1

Introduction

Forecasting 101

2

forecasting

… but also

nowcasting

backcasting

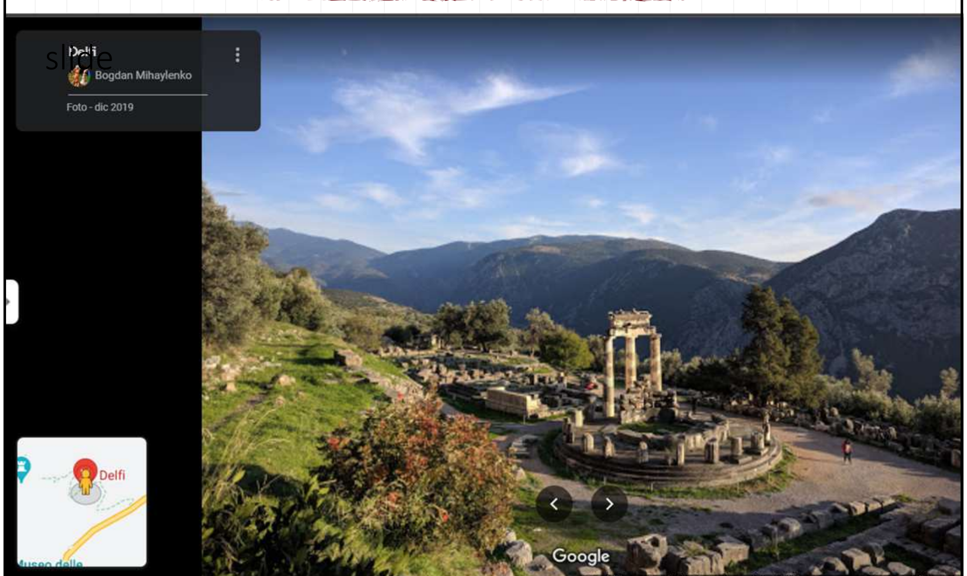
3

The buzz of divine communication

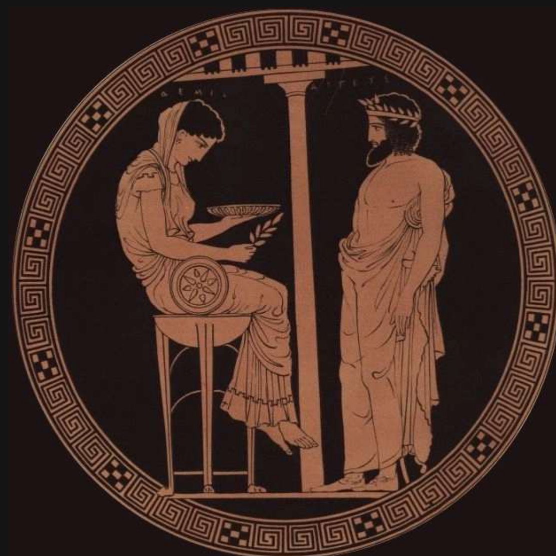
4

Sacrifice

Delphi (Δελφοί)



The pythia (Πυθία)

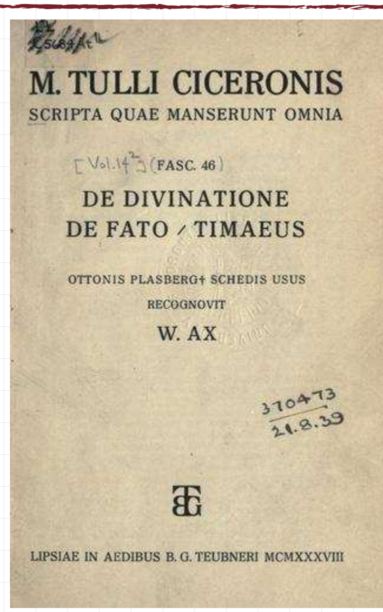


Haruspices, gut reading

[Spotify – Aruspice - song and lyrics]

Haruspices, bird flight

De divinatione



M. TULLI CICERONIS
SCRIPTA QUAE MANSERUNT OMNIA

[Vol.14^2] (FASC. 46)

