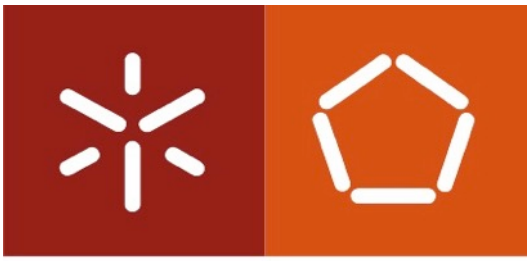

Universidade do Minho
Engineering School
Department of Information Systems

Luís Miguel Rocha Matos

Forecasting Human Entrances at a Commercial Store using facial recognition data

Master Thesis

Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação (MIEGSI)

Supervisor: Professor Paulo Cortez

Acknowledgements

I want to thank everyone that helped me elaborating this master thesis, in special my supervisor for all the hard work and availability during this thesis.

Also I want to thank Cristiano, Nuno, Carlos, Fábio, Diogo, José and Gonçalo for helping me in getting the events that occurred during the period of this study.

Also I want to thanks my mother, Maria, father, José, sister, Ana and all my friends for all the support and advices given during this study.

Last and not least, I wish to thank Duarte Duque and Nuno Santos, from the EXVA company, for kindly providing the human entrances data used in this study.

Abstract

Due to advances in Information Technology, there is a growing interest in the use of data mining to extract useful patterns from raw data in order to support decision making. In this work, a data mining approach was conducted aiming at the prediction of human entrances at a commercial store, as measured by an automatic video face detection system. In particular, a large number of experiments were held, targeting distinct types of human entrances (i.e., Female, Male, Both), forecasting periods (i.e., hourly and daily) and lookahead (horizon) predictions. Moreover, several forecasting methods were tested: conventional time series methods and time series models based on machine learning; a regression approach (e.g. using weather and special event data); and a hybrid approach that uses both time series (human entrances time lags) and regression variables. To achieve a robust evaluation, a rolling window scheme was adopted, which impled the use of a large number of model updates (trainings) and testing. For short term predictions (horizon of 1), the best performances were in general obtained by the hybrid approach, resulting in a mean absolute percentage error (MAPE) that ranges from 16.9% (all human daily entrances) to 24.8% (female hourly entrances). Such forecasting models are potentially valuable for commercial store managers. For instance, they can help in the supporting decisions related with the management of the retail store human resources and marketing campaigns.

Keywords: Business Intelligence, Data Mining, Time Series, Marketing Intelligence, Regression Methods, Forecasting.

Resumo

Devido aos avanços nas tecnologias de informação, tem surgido um maior interesse no uso de técnicas de *data mining* para extrair padrões uteis de dados em bruto com o objetivo de suportar a tomada de decisão. Neste trabalho, foi seguida uma abordagem de *data mining*,com o objetivo de prever o numero de acessos de pessoas num espaço commercial, acesso este medido através de um sistema de reconhecimento facial automatic a partir de vídeo. Em particular, foi executado um elevado número de experiências com vista a prever distintos tipos de acessos humanos (e.g., homens, mulheres, ambos), periodos de previsão (horário e diário) e horizontes temporais da previsão. Mais ainda, diversos métodos de previsão foram testados: métodos de séries temporais convencionais e modelos de séries temporais com base em modelos de *machine learning*; uma abordagem de regressão (e.g., utilizando variáveis meteorológicas e relacionadas com eventos especiais); e uma abordagem híbrida que usa variáveis de séries de temporais (*time lags* de entradas de pessoas) e de regressão. Para se obter uma avaliação mais robusta, foi utilizado um esquema de janelas deslizantes (*rolling window*) que implica um elevado número atualizações dos modelos (treinos) e testes. Para uma previsão de curto prazo (horizonte de 1), os melhores desempenhos foram obtidos, de um modo geral, pela abordagem híbrida, tendo-se obtido um erro percentual absolute médio (MAPE) que varia entre 16.9% (entradas diárias de pessoas) e 24.8% (entradas femininas horárias). Tais modelos de previsão são potencialmente valiosos para gerentes de lojas comerciais. Por exemplo, as previsões podem suportar decisões sobre a gestão dos recursos humanos de loja, bem como de campanhas de *marketing*.

Palavras-chave: *Business Intelligence*, *Data Mining*, *Time Series*, *Marketing Intelligence*, Métodos de Regressão, Simulação, Previsão.

List of Acronyms

ACF - Auto correlation function

ANN – Artificial neural network

ARIMA - Autoregressive Integrated Moving Average

CRISP-DM - Cross Industry Standard Process for Data Mining

CSV - Comma-separated values file

HW - Holt-Winters

KSVM – Kernlab package Support Vector Machine

MAE - Mean Absolute Error

MR - Multiple Regression

MLPE - Multilayer perceptron ensemble, provided by rminer package

MAPE – Mean Absolute Percentage Error

RandomForest - Random Forest algorithm provided by the randomForest package

RMSE - root mean square error

Rpart – Decision tree provided by the rpart package

PNG - Portable Network Graphics

Contents

List of figures

List of tables

1. Chapter 1 - Introduction

1.1. Context

The purpose of this work is to predict human entrances, on a daily basis or during certain relevant periods of the day, of clients in a commercial store. Such human entrances were captured using a special facial recognition system based on video analysis and that was implemented in a particular commercial store, during a pilot project. This data was kindly provided a private company, as described in the Acknowledgments section of this work. Such dataset was then enriched with other features, such as weather variables and local events that took place near the retail area. Using a data mining approach, under the CRISP-DM methodology, several time series and regression methods were explored, with the goal of obtained the best possible predictions.

1.2. Objectives

The main objective is to use a data mining approach to predict human entrances at a commercial retail store. Such human entrances are measured in terms of male, female and total entrances, under two time periods (daily and hourly) and for several lookahead predictions. As a secondary objective, several forecasting methods are compared, including conventional time series methods, time series machine learning models, a regression approach that uses weather and special event data, and a hybrid approach that uses both time series and regression variables.

1.3. Methods and tools

The CRISP-DM methodology and the R framework were adopted in this work:

- CRISP-DM: a popular methodology used to help the success of data mining projects; it is composed by six phases, namely, Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment (see Section 2.1 for more details).
- R tool: an open-source statistical programming language (see Section 2.1); all data mining procedures, including the modeling of the forecasting methods, were executed using this tool.

1.4. Organization

This document is organized in four main chapters. The first chapter provides a brief introduction of the purpose and context of this work. The second chapter presents the state of the art, including definitions used in this document and work related with the forecasting of human accesses. The third chapter describes the experiments conducted and analyses the obtained results, following the CRISP-DM methodology. Finally, the forth chapter draws conclusions in which summarizes the study and presents the main conclusions and recommendations for future work.

2. Chapter 2 - State of the art

This chapter aims to present the relevant background regarding this work. This chapter is divided into 4 sections. The first presents definitions related with this work. The second section is devoted to state of the art studies related with human entrances prediction. The third section compares the articles described in the second section. Finally, the forth section describes the associated concept matrix.

2.1. Definitions

This section describes several concepts that are relevant to this work, namely:

- **Data Mining:** a subfield of computer science, in which involves the intersection of several disciplines, such as artificial intelligence, machine learning, statistics and database systems. Data Mining has been used for extracting useful knowledge from raw data, typically by addressing huge amounts of data, in a process that is automated or semi-automated. In this field, the main outcome is a model or a technique that helps to explain patterns in the data and predict future values (Hall, 2011). Depending on the data mining goal, there are several types of tasks that can be assumed, such as stated by (Fayaad et al, 1996) :
    - **Anomaly detection**: identify errors or unusual data records.
    - **Association Rule**: this task tries to detect patterns in transitional data. Association Rules are based in the concept of string rules made by (Rakesh Agrawal, 1993), which has been used in discovering relationships between different variables. By using different measures it is possible to discover patterns, associations and correlations in all variables that compose the data. This also allows to visualize the attributes and conditions which occur frequently in a set of data.
    - **Summarization:** searches for models that summarize the data.
    - **Clustering:** find groups or items with similar characteristics.
    - **Process Mining:** data mining in special types of "logs/business processes" data.
    - **Prediction:** Predict one value (target or output variable, also known as dependent variable) using values from other variables (inputs or independent variables). A prediction task can further be devided into:
        - **Classification** - predict a discrete variable.
        - **Regression** - predict a numerical variable. There are several regression methods (Hastie et al., 2009), such as:

- **Linear Regression -** a statistical method which allows to predict a numerical variable while the other predictable variables are numerical as well. The purpose of this method consist in calculate a linear relation between the response variable and the predictable variables.
- **Regression Trees - t**his algorithm bases on the creation of a tree-data structure, which by following the leaves, which are made by the input variables, we are able to determine which factors or values should be followed to achieve the final output, which is the predicted value.
- **Regression Rules -** a forecasting model based on the if-then rules. Each rule has a condition, which will be tested with the input variables and a outcome which is the result given by the condition tested. This method allows the input variables to follow the path made by the rules given in this method and which action should it take. This regression model starts by extracting the rules of the training set and then translates it to rules.
- **Support Vector Machine(SVM) –** a statistical learning method that uses only a portion of the training set elements (the support vectors) and a kernel transformation function in order to optimize a fitting classification or regression function.

- ***R-Framework*:** corresponds to a programming language and also to a statistical and computing environment. The R tool was created by Bell Laboratories and includes a large variety of statistical methods (R Core Team, 2014). The tool can also be expanded through the installation of packages, such as the rminer and forecast packages.
- **Time Series:** corresponds to sequences of values of a particular event, generally occurring under a fixed time period (e.g., daily). Time series databases are popular in many applications, such as shopping mall analysis, economic and sales forecasting, budgetary analysis, utility studies, inventory studies, yield projections, workload projections, process and quality control, execution of natural phenomena (such as atmosphere, temperature, wind, earthquake), scientific and engineering experiments, medical treatments, education and research areas (Shaik, Rao, & Rahim, 2013). There are several time series methods (Makridakis et al., 2008), such as:

- **ARIMA Method (*AutoRegressive Integrated Moving Average*):** one of the most used methods to model and predict variables inside the time series category. It is considered to be a stochastic method, whose purpose is to calculate the probability of one variable being between two specified limits. The ARIMA Method has three parameters, which are know to be *p*, *q* and *d*. Often, these parameters are automatically estimated by using a computational method.
- **Holt-Winters Exponential Smoothing:** it analyses a time series data based on trend and seasonal components. There are several Holt-Winters variants, depending on the type of components used (trend or seasonality) and aggregation method (additive or multiplicative) for combining the components.

- ***Markov-Chains*** is a technique to estimate through stimulation, the expectation of a statistic in a complex model. Though is made through successive random selections the stationary distribution of which is the target distribution. This may prove useful to evaluate future distributions in complex Bayesian Models. According to the Metropolis–Hastings algorithm, the items are conditional distributions of single components of a vector parameter although in the end various special cases and applications should be considered (Gilks, 2005).
- ***CRISP-DM*:** the Cross Industry Standard Process for Data Mining (*CRISP-DM*) is a popular methodology used for data mining processes (Chapman et al, 2000). It is composed of six main phases:
  - **Business Understanding**: this phase focus on understanding the project objectives and requirements.
  - **Data Understanding**: this phase starts by collecting data and understanding it. This phase also aims to identify quality problems in the data collected.
  - **Data Preparation**: this phase envolves several data processing steps, such as handing of missing data, outlier detection and removal, data and variable selection..
  - **Modeling**: phase in which several machine learning methods are explored to fit the data-driven models.
  - **Evaluation**: analysis of the best data-driven models obtained in the previous phase. In particular, aspects such as novelty, relevance and/or usefulness to the business domain are considered in this phase.

- **Deployment**: If previous phase is successful, then models are deployed into a real environment. The goal of this phase is to implement and maintain the data mining models, such that a business value can be obtained.

In each phase results in a output, as described in Table 1, in which the bold represents the generic tasks and the italic represents the output of that generic tasks.

Table 1 CRISP-DM phase results output adapted from (Chapman et al., 2000)

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives 1. *Background* 2. *Business Objectives* 3. *Business Success Criteria*	**Collect Initial Data** 1. *Initial Data Collection Report*	**Select Data** 1. *Rationale for inclusion/exclusion*	**Select Modeling Techniques** 1. *Modeling Technique* 2. *Modeling Assumptions*	**Evaluate Results** 1. *Assessment of data mining Results w.r.t* 2. *Business Success Criteria* 3. *Approved Models*	**Plan Deployment** 1. *Deployment plan* **Plan Monitoring and Maintenance** 1. *Monitoring and Maintenance Plan*
Assess Situation 1. *Inventory of Resources Requirements, Assumptions, and Constraints* 2. *Risks and Contingencies* 3. *Terminology* 4. *Costs and Benefits*	**Describe Data** 2. *Data Description Report* **Explore Data** 1. *Data exploration report*	**Clean Data** 1. *Data Cleaning Report* **Construct Data** 1. *Derivate attributes* 2. *Generated reports* **Integrate Data** 1. *Merged Data*	**Generate Test Design** 1. *Test Design* **Build Model** 1. *Parameter Settings Models* 2. *Model Descriptions*	**Review Process** 1. *Review of the Process* **Determine Next Steps** 1. *List of possible actions decisions*	**Produce the final Report** 1. *Final Report* 2. *Final Presentation*
Determine Data Mining Goals 1. *Data Mining Goals* 2. *Data Mining Success Criteria*	**Verify Data Quality** 1. *Data quality Report*	**Format Data** 1. *Reformatted Data* 2. *Dataset* 3. *Dataset Description*	**Assess Model** 1. *Model Assessment* 2. *Revised Parameter Settings*		**Review Project** 1. *Experience* 2. *Documentation*
Produce Project Plan 1. *Project Plan* 2. *Initial Assessment of* 3. *Tools and Techniques*					

- The CRISP-DM is also known to have an hierarchical breakdown, described by four levels of abstraction:
    - Phase
    - Generic Task
    - Specialized Task
    - Process Instance

At the top level the data mining is described by a number of phases in which consists of several second level generic tasks. This second level is called generic because it tends to cover, generically all possible data mining situations. The third level, specialized task level, describes how actions should be carried out in some specific situations, such as cleaning numerical values versus cleaning categorical values or determine if the problem is solved by clustering or by predictive modeling. The last level, the process instance, records the actions, decisions and results of the data mining process. This instance is organized according to the tasks that were defined in the upper levels, although in the end represents the engagement that happened in a particular situation than a general situation. This methodology constitutes the basis of this same project and it will be used through out this master thesis, although it may change during the future.

2.2. State of the art work on forecasting human entrances

This section presents the state of the art in terms of forecasting human entrances. The bibliographic search strategy used to obtain the articles was based on the use of *google scholar* specialized web search engine with the keywords listed in Section 2.4 (the concept matrix).

2.2.1. A survey of time series data prediction on shopping mall sales

(Shaik et al., 2013) used several forecasting methods to predict shopping mall sales of several products. First, they explored a simple basic extrapolation method, which was based on a line drawn in the past. However, the obtained results were very inaccurate and the studied environment was considered highly dynamic, easily changed by external factors. Then, they explored a more advanced approach, building a data-driven predictive model using data mining procedures. In particular, they analysed several input features that could affect prices, namely:

1. what people expect about future discounts;
2. when the discounts are expected to be given, based on a calculated depreciation value;
3. which products are to be purchased, such that the stocks will be made available and all the products will be sold out; and
4. the quantity of risk involved.

The authors also stated that there is a necessity of data mining in a shopping mall because, data is mostly valuable when is mobilized or converted into useful information, to enhance the decision-making efficiency. Since the shopping mall is made by a stream of data, that shares a sequential nature, the preferred method to be used is the time series pattern.

The conclusion reached by the authors was to use a tree based algorithm that treats the markets behavior and interest as an input of data, filtering the desired output and using a time series approach, in which they combined the regression and time series algorithms.

2.2.2. Indoor next location prediction Wi-Fi

(Ang et al., 2014) used a technique to predict the movement pattern, also known as, ***Indoor Location Intelligence***, which allows the retail market to evaluate and make better decisions for their business. They have used the ***Markov-Chain*** method to forecast the next locations of the customers inside the shop, using the previous *n locations* that they have visited.

Considering the fact that more and more people are accessing information through mobile devices, many retailers in the United Kingdom have installed Free Wi-Fi devices in order to attract and maintain more customers inside their stores. One of these examples was the supermarket chain *Tesco*, which provided hundreds of its stores with Wi-Fi devices. As this trend started to emerge, an Indoor Location Intelligence program was launched in the United States of America using several technologies, such as Bluetooth, radio-frequency identification (RFID), **video cameras**, among many other devices, in order to track down their movement patterns. One disadvantage of this tracking method is that the Wi-Fi signals must be near of the shoppers' smartphone in order to collect the MAC address, thus the Wi-Fi emitters need to be placed in key points through out the facility, in order to keep collecting the relevant data.

The study also aimed to support the retailers in order to make them able to re-arrange their promotion shelves and carts, such that they could strategize their store layout to attract new shoppers, re-negotiate a better rent, or even stream rich advertisement data. This experiment was made at a major shopping center and the goal was to obtain a new layer of insight for analysis activities, such as shopping behavior. Their data was gathered from February 2013 to June 2013 by capturing the Wi-Fi signals emitted from the customers smartphones or tablets. It should be noted thart a limitation of the study was that no data was collected regarding shoppers that did not have an active Wi-Fi smartphone or a tablet.

The data-driven model accuracy was measured in terms of forecasting the next possible location that a individual may proceed into.

The authors started the process by loading datasets, from the database, and selected the data considered relevant to their study. In a next phase, they started

to cluster the data into one of 30 zones, corresponding to their Cartesian coordinates, then they made a transition traces chain that aimmed to identify the MAC address and its respective mobility patterns. The authors considered that a repeated movement from one location to another one was acceptable, however identical movement patterns are collapsed. Summing up, they hypothesize that "*Location I "* should have different zone numbers from "*Location 2"* whereas "*Location 3"* should be equal to "*Location 1"* . In the end, the prediction zone number in compared with the actual zone number in the testing data. This process is repeated for all valid testing data points. The obtained results ranged from 30% to 45% of accuracy.

The authors stated that to improve their work some factors needed to be taken into account, such as, reducing a single cell size could have effect in the accuracy, as the location tolerance is reduced and also they recommend an intelligent clustering algorithm to improve the models accuracy, and also made a proposal to improve the accuracy of the next location prediction that is to limit the transition traces only to neighboring zones or to zones that are not more than a couple of meter away from the current zone center.

2.2.3. Forecast movements in commercial spaces from customers positioning data

(Rodrigues, 2014) executed a forecasting approach that is similar to this work. The main difference is that Rodrigues and his collaborators used ***BIPS*** technology to measure and track customers. The *BIPS* technology reads radio frequency signals and their distances, as emitted from in mobile phones or tablets. Since temperature, electromagnetic disturbances and vast number of people could affect these type of signals, there is an algorithm that rectifies the signals in order to augment the *BIPS* overall precision. The *BIPS* solution uses several communication protocols, such as *GSM,* Wi-Fi and even Bluetooth, in order to track the devices. Also, customers do not need to install any specialized software and the BIPS authors assure that this technology does not compromise the users' privacy.

In the Master Thesis dissertation of (Rodrigues, 2014), two tasks were addressed. The first one, more directly related with this work, consisted in predicting the daily number of accesses in a retail center, while the second task involved the analysis of a commercial store layout. In his Master Thesis, Rodrigues adopted a data mining approach, under the CRISP-DM methodology. Moreover, all data analysis was conducted using the R tool. In particular, and related with the first task, a time series approach was used, using total number of human store accesses and supervised machine learning methods. The time series data was enriched with meteorological variables, such as average temperature and the type of weather (clean, clouds, drizzle, mist, partly cloudy, rain and scattered cloud). The distinct forecasting methods included:

1. **Linear Regression -** built in the R framework by using the *lm* function.
2. **Regression Trees -**built by using the the *rpart package* (*Recursive Partitioning and Regression Trees)*.. In this model, the day of the week and previous daily accesses were considered the most relevant input variables. In contrast, the weather and average temperature had a little importance.

3. **Regression Rules -** built in *R* using the package *Cubist*.. The obtained model consisted of only two rules, which separated the example cases through their day (the most important attribute of the model).
4. **Support Vector Machine -** the regression method *SVM (Support Vector Machine)* was implemented in *R* using the package *e1071*. This model according to the author, was considered to be the most conservative, since it produced few changes between the passing days.
5. **Time-Series** – a pure time series approach was tested by considering two operational research: ARIMA and Holt-Winters. The R forecast package was adopted to implement such methods.

Using the Mean Absolute Error (MAE) error metric, Rodrigues and his collaborators compared the distinct forecasting approaches. The best forecasts were obtained by the **Regression Trees.**

2.2.4. Forecasting tourism demand using time series, artificial neural networks and multivariate adaptive regression splines: evidence from Taiwan

(Lin, Chen, & Lee, 2011) conducted a research regarding the tourism in Taiwan. Their study aimed to forecast accesses from tourists into Taiwan, by using three methods, such as, ARIMA method, Artificial Neural Networks (ANN) and multivariate adaptive regression splines (MARS). This study data sample was made by the number of tourists that arrived every month between January 2004 to June 2010, covering 78 months. Their aim was to determine which was the most useful forecasting model.

When fitting all models, the authors used the first 62 months of data as the training set. The most recent 16 months (March 2009 up to June 2010) were used to test the forecasting performances and compare the methods. When using ARIMA method, the authors explored several seasonal variants. The best ARIMA model was the ARIMA(2,0,0)x(2,0,2)12. For the ANN, the authors explored the multi-layer perceptron model, with an hidden layer with 22 to 26 nodes, and one node at the output layer. In order to select the best ANN, the root mean square error (RMSE) metric was adopted over the training samples. The best ANN, with RMSE=0.206204, consisted of 25 neurons within the hidden layer and was training with a learning rate of 0.0002, being the training algorithm stopped after 150000 iterations. The last forecasting method was the MARS. The final fitted MARS model obtained a RMSE of 8895.09, a MAD of 76986.57 and a Mean Absolute Percentage Error (MAPE) around 17.72%. When comparing the three forecasting methods, the authors concluded that the best performing method was the ARIMA (2,0,0)x(2,0,2)12, which obtained lowest RMSE and MAPE values.

2.2.5. Summary

Four works related with forecasting applications were presented in the previous section. Table 2 summarizes all these studies, in terms of data approaches used, forecasting methods and results obtained.

In particular, there are two of the presented studies that are more similar to this work: (Lin et al., 2011) and (Rodrigues, 2014). Both these studies aim to forecast accesses of clients. (Lin et al., 2011) focus more attention to the tourism market instead of retail market. Nevertheless, the forecasting methods follow a time series approach. The work of (Rodrigues, 2014) is more closely related with this work: it aims to predict daily accesses at a retail store, it combines time series and regression variables (e.g., average temperature), it explored both operational research and machine learning methods, it follows the CRISP-DM methodology and it used the R tool. However, it should be stressed that there are also some relevant differences, namely:

- we used a video face recognition system that can also capture entrances of people without communication devices (e.g., smart phone);
- we predicted three types of human entrances: female, male and total human entrances;
- we used two time scales: hourly and daily; and
- we used a larger number of regression variables (e.g., sport events).

