USE OF KERNEL HILBERT SPACES FOR ONLINE DATA ANALYSIS IN MARKET RESEARCH SYSTEMS

DAN SERBANOIU

Master Degree in Smart Telecoms and Sensing Networks

31 August 2021

ASTON UNIVERSITY

Co-funded by the
Erasmus+ Programme
of the European Union

Erasmus Mundus Joint Master's Degree
"SMART Telecom and Sensing NETworks" (SMARTNET) (2019/2021 intake)
Aston University, Triangle, B4 7ET / Birmingham, UK
Email: aipt_smartnet@aston.ac.uk / Web-site: smartnet.astonphotonics.uk/

Acknowledgement

This Master Thesis has been accomplished in the framework of the European Funded Project: **SMART Telecom and Sensing Networks (SMARTNET)** - Erasmus+ Programme Key Action 1: Erasmus Mundus Joint Master Degrees – Ref. Number 2017 – 2734/001 – 001, Project number - 586686-EPP-1-2017-1-UK-EPPKA1-JMD-MOB, coordinated by **Aston University,** and with the participation of **Télécom SudParis, member of IP Paris** and **National and Kapodistrian University of Athens.**

Aston University
BIRMINGHAM UK

TELECOM SudParis
TELECOM ParisTech

National and Kapodistrian
UNIVERSITY OF ATHENS

ASTON UNIVERSITY
Use of Kernel Hilbert Spaces for online data analysis in market research systems

Dan Serbanoiu
Master Degree in Smart Telecoms and Sensing Networks
31 August 2021

Summary

The aim of this project is to develop a market research system and show how to employ big data analytics and online methods to generate affordable and high-quality data-driven marketing insights using non stationary data sets. Other such applications exist but they solve very specific business problems or require field specific knowledge. The proposed system aims at removing all the guesswork and provide automatically an array of high quality marketing information.

Keywords: Marketing Research, Big Data, Machine Learning, Kernel Machines

I lovingly dedicate my dissertation work to my family and to the friends that have helped and supported me during the difficult times that ensued as a result of the COVID-19 pandemic.

Acknowledgments

I would like to extend my heartiest thanks to all those who helped and guided me during my training period. I would like to thank my supervisor, professor Stylianos Sygletos. His guidance and help have been very helpful in putting the project on the right track and resolve some issues that occurred along the way. I would like to give a special thanks to Kety Mayelin Jimenez Tejeda and her family for their hospitality and support. A special thanks goes also to Angela Onaney Sanchez De La Cruz for the beautiful moments we shared together. Last but not least, I would like to thank my family for their presence and unconditional love.

Contents

List of Figures

Abbreviations

Market Research System (MRS)
Small and Medium-Size Enterprise (SME)
Kernel recursive least-squares regression (KRLS)
Mean Squared Error (MSE)
Continuous Integration and Continuous Delivery (CI/CD)

1 Introduction

The purpose of the current chapter is to describe the general characteristics of an AI enabled market research system, detail a list of aims and objectives, describe the adopted life cycle model, present a risk analysis and offer a project plan.

1.1 Problem Statement

Marketing research is the systematic design, collection, analysis, and reporting of data related to the market of a company. When doing market research, most SMEs don't know what to search for, struggle to find appropriate sources or lack the digital tools to properly analyse and make use of information. Nonetheless such data is vital for the success and functionality of businesses. For example, a company selling phones needs to see if the demand for mobile phones is growing or shrinking and know which accessories and phones are more popular and why. A physical restaurant should have a detailed knowledge of the dining trends of the food industry in their area. By interpreting historical data and deriving financial metrics from it such as pricing changes, sales growth, demand etc. SMEs can better understand the market they operate in, improve their competitiveness and make better decisions. Usually, large companies have their own complex systems for recording data, store it upon well-structured models, analyse it and generate useful reports. Such applications, however, are too expensive to build for SMEs and require lots of knowledge and data for their functionality to begin with [14]. The purpose of this thesis is to build a MRS that can aggregate market-related data from a variety of data sources (documents, logs, APIs, websites, online surveys etc.) and analyses it to generate useful information and data-driven marketing insights. Other MRSs already exist, such as QuantCloud, a trading service that uses predictive algorithms to find patterns within stock market data and offer stock market insights [19]. QuantCloud can calculate the seasonal volatility of log returns for grouped data, the Autoregressive-moving average (ARMA) etc. Another MRS is Google Analytics which tracks websites usage and provides businesses with prod-

uct trends, users behaviour information, custom reports etc. [6]. The mentioned systems all focus on very specific business problems. This thesis aims at building a general purpose MRS that can offer an array of different metrics and that does not require specialized knowledge to be used. Marketing data usually display non-stationary behaviour, which means that the data points have dynamic means, variances, and covariances and they are affected by a number of implicit variables. Because of this non stationary data cannot be modelled well or forecasted accurately. This thesis wants to show how to analyse such data using kernel machines and unsupervised learning.

1.2 Aims and Objectives

The main goal of this project is to develop a prototype of a Web-based MRS that leverages AI, ML and Big data technologies and that integrates and analyses data coming from different sources in order to produce marketing insights. In order to implement such a system, the following objectives have been set:

- Implement a Search Module;
- Create an Aggregator module that fetches data from various sources and integrates it on a common model;
- Build an Analyser tool that performs data analysis using k-means and KRLS methods;
- Build a data visualization dashboard;
- Configure a database to store, process and query existing market research data.

1.3 Life-cycle model

For the implementation of the marketing research system the incremental model was used. The incremental model includes phases for tracing the progress of a product from a planned idea to its final release into operation and maintenance. The main characteristics of this life cycle approach are:

1. Architectural Analysis and Design activities are not repeated: the goal is to spend sufficient time studying the problem and gathering all the requirements so that the architecture of the system is can be fully identified during the initial stages of the project. This allows for a stronger planning;
2. The implementation of the different parts of the system is incremental and is also planned in the initial stages of the project;
3. The activities of Design, Coding and Verification are repeated in order to improve existing parts of the system, correct mistakes or add new features;
4. Maintenance is a continuous activity and aims at making the product solid and bug free.

The advantages of using this model are that the requirements are treated according to their strategic importance and the implementation starts from the primary ones. Furthermore, each increment adds new functionality to the baseline and allows any shareholder to test the application and give valuable feedback. Every successive iteration of the application should become more complete, converging to an appropriate final version of the product.

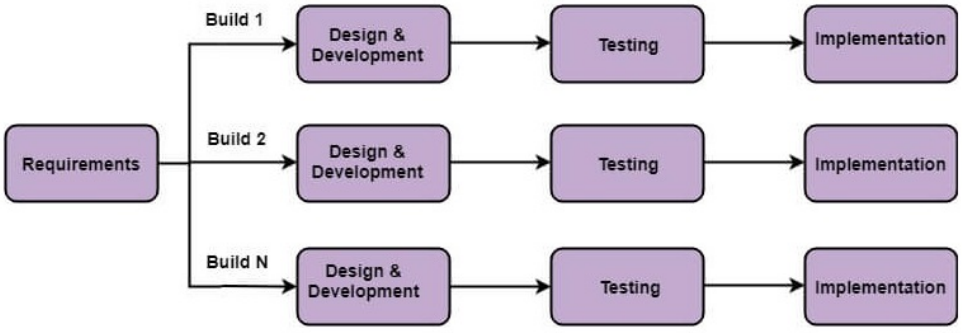

Figure 1.1: Workflow of the incremental model, where a product is built incrementally.

1.4 Risks Analysis

Every project is subject to risks. In order the manage them, the potential risks have been identified, the possible impact and consequences of each risk have been studied and preventive and corrective methods have been established:

- Technological risks: depending on type of project, some tools might be better suited than others. Choosing the wrong tools is not particularly dangerous, but it can lead to wasting time, missing deadlines and having implementation and integration issues. In order to avoid this risk a longer period of time was allocated at the beginning to compare different tools according to project needs;
- Hardware failures: all the work is done using a laptop. Any problems with this device might have repercussions on the project. This is not very dangerous but it can lead to wasting time, missing deadlines and losing work. In order to account for this risk a second device is made available and the work is backed up regularly on remote repositories;
- Poor planning: this refers to a situation where the tasks to complete are not set out or are under or over estimated. This can be caused by a misunderstanding of the requirements or a deviation from the scope of the project. This can be dangerous and it can lead to wasting time, missing deadlines, unmet expectations and to the failure of the project. To account for this a longer period of time was allocated for the requirements analysis and the literature Review and the CI/CD method was used;
- Misunderstanding the requirements: sometimes business and technical professionals will not understand each other's terminology and may have a different meaning for the same words. This can lead to misunderstanding the requirements. The danger of this happening is medium and it can lead to building the wrong product, wasting time and resources and safety issues. To avoid this risk longer time was spent in requirements analysis and the CI/CD method was used;
- Conflicts with stakeholders: sometimes conflicts might arise between the different parties involved in a project. This can be extremely dangerous and it can result in the failure of the project. To avoid this it is necessary to decide all the important things at the beginning and make sure everyone is on the same page.

