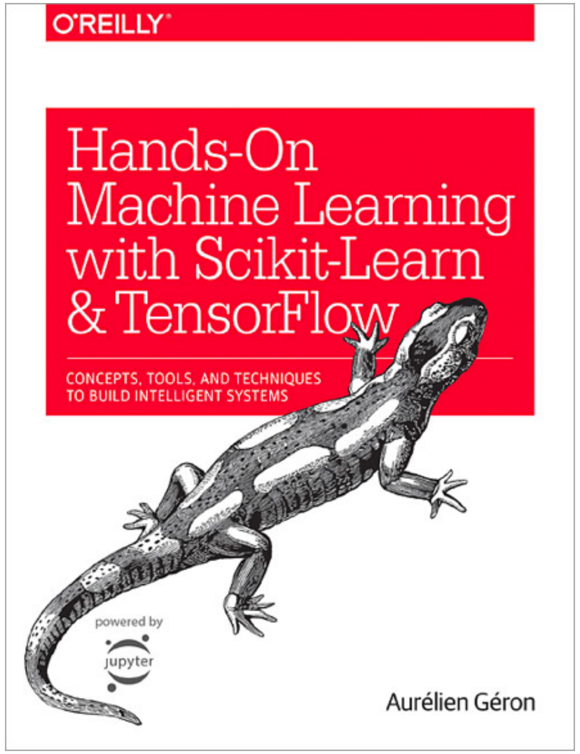
Part 1:
Machine Learning Basics
機械学習の基礎

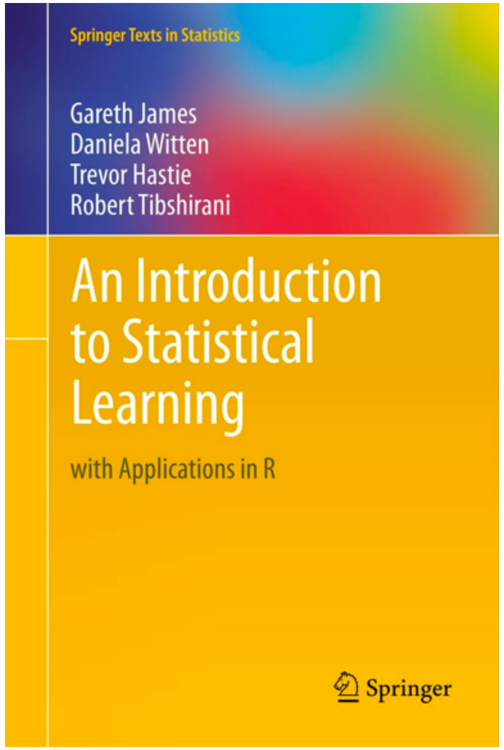
WILLIAM STEIMEL
スタイメルウィリアム

Table of Contents　目次

Sources



Chapter 1,2,4

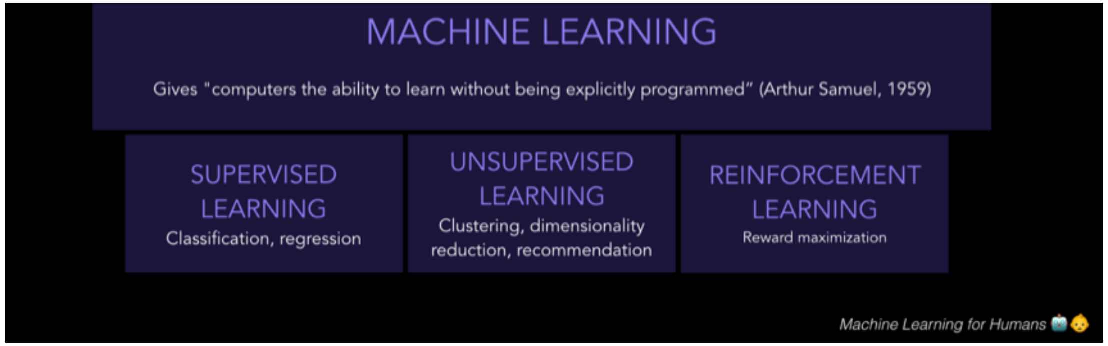


Chapter 2/3

Wine Quality Dataset:
UCI Machine Learning Repository

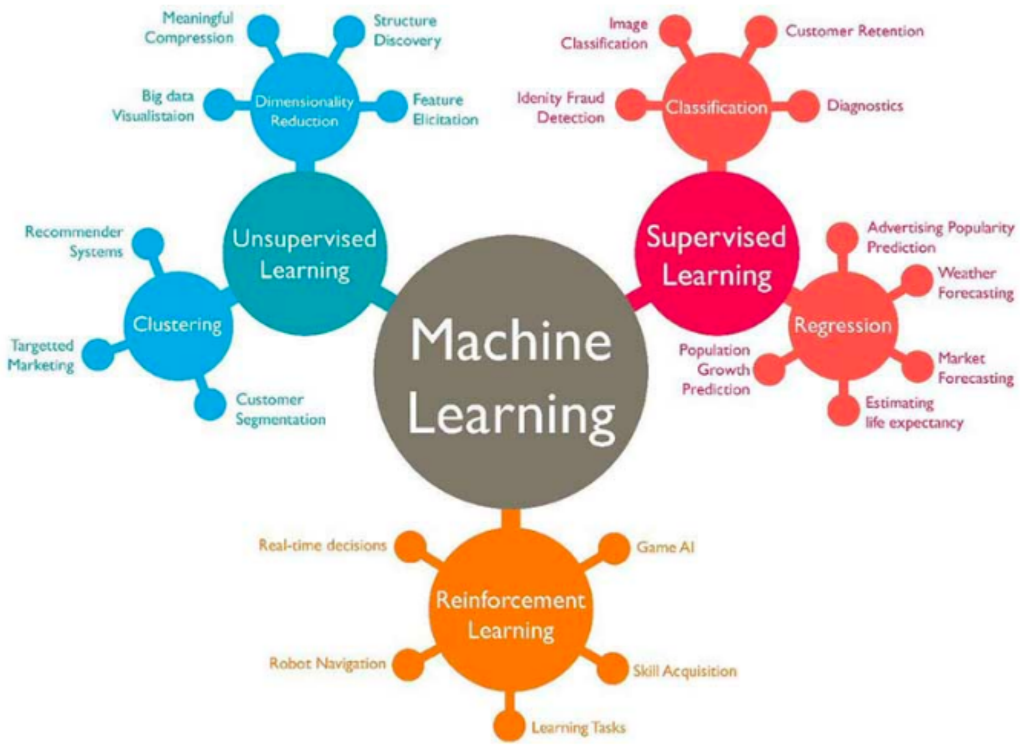
What is Machine Learning?
機械学習とは？

- **2 Famous Definitions**
  - ”Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.” – Arthur Samuel 1959
  - “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” - Tom Mitchell, 1997
- In Summary- Machine Learning is the science and art of programming computers to learn from data.



