CERTH
CENTRE FOR
RESEARCH & TECHNOLOGY
HELLAS

Data pre-processing & filtering techniques

Nikos Tsalikidis, CERTH

CERTH SmartWins Summer School: Day 3
06 July 2023
Thessaloniki

Funded by the European Union

This project has received funding from the European Union’s Horizon research and innovation programme under grant agreement No 101078997

SmartWins

Key topics

- Data merging from various sources (Energy, Weather etc.)
- Time resolution selection (5 minutes, 15 minutes, 1 hour etc.)
- Timestamp formatting (Local to UTC or UTC to Local)
- Seasonality parameters selection
- Outlier detection and replacement
- Missing values identification and filling
- Normalization/Standardization
- Sliding Window technique

Data preparation

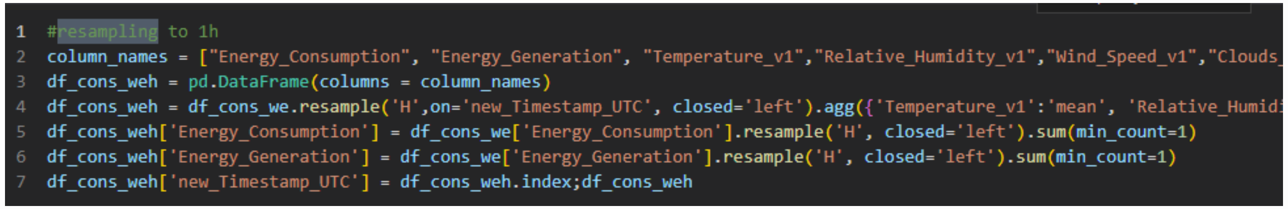
- Analyze the energy data source
  - Generation (PV, Wind etc.)
  - Load (Electricity, Heating, Cooling etc.)
- Weather Data preparation
  - Check weather APIs (MeteoStat, VisualCrossing etc.)
  - Decide about weather forecast
- Other sources (occupancy, holidays, sensors etc.)
- Check time periods
- Validate each dataset

Data merging from various sources (Energy, Weather etc.)

- Understand the data sources:
  - APIs
  - Databases
  - .csv, .xlsx files etc.
- Choose merging technique
  - Inner/outer/right/left joining
  - concatenation
  - appending
- Identify merging key (Timestamp etc.)
- Validate the merged dataset

Time resolution selection (5 minutes, 15 minutes, 1 hour etc.)

- Check each dataset’s original resolution, for example:
  - Energy dataset per 15 minutes
  - Weather dataset per one hour
  - Occupancy per 8 hours
- Define the 1st Step: One hour, 15 minutes, One Day
- Define Multi-step forecasting horizon: ex. 24 hours ahead (24 or 96 steps)
- Ensure that the timestamps in your datasets are aligned correctly
- For real time integration, check time resolution consistency!
- Data often resampled to higher or lower resolutions, to fill missing intervals:



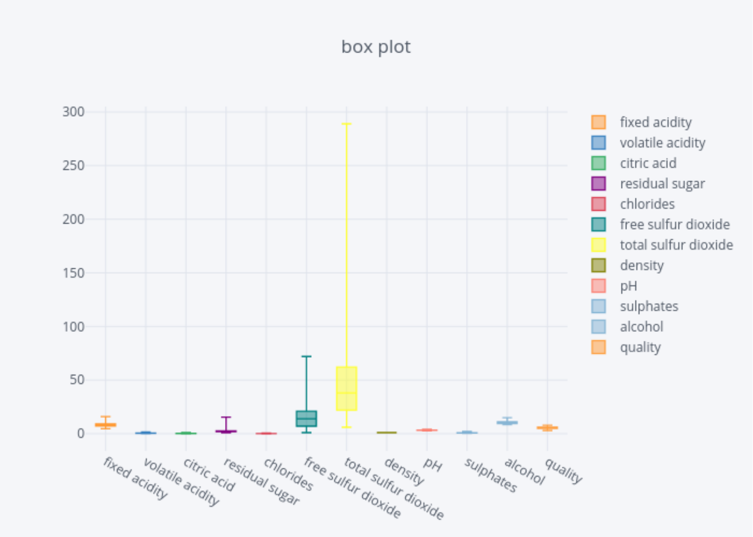
```
#resampling to 1h
column_names = ["Energy_Consumption", "Energy_Generation", "Temperature_v1","Relative_Humidity_v1","Wind_Speed_v1","Clouds_
df_cons_weh = pd.DataFrame(columns = column_names)
df_cons_weh = df_cons_we.resample('H',on='new_Timestamp_UTC', closed='left').agg({'Temperature_v1':'mean', 'Relative_Humidi
df_cons_weh['Energy_Consumption'] = df_cons_we['Energy_Consumption'].resample('H', closed='left').sum(min_count=1)
df_cons_weh['Energy_Generation'] = df_cons_we['Energy_Generation'].resample('H', closed='left').sum(min_count=1)
df_cons_weh['new_Timestamp_UTC'] = df_cons_weh.index;df_cons_weh
```

