BACHELOR'S DEGREE FINAL PROJECT

Forecast combinations

Presented by:
Cristina Tobar Fernández

Supervised by:
DR. EMILIO CARRIZOSA PRIEGO

FACULTY OF MATHEMATICS
Statistics and Operational Research Department
Sevilla, June 2023

Abstract

In the forecasting community, forecast combinations have grown dramatically. Their uses in time series span a multitude of fields, including assisting in recent years to predict COVID-19 deaths and hospital admissions with excellent accuracy, thus helping the organization of public health in different countries.

Since the 1960s, a multitude of studies have confirmed the benefits of using a combination of different base predictions. These base predictions involve a given model. They highlight the improved accuracy of the combination methods, avoiding the need to identify the "best model".

Combining techniques range from the simplest to the most challenging methods, including the optimization of different evaluation metrics. There are also many methods to measure the performance and accuracy of our predictions depending on the target of interest.

In this thesis the problem of combined point forecasts is addressed, after describing several methods we discuss their application according to the characteristics of our time series and our objectives.

Finally, we will conclude with a couple of experiments using different time series in order to empirically test our assumptions. We will also end with a proposal for different research lines for the future.

Resumen

En la comunidad científica, las predicciones por combinación han ganado terreno de manera considerable entre las técnicas de predicción. Sus usos en series temporales abarcan multitud de campos, ayudando en los últimos años a predecir muertes e ingresos hospitalarios por COVID-19 con excelente precisión. Contribuyendo así en la organización de la sanidad pública en distintos países.

Desde los años sesenta, multitud de estudios han confirmado las ventajas de utilizar combinaciones de diferentes predicciones base. Estas predicciones base suponen un modelaje de la serie temporal. Los estudios destacan la mejora en la precisión en los métodos de predicción por combinación y el ahorro de recursos a la hora de identificar el "mejor modelo".

Las técnicas por combinación abarcan desde los métodos más sencillos hasta los más complejos, pasando por la optimización de distintas métricas de evaluación. También existen diferentes procedimientos a la hora de medir el rendimiento y la precisión de nuestras predicciones en función del objetivo de interés.

En esta tesis se aborda el problema de combinación de previsiones puntuales, tras describir varios métodos se discute la aplicación de estos según las características de nuestra serie temporal y nuestros objetivos.

Finalmente concluiremos con un par de experimentos usando diferentes series temporales con el objetivo de analizar de manera empírica nuestras suposiciones. Además acabaremos con unas propuesta de distintas líneas de investigación futuras.

“Mathematics is the gate and key to science.” – Roger Bacon

Contents

Chapter 1

Introduction

Have you ever seen the typical chart of stock market evolution over the last 12 months or a plot with the increase of temperature in a place over the last years?

These are both time series, and the prediction of their future values is one of the major problems of interest. Time series are used in other disciplines like for example economic. The idea is to model the serie and do prediction based on the model assumptions.

We know many models to predict future values, ones bether than others. However, the main purpose of this tesis is not to learn how to model the time series, but to present an idea that gained importance the last century: how to combine these predictions, yielding a so-called ensemble methods .

Forecast combination starts from a very simple idea that can help us obtain better results. We aim to present an updated review of history of the methods proposed in the past five decades, from the simplest and most intuitive to the most mathematically complex. The main reason for using it is to try to collect the advantages of each and every model, both those that fit well and those that fit poorly to our target series.

The proportion of publications addressing prediction combinations among all published papers on prediction in the Web of Science has shown a general upward trend over the past 50 years, reaching 13.80% in 2021, as shown in Figure 1.1.

Our objective is to see the advantages of using ensemble methods. We will also compare which ones should be used according to our series or our base models. To compare that we also see how to measure methods performance with different types of errors and tecniques to validate it. We will also see how sometimes the most complex method do not achive the best results.

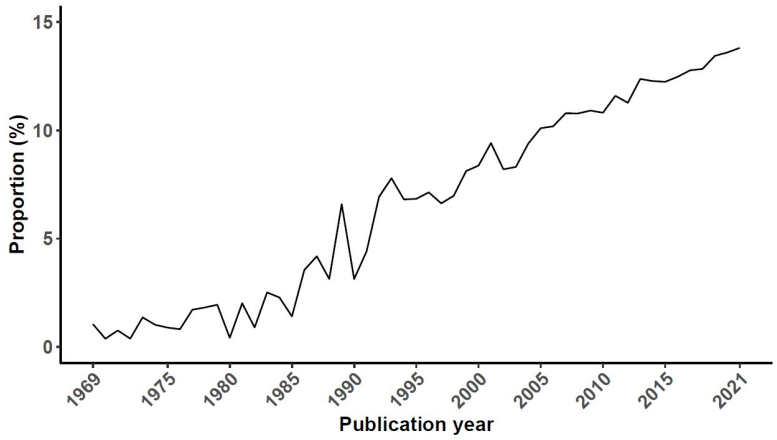

Figure 1.1: The proportion of papers that concern forecast combination among all the published forecasting papers included in the Web of Science databases from 1969 to 2021.

Source: [Wan+22]

Due to their superiority, prediction combinations have been used in a variety of fields, including epidemiology. One of the motivations for conducting this research was the incredible results obtained from 'The COVID-19 Forecast Hub'[For], a public repository for short-term forecasts of cases, hospitalizations, and deaths in the U.S. It aggregates and evaluates weekly results from many models and then generates an ensemble model. The outcome of the study, according to Nicholas Reich (a biostatistician and infectious-disease researcher at the University of Massachusetts, Amherst) is that "relying on individual models is not the best approach. Combining or synthesizing several models provides the most accurate short-term predictions." [Rob]

Despite the existence of point and probabilistic forecast, we will focus in the first one. The second one gives us not a determined value for the prediction, it allows the uncertainties of the forecasts to be evaluated. However these techniques require additional concepts, and the results are less understood than the combination of point forecasts.

Finally, we will conclude this tesis with a couple of experiments that will show us whether and how ensemble methods can improve the traditional forecasting models. These experiments will be performed with the help of a computational tool, the package Kats. We will try through the source code to extend the functionalities of this tool in order to contribute to open source.

Chapter 2

Preliminars

2.1 Introduction of time series

In this part we are going to set certaint concepts in order to understand the goal of this research. The main concept we need to make clear about is the definition of time series.

To reach this understanding we must first define a more general concept.

Definition 2.1.1 (Stochastic process)**.** *A stochastic process is a random variable family* $\{X_t, t \in T\}$ *defined over a common probability space* $(\Omega, \mathcal{A}, \mathbb{P})$ *, indexed by the element of the set* T*. [BD16]*

Definition 2.1.2 (Time series)**.** *A time series* $\{X_t\}_{t\in T}$ *is a stochastic process where its dependent variable* t *is the time. [BD16]*

A time series is a *discrete-time time series* if the set of times, with which we index the values x_t, is a discrete set, e.g., when $T = \{1, 2, 3, .., 200\}$. Similarly, a time series is a *continuous time series* if the observations are recorded continuosly over some time interval, e.g., when $T = [0, 1]$ [BD16]. The most important are the discrete ones, which will be our target.

In the practice, sometimes we see a time series like a realization of this specific type of stochastic process. In this case we denote by $\{x_t\}_{t\in T}$ the series of values observed. In addition, there is sometimes an abuse of notation by denoting T and its cardinality in the same way.

Note: We shall frequently use the term *time series* to mean both the data and the process of which it is a realization. [BD16]

Note: Sometime T refers to the set of index, and sometimes the letter T stands for $|T|$

Examples of time series abound in such fields as economics, business, engineering, etc. [Box08] Throughout this study, we will work mainly with "Air Passenger Data for Time Series Analysis" [Kaga] and "Daily Visitors to the Website Data" [Kagb].

When we have to study a time series we need to perform a time series analysis.

2.1.1 Steps of a time series analysis:

1. **Graphical representation:** To have a general idea and start to identify the time series features o characteristic.
2. **Modeling data:** We will be discussed it in more detail in the section 2.2.
3. **Model validation:** We need to check the model before making predictions, to see if they would be valid.
4. **Forecasting** the future values of the time series. It is based on the current and past values.

Next we will introduce the ideas of dependence (or autocorrelation), stationarity and seasonality, the main characteristics of a time series.

As mencioned, the **dependence** between adjacent observations is an intrinsic feature of a time series. The nature of this dependence among observations is of considerable practical interest. [Box08]. When we talk about *time series analysis* we include the techniques for the analysis of this dependence.

Definition 2.1.3 (Mean and covariance functions)**.** *Let* $\{X_t\}$ *be a time series with* $E(X_t^2) < \infty$.

The mean fuction of $\{X_t\}$ *is* $\mu_t = E(X_t)\ \forall t \in T$

The covariance fuction of $\{X_t\}$ *is* $\gamma_{r,s} = Cov(X_r, X_s) = E((X_r - \mu_r)(X_s - \mu_s))$ $\forall \{r, s\} \in T$

The conditions imposed on the time series to make them stable for prediction are known as **stationarity**.

Definition 2.1.4 (Stationarity of time series)**.** *A time series $\{X_t\}$ is stationary if :*

- *The mean is independent of t :* $\mu_t = \mu \quad \forall t \in T$
- *The covariance is independent of t for each k :* $\gamma_{t,t+k} = \gamma_k \quad \forall t \in T$

The value k refers to the lag.

Definition 2.1.5 (ACVF)**.** *Let $\{X_t\}$ be a stationary time series.*
*The **autovariance function** (ACVF) of $\{X_t\}$ at a lag k is*
$\gamma(k) = \gamma_k = Cov(X_t, X_{t+k})$
seing the autocovariance as a funtion of k.

Definition 2.1.6 (ACF)**.** *Let $\{X_t\}$ be a stationary time series.*
*The **autocorrelation function**(ACF) of $\{X_t\}$ at a lag k is*
$\rho(k) = \frac{\gamma_k}{\gamma_0} = Cor(X_t, X_{t+k})$

Example 2.1.7 (White Noise)**.** *Let $\{\epsilon_t\}$ be a stationary stochastic process, it is denoted as white noise if:*
$E(\epsilon_t) = 0 \quad \forall t$
$V(\epsilon_t) = \sigma^2 \quad \forall t$
$Cov(\epsilon_t, \epsilon_s) = 0 \quad \forall t \neq s$
This is indicated by the notation $\{\epsilon_t\} \sim WN(0, \sigma^2)$

The **seasonality** of a time series is refered to the periodic behavior of the series. The series is similar after s instants of time, where s is the *period*.

2.2 Time series models

Definition 2.2.1 (Time series model)**.** *A time series model for the observed data is a specification of the joint distributions of a sequence of random variables $\{X_t\}$ of which the values $\{x_t\}$ are postulated to be a realization. [BD16]*

Remarking the diferences between $\{X_t\}$ and $\{x_t\}$ is important. We denote with capital letters the random variable of a stochastic process, while we denote with lower case letters the observed value of this variable.

As we have already mentioned the idea is to try to model X_t for the purpose of **forecasting** .

Let us a look at the different models implemented in the *kats library*, wich will be explained in more detail in the Chapter 4. For each of them we will give an idea of the logic behind it and how we can put it into practice using this library. [Faca].

2.2.1 Models without seasonality

2.2.1.1 Linear and Quadratic

The simplest model is the linear one. We forecast the time series assuming that it has a linear relationship with other variable (or other variables).
$X_t = \beta_0 + \beta_1 Y_{1,t} + \ldots + \beta_k Y_{k,t} + \epsilon_t \quad$ for $\quad t = 1 \ldots T$ being T the number of observations.

In this case X is the **forecast variable** (or regressand) and Y the **predictor variables** (or regressors).

ϵ refers to the random error about which we also make certain assumptions:

- normality with mean zero and same variance.
- they are not autocorrelated.
- they are unrelated to the predictor variables.

In this way, we can define a confidence interval for future predictions given a confidence level, $1 - \alpha$.

$$IC(\alpha) = \hat{X} \mp Z_{(1-\alpha/2)}\hat{\sigma}_e \sqrt{1 + \frac{1}{T} + \frac{(Y - \overline{Y})^2}{(T-1)s_Y^2}}$$

We will now look at a couple of specific models that take the variable time t as a predictor variable, $Y_{k,t} = t^k$:

In the function `kats.models.quadratic_model.LinearModel` of library, they take the variable time t as a predictor variable taking values from $t = 0$ to $t = T - 1$, then apply an Ordinary Least Squares (OLS) to fit the value of our time series: $X_t = \beta_0 + \beta_1 t + \epsilon_t$. That is, taking $k = 1$.

On the other hand, the function `kats.models.quadratic_model.QuadraticModel` works likewise but taking t and t^2 as predictor variables: $X_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$. That is, taking $k = 2$

In both cases the input is the time series and the α value for the confidence level we want to obtain. [HA21]

2.2.1.2 ARIMA

ARIMA model provides another approach to time series forecasting in this case with the objective of describing the autocorrelations in the time series. Before we are going to introduce some notation:

Definition 2.2.2 (Lag operator)**.** $LX_t = X_{t-1}$

Definition 2.2.3 (Differencing operator)**.** $\nabla X_t = (1 - L)X_t = X_t - X_{t-1}$

Definition 2.2.4 (sth-order Differencing operator)**.** $\nabla_s X_t = X_t - X_{t-s}$

Note: It is not equal $\nabla^s X_t$ and $\nabla_s X_t$.

For example $\nabla^2 X_t = (1 - L)^2 X_t = (1 - L)(X_t - X_{t-1}) = X_t - 2X_{t-1} + X_{t-2}$
while $\nabla_2 X_t = X_t - X_{t-2}$

AR(p) model or autoregressive model (Auto-Regression, AR, indicates that the variable is regressed against itself) of order p can be writen as:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t \quad \{\epsilon_t\} \sim WN(0, \sigma^2)$$

MA(q) model or moving average model of order q can be griten as:

$$X_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \quad \{\epsilon_t\} \sim WN(0, \sigma^2)$$

ARIMA(p,d,q) model is the result of combining autoregression and moving average models, where I refers to "integration", because we have to apply d- differences ($\nabla^d X_t$) to the original times series to make it stationary.

$$\nabla^d X_t = c + \phi_1 \nabla^d X_{t-1} + \ldots + \phi_p \nabla^d X_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \quad \{\epsilon_t\} \sim WN(0, \sigma^2)$$

[HA21]

We can use this model with the `kats.models.arima.ARIMAModel` function. It takes as inputs:
p: order of the autoregressive part.
d: degree of the differencing.
q: order of the moving average part.

