www.turkishstudies.net/turkishstudies

Turkish Studies

eISSN: 1308-2140

Research Article / Araştırma Makalesi

INTERNATIONAL
BALKAN
UNIVERSITY

Sponsored by IBU

Time Series Performance and Limitations with SARIMAX: An Application with Retail Store Data

SARIMAX ile Zaman Serisi Performansı ve Sınırlamaları: Perakende Verileri İçeren Bir Uygulama

Emre S. Özmen*

Abstract: World giant retailers' sales data competes with stock exchange data in respect to latency, where the number one reason is the number of transactions, it is around few hundreds per second and it only goes up in time being. This emerges the idea of making an ideal application area for time series, however the field looks like lacking comparisons. This is an attempt to address its dynamics with a generic reference data that was published with Walmart retail figures. Time series has a long list of predictive models, however they are all based on regression and the problem is that regression is not always make the leader model. Here we explored the possible performance bottlenecks with most popular techniques, like ARIMA/SARIMA derivative and try to answer if regression limits time series foundations. Does the best fit with ARIMA derivatives always give the best scores? In other words, per time-series performance, does this make a lagging or leading factor? This situation created two new question groups for future studies. The first was concerned with the possibility that time series, which have not yet been exemplified in the world, are based on classification models rather than regression. In other words, the possible positive effect of fuzzy logic applications on performance, which matches the size of the classification models that produce output in binary number order with the size of turnover in pre-output estimates based on probability percentages. The second was the fact that time series, which by definition do not accept data other than time, will not produce an important factor, and to what extent this effort might be needed.

Structured Abstract: Times series is one of the oldest prediction models in a way that only accepts timedata. Therefore, it has a natural implicit stance that favors as much as timedata, in other words shorten time intervals with higher frequencies. In this 'big data' era that we live in, this type of data is almost everywhere and knowing that payment methods are miscellaneous, as well as receptive as never experienced before in the commerce history, this is even more true with sales data. However, time series applications are still relatively limited with stock exchange derivatives and limited for mass sales membranes like retailers (Raiyani, 2021; Keles et al, 2020; Chen et al, 2019)

Sales predictions is still one of the most driven topics in machine learning. Sponsors perspective, having a better ability to predict future numbers can be awarded much faster and more than any other alibi (Bera, 2021; Catal, Ece, Arslan, Akbulut, 2019; Liu et al, 2007) Therefore, few hundreds per second latency should make stakeholders to consider of producing more time series applications and more importantly its performance comparing to non-time series models. This study first aims to serve a rare time series application with real sales data then discuss its necessity in terms of value addition per its predictive ability.

* Dr. Öğr. Üyesi, Nişantaşı Üniversitesi, Lisansüstü Eğitim Enstitüsü
Asst. Prof. Dr., Nisantasi University, Graduate School of Management
ORCID 0000-0002-8384-5468
emre.ozmen@nisantasi.edu.tr

Cite as/ Atıf: Özmen, E. S. (2021). Time series performance and Limitations with SARIMAX: An application with retail store data. *Turkish Studies*, 16(5), 1583-1592. <https://dx.doi.org/10.7827/TurkishStudies.49699>

Received/Geliş: 27 February/Şubat 2021

Checked by plagiarism software

Accepted/Kabul: 25 October/Ekim 2021

CC BY-NC 4.0

Published/Yayın: 30 October/Ekim 2021

The first challenge with applications on sales data is the data itself. Since when it is about sales transactions, usually there is no real data and even if there is, there is either no permission or it is small, or the data owner company has no dimensionality in respect to company size. For few occasions, this combination exists and when it does, it is usually thru a competition, like we experienced here with Kaggle (Bojer, 2021; Miller et al., 2021). When it comes to sales, hundreds of data per second volume is only a possibility with retailers, like we have here with Walmart's 50 stores, 30 department sales data for 2 years yielding 500,000 records, in other words rows. The publicly provided data has no confidentiality close with almost half a million of records/rows, one of the most extensive lists in the world.

Three files have been merged and an extensive time have been spent with timedata attributes of the 500,000 rows sales data. Due to seasonality, statsmodels' SARIMAX was practiced with p, d, q and n attributes. As seen on "Figure 2", fitted results yielded a descent demonstration with only some slight leads. Deployed model, as seen on "Figure 4", shows a sales decline with shrank amplitudes. To check its position, other machine learnings models were applied.