DE DIVINATIONE
DE FATO / TIMAEUS

OTTONIS PLASBERG† SCHEDIS USUS
RECOGNOVIT
W. AX

370473
21.8.39

LIPSIAE IN AEDIBUS B. G. TEUBNERI MCMXXXVIII

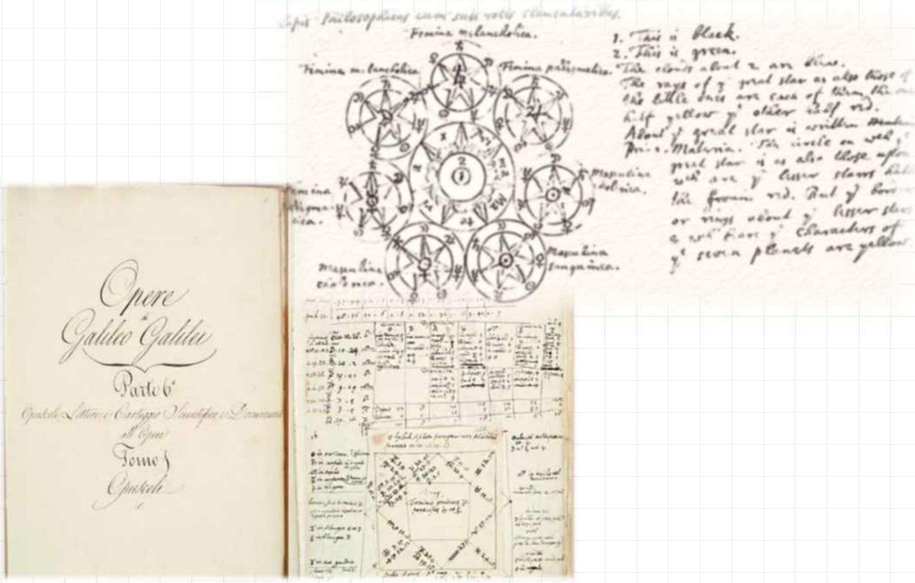
Dreams

Bible

Astrology

13

Galileo, Newton …



14

Statistics

The … “status” and means
“po…

In t… …tic
col…

For…
hu… …ary
use…

In t… …of
“st… …e
collection, summarization…

Kolmogorov laid the foun… …ed
his article Grundbegriffe … …)

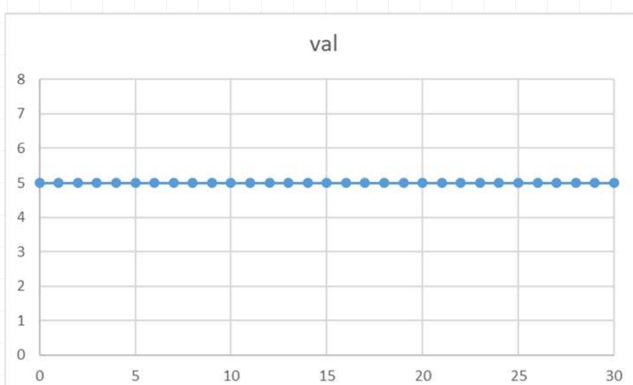
Vittorio Maniezzo - University of Bologna 15

15

Preprocessing

The easiest time series to forecast.

Data preprocessing tries to revert to this case.

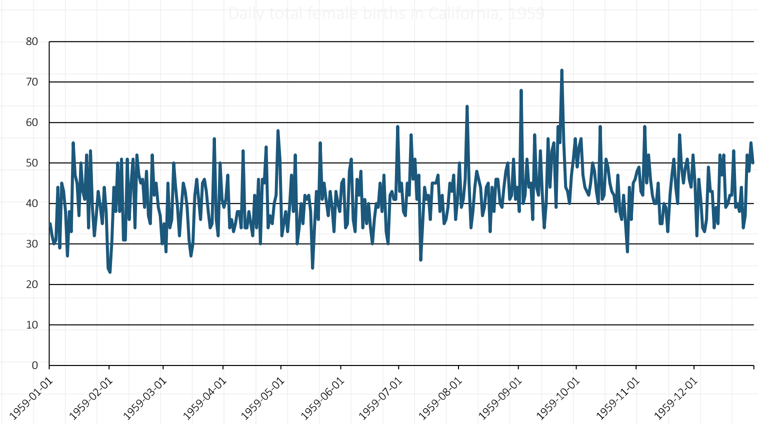


Vittorio Maniezzo - University of Bologna - 16

16

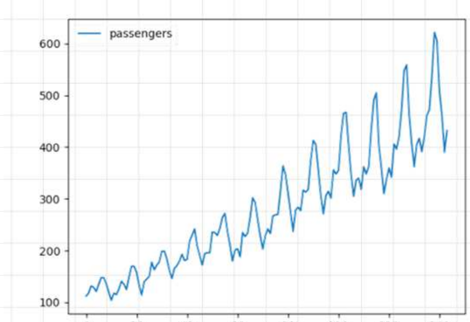
Stationary process

It has no trends (stationary in means and variance). No seasonality or cyclicity.



THE time series

International airline passengers: monthly totals in thousands. Jan 49 – Dec 60 (G.E.P. Box, G.M. Jenkins, 1976).



STL decomposition

STL stands for *Seasonal and Trend decomposition using Loess* (locally estimated scatterplot smoothing)

STL models a time series as a combination of

- Trend μ_t
- Seasonality s_t
- Random shock r_t.

