APRENMIN
OTEC

AsGeoMin

Fundamentos de Machine Learning para Geometalurgia

Regression Models

24 de abril al 4 de mayo 2023

Agenda

Machine Learning basis

python™

NumPy

pandas

matplotlib

seaborn

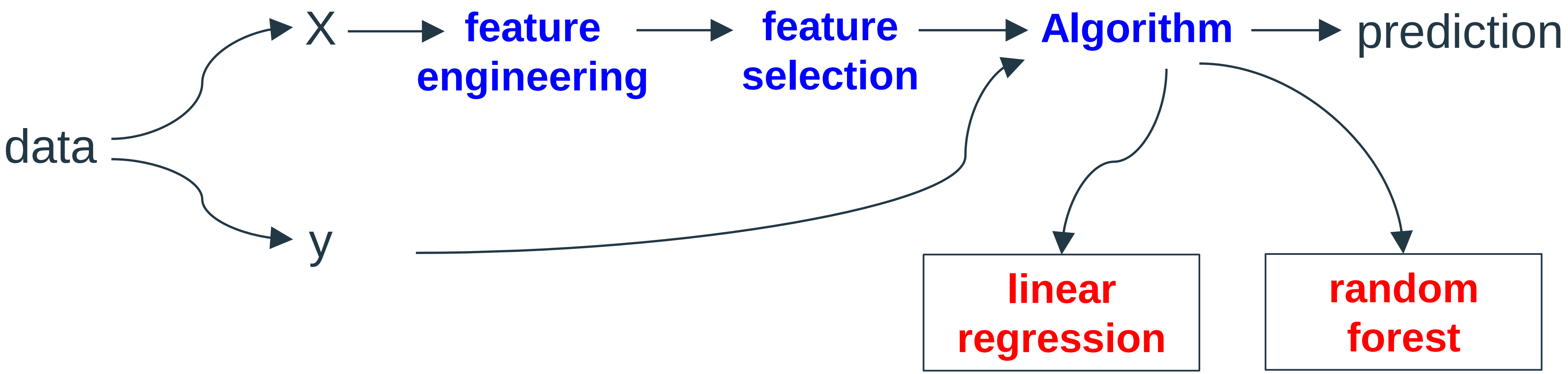
Case study

Univariate
Exploratory Data Analysis (EDA)

Data Preparation

Regression model (proxy) for geometallurgical parameter Ai

Machine Learning Process



data

X → feature engineering → feature selection → Algorithm → prediction

y → Algorithm

Algorithm → linear regression

Algorithm → random forest

X
(predictor matrix)

information such as mineralogy, minzone, etc.

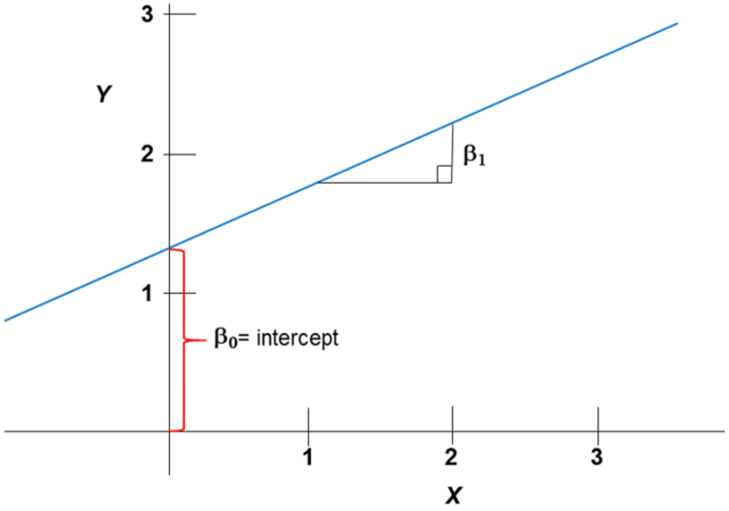
y
(target variable)

Ai values

Algorithm: Linear regression

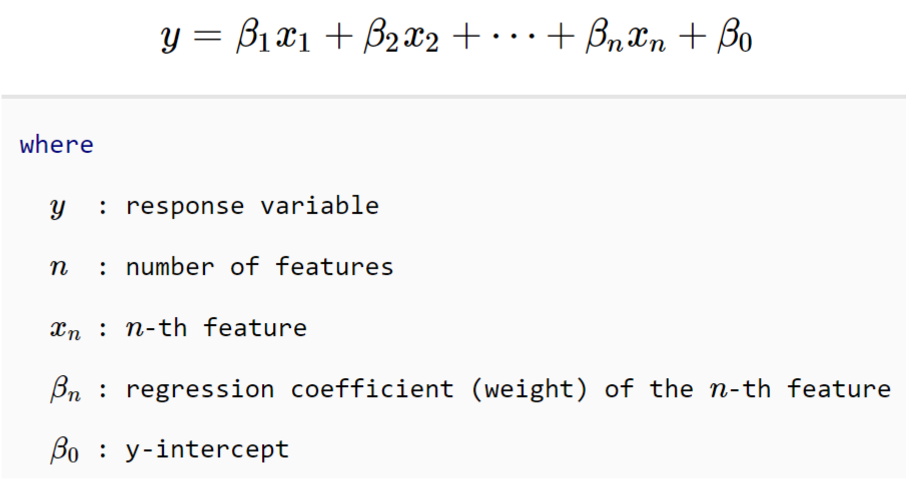
Simple linear regression can be expressed as a function of slope (β1) and the intercept (β1) of a straight line:

$$Y = \beta_0 + \beta_1 X$$



Algorithm: Linear regression

Multiple linear regression model with response Y and terms $X_1,...,X_p$ can be expressed as:



