A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

Transfer learning using deep learning for forecasting retail sales

Sebastião Santos Lessa, DCC-FCUP



Project development

Objective: Use the [Dataset](#) to understand the potential of creating hybrid models with transfer learning.

How?

Models used:

- TiDE and TSMixer (from scratch)
- Chronos (pre-trained)



1. Generating initial forecasts with the Chronos model and residuals creation
2. Using the residuals from Chronos as training input for the TSMixer and TiDE models
3. Attempting to use the Chronos forecasts as dynamic covariates in the TSMixer and TiDE models to obtain better forecasts
4. Comparing the models on 4 windows of time using the Mean Absolute Percentage Error (MAPE) metric



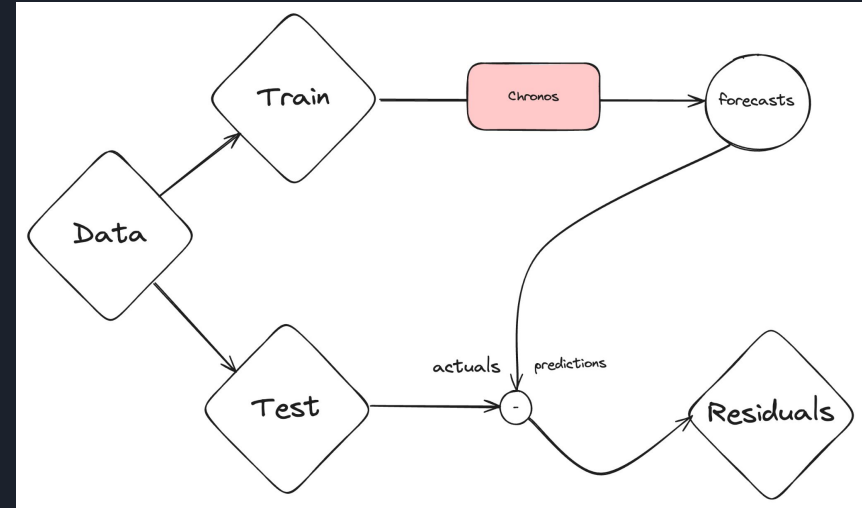
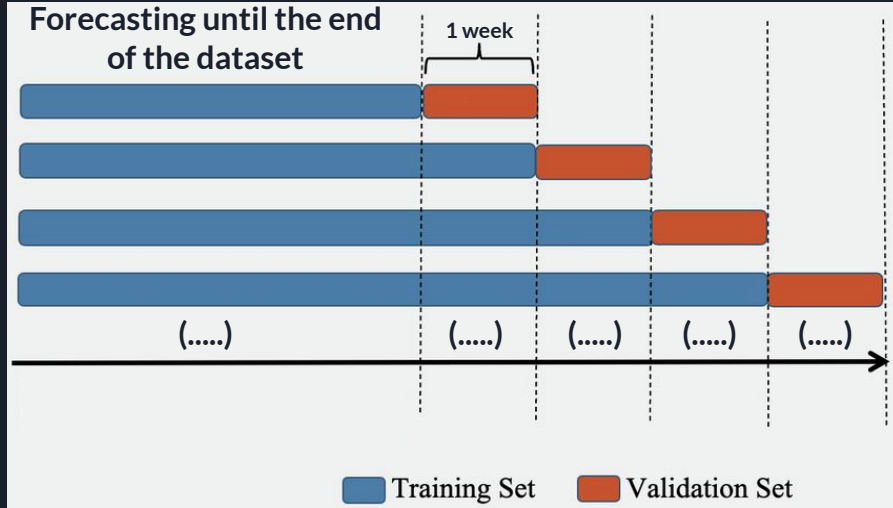
Dataset Description

The dataset consists of:

Identification values	Store and Department IDs and the Dates column
Target	<i>Weekly Sales</i>
Static covariates	Features that don't change (e.g.: <i>Type, Size</i>)
Dynamic covariates	Features that can change (e.g.: <i>IsHoliday, Temperature, Fuel_Price, CPI, Unemployment</i>)