The combination can be:

Additive : $y_t = \mu_t + s_t + r_t$

Multiplicative : $y_t = \mu_t \times s_t \times r_t$

Mixed : $y_t = \mu_t \times s_t + r_t$

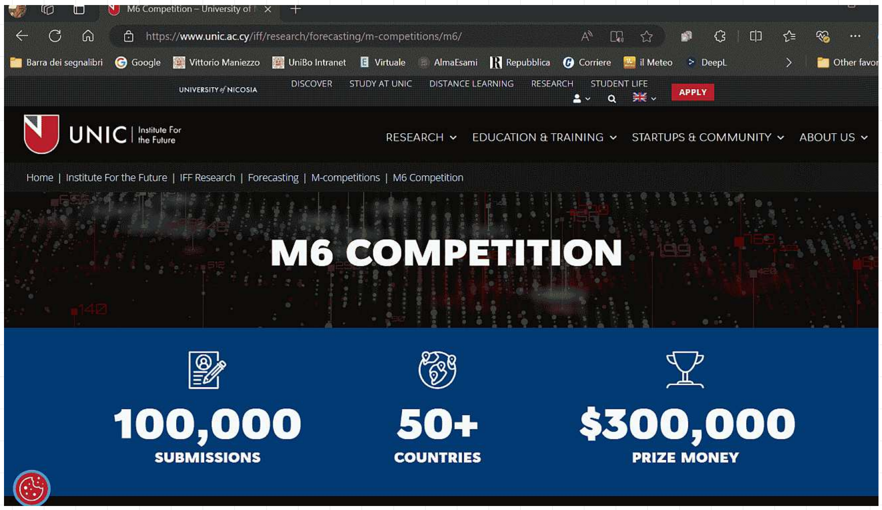
The M competitions

A series of open competitions to evaluate and compare the accuracy of different time series forecasting methods.

No.	name	Years	Num. of time series	Prizes
1	M Competition	1982	1001	
2	M2 Competition	1993	29	
3	M3 Competition	2000	3003	
4	M4 Competition	2020	100,000	$27,000
5	M5 Competition	2021 - 2022	Around 42,000 hierarchical timeseries provided by Walmart	$100,000
6	M6 Competition	2022 - 2024	Real time financial forecasting	$300,000

M6 site

https://www.unic.ac.cy/iff/research/forecasting/m-competitions/m6/



Vittorio Maniezzo - University of Bologna 21

21

... yours humbly

[Predictive Analytics for Real-time Auction Bidding Support: a Case on Fantasy Football | SpringerLink]

Operations Research Forum (2022) 3: 49
https://doi.org/10.1007/s43069-022-00160-w

ORIGINAL RESEARCH

Predictive Analytics for Real-time Auction Bidding Support: a Case on Fantasy Football

Vittorio Maniezzo[1] · Fabian Andres Aspee Encina[1]

Received: 8 April 2022 / Accepted: 4 August 2022 / Published online: 5 September 2022

Abstract
This work reports about an end-to-end business analytics experiment, applying predictive and prescriptive analytics to real-time bidding support for fantasy football draft auctions. Forecast methods are used to quantify the expected return of each investment alternative, while subgradient optimization is used to provide adaptive online recommendations on the allocation of scarce budget resources. A distributed front-end implementation of the prescriptive modules and the rankings of simulated leagues testify the viability of this architecture for actual support.

Keywords Predictive analytics · Prescriptive analytics · Lagrangian relaxation · Online decision support

Vittorio Maniezzo - University of Bologna 22

22

Fantasy sports

Fantasy sports, among which fantasy football, are more than 60 years old and permit fans to create an own league for their sport of liking.

A participant acts as owner and manager of her / his own team

- first bidding for the players to enter the team (*draft* phase)
- then deciding for each match who will be used among the acquired players (*lineup* phase).

A team does not need to be composed of players who actually play together in real-world events

The team effectiveness is computed after the actual performance of the players in real world sport events.

There exist agencies that compute for each player and for each event a score in "*fantasy points*". The success level of a fantasy team on a match day is given by the sum of the fantasy points gained by the used players in the actual real world corresponding matches.

Fantasy leagues

Each manager lists his fantasy team into a contest (a league), to be ranked against all other teams that entered the contest.

Leagues can be freely organized by groups of friends or whatever, but often there is an entry fee in order to win contest prizes.

Fantasy sports generate an economically relevant industry, moving hundreds of millions of dollars worldwide.

Entry fees for structured contests range from free to over $10,000 per entry, some contests pay out prizes up to one-million dollars.

Participants can invest money into decision making support tools, leading to an estimate of multimillion annual spending to purchase additional information and decision-making tools

The draft problem

The draft problem for Italian fantasy football requires, as other fantasy football leagues, to recruit 23 (25) players with the following roles:

- 3 goalkeepers (3),
- 7 defenders (8),
- 8 midfielders (8) among which at least 1 attacking midfielder and possibly at least 3 pure midfielders,
- 5 forwards (6).

The lineup at each match will consist of 11 players chosen among these 23, according to the allowed formations (4-4-2, 5-3-1, 4-3-2-1 and the like).

The draft problem

In the draft phase, to acquire players, the manager enters an auction with the other managers of his league, where player names will be called and bids will be placed on the currently auctioned player.

Each manager has a maximum budget B, usually 260 F.M. (FantaMillions), but also 300 or 500 (or whatever agreed upon) could be allowed.

