Regression

Supervised Learning : Regeresi

Nama Pengajar

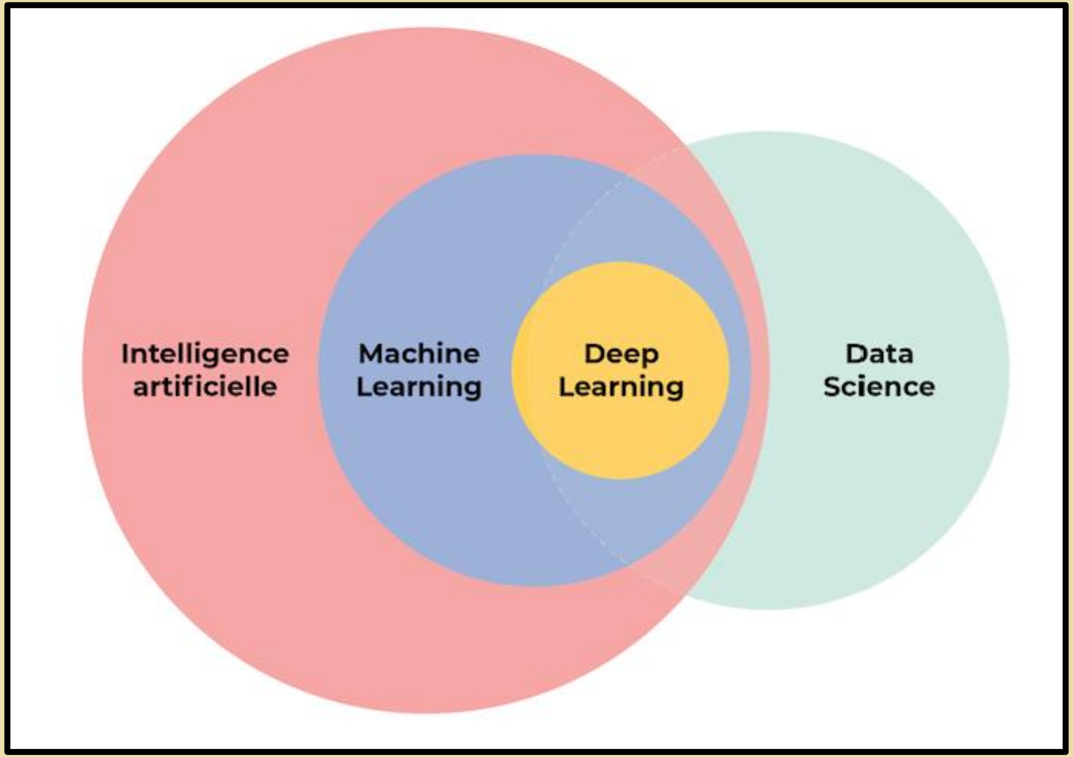
- Dian Ade Kurnia, M.Kom
- Rusnanda Farhan
- Rusnandi Fikri
- Rika Sahriana