2.2.2 Models with seasonality

This is the classical decomposition model:

$$X_t = f(T_t, S_t, I_t)$$

T_t : A slowly changing function known as a trend component.
S_t : A function (with period s) referred to a seasonal component.
I_t : A random noise component.

This model can be:

- **Additive**: $X_t = T_t + S_t + I_t$, when the seasonal variations are constant through the series. The seasonal component is expressed in absolute terms.
- **Multiplicative**: $X_t = T_t \times S_t \times I_t$, when the seasonal variations are proportional to the level of the series. In this case, the seasonal component is expressed in relative terms. [HA21]

2.2.2.1 STLF

"Seasonal and Trend decomposition using Loess, Forecast". This method is based in the STL decomposition using LOESS (local regression) to model trend and seasonal component.

In the forecast with STL we use this decomposition. Once we obtain the de-seasonalized time series component, we apply a standard forecasting (such as ARIMA, linear, quadratic) to $T(t)$ and generate an h-step ahead forecast $T(t + h)$. Finally, we sum the seasonal component to $T(t + h)$ to obtain a the forecast.

It can be used with the function `kats.models.stlf.STLFModel`. You must give it as input the standar forecasting model to fit the de-seasonalized component, and the period of the seasonal component.[Faca]

2.2.2.2 Holt-Winter

Originally, there was a simple exponential smoothing to model data without trend and seasonality. Holt (1957) extended simple exponential smoothing to allow data with a trend, and later Winter (1960) extended Holt's method in order to capture the seasonal part of the time series.

In the final model, we have three smoothing equations: one for the level ℓ_t, one for the trend b_t (both together, $\ell_t + hb_t$, take the role of the T_t component) and one for the seasonal component s_t (it takes the role of S_t) , each one with the corresponding smoothing parameters α, β, γ. Emphasize that this model ignores the random noise component I_t. In this case, we denote the period as m.

Holt-Winters' additive method:

$$\begin{aligned} X_{t+h|t} &= (\ell_t + hb_t) + s_{t+h-m(k+1)} && (2.2.1)\\ \text{with } \ell_t &= \alpha(X_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) && (2.2.2)\\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} && (2.2.3)\\ s_t &= \gamma(X_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, && (2.2.4) \end{aligned}$$

with k being the interger part of $(h-1)/m$, thus the estimations of the seasonal indices come from the last s periods of the sample. For instance, if $m = 12$ and $h = 26$ we obtain $k = 2$. Therefore, our seasonal component will be $s_{t+h-3m} = s_{t-10}$, least than a year ago since the current time.

The level equation (2.2.2) is a weigthed average between the deseasonally adjusted observation $(X_t - s_{t-m})$ and the non-seasonal forecast $(\ell_{t-1} + b_{t-1})$.

The trend equation (2.2.3) is a weigthed average between the estimated slope $(\ell_t - \ell_{t-1})$ and last estimated slope b_{t-1}.

The seasonal equation (2.2.4) in t, is a weigthed average between the estimated seasonal component $(X_t - \ell_{t-1} - b_{t-1})$ and estimation in the last same period s_{t-m}.

Holt-Winters' multiplicative method:

$$\begin{aligned} X_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)}\\ \ell_t &= \alpha\frac{X_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})\\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}\\ s_t &= \gamma\frac{X_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}. \end{aligned}$$

It has the same idea as the additive but considering that the seasonal component is multiplied, not added. [HA21]

The model can be applied by the function `kats.models.holtwinters.HoltWintersModel` with no mandatory inputs. One can give it as an input if the trend is 'additive' or 'multiplicative'.[Faca]

2.2.2.3 Prophet

"Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. Prophet is open source software released by Facebook's Core Data Science team. It is available for download on CRAN and PyPI."[DG]

It works similarly to Holt-Winter, it combines seasonality, trend, and holidays.

$$X_t = T(t) + S(t) + H(t) + \epsilon_t \tag{2.2.5}$$

where:
$T(t)$: piecewise linear or logistic growth curve that models non-periodic changes (trend).
$S(t)$: models the seasonality.
$H(t)$: models the effects of holidays with irregular schedules.
ϵ_t : error that takes into account for any other change that is not modeled. It take the role of I_t.
[DG]

The novelty of this model is the incorporation of a component $H(t)$ that models the "irregularities" inherent to the periods of a calendar year.

We can use this model through the `kats.models.prophet.ProphetModel` function with several inputs where we can choose the T function (and its parameters), and seasonalities parameters. As mentioned, $T(t)$ can be a piecewise function. Apart from choosing if the trend must be linear o logistic, we can help the model to determine the changepoints.[Faca]

By default, Prophet specifies 25 potential changepoints which are uniformly placed in the first 80% of the time series. The number of potential changepoints can be set using the argument `n_changepoints` and the range where they are uniformly placed can be set with `changepoint_range` e.g.,`changepoint_range=0.9` placed the changepoints in the first 90% of the time series. If one wishes, rather than using

automatic changepoint detection you can use `changepoints` argument specifying manually the locations of potential changepoints, e.g.,`changepoints=['2014-01-01']` [Facb]

Next we will see more complex models that do not follow the decomposition defined at the beginning of the section.

2.2.2.4 SARIMA

SARIMA (Seasonal ARIMA) is formed by including additional seasonal elements in the ARIMA models we have seen up to now.

In a $SARIMA(p, d, q) \times (P, D, Q)_s$ model the seasonal component has a period of s, thus we modeled the time series with lag s using another ARIMA model with parameters P, D and Q. [HA21]

This model can be used with the `kats.models.sarima.SARIMAModel` function. It takes as inputs:
p: order of the autoregressive part.
d: degree of the differencing.
q: order of the moving average part.
(P, D, Q, s): order of the autoregressive, differencing and moving average of the seasonal part, and its period. [Faca]

2.2.2.5 LSTM

The LSTM (Long short-term memory) model is a recurrent neural network (RNN) model that may be used for sequential data, capable of learning long-term dependencies. The RNN were introduced by Hochreiter & Schmidhuber (1997). [Lstb]

Each line in Figure 2.1 carries a full vector from one node's output to another's input. The yellow boxes are learnt neural network layers, whereas the pink circles are pointwise operations like vector addition. Concatenation is indicated by lines merging, whereas lines forking indicate that their content has been replicated and is being sent to other destinations.[Lstb]

To discuss in more detail, each layer of the RNN calculates the following function:

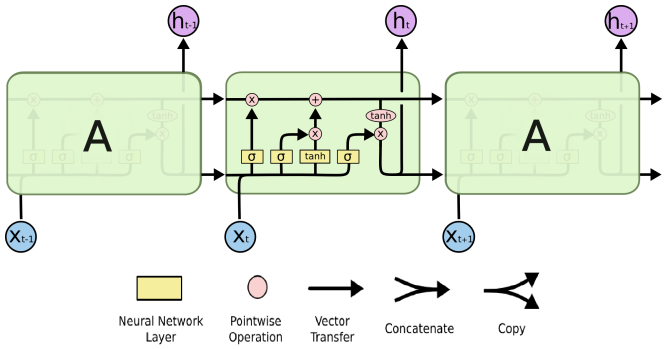

Figure 2.1: The Long Short-Term Memory (LSTM)
Source: [Lstb]

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}); \qquad f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$
$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}); \qquad o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \qquad h_t = o_t \odot \tanh(c_t)$$

where h_t is the hidden state at time t, its dimension is an input. c_t is the cell state at time t, x_t is the input at time t.

The horizontal line that runs through the top of the diagram and represents the cell state, it is the key of LSTMs. It is like a conveyor belt. With only a few tiny linear interactions, it runs directly down the entire chain. Information can very easily continue to travel along it unmodified.

The LSTM can modify the cell state by removing or adding information, which is carefully controlled via gates. They are composed out of a sigmoid neural net layer. This neural net takes values between zero and one describing how much of each component should be allowed through. A value of zero means "let nothing through," while a value of one means "let everything through!". [Lstb]

- $g_t \in (0, 1)$: cell gate, it remembers values. The information remembered depends on the three following gates. It protects and controls the cell state.
- $i_t \in (0, 1)$: input gate, it decides what new information to store in the current state.
- $f_t \in (0, 1)$: forget gate, it decides what information to discard from a previous state.
- $o_t \in (0, 1)$: output gate, it decides what actual information to output.

- $W_{**}, b_{**} \in \mathbb{R}$: weights and bias vector of each gates, that must be learned during the training.

σ is the sigmoid function which is used as activation function, and $\odot$ is the Hadamard product. [Lsta]

In the kats library we could apply this model with the function `kats.models.lstm.LSTMModel` that takes the size of the hidden unit, the time series sequence length that feeds into the model and the number of epochs for the training process. [Faca]

2.2.2.6 Theta

The first and original description of the method was given by Assimakopoulos and Nikolopoulos (2000). It involves several algebraic concepts. We will make a general presentation of the original method, but we will focus on the method expressed much more simply. The forecast obtained are equivalent to simple exponential smoothing with drift.

First, if the period is significantly different from zero, then the data is de-seasonalized.

Let $\{x_t\}$ denoted the observed time series. Then we construct a new series $\{y_{t,\theta}\}$ such that: $\nabla^2 y_{t,\theta} = \theta \nabla^2 x_t$. It is a second-order difference equation whose solution is:

$$\nabla^2 y_{t,\theta} = a_\theta + b_\theta(t-1) + \theta x_t$$

For a fixed θ we minimize $\sum_{i=1}^{t}[x_t - y_{t,\theta}]^2$ with respect to a_θ and b_θ, obtaining the solutions $\hat{a_\theta}$ and $\hat{b_\theta}$ being the mean value of the time series the same as the original one $\overline{y_\theta} = \overline{x}$

Forecast from the Theta model is obtained by a weighted average of forecast of $y_{t,\theta}$ for different values of θ. This is the main idea of the original model, and the reason for its name.

As mentioned above, there is another solution which is defined in the library, underlying stochastic models: we initialize the model by setting $X_1 = \ell_1$ and then for $t = 2 \dots T$

$$\begin{aligned} X_t &= \ell_{t-1} + b + \epsilon_t \\ \ell_t &= \ell_{t-1} + b + \alpha\epsilon_t \\ \{\epsilon_t\} &\sim WN(0, \sigma^2) \end{aligned}$$

Then X_t follows forecasts equivalent to SES (Simple Exponentias Smoothing): note that $X_t = X_{t-1} + b + (\alpha - 1)\epsilon_{t-1} + \epsilon_t$ is an ARIMA(0,0,1).

As a result we notice that:

$$\begin{aligned} X_{t+h} &= \ell_t + hb \\ X_{t+1} &= X_t + b + (\alpha - 1)\epsilon_t \\ \text{and } \epsilon_t &= X_t - X_{t-1} - b + (\alpha - 1)\epsilon_{t-1} \end{aligned}$$

by repeatedly substituting these equations we obtain:

$$\begin{aligned} X_{t+1} &= \widetilde{X}_{t+1} + \frac{b}{\alpha}[1 - (\alpha - 1)^n] \\ X_{t+h} &= \widetilde{X}_{t+1} + b[h - 1 + \frac{1}{\alpha} - \frac{(\alpha - 1)^n}{\alpha}] \end{aligned}$$

where $\widetilde{X}_{t+h}$ refers to the SES forecast since $\epsilon_1 = 0$

Finally, the forecasts are reseasonalized if it is needed.[HB03]

To apply this model implemented in the library, one needs to use the function `kats.models.theta.ThetaModel` which takes the period m as input.[Faca]

2.2.2.7 Harmonic Regression

For long seasonal periods we can use Fourier terms in order to model the seasonal part of the time series. Jean-Baptiste Fourier was a French mathematician, born in 1768, who demonstrated that a series of sine and cosine terms of the appropriate frequencies can approximate any periodic function. We can use them for seasonal curves.

If s is the seasonal period, the first few Fourier terms are:
$x_{1,t} = \sin\left(\frac{2\pi t}{s}\right), x_{2,t} = \cos\left(\frac{2\pi t}{s}\right), x_{3,t} = \sin\left(\frac{4\pi t}{s}\right),$
$x_{4,t} = \cos\left(\frac{4\pi t}{s}\right), x_{5,t} = \sin\left(\frac{6\pi t}{s}\right), x_{6,t} = \cos\left(\frac{6\pi t}{s}\right),$

A regression model containing Fourier terms is often called a harmonic regression. This makes them useful for weekly data, for example, because $s \approx 52$. [HA21]

In kats library we can apply this model with the `kats.models.harmonic_regression.HarmonicRegressionModel` function, giving to it the period s and the max order of the Fourier terms to be used.[Faca]

2.3 How to measure error

We can select as loss function mean squared error ($MSE = mean(e_t)$) but we can also take other types of metrics such as:

- Mean absolute percentage error : $MAPE = mean(|p_t|)$
- Symmetric mean absolute percentage error: $sMAPE = mean(200 \cdot |e_t|/(x_t + \hat{x}_t))$.
- Mean absolute error: $MAE = mean(|e_t|)$
- Mean absolute scaled error: $MASE = mean(|q_t|)$
- Root mean squared error: $RMSE = \sqrt{mean(e_t^2)}$

Indicating the error as $e_t = x_t - \hat{x}_t$, the percentage error as $p_t = 100 \cdot e_t/x_t$ and the scaling error as $q_t = \frac{e_t}{\frac{1}{T-1}\sum_{i=2}^{T}|x_i - x_{i-1}|}$.

But before choosing a loss function to minimize, we have to define how to measure the error e_t. What data do we take to predict the value x_t?

The first idea that comes to mind if we have $x_1, \dots x_T$ values is: to set $R < T$ and take the values $x_1, \dots, x_R$ to model the time series, then, predict $x_{R+1}, \dots x_T$ and calculate the error. We can see the idea in Figure 2.2, with the blue dots being the values used to model the predictions, and the orange dots being the known values we predict to calculate the errors.

Figure 2.2: Basic scheme
Source: [HA21]

First of all, let us take a look at the general notation of the other methods. Then we will look at them one by one in more detail.

Let $\tau \geq 1$ be the prediction horizon of interest, and we know $T + \tau$ values of the time series. There are P predictions in all. The methods use some data from period R or earlier to predict $R + \tau$, some from $R + 1$ or earlier to predict $R + \tau + 1$ event, ..., the some from $R + P - 1 \equiv T$ or earlier to predict $T + \tau$.

