.1.4 LSTM

LSTM, also known as long-short term memory, is a type of recurrent neural network (RNN) (fig. 8). This model is capable of learning the order dependence in sequence prediction problems. It is used in machine translation, speech recognition and more. The plot (fig. 9) below shows the predicted values of a test set after fitting LSTM model:

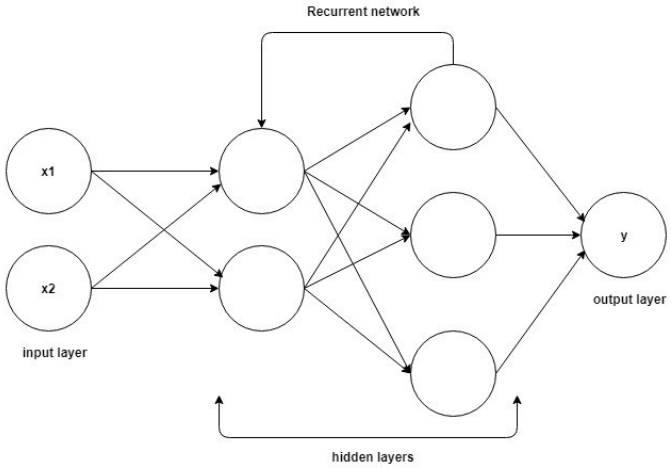

Fig. 8 LSTM RNN

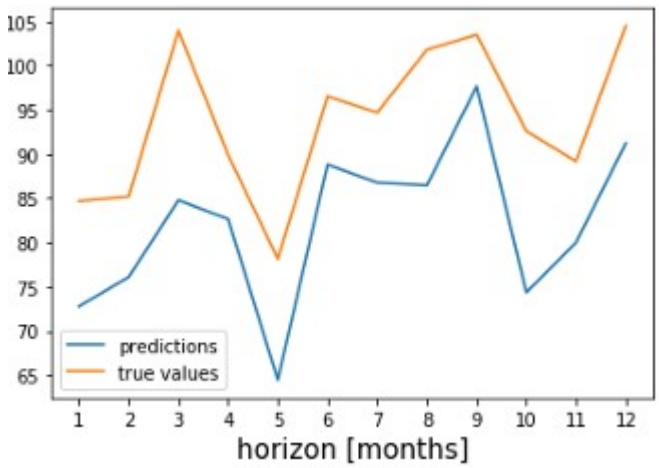

Fig. 9 LSTM forecasting model

2.1.5 Prophet

Prophet model is an open-source software released by Facebook's core data science team. It is based on an additive model, a non-parametric regression method, where the predictor is constructed according to information derived from the data. It can be decomposed as,

$\boldsymbol{y(t) = g(t) + s(t) + h(t) + \epsilon_t}$**,** where:

- $g(t)$ models trend, which describes long-term increase or decrease in the data. Prophet incorporates two trend models, a saturating growth model and a piece wise linear model, which depends on the type of forecasting problem.
- $s(t)$ models seasonality with Fourier series, which describes how data is affected by seasonal factors.
- $h(t)$ models the effects of holidays or large events that impact business time series (e.g. new product launch, Black Friday etc.)
- ϵ_t represents an irreducible error term.

The most important characteristic is that prophet model is flexible with missing values of data sets and shifts that may occur to trends, and well handler of outliers or intense changes in time series. Prophet procedure is implemented in both R and python, but they share the same underlying Stan code for fitting. Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. An example of prophet forecasting is represented from the next figure (fig. 10):

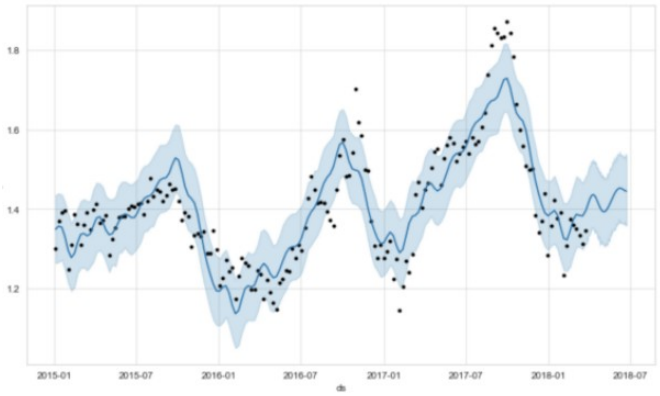

Fig. 10 Prophet model

2.2 Similarity methods

The estimation of similarity among data sets, which is based in distance measures, is a common data mining task. A similarity measure is a relation between a pair of data sets and a scalar number. There are common intervals, that are used to mapping the similarity, [-1, 1] or [0, 1], where the biggest number indicates the maximum of similarity.

Before analyzing distance functions, it is important to understand, if a distance measure can be considered metric. A metric function has to obey to four fundamental properties:

1. Non-negativity. $f(x,y) \geq 0$
2. Identity. $f(x,y) = 0$ if and only if $x = y$
3. Symmetry. $f(x,y) = f(y,x)$
4. Triangle inequality. $f(x, z) \leq f(x, y) + f(y, z)$

If any of these is not obeyed, then the distance is non-metric.

Below we will analyze some distance functions, used in Euclidean space. [14]

2.2.1 Euclidean distance

It is considered to be the most used distance function in man applications. Mathematically, it is the straight line distance between two points. [15]In Cartesian coordinates, if $\mathbf{p} = (p_1, p_2, ..., p_n)$ and $\mathbf{q} = (q_1, q_2, ..., q_n)$ are two points in Euclidean n-space, then their euclidean distance is calculated as shown in picture (fig. 11):

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Fig. 11 Euclidean distance calculation

Despite the fact that euclidean distance is so common, there are some disadvantages about it. The comparison can be done only to time series with same length, limiting many applications. In addition, it doesn't handle outliers or noise, while it is really sensitive to signal transformations as shifting.

2.2.2 Manhattan distance

Manhattan distance, also known as taxicab metric, is a kind of geometry. It is calculated as the sum of the absolute differences of the Cartesian coordinates of two points. Mathematically, the distance d between two vectors p,q is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes:

$$d(p,q) = ||p\text{-}q|| = \sum_{i=1}^{n} |pi - qi| \quad ,$$

where (p,q) are vectors

$p = (p_1, p_2, \ldots, p_n)$ and $q = q(q_1, q_2, \ldots, q_n)$.

The following picture (fig. 12) represent taxicab geometry, versus Euclidean distance.[16] The shortest path for Euclidean distance is indicated by the green line, whereas the red, blue and yellow are lines of same distance and belong to taxicab geometry.

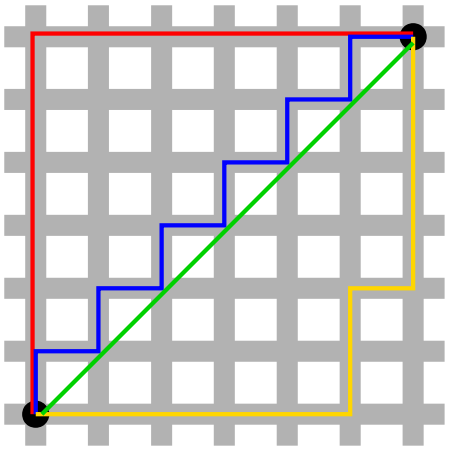

Fig. 12 Taxicab geometry vs Euclidean distance

Some frequent applications of Manhattan distance are:

1. **Regression analysis**: It is used in linear regression to find a straight line that fits a given set of points.
2. **Compressed sensing**: In solving an under determined system of linear equations, the regularization term for the parameter vector is expressed in terms of Manhattan distance. This approach appears in the signal recovery framework called compressed sensing.
3. **Frequency distribution**: It is used to assess the differences in discrete frequency distributions.

