TU
جامعة الطائف
TAIF UNIVERSITY

IEEE
WESTERN SAUDI ARABIA SECTION

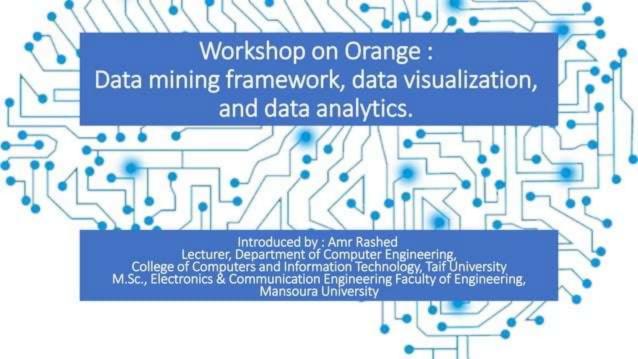# Data mining, analysis and visualization Using Python-Orange

مقدم الورشة

Eng. Amr Rashid

تاريخ الورشة

التاريخ 27/3/2021
الوقت 8:30 - 10:30 PM

# Workshop on Orange :
# Data mining framework, data visualization, and data analytics.

Introduced by : Amr Rashed
Lecturer, Department of Computer Engineering,
College of Computers and Information Technology, Taif University
M.Sc., Electronics & Communication Engineering Faculty of Engineering,
Mansoura University

# Agenda

➢ Topics:

➢ Introduction & overview

➢ Application 1:

Fault Detection for Attaining Service Continuity of Photovoltaic Power System

➢ Application 2:

Data Mining for Diagnosis of Breast Cancer in Medical Ultrasonic Images

Project 1

Project 2

**Applied Machine Learning Process**

| Define | Define the Problem |
| Prepare | Prepare Data |
| Spot | Spot Check Algorithms |
| Present | Present Results |
| Improve | Improve Results |

# Define the Problem

**Step 1: What is the problem?**

**Step 2: Why does the problem need to be solved?**

**Step 3: How would I solve the problem?**

# Common Data Preparation Tasks

- **Step 1**: Define Problem.
- **Step 2**: Prepare Data.
- **Step 3**: Evaluate Models.
- **Step 4**: Finalize Model.

## Data Cleaning

**Data cleaning** involves fixing systematic problems or errors in "*messy*" data.

Using statistics to define normal data and identify outliers.

Identifying columns that have the same value or no variance and removing them

Identifying duplicate rows of data and removing them.
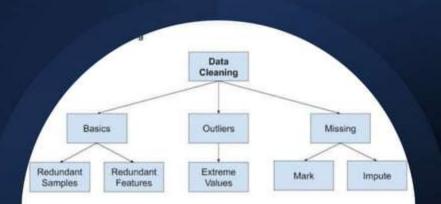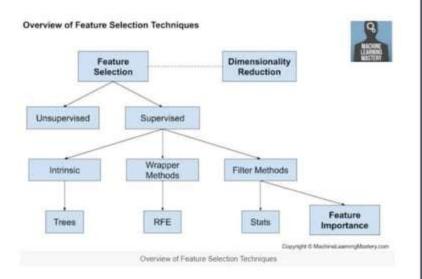
Marking empty values as missing.
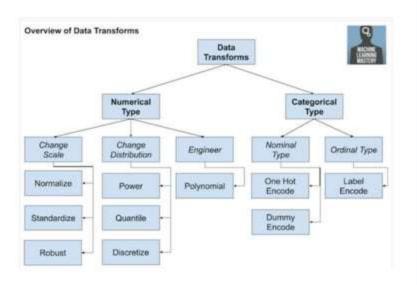
Imputing missing values using statistics or a learned model.

# Overview of data cleaning

Overview of feature selection techniques

Overview of Data Transforms

# Feature Engineering
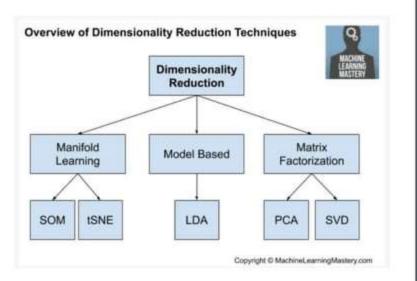
Feature engineering is the process of creating new input variables from the available data.

