Comparative study of machine learning algorithms in economics's interest time series's data

Michail Anagnostakis RN : 245

July 22, 2022

Supervisor : Associate Professor Tsiotas G.
Member of the board : Associate Professor Genius M.
Member of the board : Assistant Professor Tsagris M.

ABSTRACT

At the first part of this master thesis, initially we will explain neural networks and KNN, the theoretical background of applying k-nearest neighbor to time series forecasting. Afterwards we will make an application in monthly TOYOTA car sales (registrations) in Greece, data from AMVIR from January 2010 until and October 2021. We generate multiple KNN regression models, in order to make comparison. The simpler of them, with k=3, k=2, k=1, produce encouraging predictive results for our data set. In addition we generate an exponential smoothing model as a strict comparing basis for the kNN regression models.

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή διερευνά τα παρακάτω ερευνητικά ερωτήματα :

1. Είναι δυνατο να χρησιμοποιήσουμε τεχνικές μηχανικής μάϑησης σε δεδομένα χρονολογικών σειρών ώστε να κάνουμε πρόβλεψη και πιο συγκεκριμένα τον KNN αλγόριϑμο ;

2. Ποια μεϑοδολογία ϑα χρησιμοποιήσουμε και ποιος ϑα είναι ο αριϑμός των k ;

3. Το αποτέλεσμα ϑα ειναι αξιόπιστο, σε σύγκριση με άλλες οικονομετρικές τεχνικές ;

Τα στατιστικά νευρωνικά δίκτυα βασίζονται σε στατιστικές μεϑόδους και τη ϑεωρία πιϑανοτήτων και είναι ένας πολύ δημοφιλής και σημαντικός τύπος νευρωνικού δικτύου. Οι τρεις κύριοι τύποι αυτών των δικτύων ειναι τα RBFNNs, τα PNNs και τα GRNNs.

Ο KNN αλγόριϑμος δεν μαϑαίνει μια διακριτική λειτουργία από τα δεδομένα προς εκπαίδευση, αλλά αντί αυτού ϑυμάται τα προς εκπαίδευση δεδομένα. Στον KNN δεν υπάρχει χρόνος εκπαίδευσης, ούτε φάση εκπαίδευσης. Ο KNN αλγόριϑμος ϑυμάται παραδείγματα εκπαίδευσης αντί να μπει στον κόπο να μον-

τελοποιήσει τα δεδομένα. Ο KNN αποδίδει καλά όταν έχουμε πολλές περιπτώσεις (σημεία) και λίγες διαστάσεις. Ο KNN ανήκει στα νευρωνικά δίκτυα παλινδρόμισης και πιο συγκεκριμένα στα δίκτυα μη παραμετρικής παλινδρόμισης και ταξινόμησης.

Συχνά στην πραγματική ζωή τα φαινόμενα δεν μπορούν να μοντελοποιηθούν σε ικανοποιητικό βαθμό χρησιμοποιώντας γραμμικά μοντέλα. Έτσι χρειαζόμαστε τα μη γραμμικά μοντέλα. Βάση αυτού, εδώ το πρόβλημα είναι πως και τι μη γραμμική συνάρτηση να διαλέξουμε. Αυτή μπορεί να είναι γραμμική συνάρτηση, ημίτονα, συνημίτονα, αθροίσματα και άλλα. Έτσι ότι επιλέξουμε έχει ως συνέπεια να επηρεάζει το αποτέλεσμα της μοντελοποίησης. Σε πολλές περιπτώσεις δεν γνωρίζουμε πολλά για την υπόγεια φύση της διαδικασίας που πρόκειται να μοντελοποιηθεί, ή η μοντελοποίηση είναι δύσκολο να προσδιοριστεί με ακρίβεια. Δεν υπάρχουν πολλές επιλογές στη μηχανική μάθηση που χρησιμοποιούνται με ικανοποιητικά αποτελέσματα στη μοντελοποίηση για διάφορα προβλήματα. Αυτές οι μέθοδοι είναι η παλινδρόμηση ακτινωτής συνάρτησης, τα τεχνητά νευρωνικά δίκτυα και ο KNN αλγόριθμος.

Ο αλγόριθμος των K κοντινότερων γειτονικών παρατηρήσεων (KNN) είναι μια μη γραμμική μέθοδος τεχνητής νοημοσύνης η οποια χρησιμοποιεί ένα παράδειγμα - μέτρο, ώστε να βρει τις K πιο σχετικές παρατηρήσεις στο σετ δεδομένων προς εκπαίδευση, για μια νέα παρατήρηση και παίρνει ως πρόβλεψη το μέσο αποτέλεσμα των γειτονικών παρατηρήσεων.

Για να εφαρμόσουμε την KNN παλινδρόμηση σε ένα μονομεταβλητό υπόδειγμα χρονολογικών σειρών, ώστε να κανουμε πρόβλεψη, οι ερμηνευτικές μεταβλητές είναι μεταβλητές με υστέρηση των ερμηνευόμενων μεταβλητών ή μεταβλητών προς πρόβλεψη. Η κεντρική ιδέα ώστε να χρησιμοποιήσουμε τον KNN για πρόβλεψη χρονολογικών σειρών, είναι ότι οποιαδήποτε χρονολογική σειρά περιλαμβάνει επαναλαμβανόμενα μοτίβα, δηλαδή μπορούμε να βρουμε προηγούμενα παρόμοια μοτίβα στην υπάρχουσα δομή των σειρών και να χρησιμοποιήσουμε τα μεταγενέστερα μοτίβα ώστε να προβλέψουμε τη μελλοντική συμπεριφορά.

Υπάρχουν τρεις στρατηγικές για να επιλέξουμε την τιμη του K. Η πρώτη είναι να θεσουμε το K ίσο με την τετραγωνική ρίζα του αριθμού των παρατηρήσεων προς εκπαίδευση. Η δεύτερη ειναι το να χωρίσουμε το σετ προς εκπαίδευση σε ένα σετ εκπαίδευσης και ένα σετ επικύρωσης. Με αυτόν τον τρόπο επιλέγουμε το K έτσι ώστε αυτό να ελαχιστοποιεί ένα μέτρο ακρίβειας της πρόβλεψης για το σετ επικύρωσης, ενώ χρησιμοποιούμε τα προς εκπαίδευση δεδομένα. Ένα μειονέκτημα αυτής της στρατηγικής είναι ότι είναι χρονοβόρα. Η τρίτη στρατηγική προσπαθεί να ισορροπήσει τα πλεονεκτήματα της αποτελεσματικότητας και της σχολαστικής έρευνας διαμέσου συνδυασμών των μοντέλων πρόβλεψης χρονολογικών σειρών. Από αυτό, αυτή συνδυάζει τις προβλέψεις διάφορων KNN μοντέλων, με διαφορετικές τιμές K, ώστε να παράξουν μια τελική πρόβλεψη ως τη μέση πρόβλεψη των διάφορων μοντέλων.

Εάν η τιμη του K είναι πολύ μεγάλη ή πολύ μικρή αυτό θα αυξήσει την παρέμβαση στα δεδομένα και θα μειώσει την ακρίβεια της ταξινόμησης. Στην περίπτωση που το K είναι μικρό, το μοντέλο θα είναι πιο περίπλοκο και θα υπάρξει μια αύξηση στα σφάλματα εκτίμησης. Συνεπώς τα αποτελέσματα των προβλέψεων είναι ευαίσθητα στα γειτονικά σημεία των δεδομένων.

Αντίθετα, εάν το K είναι μεγάλο, αυτό μπορεί να μειώσει τα σφάλματα εκτίμησης, όμως τα κατα προσέγγιση σφάλματα θα μπορούσαν ταυτόχρονα να αυξηθούν και τα προς εκπαίδευση σημεία δεδομένων, μακρύα από τα σημεία δεδομένων εισόδου θα μπορούσαν επίσης να επηρεάσουν τα αποτελέσματα των προβλέψεων. Επομένως σε γενικές εφαρμογές του KNN αλγορίθμου, την τιμή του K τη θέτουμε ίση με μια σχετικά μικρή τιμή , αλλά πρέπει να ειναι ακέραιος αριθμός.

Οι επιλεγμένες τεχνικές προεπεξεργασίας για την εφαρμογή στα δεδομένα ταξινομήσεων αυτοκινήτων ειναι οι παρακάτω :

1. Δεν επιλεξαμε κανονικοποίηση των δεδομένων

2. Χρησιμοποιησαμε τεχνικές εύρεσης και προσαρμογής των ακραίων τιμών.

3. Δεν χρησιμοποιήσαμε μετασχηματισμούς Box - Cox.

4. Δεν χρησιμοποιήσαμε τεχνικές για να αφαιρεσουμε την τάση.

5. Δεν χρησιμοποιησαμε τεχνικές που αφαιρούν την εποχικές διακυμάνσεις.

Έπειτα κανουμε μια εφαρμογή σε μηνιαία δεδομένα ταξινομήσεων αυτοκινήτων TOYOTA στην Ελλάδα, στοιχεία αντλημένα απο το ΣΕΑΑ, από τον Ιανουάριο του 2010 ως τον Οκτώβρη 2021. Παράγουμε διάφορα μοντέλα KNN παλινδρόμησης για k=1, k=2, k=3, k=5, k=12, k=24 και ένα διάνυσμα με k=3, 5, 7, με σκοπό να τα συγκρίνουμε. Τα απλούστερα από αυτά , με k=3, k=2, k=1, έχουν ενθαρυντικές προβλεπτικές ικανότητες για τα δεδομένα μας. Παρατηρούμε ότι όσο το k αυξάνεται (k>3) τα αποτέλεσματα γίνονται αμφιλεγόμενα κοιτώντας τις προβλέψεις των τιμών και την ακρίβεια αυτών. Επιπρόσθετα, τρέξαμε ένα καθαρά οικονομετρικό μοντέλο εκθετικής εξομάλυνσης, ως ένα αυστηρό συγκριτικό κριτήριο για τα KNN μοντέλα.

Contents

INTRODUCTION

The k-nearest neighbors (or KNN) algorithm is an , non linear , artificial intelligence method that uses an instance measure in order to find the k most relative observations in the training data for a new observation and takes the mean outcome of the neighbor as the prediction.

Given some features or explanatory variables of a new instance to be regressed on the k-NN finds the k training instances that are closest to the new instance according to some distance metric and returns their majority class or average explained variable.

This master thesis is organized as follows. In Part I we explain neural networks and KNN , in Part II will be described a methodology about applying k-nearest neighbor in time series forecasting according Francisco Martinez, Maria Pilar Frias, Maria Dolorez Perez, Antonio Jesus Rivera (2017). In part III we will explain the modeling techniques in order to model via KNN. In part IV we explain the econometric technique we used in order to make comparison, in Part V we analyze our data set. In part VI we generate our applications and finally in Part VII we make our comparative comments.

Part I

Presentation of a methodology for applying k-nearest neighbor to time series forecasting. Theoretical background.

1 Statistical neural networks

Statistical neural networks are based on statistical methods and probability theory and are a very popular and important type of neural network. The three main types of these networks are: the Radial Basis Function Neural Networks (RBFNNs), the Probabilistic Neural Networks (ΠΝΝς), and the General Regression Neural Networks (GRNNs). Tsiotas G. (personal contact, 15/04/2022)

2 KNN and neural networks

KNN does not learn a discriminating function from the training data, but memorizes the training data set instead. There is no training time in KNN, no training phase, it memorizes training examples rather that putting in the effort to model the data. KNN performs well when we have many instances (points) and few dimensions. KNN belongs to Regression neural networks and more specifically to the non parametric regression and classification networks.

