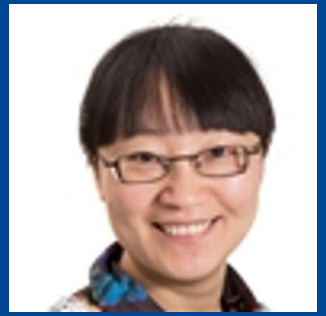
NTNU | Norwegian University of Science and Technology

Time Series Forecasting

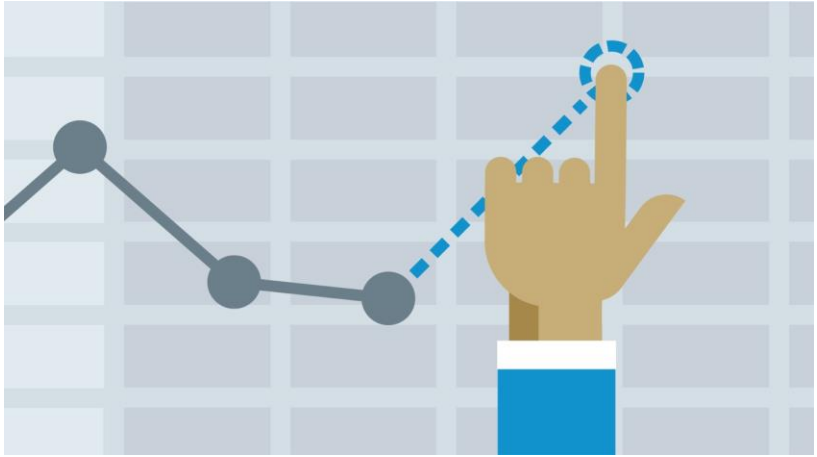
Course - TDT4173 - Machine Learning - Fall 2023

Liyuan Xing
Senior Machine Learning Engineer in ANEO
Adjunct Associate Professor at NAIL/IDI



Outline

- What is time series data and forecasting
- Forecasting steps and approaches
- Classic time series forecasting - Traditional
- Machine learning workflow – Modern
- Uncertainty and probability forecasting
- Forecasting competitions
- Trends and best practices from practitioners
- Automated machine learning
- Reading list

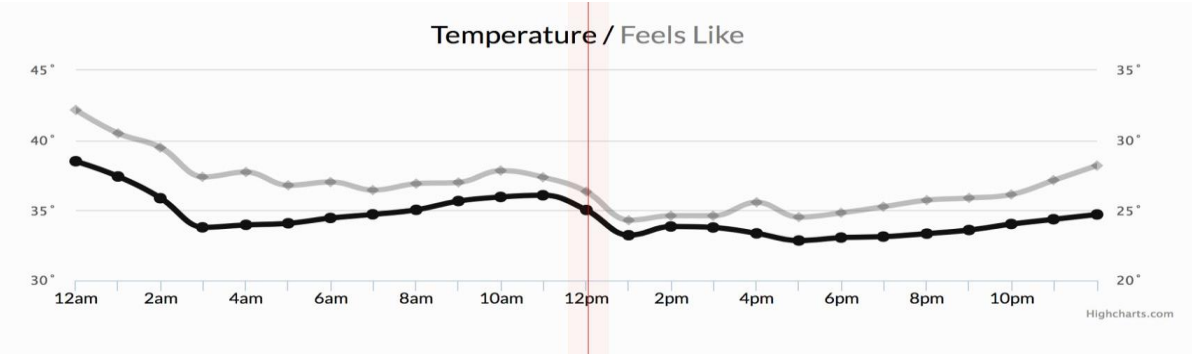


Time series data - chronological order

A sequence of data points indexed in time order

A collection of observations obtained through repeated measurements over time

One of the dimensions would always be time



time	ActivePower	Ambient-Temperature	Nacelle-Temperature
2023-08-06 07:10:00	-3.26093	9.93000	23.00000
2023-08-06 07:20:00	-4.08909	9.92167	23.00000
2023-08-06 07:30:00	-3.40830	9.70601	23.00000
2023-08-06 07:40:00	-5.95700	10.03299	23.00000
2023-08-06 07:50:00	-7.48506	10.35333	23.00000
2023-08-06 08:00:00	-4.84470	10.15434	23.00000
2023-08-06 08:10:00	-6.62998	9.69500	23.00000
2023-08-06 08:20:00	-4.76331	9.93834	23.00000
2023-08-06 08:30:00	-5.70828	10.32667	23.00000
2023-08-06 08:40:00	-4.86726	10.39334	23.00000
2023-08-06 08:50:00	-5.00134	10.76066	23.08833
2023-08-06 09:00:00	-7.88312	10.51000	23.25566
2023-08-06 09:10:00	-5.20199	11.00000	23.89667
2023-08-06 09:20:00	-6.42336	11.07833	23.99833
2023-08-06 09:30:00	-4.40971	11.24066	24.00000
2023-08-06 09:40:00	-6.11000	11.59767	24.00000
2023-08-06 09:50:00	-3.95669	11.00000	24.00000
2023-08-06 10:00:00	-5.06499	11.00000	24.00000
2023-08-06 10:10:00	-5.71667	10.88167	24.00000
2023-08-06 10:20:00	-5.08827	10.09434	24.00000
2023-08-06 10:30:00	-3.67998	10.21000	24.00000
2023-08-06 10:40:00	-3.36169	10.18333	24.00000
2023-08-06 10:50:00	-8.55030	10.77233	24.00000
2023-08-06 11:00:00	-5.69968	11.00000	24.00000

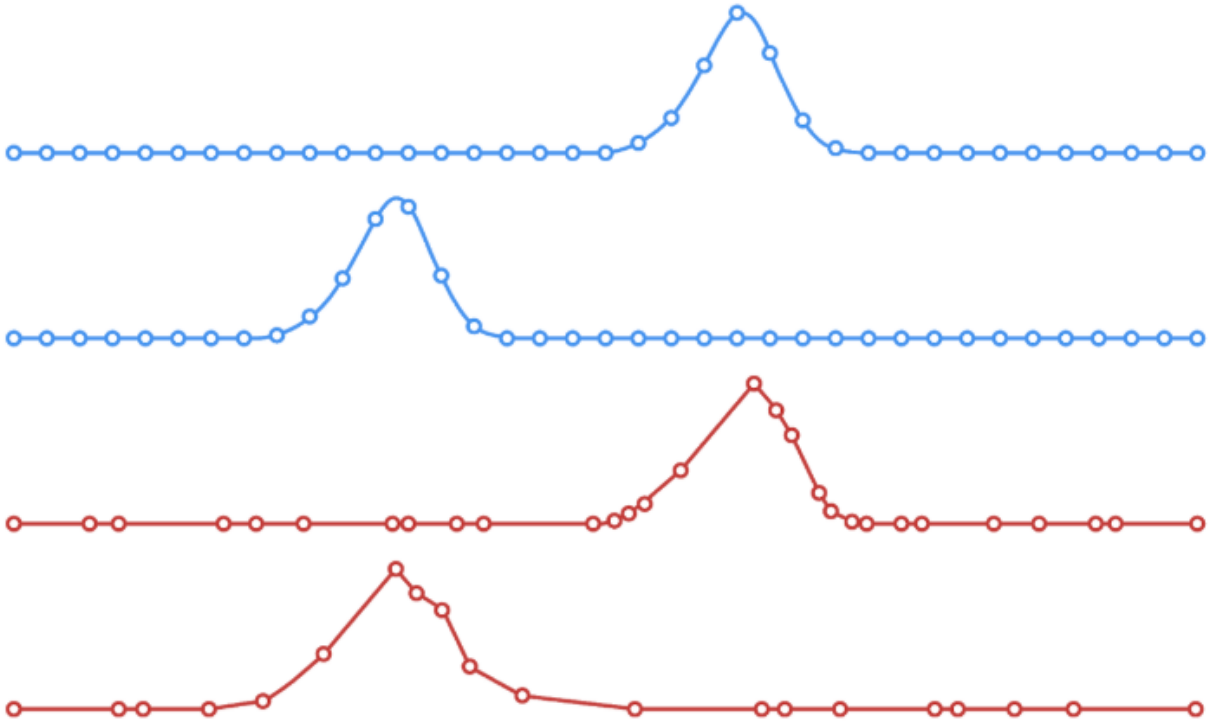
Time series data - time intervals

Metrics (Regular)

Measurements gathered at regular time intervals

Events (Irregular)

Measurements gathered at irregular time intervals



Time series examples - everywhere

Meteorology: weather variables, like temperature, pressure, wind

Economy and finance: economic factors (GNP), financial indexes, exchange rate, spread

Marketing: activity of business, sales, customers' experience, budget estimation

Industry: electric load, power consumption, voltage, sensors

Biomedicine: physiological signals (EEG), heart-rate, patient temperature

Internet of Things: sensor data prediction, data losses, device replacement

Performance and health monitoring: App performance, button clicks, server metrics, network data

Social sciences: unemployment rates, birth rates, population, migration data, political indicators

Epidemiology: disease rates, mortality rates, mosquito populations

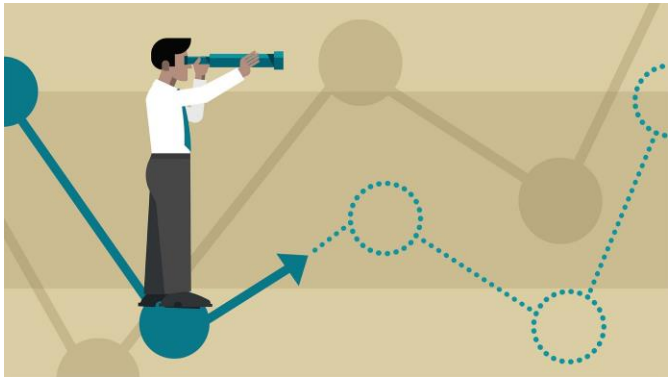
Time series forecasting - to predict the future

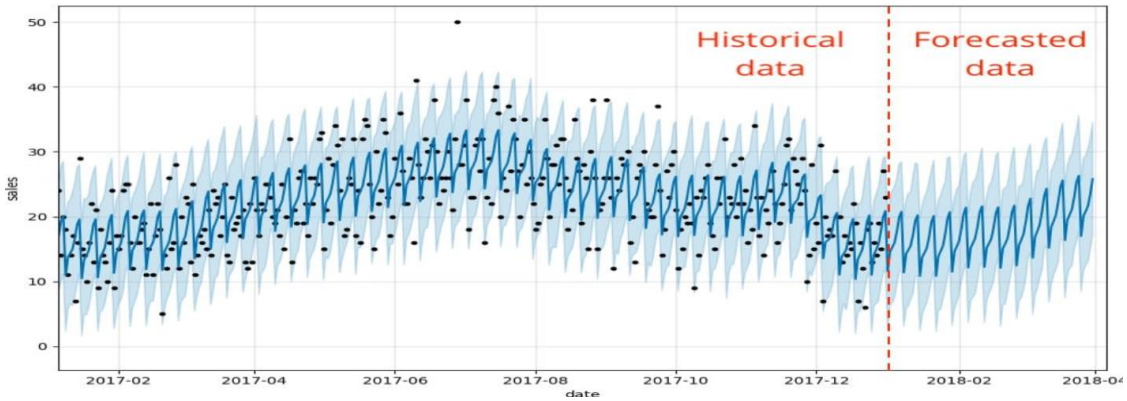
Stand in the present and forecast the future based on the past time series data