We can apply three types of structures taking into account that the series values have a fixed order.

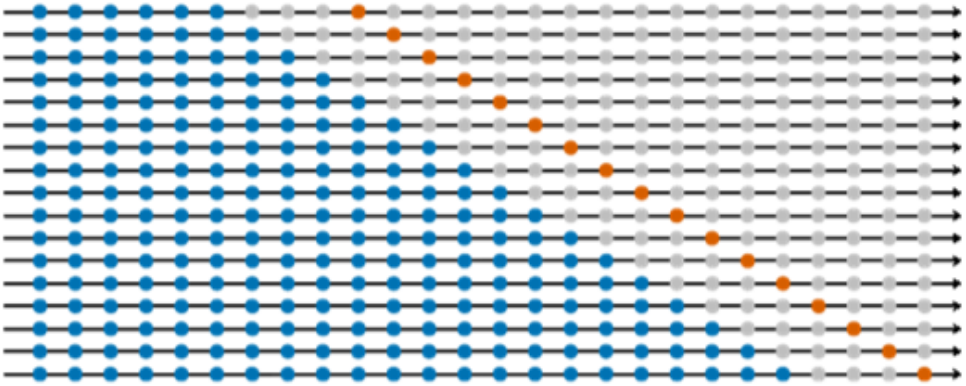

Figure 2.3: Recursive or expansive scheme
Source: [HA21]

The first one, which we call *recursive*, uses the data from 1 to R to calculate the first error, then uses from 1 to $R+1$ for the second,..., and finally with the data from 1 to T calculates the last error. This structure is represented in Figure 2.3.

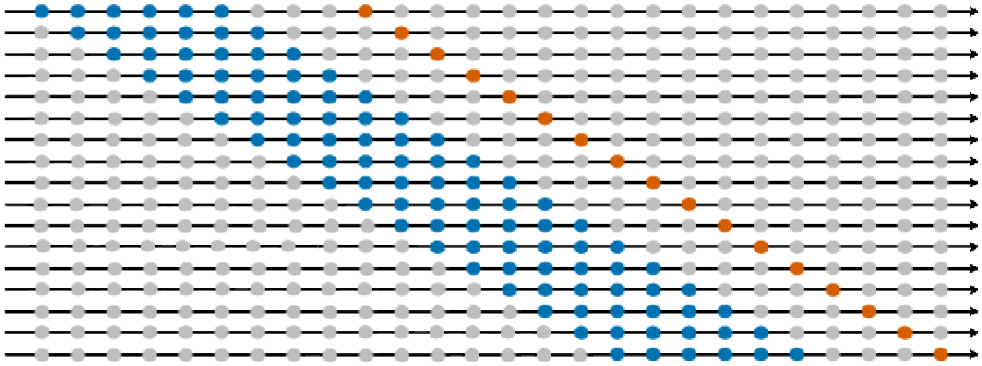

Figure 2.4: Rolling scheme
Source: Own creation based on [HA21]

The second scheme, which we call *rolling*, uses the data from 1 to R to calculate the first error, then uses from 2 to $R+1$ for the second,..., and finally with the data from $T-R+1 \equiv P$ to T calculates the last error. Its representation is in Figure 2.4. Its name comes from the fact that the origin at which the forecast is based rolls forward in time.

And the last structure, which we call *fixed*, uses the data from 1 to R to compute all P predictions with their corresponding errors. Its representation can be seen in Figure 2.5.[WM98]

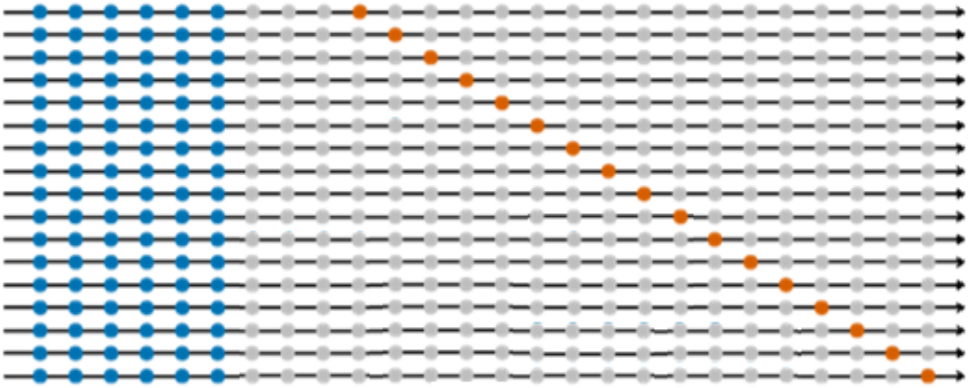

Figure 2.5: Fixed scheme
Source: Own creation based on [HA21]

Aditionally, if we apply those ideas with $\tau > 1$, instead of with $\tau = 1$, we can focus on the long-range forecasting capability of the model.

Chapter 3

Point forecast combinations

Forecast combinations of multiple individual forecasts dates back at least to Francis Galton, who in 1907 asked 787 villagers to guess the weight of an ox. None of them got the correct answer. Sir Galton averaged their guesses estimations and he arrived at a near perfect estimate. That was one of the motivations of "wisdom of the crowds" book. [Yon13]

About sixty years later, in 1969, the work of Bates and Granger [BG69] popularized the idea of forecast combination. The proportion of papers that concern forecast combinations among all published forecasting papers has been increased from near to 0% to almost 15% in the last fifty years, a symbol of the importance and usefulness of this techniques.[Wan+22]

We take the individual forecasts to be combined as given, we do not study how it has been generated. We focus our attention to combinations of multiple forecasts derived from separate modeling for the time series.

When several forecast are available, it is natural to try and find a linear combination of these forecasts that is the "best" in certain term.

3.1 Bates & Granger

As mentioned, Bates and Granger began with the discussion of combining predictions. In their article, they are in the case in which two forecasts have been made. The first reaction is to attempt to discover which is the best forecast. They noticed that we should not discard any forecast, since it always contains some useful independent

information. This information may be:

- A forecast is based on variables that are not taken into account in other forecasts, although this does not always help us to achieve a better solution.
- The assumptions about the relationship between variables in the discarded forecast are different.

They had an important assumption: the individual forecasts should be unbiased.

The equal-weights combined forecast is a good option in some cases, as we will see in section 3.5. They proposed to achive weight to forecasts that give us the smallest errors (mean squared). The problem is what is the best way to do this, as there are many ways to determine this.

The objective was to choose the combination that would produce the least forecasting errors, their first idea was derived in the following way. They assumed that the individual forecast are consistent, that is, the variance of error are independent of time, it could be denoted by σ_1^2 and σ_2^2 for all values of t.

First we will lay the foundations of the different methods proposed by Bates and Granger.

The combined forecast would be obtained by a linear combination, giving a weight k to the first forecast and $(1-k)$ to the second one. They denoted as σ_c^2 the variance of errors in the combined forecast:

$$\begin{aligned}\sigma_c^2 &= k^2\sigma_1^2 + (1-k)^2\sigma_2^2 + 2k(1-k)\sigma_{12} \\ &= k^2\sigma_1^2 + (1-k)^2\sigma_2^2 + 2\rho k\sigma_1^2(1-k)\sigma_2^2\end{aligned}$$

ρ being the correlation coefficient between the errors in the first forecast and those in the second one.

We take σ_c^2 as a function of k. Differentiating with respect to k and equating to zero, we obtain:

$$\begin{aligned}(\sigma_c^2)' &= 2k\sigma_1^2 - 2(1-k)\sigma_2^2 + 2(1-2k)\sigma_{12} \\ &= 2k(\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}) - 2(\sigma_2^2 - \sigma_{12}) \quad = 0\end{aligned}$$

$$\begin{aligned}k &= \frac{\sigma_2^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}} \\ &= \frac{\sigma_2^2 - \rho\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1^2\sigma_2^2}\end{aligned} \tag{3.1.1}$$

Note: In the scenario where $\rho = 0$, the equation (3.1.1) reduces to: $k = \sigma_2^2/(\sigma_1^2+\sigma_2^2)$

If k is determined by equation (3.1.1), one can show that the value σ_c^2 is not greater than the smaller of the two individual variances, i.e, $\sigma_c^2 \leq min\{\sigma_1^2, \sigma_2^2\}$

As mentioned above, the equation (3.1.1) is used as basis for some methods that follow shortly.

At the begining we do not know the optimal value for k, it would change as empirical evidence on the relative effectiveness of the two original forecasts accumulates. Therefore the weights must be dynamic. Thus the combined forecast for the period T, C_T is defined as follow.

$$C_T = k_T F_{1,T} + (1 - k_T) F_{2,T}$$

$F_{1,T}$ and $F_{2,T}$ being the forecast at time T from the first and second model, respectively.

We will discuss the desirable properties that methods for determining k values should have.

- The average weight of k should be around the optimal value defined in 3.1.1 when the number of forecast increased.
- The dispersion should be small. Weights should vary only slightly above the optimal value.
- The weights must be addapted rapidly to the new values if there is a durable change in the performance of one of the forecasts.

Apart from these properties, moderately simple methods are desirable. Bates and Granger examined five methods in their paper. The weights k_T have all cases been determined from past errors of the two series denoted as $e_{1,1}, e_{1,2} \ldots e_{1,T-1}$ and $e_{2,1}, e_{2,2} \ldots e_{2,T-1}$. Except for k_1, since it has no past values, we can take a random value, for example 0.5.

The methods are:

(i) Taking $E_2 = \sum_{t=T-v}^{T-1} (e_{2,t})^2$, similarly E_1, v being the largest lag used to measure the error

$$k_T = \frac{E_2}{E_1 + E_2}$$

(ii)

$$k_T = \alpha k_{T-1} + (1 - \alpha) \frac{E_2}{E_1 + E_2}$$

where $\alpha \in [0, 1]$ is a constant that measures the importance given to the last value against the last $v-$ values.

(iii) Let us take $S_2^2 = \sum_{t=1}^{T-1} w^t (e_{2,t})^2$ where: w is the weight given to the first squared error, w^2 to the second squared error and so on. Usually $w \geq 1$ because it gives more weight to recent error variances.

$$k_T = \frac{S_2^2}{S_1^2 + S_2^2}$$

(iv) Following the same logic as (iii) Bates and Granger take $C = \sum_{t=1}^{T-1} e_{1,t} e_{2,t}$ as the weighted covariance

$$k_T = \frac{S_2^2 - C}{S_1^2 + S_2^2 - 2C}$$

(v) This is similar to (ii) but in this method we only take information from the last absolute value of error

$$k_T = \alpha k_{T-1} + (1 - \alpha) \frac{|e_{2,T-1}|}{|e_{1,T-1}| + |e_{2,T-1}|}$$

Not all methods have all the desirable properties, for example the method (v) fails to satisfy that the average weight should be around the optimal value.

One of the techniques to achieve the third property is to give more importance to recent values. For this reason, methods (iii), (iv) and (v) give us very good results for time series with sudden changes.

Bates and Granger concluded that the proposed methods for combining forecasts with dynamic weights can often result in better forecasts than those resulting from applying a static weighting determined after taking note of all individual forecast errors.

As mentioned, these methods are applicable only when we have two individual predictions. As a result of this, other techniques have emerged with the intention of generalizing these ideas. [BG69]

3.2 Newbold & Granger

Newbold and Granger [NG74] extended the method to combinations of more than two forecasts.

Suppose we have M forecasts of X_T denoted by $\mathbf{F}'_{\mathbf{T}} = (F_{1,T}, F_{2,T} \dots F_{M,T})$ these individual forecasts being unbiased. Thus, the following linear combination will also be unbiased:

$$\begin{array}{l} \mathbf{C_T} = \mathbf{k}'_{\mathbf{T}}\mathbf{F_T} \quad \mathbf{k}'_{\mathbf{T}}\mathbf{1} = 1, \quad 0 \leq k_{i,T} \leq 1 \quad \forall i \in \{1 \dots M\} \\ \text{where} \quad \mathbf{k}'_{\mathbf{T}} = (k_{1,T}, k_{2,T} \dots k_{M,T}) \quad \text{and} \quad \mathbf{1}' = (1, 1 \dots 1) \end{array} \tag{3.2.1}$$

It is straightforward to show that taking

$$\mathbf{k_T} = (\Sigma^{-1}\mathbf{1})/(\mathbf{1}'\Sigma^{-1}\mathbf{1}) \quad \text{where} \quad \Sigma = \mathbf{E}(\mathbf{e_T}\mathbf{e}'_{\mathbf{T}}) \quad \text{and} \quad \mathbf{e_T} = X_T\mathbf{1} - \mathbf{F_T}$$

we obtain the minimum variance of the combined forecast error. Therefore, in general, we can find a smaller error with the combined forecast C_T.

In practice, we do not know the covariance matrix Σ, so we need to estimate it or to assume it diagonal, something similar to what Bates and Granger did with the correlation coefficient. Following the ideas that Bates and Granger proposed, the new suggestions are based on two principles: more importance should be assigned to the forecast that has performed better in the immediate past, and weight should be adapted to a possible non-stationary relationship over time.

Five of their suggested choices for one-step-ahead forecast for more that two individual forecasts are:

(i)
$$k_{i,T} = \left(\sum_{t=T-v}^{T-1} e_{i,t}^2\right)^{-1} \Bigg/ \left\{\sum_{j=1}^{M}\left(\sum_{t=T-v}^{T-1} e_{j,t}^2\right)^{-1}\right\}$$

(ii)
$$\mathbf{k_T} = (\hat{\Sigma}^{-1}\mathbf{1})/(\mathbf{1}'\hat{\Sigma}^{-1}\mathbf{1}) \quad s.t \quad 0 \leq k_{i,T} \leq 1 \quad \forall i \in \{1 \dots M\}$$
with $(\hat{\Sigma})_{i,j} = \frac{1}{v}\sum_{t=T-v}^{T-1} e_{i,t}e_{j,t}$

(iii)
$$k_{i,T} = \alpha k_{i,T-1} + \left[(1-\alpha)\left(\sum_{t=T-v}^{T-1} e_{i,t}^2\right)^{-1} \Bigg/ \left\{\sum_{j=1}^{M}\left(\sum_{t=T-v}^{T-1} e_{j,t}^2\right)^{-1}\right\}\right] \quad \alpha \in [0,1]$$

(iv)
$$k_{i,T} = \left(\sum_{t=1}^{T-1} W^t e_{i,t}^2\right)^{-1} \Bigg/ \left\{\sum_{j=1}^{M}\left(\sum_{t=1}^{T-1} W^t e_{j,t}^2\right)^{-1}\right\} \quad W \geq 1$$

(v)

$$\mathbf{k_T} = (\hat{\Sigma}^{-1}\mathbf{1})/(\mathbf{1}'\hat{\Sigma}^{-1}\mathbf{1}) \quad s.t \quad 0 \leq k_{i,T} \leq 1 \quad \forall i \in \{1 \dots M\}$$

$$\text{with } (\hat{\Sigma})_{i,j} = \left(\sum_{t=1}^{T-1} W^t e_{i,t} e_{j,t}\right) \Big/ \left(\sum_{t=1}^{T-1} W^t\right) \quad W \geq 1$$

In their evaluation studies on the combination of forecasts they had considered in detail only one-step ahead forecasts, although the methods proposed above can be readily extended to deal with forecasting several steps ahead.

