Yet Another an End-to-End Time Series Project Tutorial

By George Vinogradov · towardsdatascience.com · 12 min

View Original

source: unsplash.com

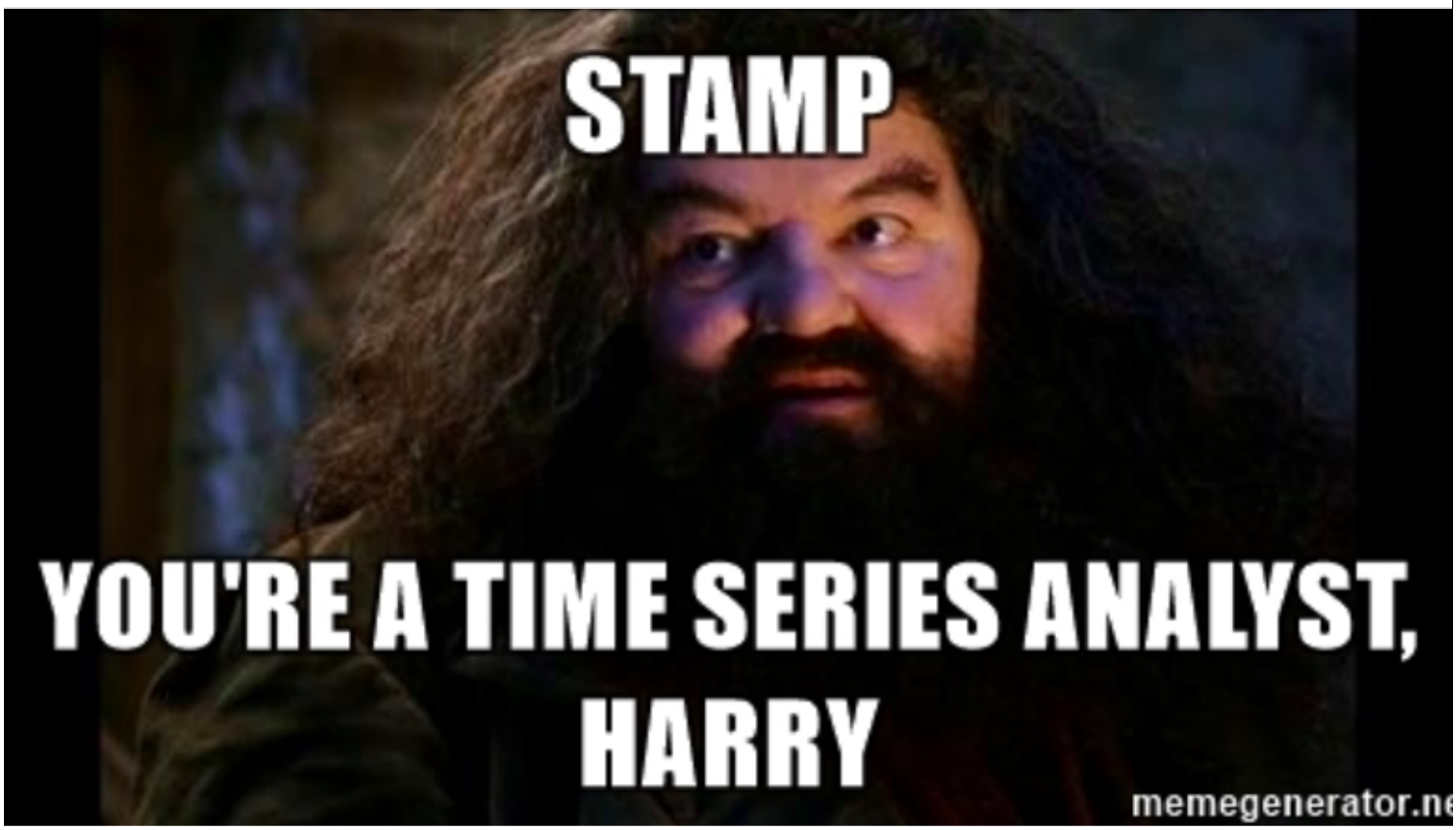
Yet Another an End-to-End Time Series Project Tutorial

Apr 19 · 13 min read★

> *Time series analysis is one of the most common and at the same time most requested tasks of the classical data science area. You can observe this type of data everywhere: stocks and financial trading, data generated by IoT, online and offline retail sales and even medical records such as heart rate, EKG, EEG, and MRI.*

In this article I described in detail the general approach to execution from scratch of a time-series project from scratch with certain examples. First and foremost, newbie analysts and project managers from the data science area could consider this content useful, but if you are kinda crusty analyst I believe you also might be pleased to see some interesting thoughts.