Forecasting one step or/and multi steps ahead

Uncertainty is often as important as the forecast itself

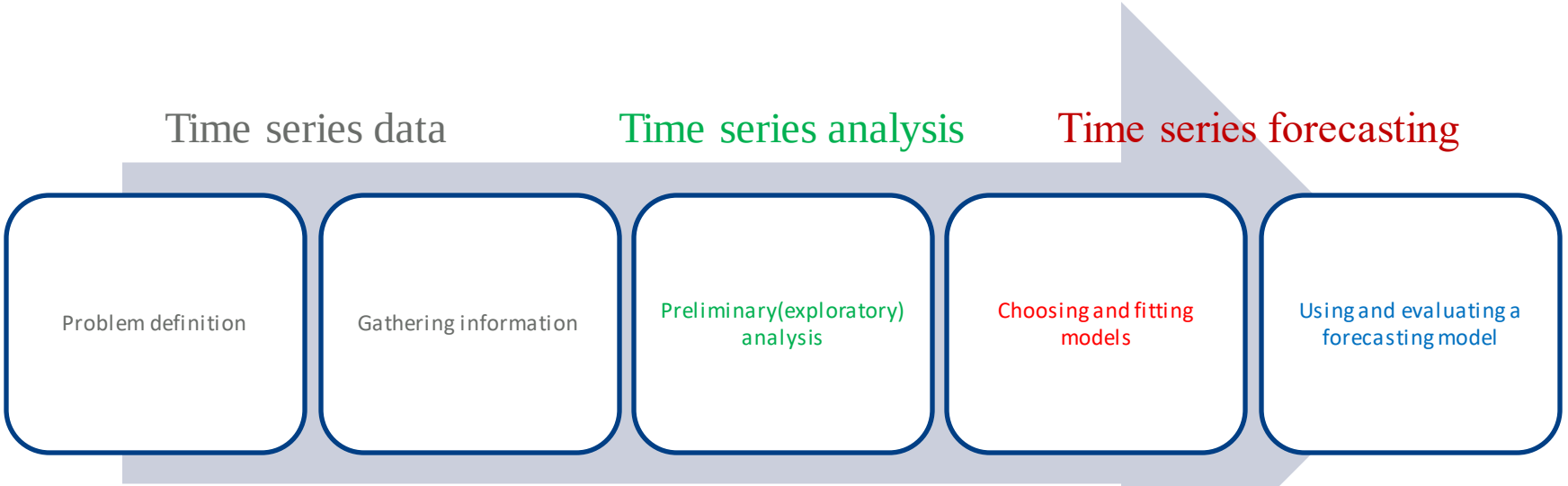
- Point forecasting: predict the mean/expectation
- Probabilistic forecasting: represent the prediction distribution





t-T t t+h

Basic steps and approaches



Get the data
- Historical values
- Impacting factors

Know the data and preprocessing
- Trend, seasonal, cyclic, (auto)correlation
- Cleaning
- Transformation to stationary
- Normalization

Build the model
- Domain based models
  - physical, econometric
- Statistic models
  - ETS, ARIMA
- Machine learning models
  - Surpervised learning, local learning, deep learning, ensemble learning

Forecasting and evaluation
- Error measures
- (Nested) cross validation

Time series analysis vs forecasting

Time series analysis is all about understanding the dataset

- Extract meaningful statistics and characteristics of the data
- Identify trends, cycles, and seasonal variances to aid in the forecasting of a future event
- It is super important in traditional time series forecasting, which is used to choose suitable statistical models with fixed formula
- While it is weakened in machine learning model that is able to learn the relationships and patterns from the data itself

Time series forecasting is all about predicting the future

- Learn a model: based on previously observed data to build a model
- Make predictions: use the learned model on new data to predict future values
- Statistical models vs machine learning models

Classic time series forecasting - Traditional

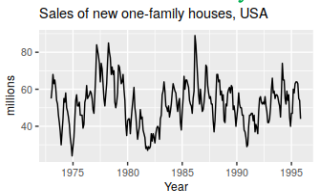
Explore the data characteristics

Choose suitable models

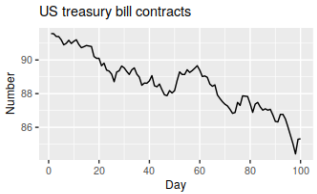
Estimate model parameters

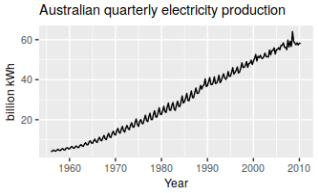
Select best model

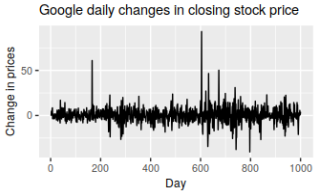
Seasonal + Cyclic



Trend







Trend + Seasonal

Stationary

ETS models

ADDITIVE ERROR MODELS

Trend	Seasonal: N	Seasonal: A	Seasonal: M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

MULTIPLICATIVE ERROR MODELS

Trend	Seasonal: N	Seasonal: A	Seasonal: M
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A_d	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

ARIMA models

$$\underbrace{(1-\phi_1 B - \cdots - \phi_p B^p)}_{\uparrow\ \mathrm{AR}(p)} \underbrace{(1-B)^d y_t}_{\uparrow\ d\text{ differences}} = c + \underbrace{(1+\theta_1 B + \cdots + \theta_q B^q)}_{\uparrow\ \mathrm{MA}(q)} \varepsilon_t \quad \underbrace{(p,d,q)}_{\uparrow\ \text{Non-seasonal part of the model}} \quad \underbrace{(P,D,Q)_m}_{\uparrow\ \text{Seasonal part of the model}}$$

Using information criteria

ETS:

$$\mathrm{AIC} = -2\log(L) + 2k$$

$$\mathrm{AIC_c} = \mathrm{AIC} + \frac{k(k+1)}{T-k-1}$$

$$\mathrm{BIC} = \mathrm{AIC} + k[\log(T) - 2]$$

ARIMA:

$$\mathrm{AIC} = -2\log(L) + 2(p+q+k+1)$$

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}$$

$$\mathrm{BIC} = \mathrm{AIC} + [\log(T) - 2](p+q+k+1)$$

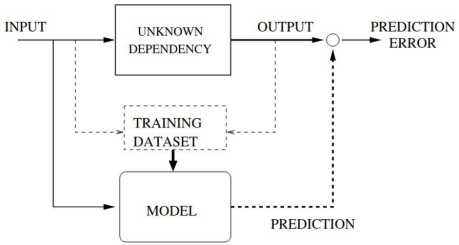
Machine learning workflow - Morden

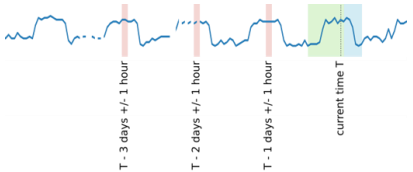
Data preprocessing: cleaning, transformation, normalization

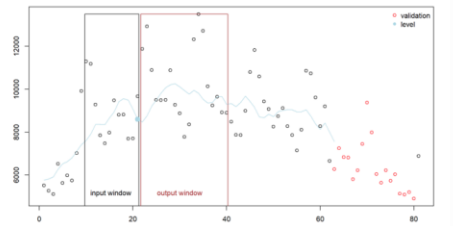
Feature engineering: construction, selection

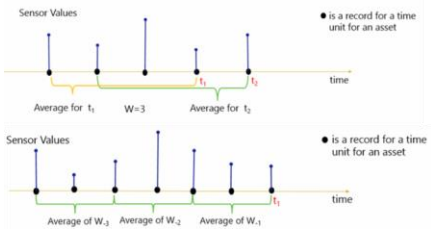
Model engineering: construction, selection, hyperparameter tuning

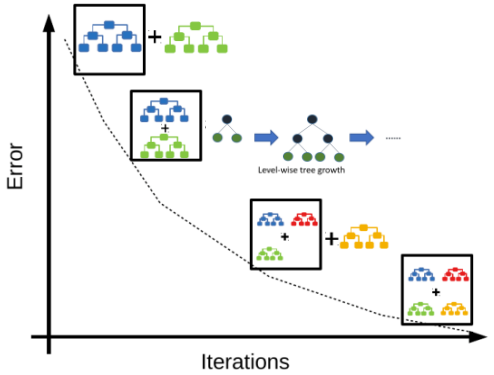
Evaluation: cross-validation

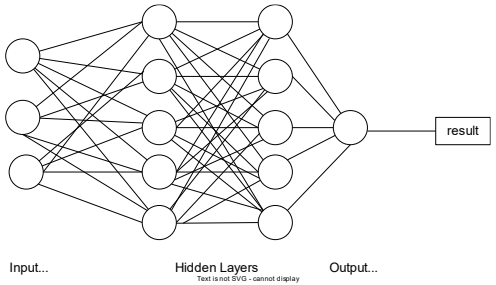


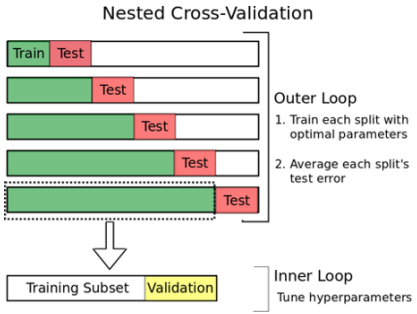












Time series forecasting methods

Classical statistical methods

- Formulate an underlying theory about
  - the dynamics of a time series
  - the statistics describing the noise and uncertainty in its behavior
- Make predictions and estimate the degree of uncertainty
  - by using the hypothesized dynamics of the process

Machine learning methods

- Data-driven: do not posit an underlying process or any rules about the underlying process
- Feature learning: allow for relationship to be learned
- Focus on identifying patterns that describe the process's behavior in ways relevant to predicting the outcome
  - Supervised learning
  - Unsupervised learning - clustering

Time series forecasting models

Classical statistical models

- Decompositional models
  - Explore historical changes over time by separating trend, seasonality, residual
- Exponential smoothing and ETS models
  - Describe the trend and seasonality
  - Additive or multiplicative error models
- ARIMA and SARIMA models
  - Describe the autocorrelations
  - Autoregression of past values
  - Moving average of past forecast errors
- Regression model
  - Capture relationships between forecast variable and predictor variables

Machine learning models

- Supervised learning
  - Linear regression (LinearRegression, Lasso, Ridge, ElasticNet)
  - Support vector regression
  - Decision trees
  - Ensemble learning (Random Forest, Gradient Boosting, AdaBoost, CatBoost, XGBoost, lightGBM)
  - Deep learning neural networks (MLP, CNN, RNN, LSTM, Attention-based model)
  - Naive bayes
- Instantiation with local learning
  - K-Neighbors Regressor
  - Lazy learning

Standarized steps and codes

- Import the necessary libraries
- Load the dataset and split into train and test
- Instantiate classifier/regressor and fit the model
- Predict the test set and compute the accuracy

Python3

```python
# Import models and utility functions
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_digits

# Setting SEED for reproducibility
SEED = 23

# Importing the dataset
X, y = load_digits(return_X_y=True)

# Splitting dataset
train_X, test_X, train_y, test_y = train_test_split(X, y,
                                                    test_size = 0.25,
                                                    random_state = SEED)

# Instantiate Gradient Boosting Regressor
gbc = GradientBoostingClassifier(n_estimators=300,
                                 learning_rate=0.05,
                                 random_state=100,
                                 max_features=5 )
# Fit to training set
gbc.fit(train_X, train_y)

# Predict on test set
pred_y = gbc.predict(test_X)

# accuracy
acc = accuracy_score(test_y, pred_y)
print("Gradient Boosting Classifier accuracy is : {:.2f}".format(acc))
```

Python3

```python
import lightgbm as lgb
from lightgbm import LGBMRegressor
lgr = LGBMRegressor()
lgr.fit(X_train, y_train)
y_pred5 = lgr.predict(X_test)
print("LightGBM - R2: ",
      r2_score(y_test, y_pred5))
```

Python3

```python
from xgboost import XGBRegressor
xgr = XGBRegressor()
xgr.fit(X_train, y_train)
y_pred2 = xgr.predict(X_test)
print("XGBoost - R2: ",
      r2_score(y_test, y_pred2))
```

Python3

```python
from catboost import CatBoostRegressor
cbr = CatBoostRegressor(iterations=100,
                        depth=5,
                        learning_rate=0.01,
                        loss_function='RMSE',
                        verbose=0)
cbr.fit(X_train, y_train)
y_pred4 = cbr.predict(X_test)
print("CatBoost - R2: ",
      r2_score(y_test, y_pred4))
```

What left for us to do?

