

Machine Learning for Renewable Energy Forecasting

19.03.2026

Shiliang Zhang
University of Oslo, Norway

Industry Invited Talk Today



Kjetil Thøgersen

Data Scientist, StatKraft

Weather and climate services for renewable energy production

Statkraft

Statkraft: Europe's largest producer of renewable energy, developing, building and operating hydro, wind, solar, gas and biomass assets.

Learning Objectives

Throughout this lecture, it is aimed for you to be able to:



Forecasting

Understand forecasting uncertainty and why we need to forecast renewable energy production



Machine Learning

Learn basic machine learning techniques in plain language, programming language, and mathematical language



Applications

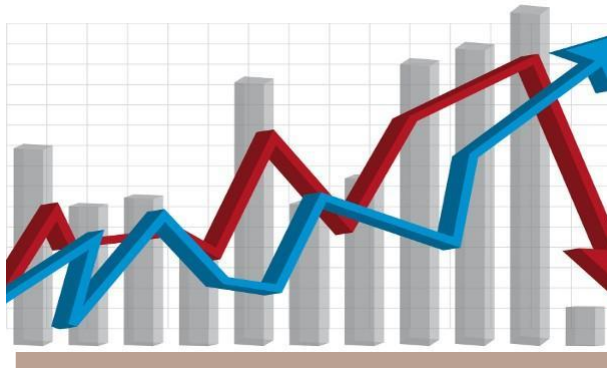
Apply different machine learning techniques for wind energy forecasting

Outline



Wind Energy

Concept
Example
Uncertainty



Forecasting

Wind power & wind speed curve; Forecasting definition



Machine Learning

Linear Regression;
K-Nearest Neighbor;
Support Vector Regression

01

WIND ENERGY: CONCEPT

Renewable Energy Forecasting



Solar energy

- Depend on sun, cloud, and weather in general



Wind energy

- Depend on wind and weather in general

Main focus in this lecture



Wave energy

- Wave energy converter are floating on the ocean surface waves. Wave power is the energy from ocean surface waves.

Offshore & Onshore

Wind turbines can be installed:

- **Onshore: on land**
  - **Cheaper installation**
  - **Cheaper integration**
  - **Cheaper maintenance**
- **Offshore: on sea**
  - **Less obstruction**
  - **Higher and more steady wind speed**

Q: what is the advantage of offshore wind farm?

Onshore wind farm

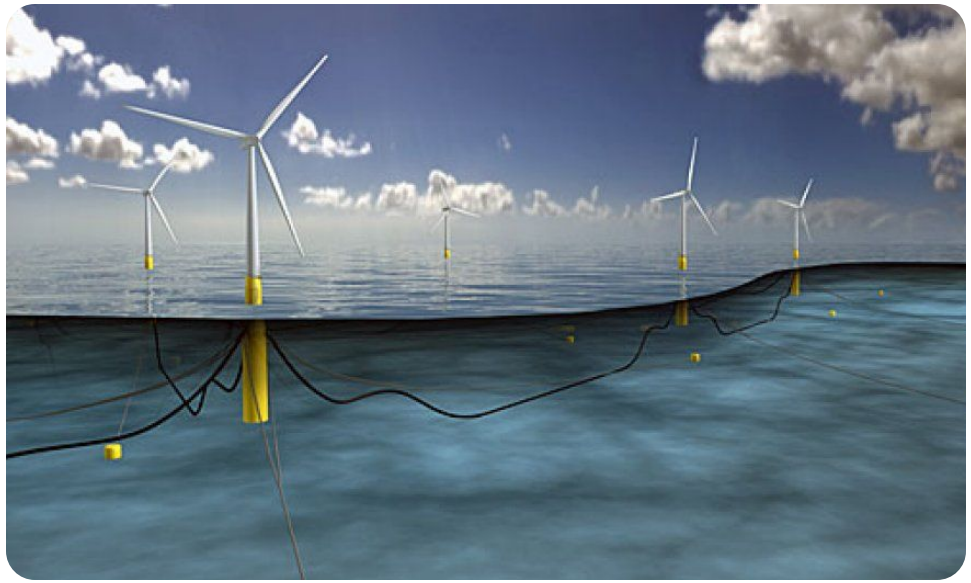


Offshore wind farm



You may watch 2-minute video at: http://www.youtube.com/watch?v=tsZITSeQFR0

Equinor (original name: Statoil) has great interest in offshore wind power



Floating wind turbines at Hywind SCOTLAND



ARKONA, Germany

More information:
https://www.equinor.com/en/what-we-do/wind.html

Wind forecasting and wind energy forecasting are important

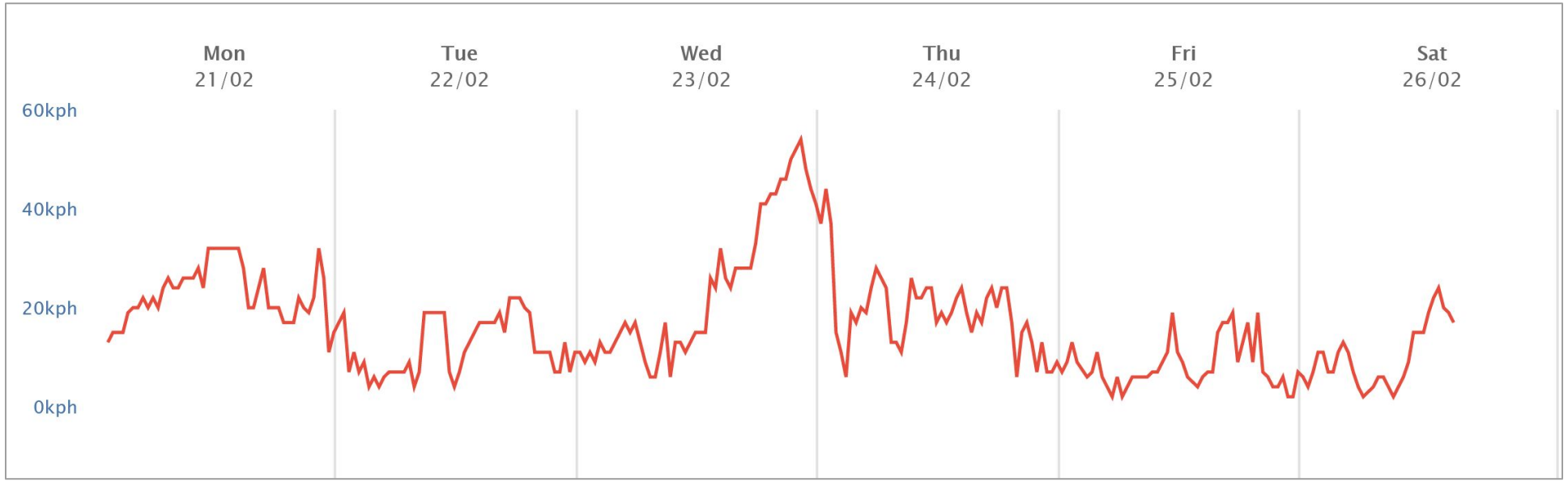


A £2million, 100metre-tall wind turbine caught fire in hurricane force wind at Ardrossan, North Ayrshire, Scotland (2011).
http://www.dailymail.co.uk/news/article-2071633

- **Energy cost of intermittent wind:** unforecasted spin-regulation waste 2.5-7.5% of total energy
- **System reliability:** unforecasted ramp events may compromise reliability
- **Effective power grid management:** forecasts are essential for effective grid management with high wind penetrations (>5%)
- **Economic value:** good forecasts may lead to high economic value

Wind speed forecasting is very challenging

Wind speed in Gardermoen (Oslo Airport) during February 21-26, 2022



- **The key problem is the intermittency. Changes in wind speed will result in changes in wind power production.**

Wind energy forecasting is challenging

- Small pressure gradients over large distances: hard to forecast accurately
- Local topography can have strong influence: not in standard weather models
- Turbulent & chaotic processes are important: even harder to forecast
- Abnormal plants activities: malfunctions, downtime and sub-optimality
- Wind-power curves are highly nonlinear: small errors in wind lead to big errors in power

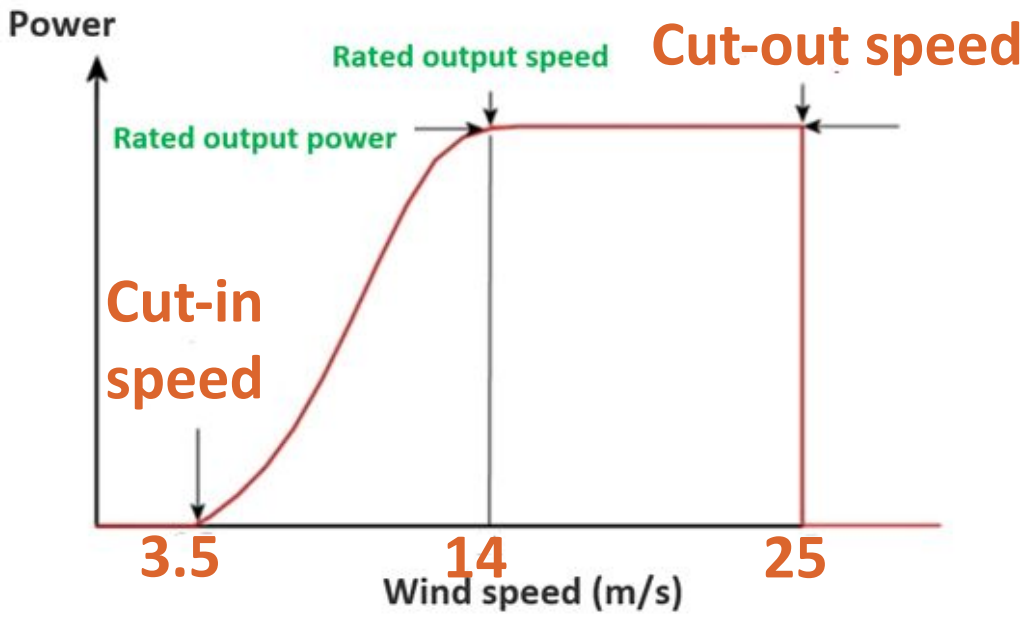


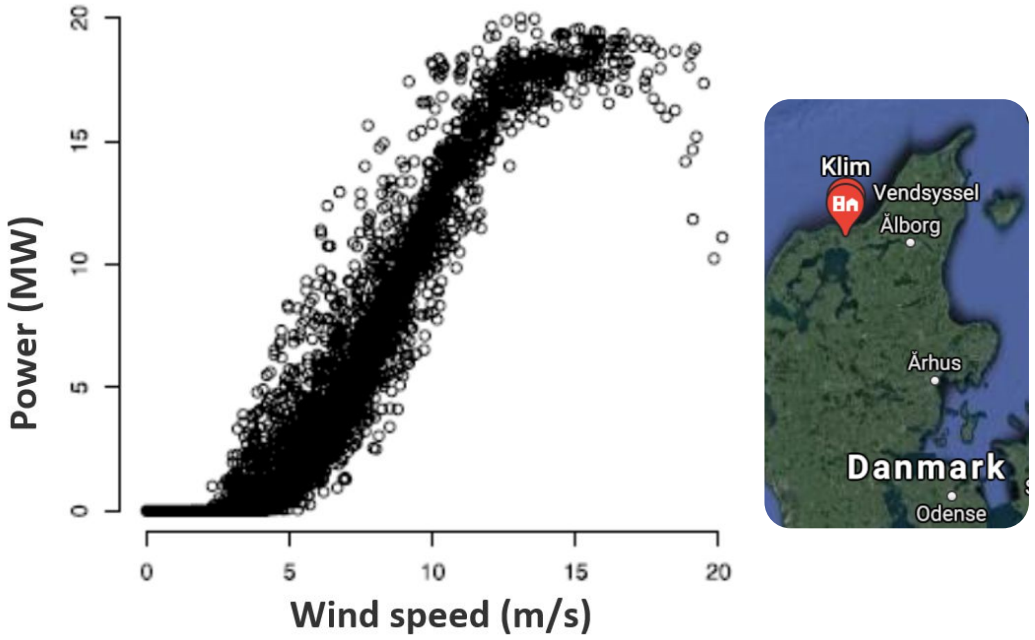


02

FORECASTING PROBLEM

Wind Power & Speed Curve

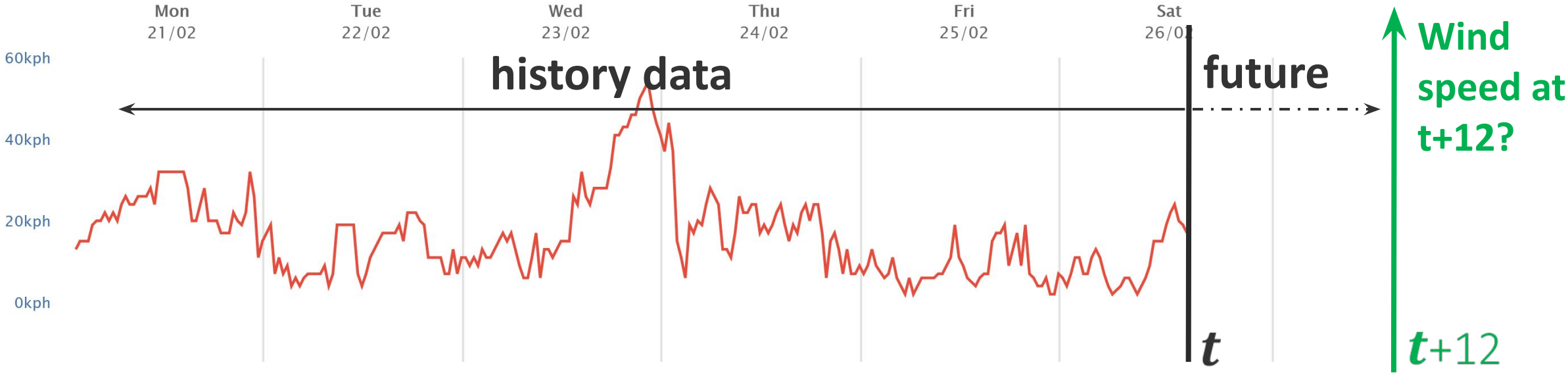




- Wind power depends on wind speed. Predicting wind speed can help us predict wind power.
- A minimum cut-in speed is needed to start power generation

- Denmark Klim wind farm: example empirical power curve for the wind farm over a 6-month period in 2002, based on hourly measurements of wind speed and corresponding power output.