MP formulation

I : index set of all auctioned players, x_i : binary decision variable associated with each player $i \in I$. An IP formulation is:

$$
\begin{array}{llll}
\text{(P)} & z_P = max & \sum_{i \in I} p_i x_i & (1) \\
& s.t. & \sum_{i \in G} x_i = 3 & (2) \\
& & \sum_{i \in \mathrm{D}} x_i = 7 & (3) \\
& & \sum_{i \in \mathrm{M}} x_i = 8 & (4) \\
& & \sum_{i \in \mathrm{T}} x_i \geq 1 & (5) \\
& & \sum_{i \in \mathrm{M} \setminus \mathrm{T}} x_i \geq 3 & (6) \\
& & \sum_{i \in \mathrm{F}} x_i = 5 & (7) \\
& & \sum_{i \in \mathrm{I}} \mathrm{c}_i x_i \leq \mathrm{B} & (8) \\
& & x_i \in \{0,1\}, \quad i \in I & (9)
\end{array}
$$

The o.f. maximizes the sum of the expected payoffs p_i of the acquired players, subject to constraints on the number of goalkeepers (2), the number of defenders (3) , the number of midfielders (4,6) , the number of attacking midfielders (5) and on the number of forwards (7), where $I = G \cup D \cup M \cup A$ and $T \subseteq M$. Constraint (8) imposes the budgetary restriction.

Sharpe's index

Purchasing a player is an investment, having the double objective of maximizing profit and minimizing risk.

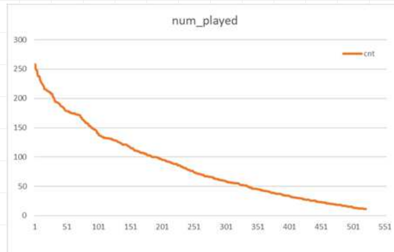
Finance literature contains different functions to consider them both, we adapted Sharpe's index (ratio, Sharpe, 1966) to our case. In finance, it is defined as the difference between the returns of the investment and the risk-free return, divided by the standard deviation of the investment (i.e., its volatility).

- risk-free return was the *minimum forecast profit* among players with the same role.
- volatility was measured as the average negative deviation, for each player.

Data acquisition

We downloaded the data series of all players for 7 seasons, amounting to 266 matches (*serie A* league is played by 20 teams).

https://www.gazzetta.it/calcio/fantanews/voti/serie-a-2020-20/giornata-19



Analytics:

1623 players
621 played in season 2020-21
586 players in the first half season
511 of these played also in the second half season (can be bought)
481 of these played at least two seasons (min series length)
161 of these played more than half of the matches in each half season (min data)

Proficiency forecast

We finally moved to forecasting future player performance (fantasy points).

The horizon of interest was half a season.

We downloaded all data at the end of the first half season 2020-2021, our training set, and forecast the second half season, which was unknown to us.

The draft problem would be sufficiently solved by forecasting the average profit for each player (its p_i). As we cover also the lineup problem, we tried however to get an as detailed as possible forecast.

It is notoriously implausible to reliably forecast detailed sports results, however, averages and maybe trends are more amenable to forecast.

We worked on series of rolling averages over three fantasy point values, the least window size that provided significant results.

We then moved on to forecasting trying different methods.

Autoregressive models AR(p)

In autoregressive models (of order p, AR(p)) the future value of a variable (added to a random component and to a constant) is assumed to be a linear combination of the last p observations

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t$$

Feasible only for stationary processes! It only models a linear dependency.

where:

y_t value at time t

ε_t random error at time t

φ_i (i = 1, … , p) model parameters

c constant

Moving Average models MA(q)

In moving average models (of order q, MA(q)) the future value of a variable is assumed to be equal to the average of the observations, μ, added to a linear combination of the last q errors:

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

Where:

μ average of the series

θ_j (j = 1, … , q) model parameters

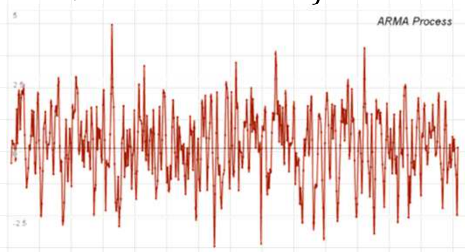
Random errors are usually assumed to have a normal distribution (mean 0, variance σ^2)

ARMA models

Autoregressive models (AR) and moving average models (MA) can be combined in a more general class, the ARMA models.

In mathematics, an $ARMA(p,q)$ model is:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$



SARIMA models

An extension, proposed by Box, Jenkins (1970), of ARIMA models which can be applied also to data with seasonality.

A seasonal diff is included.

Model identification can be complex.

Notationally, they are denoted by $ARIMA(p,d,q)(P,D,Q)_m$

where:

- p,d,q are ARIMA parameters referring to periods (ex., week),
- P,D,Q the same parameters referring to seasons (ex., trimester)
- m number of period in one season (ex., weeks in a trimester. Multiple seasonalities can be addressed).

SARIMAx

Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)

SARIMAX is an extension of SARIMA that also includes the modeling of exogenous variables.

Exogenous variables (also called covariates or predictors or externals) are parallel input sequences that have observations at the same time steps as the original series.

The primary series is referred to as endogenous data to contrast it from the exogenous sequence(s).

The observations for exogenous variables are included in the model directly at each time step and are ***not*** modeled in the same way as the primary endogenous sequence (as an AR, MA, etc. process).

Simple Exponential Smoothing

Simple Exponential Smoothing (SES), also *single exponential smoothing*, models the next time step as an exponentially weighted linear function of observations at prior time steps. It requires a single parameter, α, called the smoothing factor or smoothing coefficient.