Prepare the dataset

- Do the necessary preprocessing
- Choose the related features
- Decide the right horizon of input and output window

Choose the models

- Understand pros and cons of different models
- Design and apply new ML models based on data (advanced, see competitions)

Evaluate the models

- Cross validation
- Evaluation metrics

Preprocessing

Data cleaning

- Removing outliers
- Imputation – filling missing values

Data transformations - eliminate or reduce trend and seasonality

- Stationarity transformation
  - Log – stabilize variance
  - First or second order, or seasonal differencing – stabilize mean
- Seasonal and trend decomposition
  - Additive or multiplicative method
  - STL, ETS decomposition

Normalization – brings values to the same scale, divide them by a stable value

- median over the last season
- exponentially smoothed values

Preprocessing – into a supervised learning problem

Time series data can be phrased as supervised learning

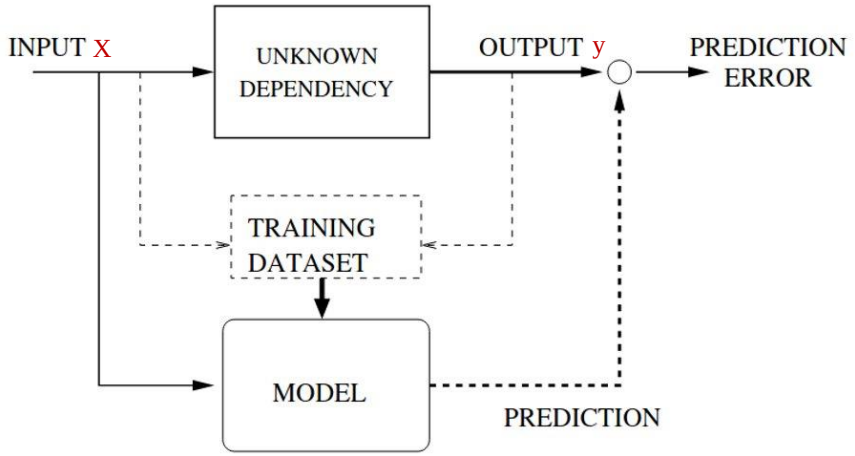
- where we have to infer from historical data the possibly nonlinear dependence between the input (past embedding vector X) and the output (future value y)

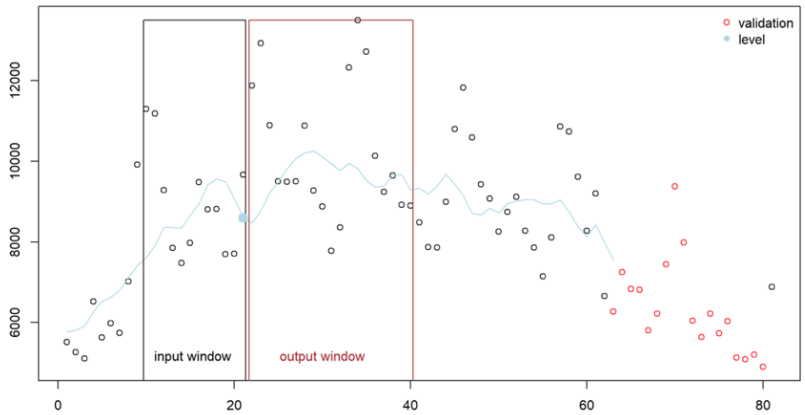
Sliding input window

- set up by experimentation for window width
- cover the period having certain data characteristics which are planned to be learned

Sliding output window

- equal to the prediction horizon





Sliding window for framing univariate time series

-only a single variable is observed at each time

Time series data
• time, measure • 1, 100 • 2, 110 • 3, 108 • 4, 115 • 5, 120

Framing for supervised learning
• X, y • ?, 100 • 100, 110 • 110, 108 • 108, 115 • 115, 120 • 120, ?

X is lag(-1) feature
Input window is 1
Output window is 1

Observations
• Each row has the previous time step as input (**X**) and the next time step as output (**y**) • The order between the observations is preserved • First and last rows with missing values need to be deleted

Sliding window for framing multivariate time series

- two or more variables are observed at each time

Time series data

- time, m1, m2
- 1, 0.2, 88
- 2, 0.5, 89
- 3, 0.7, 87
- 4, 0.4, 88
- 5, 1.0, 90

Framing for supervised learning predicting one measure

- X1, X2, X3, y
- ?, ?, 0.2, 88
- 0.2, 88, 0.5, 89
- 0.5, 89, 0.7, 87
- 0.7, 87, 0.4, 88
- 0.4, 88, 1.0, 90
- 1.0, 90, ?, ?

X1 and X2 are lag(-1) features
X3 is explanatory feature
Input window is 1
Output window is 1

Sliding window for framing multivariate time series

Time series data

- time, m1, m2
- 1, 0.2, 88
- 2, 0.5, 89
- 3, 0.7, 87
- 4, 0.4, 88
- 5, 1.0, 90

Framing for supervised learning predicting both measures

- X1, X2, y1, y2
- ?, ?, 0.2, 88
- 0.2, 88, 0.5, 89
- 0.5, 89, 0.7, 87
- 0.7, 87, 0.4, 88
- 0.4, 88, 1.0, 90
- 1.0, 90, ?, ?

X1 and X2 are lag(-1) features
Input window is 1
Output window is 1 but for 2 y

Sliding window for multi-step forecasting

- two or more future time steps are to be predicted

Time series data
• time, measure • 1, 100 • 2, 110 • 3, 108 • 4, 115 • 5, 120

Framing for supervised learning predicting multi-step
• X, y1, y2 • ? 100, 110 • 100, 110, 108 • 110, 108, 115 • 108, 115, 120 • 115, 120, ? • 120, ?, ?

X is lag(-1) features
Input window is 1
Output window is 2

Time series features (1)

Autoregressive features

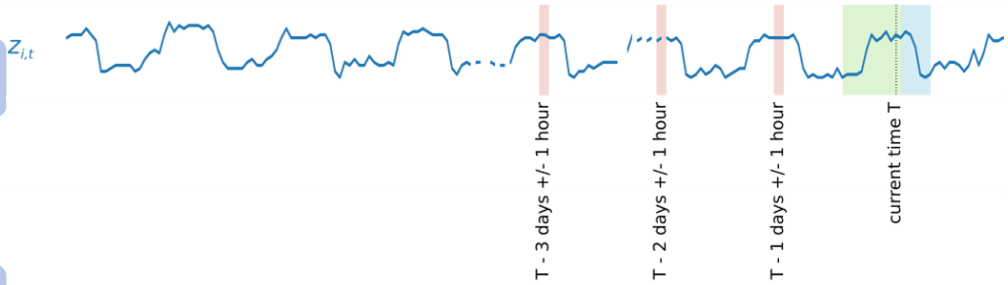
- Lag features
- Rolling aggregates
- Tumbling aggregates

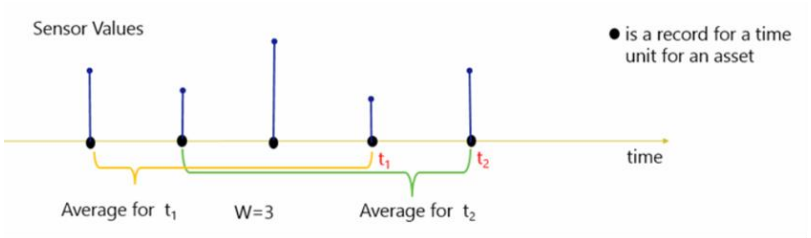
Calendar features

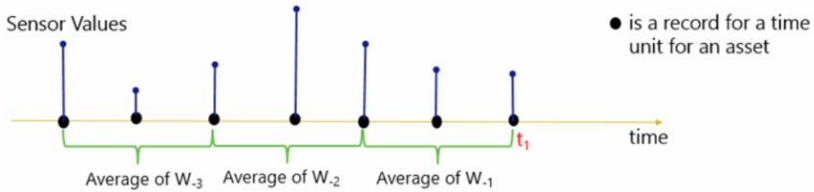
- Minute, hour, weekday, day of month, week of year, month
- holiday (yes or no)

Regressors – explanatory features

- Factors that affect the target value







Time series features (2)

Intervention features (e.g. competitor activity)

- 'Spike' feature: takes value one in the period of the intervention and zero elsewhere
- 'Step' feature: takes value zero before the intervention and one from the time of intervention onward

Trading days (e.g. sales)

- The number of trading days in a month

Distributed lags (e.g. advertising expenditure)

- Several features with different lagged values

$$
\begin{aligned}
x_1 &= \text{advertising for previous month;} \\
x_2 &= \text{advertising for two months previously;} \\
&\vdots \\
x_m &= \text{advertising for } m \text{ months previously.}
\end{aligned}
$$

Feature selection problem

Machine learning algorithms are known to degrade in performance when faced with many input/features that are not necessary for predicting the desired output.

Using all available features may negatively affect generalization performance, especially in the presence of irrelevant or redundant features.

Feature selection is to select some subset of features upon which to focus its attention, while ignoring the rest.

Feature selection can be seen as an instance of model selection problem.