2.2.3 Mahalanobis distance

Mahalanobis distance is used for multivariate data, where there are more than one variables at the same time. For its calculation, is used both the matrix of distances of an observation and the mean, and the covariance matrix.

Specifically, the Mahalanobis distance of an observation $\vec{x}=(x1,x2,\ldots,xn)^{T}$ from a data set with mean $\vec{\mu}=(\mu1,\mu2,\ldots,\mu n)^{T}$ and covariance matrix S, is defined as (fig. 13) [17]:

$$D_M(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1}(\vec{x}-\vec{\mu})}.$$

Fig. 13 Mahalanobis calculation

Applications of Mahalanobis distance can be found at:

1. **Clustering analysis.**
2. **Classification techniques**. To examine if a data point belongs to a class or not, the covariance matrix is being calculated, based on the samples that belong to each class. Then, for the given test point, the Mahalanobis distance is being calculated for each class and the smaller is chosen for classifying the test point.
3. **Detection of outliers**, especially in the development of linear regression analysis. A data point that has a greater Mahalanobis distance from the rest of the set of points influence more the slope or coefficients of the regression equation.

2.2.4 Dynamic time warping (DTW)

DTW method is more flexible to the similarity computation than the others techniques. [18]The major difference is that it compares series with different length and replaces the one-to-one point comparison (fig. 14), with one-to-many (fig. 15). It, also, recognizes similar shapes, even with the presence of signal transformations, such as shift or scale.

Practically, DTW calculates the optimal match between two time series, based on certain restrictions. [19]The optimal match is chosen by the minimal cost that is produced, which is the sum of the absolute differences for each matched pair of indices, between their values. Apart from the similarity measure, dynamic time warping algorithm produces a warping path, according to which the two series may be aligned in time. The last element of this path corresponds to the calculated distance.

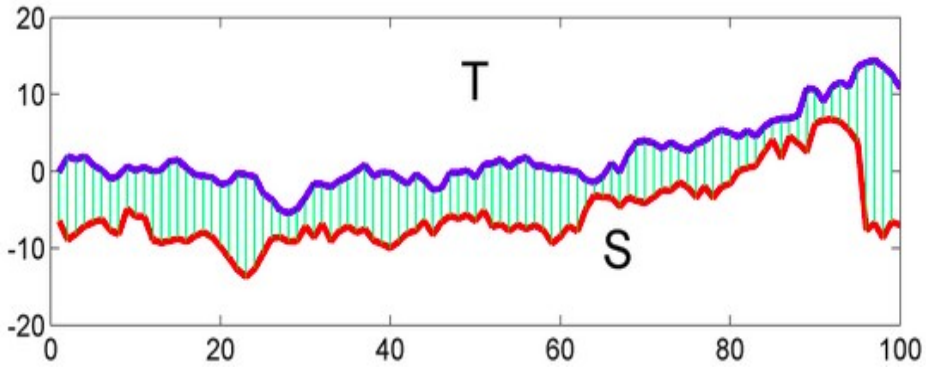

Fig.14 Euclidean distance: one-to-one

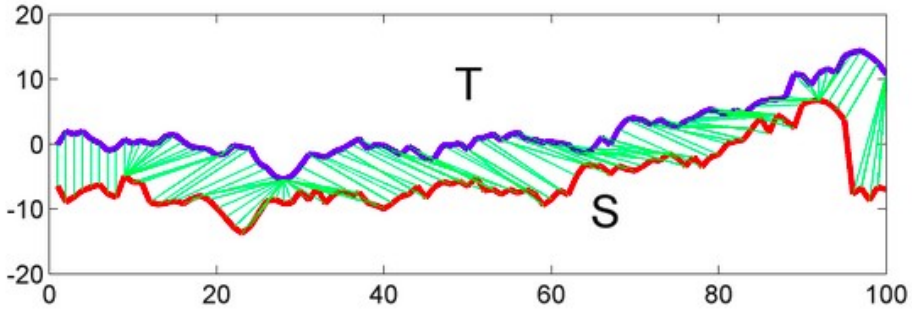

Fig. 15 DTW calculation: one-to-many

DTW can be applied to many different domains, such as:

1. **Speech recognition**. Note that there are different speeds of speaking.
2. **Signature recognition**.
3. **Financial markets**, comparing stock trading data over similar time frames, even if they do not match up perfectly.
4. **Wearable fitness trackers**, calculating more accurately a walker's speed and the number of steps, even if there is a variation of speed over time.

CHAPTER 3

Software analysis

3.1 Introduction

The process of the analysis based on two widespread models, Prophet model and Dynamic Time Warping model. To demonstrate the function of the two algorithms, a basic user interface has been set. Users interact with the program through the interface, defining the parameters of the execution. They select the dataset file, the model of the analysis and the data on which it will be based.

The archiving of the input data is achieved with a '.txt' file, which works as a database. The program connects with database, in order to initialize each execution. After completing the analysis, the results are saved in a temporary file, which will be used for creating the personalized '.txt' file for user, or it will be discarded.

To conclude, the current thesis is about displaying a simulation of how these models react to input of data sets that user gives in every execution.

3.2 Algorithms

3.2.1 Prophet model

The prophet model has specific requirements to make predictions. [20] First and foremost, the input is a dataframe with two columns, ‘ds’, ‘y’. The ‘ds’ column should be a format that is expected by pandas. Ideally, it should apply to the following format, YYYY-MM-DD for a date, or YYYY-MM-DD HH:MM:SS for a timestamp. The ‘y’, which represents the data that will be forecasted, must be numeric [21]. So, in the program each data to be forecasted from the list is set to this format.

for every data in data_to_forecast_list:

data.columns = ['ds', 'y']

The next step is about setting the prediction size, the number of periods to forecast forward. For example, if periods=60,the model will predict the views of the next 6o days.

prediction_size = 300

After completing the commands that mentioned above, prophet model is initialized. The data are fitted and the model starts makes predictions.

m = Prophet()

m.fit(data)

future = m.make_future_dataframe(periods = prediction_size)

forecast = m.predict(future)

The ‘make_future_dataframe’ command creates a new dataframe that extends into future a specified number of days. By default , it includes the dates from the history.

The ‘predict’ command returns a pandas dataframe and consists of a lot of fields. With this method, each row in future will be assigned a predicted value, which name is yhat and has specific range. Yhat_lower is the lower bound of prediction, whereas yhat_upper is the upper bound. For instance, in order to see the last five (5) predicted values, a tail function is used.

```
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())
```

Plotting the forecast will visualize the results of the analysis, whereas plotting the components will help recognizing the seasonality.

```
for data in forecast_list:
    m.plot(data)
    m.plot_components(data)
```

3.2.2 Dynamic Time Warping (DTW) model

The DTW analysis for the input time series bases on dtaidtistance library, which is used in DTAI Research group. This library is offered in pure python implementation. [22]

Specifically, the command ,that is used in the program

```
distance, paths = dtw. warping_paths(timeSeries1, timeSeries2)
```

,measures the distance of the two sequences of numbers (timeSeries1, timeSeries2) and calculates all the possible warping paths. It takes as arguments the two timeseries as lists.