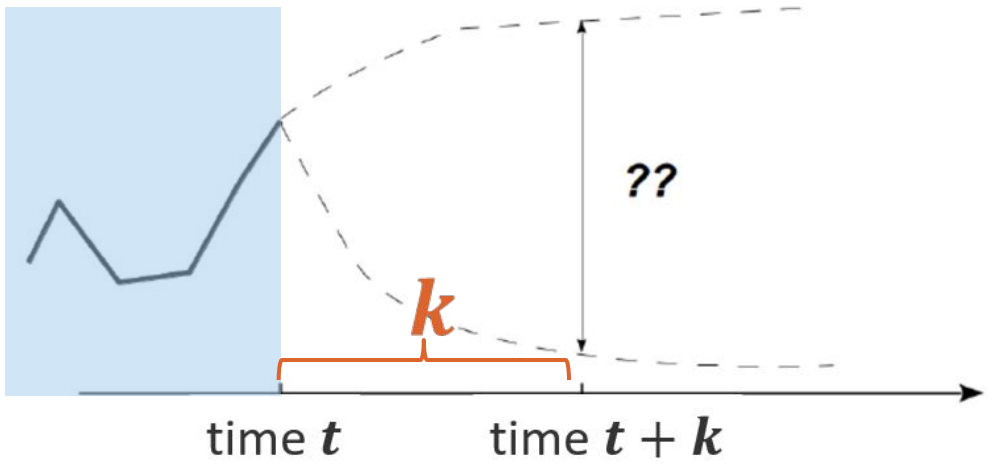
Forecasting – wind speed in Oslo Airport as an example



Wind speed in Gardermoen (Oslo Airport) during February 21-26, 2022

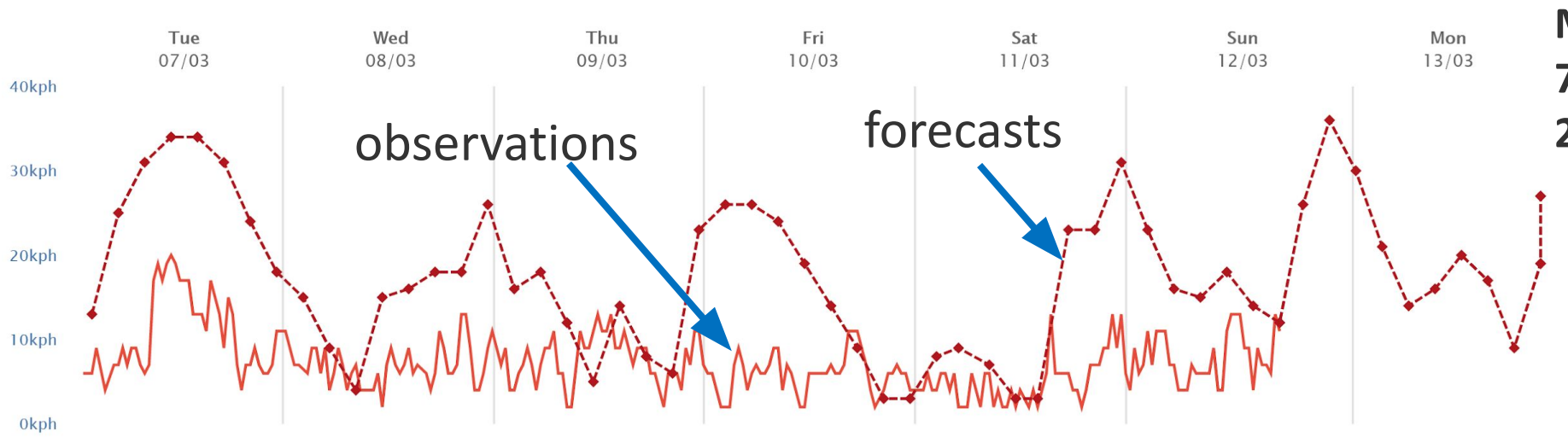
- A forecast: an estimate of wind speed at time (t +12) conditional to all history data up to time t

Forecasting definition



- The practical setup: We are at time t (e.g., at 9am) and have all information up to time t ; and interested in what will happen at time $t + k$ (e.g., at 11am)
- k : referred to as the lead time
- Y_{t+k} : the random variable "power generation at time $t + k$ "
- $\widehat{Y}_{t+k|t}$: the conditional expectation of power generation at time $t + k$, conditional to all information up to time t

Forecasting definition – mathematical model



Wind speed observations and forecasts in Gardermoen (Oslo Airport) during March 7-13, 2023

$$\hat{Y}_{t+k|t} = f(Y_{t+k}|Y_t, w_t, T, P, S)$$

- Y_t: history data up to time t
- Y_{t+k}: power generation at time t+k
- $\hat{Y}_{t+k|t}$: predicted value of power generation at time t+k
- w_t: wind speed at time t
- T: temperature
- P: pressure
- S: other parameters (e.g. wind direction)

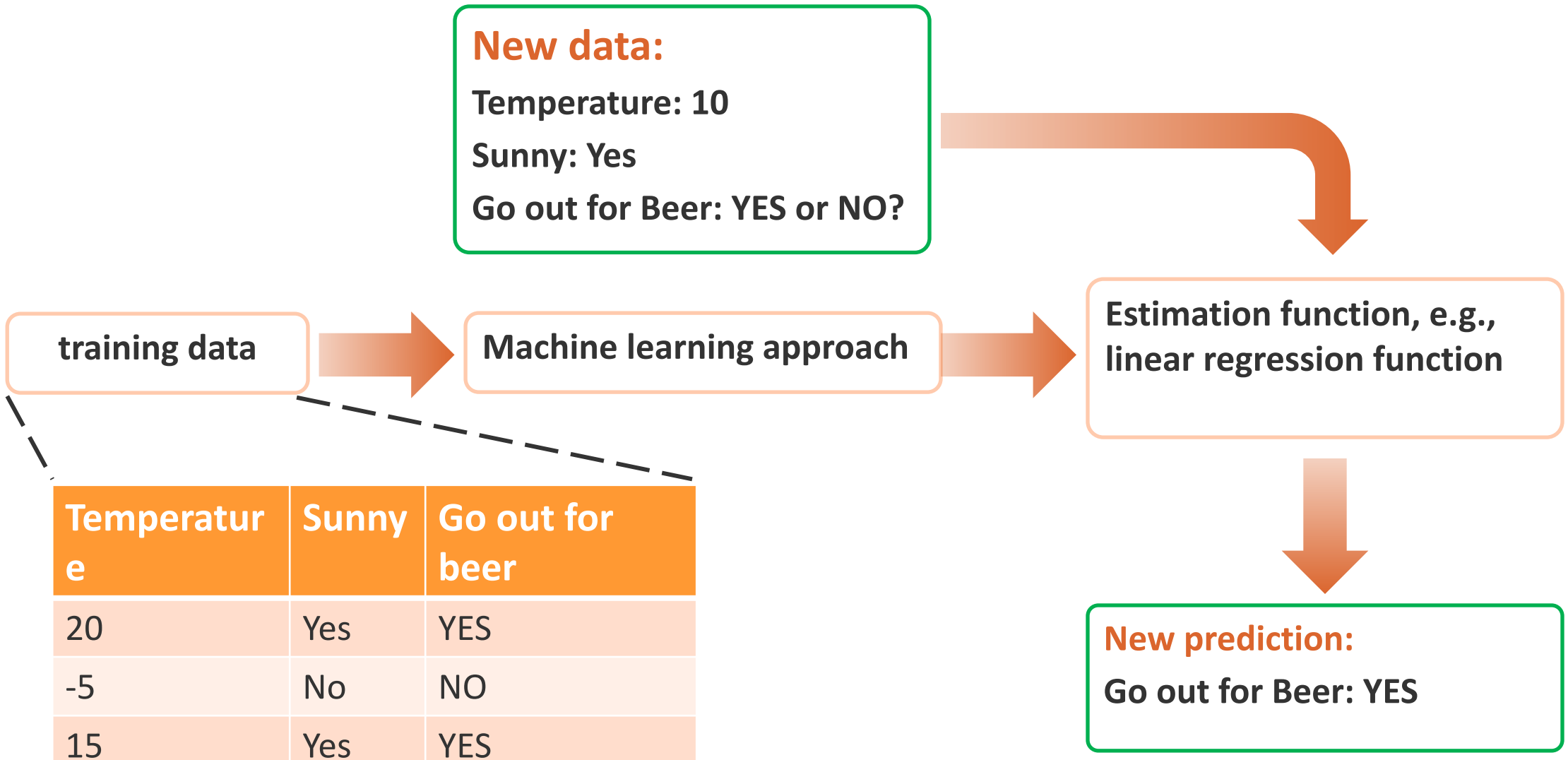
Different lead time *k* for time-scale classification for wind energy forecasting

Time-scale	Range	Applications
Very short-term	Few minute to 1 hour ahead	• Electricity market clearing • Real-time grid operations • Regulation actions
Short-term	1 hour to several hours ahead	• Economic load dispatch planning • Load reasonable decisions • Operational security in electricity market
Medium-term	Several hours to 1 week ahead	• Unit commitment decisions • Reserve requirement decisions • Generator online/offline decisions
Long-term	1 week to 1 year or more ahead	• Maintenance planning • Operation management • Optimal operating cost • Feasibility study for design of the wind farm

03

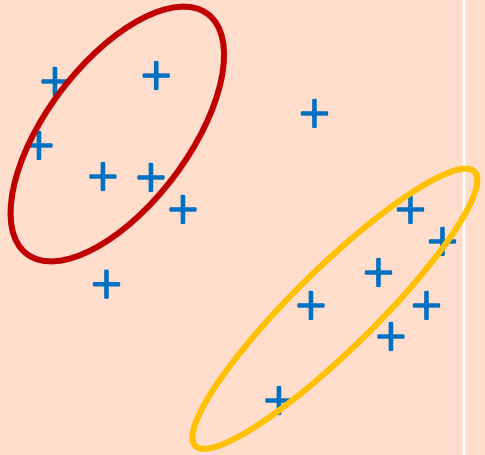
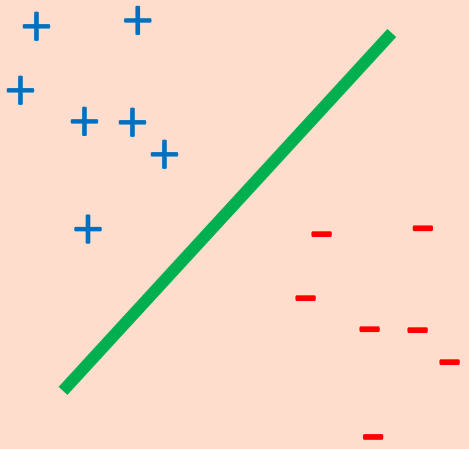
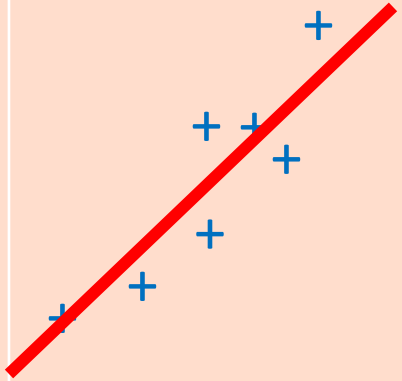
MACHINE LEARNING REGRESSION TECHNIQUES

What is machine learning? – a typical machine learning process

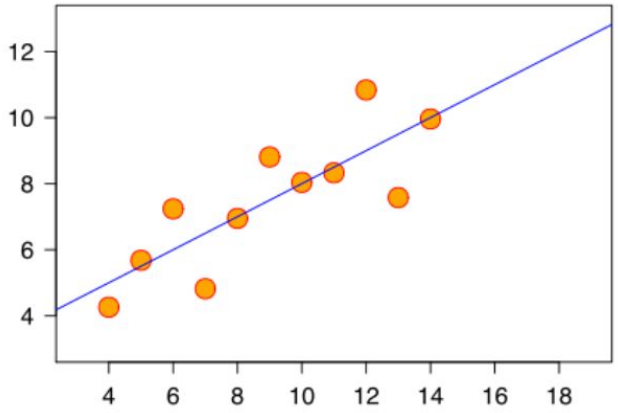


Temperature	Sunny	Go out for beer
20	Yes	YES
-5	No	NO
15	Yes	YES
2	Yes	NO

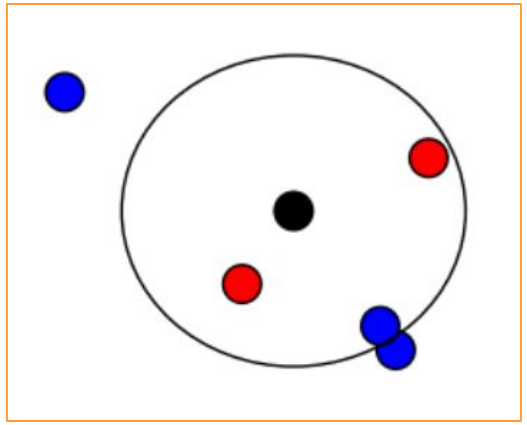
Machine learning overview

Clustering	Classification	Regression (this lecture)
		
K-means	• Decision tree • Linear Discriminant Analysis • Neural Networks • Support Vector Machines	• Linear regression • k Nearest Neighbors • Support Vector Regression
Group data based on their characteristics	Separate data based on their labels	Find a model that can explain the output given the input

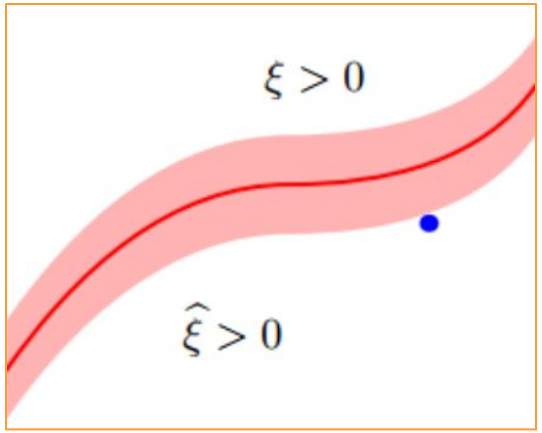
Three Regression Techniques



- Linear Regression



- K-Nearest Neighbor Regression (kNN)

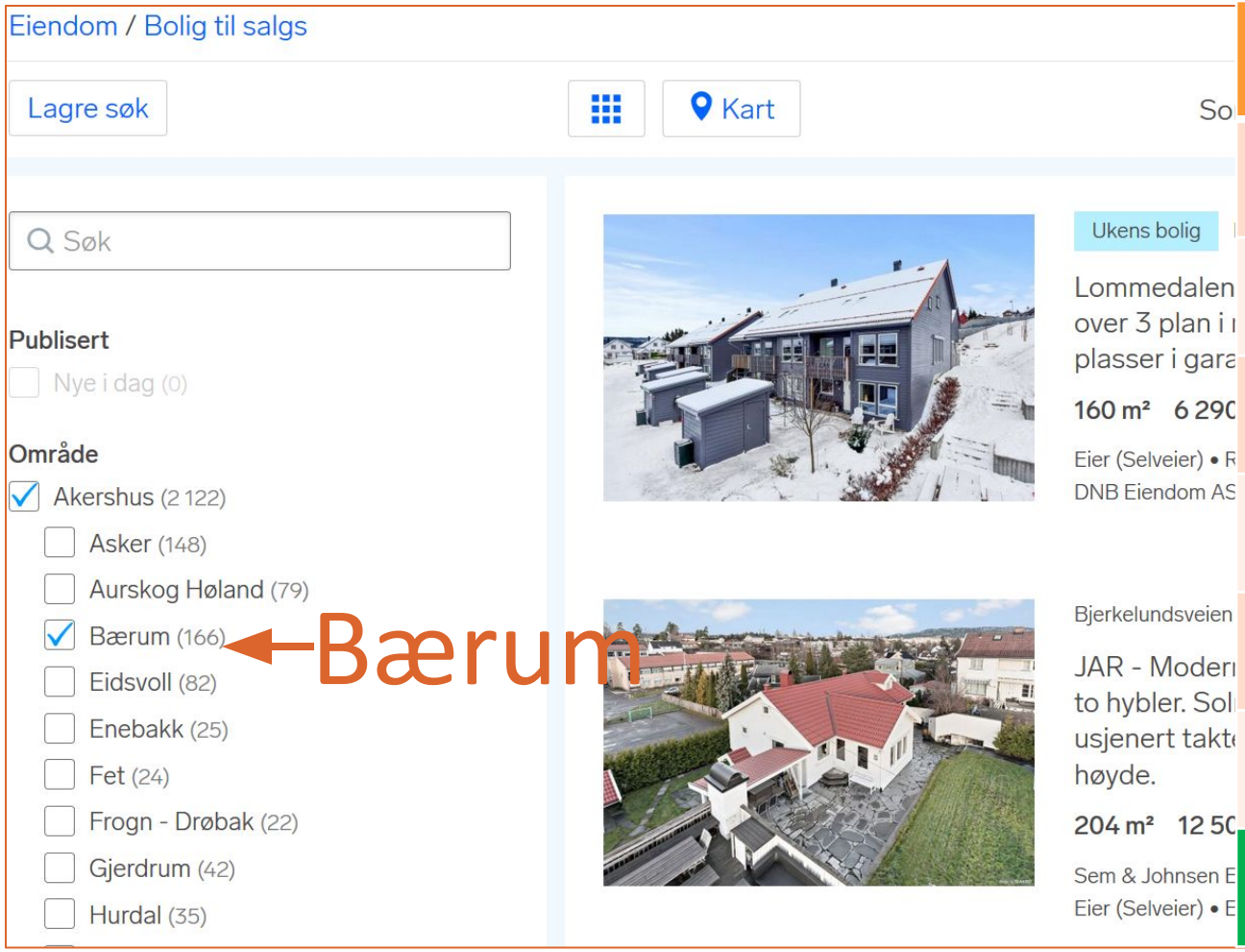


- Support Vector Regression (SVR)

03. 01

LINEAR REGRESSION AND ITS APPLICATION FOR WIND ENERGY FORECASTING

Linear Regression Model - real estate example



←Bærum

Area (m^2)	Price (Million Kr)
177	10.8
204	12.5
81	4
150	6.9
106	9.65
63	2.95
132	?