Types of Machine Learning Systems
機械学習の基礎

- Supervised Learning　（教師あり学習）
- Unsupervised Learning　（教師なし学習）
- Reinforcement Learning　（強化学習）

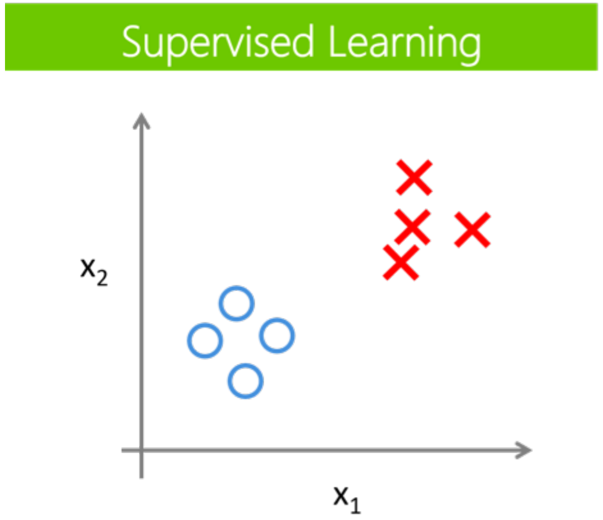


Supervised Learning
教師あり学習

- “**Supervised learning** is the machine **learning** task of inferring a function from labeled training data. “
  - The training data you feed the algorithm includes the desired solutions/labels.
  - Develop a predictive model based on both the input and output data
  - Includes Classification / Regression Problems

Sample Algorithms

- K-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Regression
- Decision Trees and Random Forests
- Neural Networks (Sometimes)



Supervised Learning Labels
教師あり学習

Labels

y

```
In [326]: data.head()
```

Out[326]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Characteristics/Predictive Features

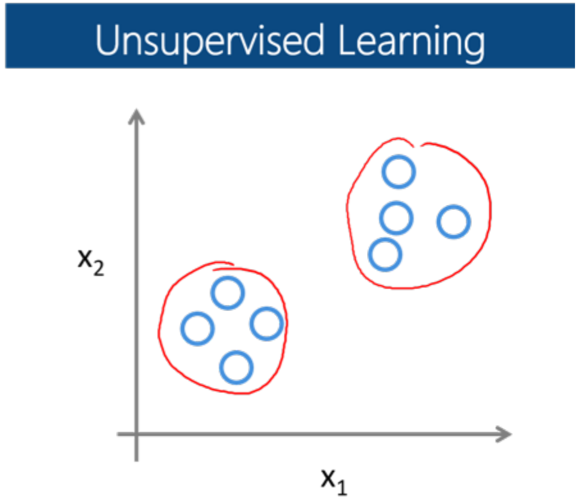
X1 - Xn

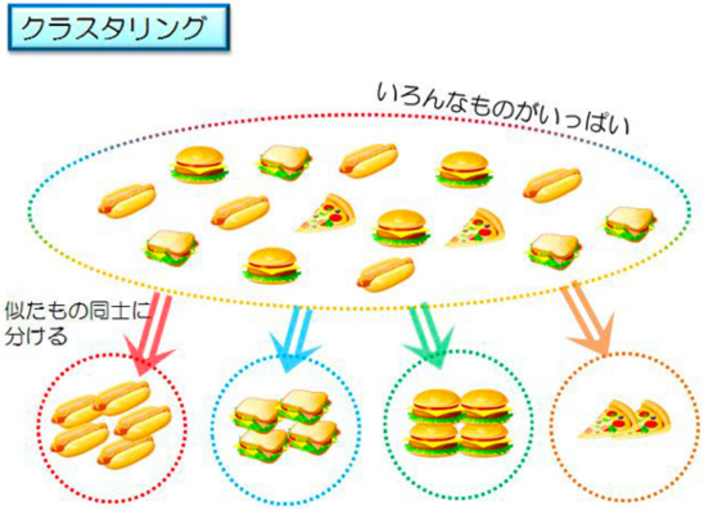
Unsupervised Learning
教師なし学習

- "**Unsupervised machine learning** is the machine learning task of inferring a function to describe hidden structure from "unlabeled" data"
  - The training data you feed the algorithm does not include labels and tries to learn without a teacher. (教師なし)
  - Group and interpret data based on only input data/characteristics

Sample Algorithms

- **Clustering**
  - K-means
  - Hierarchical Cluster Analysis (HCA)
  - Expectation Maximization
- **Visualization and dimensionality reduction**
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - T-distributed Stochastic Neighbor Embedding (t-SNE)
- **Association rule learning**
  - Apriori
  - Eclat





Reinforcement Learning
強化学習

- "**Reinforcement learning** (**RL**) is an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take *actions* in an *environment* so as to maximize some notion of cumulative *reward*."
  - The Learning System is called the "agent"
  - The Agent Learns the best strategy or "policy" to get the most reward over time
  - Policy defines what action the agent takes at each step