In addition, in the current algorithm has been applied a library for normalization of the result. ‘Sklearn.preprocessing’ is a package provides several common utility functions and transformer classes to change the vectors into representations that is more suitable for the calculations [23]. ‘Sklearn.preprocessing. normalize’ is the basic command that scales the input time series individually to unit norm.

```
timeSeries_normalized = preprocessing.normalize([timeSeries1])
```

An example of normalization of an execution is presented below. The input timeseries of figure 16, turn into the normalization shown on figure 17. Similarly, figures 18 and 19 represents timeseries2.

```
0       11.971429
1        8.662500
2       12.375000
3       12.225000
4        5.808333
           ...
```

Fig. 16 Input timeseries1

```
[0.05613959 0.04062249 0.05803212 0.0573287  0.02723797 0.06057225
 0.0760101  0.06342501 0.0629756  0.06194001 0.03409631 0.03405724
 0.05515983 0.04331893 0.06321007 0.04894629 0.10574742 0.08374602
```

Fig. 17 Normalization of timeseries1

```
0       880.666667
1       918.083333
2       896.791667
3       740.916667
4       880.083333
            ...
```

Fig. 18 Input timeseries2

```
[0.0527328  0.05497324 0.05369834 0.04436481 0.05269787 0.0406324
 0.03619511 0.03619392 0.08923859 0.05054724 0.05721369 0.06042217
 0.04598402 0.04603641 0.04005607 0.04030307 0.03574234 0.05308957
```

Fig. 19 Normalization of timeseries2

CHAPTER 4

Documentation

Documentation has a major role for a productive system, which will be maintained by other people and whom, will have responsibilities for its support. There are complex algorithms ambiguous codes, that have to be analyzed and commented.

First and foremost, the source code of the current program is written in pure python programming language. In order to be functional, there are some installations that need to be done.

4.1 Installations requirements

The first requirement is the installation of Python and its dependencies. Specifically, with the command ‘pip install name_of_package’ in terminal, **numpy**, **cython, pandas, matplotlib** and **pystan** are installed in each execution.

Especially for the prophet model, as mentioned above, it fits data in Stan so as to get forecasts. Pystan provides an interface to Stan. After meeting these requirements, fbprophet can be installed with the command, ‘pip install fbprophet’.

4.2 File requirements

The input file of the data set has to apply to specific requirements. The type should be a ‘.csv’ file and the separating element ‘;’. As the analysis bases on time series, ‘Date’ column is necessary among with ‘Time’, as there exist measurements through time for the same day.

The name of the file, which produced by saving the result of the analysis, depends on user’s preference. However, it should have a specific type, ‘.txt’.

4.3 Execution of the program

There are four files in total that consists the current program. ‘UI.py’, ‘prohet.py’, ‘dtw.py’ and ‘edit_files.py’. The execution begins from the terminal with the command ‘python UI.py’, which gives rise to activate the interface. All the interactions are happening through this, and by choosing the ‘exit’ from menu, the program stops.

4.4 Interface

User interface (UI) is based on tkinter package. Tkinter package is the standard python interface to the tk GUI toolkit. UI consists the means in which the user will control the program and it creates the dialogs between the user and the system.

Specifically, the user has access to the main menu and the help menu (fig.20).

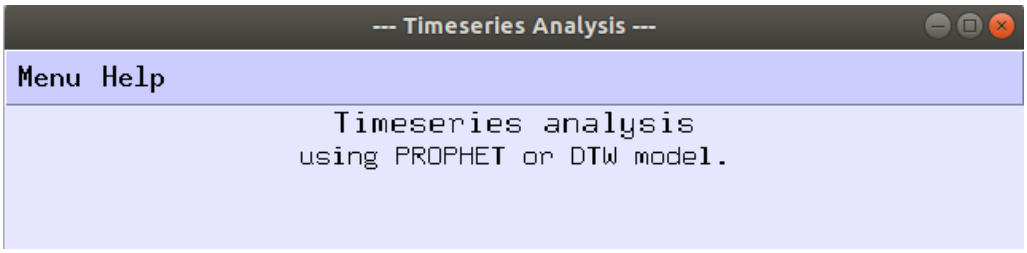

Fig. 20 Main view

Main menu (fig. 21) has the choices that are referred below:

- **New_analysis**, creates a window for every new analysis, initializing at the same time the program.
- **Refresh**. Since a ‘back’ button for previous situation

  does not exist, the choice of refresh, resets the program.
- **Save**. User can save the results of the analysis.

  Through a pop-up window, he selects a path from t

  he system and sets a name of his choice.

  The file is saved with the ‘.txt’ extension.

- **Exit**. Exit button destroys the program and exits.

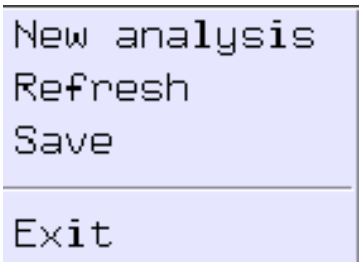

Fig. 21 Main menu

Help menu (fig. 22) provides user with information regarding the algorithm and the format of the input file.

- **Algorithm**, (fig. 23) describes the algorithm that user chose for the analysis. Some more information are shown as a pop-up window.
- **Format of file**,(fig. 24) sets the restriction that '.csv' file must obey. '.Csv' file constitute the file of data set. The function works as the algorithm pop-up window.

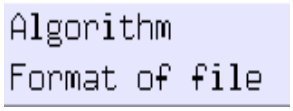

Fig. 22 Help menu

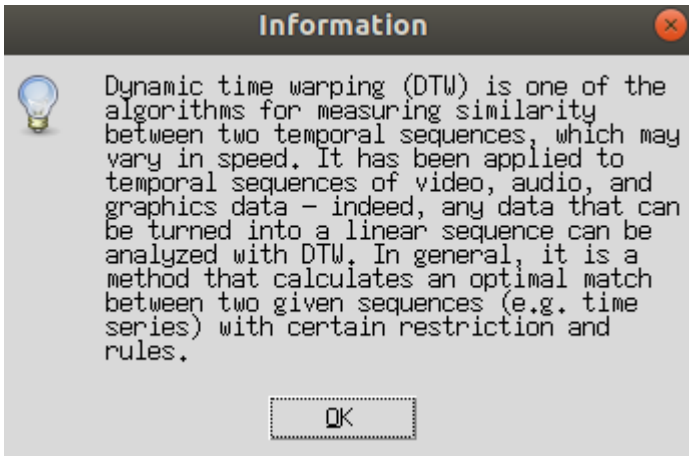

Fig. 23 Information about the algorithm

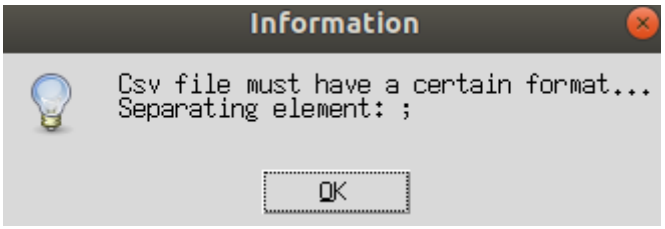

Fig. 24 Information about the format of the file

4.4.1 First menu

The choice of new analysis creates the first window/menu (fig. 25), of browsing the data set file and choosing the model for the analysis. Particularly, there are two buttons, that create dialogs between the user and the program. The first refers to a a drop-down menu with a pop-up window for browsing the file from all the available paths of the system, whereas the second is a drop-down menu, which displays the available models.

Fig. 25 First menu

After ‘open file’ is selected, it appears the following window (fig. 26).

