FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Evaluating the impact of climatic variability in wine production

Rui Miguel Cruz Soares Pinto

FINAL VERSION

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Pedro Carvalho Leal Mendes Moreira

Supervisor: Mário Manuel de Miranda Furtado Campos Cunha

July 28, 2016

Evaluating the impact of climatic variability in wine production

Rui Miguel Cruz Soares Pinto

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Daniel Silva
External Examiner: Paulo Cortez
Supervisor: João Pedro Moreira
Supervisor: Mário Cunha
July 28, 2016

Abstract

In the wine production business, climate variability is one of the most critical conditions, being essential in regards to the process of ripening fruits so that it possesses the required characteristics to produce a good wine. Adding to this factor, climatic variations may have disastrous consequences not only for wine producers and workers but also for the land used for vineyards. Performing a good forecasting and statistical analysis of the wine productions can help businesses save money and preserve the environment. In regards to this problem, new solutions arise for the processing and information extraction of "datasets". In this particular case and based on data from winery production of past years and with further analysis, it is possible to achieve and identify the different climatic components and their impact on wine production. The solution presented on this paper would be based on the premise of "Machine Learning" consisting in building a model based on existing data provided by the "Dataset" [3] in order to be able to group similar data into subgroups according to its characteristics and consequently giving it the ability to predict the production based on a set of meteorological conditions. This grouping data process, would somewhat prove the relationship between the meteorological series and its impact on winery production.

The implementation of this solution would have a good innovation component, since the use of decision trees applied to multivariate time series is still in its early stages and much discussion is had about this subject. Another advantage of this project is the ability to create a model in a form of a decision tree which can be an easy to interpret graphic that even people outside the Data Mining world can understand and benefit from such, since although this thesis is about the impact of meteorological conditions in wine production, this kind of model can be applied to a plethora of other subjects.

Resumo

No negócio de produção de vinhos a variabilidade do clima é uma das condicionantes mais importantes, sendo o aspeto mais crítico no que diz respeito ao processo de amadurecimento do fruto de maneira a que este possua as caracteristicas necessárias para a produção de um bom vinho. Acrescentando a esse fator, as variações climáticas têm consequências nefastas não só para os trabalhadores dessa área como também nos terrenos utilizados para as vinhas. A realização de uma boa previsão e análise estatística das produções vinícolas anteriores podem ajudar empresas a poupar dinheiro e a preservar o ambiente. Assim, como possíveis soluções surgem diferentes alternativas para o processamento dos “datasets” de produção vinicola de anos passados e posterior análise, conseguindo assim identificar os diferentes componentes climáticos e os seus impactos na produção de vinhos. A opção presenteada neste paper poderá ser baseada na premissa de “Machine Learning” que consiste na construção de um modelo baseando-se nos dados já existentes fornecidos pelo “Dataset”[3] , de forma a conseguir agrupar os dados de teste semelhantes em subgrupos de acordo com as suas características e consequentemente dando a abilidade ao modelo de prever qual seria a produção para um determinado conjunto de dados meterologicos. Este processo evidenciaria a relação existente entre as séries meteorólogicas e o seu impacto na produção vinicola.

A implementação desta solução teria uma boa componente de inovação, visto que o uso de árvores de decisao aplicado a séries temporais com multiplas variáveis ainda está numa fase inicial e existe muita discussão nesta temática. Outra vantagem deste projeto e o de permitir a criação de um modelo em forma de árvore de decisão que se pode transformar num grafico que seja fácil de interpretar mesmo para pessoas fora do mundo de Data Mining. Por fim, apesar de esta tese ser acerca do impacto das condições meterologicas na produção de vinhos, este tipo de modelos pode ser aplicado a uma panoplia de outros temas.

Acknowledgements

The experience of being able to attend an university as renown as Faculdade de Engenharia da Universidade do Porto comes as a great privilege as I feel they gave me all the required tools to have great success in the working world. My wholeheartedly thanks to this great faculty and all the people involved in this project of excellence and ambition.

I would not be able to have as much success as I did without the full support of my family, who always helped me and supported me since the beginning. A special thanks to both my parents, Carmo e Jorge, who made all of this possible and never stopped supporting me, and also my brother, Pedro that always was available for me, and help made this journey a success. A special thanks to my friends that accompanied me during these 5 years of learning, who made all this learning experience even more enjoyable.

Lastly, I would like to thank both my counselors, João Mendes Moreira from FEUP and Mário Cunha from FCUP, for all the guidance, availability and support that made this thesis possible.

Rui Pinto

“There are some ideas so wrong that only a very intelligent person could believe in them”

George Orwell

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

DTW	Dynamic Time Warping
DT	Decision Trees
EDM	Euclidean Distance Method
MTS	Multivariate Time Series
CART	Classification and Regression Trees
LSD	Least Squares Deviation

Chapter 1

Introduction

1.1 Context

As the world continues to develop into the XXI century, the amount of technology used and also the vast chunks of information and data being transferred brings a whole new set of problems and challenges. This was also made possible by the evolution of hardware that more and more allow computers to perform complex operations and store colossal size of information. With the evolution of technology, analyzing data has become a prime concern in the present days due to the fact that with more information, more conclusions can be taken from it, like for example in substantially different areas such as Medicine, Sports or even Wine Production. Due to this issue, there is a need to develop and implement more algorithms that are able to efficiently retrieve important information from datasets. The goal of this project is to be able to develop a new algorithm associated with the subgroup discovery area of Data Mining and apply it to a vast dataset of Wine production in the Douro Region represented by Time Series.

1.2 Motivation and Objectives

This thesis has an innovation component in which it is asked to implement a new solution involving Machine Learning and Subgroup Discovery to a Multivariate Time Series problem. This innovation, comes from the fact that in this area the use of Decision Trees for subgroup discovery in Multivariate Time Series is still in its early stages.The advancements on this subject may lead to new ways to be able to solve these types of problems as well as a simple and concise way to represent the solutions through decision trees and rule sets. These factors joined the utility that the decision trees model bring that allow an average person to easily interpret one of these models, makes it so that this project has both an innovation part and some real life practical utility in helping companies and people better predict their outcomes

1.3 Structure

After a short introduction about this project, this dissertation will present some topics about the utility of decision trees and how they can be applied to multivariate time series.

Besides the introduction, this dissertation contains 4 more chapters.

In chapter 2, the state of the art is described in addition to some similar solutions in the field being presented and analyzed.

In chapter 3, the core of the implementation as well as choices that were made and their explanation. Also the validity of the solution is analyzed.

Chapter 4. will analyze the results and problems obtained and take conclusions on why they happened.

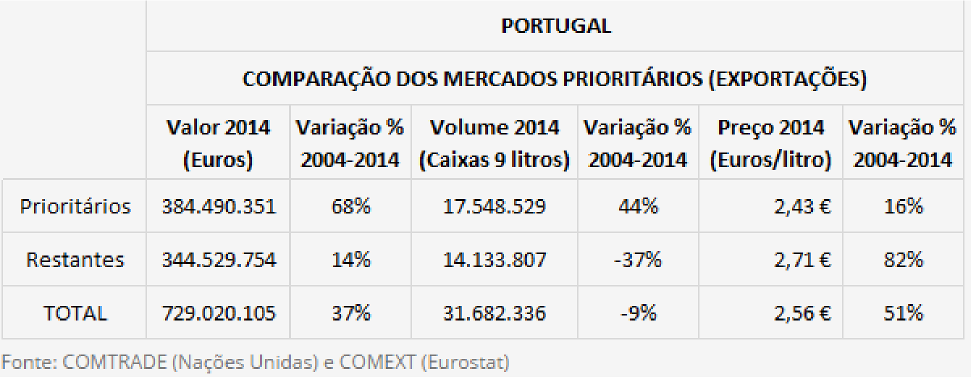
the last chapter 5 is where the conclusions and general appreciation of the project are made as well as what were the limitations and a path for future work regarding this dissertation.

Chapter 2

Literature Review

2.1 Wine Production

Wine production is in Portugal a main drive in economic exports, due to that nature it has become a focus of study regarding the development and production related to this art. Regarding regional wine production, it is characterized by large inter-annual fluctuations which is adverse to everyone involved in the process, from the producers, to the people working on the vineyards and even the environment [CR12]. In 2014, Portugal managed to export 725 millions liters of wine, complying 1.5% of total exports, making Portugal the 9th country in the world regarding wine commerce and 12th in terms of wine production [Vin].



