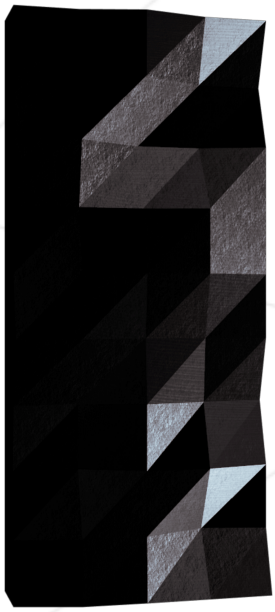

Uber

Taxi Demand Forecast

Summary: This project is an introduction to time series analysis: stationarity, exponential smoothing, SARIMA

Version: 1

Contents

Chapter I

Preamble

Does time exist? Is there absolute time (the idea supported by Isaac Newton) or is time just a relation between events and cannot be an independent measure (the idea supported by Leibniz)? Do the past and the future exist and, thus, could we hypothetically travel in time? Or they have no existence on their own and represent only changes that happened or will happen?

As you see, there are many debates about time in philosophical terms. But the same is with physics. The famous scientists applying modern theories of quantum mechanics, general and special relativity, and string theory try to answer the same questions. Using different equations they realized that in theory, we could travel in time if we had some loops in spacetime geometry (closed time-like curves). As Albert Einstein said: “In that case the distinction "earlier-later" is abandoned for world-points which lie far apart in a cosmological sense, and those paradoxes, regarding the direction of the causal connection, arise”.

Some paradoxes appear indeed. For example, the grandfather paradox: what would happen if one traveled back in time and killed one’s grandfather before the father was conceived? To solve it the Novikov self-consistency principle was proposed. The principle asserts that if an event exists that would cause a paradox or any "change" to the past whatsoever, then the probability of that event is zero. It would thus be impossible to create time paradoxes.

All in all, what we actually can say is that what we know for sure about time is literally nothing. We do not know it exists in absolute terms. We do not know if we can travel in time and what rules are applied. We do not know what devices can be created to perform time travel.

Anyway, it does not really matter whether time is an absolute term or it is just a relation to the present when we deal with some practical and day-to-day tasks. We know for sure that there will be some changes. If we can predict it by time traveling or by simpler methods, we will benefit from it.

Chapter II

Introduction

It is said that forecasting is like driving a car while looking in the rearview mirror. It is a beautiful analogy, but is it actually true? In some way yes. We do look at the past to predict the future: if something was absent in the past, it cannot be predicted in the future. But what can be a better analogy is that we train a driver who experiences many different situations on the road and who can make a good prediction of what may happen in the next few moments. She occasionally looks in the rearview mirror: not to learn something from the past, but to better see the present and to predict where all the actors around the car can be just in a moment.

Let us see what kind of drivers we might have to predict something in the future. What you did before was making predictions, but it was not about the state of an object in t+1 moment. It was a prediction about the state of an object in general, in static whether it was a classification problem or a regression one. When we talk about dynamic forecasting, we need different algorithms – algorithms from time series analysis (TSA).

Imagine that we would need to make a weather forecast (temperature) for tomorrow using only the available statistics from the past. We could use the average temperature based on the whole dataset. Another option would be to use the average temperature based on the past several years. It could be a better idea taking into account global warming as a trend component.

Another approach is to use the average temperature from the past several days or their weighted sum. It is called weighted moving average. The more sophisticated approach is to look in the past, but to give more weight to the recent data, rather than the old. It is called exponential smoothing. It does not work well though if there is a trend. It is better to use double exponential smoothing in order to deal with it. But if you have not only a trend but a seasonal component, than it is better to use triple exponential smoothing (Holt-Winters model). It applies exponential smoothing three times.