- Q: what is the price for the new house with area 132 m^2?

Linear regression model – main idea

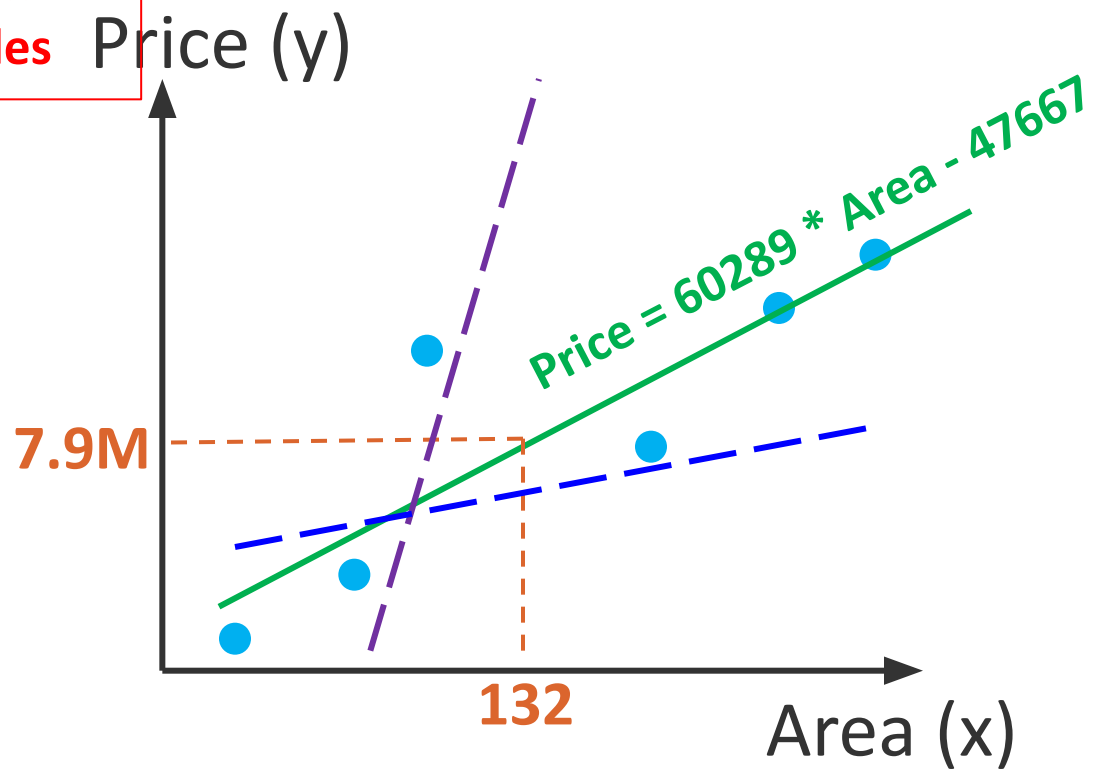
Correction here: This is w (double u), not "omega". This correction applies to all the following slides

- 6 points are plotted
- Find a straight line that models price as a linear function of living area, i.e., $\mathbf{y} = \boldsymbol{wx} + \boldsymbol{b}$. For example:

Price = 60289 * Area - 47667

- Then, we can calculate the predicted price for the new house with area $132m^2$:

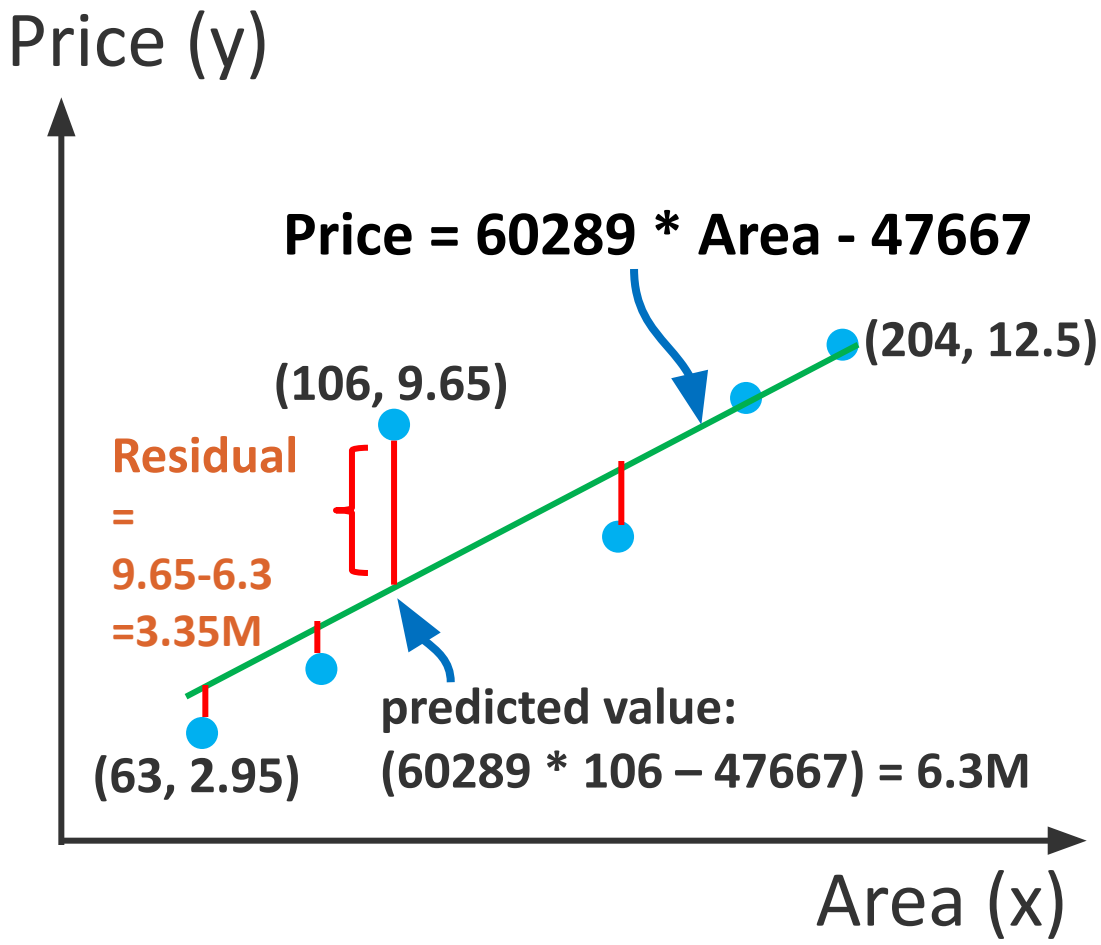
Predicted price = 60289 * 132 - 47667 = 7.9M



- **Q: how did you get the linear function "Price = 60289 * Area – 47667" ?**

Linear regression model – *residual*

- Each house has true value of the price. It also has predicted value of its price calculated by the linear model. But, there is error between the true value and the predicted value. Not all observations are perfectly on the line.
- *Residual*: defined as the difference between the true value and the predicted value calculated by the linear model.
- In principle, we shall minimize the overall error between the predicted value and the true value to find the linear model. Here, overall error refers to: the sum of residual squares.
- Q: why not simply minimize the sum of residual?

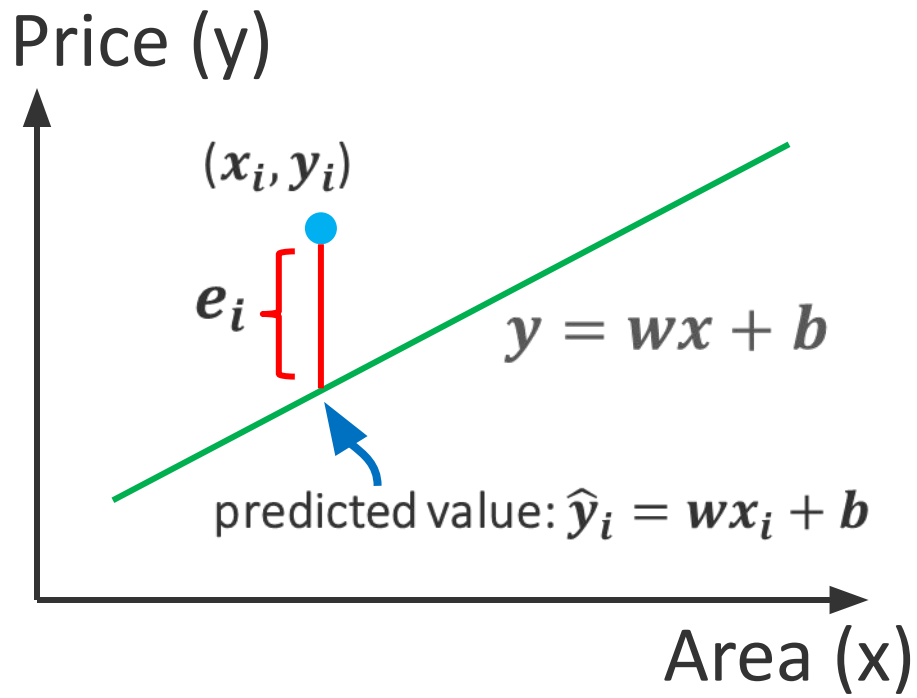


Linear regression model

- Given N observations (x_i, y_i); $(\mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots \mathbf{N})$.
- *Goal:* to find the linear function to minimize the overall error between the model and the actual observations.
- The linear model is defined as: $y = wx + b$
- The residual for observation i is given by

$$e_i = y_i - (wx_i + b)$$

true value — predicted value



- The overall error between the linear model and the true value is defined as the sum of residual squares. Hence, we need to find coefficients w and b that minimize

$$\min_{w,b} \sum_{i=1}^{N} e_i^2 = \min_{w,b} \sum_{i=1}^{N} [y_i - (wx_i + b)]^2$$

Least square is a standard approach to solve this problem

- Least square: a mathematical procedure for finding the best estimation function to a given set of points by minimizing the sum of the squares of the residuals of the points. Main idea to solve the optimization problem:
  - Define $\mathbf{E}(\boldsymbol{w}, \boldsymbol{b}) = \sum_{i=1}^{N}[\boldsymbol{y_i} - (\boldsymbol{wx_i} + \boldsymbol{b})]^2$, then function $\mathbf{E}(\boldsymbol{w}, \boldsymbol{b})$ is a convex function, we can find the optimal result if we set its derivative function as zero.
  - Take the partial derivative with respect to $\boldsymbol{w}$ and $\boldsymbol{b}$; and set them into zero

$$\frac{\partial E(w,b)}{\partial w} = 0; \qquad \frac{\partial E(w,b)}{\partial b} = 0$$

- Results have closed-form expressions

$$\begin{cases} w = \sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) / \sum_{i=1}^{N}(x_i - \bar{x})^2 \\ b = \bar{y} - w\bar{x} \end{cases}$$

where $\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}; \bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$

Linear regression in Python or Matlab

- **Python: Scikit-learn is an open sources machine learning tool in Python. It is built on NumPy, SciPy, and matplotlib.**
- **More information: www.scikit-learn.org**

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model

# Create linear regression object regr =
linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train, y_train)
```

- **Matlab: polyfit function solves linear regression problem**

```
p = polyfit(x,y,n)
```

→ **x, y: two vectors of data value**

→ **p: coefficient for the polynomial with degree n**

coeffecient = polyfit(x,y,1)

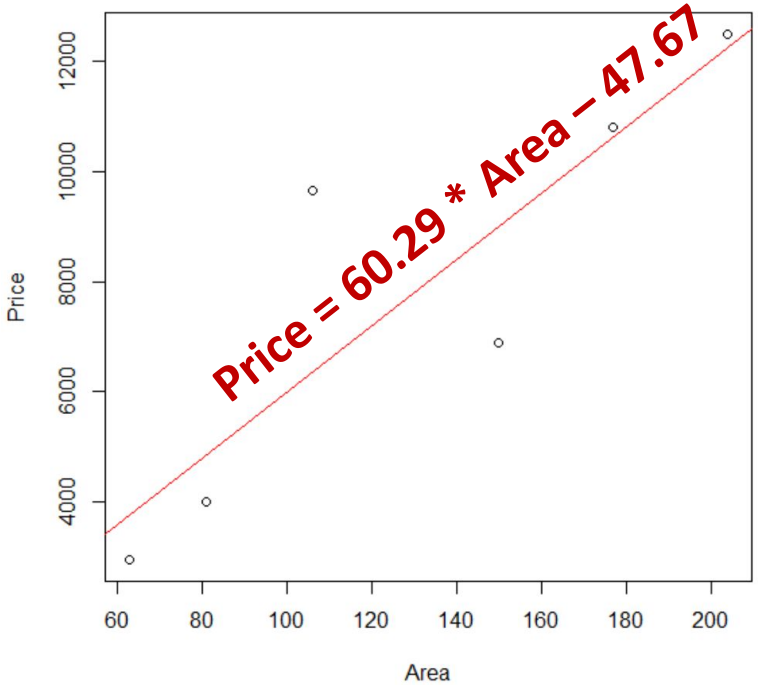
For more information: https://se.mathworks.com/help/matlab/ref/polyfit.html

Linear regression in R

- R is a free software environment for statistical computing and graphics
- Installation and more information: https://www.r-project.org/

```
#read CSV file
MyData <- read.csv("MyData.csv", sep=";", header=TRUE)
#scattplot points
plot(Price ~ Area, data = MyData)
#linear regression function
lm.out = lm(Price ~ Area, data = MyData)
#plotting the regression line on an existing scatterplot
abline(lm.out, col='red')
lm.out
```

linear regression



MyData.csv

Area	Price
177	10800
204	12500
81	4000
150	6900
106	9650
63	2950

Output :