The code of the entire project is **available** on the **GitHub** repository



source: memegenerator

Time series prediction or forecasting is the top of the most common problems that analysts or data scientists try to figure out when they are faced with that kind of data. Also, the developing internet of things area creates anomaly detection tasks, however, in this article we are going into details of sequence series models creation for future prediction.

Before using the modern machine learning models under time series highly important to take into account some their essential concepts:

- Machine learning models have no awareness of time

- Information are stored in historical data indicates signals of time-series behavior in the future
- Tree-based algorithms are unable to predict a trend, i.e., they do not extrapolate

Pay attention, you will not find here any mention of classical statistics and econometrics models. Only machine learning algorithms are described here. There is a reason why the amount of time you spend on selection of hyperparameters of econometrics models is not justified, when enough data for machine learning models is used.

General timeline of the time series project

In my experience, sequence analysis projects have been realized by a similar classical machine learning projects scenario. However, there are important differences in parts of exploratory data analysis and feature generation.

Below, I enumerated the general stages with their description in the time series analysis project.

Problem definition or target setting — the first stage of the project is the base of further execution. In this phase project manager, data science team or analyst meet a business owner that could represent an internal department or external client. In this meeting it is highly important to go down the following checklist:

- Define a whole scope of available data and documentation where sources and methods of data generation are described
- Define duration of the forecast horizon and its discreteness (hourly, daily, weekly, etc)
- Together with business owners and someone from the data science team define a key metric for future estimation of the model quality and its target value which those analysts will try to reach.

Exploratory data analysis (EDA) — that is a common stage for all data science projects where analysts face a new dataset and try to investigate that. Also, they try to evaluate its characteristics such as missing values, value distribution existence of stationary or trends in data, etc.

Data preprocessing and features generation — essential stage of any machine learning project where you have to be especially creative. On this stage data scientists generate predictors that will be the base of future models. The accuracy of the models and theirs ability of generalization depends on those features as well. Also, we should not skip normalization or standardization steps for the predictors of models. Roughly speaking, the entire set of features can be divided into 3 main groups:

- **Shifts** — that technique is used frequently in time series models. The main point is that historical data points include information about behavior of time steps in the future.
- **Math and statistics features** — the key idea is the same as in the previous one, but here we are going to calculate statistics such as

- **Domain knowledge** — traditionally it's the most useful and valuable information, but significantly complex for extraction. There can be external additional data or special statistics that are familiar only to domain experts. For instance, if you work with retail sales transaction data, information about marketing campaigns might be highly important for your models.

Training — that is a simple and straightforward process. However, you should always take into account cross-validation during training of your models. When you work with time-series data make sure you choose a relevant type of cross-validation. Finally, don't forget to look at model accuracy at the end of this stage.

Validation — give new information to your models, that it has never seen before, and estimate their value. That is a crash test for your models. If you observe significant differences between train and test's accuracy most likely they are overfitted or met completely new information in the test data.

The baseline — when you accomplished the first train — test iteration you get a baseline. After that, your goal is to jump over the baseline accuracy via tuning hyperparameters.

Selection — often you don't have enough time to set up hyperparameters for the entire set of your models, so after the first train-test iteration you are likely to select the best one and then try to improve its quality.

Shaping — an essential project stage where you work under a tuning of hyperparameters of your model with the aim to improve its prediction

GO live — so, in this step we have a model with enough level of accuracy and that works for business owners. That means it is time to put our model into a production environment. Before that step, your model is likely to work only in your Jupyter Notebook, but it's unavailable for users. With an aim to extract value from your work for the business to the maximum extent possible you should place the model in an environment with interface. That could be a part of anyone's software or you can also use Flask/Django or even integrate into business intelligence applications such as Tableau, QlikSense or Power BI.

Allow me to neglect to mention certain details and skip the first step of the project, despite it being the key factor of the project's success, and make focus on other ones.

In further explanation, I used the Kaggle dataset that consists of 600k transactional data collected during the period of 2014–2019, indicating date and time of sale, pharmaceutical drug brand name and sold quantity, exported from the Point-of-Sale system in the individual pharmacy. The selected group of drugs from the dataset is classified into the following Anatomical Therapeutic Chemical (ATC) Classification System categories.