Outline

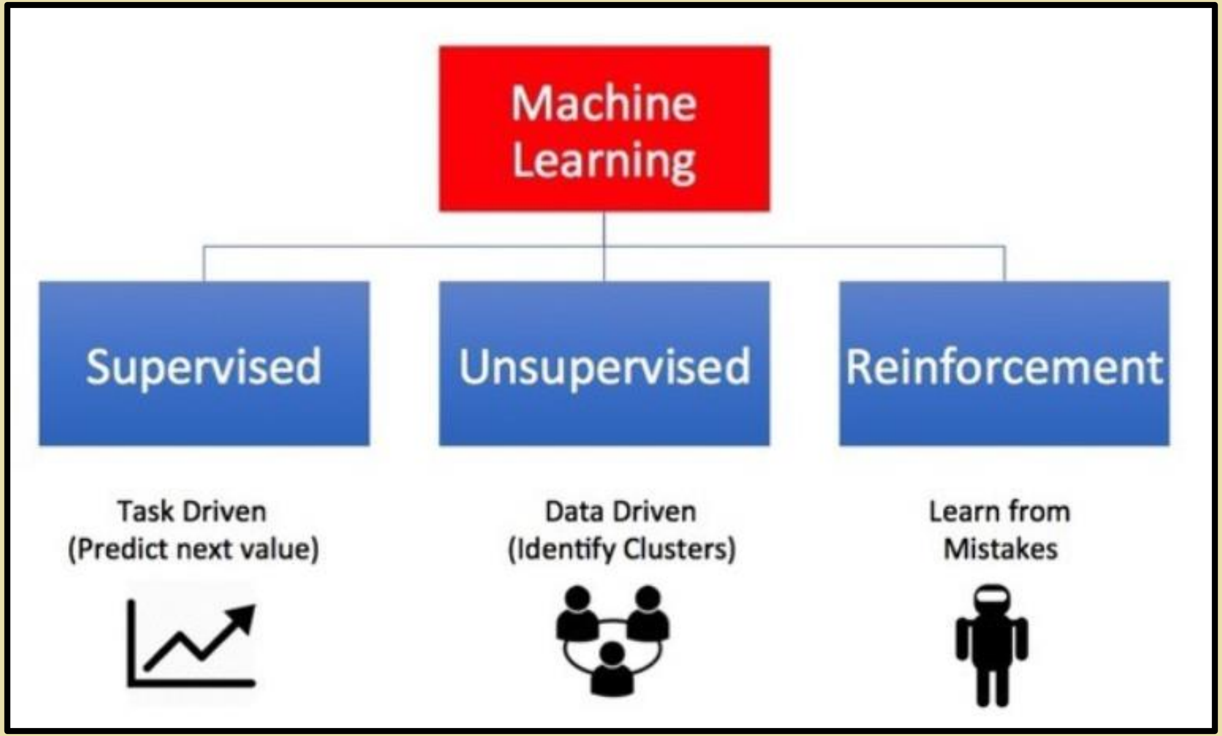
- Introduction Machine Learning
- Introduction Regression
- Linear Regression
- Decision Tree
- SVM

Introduction Machine Learning

Machine Learning is part of Artificial Intelligence, and intersects with Data Science. Machine Learning is the study of computer algorithms that can improve automatically through experience and by the use of data (training data). As the result of Machine Learning. it usually builds/produces a model.