	PORTUGAL					
	COMPARAÇÃO DOS MERCADOS PRIORITÁRIOS (EXPORTAÇÕES)					
	Valor 2014 (Euros)	Variação % 2004-2014	Volume 2014 (Caixas 9 litros)	Variação % 2004-2014	Preço 2014 (Euros/litro)	Variação % 2004-2014
Prioritários	384.490.351	68%	17.548.529	44%	2,43 €	16%
Restantes	344.529.754	14%	14.133.807	-37%	2,71 €	82%
TOTAL	729.020.105	37%	31.682.336	-9%	2,56 €	51%

Fonte: COMTRADE (Nações Unidas) e COMEXT (Eurostat)

Figure 2.1: Portugal exports in the year 2014 [Vin]

The main issue with wine production is that it does not follow a linear increase since the many features that affect it (meteorological conditions, technological advancements, or even the increase in the vineyards area) provoke fluctuations in that regard. This leaves us with one of the goals of this paper which is to measure the impact of five important climatic characteristics: *minimum temperature, precipitation, maximum temperature, medium temperature, and amount of water in soil*. Just for the sake of an easier visualization, in image 2.2, it is possible to see the deviation of production according to the "trend".

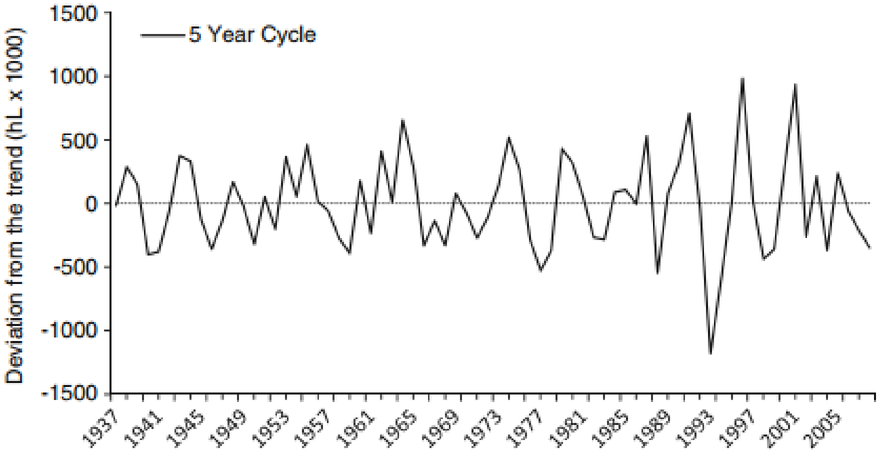


Figure 2.2: Deviation of production along the years compared to the trend [CR12]

2.2 Time Series

2.2.1 What is it?

A time series is the most frequent type of data in Data Mining problems [LKT03], time oriented data is present in the most diverse fields of interest, from measuring the performance of a sports athlete to analyzing the variation of stock prices to even measuring the meteorological conditions through time. Due to the importance of this type of data several methods for the most diverse tasks were invented such as classification, clustering, prediction and anomaly detection, whereas classification and regression are the most common [LKT03].

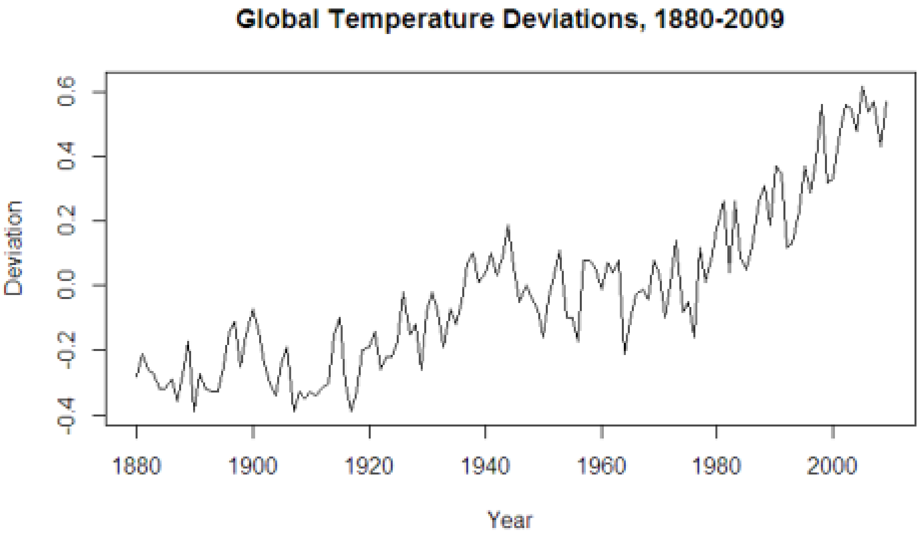


Figure 2.3: Example of a time series

2.2.2 How to compare Time Series?

One of the most common goals related to time series, is being able to check the similarity between two or more of this data representation. And therefore several methods were created in order to calculate this similarity.

2.2.2.1 Euclidean Distance

The most simple method in order to find similarity between two time series is by calculating the Euclidean distance between the two of them. Despite this method being simple it contains two major advantages, the first being the order of complexity which is simply O(n) and the second being allowing scalable solutions to other problems such as clustering [GF]. However, this method has one major flaw which is its downfall. The Euclidean distance method only allows comparison between two points at the same time not allowing a good comparison between two time series with unequal length.

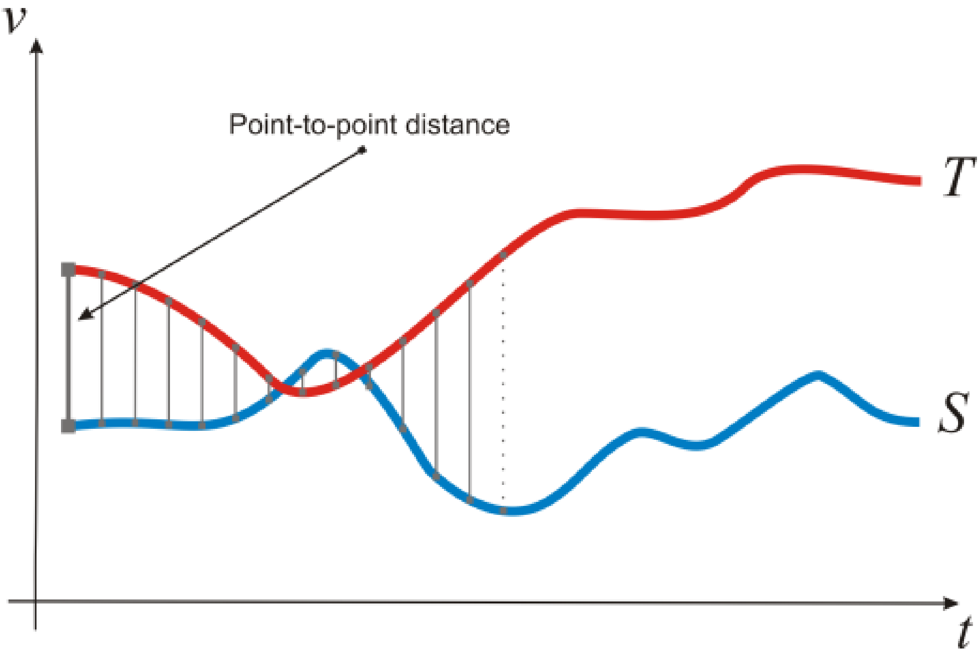


Figure 2.4: Example of Euclidean Distance between two time series [CMA$^+$12]

As we can see in image 2.4 the vertical lines, are the several distances between the two times series, and they are bound to a specific moment in time. Nonetheless, what happens if instead we want to compare points of interest (two maximums) but they are in different moments of the time axis? It becomes obvious that this solution cannot solve these types of problems, and so another method has to be used like Dynamic Time Warping for example.

2.2.2.2 Dynamic Time Warping

DTW is another method for comparing time series and compared to the euclidean distance alternative, it has the main advantage of being able to compare two time series with unequal length. However, this advantage comes with a price which is the quadratic complexity O(n*n) , making it very costly to use on lengthy time series. [Li15].

The best warping path is found in the cost matrix of distances between the two time series [Li15]. So in order to better understand the differences between the Euclidean and DTW method, its best to take a look at image 2.5.

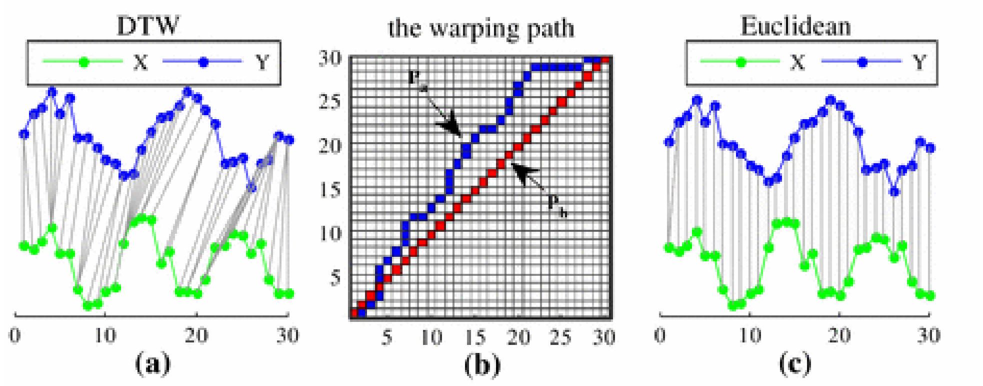


Figure 2.5: Comparisons between DTW and Euclidean Distance methods [Li15]

As we can see in image 2.5, we have in a) the DTW method being represented with the lines being distorted from the typical vertical lines. In the second image b) there is the blue image P(a) and the red image P(b), representing respectively the warping path of both a) and c). Finally in c) it is represented the normal Euclidean Distance Method. It is possible to verify that in a) the points with same shape match while that does not happen in image c). In an arbitrary warping path, there are three major constraints, they are *boundary*, *continuity* and *monotonicity*. The most common algorithm to construct the best warping path is the Dynamic Programming is used to construct the cost matrix [Li15].