```
Coefficients:
(Intercept)         Area
     -47.67        60.29
```

Linear Regression for Wind Energy Forecasting

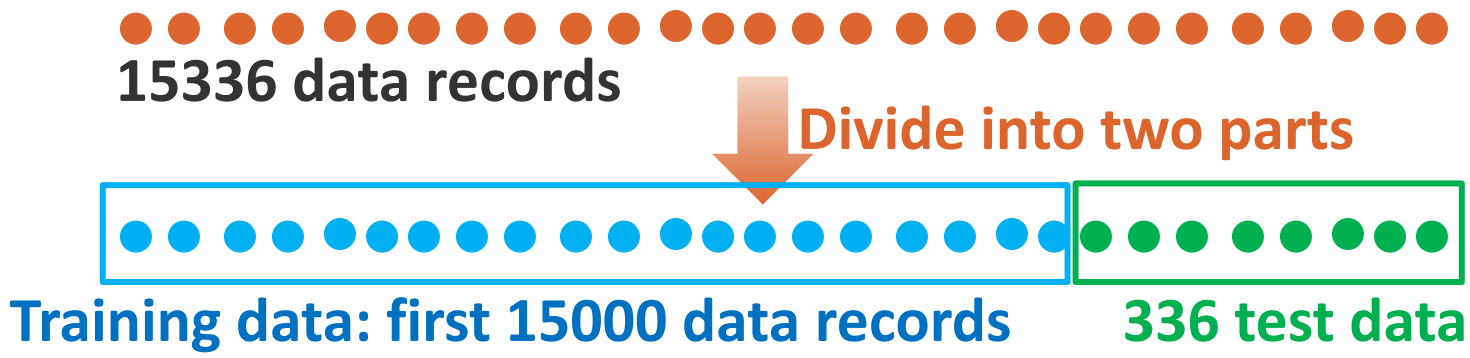
	A	B	C	D	E	F	G
1	TIMESTAMP	POWER	U10	V10	WS10		
2	20120101 1:00	0.2736781568	0.5348940035	-3.6602432799	3.6991204986		
3	20120101 2:00	0.0867959455	0.3308130989	-2.6764297561	2.6967969048		
4	20120101 3:00	0.0068114015	-0.0658387244	-2.0290719396	2.0301398163		
5	20120101 4:00	0.0186459868	-0.4195494463	-1.7990895574	1.847361625		
6	20120101 5:00	0.0348118328	-0.7542244222	-1.6615260214	1.8246981117		

Normalized wind power

Wind speed above ground 10m

- The data is between 2012.01.01-2013.10.01 (21 months), which has 15336 data records
- "WS10": wind speed above ground 10m
- The file contains hourly wind power measurements and wind speed
- "POWER": real wind power measurement (normalized) for a wind farm in Australia (Q: why normalized wind power data?)
- Normalized data to preserve anonymity. Then, you are not able to trace back to the specific wind park.

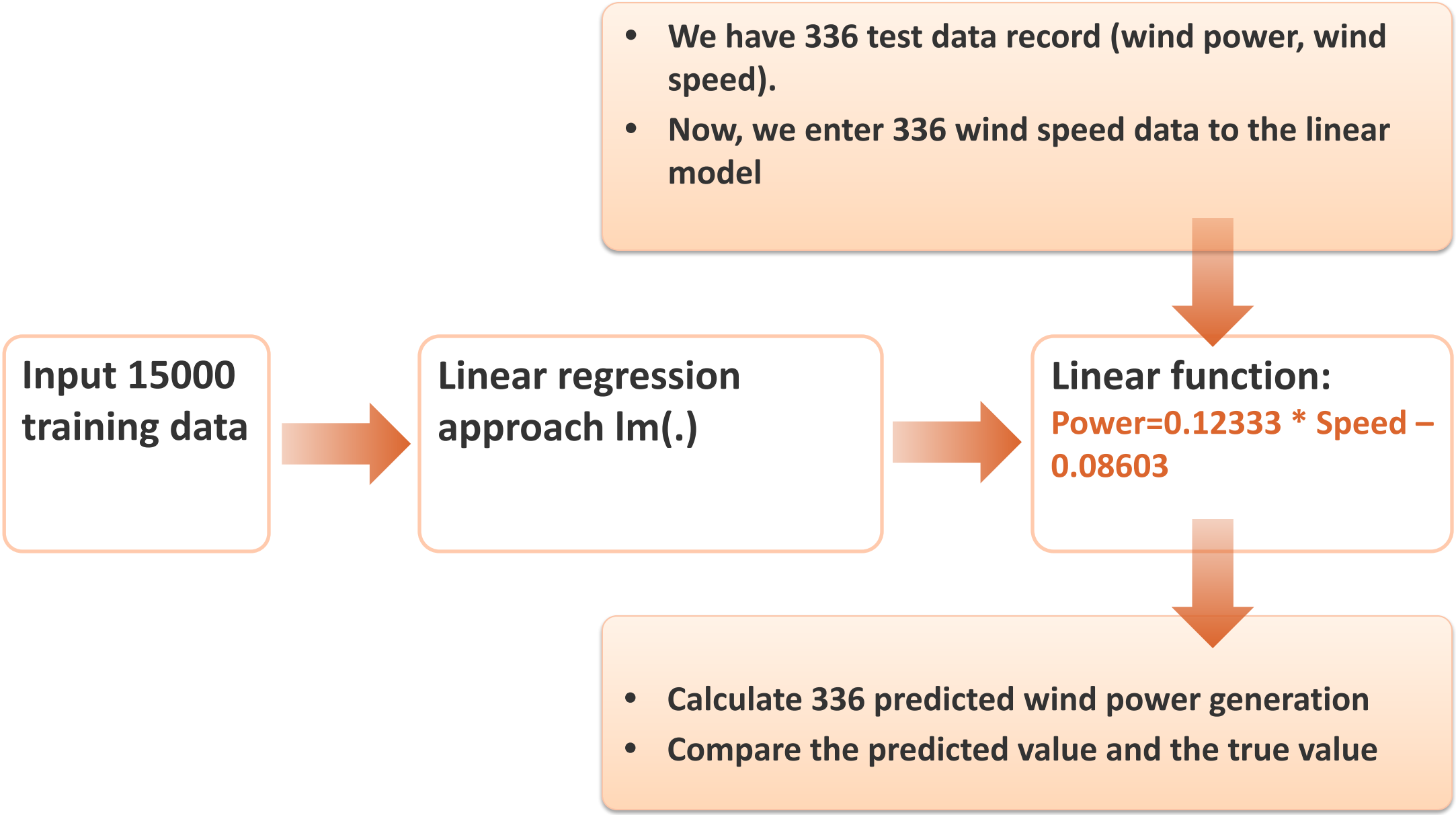
Training data and test data



- For all 15336 data records, we divide them into two parts.
- Training data: The first 15000 data records are used as training data to find the linear function (i.e., w and b)
- Test data: The last 336 data records are used as test data. This shows the period (from 2013-09-17 01:00:00 to 2013-10-01 00:00:00). Remind that these data are true value of wind power and wind speed in such period. We can calculate 336 predicted wind power generation based on the linear regression model. Then we compare the difference between the predicted value and the true value to calculate the prediction accuracy.

- **In general case, we may use 80% data of all data records as the training data while the last 20% data as the test data.**

Flowchart of linear regression approach for wind energy forecasting



Find the Training Model, i.e., linear function

- **The wind speed in the 336 test data records are used in the linear function to find the predicted wind power generation.**

- **The true wind power in the test data are then used to compare with the forecasted power generation. This can show if the training model (i.e., linear function) can provide accurate prediction.**

```
#read data from CSV file
data <- read.csv("WindPowerData.csv)

#the first 15000 data are training data
powerTrain = data$POWER[1:15000]
wsTrain = data$WS10[1:15000]