$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \beta_0$$

where

y : response variable

n : number of features

x_n : n-th feature

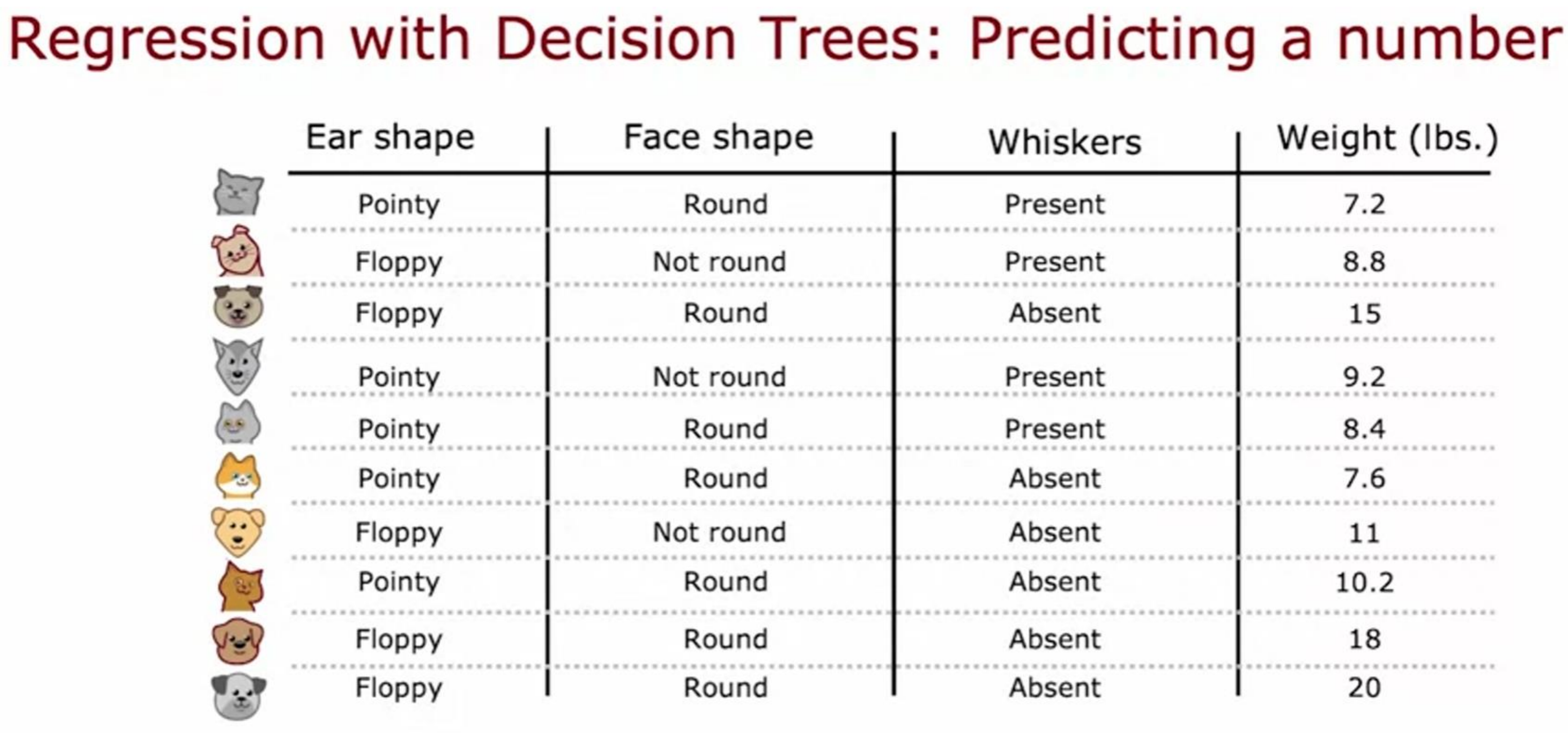
β_n : regression coefficient (weight) of the n-th feature

β_0 : y-intercept

Animation:
https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

Algorithm: regression tree



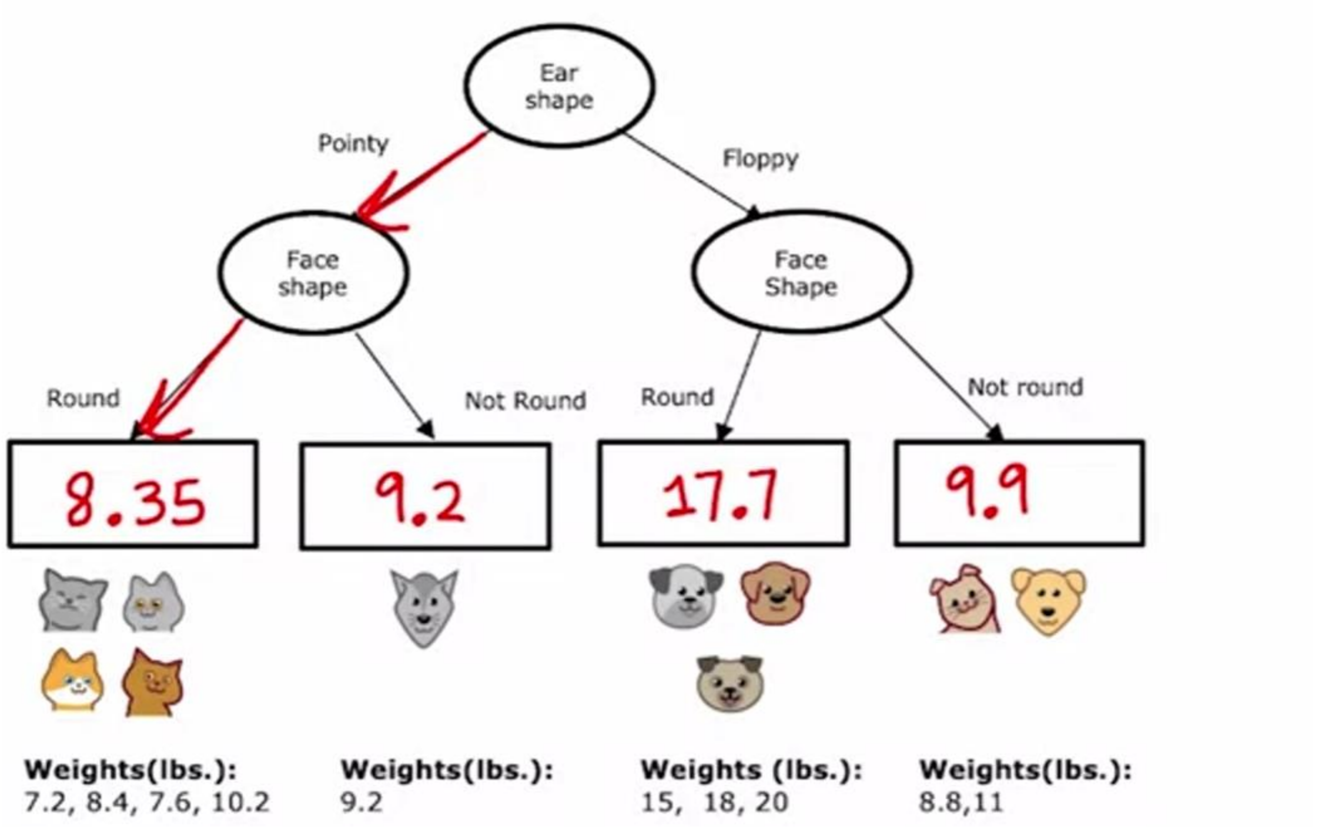
Regression with Decision Trees: Predicting a number