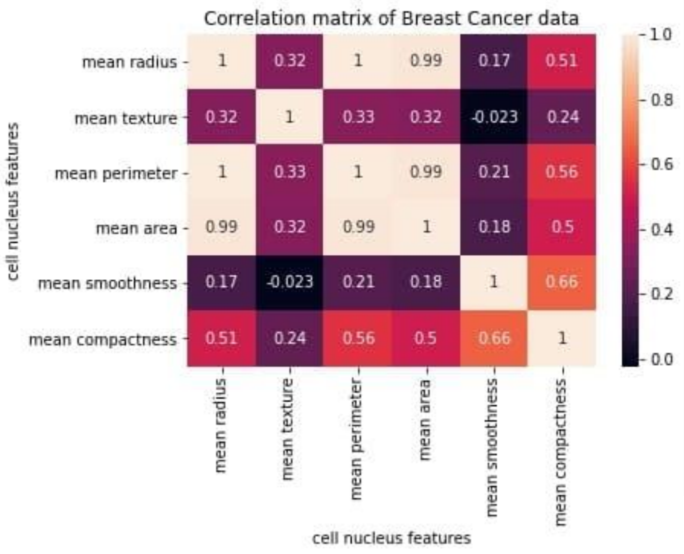
Timestamp formatting

- Local to UTC or UTC to Local: Converting timestamps between local time and UTC (Coordinated Universal Time) is a common task in handling time-related data
- Understand time zones
- Consider daylight saving time changes
- Historical time zone differences
- Use libraries and modules like:
  - datetime
  - pytz
  - strftime()

EDA & Seasonality parameters selection

- Understand the data—> Exploratory Data Analysis(EDA)
- EDA: applying a set of statistical techniques aimed at exploring, describing & summarizing the nature of the data
- Map influence by relevant exogenous factors (e.g., for energy load: ambient temperature)
- Correspond with the forecasting step and horizon
- Identify overall time period of the data
- Don’t be afraid to select extra parameters at first
- Use correlation tables, heatmaps

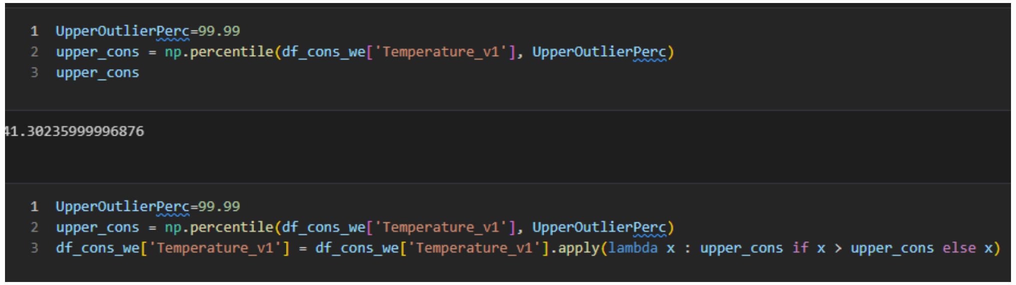



Correlation matrix of Breast Cancer data

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness
mean radius	1	0.32	1	0.99	0.17	0.51
mean texture	0.32	1	0.33	0.32	-0.023	0.24
mean perimeter	1	0.33	1	0.99	0.21	0.56
mean area	0.99	0.32	0.99	1	0.18	0.5
mean smoothness	0.17	-0.023	0.21	0.18	1	0.66
mean compactness	0.51	0.24	0.56	0.5	0.66	1