1.5 Planning

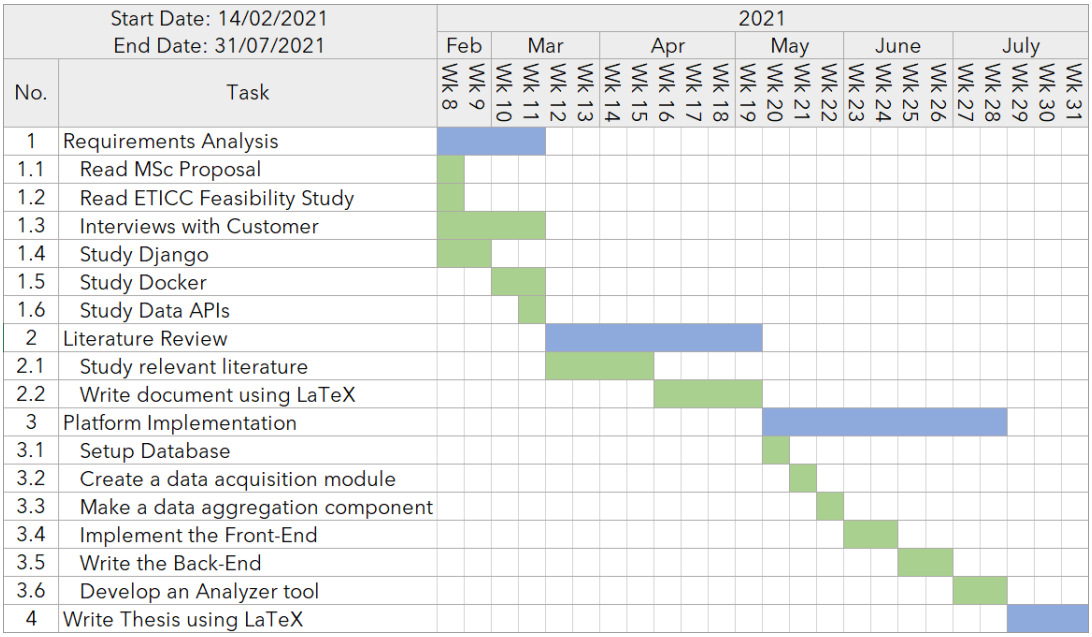

Figure 1.2: The Gantt chart shows the activities involved in a project.

The activities and their subtasks have been scheduled using a Gantt chart, which is a useful tool for seeing the various dependencies between activities and sub-activities. In a Gantt chart is possible to represent, through black rhombuses, milestones that coincide with the end of a period or objectives to be achieved. The project started as part of an internship within a company located in Birmingham, but due to an unresolvable dispute over IP, the project was ultimately submitted under a different supervision.

2 Literature Review

The purpose of the current chapter is to provide knowledge of the subject areas relevant to the project including current developments, controversies, breakthrough, previous research and relevant background theory.

2.1 Market Research Systems

MRSs use big data analytics and online methods to produce high quality marketing information that medium sized companies can use to understand their market and be more competitive [14]. Marketing research is useful in the area of product planning and development such as when evaluating the need for a new product or its positioning in the market. It has applications in the area of advertising and copy testing or to identify existing and potential distribution channels. It can help to deduce the pricing expectations of consumers and their reactions and responses to different price levels of products. Marketing research can identify the forces operating in a market and assess the market trends, the size of present and potential markets of the company, the evolution of the impact of government legislation, policies, and schemes on the performances of marketing operations of the company. It can also study the sales potential of the company's products and the evolution of the company's sales performance. Finally, it can assess the environmental fitness of the firm. [1]. A study conducted by the Alibaba Group in China has also shown a way of calculating the impact of brands on society, by combining the impact brands have on the media, the government Impact and the personal impact. [9] The MRS can obtain all these analytics by integrating several data sources and where necessary apply AI and Machine learning to deduce useful results [16]. Platforms like QuantCloud, a trading service [19], or Google Analytics, a websites activity tracking platform [6], all offer insights within specific areas and require specialized knowledge to be used. The goal is to build a prototype of a MRS that generates a variety of marketing information automatically and that does not require specialized knowledge.

2.2 AI, ML and Big Data

Big data is a term used to describe the huge amount of data that is being generated by organizations in today's business environment. Such data usually comes from three primary sources: social data such as likes, tweets, comments, video uploads, queries in search engines etc., machine data generated by sensors installed in industrial equipment and transaction data. Machine learning is a branch of artificial intelligence that provides tools and techniques for working on all this data to produce useful results. It is based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Machine learning algorithms are classified into supervised learning, unsupervised learning and reinforcement learning [16]:

2.2.1 Supervised learning

Supervised learning is the task of training a function with labelled training data in order for this function to be able to learn a general rule and map any input to its appropriate output. Machine learning models are usually very good at performing regression and classification:

- Statistical modelling techniques are used to find a parametric function that best fits a set of observations and train it appropriately so that is can predict new observations. Statistical models (i.e. logistic models, polynomials, linear regression models) are very good at modelling and predicting trends;
- Neural networks are powerful tools used to perform regression and classification of non linearly separable data. For example, convolutional neural networks can be used to classify consumers according to their preferences so that they can be targeted by specific ads or marketing strategies [3]. Neural Networks, paired with Locality sensitive hashing [7] can be used to improve de-duplication, that is to identify chunks that are already stored in a large database;
- Kernel Machines can perform accurate regression and classification by mapping the data feature space into an equivalent higher order representation where the problem becomes linearly separable. They are very good at analysing non stationary data such as marketing trends [15];
- Support Vector Machines are excellent regression and classification algorithms that

work by separating the data set with an hyperplane and maximizing the margin between the closest points. They can be employed to classify social media data such as tweets, posts, comments etc. according to the emotions they emit (positive, negative and neutral). These results can aid in understanding if people have a good or bad opinion about something and take action accordingly [13].

2.2.2 Unsupervised learning

Unsupervised learning is a machine learning technique in which a model works on its own to discover hidden patterns and separate unlabelled information into appropriate clusters, if they exist. It is useful for finding associations between different parameters in the available data, for example people that buy X might also buy Y. It can also be used to find the anomalous elements in a data set [2]. There exist several clustering methods:

- Hierarchical clustering is a naive clustering method that builds a hierarchy of clusters by iteratively joining together the closest elements;
- K-means is a set of unsupervised clustering algorithms that can divide data points into k clusters [2];
- K-Nearest Neighbors is an algorithm that finds the k nearest points to the point of interest.

2.2.3 Reinforcement learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment [12].

- Monte Carlo methods are a way of obtaining estimates when working with uncertain phenomena. They use randomness to obtain meaningful information and are effective for calculating business risks and predicting costs or scheduling overruns;
- For any finite Markov decision process, Q-learning algorithms are able to find an optimal path by maximizing the expected value of the total reward over any and all successive steps, starting from a specific state. These algorithms can be used to make decisions in business environments, for example in order to determine whether to hold, buy, or sell at any point in time.

2.3 ML workflow

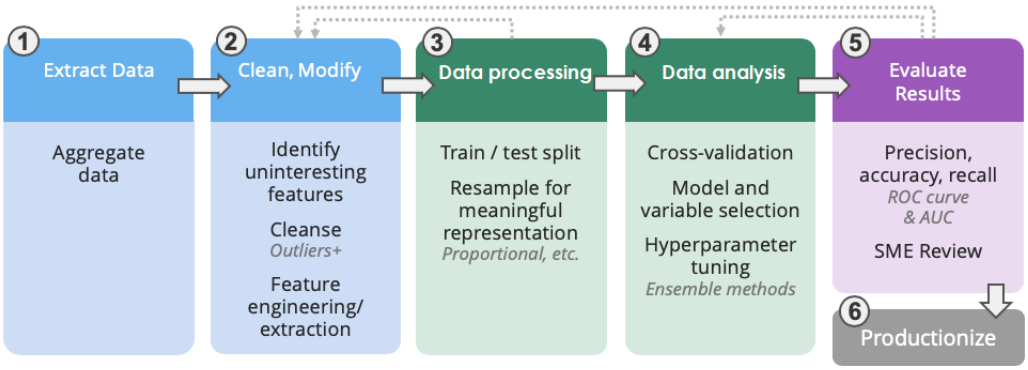

Figure 2.1: A machine learning task includes data gathering/extraction, data cleaning, data processing, data analysis, training and testing and finally the evaluation phase.

A machine learning task typically involves collecting, transforming, cleaning, and modelling data with the goal of discovering new information [11].

2.3.1 Data aggregation

Data aggregation is the process of gathering data and presenting it in a summarized format. The data may be gathered from multiple local or remote data sources with the intent of combining these data sources into a common model that could be virtual. Time aggregation collects data points for a single resource over a specific time frame, while spatial aggregation gathers a time series from a group of resources. It is possible to store incoming data into a Data warehouse, which is a large centralized repository of data.