After analyzing both methods, it is clear to see that although the Euclidean Distance method can be used for series with same size and for its better complexity, its lack of flexibility leaves DTW as a good solution for the remaining type of problems with different length time series, despite its higher cost.

2.3 Data Mining Techniques

Data Mining's main goal is to be able to extract relevant information from databases.However, nowadays, databases are becoming larger in size and as a result there is a need to improve the efficiency on how to extract the information from the complex data [SMMA16]. In this dissertation, the focus is both on classification and regression methods.

2.3.1 Classification

Classification is used to label each one of the items present in a dataset into a predefined group of classes or groups [LKT03]. The solution to these types of problems can be defined by either a set of rules or a decision tree. For example, let us imagine that the mileage of a car (High or Low) is based on two attributes, the car's weight and its horsepower, so basically the objective is to model the main target(mileage) based on the two attributes referred. In 2.6, it is possible to see this problem modeled in a decision tree.

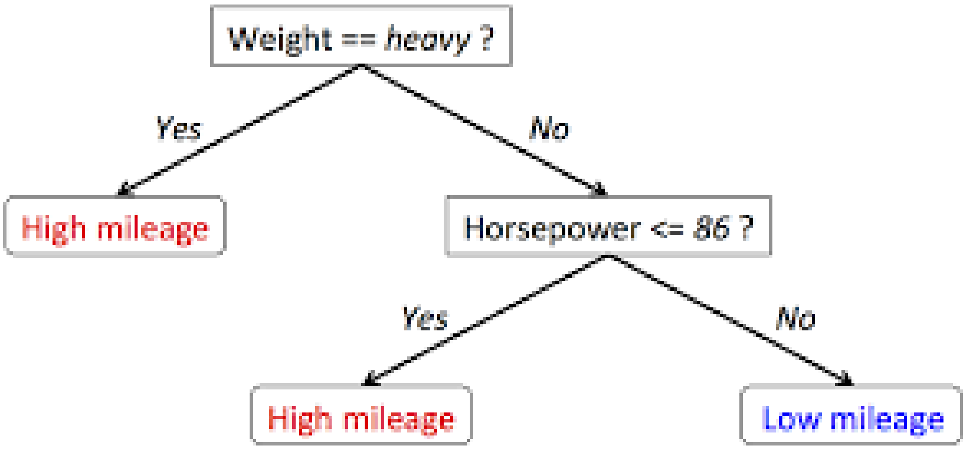


Figure 2.6: Simple Decision Tree [AB]

2.3.2 Regression

Although similar to classification, regression is applicable to situations where the target variables do not have labels [CK15], as for example, based on a set of conditions one model could predict the stock market values for the following weeks. It is possible to model an algorithm which will "learn" and based on those conditions can group the data and calculate an outcome which will be the prediction. Another example could be something in the context of this project which is given a set of meteorological conditions, a model that can predict the wine production based on the temperature. So after the problem is established, there is a need for a set of rules that models the solution.

In order to solve the problem of 2.7 a rule set is needed in order to obtain the different existing classes. A rule set is a group of rules that models a certain class, for example in the above problem a possible rule of the rule set could be: *temperature >= 25.5 & temperature < 28.5 => play = no*. The joint group of these type of rules constructs the rule-set and models the different classes.

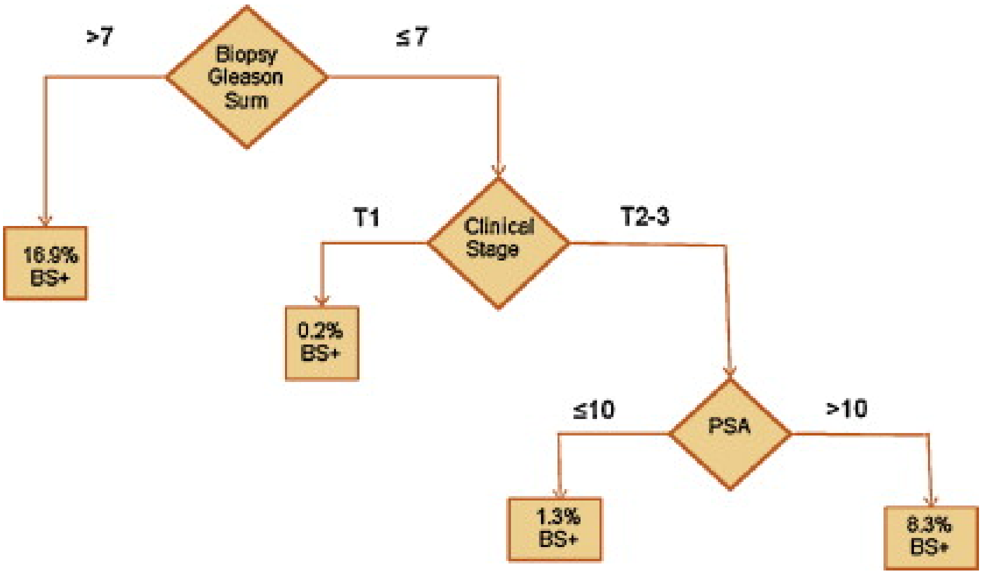


Figure 2.7: Example of a classic regression tree [K+10]

2.4 Time Series and Decision Trees

The use of decision trees in normal data is common practice in the matter of fact that it is easy to perform the splitting of the data inside the trees since the comparative process is straight-forward. However, when the dataset switches from being a normal set of data to a multivariate time series problem, there is an issue in which the implementer has to create a methodology to decide if a time series is higher or lower than the other. As mentioned in sections 2.2.2.1 and 2.2.2.2, both DTW and Euclidean Distance give a metric in which it is possible to calculate the absolute distance between two time series. However this process does not decide which series is above the other, it only gives the user how far apart they are. Over the years, there have been some suggestions and alternatives to the study of multivariate time-series and each one has its advantages and disadvantages.

2.4.1 Problems of Other Methods

In relation to Kadous proposal, he suggests a feature extraction to handle the time series data as a traditional classification problem[Kad99], however by utilizing this transformation, it produces less comprehensive classifiers than the direct approach[YSYT03]. Among other solutions there is also the Naïve approach which takes the average value of a time series and applies a nearest neighbor method for the dissimilarity measure[YSYT03]. This approach has a problem in which by calculating the average of a time series it completely neglects it's structure and can consider two largely different series as similar. In the model that will be presented in the next chapter, a time series is treated normally, and the method for comparison between two time series, is the Euclidean Distance method used for the simplicity of it and in this particular case it makes sense from a logical point of view. To better understand the other solutions suggested by other authors to solve this problem of Time series classification, two papers about this subject will be presented.

2.4.2 Decision Tree Induction from Time Series Data

In this paper by Yuu Yamada, Einoshin Suzuki, Hideto Yokoi and Katsuhiko Takabayashi, they propose a split test which finds the best time series in the data by using exhaustive search [YSYT03]. This technique is applied in this dissertation since to find the best split, the model performs an iteration through all the possible splits and calculates the gain through the LSD method, choosing the best one in the end. The main difference between our and Yamada approach is that they utilize gain ratio method to calculate the split gain.

Both this alternative and the present dissertation assume a decision tree made by the CART algorithm. However while Yamada uses it for Classification, our solution applies it to Regression.

The dissimilarity was calculated using the DTW measure to calculate the dissimilarity was used. This method instead of evaluating the time sequences vertically, can warp the path to allow relations between different points of both series. This characteristic allows the comparison of two time series with different sizes, and also fits human intuition better since a human can notice the trends between the two series[YSYT03].

Even though the method in this dissertation and in this paper are similar, there are some significant differences mainly, on both the criterion splits and the main objective of the tree. One is used for Classification while our approach is used for Regression. Also the way to calculate the dissimilarity between time series is different since we use the Euclidean Distance while the other utilizes Dynamic Time Warping.

2.4.3 Time Series Shapelets

In this paper by Lexiang Ye and Eamonn Keogh, it is proposed that instead of treating the time series as a whole, some sub-sequences of the time series are picked which are representative of the whole class. This factor would minimize the weaknesses of the Nearest Neighbor Algorithm which is the most accurate and robust method, but also more space and time requirements [YK11].