Outlier detection and replacement

- **Outlier Detection**
  - Distribution based (percentiles, quantiles etc.)
    - E.g., **99.9th percentile**: 99.9% of the temperature data points are lower than this value
  - ML based (clustering, classification etc.)
  - Empirical based
- **Outlier Replacement**
  - Closest non outlier value
  - Regression



```
UpperOutlierPerc=99.99
upper_cons = np.percentile(df_cons_we['Temperature_v1'], UpperOutlierPerc)
upper_cons
```

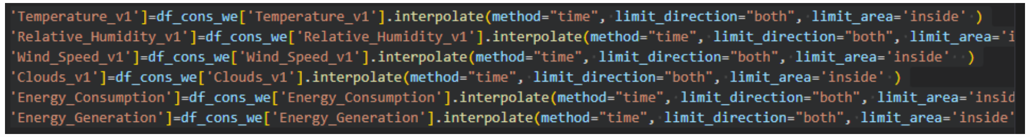
```
41.30235999996876
```

```
UpperOutlierPerc=99.99
upper_cons = np.percentile(df_cons_we['Temperature_v1'], UpperOutlierPerc)
df_cons_we['Temperature_v1'] = df_cons_we['Temperature_v1'].apply(lambda x : upper_cons if x > upper_cons else x)
```

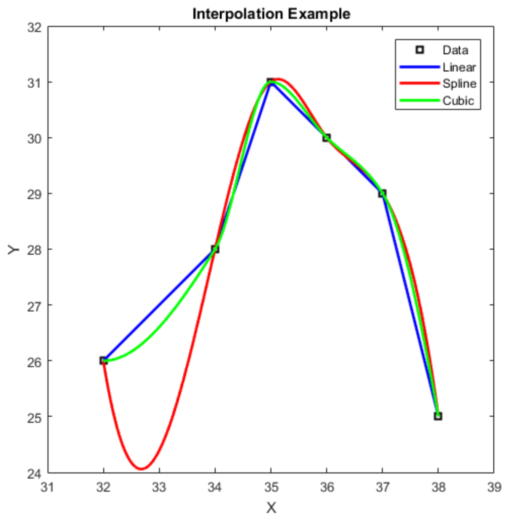
Missing values identification and filling (1/2)

Missing values in a dataset ->can lead to biased or inaccurate analyses.

- **Interpolation**: estimate missing values in a dataset by assuming that the missing values follow a smooth pattern defined by the existing data points. Prominent techniques:
  - **Linear**: A straight line between the two nearest neighboring data points
  - **Polynomial**: fits a polynomial curve to the data points ▯ estimates missing values. Degree▯ determines complexity of the curve (e.g., cubic)
  - **Spline**: divides the dataset into smaller segments and fits separate polynomial curves to each segment



```
'Temperature_v1']=df_cons_we['Temperature_v1'].interpolate(method="time", limit_direction="both", limit_area='inside' )
'Relative_Humidity_v1']=df_cons_we['Relative_Humidity_v1'].interpolate(method="time", limit_direction="both", limit_area='i
'Wind_Speed_v1']=df_cons_we['Wind_Speed_v1'].interpolate(method="time", limit_direction="both", limit_area='inside'  )
'Clouds_v1']=df_cons_we['Clouds_v1'].interpolate(method="time", limit_direction="both", limit_area='inside' )
'Energy_Consumption']=df_cons_we['Energy_Consumption'].interpolate(method="time", limit_direction="both", limit_area='insid
'Energy_Generation']=df_cons_we['Energy_Generation'].interpolate(method="time", limit_direction="both", limit_area='inside'
```



Missing values identification and filling (2/2)

- **Imputation**: can be based on statistical techniques, pattern recognition, or predictive models-> depends on nature of the data, missing data mechanism, analysis objectives.
  - **Mean/Median**: missing values are replaced with the mean or median value of the respective feature
  - **Mode**: Used for categorical variables ⍰ replaces missing values w/ most frequent value of the respective category.
  - **Backfill/Forward Fill**: propagate the last known value backward or the next known value forward to fill in missing values in time series data.
  - **Regression**: regression model built using the non-missing data points⍰ used to impute the missing values.