Exploratory data analysis (EDA)

Preliminary data analysis is an essential part of any data science project. There are 3 main types of data exploration for working with time series:

- Data quality checking by the completeness and consistency -
- Visual analysis

Data quality and structure

```
df.info()
```

```
ass 'pandas.core.frame.DataFrame'>
geIndex: 2106 entries, 0 to 2105
a columns (total 13 columns):
um            2106 non-null object
AB            2106 non-null float64
AE            2106 non-null float64
BA            2106 non-null float64
BE            2106 non-null float64
B             2106 non-null float64
C             2106 non-null float64
              2106 non-null float64
              2106 non-null float64
r             2106 non-null int64
th            2106 non-null int64
r             2106 non-null int64
kday Name     2106 non-null object
pes: float64(8), int64(3), object(2)
ory usage: 214.0+ KB
```

So, there is perfectly rafinated data from Kaggle comprises two type of variables: float64, int64 and no one NaN/missing value.
There is strong likelihood that you never meet so clear data from the real world.

```
df.head()
```

latum	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Year	Month	Hour	Weekday Name
/2/2014	0.0	3.67	3.4	32.40	7.0	0.0	0.0	2.0	2014	1	248	Thursday
/3/2014	8.0	4.00	4.4	50.60	16.0	0.0	20.0	4.0	2014	1	276	Friday
/4/2014	2.0	1.00	6.5	61.85	10.0	0.0	9.0	1.0	2014	1	276	Saturday
/5/2014	4.0	3.00	7.0	41.10	8.0	0.0	3.0	0.0	2014	1	276	Sunday
/6/2014	5.0	1.00	4.5	21.70	16.0	2.0	6.0	2.0	2014	1	276	Monday

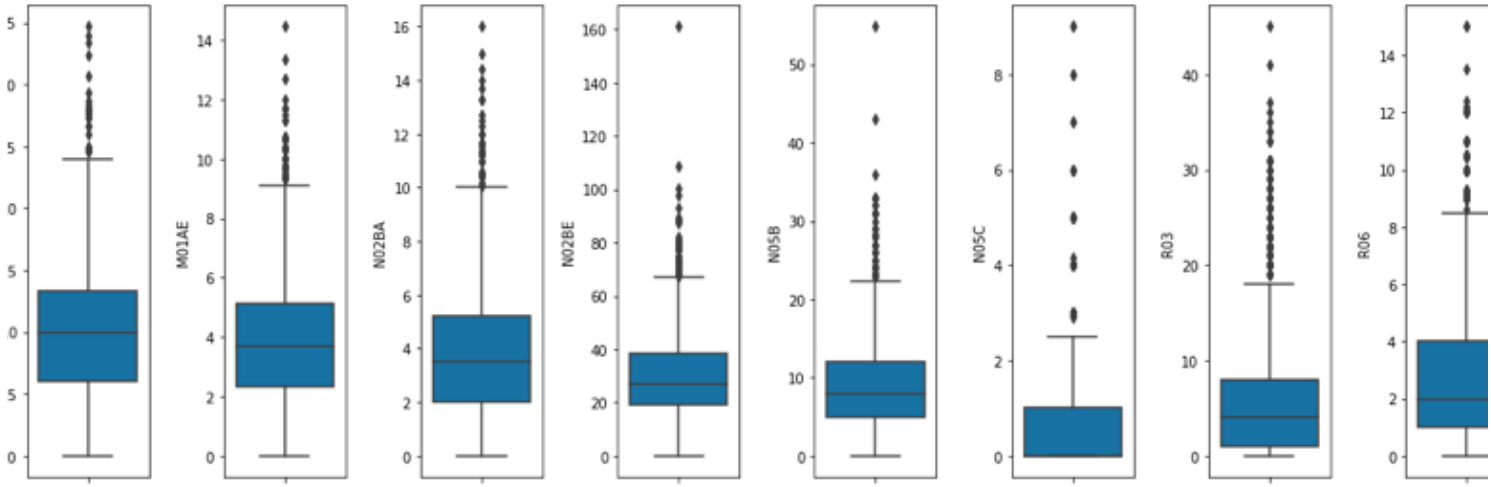
- M01AB — Anti-inflammatory and antirheumatic products, non-steroids, Acetic acid derivatives and related substances
- M01AE — Anti-inflammatory and antirheumatic products, non-steroids, Propionic acid derivatives
- N02BA — Other analgesics and antipyretics, Salicylic acid and derivatives
- N02BE — Other analgesics and antipyretics, Pyrazolones and Anilides
- N05B — Psycholeptics drugs, Anxiolytic drugs
- N05C — Psycholeptics drugs, Hypnotics and sedatives drugs
- R03 — Drugs for obstructive airway diseases
- R06 — Antihistamines for systemic use

```
df.describe()
```