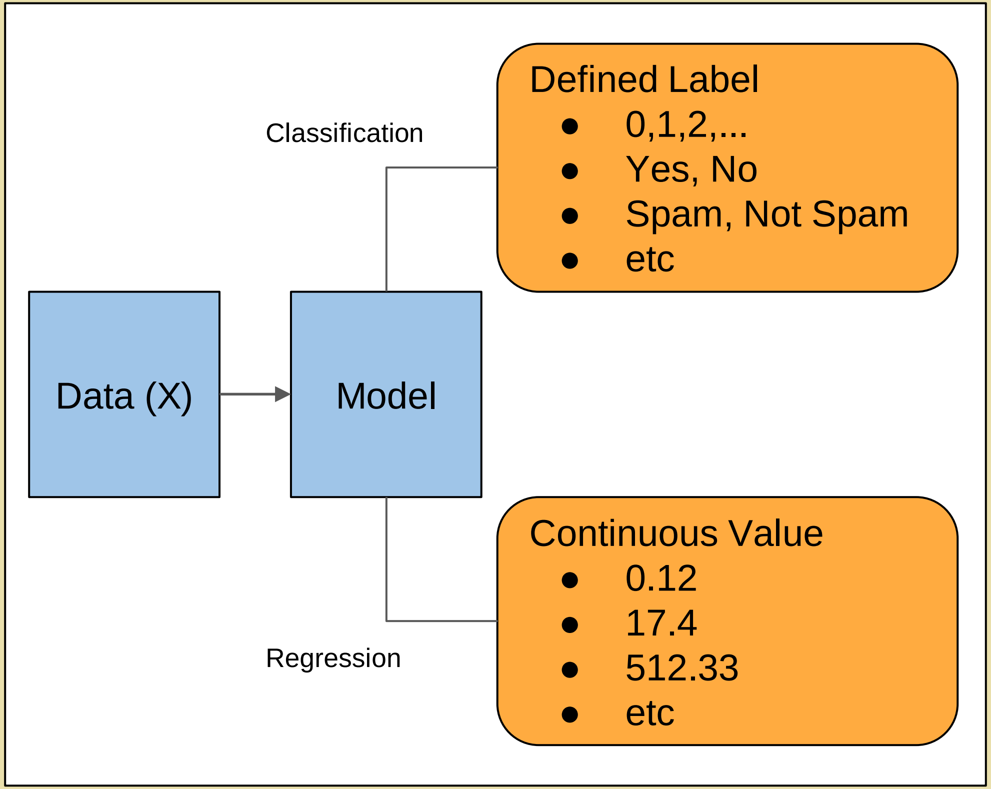


Kind of Machine Learning



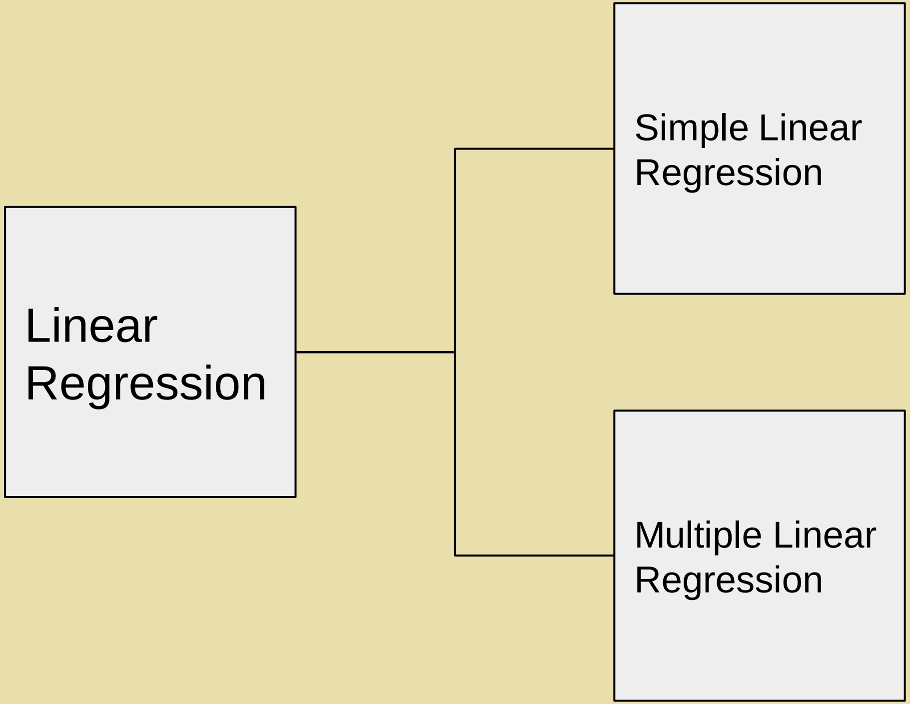
Introduction Regression

Regression. A data science task of predicting the value of target (numerical variable) by building a model based on one or more predictors (numerical and categorical variables).



Linear Regression

Linear Regression will build a linear equation based on data. We have to find the best parameter that fit to the equation in order to estimate dependent variable accurately (minimum error).