Table 2 Results and conclusions reached by the studies presented in the state of art

Reference	Data Approach	Prediction method	Results obtained
(Lin et al., 2011)	Tourist Sample	ARIMA, ANN, MARS	• ARIMA RMSE presented a score of 37466.43 • BPN RMSE presented a score of 80202.12 • MARS RMSE presented a score of 88895.09 •
(Shaik et al., 2013)	Shopping mall dataset	Time Series	• not shown
(Ang et al., 2014)	Shopping Wi-Fi Accesses	Markov Chain	• *n=1 model* reached 30% up to 38% accuracy • *n=2 model* reached 48% accuracy
(Rodrigues, 2014)	BIPS dataset	Linear Regression, Regression Trees, Regression Rules, SVM, Time-Series (ARIMA, Exponential Method)	• Linear Regression present with MAE of 2.45 • Regression Rules present with MAE of 2.4 • Regression Trees present with MAE of 2.89

2.3. Concept matrix

Table 3 presents the concept matrix of the state of the art, where the columns denote relevant keywords and the rows show the related work references.

Table 3 Concept Matrix for the related work

	Emporium Prediction environment	Shopping mall Data Mining Prediction	Time Series and Malls	Regression Tree's and Shopping Malls	Prediction Shopping Mall	Data Mining in shopping malls	Shopping centers Predictions	Time Series in Shopping malls	Client Support prediction Shopping Malls
(Lin et al., 2011)	X	X	X	X	X	X	X	X	X
(Rodrigues, 2014)					X				X
(Ang et al., 2014)	X	X			X		X		X
(Shaik et al., 2013)		X	X			X	X	X	X

3. Forecasting human entrances at a commercial store using facial recognition data

3.1. Introduction

In this chapter we describe the data mining process that we took on the development of this master thesis. Following the *CRISP-DM* methodology, each section of this chapter describes the details and decisions made in order to obtain the final results.

3.2. Business understanding phase

Prior to this study, a pilot was installed and conducted in a commercial store, under the responsibility of th EXVA company. In the pilot, a digital camera was installed at the entrance of a retail store and positioned in such a way that it could capture all costumers that entered the store. The camera was integrated with a video system developed by EXVA that had the ability to automatically recognize human faces and allowed to obtain the key data used in this study. The collected data contained accesses during the year of 2013, from April to December. The video system is capable of capturing several faces from a single image. Each detected face signals the creation of an event, meaning that a person has entered the store. Table 4 summarizes the main attributes of this data.

The motivation of this pilot was to assist the management of the store by providing an automatic method to count costumer accesses to the store. Using such data, the goal is to assist in store management decisions, such as controlling stocks and product inventory, managing human resources and also support marketing operations, such as the announcement of discounts or promotions.

Table 4 Description of the data attributes obtained from the pilot

Field	Description	Range
event_date	The time of the event	Within the range from [0;+inf]
face_uptime	Total Time that the face was identified by the camera	Within the range from [0;+inf]
face_score	Score given in recognizing the face	Within the range from [0;100]
gender	Estimated gender of the person	0 – unrecognizable; 1-Male;2-Female
age	Estimated age of the person	Within the range from [0;+inf]
age_deviation	Estimated age interval	Within the range from [0;+inf]
anger	Estimated anger score of the person	Within the range from [0;100]
happiness	Estimate happiness score of the person	Within the range from [0;100]
sadness	Estimated sadness score of the person	Within the range from [0;100]
astonishment	Estimated astonishment score of the person	Within the range from [0;100]

3.3. Data understanding and preparation phases

In this study, we only considered pilot attributes that the EXVA company considered to be highly accurate, namely: event_date, face_uptime and gender. First, we had to convert the Unix timestamp (of the event_data attribute) to a normal date timestamp. For this, we used a package on R called Lubridate. In addition to the pilot data, we retrieved other data variables. Using the R package CURL, we managed to get all weather information from the specialized wunderground service («Wunderground», 2010) from the same period of time that the pilot was conducted. .As another external source of data, we had access

to an event database, related with important human gathering events that took place within the pilot period. Examples of these events included football matches (sport event), music concert very near the store (local event) and other entertainment events that took place in the city where the store is placed (other event). This event database was retrieved from the Web, as kindly provided by the group of students mentioned in the Acknowledgements section. Using the R tool, the three data sources were first merged and then split into 6 distinct .csv files, according to the type of entrance frequency (hourly or daily), gender (female, male, all): femaleWeatherhourly, maleWeatherhourly, AllWeatherhourly, maleWeatherdaily, femaleWeatherdaily and AllWeatherdaily. The tables 5 and 6 describe the data used on the .csv files. It should be noted that the considered store is opened during all alls of the week, during a period of 12 hours each day.

Table 5 Data features used on the daily datasets

Column	Description
time	Date of time (day of the year)
Ts	Time series with number of accesses for that time
temperature	Exterior temperature in that time
Weather	Exterior weather of that time
Humidity	Exterior humidity
Max.Wind.Speed	Exterior maximum wind speed
type	Type of special events: • None • Sports (e.g. football match) • Local (e.g., music concert very near the store) • Other (important human gathering event in the city)
Weekday	Day of the week: • 1 – Sunday • 2 – Monday • 3 – Tuesday • 4 – Wednesday • 5 – Thursday • 6 – Friday • 7 - Saturday
Holiday	Determines if current day is a holiday or not: • 0 – No • 1 – Yes
Weekend	Determines if current day is part of the weekend: • 0 – No • 1 – Yes

Table 6 Data features used on the hourly datasets

Column	Description
time	Date of that time (hour and day of the year)
Ts	Time series with number accesses for that time
temperature	Exterior temperature in that time
Weather	Exterior weather of that time
Humidity	Exterior humidity
type	Type of special events: • None • Sports (e.g. football match) • Local (e.g., music concert very near the store) • Other (important human gathering event in the city)
Weekday	Day of the week: • 1 – Sunday • 2 – Monday • 3 – Tuesday • 4 – Wednesday • 5 – Thursday • 6 – Friday • 7 - Saturday
Holiday	Determines if current day is a holiday or not: • 0 – No • 1 – Yes
Weekend	Determines if current day is part of the weekend: • 0 – No • 1 – Yes

The Figures 1 to 7 represent graphical information in terms of histograms and barplots that provide the overall frequency values of the features mentioned in Tables 5 and 6..

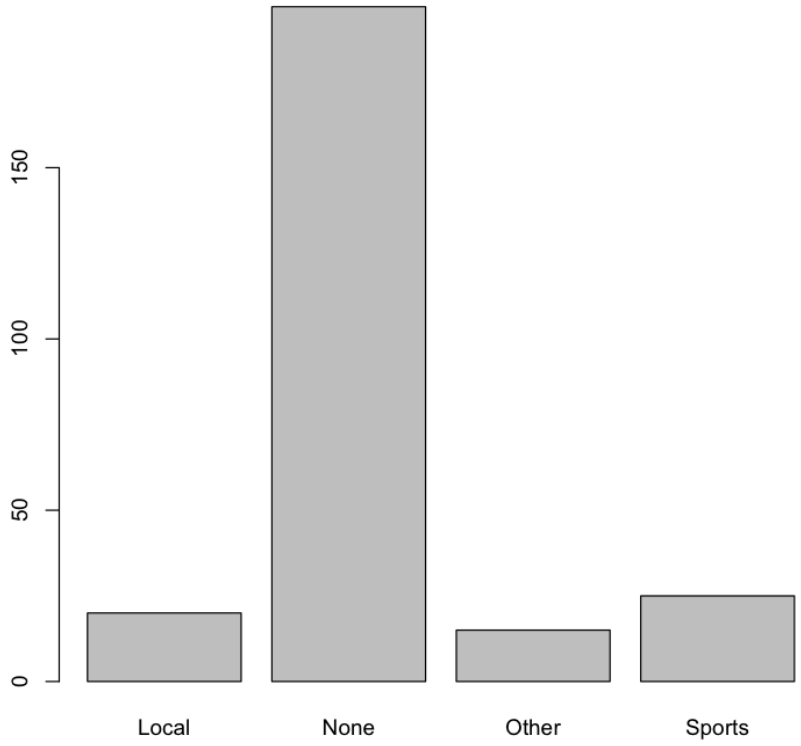

Figure 1 Barplot with frequencies of the special daily events.

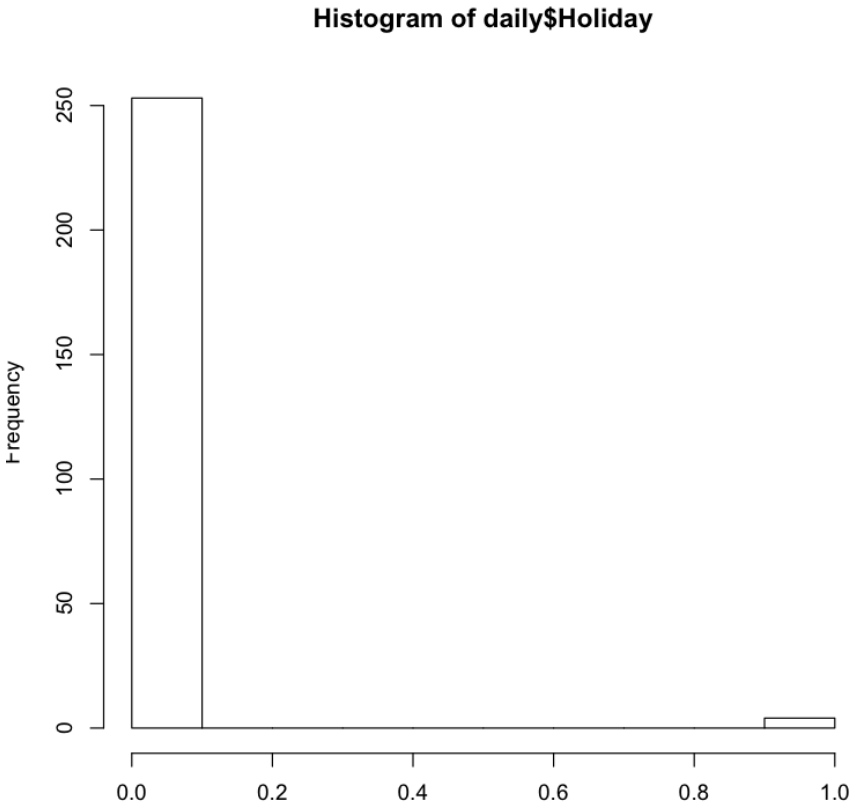

Figure 2 Histogram about the number of holidays

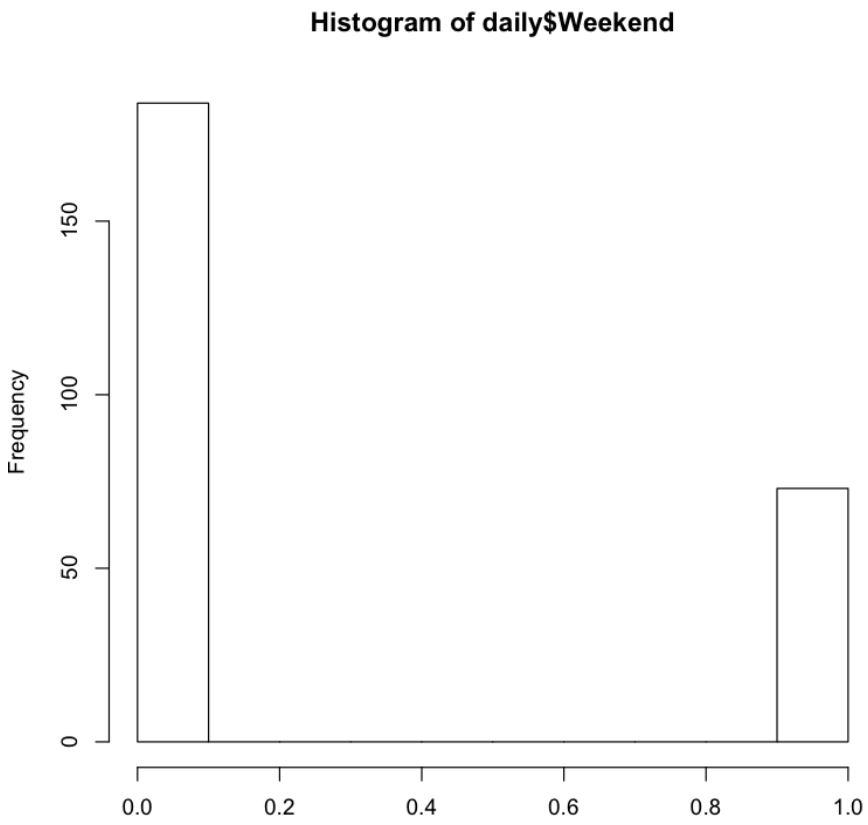

Figure 3 Histogram with the number of weekends days

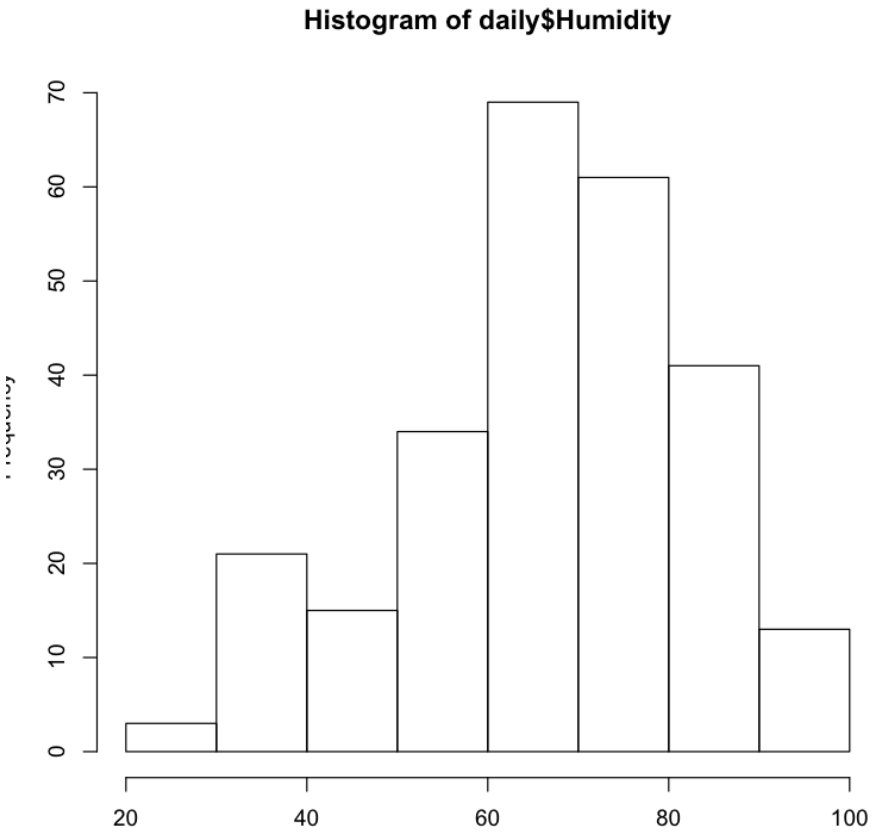

Figure 4 Histogram of the humidity variable

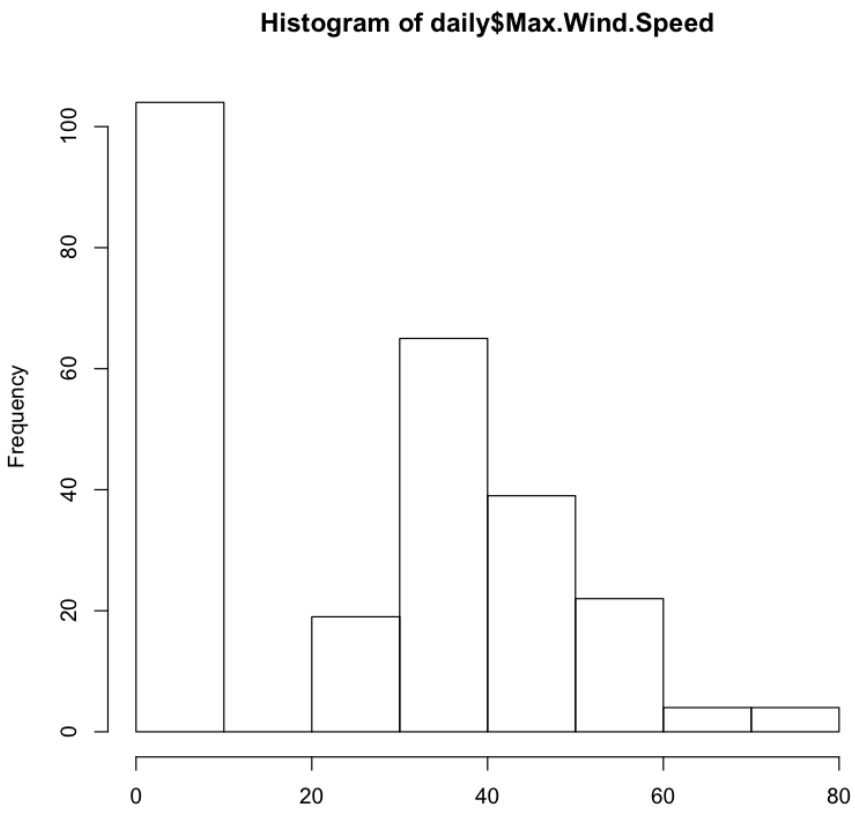

Figure 5 Histogram of maximum wind speed

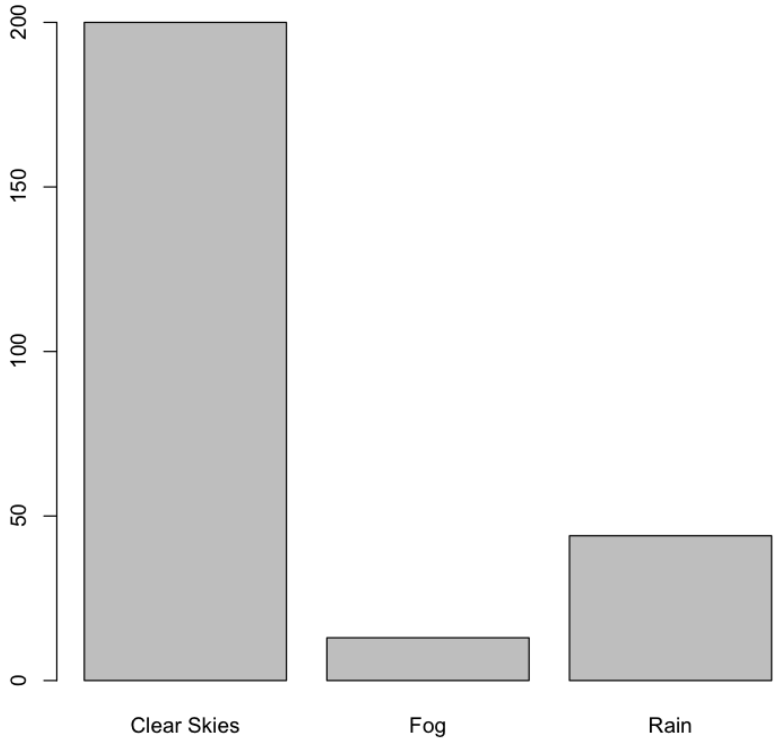

Figure 6 Barplot with the frequency of external weather condition

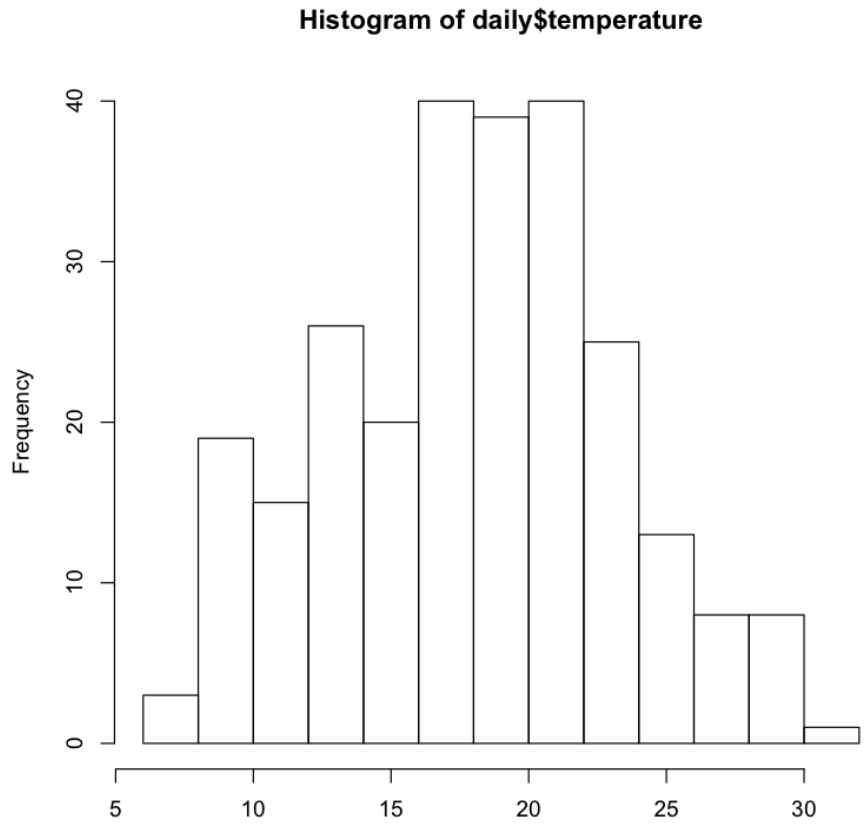

Figure 7 Histogram of temperature

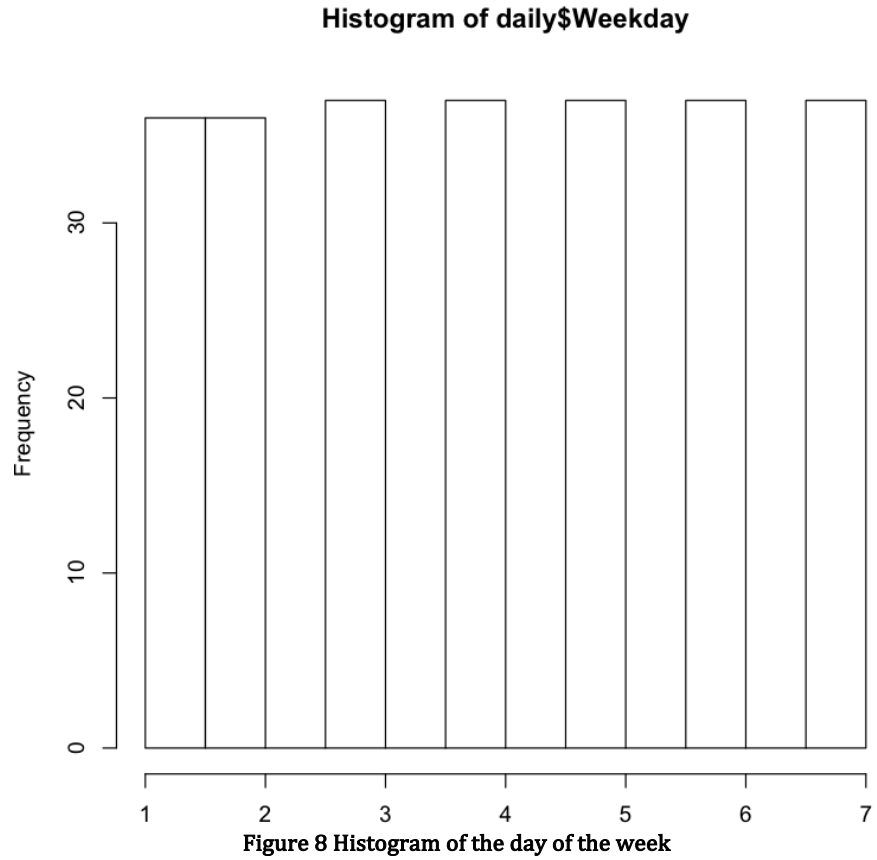

Figure 8 Histogram of the day of the week

The Table 7 summarizes the major quality issues found in collected the data, as well as the decisions made to solve these issues.