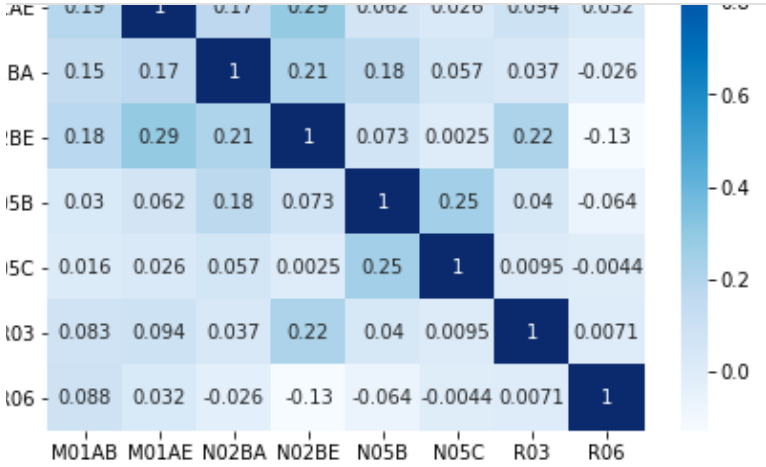
	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06
count	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000
mean	5.033683	3.895830	3.880441	29.917095	8.853627	0.593522	5.512262	2.900198
std	2.737579	2.133337	2.384010	15.590966	5.605605	1.092988	6.428736	2.415816
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3.000000	2.340000	2.000000	19.000000	5.000000	0.000000	1.000000	1.000000
50%	4.990000	3.670000	3.500000	26.900000	8.000000	0.000000	4.000000	2.000000
75%	6.670000	5.138000	5.200000	38.300000	12.000000	1.000000	8.000000	4.000000
max	17.340000	14.463000	16.000000	161.000000	54.833333	9.000000	45.000000	15.000000

at the range between 75th %tile and max values.
Let's figure it out how many outliers we have and where they are.

Visual analysis



Look at that! All of them includes outliers.
In accordance with statistical terms, outliers are defined as how values those out of 3 IQR.



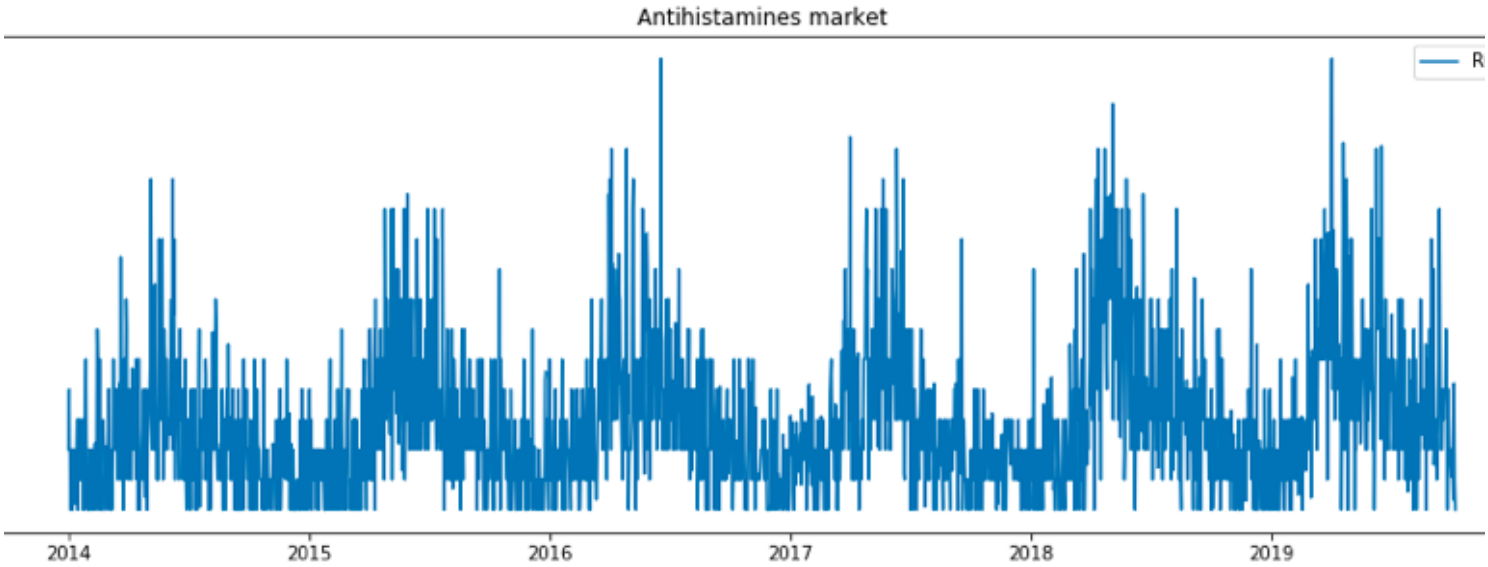
As part of exploratory data analysis sometimes incredibly useful to look at correlation matrix to future models tuning or further data exploration. On the graph above dark shades represents positive correlation while lighter shades negative. If you, for instance, would like to use linear regression, you have to consider removing correlated variables to improve forecast accuracy.
In our case, we can see a weak correlation between related categories (N05B, N05C, and N02BA, N02BE) and an interesting association between M01AE, N02BA codes. Defining the reasons for that could be the goal of further analysis.