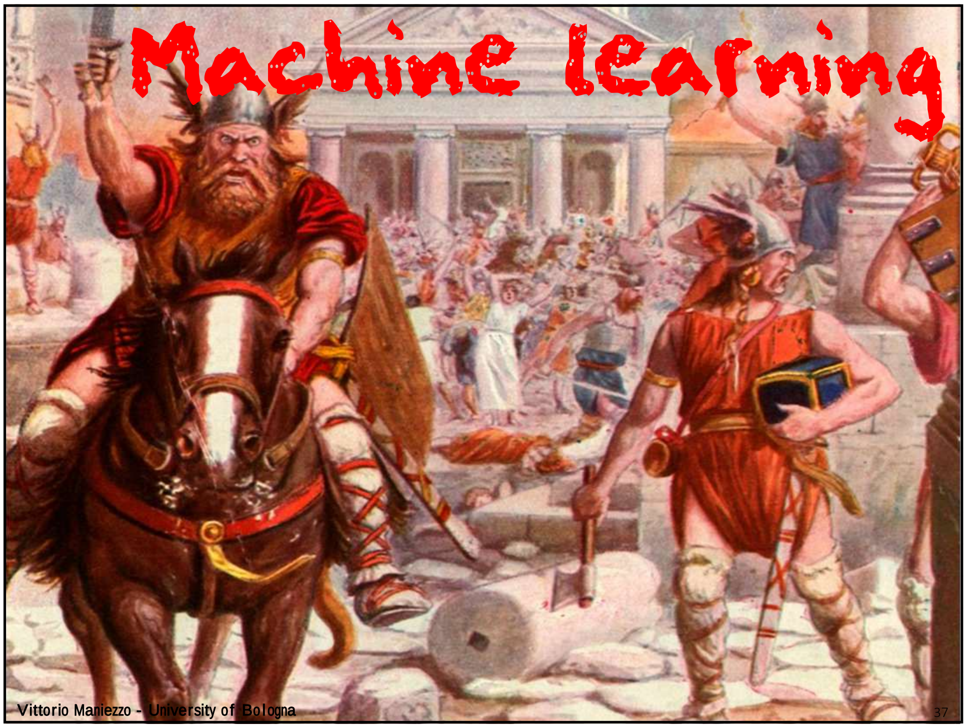
The equation is simply

$$y_{t+1} = \alpha y_t + (1-\alpha) y_{t-1}$$

That, when projected back in time, corresponds to

$$y_{t+1} = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \dots$$

The parameter controls the rate at which the influence of the observations at prior time steps decay exponentially. Values close to 1 mean that the model pays attention mainly to the most recent observations, whereas values close to 0 mean that more of the history is taken into account.



37

NN for forecast

Neural networks (NN) are very effective in identifying non linear models.

Even with input data deriving from complex, nonlinear processes, NN are able to combine linear and nonlinear models to get whichever accuracy level is requested (in prediction):

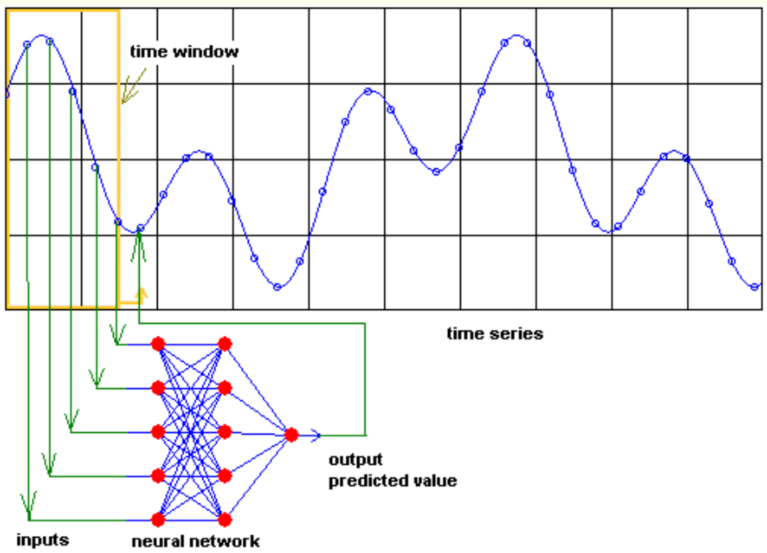
- NN can learn patterns in linear time series
- NN can learn patterns in nonlinear time series
- NN can generalize linear and nonlinear patterns

NN are nonparametric, they do not make hypotheses on noise distribution (gaussian or else).

NN identify a model from input data, essentially defining a model of the generating process.