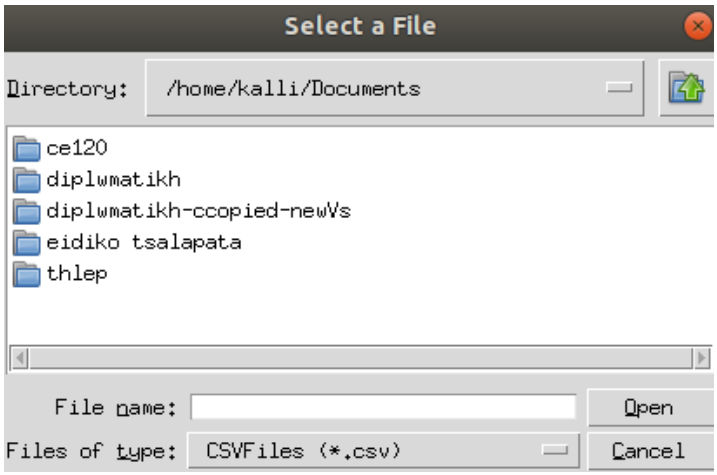

Fig. 26 File dialog window

The ‘next’ button is activated, after these two selections have been completed. The second window/menu is created.

4.4.2 Second menu

The second menu (fig. 27) deals with the selection of data for forecasting or measuring the similarity and executing the analysis.

A list of columns of the ‘.csv’ file is being displayed as a slide window. The buttons ‘Import’ and ‘Reset’, work as importing the data to the program and refreshing the list for the user.

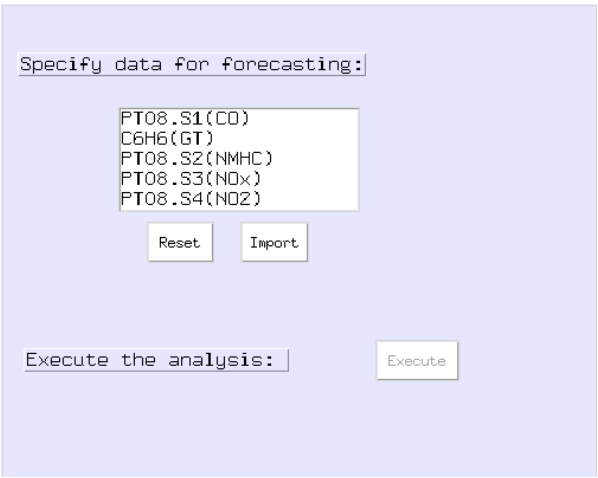

Fig. 27 Second menu

After import is selected, the ‘Execute’ button is activated (fig. 28). Depending on the algorithm that user have chosen, the program executes the ‘prophet.py’ file, or the ‘dtw.py’ file.

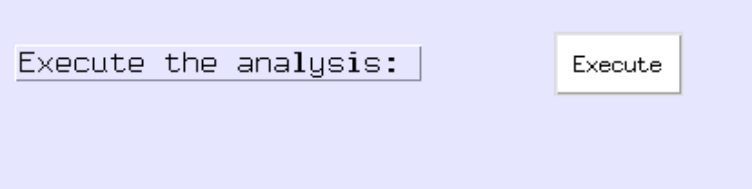

Fig. 28 Activation of ‘execute’ button

4.4.3 Display of results

The last view of the interface shows the results of the current analysis and concludes the process. Specifically, it displays the path of the file, the chosen model of the analysis, the data that it based on and the results (fig. 29).

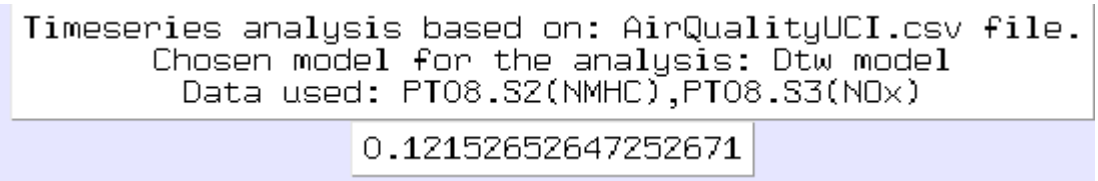

Fig. 29 Results of an analysis

4.5 Database.txt file

Database is essential for saving the input data and communicating with the program, in order to commit the execution. ‘Database.txt’ file simulates a database for the program. Specifically, it saves the inputs of user for every specific execution, in a specific structure. Each analysis imports the data, initializing the program and setting the parameters. For every new entry the structure is as followed (fig. 30):

- **New entry**. State that defines the new execution.
- **Data path**, of the input file ‘.csv’.
- **Model**, that will be used for the analysis (Prophet-Dtw).
- **Edited_file**, that occurs from editing the input ‘.csv’ file.
- **Time series**, on which the analysis will be applied.

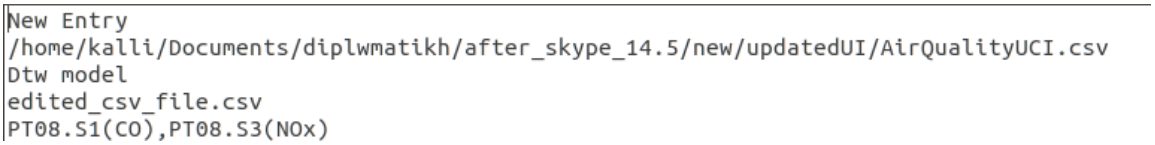

Fig. 30 New entry

The current file can not be accessed or edited by the users.

4.6 Edit_files (edit_file_csv) algorithm

Basic part of the program consists the edit_files program. It contains two (2) functions that edit given files.

4.6.1 Get_data() function

The first function **get_data()**, connects with the ‘database’ - database.txt file. There are no arguments. For every new entry reads the file and returns the data ,as a list, that will be used for the execution.

```
data = []
file_data = open(database.txt)
for every line in database:
        data.append(line)
file_data.close()
return data
```

4.6.2 Get_csv_columns(datapath)

The next function **get_csv_columns(datapath)** has two operations, editing the input '.csv' file and exporting its columns. It takes as an argument the datapath of the '.csv' file and returns the columns of the data set.

To begin with the editing of file, the steps that are followed concerns the cleaning of the file. First, the instances, where there is an empty value, are removed.

```
dataset = pd.read_csv(datapath, sep = ';')
dataset.dropna(axis = 1, how = 'all', inplace = True)        #column
dataset.dropna(axis = 0, how = 'all', inplace = True)        #row
```

The column 'Date' is parsed as a datetime and all measurements are turned into floats, as follows,

```
dataset['Date'] = pd.to_datetime(dataset['Date'])
for every column in dataset.columns:
    if dataset[col].dtypes == object:
        dataset[col] = dataset[col].str.replace(',','.')
        dataset[col] = dataset[col].astype('float')
```

Next in order is the removal of 'Time' column, if it dies exist and the aggregation of data by day, using their positive average.

```
daily_data = dataset.drop('Time', axis = 1)
daily_data = daily_data.groupby('Date')
daily_data = daily_data.apply(positive_average)
```

Finally, to remove the rest NaN values of each column, an array(column) of True / False is calculated. If there are more than 8 NaN values in each column or not, they are removed from data set.

```
column = (daily_data.isna().sum() <= 8).values
data = data.iloc[:, column]
data = data.dropna()            #removes missing values
```

The updated ‘.csv’ file is saved to improve the performance if the program.

```
data.to_csv(r'./edited_csv_file.csv')
```

The function returns the columns of the file as list.

```
columns = []
for column in data.columns:
        columns.append(column)
return columns
```

4.7 Prophet model

4.7.1 Prohet plots

The plots that occur after executing an analysis, based on prophet model have a specific description.