Ear shape	Face shape	Whiskers	Weight (lbs.)
Pointy	Round	Present	7.2
Floppy	Not round	Present	8.8
Floppy	Round	Absent	15
Pointy	Not round	Present	9.2
Pointy	Round	Present	8.4
Pointy	Round	Absent	7.6
Floppy	Not round	Absent	11
Pointy	Round	Absent	10.2
Floppy	Round	Absent	18
Floppy	Round	Absent	20

Andrew Ng, “machine learning specialization” (Coursera)

Algorithm: regression tree

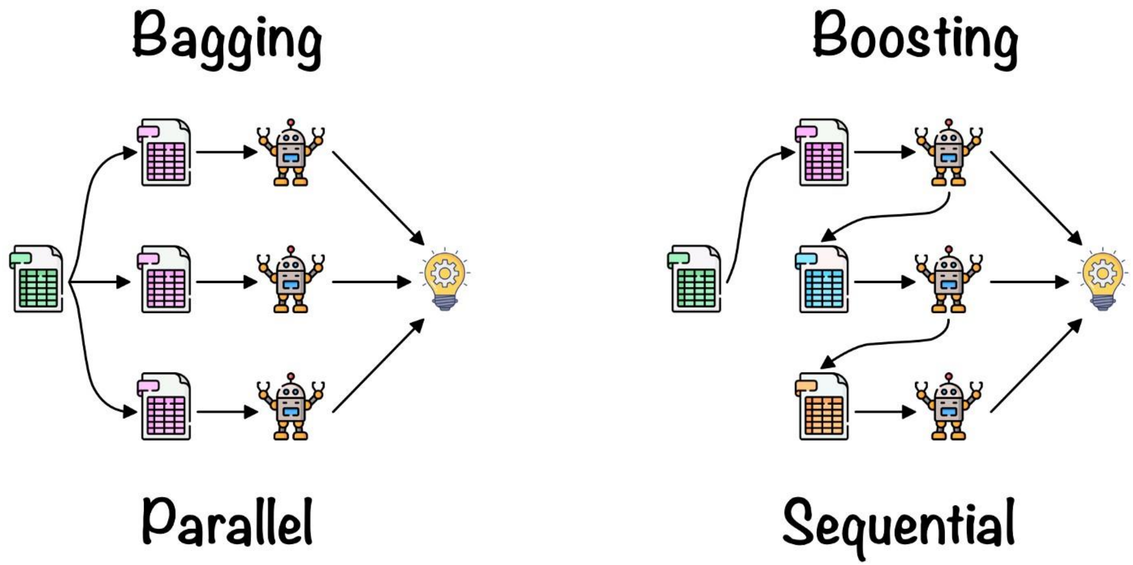
Built through binary recursive partitioning, and then continues splitting each partition into smaller groups.



Andrew Ng, “machine learning specialization” (Coursera)

Algorithm: random forest

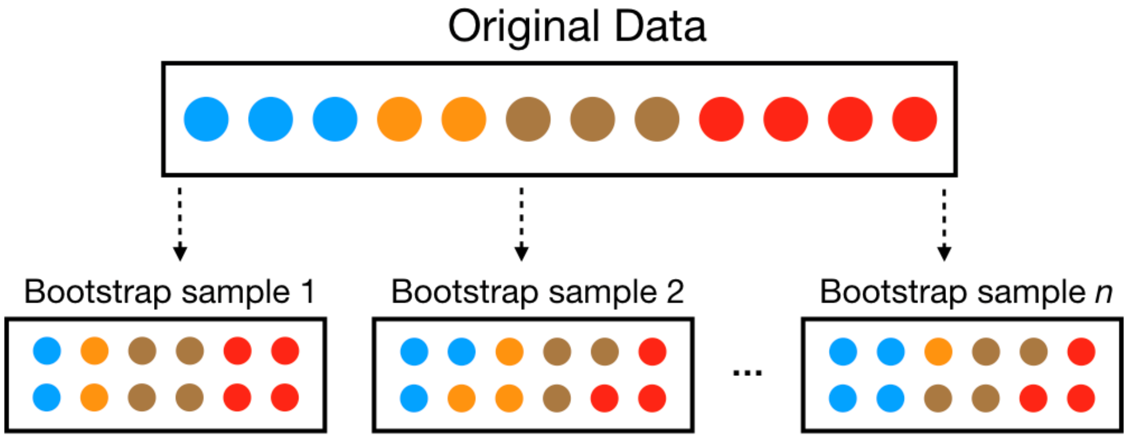
Ensemble learning is the process of using multiple models, trained over the same data, averaging the results to find a better predictive result. Combining weak learners to build a stronger learner usually increase the model performance. Random forest is a bagging ensemble.



https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422

Algorithm: random forest

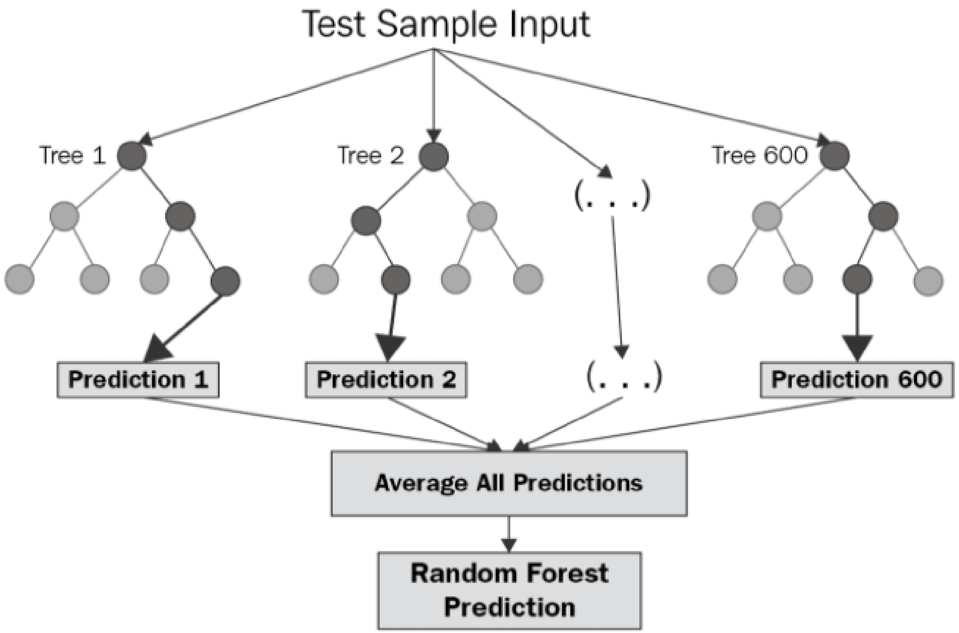
Bootstrapping is the process of randomly sampling subsets of a dataset over a given number of iterations and a given number of variables. Since samples are drawn with replacement, each bootstrap sample is likely to contain duplicate values.