2.3.2 Data cleaning

Before data can be fed to a training algorithm or used for numerical analysis, it is important to prepare it by removing or changing rows that are incorrect, incomplete, irrelevant, duplicated, or improperly formatted. In some cases it is necessary to map data to another model. Skipping this step is not recommended because it can lead to runtime errors, inaccurate results and false conclusions [19].

2.3.3 Data processing

The data is divided into 3 three different sets, the training set, the validation set and the test set. During this phase some dimensions of the data might be removed, merged together or reshaped depending on the kind of task to be performed.

2.3.4 Data analysis

Predictive analytics, artificial intelligence (AI) or machine learning algorithms can be applied to the collected data for new insights. Depending on the task at hand (regression or classification) an array of ML models and sets of optimal hyper parameters are chosen for training.

2.3.5 Evaluation

The chosen models are trained with the training data and their precision, accuracy and recall are compared in order to find the best model and understand how well the chosen model can work in the future.

2.3.6 Deploying to production

Only the most accurate models are deployed to production. Marketing data display non stationary behaviour and cannot be accurately modelled by statistical models. On the other hand kernel machines are able to model well data that displays dynamic means, variances, and covariances.

2.4 Kernel recursive least-squares regression methods

Marketing data is non-stationary which means that it is affected by several implicit variables; some data points might belong to different distributions or they might be affected by seasonal patterns and other external influences. Because of this it is very difficult to model marketing data and perform accurate forecasting. Kernel machines offer a way of modelling such unpredictable data sets. Kernel methods use the kernel trick and adaptive filters such as the Recursive Least Squares (RLS) method to update sample-by-sample a regression model [15]. A linear regression model $f(x) = w^{\mathsf{T}}x$ with $x, w \in \mathbb{R}^{\mathsf{D}}$ and $y \in \mathbb{R}$ can be reformulated into a convex optimization problem $f(x) = \sum_{n=1}^{N}[\alpha(n)\kappa(x_n, x)]$ where $\alpha(n) \in \mathbb{R}$ are the expansion coefficients and x_n are the training data or bases. In this reformulation the kernel function $\kappa(x, x^{'}) = \phi(x)^{\mathsf{T}}\phi(x^{'})$ can be interpreted as inner products in a high-dimensional feature space (more formally called Hilbert space), and these can be calculated without having direct knowledge of the higher order feature space. By doing this operation a non linearly separable problem can be brought in a space where it is linearly separable. The least squares minimization $\frac{\partial J(w)}{\partial w} = 0$ yields the coefficients $\hat{w} = \phi^{\mathsf{T}}a$, that can be substituted into $J(w)$ creating the dual representation $J(a)$. The minimization $\frac{\partial J(a)}{\partial a} = 0$ yields the coefficients $\hat{a} = (K + \lambda I_N)^{-1}t$, which can be substituted inside the linear regression model to obtain a kernel-based prediction model:

$$f(x) = w^{\mathsf{T}}\phi(x) = a^{\mathsf{T}}\phi\phi(x) = k(x)^{\mathsf{T}}(K + \lambda I_N)^{-1}t$$

Because the solution to the least-squares problem is expressed as a linear combination of the training data and in terms of the parameter vector w and the kernel function k, it does not matter how complex the feature vector $\phi(x)$ is and it can even have infinite dimensionality if that suits the problem at hand.

When using this method in an online scenario if a new data point is made available it's possible to update the weights of the model recursively by the previous solution calculated with the last $n - 1$ data points. The updated solution $\alpha(n)$ is obtained by recursively applying:

$$\alpha(n) = \begin{bmatrix} \alpha(n-1) - \frac{a_n e_n}{\gamma_n} \\ \frac{e_n}{\gamma} \end{bmatrix}$$

where $e_n = y_n - \hat{y_n}$, $a_n = K_{n-1}^{-1}k_n$ and $\gamma_n = k_{nn} + c - k_n^{\mathsf{T}}a_n$.

The algorithm can be optimized further by using a sliding window that stores the last M data points. When a new data point is made available, before updating the solution, the algorithm adds the new datum to the sliding window and discards the oldest datum or the data point that causes the least error upon being discarded.

Finally it's possible to improve accuracy further by using a Gaussian process model $y = f(x) + r$ which has a zero mean GP prior on f(x) and a Gaussian prior on r. It can be found that the posterior over the latent vector f is $p(f|y) = \mathcal{N}(f|\mu\Sigma)$ which means that the result of this model gives not only the mean value for the predicted solution but also its entire posterior distribution. The recursive updates of the model are: $p(f_n|X_n, y_n) = \mathcal{N}(f_n|\mu_n, \Sigma_n), \mu_n = \begin{bmatrix} \mu_{n-1} \\ \hat{y}_n \end{bmatrix} + \frac{e_n}{\hat{\sigma}^2_{y_n}} \begin{bmatrix} h_n \\ \hat{\sigma}^2_{f_n} \end{bmatrix}, \Sigma_n = \begin{bmatrix} \Sigma_{n-1} & h_n \\ h_n^T & \hat{\sigma}^2_{f_n} \end{bmatrix} - \begin{bmatrix} h_n \\ \hat{\sigma}^2_{f_n} \end{bmatrix} \begin{bmatrix} h_n \\ \hat{\sigma}^2_{f_n} \end{bmatrix}^T$, where $h_n = \Sigma_{n-1} K^{-1}_{n-1} k_n$ and $\hat{\sigma}^2_{f_n}$ and $\hat{\sigma}^2_{y_n}$ are the predictive variances of the latent function and the new output calculated at the new input. The underlying Gaussian process is described at every point by μ_n and Σ_n. The entire KRLS-T algorithm is summarized in algorithm 1. KRLS-T was used in this thesis to implement regression and perform forecasting.

Algorithm 1 KRLS Tracker (KRLS-T) algorithm.

for $i = 1, 2, ...$ **do**
  FORGET: $\mu_{n-1} \leftarrow \sqrt{\lambda}\mu_{n-1}$, $\Sigma_{n-1} \leftarrow \lambda\Sigma + (1 - \lambda)K$.
  Observe input x_n.
  Calculate predictive mean: $\hat{y_n} = k_n K^{-1}_{n-1} \mu_{n-1}$
  Calculate predictive variance: $\hat{\sigma}^2_{yn}$.
  Observe true output: y_n Compute μ_n, Σ_n, K^{-1}_n.
  Add basis x_n to the dictionary.
  if number of bases in the dictionary > M **then**
    Determine the least relevant basis, u_m
    Remove basis u_m from μ_n, Σ_n, K^{-1}_n.
    Remove basis u_m from the dictionary.
  end if
end for

2.5 Tools and Techniques

There are many tools and technologies that have been developed for data extraction, integration, processing and analysis. The following tools and techniques have several features that are useful for Big Data integration and Machine learning tasks.

2.5.1 Aurelia

Aurelia is a client-side JavaScript framework that implements the Model-View-View-Model (MVVM) pattern and supports ES6 and TypeScript. It has a very good performance, an extensive ecosystem, it has a very solid and intuitive routing system and employs reactive binding. In Aurelia the presentation layer is completely separated from the application logic and as a result Aurelia applications are very maintainable, testable and extensible. In this project Aurelia has been used to implement a browser based Front-End of the MRS which allows the user to perform a search and view the results.

Figure 2.2: Aurelia is a JavaScript client framework used to build web, mobile and desktop applications.

2.5.2 Django

Figure 2.3: Django is a Python framework useful for writing scalable web applications

Django is a Python framework [10] and is used to write web applications. It comes with a lot of useful APIs, such as an Object relational mapping API, an User authentication module, a HTTP Session handling module etc. The framework employs the Model Tem-

plate View pattern (MTV), which is a variation of the Model View Controller (MVC). In Django the Model components represent the data and the model classes are mapped automatically to database tables. The View components handle the HTTP requests and return responses. The templates are used to serve the HTML or data for the websites. Django has been used to implement the Back-End of the MRS which performs the data gathering and analysis.

2.5.3 Spark

Figure 2.4: Apache Spark is an analytics engine used for big data processing, with modules for streaming, SQL, machine learning and graph processing.

Spark is a framework developed in 2009 at UC Berkeley [18] that provides primitives for data mining and offers a very good implementation of MapReduce, a programming model for processing and generating big data sets with a parallel, distributed algorithm on a cluster [5]. In Spark every object is a Resilient Distributed Dataset (RDD). The elements of an RDD are not stored in memory but rather every RDDs maintains lineage information for fetching data sets from their source when needed. RDDs can be computed from a HDFS Hadoop Distributed File System (HDFS) or from an existing RDD. In Spark data is split among the different workers by a main driver. Every worker performs its assigned task and then sends the result back to the origin. Spark is up to 100 times faster than Hadoop, because the latter only processes data on disk, while Spark can also load data in memory. As a result, Spark can read 1 TB of data in 4 or 7 seconds. Spark can run on clusters created using many different technologies, such as Apache Mesos, Kubernetes, Amazon EMR etc. and a Spark cluster can scale very easily.