For its experiment it is concluded that even though DTW and Euclidean distances which are usually very competitive measures do not outperform random guessing, due to the fact that the data is noisy, and this noise is enough to disrupt the subtle differences between the features[YK11].

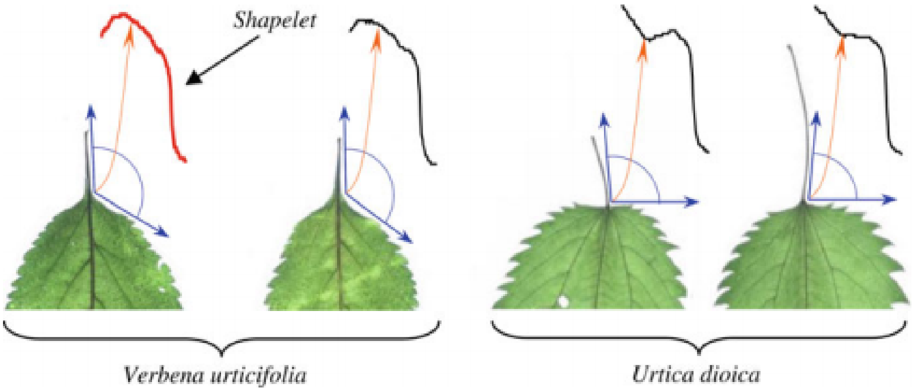


Figure 2.8: Example of a shapelet [YK11]

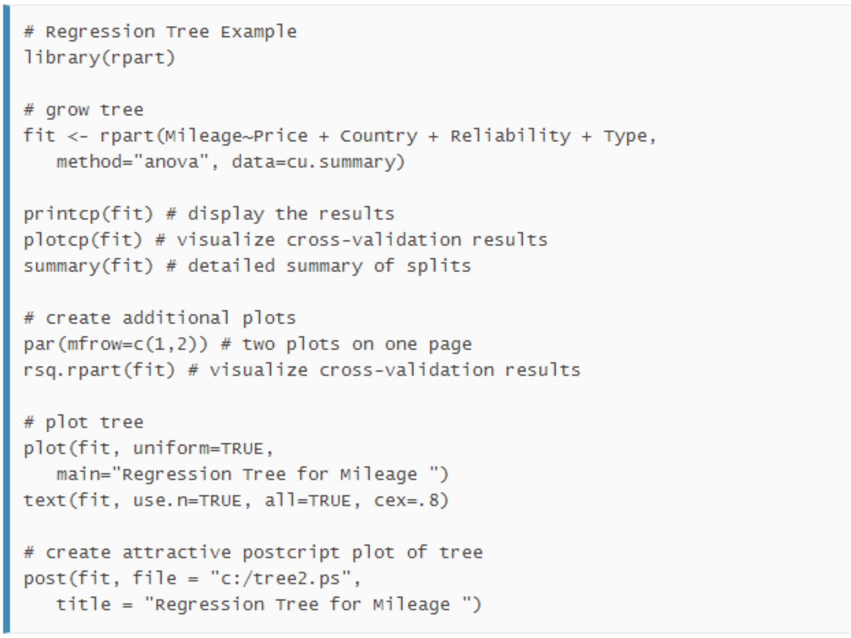
As seen in figure 2.8, for that specific case the 2nd set of images has the shapelet that better discriminates the two classes.

To obtain these shapelets, the method used is the sliding window. The sliding window method consists in acquiring all subsections of the full time series with a length of L, defined by the implementer. To find the best shapelet, several methods like Brute Force Algorithm or other more efficient methods can be used.

This method in contrast to the one utilized in this dissertation, does not treat the time series as a whole not respecting its full shape. In addition to that, the method utilized for measuring the dissimilarity between two time series, is neither DTW or Euclidean Distance based on the data they used which possesses a large amount of noise, they decided that random guess would be a better metric. Lastly, even though it is said that this method can be used for other uses than classification, they don't utilize CART for regression like in this project.

2.5 R Language

Over the last decade R has been a very commonly used tool for implementing data analysis algorithms in the most varied fields ranging from computational biology to political science. This software was created by Ross Ihaka and Rober Gentleman in 1993 [MHOV12]. The language is uncommon since it acts as a mixture of different paradigms. It's a dynamic language in the spirit of Scheme or JavaScript, however the basic data type is the vector. At the same time it is also functional since functions are first class values and arguments are passed by deep copy. Finally the language is also object oriented since it supports the creation of class objects [MHOV12]. The main advantage of using R is the fact that it contains numerous graphical and statistical models which helps user understand the data and their problems better.



```
# Regression Tree Example
library(rpart)

# grow tree
fit <- rpart(Mileage~Price + Country + Reliability + Type,
   method="anova", data=cu.summary)

printcp(fit) # display the results
plotcp(fit) # visualize cross-validation results
summary(fit) # detailed summary of splits

# create additional plots
par(mfrow=c(1,2)) # two plots on one page
rsq.rpart(fit) # visualize cross-validation results

# plot tree
plot(fit, uniform=TRUE,
   main="Regression Tree for Mileage ")
text(fit, use.n=TRUE, all=TRUE, cex=.8)

# create attractive postcript plot of tree
post(fit, file = "c:/tree2.ps",
   title = "Regression Tree for Mileage ")
```

Figure 2.9: Snippet of code written in R to construct a Regression tree

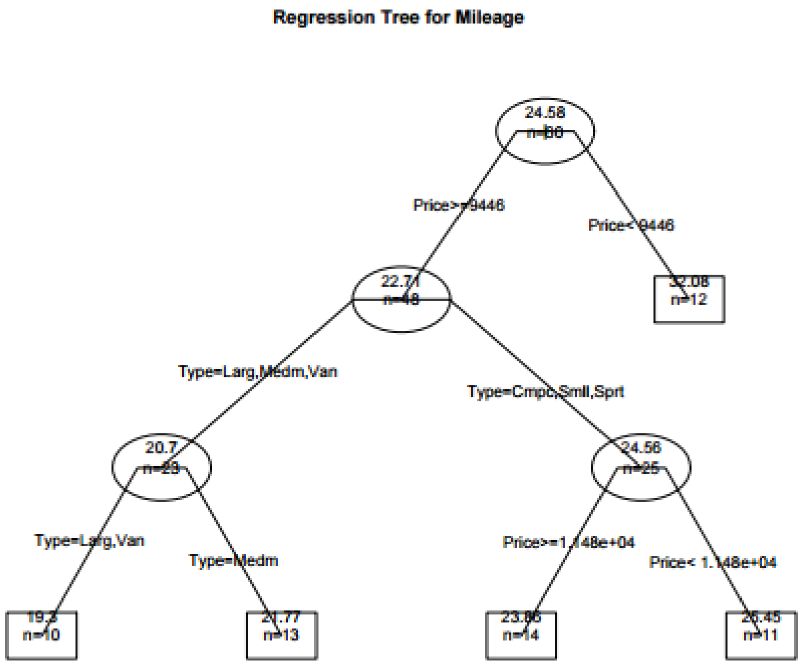


Figure 2.10: Output of code generated by 2.9

The code featured in Figure 2.9 is an easy to understand and short code that produces a simple regression tree as presented in figure 2.10.

2.6 Python Language

The Python programming language is establishing itself as one of the most popular languages for scientific computing. [PVG+11] Python is an interpreted, object-oriented, high-level programming language that possesses dynamic semantics, it also possesses a library dedicated to data mining and data analysis, named *Scikit-Learn*. This module maintains an easy to use interface integrated in the Python language. This module comes as an answer to the ever-growing need for statistical data analysis by people outside of the computer science area that need models easy to interpret, like in the area of medicine or physics [PVG+11]. In Listing [] it is possible to see a small code utilizing the Scikit library to create a decision tree.

```python
from sklearn.datasets import load_iris
from sklearn import tree
iris = load_iris()
clf = tree.DecisionTreeClassifier()
clf = clf.fit(iris.data, iris.target)
```

Listing 2.1: Example of Python code for Decision Tree

2.7 Final Remarks

This chapter was created to provide the user with some basic knowledge about some of the most common concepts involving this dissertation. The notion and definition of what a time series is as well as what ways exist to analyze them is imperative for this dissertation. Also knowing the limitations of other types of solutions is useful in understanding the solution implemented. Lastly but not least, learning about the algorithms involved in the Classification/Regression methods and how these paradigms work and their difference is vastly relevant for this thesis.

Chapter 3

Implementation

The main goal of this chapter is to take a deeper look at the implementation of the solution used to solve the problem presented. Firstly, a brief description of the dataset will be utilized in order to understand better what kind of data and it's relevance in regards to the problem. Then a chapter about CART (Classification and Regression Trees) which was the main algorithm used in this project will be presented and explained so it's easier to understand the underlying logic of the solution.

