Universidad del País Vasco Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA
Konputazio Ingeniaritza eta Sistema Adimentsuak Unibertsitate Masterra
Máster Universitario en Ingeniería Computacional y Sistemas Inteligentes

INGENIERÍA COMPUTACIONAL Y SISTEMAS INTELIGENTES

Introduction to Time Series Data Analisis

Time Series Prediction

Estudiante: Eneko Perez

Estudiante: Alan García

Curso: 2024-2025

Fecha: December 31, 2024

Description of the time series

We will use hierarchical sales data from Walmart, the world's largest company by revenue, to forecast daily sales for the next 28 days. The data covers stores in three US states (California, Texas and Wisconsin) and includes item level, department, product categories, and store details. In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust data set can be used to improve forecast accuracy [1].

The dataset is divided into four different files: **calendar.csv**, which contains information about the dates on which the products are sold; **sell-prices.csv**, which provides details on the price of products sold per store and date; **sales-train-validation.csv**, which includes historical daily unit sales data per product and store for 1913 days; and **sales-train-evaluation.csv**, which contains historical daily unit sales data per product and store for 1941 days.

With this data, various time series could be generated, such as the sales of a specific product in a store over time, the total sales of a store over time, and so on. However, the time series we will use and predict will be the total sum of sales across all stores over time. The data splits used to train the algorithms and perform the predictions are represented in Figure 1, where a prediction length of 28 days is used. Only the training and evaluation data has been used to adjust the model hyperparameters.

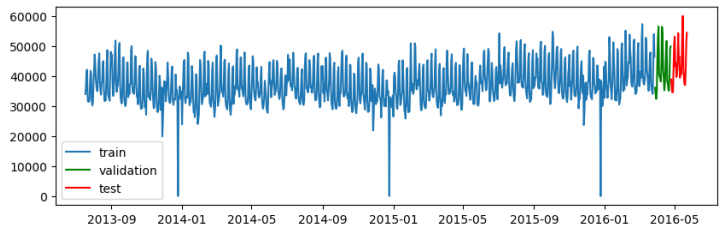

Figure 1: Data splits used

For a more extensive analysis of the data, refer to [2], where a detailed exploration of trends and patterns is presented.

Description of the chosen model

Three different models have been used to predict the time series.

XGBoost

XGBoost (eXtreme Gradient Boosting) is a machine learning algorithm based on gradient boosting that uses decision trees as base models. To use this algorithm, a transfor-

mation of the data has been made, extracting features such as the day of the week, the month, the year, and others.

It works by sequentially building decision trees, where each tree is trained to correct the residual errors of the previous model. The final prediction is a weighted sum of all the trees. It supports regularization techniques (L1 and L2), making it robust against overfitting.

The algorithm is particularly effective for large datasets and complex problems, and its flexibility allows customization through hyperparameters, such as the number of trees, the learning rate, and tree depth [3].

After several training iterations, the best-performing model was achieved with 270 estimator trees, an initial learning rate of 0.1, and a maximum tree depth of 5, resulting in RMSE values of 1866.36 for the validation set and 3288.07 for the test set.

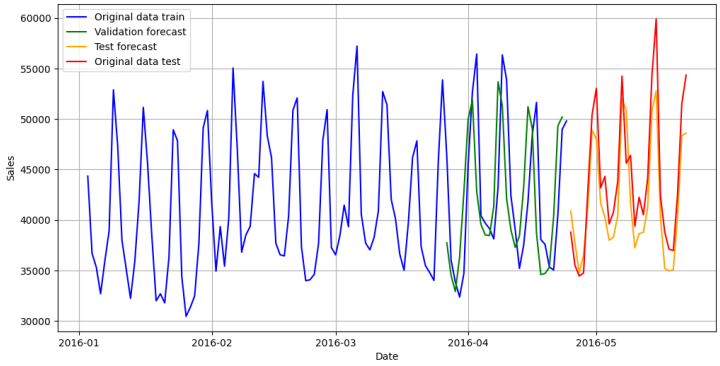

Figure 2: Forecast with XGBoost

ARIMA

ARIMA (AutoRegressive Integrated Moving Average) is a statistical model commonly used to analyze and forecast time-series data. It combines three components:

- **AR (AutoRegressive):** Uses the dependency between an observation and a number of lagged observations (past values).
- **I (Integrated):** Involves differencing the series to make it stationary, removing trends and seasonality.
- **MA (Moving Average):** Models the dependency between an observation and residual errors from a moving average model applied to lagged observations.

These models are suitable for univariate time series data and require careful tuning of parameters p (AR order), d (degree of differencing) and q (MA order). Once fitted, the model can provide insight into the dynamics of the series and forecast future values with confidence intervals [4].