Indeed, according to the study data, we have observed that time series, which give a successful fit, fell behind other models in machine learning such as 'Decision Tree' and 'Random Forest'. For this reason, we noted that the relatively low number of time series sales prediction trials may not only be due to the laborious preparation phase of time series. Knowing that the natural comparison environment is due to regression may also reduce the motivation for the effort of a possible picture. This study emerged into two avenues for future studies:

- The first was concerned with the possibility that time series, which have not yet been exemplified in the world, are based on classification models rather than regression. In other words, the possible positive effect of fuzzy logic applications on performance, which matches the size of the classification models that produce output in binary number order with the size of turnover in pre-output estimates based on probability percentages.

- The second was the fact that time series, which, by definition, do not accept data other than time, will not produce an important factor, and to what extent this effort might be needed.

Time series, by definition, accept nothing but timedata and even it is not explicitly noted, is being considered with better performing metrics with higher frequencies. Having the golden age of big data gives nothing but an opportunity this is to be tested. After all, the tacitly well-embraced dichotomy that time series will perform only better with more timedata does not even seem to be correct. Number of records does not change the length of prediction, it is considered as 100 days, either it's trained by 2-year data or 10-year data. In other words, even if there is no label other than timedata and the only option is the time series, big data does not seem helping time series performance.

The data observed is significant, it has 421,570 rows with 18 features including the timestamp, where the prediction label set as the retail revenue. The inclusion of timestamp brings an implicit intention for the utilization of time series. This paper questions the unconditional acceptance of timestamp as a leader model. As a matter of fact, later findings barely suggest a regression structure, which makes the mathematical basis of a time series. The vast majority of the leader board is dominated by classification (Random Forest, Gradient Boost and Decision Trees) models, rather than regression (which is unfortunately ranked as the last one on the list, GLM) models including timestamp, that also recommends to revisit our tacit knowledge which is considering time series as supposedly the best, just because we observe timestamp as a part of the data. It is suggested to validate with all regressors, as well as classifiers which resume the best performance here, the first 6 out of 7 best predictors were compartmentalized as classifiers, not regressors with or without time series.

Keywords: Computer science, Time series, SARIMAX, Retail, Classification

Öz: Dünya devi perakendecilerin satış verileri yoğunluğu borsa verileriyle rekabet etmektedir. Bir numaralı nedenin işlem sayısı olduğu bu dünyada yenilenme oranı saniyede birkaç yüz civarında seyretmekte, bu da satış tahmin noktasında örnekleri az olan zaman serileri için ideal bir uygulama alanı yapma fikrini ortaya çıkarmaktadır. Zaman serilerinin uzun bir tahmin modelleri listesi olduğu bilinmektedir, ancak hemen hepsi regresyona dayanır ve regresyon her zaman lider model değildir. Burada ARIMA / SARIMA türevi gibi en

popüler zaman serisi teknikleriyle olası performans darboğazlarını araştırdık ve regresyonun zaman serilerinin temellerini sınırlayıp sınırlamadığını yanıtlamaya çalıştık. Bu durum ilerideki çalışmalar için iki yeni soru grubu yarattı. Birincisi, dünyada henüz örneklenmemiş olan zaman serilerinin regresyondan ziyade sınıflandırma modellerine dayanma olasılığı ile ilgili oldu. Diğer bir deyişle, ikili sayı sıralamasında çıktı üreten sınıflandırma modellerinin boyutu ile olasılık yüzdelerine dayalı ön çıktı tahminlerinde ciro büyüklüğünü eşleştiren bulanık mantık uygulamalarının performans üzerindeki olası olumlu etkisi incelendi. İkincisi, tanımı gereği zaman dışında veri kabul etmeyen zaman serilerinin önemli bir faktör üretmeyeceği ve bu çabaya ne ölçüde ihtiyaç duyulabileceği ile ilgili oldu. Sonuç olarak, zaman serilerinin ancak daha fazla zaman verisiyle daha iyi performans göstereceği şeklindeki zımnen doğru kabul edilen gerçek şüphe uyandırmış, yeni ikilemler doğurmuştur, ki ileri araştırmalar için bir fırsat olarak görülebilir.

Anahtar Kelimeler: Bilgisayar Bilimleri, Zaman serileri, SARIMAX, Parakende, Klasifikasyon

Walmart Store Sales Forecasting: Official data description

Data is provided with around 50 Walmart stores located across the United States. Each store is compartmentalized per number of divisions, so categories can be scrutinized with ease. Kaggle also notes that Walmart runs campaign throughout the year. This is especially more valid when it comes to big holidays. There are four big holidays during the year and the weeks consisting a big holiday have five times higher sales than the normal week run rates. What make this trophy challenging is that the way the data presented by this competition is not an ideal historical data, missing figures with irregular weeks (Kaggle, 2020).