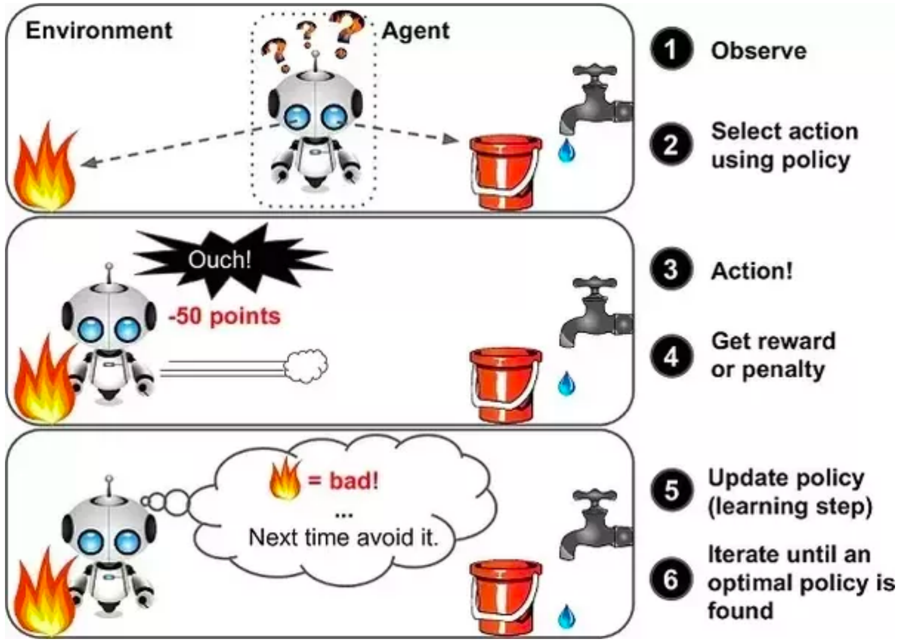


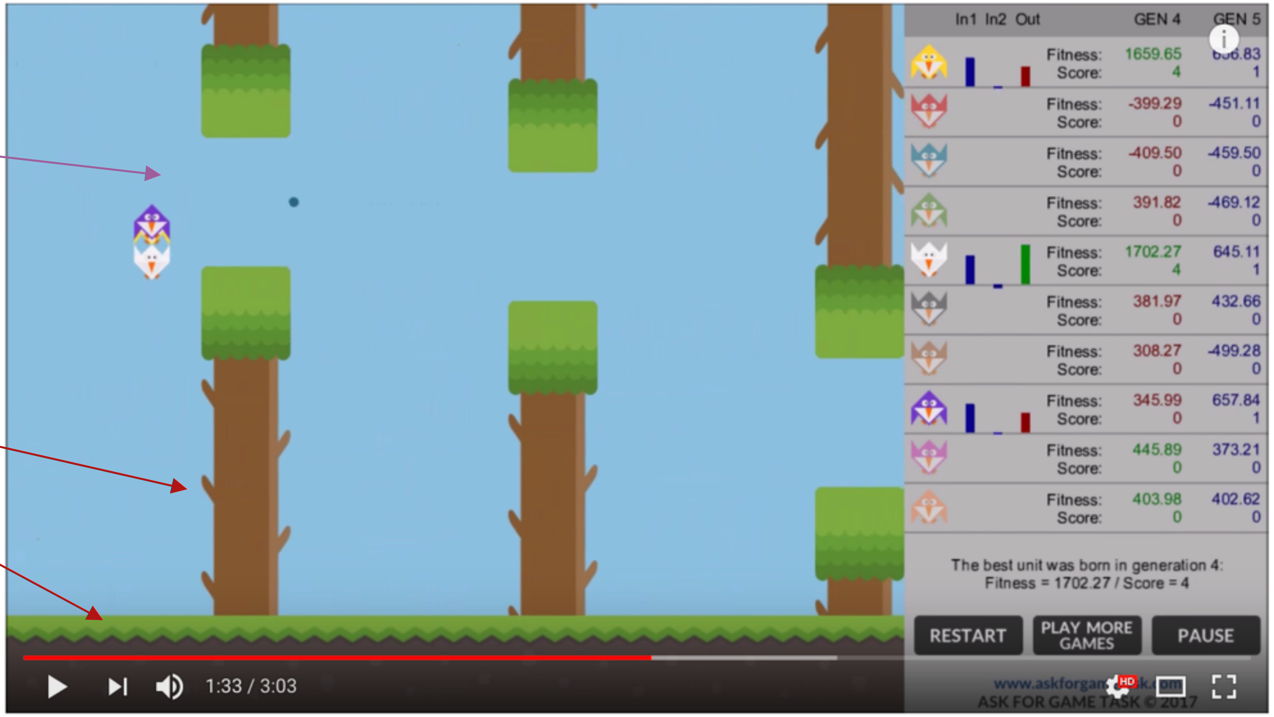
Diagram from Hands On Machine Learning with SciKit Learn & Tensor Flow

Reinforcement Learning Flappy Bird

Reward

Penalty

Penalty



Machine Learning for Flappy Bird using Neural Network & Genetic Algorithm

From YouTube <<

Flappy Bird

Batch vs Online Learning

- Machine Learning systems are also categorized based on their learning methods

- Batch Learning
  - The system must be trained using all available data (batch)
  - *The system is trained on all of the data then launched into production environment.*
    - Requires a lot of computing resources (Storage, Memory, etc.)
    - Not as reactive as online learning
- Online Learning
  - Continuous stream of data
  - Learns incrementally via new data and previous predictions (Good for stock data/other quickly changing systems)

Instance vs Model Based Learning

- Another way to categorize machine learning systems is based on how they generalize

- Two approaches to generalization
  - Instance-Based Learning
    - Machine Learning System learns the examples by heart, then classifies new cases using a measure of similarity
    - Spam E-mail example: The amount of words two e-mails have in common.
    - Example: K-nearest Neighbor Algorithm
  - Model-Based Learning
    - Machine Learning System learns via building a model of examples to make predictions
    - Example: Linear Model

Machine Learning Challenges

- Not enough Training Data
- Non-representative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting/ Under fitting

Need More Training Data !

- Data is important for machine learning (But Training Data can sometimes be hard to get)
- Insufficient data can affect machine learning models
- Effectiveness of Data
  - A study by Microsoft researchers revealed that different machine Learning algorithms performed almost identically when given enough data.

Non-Representative Training Data

- When training a machine learning model it is important for the training set to be representative of the predictions desired.
  - Example: If we train an algorithm to classify an observation as hotdogs, hamburgers, and Pizza. If the algorithm is trained on a dataset that does not include any pizza observations, It will be impossible to predict the observation is a pizza.
- Small Samples – **Sampling Noise** can occur due to chance
  - (Not enough observations so the training set is flawed)
- Large Samples- Flawed Sampling Methods can lead to Sampling Bias
  - **Sampling Bias** – Choosing a Sample is flawed which leads to inaccurate predictions
    - Example: We are building a predictive model to predict who will become prime minister of Japan in the next election and we only interview students at Sophia University.
    - There will be a sampling bias towards young people and the populace will differ greatly.