38

Sliding window

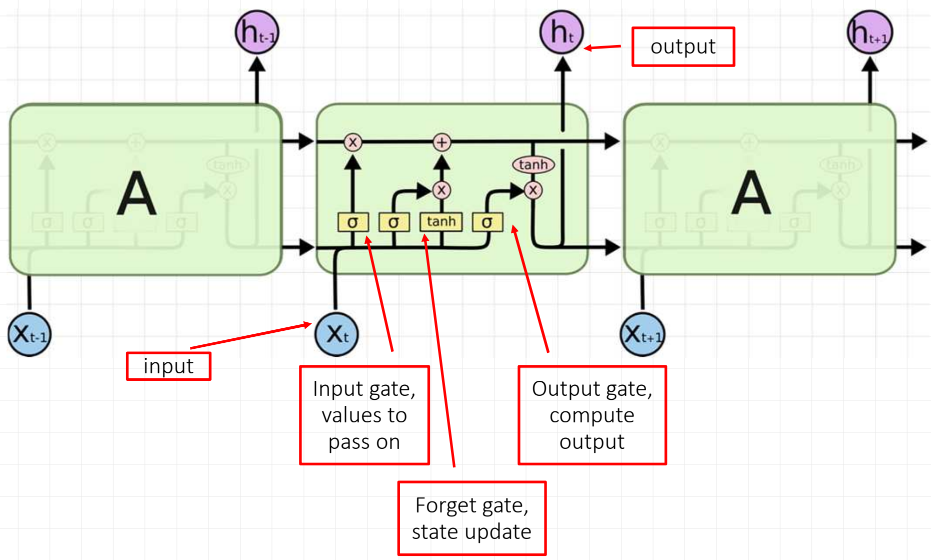


Recurrent NN

The most effective neural model for forecasting so far.

- They have at least one backward connection (*feedback loop*)
- They can keep activations even with no input → they have an *internal state*.
- Even with one single neuron, we can input a time series and get a series in the output.
- They can approximate any dynamical system.
- Very complex mathematical analysis.
- Very complex learning algorithms, hard to control. The most used one is Backpropagation Through Time (BTT).

LSTM

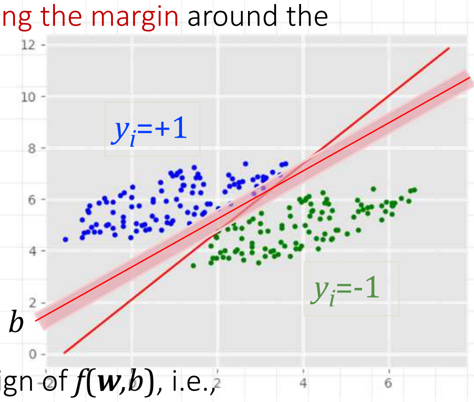


41

Support Vector Machines

Support Vector Machines are machine learning classifiers which, given labeled training data (supervised learning), compute an optimal hyperplane which separates (categorizes) the examples.

Optimality comes from maximizing the margin around the separating hyperplane.



This increases robustness in classification.

The separating hyperplane is determined by coefficients $(\boldsymbol{w},b)$:

$$f(\boldsymbol{x_i}) = w_1 x_{i1} + \cdots + w_n x_{in} + b$$

The classes are labelled by the sign of $f(\boldsymbol{w},b)$, i.e.,

$$y_i = sign(f(x_i)) \in \{-1,1\}$$

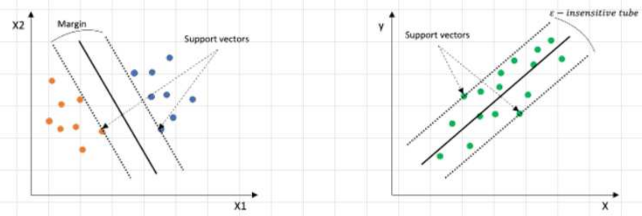
42

Support Vector Regression

SVR uses hyperplane and margin too, but with differences in their definitions.

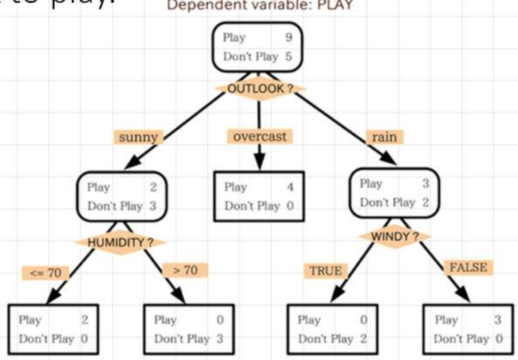
The margin, called the ε-insensitive tube, is the error tolerance of the model. This tube allows some deviation of the data points from the hyperplane.

The hyperplane is the best fit possible to the data that fall within the ϵ-insensitive tube.

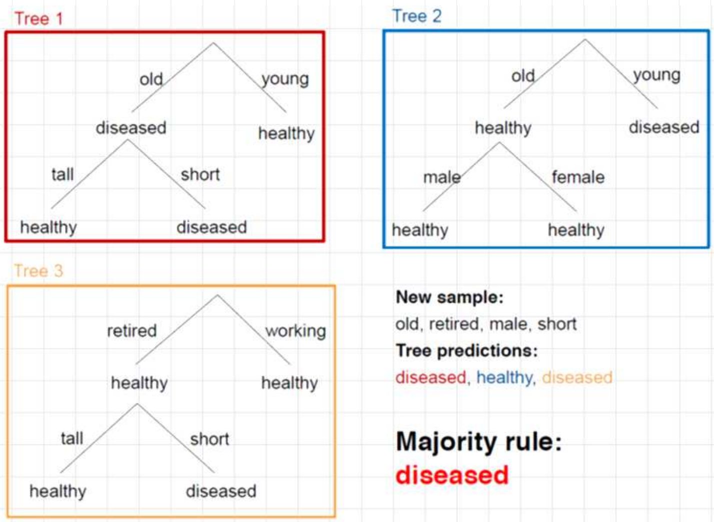


Decision tree

Example of a decision tree. In this case, the tree advises us, based upon weather conditions, whether to play ball or not. For example, if the outlook is sunny and the humidity is less than or equal to 70, then it's probably OK to play.



