Artificial Intelligence for Medical Data with Python

10 SAMPLE SLIDES

3 th session – Classification Algorithms, Regression and Time Series

UNIVERSITY OF THE AEGEAN
SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATION AND COMMUNICATION SYSTEMS ENGINEERING

Presenter: Panagiotis Symeonidis

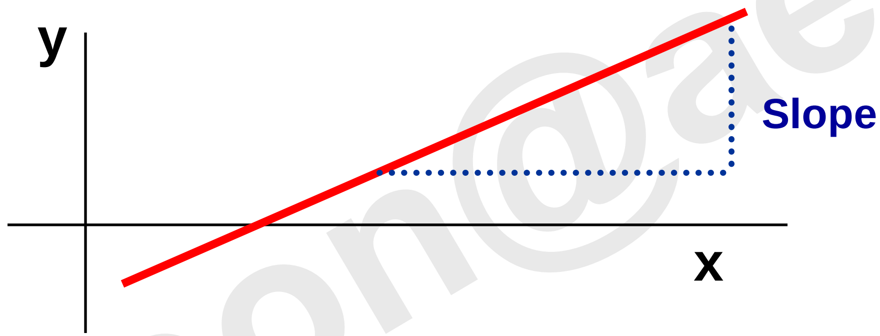
Associate Professor

http://panagiotissymeonidis.com

psymeon@aegean.gr

Simple linear regression

- Relation between 2 continuous variables (SBP and age)