There is another class of possible solutions – ARIMA (autoregressive integrated moving average). Their core idea is that the main predictor of the series is the series itself. Your current data may have a good correlation with the data from t-1, t-2, t-3, ... period (lag). This is what is called autoregression. Integrated means that in order to make the series stationary, we have to differentiate it. A moving average model takes the lagged prediction errors as inputs. ARIMA models have three hyperparameters for each of these parts: p (AR), d (I), q (MA). There are some rules how to choose them.

There is another branch of ARIMA that is called SARIMA – seasonal ARIMA. It has four more parameters, but for seasonal component. P – seasonal autoregressive order, D

– Seasonal difference order, Q – seasonal moving average order, m – the number of time steps for a single seasonal period.

What else can you do? Well, you still may apply classical machine learning algorithms. What you need to do is to extract features as usual (weekday, holidays, time of the day, etc.) and to make predictions based on it solving a regression task.

And of course, since time series is a sequence, you may apply RNNs that you used before to work with texts.

A lot of things to try, right?

Chapter III

Rules of the project

The goal of this project is to give you a first approach to Time Series Analysis. You will try different algorithms to predict taxi demand in different city areas for the next week. The efficiency of any taxi business depends on it.

Chapter IV

Instructions

- This project will only be evaluated by humans. You are free to organize and name your files as you desire.
- Here and further we use Python 3 as the only correct version of Python.
- For training deep learning algorithms you can try Google Colab. It offers kernels (Runtime) with GPU for free that are faster than CPU for such tasks.
- The norm is not applied to this project. Nevertheless, you are asked to be clear and structured in the conception of your source code.
- Store the datasets in the subfolder data

Chapter V

Mandatory part

a. Task

In this project, you will work on demand forecasting. This task might be useful for many different industries: manufacturers, retailers, banks, etc. This time you will help a taxi company optimize their business. If you can predict that in certain areas tomorrow X taxis will be needed at that time, you can reduce arrival time and be better than the competitors.

In order to do this you will need to try different architectures:

1. naive averages,
2. moving averages,
3. exponential smoothing algorithms,
4. ARIMA models,
5. classical machine learning,
6. Deep Learning algorithms for sequences.

b. Dataset

You will work with the dataset of taxi drives. It contains data from 2019-04-01 to 2019-06-23 for 77 different areas of the city. And you will need to make predictions for the next 7 days in 15-minutes intervals (673): how many taxi orders expected in each area.

You can find the dataset in the project page :
1.taxi_pickups_area.csv
2.taxi_submission_file.csv

c. Implementation

You can work in Google Colab or Jupyter Notebooks on your computer.
You can use any library or any framework that you find convenient.
You should keep a research diary with all information about the used approaches and their metrics.

Naive averages

1. For each area make a prediction with the global average for this area for the next 673 intervals.

2. Calculate mean absolute error (MAE) for your own validation dataset.

Moving averages

1. For each area make a prediction with different moving averages for this area for the next 673 intervals.

2. Calculate mean absolute error (MAE) for the validation dataset.

Exponential smoothing

1. For each area make a prediction with 3 different exponential smoothing algorithms for this area for the next 673 intervals. Optimize the weights.

2. Calculate mean absolute error (MAE) for the validation dataset.

ARIMA

1. For each area make a prediction with the best SARIMA model according to AIC metrics for this area for the next 673 intervals.

2. Calculate mean absolute error (MAE) for the validation dataset.

d. Submission

Choose your best approach and save the prediction in the file for submission (taxi_submission_file.csv in attachment). You can use any heuristic above algorithms that you may find useful. For example, turning negative numbers into zeros, etc.

You need to achieve average MAE at most equal to 10.0 on the test dataset.

Your repository should contain one or several notebooks with your solutions.

Chapter VI

Bonus part

- Try to use machine learning algorithms for time series analysis. Make a prediction for each area for the next 673 intervals.
- Try to use deep learning algorithms for time series analysis. Make a prediction for each area for the next 673 intervals.
- Try to achieve an even better average MAE on the test dataset – 8.0.