https://bradleyboehmke.github.io/HOML/process.html

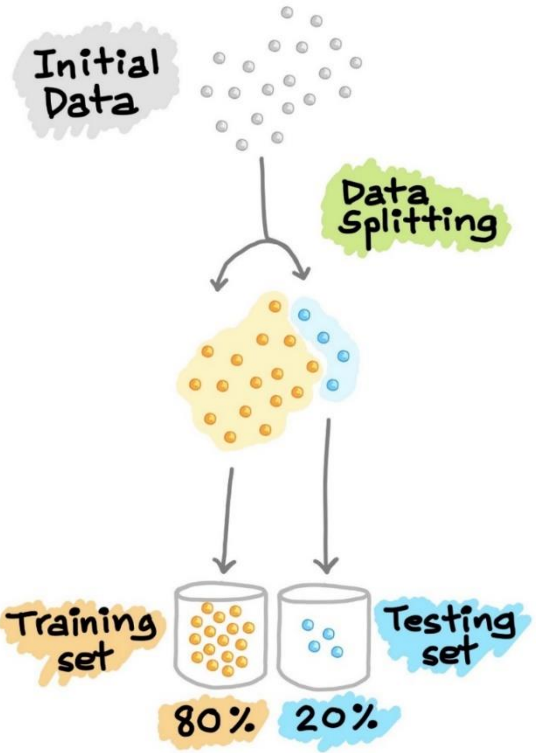
Algorithm: random forest

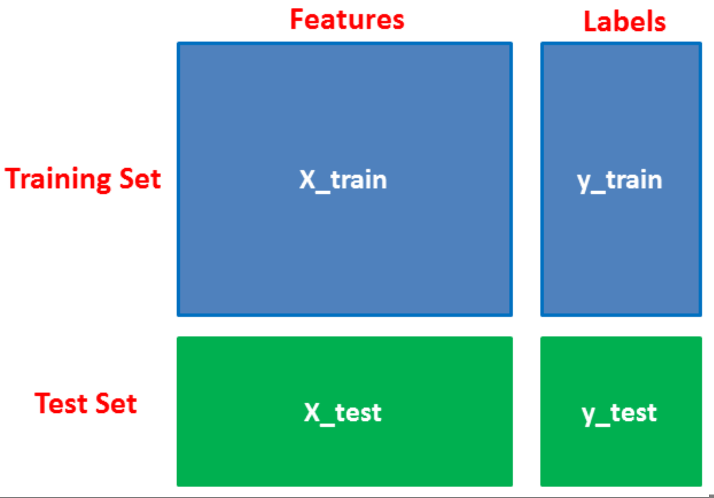
Bootstrapping algorithm that ensemble multiple randomly drawn decision trees from the data, averaging the results to obtain the prediction. In addition, a subset of the features is randomly selected at each node.



https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/

Data splitting





	Features	Labels
Training Set	X_train	y_train
Test Set	X_test	y_test

Trained model must perform well on new, unseen data. In order to simulate the new, unseen data, the available data is subjected to data splitting whereby it is split into 2 portions. 80% of the original data is used as the training set and the remaining 20% is used as the testing set

Tune hyperparameters with GridsearchCV

Hyperparameters are variables that the user specify usually while building the Machine Learning model and are used to evaluate optimal parameters of the model. Example: **max_depth** in Random Forest. But, How can we find the best hyperparameters values to get the best prediction results from our model?

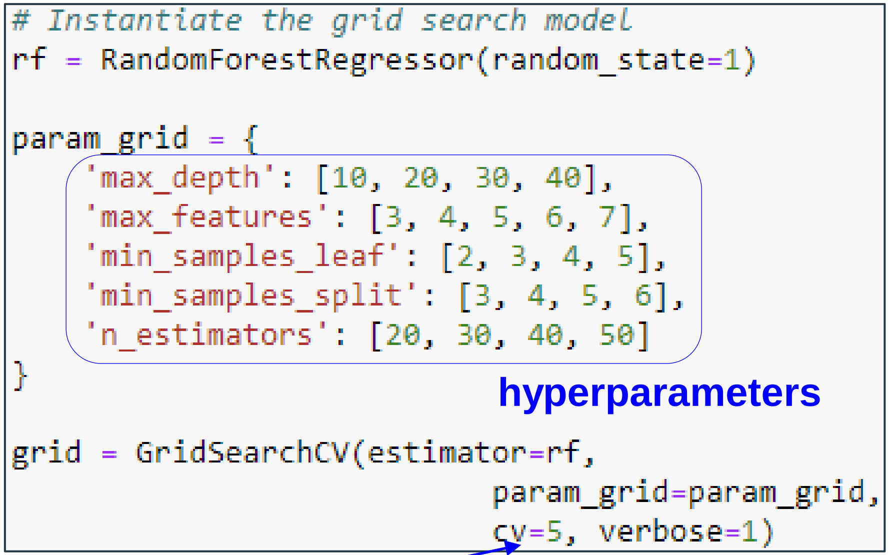
Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters.

https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

Tune hyperparameters with GridsearchCV

In GridSearchCV, along with Grid Search, cross-validation (CV) is also performed. In CV, train data is divided into two parts: train data and validation (test) data.

GridsearchCV

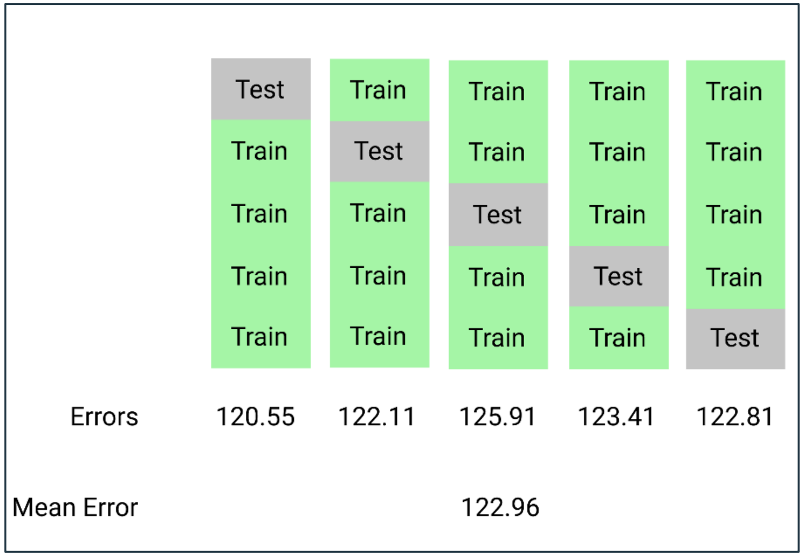


```
# Instantiate the grid search model
rf = RandomForestRegressor(random_state=1)

param_grid = {
    'max_depth': [10, 20, 30, 40],
    'max_features': [3, 4, 5, 6, 7],
    'min_samples_leaf': [2, 3, 4, 5],
    'min_samples_split': [3, 4, 5, 6],
    'n_estimators': [20, 30, 40, 50]
}

grid = GridSearchCV(estimator=rf,
                           param_grid=param_grid,
                           cv=5, verbose=1)
```