Nonlinear regression

Hertzmann and Fleet (2010) support that often in real life phenomena cannot be modeled in a great way using linear models, thus we need non linear models. Hence, the problem here is how and what non linear function to choose. It can be linear functions, sines and cosines, summations, etc. So what we choose affects the result of the effectiveness of the modeling. In a lot of cases we don't know a lot about the underlying nature of the process being modeled or modeling is difficult to precise. There are not a lot of options in machine learning that are used with satisfactory results in modeling for a variety of problems. These methods are basis function regression, artificial neural networks and k- Nearest Neighbors.

2.1 Basis function regression

A basis function for one dimension is

$$y = f(x) = \sum_k w_k b_k(x) \tag{1}$$

Basis functions are represented from the functions b$_k$(x).

Using vectors we can rewrite the formula :

$$y = f(x) = b(x)^T w \tag{2}$$

where $b(x) = [b_1(x), ..., b_M(x)]^T$, $w = [w_1, ..., w_M]^T$ and M is the number of basis functions.

Polynomials and Radial Basis Functions (RBF) are two often used basis functions. Simpler forms are the monomials such as $b_0(x) = 1$, $b_1(x) = x$,$b_2(x) = x^2$, etc

Using monomials the regression model is expressed :

$$f(x) = \sum w_k x^k \tag{3}$$

and gives the Radial Basis Functions :

$$b_k(x) = e^{-\frac{(x-c_k)^2}{2\sigma^2}} \tag{4}$$

and

$$f(x) = \sum w_k e^{-\frac{(x-c_k)^2}{2\sigma^2}} \tag{5}$$

is the regression model.

Where c$_k$ stands for the center of the basis function and σ^2stands for the width of the basis function.

2.2 Artificial Neural Networks

When we choose a sigmoid function as basis function we usually choose the function :

$$g(a) = \frac{1}{1+e^{-a}} \tag{6}$$

and when we combine sigmoids we have as a result an artificial neural network (ANN).

For one dimension ANN the model is

$$y = f(x) = \sum_j wj^{(1)} g(w_j^{(2)} x + b_j^{(2)}) + b^{(1)} \tag{7}$$

where w$_j^{(1)}$ are the outer weights and w$_j^{(2)}$are the inner weights.

2.3 K - Nearest Neighbors

A researcher who does not want to decide about the locations of the basis functions and other sensitive training data smoothing techniques, can opt for the KNN regression method. The only decision he has to make, is the number of k.

3 Theoretical Background - Classification and Regression with KNN

Imandoust et al (2013) support that in order to discover patterns and relationships inside data, we perform data mining that takes advantage of a lot of data analysis tools, aiming for successful predictions. Some tools that data mining uses are artificial intelligence techniques and statistics in order to solve problems of pattern recognition and classification. There are two categories in order to have a forecast with data mining techniques. The first is classification that forecasts the category in which a case belongs. The second is regression which performs a forecast on what scalar value will be for a variable or a time series.

With K-Nearest Neighbor method is possible to generate both classification and regression models as well.

This technique of classification, classifies each instance similarly to the instances near it. Hence, it can be classified based on the classification of its surrounding instances, if we don't know what classification class it belongs. That is the notion of this classification technique and the way it does forecast.

In practice the KNN algorithm calculates the distances between the unknown instances and the known ones (training set). Thus, the unknown instances can be classified according to the smallest distances of their nearest neighbors.

Both following figures explain the KNN decision's rules according to Imandust et al (2013) :

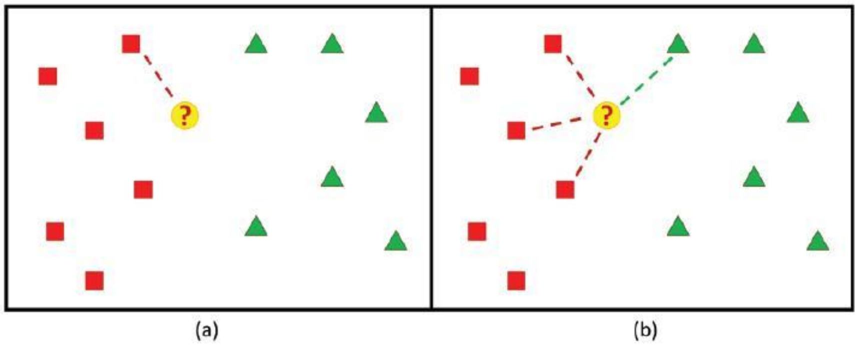


FIGURE 1: classification KNN rule for (a) K=1, (b) K=4

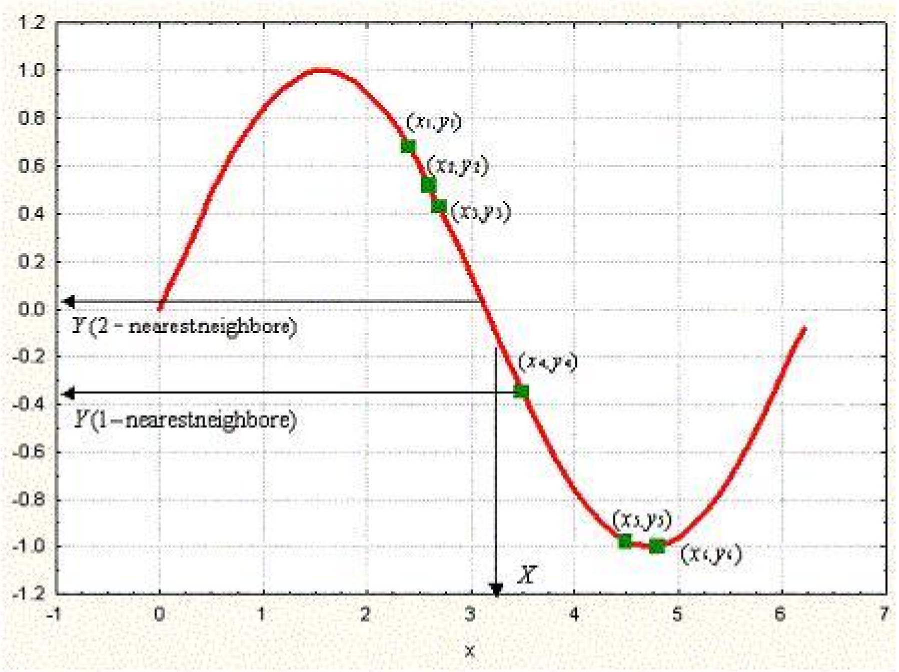


Figure 2: The KNN decision rule for regression

The KNN model's performance mainly depends on the number of the selected K. In addition with the work of S.B. Imandoust et al (2013) and Gaganis and Zopounidis (2008) that argue on the use of small and large K, in this study we are going to to test and the thumbnail rule, which is K = $\sqrt{n}$, where n = the number of observations. We can use the same method, except for classification, for regression by giving the property value for the object, equal to the average of the values of K nearest neighbors. Proceeding on Imandust et al (2013) the main advantages of KNN are " simplicity , effectiveness, intuitiveness and competitive classification performance." Also robustness to noisy data and effectiveness to large training data.

The main disadvantages of KNN are : poor run-time performance in large training sets , KNN is very sensitive to irrelevant features. It also needs high computation cost and time. Finally, the KNN techniques can be applied in a big variety of applications of real life.Varian (2014) supports that machine learning techniques can be improved from various econometric techniques such as causal modeling, non IID data, causal inference, confounding variables, natural and explicit experiments, regression discontinuity, difference in differences and instrumental variables. On the other hand Varian (2014) supports that econometrics can be improved from machine learning with techniques as : " Big data, train- test -validate to avoid over fitting, cross validation, non linear estimation, bootstrap, model averaging, Bayesian methods etc. "

4 The K-NN algorithm

Gaganis and Zopounidis (2008) state clearly the K-NN algorithm for classification : "

> Assuming that each alternative $\{\overrightarrow{x}, C\}$ that exists in the training sample includes n input features that are described from the vector $<(g_1(x), g_2(x), ..., g_m(x))>$ and f: R^n→ C, C is a function with C a limited set$\{C_1, C_2, ..., C_q\}$ of discrete categories. Then the euclidean distance between the two alternatives$\{x_1, x_q\}$ equals to : d$(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{m}(g_r(x_i) - g_r(x_j))^2}$
>
> The KNN algorithm follows three stages :
>
> i. The training algorithm.

For each alternative of the training sample <x, f(x)> it adds the alternative in the list with the example of training sample.

ii. The classification algorithm

In order to classify an alternative for estimation x$_q$. If the set of the alternatives in the learning sample is x$_1$,x$_2$,...,x$_k$then find K alternatives of the learning sample that are nearest to x$_q$.

iii. Return

$$\hat{f}(x) \leftarrow argmax_{c\in C}\sum_{i=1}^{k}\delta(c, f\,(x_i))$$

where δ$(a, b))$ = 1if a=b, else δ(a,b))=0

"

Part II

Towards an automated methodology for applying K-NN to time series

F. Martinez et al (2017) support that in order to apply the KNN regression in a uni variate time series forecasting setting, the explanatory variables are lagged values of the explained or forecast variable. The notion to use KNN for forecasting time series is that any time series contains repetitive patterns, so we can find previous similar patterns to the current series structure and use their subsequent patterns to predict the future behavior.

5 First stage : Preprocessing task

In order to improve the forecast accuracy of a KNN regression model we may apply the following strategies :

5.1 Normalization

According Google developers :”

The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model. Four common normalization techniques may be useful:

a. scaling to a range, b. clipping, c. log scaling, d. z-score. “

Normalization is applied in order the distance metric is not to biased towards attributes that have larger scales of measurement. However, as all the observations of a time series are in the same scale we consider that normalization is not necessary for doing KNN regression on a time series.

5.2 Outliers

According the Engineering Statistics Handbook: “

Definition of outliers : An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Outliers may contain important information. Outliers should be investigated carefully. Often they contain valuable information about the process under investigation or the data gathering and recording process. Before considering the possible elimination of these points from the data, one should try to understand why they appeared and whether it is likely similar values will continue to appear. Of course, outliers are often bad data points. “

Outliers are extreme observations that, if not adjusted, may cause serious estimation and forecast errors. An extreme observation should be adjusted by an expert after careful thought depending on whether the extreme phenomenon can happen again in the future. However, a useful heuristic tool for detect and adjust outliers is the following :

an observation is deemed an outlier if its absolute value is four times greater than the absolute medians of the three consecutive points before and after the observation.

$$|o_i| \geq 4xmax\left\{|m_b|, |m_a|\right\} \tag{8}$$

where m$_b$= median(o$_{i-3}, o_{i-2}, o_{i-1}$) and m$_a$=median(o$_{i+1}, o_{i+2}, o_{i+3}$)

When an observation is considered an outlier its value is replaced by the average value of the two observations that are immediately before and after the outlier.

5.3 Box-Cox transformations

Definition of Box - Cox transformations :

$$w_i = \begin{cases} log(x_i) if \lambda{=}0; \\ \\ \left(x^{\lambda}{-}1\right)/\lambda otherwise. \end{cases} \tag{9}$$

These transformations are used for example in ARIMA modeling to stabilize the variance in series with multiplicative seasonality. These series are also problematic for KNN regression as the seasonal component is increasing over time.

According to Athanasopoulos, George, Hyndman, Rob J. - Forecasting Principles and Practice (2018) : “

The logarithm in a Box-Cox transformation is always a natural logarithm (i.e., to base e). So if λ = 0 , natural logarithms are used, but if λ ≠ 0 , a power transformation is used, followed by some simple scaling. If λ = 1 , then w $_i$ = y $_i$ − 1 , so the transformed data is shifted downwards but there is no change in the shape of the time series. But for all other values of λ , the time series will change shape.”