Table 7 Problems found during the data preparation phase

Field	Description	Possible Problem	Decision
Face_uptime	Values above 300 seconds	The camera may have been identifying a marketing billboard instead of a human face	Omit the values that were above the 300 threshold.
TS	During a certain times the camera did not record any accesses (missing values).	The camera may have been shut down or stopped working during that time	Replace missing values with a simple imputation method that used the same values available in the previous week.
Event_date	Values in Unix Timestamp	---	Usage of the *lubridate* package to convert to a normal date/date-time timestamp.
Weather attributes	The camera did not record the any weather information.	The camera was not programmed to do so.	Usage of the package *Curl* and the website *Wunderground* to complement the original data (see **Appendix A.5)**

3.4. Modeling phase

During the modeling phase, we decided to address 4 approaches: conventional time series forecasting, in which we recurred to ARIMA and Holt-winters models; time series modeling by machine learning methods (e.g., decision trees, random forest, support vector machines) (ksvm); a pure regression approach (with weather, date and special event data); and a hybrid approach that uses both time series and regression variables.

All four forecasting approaches were implement in the R tool. As the validation procedure, we adopted a rolling window scheme (Cortez, 2015), as shown in Figure 9. This rolling window validation is more realistic and allows also for a more robust evaluation, as several models are trained, resulting in several test sets. In particular, a modeling function was implement using R code, which receives: a scenario that we want to analyse (daily or hourly); an forecasting horizon (*H*) (set up to *7* for daily and 12 for hourly time periods, under the pure series approaches); the rolling windows size (*W*), set to 200 training elements for daily scenario and 2000 for the hourly data; and a variable (*SW*) that increments the rolling windows after each iteration, *(SW* = 1 day for the daily approach; and *SW* = 12 for hourly approach, corresponding to one day as well). The forecast package was used to build the ARIMA and Holt-Winters models, while the rminer package was used to train and test the machine learning methods.. The R code used for this study is presented in the appendix chapter, Sections **A.1), A.2), A.3), A.4), A.5)**.

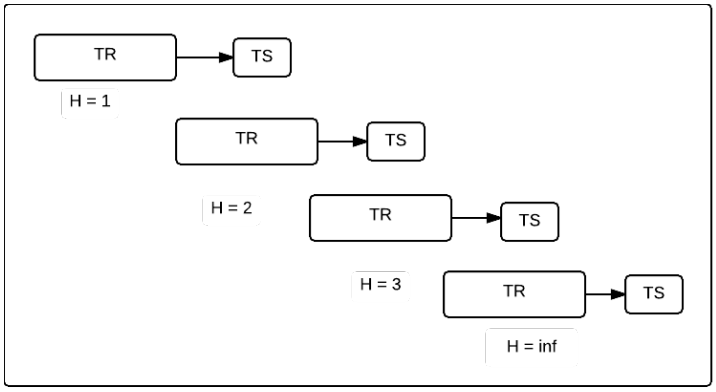

Figure 9 Rolling window process

An example R code that generates the rolling window, is provided below:

#rolling windows

HO=holdout(checkcenariohora(cenario), ratio=H, mode="rolling", iter=i, window=W, increment=SW)

3.4.1. Time series

3.4.1.1. Conventional methods (ARIMA and HoltWinters)

ARIMA

The following models presented were built using the forecast package (Hyndman, 2015). Figures 9 to 10 represent the best models generated by this package and they are followed by the actual values (targets) that happened in that time, which are marked as a black line, and with a blue line (predictions), which represent what the forecast package predicted for that time. For this model we used the auto.ARIMA function (Hyndman, 2015), which builds the ARIMA models for this study using a TimeSeries (TR) and a Horizon that we want to analyze (H) For more information about these models check Chapter 2 section 2.1. The code that originated this models is as follows, for more information check **Appendix A.1)**:

AR=auto.arima(TR)

F1=forecast(AR,h=H)

Pred1=F1$mean[1:H]

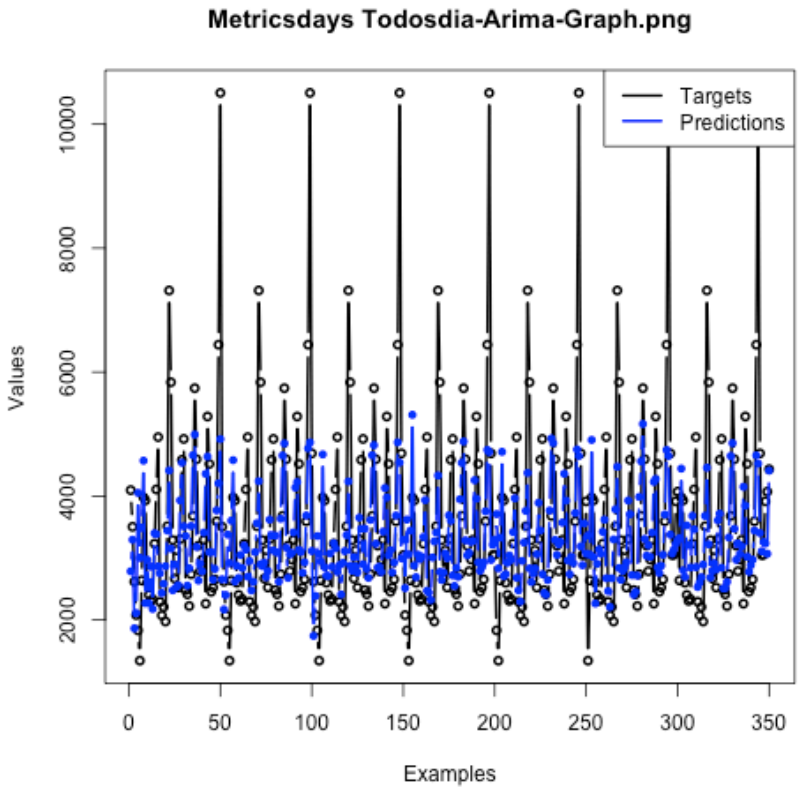

Figure 10 All data forecasts using ARIMA

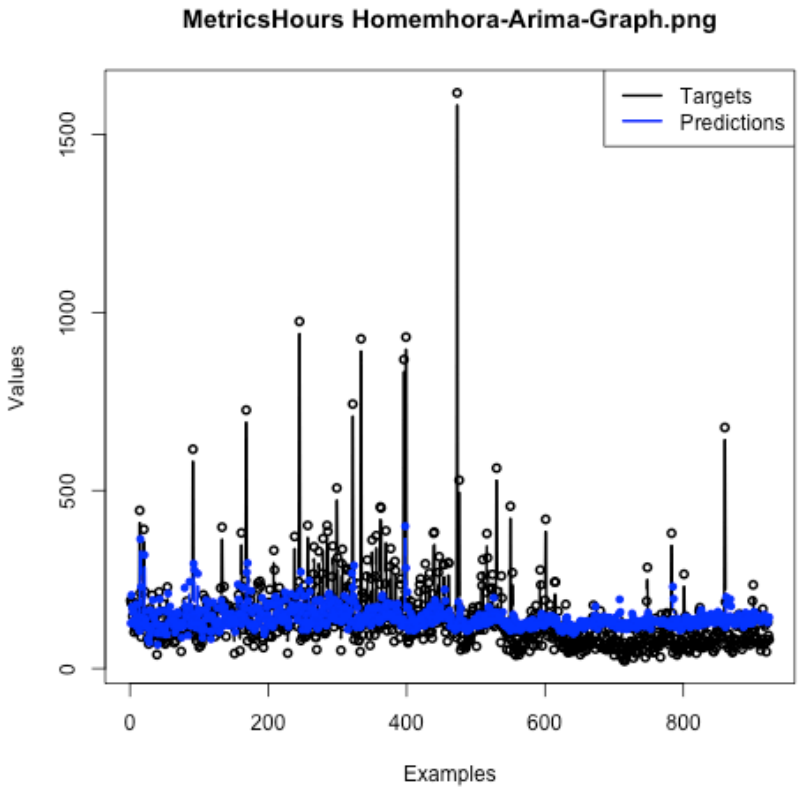

Figure 11 Hourly male entrances forecasts using ARIMA

Holt-Winters

These models were also built using the forecast package (Hyndman, 2015), and it follows the similar patterns to the ARIMA model function (Figures 11 and 12 present the best models generated). The code, used to generate these models, is as follows, for more information check **Appendix A.1)**:

 # holt winters:

HW=HoltWinters(TR)

F=forecast(HW,h=H)

Pred=F$mean[1:H]

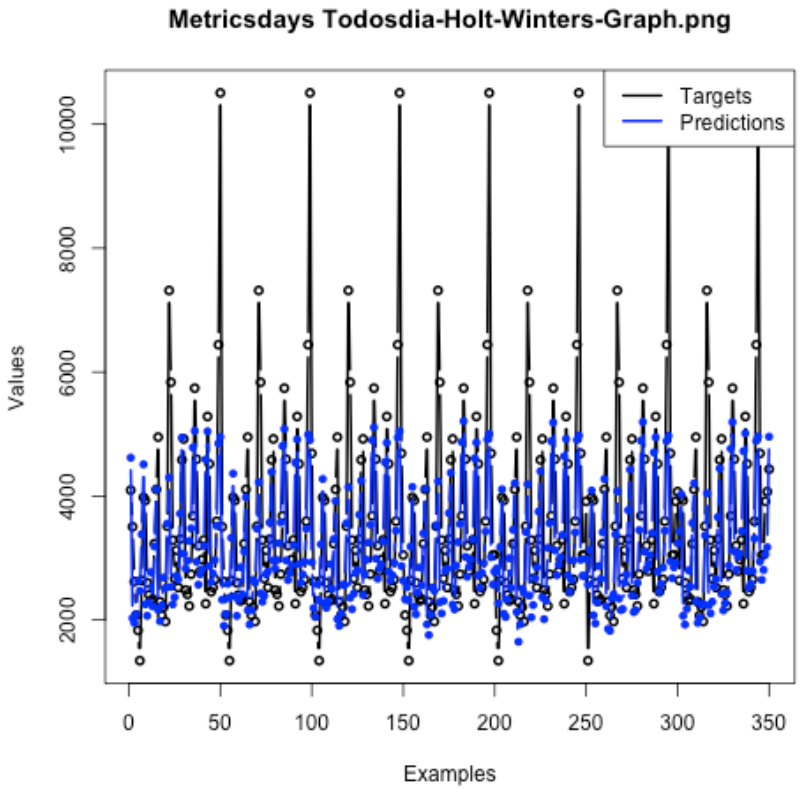

Figure 12 All daily forecasts using Holt-Winters

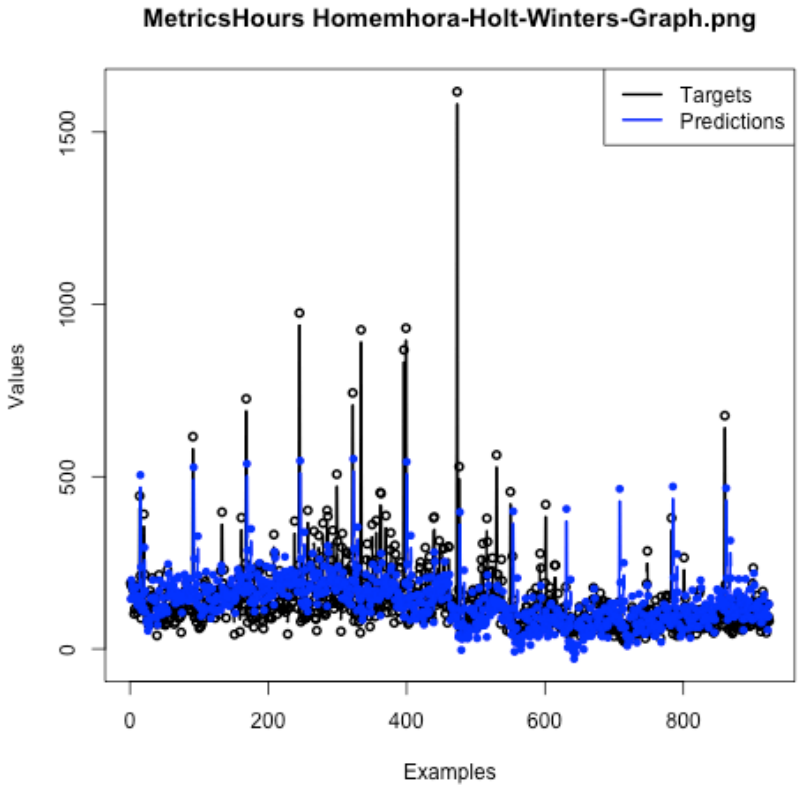

Figure 13 Hourly male entrances forecasts using Holt-Winters

3.4.2. Machine learning

For this approach, we use Machine Learning techniques under the rminer package in **R** (Cortez, 2015). In particular, we explored 5 distinct learning models: random forests, regression tree (rpart), multiple regression (mr), support vector machine (ksvm), neural network ensemble of the multi-layer perceptron type (mlpe). The respective R code appears at **Appendix A.2)**. In particular, we highlight that rminer uses a sliding window to create a supervised training set from a time series (function CasesSeries). In this work, we used the following time lags to create such training data: from 1 to 13, at the hourly scale, and from 1 to 8, at the daily time period.

3.4.2.1. RandomForests

Figures 13 and 14 show the forecasts provided by this model.

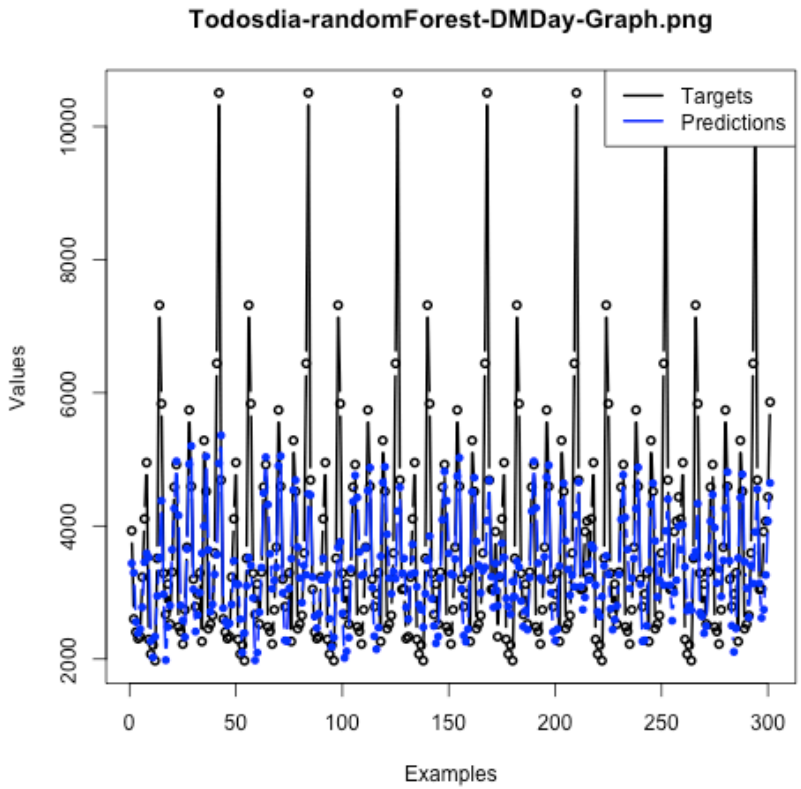

Figure 14 Random Forest modeling using all data(daily)

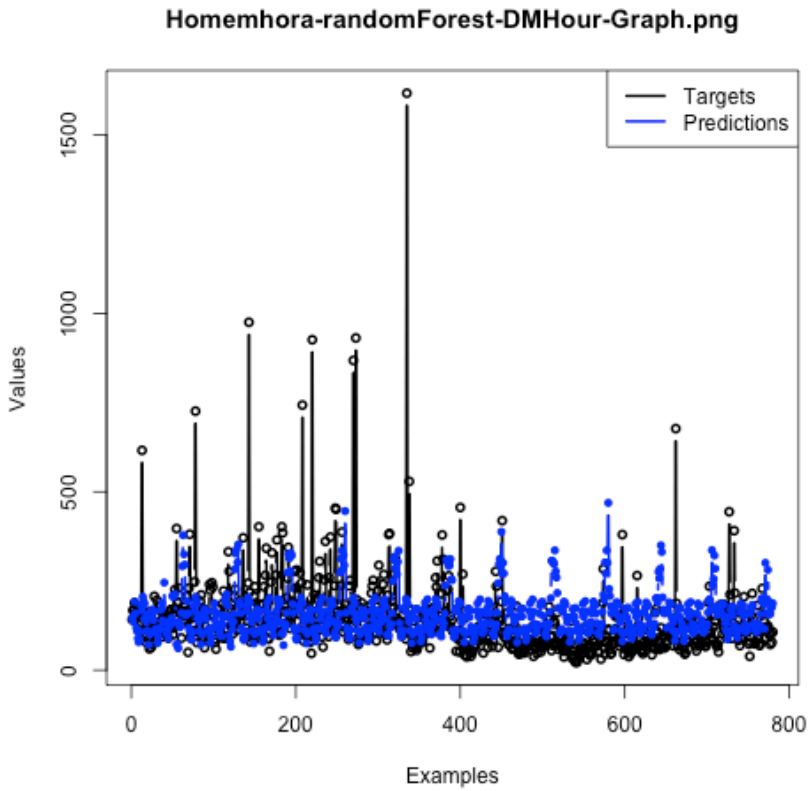

Figure 15 Random Forests forecasts using male data (hourly)

3.4.2.2. Rpart

Figures 15 and 16 show examples of the regression trees forecasts.

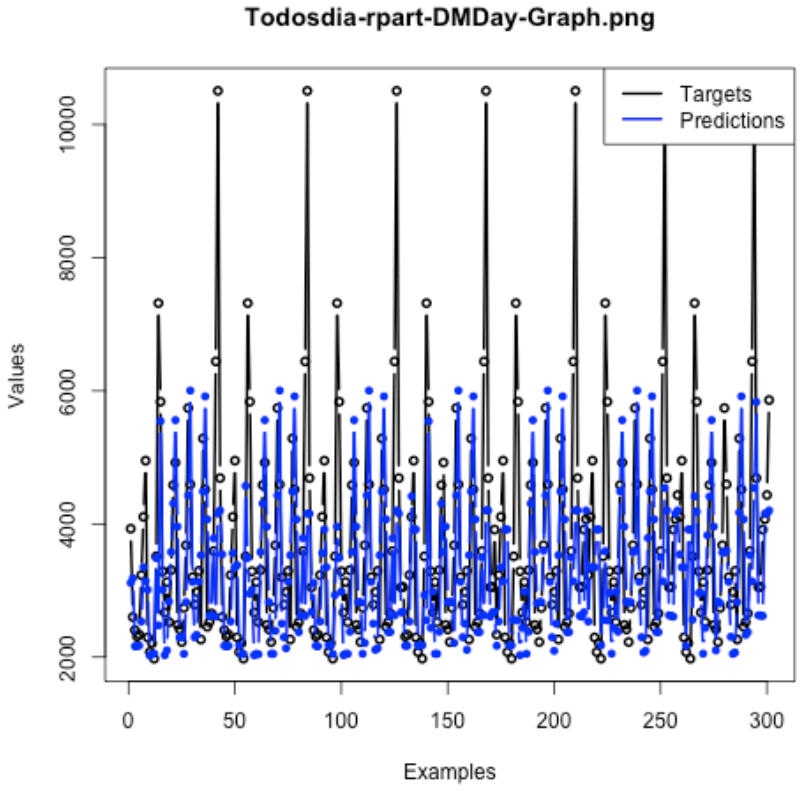

Figure 16 Daily (all data) rpart model forecasts

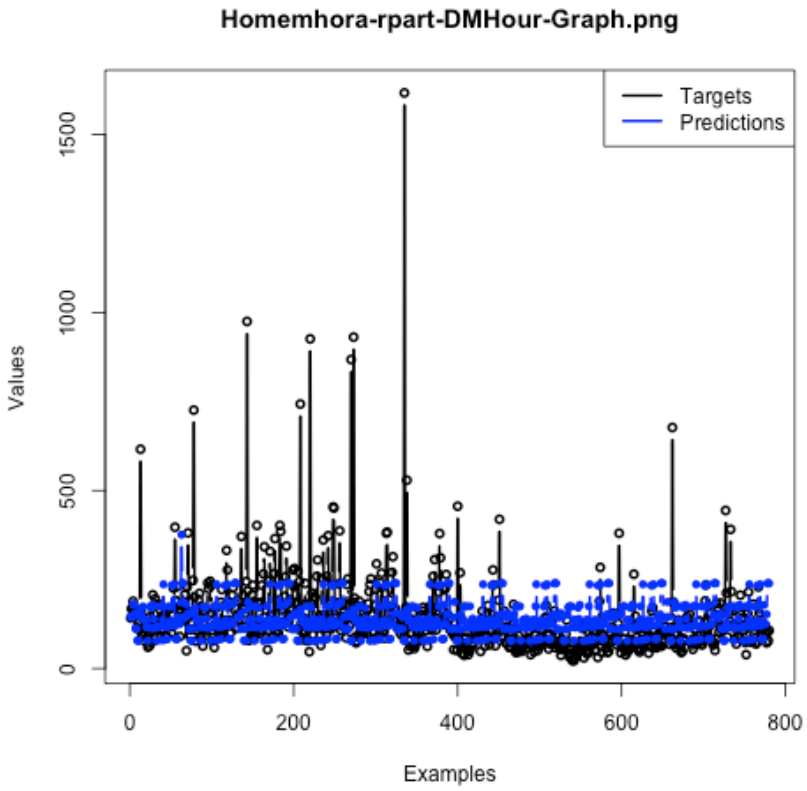

Figure 17 Hourly males rpart model forecasts

3.4.2.3. KSVM

Figures 17 and 18 present examples of predictions provided by the support vector machine model.