An interesting note Newbold and Granger made is that sometimes optimal forecasts can be obtained from a single model, but we can never be absolutely sure that particular model is the right one. In fact, for small samples, the degree of uncertainty can be very high. It might be a better approach, given M univariate forecasting models, to stipulate a subjective probability k_i as the degree of belief that the i-th model represents the ground-truth stochastic process.

Suppose now that we wish to predict X_T from the past values $\{X_{T-j}, j > 0\}$. Assume that we knew that there exists a correct model being the ith one. We denote the density function $f_i(x_T)$, which measures the probability that, assuming model i, the time series will take the value x_T. Its mean is $F_{i,T}$. Given this assumption the optimal quadratic loss predictor would be $\hat{x_T} = F_{i,T}$. Now, in subjective terms, our intuition about the density of X_T are represented by the function:

$$f(X_T) = \sum_{i=1}^{M} k_i f_i(X_T)$$

The mean of this density function, which provides the optimal predictor in terms of quadratic loss, is given by Equation 3.2.1 and, therefore, one is naturally driven to look for forecasts of this form.

3.3 With L-estimators

One class of used location estimators is the family of L-estimators, which stands for "linear combinations of order statistics". Mean and median are included in this family, as well as trimmed means and Winsorized means.

If $F_{(i),T}$ is the ith order statistic for the individual forecasts $F_{1,T}, \dots F_{M,T}$ then, trimmed and Winsorized means are defined as follows:

- Trimmed Mean: $T(i) = \frac{1}{M-2i} \sum_{k=i+1}^{M-i} F_{(k),T}$

- Winsorized Mean: $W(i) = \frac{1}{M}\left[iF_{(i+1),T} + \sum_{k=i+1}^{M-i} F_{(k),T} + iF_{(M-i),T}\right]$

where i is an integer with $0 \leq i \leq n/2$.

These measures involve taking the i smallest and i largest forecasts and either deleting them or setting them equal to the $(i+1)$th smallest and $(i+1)$th largest forecast.

Victor Richmond R. Jose and Robert L. Winkler studied these methods empirically and they concluded: performance of the Winsorized mean seems to be a little less sensitive to the choice of i than the trimmed mean.[JW08]

Another of the simplest methods to cluster M forecasts is to choose the median of $F_{i,T}$, with $F_{i,T}$ being the prediction of the i-th model. If M is even, we can take the mean of the two central values.

As we have already mentioned our first idea once we have the different predictions is to do the simple average. However, here we have presented other methods that perform better in some cases, for example with outlier forecast values.

3.4 Regression-based weights

We will consider three alternative approaches to obtain linear combinations. It is shown that the best method is to add a constant term and not force the weights to sum to unity.

First we will set some general notation:

- $x = (x_1, x_2, \ldots, x_T)'$ is a $T \times 1$ vector of values of X_t, being the series $x_t, t = 1, \ldots, T$.
- $F_j' = (F_{j,0}, F_{j,1}, \ldots, F_{j,T-1})$ is a $1 \times T$ vector of the forecasts from the jth model.
- $\mathbf{F} = (F_1, F_2, \ldots, F_M)$ is a $T \times M$ matrix of forecasts values.
- $\mathbf{1}$ is a vector of 1s of appropiate dimension.

3.4.1 Method A

Let $\mathbf{F}\alpha$ be the unconstraint forecast, where α is a $M \times 1$ vector of weights for the individuals forecasts. The forecast error is $e_A = x - \mathbf{F}\alpha$. Suppose that α is determined so as to minimize the sum of squared errors of forecasts. That is,

$$\min_{\alpha} \quad (x - \mathbf{F}\alpha)'(x - \mathbf{F}\alpha)$$

The solution is given by:

$$\mathbf{F}'(x - \mathbf{F}\alpha) = 0 \quad or \quad \hat{\alpha} = (\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'x$$

Thus, the combined forecast is $C_A = \mathbf{F}\hat{\alpha} = \mathbf{F}(\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'x$ attaining as min sum of squared error:

$$E_A = (x - C_A)'(x - C_A) = x'x - x'\mathbf{F}\hat{\alpha}$$

This is nothing more than a regression of x against $F_1, \ldots, F_M$ with no constant term.

A pair of conditions are sufficient for zero combined forecast bias :

a) each forecast has zero error mean : $\mathbf{1}'x = \mathbf{1}'F_j$

b) the weights add up to 1: $\mathbf{1}'\hat{\alpha} = 1$

3.4.2 Method B

Now consider the case in which the weights are constrained to sum to unity.

$$\begin{aligned} \min_{\beta} \quad & (x - \mathbf{F}\beta)'(x - \mathbf{F}\beta) \\ \text{s.t.} \quad & \mathbf{1}'\beta = 1 \end{aligned}$$

Considering $\min_\beta (x-\mathbf{F}\beta)'(x-\mathbf{F}\beta)+2\lambda_B(\mathbf{1}'\beta-1)$ λ_B being a Lagrangian multiplier.

The solution is given by:

$$\mathbf{F}'(x - \mathbf{F}\beta) - \lambda_B\mathbf{1} = 0 \quad or \quad \hat{\beta} = \hat{\alpha} - \lambda_B(\mathbf{F}'\mathbf{F})^{-1}\mathbf{1}$$

with $\lambda_B = (\mathbf{1}'\alpha - 1)/[\mathbf{1}'(\mathbf{F}'\mathbf{F})^{-1}\mathbf{1}]$ due to the constraint $\mathbf{1}'\beta = 1$

Thus, the combined forecast is $C_B = \mathbf{F}\hat{\beta}$ attaining as min sum of squared error:

$$E_B = E_A + \lambda_B^2[\mathbf{1}'(\mathbf{F}'\mathbf{F})^{-1}\mathbf{1}]$$

Evidently $E_B \geq E_A$ and, therefore, there is a loss in the mean square error due to the constraint. This method is equivalent to regresing $(x - F_M)$ against $(F_1 - F_M), \ldots, (F_{M-1} - F_M)$ without intercept, taking the weight for F_M as 1- (sum of the remaining forecasts).

From condition a) of method A, the combined forecast is unbiased when each individual forecast is unbiased.

3.4.3 Method C

In this method of combining without constraint on the weights we add a constant term. Consider:

$$\min_{\delta} \quad (x - \delta_0 \mathbf{1} - \mathbf{F}\delta)'(x - \delta_0 \mathbf{1} - \mathbf{F}\delta)$$

δ_0 being the intercept and δ the weights for the M forecasts.

The normal equations are given by:

$$\mathbf{F}'(x - \delta_0 \mathbf{1} - \mathbf{F}\delta) = 0 \quad \text{and} \quad \mathbf{1}'(x - \delta_0 \mathbf{1} - \mathbf{F}\delta) = 0$$

obtainig

$$\begin{aligned} \hat{\delta} = &\quad \hat{\alpha} - \hat{\delta}_0 (\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'\mathbf{1} \\ \hat{\delta}_0 = &\quad (\mathbf{1}'x - \mathbf{1}\mathbf{F}\hat{\delta})/n \end{aligned}$$

Thus, the combined forecast is $C_C = \hat{\delta}_0 \mathbf{1} + \mathbf{F}\hat{\delta}$ achieving as min sum of squared error respect method A:

$$E_C = E_A - \frac{(\mathbf{1}\hat{e}_A)^2}{n - \theta} \leq E_A$$

where $\theta = \mathbf{1}'\mathbf{F}(\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'\mathbf{1}'$ and $\hat{e}_A$ is the vector of errors using method A.

The important idea is: method C is crearly the best because it gives the smallest mean squared error and has unbiased combined forecast even if individual forecasts are biased, that because we correct the bias with the constant term.

The usual practice of obtaining a weighted average of alternative forecasts should be rejected in favor of an unconstrained linear combination that incorporates a constant term.[GR84]

3.5 Forecast combination puzzle

Despite the complexity of the above sophisticated combination techniques, there are some empirical evidence to show that the simple average with equal weight sometimes outperforms complicated schemes. Stock and Watson coined the term "forecast combination puzzle" for this phenomenon [SW04].

What is the explanation for the robutness of the simple average of forecasts?

For example, Timmermann [Tim06] noted that the success of simple combination is due to the increased parameter estimation error with weighted combinations. The results of empirical studies claim that the cost of estimating weighted averages when the optimal weights are approximately equal is considerably high, which is an empirical explanation of the puzzle.

Explainig this requires a hypothesis that the potential gains from the "optimal" are not too large, so the estimator error overwhelms it. There are special cases, like where the covariance matrix has equal variances on the diagonal and all off-diagonal value are equal to a constant (so you have to estimate only two values for the matrix) that are ilustrated by Timmermann [Tim06] and Hsiao and Wan [HW14] to arrive at equivalence between the simple and the optimal average. Other researchers even characterized the potential bounds on the size of gains for which this occurs.

These studies give us some rules or markers to identify which combination method should be chosen in a specific forecasting problem:

- Too small sample size may be unable to provide robust estimates of the weights. Therefore, if we have access to limited historical data, simple mean or weights estimated by taking covariances between forecast errors as zero are recommended.
- Structural changes which may cause different weight estimates in the training and evaluation samples tend to impact sophisticated combination approaches more than the simple average. This case makes the simple average the best choice. Forecast combinations using changing or dynamic weights can also be considered as a means to cope with structural changes. [Wan+22]

Chapter 4

Computational implementation

In this section we will establish the necessary computational information to perform experiments in order to reaffirm the results studied.

We will make an introduction to the tool used, and then we will describe both, the functionalities that were already created and those that have been created for the research.

4.1 Kats

Kats (Kits to Analyze Time Series) is a user-friendly and flexible framework for doing time series analysis. It is a toolkit for analyzing time series data. It helps us to identify the most important statistics and features, spot anomalies or change point, and predict future. Using Kats, time series analysis can be done all in one place, including detection, forecasting, feature extraction and embedding, multivariate analysis, etc. Kats has been published by Facebook's Infrastructure Data Science team.[Faca]

The part that concerns us is forecasting and embedding. The experiments will be performed in Python 3.9 with the help of the version 0.2.0 of this tool that can be downloaded at the following link [Dow].

Kats gives us a set of tools for forecasting that includes the individual forecasting models showed in Chapter 2. In addition, it provides a set of functions for the emsembling methods studied. However, Bates and Granger method is not implemented in the framework. For the purpose of comparing methods, we will create a code extension to define a function that provides us with the ability to apply this ensemble method.

4.1.1 Data preparation

We need a specific type object to apply the functionalities to, `kats.consts.TimeSeriesData` is the basic data structure in Kats to define time series. Two ways to initialize it are available:

1. `TimeSeriesData(df)`: being `df` a `pd.DataFrame` with a column named 'time'. If the column with the time values is not called 'time', one can specify its name with the param `time_col_name`.

2. `TimeSeriesData(time,value)`: being `time` a `pd.Series` or `pd.DatetimeIndex` object and where `value` is a `pd.Series` or `pd.DataFrame`.

In a `TimeSeriesData` the time can be expressed as a variety of different types, including standard `datetime`, `pd.Timestamp`, string (in wich case we should use `date_format` argument to specify the structure) or even an integer (i.e unix time).

If we use unix time, we have to use the argument `use_unix_time=True` and we can specify the unit with the argument `unix_time_units` (by default nanosecod,'ns').

Several of the operations supported by `pd.DataFrame` are also supported by the `TimeSeriesData` object. For example:

- slicing
- math operations: sum, equality, etc.
- extend: with the method `ts.extend(ts_2)`
- ploting: with the method `ts.plot(cols = ['value'])`, we must pass the names of the value columns to plot.
- convert to `pd.DataFrame` with the method `ts.to_dataframe()`
- convert to `np.array` with the method `ts.to_array()`
- check basis characterictics of the time series: `ts.is_empty()` or `.is_univariante()`

4.1.2 Hyperparameter tuning

Kats offers classes that help one to quickly determine the optimal hyperparameters to utilize for a particular forecasting model. The method used is a static one called `create_search_method`. We have to specify:

- **selected_search_method** the type of search, usually `GRID_SEARCH`.
- **parameters** the search space for the parameters, defining a dictionary for each parameter and combining them into a list.
- **objective_name** string with the name o the objective function used for the search, usually "evaluation_metric".

4.1.3 Backtesting

We are going to do a short overview of the `kats.utils.backtesters` module. It is a module that makes it easy to compare and evaluate different forecasting models.

This module allows one to include multiple error metrics in a single function call. It supports the metrics presented in the Section 2.3.

Once we choose the metrics, Kats provides several types of backesters. We have the class `kats.utils.backtesters.BackTesterSimple` which executes a simple train/test backtest as we saw in Figure 2.2.

In addition, we have three different classes to apply the methods defined in abovementional Section 2.3:

- `BackTesterExpandingWindow` the expansive scheme studied.
- `BackTesterRollingWindow` the rolling scheme.
- `BackTesterFixedWindow` the fixed scheme.

4.1.4 Ensemble forecasting methods

As it has been mentioned, in addition to the individual forecast studied, this framework provides us with the following class in the `kats.models.ensemble` .

We have the base class on which the other classes are defined:

```
kats.models.ensemble.ensemble.BaseEnsemble(
    data:kats.consts.TimeSeriesData,
    params:kats.models.ensemble.ensemble.EnsembleParams)
```

The ensemble method assumes we have M base models whose names and parameters are defined in `params`.

In the following, we will study the different specific ensemble methods that are already implemented.

Median ensembling method:

```
kats.models.ensemble.median_ensemble.MedianEnsembleModel(
    data:  kats.consts.TimeSeriesData,
    params:  kats.models.ensemble.ensemble.EnsembleParams)
```

We take as a final prediction the median of the individual predictions. If M is even, we will take the mean of the center values.

