Neural networks for time series analysis

Oleksandr Honchar | University of Verona

March 2018

Day 1: introduction

who I am?

Industry experience

Inma.AI, USA

machine learning developer

NDA “Gambling”, Ukraine-Russia

mathematician / developer

Mlvch, Russia

senior machine learning developer

ARVI Labs, Ukraine

senior machine learning developer

HPA Srl, Italy

CTO and co-founder

Sincere Tech, Ukraine

head of AI and shareholder

Teaching experience

3 technologies lectures, Ukraine

speaker and organizer

1 deep learning workshop, Ukraine

speaker and organizer

3 machine learning seminars, Italy

lecturer, UNIVR

5 AI applications webinars

speaker and/or organizer

AI blogger, Medium

> 40000 views monthly

what is this course about?

NOT about programming neural networks
NOT about mathematics
NOT about making a lot of easy money

HOW TO apply neural nets to time series
HOW TO design machine learning solutions
HOW TO test machine learning ideas for trading

Introduction to machine learning and time series analysis

Data preparation and feedforward neural networks

Convolutional and recurrent neural networks

Building a trading strategy and further applications

Literature:

Deep Learning: Ian Goodfellow, Yoshua Bengio, Aaron Courville
Neural Networks and Deep Learning: Michael Nielsen
Time series analysis with applications: Cryer, Jonathan D., Chan, Kung-Sik
Advanced algorithmic trading: Mike Halls-Moore

https://medium.com/@alexrachnog

time series applications

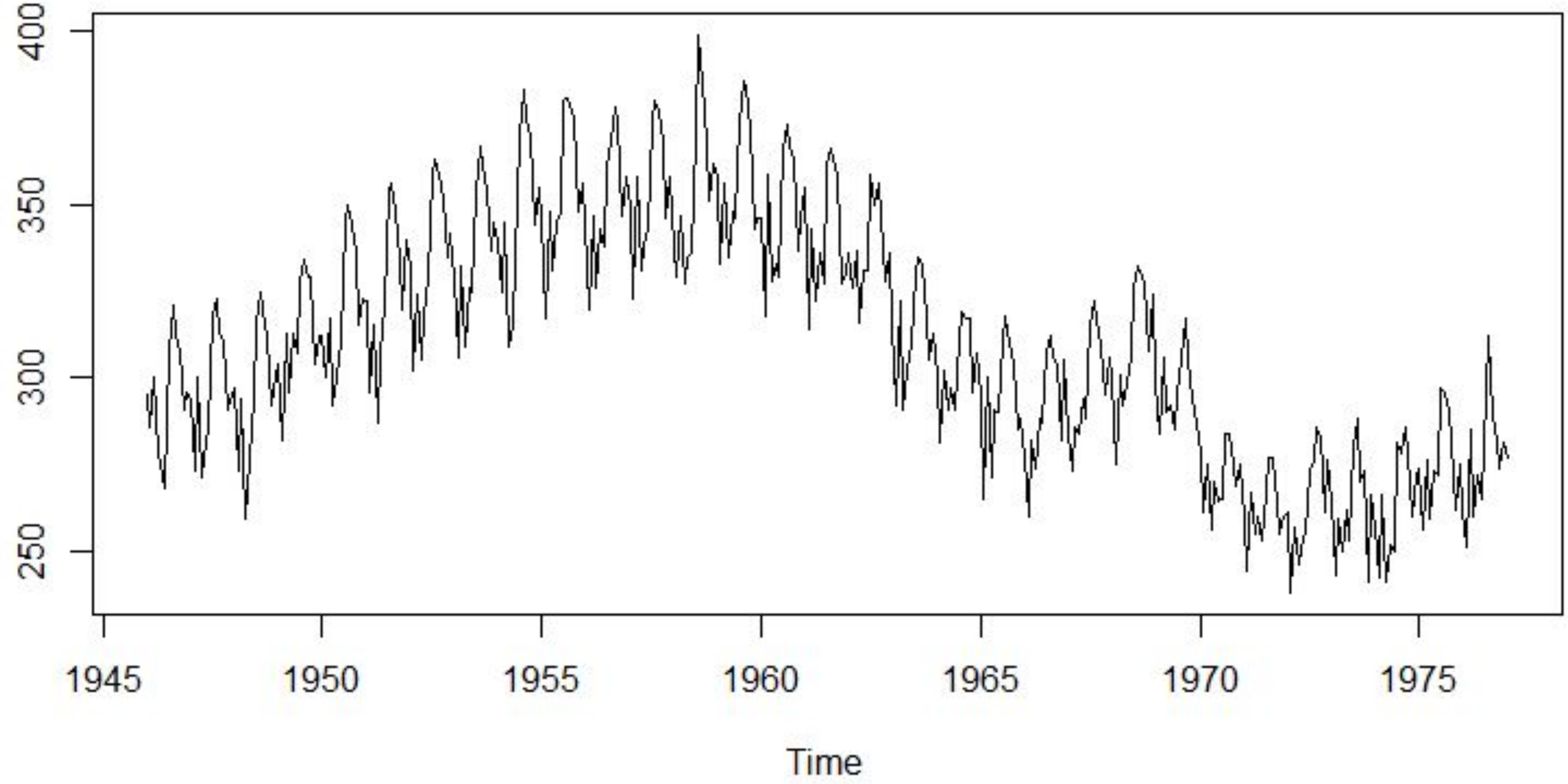
12:32
Walking
12AM
12PM
54
Walking Average
104 BPM
Today