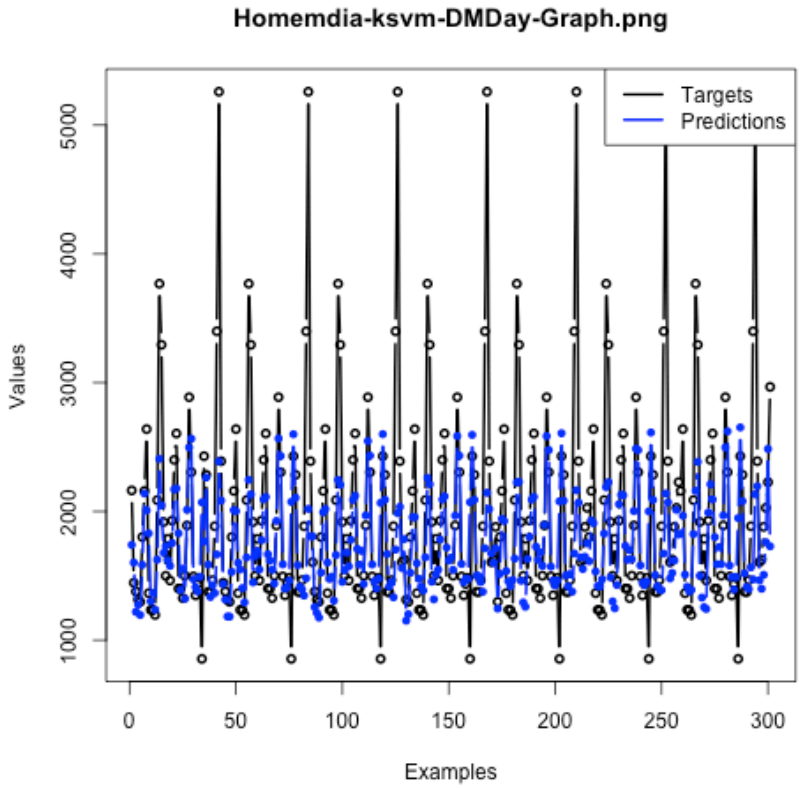

Figure 18 Daily males KSVM model forecasts

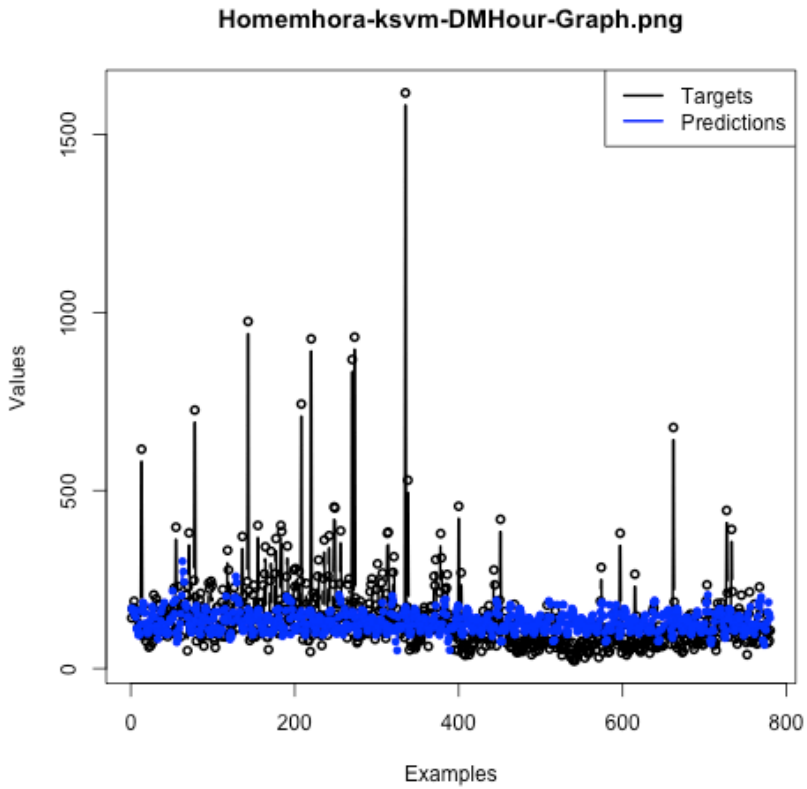

Figure 19 Hourly males KSVM model forecasts

3.4.2.4. MR

Figures 19 and 20 present examples of forecasts generated by the multiple regression model..

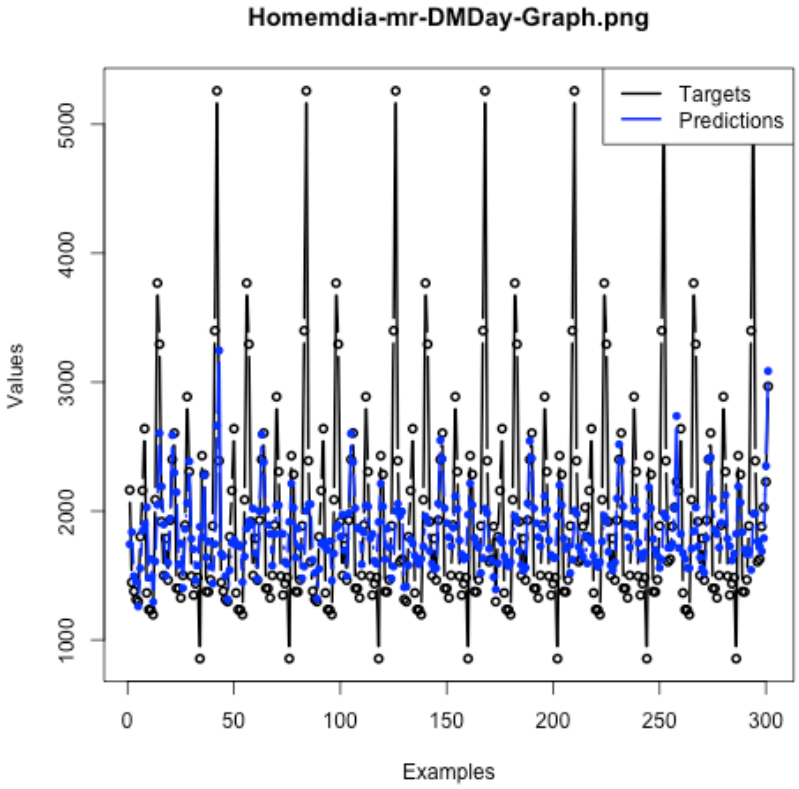

Figure 20 Daily males MR model forecasts

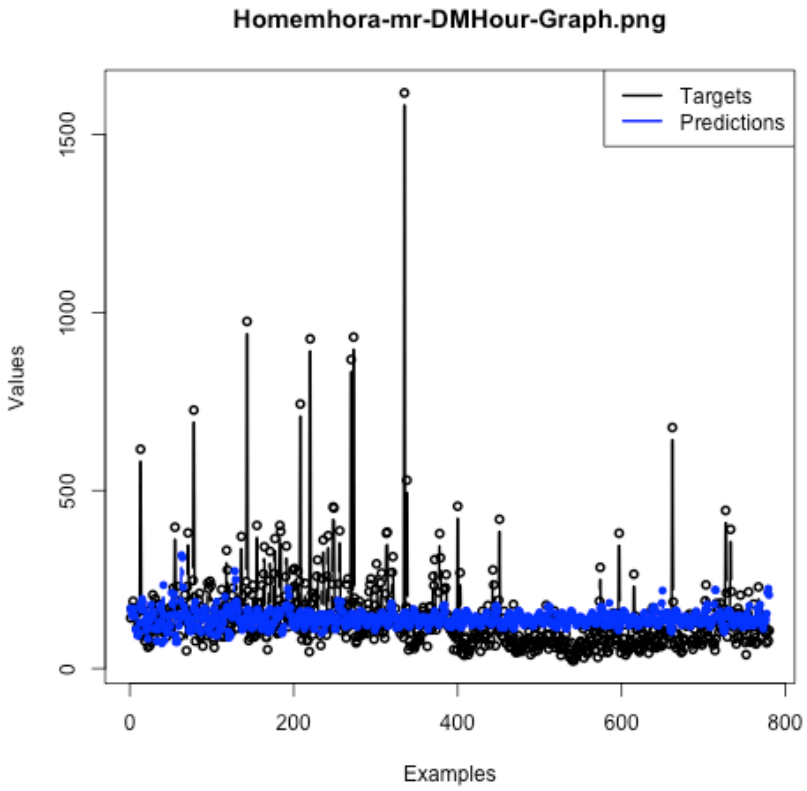

Figure 21 Hourly males MR model forecasts

3.4.2.5. MLPE

Figures 21 and 22 shown examples of forecasts of the neural network model.

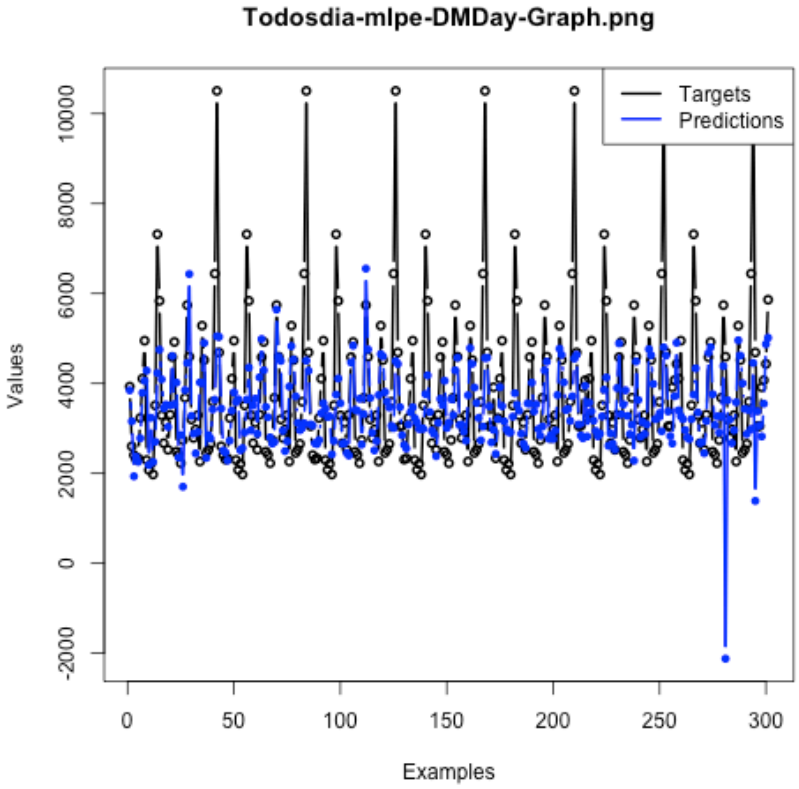

Figure 22 Daily (all data) MLPE model forecasts

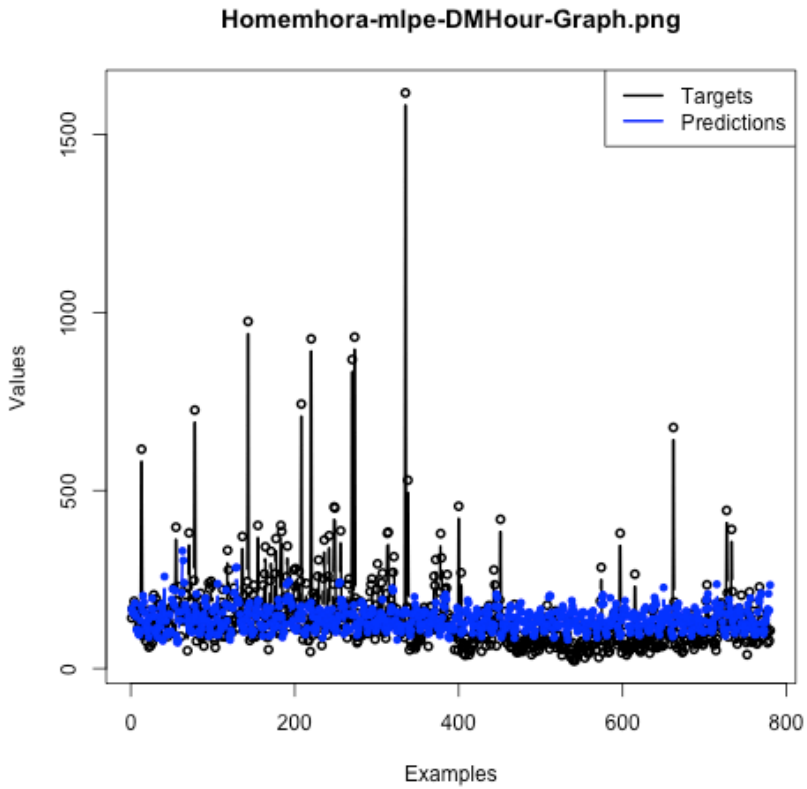

Figure 23 Hourly males MLPE model forecasts

3.4.3. Regression

For the regression approach, although we kept the same rolling window validation technique, we used different data variables, not based on time lags of human accesses. In this particular case, we use weather data, date-related data (day of the week, holiday etc.) and events in order to forecast the number of accesses that may arise (see Tables 5 and 6). The code also suffered a slight change, since the lforecast function of the rminer package, which produces up to H ahead predictions, only works with pure time series datasets. As such, we are only able to predict up to a horizon of 1 (i.e., predict the next value), for the regression based approaches (including the hybrid approach). The implemented R code is presented in **Appendix A.4). Using the pure regression database, we tested in this approach the same 5 distinct machine learning methods used in Section 3.4.2.**

3.4.3.1. RandomForests

Figures 23 and 24 show the random forest pure regression forecasts.

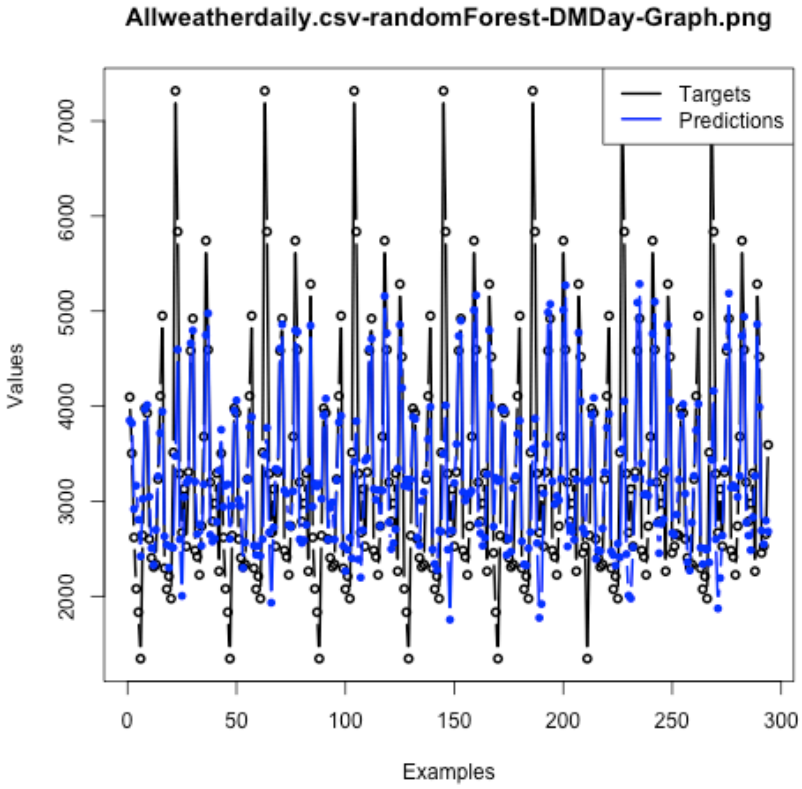

Figure 24 Daily (all data) RandomForest model (Regression) forecasts

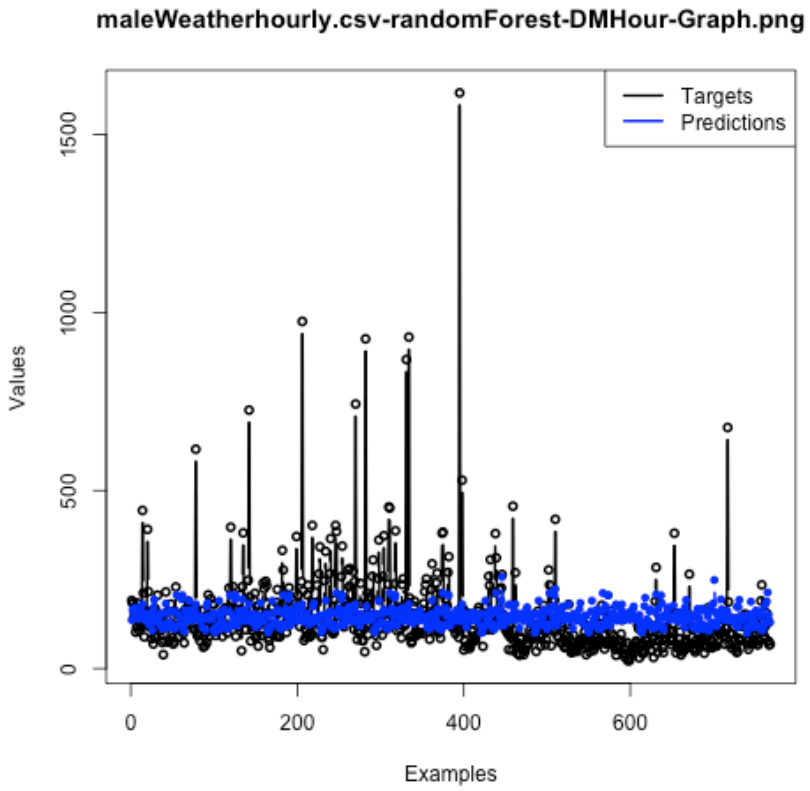

Figure 25 Hourly males RandomForest model (Regression) forecasts

3.4.3.2. Rpart

Figures 25 and 26 show the regression tree pure regression forecasts..

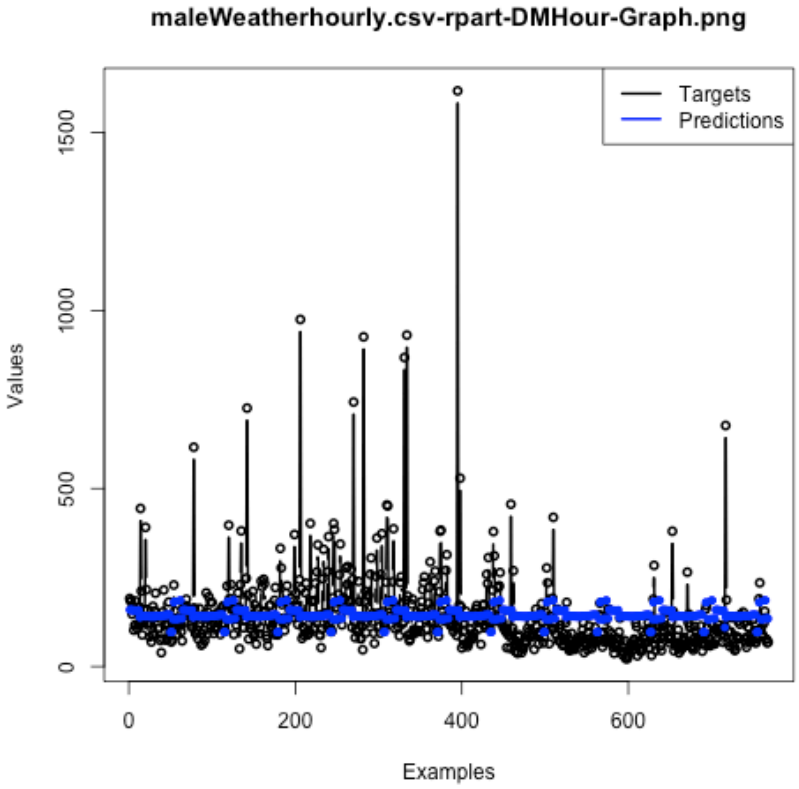

Figure 26 Hourly males rpart model (Regression) forecasts

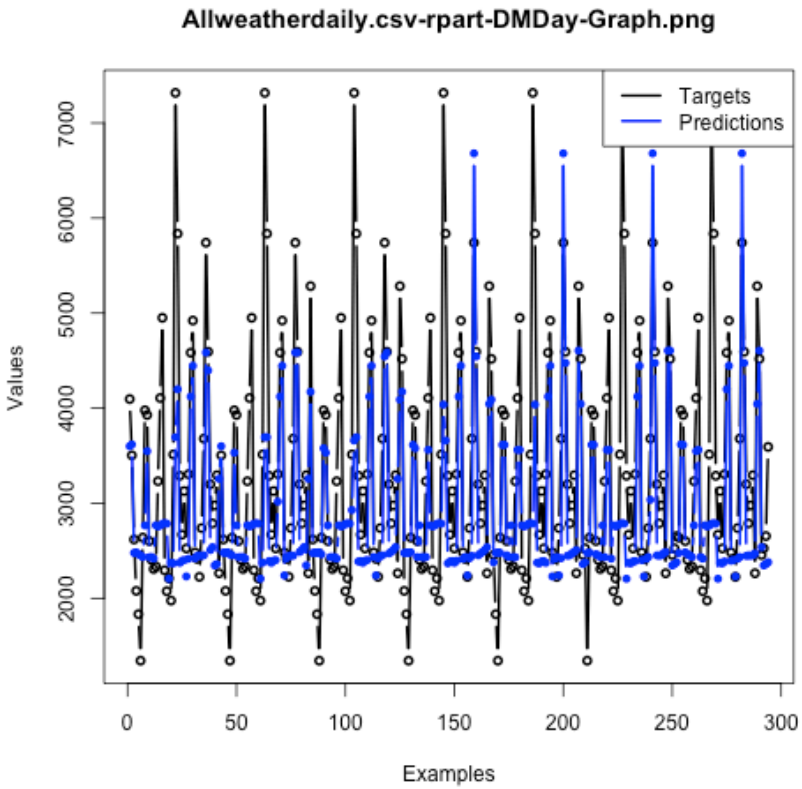

Figure 27 Daily (all data) rpart model (Regression) forecasts

3.4.3.3. KSVM

Figures 27 and 28 show the support vector machine pure regression forecasts.

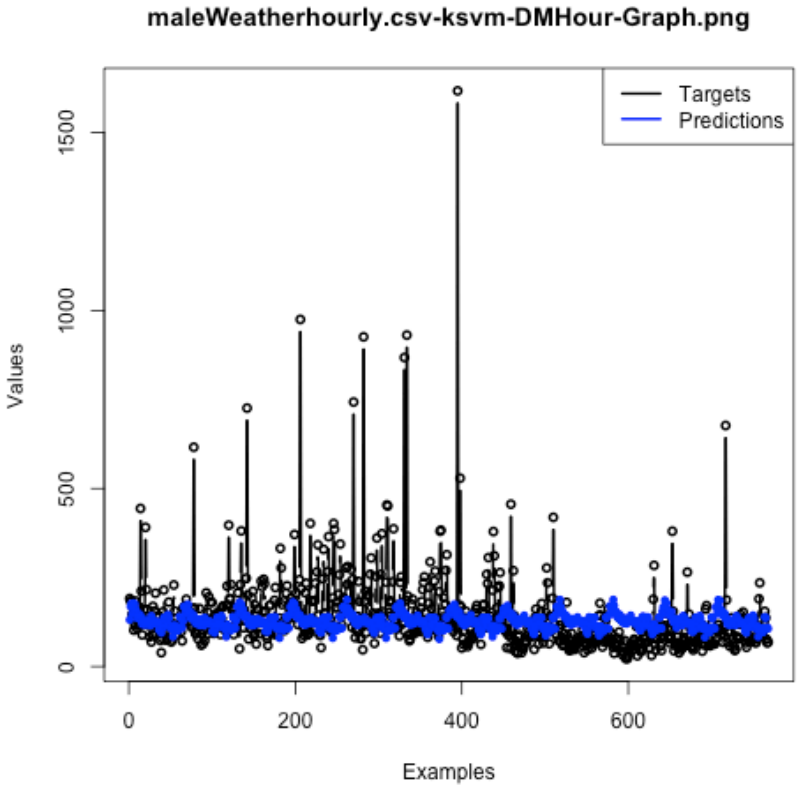

Figure 28 Hourly males KSVM model (Regression) forecasts

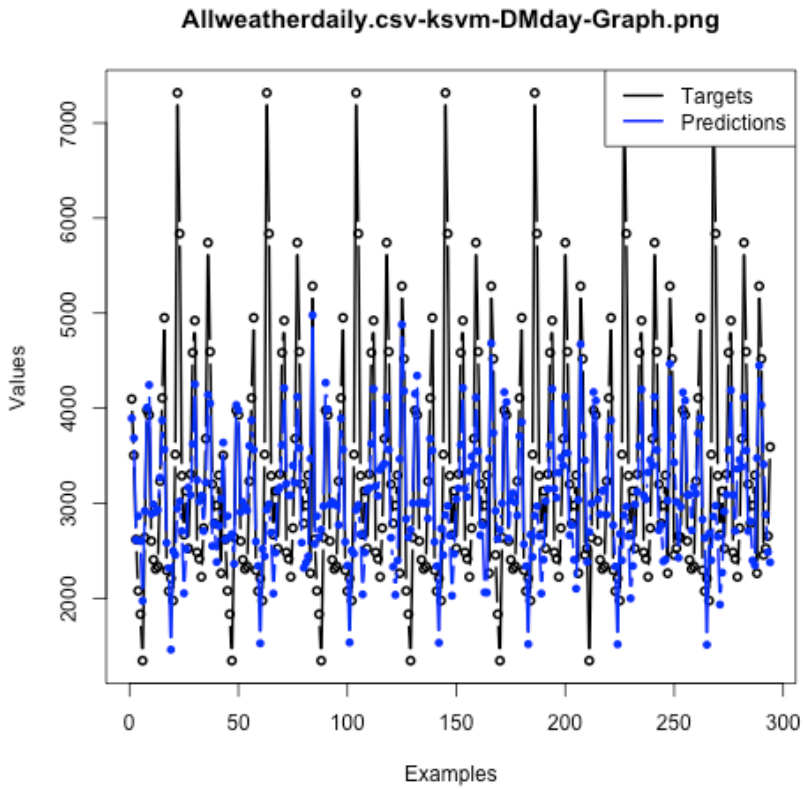

Figure 29 Daily (all data) KSVM model (Regression) forecasts

3.4.3.4. MR

Figures 29 and 30 show examples of the multiple regression pure regression forecasts.