5.4 Trend

KNN is not suitable for forecasting time series with a global trend and some tools are needed to remove the trend component :a. Differencing, b. STL decomposition

According to Katos A (2004) and Christou (2003) :”With the term trend we assume the continuous over time, increase or decrease of the values of a time series. It is obvious that if a variable is characterized by trend, the mean and possibly and the variance are changing over time, that means that the specific time series is not stationary. According to Granger and Newbolt, if in a regression the value of $\mathrm{R}^2$and the value of Durbin -Watson statistic is very low and specifically if $\mathrm{R}^2 > d$, is very possible the regression not to be real but a spurious regression. In this case it is preferable, to estimate the relationship between the first differences and not on the levels of variables, for example, in order not to estimate the model :

$$Y_i = \beta_o + \beta_1 X_i + u_i \tag{10}$$

but to estimate the model :

$$\Delta Y_i = \beta_0 + \beta_1 \Delta \mathrm{X}_i + u_i^* \tag{11}$$

, where

$$u_i^* = u_i - u_{i-1} \tag{12}$$

The reason for using the first differences is that the usage of the first differences is that a lot of times series in the economy have the characteristics of a random walk. Hence,the usage of the first differences is transforming them stationary. In every case, is mandatory to check for stationary of the time series that used in each model though.”

According to Athanasopoulos, George Hyndman, Rob J. - Forecasting Principles and Practice (2018) : “

STL is a versatile and robust method for decomposing time series. STL is an acronym for “Seasonal and Trend decomposition using Loess”, while Loess is a method for estimating nonlinear relationships. The STL method was developed by Cleveland, Cleveland, McRae, & Terpenning.

Athanasopoulos and Hyndman (2018) support that the main advantages of STL decomposition are:

a. the ability to handle all types of seasonality

b. the part that deals with seasonality can change over time and degree of change can be manipulated by the researcher

c. the researcher can also manipulate how smooth is the trend cycle

d. robustness to extreme values

Proceeding to Hyndman and Athanasopoulos (2018)the main disadvantages of the STL method are :

a. it is not possible to work by itself with little or a no direct human control with trading day or calendar variations

b.only additive decompositions can be handled easily by the researcher.

5.5 Seasonality

Robert Kabacoff (2015) explains the seasonal decomposition of time series :

“Seasonal decomposition Time-series data that have a seasonal aspect (such as monthly or quarterly data) can be decomposed into a trend component, a seasonal component, and an irregular component. The trend component captures changes in level over time. The seasonal component captures cyclical effects due to the time of year. The irregular (or error) component captures those influences not described by the trend and seasonal effects. The decomposition can be additive or multiplicative. In an additive model, the components sum to give the values of the time series. Specifically,

$$Yt = Trendt + Seasonalt + Irregulart \tag{13}$$

where the observation at time t is the sum of the contributions of the trend at time t, the seasonal effect at time t, and an irregular effect at time t. In a multiplicative model, given by the equation

$$Yt = Trendt * Seasonalt * Irregulart \tag{14}$$

the trend, seasonal, and irregular influences are multiplied."

KNN can model additive seasonality, but in order to experiment, we test two strategies for removing seasonality in order to verify if they can improve the forecast accuracy of a KNN model.

a. Taking differences at lag n, where n is the period of seasonality

b.Another strategy may used whether seasonality is detected, deseasonalizing is performed by subtracting the seasonal average.

Javier Sáez Martínez (2021) uses sine and cosine transformations to deal with seasonality in business problems.

Part III

Second stage: Modeling via the k-NN algorithm

In order to perform a forecast via the KNN for time series, we need first to decide : what the values of k will be, the choice of variables, the steps ahead strategy, the distance measure and how the targets are combined.

6 Choosing k

There are two common approaches to choose the number of the neighbors.

i.Setting k equal with the square root of the number of training observations

ii.To divide the training set into a training set and a validation set. Hence k is chosen thus it will minimize an accuracy measure of the forecast for the validation set,while using the training data. A disadvantage is that it is time consuming.

Third strategy : It tries to balance the advantages of efficiency and an exhaustive research via combinations of models predicting time series. Hence,it combines the forecasts of various KNN models, with different values of k, generating a final forecast as the mean prediction of different models.

Guo -Feng Fan et al (2019) explains the pros and cons of a choice of small or large number of neighbors : “Based on the K-NN algorithm, k is a user-defined neighbor parameter, which is used to classify samples to be classified according to the category label with the highest frequency of occurrence among the k training samples that are closest to the selected data point. If the value of k is too large or too small, it will increase the interference to the data and reduce the classification accuracy. In the case where the value of k is small, the complexity of the model is higher (i.e., it is easy to suffer from the over-fitting problem), and there is an increase of the estimation errors. Eventually, the forecasting results are very sensitive to the neighbor data points. On the contrary, in the case where the value of k is large, it would reduce the estimation errors; however, the approximation errors would be simultaneously increased, and the training data points farther from the input data point will also affect the forecasting results. Therefore, in general applications of the K-NN algorithm, the value of k is often set as a relatively small value, but must be an integer.”

7 Choosing the input variables

More specifically with what way we are going to choose the lagged variables of a time series, as input variables of the KNN regression model.

i.Naive method : choosing a default number of lags

ii. Using PACF we choose as variables the lags with significant autocorrelation.

iii. Based on their predictive performance in a validation set, using the prediction algorithm. Here the prediction algorithm is the KNN regression and the rolling origin technique has been used to assess the forecast performance.

8 Selecting a multi-step ahead forecasting approach

The classical methodologies are recursive approach and direct approach. Recursive approach can only predict an one step ahead and the direct methodology is very time consuming. Both are backward looking methodologies.

8.1 Recursive strategy

Souhaib Ben Taieb and Rob J Hyndman (2012) define the recursive strategy : ”in the recursive strategy, we estimate the model

$$y_t = m(x_{t-1}; \theta) + e_t \tag{15}$$

, where

$$x_t = [y_t, ..., y_{t-p+1}]' \tag{16}$$

and

$$E[e_t] = 0 \tag{17}$$

. In standard step optimization with squared errors, the objective function of the recursive strategy is

$$E[(y_{t+1} - m(x_t;\theta))^2|x_t] \tag{18}$$

,

and the parameters ϑ are estimated by

$$\hat{\theta} = argmin_{\theta\in\Theta} \sum_{\mathrm{t}} (y_t - m(x_{t-1};\theta))^2 \tag{19}$$

,

where Θ denotes the parameter space. We compute forecasts recursively:

$$\hat{m}^{(h)}\left(x_t\right) = \begin{cases} \hat{m}([\hat{m}^{(h+1)}(x_t), ..., \hat{m}^{(h-p)}(x_t)'], & if h > 0; \\ x_t^{'} w_h, & if 1 - p \leq h \leq 0; \end{cases} \tag{20}$$

where $\hat{m}(x)$is a shorthand notation for m(x; $\hat{\theta}$). These forecasts are also sometimes called “iterated multi-step” (IMS) forecasts.

The choice of $\hat{\theta}$ given by eq. (19) minimizes the mean squared error of the one-step forecasts, and so ensures that

$$m^{(1)}(x_t) = \mu_{t+1|t} \tag{21}$$

provided m$\asymp$ f and p = d.”

Proceeding on Souhaib Ben Taieb and Rob J Hyndman (2012) the main advantage of the recursive strategy : only one model is required , hence less computing time is needed. One disadvantage of the recursive strategy is that the forecasts are not equal to the conditional mean.

8.2 Direct strategy

With the direct strategy as defined from Souhaib Ben Taieb and Rob J Hyndman (2012) :" different forecasting models are used for each forecast horizon:

$$y_t = m_h(y_{t-h}, ..., y_{t-h-p_h}; \theta_h) + e_{t,h} \tag{22}$$

where h = 1,...,H.

For each model, the parameters ϑ_h are estimated as follows

$$\hat{\theta}_h = \underset{\theta_h \in \Theta_\mathrm{h}}{argmin} \sum_t [y_t - m_h(x_{t-h}; \theta_h)]^2 \tag{23}$$

Then forecasts are obtained for each horizon from the corresponding model, $\hat{m}^{(h)}(x_t) = m_h(x_t; \hat{\theta}_h)$.

This is sometimes also known as "direct multi-step" (DMS) forecasting."

8.3 Multi Input Multi Output strategy

Another methodology is the Multi Input Multi Output strategy (MIMO). The MIMO strategy learns one multiple output for the time series, that predicts all the time points at once.

Souhaib et al (2011) define the MIMO strategy : " The MIMO strategy learns one multiple-output model F from the time series [y$_1$,...,y$_N$] where

$$[y_{t+H}, ..., y_{t+1}] = F(y_t, ..., y_{t-d+1}) + w \tag{24}$$

,

with t ∈ {d,...,N −H}, F : $\mathbb{R}^d \rightarrow \mathbb{R}^{\mathbb{H}}$ is a vector-valued function, and w $\in \mathbb{R}^{\mathbb{H}}$ is a noise vector with a covariance that is not necessarily diagonal.The forecasts are returned in one step by a multiple-output model $\hat{F}$ where

$$[\hat{y}_{t+H}, ..., \hat{y}_{t+1}] = \hat{F}(y_N, ..., y_{N-d+1}) \tag{25}$$

. The rationale of the MIMO strategy is to preserve, between the predicted values, the stochastic dependency characterizing the time series. This strategy avoids the conditional independence assumption made by the Direct strategy as well as the accumulation of errors from which plagues the Recursive strategy. So far, this strategy has been successfully applied to several real-world multi-step ahead time series forecasting tasks. However, the need to preserve the stochastic dependencies by using one model has a drawback as it constrains all the horizons to be forecasted with the same model structure. This constraint could reduce the flexibility of the forecasting approach."

8.4 Selecting distance metric and the combination of targets

There are two main distance metrics. The Euclidean distance and Manhattan distance. The Euclidean distance is the most used distance metric.

8.4.1 Euclidean distance

Michael Greenacre defines the Euclidean distance between two J dimensional vectors x and y as :

$$d_{x,y} = \sqrt{\sum_{j=1}^{J}(\ x_j - y_j)^2} \tag{26}$$

8.4.2 Manhattan Distance

Fürnkranz, J. et al (2011) define the Manhattan distance metric : "

Definition of the Manhattan distance between two points x = (x$_1$,x$_2$,...,x$_n$) and y= (y$_1$,y$_2$,...,y$_n$) in n-dimensional space is the sum of the distances in each dimension. $d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$. It is called the Manhattan distance because it is the distance a car would drive in a city (e.g., Manhattan) where the buildings are laid out in square blocks and the straight streets intersect a triangles . This explains the other terms City Block and taxicab distances. The terms L and -norm distances are the mathematical descriptions of this distance. "

8.4.3 Combination of targets

Given the k targets for the nearest neighbor, they have to be combined to produce the forecast, more specifically :

the mean value, the median value, a weighted mean (closer neighbors have higher weights) and a trimmed mean (the lowest and the highest values are removed from the mean).

According to National Institute of Standards and Technology: " The mean is the sum of the observations divided by the number of observations. The mean can be heavily influenced by extreme values in the tails of a variable. The trimmed mean compensates for this by dropping a certain percentage of values on the tails. For example, the 50% trimmed mean is the mean of the values between the upper and lower quartiles. The 90% trimmed mean is the mean of the values after truncating the lowest and highest 5% of the values."

9 Third stage : Experimentation, verification of the methodology, selecting the methods

According the study of Francisco Martinez, Maria Pilar Frias, Maria Dolorez Perez, Antonio Jesus Rivera (2017) that used the data from the NN3 competition in order to select the most suitable strategies and methodologies and select one of each, from those that were already described.

Lichtendahl and Winkler (2019) define the SMAPE method in order to assess the forecasting accuracy: "

The evaluation measure is the SMAPE (symmetric mean absolute percentage error) for the forecast accuracy. For forecasts of time series i at various forecast horizons, the SMAPE is given by :

$$sMAPE\left(y_i, \hat{y_i}\right) = \frac{1}{H_i}\sum_{h=1}^{H_i}\frac{|y_{i,h} - \hat{y}_{i,h}|}{\frac{|y_{i,h}|+|\hat{y}_{i,h}|}{2}}100 \tag{27}$$

where y$_i$=$(y_{i,1}, ..., y_{i,H_i})$is the vector of the time series i's actual values in the testing set and

$$\hat{y_i} = (\hat{y}_{i,1}, ..., \hat{y}_{i,H_i}) \tag{28}$$

is the vector of forecasts from one to H$_i$ steps ahead. "

Here H$_i$ = 18, the last 18 observations of the training set are used as validation set.