Ensemble models with weighted average individual models:

```
kats.models.ensemble.weighted_avg_ensemble.WeightedAvgEnsemble(
    data:  kats.consts.TimeSeriesData,
    params:  kats.models.ensemble.ensemble.EnsembleParams)
```

It is based on backtesting results, we determine the weight for each model; a model with greater performance should have larger weight. We choose the function that measure the error with the argument `error_method`. By default it is mape (mean absolute percentage error).

Kats ensemble model:

```
kats.models.ensemble.kats_ensemble.KatsEnsemble(
    data:  kats.consts.TimeSeriesData,
    params:Dict[str, Any])
```

This is a specific pipeline implemented to improve the methods above. It begins by looking for seasonality. If it finds any, it proceeds to STL decomposition, as described in Section 2.2.2.1. Then it fits forecasting models to de-seasonalized components and finally it aggregates them. If it does not find sesonality, it just uses individual forecasting models and ensembling. This last part is made using median ensemble method or weighted average method, depending on the values defined in the dictionary.

Once the model has been instanced, it works like the models of the well-known Sklearn [Ped+11]. We fit the model simply by calling `model.fit()` and we make predictions with the method `model.predict(steps:int)` where `step` is the length of forecasting horizont. It returns the results as a `pd.DataFrame`.

4.2 Implementing an extension of Kats

Lastly, to increase the versatility of the package, we took the original repository and added and altered a few defined classes. Object-oriented programming (OOP) is the foundation of the Kats framework, which means that it organizes around objects rather than functions and logical expressions. Because of this, classes and the appropriate properties or methods are used to define all of these functionalities.

4.2.1 Mean Ensemble implementation

As we saw in Section 3.3, we can combine methods using L-estimators. The median one is already implemented in Kats.

Taking `MedianEnsembleModel` class as base, we create the `kats.models.ensemble.mean_ensemble.MeanEnsembleModel` class.

This method takes the M different predictions and returns the average value.

4.2.2 Bates & Granger implementation

Finally, we will study how the Bates and Granger method has been implemented.

This procedure, as we learned in Section 3.1 was only intended to aggregate two invididual forecasts. Due to this restriction, we develop a model based on the R function `comb_BG()` approach[Rdo]. The idea is essentially the same that was defined by Newbold and Granger years later.

We define the class:

`kats.models.ensemble.bates_granger_ensemble.BatesGranger Ensemble` based on the class made for weighted average method. The measured error was changed to mse (mean squared error) instead of mape (mean absolute per-

centage error). In addition, the expression of the weights was changed to those of the `comb_BG()` function, i.e, $w_i = \frac{\hat{\sigma}_i^{-2}}{\sum_{j=1}^{M} \hat{\sigma}_j^{-2}}$ where $\hat{\sigma}_i$ is the estimated mean squared prediction error of the i-th model.

By default the simple backtester is used as in the weighted average method. However we create the attribute `back_method` to the created class. By changing that with any of the two values we will mention below, we can control how we evaluate the error.

We will denote an instance of the defined class as `bt`.

If we put `bt.back_method = 'simple'` the error is measured using the 80% of the data for training and the 20% for testing, as described in Section 2.3.

If we define `bt.back_method = 'fixed'`, we use the researched fixed strategy, as discussed in Section 2.3. In this case training uses 75% of the data and testing uses 15%. We established a 10% data gap between the train and test data sets.

We choose the fixed window method because of its advantages in predicting the future with a certain time lag.

4.2.3 Other modifications

In order to keep the same structure in all classes, the attribute `back_method` was also included in the class `kats.models.ensemble.weighted_avg_ensemble.WeightedAvgEnsemble` with the same dynamic.

In addition, the class `kats.models.ensemble.kats_ensemble.KatsEnsemble` has also been modified to support Bates & Granger as base method. By default the class uses mape to measure the error, unless we use Bates & Granger, in which case the class uses mse.

All this new code can be consulted in the next repository [Cri].

Chapter 5

Experiments

For the purpose of illustrating the concepts presented in this tesis, a series of experiments will be conducted using real data in this chapter.

5.1 Data sets

5.1.1 Air Passenger Data

As mentioned at the begining, we will use *Air Passenger Data* [Dat] which contains San Francisco International Airport Report on Monthly Passenger Traffic Statistic by Airline.

There is a list of passenger data from 1949 to 1960. These data are seasonal in nature, since the behavior throughout the year is similar. In this case we have monthly data, so we will expect a period of 12.

This dataset contains 144 records of data. It is a powerful dataset to practice time series analysis, especially with seasonality. In addition, this data set has no missing value. [Kaga]

The data has only two columns:

- **Month:** A variable of type DateTime, ranging from January 1, 1949 to December 1, 1959. Its period is monthly.
- **Passenger:** Positive integer variable that determines the number of passengers

of the airline in the corresponding month.

Let us make a brief descriptive analysis of the data in Table 5.1.

	count	mean	std	min	25%	50%	75%	max
Passenger	144	280.30	119.97	104	180	265	360	622

Table 5.1: Table to describe the Air Passenger Data variable.

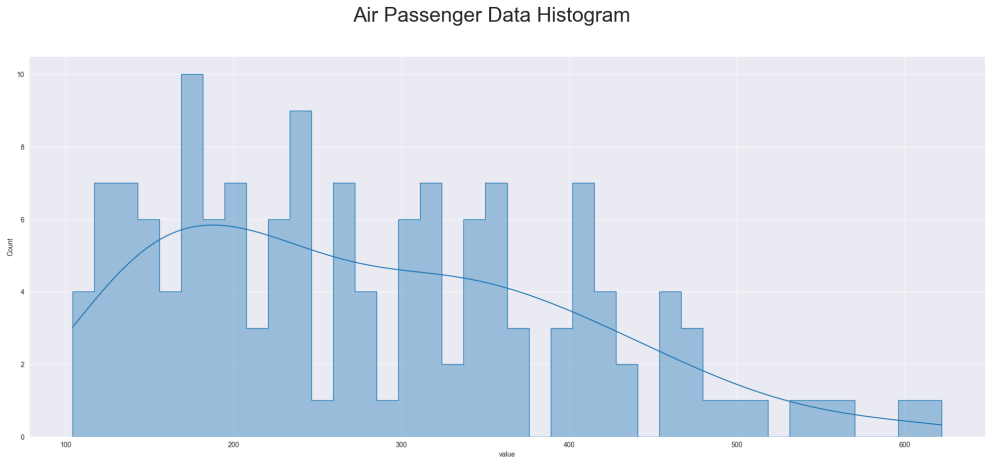

Figure 5.1: Air Passenger Data histogram

First we look at the histogram in Figure 5.1 in order to keep in mind the distribution of the variable. Finally, we plot the time series. As we can see in the Figure 5.2, the series has a repetitive behavior every year.

5.1.2 Daily Visitors to the Website Data

In order to have a more complete study, we will also examine *Daily Visitors to the Website Data*.[Kagb]

This data set includes five years worth of daily time series data returning four (although we only use three) traffic metrics from the statforecasting.com website. The variables exhibit seasonality that is correlated with both the academic calendar and the day of the week.

We are going to use four columns of the data:

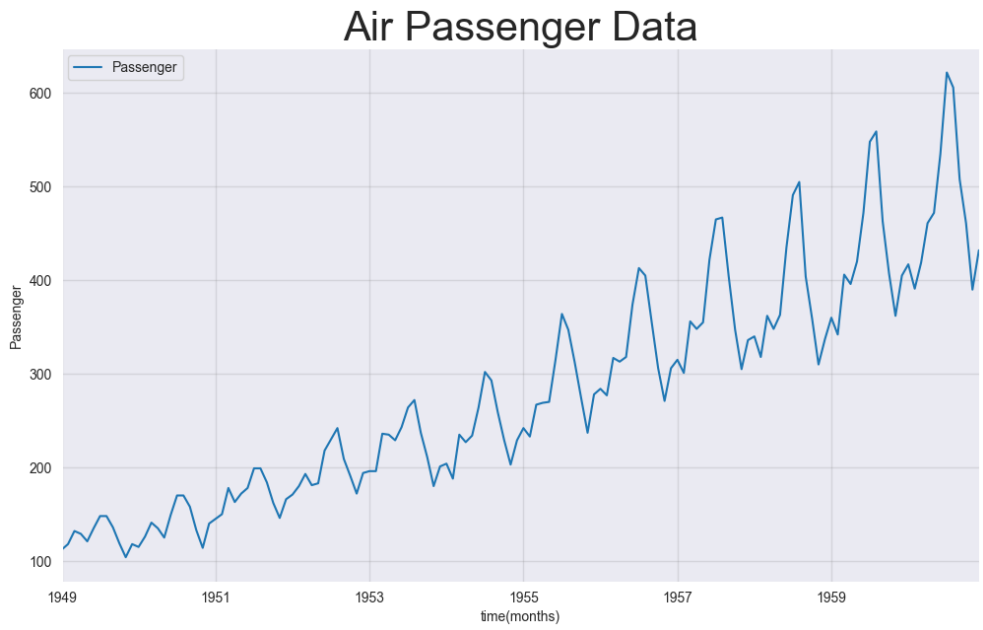

Figure 5.2: Air Passenger Data Time Series

- **Date:** A DateTime variable, ranging from September 14, 2014 to August 19, 2020.
- **Page Loads:** A variable of type integer, daily number of pages loaded.
- **Unique Visits:** Daily number of visitors from whose IP addresses there have not been hits on any page in over 6 hours.
- **First Time Visits:** Integer, that determines the number of distinct visitors who are not already identified by a cookie as past customers.

After a small descriptive preliminary study, we obtained the results shown in Table 5.2:

	count	mean	std	min	25%	50%	75%	max
Page Loads	2167	4116.99	1350.98	1002	3114.5	4106	5020.5	7984
Unique Visits	2167	2943.65	977.89	667	2226	2914	3667.5	5541
First Time Visits	2167	2431.82	828.70	522	1830	2400	3038	4616

Table 5.2: Table to describe the Daily Visitors to the Website Data variables.

In addition to the descriptive analysis performed, we will plot the histograms of its three variables. These can be seen in Figure 5.3.

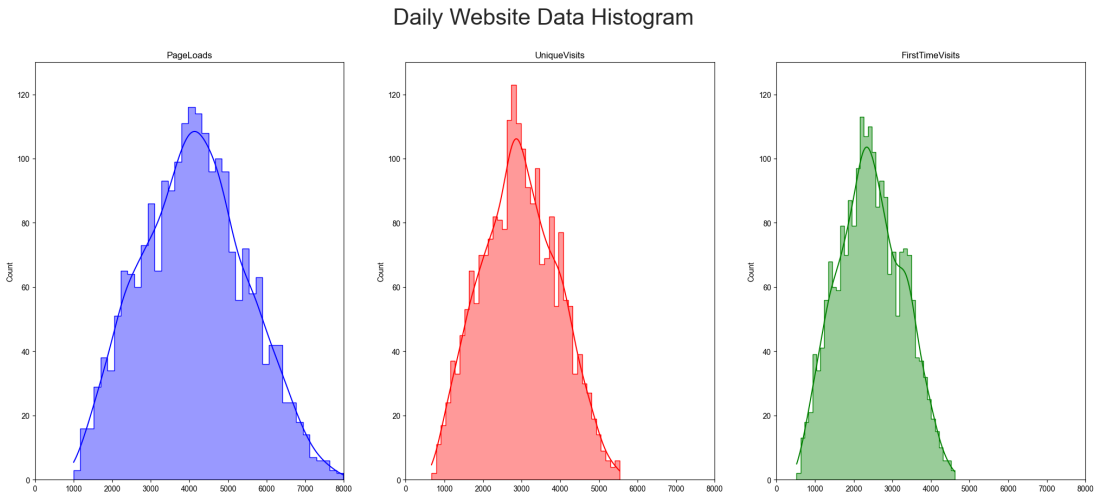

Figure 5.3: Daily Visitors to the Website Data

Now, we will look in Figure 5.4 at the time series plot, in order to identify seasonality.

We have a large amount of data, so to clarify the series, on the one hand we will represent daily data of the first year in Figure 5.5 and on the other hand we will represent in Figure 5.6 the data collected on Mondays throughout the 6 years.

5.2 Air Passenger Data results

In this section we will use the Air Passenger Dataset. We will study the performance of several simple methods versus the performance of the four ensemble methods studied (using the same hyperparameters in both).

First we will use the simple backtester and all of the error-accounting strategies listed in Section 2.3 to measure performance. Next, we will use fixed-window backtester. As individual method we choose the SARIMA, ARIMA, Prophet, Holt-Winter and Theta methods.

5.2.1 Individual methods

The hyperparameters were selected by hyperparameter tuning assuming that the seasonal period are 12. We chose the hyperparameters that reach the minimum mean absolute error in the test set. First by dividing the training data as follows: 80%-20% for simple backtester, and 75%-10%-15% being the 10% the gap between sets for fixed-window backtester.

SARIMA In this case, the best parameters are `{'p': 2, 'd': 1, 'q': 1}` for both, simple and fixed-window method. Appart of that we selected the trend both linear or constant and the seasonal order as `(P,Q,D,s)=(1,0,1,12)`.

ARIMA The optimal parameters in this situation are `{'p': 2, 'd': 1, 'q': 1}` for both too.

Prophet In this case we study the seasonal mode (multiplicative or additive), the number of change points and the initial range where we take the changepoints. We detected that the only hyperparameter that influences the mae is the seasonal mode. We chose the additive mode.

Holt-Winter After the study, we chose multiplicative mode in the trend, and additive mode for the seasonal part. The study also determinated that the trend should not be damped or smothed.

Theta For this model we do not need any hyperparameters, since we knew from the nature of the data set that the period was 12.

Once we have selected the parameters of the models, we calculate the mae, mape, mase, mse, rmse, and smape for each one, as we described in Section 2.3. We can see the results in Table 5.3.

5.2.2 Ensemple methods

As we mentioned, the hyperparameters selected are the same as the individual ones. In this case we use median, mean, weighted average and Bates & Granger method, as defined in Chapter 3. We obtain the performance achived by ensemble of the individuals methods, the result can be also see in Table 5.3.

5.2.3 Comparison

In addition, we have a couple of plots that will help us to see the results of Table 5.3 in a more visual way. On the one hand we can see in Figure 5.7 the results with the simple backtester. On the other hand the results with fixed-window backtester are shown in Figure 5.8.