- Dark points/dots: Data points used to train the model.
- Wide blue area: Prediction that has occurred based on the prophet model.
- Light-blue shaded regions are the uncertainty intervals. Since we have not specified cap and/or floor, the model plots with uncertainty.

The image below (fig. 31) is an example of a plot based on time series of the log daily page views for the Wikipedia page for Peyton Manning.

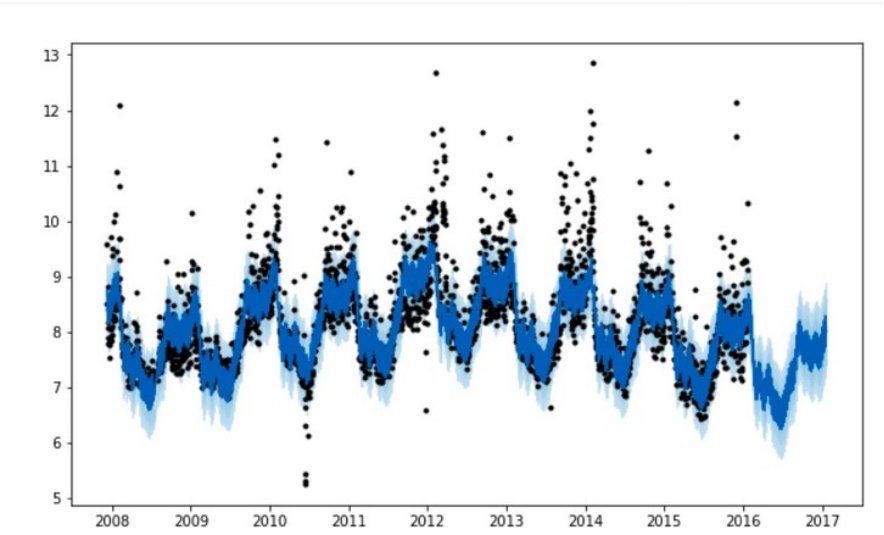

Fig. 31 Forecast of the log daily page views

4.7.2 Function plot_data(column)

Plot_data() is a function for plotting input data during time. It takes as an argument the data points of the corresponding column and displays the plot with some specifications.

```
plot.figure(figure_size = (17,8))
plot.plot(column)
plot.xlabel('Time')
plot.ylabel(column)
plot.grid(True)
plot.show()
```

4.8 DTW model

4.8.1 Function plot_data(title, column)

The same function has been defined for the DTW model with a second argument. It refers to the name of the column, so as to specify the y axis. The difference in the code is located in this command,

```
plot.ylabel(title).
```

CHAPTER 5

Evaluations

5.1 Dataset

The dataset that is used for the analysis is the ‘airquality.csv’. It contains information about the concentrations of gases, that has been recorded every hour per day. Generally, not all features of data sets are important to be analyzed. It is common, sometimes, to explore the behavior from only one characteristic, for example the price of the main data. For the chosen data set, NOx is the desirable for the analysis, not only for its length, but also for the effects that causes to both humans and the environment.

5.2 Prophet model

A simulation of an analysis with the prophet model is cited below.

After the interaction with the interface (fig. 32), ‘airquality.csv’ data set and prophet model have been defined.

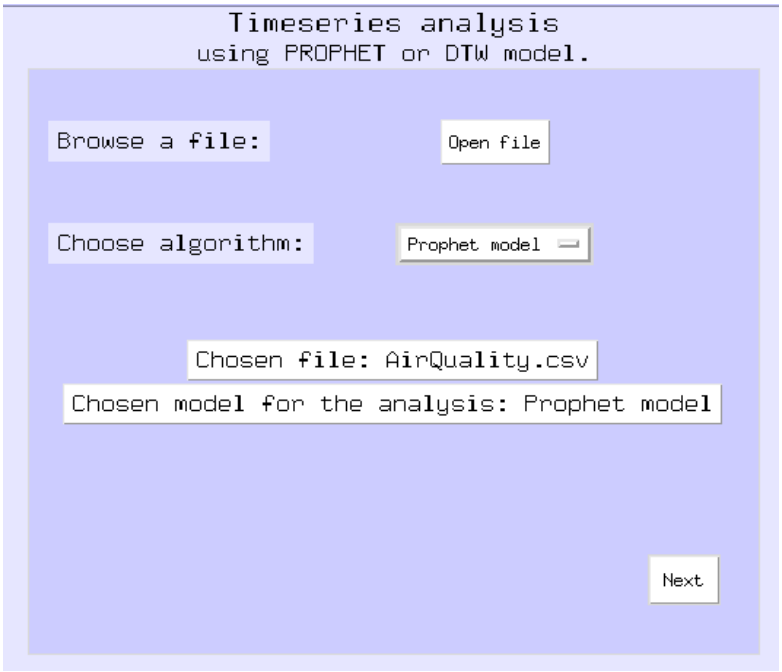

Fig. 32 Execution with prophet model

From the list of all features of the dataset, ‘PT08.S3(NOx)’ and ‘PT08.S5(O3)’ have been selected. The plots of components (fig. 33) and prediction (fig. 34) for the first one are:

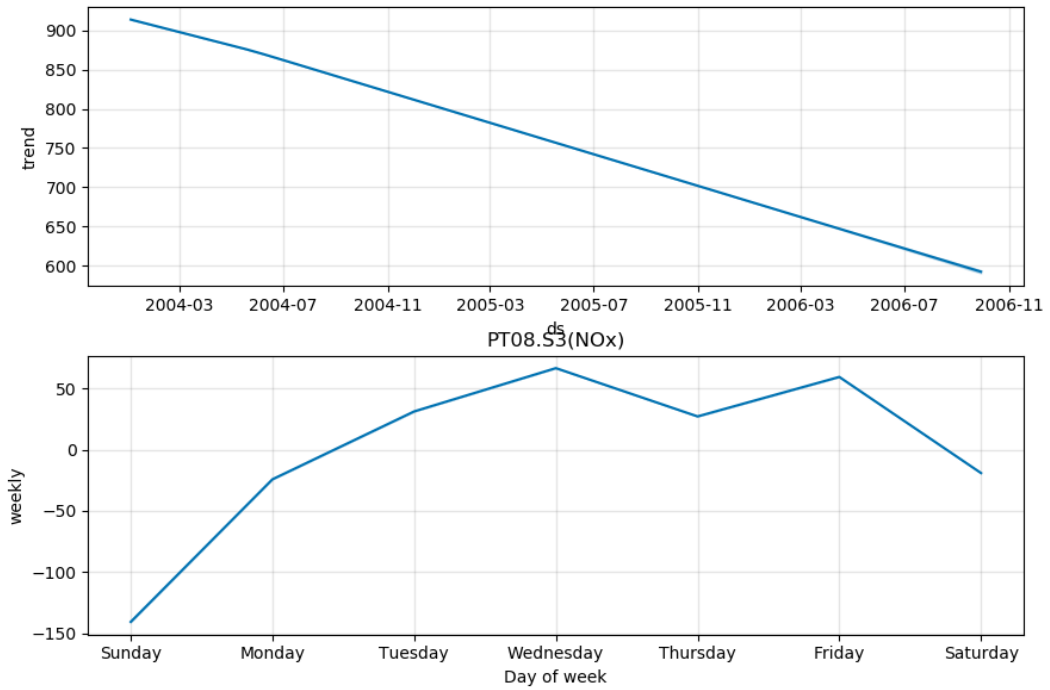

Fig. 33 Components plots of PT08.S3(NOx)