10 Results - Selecting the modeling strategies

10.1 Selecting k

After comparing all the strategies about selecting k, here we opt for the approach of the combinations due to its robustness. More precisely we combine three models with k equals to 3, 5 and 7 respectively and use their forecasts averaged.

10.2 Selecting the distance metric

Here we opt for the Euclidean distance in contrast with Manhattan distance , because of the fact is the common distance metric used in time series forecasting with k-NN regression.

10.3 Selecting how to combine the k targets

In this part we select the mean because it obtains the best accuracy.

10.4 Selecting the multi-step ahead forecasting strategy

After comparing the direct , the recursive and the MIMO strategies for 18 forecasted future points, the recursive strategy is selected because its simpler and slightly faster.

10.5 Selecting the input (explanatory) variables

In this part we select the forward selection strategy due to its better results.

10.6 Selecting the preprocessing approaches

1. Our data do not need normalization for doing kNN regression because they are in the same scale.

2. Despite the fact that outliers are rare in our data, we include this strategy in our methodology.

3. Using Box - Cox transformation in our methodology, it yields better results, in contrast to being omitted.

4. We will not include strategy of removing the trend , due to the fact that after studying the patterns from our data, these strategies are not effective.

5. We will not include strategies for removing the seasonality, because the fact that kNN algorithm can deal with seasonal patterns without preprocessing.

11 SELECTED METHODOLOGY FOR APPLYING KNN IN TIME SERIES FORECASTING

Thus, based on our experimentations, we are going to use the following methodology :

1. Detect and remove the outliers.
2. Use forward selection on a validation set to select input variables I.
3. Use I to build three kNN models with k = 3, 5, 7
4. Every model generates its forecasts using the recursive approach
5. The final forecast is the mean of the forecasts of the three models

This methodology works better with the seasonal series. When several design alternatives are attractive, it is worth trying to combine them. Hence we use a combination of models with different schemes for selecting input variables. This methodology seems to improve forecast accuracy.

Part IV

Econometric techniques : smoothing models.

According to Katos (2004) : ”As smoothing of a time series, we mean the removal or the reduction of fluctuations of that series. For the case of unstable short run fluctuations, then we refer to smoothing of that time series. And for the case of referring on seasonal or cyclic fluctuations, then we refer on seasonal smoothing of that series.”

Exponential smoothing Rob J Hyndman and George Athanasopoulos, 2018 support : “

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry. ”

Katos (2004) defines the simple exponential smoothing model :”

$$\tilde{Y_t} = aY_t + a(1-a)Y_{t-1} + a(1-a)^2 Y_{t-2} + ... \tag{29}$$

or

$$\tilde{Y_t} = aY_t + (1-a)\tilde{Y}_{t-1} = a\sum_{k=0}^{t-1}(1-a)^k Y_{t-k} \tag{30}$$

where 0$<$ a ≤ 1 is the smoothing coefficient. The smaller ’a’ value is, the greater is the smoothing of the series.

This method is suitable for forecasting data with no clear trend or seasonal pattern.”

Kabacoff (2015) explains very well the model : “

Simple exponential smoothing uses a weighted average of existing time series values to make a short term prediction of future values. The weights are chosen so that observations have an exponentially decreasing impact on the average as you go back in time. The simple exponential smoothing model assumes that an observation in the time series can be described by

Y$_t$ =level + irregular$_t$

The prediction at time Y$_{t+1}$(called the one step ahead forecast) is written as

Y$_{t+1}$ =c$_0 Y_t$+c$_1 Y_{t-1} + c_2 Y_{t-2} + ...$

where c$_i$ = $a(1-a)^i$,i = 0, 1, 2, ... and 0$<$a$\leq$ 1. The c$_i$weights sum to one, and the 1-step ahead forecast can be seen to be a weighted average of the current value and all past values of the time series. The 'a' parameter controls the rate of decay for the weights. The closer 'a' is to 1, the more weight is given to recent observations. The closer 'a' is to 0, the more weight is given to past observations. The actual value of 'a' is usually chosen by computer in order to optimize a fit criterion. A common fit criterion is the sum of squared errors between the actual and predicted values.

Holt and Holt-Winters exponential smoothing The Holt exponential smoothing approach can fit a time series that has an overall level and a trend (slope). The model for an observation at time t is Y$_t$ $=$ *level*$+$ slope*t $+$ irregular$_t$

An alpha smoothing parameter controls the exponential decay for the level, and a beta smoothing parameter controls the exponential decay for the slope. Again each parameter ranges from 0 to 1, with larger values giving more weight to recent observations. "

Makridakis et al (1997) defines that : "

Holt's linear method extended single exponential smoothing to linear exponential smoothing to allow forecasting of data with trends. The forecast for Holt's linear exponential smoothing is found using two smoothing constants, α and β (with values between 0 and 1), and three equations:

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + b_{t-1}) \tag{31}$$

,

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \tag{32}$$

,

$$F_{t+m} = L_t + b_t m \tag{33}$$

.

Here L $_t$ denotes an estimate of the level of the series at time t and b $_t$ denotes an estimate of the slope of the series at time t. Equation (31) adjusts L $_t$ directly for the trend of the previous period, b $_{t-1}$, by adding it to the last smoothed value, L $_{t-1}$. This helps to eliminate the lag and brings L $_t$ to the approximate level of the current data value. Equation (32) then updates the trend, which is expressed as the difference between the last two smoothed values. This is appropriate because if there is a trend in the data, new values should be higher or lower than the previous ones. Since there may be some randomness remaining, the trend is modified by smoothing with β the trend in the last period (L $_t$ − L$_{t-1}$), and adding that to the previous estimate of the trend multiplied by (1 − β). Thus, (32) is similar to the basic form of single smoothing but applies to the updating of the trend. Finally, equation (33) is used to forecast ahead. The trend, b$_t$, is multiplied by the number of periods ahead to be forecast, m, and added to the base value, L$_t$. ”

Kabacoff (2015) supports : ”

The Holt-Winters exponential smoothing approach can be used to fit a time series that has an overall level, a trend, and a seasonal component. Here, the model is

$$Y_t = level + slope * t + s_t + irregular_t \tag{34}$$

where s$_t$ represents the seasonal influence at time t. In addition to alpha and beta parameters, a gamma smoothing parameter controls the exponential decay of the seasonal component. Like the others, it ranges from from 0 to 1, and larger values give more weight to recent observations in calculating the seasonal effect. ”

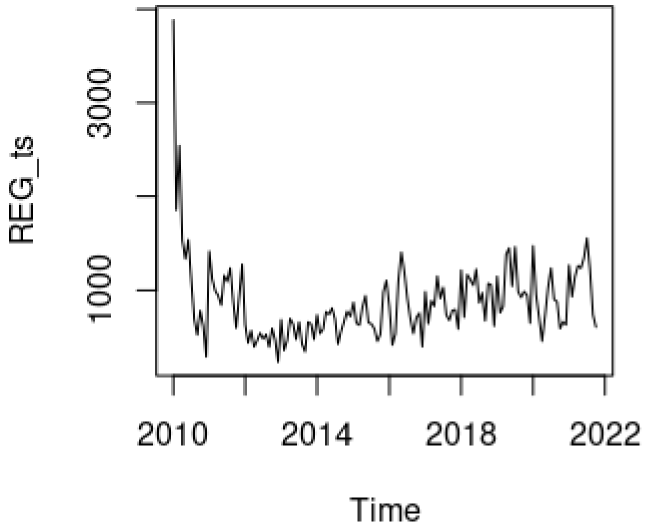
Part V

Data set

The data used in the study were from the Hellenic Association of Motor Vehicle Representatives (AMVIR). They are the monthly registrations data for TOYOTA, from January 2010 until and October 2021.

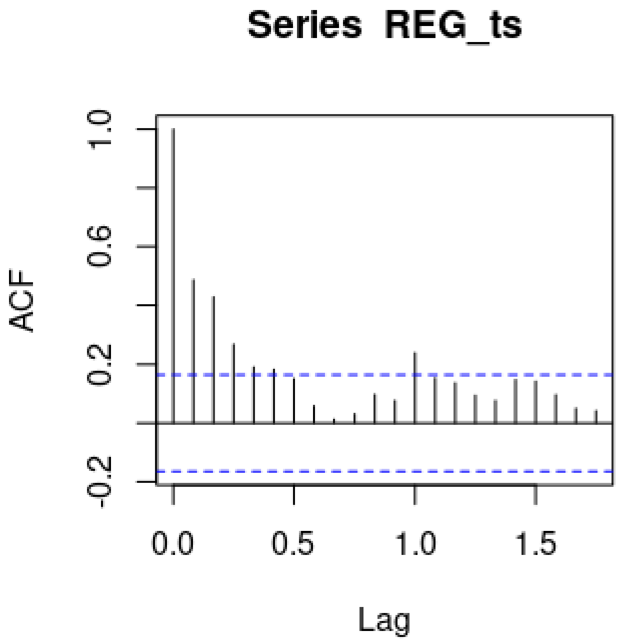
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
234.0	619.8	778.0	872.0	1069.0	3884.0	428.2258

The sample size is n=142 observations. The standard deviation of our series is 428.2258 and the mean is 872 registrations.



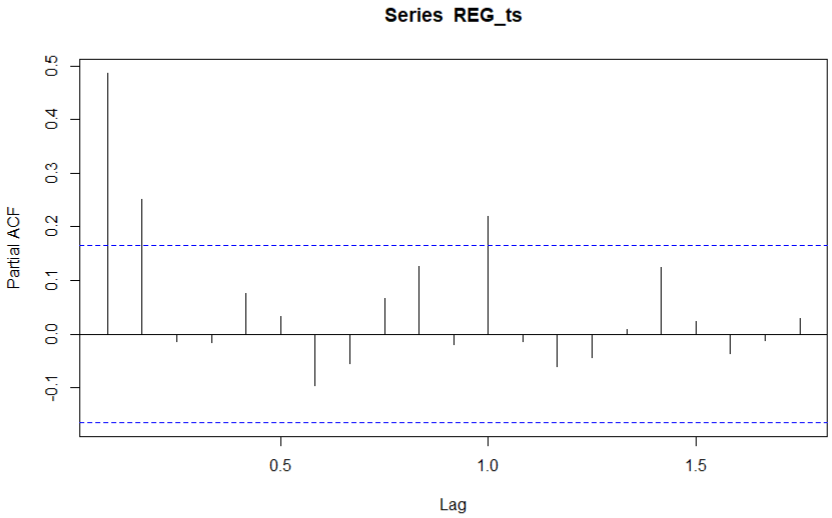
PLOT 1 : Our time series

Performing a visual check the sample distribution is not normal.