For the simple backtester, the best performing single method is Holt-winter, followed by SARIMA. On the contrary, the worst method is ARIMA. In this case, only the first two improve the results obtained with the ensemble methods. We see a more stable performance with the ensemble methods, obtaining values closer to the best than to the worst performance.

Comparing the ensemble methods, the median is the best in this case, followed by the mean. This could be due to the number of parameters to estimate. We ensemble 5 simple models with their own parameters. In cases like this, the forecast combination puzzle makes sense. One reason why we get better results with the median than with the mean may be the performance of the ARIMA outliers.

Now, we will discuss the results with the fixed window backtester. In this case, we measure the predictive ability with lag. This justifies that more complex methods obtain better results in this case. As far as individual methods are concerned, SARIMA obtains the best results, and as far as ensemble methods are concerned, the weighted average wins. We must make an important point: the performance of the weighted average ensemble method beats the SARIMA method in terms of mse and rmse and

equals it in mape and smape, even with SARIMA being the best in the group of simple methods.

In summary, in general we see results (although not necessarily better), more stable with ensemble methods. If we have to ensemble many models, it is better to resort to simple ensemble methods such as mean or median. Otherwise, if we do not want to predict the immediate future, then it is better to use more complex ensemble models that compensate for the efficiencies and shortcomings of the simple methods.

5.3 Daily Visitors to the Website Data results

In this section we will only use the simple backtester. We start with three individual models: SARIMA, linear and quadratic. We study the results applying these methods and the ensemble methods on the three variables separately.

5.3.1 Individual methods

As in the previous experiment, we choose the hyperparameters that reach the lowest mean absolute error in the test set dividing the training data as follows: 80%-20%. We will use simple backtester for all variables.

SARIMA In this case, the best parameters are: `{'p': 2, 'd': 1, 'q': 2}` for *Page Loads* and `{'p': 1, 'd': 1, 'q': 1}` for both *Unique Visits* and *First Time Visits*. Appart of that we selected the trend both linear or constant and the seasonal order as `(P,Q,D,s)=(0,0,0,7)`.

Linear and quadratic In these models, all parameters are optional. Therefore, we do not adjust any hyperparameters.

5.3.2 Ensemple methods

As in section before, the hyperparameters selected for ensemble methods are the same as the individual ones. We use median, mean, weighted average and Bates & Granger methods. The results are shown in Table 5.4.

5.3.3 Comparison

We will now compare the effectiveness of the ensemble strategies with respect to basic ones. Table 5.4 contains the outcome. Appart from this table, we present plots to summarize the information for each variable. The results are shown in Figure 5.9, Figure 5.10 and Figure 5.11 for the variables *Page Loads, Unique Visits* and *First Time Visits* respectively.

In this experiment we use few and simple methods, without many parameters to estimate, so in this case it would be normal if the best ensemble method was one of the complex ones.

As for the variable *Page Loads*, the best single model is SARIMA, and the worst is quadratic. In this case, the Bates and Granger method obtains better results than the weighted average, in contradiction with the first experiment. However, the median beats both, obtaining the best results in the ensemble method group.

As we commented in the last experiment, the ensemble methods obtain similar results while the results of the individual methods can be more variable and unexpected. To show this aspect, in Table 5.5 we can see the variance between individual methods and between ensemble methods. In addition we can see the ratio $\frac{\text{variance individual models}}{\text{variance ensemble methods}}$. In this variable we can observe a performance variance of up to 1546 times higher.

Focusing on *Unique Visits* variable, the linear model obtains the best results among the individual models and the Bates and Granger method wins among the ensemble models. In this case, we obtain variances between 498 (for mape) and 2666 (for rmse) times greater among the individual models than among the ensemble models.

Finally, for the variable *First Time Visits* we can see a variance between 62 and 16621 times lower in the ensemble method group. In this case we have a tie between models: SARIMA and median being in both cases the best in their group.

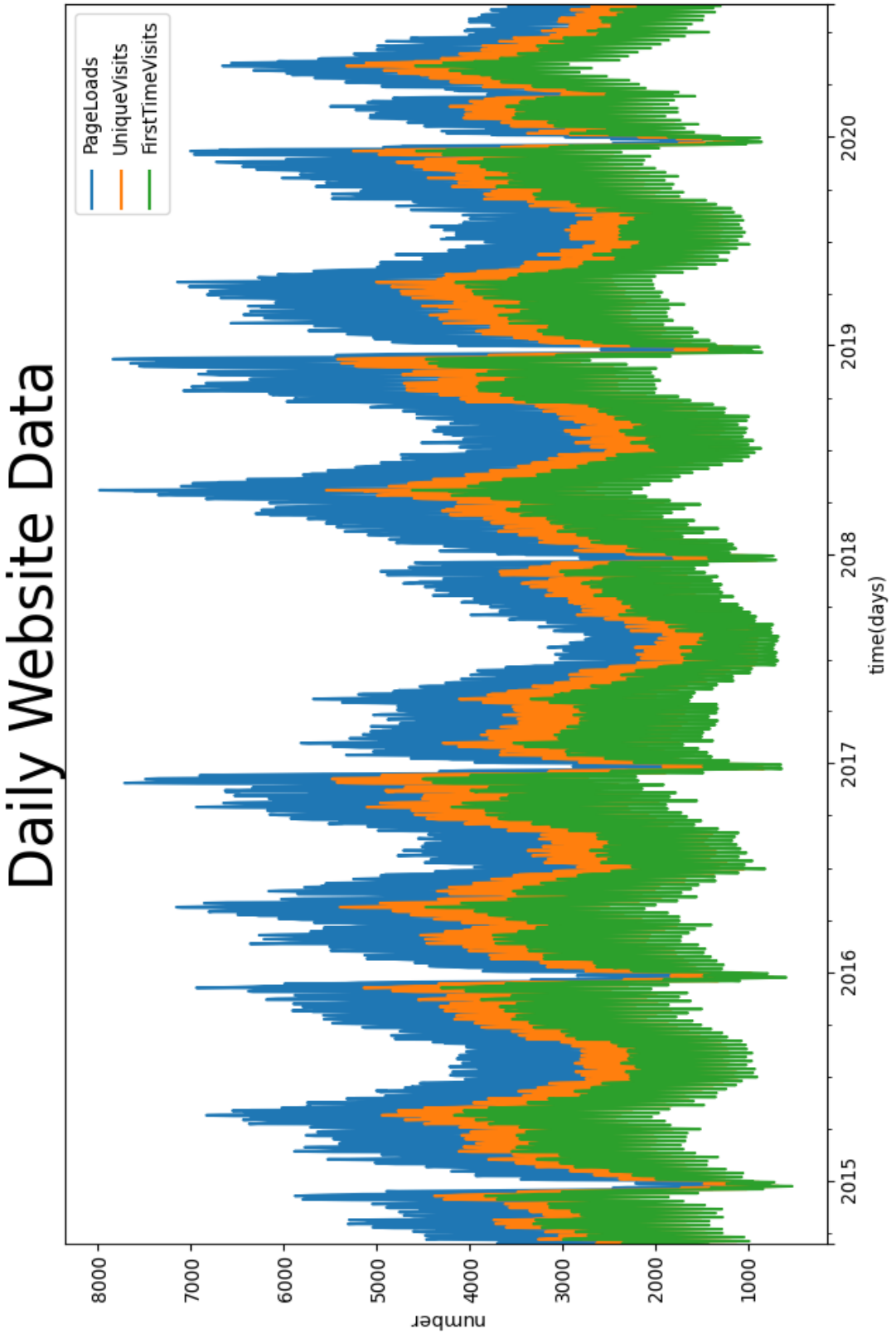

Figure 5.4: Daily Visitors to the Website Data

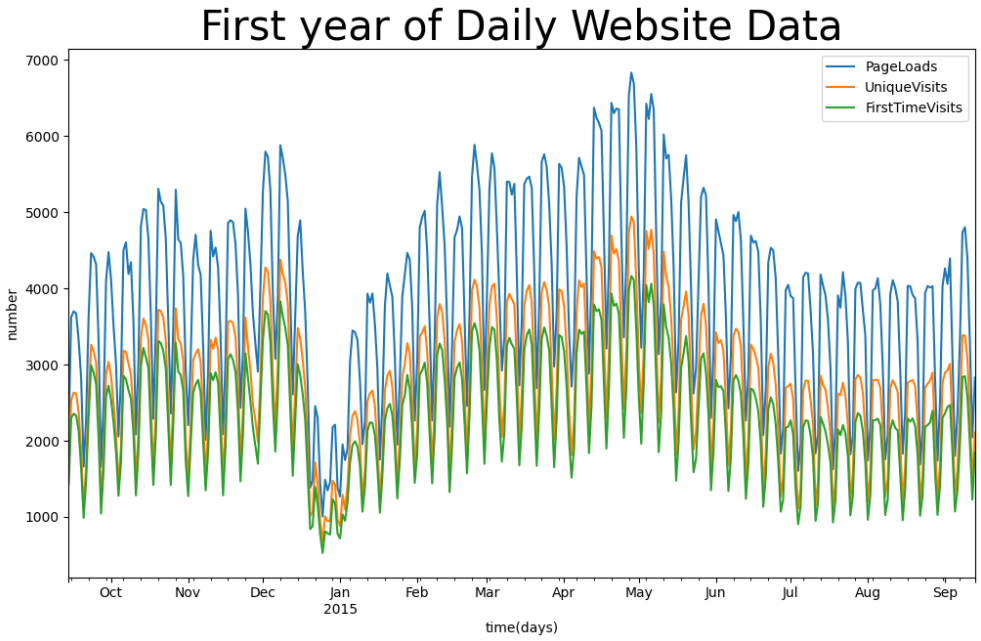

Figure 5.5: First year Daily Visitors to the Website Data

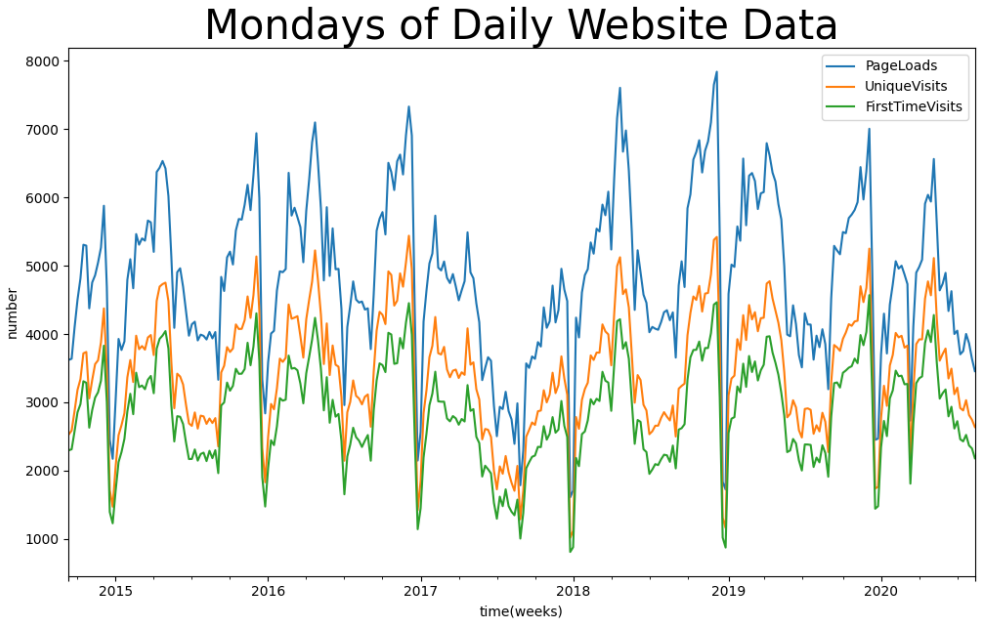

Figure 5.6: Mondays of Daily Visitors to the Website Data

Simple Backtester

	Individual methods					Ensemble methods			
	sarima	arima	prophet	holt	theta	median	mean	weighted average	bates & granger
mae	18.88	52.88	34.07	12.30	40.48	24.75	25.95	26.73	36.70
mape	0.04	0.11	0.08	0.03	0.09	0.05	0.06	0.06	0.08
mase	0.41	1.15	0.74	0.27	0.88	0.54	0.56	0.58	0.80
mse	499.24	4875.09	1739.03	295.08	2332.12	1093.55	1154.02	1256.26	1977.62
rmse	22.34	69.82	41.70	17.18	48.29	33.07	33.97	35.44	44.47
smape	0.04	0.12	0.08	0.03	0.09	0.05	0.06	0.06	0.08

Fixed-Window Backtester

mae	20.76	57.13	36.47	44.49	40.28	27.59	26.01	21.17	28.38
mape	0.04	0.11	0.08	0.10	0.08	0.06	0.05	0.04	0.07
mase	0.45	1.25	0.80	0.97	0.88	0.60	0.57	0.46	0.62
mse	819.57	6113.02	1853.15	2459.07	2187.66	1218.03	1163.34	728.64	1027.92
rmse	28.63	78.19	43.05	49.59	46.77	34.90	34.11	26.99	32.06
smape	0.04	0.12	0.08	0.10	0.09	0.06	0.05	0.04	0.06