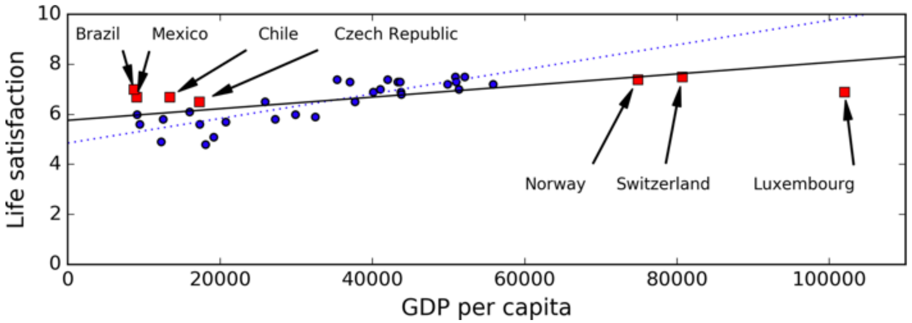


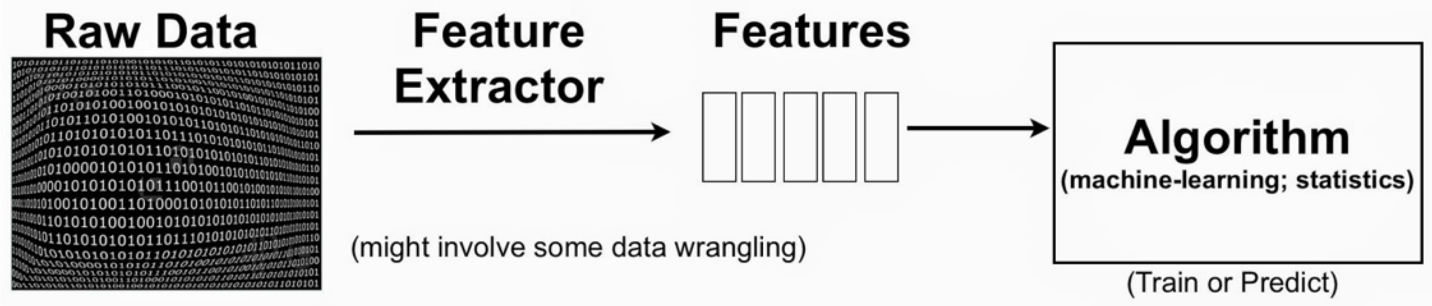
Diagram from Hands On Machine Learning with SciKit Learn & Tensor Flow

Poor-Quality Data

- "Garbage in – Garbage out"
  - If training data is full of errors, outliers, and noise the predictive model will not perform well
- If possible Outliers/Null values should be removed or adjusted
- Dirty data is very common

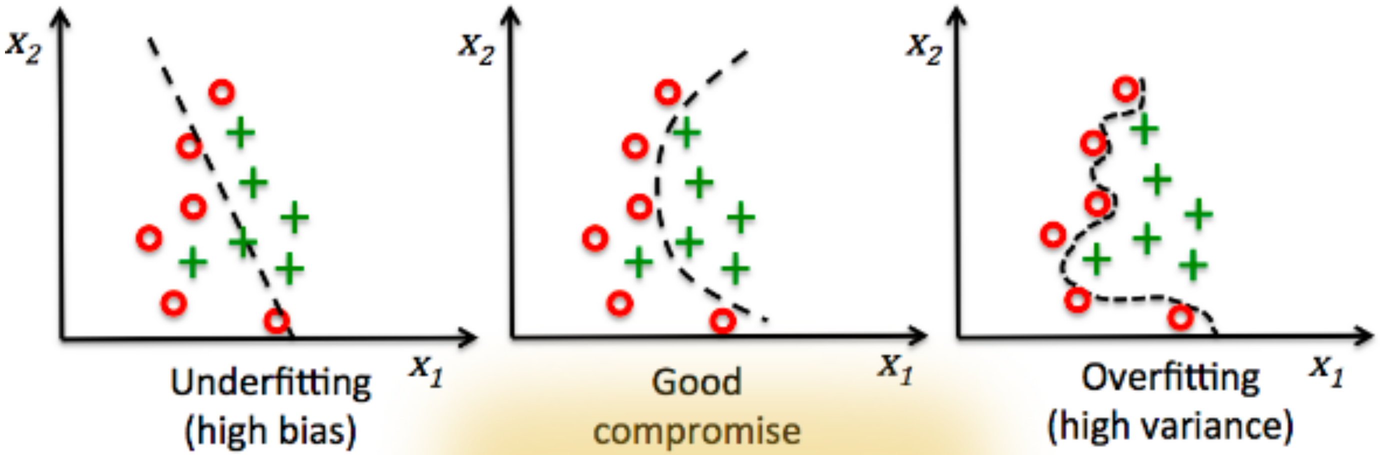
Irrelevant Features

- A critical part of machine learning is training a model on good features
- Feature Engineering – Transforming Data/Analysis to make the ideal inputs for prediction
- Feature Engineering is important !
  - Feature Selection (Select the most relevant/correlated features)
  - Feature Extraction (Combining existing features to make more useful ones)
  - Creating new features by gathering data



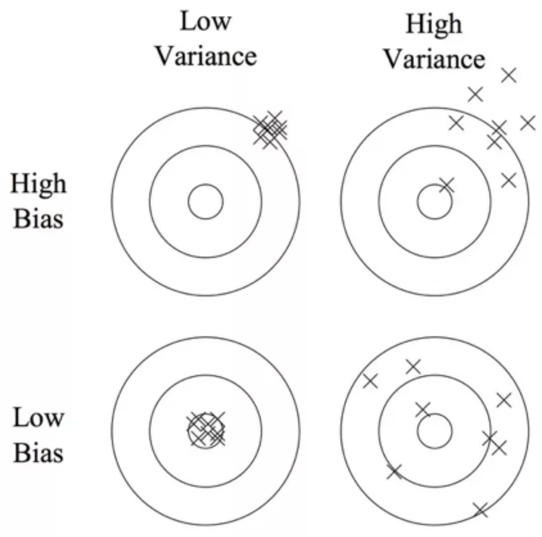
Overfitting/Underfitting

- Overfitting and Under fitting are two big challenges to Machine Learning



高バイアス

高バリアンス

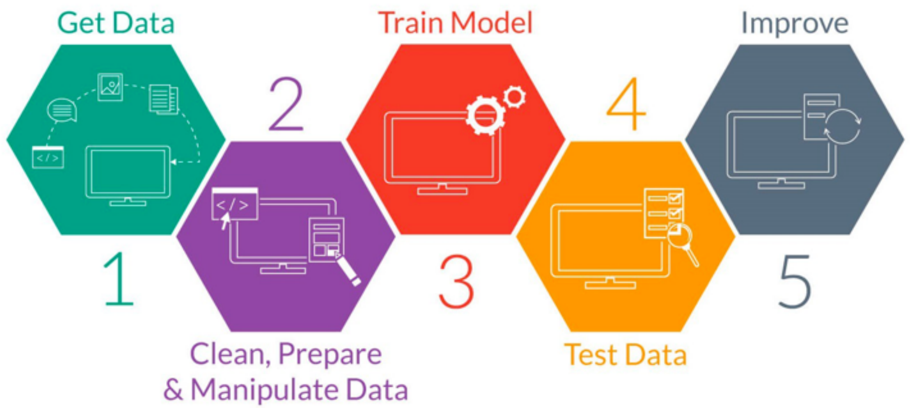


Part 2:
End to End Machine Learning
エンドツーエンド機械学習

End to End Machine Learning
エンドツーエンド機械学習

- The Machine Learning Process (Varies)
  - Look at the big picture
  - Get the data
  - Discover and visualize the data to gain insights
  - Prepare the data for Machine Learning algorithms
  - Select a model and train it
  - Fine-tune your model
  - Present your solution