Statistical analysis

In this part of research we want to estimate several essential statistical parameters:

> ***Trend*** *— a long-term movement in the one way in an ordered time-series.*

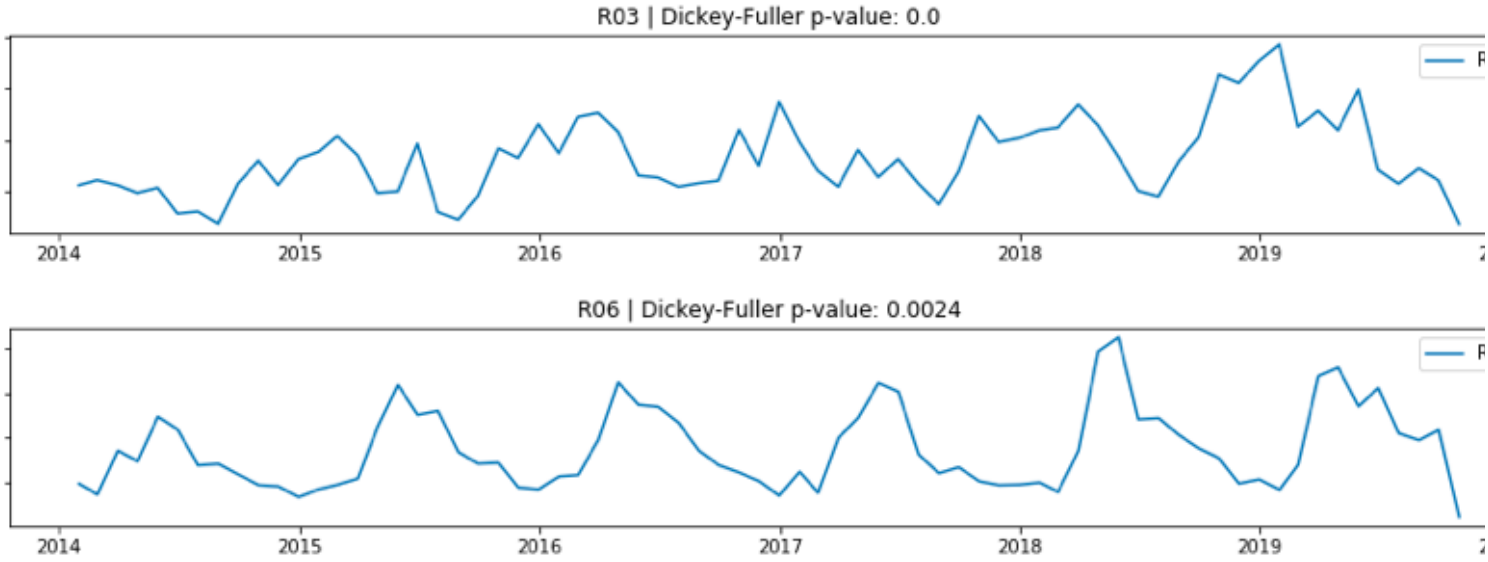
seasonality to remove the seasonal component and extract additional information about that with the aim to improve model performance. Be attentive and don't mix up seasonality with cycles.