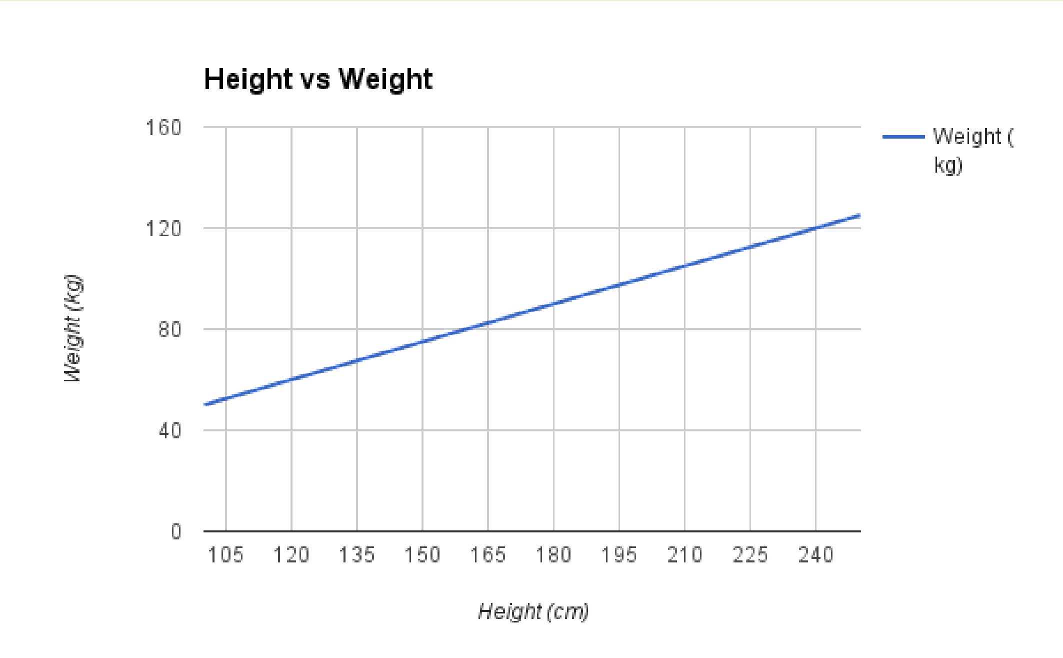


Linear Regression

Simple Linear Regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable (or independent variable), and the other is considered to be a dependent variable.

$$Y = aX+b$$

where **X** is the explanatory variable and **Y** is the dependent variable. The slope of the line is **a**, and **b** is the intercept (the value of y when x = 0).

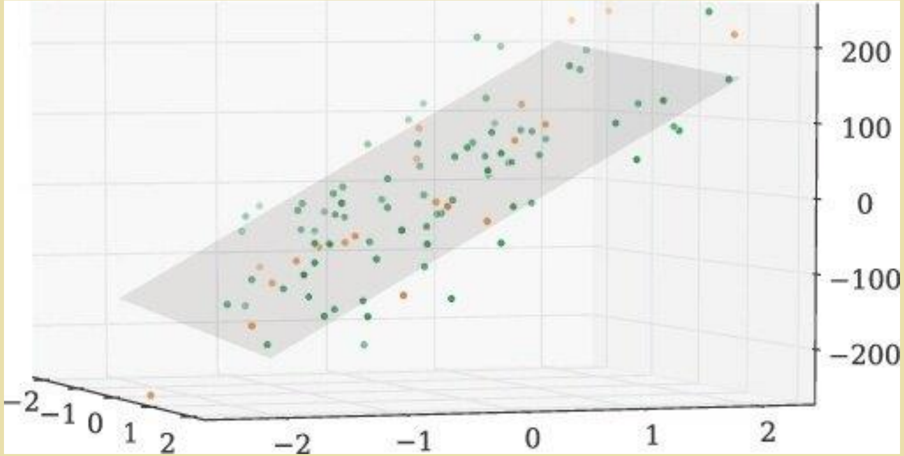


Linear Regression

Much like Simple Linear Regression, Multiple Linear Regression works by generate equation with parameters. In Multiple Linear Regression, however, we could have any number of parameters. Let's take a look at Multiple Linear Regression's equation to visualize this.

$$Y = a_1X_1+a_2X_2+...+a_nX_n+b$$

- Y is dependent variable
- X_1 through X_n are independent variables
- b is the intercept
- a_1 through a_n are coefficient parameters

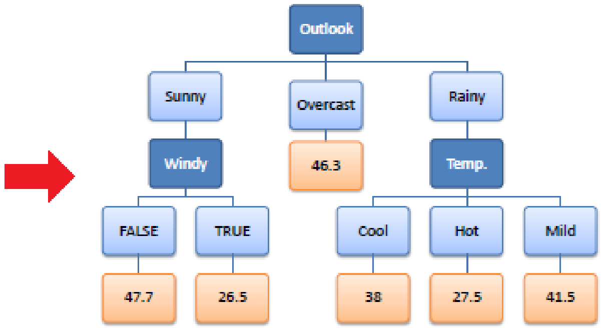


Decision Tree

Decision Tree is one of the most popular machine learning algorithms. Decision Tree not only provided for classification task but could also be used for regression task.