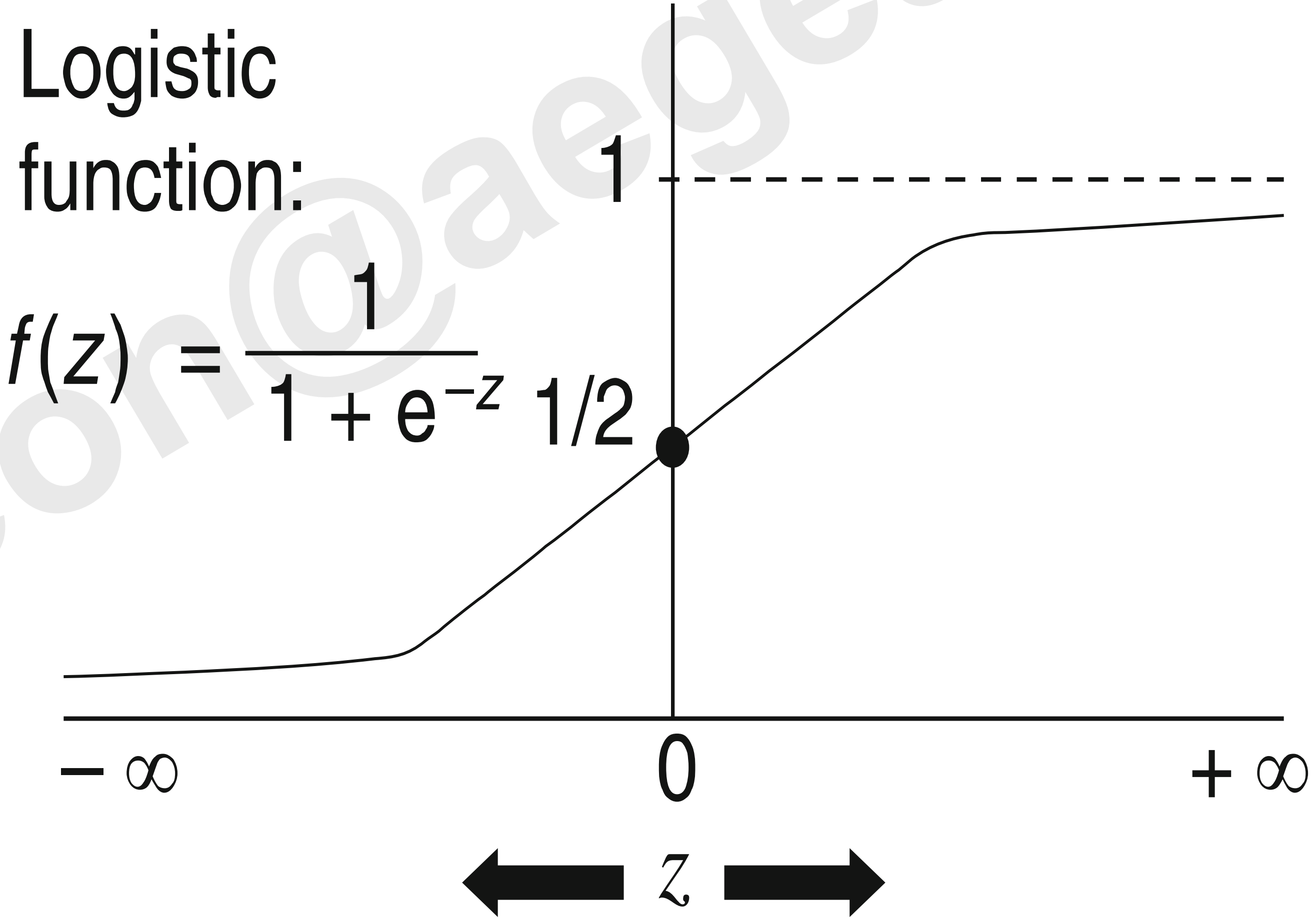
$$y = \alpha + \beta_1 x_1$$

- α coefficient : The intercept of the line, which is the value of y when x=0 It represents the point where the line crosses the y-axis.

- β_1 coefficient : The slope of the line.
  - Amount by which y changes on average when x changes by one unit
  - It indicates how steep the line is.

Visualization of the Logistic function

Probability of the disease ranges in [0,1]



Coronary Heart Disease (CHD) Prediction Example

ABOUT PATIENTS' DATA

THE FOLLOWING FOUR ATTRIBUTES ARE COLLECTED

probability of CHD ← •Dependent variable

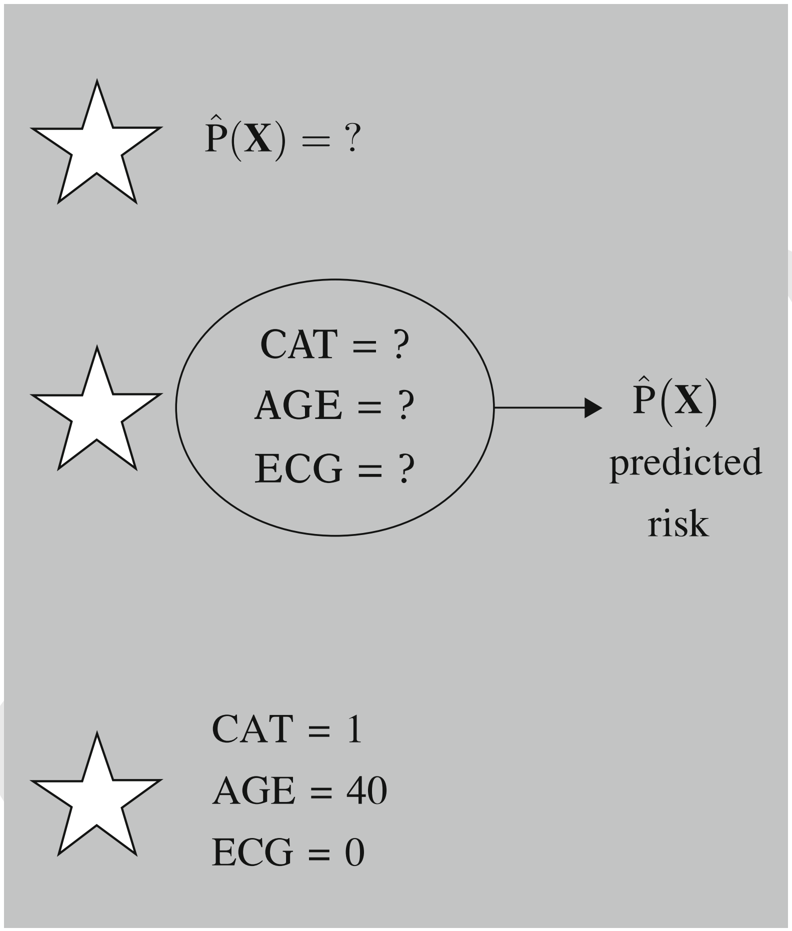
catecholamine level (0=low, 1=high) ← •Independent variable

ECG (0=normal,1=abnormal) ← •Independent variable

age (in years) ← •Independent variable

Test the predicting power of the trained Model

Let s assume a patient with the following clinical status



Predicting the Risk for Coronary Heart Disease

$$\hat{P}(\mathbf{X})$$

$$= \frac{1}{1 + e^{-[-3.911 + 0.652(1) + 0.029(40) + 0.342(0)]}}$$

$$= \frac{1}{1 + e^{-(-2.101)}}$$

$$= \frac{1}{1 + 8.173}$$

$$= 0.1090, \text{ i.e., } \text{risk} \simeq 11\%$$

Time Series

- A time series is a set of successive values of a characteristic over a period of time.