Easy to see a clear seasonality of sales of antihistamines and analgesics products.
Also, on first sight, data looks quite stationary.
Let's determine is a time series is stationary or not by applying the Dickey-Fuller test

Stationarity *— means that the statistical properties of the time-series do not change over time. Accomplishing a research of that property is important because the future values could be easily predicted if statistical properties in the future will not differ significantly. The Dickey–Fuller test or Student's t-test can be used to observe stationarity. However, remember that trends and seasonality obviously makes data not stationary, but it's not clear*

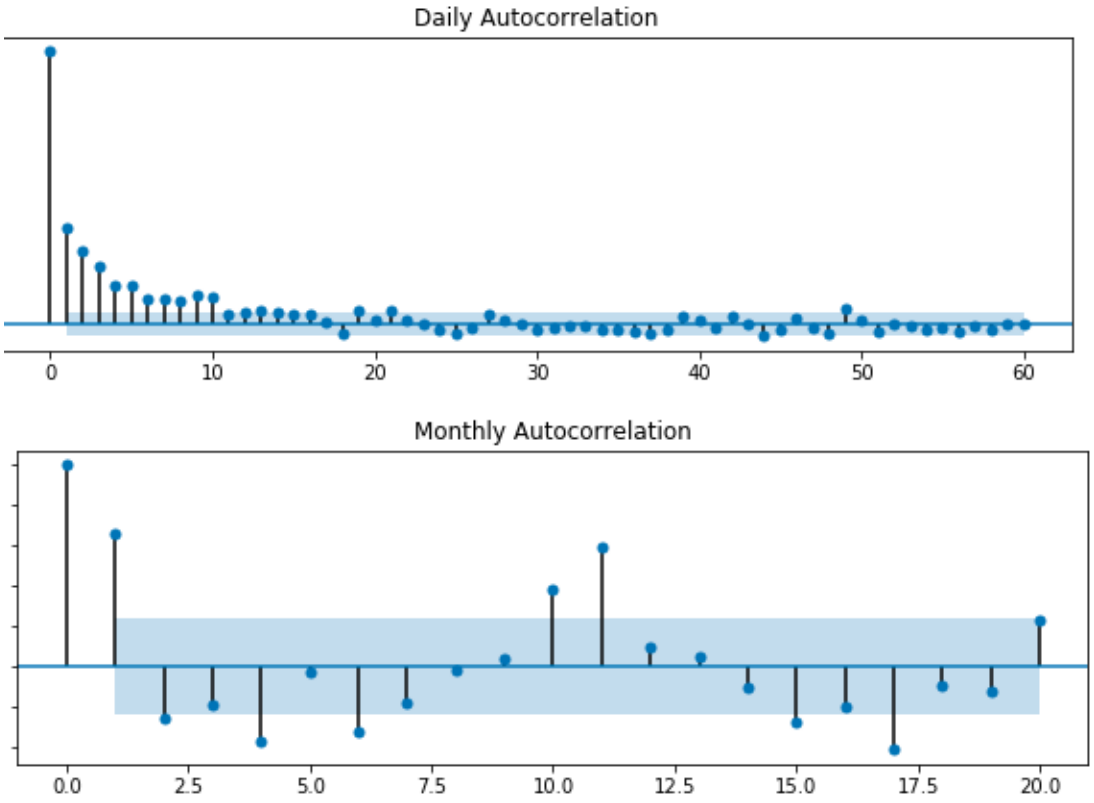
methods also benefit from the clearer signals in data.



How you can see we have changed a frequency of the data from daily to monthly
That made data is much more readable and we can see that all of the markets except one have a seasonality
At the same time, you can see how increased the number of sales during the 2015 year on psycholeptics market and sharply declined then
Sales of the airway disease market except seasonality includes an improving trend
By the way, you may have noticed in the title of the values of the chart of the p-value of the Dickey-Fuller test:

- when p-value > 0, and the data is not stationary.
- otherwise, p-value = 0, the null hypothesis is rejected, and the process is considered to be stationary.

important part of performing regression analysis because regression models assume that data has no autocorrelation, otherwise should not expect high quality of models.



To demonstrate how the .plot_pacf function works we took data with seasonality.

If autocorrelation values are close to 0, it is mean that values between lagged observations are not correlated with one another. At the same time, partial autocorrelations with values close to 1 or -1 demonstrate that there exist strong positive or negative correlations between the lagged observations of the time series.

The blue shaded regions indicate borders of the confidence intervals. If

statistically significant.