3.1 Dataset

For this project work, considering the main objective is to measure the impact of climate variability in wine production, a large set of data is needed in order to establish a good analysis and identify subgroups of these objects. Considering Portugal is the fifth wine producer in Europe, with the Douro region being the most known one [CR12] there is an interest from both the technology part of the problem but also from the producers side in having more tools to being able to both: (1) identify subgroups and; (2) predict future productions. The data gathered is split into two components, the Wine Production data and the Meteorological data.

3.1.1 Wine Production Data

The production (in hL) for the Douro region was obtained through the *Instituto dos Vinhos do Douro Porto* (IVDP,2015), supplying the data for the years from 1933 to 2013. However, the wine yield estimations fails to account for the dynamics of new plantings, replanting, removal and age composition of the vineyard [CR16]. For this problem at hand, it is assumed that these new dynamics of young vineyards do not jeopardize the stability of the productivity link.

As seen on 3.1 there is a clear evidence that suggests an upward trend in the production of wine.

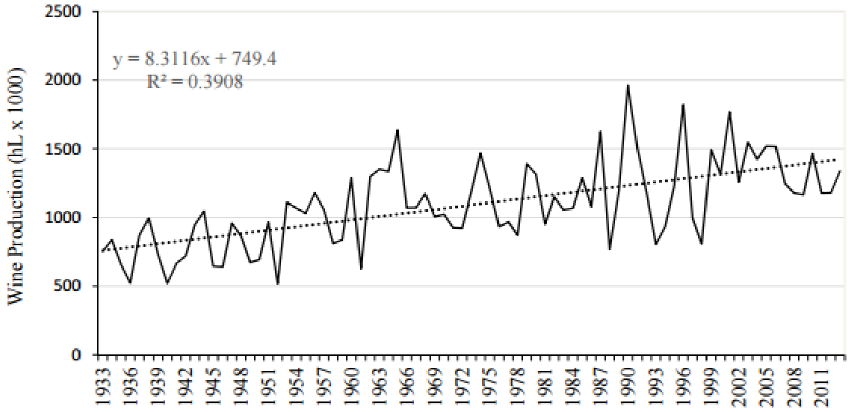


Figure 3.1: Time series and estimated linear trend of wine production for the period 1933 to 2013 in the Douro wine region [CR16]

3.1.2 Meteorological Data

The meteorological observations for the same years as the Production Data (1933-2013) were collected in the weather station of Peso Da Regua (41°10'N, 7°47'W), which is located in the Douro Region. The meteorological data consists of daily observations (365 days) and possesses 5 different features: Minimum Temperature, Maximum Temperature, Mean Temperature, Precipitation and available soil water. The daily climate data for the years 1933 to 1950 were obtained from the "Serviço Metereológico Nacional - Mapa de apuramento mensal". For this time period the data is complete and had no missing values, however the quality control used for climate data inspection isn't known. The remainder of the years (1950-2013) was subject to manual examination to complete some missing data from some of the years.

3.1.3 Final Remarks

The final remarks about the dataset and concurrently what also gives this project a factor of innovation is the fact that decision trees are being applied to time series instead of classical values. This factor creates some problems since with normal values, there is a linear way to compare, for example: 5.5 is lower than 9.1, however with time series, there is a need to create a method in which it is possible to say that, as an example, *Tmax2005* is lower than *Tmax1965*. This is one of the main focus of the project which is to apply the decision trees to these kinds of new problems.

3.2 Decision Trees

To tackle the issue of identifying subgroups among the dataset, the chosen solution revolves around the creation of a decision tree that sorts the different groups based on the several meteorological features. In this type of solution regarding decision trees, it can usually be split in Classification trees or Regression trees, which are the most common types of trees present in machine learning problems. The first one, requires that the target variable takes a finite set of values, which means there pre-defined labels for the outcome. On the other hand, in Regression trees, the target variable can take continuous values which are not pre-defined. Considering the target feature at hand (the quantity of wine produced), which can take continuous values, this is a classic problem where Regression Trees should be used instead of Classification. For the implementation of the Regression Tree, the CART (Classification and Regression Trees) algorithm, created by Leo Breiman, Jerome Friedman, Richard Olsen and Charles Stone in 1984, was used.

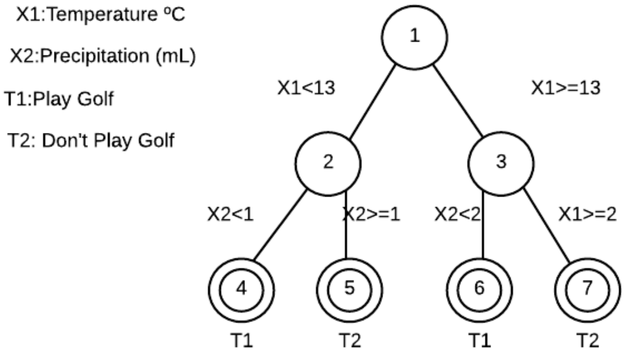


Figure 3.2: Example of a classification tree

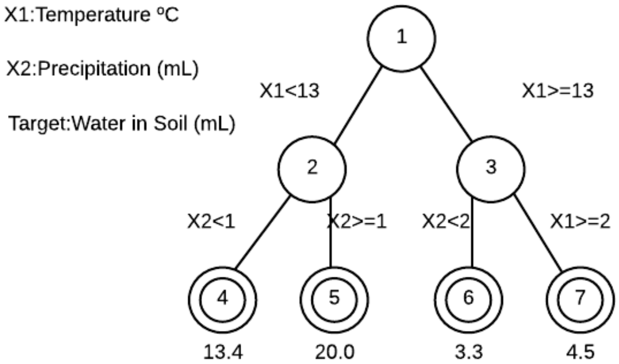


Figure 3.3: Example of a regression tree

3.2.1 CART

The basic idea of tree growing is choosing the best split value among all the possible splits for each node present in the tree, in order that the resulting child nodes are the purest i.e. lowest variance among the data present in that node. For this type of problems, a splitting criterion has to be chosen based on the type of data utilized, since the Y variable (target) is continuous, the splitting criterion should be one that is adequate to this. In CART, when Y is continuous, the split criterion used is the one present in 3.1. Utilizing the Least Squares Deviation (LSD) method to calculate the impurities.

$$\Delta\, i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{3.1}$$

In the 3.1 equation, the value for the node is given by calculating the impurity measure of the root node for that split *i(t)* and subtracting the impurities of both the left node and the right node multiplied by their proportions.

3.2.1.1 Least Squares Deviation

Least Squares Deviation is the method used in this solution to calculate the values of impurities for each possible split. Firstly it is needed to take a look at the main equation and interpret what each of the values represent.

$$i(t) = \frac{\sum\limits_{n \in h(t)} w_n f_n (y_n - \bar{y}(t))^2}{\sum\limits_{n \in h(t)} w_n f_n} \tag{3.2}$$

In this equation, the w_n refers to the weight each value of Y has, in this particular case, the value is equal for each instance, which equates to 1/n , where n is the total number of elements. Secondly, f_n value refers to the frequency a certain element appears in the whole set. For example, if the set is (1,2,3,1), the value for f_n in the first iteration would be 2, since the number 1 appears two times. For the expression within brackets y_n - $\bar{y}(t)$, the first element refers to the value of the element in the iteration, while the second is the average of all elements in the set, calculated through the expression in 3.3.

$$\bar{y}(t) = \frac{\sum\limits_{n \in h(t)} w_n f_n y_n}{N_w(t)} \tag{3.3}$$

where $N_w(t)$ is the number of elements calculated with the following expression 3.4:

$$N_w(t) = \sum_{n \in h(t)} w_n f_n \tag{3.4}$$

To complete the equation in 3.1, the only components missing are the proportions which are calculated through a simple division between the elements in both the left and the right nodes and all the elements involved in the split, represented respectively in 3.5 and 3.6.

$$p_L = \frac{N_w(t_L)}{N_w(t)} \tag{3.5}$$

$$p_R = \frac{N_w(t_R)}{N_w(t)} \tag{3.6}$$

3.2.2 Tree Growing and Stopping Proccess

The tree growing process is simple as soon as the method of split is chosen, however if the tree grows uncontrollably it will enter in a traditional problem of *overfitting*. Overfitting occurs when the tree model gets too complex which leads to poor predictive performance as it overreacts to minor alterations in the training data. There are several ways to prevent this phenomenon from happening, in this solution the methods chosen to stop the tree from growing are to limit the max depth of the tree, and also to stop splitting the node when it becomes pure.