Typical Workflow



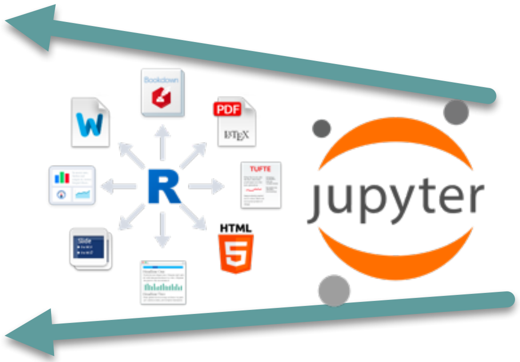
End to End Machine Learning
エンドツーエンド機械学習

- https://github.com/steimel64/steimel64.github.io

Data Scientist/Machine Learning Engineer/Research Student

kaggle

GitHub



Motivations
動機

- I implemented an End to End Machine Learning Project Last presentation but wanted to study the best practices and improve upon my process
- The book "Hands-On Machine Learning with Scikit-Learn & Tensor Flow' implements a very thorough end to end Machine Learning project with a California Census Dataset
- Before studying more advanced algorithms, I feel that process will be key to creating meaningful machine learning projects
- I will use a project I am currently working on to illustrate the End to End Process called **Regression Applications to Wine Quality Prediction**

乾杯！

Step I: Look at the big picture

- Frame the Problem –
  - Summarize the dataset (What Data? Where is it From?)
  - Define the Problem (What is the problem we are trying to solve?)
  - Define the Objective (What do we want to accomplish?)
  - Software/Libraries (What software/packages were used in this project?
- Select a performance measure to evaluate all models on
  - MSE
  - Accuracy Rate/Confusion Matrix
  - Etc.
- Check the Assumptions
  - Talk with stakeholders who are familiar with the data
  - General Knowledge about the area

Step II: Get the Data

- Download the Data/Import Via Python
- Take a quick look at the Data Structure (Best Functions are head(), describe(),info()
  - Attributes
  - Data Description
  - Numerical Attributes (Count, Mean, STD, Min, 25 %, 50 %, 75 %, Max)
- Get a feel for the data (Correlation Analysis/Histograms/Scatter Plots)
  - Exploratory Analysis (Visualization)
- **Create a test set**

Step III: Discover and visualize the data to gain insights

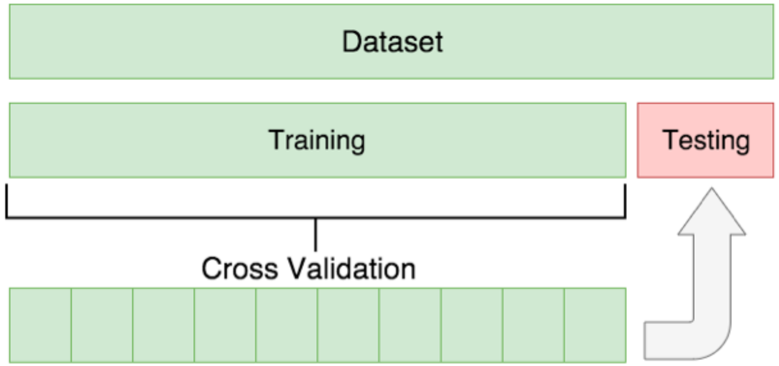
- Training Set
  - Visualize the data more in depth
  - Look for Correlations
  - Experiment with Attribute Combinations

Step IV: Prepare the data for Machine Learning algorithms

- Data Cleaning
    - Remove Nulls
    - Get rid of whole attributes
    - Set values to some value (mean,median,0 etc.)
  - Handling Text and categorical Attributes
- Feature Scaling (Normalizing Features)
- **Transformation Pipeline (Automate the data cleaning/feature scaling process via functions)**

Step V: Select a model and train it

- Training and Evaluating on the Training set
  - Test your Model's effectiveness on the training sets (Pick a model that may solve the problem- Linear Regression, Decision Tree, SVR etc.)
- **Better Evaluation Using Cross-Validation**
  - **Score Each Model with an evaluation Metric (Accuracy, RSE, etc.)**
  - **Average all scores to get accuracy**



Source: Joseph Nelson

Step VI: Fine-tune your model

- HyperParameter Tuning
  - Grid Search (Yagi Kun)
  - Randomized Search (Method similar to Randomized Search)
- Ensemble Models (Combine Models to improve results)
- Analyze the Best Models and Their Errors
- Evaluate Your Model on the test set

Step VII: Present your solution

- Revisit the Problem Statement
- In the business world we often need to present to stakeholders/executives
- Highlight what was learned
  - What Worked ?
  - What Didn't Work?
  - What system Limitations existed?
  - Document Everything
  - Create nice presentations with clear visualizations

Jupyter Notebook Project

Wine Project End-to-End

https://github.com/steimel64/steimel64.github.io/blob/master/Notebooks/Wine.ipynb

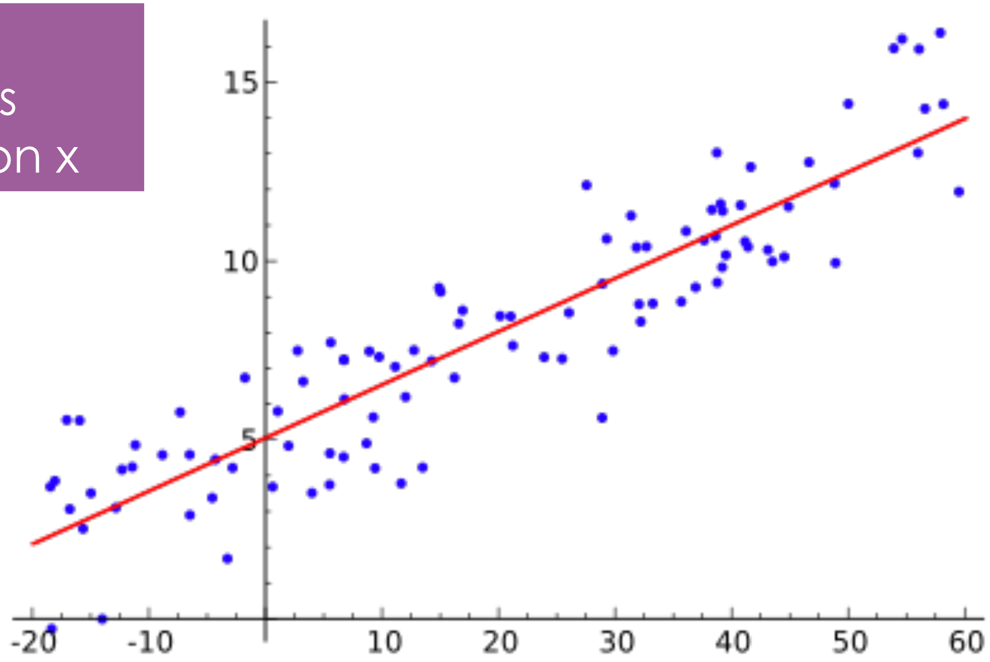
Part 3- Linear Regression

Regression (A form of Supervised Learning)

Regression Problems predict a **continuous target variable Y**. It allows you to estimate a value, such as housing prices or stock exchange, based on input data X.