Pay your attention that daily values highly correlated with nearest observations, otherwise on the monthly chart you can notice a significant correlation with the same month last year.

Data preprocessing and feature generation

Once we accomplished an exploratory data analysis and detected data imperfection time to shape the dataset for the next steps.

There are 3 widely distributed methods to processed data more clearer for further forecasting:

- The stabilization of the dispersion by apply the box-cox transformation
- Differentiation or in other words moving to the pairwise differences of the neighboring observations
- Seasonal differentiation

TIPS

- *In case your dataset includes negative observations the box-cox transformation is unavailable, however you can add positive constant value for the entire dataset and apply the transformation.*
- *In most cases, the box-cox transformation does not strongly affect the quality of predictions, instead impacts on forecasts' horizon.*

- *Both differentiation methods can be used together, but if time series has pronounced seasonality better begin with the seasonal differentiation*

After we achieved a stationarity for the entire dataset it's time to move to the part of features generation. There we will create predictors that will be the base of the models.

By way of example I used multiple lagged values, categorical features characterizes period of the time and linear regression. Actually try to be more creative than me now and generate valuable features based on your domain knowledge that significantly improve forecasting accuracy.

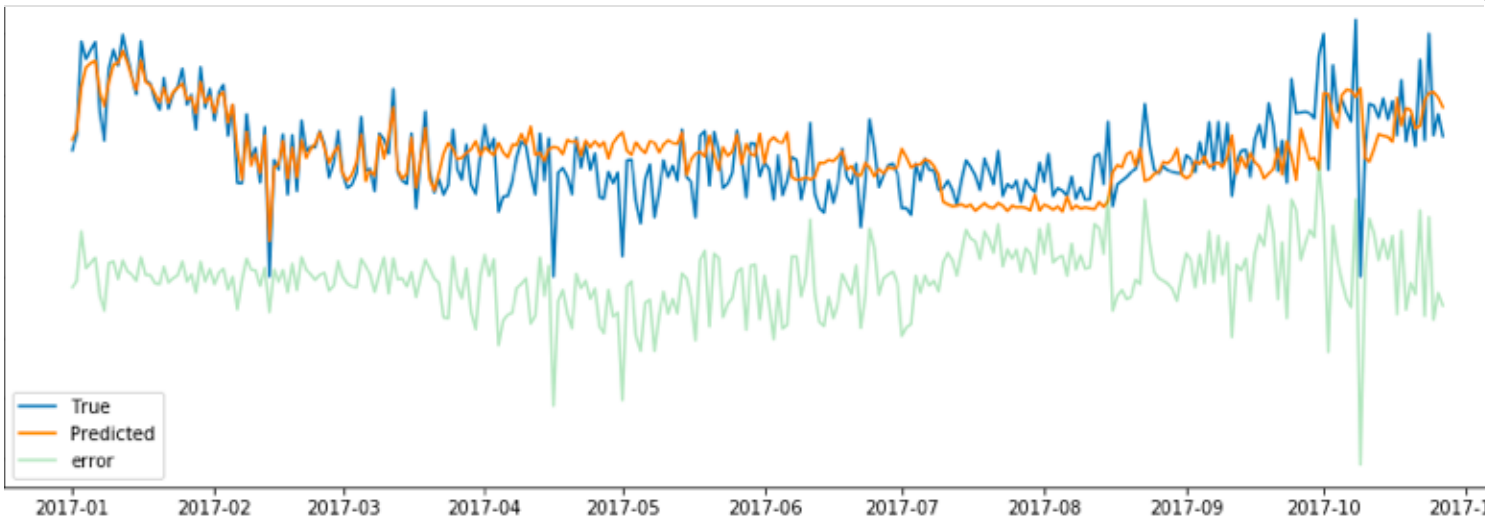
In the final phase of feature generation don't miss to add in your pipeline a step of data normalization.

Training

After fascinating exploratory analysis and following rigorous preprocessing we are ready to begin to train models.

As I mentioned above, setting up parameters of statistical models such as ARMA, ARIMA, SARIMA, etc is not worth it when enough observations are available and you are able to use machine learning algorithms.

So, let's train several models at the same time and see their quality via MAE and make emphasis on parts of the error line on the charts thouse will demonstrate where models fail.