3.3 Tree Implementation

The first step after studying the mathematical equations and understanding what they mean is to define a structure for a Node. Through that, it's possible to define a Tree as a set of nodes, where each node stores the connection to the child nodes, and also other useful information like: depth, feature of the split, value of the split, and the elements present in that node. This model is represented in 3.1

```python
class Node:
    def __init__(self,t,L,R,D,S,V,M,X):
        self.t=t
        self.L=L
        self.R=R
        self.D=D
        self.S=S
        self.V=V
        self.M=M
        self.X=X

#t Index of Node
#L Index of Left child
#R Index of right child
#D Depth of the Node
#S Value of split
#V Feature of split
#M Subset array
```

```
#X Execution Flag
```

Listing 3.1: Class for Node Structure

With this class, we can add to the tree in each iteration both child nodes of the Root node, based on what split criterion was used utilizing the method present in chapter 3.2.1.1. As soon as this is defined, a preliminary implementation of the tree was tested with regular data instead of time series, to verify if all the formulas were acting correctly and to also confirm if this implementation of the tree was valid and made sense to the user. Regarding the formulas present in the previous chapter, the results obtained for the variances of each node were compared to the python function *numpy.var* from the *numpy* library, and the outcomes gave exactly the same result in every experimentation. In order to better understand the proccess implemented, the main function will be presented and analyzed in A.1

```

def main(depth):
    temp=[]
    for index,node in enumerate(Tree):
        temp.append(Tree[index].D)

#Expand tree, using last node to compute split

    for index,node in enumerate(Tree):
  #Depth condition
        if (node.D==temp[-1] and node.D<=depth):
            if node.X<>1:
                if len(Tree[index].M)>1:
                    LSD(Tree[index].M)
                    #print sum(maxLSD[:,2])
                    if sum(maxLSD[:,2]) > 0 :
                        Build(Tree[index])
                        main(depth)
```

Listing 3.2: Main Function of preliminary implementation

The main function presented is the one called at the beginning of the program, and it takes 1 parameter which is the max depth of the tree. After that, it iterates through all the nodes in the List Tree, checking for each node if it should still be split or ignored through the *if (node.D==temp[-1] and node.D<=depth)* expression which checks for both max depth and if the depth present in the node is correct. If all the conditions pass, the impurities will be calculated in the *LSD* function in order to choose the split which will provide maximum gain. This is done recursively for each node, until either the max depth condition is met, or every leaf node has exactly one element.

After this experiment, there is a need to figure out how to work with time series instead of traditional data, since time series aren't easily comparable like numbers. However, the way to calculate the splitting criterion method is still the same since it applies only to the target variable, which has 1 element per set of time series.

```
[Production][TMAX][TMIN][TMED][PRECIP][SOIL]
```

Listing 3.3: Example of 1 Element

As seen in 3.3 instead of having 1 array of values leading up to a target result, the dataset has a conjunction of several arrays that leads to a target production. So in order to be able to solve this issue, first a metric to declare that a time series is lower or higher than the other has to be decided. This is essential since for the algorithm there needs to be a decision in sending elements to the left node and to the right node.

The method chosen in this solution was an alteration to the classic Euclidean Distance method that measures the absolute distance between two series, by summing up the square of difference between all points from the two series vertically. This method is very useful but it doesn't define if one time series is above or below the other. For that matter the alteration created was to still make the sum of the differences without squaring to verify the sign between the two series. With the sign known, the distance can be calculated normally through Euclidean Distance. The proccess in which the comparation of the two time series is made is present in the small piece of code 3.4

```

for y in range(len(a[i])):

    if(series1[i] - series2[i]) < 0):
      euc = euc - (series1[i] - series2[i])^2
    else:
      euc = euc + (series1[i] - series2[i])^2
```

Listing 3.4: Function used to perform the split

With both the splitting criterion implemented and a way to compare two time series, allowing the algorithm to choose which elements go to each side, the tree can be done traditionally following the CART algorithm allowing the identification of sub-groups that possess less variance than the root node.

3.3.1 Prediction Component

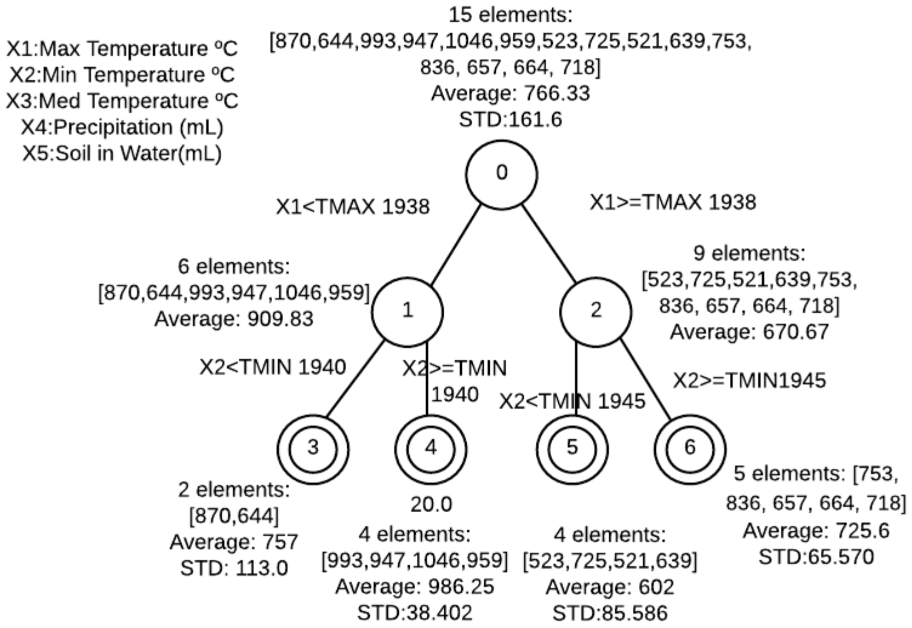
The tree implementation also gives the user the possibility for a prediction component. It is possible to split the dataset into training data and testing data. With that split it is possible to see for each instance of the testing data where it would fit in the tree created from the training data, and perform a "prediction" based on which node that instance ends up in. For that prediction number for production, the method chosen was to perform the average of productions of all the elements present in that leaf node.

Chapter 4

Result Analysis

4.1 Tree Analysis

Following the tree implementation described in chapter 3, the algorithm is now ready to perform experiments with the dataset. Firstly, a small subset of the data was chosen to verify if the results were acceptable and to take some conclusions on the effect that the several features had in the final outcome of the tree. The small subset included all the data from the first 15 years (1933-1947), and delimited by maxdepth = 2 produced the tree represented in 4.1.



X1:Max Temperature ºC
X2:Min Temperature ºC
X3:Med Temperature ºC
X4:Precipitation (mL)
X5:Soil in Water(mL)
15 elements: [870,644,993,947,1046,959,523,725,521,639,753, 836, 657, 664, 718] Average: 766.33 STD:161.6
0
X1<TMAX 1938
X1>=TMAX 1938
6 elements: [870,644,993,947,1046,959] Average: 909.83
1
2
9 elements: [523,725,521,639,753, 836, 657, 664, 718] Average: 670.67
X2<TMIN 1940
X2>=TMIN 1940
X2<TMIN 1945
X2>=TMIN1945
3
4
5
6
2 elements: [870,644] Average: 757 STD: 113.0
20.0
4 elements: [993,947,1046,959] Average: 986.25 STD:38.402
4 elements: [523,725,521,639] Average: 602 STD:85.586
5 elements: [753, 836, 657, 664, 718] Average: 725.6 STD:65.570

Figure 4.1: First experiment with first 15 years of dataset

Analyzing the tree, at first glance only two of the five features were used to perform all the splits. However as will be seen later in the analysis there is a reason for some of the features not being used compared to the others. As opposite to classical trees, it is possible to see that the splits are made not by numeric numbers but by time series i.e. *TMAX 1938* and as expected all the Standard deviations values on the leaf nodes are lower than in the root node, that is one of the indicators that the tree is valid since it is the main objective of a decision tree problem.

From the data itself and the way it is grouped in the leaf nodes, it is possible to verify that the highest productions of this dataset are being grouped into node 4, which leads to the conclusion that the conditions that nurture a higher production are when the Max Temperature time series is lower than *TMAX 1938* and the Minimum temperature time series is higher than *TMIN 1940*. By the same analysis inserted in node 5 are the 3 lowest productions [523,521,639], however there is also an element which seems that it does not fit totally into the set, the year in which the production equates to 725. This stems from a problem in which there is an assumption that the productivity of a year is directly correlated only to the meteorological conditions which is not entirely true. While features like temperature and precipitation are very important in the process of wine production, the lack of data regarding other issues like for example: diseases affecting the plants negatively or even just lower funds used by the companies, will lead to sporadic errors in the grouping of the data.

After performing this analysis on the small set of data, the next step is to use the full dataset (1933-2013) and verify if the same conclusions and assumptions from the previous experiment still stand. For that, there is the tree represented in 4.2 for this experiment.