Creating the Residuals dataset

Using Chronos, with context from the beginning of the dataset up to the starting point for predicting the following week. The prediction for that week is made, and then this process is repeated to make the prediction for the next week. As can be seen in the image below:

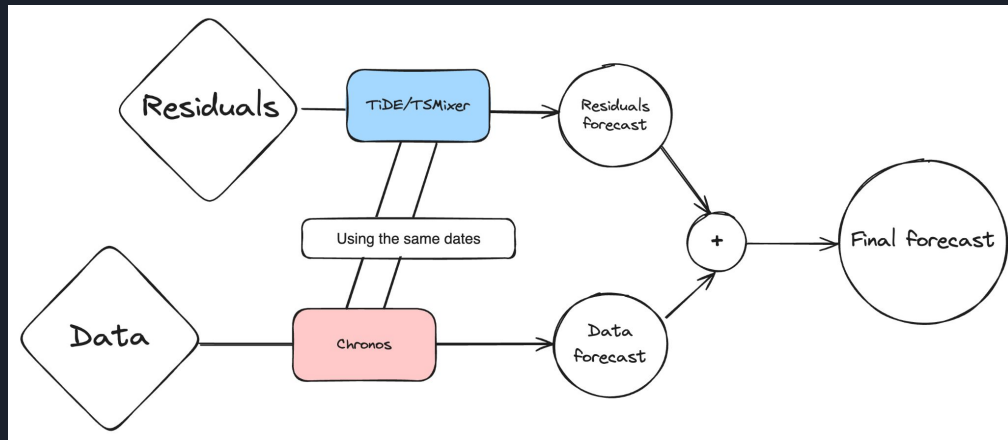


Training with the Residuals

Using the residuals dataset, I trained the TiDE and TSMixer models to try to predict the future residuals of future Chronos forecasts.

By combining the two, we theoretically have a better prediction from Chronos.

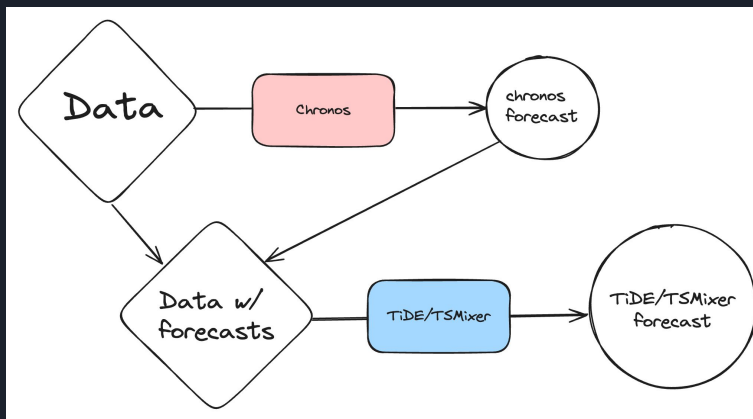
These TiDE and TSMixer models underwent hyper parameter tuning. We can assess the difference in the end.