#linear regression
lmOut = lm(powerTrain ~ wsTrain)
#get the coefficients in the linear model
and save in coeffs
coeffs = coefficients(lmOut)
```

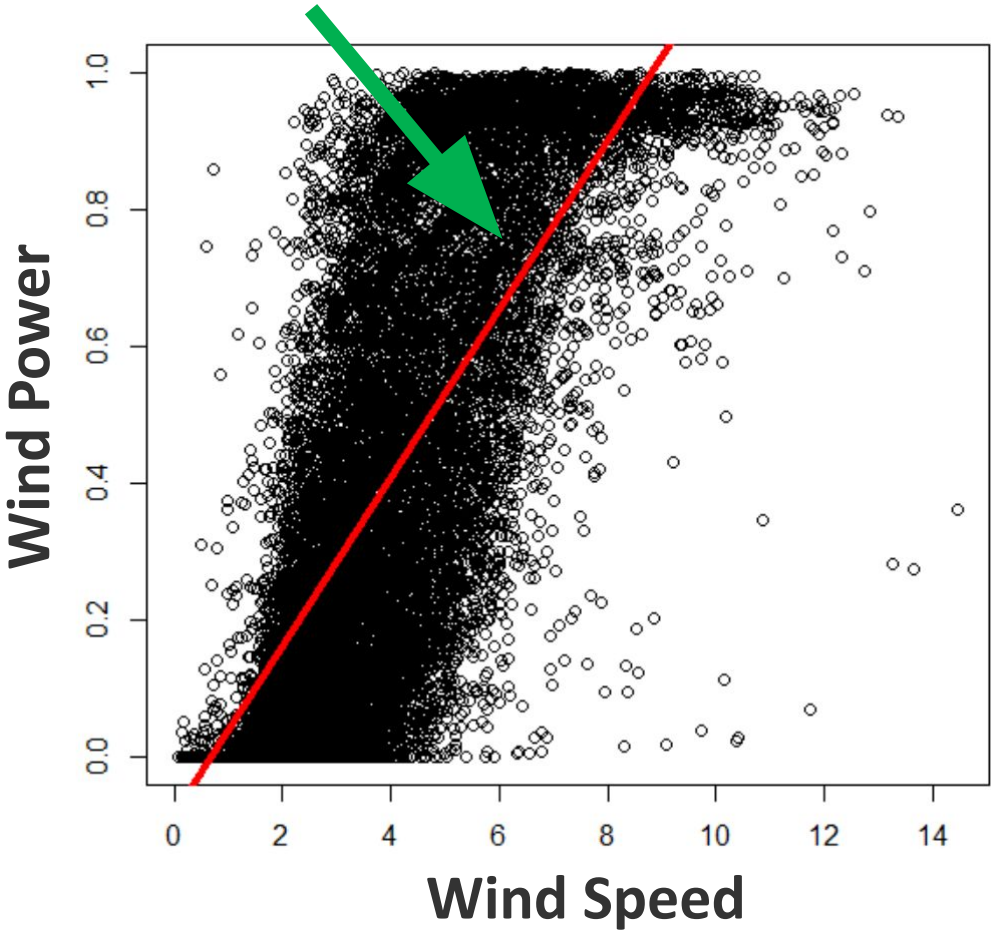
linear regression

Linear Regression for Wind Energy Forecasting

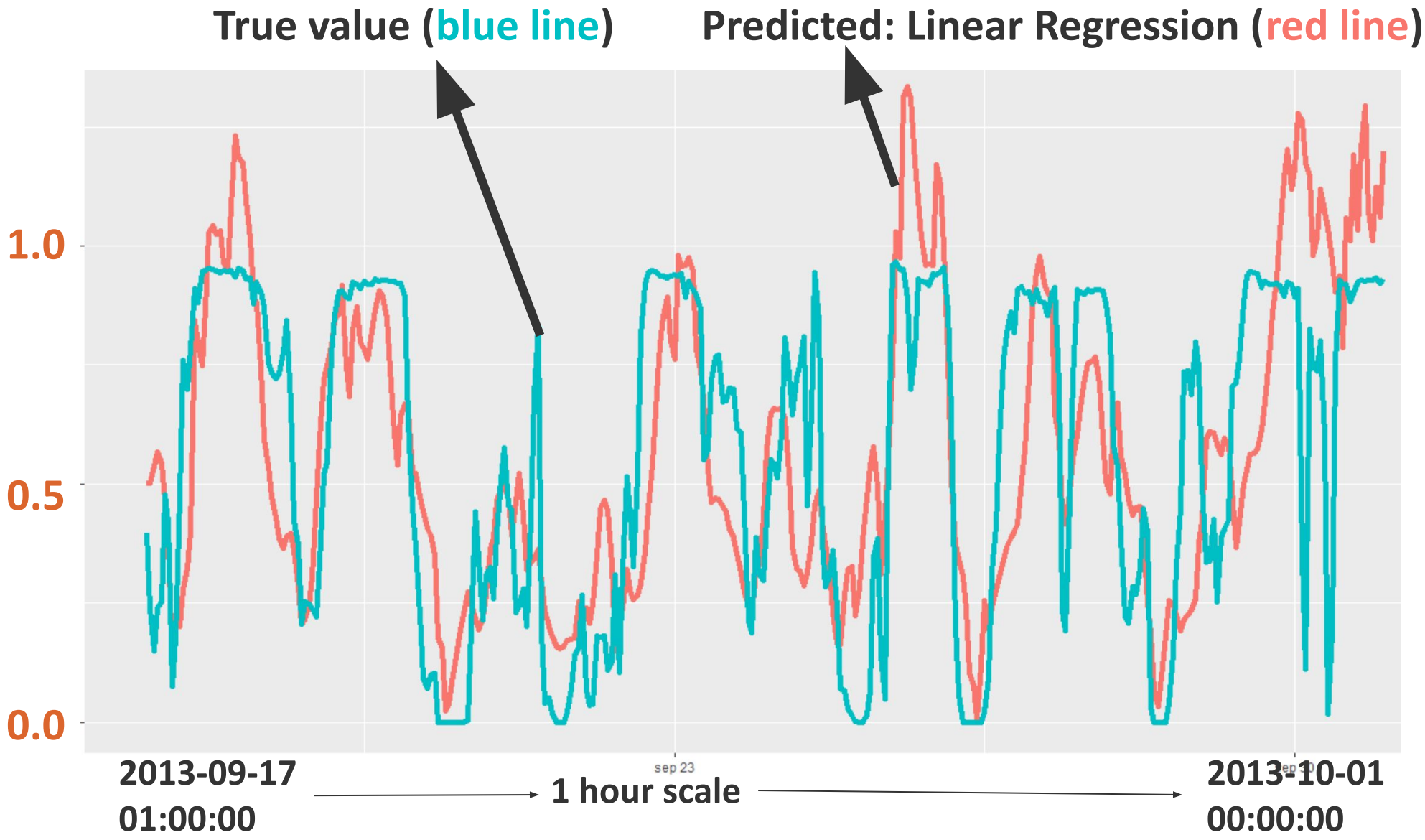
- Black circle “o”: the scatter plotting of (wind speed, wind power) pairs for the training data
- We assume linear relationship between wind power and wind speed. Then, we use linear regression to demonstrate their relationship.

- Q: is linear relationship a good model?

Power=0.12333 * Speed – 0.08603



True value & Predicted Wind Power for the Test Data (the last two weeks data records)



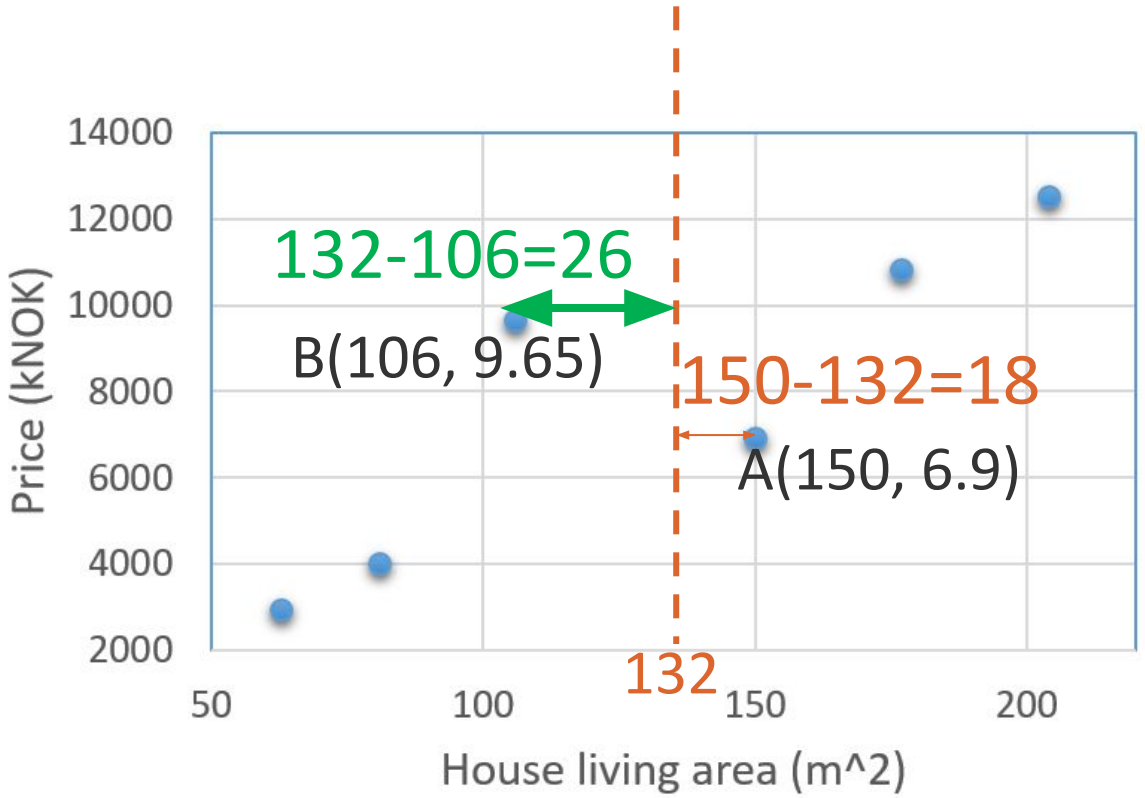
03. 02

KNN AND ITS APPLICATION FOR WIND ENERGY FORECASTING

k-Nearest Neighbor (kNN) Regression. Here, '*k*' is a variable, k=1,2,3….

- Q: what is the price for a new house with area 132 m^2?

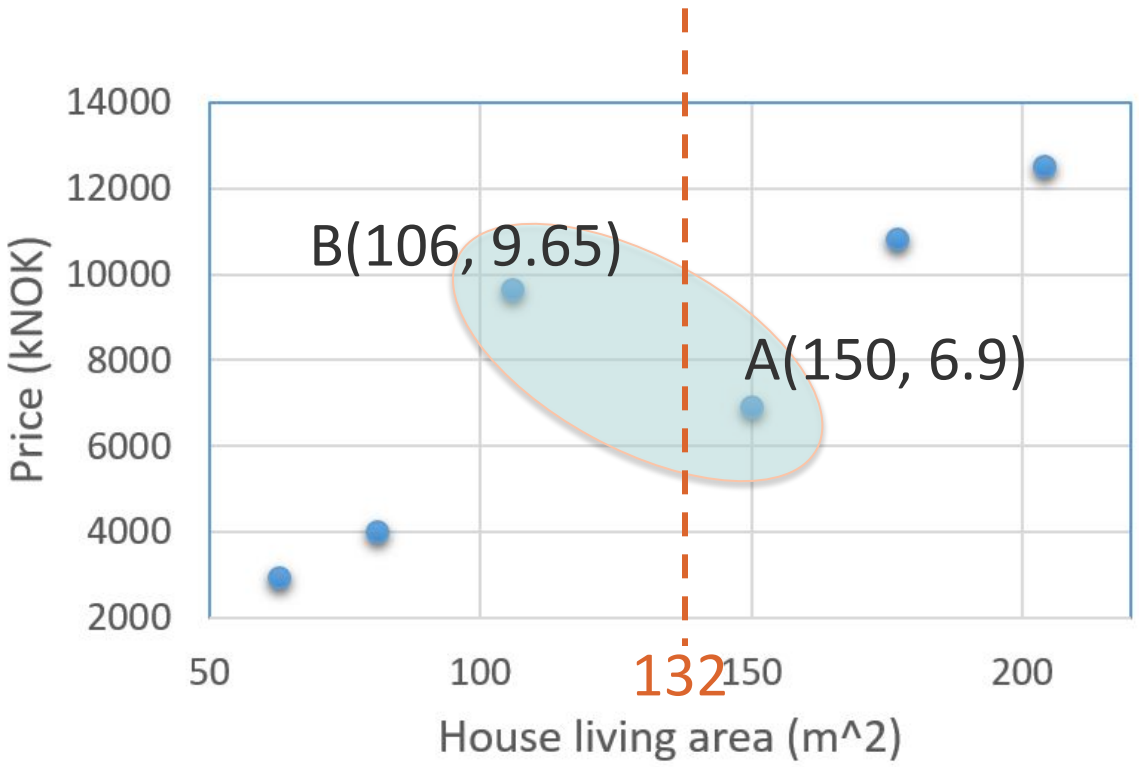
- 1-NN (one nearest neighbor, k=1)
  - distance: the difference between living area size
  - the observation A(150, 6.9) is the nearest point from the area 132 m^2
  - then, the predicted price has the same price as the house with area 150 m^2, which is 6.9M



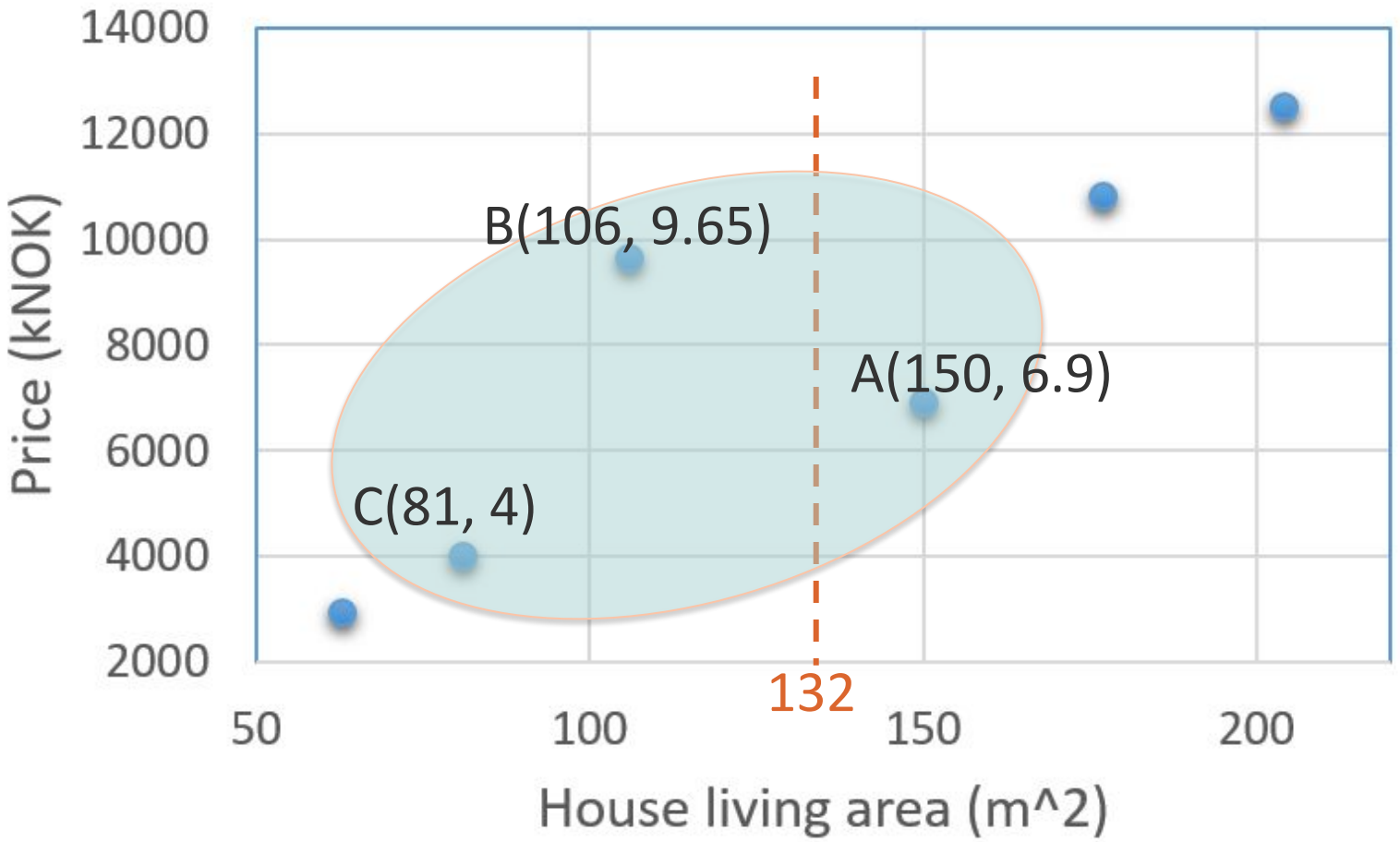
2-Nearest Neighbor (2NN)

- Q: what is the price for the new house with area 132 $\boldsymbol{m}^2$?

- 2-NN (two nearest neighbors, k=2)
  - the observations A(150, 6.9) and B(106, 9.65) are the two nearest node from the area 132 $\boldsymbol{m}^2$
  - Then, the predicted price is the average price of these two houses, i.e., (9.65+6.9)/2= 7.825M



K-Nearest Neighbor Regression (kNN)



- 3-NN (three nearest neighbors): the observations A(150, 6.9), B(106, 9.65) and C(81, 4) are the three nearest nodes from the area 132 $\boldsymbol{m}^2$, so, the prediction is the average of 9.65M, 6.9M and 4M, which is 6.55M

kNN Algorithm

Given:

- Training set (x_i, y_i), $(i = 1, 2 \ldots N)$
  - x_i: attribute-value representation of examples (e.g., living area)
  - y_i: real-valued target (e.g., price)
- Testing point x that we want to predict the target

Algorithm

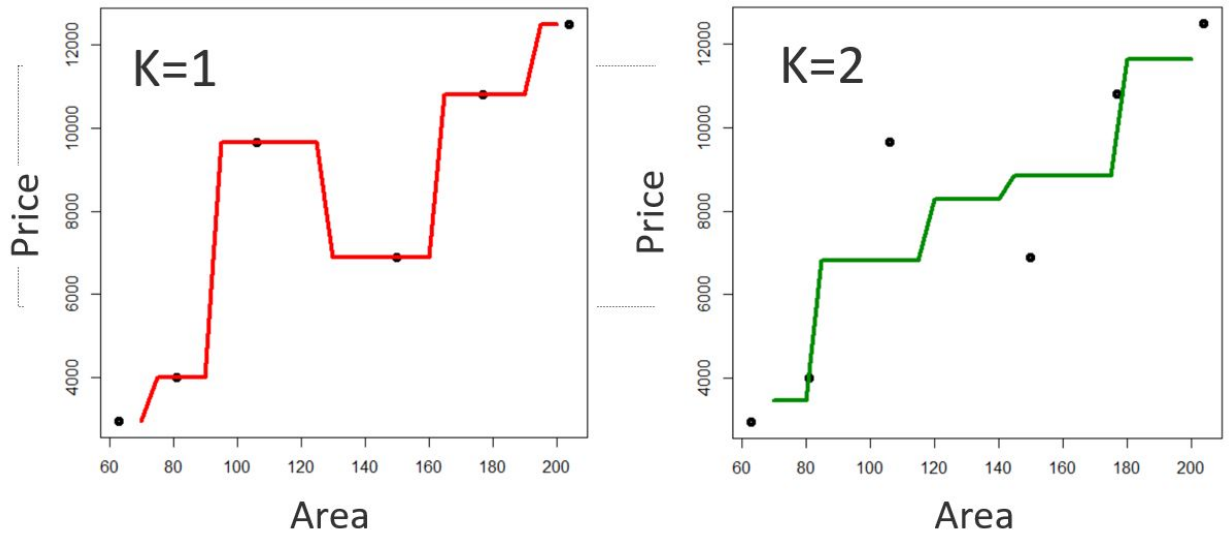
- Compute the distance $D(x, x_i)$ to every training example x_i: $D(x, x_i) = |x - x_i|$
- Select k closest points $(x_{i1}, x_{i2}, \ldots, x_{ik})$ and their values $(y_{i1}, y_{i2}, \ldots, y_{ik})$
- Output the average value of $y_{i1}, y_{i2}, \ldots, y_{ik}$:

$$\overline{y} = \sum_{j=1}^{k} y_{ij}/k$$

Q: House price may depend on the area size, location, and public transport etc. In this case, we have training set (x_i, y_i, z_i) where x_i represents the living area, y_i represents the location, and z_i represents the house price. How to calculate the distance for the new house with the area size and location (x, y)?

$$D = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

kNN in R language and illustration




```
MyData <- read.csv("MyData.csv", sep=";", header=TRUE)

AreaTest = seq (70, 200, by=5); #test data set

#kNN regression on the training set

knnmodel = knn.reg(train=matrix(MyData$Area,ncol=1), test=matrix(AreaTest,ncol=1), y=MyData$Price, k=1)

plot(MyData$Area, MyData$Price, lwd=4, xlab="Area", ylab="Price")

lines(AreaTest, knnmodel$pred,col="red",lwd=4)
```

kNN regression in R

kNN for Wind Energy Forecasting

- Same real wind power data for a wind farm in Australia
- For 15336 data records, we divide them into two parts. The first 15000 data records are used as training data to build kNN model. The wind speed in the rest 336 records are used as test data to predict the wind power generation