There are three files that train directory comes in a hybrid stationary form, where the apparent seasonality can be categorized as a variation of non-stationary time series. Store file contains the type and size of store. The training data, having half a million rows, covers from 5th of May 2010 to the 1st November 2012. Test file is around 20% split and identical to train, with the weekly sales exception. Besides holiday stops, some attributes that can be tied with shopping terms were also included. In this regard, weather conditions, fuel consumption level, purchasing power, unemployment indices were mentioned.

Visiting ARIMA, SARIMA, SARIMAX models background

ARIMA (Autoregressive Integrated Moving Average) is a variation of ARMA (Autoregressive Moving Average) and is being utilized in the sake of understanding time series in a better way and more often predicting with higher accuracy. ARIMA has typically three parameters referring to p, d and q with autoregressive, differencing and moving average respectively. ARIMA is interpreted as follows, where L is lag operator, ε_t is error (Arunraja, N.S., Ahrens, D., Fernandes, M. and Müllera, M., 2014).

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \varepsilon_t$$

Seasonality exists in patterns and it favors occurrences within a particular frequency. Formula is similar but with an upper-case utilization of autoregressive, differencing and moving average parameters and most importantly an n that refers to trend.

$$\text{ARIMA} (p, d, q) \quad \text{vs} \quad \text{SARIMA} (P, D, Q, n)$$

$$\text{Non-seasonal} \quad \text{vs} \quad \text{Seasonal}$$

For instance, an ARIMA(1,1,1)(1,1,1, 12) model (without a constant) is for monthly data (n=12), and can be written as below, where the non-seasonal ones are expressed with lower-case ϕ and θ , while their seasonal counterparts are expressed with upper-case Φ and Θ respectively.

$$(1 - S)(1 - \phi_1 S_{12})(1 - S)(1 - S_{12})y_t = (1 + \Phi_1 S)(1 + \Theta_1 S_{12})\varepsilon_t$$

SARIMA is also called as SARIMAX because the “X” addition to the method name accepts that the implementation also supports exogenous variables.

Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model

The module was originally developed by Jonathan Taylor, Josef Perktold, Skipper Seabold. At the beginning, it was part of scipy but was later removed. During the Google Summer of Code 2009, statsmodels was corrected, tested, improved and released as a new package. Since then, the statsmodels development team has continued to add new models, plotting tools, and statistical methods (Statsmodel, 2020). Per readability perspective, the library imports were not separately mentioned in the codes. Most dominantly numpy and pandas were utilized for data wrangling, datetime was used for timestamps, matplotlib for graphical representation and sklearn for data imputer and label encoding for preprocessing, alternate machine learning models, respectively, linear regression, k neighbors regression, ridge, lass, decision tree and random forest for model alternatives.

Statsmodelstatsmodels.tsa.statespace.sarimax.SARIMAX

The usage model is as follow;

```
class statsmodels.tsa.statespace.sarimax.SARIMAX(endog, exog=None, order=1, 0, 0,
seasonal_order=0, 0, 0, 0, trend=None, measurement_error=False, time_varying_regression=False,
mle_regression=True, simple_differencing=False, enforce_stationarity=True,
enforce_invertibility=True, hamilton_representation=False, concentrate_scale=False,
trend_offset=1, use_exact_diffuse=False, dates=None, freq=None, missing='none',
validate_specification=True, **kwargs)[source]
```

Arrangements for timedates;

Data is observed with 1,000 points and stationary stance with modalities thru seasonal effects. This can also be perceived as non-stationary with no seasonal aspects. There are two major and two minor peaks with decreasing magnitudes. Non-seasonal elements look like a having a more fluctuating pattern comparing to past years' same period. (Figure 1)

```
1 data = train.merge(features, on=['Store', 'Date'], how='inner').merge(stores, on=['Store'], how='inner')
2 from dateutil.parser import parse
3 dates = pd.DatetimeIndex([parse(d).strftime('%Y-%m-%d') for d in data['Date']])
4 data.insert(2, 'dates', dates)
5 #data.set_index(dates, inplace=True)
6 print(data.shape)
7 display(data)
8 type(data)
9 ts=data.groupby('dates').Weekly_Sales.sum()
10 display(ts)
11 type(ts)
12 tsdf=ts.to_frame()
13 display(tsdf)
14 # tsdf.Date
15 plt.figure(figsize=(20,8))
16 ts.plot()
```