Adding a Boolean flag variable for some state.

Adding a group or global summary statistic, such as a mean.
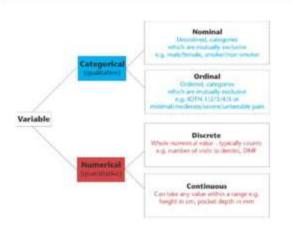
Adding new variables for each component of a compound variable, such as a date-time.
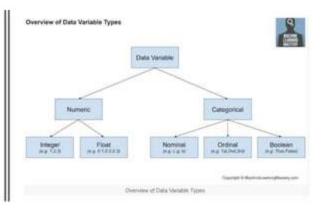
Polynomial Transform: Create copies of numerical input variables that are raised to a power(raising them to a power or multiplied with other input variables).

Overview of Dimensionality Reduction Techniques

# Overview of data types



Variable

Categorical (qualitative)

**Nominal**
Unordered, categories
which are mutually exclusive
e.g. male/female, smoker/non-smoker

**Ordinal**
Ordered, categories
which are mutually exclusive
e.g. IOTN 1/2/3/4/5 or
minimal/moderate/severe/unbearable pain

Numerical (quantitative)

**Discrete**
Whole numerical value - typically counts
e.g. number of visits to dentist, DMF

**Continuous**
Can take any value within a range e.g.
height in cm, pocket depth in mm

Overview of Data Variable Types

Data Variable

Numeric

Integer
(e.g. 1,2,3)

Float
(e.g. 0.1,0.2,0.3)

Categorical

Nominal
(e.g. c,g, b)

Ordinal
(e.g. 1st,2nd,3rd)

Boolean
(e.g. True,False)

Overview of Data Variable Types

# Orange installation

# Cont.



orange

Screenshots    Workflows    Download    Blog    Docs    Workshops

Windows          macOS          Linux / Source

Download the latest version for Windows

Download Orange 3.28.0

# Cont.

Standalone installer (default)

Orange3-3.28.0-Miniconda-x86_64.exe (64 bit)
Can be used without administrative priviledges.

Portable Orange

Orange3-3.28.0.zip

Anaconda

If you are using python provided by Anaconda distribution, you are almost ready to go. Add conda-forge to the list of channels you can install packages from

```
conda config --add channels conda-forge
```

and run

```
conda install orange3
```

Pip
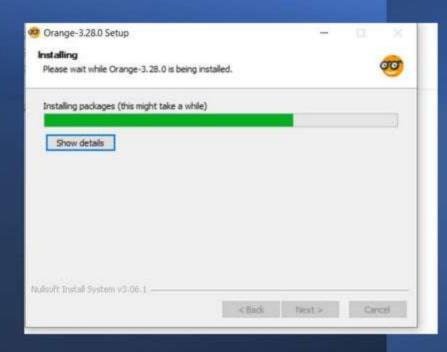
Orange can also be installed from the Python Package Index:

```
pip install orange3
```

**Today**

**Orange3-3.28.0-Miniconda-x86_64.exe**

https://download.biolab.si/download/files/Orange3-3.28.0-Miniconda-x86_64.exe

376 KB/s - 29.6 MB of 414 MB, 17 mins left

Pause     Cancel

# Orange-3.28.0 Setup

□ ‒ ✕

## Installing

Please wait while Orange-3.28.0 is being installed.