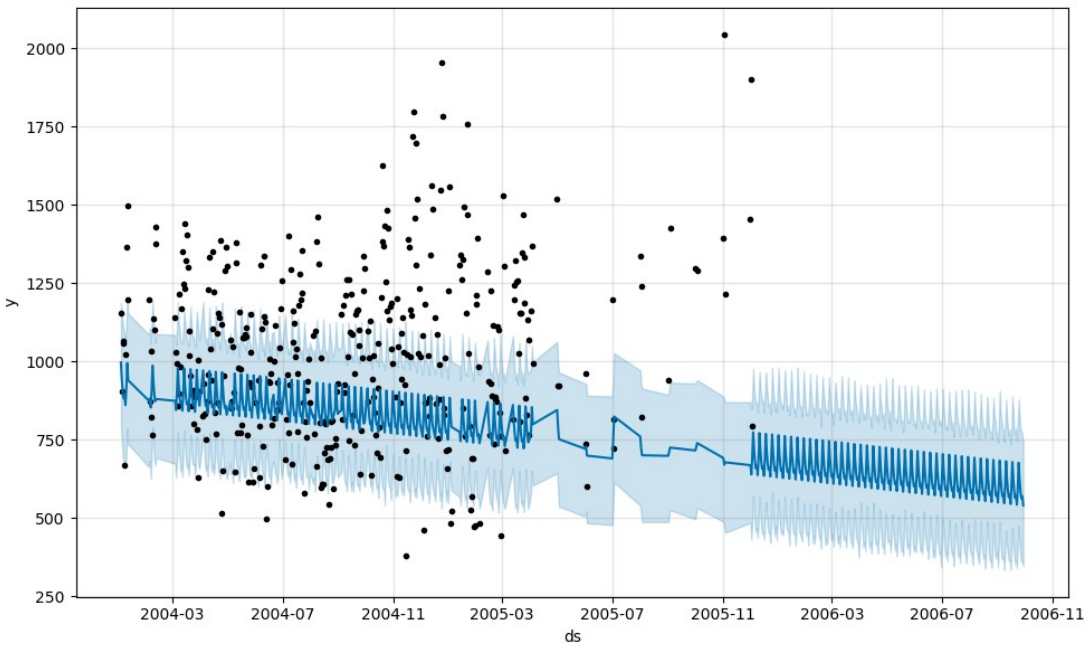

Fig. 34 Forecasting plot of PT08.S3(NOx)

The plots for the second column are presented below (fig. 35) (fig. 36):

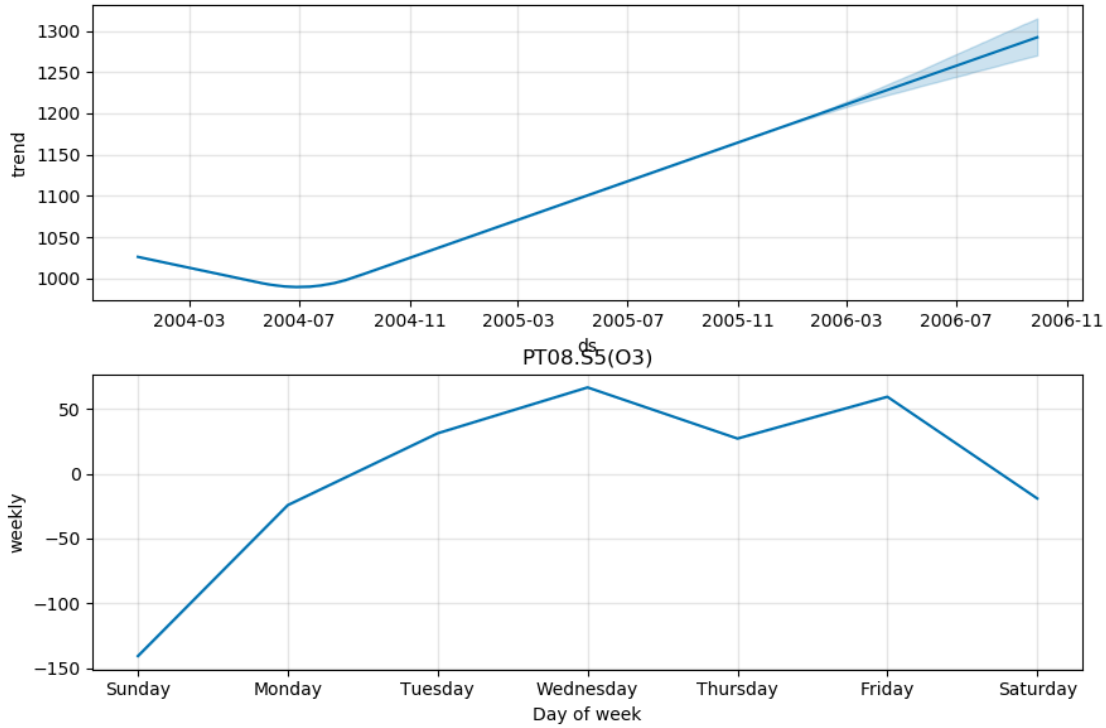

Fig. 35 Components plots of PT08.S5(O3)

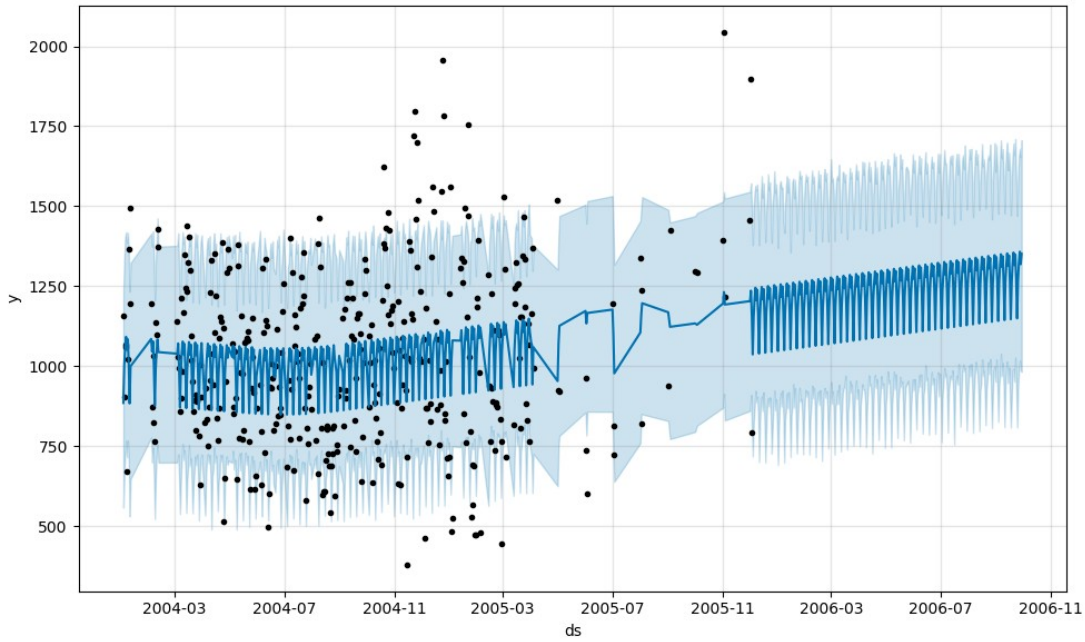

Fig. 36 Forecasting plot of PT08.S5(O3)

5.3 Explanation

5.3.1 'PT08.S3(NOx)'

Starting from the first feature, 'PT08.S3(NOx)' , the plot of its data points through time (fig. 37) is the following:

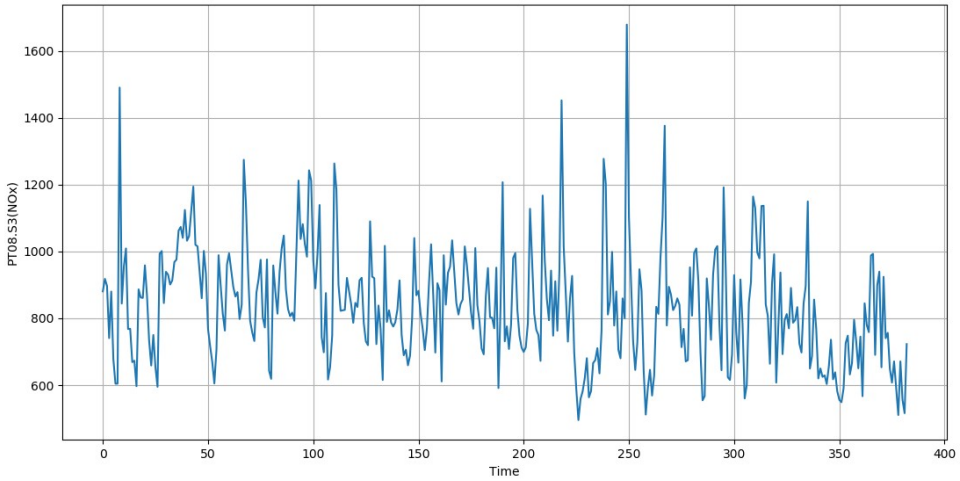

Fig. 37 Plot of PT08.S3(NOx)

It shows the evolution of data points through time.

The values of this set after initializing the column to 'ds', 'y', before the prediction (fig. 38) are:

	ds	y
378	2005-11-02	510.666667
379	2005-11-03	671.130435
380	2005-12-01	556.166667
381	2005-12-02	516.250000
382	2005-12-03	722.916667

Fig. 38 Data before prediction

After the prediction (fig. 39), including the old values and with periods of prediction equal to 300,

```
***************PT08.S3(NOx)***************
           ds        yhat  yhat_lower   yhat_upper
0  2004-01-04  996.754768  787.038343  1184.022442
1  2004-01-05  941.954237  734.583366  1148.386778
2  2004-01-06  903.624827  696.723753  1121.233491
3  2004-01-07  882.330670  668.119301  1072.344876
4  2004-01-08  889.906155  686.040001  1100.334761
             ds        yhat  yhat_lower  yhat_upper
678  2006-09-25  621.965049  426.862652  836.605649
679  2006-09-26  583.584273  372.830963  787.534243
680  2006-09-27  562.238750  376.281811  757.427533
681  2006-09-28  569.762869  373.007603  782.316082
682  2006-09-29  540.444063  337.007535  753.561975
*************** ***************
```

Fig. 39 Data after prediction

For the next periods, the predicted values are getting lower and this is being visualized on plot (fig. 34). Based on components plot, it is clear that there is a downward trend with no seasonality for ‘PT08.S3(NOx)’. Continuing with the weekly seasonality, the value in the plot for any particular day is how much y is added to the trend on that day, just due to weekly seasonality.

5.3.2 'PT08.S5(O3)'

The plot of data points of this column, through time (fig. 40) is the following:

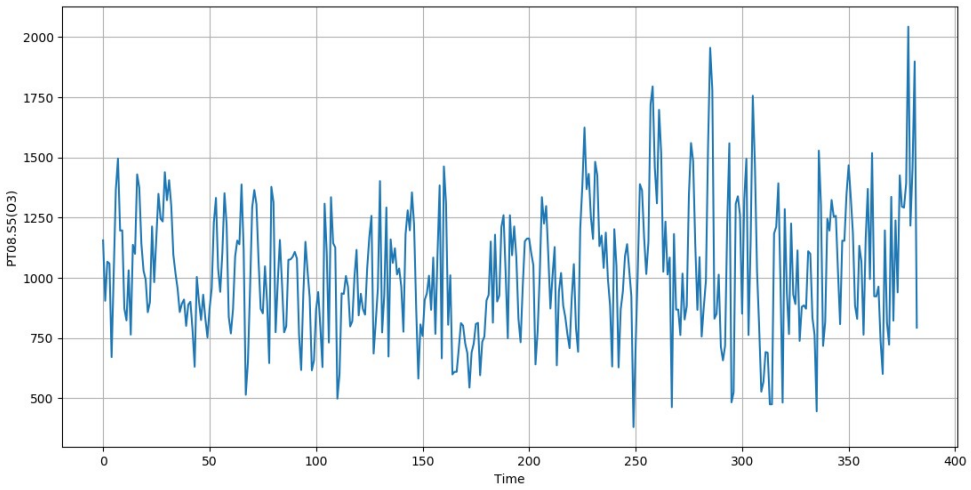

Fig. 40 Plot of PT08.S5(O3)

Similarly, the values of the column 'ds', 'y', before the prediction (fig. 41) are,

	ds	y
378	2005-11-02	2043.666667
379	2005-11-03	1216.869565
380	2005-12-01	1456.000000
381	2005-12-02	1899.125000
382	2005-12-03	792.875000

Fig. 41 Data before prediction

and after the prediction with prophet model (fig. 42):

```
***************PT08.S5(O3)***************
           ds         yhat  yhat_lower   yhat_upper
0  2004-01-04   885.299028  541.054948  1237.089722
1  2004-01-05  1001.565335  647.424214  1330.126566
2  2004-01-06  1056.783076  718.724055  1368.134064
3  2004-01-07  1091.930991  741.393900  1407.797003
4  2004-01-08  1052.246668  745.322413  1386.141688
             ds         yhat   yhat_lower   yhat_upper
678  2006-09-25  1266.439097   938.047040  1614.220849
679  2006-09-26  1322.272810   966.603167  1666.048990
680  2006-09-27  1358.036696  1000.476953  1679.655816
681  2006-09-28  1318.968344   979.545846  1664.162624
682  2006-09-29  1351.603819  1033.407043  1694.514367
```

Fig. 42 Data after prediction

As shown in the previous figure (fig. 42), the predicted values are increasing in time. This fact is explained through the corresponding plot (fig. 36). For the components plot, the trend has a cathodic route and then continues upwards with no seasonality,

5.4 DTW model

Likewise, below is reported an analysis that has occurred from the DTW model. The dataset remains the same and there is a difference in features. ‘PTO8.S1(CO)’ and ‘PTO8.S2(NMHC)’ have been selected for the simulation (fig. 43).