Simple Linear Regression

- Simplest Approach to Supervised Learning
- The task of fitting a straight line through a set of points
- Assumes outcome of y is a linear relationship based on x



Linear Regression Benefits

- Widely used
- Runs fast
- Easy to use (not a lot of tuning required)
- Highly interpretable
- Basis for many other methods (Linear regression concepts are very important foundations for other forms of supervised learning techniques)

Supervised Learning: Regression

Goal/Problem: Predict how many Japanese Vocabulary(y) a student knows after 2 and 4 years of Japanese Studies(X).

Training Set

Observation #	Years Studying Japanese (X)	Vocabulary Total (y)
1	3	3,000
2	2	2,300
3	1	1,000
4	6	5,400

Test Set

Observation #	Years Studying Japanese (X)	Vocabulary Total (y)
1	4	???
2	2	???

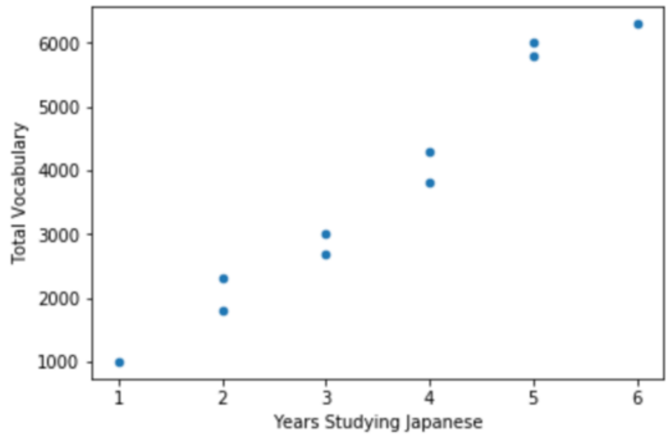
Sample Dataset (Japanese Studies)

10 Students

Observation	Years Studying Japanese	Total Vocabulary
1	3	3000
2	2	2300
3	1	1000
4	6	6300
5	5	6000
6	5	5800
7	4	4300
8	2	1800
9	3	2700
10	4	3800

In [64]:
```
data.plot(kind='scatter',x='Years Studying Japanese', y='Total Vocabulary')
```

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x1c186aba20>



Linear

Sample Dataset (Continued)

- Questions we might ask:
  - Is there a relationship between years studying Japanese and how many vocabulary a person has?
  - How accurately can we predict a person's vocabulary total?
  - Is the relationship linear?

Observation	Years Studying Japanese	Total Vocabulary
1	3	3000
2	2	2300
3	1	1000
4	6	6300
5	5	6000
6	5	5800
7	4	4300
8	2	1800
9	3	2700
10	4	3800

Simple Linear Regression

Simple linear regression is an approach for predicting a **quantitative response** using a **single feature** (or "predictor" or "input variable")

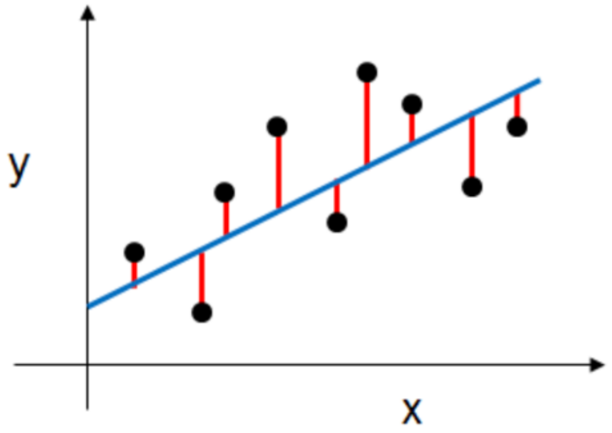
$$y = \beta_0 + \beta_1 x$$

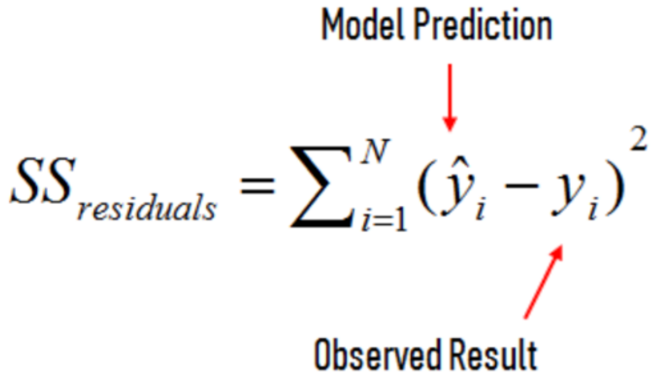
- y is the response
- x is the feature
- β_0 is the intercept
- β_1 is the coefficient for x

Total Vocabulary

Years Studying Japanese

Finding the Optimal Line





Coefficients (B0 & B1) are estimated using the **least squares criterion**, which means we find the line (mathematically) which minimizes the **sum of squared residuals** (or "sum of squared errors"):

- The black dots are the **observed values** of x and y.
- The blue line is our **least squares line**.
- The red lines are the **residuals**, which are the distances between the observed values and the least squares line.

Interpreting Model Coefficients

$$y = \beta_0 + \beta_1 x$$

Out[63]:

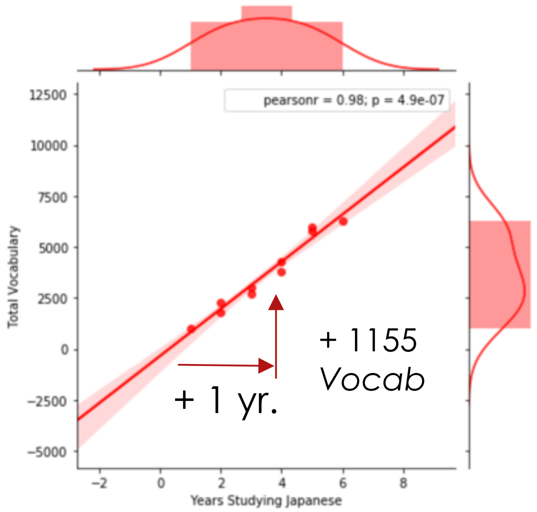
OLS Regression Results

Dep. Variable:	Total Vocabulary	**R-squared:**	0.964
Model:	OLS	**Adj. R-squared:**	0.959
Method:	Least Squares	**F-statistic:**	211.7
Date:	Tue, 05 Dec 2017	**Prob (F-statistic):**	4.88e-07
Time:	22:06:35	**Log-Likelihood:**	-72.390
No. Observations:	10	**AIC:**	148.8
Df Residuals:	8	**BIC:**	149.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-344.4444	302.449	-1.139	0.288	-1041.893	353.004
Years Studying Japanese	1155.5556	79.427	14.549	0.000	972.397	1338.714