2.5.4 Docker

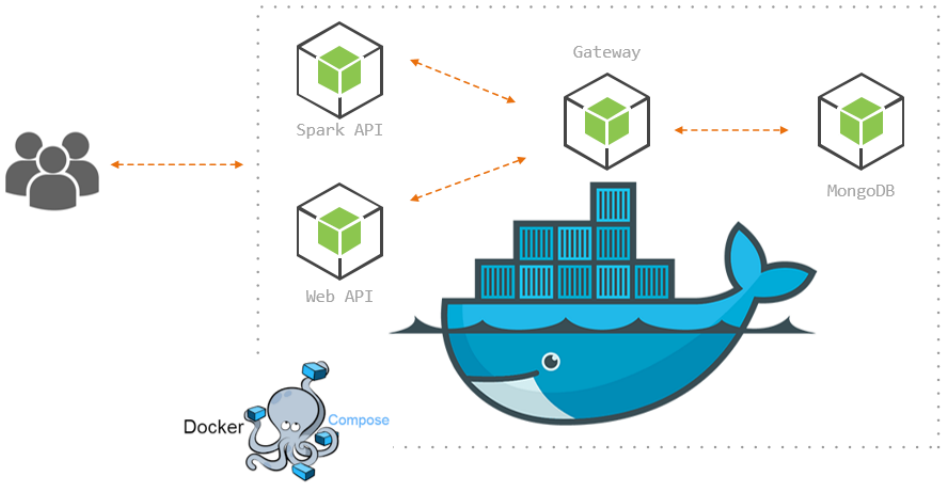

Figure 2.5: Docker is platform that can package applications into self contained containers that are able to run in any environment.

Docker is a platform for packaging and delivering independent software units in portable and self-sufficient containers that have all the dependencies required for the functionality of their applications. Every container is derived from an image, which essentially is a read-only template. An image provides its container with environmental variables, a command line and a filesystem that can store operating systems, packages, libraries and every other dependency a container needs. The main advantage of encapsulating all the dependencies of a service inside a self-sufficient entity it's ensuring that the development platform mimics perfectly the target platform. This makes it very easy to continuously integrate and deploy, increases productivity and encourages a frequent feedback loop. With Docker it's possible to package the different types of nodes of an application [4]. The Front-End and Back-End that implement MRS have been packaged into Docker containers.

2.6 Query optimization through hashing

A very good algorithm for optimizing comparison between items is locality sensitive hashing (LSH). LSH uses the properties of hashing in order to find similarities between big sized entities [7]. LSH can be used to improve query efficiency through an approximate Membership Query (AMQ) scheme. In such a system, the items are hashed with a similar process as LSH, with the only difference being that instead of hashing the signature into buckets, a Bloom filter is used in the last step to map every item into L bits. All the items that have the same L bits set to 1 are determined to be approximate members of the same data set S. AMQ can be used to produce results within a search engine in O(1) time and has demonstrated better accuracy than LSB-trees and other algorithms than run in at least logarithmic time. However, a good configuration is needed in order to minimize false positives and false negatives [8].

3 Implementation

This chapter provides the functional requirements of the application, explains the architecture of the developed research platform and presents some meaningful results and some steps forwards.

3.1 Functional Requirements

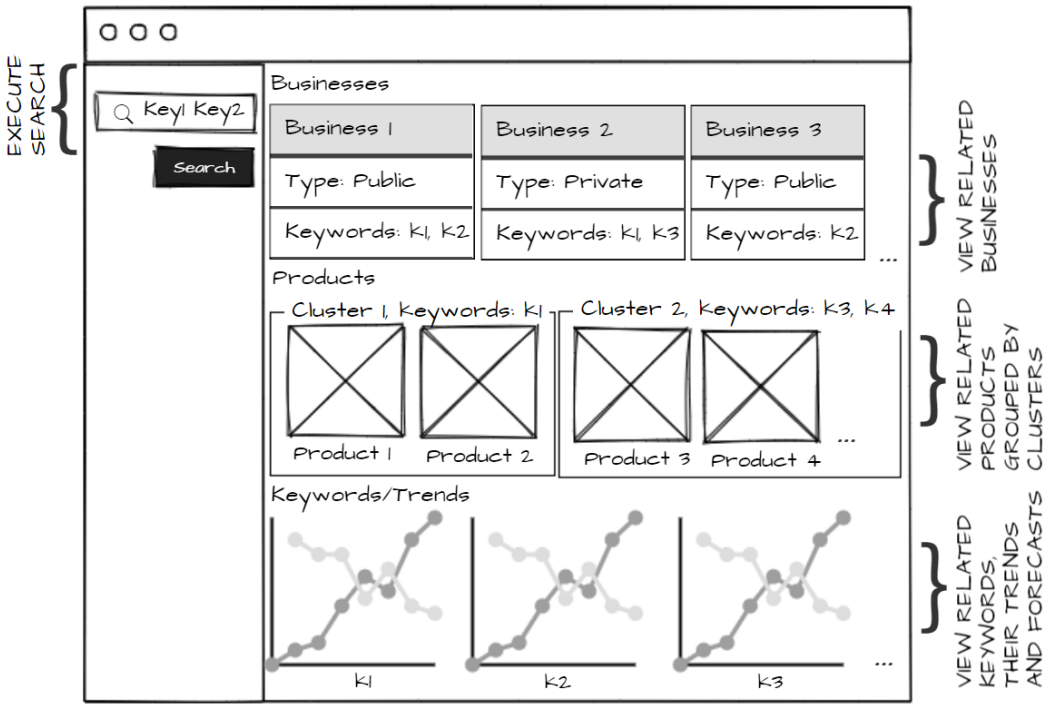

Figure 3.1: This wireframe shows the essential functionality of the implemented MRS.

Functional requirements define the system behaviour and explain how the system responds to inputs. The minimum requirements that have been implemented are shown in figure 3.1. The MRS allows an user to perform a search by proving a list of keywords. When the user presses the search button the browser infers the location and language of the user and makes a request to the back-end which starts gathering and processing the data relevant to the keywords that have been provided. When this operation is successful, the system shows the results to the user in a report. The user can view the keywords that are related to the ones he provided, he can see the related businesses and the related products. The system is able to calculate a list of clusters for the products. The user can see the trends associated with the found keywords and for each trends view the forecasts.

3.2 System Architecture

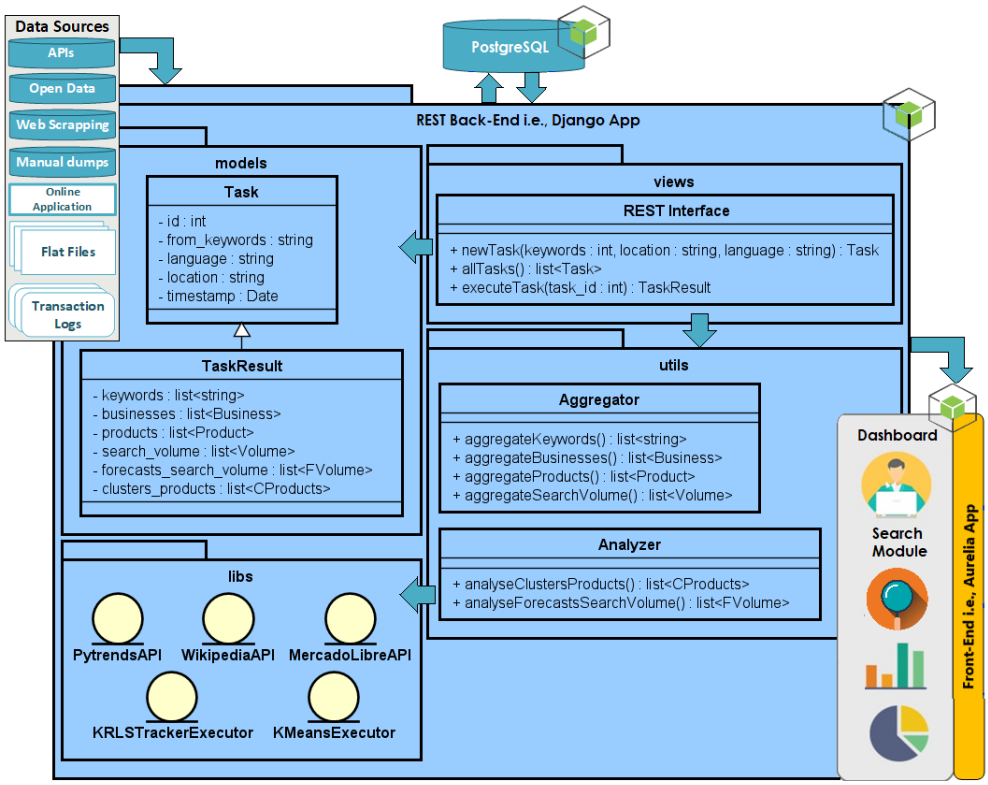

Figure 3.2: This diagram contains the modules that implement the most recent version of the MRS.