(421570, 18)

Figure 1: Holistic representation

Excellent Model Fit

Per p, d, q order, although derivatives with 0, 1 and 2 were attempted for combinations, default system values were applied with 1, 0 and 0, where d refers to integration and p, q favor AR, MA respectively. Trend eliminates linear (c) and polynomial (t) patterns. Using the same exact training data, test fitting was observed with slight minimal delays however model parameters are integers and they are not sensitive enough to make further arrangements (Slimani, El Farissi & Al-Qualsadi, 2016). For P, D, Q, same default values were applied. For S was subject for few variations. First attempt was made with 4, in other words a quarter per year, then 2 as the semi-annually, 12 as monthly and lastly 52 as weekly were applied. It is observed that 52 week yields the best fit, however increasing this was not a possibility, due to annual frequencies. (Figure 2)

```
1 from statsmodels.tsa.arima_model import ARIMA
2 # include the Seasonal Arima Model for forecasting, SARIMAX
3 from statsmodels.tsa.statespace.sarimax import SARIMAX
4 model = SARIMAX(ts, trend='n', order=(1,0,0), seasonal_order=(1,0,0,52))
5 results_ARIMA = model.fit(dis=-1)
6 plt.figure(figsize=(20,8))
7 plt.plot(ts, label='original')
8 plt.plot(results_ARIMA.fittedvalues, color='red', label='fitted')
9 plt.title('RSS: %.4f'% sum((results_ARIMA.fittedvalues-ts)**2))
10 plt.legend()
11 #type(ts)
```

Figure 2: Model fitting

To have a better view, date interval was delimited by one third and it is noted that slight lagging figures become more apparent. The data does not overlap fully, however revenue wise the delta is only about 9% less than the actual data. At this point, an interesting remark that might need a decision arises. What is more favorable? Being 9% more or less than actual is a true management question, rather scientists. Hypothetically, this answer may also change in time being and may not have a definitive conclusion.

Per timestamp perspective, the delay refers to a week. Time series data are good for at least 1000 entries as the training data and try to predict for the next 100, as the test. Here we have a

1000-day data, and 7-week delay can be judged by 100 days in total. 7% might be acceptable all means, even though we predicted a bit more than this, it is around 4 months. (Figure 3)

```

1 df = pd.DataFrame(ts)
2 # Add a forecast column to the dataframe that includes our predictions from the model
3 df['forecast'] = results_ARIMA.predict(start = '2010-02-05', end= '2012-10-26', dynamic= False)
4 #display(df)
5 # Plot the Dataframe data
6 df[['Weekly_Sales', 'forecast']].plot(figsize=(12, 8))
7 # Show the plot
8 plt.show()

```

Figure 3: Model fitting - Closer look

Model deployment

To burst a better match, we declared a 52-week seasonality, instead of 4-quarter verbiage, as the *s* parameter. To remember, per *p*, *d*, *q* order, default system values were applied with 1, 0 and 0, where *d* refers to integration and *p*, *q* favor AR, MA respectively. Trend eliminates linear (*c*) and polynomial (*t*) patterns. Using the same exact training (from the 5th of February, 2010 to the 26th of October, 2012) set, we extended with the test (from the 27th of October, 2012 to the 26th of February) data back-to-back and we observed a significant fall in sales, shown in orange color. Drop is aligned with our preliminary findings about decreasing amplitudes of peaks, it shows a fading pattern for holiday shopping. This might also be related to a cultural shift of modern era. Do holidays mean less than it was earlier? Nevertheless, its weight in overall numbers is significant and a fading pattern does not look like helping for the greater good of the company. (Figure 4)

```

1 pred=results_ARIMA.predict(start = '2010-02-05', end= '2012-10-26', dynamic= False)
2 pred.plot()
3 pred=results_ARIMA.predict(start = '2012-10-27', end= '2014-02-26', dynamic= False)
4 pred.plot()