An example of X and y

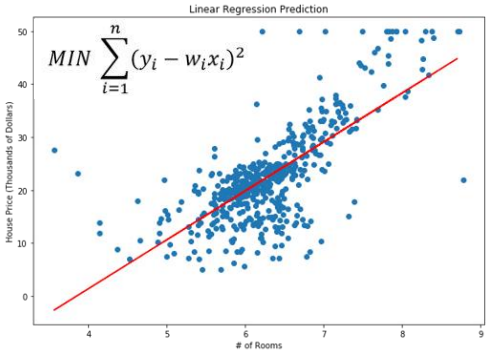
Lag features of explanatory variable | Lag features of target | Calendar features | Target y

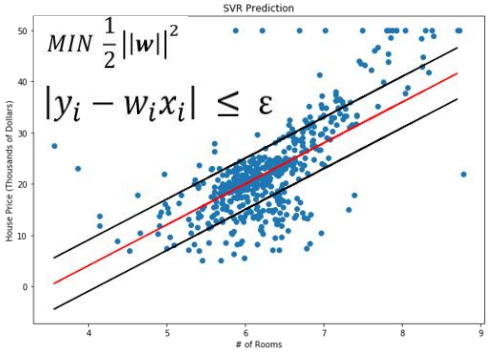
WindSpeed.lag(-2)	WindSpeed.lag(-1)	WindSpeed	TempMainBearing.10m.SQL.lag(-2)	TempMainBearing.10m.SQL.lag(-1)	Hour	Minute		TempMainBearing.10m.SQL
9.14998	8.81014	9.19353	47.00000	46.00000	13	00	13:00:00	46.00000
8.81014	9.19353	9.76677	46.00000	46.00000	13	10	13:10:00	46.00000
9.19353	9.76677	9.30929	46.00000	46.00000	13	20	13:20:00	46.00000
9.76677	9.30929	9.42590	46.00000	46.00000	13	30	13:30:00	47.00000
9.30929	9.42590	8.66883	46.00000	47.00000	13	40	13:40:00	47.00000
9.42590	8.66883	7.74852	47.00000	47.00000	13	50	13:50:00	47.00000
8.66883	7.74852	7.78619	47.00000	47.00000	14	00	14:00:00	46.00000
7.74852	7.78619	7.85361	47.00000	46.00000	14	10	14:10:00	46.00000
7.78619	7.85361	7.70215	46.00000	46.00000	14	20	14:20:00	46.00000
7.85361	7.70215	7.87644	46.00000	46.00000	14	30	14:30:00	46.00000
7.70215	7.87644	7.20292	46.00000	46.00000	14	40	14:40:00	46.00000
7.87644	7.20292	7.06888	46.00000	46.00000	14	50	14:50:00	46.00000
7.20292	7.06888	7.36850	46.00000	46.00000	15	00	15:00:00	45.00000
7.06888	7.36850	7.12104	46.00000	45.00000	15	10	15:10:00	45.00000
7.36850	7.12104	7.35081	45.00000	45.00000	15	20	15:20:00	45.00000
7.12104	7.35081	7.92635	45.00000	45.00000	15	30	15:30:00	45.00000
7.35081	7.92635	6.98474	45.00000	45.00000	15	40	15:40:00	45.00000
7.92635	6.98474	7.12135	45.00000	45.00000	15	50	15:50:00	45.00000
6.98474	7.12135	7.05697	45.00000	45.00000	16	00	16:00:00	44.00000

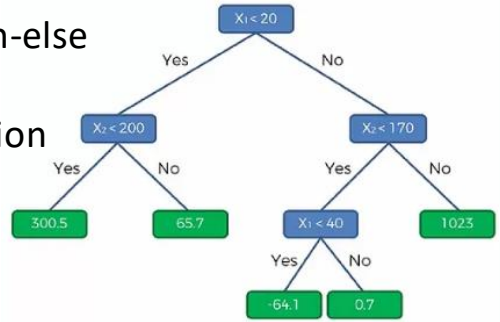
Normal machine learning models

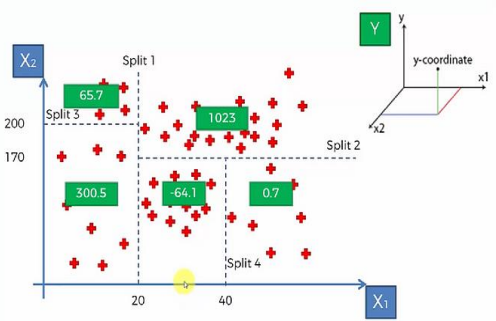
Supervised learning without special consideration of temporal dependence but with calendar features

- Linear regression (LinearRegression, Lasso, Ridge, ElasticNet)
  - Models the relationship between a scalar response and one or more explanatory variables
- Support vector regression (SVR)
  - Gives the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data
- Decision trees
  - Predicts the target value by learning simple if-then-else decision rules inferred from the data features
  - A tree is seen as a piecewise constant approximation







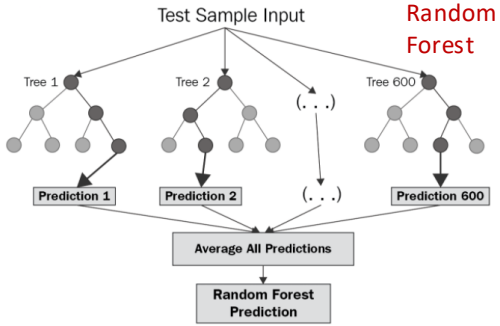


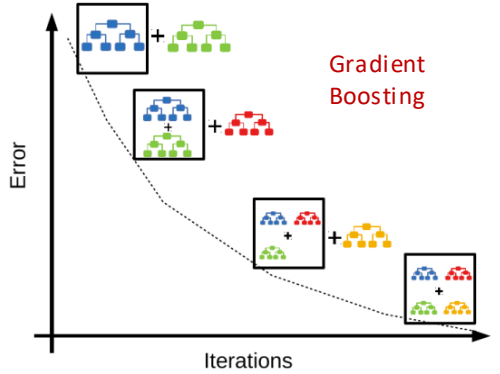
Normal machine learning models – cont.

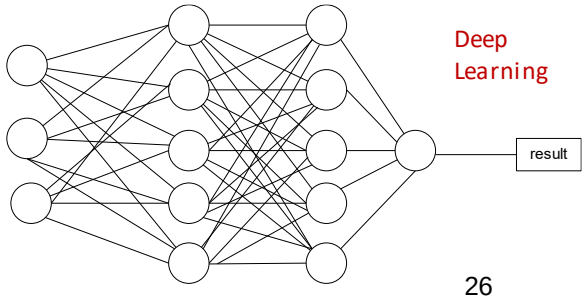
Supervised learning without special consideration of temporal dependence but with calendar features

- Ensemble learning (Random Forest, Gradient Boosting, AdaBoost, CatBoost, XGBoost, lightGBM)
  - Gives an additive (more powerful) prediction model in the form of an ensemble of weak prediction models, typically decision trees
- Deep learning neural networks (MLP, CNN, RNN, LSTM, Attention-based model)
  - ANN consists of interconnected neurons which process data through three or more layers, including input, output and hidden layers
  - A DNN is simply an ANN with deep layers
- Naive Bayes
  - Applies Bayes' theorem with strong (naïve) independence assumptions between features

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \qquad \text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$







More machine learning models

Designed for sequences

- See the input data as a sequence and learn temporal dependence
- Methods
  - RNN: LSTM
  - Attention-based model

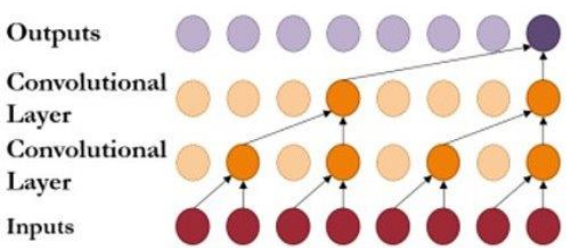
Local learning

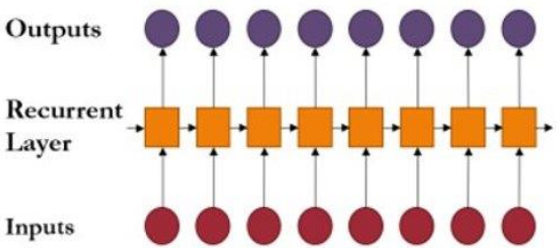
- Not supervised learning, but unsupervised clustering
  - Reduced number of assumptions
  - On-line learning capability
  - Modelling non-stationarity
- Methods
  - K-Neighbors Regressor
  - Lazy learning

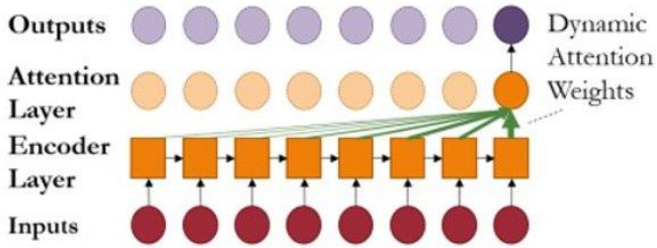
Learn temporal dependence

Deep learning models

- Multilayer perceptrons (MLPs)
  - Robust to noise, nonlinear, multivariate inputs, multi-step forecasts
- Convolutional neural networks (CNNs)
  - Feature learning
- Recurrent neural networks (RNNS): LSTM
  - Native support for sequences and can learn temporal dependence
- Attention-based model
  - Aggregate temporal features using dynamically generated weights







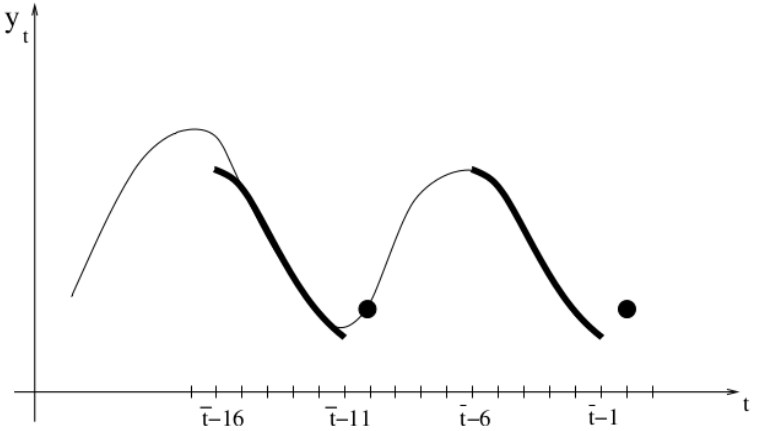
Local learning: K-neighbors

1. Compute the distance between the query x_q and the training samples according to a predefined metric.

2. Rank the neighbors on the basis of their distance to the query.

3. Select a subset of the *k* nearest neighbors according to the *bandwidth* which measures the size of the neighborhood.

4. Fit a *local* model(e.g. constant, linear,...)



Nearest-neighbor one-step-ahead forecasts. We want to predict at time $t-1$ the next value of the series y of order $n=6$. The pattern $y_{t-16}, y_{t-15}, \ldots, y_{t-11}$ is the most similar to the pattern $\{y_{t-6}, y_{t-5}, \ldots, y_{t-1}\}$. Then, the prediction $\hat{y}_t = y_{t-10}$ is returned.

Model evaluation

Training error:

- is not a good estimator (i.e. it is too optimistic and can be overfitting) of the generalization capability of the learned model.