```
powerAll = data$POWER #all wind power data set

wsAll = data$WS10 #all wind speed data set

trainLength = 15000 #number of data records to be used in training set

testLength = length(wsAll) - trainLength

wsTrain = wsAll[1:trainLength] #wind speed - training data set

powerTrain = powerAll[1:trainLength] #wind power - training data set

wsTest = wsAll[trainLength+1 : length(wsAll) ] #wind speed - test data set

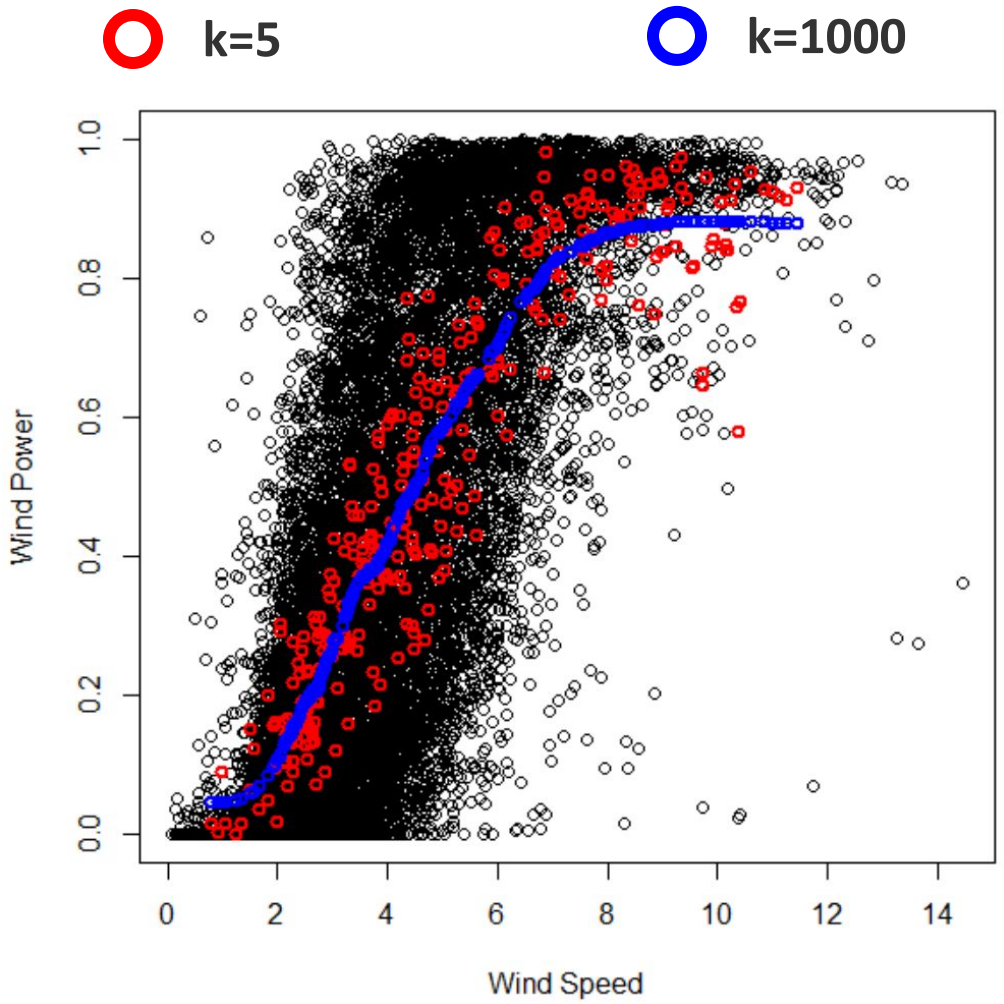
wsTest = wsTest[1:testLength] #wind speed - test data set