- Given an attribute A, a time series is the set of N observed pairs:

{(t1, a1), (t2, a2), ..., (tn, an)},

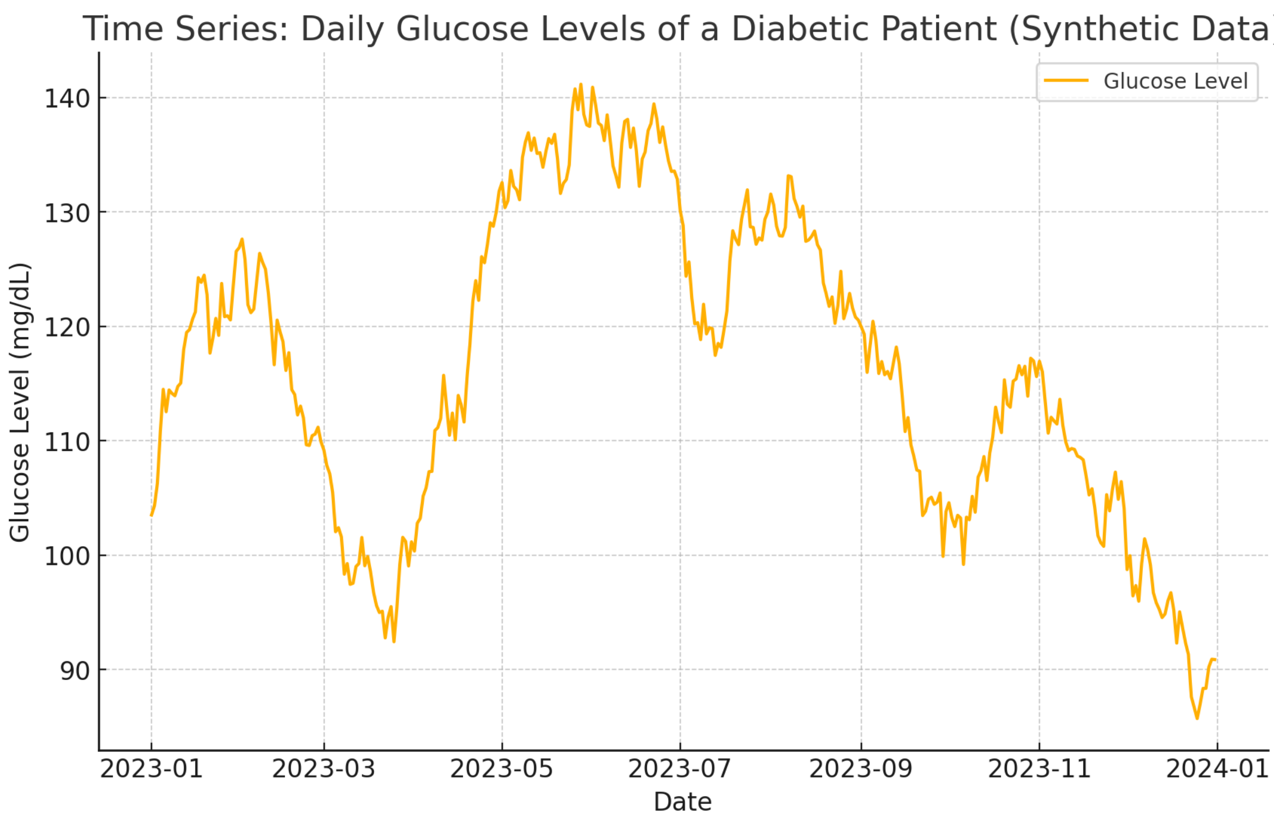
where

T = {1, 2, ..., n} is a sequence of successive time points for an observed value of the attribute A at time n.

Example of a time series

The collection of daily average values of glucose level of a patient for year constitutes a time series.

Date	Glucose Level
2023-01-01	104
2023-01-31	127
2023-03-02	108
2023-04-01	100
2023-05-01	133
2023-05-31	137
2023-06-30	133
2023-07-30	129
2023-08-29	122
2023-09-28	105
2023-10-28	114
2023-11-27	107
2023-12-27	88



Preparing Data for Random Forest Regressor

1. **Generate Synthetic Data**: We simulate daily glucose levels for a diabetic patient over one year using a random walk.

```python
dates = pd.date_range('2023-01-01', periods=365)
glucose_levels = np.cumsum(np.random.randn(365) * 2) + 100
data = pd.DataFrame({'Date': dates, 'Glucose Level': glucose_levels})
```

2. **Lag Features**: Create lagged features to use past glucose levels as predictors.

```python
data['Lag_1'] = data['Glucose Level'].shift(1)
data['Lag_2'] = data['Glucose Level'].shift(2)
data['Lag_3'] = data['Glucose Level'].shift(3)
data.dropna(inplace=True)
```