Installing packages (this might take a while)

Show details

Nullsoft Install System v3.06.1

< Back    Next >    Cancel

# Create Dataset for the First Project (250kw grid connected PV array)
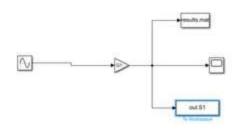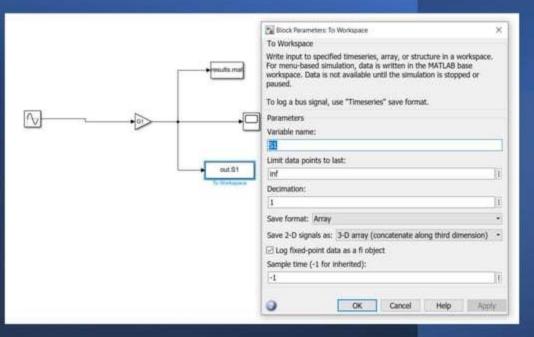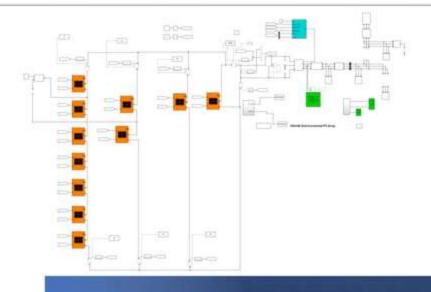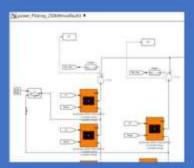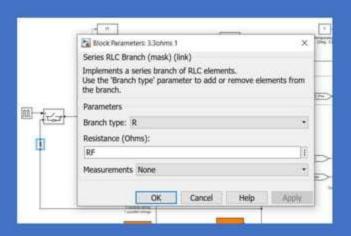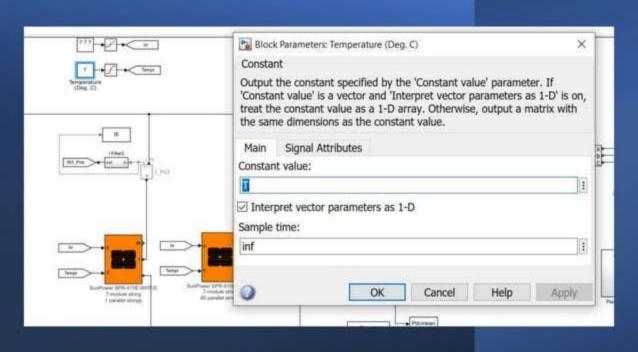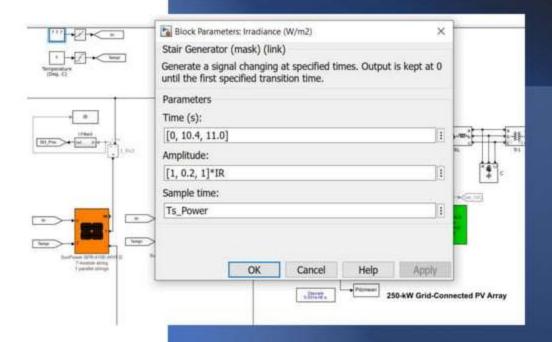
# MATLAB Script

```matlab
1 -    clc;clear;close all
2 -  □ for i=1:5
3 -    G1=randi([1,10],1);
4      %simulation
5      a=sim('E:\FE project\all codes\new1','SimulationMode','normal');
6      %extract data generated by simulink model
7 -    S1 = a.get('S1');
8 -    m1=mean(S1)
9 -  └ end
10
```

Command Window

m1 =

    2.3837

m1 =

    6.4718

m1 =

    1.7437

m1 =

    1.2849

m1 =

    1.7437

Block Parameters: Irradiance (W/m2)

Stair Generator (mask) (link)

Generate a signal changing at specified times. Output is kept at 0 until the first specified transition time.

Parameters

Time (s):

[0, 10.4, 11.0]

Amplitude:

[1, 0.2, 1]*IR

Sample time:

Ts_Power

OK    Cancel    Help    Apply

250-kW Grid-Connected PV Array

Block Parameters: To Workspace

**To Workspace**

Write input to specified timeseries, array, or structure in a workspace. For menu-based simulation, data is written in the MATLAB base workspace. Data is not available until the simulation is stopped or paused.

To log a bus signal, use "Timeseries" save format.