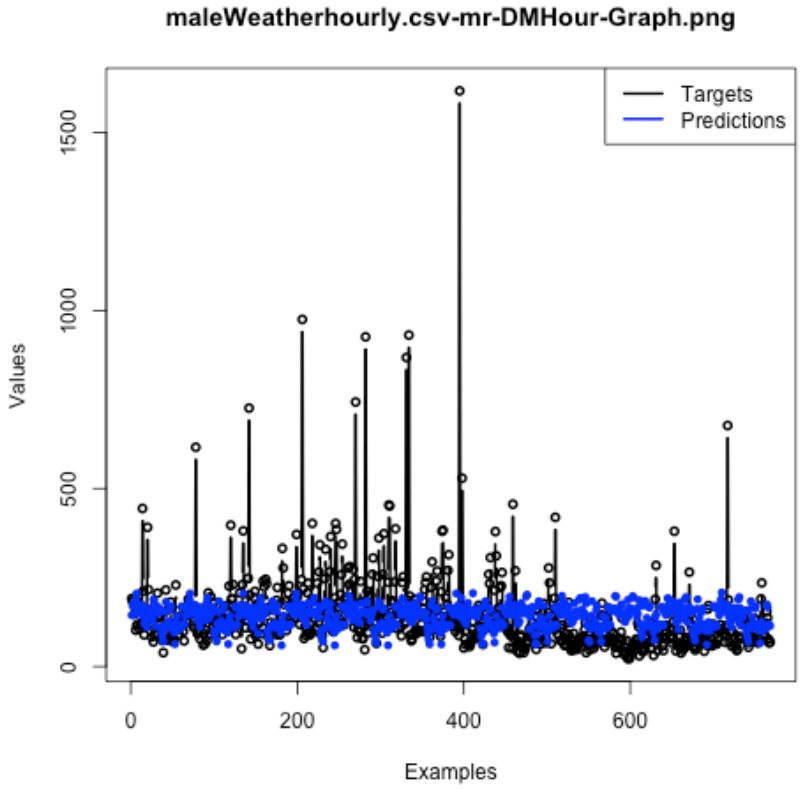

Figure 30 Hourly males MR model (Regression) forecasts

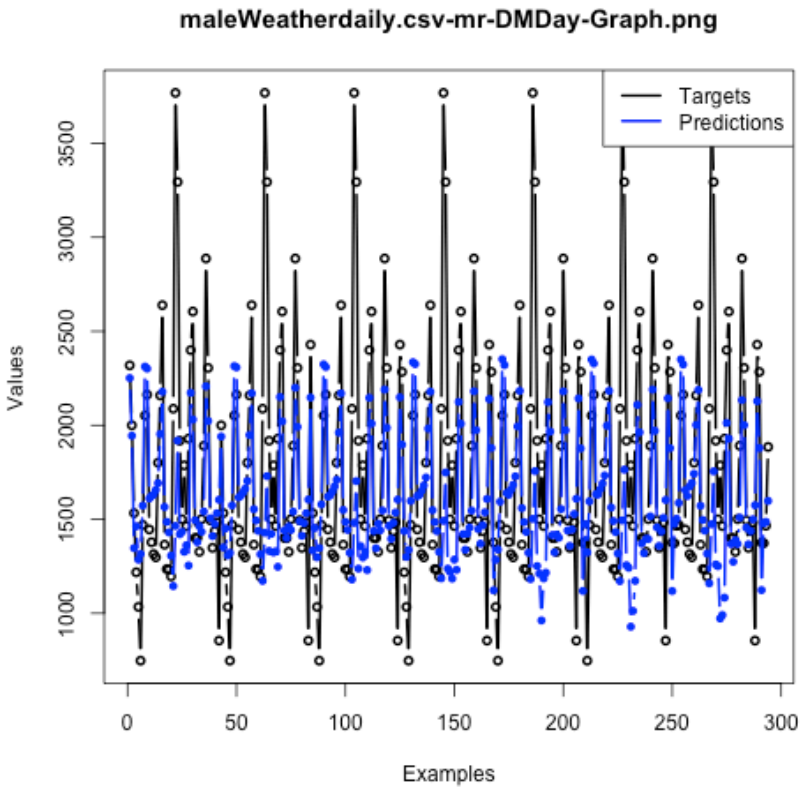

Figure 31 Daily males MR model (Regression) forecasts

3.4.3.5. MLPE

Figure 31 and 32 present examples of the neural network pure regression forecasts.

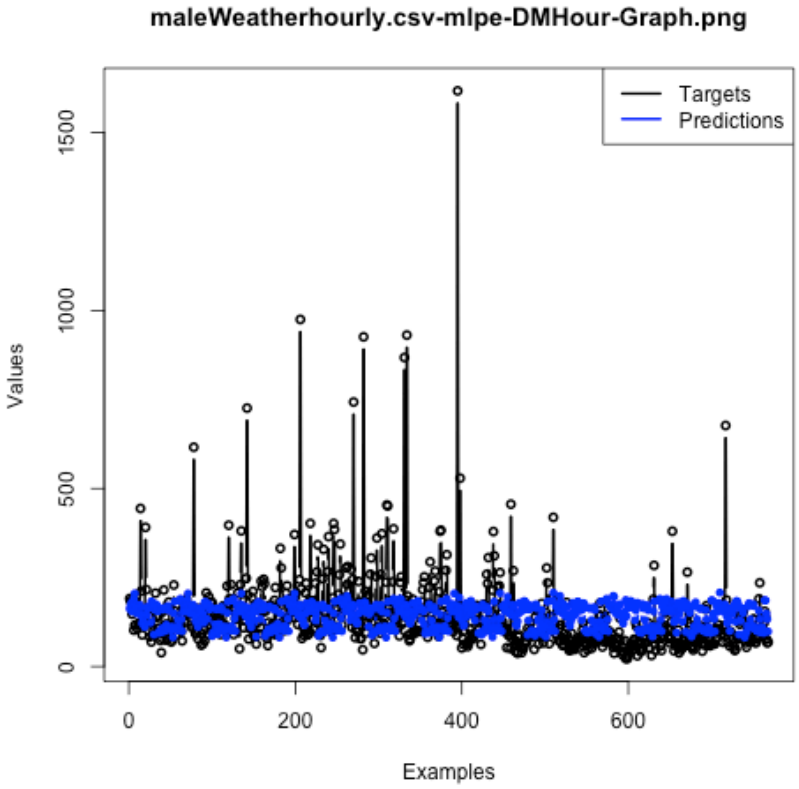

Figure 32 Hourly males MLPE model (Regression) forecasts

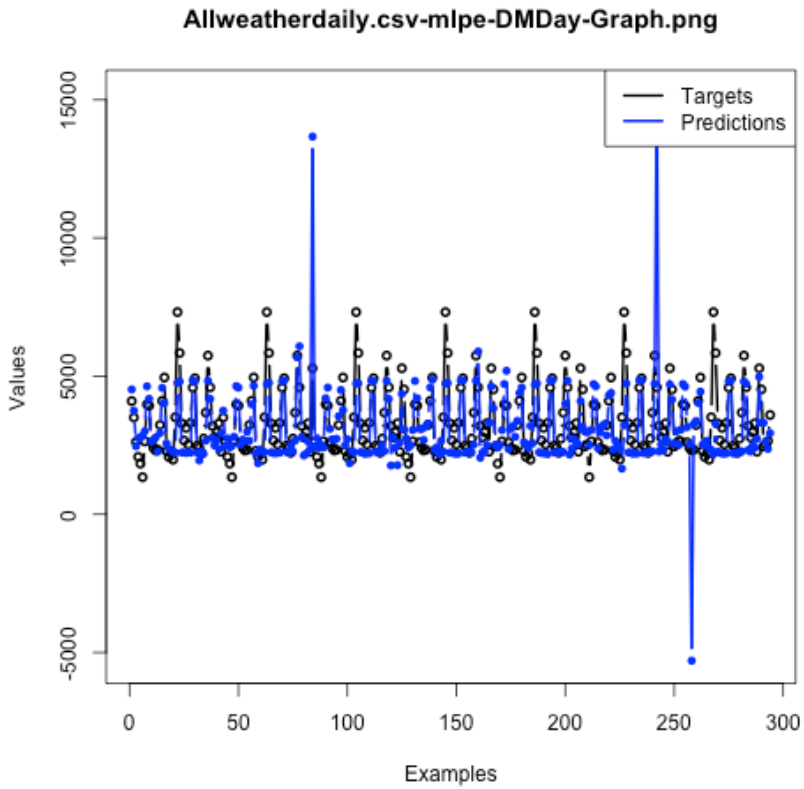

Figure 33 Daily (all data) MLPE model (Regression) forecasts

3.4.4. Hybrid approach

This approach is a hybrid of the time series modeling and the pure regression machine learning modeling. Using the rminer package in **R,** we were able to bind the time series lags with regression variables. For instance, for a daily scale, the prediction models will use all regression variables and also all human entrance time lags from 1 to 8. As previously explained in Section 3.4.3, due to limitations of the lforecast function, we were only able to predict up to a horizon of 1. The respective R code is made available at the **Appendix** section **A.3)**.

3.4.4.1. RandomForests

Figures 33 and 34 show the best forecasts provided by the random forest and hybrid approach.

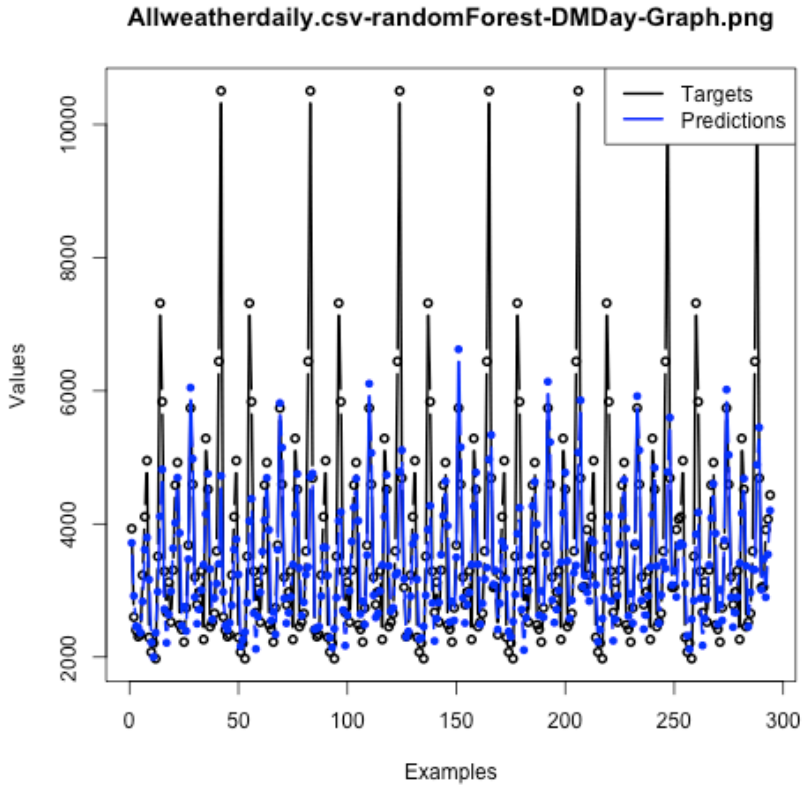

Figure 34 Daily (all data) RandomForest model (Hybrid) forecasts

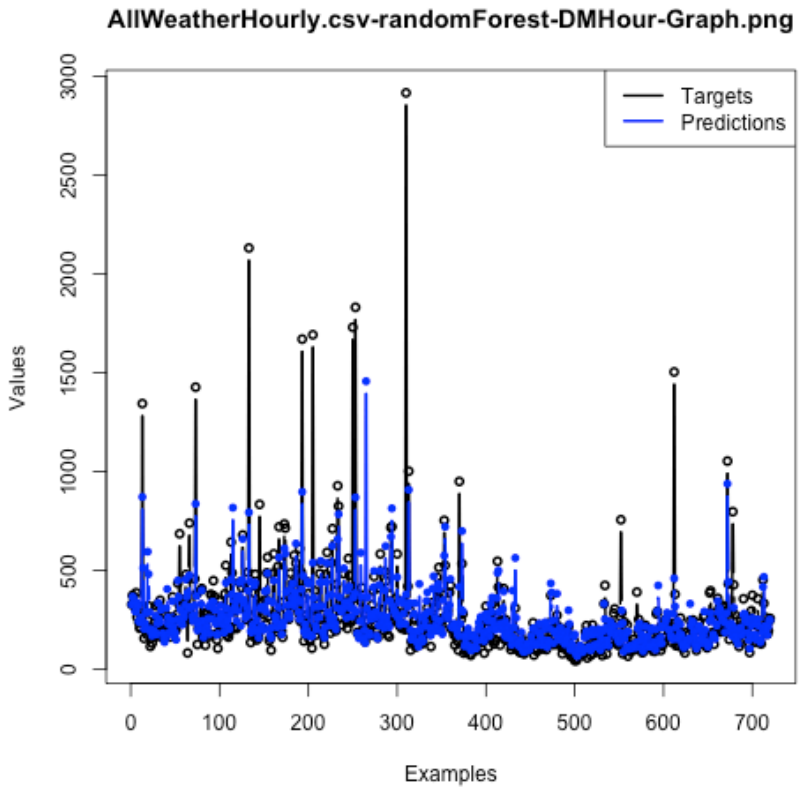

Figure 35 Hourly (all data) RandomForest model (Hybrid) forecasts

3.4.4.2. Rpart

Figures 35 and 36 plot the regression tree forecasts for the hybrid approach.

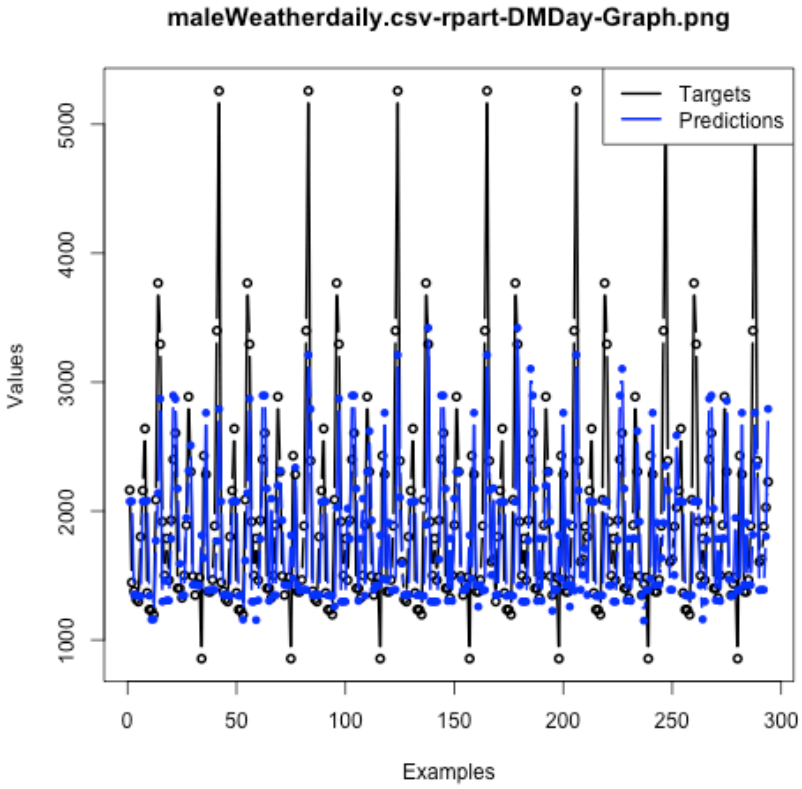

Figure 36 Daily males rpart model (Hybrid) forecasts

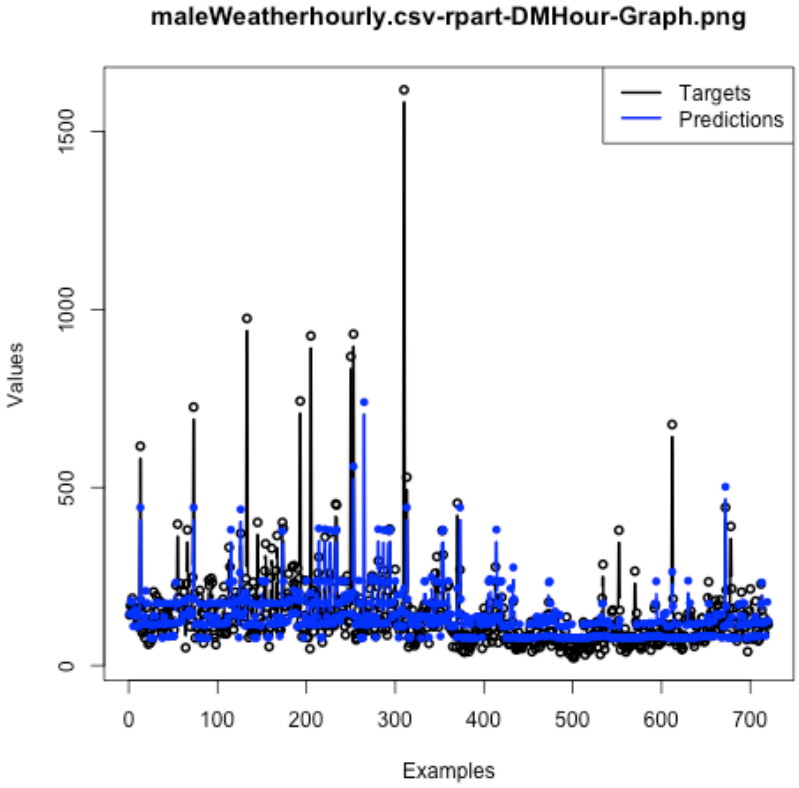

Figure 37 Hourly males rpart model (Hybrid) forecasts

3.4.4.3. KSVM

For the support vector machine and hybrid approach, Figures 37 and 38 show examples of the obtained forecasts..

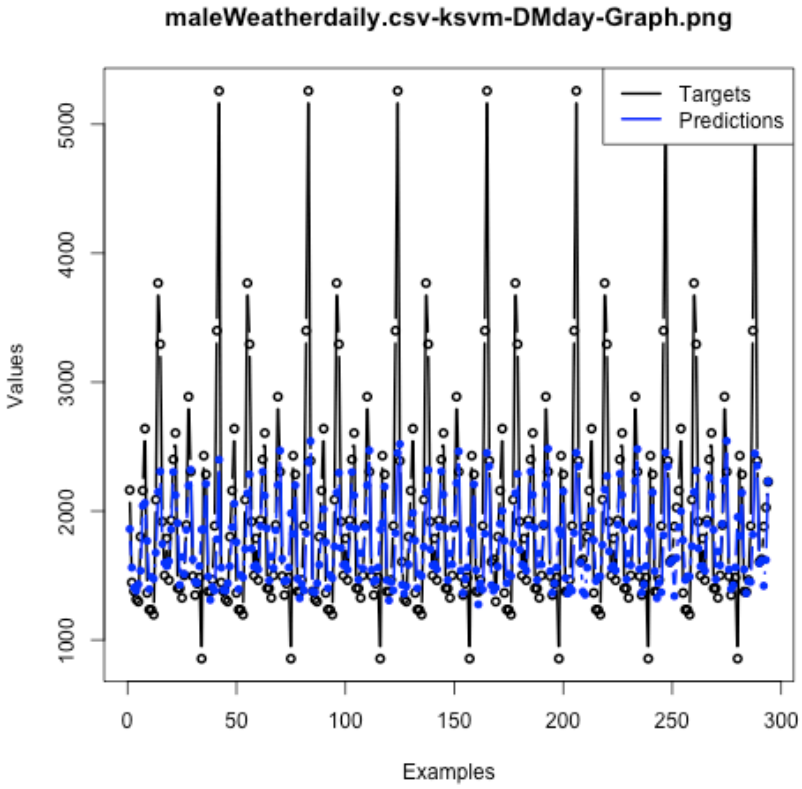

Figure 38 Daily males KSVM model (Hybrid) forecasts

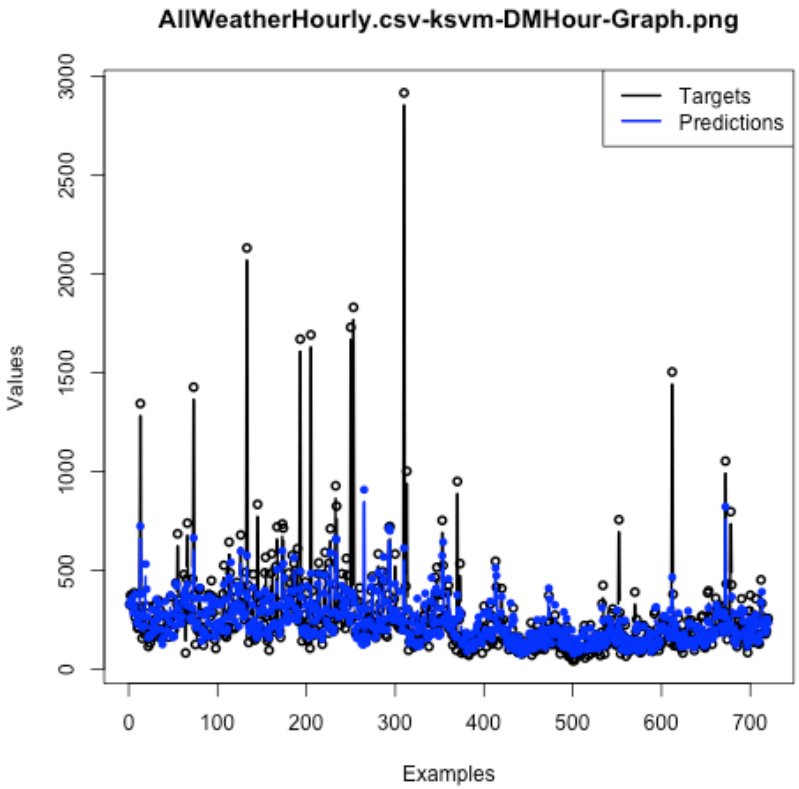

Figure 39 Hourly (all data) KSVM model (Hybrid) forecasts

3.4.4.4. MR

Figures 39 and 40 show the best forecasts using the multiple regression method and hybrid approach.