Similar to DT in classification task, DT for regression will try to make a structure from features. We have to calculate and determine the feature for top of tree and features in branch.

Predictors				Target
Outlook	**Temp.**	**Humidity**	**Windy**	**Hours Played**
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Decision Tree

Standard Deviation

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). We use standard deviation to calculate the homogeneity of a numerical sample. If the numerical sample is completely homogeneous its standard deviation is zero.

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$\mathit{Count} = n = 14$$

$$\mathit{Average} = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$\mathit{Standard\ Deviation} = S = \sqrt{\frac{\sum (x - \bar{x})^2}{n}} = 9.32$$

$$\mathit{Coeffeicient\ of\ Variation} = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

- Standard Deviation (S) is for tree building (branching).
- Coefficient of Variiation (CV) is used to decide when to stop branching. We can use Count (n) as well.
- Average (Avg) is the value in the leaf nodes.

Decision Tree

Standard Deviation for Feature

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

Standard Deviation Reduction

The standard deviation reduction is based on the decrease in standard deviation after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest standard deviation reduction (i.e., the most homogeneous branches).

$$SDR(T,X) = S(T) - S(T,X)$$

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

Decision Tree

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR= 0.48		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

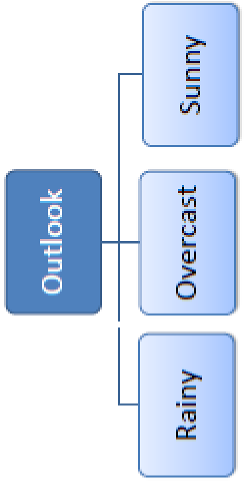
		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Largest SDR

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

Decision Tree

Construct the tree with the feature that contain largest SDR as the top of tree

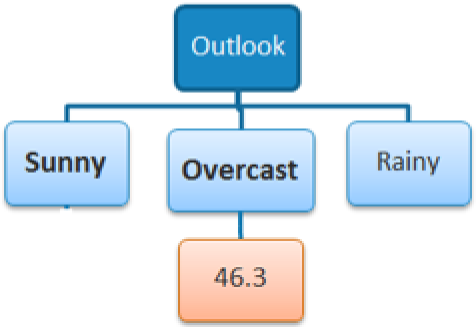


Outlook	Temp.	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

Decision Tree

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



"Overcast" subset does not need any further splitting because its CV (8%) is less than the threshold (10%). The related leaf node gets the average of the "Overcast" subset.

Decision Tree

Outlook - Sunny

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Cool	Normal	TRUE	23
Mild	Normal	FALSE	46
Mild	High	TRUE	30
			S = 10.87
			AVG = 39.2
			CV = 28%

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

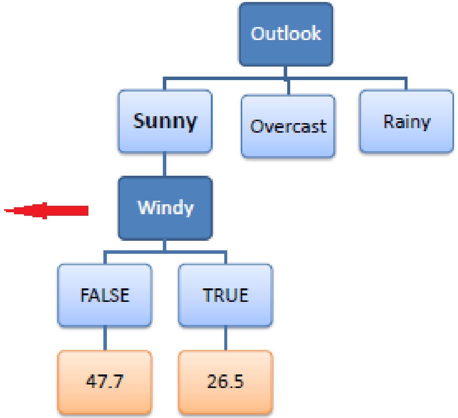
SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

In "Sunny" branch has an CV (28%) more than the threshold (10%) which needs further splitting. We select "Temp" as the best best node after "Outlook" because it has the largest SDR.

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Mild	Normal	FALSE	46
Cool	Normal	TRUE	23
Mild	High	TRUE	30



Because the number of data points for both branches (FALSE and TRUE) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.

Decision Tree

Now, calculate the SDR to determine the best node after “Rainy”. We select “Temp” as the largest SDR. Because the number of data points for all three branches (Cool, Hot and Mild) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.