Two alternative exists:

- Complexity-based penalty criteria
- Data-driven validation techniques
  - Testing
  - Holdout
  - K-fold cross-validation
  - Nested cross-validation

Nested CV: Almost unbiased estimate of true error

Outer loop for error estimation

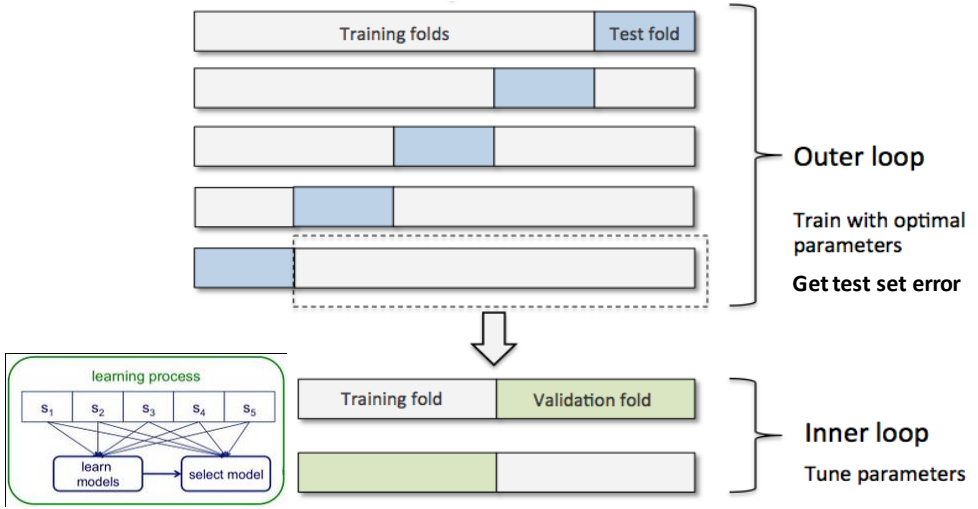
- Outer k-fold CV loop to split the data into training and test folds

Inner loop for parameter tunning

- Inner loop is used to select the model via k-fold cross-validation on the training fold

Connection between inner and outer loop

- For a given training set in outer loop, the optimal parameters tuned from its inner loop are used to train the model using the given entire training set



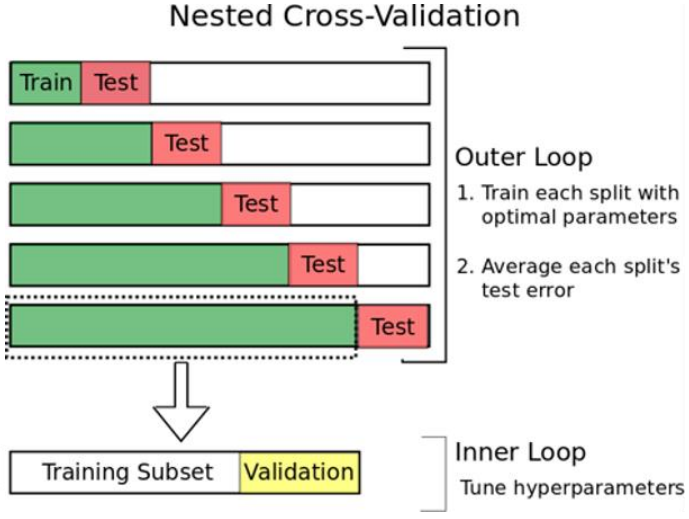
(Nested) CV for time series

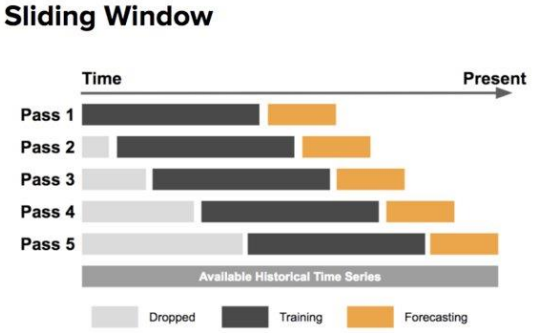
Normal CV (like k-fold) should not be used for two reasons

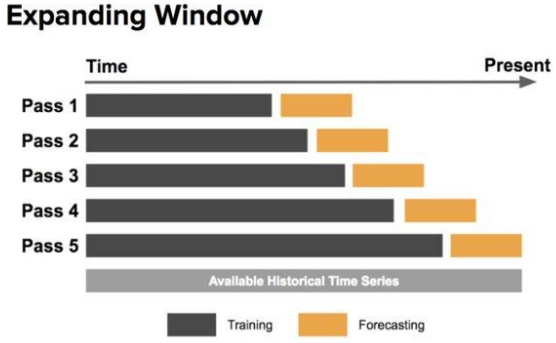
- Time series has temporal dependencies
- Arbitrary choice of test set by normal CV results in data leakage

Adaption

- Train on a set of data that is older than the test data
- Use expanding window when there is limited time series data

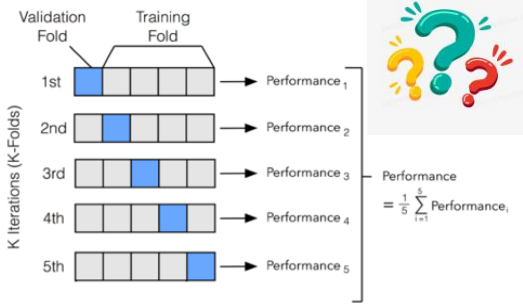






NTNU

Supervised metric types



Classification metrics

- Binary classification
  - Accuracy, Precision-Recall, F1 score, Specificity-Sensitivity, PR/AUC, ROC/AUC, Log loss
- Multi-task classification
  - Precision: Micro, macro, weighted
  - Multi-class log loss

Regression metrics

- Point estimation
  - MAE, RMSE, MAPE, sMAPE, MASE, R^2
- Probability estimation
  - Reliability: statistical tests
  - Sharpness: pinball loss, CRPS

Practical issues for building models

Data augmentation

- Limited data for deep learning or cross-validation

Retrain your model often

- Distribution is not stationary

No data leakage

- Do not learn from data/features that wouldn't be available at the time of prediction

Reasonable predictions even with missing features

- Missing data at the time of prediction

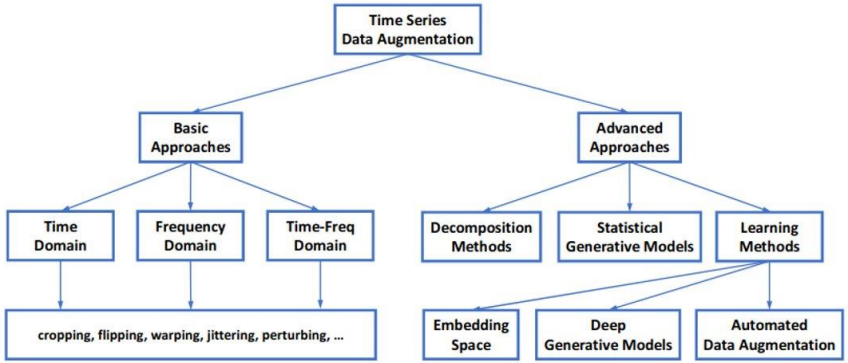
Data augmentation for time series data

Limited time series data

- Time series data is often very small compared to the data sets used in image processing or natural language processing.
- Deep learning relies heavily on a large number of training data to avoid over-fitting.
- (Nested) cross validation requires enough train/test splits

Data augmentation techniques for time series data

- See figure



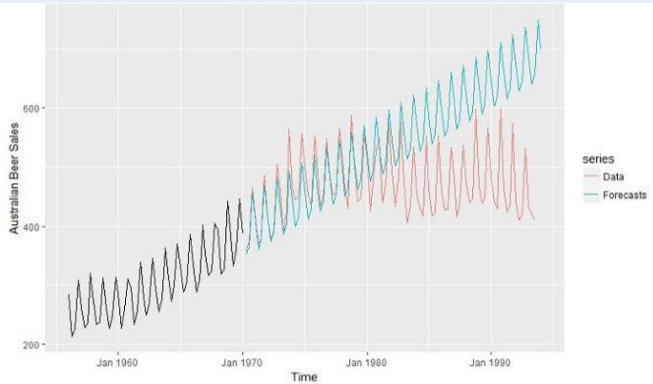
Retrain the time series model often

Image classification

- The visual properties of cats are stable over time
- Features that define cats are going to remain the same for the foreseeable future

Time series

- Real world business time series data are *not* stationary
- The statistical properties of your distribution will keep shifting as new actuals come in



No data leakage

Data leakage is the introduction of a feature that can not available at the time of prediction

- When working with time series
  - If our model learns to use tomorrows prediction for predicting today, the model cannot be generalized and used to predict future order demands.
- Choosing the right features for prediction
- Labeling our output values

Pitfalls of data leakage

- The trained model with data leakage is not usable in the production since the needed features are not available

Date	Output
2019/08/26	30
2019/08/27	35
2019/08/28	45
2019/08/29	??
2019/08/30	20
2019/08/31	16
2019/09/01	18

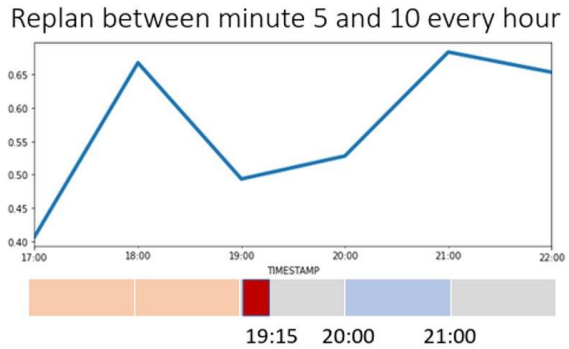
Reasonable predictions even with missing features

Missing or stale data at the time of prediction due to

- Collection from low-quality sensors and the nature of data recording process to concrete devices
- Technical problems in communication infrastructure
- External APIs or service providers is out of service, e.g. forecasting weather or price
- Human errors

Methods

- Omitting those missing values may cause temporal discontinuity
- Data imputation may alter the original time series
- Forecasting it without data imputation if possible and needed
- Building multiple models on subsets of features for selecting



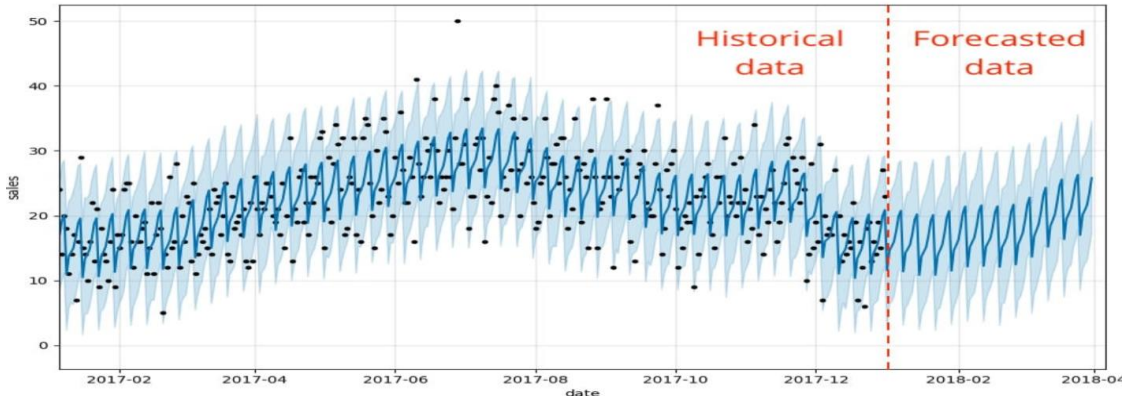
More requirements for time series forecasting

Extendable to multi-steps forecasting

- short, median, long term horizons

Probabilistic forecasting

- Uncertainty
- Predictive intervals or quantiles
- Quantile loss and machine learning algorithms
- Probability estimation