Using the forecast directly as Dynamic covariates

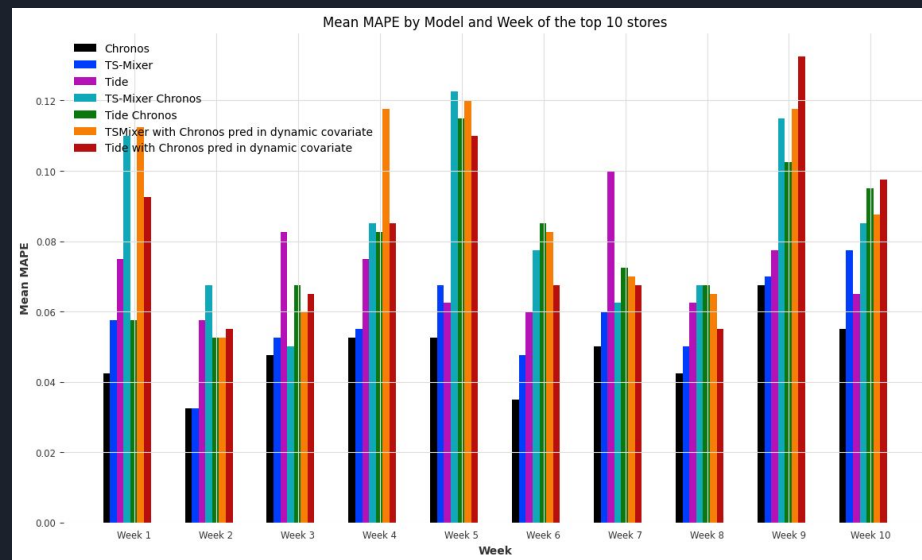
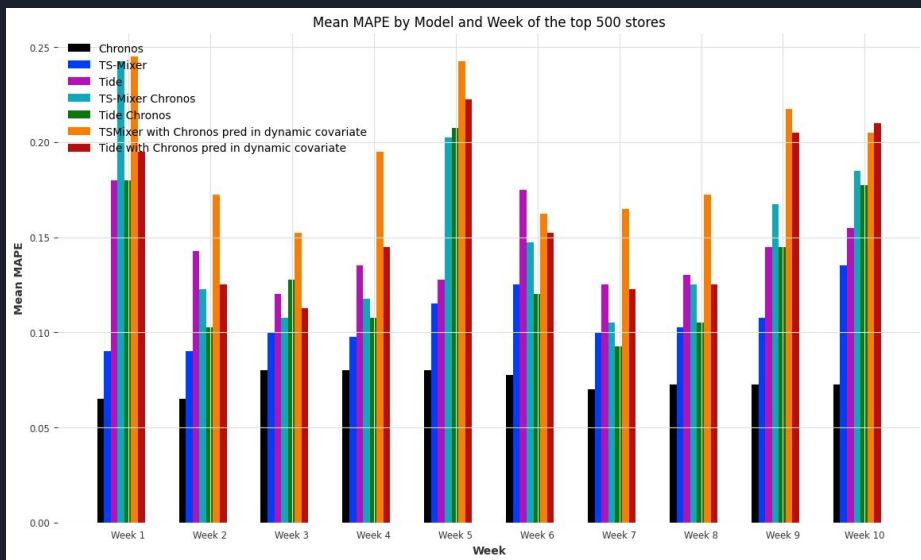
I added the chronos forecasts as a feature to the dataframe and trained the TiDE and TSMixer models, with this extra feature as a dynamic covariate (the chronos forecasts themselves).

Then, taking the implementation of TiDE and TSMixer, I added only the column of Chronos forecasts and will carry out predictions for these new models.