The MRS is composed of a Back-End and a Front-End. The Back-End is served through Nginx and runs inside a Docker container. It is written using Django, a Python framework, and stores its data in a local PostgreSQL database. The main components of the server are the Aggregator and the Analyser. The Aggregator module uses various APIs and scraping objects to fetch data from the web and maps the collected information on a common model. In its current version, when an user performs a search, the system uses pytrends, a Python API that interacts with Google Trends, to generate 400 keywords and a time series with their interest over time. The interest over time trends come in the form of a value that goes from 0 to 100 and that is relative to the most searched term for a given location and time. Using the generated keywords the system fetches the info-boxes of businesses from Wikipedia and products information. The Analyser then takes care of analysing the data and performs regression and classification in order to extract useful marketing insights. The implemented market research platform can perform clustering on the products using k-means and do forecasting on trends using an online method known as Kernel Recursive Least Squares Tracker (KRLS-T). All this information is served to customers in JSON format through a REST interface. The Front End application was built using Aurelia, a JavaScript framework, and allows the users to use the functions of the MRS and consume any marketing information through lists, graphs and tables. The products, businesses and related keywords are appropriately shown in a web browser with a design similar to figure 3.1.

3.3 Forecasting using KRLS methods

When a user performs a search the MRS looks for a list of keywords associated with the keywords provided by the user and for every keywords is able to download a 5 years time series of trends that have a value that goes from 0 to 100 and that is relative to the most searched term on the Google search engine for a given location and time. The system then applies KRLS regression to each 5 years time series set to perform forecasting and derives the values of these trends 2 years into the future. Before giving it as input to the algorithm every time series is prepared by mapping the dates to a 10 dimensional array of random values where the time positioning is achieved by shifting every row by one for every successive date. Algorithm 2 shows this process more in detail.

Algorithm 2 KRLS Tracker (KRLS-T) executor

Fetch time series of keywords
λ = forgetting factor
c = regularization
M = dictionary size
for $i = 1, 2, ..., M$ **do**
  Consider $dates_i$ and $values_i$ of i-th keyword
  Map every item of $dates_i$ to a 10 dimensional vector $dates_i^{'}$
  forecasts i-th keyword are calculated with $KRLST(dates_i^{'}, values_i, \lambda, c, M)$.
end for

pos	dates	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	trends
1	14/08/2016	-0.058	0	0	0	0	0	0	0	0	0	65
2	21/08/2016	0.054	-0.058	0	0	0	0	0	0	0	0	70
3	28/08/2016	-0.729	0.054	-0.058	0	0	0	0	0	0	0	71
4	04/09/2016	0.491	-0.729	0.054	-0.058	0	0	0	0	0	0	74
5	11/09/2016	-0.61	0.491	-0.729	0.054	-0.058	0	0	0	0	0	72
6	18/09/2016	-0.904	-0.61	0.491	-0.729	0.054	-0.058	0	0	0	0	72
7	25/09/2016	1.012	-0.904	-0.61	0.491	-0.729	0.054	-0.058	0	0	0	72
8	02/10/2016	-0.016	1.012	-0.904	-0.61	0.491	-0.729	0.054	-0.058	0	0	73
9	09/10/2016	-1.658	-0.016	1.012	-0.904	-0.61	0.491	-0.729	0.054	-0.058	0	73
10	16/10/2016	-0.165	-1.658	-0.016	1.012	-0.904	-0.61	0.491	-0.729	0.054	-0.058	72
11	23/10/2016	2.062	-0.165	-1.658	-0.016	1.012	-0.904	-0.61	0.491	-0.729	0.054	71
12	30/10/2016	-0.903	2.062	-0.165	-1.658	-0.016	1.012	-0.904	-0.61	0.491	-0.729	74
13	06/11/2016	1.809	-0.903	2.062	-0.165	-1.658	-0.016	1.012	-0.904	-0.61	0.491	73
14	13/11/2016	-0.236	1.809	-0.903	2.062	-0.165	-1.658	-0.016	1.012	-0.904	-0.61	75
15	20/11/2016	-1.554	-0.236	1.809	-0.903	2.062	-0.165	-1.658	-0.016	1.012	-0.904	72
					...							

Figure 3.3: This table shows how the dates of a time series are mapped to 10 dimensions filled with random values where time representation is obtained by shifting each successive date by one position.

When the MRS finishes calculating all the forecast the results are returned to the user which is able to visualize a time series for every keyword showing the actual demand and the forecasted demand. The interface the user sees looks very similar to figure 3.4.

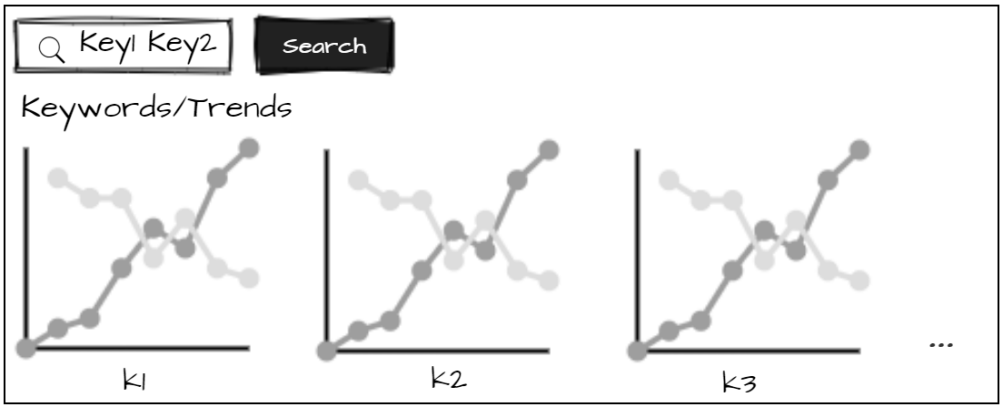

Figure 3.4: This wireframe shows how the system displays the demand for each keyword.

Marketing data is non-stationary which means that it is affected by several implicit variables. The goal of the current section is to show that KRLS regression techniques are able to model such data and perform forecasting accurately. In order to understand which KRLS variant is the most accurate, the algorithm was tested with different parameters and the corresponding MSEs were calculated. The MSEs have been calculated using a training set of 200 points and a test set of 60 points. Every time a new training point was added to the algorithm, the MSE was calculated by subtracting the estimated values and the actual values of the test set. Only two kernel were used, the first being the radial basis function that has the following form:

$$K(X_i, X_j) = \exp(-\frac{|| X_i - X_j ||^2}{2\sigma^2})$$

The second is the a rational quadratic function that has the form:

$$K(X_i, X_j) = (1 + \frac{|| X_i - X_j ||^2}{2\alpha l^2})^{-\alpha}$$

When the algorithm uses no forgetting factor it means that it gives equal importance to all the training data that is available to it. On the contrary, the algorithm uses some weights for each base. The dictionary size drives how much of the data is kept as the basis of the algorithm. The data used for analysis is the demand relative to the keyword food.

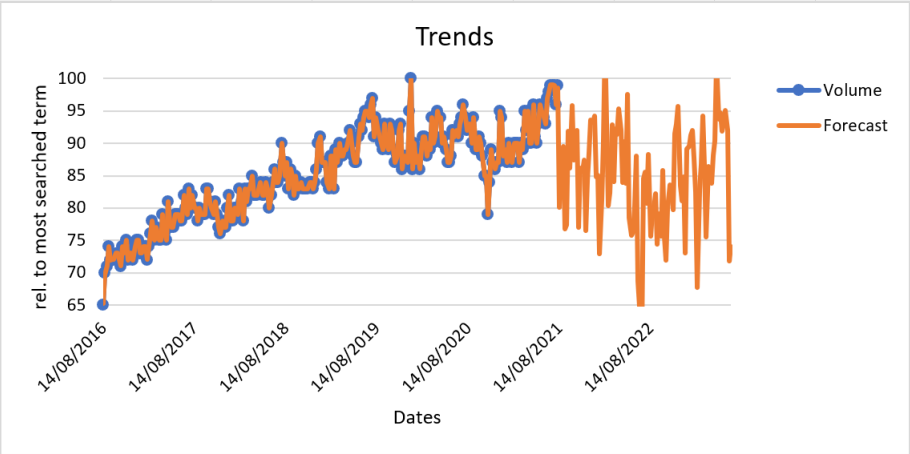

Figure 3.5: The figure shows the search indexes over time relative to the keyword 'food' together with the estimated indexes calculated with a KRLS-T regressor that uses a RBF(2) kernel, a forgetting factor of 1, a regularization of 1e-4 and a dictionary size of 300 elements.