PLOT 2 : The ACF

Box-Ljung test
data: REG_ts
X-squared = 34.247
df = 1
p-value = 4.854e-09
H_0: all $p_k = 0$ (The data are independently distributed)



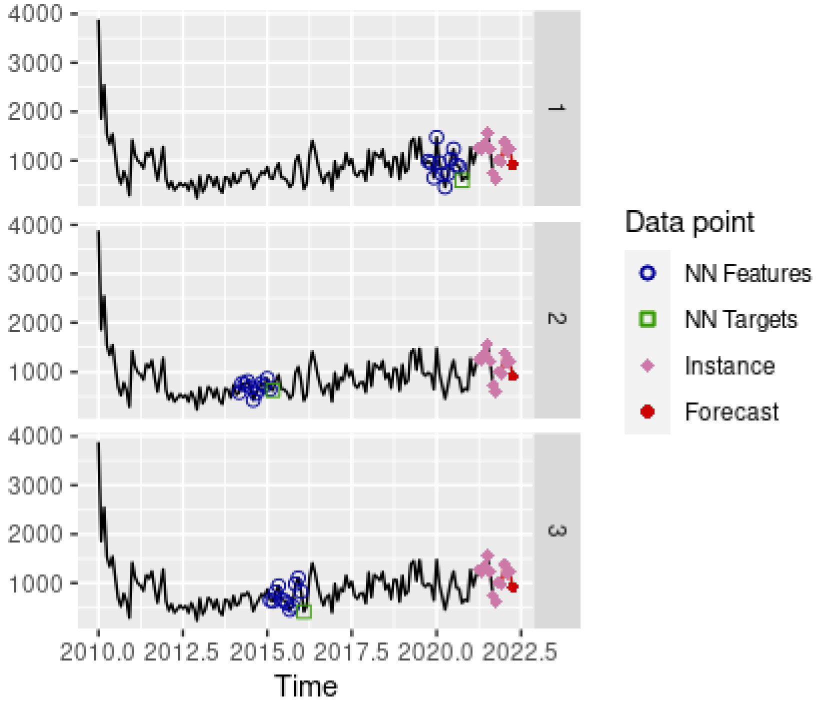
PLOT 3 : PACF

As we notice at the ACF and PACF plots the zero lag is statistically important and after the first lag follows (here the first lag equals 12 lags). Additionally, with the Ljung-Box Test we see a p-value much smaller than .01, thus we can reject the null hypothesis, indicating the time series does contain an auto correlation.

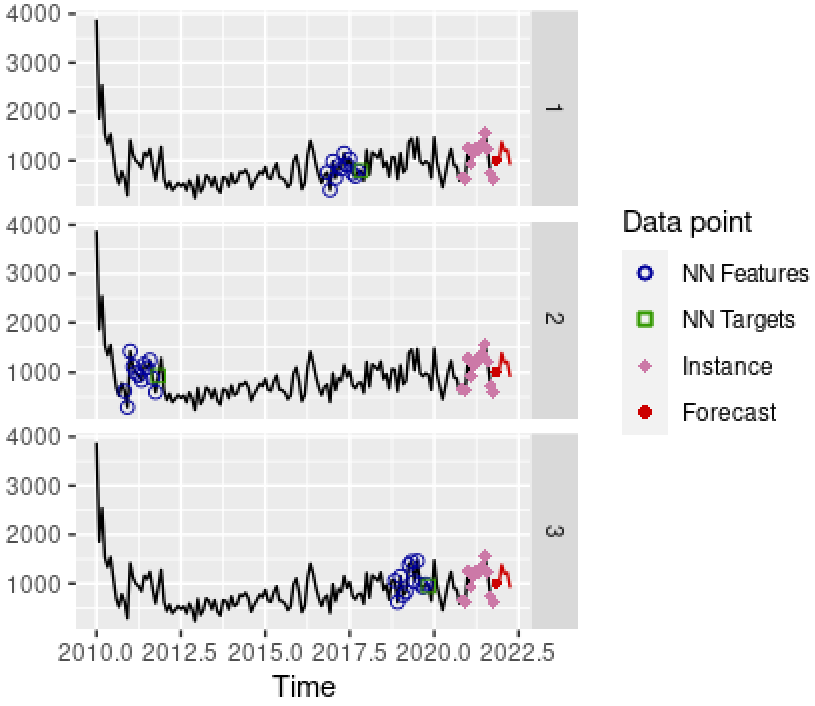
Part VI

Applications

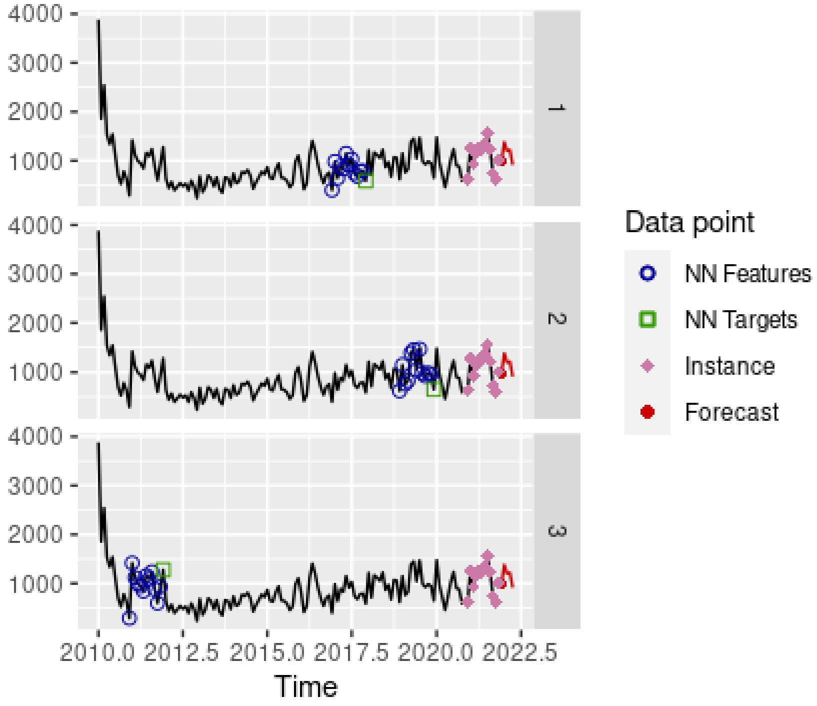
Using the tsfknn R package with our data set (we note that we did not use any Box- Cox transformation in none application), initially we try to make a prediction with six steps ahead and using lags 1-12 as features and k=3 neighbors. The features associated with the next future value of the time series are the last 12 values of the time series vector. The three most similar examples (nearest neighbors) of this instance are vectors whose targets are averaged to produce the prediction. Each nearest neighbor has been plotted in different plot. The neighbors in the plots are sorted according to their distance to the instance, being the neighbor in the top plot the nearest neighbor. All this procedure is illustrated at the following 6 plots (PLOT 4 - PLOT 9).



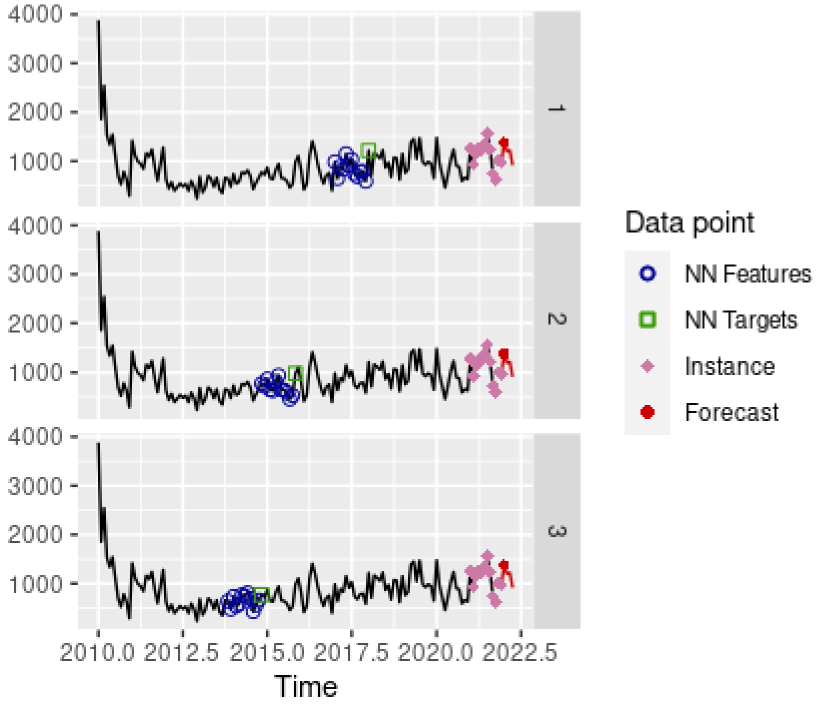
PLOT 4: For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.



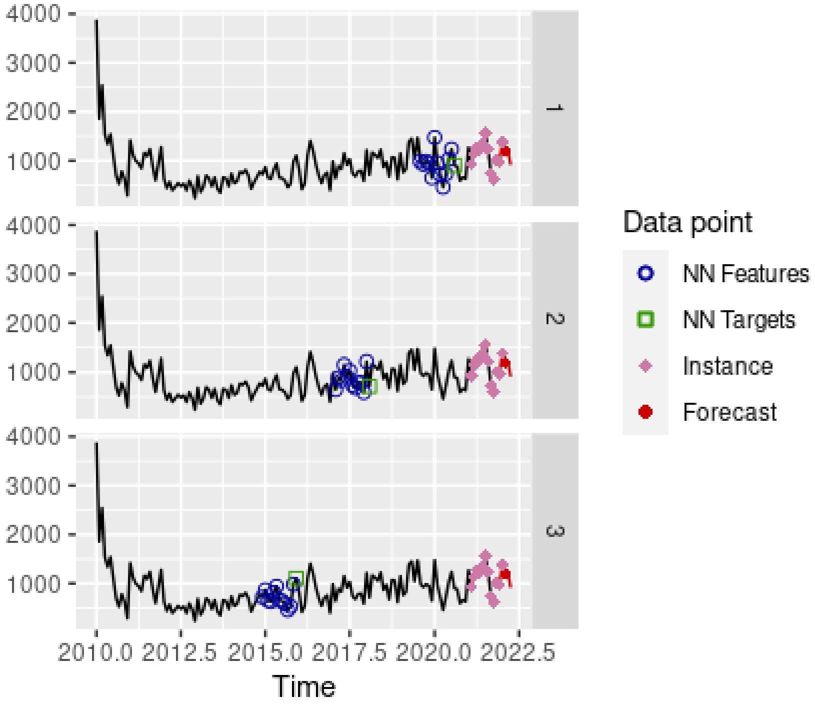
PLOT 5 : For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.



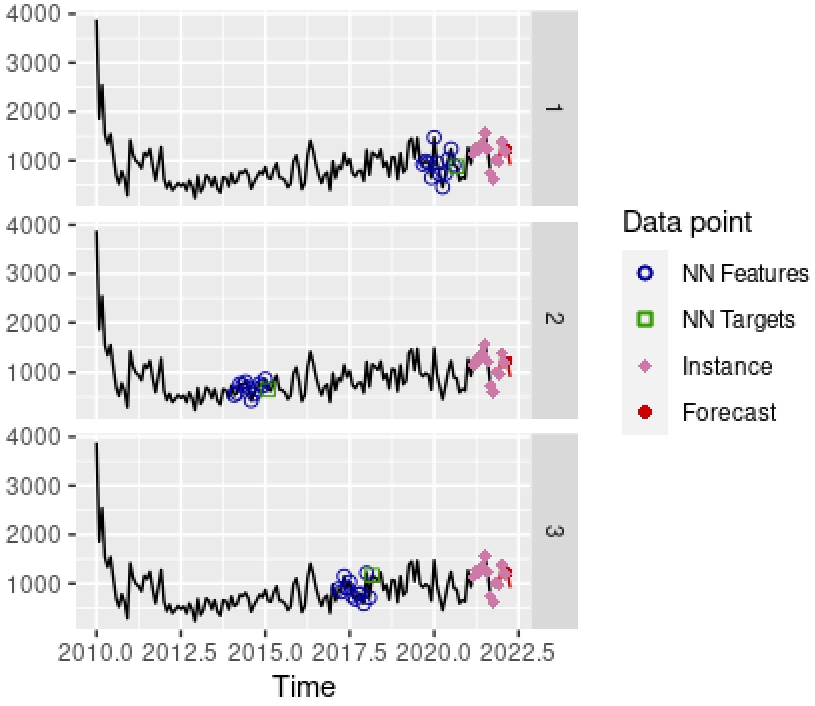
PLOT 6 : For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.