Table 5.3: Air Passenger Data results

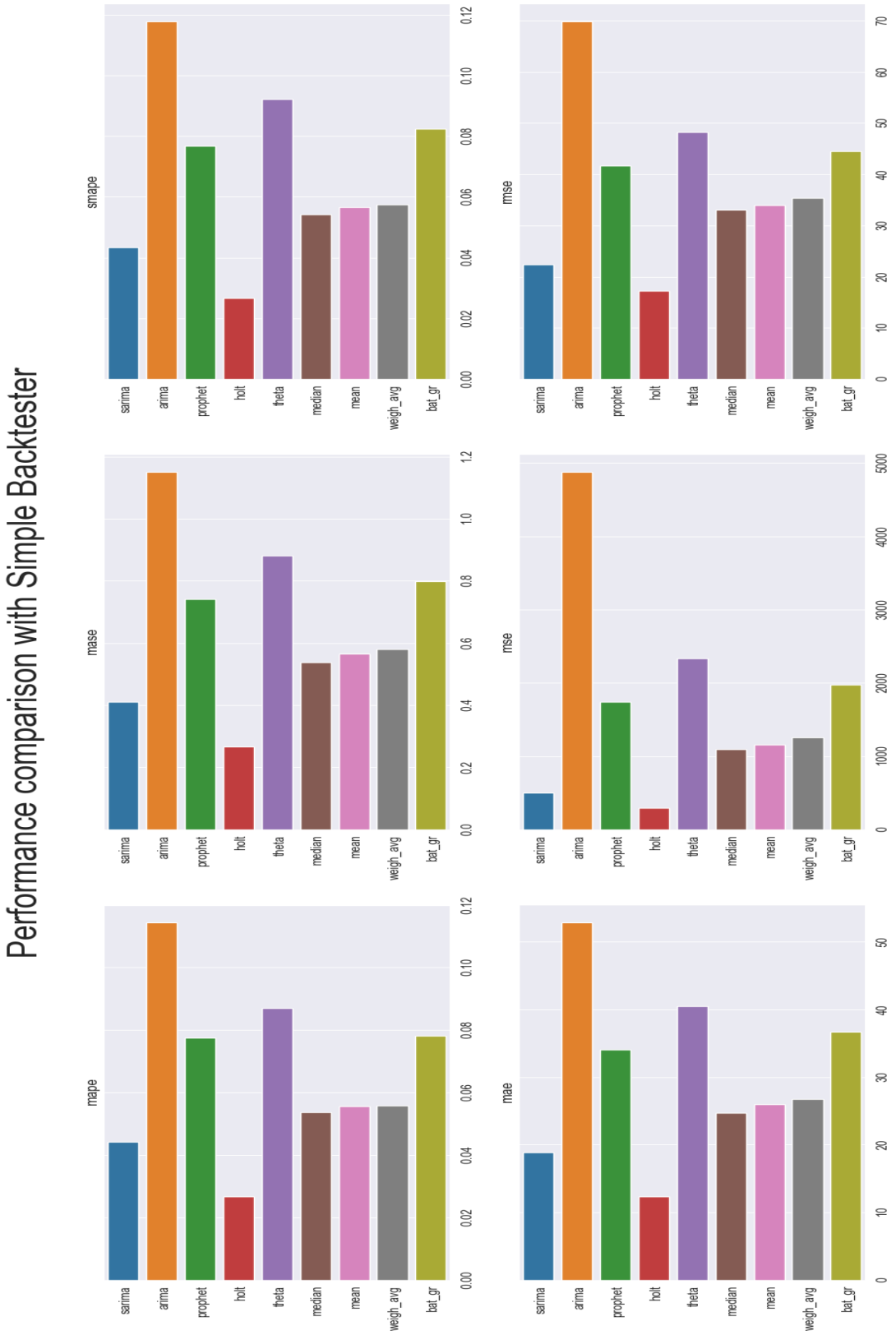

Figure 5.7: Models performance with simple backtester

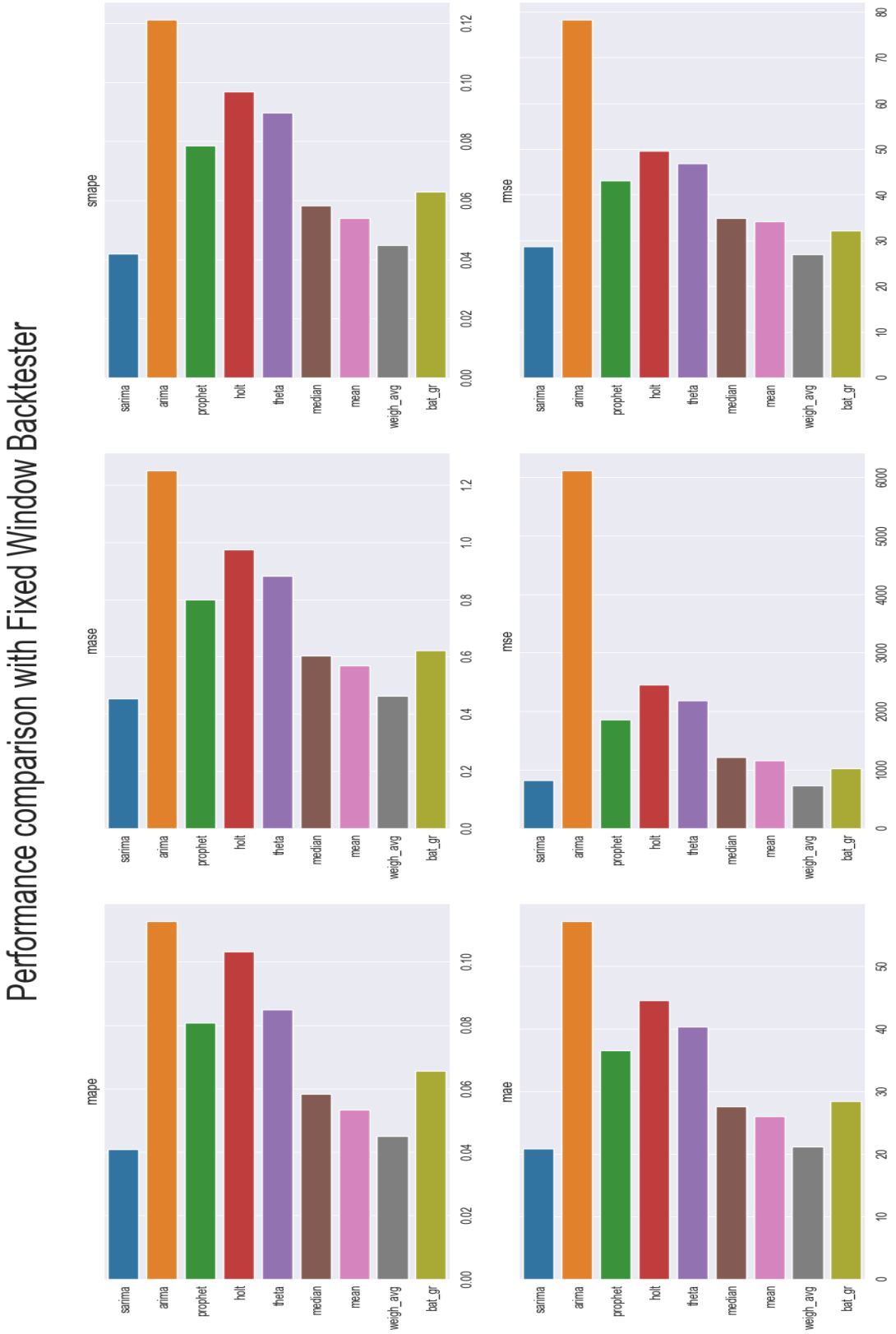

Figure 5.8: Models performance with fixed-window backtester

Page Loads							
	Individual methods			Ensemble methods			
	sarima	linear	quadratic	median	mean	weighted average	bates & granger
mae	958.02	1023.99	1307.68	1038.03	1051.42	1043.58	1040.67
mape	0.28	0.33	0.43	0.33	0.34	0.33	0.33
mase	1.46	1.56	1.99	1.58	1.60	1.59	1.58
mse	1381966.01	1569372.07	2468249.75	1604633.53	1648745.74	1624946.30	1615634.43
rmse	1175.57	1252.75	1571.07	1266.74	1284.03	1274.73	1271.08
smape	0.24	0.26	0.31	0.26	0.26	0.26	0.26
Unique Visits							
mae	751.47	736.84	781.25	751.47	753.31	753.51	748.53
mape	0.31	0.30	0.34	0.31	0.32	0.32	0.31
mase	1.54	1.51	1.60	1.54	1.54	1.54	1.53
mse	831969.76	819661.26	891076.12	831969.76	834900.42	835250.07	828059.15
rmse	912.12	905.35	943.97	912.12	913.73	913.92	909.98
smape	0.26	0.25	0.27	0.26	0.26	0.26	0.26
First Time Visits							
mae	637.13	641.09	684.48	637.13	643.27	645.36	644.35
mape	0.30	0.29	0.35	0.30	0.31	0.31	0.31
mase	1.52	1.53	1.63	1.52	1.54	1.54	1.54
mse	613696.14	635067.35	673915.13	613696.14	611460.96	612648.98	612268.70
rmse	783.39	796.91	820.92	783.39	781.96	782.72	782.48
smape	0.26	0.26	0.27	0.26	0.26	0.26	0.26

Table 5.4: Daily Visitors to the Website Data results

	Page Loads			Unique Visits			First Time Visits		
	Var individual	Var ensemble	Var.ind/Var.ens	Var individual	Var ensemble	Var.ind/Var.ens	Var individual	Var ensemble	Var.ind/Var.ens
mae	5.190956e+04	3.357020e+01	1546	6.902323e+03	5.330800e+00	1295	3.794260e+03	13.6793	277
mape	9.200000e-03	0.000000e+00	1320	2.200000e-03	0.000000e+00	498	2.200000e-03	0.0000	62
mase	1.199000e-01	1.000000e-04	1546	2.880000e-02	0.000000e+00	1295	2.160000e-02	0.0001	277
mse	4.244755e+11	3.524165e+08	1204	2.462797e+10	1.110637e+07	2217	1.244595e+10	861550.1279	14446
rmse	6.496792e+04	5.413020e+01	1200	8.901198e+03	3.338500e+00	2666	5.843978e+03	0.3516	16621
smape	2.400000e-03	0.000000e+00	2078	8.000000e-04	0.000000e+00	1746	5.000000e-04	0.0000	324