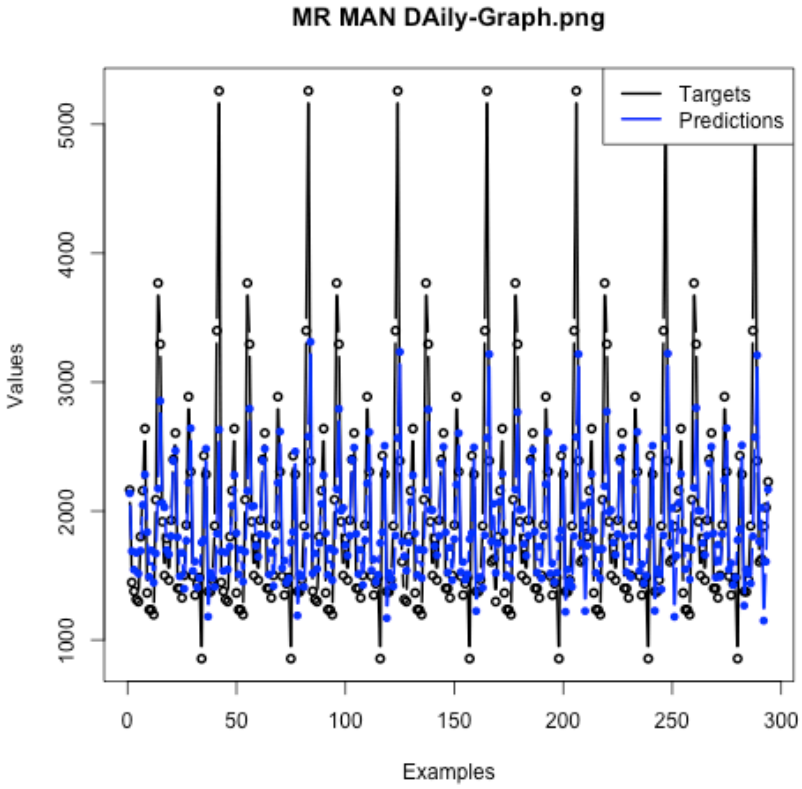

Figure 40 Daily males MR model (Hybrid) forecasts

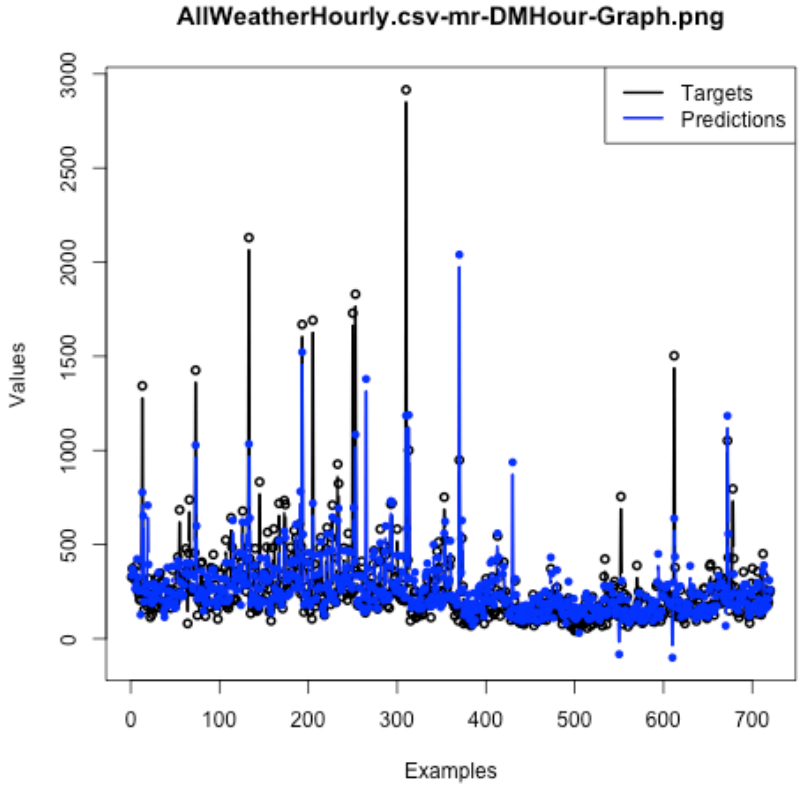

Figure 41 Hourly (all data) MR model (Hybrid) forecasts

3.4.4.5. MLPE

Figures 41 and 42 plot the forecasts for the neural nework model and hybrid approach.

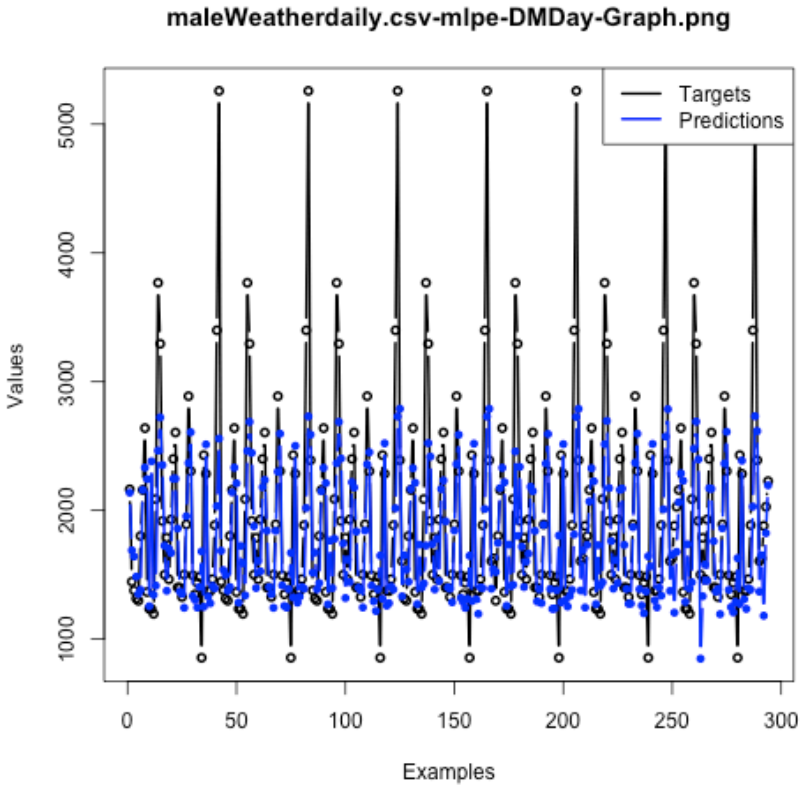

Figure 42 Daily males MLPE model (Hybrid)forecasts

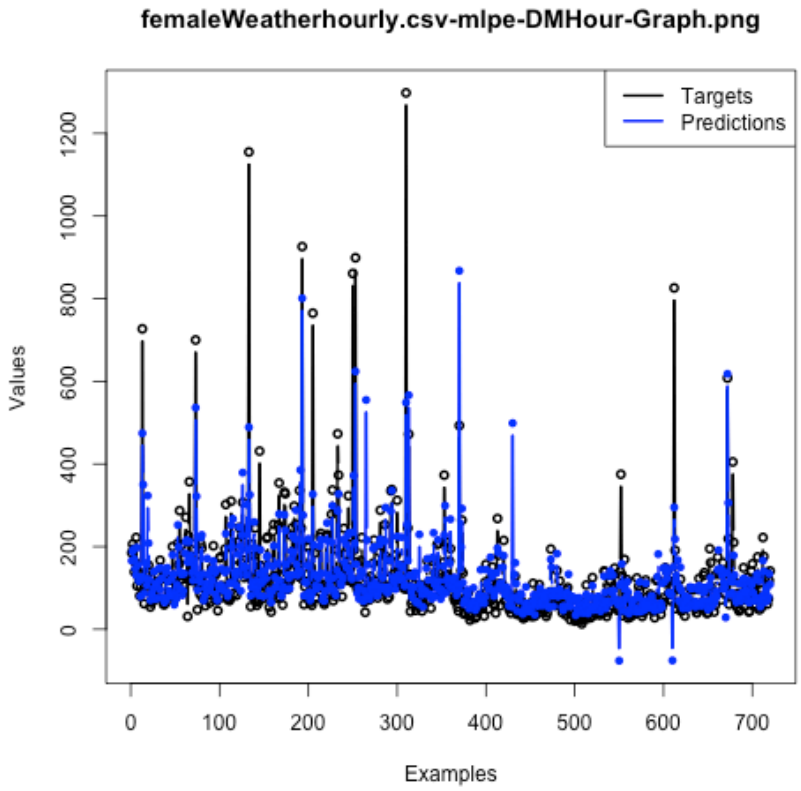

Figure 43 Hourly females MLPE model (Hybrid) forecasts

3.5. Evaluation phase

In this section, we analyze the modeling phase results. The daily analysis is provided in Tables 8 to 11, while the hourly results appear in Tables 12 to 15. As the evaluation error metric, we resorted to the MAPE metric, which stands for Mean Absolute Percent Error and that is a popular forecasting metric («Demand Planning.Net: Are you Planning By Exception?», 2015). One advantage of the MAPE metric is that it is scale independent, where the lower the metric, the better are the forecasts. In the tables, the best values are highlighted using **bold**.

Table 8 Daily analysis of ARIMA and HoltWinters (HW) MAPE values

Time Series Modeling								
Data Modeling	H1	H2	H3	H4	H5	H6	H7	Avg H
Holt-Winters Female	26.05	25.76	24.99	25.50	26.07	24.69	**22.63**	25.10
ARIMA Female	28.42	30.08	29.75	29.09	28.33	27.18	**25.92**	28.40
Holt-Winters Male	24.52	24.86	24.95	24.87	25.24	24.63	**22.19**	24.46
ARIMA Male	25.05	27.15	26.45	25.73	25.91	25.19	**21.23**	25.24
Holt-Winters All	23.16	23.12	22.81	22.76	23.44	22.35	**19.73**	**22.48**
ARIMA All	25.45	26.80	26.55	26.40	26.32	24.12	**20.87**	**25.22**

Table 9 Daily time series machine learning MAPE values

Machine Learning								
Data Modeling	H1	H2	H3	H4	H5	H6	H7	Avg H
All MR	**22.56**	24.76	24.89	23.40	23.49	23.16	22.60	23.55
All MLPE	**21.36**	23.37	24.15	24.27	21.57	23.42	25.78	23.42
All KSVM	18.19	17.75	**17.68**	17.78	20.44	21.02	21.62	**19.21**
All Random-Forest	**19.48**	21.27	22.60	22.60	22.68	22.58	21.84	21.87
All Rpart	**20.67**	23.92	24.42	24.09	25.88	27.77	25.66	24.63
Female-mr	**25.60**	28.94	29.15	27.66	27.67	27.30	26.84	27.59
Female-mlpe	**24.86**	32.01	30.93	27.68	28.18	27.33	25.62	28.09
Female-ksvm	**19.76**	22.78	23.57	24.79	26.78	25.60	25.69	**24.14**
Female-randomForest	**22.58**	25.45	26.36	27.95	27.50	27.04	26.58	26.21
Female-rpart	29.65	29.30	29.86	29.21	28.87	26.76	**25.95**	28.51
Male-mr	**22.31**	24.63	24.67	23.30	23.02	23.05	22.34	23.33
Male-mlpe	**23.29**	28.56	28.50	24.87	27.68	25.72	24.49	26.16
Male-ksvm	**17.64**	17.86	17.92	18.00	18.60	18.35	19.27	**18.23**
Male-randomForest	**19.77**	20.99	22.51	22.79	21.95	21.69	23.95	21.95
Male-rpart	**20.14**	28.39	32.98	34.90	29.00	24.26	22.88	27.51

Table 10 Daily hybrid approach MAPE values

Hybrid	
Data Modeling	**H1**
All MR	19.54
All MLPE	20.70
All KSVM	17.82
All Random-Forest	**16.88**
All Rpart	21.55
Female-mr	23.04
Female-mlpe	21.17
Female-ksvm	20.83
Female-randomForest	**19.90**
Female-rpart	29.63
Male-mr	18.73
Male-mlpe	20.54
Male-ksvm	**17.26**
Male-randomForest	17.62
Male-rpart	18.56

Table 11 Daily pure regression approach MAPE values

Regression	
Data Modeling	**H1**
All MR	21.68
All MLPE	18.86
All KSVM	20.62
All Random-Forest	**18.08**
All Rpart	19.98
Female-mr	27.28
Female-mlpe	25.48
Female-ksvm	24.99
Female-randomForest	26.09
Female-rpart	**23.93**
Male-mr	**18.86**
Male-mlpe	20.05
Male-ksvm	21.23
Male-randomForest	19.59
Male-rpart	20.13

Table 12 Hourly Holt-Winters and ARIMA MAPE values

Time Series Modeling													
Data Modeling	**H1**	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**	**H9**	**H10**	**H11**	**H12**	**H AVG**
Female ARIMA	46.83	**39.67**	58.49	55.36	52.07	42.01	40.66	79.18	102.93	185.20	117.33	91.74	75.96
Male ARIMA	34.99	**33.47**	45.53	40.96	40.87	37.56	33.90	53.17	66.82	114.79	85.42	62.87	**54.20**
All ARIMA	37.44	**34.04**	49.53	45.85	44.68	38.14	34.66	62.73	78.18	134.20	93.18	70.90	60.29
Female Holt-Winters	51.19	46.39	73.25	68.02	63.65	**46.34**	45.07	61.00	86.25	121.13	107.01	88.91	71.52
Male Holt-Winters	44.81	**39.15**	60.43	55.12	51.88	44.07	39.12	47.88	56.22	84.07	74.39	66.62	**55.31**
All Holt-Winters	43.69	**39.65**	63.87	58.67	55.16	41.86	39.97	52.78	67.37	94.64	84.08	73.89	59.64

Table 13 Hourly time series machine learning MAPE values

Machine Learning													
Data Modeling	**H1**	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**	**H9**	**H10**	**H11**	**H12**	**H AVG**
All MR	47.60	55.31	46.31	45.02	**34.55**	42.87	90.04	116.01	178.18	109.77	82.54	52.57	75.06
All MLPE	49.24	57.15	47.91	47.91	**36.34**	49.42	94.41	118.08	183.46	108.38	81.74	57.17	77.60
All KSVM	45.93	53.39	46.48	43.90	**33.25**	43.92	85.20	107.53	167.35	98.40	74.18	51.48	**70.92**
All Random-Forest	49.84	60.22	54.37	51.30	**39.84**	47.95	93.14	120.17	181.00	108.67	83.76	54.19	78.70
All Rpart	45.41	58.60	51.25	49.73	**43.66**	53.84	97.79	114.97	182.65	114.32	85.15	59.40	79.73
Female-mr	63.52	69.76	58.90	52.63	**43.76**	55.04	128.14	166.48	253.47	151.71	114.75	76.24	102.87
Female-mlpe	64.47	70.74	60.03	54.36	**46.51**	59.83	133.43	174.37	261.30	158.84	123.95	86.47	107.86
Female-ksvm	59.40	67.22	56.49	50.47	**41.79**	54.45	120.85	155.02	239.36	139.77	106.27	72.01	**96.92**
Female-randomForest	63.76	75.51	64.42	58.08	**50.38**	59.36	131.24	171.36	255.74	150.13	114.90	77.87	106.06
Female-rpart	62.04	64.33	54.88	50.03	**48.32**	59.96	128.91	159.21	255.87	151.76	115.24	78.37	102.41
Male-mr	41.34	45.31	40.05	41.83	**32.50**	37.15	70.08	88.12	142.73	87.29	65.61	42.25	61.19
Male-mlpe	43.76	47.41	42.52	43.72	**33.03**	39.77	70.22	86.98	139.85	84.01	63.95	45.14	61.70
Male-ksvm	39.65	43.71	40.60	40.12	**33.12**	38.50	66.14	80.16	132.41	74.93	57.66	40.11	**57.26**
Male-randomForest	44.50	54.23	51.08	53.51	**42.24**	47.72	82.60	112.26	168.98	105.38	76.68	52.53	74.31
Male-rpart	40.29	47.62	42.15	44.77	**39.15**	41.53	68.32	84.21	141.41	91.46	74.55	53.27	64.06

Table 14 Hourly hybrid MAPE values

Hybrid	
Data Modeling	**H1**
All MR	27.52
All MLPE	33.75
All KSVM	**22.64**
All Random-Forest	22.95
All Rpart	24.54
Female-mr	28.88
Female-mlpe	30.18
Female-ksvm	**24.80**
Female-randomForest	25.67
Female-rpart	25.72
Male-mr	30.70
Male-mlpe	30.81
Male-ksvm	24.87
Male-randomForest	24.39
Male-rpart	**24.21**

Table 15 Hourly pure regression approach MAPE values

Regression	
Data Modeling	**H1**
All MR	39.44
All MLPE	39.33
All KSVM	***33.43***
All Random-Forest	40.82
All Rpart	43.83
Female-mr	60.23
Female-mlpe	57.70
Female-ksvm	***47.02***
Female-randomForest	58.25
Female-rpart	63.08
Male-mr	31.79
Male-mlpe	32.78
Male-ksvm	***28.94***
Male-randomForest	34.48
Male-rpart	36.43

Table 16 summarizes the best forecasting models, condensing the MAPE values presented in Tables 8 to 15. To simplify the comparison among distinct method, only H=1 results were selected in Table 16. Analysing this table, it its clear that the hybrid approach produces the best forecasts in all cases except for the daily female entrances case, where the pure time series machine model gets the best result.

Table 16 Best models obtained from the modeling phase, regarding a horizon of 1 (H = 1)

		Time Series Modeling	Machine Learning Modeling	Regression Modeling	Hybrid Modeling
Hourly	**Male**	ARIMA 34.99	KSVM 39.65	KSVM 28.94	**Rpart 24.21**
	Female	ARIMA 46.83	KSVM 59.40	KSVM 47.02	**KSVM 24.80**
	All	ARIMA 37.44	KSVM 45.93	KSVM 33.43	**KSVM 22.64**
Daily	**Male**	Holt-Winters 24.52	KSVM 17.64	MR 18.86	**KSVM 17.26**
	Female	Holt-Winters 26.05	**KSVM 19.76**	Rpart 23.93	RandomForest 19.90
	All	Holt-Winters 23.16	KSVM 18.19	RandomForest 18.08	**RandomForest 16.88**

3.6. Implementation

With this study, it was possible to create several interesting forecasting, which may consist a valuable asset for the retail store, assisting in decisions related with management of its human resources, product inventory or marketing operations. However, due to time restrictions, implementation of this CRISP-DM phase was not possible, it stays here only mentioned as future work .

3.7. Summary

During this study a large number of experiments were held, targeting distinct types of human entrance predictions (i.e., female, male, both genders), forecasting periods (i.e., hourly and daily) and lookahead (horizon) predictions. Moreover, several forecasting methods were tested: conventional time series methods and time series models based on machine learning; a regression approach (e.g., using weather and special event data); and a hybrid approach that uses both time series (human entrances time lags) and regression variables.

When comparing the distinct forecasting approaches, the best results were achieved (in general) by the hybrid approach. Also, although most of the results proved to be promising, we regarded the hourly models, such as the conventional time series approach and the machine learning approach to be the worst models conceived by overall, since all of them proved to have a MAPE above 50%, however the hourly hybrid approach proved to be fairly promising as well, as stated on the Table 14, since all models have a MAPE below the 33%. The best model, according to the metrics presented in the section 3.6, proved to be **daily hybrid modeling**, more specifically the RandomForest model (for both sex) and KSVM model (for males), since it predicts with the best accuracy, although includes more variables, and also has the lowest error margin according to the MAPE metric, roughly 16.88 % and 17.26% respectively. Such forecasting models are potentially valuable for commercial store managers. For instance, they can help in the management of the human resources and marketing campaigns.

4. Conclusions

4.1. Summary

The purpose of this work was to predict human entrances, on a houly or daily basis, of clients in a commercial store. Such human entrances were captured using a special facial recognition system based on video analysis and that was implemented in a particular commercial store, during a pilot project. This data were kindly provided a private company, as described in the Acknowledments section of this work. Such dataset was then enriched with other features, such as weather variables and local events that took place near or in the city surrounding the retail area. Using a data mining approach, under the CRISP-DM methodology and the R tool, we were able to explored create various approaches in which resulted in various models that could prove useful to this study as well.

We surveyed four studies that involve forecasting application studies. In particular, two of these studies are more relared with this work. Yet, there are different opinions and methodologies used regarding the best models used. For example (Rodrigues, 2014) clearly states that the best model to use in the matter is the Regression Tree model, while (Lin et al., 2011) concluded that the ARIMA (2,0,0)x(2,0,2)12 predicted better when compared with other machine learning methods.

During this study a large number of experiments were held, targeting distinct types of human entrances (i.e., female, male, all), forecasting periods (i.e., hourly and daily) and lookahead (horizon) predictions. Moreover, several forecasting methods were verified such as: conventional time series methods and time series models based on machine learning; a regression approach (e.g. using weather and special event data); and a hybrid approach that uses both time series (human entrances time lags) and regression variables. The conclusion reached in the end, is that the hybrid approach tends to provide the best forecasts. In particular, the daily hybrid modeling, more specifically the random forest model (for all data) and support vector machine algorithm (for males), obtained the lowest MAPE values (16.88% and 17.26%, respectively). We belive that such forecasting models are potentially valuable for commercial store managers. For instance,

they can help in making decisions about the management of the retail store and related with product inventory, human resources and marketing operations.

4.2. Discussion

This study provides new insight regarding the use of data mining techniques to forecast human accesses in a commercial retail store. Using the data obtained through the camera, we were able to measure those same accesses, from April 2013 to December 2013, having relevant predictive information such as the weather, local events and date related measures (e.g. Monday, Holiday, among others). At the end of the modeling phase, we created several models containing various approaches, that in some cases proved not to be the most reliable models to consider, such as the hourly analysis using the machine learning approach and the conventional time series approach. As already explained, the best forecasting approach tends to be the hybrid one. In particular, best models are related with the daily hybrid methods, which provided a reasonable low MAPE error.

This study also presents some limitations. First of all, all data analysis was conducted offline, i.e., after the data was collected and we did not have direct access to the retail store managers or pilot implementors. This restricted the type and quality of data attributes that we could collect. Also, limited the type of feedback that our forecasting models could provide for the retail store managers. In other words, difficulted a real assessment of the business impact of the developed data-driven models. The second limitation is that all data analysis was dependent on the quality of the video detection system, which is assumed to be of high quality but whose working mechanism is not disclosed, thus being black-box.The third limitation was the time available to execute this work. Due to the lack of time, other intesting modeling approaches are left for future work.

4.3. Future Work

In this section, we detail several future directions that could be followed after the conclusion of this work:

- It would be interesting to have the best forecasting models implemented in a real retail store, providing forecasts in real-time and obtaining feedback from the store managers in terms of their usefulness for supporting their decisions.

- Another interesting future research topic is the forecast of intervals instead of single values. Using machine learning models, it is possible to forecast a interval range of human entrances, predicting a minimum and maximum value (each one with a desirable high confidence) of such entrances.
- Collect and test the value of more regression variables, such as other types of human interesting events, such as traffic congestion or public transportation company strikes.
- Conduct similar pilots and forecasting approaches at other retail stores.
- Adapt the rminer functions to multivariate data, allowing the use of several lookaheads predictions (horizon>1) when non lagged regression variables are also included in the forecasting model.