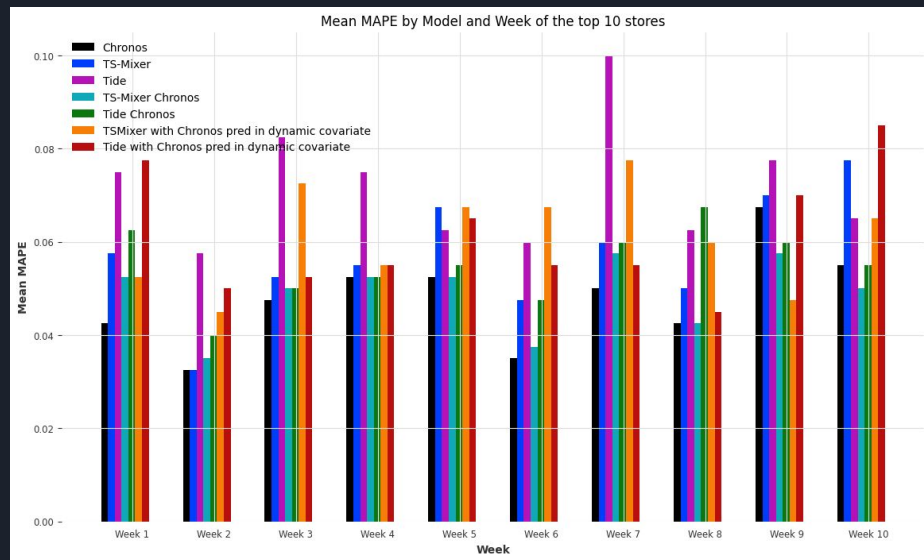
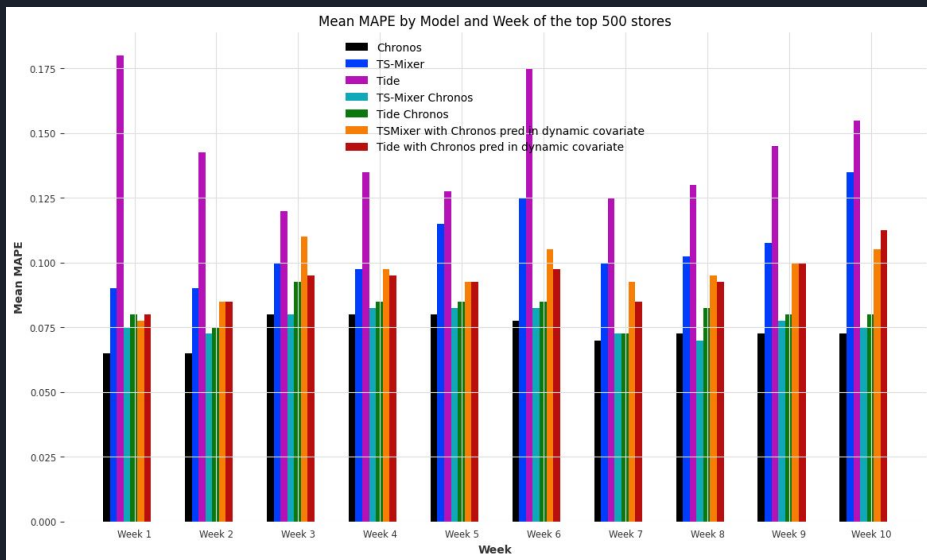
Outcomes

Using Mean absolute percentage error to compare each model, here's the results:



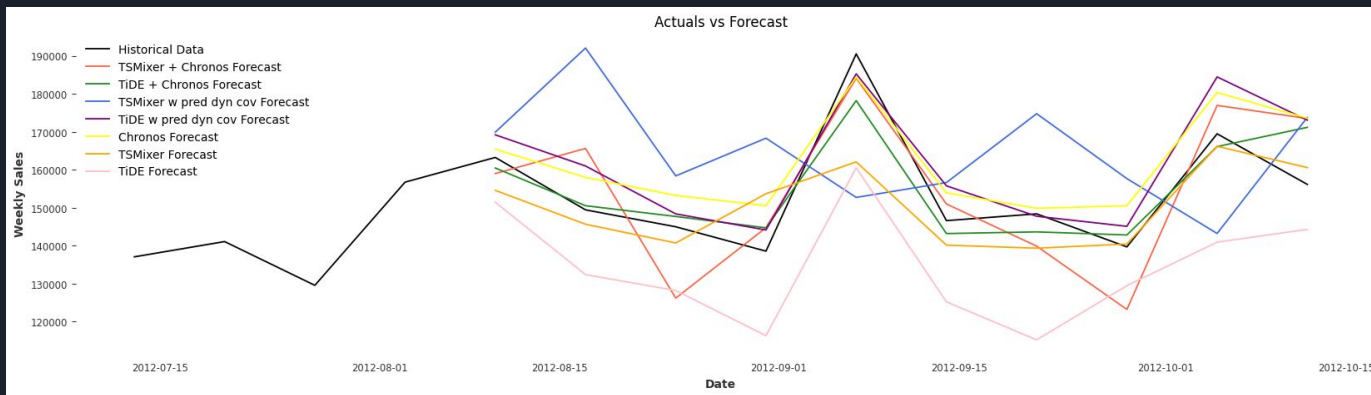
Hyper parameter tuning

For all the hybrid models a search for the best hyperparameters was conducted. Ultimately, we achieved a significantly better result.



Conclusions

- The best model is always the Chronos
- The TiDE model, trained with the residuals from Chronos and combined with Chronos, is the model with the most potential to, with more fine-tuning, become better than Chronos itself
- Adding the forecast to the dynamic covariates initially had a negative impact, but with further fine-tuning, it could also show potential



Graph with predictions from all models



Thank you!