PLOT 7 : For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.



PLOT 8: For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.

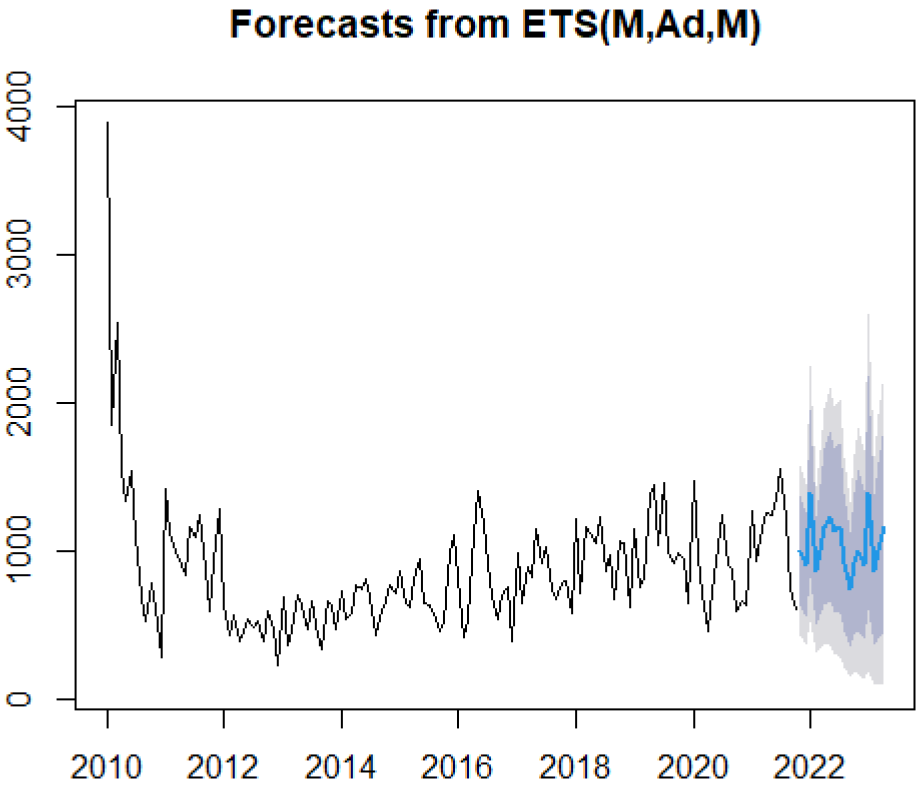


PLOT 9 : For K=3 we have used lags 1-12 to specify the features in order to predict the future point in red. The three most similar examples (NN) of this instance whose targets are averaged to produce the forecast.

Our next step was to generate a kNN regression with a vector of k=(3, 5, 7) and 12 lags and as multiple step ahead strategy a vector including recursive and MIMO strategies.

Afterwards, using the R package : forecast, we generated exponential smoothing models using the ets() function and automated forecasting. You can use this function to fit exponential models that have multiplicative components, add a dampening component and perform automated forecasting.

Both techniques (kNN regression and exponential smoothing) were regenerated in various forecasts for six, nine, twelve and eighteen steps ahead.



PLOT 10. Exponential Smoothing predictions

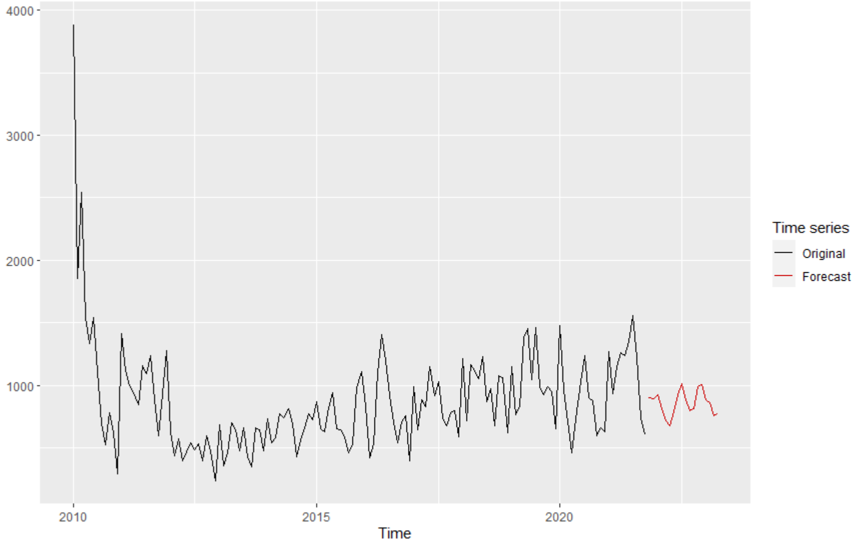
Kabacoff (2015) supports : “Exponential models can be fit with either the HoltWinters() function in the base installation or the ets() function that comes with the forecast package. The ets() function has more options and is generally more powerful.

The format of the ets() function is

ets(ts, model=”ZZZ”)

where ts is the time series and the model is specified by three letters. The first letter denotes the error type, the second letter denotes the trend type, and the third letter denotes the seasonal type. Allowable letters are A for additive, M for multiplicative, N for none, and Z for automatically selected."

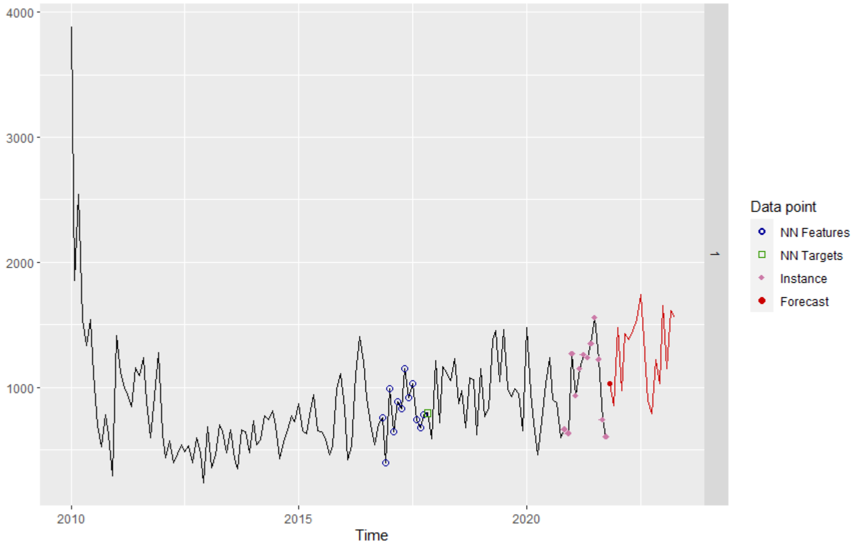
In our application, we chose the automatic selection for all types. Thus the function automatically selected Multiplicative for the error type, Additive for the trend type and Multiplicative for the seasonal type.



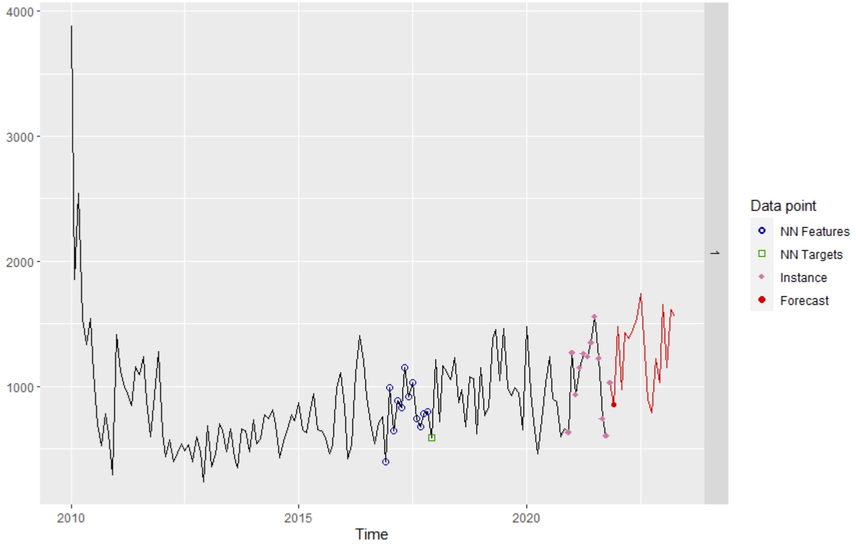
PLOT 11: KNN VECTOR FORECAST

Afterwards we generated another four different KNN regression models for K=1, K=2, K=5 ,K=12 and K=24 via the tsfknn package with R.

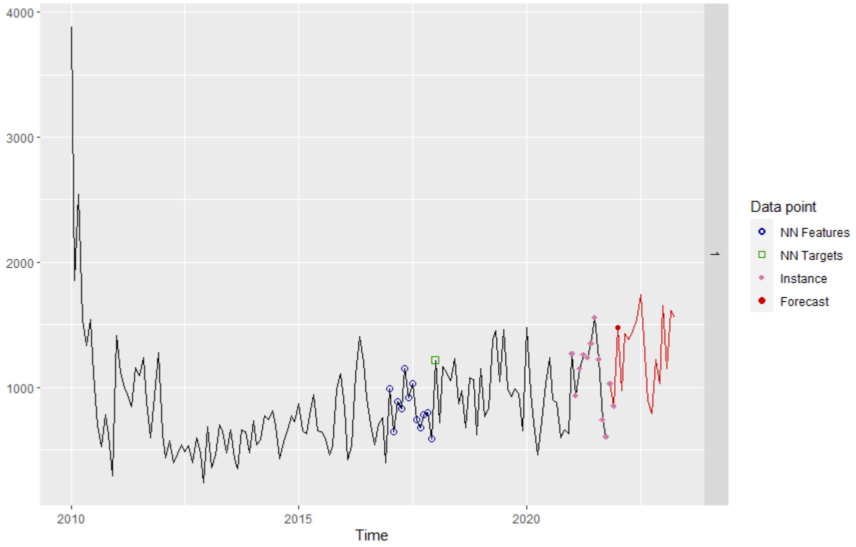
For K = 1 we present the full plot procedure as follows, and for K=2, K=5 and K =12 only selected plots :