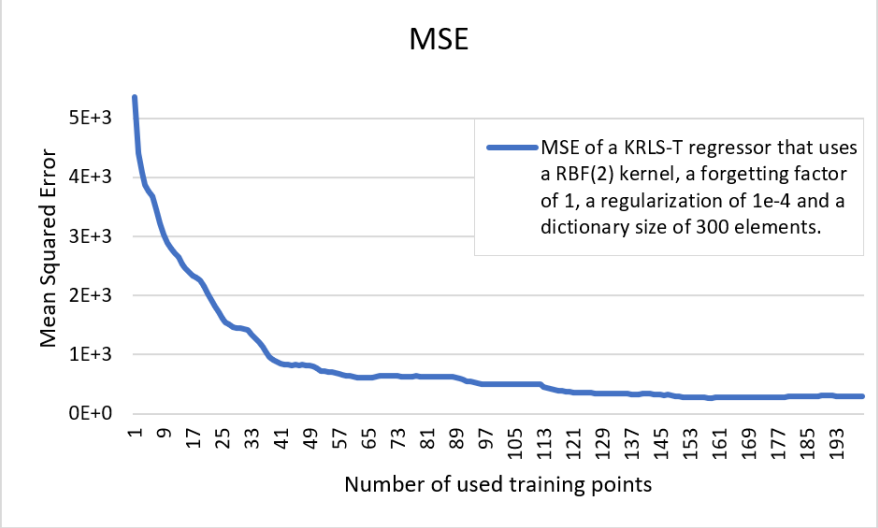

Figure 3.6: The figure shows the improvement over time of the Mean Squared Error of a KRLS-T regressor that models the demand of the keyword 'food' and uses a RBF(2) kernel, a forgetting factor of 1, a regularization of 1e-4 and a dictionary size of 300 elements.

The algorithm utilized in figure 3.5 uses a radial basis function kernel, a dictionary of 300 data points and no forgetting factor. The results show that the chosen kernel does not perform very well with the provided data. Figure 3.6 demonstrates that the algorithm becomes more and more accurate as more training data is used, however it eventually stops improving; the forecasts are not very good as they deviate from the expectation.

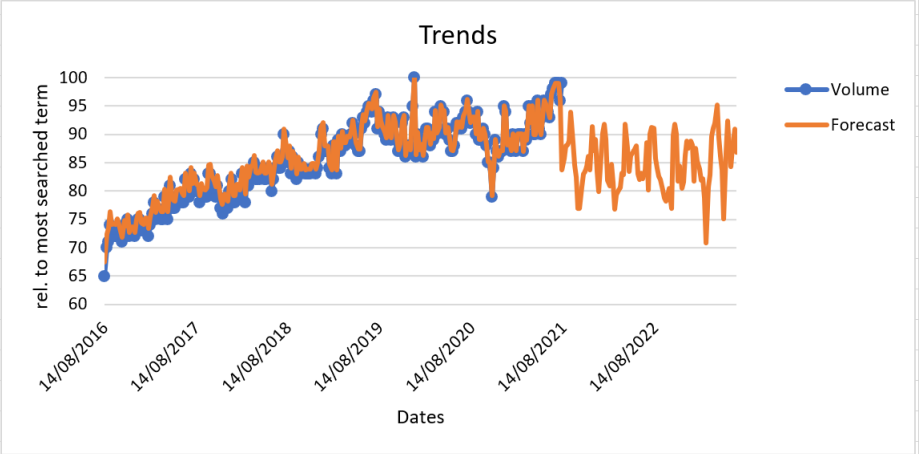

Figure 3.7: The figure shows the search indexes over time relative to the keyword 'food' together with the estimated indexes calculated with a KRLS-T regressor that uses a RQ(2) kernel, a forgetting factor of 0.99999, a regularization of 1e-4 and a dictionary size of 300 elements.

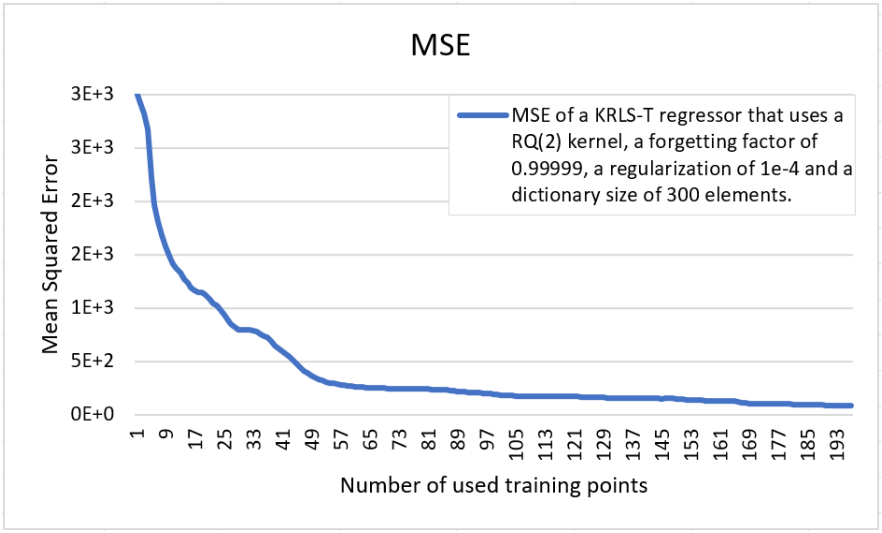

Figure 3.8: The figure shows the improvement over time of the Mean Squared Error of a KRLS-T regressor that models the demand of the keyword 'food' and uses a RQ(2) kernel, a forgetting factor of 0.99999, a regularization of 1e-4 and a dictionary size of 300 elements.

The results shown in figure 3.7 have been calculated using a KRLS regressor that employs a rational quadratic function, a dictionary of 300 elements and a back-to-prior forgetting factor of 0.99999. The performance appears much better than the previous case. The function does not approximate the data as well as before, but the forecasts are much closer to the expectation. Figure 3.8 demonstrates how the MSE gets much lower than

the previous case and in general improves a lot faster with training without reaching a plateau. Because of the performance shown by this kernel it has been further tested with various other parameters.

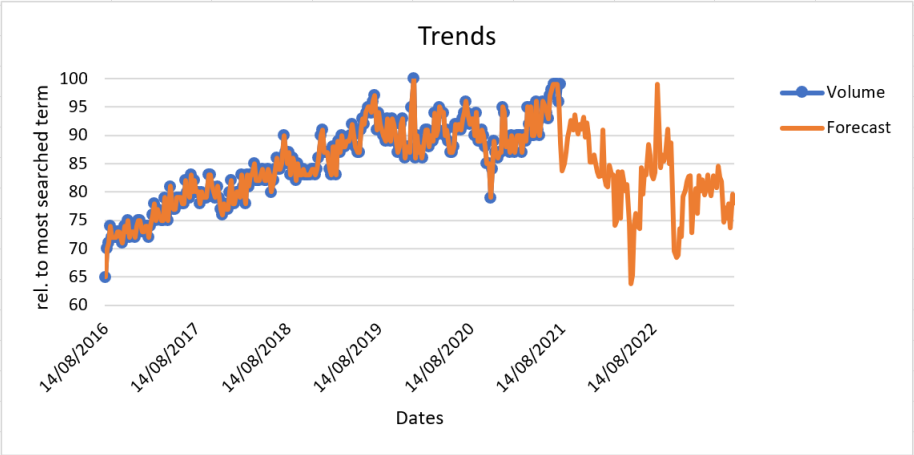

Figure 3.9: The figure shows the search indexes over time relative to the keyword 'food' together with the estimated indexes calculated with a KRLS-T regressor that uses a RQ(2) kernel, a forgetting factor of 1, a regularization of 1e-4 and a dictionary size of 300 elements.

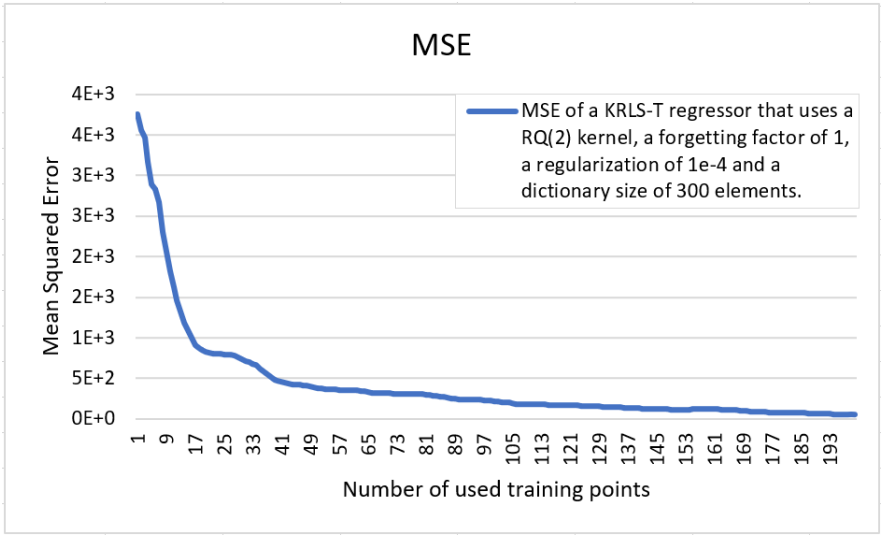

Figure 3.10: The figure shows the improvement over time of the Mean Squared Error of a KRLS-T regressor that models the demand of the keyword 'food' and uses a RQ(2) kernel, a forgetting factor of 1, a regularization of 1e-4 and a dictionary size of 300 elements.