Omnibus:	1.138	**Durbin-Watson:**	1.278
Prob(Omnibus):	0.566	**Jarque-Bera (JB):**	0.684
Skew:	0.157	**Prob(JB):**	0.710
Kurtosis:	1.758	**Cond. No.**	10.2

Beta 0: -344.4444 (intercept)
Beta 1: 1155 *(Model adds 1155 vocab per every year of Japanese Studies)*



Observation #	Years Studying Japanese (X)	Vocabulary Total (y)
1	4	**4964 Y(hat)**
2	2	**2654 Y(hat)**

Assumptions of Linear Regression

- Linear Relationship
  - Input and output relationship needs to be linear
- Multivariate Normality
  - Multiple regression assumes that the residuals are normally distributed.
- No or little multicollinearity
  - Multicollinearity occurs when the independent variables are too highly correlated with each other
  - Risk of over-fit data with highly correlated input variables
- No Auto-Correlation
- Homoscedasticity
  - No Outliers (Variance cannot be high around the regression line)

Multiple Linear Regression

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

- Simple Linear Regression can be extended to Multiple Features
- Each feature gets its own coefficient (n features)
- In Multiple Linear Regression, the least squares regression line becomes a plane
- Goal is also to reduce the RSS like simple linear model

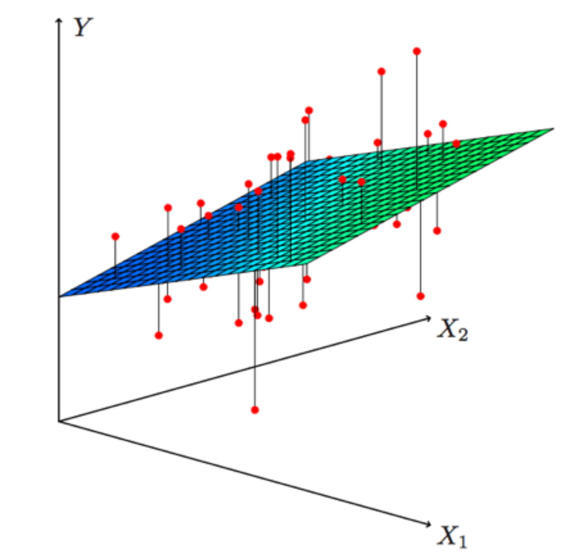
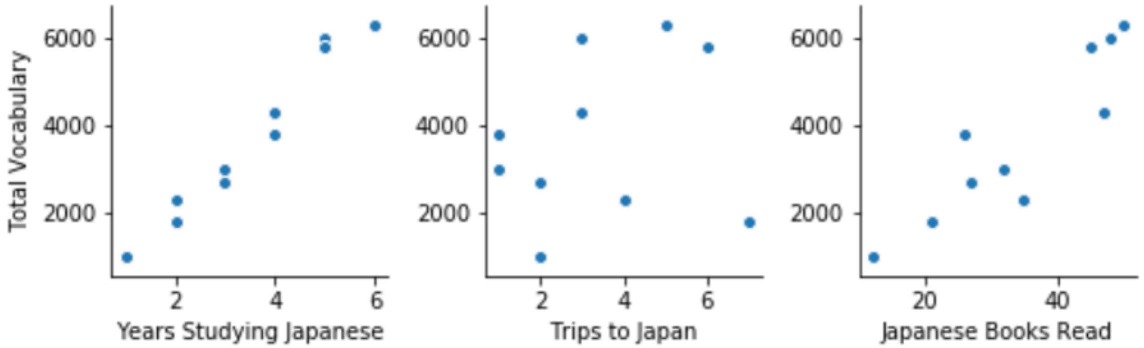


Figure Source: An Introduction to Statistical Learning

Sample Dataset (Japanese Studies)

Years Studying Japanese	Trips to Japan	Japanese Books Read	Total Vocabulary
3	1	32	3000
2	4	35	2300
1	2	12	1000
6	5	50	6300
5	3	48	6000
5	6	45	5800
4	3	47	4300
2	7	21	1800
3	2	27	2700
4	1	26	3800

10 Students



Linear

No Correlation

Somewhat Linear

Multiple Linear Regression Model (Calculation/Coefficients)

Coefficients

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

In [101]:
```
lm1.params
```

Out[101]:
```
Intercept                  -698.496461
YearsStudyingJapanese       938.507576
TripstoJapan                 21.714596
JapaneseBooksRead            30.317502
```

In [106]:
```
## Manual Calculation
##2 Years Studying Japanese
##5 Trips to Japan
##26 Japanese Books Read

y = -698 + (938*2) + (21*5) + (30*26)
y
print('It is Predicted that this students total Japanese Vocabulary is',y)
```

```
It is Predicted that this students total Japanese Vocabulary is 2063
```

Can we Improve the model?

Model Summary

```
In [102]: lm1.summary()
Out[102]:
```

OLS Regression Results

Dep. Variable:	TotalVocabulary	**R-squared:**	0.977
Model:	OLS	**Adj. R-squared:**	0.966
Method:	Least Squares	**F-statistic:**	86.30
Date:	Wed, 06 Dec 2017	**Prob (F-statistic):**	2.52e-05
Time:	02:53:18	**Log-Likelihood:**	-70.015
No. Observations:	10	**AIC:**	148.0
Df Residuals:	6	**BIC:**	149.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-698.4965	341.244	-2.047	0.087	-1533.490	136.497
YearsStudyingJapanese	938.5076	139.627	6.722	0.001	596.853	1280.162
TripstoJapan	21.7146	57.552	0.377	0.719	-119.109	162.539
JapaneseBooksRead	30.3175	17.365	1.746	0.131	-12.172	72.807

Omnibus:	1.947	**Durbin-Watson:**	2.094
Prob(Omnibus):	0.378	**Jarque-Bera (JB):**	1.276
Skew:	0.686	**Prob(JB):**	0.528
Kurtosis:	1.915	**Cond. No.**	116.

R-Squared helps determine goodness of fit (97 % fit)

Important Questions Regarding Multiple Linear Regression

- Is at least one of the predictors X1, X2, X3, Xn useful in predicting the response?
  - (Does Years studying Japanese impact Japanese Vocabulary total? Etc.)
- Do all the predictors help to explain Y, or is only a subset of the predictors useful?
  - Do all three predictors explain Total Vocabulary or are only certain predictors relevant?
- How well does the model fit the data?
  - Is this linear model the best model or can we do better?
- Given a set of predictor values, what response value should we predict and how accurate is our prediction?
  - Given a student has studied Japanese for 2 years, has read 20 books, and been to Japan 3 times how many Japanese Vocabulary should they know?