Table 5.5: Daily Visitors to the Website Data performance variance results

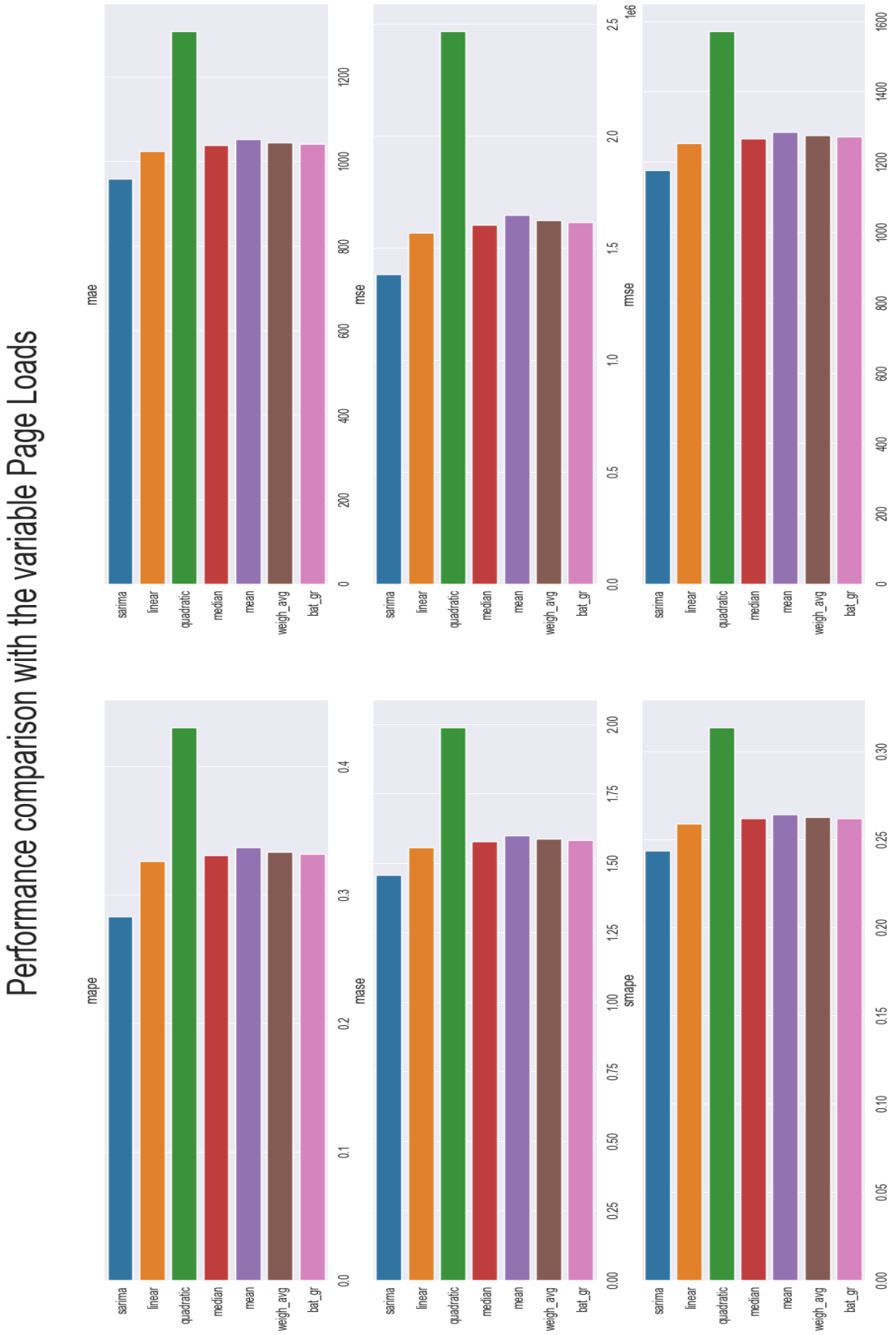

Figure 5.9: Performance comparison with the variable Page Loads

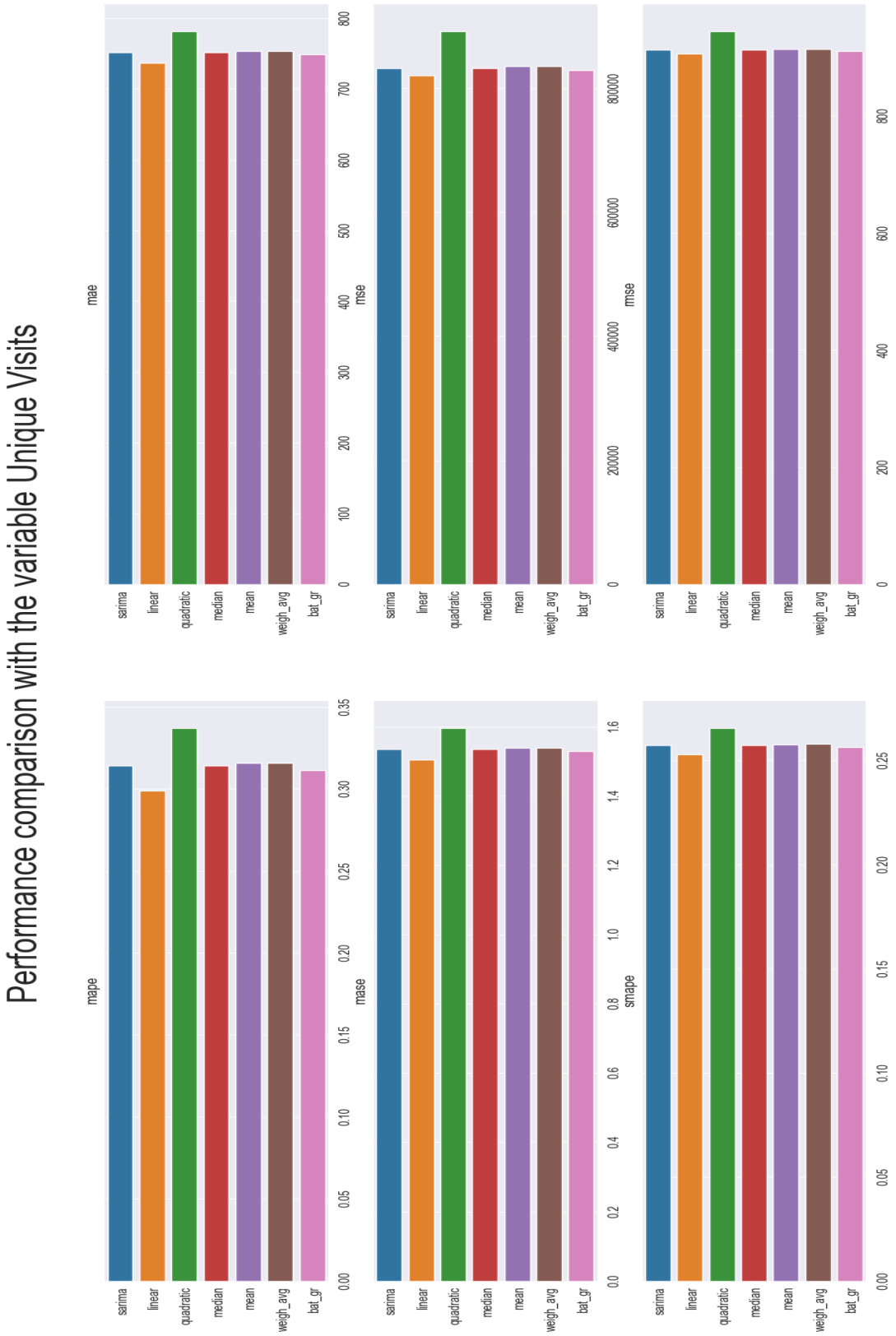

Figure 5.10: Performance comparison with the variable Unique Visits

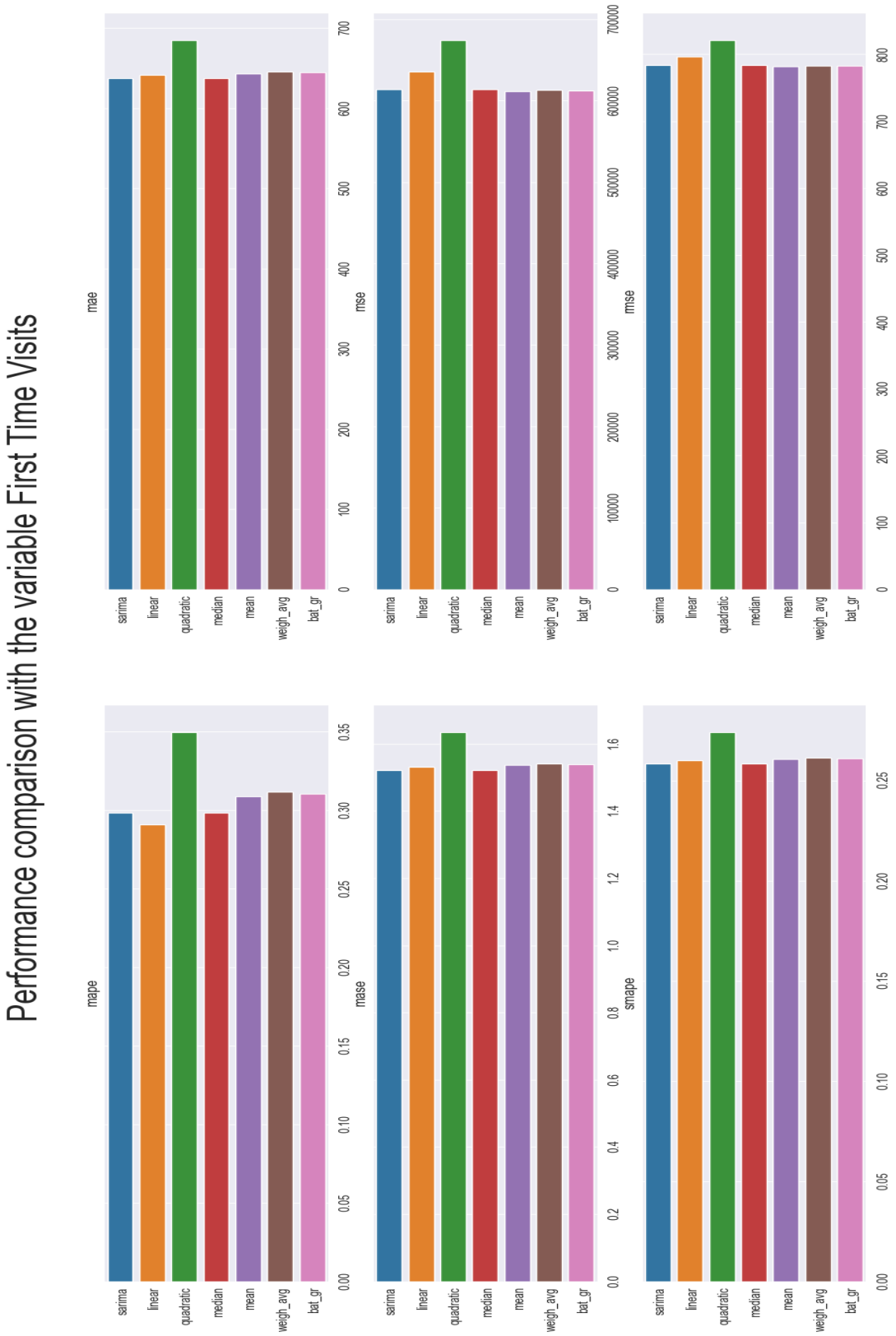

Figure 5.11: Performance comparison with the variable First Time Visits

Chapter 6

Conclusions

As we saw at the beginning of this research, forecasting has great importance today for making decisions in different fields. The success of these decisions depends on the accuracy of these predictions.

We have many models that give us different performances in each case: a small noise or modification in a time series can mean a significant increase or decrease in the performance of a model prediction. For this reason, ensemble methods have been studied. They take information from all model bases and use it to create a more robust and stable prediction. In our experiments we could observe a big difference between the variance obtained by simple models and the one obtained by ensemble methods, the last one being considerably lower.

We have also seen which method to use in each case and which is the most appropriate way to evaluate the performance depending on the main objective. If we have complex base models, it is usually better to use simple methods such as mean or median. Even if we have outliers in the prediction model we can use Winsorized and Trimmed mean. This is because with other methods we can suffer overestimates due to the huge number of parameters; also due to the fact that by making so many estimates the errors are added, concluding with predictions that are not very reliable.

However in some cases when we have few and simple base models it is more convenient to use the weighted mean. It could be with an evaluation metric to minimize, with or without the constraint of adding the unit, with or without an intercept, or even using the method proposed by Bates & Granger. We have also seen different techniques to evaluate performance once the evaluation metric has been chosen : if we want to predict short-term future values it is better to use moving windows without a gap, but if on the contrary we want to measure performance in predicting long-term

values we should use fixed windows with a gap.

Finally, we will mention some potential findings for future investigations: focus on probabilistic forecasting, study non linear combination of the point forecasts, study the extension of these techniques to multivariate time series and complete the open source contribution of the kats library to the above topics.

Bibliography

[Wan+22] Xiaoqian Wang et al. "Forecast combinations: An over 50-year review". In: *International Journal of Forecasting* (Dec. 2022). ISSN: 0169-2070. DOI: 10.1016/J.IJFORECAST.2022.11.005.

[For] *The COVID-19 Forecast Hub*. URL: https://covid19forecasthub.org/ (visited on 05/09/2023).

[Rob] Siobhan Roberts. *All together now: the most trustworthy covid-19 model is an ensemble*. URL: https://www.technologyreview.com/2021/05/28/1025478/covid-ensemble-model-forecast-trustworthy/ (visited on 05/09/2023).

[BD16] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Third Edition. Springer International Publishing, 2016. ISBN: 978-3-319-29852-8. DOI: 10.1007/978-3-319-29854-2.

[Box08] George E. P. Box. *Time series analysis forecasting and control*. Ed. by Gwilym M. Jenkins and Gregory C. Reinsel. 4th ed. John Wiley, 2008. ISBN: 1-118-61919-6.

[Kaga] *Air Passenger Data for Time Series Analysis*. URL: https://www.kaggle.com/datasets/ashfakyeafi/air-passenger-data-for-time-series-analysi (visited on 02/23/2023).

[Kagb] *Daily website visitors (time series regression)*. URL: https://www.kaggle.com/datasets/bobnau/daily-website-visitors (visited on 04/02/2023).

[Faca] Facebook. *kats One stop shop for time series analysis in Python*. URL: https://facebookresearch.github.io/Kats/ (visited on 02/23/2023).

[HA21] Rob J Hyndman and George Athanasopoulos. *Forecasting*. 3rd ed. Australia: OTexts, May 2021.

[DG] Anais Dotis-Georgiou. *Forecasting with FB Prophet and InfluxDB*. URL: https://w2.influxdata.com/blog/forecasting-with-fb-prophet-and-influxdb/ (visited on 02/25/2023).

[Facb] Facebook. *Prophet*. URL: https://github.com/critobfer/prophet/blob/main/notebooks/trend_changepoints.ipynb (visited on 03/06/2023).

[Lstb] *Understanding LSTM Networks*. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 04/19/2023).

[Lsta] *LSTM with PyTorch*. URL: https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html (visited on 02/26/2023).

[HB03] Rob J. Hyndman and Baki Billah. "Unmasking the Theta method". In: *International Journal of Forecasting* 19 (2 Apr. 2003), pp. 287–290. ISSN: 0169-2070. DOI: 10.1016/S0169-2070(01)00143-1.

[WM98] Kenneth D West and Michael W McCracken. "Regression-Based Tests of Predictive Ability". In: *International Economic Review* 39 (4 1998), pp. 817–840. ISSN: 00206598, 14682354. DOI: 10.2307/2527340. URL: http://www.jstor.org/stable/2527340.

[Yon13] Ed Yong. *The Real Wisdom of the Crowds*. 2013. URL: Yong,E.(2021)Therealwisdomofthecrowds,Science.NationalGeographic.Availableat:https://www.nationalgeographic.com/science/article/the-real-wisdom-of-the-crowds(Accessed:March6,2023).Yong,E.(2021)Therealwisdomofthecrowds,Science.NationalGeographic.Availableat:https://www.nationalgeographic.com/science/article/the-real-wisdom-of-the-crowds(Accessed:March6,2023)..

[BG69] J M Bates and C W J Granger. "The Combination of Forecasts". In: *Journal of the Operational Research Society* 20 (4 1969), pp. 451–468. ISSN: 1476-9360. DOI: 10.1057/jors.1969.103. URL: https://doi.org/10.1057/jors.1969.103.

[NG74] P Newbold and C W J Granger. "Experience with Forecasting Univariate Time Series and the Combination of Forecasts". In: *Journal of the Royal Statistical Society. Series A (General)* 137 (2 1974), pp. 131–165. ISSN: 00359238. DOI: 10.2307/2344546. URL: http://www.jstor.org/stable/2344546.

[JW08] Victor Richmond R Jose and Robert L Winkler. "Simple robust averages of forecasts: Some empirical results". In: *International Journal of Forecasting* 24 (1 2008), pp. 163–169. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2007.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207007000878.

[GR84] CLIVE W J GRANGER and Ramu Ramanathan. "Improved Methods of Combining Forecasts: ABSTRACT". In: *Journal of Forecasting (pre-1986)* 3 (2 1984). Copyright - Copyright Wiley Periodicals Inc. Apr-Jun 1984 Última actualización - 2011-10-07 CODEN - JOFODV, p. 197. ISSN: 02776693. URL: https://www.proquest.com/scholarly-journals/improved-methods-combining-forecasts/docview/224795448/se-2?accountid=14744https://cbua-us.primo.exlibrisgroup.com/discovery/openurl?institution=34CBUA_US&vid=34CBUA_US:VU1&lang=es?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=unknown&sid=ProQ:ProQ%3Apq1busgeneral&atitle=Improved+Methods+of+Combining+Forecasts%3A+ABSTRACT&title=Journal+of+Forecasting+%28pre-1986%29&issn=02776693&date=1984-04-01&volume=3&issue=2&spage=197&au=CLIVE+WJ+GRANGER%3BRamanathan%2C+Ramu&isbn=&jtitle=Journal+of+Forecasting+%28pre-1986%29&btitle=&rft_id=info:eric/&rft_id=info:doi/.

[SW04] James H Stock and Mark W Watson. "Combination forecasts of output growth in a seven-country data set". In: *Journal of Forecasting* 23 (6 2004), pp. 405–430. DOI: https://doi.org/10.1002/for.928. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.928.

[Tim06] Allan Timmermann. "Chapter 4 Forecast Combinations". In: *Handbook of Economic Forecasting* 1 (Jan. 2006), pp. 135–196. ISSN: 1574-0706. DOI: 10.1016/S1574-0706(05)01004-9.

[HW14] Cheng Hsiao and Shui Ki Wan. "Is there an optimal forecast combination?" In: *Journal of Econometrics* 178 (PART 2 Jan. 2014), pp. 294–309. ISSN: 0304-4076. DOI: 10.1016/J.JECONOM.2013.11.003.

[Dow] URL: https://pypi.org/project/kats/.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

[Rdo] *Bates/Granger (1969) Forecast Combination Approach*. URL: https://search.r-project.org/CRAN/refmans/GeomComb/html/comb_BG.html (visited on 04/04/2023).

[Cri] *critobfer/Kats*. URL: https://github.com/critobfer/Kats.git.

[Dat] *Air Traffic Passenger Data*. URL: https://data.world/data-society/air-traffic-passenger-data (visited on 03/30/2023).