Normalization – Standardization

Techniques used to scale the features in a dataset to a similar range:

- **Necessary for non-tree-based models (e.g. Linear regression), Distance-Based Algorithms(KNN,SVM) & ANNs**
- **All features contribute equally to the model and to prevent features with larger values from dominating**

Normalization (MinMax)	Standardization
Rescales values to a range between 0 and 1	Centers data around the mean and scales to a standard deviation of 1
Useful when the distribution of the data is unknown or not Gaussian	Useful when the distribution of the data is Gaussian or unknown
Sensitive to outliers	Less sensitive to outliers
Retains the shape of the original distribution	Changes the shape of the original distribution
May not preserve the relationships between the data points	Preserves the relationships between the data points
Equation: (x – min)/(max – min)	Equation: (x – mean)/standard deviation

Synthesize time lagged values

In time-series data, relationship between independent variables and the predicted value can vary over time

To capture these dynamics, time lagged features are synthesized-> incorporating past values of the variables as inputs to the model

Enhances learning from historical patterns-> **improves predictive performance.**

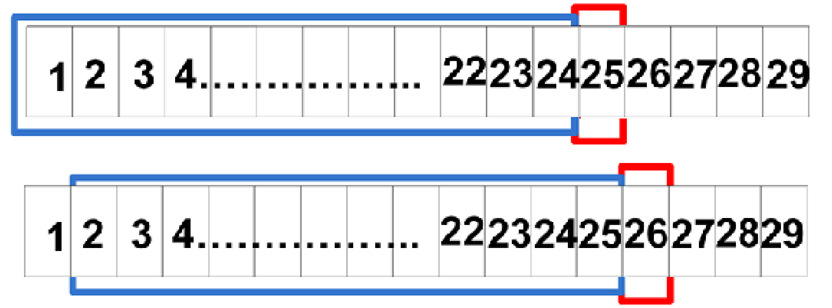
Most prominent technique: Sliding Window

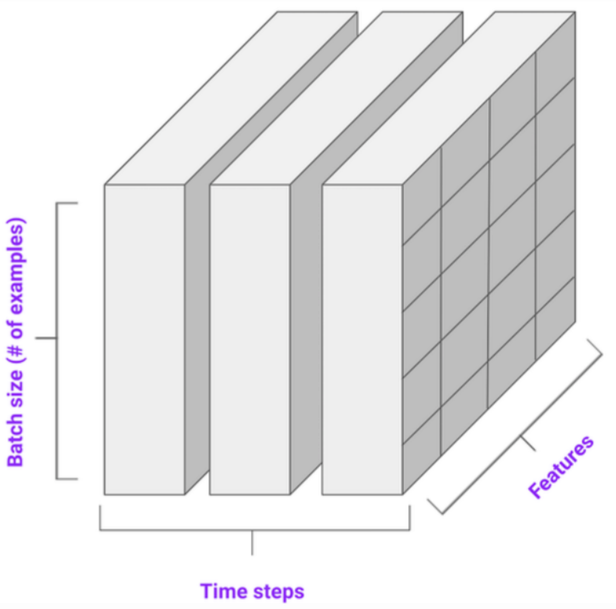
Often time-series data are shifted (24/48/96 steps) ahead based on the asset's resolution, in order to produce datasets with the information of the previous day

- 3D for LSTMs
  - **time-steps**
  - **rows**
  - **parameters**

- 2D for others

(Rows, parameters for all time-steps)





Boosting Research for a Smart and Carbon Neutral Built Environment with Digital Twins – **SmartWins**

Project Partners

ktu kaunas university of technology 1922

CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

POLITECNICO MILANO 1863

Contecht

INNOTROPE

Funded by the European Union

This project has received funding from the European Union’s Horizon research and innovation programme under grant agreement No 101078997

SmartWins