Bibliography

Ang, B.-K., Dahlmeier, D., Lin, Z., Huang, J., Seeto, M.-L., & Shi, H. (2014, 2014). *Indoor Next Location Prediction with Wi-Fi.* Paper presented at the The Fourth International Conference on Digital Information Processing and Communications (ICDIPC2014).

Carr, A. (2011). Strip Development and Community: Maintaining a Sense of Place. *Masters Theses*.

Chapman, C. J., Kerber Randy ,, Khabaza Thomas , R. T., & Rüdiger, S. C. a. W. (2000). *CRISP-DM 1.0*: SPSS.

Gilks, W. R. (2005). Markov Chain Monte Carlo *Encyclopedia of Biostatistics*: John Wiley & Sons, Ltd.

Hall, I. H. W. E. F. M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques*: ELSEVIER.

Hastie, T., Tibshirani R., Friedman J., Hastie T., Friedman J., & Tibshirani, R. (2009). *The elements of statistical learning (Vol. 2, No. 1)*: New York: springer.

Lin, C. J., Chen, H. F., & Lee, T. S. (2011). Forecasting tourism demand using time series, artificial neural networks and multivariate adaptive regression splines: evidence from Taiwan. *International Journal of Business Administration, 2*(2), p14.

Rakesh Agrawal, T. I., and Arun Swami. (1993). *Mining association rules between sets of items in large databases.* : SIGMOD Rec., 22(2):207216, June 1993.

Rodrigues, T. O. (2014). *Previsão de Movimentos em Espaços Comerciais a Partir de Dados de Posicionamento de Clientes.* (Tese de Mestrado), Faculdade de Ciências da Universidade do Porto, Porto.

Shaik, M. A., Rao, S. N., & Rahim, A. (2013). A SURVEY OF TIME SERIES DATA PREDICTION ON SHOPPING MALL. *Indian Journal of Computer Science and Engineering, 4*(2), 174-184.

Song, H., & Li, G. (2008). Tourism demand modelling and forecasting—A review of recent research. *Tourism Management, 29*(2), 203-220. doi: http://dx.doi.org/10.1016/j.tourman.2007.07.016

Team, R. C. (2014). Vienna, Austria Patent No.: R. F. f. S. Computing.

Williams, G., Culp, M. V., Cox, E., Nolan, A., White, D., Medri, D., . . . print.summary.nnet, B. R. (2014). rattle: Graphical user interface for data mining in R (Version 3.3.0). Retrieved from http://cran.r-project.org/web/packages/rattle/index.html

Cortez, P. (2015). *Data Mining Classification and Regression Methods* [Package]. R. Obtido de http://cran.r-project.org/package=rminer http://www3.dsi.uminho.pt/pcortez/rminer.html

Demand Planning.Net: Are you Planning By Exception? (2015, Novembro 10). Obtido 11 de Outubro de 2015, de http://demandplanning.net/MAPE.htm

Fayaad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Database. Obtido de http://www.csd.uwo.ca/faculty/ling/cs435/fayyad.pdf

Hyndman, R. (2015). Forecasting Functions for Time Series and Linear Models (Versão 6.1) [R]. Obtido de https://cran.r-project.org/web/packages/forecast/forecast.pdf

Wunderground. (2010). [Weather Reports]. Obtido de http://www.wunderground.com

Appendices

Appendix A

A.1) Time Series ARIMA and Holtwinters code

```
library(forecast)
library(lubridate)
library(rminer)



#geting updated models
femaleWeatherdaily = read.csv("femaleWeatherdaily.csv",sep=";")
maleWeatherdaily = read.csv("maleWeatherdaily.csv",sep=";")
femaleWeatherhourly = read.csv("femaleWeatherhourly.csv",sep=";")
maleWeatherhourly = read.csv("maleWeatherhourly.csv",sep=";")
allWeatherdaily = read.csv("Allweatherdaily.csv",sep=";",dec = ",")
allWeatherhourly = read.csv("AllWeatherHourly.csv",sep=";",dec = ",")
#modeling phase!
 cenarios_day=c("Todosdia","Homemdia","Mulherdia")
cenarios_hour=c("TodosHora","Homemhora","MulherHora")

cenariostotais=c("Todosdia","TodosHora","Homemdia","Mulherdia","Homemhora",
 "MulherHora")
directory = "/Users/user/Desktop/Modulation Phase 3 Time Series"
#predict
checkcenariodia= function(st) {
 if(st=="Todosdia") st = allWeatherdaily$ts
 if (st=="Homemdia") st= maleWeatherdaily$ts
 if (st=="Mulherdia") st= femaleWeatherdaily$ts



return(st);
}
checkcenariohora= function(st) {
```

```
 if(st=="TodosHora") st = allWeatherhourly$ts
 if (st=="Homemhora") st= maleWeatherhourly$ts
 if (st=="MulherHora") st= femaleWeatherhourly$ts


return(st);
}

#Save metrics on a file
write.table("Metrics","metrics.txt",sep="\n",eol="\n",  col.names=FALSE,
 row.names=FALSE)
modeling=function(frequencytime = "day" , H = 7, SW = 1, W= 200){


if(frequencytime == "hour"){
for(cenario in cenarios_hour)
{

# show visually the time series:
#tsdisplay(checkcenario(cenario), main=cenario)
#mpause(cenario)

L=length(checkcenariohora(cenario))
V = ((L-W)/SW)
V = V -H
VectorHWH = vector("list",V)
VectorARH = vector("list",V)

mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
for(i in 1:V){
#rolling windows
HO=holdout(checkcenariohora(cenario),  ratio=H,  mode="rolling",  iter=i,
 window=W, increment=SW)
```

```
write.table(cenario,"metrics.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table("Holt Winters","metrics.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)

# time series week object:

TR=ts(checkcenariohora(cenario)[HO$tr],frequency=12,start=1,end=length(HO$t
 r))
 target=checkcenariohora(cenario)[HO$ts]



# holt winters:
HW=HoltWinters(TR)
F=forecast(HW,h=H)
Pred=F$mean[1:H]
#mgraph(target,Pred,graph="REG",Grid=10,col=c("black","blue"),leg=list(pos="t
 opleft",leg=c("target","predictions")), main=c("holt winters:",cenario))
 cat("iteration:",i,"\nTarget:",target,"\n")
print(c("holt winters: ",cenario))
hwmmetric=mmetric(target,Pred,metric=c("MAE", "RAE", "RMSE", "RRSE", "R22"))
print(hwmmetric)
write.table(mmetrics,"metrics.txt",sep="\n", eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(hwmmetric,"metrics.txt",sep="\t", eol="\t", append=TRUE,
 col.names=FALSE, row.names=FALSE)



# ARIMA modeling:
AR=auto.ARIMA(TR)
```

```
F1=forecast(AR,h=H)
Pred1=F1$mean[1:H]
#mgraph(target,Pred1,graph="REG",Grid=10,col=c("black","blue"),leg=list(pos="
 topleft",leg=c("target","predictions")), main=c("ARIMA:",cenario))



print(c("ARIMA modeling:",cenario))
 cat("iteration:",i,"\nTarget:",target,"\n")
armmmetric=mmetric(target,Pred1,metric=c("MAE",  "RAE",  "RMSE",  "RRSE",
 "R22"))
print(armmmetric)
write.table(mmetrics,"metrics.txt",sep="\n",  eol="\n",  append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(armmmetric,"metrics.txt",sep="\t",  eol="\t",  append=TRUE,
 col.names=FALSE, row.names=FALSE)
VectorHWH[[i]]=list(target=target,pred=Pred)
VectorARH[[i]]=list(target=target,pred=Pred1)
save(VectorHWH, VectorARH,file=paste("MetricsHours",cenario))
}

} }else {
 for(cenario in cenarios_day)
{
# show visually the time series:
#tsdisplay(checkcenario(cenario), main=cenario)
#mpause(cenario)

L=length(checkcenariodia(cenario))
V = ((L-W)/SW)
V = V -H
VectorHWD = vector("list",V)
VectorARD = vector("list",V)
```

```
# time series week object:

mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")

for(i in 1:V){
 write.table(cenario,"metrics.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table("Holt Winters","metrics.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
 #rolling windows
HO=holdout(checkcenariodia(cenario), ratio=H, mode="rolling", iter=i,
 window=W, increment=SW)

TR=ts(checkcenariodia(cenario)[HO$tr],frequency=7,start=1,end=length(HO$tr))
 target=checkcenariodia(cenario)[HO$ts]


# holt winters:
HW=HoltWinters(TR)
F=forecast(HW,h=H)
Pred=F$mean[1:H]


#mgraph(target,Pred,graph="REG",Grid=10,col=c("black","blue"),leg=list(pos="t
 opleft",leg=c("target","predictions")), main=c("holt winters:",cenario))


print(c("holt winters: ",cenario))
 cat("iteration:",i,"\nTarget:",target,"\n")
hwmmetric=mmetric(target,Pred,metric=c("MAE", "RAE", "RMSE", "RRSE", "R22"))
print(hwmmetric)
write.table(mmetrics,"metrics.txt",sep="\n", eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
```

```
write.table(hwmmetric,"metrics.txt",sep="\t", eol="\t", append=TRUE,
 col.names=FALSE, row.names=FALSE)


# ARIMA modeling:
AR=auto.ARIMA(TR)
F1=forecast(AR,h=H)
Pred1=F1$mean[1:H]
#mgraph(target,Pred1,graph="REG",Grid=10,col=c("black","blue"),leg=list(pos="
 topleft",leg=c("target","predictions")), main=c("ARIMA:",cenario))



print(c("ARIMA modeling:",cenario))
 cat("iteration:",i,"\nTarget:",target,"\n")
armmmetric=mmetric(target,Pred1,metric=c("MAE", "RAE", "RMSE", "RRSE",
 "R22"))
print(armmmetric)
write.table(mmetrics,"metrics.txt",sep="\n", eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(armmmetric,"metrics.txt",sep="\t", eol="\t", append=TRUE,
 col.names=FALSE, row.names=FALSE)
VectorHWD[[i]]=list(target=target,pred=Pred, cenario = cenario)
VectorARD[[i]]=list(target=target,pred=Pred1, cenario = cenario)

save(VectorHWD,VectorARD,file=paste("Metricsdays",cenario))


}


}
}
```

```
}


vectorgraph=function(nametoload = "Metricsdays Todosdia", desiredmodel="Holt-
 Winters",H = 7, metrics = "MAPE",  graphs = "REG"){
  load(nametoload)
  #H defines days or hours days are 7 days per week and hours is 12 hours per day...
 if(H==7){
  if(desiredmodel=="Holt-Winters"){
       v = VectorHWD
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
  htarg[i,]=v[[i]]$target
  hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
      name = paste(nametoload, desiredmodel,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)
```

```
 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = name)
 dev.off()
          write.table(nametoload,"model.txt",sep="\n",eol="\n",  append=TRUE,
 col.names=FALSE, row.names=FALSE)
}else{

          v = VectorARD
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
     name = paste(nametoload,desiredmodel,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)
```

```
 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = name)
 dev.off()
        write.table(nametoload,"model.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
 }

}else{

 #hourly vectors....
 if(desiredmodel=="Holt-Winters"){
     v = VectorHWH
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say "MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
     name = paste(nametoload, desiredmodel,"Graph.png", sep="-")
```

```
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = name)
 dev.off()
 write.table(nametoload,"model.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)

}else{

        v = VectorARH
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
     name = paste(nametoload, desiredmodel,"Graph.png", sep="-")
```

```
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = name)
 dev.off()
        write.table(nametoload,"model.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)

 }
 }
}
```

A.2) Pure Time Series using Datamining Modeling Code

```
 #phase4 modeling with Data Mining models
library(forecast)
library(lubridate)
library(rminer)
femaleWeatherdaily = read.csv("femaleWeatherdaily.csv",sep=";")
maleWeatherdaily = read.csv("maleWeatherdaily.csv",sep=";")
femaleWeatherhourly = read.csv("femaleWeatherhourly.csv",sep=";")
maleWeatherhourly = read.csv("maleWeatherhourly.csv",sep=";")
allWeatherdaily = read.csv("Allweatherdaily.csv",sep=";",dec = ",")
allWeatherhourly = read.csv("AllWeatherHourly.csv",sep=";",dec = ",")
#modeling phase!
 cenarios_day=c("Todosdia","Homemdia","Mulherdia")
cenarios_hour=c("TodosHora","Homemhora","MulherHora")

cenariostotais=c("Todosdia","TodosHora","Homemdia","Mulherdia","Homemhora","Mu
 lherHora")
directory = "/Users/LuisRMatos/Desktop"
#predict
```

```
checkcenariodia= function(st) {
 if(st=="Todosdia") st = allWeatherdaily$ts
 if (st=="Homemdia") st= maleWeatherdaily$ts
 if (st=="Mulherdia") st= femaleWeatherdaily$ts


return(st);
}
checkcenariohora= function(st) {
 if(st=="TodosHora") st = allWeatherhourly$ts
 if (st=="Homemhora") st= maleWeatherhourly$ts
 if (st=="MulherHora") st= femaleWeatherhourly$ts


return(st);
}

models=c("rpart","mlpe","ksvm","randomForest","mr")

 vectorDays=c("")
vectorDays=c("Mulherdia-rpart-DMDay",
"Mulherdia-mlpe-DMDay",
"Mulherdia-randomForest-DMDay",
"Mulherdia-mr-DMDay",
"Homemdia-rpart-DMDay",
"Homemdia-mlpe-DMDay",
"Homemdia-randomForest-DMDay",
"Homemdia-mr-DMDay",
"Todosdia-rpart-DMDay",
"Todosdia-mlpe-DMDay",
"Todosdia-randomForest-DMDay",
"Todosdia-mr-DMDay",
"Todosdia-ksvm-DMDay",
```

```
"Homemdia-ksvm-DMDay",
"Mulherdia-ksvm-DMDay")
vectorHours = c("MulherHora-rpart-DMHour","MulherHora-mlpe-
 DMHour","MulherHora-mr-DMHour","Homemhora-rpart-DMHour","Homemhora-
 mlpe-DMHour","Homemhora-mr-DMHour","TodosHora-rpart-DMHour","TodosHora-
 mlpe-DMHour","TodosHora-mr-DMHour","MulherHora-randomForest-
 DMHour","Homemhora-randomForest-DMHour","TodosHora-randomForest-
 DMHour","TodosHora-ksvm-DMHour","MulherHora-ksvm-DMHour","Homemhora-
 ksvm-DMHour")


 modeling=function(frequencytime = "day" , H = 7, W = 200,SW=1 ){


if(frequencytime == "hour"){
 #models=c("naive","rpart","kknn","mlp","mlpe","ksvm","randomForest","mr","cubist",
 "pcr", "plsr", "cppls") #rvm model not used because of compute problems (only in the
 hourly cenarios...)
for(cenario in cenarios_hour)
{
write.table(cenario,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
 L=length(checkcenariohora(cenario))
V = ((L-W)/SW)
V = V -H*2 #aqui 5 iterações que criou que estão erradas do genero (target : 1 2 3 4 5 6
 7) os ultimos 5 corresponderiam a ( 2, 3, 4, 5,6,7) (3,4,5,6,7)....
Vectormodels = vector("list",V)


#dtr=1:(L-H)
testes = CasesSeries(checkcenariohora(cenario), W = c(1,2,3,4,5,6,7,8,9,10,11,12,13))

for(model in models)
```

```
{

    s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto",
 4/5),metric="SAE")

 modelfinal=paste(cenario,model,";")
 write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")


 b = 1
 c=1
 for(b in 1:V){
    modelfinal=paste(cenario,model,"rolling window",";")
 write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
   HO=holdout(testes$y, ratio=H, mode="rolling", iter=b, window=W, increment=SW)
   cat("iteration:",b,"\nTarget:",testes$y[HO$ts],"\n")
   M=fit(y~.,testes[HO$tr,],model=model, search = s)
     PredMR=predict(M,testes[HO$ts,])
   cat("PRED:",round(PredMR,digits=0),"\n")

    mmetric=mmetric(testes$y[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
 "RRSE", "R22","MAPE"))
print(mmetric)
write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)


 c=c+1#contador para o start do lforecast
```

```
 Vectormodels[[b]]=list(target=testes$y[HO$ts],pred=PredMR,model = model ,
 cenario = cenario)
 write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)

}

  file =paste(cenario,model,"DMhour", sep="-")

 save(Vectormodels,file=file)
}
}

}else {

 for(cenario in cenarios_day)
{



L=length(checkcenariodia(cenario))




V = ((L-W)/SW)
V = V - H*2 #isto so permite ir ate  ás iterações possiveis senão começa a reduzir de 7
 para 6,5,4,3,2,1 que corrspondem aos ultimos targets conhecidos

dtr=1:(L-H)
testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))
```

```
Vectormodels = vector("list",V)
for(model in models)
 {

 s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto"),m
 etric="SAE")
  b = 1
 modelfinal=paste(cenario,model,";")
 write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
c=1
 for(b in 1:V){ # beginning of the rolling windows predictions using rminer package...


 modelfinal=paste(cenario,model,"rolling window",";")
 write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
   HO=holdout(testes$y, ratio=H, mode="rolling", iter=b, window=W, increment=SW)
   cat("iteration:",b,"\nTarget:",testes$y[HO$ts],"\n")

   M=fit(y~.,testes[HO$tr,],model=model,search=s)
   PredMR=predict(M,testes[HO$ts,])
      cat("PRED:",round(PredMR,digits=0),"\n")

     cat("Model:",model,"MAE:",mmetric(testes$y[HO$ts], PredMR,"MAE"),"\n")

   mmetric=mmetric(testes$y[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
 "RRSE", "R22","MAPE"))
print(mmetric)
write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
```

```
 col.names=FALSE, row.names=FALSE)
name = paste(cenario, model ,b, 'rollingwindow.png', sep="-")

 c=c+1#contador para o start do lforecast


    Vectormodels[[b]]=list(target=testes$y[HO$ts],pred=PredMR,model = model ,
 cenario = cenario)
write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)



}
file =paste(cenario,model,"DMday", sep="-")
 save(Vectormodels,file=file)




 }

}
}
}

#this function helps generating the REC graphics for all models using in the datamining
 modeling which allows to compare them afterwards! and other graphics such as scatter
```

```
 plot or REG
recGraph = function(time = "day", H=7, graphic = "REC"){
 if(time == "hour"){
        for(cenario in cenarios_hour)
{
 testes = CasesSeries(checkcenariohora(cenario), W = c(1,2,3,4,5,6,7,8,9,10,11,12,13))

M1=mining(y~.,data=testes,method=c("holdoutorder",H),model="rpart")
M2=mining(y~.,data=testes,method=c("holdoutorder",H),model="mlpe")
M3=mining(y~.,data=testes,method=c("holdoutorder",H),model="ksvm")
M4=mining(y~.,data=testes,method=c("holdoutorder",H),model="randomForest")
M5=mining(y~.,data=testes,method=c("holdoutorder",H),model="mr")


VEC=vector("list",5); VEC[[1]]=M1;
 VEC[[2]]=M2;VEC[[3]]=M3;VEC[[4]]=M4;VEC[[5]]=M5;
 name = paste(cenario,'RecCurve.png', sep="_")
 fileName= paste(directory, name, sep="/")#
 png(filename=fileName)
 mgraph(VEC,graph="REC",col=c("black","blue","red","green","yellow"),leg=list(pos="
 topright",leg = c("rpart","mlpe","ksvm","randomForest","mr")),main=paste("REC
 curve for all models", cenario, sep = "-"))
 dev.off()
}
}
else{

 for(cenario in cenarios_day)
{
        testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))

M1=mining(y~.,data=testes,method=c("holdoutorder",H),model="rpart")
M2=mining(y~.,data=testes,method=c("holdoutorder",H),model="mlpe")
```

```
M3=mining(y~.,data=testes,method=c("holdoutorder",H),model="ksvm")
M4=mining(y~.,data=testes,method=c("holdoutorder",H),model="randomForest")
M5=mining(y~.,data=testes,method=c("holdoutorder",H),model="mr")
        VEC=vector("list",5); VEC[[1]]=M1;
 VEC[[2]]=M2;VEC[[3]]=M3;VEC[[4]]=M4;VEC[[5]]=M5;
 name = paste(cenario,graphic,'Curve.png', sep="_")
 fileName= paste(directory, name, sep="/")#
 png(filename=fileName)
 mgraph(VEC,graph=graphic,col=c("black","blue","red","green","yellow"),leg=list(pos
 ="topright",leg =
 c("rpart","mlpe","ksvm","randomForest","mr")),main=paste(graphic,"curve for all
 models", cenario, sep = "-"))
 dev.off()

 }
}
}

vectorgraph=function( H = 7, metrics = "MAPE",  graphs = "REG"){


 if(H == 7){
    for(nametoload in(vectorDays)){
    load(nametoload)
  v = Vectormodels
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
```

```
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
 hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
 print(hmae)
    name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = "Target vs
 Prediction")
 dev.off()

 write.table(nametoload,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)

 }
}else{
 for(nametoload in (vectorHours)){
 load(nametoload)
 print(nametoload)
  v = Vectormodels
 N=length(v)-4 #this is because the hourly model generates 4 null pointers at the end...
```

```
htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
 name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"),
 leg=list(pos="topright",leg = c("Targets", "Predictions")),main = "Target vs
 Prediction")
 dev.off()
    write.table(nametoload,"MetricsHours.csv",sep=";",eol="\n", dec =
 ",",append=TRUE, col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsHours.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)
 }
 }
```

```
}
```