Cross-validation (CV)



	Test	Train	Train	Train	Train
	Train	Test	Train	Train	Train
	Train	Train	Test	Train	Train
	Train	Train	Train	Test	Train
	Train	Train	Train	Train	Test
Errors	120.55	122.11	125.91	123.41	122.81
Mean Error			122.96		

number of groups or folds for cross-validation

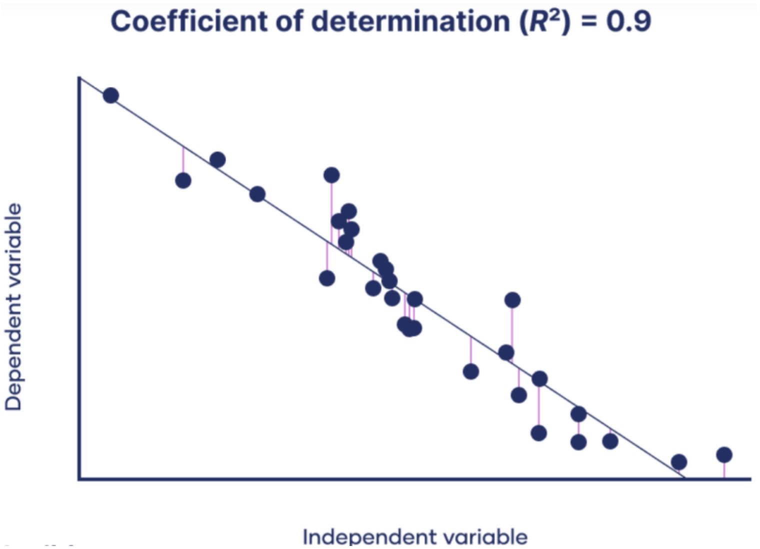
https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

Error Metrics

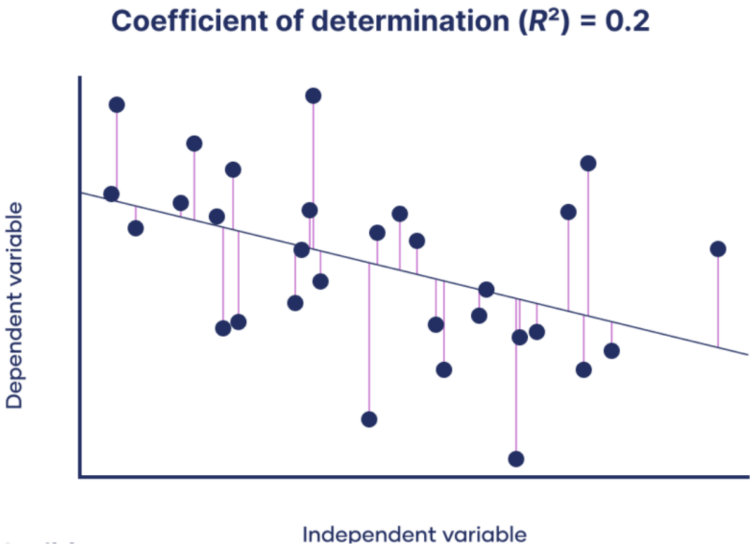
RMSE is the most used metric in regression.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Coefficient of Determination (r^2) determines the proportion of variance in the dependent variable that can be explained by the dependent.



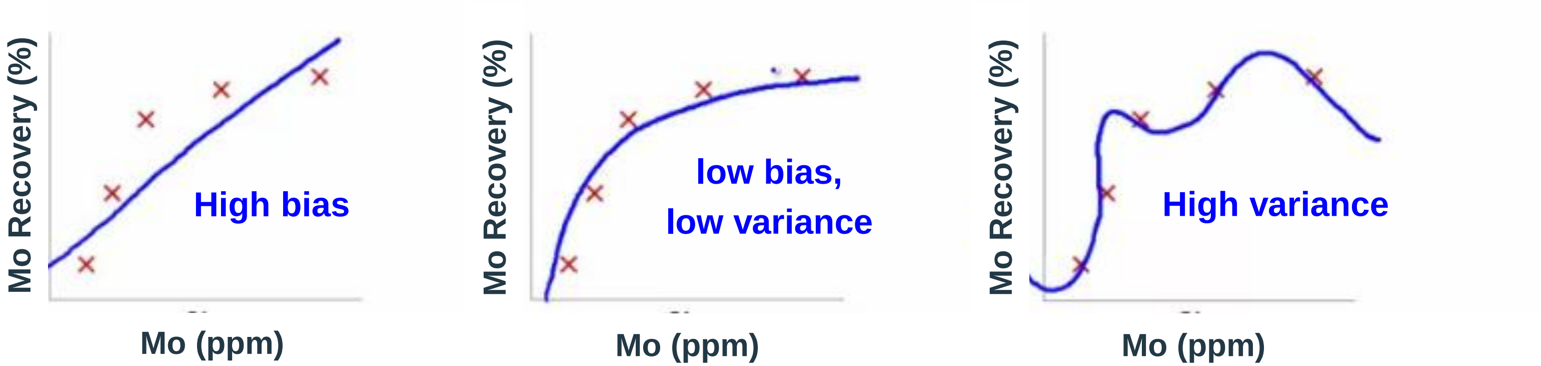
Coefficient of determination (R^2) = 0.9