Outlook - Rainy

Temp	Humidity	Windy	Hours Played
Hot	High	FALSE	25
Hot	High	TRUE	30
Mild	High	FALSE	35
Cool	Normal	FALSE	38
Mild	Normal	TRUE	48
			S = 7.78
			AVG = 35.2
			CV = 22%

		Hours Played (StDev)	Count
Temp	Cool	0	1
	Hot	2.5	2
	Mild	6.5	2

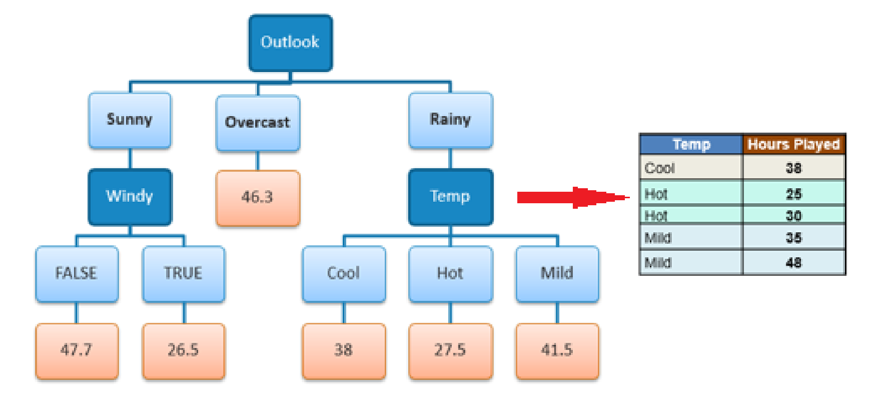
SDR = 7.78 - ((1/5)*0+(2/5)*2.5 + (2/5)*6.5) = 4.18

		Hours Played (StDev)	Count
Humidity	High	4.1	3
	Normal	5.0	2

SDR = 7.78 - ((3/5)*4.1 + (2/5)*5.0) = 3.32

		Hours Played (StDev)	Count
Windy	False	5.6	3
	True	9.0	2

SDR = 7.78 - ((3/5)*5.6 + (2/5)*9.0) = 0.82



Support Vector Machine

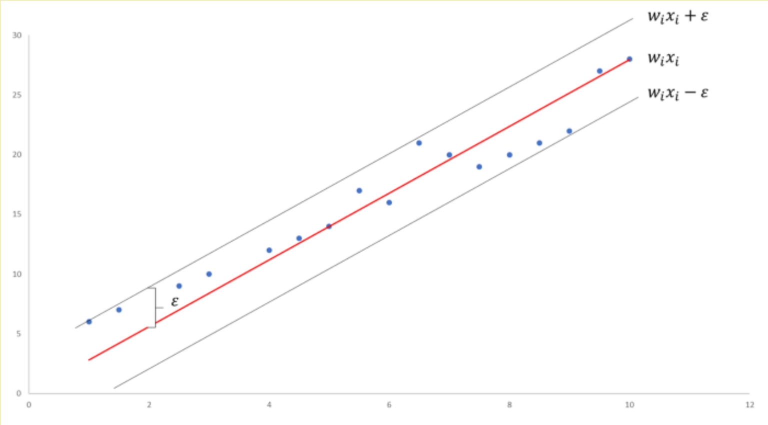
Just like Decision Tree, Support Vector Machine could also be used for regression task.

Support Vector Machine that be used as a regression task, maintaining all the main features that characterize the algorithm (maximal margin). In this case we use the same principles as the SVM for classification, with only a few minor differences.

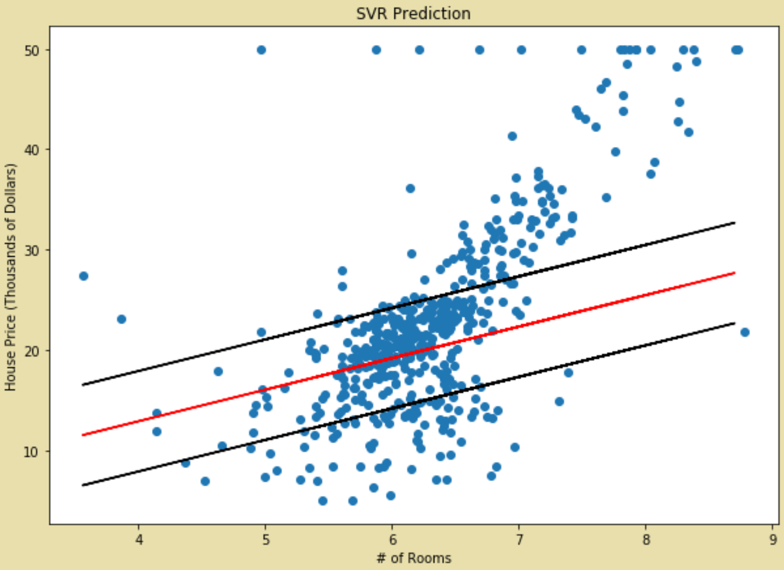
Se set the absolute error less than or equal to a specified margin, called the maximum error, ϵ (epsilon). We can tune epsilon to gain the desired accuracy of our model.