Parameters

Variable name:

simout

Limit data points to last:

inf

Decimation:

1

Save format: Array

Save 2-D signals as: 2-D array (concatenate along first dimension)

☑ Log fixed-point data as a fi object

Sample time (-1 for inherited):

-1

OK    Cancel    Help    Apply

```matlab
solar3.m    ×    aas.m    ×    +

2 -    clc;clear;close all ;
3 -    format short
4      %uiopen('E:\FE project\all codes\matlab simulink\power_PVarray_250kWmod.slx',1)
5      %sim
6      %define model constant
7 -    simout1=zeros(1,30);
8 - ☐ for i=1:2%10:50
9 -        disp(i)
10       %Ir=0:1000
11       %rf=0.001:2000ohm
12 -      T=randi([10,35],1);%1-var
13 -      IR=randi([100,1000],1);%2-var
14 -      LOC=26;%3-var
15       %LOC=0;%normal condition
16 -      Rf=randi([1,2000],1);%4-var
17       %RF=10000;%normal condition
18       %tsample=0.0005;
19 -      tfinal=0.4;%sim stop time %5-var
10       %samplet=Auto;%sample time 2*0.005
11 -      timef=0.2;%fault time
12       %simulation
13 -      a=sim('E:\FE project\all codes\matlab simulink\power_PVarray_250kWmodfault3','Simulation
14       %extract data generated by simulink model
15 -      I1 = a.get('I1');
16 -      I2 = a.get('I2');
17 -      I3 = a.get('I3');
18 -      I4 = a.get('I4');
19 -      I5 = a.get('I5');
```

| U | V | W | X | Y | Z | AA | AB | AC | AD |
|---|---|---|---|---|---|---|---|---|---|
| Vdcmean1 | Vdcmax1 | Vdcmin1 | Pdcmean1 | IR | T | LOC | RF | simstop | tir class |
| 483.2946 | 483.6525 | 482.3762 | 58.59916 | 242 | 35 | 26 | 1942 | 0.4 | 3 |
| 494.6765 | 494.9698 | 494.4559 | 133.0941 | 537 | 34 | 26 | 1601 | 0.4 | 3 |

## Test and Score

### Sampling

- ● Cross validation
  - Number of folds: 5
  - ☑ Stratified
- ○ Cross validation by feature
- ○ Random sampling
  - Repeat train/test: 10
  - Training set size: 66 %
  - ☑ Stratified
- ○ Leave one out
- ○ Test on train data