```

Figure 4: Prediction

Data wrangling for alternate machine learning models

To mitigate the anomalies that we detected with slight lagging figures for both timestamp and the revenue, we favor alternate models and check the possibilities with few more, including linear regression, k neighbors regression, ridge, lasso, decision tree and random forest (Vishwas & Patel, 2020, Lai, Fana, Huang & Chang, 2009). Before we proceed, few steps with imputing and encoding the date are needed with labels. Markdown, referring to holidays, were split into 5 columns for float to binary positioning.

```

1 df = train.merge(features, on=['Store', 'Date', 'IsHoliday'], how='inner').merge(stores, on=['Store'], how='inner')
2 print(data.shape)
3
4 markdown=pd.DataFrame(SimpleImputer().fit_transform(df[['Markdown1', 'Markdown2', 'Markdown3', 'Markdown4', 'Markdown5']]),
5                       columns=['Markdown1', 'Markdown2', 'Markdown3', 'Markdown4', 'Markdown5'])
6 df = df.drop(['Markdown1', 'Markdown2', 'Markdown3', 'Markdown4', 'Markdown5'], axis=1)
7 df=pd.concat([df, markdown], axis=1)
8
9 df['Date']=pd.to_datetime(df['Date'])
10 df['year']=df['Date'].dt.year
11 df['month']=df['Date'].dt.month
12 df['day']=df['Date'].dt.day
13 del df['Date']
14
15 tf['Date']=pd.to_datetime(tf['Date'])
16 tf['year']=tf['Date'].dt.year
17 tf['month']=tf['Date'].dt.month
18 tf['day']=tf['Date'].dt.day
19 del tf['Date']
20
21 df['Type']=LabelEncoder().fit_transform(df['Type'])
22 df['IsHoliday']=LabelEncoder().fit_transform(df['IsHoliday'])
23 tf['IsHoliday']=LabelEncoder().fit_transform(tf['IsHoliday'])
24 df.dtypes

```

Figure 5: Feature engineering

Stacked Ensemble is the leader model, not regression/autoregression/SARIMAX

Stacked ensemble is the leader model with the top 10% ranking of the participant list, where the Gradient boosting is the follower. ARIMA/SARIMAX models do not look like make a robust model, other than a demonstration (Geurts, 2002, Torgo and Gama, 1997). Furthermore, even its demonstration might be a false signal, since the fit cannot be better by the look. This situation emerges more questions than answers and questions the reliability of existing offerings.

Which one should may depend on, perfect fit or predicting quality? Normally, this might be question to brough up the overfitting, however in the time series world overfitting was never be a valid topic to dig in, since it is a part of its signature. A need to be validated might look like a time-consuming task and cause a juxtaposition.

Table 1: Model leaderboard

	model_id	mean_residua_devia	rmse	mse	mae	rmsle
	StackedEnsemble_BestOffFamily_AutoML_2021	2.029e+07	3243.8	2.02934e+0	2056.6	nan
	StackedEnsemble_AllModels_AutoML_20210118_	2.22934e+07	3928.8	2.224e+07	2256.6	nan
	GBM_3_AutoML_20210118_172408	2.51177e+07	5011.7	2.517e+07	2661.5	nan
	GBM_2_AutoML_20210118_172408	3.26845e+07	5717.0	3.265e+07	3076.1	nan
	GBM_1_AutoML_20210118_172408	3.80761e+07	6170.5	3.801e+07	3392.2	nan
	DRF_1_AutoML_20210118_172408	5.51084e+07	7423.5	5.514e+07	4015.3	nan
	GLM_1_AutoML_20210118_172408	5.15796e+08	22711.	5.156e+08	15161.	nan

Discussions

Walmart knew what participants did not, the predictions were supposedly showing that the sales will drop. Independent from the context, the confidence of Walmart with sharing an implicitly decreasing sales number is remarkable. Drop is aligned with our preliminary findings about decreasing amplitudes of peaks, that it shows a fading pattern for holiday shopping. This might also be related to a cultural shift of modern era. Do holidays mean less than it was earlier? Nevertheless, its weight in overall numbers is significant and a fading pattern does not look like helping for the greater good of the company.

Per model discussions, we were targeted on comparing regression-based time series predictions to all available others. Sarimax, in the extent of statsmodel, has a perfect model fit. However, it did not necessarily produce the best scores and the reason looks like being a part of regression models' overall limitations. From deductive perspective, as long as time series make variations of regression models, it will always face challenge from classification category. When it happens, it tacitly compartmentalizes the context into dstacked ensemble or gradient boosting deployment, like it is practiced here in this study.

Overfitting was discussed as a possible reason for this, however, when it comes to time series, having a perfect fit was never considered as a hurdle. In the contrary, not being able to find a perfect match usually is the problem self. In the end, from performance standpoint, problems of ARIMA derivatives need to have a resolution for better results. Otherwise, it will be fair to explicitly note that those were limited with nice graphical representation but no efficient yield.