Minimize: $$MIN\ \frac{1}{2}||\mathbf{w}||^2$$

Constraints: $$|y_i - w_i x_i| \leq \varepsilon$$



Support Vector Machine

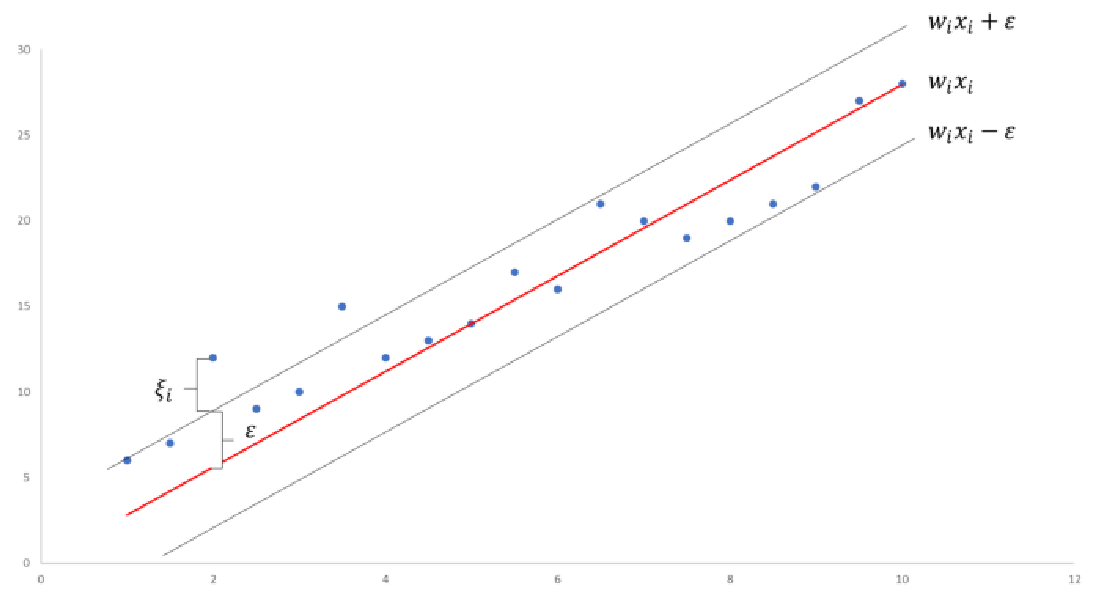


With the illustration, we can see that that this algorithm doesn’t work for all data points. The algorithm solved the objective function as best as possible but some of the points still fall outside the margins. As such, we need to account for the possibility of errors that are larger than ϵ.