Coefficient of determination (R^2) = 0.2

https://www.scribbr.com/statistics/coefficient-of-determination/

Bias-variance Tradeoff



underfitting

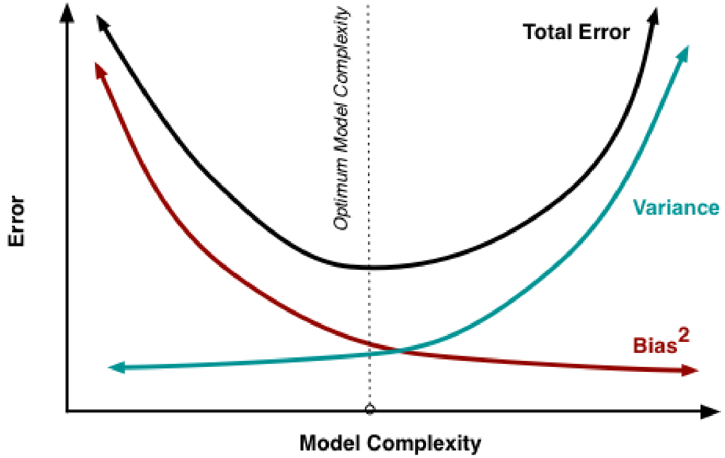
Good balance

overfitting

Model complexity

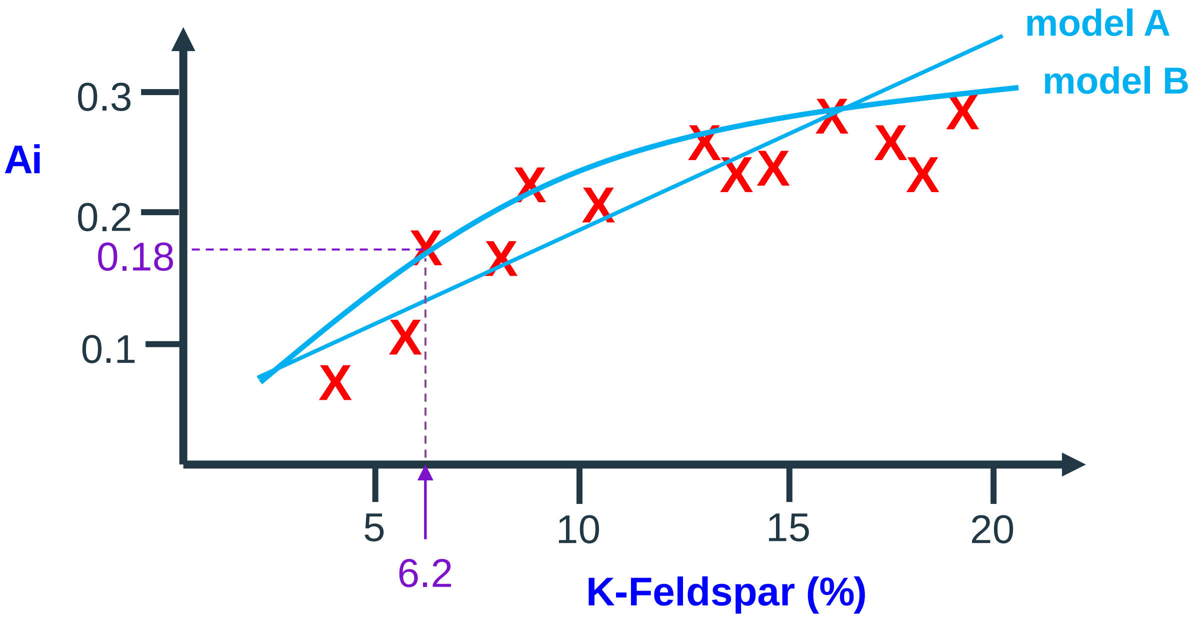
bias: difference between the model prediction and the actual value. Leads to high error on training and test data.

variance: variability of model prediction. Perform well on training data but has high error on test data.



https://www.endtoend.ai/blog/bias-variance-tradeoff-in-reinforcement-learning/

How does a regression algorithm learn?



predictor K-Feldspar (%)	target Ai
10.4	0.21
6.2	0.18
15.9	0.28
…	…

Regression: predict an infinite number of posible outputs. Model:

$$y = wx + b \rightarrow Ai = w * [K\text{-}Feldspar\ (\%)] + b$$

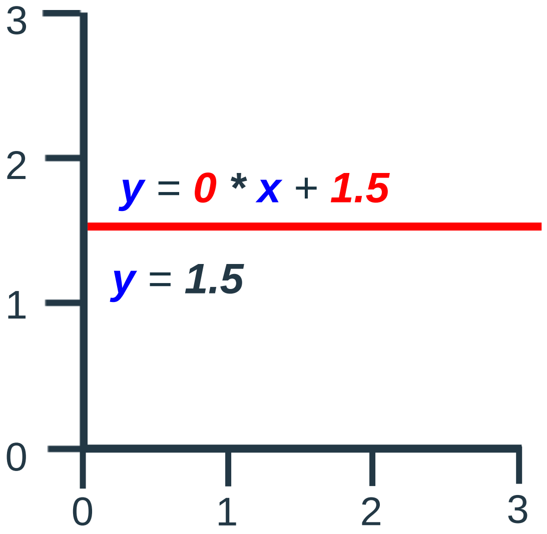
Our goal is to find an algorithm that selects the most appropriate line/curve to fit the data. Which model is better, **model A** o **model B**? How can we choose the best?

Modified from Andrew Ng, “machine learning specialization” (Coursera)

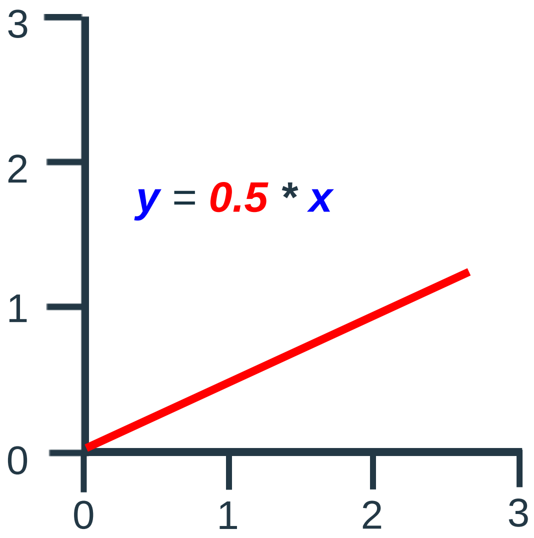
How does a regression algorithm learn?