Tree ensembles

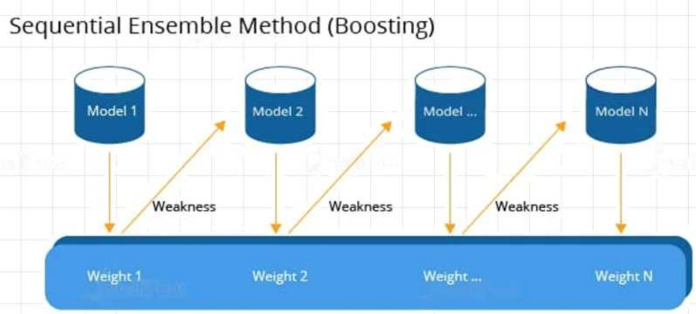


New sample:
old, retired, male, short
Tree predictions:
diseased, healthy, diseased

Majority rule:
diseased

Boosting (sequential)

In boosting we train the individual models in a sequential way. Each individual model learns from mistakes made by the previous model. AdaBoost, Gradient Boost etc. are different types of boosting methods.



Bagging (parallel)

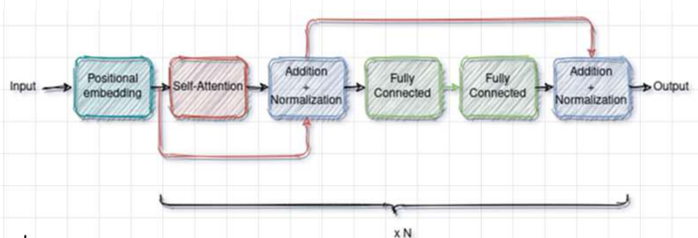
Bagging: decision trees depend onn the data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree and the predictions can be quite different.

Bagging creates N learners (decision trees) and produces N new training data sets by random sampling with replacement from the original set. The final prediction is the average of prediction from N decision trees.

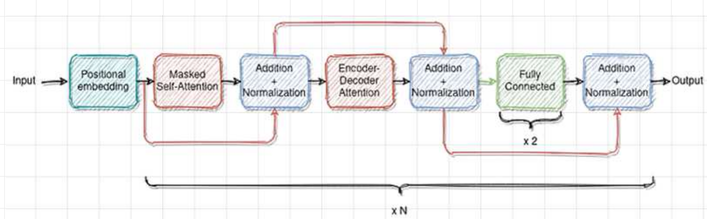
Random Forest is a bagging method. It also does a random selection of features rather than using all features to grow trees.

Transformers

Encoder



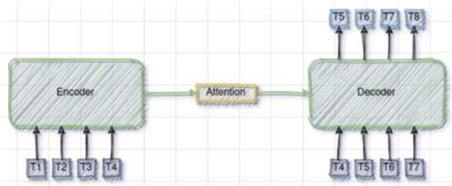
decoder



A transformer model

One can use an encoder-decoder transformer where the encoder takes as input the history of the time series while the decoder predicts the future values.

The decoder is linked with the encoder using an attention mechanism that learns to “attend” to the most useful part of the time series historical values before making a prediction.



Fantasy football: forecasts

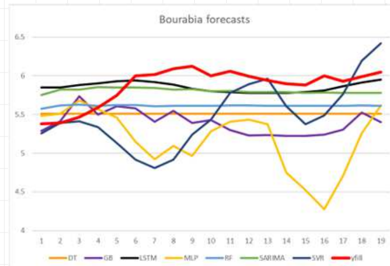
We applied and compared 6 univariate forecast methods:

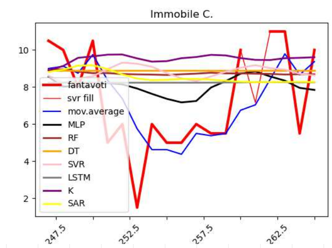
- SARIMA (seasonality allowed, turned out to be useless)
- MLP (rolling window, one hidden layer)
- SVR (the same as in filling)
- LSTM
- Decision Tree (sklearn)
- Random forest

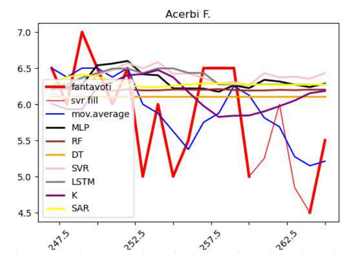
We recorded the results and eventually drafted a team based on the forecast of each of them.

Forecasts, whole half-season

Some examples. Drafts are made on these.