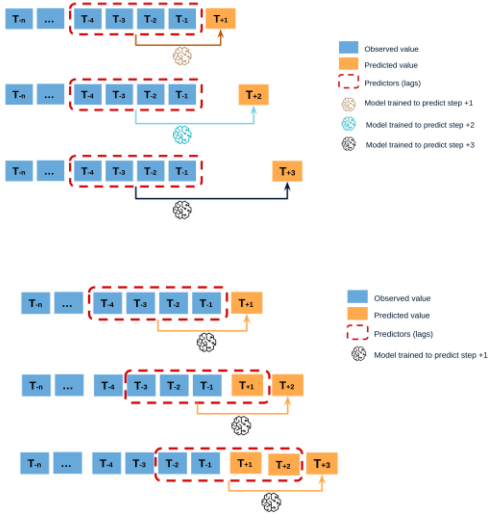


t-T t t+h

Multi-step forecasting

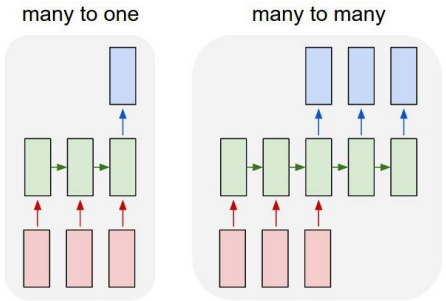
Direct, Recursive, DirRec strategies

- Direct: developing a separate model for each forecast time step
- Recursive: using a one-step model multiple times where the prediction for the prior time step is used as an input for making a prediction on the following time step
- DirRec: combining to offer the benefits of both methods



Multiple output strategies

- Multi-Input Multi-Output (MIMO): predicting the entire forecast sequence in a one-shot manner
  - Deep learning neural networks
  - Local learning
- DIRMO(DIRect+MIMO): partitioning the horizon in several blocks, and using MIMO to forecast the values inside each block



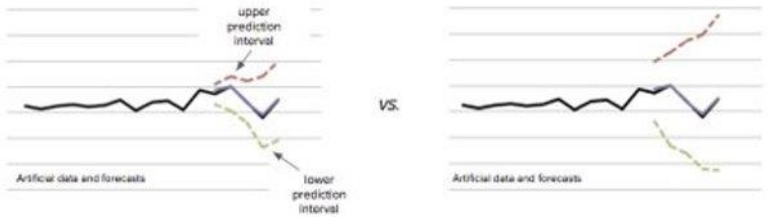
Uncertainty and probability forecasting

The thing we are trying to forecast is unknown (or we would not be forecasting it), and so we can think of it as a random variable

- Example, the total sales for next month could take a range of possible values, and until we add up the actual sales at the end of the month, we don't know what the value will be. So until we know the sales for next month, it is a random quantity
- Predictions are never absolute, and it is imperative to know the potential variations

Prediction intervals are just as important as the point forecast itself

- The difference in prediction intervals results in two very different forecasts
- The second forecast calls for much higher capacity reserves to allow for the possibility of a large increase in demand
- The further ahead we forecast, the more uncertain we are



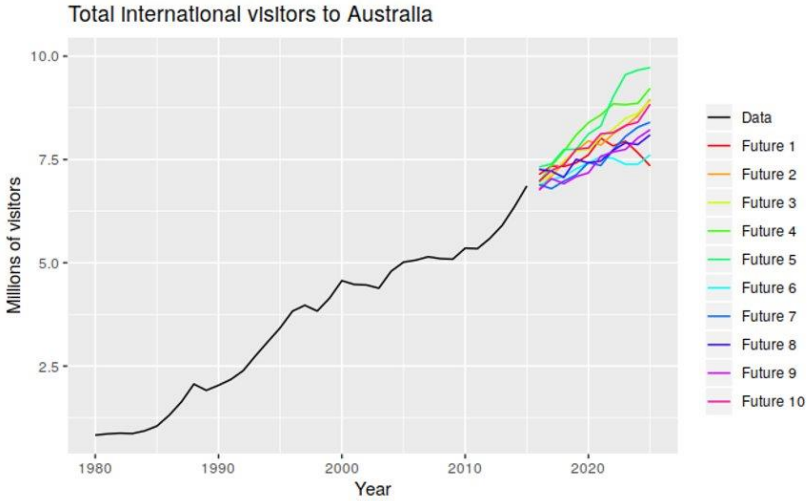
Predictions are themselves random variables with distribution

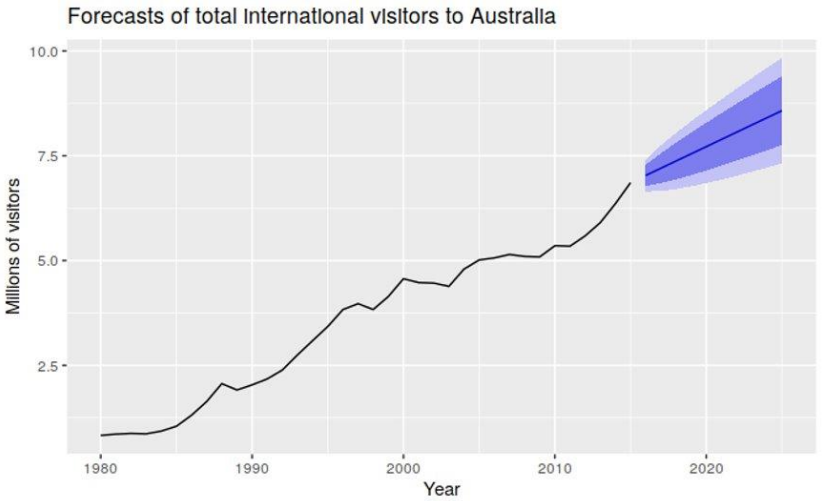
Point forecasts

- Only represent the expected prediction
- Does not model uncertainty
- The middle of the range of possible values the random variable could take

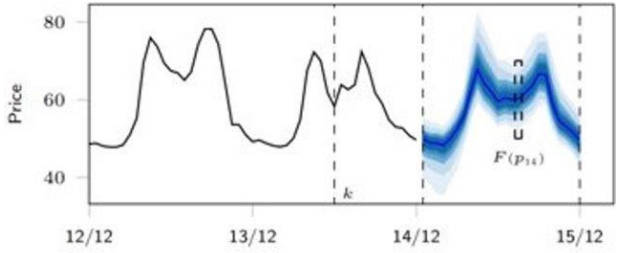
Forecast distribution

- Represent the prediction distribution
- Model the uncertainty of forecasting error
- A prediction interval giving a range of values the random variable could take with relatively high probability



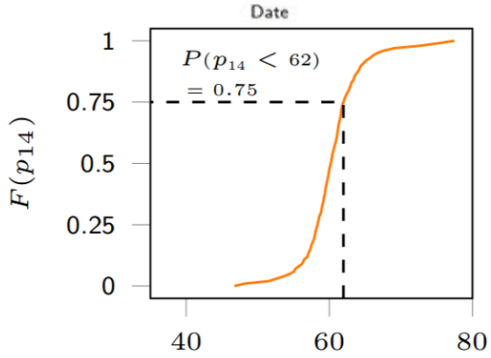


Probability forecasting methods



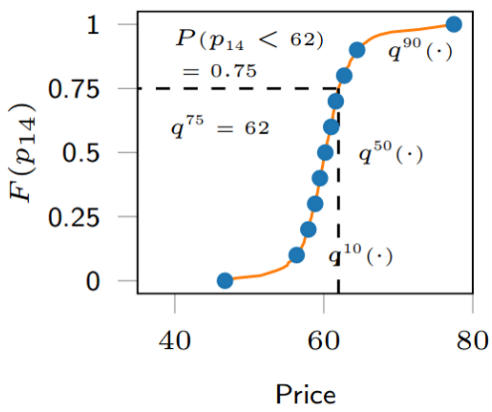
Parametric models

- The forecast is given by a full parameterization of the probability distribution, which is defined by random variable *p* and its cumulative distribution CDF *F(p),* continuous orange curve
- Limited by the distribution assumption



Quantile models

- *F(p)* is approximated building quantile models $q^{\alpha}(\vartheta,x)$, discrete blue points
- Quantile q^{α} of *p* is the value at which the probability of *p* is less than or equal to α, i.e. $\alpha=F(q^{\alpha})$
- More general: no assumptions needed



Quantiles

Quantiles are a generalization of prediction intervals and no assumption about distribution is needed

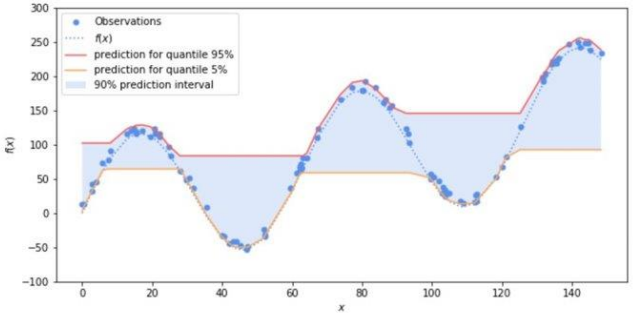
- Example: 90% prediction interval equals the interval $[q_5, q_{95}]$ of quantiles

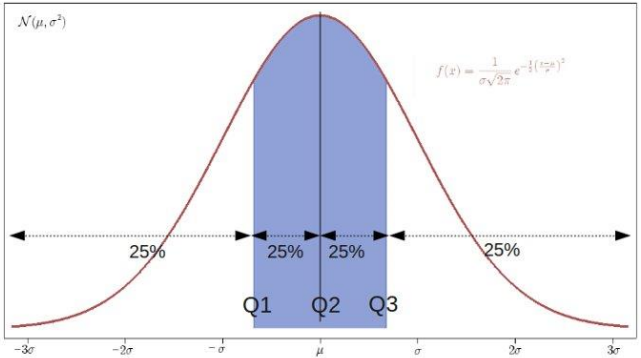
Cut points

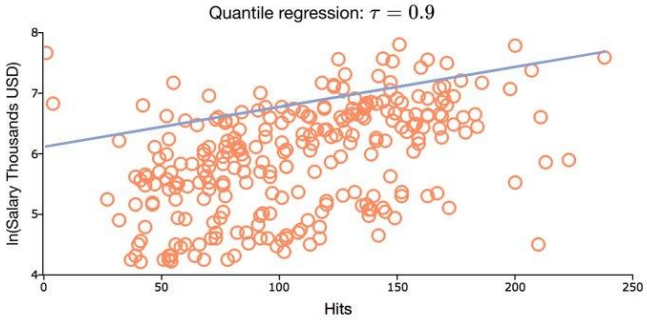
- dividing the range of a probability distribution into continuous intervals with equal probabilities
- dividing the observations in a sample in the same way

A quantile is the value below which a fraction of observations in a group falls

- Example: A prediction for quantile 0.9 should over-predict 90% of the times





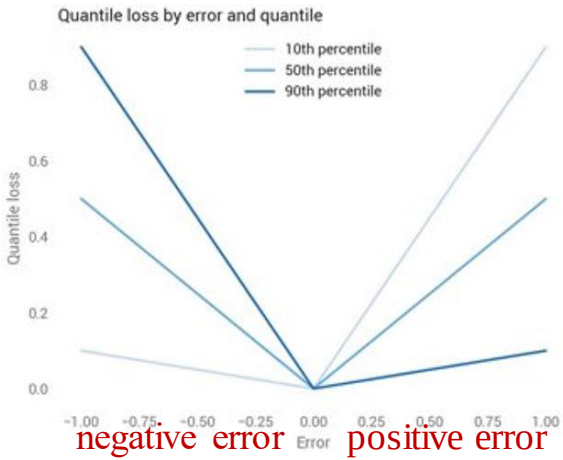


Quantile loss

$$L_q(y, \hat{y}) = (q-1) \sum_{i \in y_i < \hat{y}_i} |y_i - \hat{y}_i| + q \sum_{i \in y_i \geq \hat{y}_i} |y_i - \hat{y}_i|$$

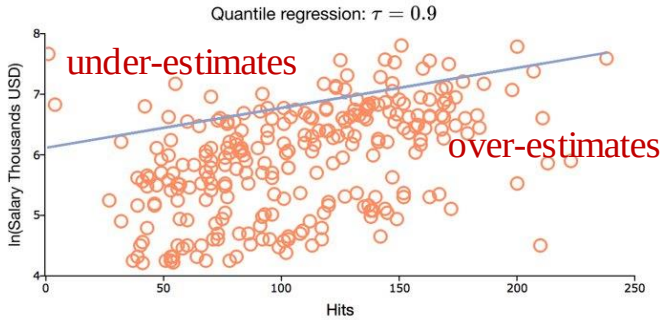
weight — over-estimates positive error

weight — under-estimates negative error



For true values *y*, the predicted values *y_hat*, and a desired quantile *q*

- Quantile 50% weights underestimates equally to overestimates
- The closer the desired quantile gets to 100%, the more the loss function penalizes under-estimates, meaning assign more of a loss to under-estimates/negative error than to over-estimates/positive error.