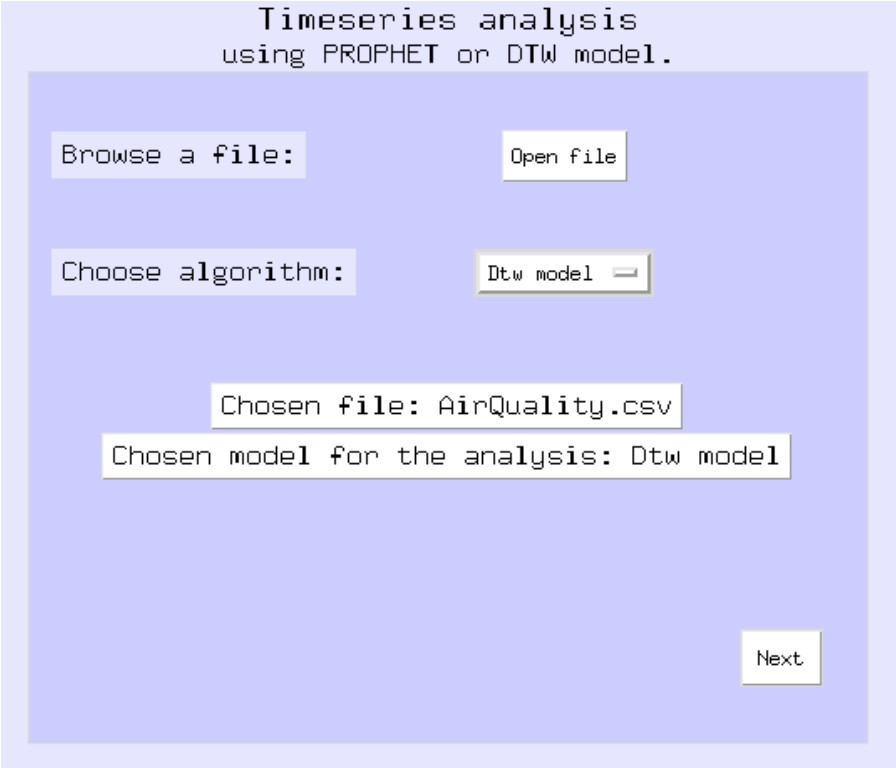

Fig. 43 Execution with DTW model

The calculated distance is cited below (fig. 44),

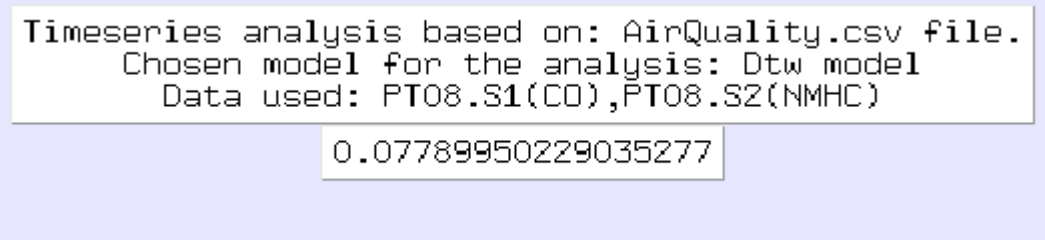

Fig. 44 Results of DTW analysis

5.5 Explanation

The calculated distance is a number that indicates how similar the two timeseries are. It is important to note, that this number has been normalized in a range [0,1]. So if distance is small, there is high degree of similarity. If the distance is a large number,then there is a small degree of similarity vice versa.

For the current execution, we notice that the occurring distance is small and close to zero (0). As a result, the two time series are quite similar.

CHAPTER 6

Conclusion and future improvements

6.1 Conclusion

The current thesis is about a functional simulation of timeseries analysis. Utilizing two state-of-the-art algorithms, we manage to understand and analyze the behavior of the data sets, explore future values, make forecasts and find the similarity between features. The results of these achievements consist the milestone for data mining, decision making supporting any field, from finance and biology, to the system of a government.

The first part of the thesis supports an analysis about forecasting. After searching and making an in-depth analysis of prediction models, we came to the conclusion that prophet model is accurate and fast, producing forecasts that are more reliable than other techniques. In some cases, other algorithms need further configurations in order to produce reasonable results. Furthermore, as mentioned at the previous chapters, prophet handles missing values and outliers, which are commonly found at data sets.

The second part is related with the similarity between time series. The fact that data sets have not the same length, led us at the choice of dynamic time warping model. It is almost impossible to compare time series of same length. That it the main advantage of DTW, replacing the one-to-one with one-to-many point comparisons.

6.2 Future improvements

For analysts and researchers, the functionality of the algorithms and the conclusions that are exported through them, define their work. It supports setting new goals and also optimize their brainstorming sessions.

However, the need for improvement and development is an integral part of the work, not only for collecting more exact results, but also for faster calculations. To begin with, as described above, the system connects with a file, where all the inputs are recorded. A future goal is about replacing this file with a database, which will organize all the collections of the data.

Furthermore, the plethora of state-of-the-art algorithms is a major motivation for improving the methods of analysis. Not only analysts, but also users will have a better perception of the behavior of timeseries. They will be able to compare the results and end up with the appropriate and most effective algorithms. The fact that there are different parameters will highlight the advantages and features of every model, based on the corresponding data set.

The last improvement refers to the restriction of the separator element of the ‘.csv’ file. The range of data sets will increase rapidly and as result it will be possible to make analysis on more fields.

Bibliography

[1] Article, Will Kenton, What Is a Time Series?, Mar 31 2020

[2] https://en.wikipedia.org/wiki/Time_series, Wikipedia, Time series, 11 June 2020

[3] Article, Swapnil Yeolekar, What is the importance and usage of Time Series? , April 23, 2017

[4] Data Analytics Blog, Introduction to the Fundamentals of Time Series Data and Analysis, September 13, 2019

[5] Digital platform www.toppr.com, Time Series Analysis Components of Time Series

[6] Book, Chris Chatfield, TIME-SERIES FORECASTING, Published October 25, 2000 by Chapman and Hall/CRC. ISBN 9781584880639

[7] Book, Rob J Hyndman and George Athanasopoulos, Monash University, Australia, Forecasting: Principles and Practice, Published by OTexts, 2ND EDITION

[8] Article, Johnny Lui, Forecasting 101: Why Every Business Needs It, Apr 14 2016

[9] Wikipedia, https://en.wikipedia.org/wiki/Forecasting, Forecasting

[10] Article, Shriya Gupta, Top 5 Distance Similarity Measures implementation in Machine Learning, Sep 30, 2019

[11] Wikipedia, https://en.wikipedia.org/wiki/Dynamic_time_warping, Dynamic time warping

[12] Article, https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/6224, Claire Whittaker , 7 Innovative Uses of Clustering Algorithms in the Real World, April 4, 2019

[13] Article, Towards data science, https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb, Davide Burba, An overview of time series forecasting models, Oct 3 2019

[14] Book, Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata and Alfredo Pulvirenti, Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining, Submitted: January 10th 2012, Reviewed: May 21st 2012, Published: September 12th 2012, DOI: 10.5772/49941, Published by IntechOpen

[15] https://en.wikipedia.org/wiki/Euclidean_distance, Wikipedia, Euclidean distance

[16] https://en.wikipedia.org/wiki/Taxicab_geometry, Wikipedia, Taxicab geometry

[17] https://en.wikipedia.org/wiki/Mahalanobis_distance, Wikipedia, Mahalanobis distance

[18] Article, https://medium.com/@shachiakyaagba_41915/dynamic-time-warping-with-time-series-1f5c05fb8950, Shachia Kyaagba, Dynamic Time Warping with Time Series, Sep 7 2018

[19] Research Article, Yingmin Li, Huiguo Chen , Zheqian Wu, Dynamic Time Warping Distance Method for Similarity Test of Multipoint Ground Motion Field, Hindawi Publishing Corporation, Mathematical Problems in Engineering,Volume 2010, Article ID 749517, Received 1 January 2010, Revised 6 April 2010, Accepted 16 May 2010,Academic Editor: Carlo Cattani

[20] Documentation, https://facebook.github.io/prophet/docs/quick_start.html, Prophet, Quick Start Python API

[21] Article, Towards data science, https://towardsdatascience.com/predicting-the-future-with-facebook-s-prophet-bdfe11af10ff, Parul Pandey, Predicting the 'Future' with Facebook's Prophet, Mar 22, 2019

[22] Documentation, https://dtaidistance.readthedocs.io/en/latest/sagedtw.html, DTAIDistance, DTW Distance Measure Between Two Series

[23] scikit-learn Machine Learning in Python, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html, 6.3.3. Normalization