Conclusion

The data observed is significant, it has 421,570 rows with 18 features including the timestamp, where the prediction label set as the retail revenue. The inclusion of timestamp brings an implicit intention for the utilization of time series. This paper questions the unconditional

acceptance of timestamp as a leader model. As a matter of fact, later findings barely suggest a regression structure, which makes the mathematical basis of a time series. The vast majority of the leader board is dominated by classification (Random Forest, Gradient Boost and Decision Trees) models, rather than regression (which is unfortunately ranked as the last one on the list, GLM) models including timestamp, that also recommends to revisit our tacit knowledge which is considering time series as supposedly the best, just because we observe timestamp as a part of the data. It is suggested to validate with all regressors, as well as classifiers which resume the best performance here, the first 6 out of 7 best predictors were compartmentalized as classifiers, not regressors with or without time series.

References

- Arunraja, N. S., Ahrensb, D., Fernandes, M. & Müllera, M. (2014). Time series sales forecasting to reduce food waste in retail industry. *The 34th International Symposium on Forecasting*, Tokyo, 118-134.
- Bera, S. (2021). An application of operational analytics: For predicting sales revenue of restaurant. *In Machine Learning Algorithms for Industrial Applications* (pp. 209-235). Springer, Cham. <https://doi.org/10.1007/978-3-030-50641-4>
- Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2), 587-603. <https://dx.doi.org/10.1016/j.ijforecast.2020.07.007>
- Chen, D., Liu, Q., Yang, X., Yu, X., Ma, F., & Su, J. (2019, October). Self-Adaptive Particle Filter Based Time Series Prediction of Online Retailer Daily Sale. *12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (pp. 1-8). IEEE. <https://dx.doi.org/10.1109/CISP-BMEI.2018.8633040>
- Catal, C., Ece, K., Arslan, B., & Akbulut, A. (2019). Benchmarking of regression algorithms and time series analysis techniques for sales forecasting. *Balkan Journal of Electrical and Computer Engineering*, 7(1), 20-26. <http://dx.doi.org/10.17694/bajece.494920>
- Geurts, P. (2002). *Contributions to decision tree induction: bias/variance tradeoff and time series classification* (Doctoral dissertation, ULiège-University of Liège).
- Kaggle. (2020, November 1). Walmart Recruiting - Store Sales Forecasting: Use historical markdown data to predict store sales. <https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/overview>
- Keleş, M. B., Keleş, A., & Keleş, A. (2020). Yapay zekâ teknolojisi ile uçuş fiyatı tahmin modeli geliştirme. *Turkish Studies*, 15(4), 511-520. <https://dx.doi.org/10.47844/TurkishStudies.45993>
- Lai, R. K., Fan, C. Y., Huang, W. H., & Chang, P. C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36(2), 3761-3773. <https://doi.org/10.1016/j.eswa.2008.02.025>
- Liu, Y., Huang, X., An, A., & Yu, X. (2007, July). ARSA: a sentiment-aware model for predicting sales performance using blogs. *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 607-614). ACM. <https://doi.org/10.1145/1277741.1277845>
- Miller, C., Picchetti, B., Fu, C., & Pantelic, J. (2021). Limitations of machine learning for building energy prediction: ASHRAE Great Energy Predictor III Kaggle competition error analysis. arXiv preprint arXiv:2106.13475.

- Raiyani, A., Lathigara, A., & Mehta, H. (2021). Usage of time series forecasting model in Supply chain sales prediction. In *IOP Conference Series: Materials Science and Engineering* (pp. 1-8). IOP Publishing. <http://dx.doi.org/10.1088/1757-899X/1042/1/012022>
- Slimani, I., Farissi, I. E., & Achchab, S. (2017). Configuration and implementation of a daily artificial neural network-based forecasting system using real supermarket data. *International Journal of Logistics Systems and Management*, 28(2), 144-163. <https://dx.doi.org/10.1109/GOL.2016.7731709>
- Statsmodel. (2020, November 1). Statsmodels.tsa.statespace.sarimax.SARIMAX. <https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>
- Torgo, L. and Gama, J. (1997). Regression using classification algorithms. *Intelligent Data Analysis*, 1(4), 271-293. [https://doi.org/10.1016/S1088-467X\(97\)00013-9](https://doi.org/10.1016/S1088-467X(97)00013-9)
- Vishwas, B. V., & Patel, A. (2020). Regression Extension Techniques for Time-Series Data. In *Hands-on Time Series Analysis with Python*. 99-184. Apress (Springer)