Support Vector Machine

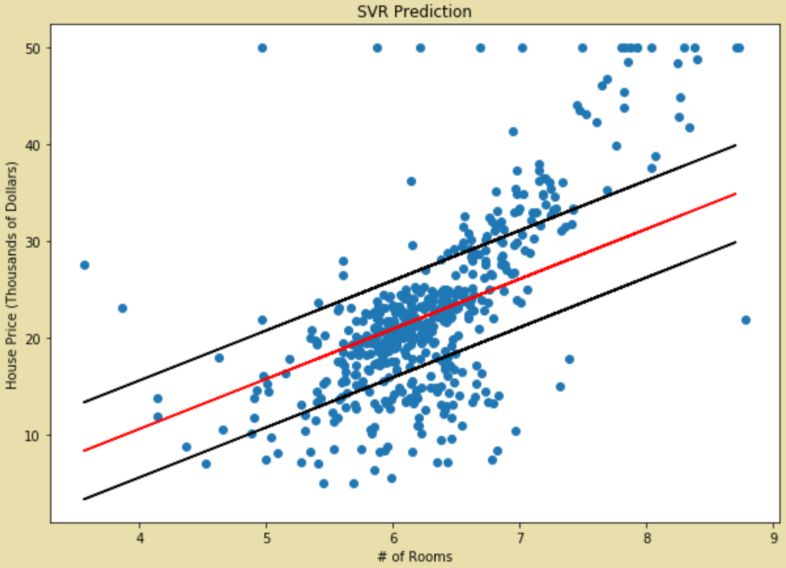
Minimize: $$MIN\ \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n}|\xi_i|$$

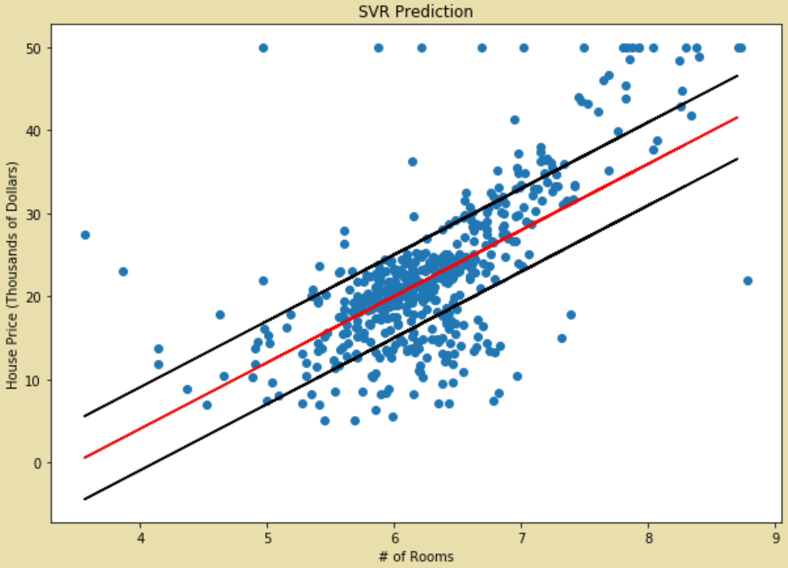
Constraints: $$|y_i - w_i x_i| \leq \varepsilon + |\xi_i|$$



We now have an additional hyperparameter, C, that we can tune. As C increases, our tolerance for points outside of ϵ also increases. As C approaches 0, the tolerance approaches 0 and the equation collapses into the simplified (although sometimes infeasible) one.

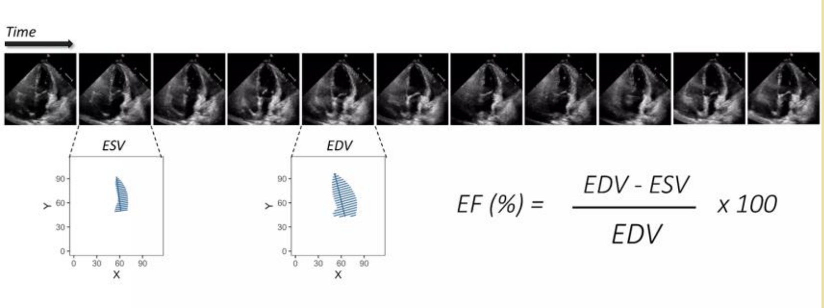
Support Vector Machine





Real Case In Regression Task

Eject Fraction Prediction From Echocardiogram Videos



House Price Prediction

Crowd Counting

Forecasting

Conclusion

- With Machine Learning we can build models that are very important in an intelligent system or AI
- In regression we can estimate the continuous value from the feature given.
- Not all of the cases in regression task could solve with linear regression, maybe we need nonlinear form.
- In industrial cases, regression not only can be used in structured data but also can be used for unstructured data such as images, videos, signal, etc.

References

https://www.saedsayad.com/

https://towardsdatascience.com/

https://en.wikipedia.org/wiki/Machine_learning