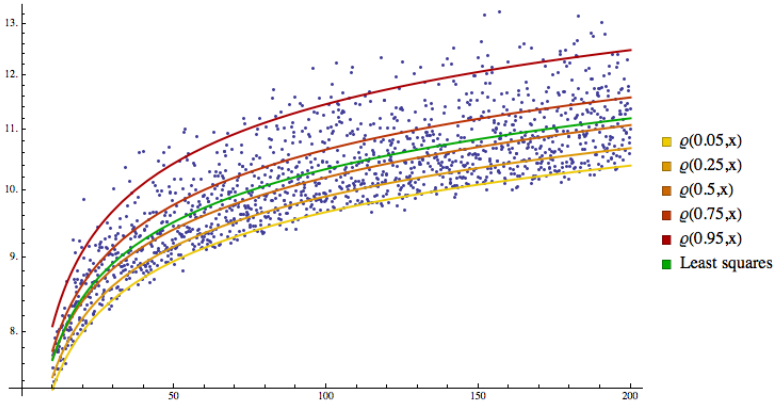


This quantile loss can be used to calculate prediction intervals for

- Regression, neural networks, tree based models

Linear quantile regression

$$y_i = \beta' X_i + \epsilon_i$$



OLS: Regressions minimize the squared-error loss function to predict a mean

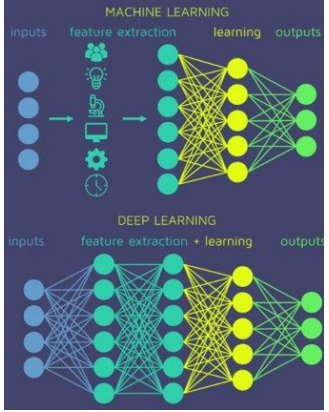
- Prediction intervals are calculate based on standard errors and the inverse normal CDF $\sum_i^N (y_i - \hat{y}_i)^2$

Quantile regressions minimize the quantile loss in predicting a certain quantile

- Optimizing this loss function results in an estimated linear relationship between y_i and x_i where a portion of the data, τ, lies below the line and the remaining portion of the data, 1-τ, lies above the line

$$\tau \sum_{y_i > \hat{\beta}_\tau' X_i} |y_i - \hat{\beta}_\tau' X_i| + (1-\tau) \sum_{y_i < \hat{\beta}_\tau' X_i} |y_i - \hat{\beta}_\tau' X_i|$$

Deep learning and quantiles



Deep learning vs traditional machine learning

- Large and deep neural networks
- Feature learning, such as automatic feature extraction

Quantiles are predicted in DL by passing the quantile loss function, neither RMSE nor MAE

- Keras: each quantile must be trained separately
- Tensorflow: to leverage patterns common to the quantiles
  - Co-learning across the quantiles in its predictions, where the model learns a common kink rather than separate ones for each quantile

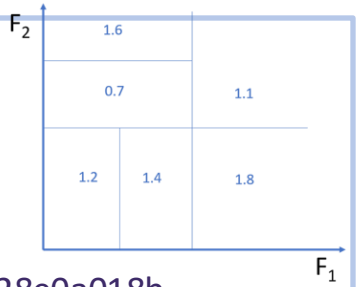
Quantile regression on gradient boosting

$$L_q(y, \hat{y}) = (q-1) \sum_{i \in y_i < \hat{y}_i} |y_i - \hat{y}_i| + q \sum_{i \in y_i \geq \hat{y}_i} |y_i - \hat{y}_i|$$

Using a second-order approximation of the quantile loss function

- Tree boosting vs gradient boosting in splitting procedure
  - every (dimnesion, cutoff) pair vs a single pass per dimension (next-largest cut point)
- Point forecasts vs forecast distribution

Implemented differently, but all support explicit quantile prediction



- Scikit-learn's implementation GradientBoostingRegressor
- LightGBM
  - http://jmarkhou.com/lgbqr/
- Xgboost
  - https://towardsdatascience.com/regression-prediction-intervals-with-xgboost-428e0a018b
- Catboost

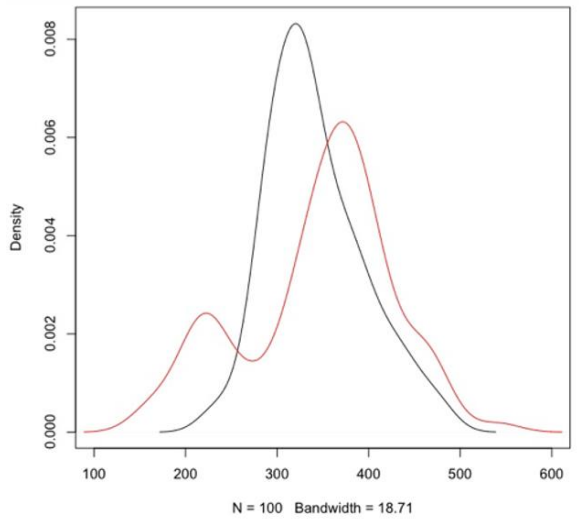
Regression metrics - Probability estimation

Qualify the performance by comparing two distributions

- True distribution: missing, can not be observed
  - Only the past observations
- Predictive distribution: from regression
  - Often not cumulative distribution but prediction interval, quantiles

How to evaluate probabilistic forecasts?

- Reliability: statistical consistency between distributional forecasts and observations
  - Joint property of the predictions and the observations
  - Formal statistical tests: Kupiec test, Christoffersen test, Berkowitz test
- Sharpness: how tightly the predictive distribution covers the actual one, i.e., to the concentration of the predictive distributions
  - Property of the predictions only
  - A-single-number metrics: pinball loss, winkler score, CRPS(continuous rank probability score)



Forecasting competition – M4 started on January 1, 2018, ended on May 31, 2018

100 000 time series

- Domains: Micro, Macro, Industry, Finance, Demographic
- Intervals: Yearly, Quarterly, Monthly, Weekly, Daily, Hourly

Submitted 61 methods

- 53 statistical, 6 pure ML, 2 hybrid

Result

- 12 of top 17 methods were combinations of mostly statistical methods
- Top 2 are hybrids that combine statistical and machine learning methods
- 1st is a hybrid between statistical and ML method (ES-RNN)
- 2nd is a combination of 7 statistical and one ML method, weighted by ML method
- Pure ML methods performed worse than ARIMA or Exponential Smoothing, only one was better than seasonally adjusted naïve forecast
- Top 3 use information from multiple time series (global models)

Winner of M4: ES-RNN

Combines hand-coded exponential smoothing with a black-box recurrent neural network (RNN)

- Forecast equation
- Preprocessed input for RNN
- Level equation
- Season equation

$$\hat{y}_{t+1..t+h} = RNN(X_t) * l_t * s_{t+1..t+h}$$

$$x_i = \frac{y_i}{l_t s_i}$$

$$l_t = \alpha(y_t/s_t) + (1-\alpha)l_{t-1}$$

$$s_{t+m} = \gamma(y_t/l_t) + (1-\gamma)s_t$$

Combines a global model trained on multiple time series with a local model specific to target

- Level and season equations are per time-series (local)
- RNN is trained on multiple time-series (global) time series

Beating M4's Winner Score: N-BEATS 2019

- N-BEATS: Neural Basis Expansion Analysis for interpretable Time Series forecasting
- New kid on the block
- End-to-end Deep Learning approach
- https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/133551

Forecasting competition – M5 started on March 3, 2020, ended on July 1, 2020

Around 42,000 hierarchical time series

- Estimate the unit sales of Walmart retail goods

Companion competitions and participants

- Accuracy (point forecasts): 7,092 participants on 5,507 teams from 101 countries
- Uncertainty (Probability forecasting): 1,137 participants on 892 teams from 94 countries

Extended the results of the previous four competitions by:

- significantly expanding the number of participating methods, especially those in the category of Machine Learning
- including exogenous/explanatory variables in addition to the time series data
- using grouped, correlated time series
- focusing on series that display intermittency

Forecasting competition – M5 findings

Winning methods in top 50

- Accuracy: Most of the methods examined utilized LightGBM
- Uncertainty: Most of the methods examined utilized LightGBM, the rest mostly on Long Short-Term Memory Neural Networks (LSTM NNs)

Findings

- The superiority of simple ML methods
  - all top-performing methods were both "pure" ML ones and significantly better than all statistical benchmarks and their combinations
- The value of combining and cross-learning
- The beneficial effect of external adjustments.
- The value added by effective CV strategies and data augmentation.
- The importance of exogenous/explanatory variables.

