Evaluation of Machine Learning Techniques to forecast OList sales.

Malhar Thombare
Computer Science
University of California Riverside
Rivercide, CA
mthom121@ucr.edu

Bhagyesh Gaikwad
Computer Science
University of California Riverside
Rivercide, CA
bgaik001@ucr.edu

Chirag Rajavat
Computer Science
University of California Riverside
Rivercide, CA
craja008@ucr.edu

Gandharva Deshpande
Computer Science
University of California Riverside
Rivercide, CA
gdesh002@ucr.edu

Introduction

Time series prediction has always received special treatment when compared with other prediction problems. Its temporal nature makes it different from the others. We are trying to predict a time series of sales from OList, a Brazilian e-commerce website. The data only covers one and a half years, making it a bad representative of yearly seasonality. The small amount of data is also likely to overfit the models trained on it. Furthermore, It is a growing time series hence it is also essential for models to learn the growth patterns in it. We are evaluating four techniques for this prediction: XGBoost regression, Random Forest regression, LSTM neural network, and Deep Neural Network. Except for LSTMs, none of them are traditionally used for time series prediction. The XGBoost and Random forest typically need small amounts of data to train which makes them good candidates. Although LSTMs are used in time series problems we would like to see how challenging it could be for them to work in such a small amount of data. A simple Deep Learning model is tried to see where it stands when compared with other sophisticated techniques. We tried the techniques for different variations of the same problem. Various window sizes(no. of past days) are provided as input. Each window size is used to predict a different sales value. That is accumulated sales of 1, 7, 14, and 30 days in the future. We discuss the results and try to give explanations for them and also evaluate for ease of use (for LSTM and Deep Learning) and implementation(For XGBoost and Random Forest).

Relevant Work

Transformer is the newest addition in the favorite choices for series prediction and it took the spot for being the most favorite choice. The benefit of a transformer over recurrent nets is that it processes the whole input at once. This lets the transformer naturally give equal importance over the inputs at all time steps than just the recent ones. Other models like plain deep neural networks can take the whole input at once like the transformers but that may make them over parameterized and very difficult to train. Transformers need a large input to perform well and hence have a huge number of parameters. This makes transformer training difficult and time-consuming. Hence for the smaller data size transformers are reconsidered in case of small datasets. in In this sample-structured document, neither the cross-linking of float elements and bibliography nor metadata/copyright information is available. The sample document is provided in “Draft” mode and to view it in the final layout format, applying the required template is essential with some standard steps.

Long Short-Term Memory (LSTM) has been long known to solve the time series prediction problem in this research space. The paper: ‘Applying LSTM to Time Series Predictable through Time-Window Approaches’[4], explores LSTM’s application on the mentioned problem. Here the approach was taking every time step by looking at some past observations. One major observation is LSTM does not show much improvement even with some tasks that can be solved with timed window approaches. Consequently, the argument is found that if lagged observations are close to the time being forecasted, LSTM may not be the best choice for forecasting. The problem that we are solving deals with less amount of data and is also susceptible to seasonality, meaning that the observations are close to the time at which they are being forecasted. We further explore by applying LSTM as one of the implementation methods for our problem.

Decision Tree algorithms, along with SVM, are commonly used to development of the regression analysis models, in addition to solving data classification problems. The paper "Development and Research of the Forecasting Models Based on the Time Series Using the Random Forest Algorithm " discusses the random forest algorithm to the problem of predicting the remaining useful life of the equipment of a complex technical system by developing the appropriate regression model using datasets based on multivariate time series. A method for improving the accuracy of forecasting the remaining useful life(RUL) of equipment by engineering additional features produced using the original dataset based on multivariate time series is studied in this paper. The regression model was created for number of trees from 10 to 200, with steps of 10. It was observed that number of trees equal to 100 was optimal. For values less than 100, the expectation by the number of features is increased significantly. The features were also sorted in dcreasing order of their importance, resulting in desired score and decrease in variance. Also, with decreased in features the MAE (mean absolute error) is decreased. Overall, it was concluded that it is possible to use Decision Trees for development of Regression models, and forecasting accuracy improves with choosing optimal parameters.

Proposed Techniques

XGBoost:

XGBoost combined many past ideas to create a strong algorithm. For long time trees are used for prediction tasks. It is an improvement to gradient boosted trees by adding more regularization.

The XGBoost method is an ensemble model of regression trees. Each tree leaf is given a weight that acts as a prediction value. The trees are built sequentially each tree moves its prediction in a direction of the gradient.

The very first prediction is a tree with just a leaf node. The initial prediction that is the score on this leaf could be either zero or the mean of outputs. Using this prediction gradients are calculated and a new tree is added which in a way predicts these gradients. The scaled output of the new tree is added to the existing prediction to improve it. The scaling factor is a step size or a learning rate. This ensures the model doesn’t overfit a single tree to the data and instead becomes robust by using multiple trees.

