DataStorm 3.0
Storming Round 1 Report

Team Crypto
University of Moratuwa

March 29, 2022

Team member details

Team members 1: **Manjitha Kularatne**
Team members 2: **Nuwan Bandara**
Team members 3: **Dasun Premathilake**

Other details

Github repository :
Lowest Total MAPE score : 77.655

1 Introduction

Many businesses and industries are in a deliberate requirement for accurate forecasting since of the uncertain nature of their business environments due to various factors such as the innovation-based transformations on different verticals. These challenges are critical and decisive in several industries, including retail businesses, where stakeholders are intuitively obligated to make key decisions in a short amount of time. These circumstances lead the business forms to turn towards advanced analytics for better outcomes through data-driven decision making.

Therefore, forecasting sales is one of the most fundamental problems most business chains have which directly result in improving project revenues, adequate preparation for the necessary supply, reduction of wastage, and better manage storage warehousing if the forecast is accurately positioned. In the given challenge, it is required to derive insights from the data and better estimate the sales four weeks for a retail chain which has previously used traditional forecasting methods to estimate projected sales for each item across stores and found those approaches to be inaccurate.

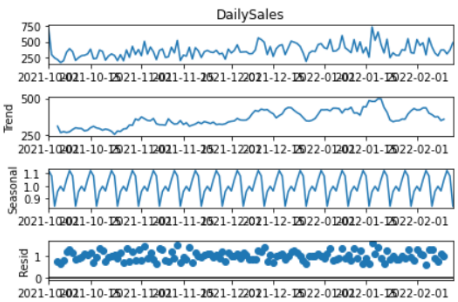
In the state-of-art machine learning (ML) approaches for time series forecasting, global forecasting models (GFM) are found to outperform the traditional univariate models which work on the isolated series. However, one key problem of GFMs is that they are not being localized enough to a particular series even though they share the same set of parameters across all local time series. Therefore, in this challenge, we explored the possibility of using a ensembling method of univariate models to obtain a performance similar to that of GFMs in time series forecasting. Our approach incorporates four base forecasting models: long short-time memory (LSTM) model, transformer architecture-based model, Prophet statistical model (Prophet) and seasonal autoregressive integrated moving average (SARIMA) model, which were then combined through linear stacking ensembling methodology. In the evaluation of the challenge, it is found that our ensembling approach outperforms the localized univariate models and GFMs.

2 Methods

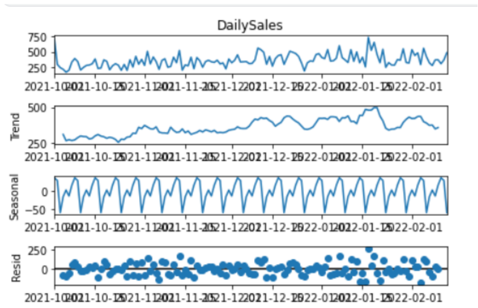
2.1 Pre-processing and feature engineering

As the first stage of the pre-processing data, we have analyzed the seasonal patterns and the frequency changes of the time-series dataset if any. Through our visual observations and analysis, we realized that there exists a slight biweekly seasonality in a group of time series data in certain items while most of the other patterns were seemingly random. Then we experimentally changed the daily formatted data of the training dataset to the weekly basis with the help of **datetime** python in-build library. Then sorted according to item-Code and the week basis data while adding a unique ID to the help of the model train. Howsoever, the abrupt reduction of data from daily to weekly basis resulted in a considerate insufficiency in data, manipulating the models to perform weekly in a generic circumstances. Therefore, we have decided to use the daily basis for all the training, validation, and testing phases of all models.

The data frames have to post-processed as the 4 models run and predicted data receives, it has to arrange according to the submission format by sorting and filtering.



(a) Seasonal decomposition of category 1 using multiplicative model



(b) Seasonal decomposition of category 1 using additive model

2.2 Localized univariate models

2.2.1 Transformer-based model

Transformers have been utilized for time series forecasting tasks in recent years because of their strong characteristic of self-attention in which they retain direct connections to all previous timestamps, thus allowing the information to flow in long sequences. In our approach, we utilized both the single-step transformer model and the multi-step transformer model to validate the best performance with a minimal number of model depth. Both the architectures consist of a positional enocoder, transformer encoder and a decoder layer while the single-step output window size is one whereas the multi-step output window size is seven. The optimizer for the models was Adam and the utilized loss function was the mean absolute error. Through our experiments in both models, we found that since the input length of the model through the daily sales data is not sufficient enough for a deeper architecture, the single-step transformer performed better than its counterpart. Further, we found that these generic transformers performed poorly in decomposition and thus, looses the sufficient accuracy in the evaluation.

2.2.2 LSTM + Conv1D

In this approach, 1D convolution was first applied to the sales vector to extract the features. 60 such filters were used, and the extracted features were then fed to two conjoined LSTM layers. Here, LSTM layers were used instead of simple RNN layers to avoid the problem of vanishing gradients by allowing the model to remember long-term dependencies. Each LSTM layer consisted of 30 units and the outputs from the second LSTM layer are fed to a fully connected layer which is then directed to a single output node indicating the next term in the sequence or in other words, the predicted sales value. The weights of the model were updated based on the corresponding derivates of the Huber loss function and SGD with momentum was used as the optimizer of the model.

2.2.3 Prophet model

The objective behind the utilization of Prophet model is to capture any existing long-term seasonal patterns in the dataset if there is any such behaviour. The upper and lower bounds of the predictions were set in a manner that corresponds with the existing maximum and minimum values of the time series data, while the weekly seasonality was set to true, considering the generic observations of the dataset. The evaluation metric was mean absoluter percentage error as the challenge determined. After the evaluations, we concluded that the non-linear additive nature of the model strongly enforced the model to perform seasonally in prediction where it was always not the case for the challenge dataset.