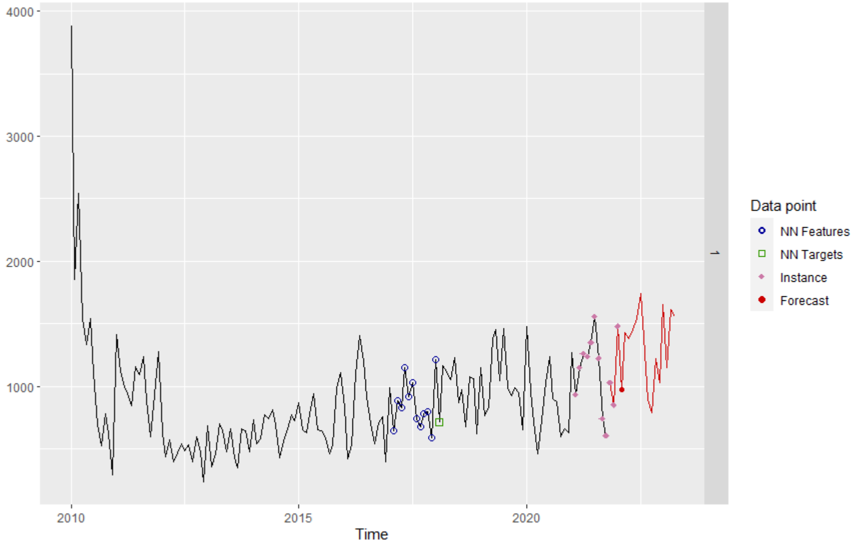
PLOT 12: Prediction point for K=1



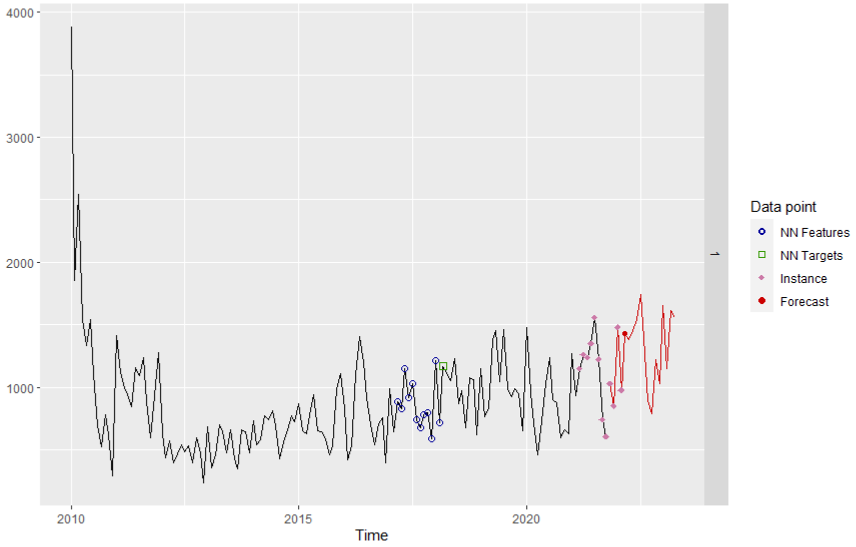
PLOT 13: Prediction point for K=1



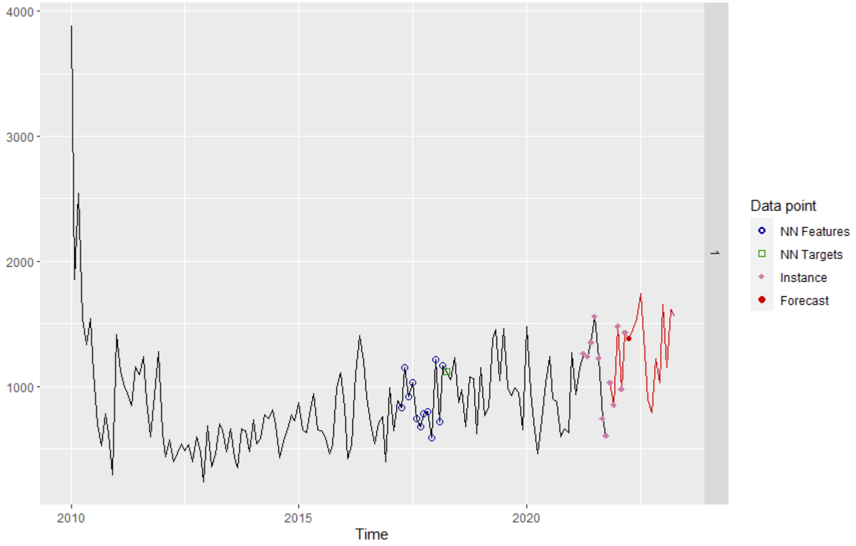
PLOT 14 : Prediction point for K=1



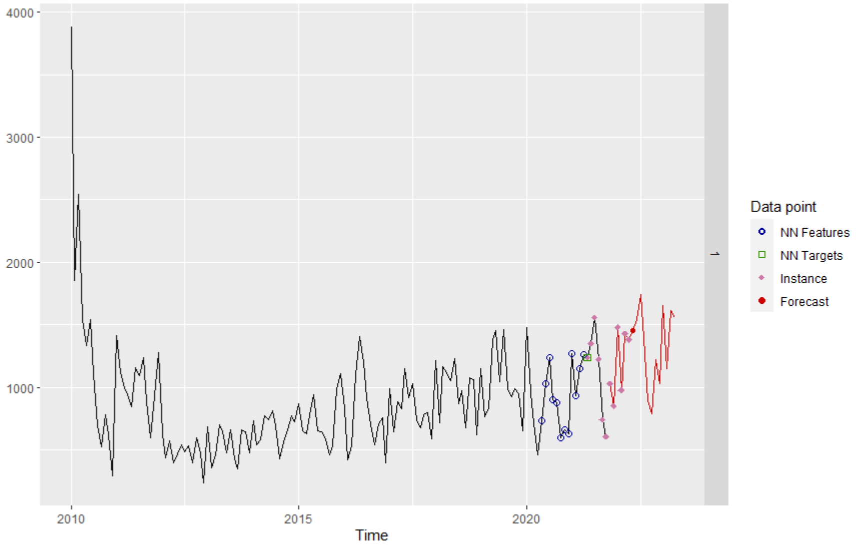
PLOT 15 : Prediction point for K=1



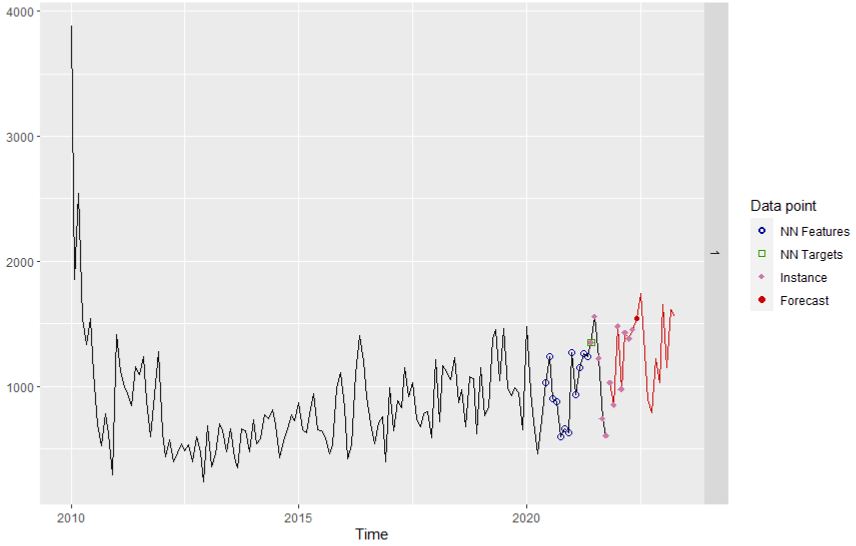
PLOT 16: Prediction point for K=1



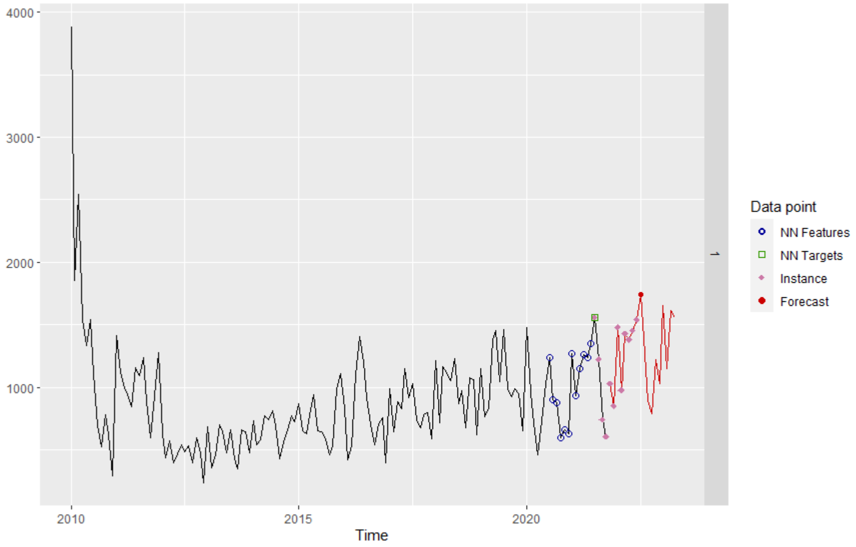
PLOT 17: Prediction point for K=1



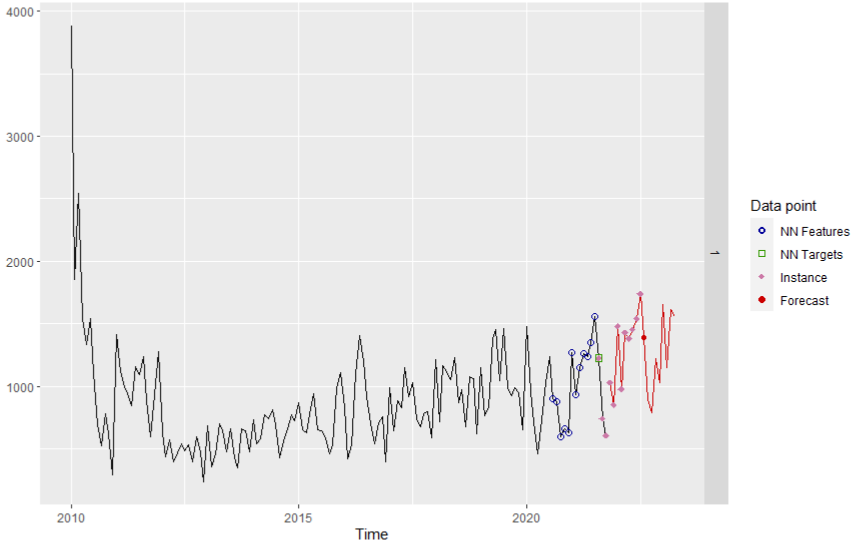
PLOT 18: Prediction point for K=1



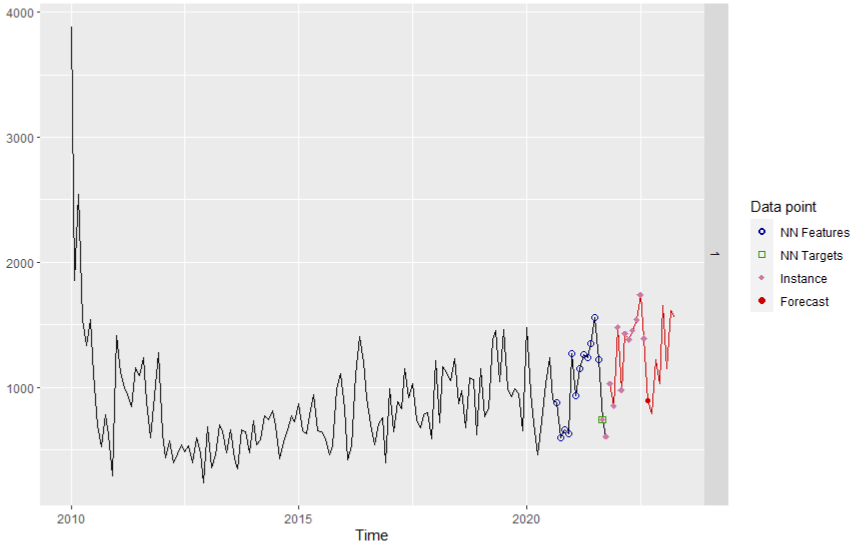
PLOT 19: Prediction point for K=1



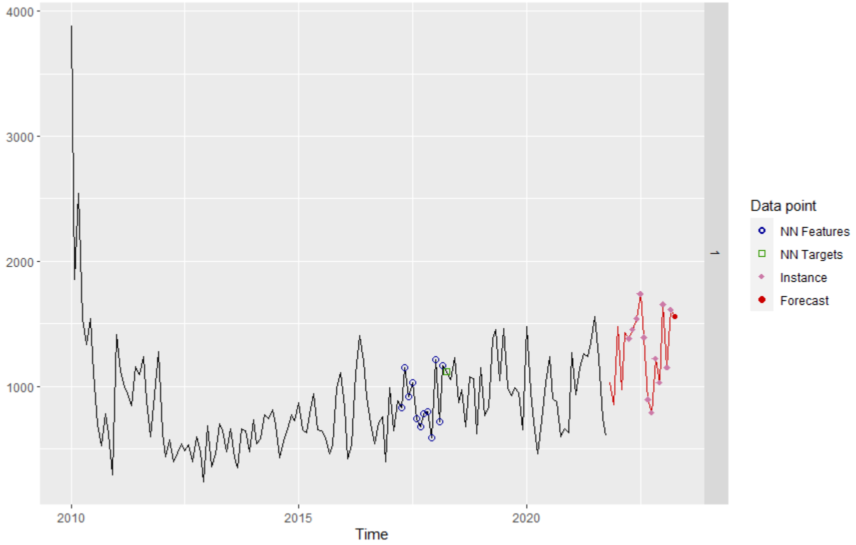
PLOT 20 : Prediction point for K=1



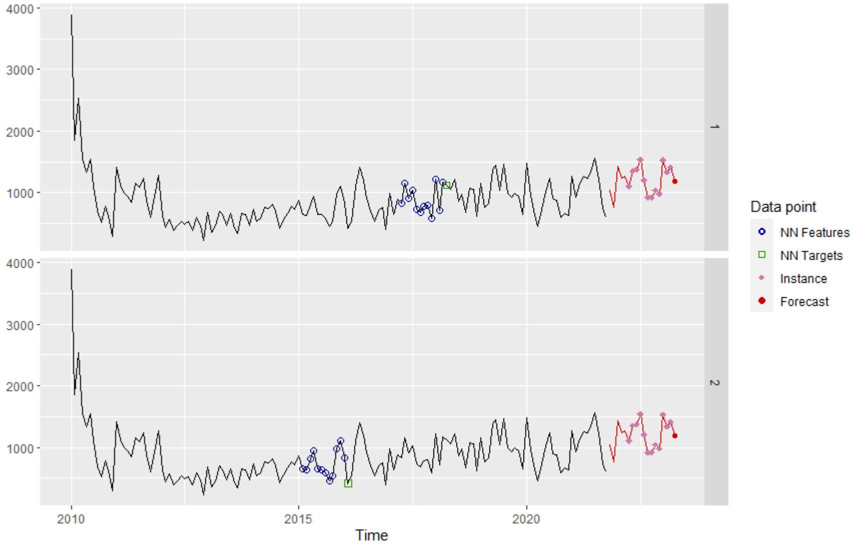
PLOT 21: Prediction point for K=1



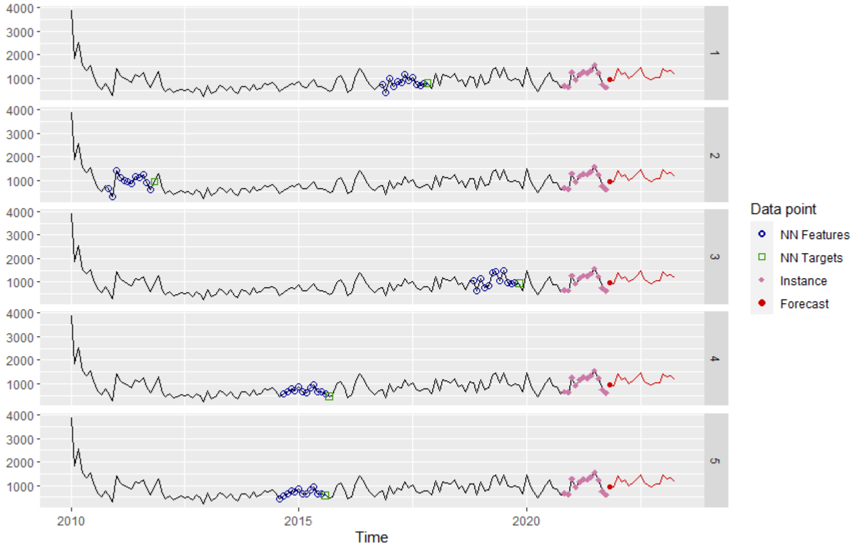
PLOT 22: Prediction point for K=1



PLOT 23: Prediction point for K=1



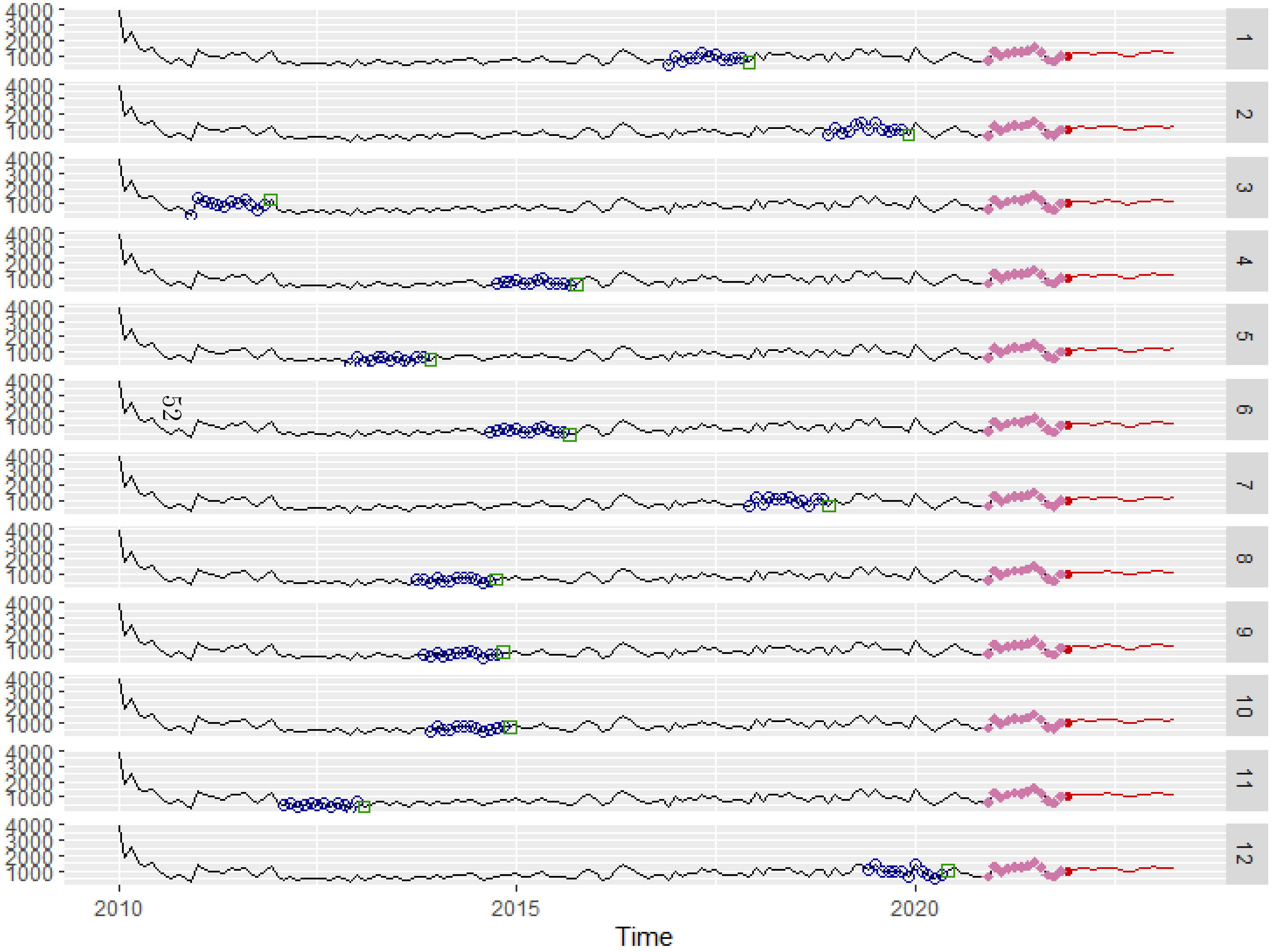
PLOT 24 : a K=2 prediction point



PLOT 25: a K=5 prediction point

PLOT 26

For K=12 and up to 18 steps ahead predictions



Part VII

Comparative comments on the results

Accuracy metric	RMSE	MAE	MAPE
k=1	453.5550	375.5112	36.8727
k=2	360.8493	304.5698	30.4590
k=3	308.69600	278.42197	32.67946
k=5	325.22561	291.25691	29.51154
k=12	312.55986	286.37012	29.65794
k=24	314.37759	286.64512	29.72298
KNN VECTOR	294.48650	259.36599	28.31113

Table 1: KNN regressions forecast accuracy

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
training set	-6.858196	295.0942	202.6776	-8.227146	25.15197	0.7183035	0.2797846

Table 2: Exponential smoothing forecast accuracy

As we look at the results of values predicted and the accuracy of the predictions,

we notice that as k increases (k>3) the results become controversial. The forecasts of the kNN regression for k=3 are closer to those of the exponential smoothing, in contrast with those of the kNN regression vector k=3, 5, 7, that are quite different (smaller values) comparing with those with k= 3 but also with those of the exponential smoothing model.

In addition, in terms of accuracy of the forecasting methods, we observe that the predictions of exponential smoothing are those with the best accuracy (smaller errors), from the of the kNN vector and the simple k=3 KNN regression, in terms of RMSE, MAE and MAPE.

Hence, in the terms of our study, the best forecasting technique is the exponential smoothing, the next one is the KNN regression with k=3 number of neighbors.