Forecasting competition – M6
started on February 2022, ended on January 8, 2023

Real time financial forecasting competition

- consisting of 50 stocks from the Standard and Poor's (S&P) 500 index and 50 international exchange-traded funds (ETFs)
- covering a variety of asset categories and countries
- 12 rolling origins (every four weeks) for participants to provide their submissions and be evaluated once the actual data becomes available

Aims and focuses

- empirically identify the most appropriate way of forecasting financial (stock and ETF) prices
- investigate the connection between the accuracy/uncertainty of such forecasts and the associated returns on investment (investment decisions made based on the use of forecasts)
- overall performance of forecasts and investment decisions

Forecasting competition – M6 notes

Winners

- Overall performance: by StanekF_STU
  - MtMs: a meta-learning approach that is based on an encoder-decoder hypernetwork, capable of identifying the most appropriate parametric model for a given family of related prediction tasks.
  - Relative good forecasting performance and return performance
- Return (IR) performance: by Galloping Grgo
  - AutoTS package (subclass of AutoML) has many preprocessing methods and many of forecasting models. The magic comes in how it combines them, optimizes them with a genetic algorithm, and evaluates and chooses the best.
  - From forecasts to investment decisions is very simple, a 'hinge' method is used that is the simple average of the upper and lower forecasts (probabilistic forecasting) and used that instead of the point forecast. Then this hinge forecast is turned into a percentage return for the month by simply taking the forecast price against the starting price, normalized it with the other stocks returns, and rounded down, thus generating the investment amount for the month.
  - Relative bad forecasting performance, and no fancy investment strategy or any stock market type knowledge applied, but ranked first in return performance

Findings

- *Foresight* Practitioner Conference – Future of Forecasting and the M6 Competition
  - November 6-7, 2023 New York
  - https://forecasters.org/events/foresight-practitioner-conference/

M6 forecasting, Overall Competition winners: Total prize money $124,000

Overall Performace	Prize Money	Forecasting (RPS) Performance	Prize Money	Return (IR) Performance	Prize Money
StanekF_STU (CERGE-EI	$32,000	Dan	$16,000	Galloping Gargo	$16,000
MP - Miguel Pérez Michaus	$16,000	MP - Miguel Pére	$8,000	Selamlar H.T.(Ata	$8,000
Peters_STU	$8,000	SebastianR	$4,000	Quantmetry	$4,000
OPPO XZ Lab	$4,000	AdaGaussMC_ST	$2,000	LAIR	$2,000
Microprediction	$2,000	StanekF_STU (CE	$1,000	Tilefish Poele	$1,000
Total	**$62,000**		**$31,000**		**$31,000**

Position	Team	Overall Rank (OR)	Performance of Forecasts (RPS)	Rank (Forecasts)	Performance of Decisions (IR)	Rank (Decisions)
20	bbb3c357 Galloping Gargoyles	46.5	0.16389	92	32.88981	1
43	f8ffbd15f Selamlar H.T.(Ataforecasting)	56	0.17028	110	28.65539	2
48	ccb2bf59 Quantmetry	58	0.17089	113	24.72015	3
15	9b7bbf059 LAIR	41.5	0.16122	79	21.53685	4
82	872bdbc9 Tilefish Poele	81.5	0.30977	158	17.94236	5
1	ed597d34 StanekF_STU (CERGE-EI)	5	0.15689	4	13.79121	6
72	75097136 DeepFuture	77.5	0.23052	148	12.47064	7

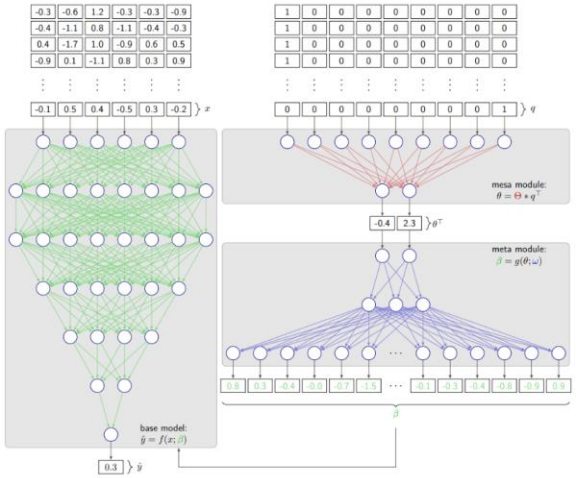


Figure 1: A diagram illustrating MtMs
We denote $q \in \{0,1\}^M$ as the indicator of the task under consideration and $\Theta = (\theta_1, \theta_2, \ldots, \theta_M)$.

Forecasting competition – GEFCom 2012
-Global Energy Forecasting Competition 2012

Point forecasts on four tracks on energy related time series

- Wind production, solar power, electric load, electricity price

Winner of wind production

- Pre-processing: K-MEANS similarity model to detect farm and overall intertia components
- Models: GBM for each farm, for specific time intervals, and all data instances
- Post processing: ensemble and smooth the predictions

Dataset

- https://www.kaggle.com/c/GEF2012-wind-forecasting/leaderboard

Forecasting competition – GEFCom 2014
-Global Energy Forecasting Competition 2014

Probability forecasting on four tracks on energy related time series

- Wind production, solar power, electric load, electricity price

Top five winners of wind production

- Gradient Boosting Machines (GBM)
- Quantile Regression Forest (QRF) and Gradient Boosting Decision Trees (GBDT)
- k-Nearest Neighbor (k-NN)
- Multiple Quantile Regression (MQR)
- k-Nearest Neighbor (k-NN) and Kernel Density Estimation (KDE)

Dataset

- http://blog.drhongtao.com/2017/03/gefcom2014-load-forecasting-data.html

Trends

1. Time intervals are getting shorter, e.g. hourly, 5-minute, 1-minute

2. Data is becoming more messy

3. Time series are often hierarchical, e.g. many correlated timeseries organized by geographic areas

4. Forecasts often require more information than just the time series e.g. categorical variables, event streams, relations between timeseries, logical constraints

5. Prediction intervals are often more important for decision making than point forecasts

Best practices

Statistical models work best with large intervals, data is clean and small and there are clear trends and seasonalities

Physical or econometric models work best when physical process or theory is well known and can be formalized in equations

Machine learning models work well when intervals are small, data is large and messy and there are a lot of explanatory features

If you have many related time series, consider a global machine learning model

LightGBM displays several advantages over other ML alternatives in forecasting tasks

- effective handling of multiple features (past and exogenous/explanatory variables) of various types (numeric, binary, and categorical)
- fast to compute compared to typical GBM implementation
- does not depend on data pre-processing and transformations
- requires the optimization of only a relatively small number of parameters
- both for point forecasts and probability forecasting

Automated machine learning (AutoML)

Automating the process of applying machine learning

- Data pre-processing, Feature engineering, Model selection, Hyperparameter optimization

AutoML for time series forecasting

- Facebook Prophet
  - Package for Python and R
  - Statistical model that handles non-linear trend, multiple seasonalities, holidays, missing data and outliers
- Microsoft Azure AutoML
  - Cloud service
  - Uses a recommender system to search for optimal ML pipeline including data preprocessing, feature and model engineering steps
- Amazon SageMaker DeepAR
  - Cloud service
  - RNN model for training on multiple timeseries
- OpenAI ChatGPT
  - Can create one for microprediction (effecting autonomous prediction using lightweight markets instead of models)
    - https://medium.com/geekculture/chatgpt-acquires-realtime-operational-intelligence-23675d2e0f3b

Advanced topics

Modern machine learning for time series forecasting on "irregular" data

- Use of FNN, LSTM and CNN for time series modelling and forecasting.
- Attention mechanism in LSTM-based architecture for time series forecasting.
- The problem of small data and low-data regime in the time series domain.
- Unsupervised and Self-Supervised Learning for different time series related tasks.
- Transfer Learning and Transformer architecture.
- Few-Shot Learning and TS Classification in low-data regime.
- GAN for time series analysis (i.e. Anomaly Detection, Data Imputation, Data Augmentation, Data Generation, Privacy).
- (Deep) Echo State Network and Spiking Network for Time Series Analysis

Statistical models: Vector autoregression (VAR, VMA, VARMA)

- Two or more time series influence each other as they change over time

Reading list

Book: FORECASTING: Principles and Practice

- https://OTexts.org/fpp3/
- https://github.com/robjhyndman/fpp3-package

Video: Time series forecasting

- From Udacity: https://classroom.udacity.com/courses/ud980

Competitions

- M4: https://forecasters.org/resources/time-series-data/m4-competition/
- M5 Accuracy: https://www.kaggle.com/c/m5-forecasting-accuracy
- M5 uncertainty: https://www.kaggle.com/c/m5-forecasting-uncertainty
- M6: https://mofc.unic.ac.cy/the-m6-competition/
- GEFCom 2012: http://www.drhongtao.com/gefcom/2012
- GEFCom 2014: http://www.drhongtao.com/gefcom/2014

Reading list - Notebooks

Jupyter notebooks from Kaggle that you can edit and run

- 1. https://www.kaggle.com/jagangupta/time-series-basics-exploring-traditional-ts
- 2. https://www.kaggle.com/freespirit08/time-series-for-beginners-with-arima
- 3. https://www.kaggle.com/surajyathinatti/time-series-regressionmodels
- 4. https://www.kaggle.com/robikscube/tutorial-time-series-forecasting-with-xgboost
- 5. https://www.kaggle.com/dimitreoliveira/deep-learning-for-time-series-forecasting
- 6. https://www.kaggle.com/mayer79/rnn-starter-for-huge-time-series
- 7. https://www.kaggle.com/amirrezaeian/time-series-data-analysis-using-lstm-tutorial
- 8. https://www.kaggle.com/rohanrao/a-modern-time-series-tutorial
- 9. https://www.kaggle.com/robikscube/time-series-forecasting-with-prophet

Reading list – papers (1)

- 1. Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Ael Le Borgne, "Machine Learning Strategies for Time Series Forecasting," Chapter in Lecture Notes in Business Information Processing (2013)
- 2. Lim, Bryan, and Stefan Zohren, "Time Series Forecasting with Deep Learning: A Survey," (2020)
- 3. Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods", International Journal of Forecasting 36, no. 1 (2020): 54–74.
- 4. Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: Results, Findings, Conclusion and Way Forward," International Journal of Forecasting 34, no. 4 (2018): 802–808.
- 5. Barker, Jocelyn, "Machine learning in M4: What makes a good unstructured model?" International Journal of Forecasting 36, no. 1 (2020): 150–155.
- 6. Spyros Makridakisa, Evangelos Spiliotis, Vassilios Assimakopoulos, "The M5 Accuracy competition: Results, findings and conclusions", International Journal of Forecasting, 2020
- 7. Spyros Makridakisa, Evangelos Spiliotis, Vassilios Assimakopoulos, "The M5 Uncertainty competition: Results, findings and conclusions", International Journal of Forecasting, 2020

Reading list – papers (2)

- 8. Hong, Tao, Pierre Pinson, and Shu Fan, "Global Energy Forecasting Competition 2012," International Journal of Forecasting 30, no. 2 (2014): 357–363.
- 9. Silva, Lucas, and Belo Horizonte, "A Feature Engineering Approach: GEFCom 2012 – Wind Power Forecast," International Journal of Forecasting 30, no. 2 (2014): 395–401.
- 10. Fry, Chris, and Michael Brundage. "The M4 Forecasting Competition-A Practitioner's View," International Journal of Forecasting 36, no. 1 (2020): 156–160.
- 11. Hong, Tao, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman, "Probabilistic energy forecasting - Global Energy Forecasting Competition 2014 and beyond", International Journal of Forecasting 32, no. 3 (2016): 896–913.

Reading list - links

- 1. “An Introduction to Support Vector Regression (SVR).” Accessed August 31, 2022. https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2
- 2. “Decision Trees.” Accessed August 31, 2022. https://scikit-learn.org/stable/modules/tree.html
- 3. “M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model.” Accessed August 31, 2022. https://eng.uber.com/m4-forecasting-competition/
- 4. “Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python.” Accessed August 31, 2022. https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/
- 5. “5 Regression Loss Functions All Machine Learners Should Know.” Accessed August 31, 2022. https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
- 6. “Quantile regression, from linear models to trees to deep learning.” Accessed August 31, 2022. https://towardsdatascience.com/quantile-regression-from-linear-models-to-trees-to-deep-learning-af3738b527c3
- 7. “M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model.” Accessed August 31, 2022. https://eng.uber.com/m4-forecasting-competition/
- 8. “1st Place Solution | Kaggle.” Accessed August 31, 2022. https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/43795#latest-567361
- 9. “Prophet.” Prophet. Accessed August 31, 2022. http://facebook.github.io/prophet/.
- 10. “What Is Automated ML / Automl - Azure Machine Learning.” Accessed August 31, 2022. https://docs.microsoft.com/en-us/azure/machine-learning/service/concept-automated-ml.
- 11. “DeepAR Forecasting Algorithm - Amazon SageMaker.” Accessed August 31, 2022. https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html.