2.2.4 SARIMA model

A SARIMA model was implemented to highlight the short-term seasonal behavior of the sales if there was any such periodic behavior. The optimum values for the parameters p, q, P, and Q were found after considering the AIC values for all the possible permutations of the selected ranges of those parameters. The parameters that yielded the minimum AIC value were used in the final SARIMA model. After a thorough inspection, we realized that the periodicity of the sales values was not that prominent. Yet, we decided to use the results of the SARIMA model in one of our ensemble model with negative weights.

2.2.5 Ensemble Approach

Testing results from four different models; transformers, LSTM + Conv1D, Prophet, and SARIMA were weighted to obtain the result. The intention was to combine the statistical and/or seasonal features extracted from statistical models with behavioral features extracted from the NN models. Higher weights were given to NN models considering their adaptability to a given dataset.

$$Output = \alpha * output_{transformer} + \beta * output_{LSTM} + \gamma * output_{SARIMA} + \delta * output_{prophet} \quad (1)$$

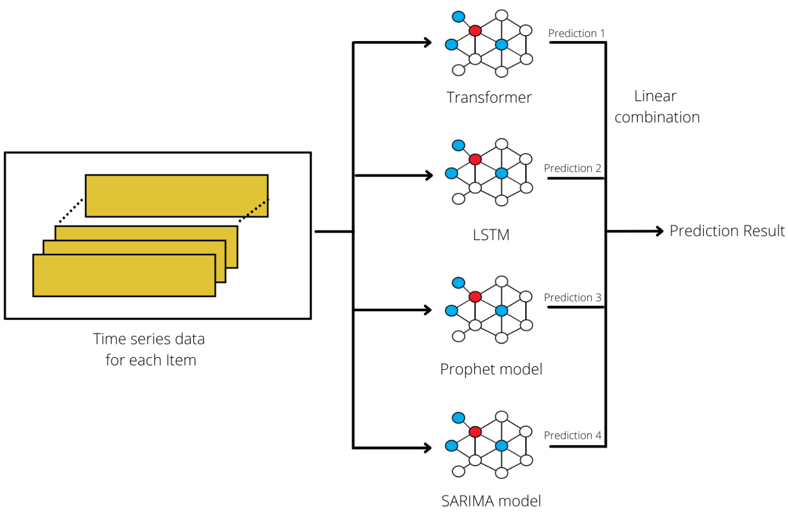


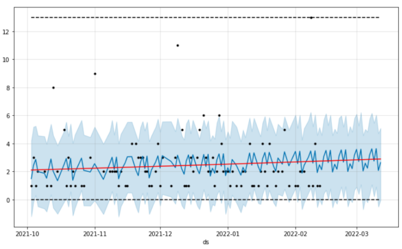
Figure 2: Proposed ensemble model with four base architectures

3 Results

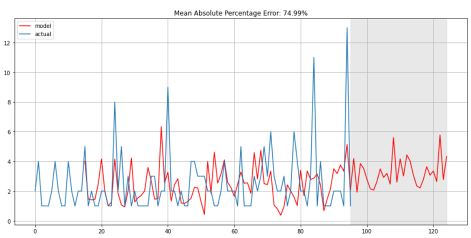
Through our four models, the following results were able to obtained and the corresponding python implementations are added to the GitHub repository. Through exploration of the results, we realized that the transformer model is slightly inefficient in learning long-term forecasting, while the LSTM model can learn in a decent manner with fewer layers. However, there is no room for a deeper network in LSTM domain because of the insufficient data with respect to particular one series. In addition, both statistical models perform poorly in the time series due to the lack of seasonality in the dataset. Furthermore, we showed that the assembly approach could lead to better results compared to univariate models that only consider one particular time series. We believe that the success of the ensemble model is an obvious result of the different features learned from different modalities in statistical and behavioural domains.

Table 1: In the ensemble model 1*, the values of the linear stacking parameters were $\alpha = 0.42$, $\beta = 0.42$, $\gamma = 0.08$ and $\delta = 0.08$ while in ensemble model 2**, the corresponding values were 0.85, 0.25, -0.05 and -0.05 respectively. Here, considered score is mean absoluter percentage error

Results from Kaggle submission		
Models	Private Score	Public Score
Transformer only	102.22	67.56
LSTM only	85.17	73.26
Ensemble model 1*	85.19	71.49
Ensemble model 2**	92.28	64.40



(a) Through Prophet model



(b) Through SARIMA model

Figure 3: Plotted resulting temporal distribution for item code 9925

4 Discussion

We believe that the performance of the ensemble model could be further improved using optimum linear coefficients for stacking and by exploring other techniques for ensembling such as using a fully connected neural network. Thus, it is obvious to say that ensembling methods of different selected models are a great pathway to study further to obtain better forecasting results which satisfactorily perform in the presence of abrupt technological innovations in the industries.

We believe that retail management should obtain a defined set of external parameters which affect each item since it could lead to building a multivariate ensemble model which is possibly better than a ensemble of univariates. In addition, it is better for the management team to intervene in a deep market analysis to understand the technological and research reasons behind the abrupt changes in the market in parallel with these kinds of forecasting models. Furthermore, the failure of generic statistical model depicts that the stability of the market is not assured and thus, should not be expected and thus, we believe that the forecasting strategy of the company should further investigate about deeper ML models which studies the possible market decisive factors which may be placed in collaboration with a ensembled sample space.