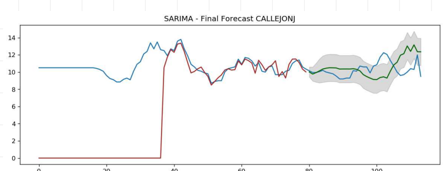


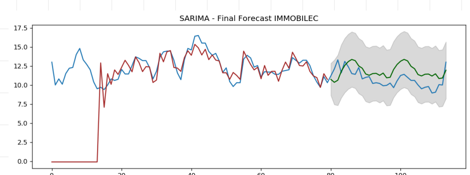


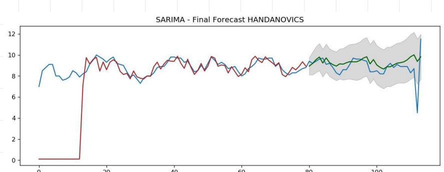
51

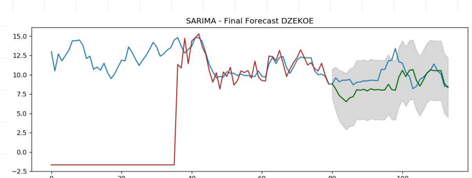
Forecasts, rolling

Rolling windows. Lineups are made on these









52

Deployment: Lagrangian relaxation

The complexity is due to the budgetary restriction. Associating a Lagrangian penalty λ with it, and relaxing the constraint, we have the formulation LP:

$$\text{(LP)} \quad z_{LP} = max \quad \sum_{i \in I} (p_i - \lambda c_i) x_i + \lambda B \qquad (10)$$

$$s.t. \quad (2), (3), (4), (5), (6), (7), (9)$$

$$\lambda \geq 0 \qquad (11)$$

Problem LP can be decomposed into four subproblems, defined on G, D, M and F, which can all be solved by inspection.

Having relaxed only one inequality constraint, the optimization of the Lagrangian dual can be expected to be quite fast and effective.

Auction support on the edge

LP could easily be solved by state of the art MIP solvers, but we want to support real time bidders, likely young people using a smartphone.

Currently, smartphones (mobile devices in general) have enough power to support advanced optimization, but the efficiency of mobile MIP solvers is still far from that of the desktop ones.

On the contrary, Lagrangian optimization in our case is much less computationally demanding and can be easily implemented using web languages such as javascript.

Deployment

Forecast results were persisted and are made available to mobile optimization clients, which implement the bidding support.

Each client runs its own optimization code, a subgradient optimization of the Lagrangian dual of problem LP paired with a Lagrangian heuristic.

Before the auction begins, a first optimization determines a target ideal team, as a function of the available budget.

As the auction proceeds, problem data can be updated and the optimizer restarted over the new values, adapting the incumbent Lagrangian multipliers and heuristic solution.

Reoptimizations run in a couple of seconds CPU time.

Lineup

For each match, 11 out of the 23 players must be selected.

The selected players must conform to one of the allowed formations: 3-4-3, 3-4-1-2, … , 4-4-1-1, 4-2-3-1 .

Lineups are defined:

- Selecting for each formation the player with the highest expected profits in the role,
- then selecting the formation that maximizes the total expected profit.

Nonconstant forecasts permit different lineups for different matches.

Comput. results: leagues

We computed a team for each forecast algorithm and for 4 budget levels.

We benchmarked the teams against one composed by the maximum cost players for each role, as of end Jan, 2021 (*not all of them available for drafting*, having been in Italy for too short), and against teams optimized projecting past profit means (no forecast, no sharpe's index).

Results: forecast leagues

Team	avg points	avg position
bestteam_400_RF	56.75	1.33
bestteam_400_SVR	54.75	1.79
best_400_MLP_NO	50.17	3.21
best_400_SAR_NO	47.83	3.96
bestteam_400_MLP	44.33	4.96
bestteam_400_SAR	42.17	5.75
best_400_RF_NO	36.92	7.29
bestteam_400_LSTM	34.25	8.38
best_400_SVR_NO	33.67	8.71
best_300_SVR_NO	31.25	10.17
bestteam_300_LSTM	30.50	10.46
best_300_MLP_NO	26.50	12.17
best_300_SAR_NO	24.00	12.83
bestteam_300_SVR	21.33	14.08
best_400_LSTM_NO	19.17	14.92
bestteam_300_RF	16.42	16.17
bestteam_300_SAR	14.67	16.83
best_300_RF_NO	11.50	18.04
bestteam_300_MLP	9.33	18.96
best_300_LSTM_NO	7.33	20.00

Results: expensive team

team	avg points	avg position
bestteam_SVR	11.42	1.21
expensiveTeam	10.58	1.79
bestteam_RF	7.33	3.25
bestteam_MLP	6.67	3.83
bestteam_SAR	5.33	5.08
bestteam_LSTM	4.50	5.83

For the record:
3-4-3
Handanovic S.
Gosens R.
Hakimi A.
Hernandez T.
Insigne L.
Berardi D.
Ilicic J.
Joao Pedro G.
Ronaldo C.
Lukaku R.
Muriel L.

(no statistical significance)