Model: $y = wx + b$ | w, b = *parameters (coefficients)*

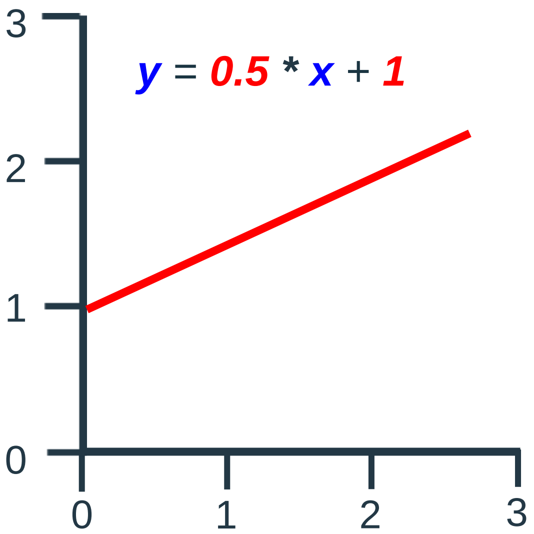
What *w and b do?*



$w = 0$
$b = 1.5$



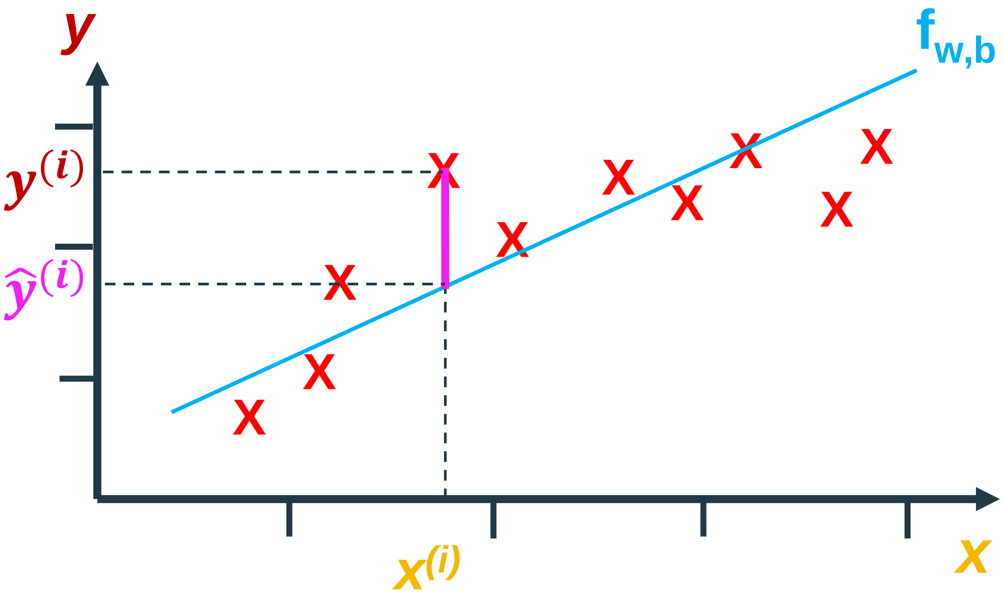
$w = 0.5$
$b = 0$



$w = 0.5$
$b = 1$

Modified from Andrew Ng, “machine learning specialization” (Coursera)

How does a regression algorithm learn?



prediction:

$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$

cost function:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

Find **w, b**:

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

Modified from Andrew Ng, “machine learning specialization” (Coursera)

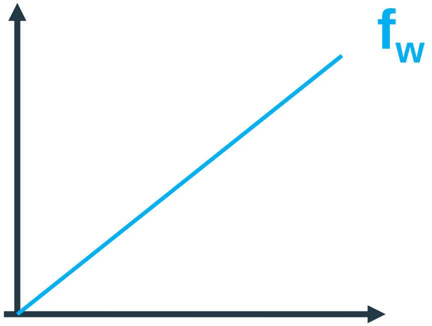
How does a regression algorithm learn?

Simplified Model:

$f_{w,b}(x) = wx + b$

if $b = 0$,

$f_w(x) = wx$

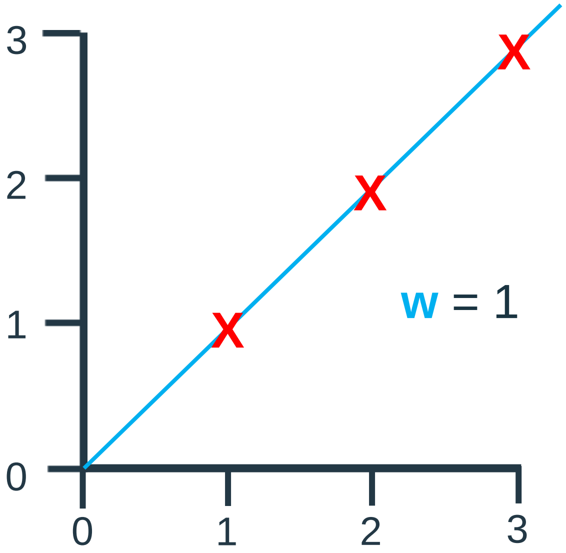


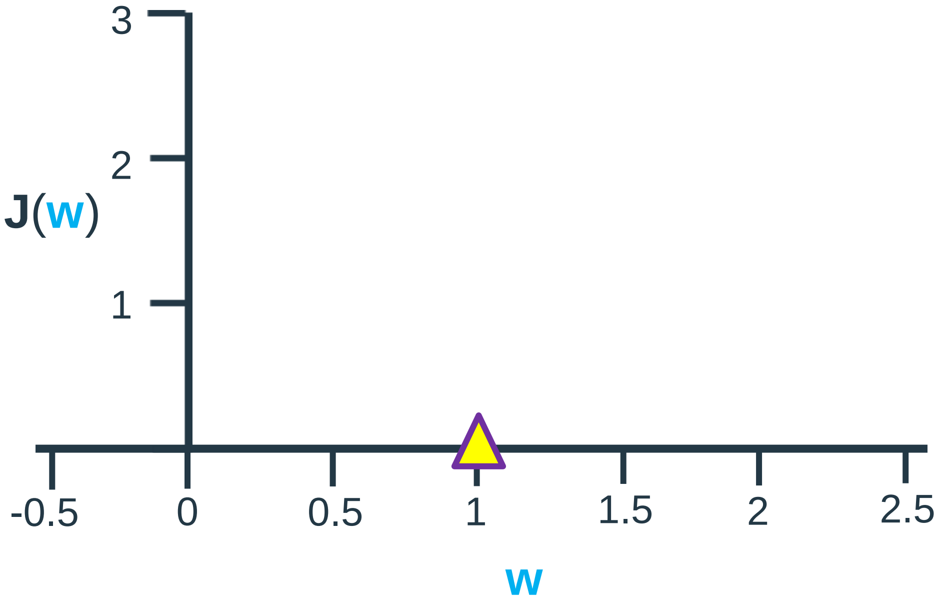
cost function: $$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

goal: minimize J(w)

Modified from Andrew Ng, "machine learning specialization" (Coursera)

How does a regression algorithm learn?