knnmodel = knn.reg(train=matrix(wsTrain,ncol=1),test=matrix(wsTest,ncol=1),y=powerTrain, k=10)
```

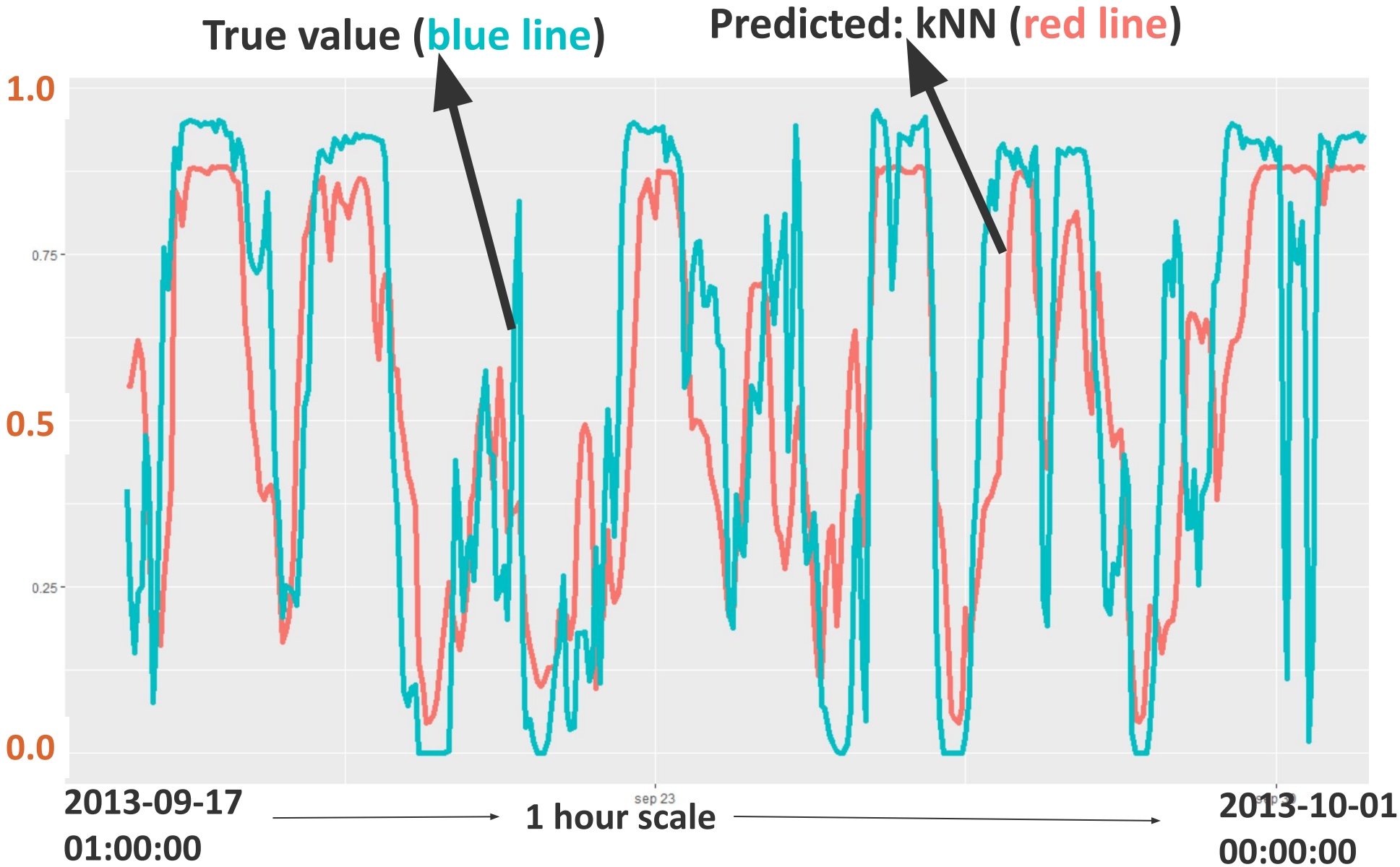
kNN regression

kNN for Wind Energy Forecasting

- Black circle "O": the scatter plotting of (wind speed, wind power) pairs in the training set
- Red circle "O": (wind speed, predicted wind power) when k=5
- Blue circle "O": (wind speed, predicted wind power) pairs when k=1000
- In this kNN model, the curve is not linear
- Q: what is an appropriate k?
- Test multiple k-values to decide an optimal value for your data
- use cross-validation to determine k. More information: http://genomicsclass.github.io/book/pages/crossvalidation.html



True value & Predicted Wind Power for the Test Data (the last 336 data records)

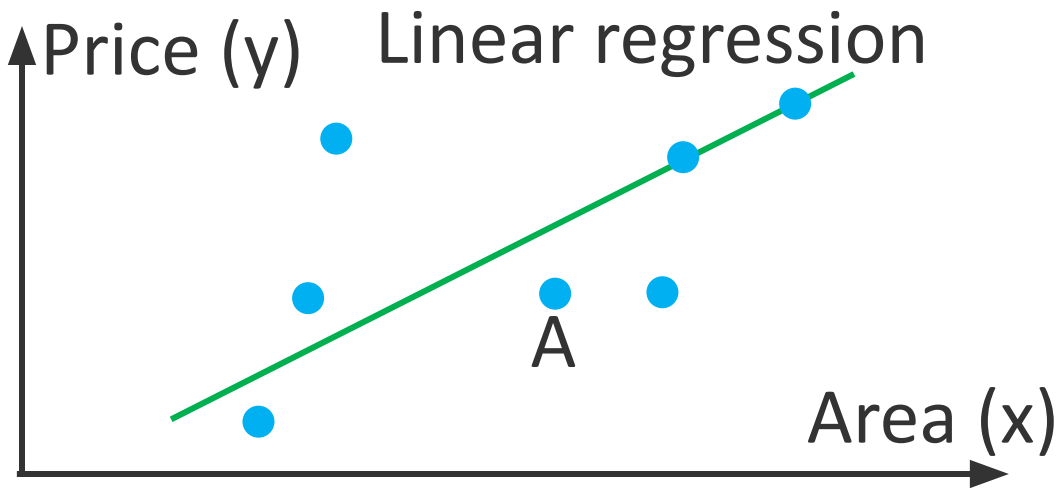


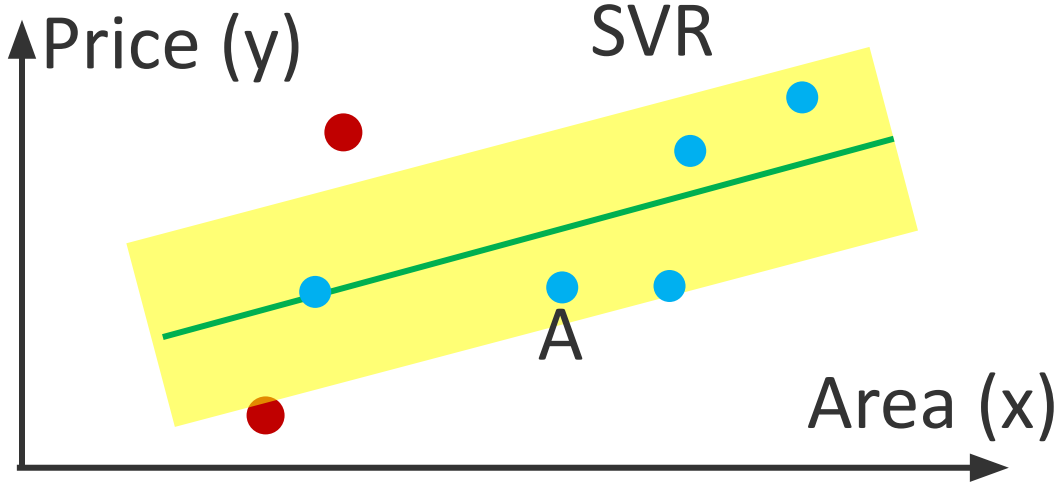
03. 03

SVR AND ITS APPLICATION FOR WIND ENERGY FORECASTING

Supported Vector Regression (SVR): main idea

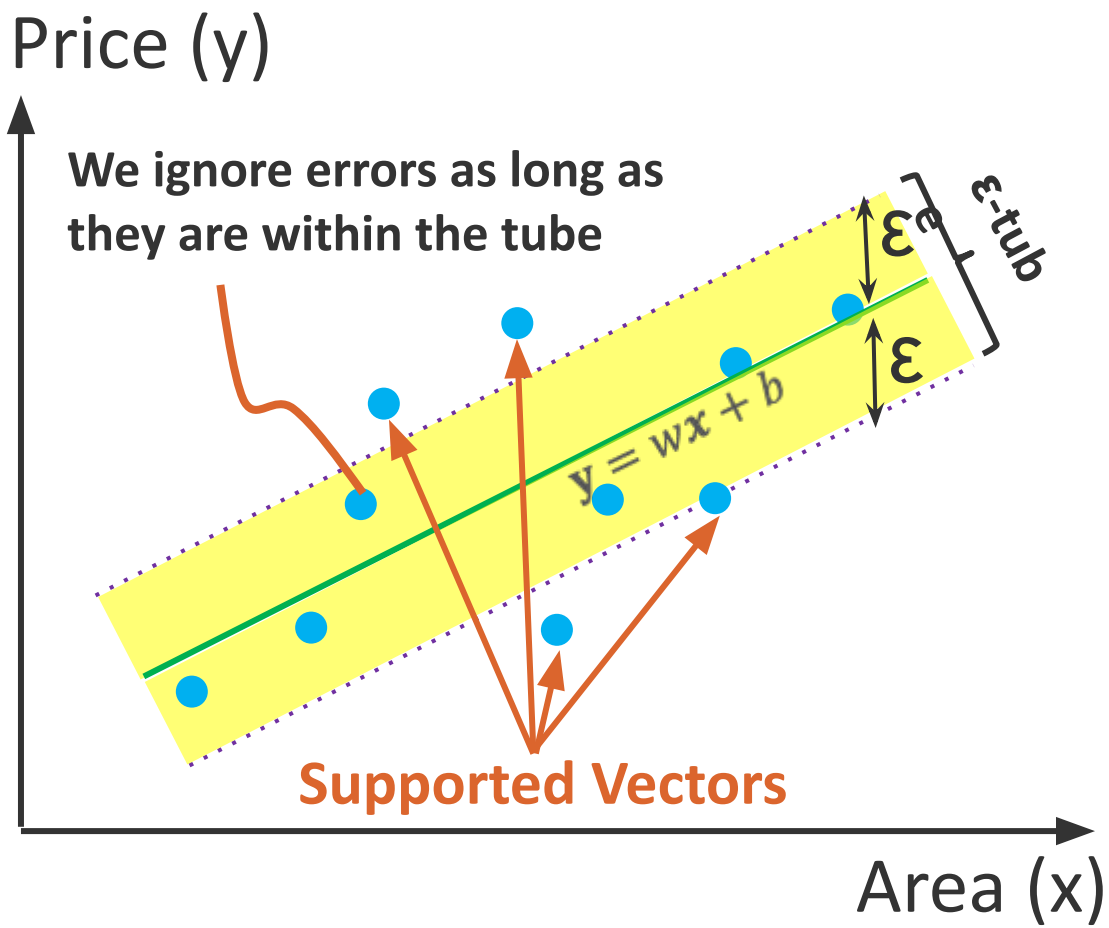
- In linear regression, the objective is to minimize the sum of residual squares. For all points, we calculate the error of each point when the predict is not the same as the true value.
- In practice, we may not care how large the errors are, as long as they fall within an acceptable range. This gives flexibility in building the model. For example,
  - it should be okay with the predicted house price being within a certain error, e.g., 50k NOK.
  - Oslo city's predicted temperature tomorrow is 10 degrees while the actual temperature is 12 degrees
- **Support Vector Regression (SVR): we ignore the errors as long as they are within the acceptable range as represented by the band, but will not accept any deviation larger**
- **For point A, we include its error in linear regression while we ignore its error in SVR**





Supported Vector Regression (SVR): main concepts

- Hyperline: the line to fit the data and show relationship between the house price and the living area, e.g., $y = wx + b$
- Margin ε: represents the error tolerance of the model. This is pre-defined parameter.
- ε-tube (or ε-insensitive tube): tube around the hyperline within which the error is ignored since the absolute values of errors are less than ε. Error is ignored if a point is inside the tube. Only the points out of the tube will be considered
- Supported Vector: points outside of ε-tube; and the points on the border of ε-tube. The smaller the value of ε, the more points that lie outside the tube and hence the greater number of supported vectors



Supported Vector Regression (SVR): width of ε-tube

- Perpendicular distance between two parallel lines.

Equations of two parallel lines

$$\begin{cases} Ax + By + C_1 = 0 \\ Ax + By + C_2 = 0 \end{cases}$$

Distance between the two lines

$$d = \frac{|C_1 - C_2|}{\sqrt{A^2 + B^2}}$$

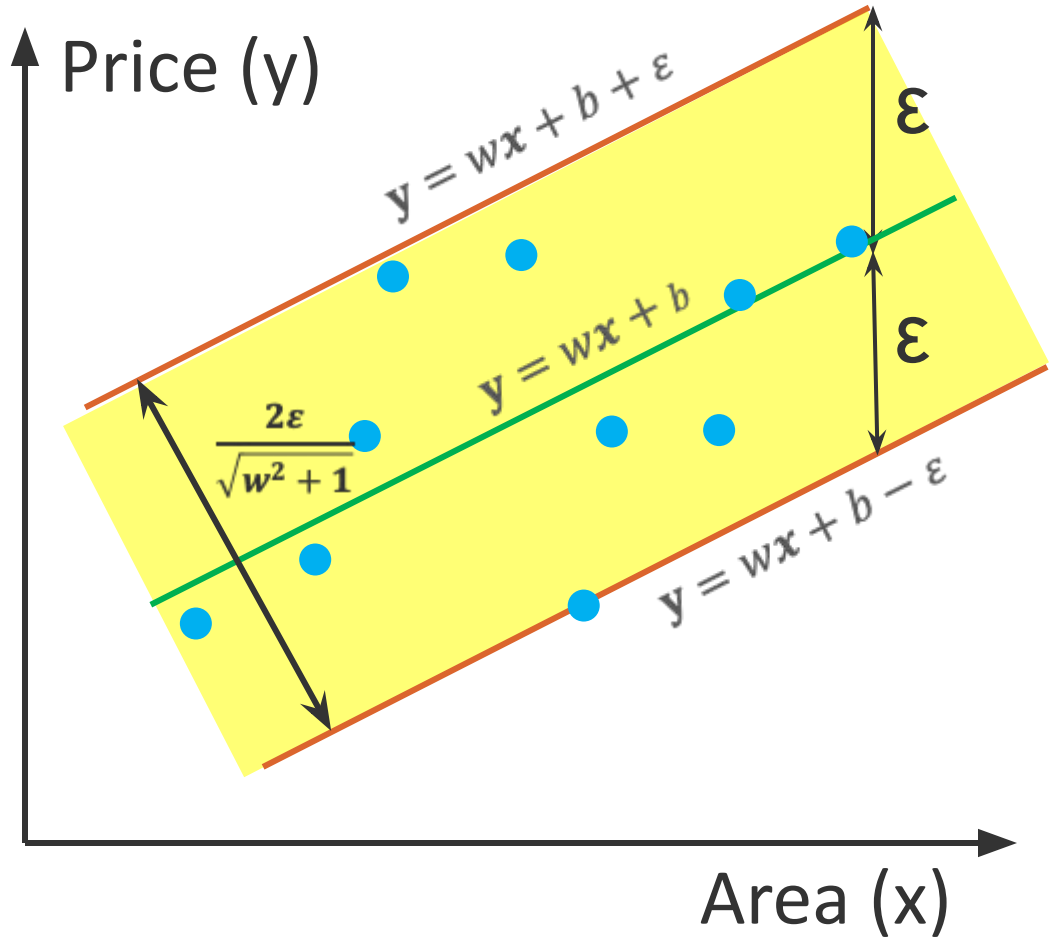
- Distance between two boundary lines of ε-tube, i.e., the width of ε-tube

Equations of two boundary lines

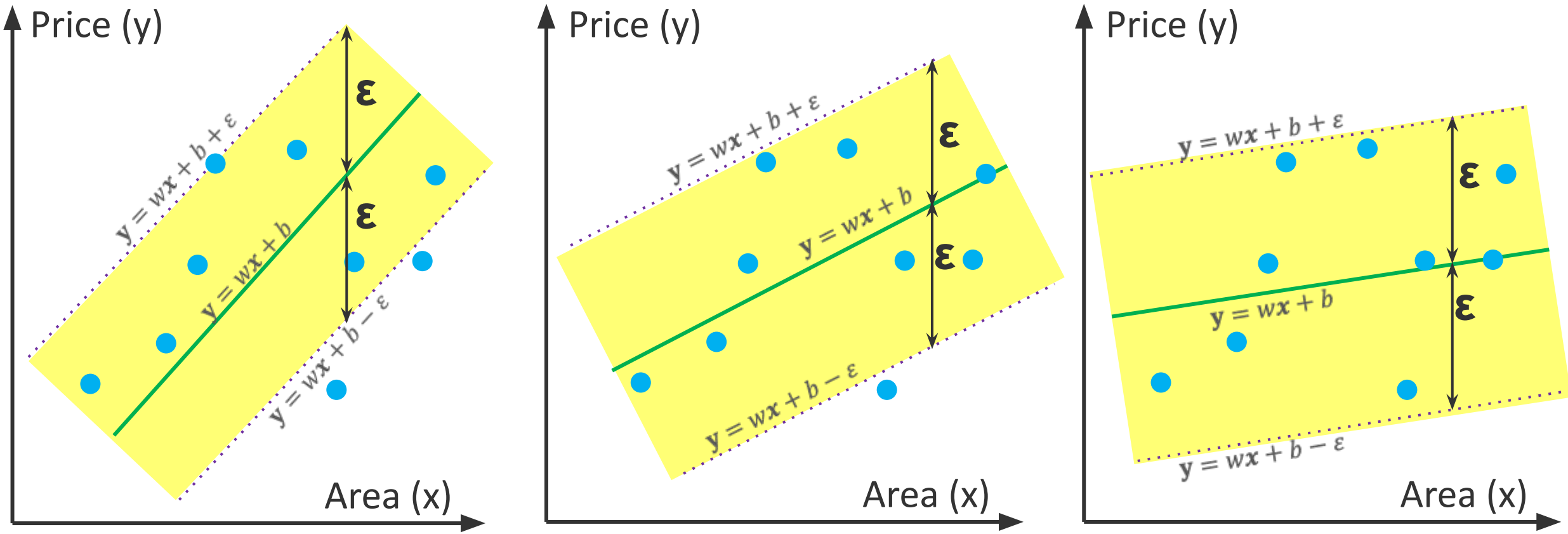
$$\begin{cases} y = wx + b + \varepsilon \\ y = wx + b - \varepsilon \end{cases}$$

Distance between the two lines

$$d = \frac{2\varepsilon}{\sqrt{w^2 + 1}}$$



Supported Vector Regression (SVR): effect of the parameter w



Large w: narrow tube, more points lie outside the tube

Small w: wide tube, less points lie outside the tube

Smaller w: wider tube, most or all points lie inside the tube

Observation: smaller $|w|$ leads to wider tube and more points are within the tube. → in order to cover as many as training points as possible, we should maximize the width of the tube, i.e., minimize $|w|$.

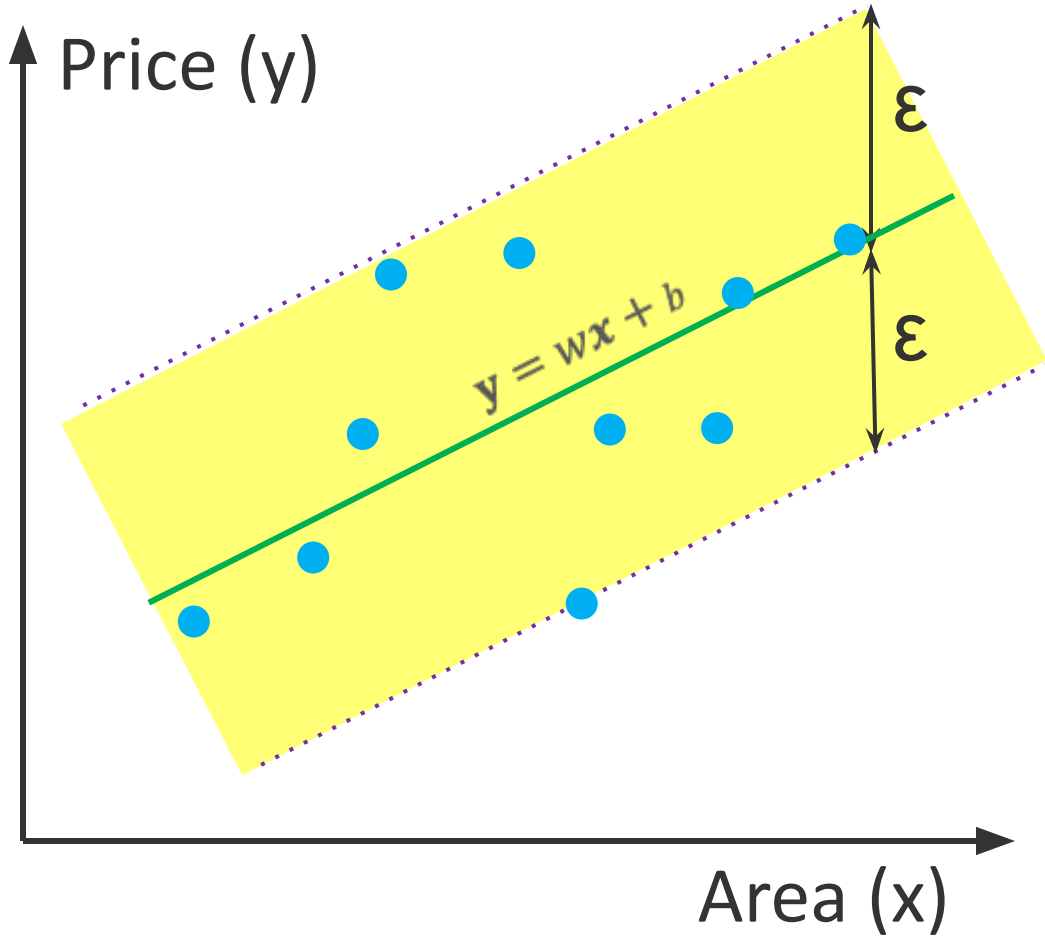
Supported Vector Regression (SVR) with hard margin when all points are within ε-tube

- *Goal*: to find a function that maximizes the width of ε-tube such that as many points as possible are within tube and hence the overall error approaches zero. This is equivalent to minimizing $|\boldsymbol{w}|$, or $\boldsymbol{w}^2$.

$$\min \frac{1}{2} w^2$$

$$subject\ to: \quad y_i - (wx_i + b) \leq \varepsilon$$
$$(wx_i + b) - y_i \leq \varepsilon$$

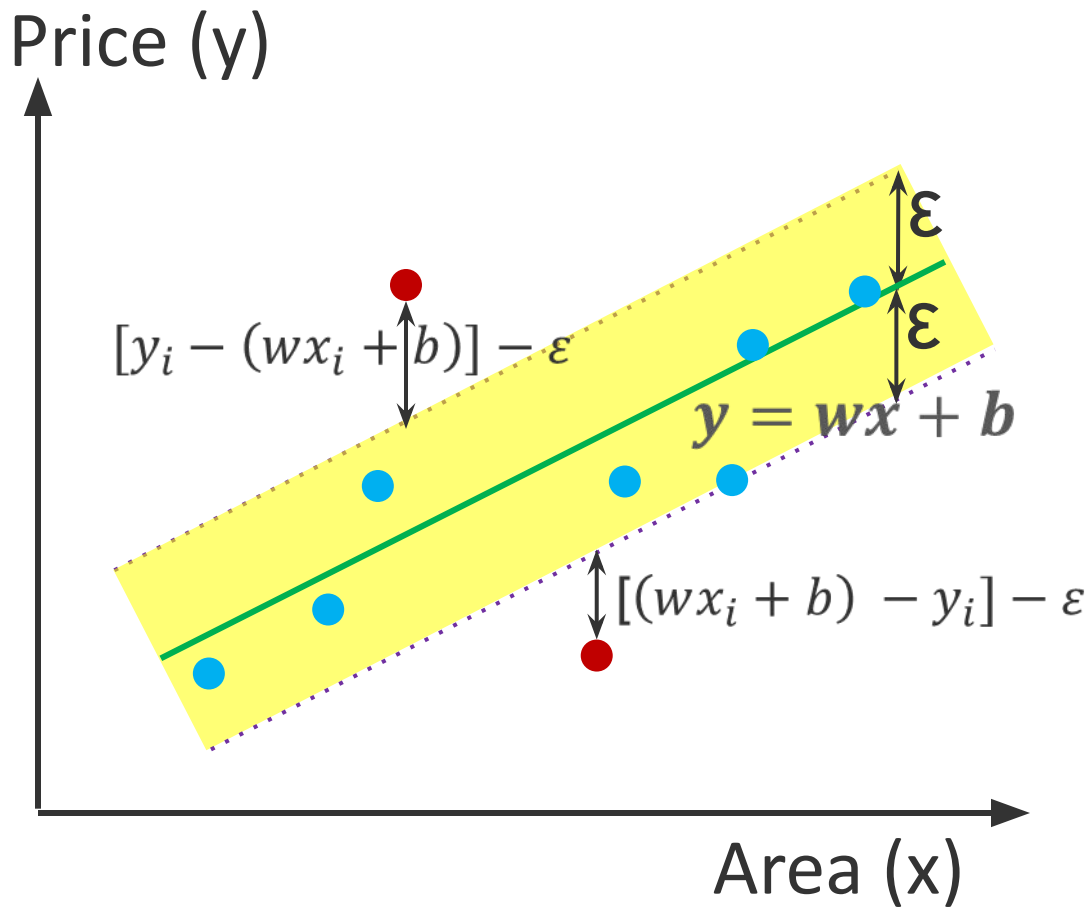
- Observation: This optimization problem can generate the function as flat as possible. With minimized $|\boldsymbol{w}|$, the function is less sensitive to small changes in the input data. This can avoid overfitting problem, i.e., the model does a perfect job of fitting the training data, but will do a bad job of predicting new data. This means the model is sensitive to new data.



Supported Vector Regression (SVR) with soft margin when we allow for errors

- In some cases, the optimization problem (previous slide) may not be feasible and not all points can fall within the tube with pre-defined ε. For points outside the tube, we need to calculate their errors and also minimize the errors. Still, error is zero if the points are inside the ε-tube.
- A point inside the tube satisfies $|\boldsymbol{y_i - (wx_i + b)}| < \boldsymbol{\varepsilon}$, then error is given by 0
- A point outside the tube satisfies $|\boldsymbol{y_i - (wx_i + b)}| > \boldsymbol{\varepsilon}$, error is given by $|\boldsymbol{y_i - (wx_i + b)}| - \boldsymbol{\varepsilon}$
- For any point (x_i, y_i), its error can be written as

$$max(|y_i - (wx_i + b)| - \varepsilon, 0)$$



Supported Vector Regression (SVR) with soft margin when we allow for errors

- *Goal*: to find a function that maximize the width of ε-tube (i.e., minimize $\frac{1}{2}w^2$ as shown already); and also minimize the errors of all points outside the tube.

$$\min_{w,b} \frac{1}{2}w^2 + C\sum_{i=1}^{N} max(|y_i - (wx_i + b)| - \varepsilon, 0)$$

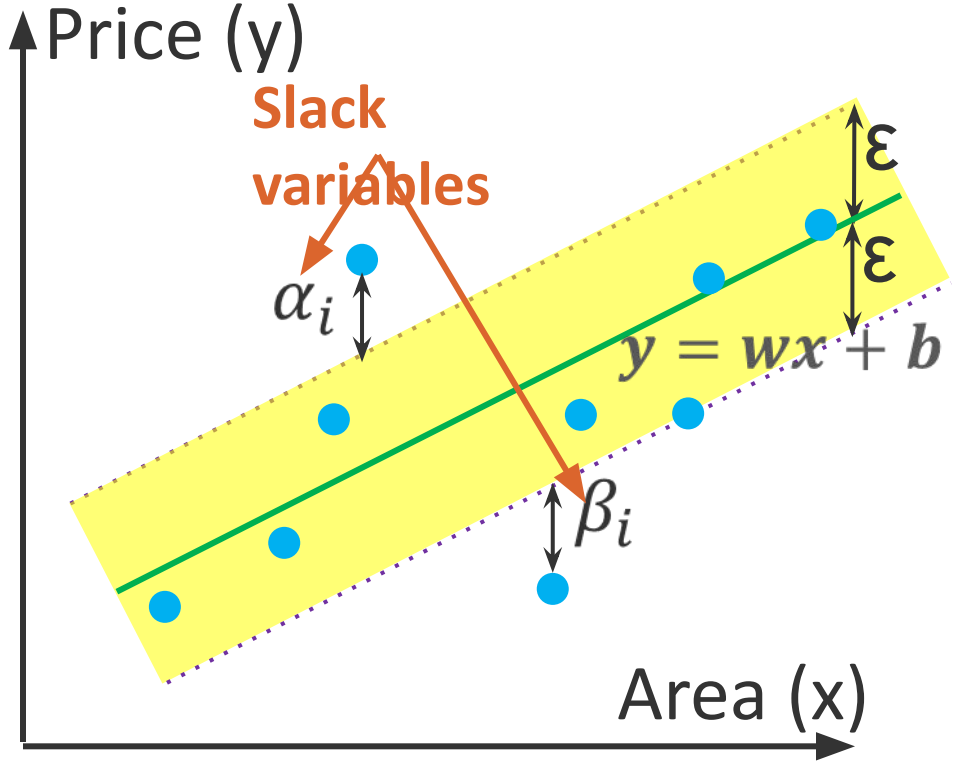
- Regularization parameter *C*: balance the tradeoff between maximizing the width of ε-tube and minimizing the error of all points outside the tube. Smaller *C* allows more errors, while larger *C* allows fewer errors. When C→∞, all points should be within the tube.
- The optimization is not easy to solve with the absolute value operation. Now, we can convert this problem into a new optimization problem after removing the absolute value operation.

Supported Vector Regression (SVR): general case when we allow for errors

- We introduce two slacking variables α_i, β_i, the SVR with soft margin optimization problem can be converted into a constrained quadratic optimization problem:

$$\min_{w,b} \frac{1}{2} w^2 + C \sum_{i=1}^{N} (\alpha_i + \beta_i)$$

$$subject\ to: \quad y_i - (wx_i + b) \leq \varepsilon + \alpha_i$$
$$(wx_i + b) - y_i \leq \varepsilon + \beta_i$$
$$\alpha_i \geq 0;\ \beta_i \geq 0, \qquad i = 1, \ldots, N$$



- Solution: can be solved by constructing the Lagrange function. Then, take partial derivatives of the Lagrange function with respect to w, b, α_i, β_i and setting them equal to zero, based on the *Karush-Kuhn-Tucker* (KKT) conditions.

$$L = \frac{1}{2} w^2 + C \sum_{i=1}^{N} (\alpha_i + \beta_i) + \sum_{i=1}^{N} \mu_i (y_i - (wx_i + b) - \varepsilon - \alpha_i) + \sum_{i=1}^{N} \mu_i^* ((wx_i + b) - y_i - \varepsilon - \beta_i) - \sum_{i=1}^{N} \varphi_i \alpha_i - \sum_{i=1}^{N} \varphi_i^* \beta_i$$

SVR for Wind Energy Forecasting

- We use the same real wind power data for a wind farm in Australia
- For all 15336 data records, we divide them into two parts. The first 15000 data records are used as training data to build SVR model. The wind speed in the rest 336 records are used as test data to predict the wind power generation