After several experiments, the final trained model achieved a test RMSE of 3441.47, as illustrated in the following forecasting figure:

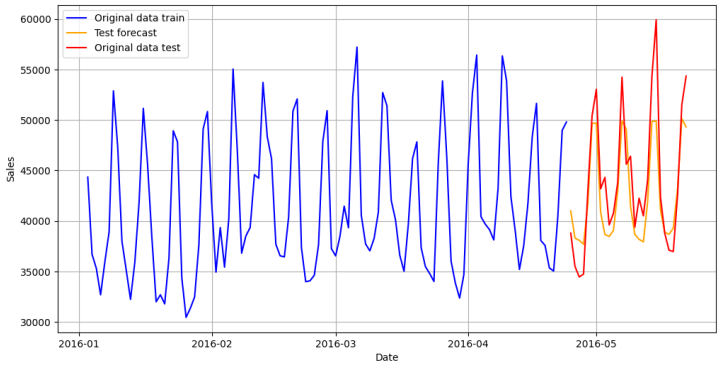

Figure 3: Forecast with ARIMA

DeepAR

DeepAR is an LSTM-based autoregressive model that generates probabilistic forecasts for future time steps by conditioning on past values of time series [5]. The model learns from the sequential patterns and dependencies in the time series data, and it outputs a probability distribution over possible future values, enabling the model to provide not only point forecasts but also confidence intervals.

DeepAR is particularly effective when dealing with large-scale datasets that consist of many similar time series, as it can model the common patterns shared across these series. However, in this case the model has been trained with just one time series, and although DeepAR accepts features such as holidays or other categorical features, only sales data and dates have been used.

The predictions of the trained model for the validation and test sets are presented in the following figure, achieving RMSE values of 1549.26 and 2887.18, respectively:

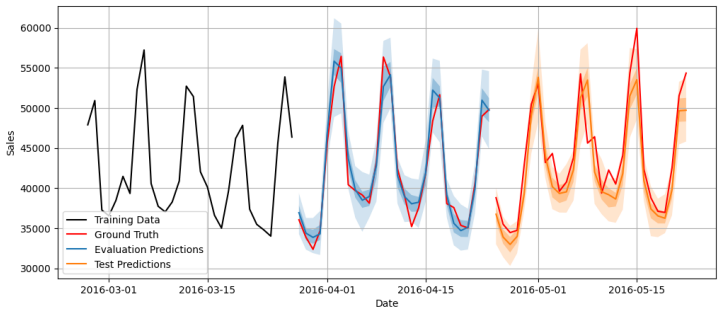

Figure 4: Forecast with DeepAR

Results and comparison

Based on the RMSE error, DeepAR is the model that best fits the actual distribution, followed by XGBoost, with ARIMA being the least accurate. Furthermore, the following figure compares the predictions of each model on the test set, supporting the hypothesis that the ARIMA model does not fit the data as well as the other two models.

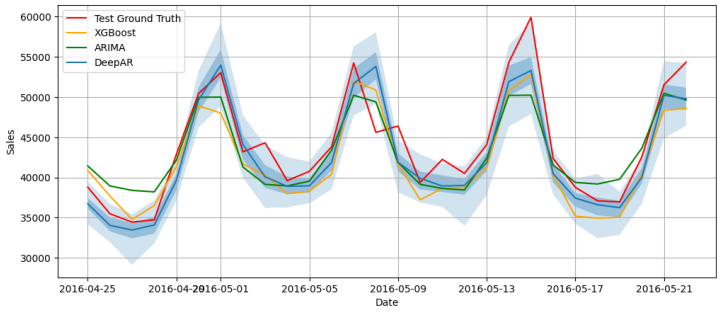

Figure 5: Comparison

Conclusions

Given the simplicity of the use case, which involves forecasting a single univariate time series, the difference in performance between traditional methods like ARIMA or XGBoost and more complex methods like DeepAR is not very significant. Therefore, in such cases, it is preferable to choose simpler methods, as they provide acceptable results without requiring substantial computational resources. Specifically, we consider XGBoost to be an excellent choice.

Additionally, during this project, efforts were made to explore cross-validation techniques for training the models and to apply Transformers as deep learning techniques for the forecasting task. However, despite being very promising methods that could potentially yield better results than those presented, significant progress in these areas was not achieved due to various technical complications, leaving them as future work.

References

[1] Addison Howard, inversion, Spyros Makridakis, and vangelis. M5 forecasting - accuracy. https://kaggle.com/competitions/m5-forecasting-accuracy, 2020. Kaggle.

[2] Headsortails. Back to (predict) the future - Interactive M5 EDA. https://www.kaggle.com/code/headsortails/back-to-predict-the-future-interactive-m5-eda, 2020. Kaggle.

[3] Javier Jesús Espinosa-Zúñiga. Aplicación de algoritmos random forest y xgboost en una base de solicitudes de tarjetas de crédito. *Ingeniería, investigación y tecnología*, 21(3), 2020.

[4] Ma Pilar González Casimiro. Análisis de series temporales: Modelos arima. *Universidad del País Basco*, 1(1):1–169, 2009.

[5] David Salinas Valentin Flunkert and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *CoRR*, abs/1704.04110, 2017.