Let me say a word about how we can monitor a forecast's bias. There are a lot of metrics allowing to track a performance of regression models, however the most popular are:

> ***MSE*** *(Mean Square Error) — is attentive to the multidirectional errors of forecasting models, but the same time sensitive to outliers and complicated to interpretation due to the lack of possibility of directly comparable to original errors.*

$$\mathrm{MSE} = \frac{\sum_{i=0}^{n} (y_i - \hat{y}_i)^2}{n}$$

values as Root-Mean Square Error, but it is still sensitive to the bad forecast points. After catching a few bad points even a good performance regression model will demonstrate a high value of error.

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

MAE *(Mean Absolute Error) — we are able to minimize an effect of bad points in the forecast due to calculating the absolute value of deviation.*

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$$

model across different datasets even with different ranges of observations. The main limitation of this metric is that it could not be used when the dataset includes zero values for any of the real instances.

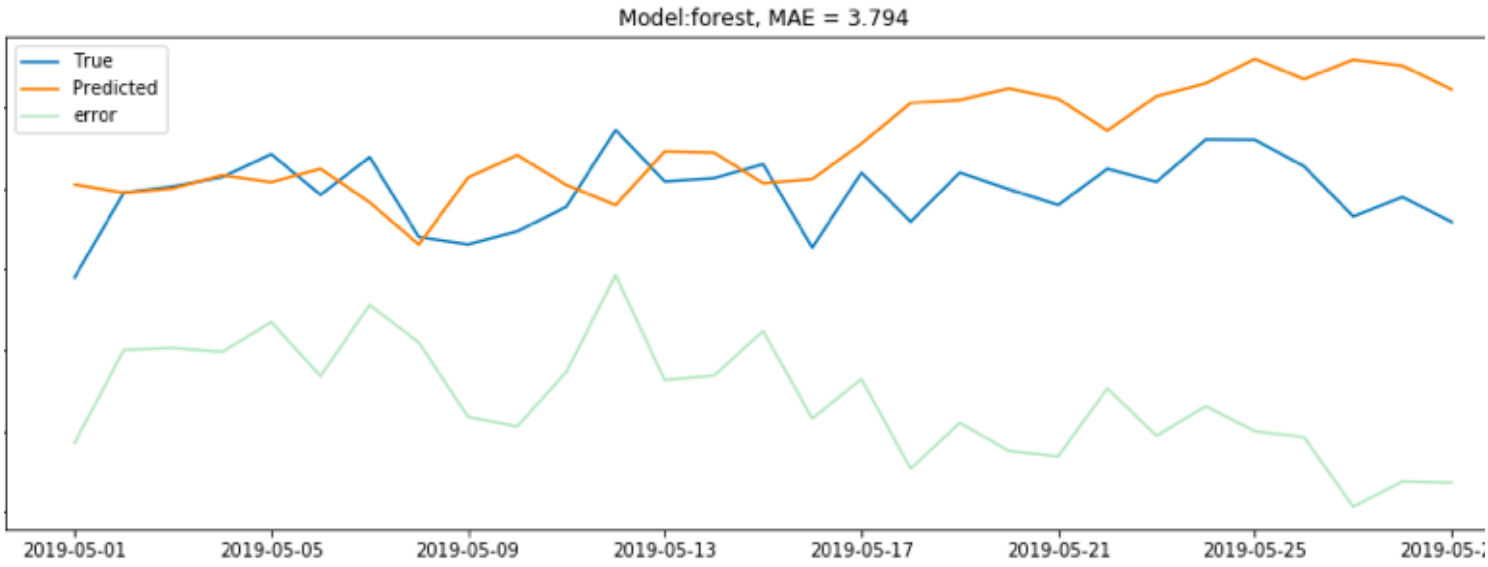
$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

R-squared *(Coefficient of Determination) — shows how well real points are explained by the model. The domain of definition can take values from 0 to 1 and the closer it to 1, the closer real observations to the model's line. Pay attention, the drawback of the coefficient is growthing it's value when additional variables are added to the model, even if they are not significant. For this reason, R-squared couldn't be used for comparing models with different amounts of variables.*

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Test

On the final step of the iteration we would like to compare predicted values and real instances that our models have never seen on the previous phases of the project. Follow the level of accuracy and places when the model fails. If you detected a significant difference between the level of chosen performance metric is calculated on the training and test datasets that would mean your model is overfitted. To fix it you should customize models parameters.



Conclusion

In the article has been described a general approach of creation forecasting model by using machine learning algorithms and examples of the first iteration have been shown. However, to reach a high quality of forecasting and extract the maximum value for business users, in your further iterations, I recommend focus on tuning of model