```
#use the same source code to get wsTrain, powerTrain, wsTest

plot(wsTrain, powerTrain, xlab="Wind Speed", ylab="Wind Power")

#Support vector machine/regression

df <- data.frame(x = wsTrain, y = powerTrain)

svrmodel <- svm(y ~ x, data = df)

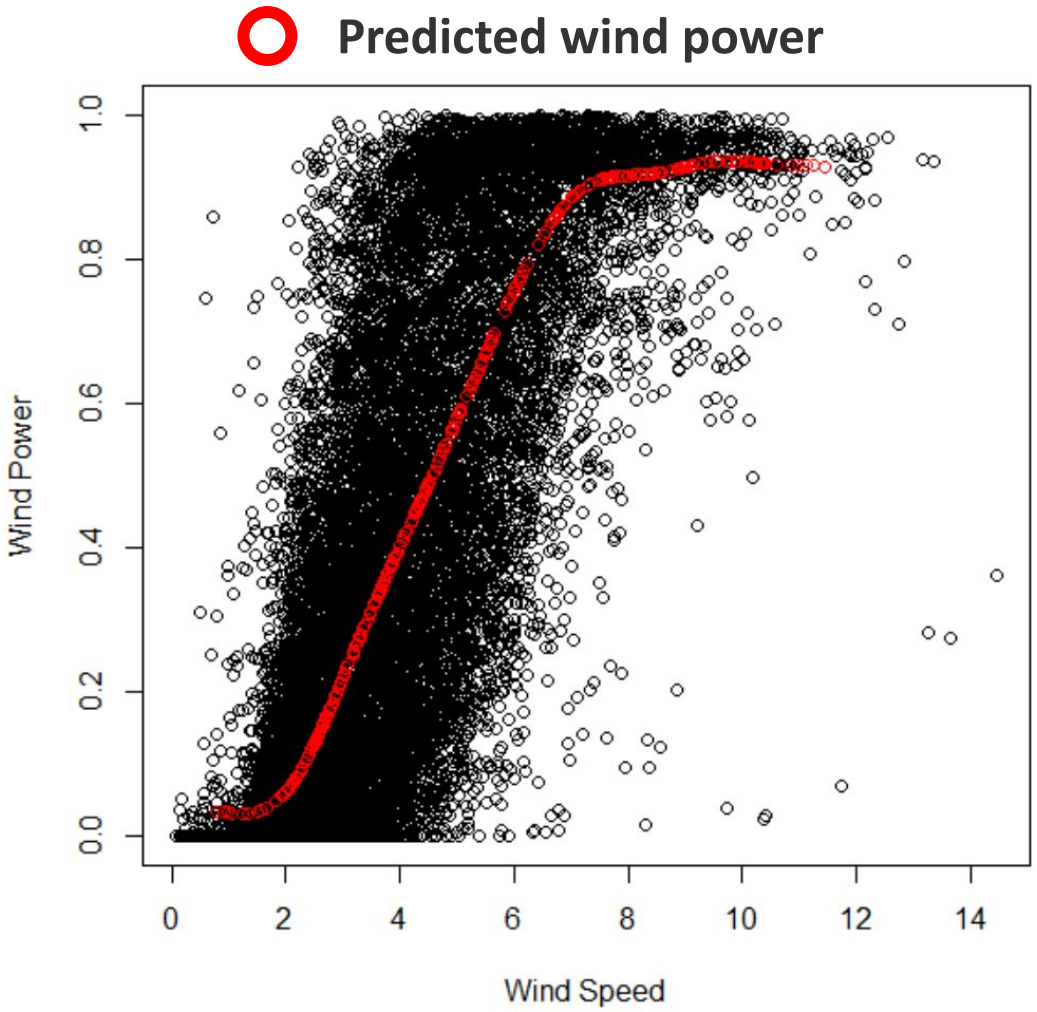
powerPredicted = predict(svrmodel, newdata = data.frame(x = wsTest))

# Add points for fitted svrmodel

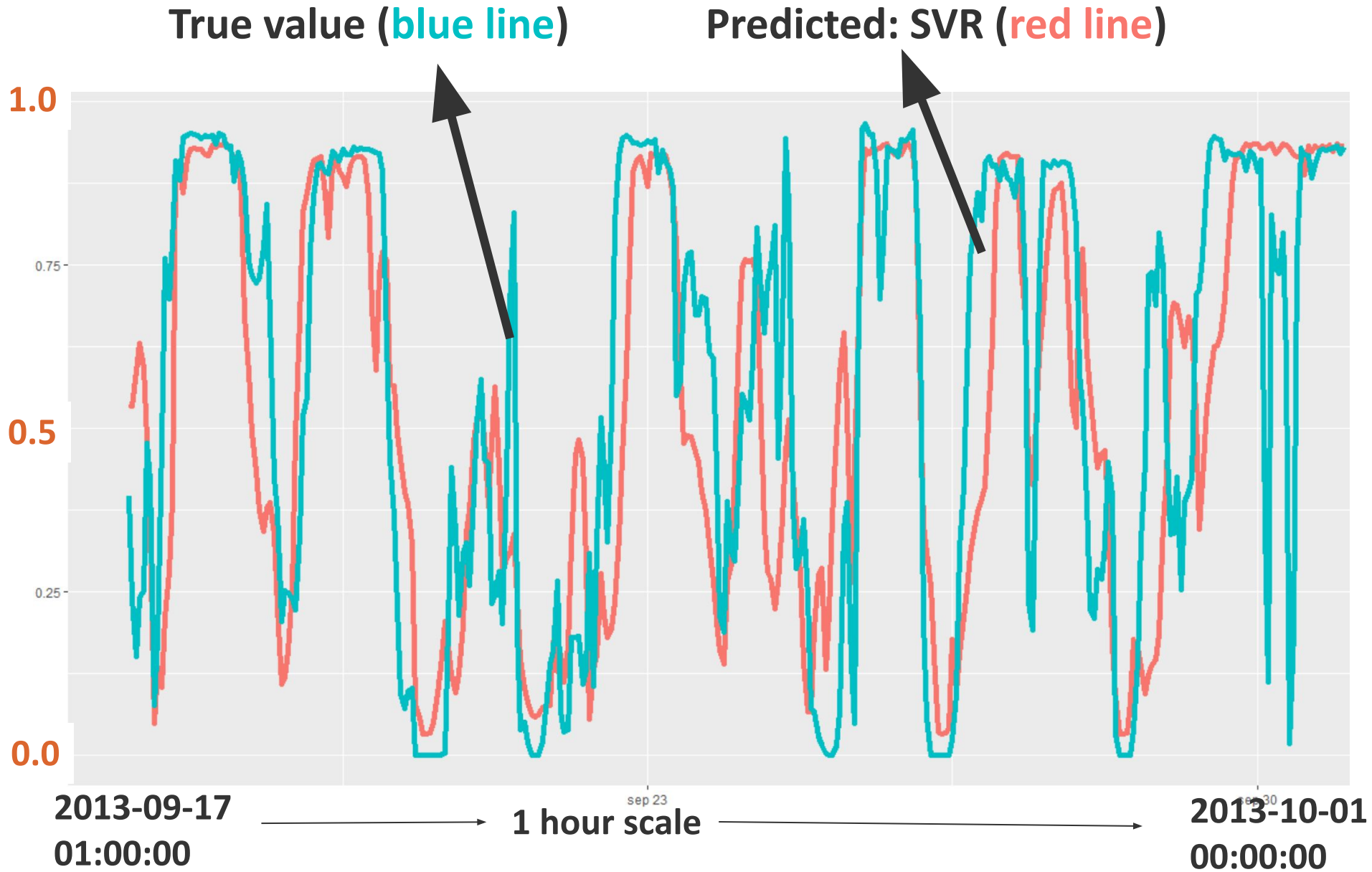
points(wsTest, powerPredicted , col = "red" )
```

SVR for Wind Energy Forecasting

- Black circle “O”: the scatter plotting of (wind speed, wind power) pairs in the training set
- Red circle “O”: (wind speed, predicted wind power) according to the wind speed in the test dataset
- Similar as kNN model, SVR-model based predicted wind power also follow a non-linear curve



True value & Predicted Wind Power for Test Data



Metrics measures the model goodness in prediction

- **Q: we discussed three models: linear regression, kNN and SVR. Which model is the best?**

- For each data point $\boldsymbol{x_i}$, its real value is $\boldsymbol{y_i}$. A model makes a prediction $\boldsymbol{\hat{y}_i}$. In order to measure how good a model is we will compute how much error it makes. We compare each real value $\boldsymbol{y_i}$ with its predicted value $\boldsymbol{\hat{y}_i}$ and calculate the difference.

- The expression $\boldsymbol{\hat{y}_i - y_i}$ is the error. For each data point, we have the sum of the squared errors, and then take the mean. We have the Mean Squared Error (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

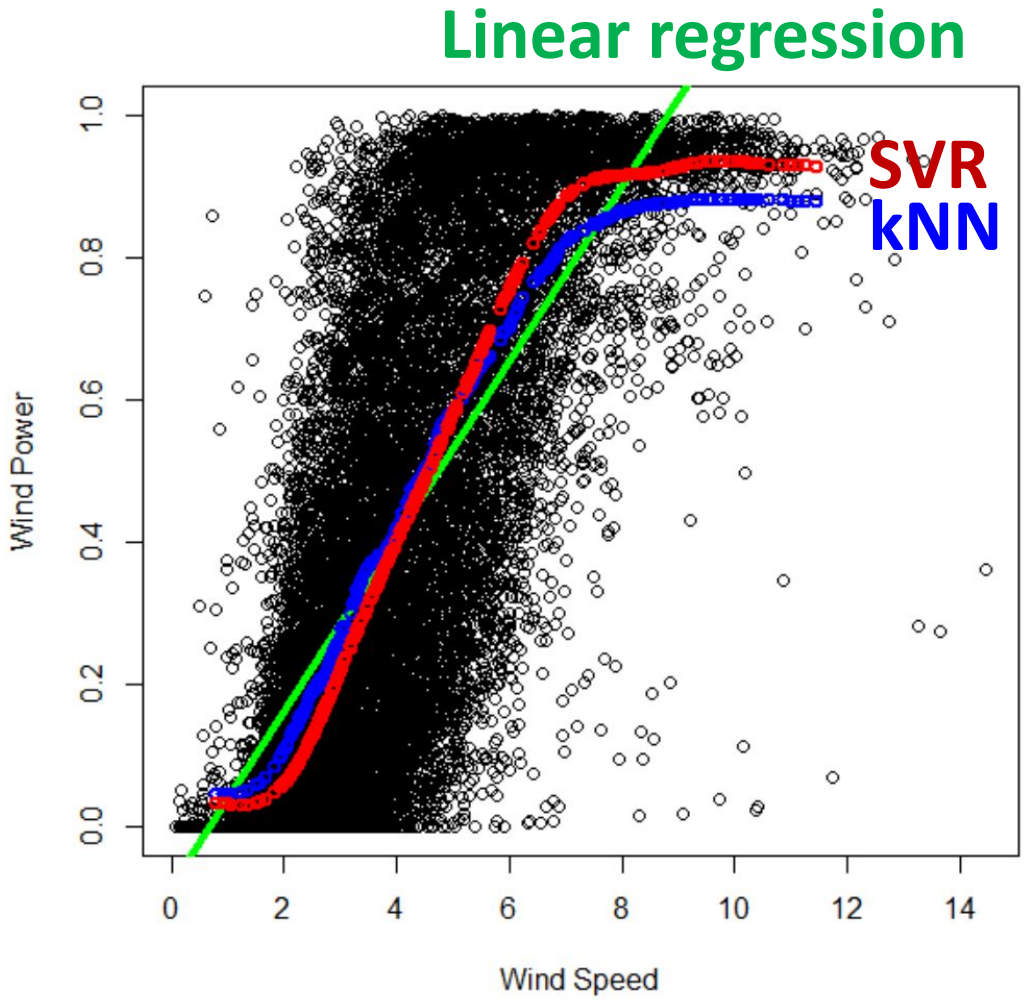
- A common way to measure error uses the Root Mean Squared Error (RMSE), we take the square root of MSE

$$RMSE = \sqrt{MSE}$$

An illustration to show the RMSE between these models

```
rmse <- function(error)
{    sqrt(mean(error^2)) }
# Calculate error
errorSVR <- powerTest - powerPredicted
rmse(errorSVR)
```

Model	RMSE
Linear regression	0.2545
kNN (k=1000)	0.2276
SVR	0.2343



04

MORE CONSIDERATIONS...

Different time horizon prediction

- Normally we need short-term forecasting for solar or wind energy power generation
- Q: in what cases, we need very short-term forecasting, e.g., 3mins?

Solar power plant in desert



More parameters

- House price is not only dependent on the area. It is also related to location, public transport, school rankings, etc. How to use machine learning to model and solve this problem?
- For example, if we consider more factors affecting house price, we have Multiple Linear Regression model

$$\text{Price} = b + w_1 * \text{Area} + w_2 * \text{Location} + w_3 * \text{SchoolQuality}$$

- Similarly, wind power generation is not only dependent on wind speed. It is also related to wind direction, temperature, and pressure.

$$\text{Wind Power} = b + w_1 * \text{WindSpeed} + w_2 * \text{WindDirection} + w_3 * \text{Temperature} + w_4 * \text{Pressure}$$

Code: lm(powerTrain ~ wsTrain + WindDirection + temperature + pressure)