Random Forest Regression:

A random forest employs ensemble learning, a technique that uses multiple classifiers to solve complex problems. A random forest algorithm is composed of many decision trees. The 'forest' of the random forest algorithm is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm ensemble that improves machine learning algorithm accuracy.

A decision tree is just a graph of trees. It is made up of leaf nodes, decision nodes, and edges. The root is one of the decision-making nodes because it is the foundation of the tree and from here we can reach any other node. The leaf nodes are always at the conclusion of the path on the graph.

The decision node checks the criteria for separating the set into disjoint subsets, and each edge represents one of the alternatives. The quality criteria (indicators) such as mean squared error (MSE) and mean absolute error (MAE) are frequently used as test conditions. The random forest algorithm determines the outcome based on the decision trees' predictions. It predicts by taking the mean of the output of various trees. The precision of the outcome improves as the number of decision trees grows.

Long Short-Term Memory (LSTM):

LSTM are a special case of recurrent neural networks (RNN), which can remember the data in memory, they are thus capable of learning order dependence solving the vanishing gradient problem and used for sequence prediction problems.

This method was a result of the analysis of error flow in existing RNNs, in the previous methods the lags over longer time steps were inaccessible thus giving us no ways to backpropagate enough to rectify the errors.The different memory blocks in the network have interconnected cells which have input, output and forget gates that provide operations of write, read and reset for the cells. Thus the Information can be added to or removed from the cell state in LSTM and is regulated by gates.

Deep feed-forward neural network:

Since linear approaches are easily understood and effective in many basic predicting situations, linear methods have traditionally dominated time series forecasting.

Deep learning neural networks can learn arbitrary complex input-output mappings automatically and handle numerous inputs and outputs.

Perceptrons with Multiple Layers (MLPs) In general, neural networks, such as Multilayer Perceptrons (MLPs), give characteristics that few algorithms can match, such as:

Noise-resistant. Neural networks are resistant to noise in the input data and the mapping function, and they can even learn and predict in the absence of data.

Nonlinear. Neural networks learn linear and nonlinear relationships without making strong assumptions about the mapping function.

Inputs that are multivariate. It is possible to provide an unlimited number of input features.Forecasts with multiple steps. There is no limit to the number of output values that can be supplied.Multi-step and even multivariate forecasting have direct support.Feedforward neural networks may be effective for time series forecasting simply because of these features

Experimental Evaluation

These algorithms are evaluated by the following evaluation criteria.

- Performance of model on different input sizes. That is data from the past 1 day, 7 days, 14 days and 30 days as seen in below figures.
- Performance of model for predicting sales of different tenures. That is cumulative sales of the next 1 day, 7 days, 14 days and 30 days.
- Mean error percentage from original value and standard deviation of it.
- Comparison of these performances with Moving Average.

We found that parameter tuning for XGBoost and Random forest did not improve the performance further hence the default parameters are used. We tuned the parameters for LSTM and DNN but still, their performance did not surpass the algorithms using trees.

As reported in the Midterm report The XGBoost and Random continued to perform better for the given problem. One of the reasons could be because of the dataset size. LSTM and Deep Learning models are susceptible to overfitting when trained with smaller datasets.

Note : Please find the comparison images at the end of this report

Future Work Discussions:

The data that we used for evaluating the techniques seems to be too small for the current techniques to be useful, for the same techniques to be fruitful we can try different time series preprocessing techniques to remove the noise. For the smaller data sets, we recommend using XGBoost or Random Forest but these models won't scale with the inputs, and hence for each new batch of data, we should consider retraining these models.

For Neural networks we can try strong regularization techniques. These models are strong learners but can also learn the noise in the data.

Conclusions:

- XGBoost and Random Forest can be used for time-series data. Their specialty is robustness to the size of the dataset.
- The results we obtained for these techniques are not promising but that might be due to noise in the underlying data set.
- DNN and LSTM are difficult to train given the small and noisy dataset.
- The data that we used is noisy and it is difficult to filter out for the models when provided as is.

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785–794. ACM, 2016.

[2] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. arXiv:2001.08317 [cs.LG]

[3] Liliya Demidova and Mariya Ivkina. Development and Research of the Forecasting Models Based on the Time Series Using the Random Forest Algorithm. Lipetsk, Russia IEEE 2020 ISBN:978-1-7281-8113-4

[4] F. Gers, D. Eck, and J. Schmidhuber. Applying LSTM to time series predictable through time-window approaches. Technical report IDSIA-IDSIA-22-00, IDSI/USI-SUPSI, Instituto Dalle Molle di studi sull' intelligenza artificiale, Manno, Switzerland, 2000. http://www.idsia.ch/~felix/Publications.html.