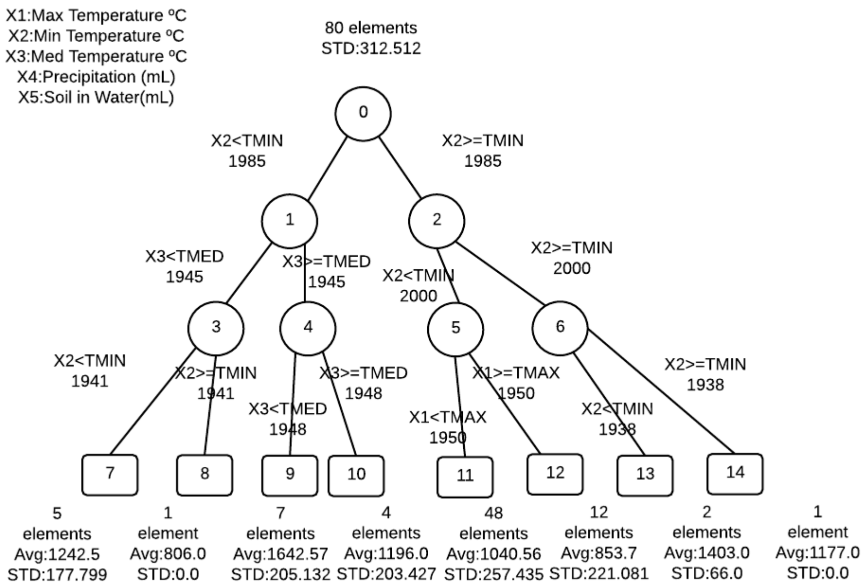


Figure 4.2: Second experiment with full dataset

At first glance, it is possible to see that compared to the first experiment, there were three features used to perform the splits (TMAX, TMIN and TMED), however the remaining two (Precipitation and Water in Soil) are not used, even if the depth of the tree is increased even further. Taking a look at the anatomy of time series of precipitation in 4.1, there are several days of the time series where the number is 0 as expected from a feature like precipitation since for a day where it doesn't rain the value would be 0. However from the Decision tree point of view, features where there are several elements with the same number, aren't usually chosen for the splits since they often offer a lesser reduction in variance for child nodes. This type of issues in datasets usually present in climate, ecological modeling and disease monitoring can sometimes degrades the overall quality of the model[AT]. However, in this particular model since it is never used to perform the splits it has no effect over the final result.

```
Year 1933:
7.2 0.2 5.4 2.6 0 0 0 0 0 0 0 0 0 0 4 14 1.2 2.6 1 0 0 0 0 0 0 1.2 3 26 22 0.6 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0.1 0.2 0 0 0 0 0 0 0 0 0 2.1 6.2 0 0 0 4.7 4.8 18.4
    5.2 4.4 13 0 0 0 12 15.4 0 0 0 0 10.4 3.6 7.6 0.6 0 0 0.8 5 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 1.4 2 0 0 0 0 0 0 5.4 11 0 0 0 0 0 0 0 0 0 0 8 6.2 1.6 3.8 3.4
     1.6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3.8 0 1.6 1.4 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0.6 4.6 0 0 0 0 0 0 0 0 1.2 1.2 14.4 8.6 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.6 0 1 1.2 0 3 0 0 0 0 0 0 0 0 0 2
    0.2 0 0 2.6 0.6 0 4 0 0.1 3 0 0 0 4.8 8.2 0 0 0 0 0 0 10.2 5.8 8.4 0 0 0.5 0 0
    0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 4.2 4.4 0 0 1.6 19.8 4.4 2 2.4 3.8 0.2 0 0 0 0 0
    0 7.2 11.8 0 1.4 16.4 0 0 0 0 0 3.8 0 0 0 0 1.5 1.9 0 0 0 0 0 0 0 0 0 0 6.4
    22.2 12 18.2 28 0 0 0
```

Listing 4.1: Example of 1 precipitation time series

Taking the first iteration of the algorithm as example, and comparing the gain values from the best year for each feature, it is possible to see a discrepancy between the chosen split and the best split for precipitation.

```
Chosen Split Gain: 1.18537038e+04
Best Precipitation Split Gain: 3.21485732e+03
```

Listing 4.2: Comparing gains based on splits on the 1st Experiment

By looking at the leaf nodes, it is possible to identify the same way as was made in the first experiment, the node that has the highest production attached to it (node 9). However, the results seem to not make as much sense as it did in the first experiment with less data, despite all of the result nodes having a standard deviation lower than the root node. This phenomenon can be attributed to a factor which is lack of data regarding other components that influence production. By trying to establish a correlation between the climate and the production of wine, even though they are correlated, factors like the advance in technology in this area are being disregarded. That is, for example, assuming that in the year 2013, the meteorological conditions were much more adverse than in the year 1943 and despite that the production in 2013 was much higher than in the earlier year, these types of situations will disrupt the model since the decision tree is not taking into account this type of data that is not being taken into consideration, leading to sometimes unsatisfactory results.

Continuing what was said in the last paragraph, it is therefore logical that in the smaller dataset with 15 years the results would be much more reliable since in 15 years the technological advancements would not interfere as much as they would in a 80 years interval. Other possible reasons for this discrepancy could be the increase in the vineyard's area which will in consequence produce more wine.

4.2 Prediction Analysis

After a better understanding of the tree and the subgroups it discovered, it is possible to perform prediction regarding new sets of data. As mentioned in the end of chapter 3, it is possible to insert new data, see in which leaf node it lands and make a prediction based on the rest of the elements present in that node.

Firstly, taking the small subset of the first 15 years, as the training set, and the next 5 years (1948-1952) as the testing set. it is possible to verify in 4.4 the results for a tree with depth 3.

```
Year: 1948, Real Production: 863, Prediction: 870.0
Year: 1949, Real Production: 669, Prediction: 870.0
Year: 1950, Real Production: 692, Prediction: 794.5
Year: 1951, Real Production: 966, Prediction: 1046.0
Year: 1952, Real Production: 518, Prediction: 644.0
Root Mean Squared Error: 121.023
```

Listing 4.3: Prediction results of First Experiment

As expected from the model, the predictions are not very precise, since there is an assumption which states that the wine production of a year is directly correlated only with the meteorologic data of that year. This level of mismatch will only get worse the more years included because as was said before, the technological advances will have an even higher effect the more years that are included into the dataset leading to an even lesser reliable model for prediction. To prove that fact, there was an experiment done with 76 years as the training set, and the 5 remaining years chosen randomly from the full dataset(81 elements), as the testing set.

```
Year: 1971, Real Production: 925, Prediction: 1181.75
Year: 1994, Real Production: 932, Prediction: 1284.0
Year: 1956, Real Production: 1177, Prediction: 1403
Year: 1962, Real Production: 1297, Prediction: 892.0
Year: 2005, Real Production: 1520, Prediction: 1049.0
Root Mean Squared Error: 251.49
```

Listing 4.4: Prediction results of Second Experiment

In this second experiment, the predictions are worse than on the first one despite having more data to rely on. That factor comes from what was mentioned before about the fact that the model is being created with missing features that have a huge impact in the target variable. For that reason it is concluded that prediction is not reliable with the amount of information provided since the correlation between the Meteorologic and the Target variable are not strong enough to overshadow the missing data.

4.3 Final Remarks

From analyzing all the results, it is possible to conclude that the model does indeed create subgroups with considerable less standard deviation than the root node, which makes sense in the decision trees paradigm. There is also an interesting fact to take about data with several elements that are equal (Precipitation and Water in soil) in which it seems they produce lesser values in gains compared to other features that possess more diversity.

Lastly, the amount of missing features like technological advancements and the assumption that the production of wine is only related to the meteorological conditions which is not true make the model not reliable for prediction in instances where the missing features have a great effect.

Result Analysis

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This area of utilizing Classification or Regression Trees to treat Time series data is still in its early stages, and so the model proposed in this dissertation has a goal to presenting a new alternative to the ones already existent. This solution presents itself as a way to give the user an easy to interpret model that can be understood by even people outside of the computer science field. For that, professionals in the medical field, or in the stock markets can make use of this decision Tree model to better understand the data and perform predictions.

Considering that the main objective of this paper was to implement a model that utilized Decision Trees to perform subgroup-discovery of Multivariate Time Series, it is concluded that this project was completed with success. Adding to this fact, there was a small prediction component made based on the Decision Tree output that went beyond the requirements proposed for this dissertation. However, there were some other aspects that could have been improved like the implementation of more than one split criterion or the study of the Meteorological Conditions trends to further prevent some of the errors from missing features. Despite those factors, this model can be adapted to other types of data as long as they all possess the same size and are composed of int or float values.

The main motivation of this work was to further attempt to study this new area with the implementation of an innovative model that could also have uses in real life for other areas, as in my opinion the ability to merge the computer science area with others is a great use of technical skills. Besides that, this new model can be used to further develop the advancements in this area in the search for something even more efficient.

5.2 Future Work

For future work, the study of alternative methods for comparing series to each other in order to perform a better split of the data like an alteration to the DTW method or other new criterions would be interesting additions to this project.

This dissertation was more focused in implementing a model that utilized the data from the meteorological conditions which all were time series composed by floats. So another step that could be taken for this thesis would be to adapt the model to be able to treat also qualitative, quantitative attributes among others. Also within this category, it would be possible to have annual measures in the same set as daily measures.