Finally, we restricted our attention in this thesis mainly to point forecasts, due to time constraints. Probability forecasts are much more informative than point forecasts, because they provide important information about uncertainty.

For future work, will be interesting to compare the performance of other new R packages for kNN regresion, such as the jenga package.

References

[1] Athanasopoulos, George Hyndman, Rob J. - Forecasting Principles and Practice (2018)

[2] Christou G, Introduction in econometrics, book in greek (2003)

[3] Guo-Feng Fan et al, Application of the Weighted K-Nearest Neighbor Algorithm for Short-Term Load Forecasting (2019)

[4] Fürnkranz, J., Chan, P. K., Craw, S., Sammut, C., Uther, W., Ratnaparkhi, A., . . . De Raedt, L. (2011). Manhattan Distance. Encyclopedia of Machine Learning, 639–639.

[5] Gaganis and Zopounidis, IDENTIFICATION OF FORGED ACCOUNTING STATEMENTS -book in greek (2008)

[6] Michael Greenacre [http://www.econ.upf.edu/~michael/stanford/maeb4.pdf]

[7] Hertzmann and Fleet, Machine Learning and Data Mining Lecture Notes. (2010)

[8] S B Imandoust et al. Int. Journal of Engineering Research and Applications Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610, Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background

[9] Robert Kabacoff-R in Action Data Analysis and Graphics with R-Manning Publications (2015)

[10] Katos A., ECONOMETRICS , theory and applications, book in greek (2004)

[11] Lichtendahl and Winkler, Why do some combinations perform better than others ?, International Journal of Forecasting (2019)

[12] S.Makridakis et al - Forecasting Methods and Applications, Wiley (1997)

[13] Francisco Martinez, Maria Pilar Frias, Maria Dolorez Perez, Antonio Jesus Rivera, A methodology for applying k-nearest neighbor to time series forecasting, Springer (2017)

[14] Martinez et al, Time Series Forecasting with KNN in R : the tsfknn Package, (2021)

[15] Javier Sáez Martínez, Finding Trends in the B2B Textile Sector using Machine Learning, (2021)

[16] R package 'forecast'

[17] R package 'tsfknn'

[18] Souhaib Ben Taieb et al, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, (2011).

[19] Souhaib Ben Taieb and Rob J Hyndman, Recursive and direct multi-step forecasting: the best of both worlds (2012)

[20] Varian H, Machine Learning and Econometrics, Google Confidential and Proprietary, (2014)

[21] https://developers.google.com/machine-learning/data-prep/transform/normalization

[22] https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

[23] National Institute of Standards and Technology [https://www.itl.nist.gov/div898/software/dataplot/refman2/ch2/trimmean.pdf]