The results displayed in figure 3.9 were calculated with a KRLS regressor that employs a rational quadratic function, no forgetting factor and a dictionary of 300 elements. The performance appears to be similar to the previous case. The function can approximate the data as well as before, while the forecasts are slightly worse than the expected behaviour.

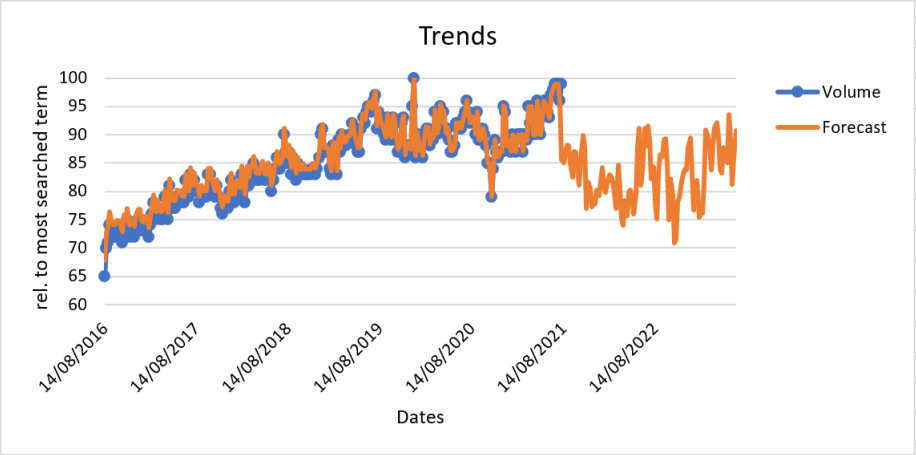

Figure 3.11: The figure shows the search indexes over time relative to the keyword 'food' together with the estimated indexes calculated with a KRLS-T regressor that uses a RQ(2) kernel, a forgetting factor of 0.99999, a regularization of 1e-4 and a dictionary size of 100 elements.

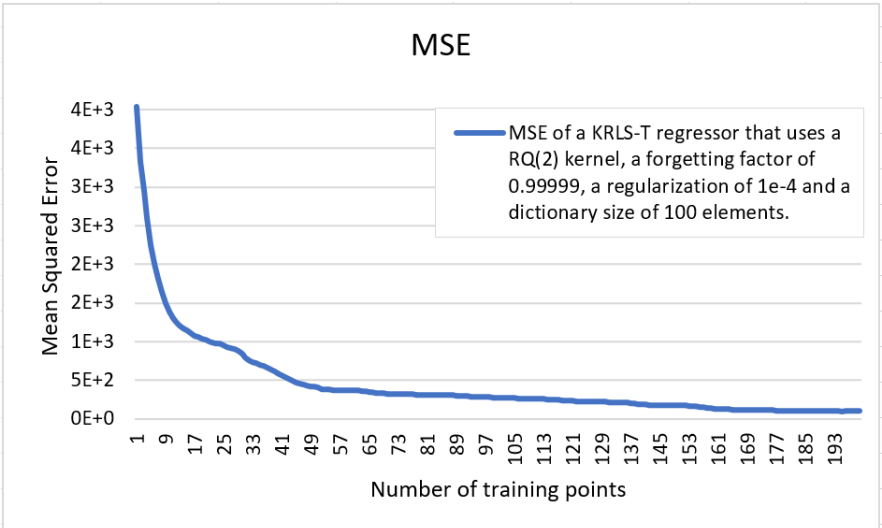

Figure 3.12: The figure shows the improvement over time of the Mean Squared Error of a KRLS-T regressor that models the demand of the keyword 'food' and uses a RQ(2) kernel, a forgetting factor of 0.99999, a regularization of 1e-4 and a dictionary size of 100 elements.

The results displayed in figure 3.11 have been calculated using a KRLS regressor that employs a rational quadratic function and a back-to-prior forgetting factor of 0.99999, but a very small dictionary. The performance is apparently good but the forecasted values don't make much sense as can be seen in 3.12. There is an abrupt drop at the intersection between the actual values and the estimated values.

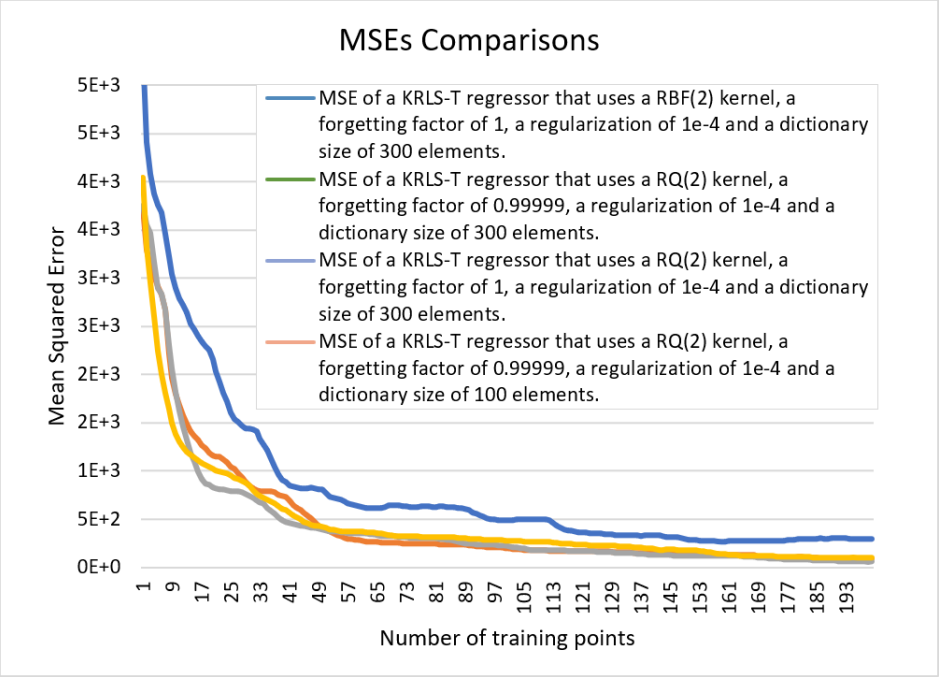

Figure 3.13: The figure shows a comparison between the MSEs obtained with different KRLS-T regressors that use different parameters. This MSEs are all relative to the modelling of the demand of the keyword 'food'.

As can be seen in figure 3.13 the KRLS-T algorithm is more accurate when it uses a rational quadratic kernel. When using this type of kernel the performance is best when using no forgetting factor and a large dictionary size. However, even if the MSE appears better in this case, the actual forecasts appear much closer to the expectation when a back-to-prior forgetting factor is being used.

3.4 Clustering products using k-means

The presented MRS also implements a clustering method in which products are clustered into categories. Before fetching it to the algorithm, every product item is converted to a set of words and then again to a word frequency vector using the Python scikit-learn library TfidfVectorizer. After this operation it is possible to cluster the vectorized representation of the products into the Euclidean space with a k-means algorithm. The algorithm produces a list of clusters similar to figure 3.14.

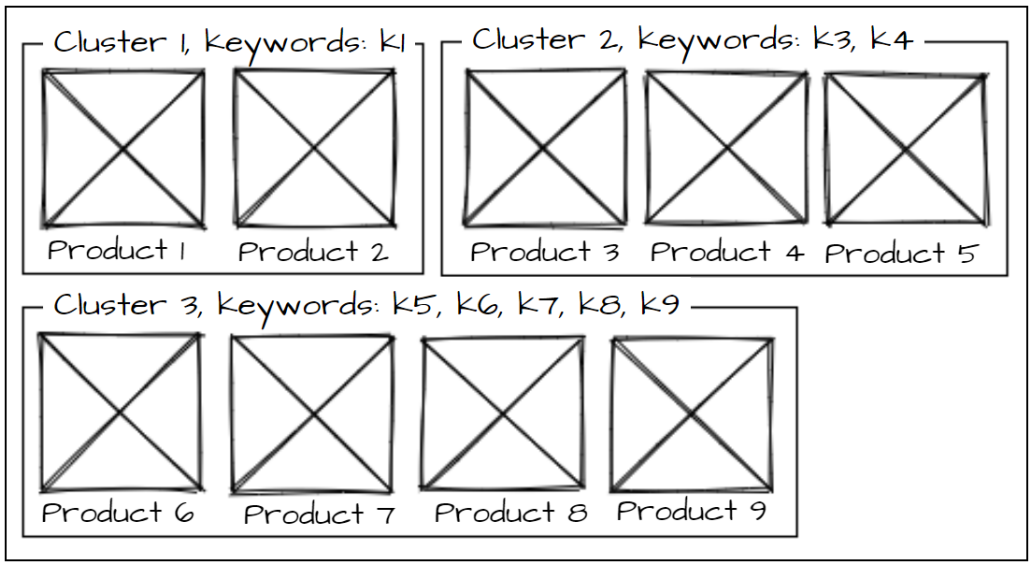

Figure 3.14: This wireframe shows how products are placed inside clusters indexed by the set of the most common keywords within every cluster.