Lastly, there could be an option for the user of this model to specify what kind of pruning method he thinks would be more appropriate, instead of using max-depth method for all experiments.

References

[AB] Manish Amde and Joseph Bradley. Scalable decision trees in mllib. September 29, 2014. Available at https://databricks.com/blog/2014/09/29/scalable-decision-trees-in-mllib.html.

[AT] Zubin Abraham and Pang-Ning Tan. An integrated framework for simultaneous classification and regression of time-series data. Available at http://www.cse.msu.edu/~ptan/papers/SDM10.pdf.

[CK15] Sneha Chandra and Maneet Kaur. Creation of an adaptive classifier to enhance the classification accuracy of existing classification algorithms in the field of medical data mining. *2nd International Conference on Computing for Sustainable Global Development (INDIACom),New Delhi*, pages 376–381, March 2015.

[CMA$^+$12] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. *Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining*. Intech, 2012.

[CR12] Mário Cunha and Christian Richter. Measuring the impact of temperature changes on the wine production in the douro region using the short time fourier transform. *International Journal of Biometeorology*, 56(2):357–370, 2012.

[CR16] Mario Cunha and Christian Richter. The impact of climate change on the winegrape vineyards of the portuguese douro region. *Climatic Change*, pages 1–13, 2016.

[GF] Dimitrios Gunopulos and Christos Faloutsos. Indexing time series. Available at http://www.cs.bu.edu/~gkollios/dm07/LectNotes/TSIndexing.ppt.

[K$^+$10] Michael W Kattan et al. Classification and regression trees versus nomograms: a bone scan positivity example. *European urology*, 57(4):559–560, 2010.

[Kad99] Mohammed Waleed Kadous. Learning comprehensible descriptions of multivariate time series. In *ICML*, pages 454–463, 1999.

[Li15] Hailin Li. On-line and dynamic time warping for time series data mining. *International Journal of Machine Learning and Cybernetics*, 6(1):145–153, 2015.

[LKT03] Jessica Lin, Eamonn Keogh, and Wagner Truppel. Clustering of streaming time series is meaningless. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 56–65, New York, NY, USA, 2003. ACM.

[MHOV12] Floréal Morandat, Brandon Hill, Leo Osvald, and Jan Vitek. *ECOOP 2012 – Object-Oriented Programming: 26th European Conference, Evaluating the Design of the R Language, Beijing, China, June 11-16, 2012. Proceedings*. Springer Berlin Heidelberg, 2012.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[SMMA16] I.S. Saeh, M.W. Mustafa, Y.S. Mohammed, and M. Almaktar. Static security classification and evaluation classifier design in electric power grid with presence of {PV} power plants using c-4.5. *Renewable and Sustainable Energy Reviews*, 56:283 – 290, 2016.

[Vin] ViniPortugal. Available at http://www.viniportugal.pt/OSector.

[YK11] Lexiang Ye and Eamonn Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery*, 22(1):149–182, 2011.

[YSYT03] Yuu Yamada, Einoshin Suzuki, Hideto Yokoi, and Katsuhiko Takabayashi. Decision-tree induction from time-series data based on a standard-example split test. In *ICML*, volume 3, pages 840–847, 2003.

Appendix A

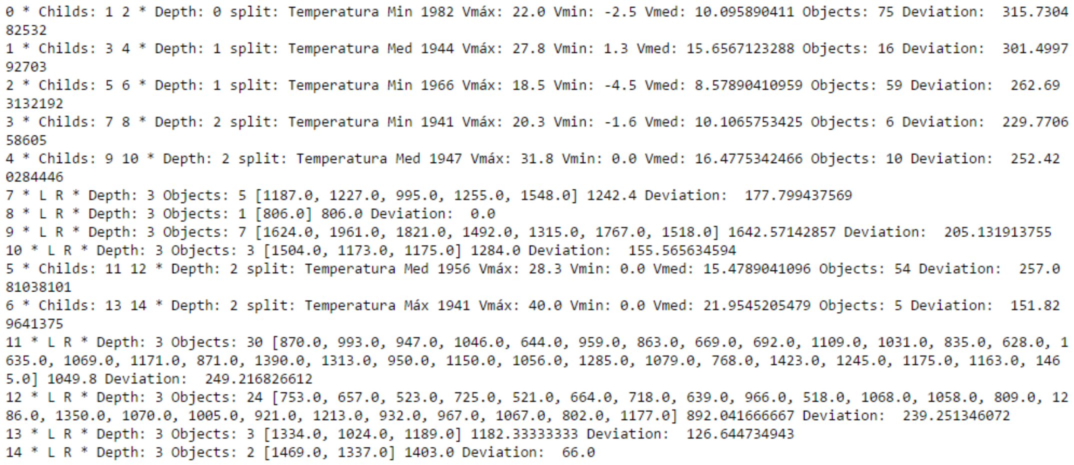
Implementation

A.1 Extract of code that performs Prediction

```

def Prediction(series,Tree,a):

    next_node = 0


    for i,node in enumerate(Tree):


        if (node.t == next_node):

            if(node.V  == "V"):
                return

            result = compare_series(node.M[1][int(node.V)][int(node.S)],series[1][node.V])
            if (result < 0):
                next_node = node.L
            else:
                next_node = node.R

        if(next_node == "L"):
            return np.mean(node.M[0])
        if(next_node == "R"):
            return np.mean(node.M[0])
```

Listing A.1: Main Function of preliminary implementation

A.2 Text Output of Tree



```
0 * Childs: 1 2 * Depth: 0 split: Temperatura Min 1982 Vmáx: 22.0 Vmin: -2.5 Vmed: 10.095890411 Objects: 75 Deviation:  315.7304
82532
1 * Childs: 3 4 * Depth: 1 split: Temperatura Med 1944 Vmáx: 27.8 Vmin: 1.3 Vmed: 15.6567123288 Objects: 16 Deviation:  301.4997
92703
2 * Childs: 5 6 * Depth: 1 split: Temperatura Min 1966 Vmáx: 18.5 Vmin: -4.5 Vmed: 8.57890410959 Objects: 59 Deviation:  262.69
3132192
3 * Childs: 7 8 * Depth: 2 split: Temperatura Min 1941 Vmáx: 20.3 Vmin: -1.6 Vmed: 10.1065753425 Objects: 6 Deviation:  229.7706
58605
4 * Childs: 9 10 * Depth: 2 split: Temperatura Med 1947 Vmáx: 31.8 Vmin: 0.0 Vmed: 16.4775342466 Objects: 10 Deviation:  252.42
0284446
7 * L R * Depth: 3 Objects: 5 [1187.0, 1227.0, 995.0, 1255.0, 1548.0] 1242.4 Deviation:  177.799437569
8 * L R * Depth: 3 Objects: 1 [806.0] 806.0 Deviation:  0.0
9 * L R * Depth: 3 Objects: 7 [1624.0, 1961.0, 1821.0, 1492.0, 1315.0, 1767.0, 1518.0] 1642.57142857 Deviation:  205.131913755
10 * L R * Depth: 3 Objects: 3 [1504.0, 1173.0, 1175.0] 1284.0 Deviation:  155.565634594
5 * Childs: 11 12 * Depth: 2 split: Temperatura Med 1956 Vmáx: 28.3 Vmin: 0.0 Vmed: 15.4789041096 Objects: 54 Deviation:  257.0
81038101
6 * Childs: 13 14 * Depth: 2 split: Temperatura Máx 1941 Vmáx: 40.0 Vmin: 0.0 Vmed: 21.9545205479 Objects: 5 Deviation:  151.82
9641375
11 * L R * Depth: 3 Objects: 30 [870.0, 993.0, 947.0, 1046.0, 644.0, 959.0, 863.0, 669.0, 692.0, 1109.0, 1031.0, 835.0, 628.0, 1
635.0, 1069.0, 1171.0, 871.0, 1390.0, 1313.0, 950.0, 1150.0, 1056.0, 1285.0, 1079.0, 768.0, 1423.0, 1245.0, 1175.0, 1163.0, 146
5.0] 1049.8 Deviation:  249.216826612
12 * L R * Depth: 3 Objects: 24 [753.0, 657.0, 523.0, 725.0, 521.0, 664.0, 718.0, 639.0, 966.0, 518.0, 1068.0, 1058.0, 809.0, 12
86.0, 1350.0, 1070.0, 1005.0, 921.0, 1213.0, 932.0, 967.0, 1067.0, 802.0, 1177.0] 892.041666667 Deviation:  239.251346072
13 * L R * Depth: 3 Objects: 3 [1334.0, 1024.0, 1189.0] 1182.33333333 Deviation:  126.644734943
14 * L R * Depth: 3 Objects: 2 [1469.0, 1337.0] 1403.0 Deviation:  66.0
```

Figure A.1: Output Produced by Code