amazon

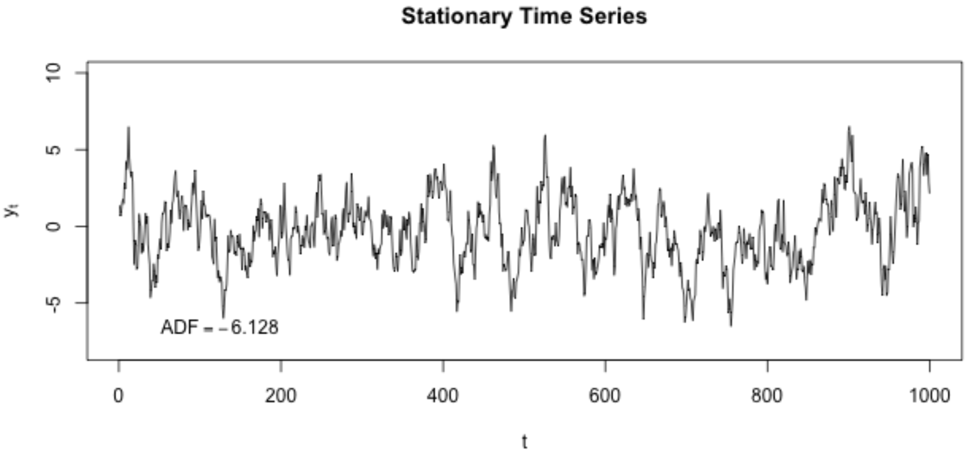
airbnb

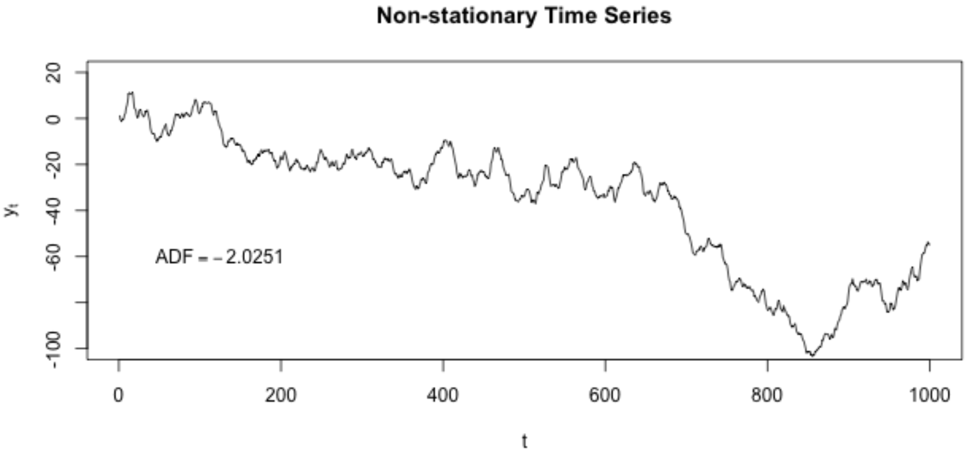
UBER

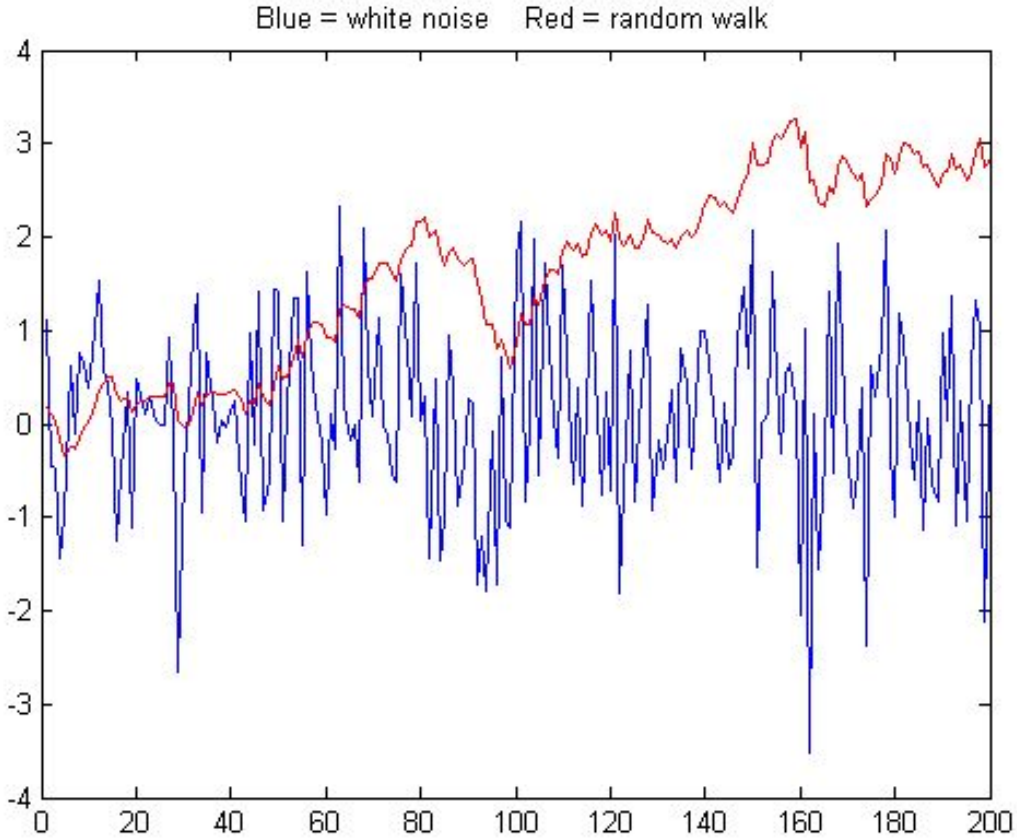
time series analysis


400
350
300
250
1945
1950
1955
1960
1965
1970
1975
Time

- trends (deterministic/stochastic)
- seasonal variation
- serial correlation


Stationary Time Series
10
5
0
-5
y_t
ADF = −6.128
0
200
400
600
800
1000
t


Non-stationary Time Series
20
0
-20
-40
-60
-100
y_t
ADF = −2.0251
0
200
400
600
800
1000
t


Blue = white noise   Red = random walk
4
3
2
1
0
-1
-2
-3
-4
0
20
40
60
80
100
120
140
160
180
200

- autoregressive model $$X_t = c + \sum_{i=1}^{p} a_i X_{t-i} + \varepsilon_t,$$