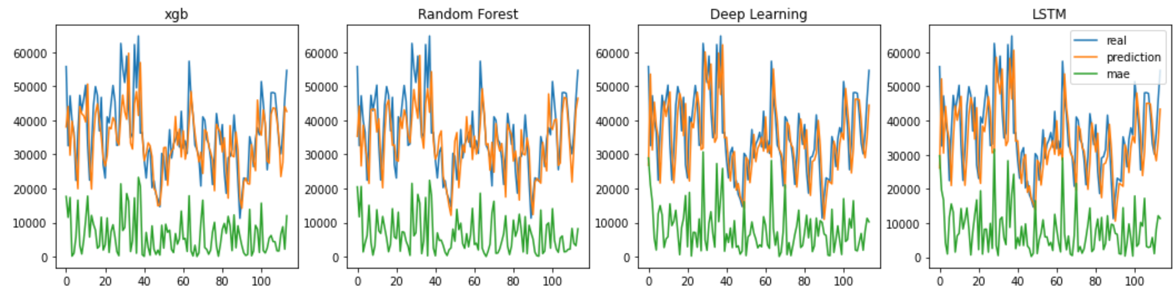

Figure 1: **Plots for: Prediction for 1 day with window size 1**

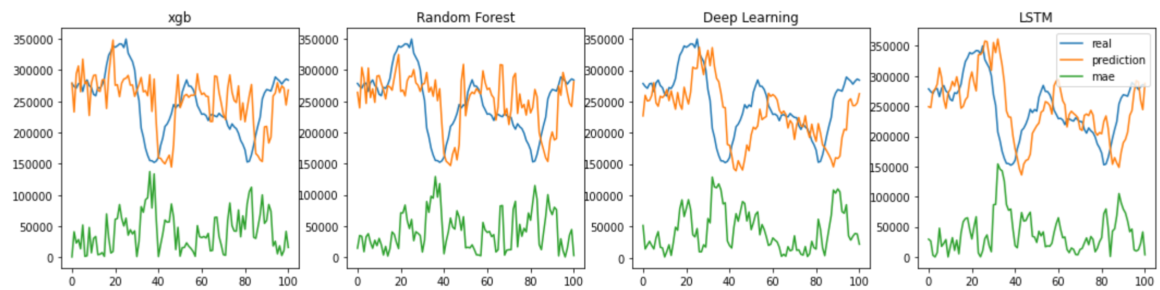

Figure 2: **Plots for: Prediction for 7 day with window size 7**

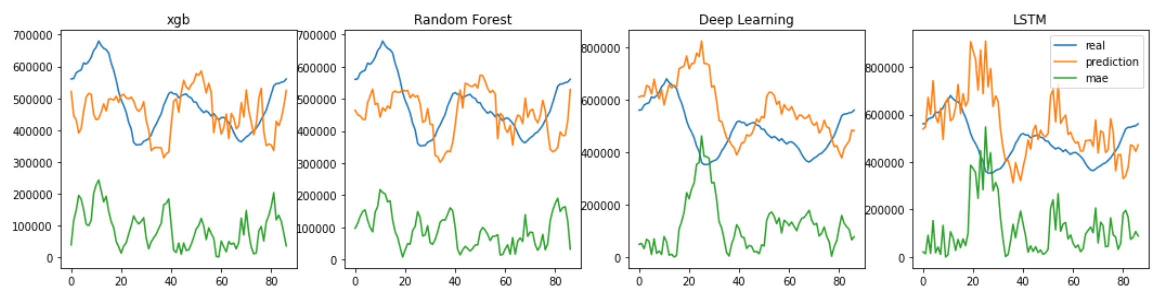

Figure 3: **Plots for: Prediction for 14 day with window size 14**

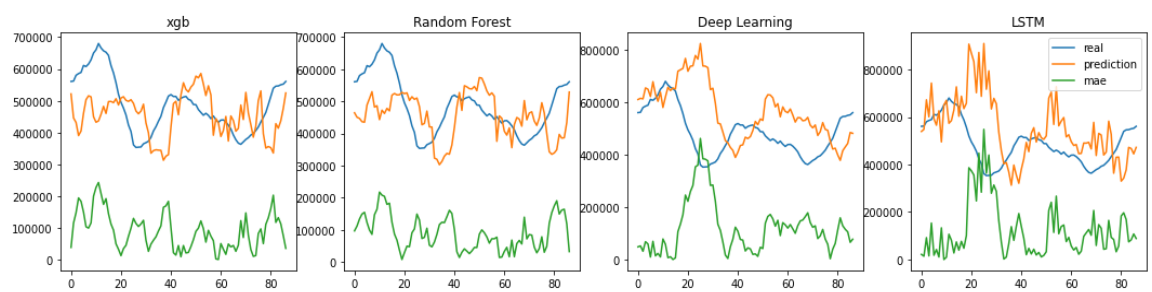

Figure 4: **Plots for: Prediction for 30 day with window size 30**