Importance of Feature Selection

- The art of determining Useful Features for predictive model
  - Better Results
  - Simpler Model Interpretation
  - Features impact predictive models used and results

Model Building

- All-In
- Backward Elimination
- Forward Selection
- Bidirectional Elimination

Stepwise Regression

All-In

- Uses all the predictive features in the model
- Preparation for Backward Elimination

All-In Would use all three predictors - Years Studying Japanese, Trips to Japan, and Japanese Books Read in the model

```
In [102]: lm1.summary()
```

Out[102]:

OLS Regression Results

Dep. Variable:	TotalVocabulary	**R-squared:**	0.977
Model:	OLS	**Adj. R-squared:**	0.966
Method:	Least Squares	**F-statistic:**	86.30
Date:	Wed, 06 Dec 2017	**Prob (F-statistic):**	2.52e-05
Time:	02:53:18	**Log-Likelihood:**	-70.015
No. Observations:	10	**AIC:**	148.0
Df Residuals:	6	**BIC:**	149.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-698.4965	341.244	-2.047	0.087	-1533.490	136.497
YearsStudyingJapanese	938.5076	139.627	6.722	0.001	596.853	1280.162
TripstoJapan	21.7146	57.552	0.377	0.719	-119.109	162.539
JapaneseBooksRead	30.3175	17.365	1.746	0.131	-12.172	72.807

Omnibus:	1.947	**Durbin-Watson:**	2.094
Prob(Omnibus):	0.378	**Jarque-Bera (JB):**	1.276
Skew:	0.686	**Prob(JB):**	0.528
Kurtosis:	1.915	**Cond. No.**	116.

Forward Selection

- Stepwise Regression (Adds one feature at a time) – Greedy Approach
- Model starts with no features
- Step 1: Select a significance level to enter in the model (e.g. SL = 0.05)
- In each iteration, add the feature which best improves the model (Lowest RSS)
  - Fit regression model for each independent variable
  - Keep variable with highest significance and perform all possible linear regressions with
    - 2 variables
    - 3 variables
    - 4 variables etc.
  - Stop when the p_value is greater than the designated threshold

Backward Selection

- Stepwise Regression (Removes features one at a time)
- Starts with all features in the model
- Step 1: Select a significance level to stay in the model (e.g. SL = 0.05)
- Step 2: Fit the full model with all predictors
  - Step 3: Remove the value with the highest P-value (If greater than significance threshold)
  - Step 4: Fit Model without this variable and repeat

Complete!

Bi-Directional Elimination (Mixed Selection)

- Stepwise Regression (Adds/Removes one feature at a time)
- Combination of Forward and Backward Selection
- Step 1: Select a significance level to stay/enter in the model (e.g. SL = 0.05)
  - Perform next step of Forward Selection
  - Perform all steps of Backward Elimination
- Model Complete

Evaluating Regression Model Performance

```
In [107]: from sklearn.metrics import mean_absolute_error
          y_true = [3, -0.5, 2, 7]
          y_pred = [2.5, 0.0, 2, 8]
          mean_absolute_error(y_true, y_pred)

Out[107]: 0.5
```

```
In [116]: from sklearn.metrics import mean_squared_error
          mse = mean_squared_error(y_true, y_pred)
          mse

Out[116]: 0.375
```

```
In [118]: import math
          RMSE = math.sqrt(mse)
          RMSE

Out[118]: 0.6123724356957945
```

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Squared Error (MSE) is the mean of the squared errors:

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

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

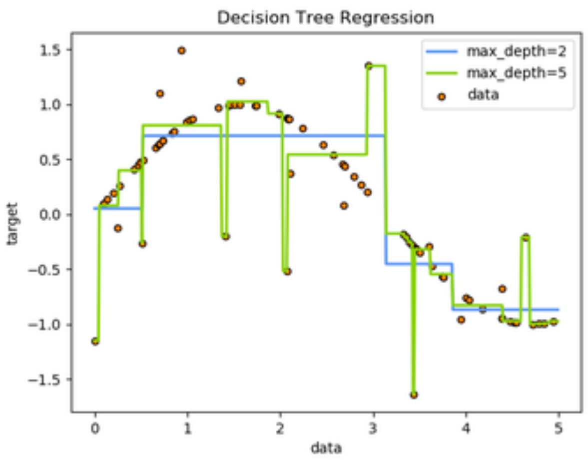
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

To evaluate you would need to compare the values to other models used like Polynomial Regression, SVR, Decision Tree. Etc.

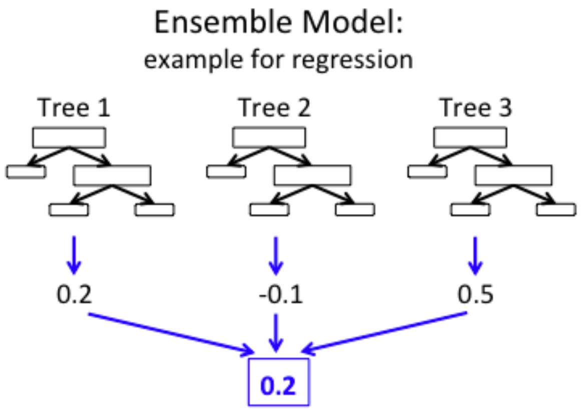
Other forms of Regression (Not Covered Today)

- Decision Tree Regression (Covered Last Week)
- Random Forest Regression (Covered Last Week)
- Support Vector Regression (Will cover with SVM Classification)
- Polynomial Regression

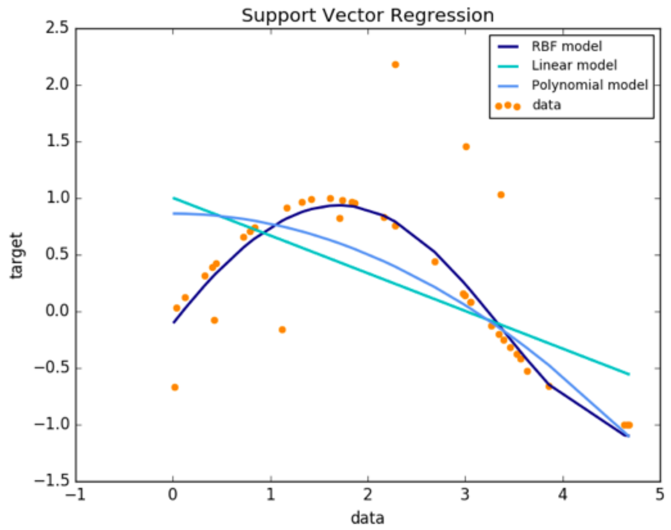
Decision Tree



Random Forest



SVR



Source ScikitLearn