- moving average model $$X_t = \sum_{j=0}^{q} b_j \varepsilon_{t-j}$$

- ARIMA
- GARCH
- State space models
- Hidden Markov models

+ clean mathematical formulation
+ seems like it works

- way too linear for the real world
- don’t have deep memory
- don’t take into account single event impact
- don’t scale well for other tasks
- don’t scale well for other data sources

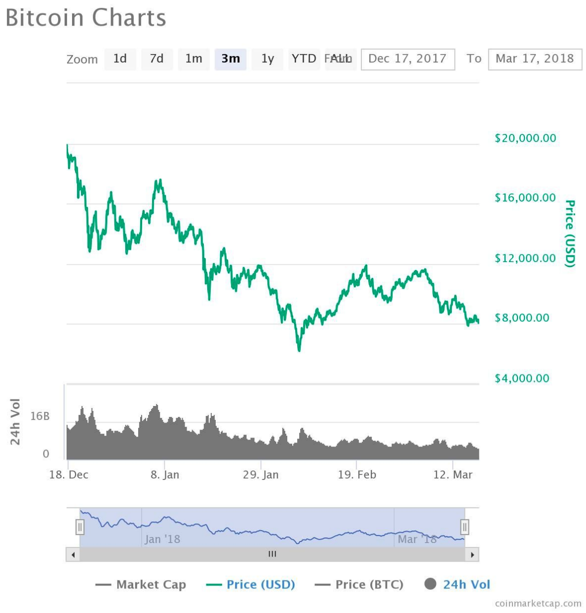
machine learning basics

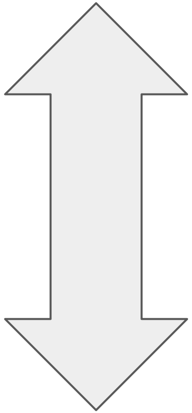
Program is said to learn from experience E with respect to some class of tasks T and performance measure P if performance improves