In order to understand which is the optimal number of clusters k, the silhouette score is used, which is a value that measures how similar a point is to its own cluster compared to other clusters [17]. The Silhouette Value s(i) for each data point i is defined as follows:

$$\begin{cases} s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}, \mid C_i \mid > 1 \\ s(i) = 0, \mid C_i \mid = 1 \end{cases}$$

Where $a(i) = \frac{1}{|C_i-1|} \sum_{j \in C_i, i \neq j} \mid\mid i - j \mid\mid$ is the measure of similarity of the point i to its own cluster and $b(i) = \min_{i \neq j} \frac{1}{|C_i|} \sum_{j \in C_i} \mid\mid i - j \mid\mid$ is a measure of dissimilarity of i from points in other clusters. The optimal k is the value for which the Silhouette Score reaches its global maximum.

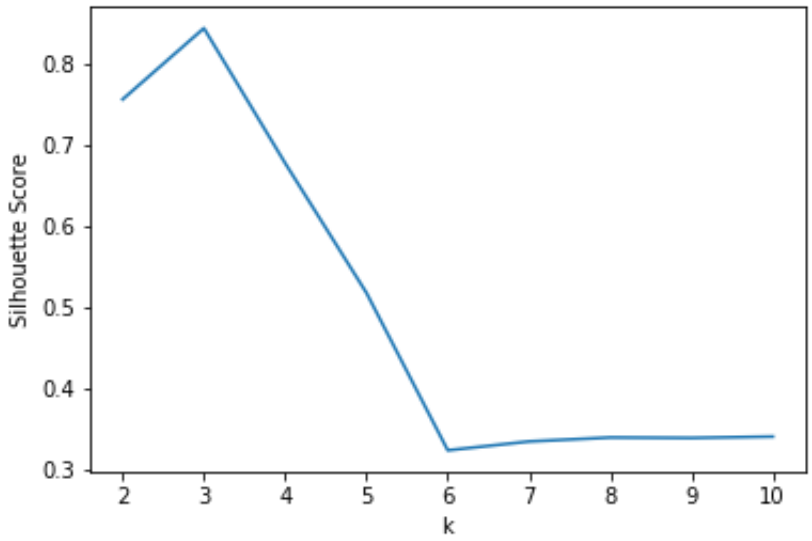

Figure 3.15: In an hypothetical scenario where the global maximum 3, the optimal number of clusters is also 3.

3.5 Future work

It's possible to optimize the currently developed MRS. In the current implementation the KRLS regression performs forecasting on one time series at a time, but it's also possible to apply the KRLS algorithms on all trends at once in a MIMO fashion. It is possible to make data gathering much faster by utilizing data sources that are local to the server rather than using remote data sources or scraping the web. It is also possible to improve the speed of data fetching by clustering data before the queries actually happen, so that the search system can simply look within the clusters relevant to the search. In general it is preferred to use data APIs than resort to web scraping and that is because the latter is very inefficient and sometimes it is not feasible, since websites change their DOM structure frequently and they might even change their DOM structure dynamically. This is why the current project uses a KRLS scraper and not an Amazon or Ebay scraper. The MRS can be made more general purpose by adding more research analytics.

4 Conclusions

This last chapter reports the evaluations and considerations regarding the completion of the set objectives and offers a summary and reflection on the research.

This project attempted to implement a prototype of a general purpose MRS that gathers data from various sources and generates an array of marketing insights. The implemented prototype is divided into a Front-End application and into a Back-End server that run into Docker containers. The implemented MRS is currently able to gather some data using some web scrapers and can analyse this data to produce forecasts and clusters for products. The proposed MRS can be optimized in various ways. For example, APIs can be used instead of web scrapers and the data sources can be brought locally to the server. Marketing data is non stationary and it is difficult to model due to data points having different distributions. This thesis shows a way to model such data and perform accurate forecasting using kernel machines. A KRLS-T model was used that is able to more accurately predict what comes next by keeping a set of distributions for each data point. It can be shown that the choice of the kernel greatly affects the performance of kernel machines. In non-stationary scenarios, this online algorithm implements a forgetting mechanism that can track the changes of the observed model by weighting past data less heavily than more recent data. The forgetting mechanism improves performance by generating a curve that does not fit so well the training data but can provide very good forecasts. The dictionary size is also a factor as more training data can empower the algorithm to also perform better. The KRLS-T algorithm achieves the lowest MSE when using a rational quadratic kernel, no forgetting factor and a large dictionary size. However, the forecasts make more sense when using also a forgetting factor. When using a radial basis function the performance is inferior.

References

[1] Bello Ayuba, Olalekan Aina, and Kazeem. The role of marketing research on the performance of business organizations. 7, 01 2015. https://www.researchgate.net/publication/344177049_The_Role_of_Marketing_Research_on_the_Performance_of_Business_Organizations.

[2] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proc. VLDB Endow.*, 5, 03 2012. https://www.researchgate.net/publication/221966107_Scalable_K-Means.

[3] H. N. Bhor, T. Koul, R. Malviya, and K. Mundra. Digital media marketing using trend analysis on social media. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 1398–1400, 2018. https://ieeexplore.ieee.org/document/8399038.

[4] Nigel Brown. Containerizing a software application with docker. https://app.pluralsight.com/.

[5] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. volume 51, pages 137–150, 01 2004. https://www.researchgate.net/publication/220851866_MapReduce_Simplified_Data_Processing_on_Large_Clusters.

[6] Susanna Galbraith. Google analytics. *Journal of the Canadian Health Libraries Association*, 34:119–122, 08 2013. https://www.researchgate.net/publication/272894748_Google_Analytics.

[7] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. *Proceeding VLDB '99 Proceedings of the 25th International Conference on Very Large Data Bases*, 99, 05 2000. https://www.researchgate.net/publication/2634460_Similarity_Search_in_High_Dimensions_via_Hashing.

[8] Y. Hua, B. Xiao, B. Veeravalli, and D. Feng. Locality-sensitive bloom filter for approximate membership query. *IEEE Transactions on Computers*, 61(6):817–830, 2012. https://ieeexplore.ieee.org/document/5928322.

[9] L. Huang and X. Zhang. A new marketing effectiveness metric based on web data mining. In *2009 1st IEEE Symposium on Web Society*, pages 5–9, 2009. https://ieeexplore.ieee.org/document/5271733.

[10] Reindert Jan. Django fundamentals. https://app.pluralsight.com/.

[11] Kipp Johnson, Jessica Soto, Benjamin Glicksberg, Shameer Khader, Riccardo Miotto, Mohsin Ali, and Euan Ashley. Artificial intelligence in cardiology. *Journal of the American College of Cardiology*, 71:2668–2679, 06 2018. https://www.researchgate.net/publication/325585229_Artificial_Intelligence_in_Cardiology.

[12] Lixin Ma and Yang Liu. Application of a deep reinforcement learning method in financial market trading. In *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 421–425, 2019. https://ieeexplore.ieee.org/document/8858758.

[13] V. S. Rajput and S. M. Dubey. Stock market sentiment analysis based on machine learning. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 506–510, 2016. https://ieeexplore.ieee.org/document/7877468.

[14] Dilber Ulas. Digital transformation process and smes. *Procedia Computer Science*, 158:662–671, 2019. 3rd WORLD CONFERENCE ON TECHNOLOGY, INNOVATION AND ENTREPRENEURSHIP"INDUSTRY 4.0 FOCUSED INNOVATION, TECHNOLOGY, ENTREPRENEURSHIP AND MANUFACTURE" June 21-23, 2019.

[15] S. Van Vaerenbergh and I. Santamaria. Online regression with kernels. In *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 477–501, New York, 2014. Chapman and Hall/CRC. https://gtas.unican.es/files/pub/ch21_online_regression_with_kernels.pdf.

[16] P. Vats and K. Samdani. Study on machine learning techniques in financial markets. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5, 2019. https://ieeexplore.ieee.org/document/8878741.

[17] Fei Wang, Hector-Hugo Franco-Penya, John Kelleher, John Pugh, and Robert Ross. An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. 07 2017. https://www.researchgate.net/publication/318109824.

[18] Matei Zaharia, Mosharaf Chowdhury, Michael Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 10:10–10, 07 2010. https://www.researchgate.net/publication/234790155_Spark_Cluster_Computing_with_Working_Sets.

[19] P. Zhang, X. Shi, and S. U. Khan. Quantcloud: Enabling big data complex event processing for quantitative finance through a data-driven execution. *IEEE Transactions on Big Data*, 5(4):564–575, 2019. https://ieeexplore.ieee.org/document/8386682.