Lagging Example

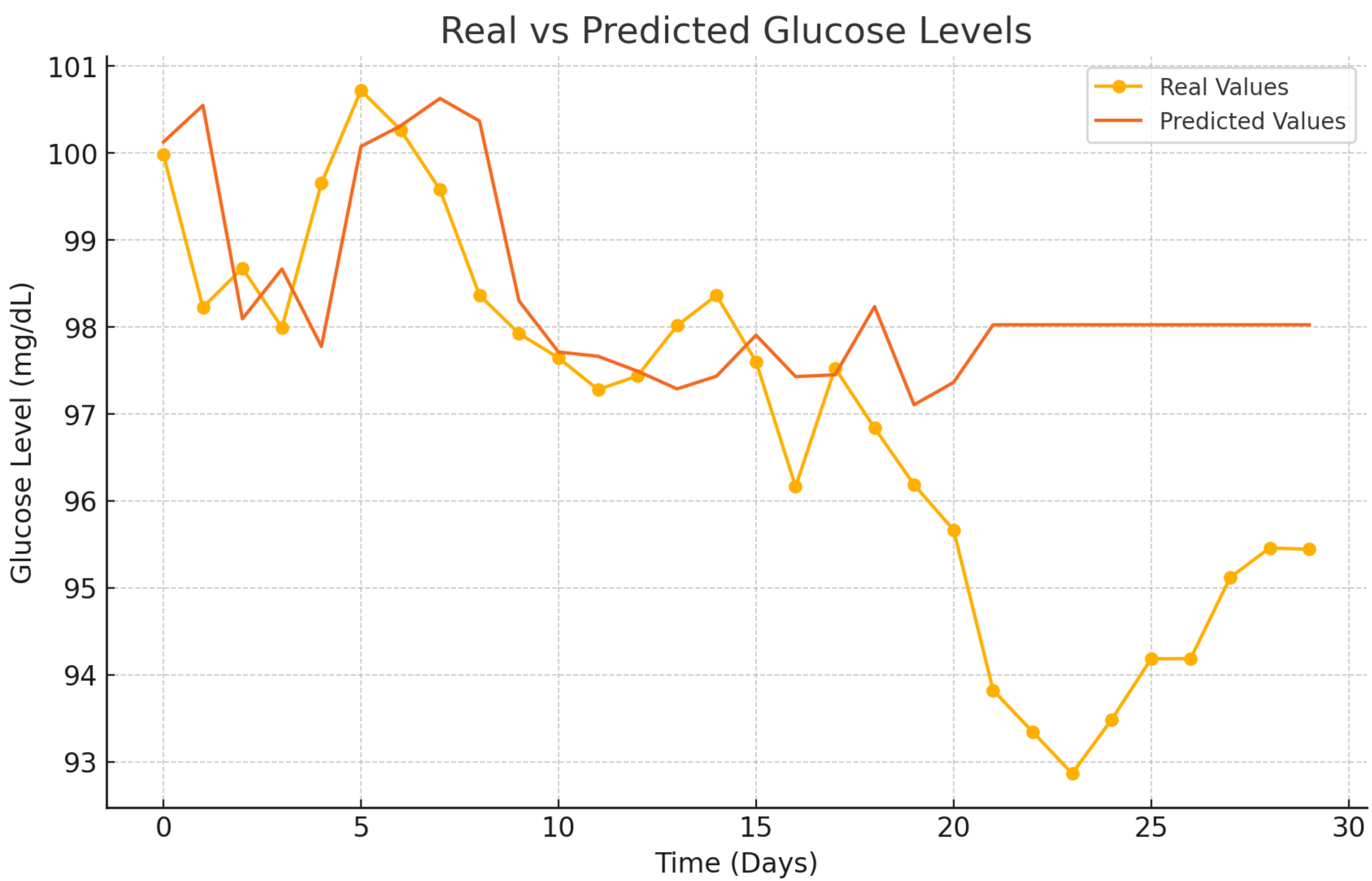
Original Glucose Data

	Date	Glucose Level
1	2023-01-01	100
2	2023-01-02	105
3	2023-01-03	102
4	2023-01-04	110
5	2023-01-05	108

```
data['Lag_1'] = data['Glucose Level'].shift(1)
data['Lag_2'] = data['Glucose Level'].shift(2)
data['Lag_3'] = data['Glucose Level'].shift(3)
```

Transformed Glucose Data With Lags

	Date	Glucose Level	Lag_1	Lag_2
1	2023-01-01	100		
2	2023-01-02	105	100.0	
3	2023-01-03	102	105.0	100.0
4	2023-01-04	110	102.0	105.0
5	2023-01-05	108	110.0	102.0


Real vs Predicted Glucose Levels
Real Values
Predicted Values
Glucose Level (mg/dL)
Time (Days)
101
100
99
98
97
96
95
94
93
0
5
10
15
20
25
30