A.3) Datamining Modeling Mix code

```
#phase 4 modeling with Data Mining models MIX
library(forecast)
library(lubridate)
library(rminer)
femaleWeatherdaily = read.csv("femaleWeatherdaily.csv",sep=";")
maleWeatherdaily = read.csv("maleWeatherdaily.csv",sep=";")
femaleWeatherhourly = read.csv("femaleWeatherhourly.csv",sep=";")
maleWeatherhourly = read.csv("maleWeatherhourly.csv",sep=";")
allWeatherdaily = read.csv("Allweatherdaily.csv",sep=";",dec = ",")
allWeatherhourly = read.csv("AllWeatherHourly.csv",sep=";",dec = ",")
#modeling phase!
#femaleWeatherhourly.csv
dataD=c("femaleWeatherdaily.csv","maleWeatherdaily.csv","Allweatherdaily.csv")
datah=c("AllWeatherHourly.csv","femaleWeatherhourly.csv","maleWeatherhourly.csv")
models=c("randomForest","rpart","mlpe","mr","ksvm")
modelsh=c("mr","rpart","mlpe")
cenariostotais=c("Todosdia","TodosHora","Homemdia","Mulherdia","Homemhora","Mu
 lherHora")
directory = "/Users/user/Desktop/Models DM Phase 3"
#predict
checkcenariodia= function(st) {
 if(st=="Todosdia") st = allWeatherdaily$ts
 if (st=="Homemdia") st= maleWeatherdaily$ts
 if (st=="Mulherdia") st= femaleWeatherdaily$ts


return(st);
```

```
}
checkcenariohora= function(st) {
 if(st=="TodosHora") st = allWeatherhourly$ts
 if (st=="Homemhora") st= maleWeatherhourly$ts
 if (st=="MulherHora") st= femaleWeatherhourly$ts


return(st);
}


vectordaily=c("femaleWeatherdaily.csv-rpart-DMDay",
"femaleWeatherdaily.csv-mlpe-DMDay",
"femaleWeatherdaily.csv-randomForest-DMDay",
"femaleWeatherdaily.csv-mr-DMDay",
"maleWeatherdaily.csv-rpart-DMDay",
"maleWeatherdaily.csv-mlpe-DMDay",
"maleWeatherdaily.csv-randomForest-DMDay",
"maleWeatherdaily.csv-mr-DMDay",
"Allweatherdaily.csv-rpart-DMDay",
"Allweatherdaily.csv-mlpe-DMDay",
"Allweatherdaily.csv-randomForest-DMDay",
"Allweatherdaily.csv-mr-DMDay",
"femaleWeatherdaily.csv-ksvm-DMday",
"Allweatherdaily.csv-ksvm-DMday",
"maleWeatherdaily.csv-ksvm-DMday")
vectorHourly = c("femaleWeatherhourly.csv-ksvm-DMHour","AllWeatherHourly.csv-
 ksvm-DMHour","maleWeatherhourly.csv-ksvm-DMHour","femaleWeatherhourly.csv-
 rpart-DMHour","femaleWeatherhourly.csv-mlpe-DMHour","femaleWeatherhourly.csv-
 mr-DMHour","maleWeatherhourly.csv-rpart-DMHour","maleWeatherhourly.csv-mlpe-
 DMHour","maleWeatherhourly.csv-mr-DMHour","AllWeatherHourly.csv-rpart-
 DMHour","AllWeatherHourly.csv-mlpe-DMHour","AllWeatherHourly.csv-mr-
 DMHour","maleWeatherhourly.csv-randomForest-
```

```
 DMHour","femaleWeatherhourly.csv-randomForest-DMHour","AllWeatherHourly.csv-
 randomForest-DMHour")


 modeling=function(frequencytime = "day" , H = 7, W = 200,SW=1){


if(frequencytime == "hour"){
 #models=c("naive","rpart","kknn","mlp","mlpe","ksvm","randomForest","mr","cubist","
 pcr", "plsr", "cppls") #rvm model not used because of compute problems (only in the
 hourly cenarios...)

for (data in datah){
 d = read.csv(data, sep = ";", dec = ",")
 L=nrow(d)

d$Weather  = delevels(d$Weather,c("Drizzle","Heavy Rain Showers","Light
 Drizzle","Light Rain","Light Rain Showers","Light Thunderstorms and
 Rain","Thunderstorm","Rain Showers"),"Rain")
d$Weather = delevels(d$Weather,c("Fog","Mostly Cloudy","Overcast","Partly
 Cloudy","Patches of Fog"),"Fog")
d$Weather = delevels(d$Weather,c("Clear","Scattered Clouds"),"Clear")
d$type = delevels(d$type,c("Sports","Sportsing"),"Sports")
 V = ((L-W)/SW)
 V = V - H*2  #isto so permite ir ate  ás iterações possiveis senão começa a reduzir de 7
 para 6,5,4,3,2,1 que corrspondem aos ultimos targets conhecidos
 V = V -1
 dtr=1:(L-H)+1
#testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))

 testests = CasesSeries(d$ts, W = c(1,2,3,4,5,6,7,8,9,10,11,12,13)) #CasesSeries do TS
 testes=cbind(d[1: (L-H)-1, c(2,4,5,6,7,8,9)],testests)#ficam todos os dados ate ao
 numero de series
```

```
Vectormodels = vector("list",V)
for(model in models)
 {

s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto"),m
etric="SAE")
        b = 1
    modelfinal=paste(data,model,";")
   write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
   mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
   c=1
   for(b in 1:V){ # beginning of the rolling windows predictions using rminer package...


        modelfinal=paste(data,model,"rolling window",";")
        write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        HO=holdout(testes$y, ratio=H, mode="rolling", iter=b, window=W,
increment=SW)
        cat("iteration:",b,"\nTarget:",testes$y[HO$ts],"\n")

              M=fit(y~.,testes[HO$tr,],model=model, search = s)

        PredMR=predict(M,testes[HO$ts,])
        cat("PRED:",round(PredMR,digits=0),"\n")

        cat("Model:",model,"MAE:",mmetric(testes$y[HO$ts], PredMR,"MAE"),"\n")

   #mgraph(testes$y[HO$ts],PredMR,graph="REG",main=modelfinal,Grid=10,col=c("
black","blue"),leg=list(pos="topleft",leg=c("target","predictions")))
        mmetric=mmetric(testes$y[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
```

```
 "RRSE", "R22","MAPE"))
            print(mmetric)
            write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
 append=TRUE, col.names=FALSE, row.names=FALSE)
            write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)
            name = paste(data, model ,b, 'rollingwindow.png', sep="-")


     #ver comentario no fundo da pagina para uma iteração interessante!!
            Vectormodels[[b]]=list(target=testes$y[HO$ts],pred=PredMR,model = model
 , cenario = data)
            write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
 append=TRUE, col.names=FALSE, row.names=FALSE)
            write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)



}


     file =paste(data,model,"DMhour", sep="-")
 save(Vectormodels,file=file)
}

 }}  else {

for(data in dataD){
d = read.csv(data, sep = ";")
L=nrow(d)

d$Weather = delevels(d$Weather,c("Fog-Rain","Rain"),"Rain")
```

```
V = ((L-W)/SW)
V = V - H*2 #isto so permite ir ate  ás iterações possiveis senão começa a reduzir de 7 para 6,5,4,3,2,1 que corrspondem aos ultimos targets conhecidos
V = V -1
dtr=1:(L-H)
#testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))

testests = CasesSeries(d$ts, W = c(1,2,3,4,5,6,7,8)) #CasesSeries do TS
testes=cbind(d[1:(L-H)-1, c(3:10)],testests)#ficam todos os dados ate ao numero de series
#print(testes)
Vectormodels = vector("list",V)
for(model in models)
 {

s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto"),metric="SAE")
        b = 1
    modelfinal=paste(data,model,";")
   write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE, col.names=FALSE, row.names=FALSE)
   mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
   c=1
   for(b in 1:V){ # beginning of the rolling windows predictions using rminer package...


        modelfinal=paste(data,model,"rolling window",";")
        write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE, col.names=FALSE, row.names=FALSE)
        HO=holdout(testes$y, ratio=H, mode="rolling", iter=b, window=W, increment=SW)
```

```
cat("iteration:",b,"\nTarget:",testes$y[HO$ts],"\n")

        M=fit(y~.,testes[HO$tr,],model=model,search=s) #o erro do KSVM
Ocorre aqui no fit!

PredMR=predict(M,testes[HO$ts,])
cat("PRED:",round(PredMR,digits=0),"\n")

cat("Model:",model,"MAE:",mmetric(testes$y[HO$ts], PredMR,"MAE"),"\n")

  #mgraph(testes$y[HO$ts],PredMR,graph="REG",main=modelfinal,Grid=10,col=c("
black","blue"),leg=list(pos="topleft",leg=c("target","predictions")))
mmetric=mmetric(testes$y[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
"RRSE", "R22","MAPE"))
print(mmetric)
write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
name = paste(data, model ,b, 'rollingwindow.png', sep="-")


  #ver comentario no fundo da pagina para uma iteração interessante!!
Vectormodels[[b]]=list(target=testes$y[HO$ts],pred=PredMR,model = model
, cenario = data)
write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
```

```
}
 file =paste(data,model,"DMday", sep="-")
 save(Vectormodels,file=file)

}









}
}
}

vectorgraph=function( H = 7, metrics = "MAPE",  graphs = "REG"){


 if(H == 7){
      for(nametoload in vectordaily){
      load(nametoload)
  v = Vectormodels
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
  htarg[i,]=v[[i]]$target
  hpred[i,]=v[[i]]$pred
}
```

```
# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
     name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"), leg=list(pos="topright",leg
 = c("Targets", "Predictions")),main = "Target vs Prediction")
 dev.off()
 write.table(nametoload,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)

 }
}else{
 for(nametoload in vectorHourly){
 load(nametoload)
  v = Vectormodels
 N=length(v)-4 #this is because the hourly model generates 4 null pointers at the end...

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
```

```
{
 htarg[i,]=v[[i]]$target
 hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
 hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
 print(hmae)
    name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"), leg=list(pos="topright",leg
 = c("Targets", "Predictions")),main = "Target vs Prediction")
 dev.off()

    write.table(nametoload,"MetricsHours.csv",sep=";",eol="\n", dec =
 ",",append=TRUE, col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsHours.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)
 }
 }
}
```

A.4) Regression Modeling code

```
#phase 3 modeling with Data Mining models
```

```
 #aqui tenho que fazer os cases series seperados e depois juntar e fazer o cases series do
 resultado junto!!
library(forecast)
library(lubridate)
library(rminer)
femaleWeatherdaily = read.csv("femaleWeatherdaily.csv",sep=";")
maleWeatherdaily = read.csv("maleWeatherdaily.csv",sep=";")
femaleWeatherhourly = read.csv("femaleWeatherhourly.csv",sep=";")
maleWeatherhourly = read.csv("maleWeatherhourly.csv",sep=";")
allWeatherdaily = read.csv("Allweatherdaily.csv",sep=";",dec = ",")
allWeatherhourly = read.csv("AllWeatherHourly.csv",sep=";",dec = ",")
#modeling phase!
#femaleWeatherhourly.csv
dataD=c("femaleWeatherdaily.csv","maleWeatherdaily.csv","Allweatherdaily.csv")
datah=c("AllWeatherHourly.csv","femaleWeatherhourly.csv","maleWeatherhourly.csv")
models=c("randomForest","rpart","mlpe","mr","ksvm")
modelsh=c("mr","rpart","mlpe")
cenariostotais=c("Todosdia","TodosHora","Homemdia","Mulherdia","Homemhora","Mu
 lherHora")
directory = "/Users/user/Desktop/Models DM Phase 3"
#predict
checkcenariodia= function(st) {
 if(st=="Todosdia") st = allWeatherdaily$ts
 if (st=="Homemdia") st= maleWeatherdaily$ts
 if (st=="Mulherdia") st= femaleWeatherdaily$ts


return(st);
}
checkcenariohora= function(st) {
 if(st=="TodosHora") st = allWeatherhourly$ts
 if (st=="Homemhora") st= maleWeatherhourly$ts
 if (st=="MulherHora") st= femaleWeatherhourly$ts
```

```
return(st);
}


vectordaily=c("femaleWeatherdaily.csv-rpart-DMDay",
"femaleWeatherdaily.csv-mlpe-DMDay",
"femaleWeatherdaily.csv-randomForest-DMDay",
"femaleWeatherdaily.csv-mr-DMDay",
"maleWeatherdaily.csv-rpart-DMDay",
"maleWeatherdaily.csv-mlpe-DMDay",
"maleWeatherdaily.csv-randomForest-DMDay",
"maleWeatherdaily.csv-mr-DMDay",
"Allweatherdaily.csv-rpart-DMDay",
"Allweatherdaily.csv-mlpe-DMDay",
"Allweatherdaily.csv-randomForest-DMDay",
"Allweatherdaily.csv-mr-DMDay",
"femaleWeatherdaily.csv-ksvm-DMday",
"Allweatherdaily.csv-ksvm-DMday",
"maleWeatherdaily.csv-ksvm-DMday")
vectorHourly = c("femaleWeatherhourly.csv-ksvm-DMHour","AllWeatherHourly.csv-
 ksvm-DMHour","maleWeatherhourly.csv-ksvm-DMHour","femaleWeatherhourly.csv-
 rpart-DMHour","femaleWeatherhourly.csv-mlpe-DMHour","femaleWeatherhourly.csv-
 mr-DMHour","maleWeatherhourly.csv-rpart-DMHour","maleWeatherhourly.csv-mlpe-
 DMHour","maleWeatherhourly.csv-mr-DMHour","AllWeatherHourly.csv-rpart-
 DMHour","AllWeatherHourly.csv-mlpe-DMHour","AllWeatherHourly.csv-mr-
 DMHour","maleWeatherhourly.csv-randomForest-
 DMHour","femaleWeatherhourly.csv-randomForest-DMHour","AllWeatherHourly.csv-
 randomForest-DMHour")


 modeling=function(frequencytime = "day" , H = 7, W = 200,SW=1){
```

```
if(frequencytime == "hour"){
 #models=c("naive","rpart","kknn","mlp","mlpe","ksvm","randomForest","mr","cubist","
 pcr", "plsr", "cppls") #rvm model not used because of compute problems (only in the
 hourly cenarios...)

for (data in datah){
 d = read.csv(data, sep = ";", dec = ",")
 L=nrow(d)

d$Weather  = delevels(d$Weather,c("Drizzle","Heavy Rain Showers","Light
 Drizzle","Light Rain","Light Rain Showers","Light Thunderstorms and
 Rain","Thunderstorm","Rain Showers"),"Rain")
d$Weather = delevels(d$Weather,c("Fog","Mostly Cloudy","Overcast","Partly
 Cloudy","Patches of Fog"),"Fog")
d$Weather = delevels(d$Weather,c("Clear","Scattered Clouds"),"Clear")
d$type = delevels(d$type,c("Sports","Sportsing"),"Sports")
 V = ((L-W)/SW)
 V = V - H*2  #isto so permite ir ate  ás iterações possiveis senão começa a reduzir de 7
 para 6,5,4,3,2,1 que corrspondem aos ultimos targets conhecidos
 V = V -1
 dtr=1:(L-H)+1
#testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))


 testes=d[c(2,3,4,5,6,7,8,9)]#ficam todos os dados ate ao numero de series

 Vectormodels = vector("list",V)
 for(model in models)
  {

 s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto"),m
```

```
etric="SAE")
        b = 1
    modelfinal=paste(data,model,";")
    write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
    mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
    c=1
    for(b in 1:V){ # beginning of the rolling windows predictions using rminer package...


        modelfinal=paste(data,model,"rolling window",";")
        write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        HO=holdout(testes$ts, ratio=H, mode="rolling", iter=b, window=W,
increment=SW)
        cat("iteration:",b,"\nTarget:",testes$ts[HO$ts],"\n")

            M=fit(ts~.,testes[HO$tr,],model=model, search = s)


        PredMR=predict(M,testes[HO$ts,])
        cat("PRED:",round(PredMR,digits=0),"\n")

        cat("Model:",model,"MAE:",mmetric(testes$ts[HO$ts], PredMR,"MAE"),"\n")


    #mgraph(testes$y[HO$ts],PredMR,graph="REG",main=modelfinal,Grid=10,col=c("
black","blue"),leg=list(pos="topleft",leg=c("target","predictions")))
        mmetric=mmetric(testes$ts[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
"RRSE", "R22","MAPE"))
        print(mmetric)
        write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
```

```
        name = paste(data, model ,b, 'rollingwindow.png', sep="-")


    #ver comentario no fundo da pagina para uma iteração interessante!!
        Vectormodels[[b]]=list(target=testes$ts[HO$ts],pred=PredMR,model = model
, cenario = data)
        write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
 append=TRUE, col.names=FALSE, row.names=FALSE)
        write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
 col.names=FALSE, row.names=FALSE)



}


    file =paste(data,model,"DMhour", sep="-")
 save(Vectormodels,file=file)
}

 }}  else {

for(data in dataD){
d = read.csv(data, sep = ";")
L=nrow(d)

d$Weather = delevels(d$Weather,c("Fog-Rain","Rain"),"Rain")


 V = ((L-W)/SW)
 V = V - H*2 #isto so permite ir ate  ás iterações possiveis senão começa a reduzir de 7
 para 6,5,4,3,2,1 que corrspondem aos ultimos targets conhecidos
 V = V -1
```

```
dtr=1:(L-H)
#testes = CasesSeries(checkcenariodia(cenario), W = c(1,2,3,4,5,6,7,8))


testes=d[c(2:10)]#ficam todos os dados ate ao numero de series
#print(testes)
Vectormodels = vector("list",V)
for(model in models)
 {

s=list(search=mparheuristic(model,n=10),smethod="grid",method=c("holdouto"),m
etric="SAE")
        b = 1
    modelfinal=paste(data,model,";")
   write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
   mmetrics= paste("MAE", "RAE", "RMSE", "RRSE", "R22", colapse=" ")
   c=1
   for(b in 1:V){ # beginning of the rolling windows predictions using rminer package...


        modelfinal=paste(data,model,"rolling window",";")
        write.table(modelfinal,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        HO=holdout(testes$ts, ratio=H, mode="rolling", iter=b, window=W,
increment=SW)
        cat("iteration:",b,"\nTarget:",testes$ts[HO$ts],"\n")

            M=fit(ts~.,testes[HO$tr,],model=model,search=s) #o erro do KSVM
Ocorre aqui no fit!

        PredMR=predict(M,testes[HO$ts,])
        cat("PRED:",round(PredMR,digits=0),"\n")
```

```
        cat("Model:",model,"MAE:",mmetric(testes$ts[HO$ts], PredMR,"MAE"),"\n")

    #mgraph(testes$y[HO$ts],PredMR,graph="REG",main=modelfinal,Grid=10,col=c("
black","blue"),leg=list(pos="topleft",leg=c("target","predictions")))
        mmetric=mmetric(testes$ts[HO$ts],PredMR,metric=c("MAE", "RAE", "RMSE",
"RRSE", "R22","MAPE"))
        print(mmetric)
        write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)
        name = paste(data, model ,b, 'rollingwindow.png', sep="-")


    #ver comentario no fundo da pagina para uma iteração interessante!!
        Vectormodels[[b]]=list(target=testes$ts[HO$ts],pred=PredMR,model = model
, cenario = data)
        write.table(mmetrics,"metricsdataming.txt",sep="\n",eol="\n",
append=TRUE, col.names=FALSE, row.names=FALSE)
        write.table(mmetric,"metricsdataming.txt",sep="\n",eol="\n", append=TRUE,
col.names=FALSE, row.names=FALSE)




}
 file =paste(data,model,"DMday", sep="-")
 save(Vectormodels,file=file)

}
```

```
}
}
}

vectorgraph=function( H = 7, metrics = "MAPE",  graphs = "REG"){


 if(H == 7){
      for(nametoload in vectordaily){
      load(nametoload)
  v = Vectormodels
 N=length(v)

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
  htarg[i,]=v[[i]]$target
  hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"

hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)
```

```
 }
  print(hmae)
     name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"), leg=list(pos="topright",leg
 = c("Targets", "Predictions")),main = name)
 dev.off()
 write.table(nametoload,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsDays.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)

 }
}else{
 for(nametoload in vectorHourly){
 load(nametoload)
  v = Vectormodels
 N=length(v) #this is because the hourly model generates 4 null pointers at the end...

htarg=matrix(nrow=N,ncol=H)
hpred=matrix(nrow=N,ncol=H)

# convert vector into htarg and hpred
for(i in 1:N)
{
  htarg[i,]=v[[i]]$target
  hpred[i,]=v[[i]]$pred
}

# for a metric, say “MAE"
```

```
hmae=vector(length=H)
for(j in 1:H){
  hmae[j]=mmetric(htarg[,j],hpred[,j],metric=metrics)

 }
  print(hmae)
     name = paste(nametoload,"Graph.png", sep="-")
fileName= paste(name, sep="/")
png(filename=fileName)

 mgraph(htarg,hpred,graph = graphs ,col=c("black","blue"), leg=list(pos="topright",leg
 = c("Targets", "Predictions")),main = "Target vs Prediction")
 dev.off()

     write.table(nametoload,"MetricsHours.csv",sep=";",eol="\n", dec =
 ",",append=TRUE, col.names=FALSE, row.names=FALSE)
 write.table(hmae,"MetricsHours.csv",sep=";",eol="\n", dec = ",",append=TRUE,
 col.names=FALSE, row.names=FALSE)




 }



 }
}
```

A.5) Weather collection code

```
 library(RCurl)
library(XML)
i = 1
```

```
data = read.csv("TimeWeather.csv",sep=";")
df = data.frame(event_date = NA,events=NA,temperature=NA,conditions=NA,
 Humidity = NA)
write.table(df,"Weather.csv",sep=";",row.names=FALSE)
while(i <= nrow(data)){
 datas = data$time[i]
 linkp1="http://www.wunderground.com/history/airport/LPPT/"
 linkp2="/DailyHistory.html?&reqdb.zip=&reqdb.magic=&reqdb.wmo=&MR=1"
 url = paste(linkp1,datas,linkp2, sep="")
 tables <- readHTMLTable(url)
 n.rows <- unlist(lapply(tables, function(t) dim(t)[1]))
 j = 1
 while(j <= nrow(tables$obsTable)){
 date = data$time[i]
 time = tables$obsTable$Time[j]
 switch(time,
"12:00 AM"={

 time ="00:00"

},
"12:30 AM"={

time ="00:30"
},
"1:00 AM"={

time ="01:00"
},
"1:30 AM"={

time ="01:30"
},
```

"2:00 AM"={

time ="02:00"
},
"2:30 AM"={

time ="02:30"
},
"3:00 AM"={
time ="03:00"
},
"3:30 AM"={
time ="03:30"
},
"4:00 AM"={
time ="04:00"
},
"4:30 AM"={
time ="04:30"
},
"5:00 AM"={
time ="05:00"
},
"5:30 AM"={
time ="05:30"
},
"6:00 AM"={
time ="06:00"
},
"6:30 AM"={
time ="06:30"
},
"7:00 AM"={

time ="07:00"

},

"7:30 AM"={

time ="07:30"

},

"8:00 AM"={

time ="08:00"

},

"8:30 AM"={

time ="08:30"

},

"9:00 AM"={

time ="09:00"

},

"9:30 AM"={

time ="09:30"

},

"10:00 AM"={

time ="10:00"

},

"10:30 AM"={

time ="10:30"

},

"11:00 AM"={

time ="11:00"

},

"11:30 AM"={

time ="11:30"

},

"12:00 PM"={

time ="12:00"

},
"12:30 PM"={

time ="12:30"
},
"1:00 PM"={

time ="13:00"
},
"1:30 PM"={

time ="13:30"
},
"2:00 PM"={

time ="14:00"
},
"2:30 PM"={

time ="14:30"
},
"3:00 PM"={
time ="15:00"
},
"3:30 PM"={
time ="15:30"
},
"4:00 PM"={
time ="16:00"
},
"4:30 PM"={
time ="16:30"
},

"5:00 PM"={

time ="17:00"

},

"5:30 PM"={

time ="17:30"

},

"6:00 PM"={

time ="18:00"

},

"6:30 PM"={

time ="18:30"

},

"7:00 PM"={

time ="19:00"

},

"7:30 PM"={

time ="19:30"

},

"8:00 PM"={

time ="20:00"

},

"8:30 PM"={

time ="20:30"

},

"9:00 PM"={

time ="21:00"

},

"9:30 PM"={

time ="21:30"

},

"10:00 PM"={

time ="22:00"

},

```
"10:30 PM"={
time ="22:30"
},
"11:00 PM"={
time="23:00"
},
"11:30 PM"={
time="23:30"
},
{
  time
}
)
 df$event_date =paste(date,time,sep=" ")
 #df$time = tables$obsTable$Time[j]
 df$events = tables$obsTable$Events[j]
 df$temperature=tables$obsTable$Temp.[j]
 df$conditions=tables$obsTable$Conditions[j]
 df$Humidity= tables$obsTable$Humidity[j]
 write.table(df,"Weather.csv",sep=";", append=TRUE , row.names=FALSE,
 col.names=FALSE)

 if(tables$obsTable$Time =="11:30 PM" && j == nrow(tables$obsTable)){

    break
    }else{
    j = j+1
    }}
    i=i+1
}
```