Mitchell, 1997

experience

{X} - historical prices windows
{Y} - historical changes after windows





tasks

Regression: predict a real value

Classification: predict a class instance

Tagging: predict a class instance for each element in the input vector

Regression: predict product sales volume next week

Classification: tell if 20 sec of ECG has arrhythmia or doesn’t

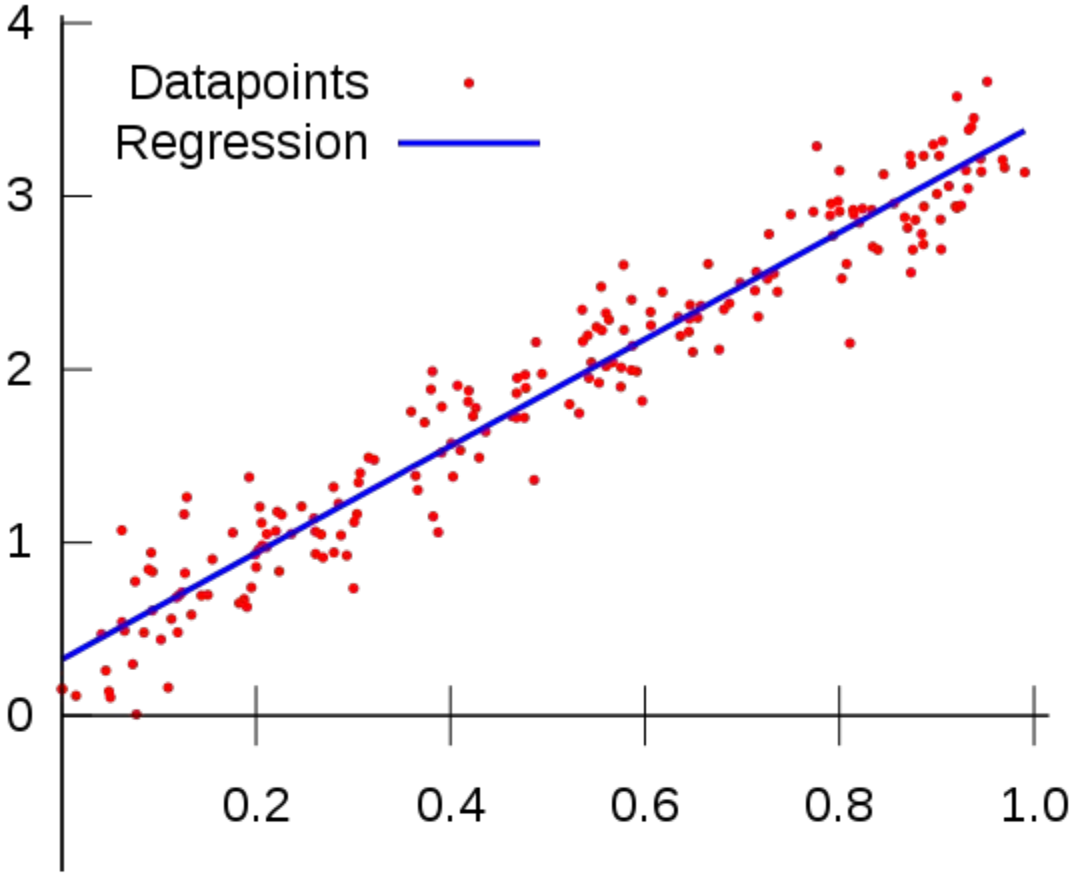
Tagging: tell what activities person was doing each minute based on one hour accelerometer data measure

mean average error	$\frac{1}{n}\sum_{t=1}^{n}\left\|x_t - \hat{x}_t\right\|$
mean squared error	$\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$
cross-entropy error	$-\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$

learn?

example:
polynomial regression

$$y = \beta_0 + \beta_1 x + \varepsilon,$$


Datapoints
Regression
4
3
2
1
0
0.2
0.4
0.6
0.8
1.0

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$

coding time!

https://github.com/Rachnog/education

rachnogstyle@gmail.com
facebook: rachnogstyle