### Evaluation Results

| Model | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|
| kNN | 0.685 | 0.420 | 0.435 | 0.471 | 0.420 |
| Tree | 0.999 | 0.998 | 0.998 | 0.998 | 0.998 |
| SVM | 0.801 | 0.522 | 0.470 | 0.490 | 0.522 |
| Random Forest | 1.000 | 0.998 | 0.998 | 0.998 | 0.998 |
| Neural Network | 0.996 | 0.963 | 0.964 | 0.964 | 0.963 |
| Naïve Bayes | 0.992 | 0.920 | 0.920 | 0.931 | 0.920 |
| CN2 rule inducer | 0.991 | 0.958 | 0.958 | 0.959 | 0.958 |
| AdaBoost | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Train time [s]
Test time [s]
✓ AUC
✓ CA
✓ F1
✓ Precision
✓ Recall
LogLoss
Specificity
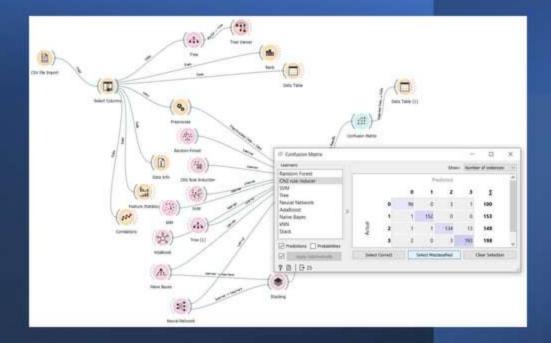
## Evaluation Results
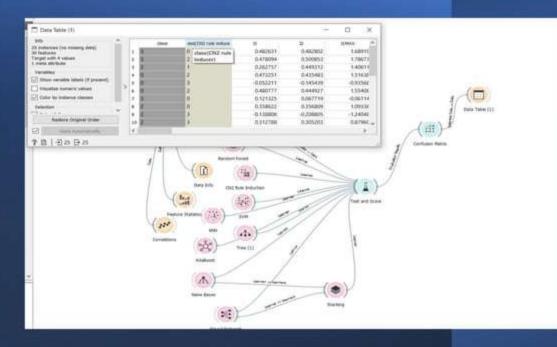
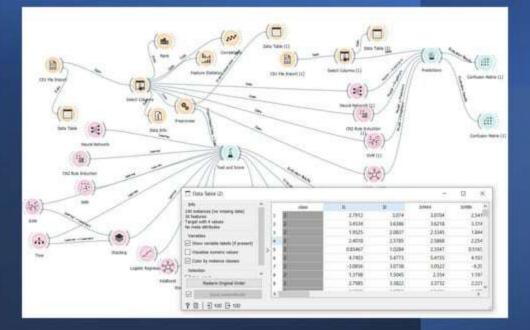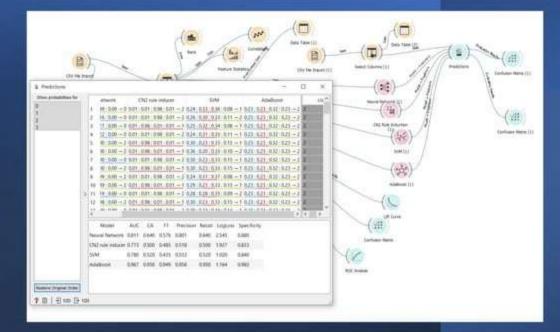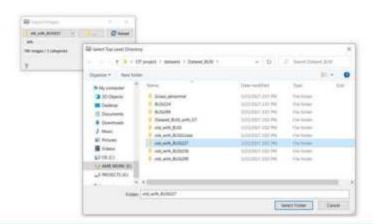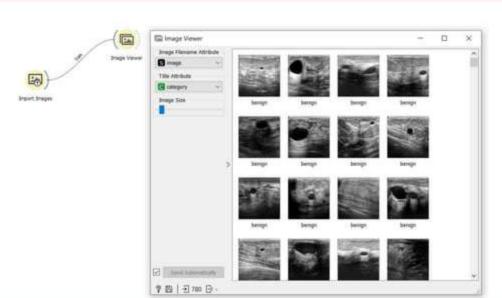| Model | Train time [s] | Test time [s] | AUC | CA | F1 | Precision | Recall | LogLoss | Specificity |
|---|---|---|---|---|---|---|---|---|---|
| AdaBoost | 0.084 | 0.020 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.000 | 1.000 |
| Random Forest | 0.111 | 0.029 | 1.000 | 0.998 | 0.998 | 0.998 | 0.998 | 0.073 | 1.000 |
| Tree | 0.097 | 0.001 | 0.999 | 0.998 | 0.998 | 0.998 | 0.998 | 0.058 | 0.999 |
| Neural Network | 3.269 | 0.014 | 0.996 | 0.963 | 0.964 | 0.964 | 0.963 | 0.293 | 0.991 |
| CN2 rule inducer | 13.204 | 0.037 | 0.991 | 0.958 | 0.958 | 0.959 | 0.958 | 0.213 | 0.983 |
| Naive Bayes | 0.154 | 0.023 | 0.992 | 0.920 | 0.920 | 0.931 | 0.920 | 0.249 | 0.979 |
| SVM | 0.457 | 0.054 | 0.801 | 0.522 | 0.470 | 0.490 | 0.522 | 1.057 | 0.798 |
| kNN | 0.075 | 0.138 | 0.685 | 0.420 | 0.435 | 0.471 | 0.420 | 5.483 | 0.820 |

# Project 2

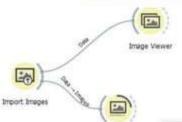- Data Mining for Diagnosis of Breast Cancer in Medical Ultrasonic Images