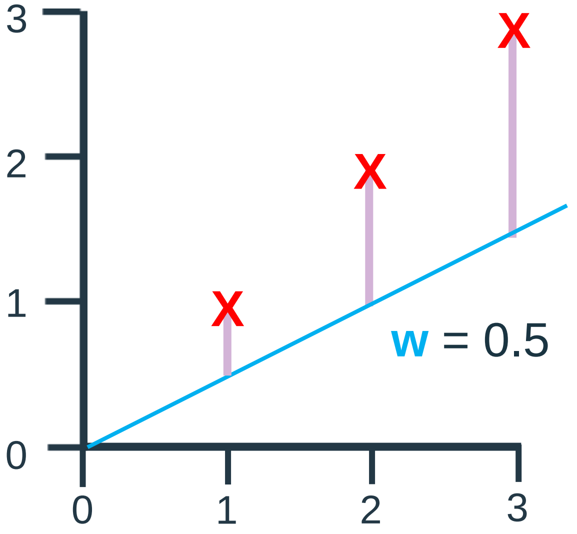


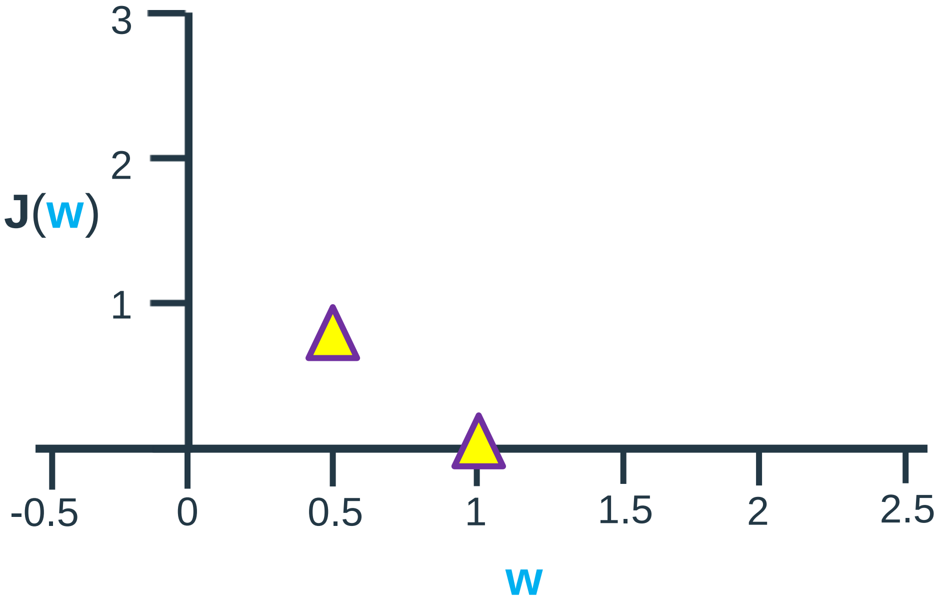


$$J(w = 1) = \frac{1}{2m}\sum_{i=1}^{m}(\,\hat{y}^{(i)} - y^{(i)})^2 = \frac{1}{2m}\,(0^2 + 0^2 + 0^2) = 0$$

Modified from Andrew Ng, “machine learning specialization” (Coursera)

How does a regression algorithm learn?

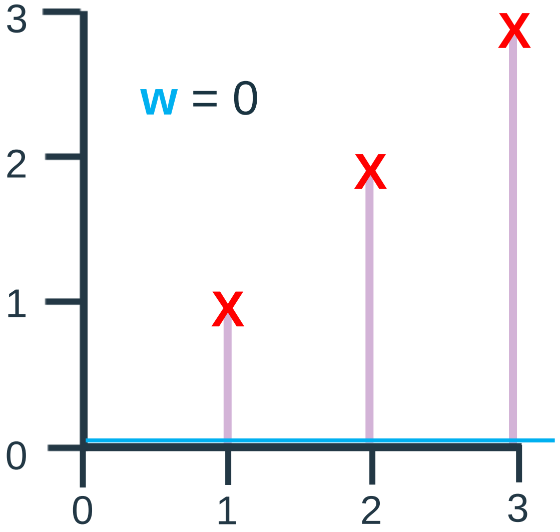


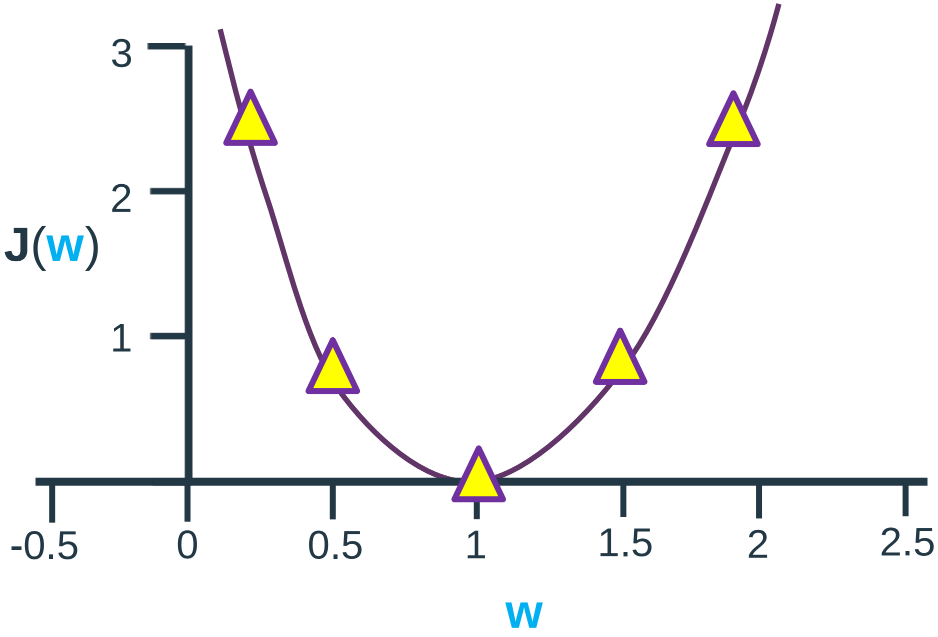


$$J(w = 0.5) = \frac{1}{2m}\sum_{i=1}^{m}(\,\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)})^2 = \frac{1}{2m}\,[(0.5-1)^2 + (2-1)^2 + (1.5-3)^2)] = \frac{1}{2*3}\,[3.5] = 0.58$$

Modified from Andrew Ng, “machine learning specialization” (Coursera)

How does a regression algorithm learn?





$$J(w = 0) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} \text{-} y^{(i)})^2 = \frac{1}{2m}(1^2 + 2^2 + 3^2)] = \frac{1}{2*3}[14] = 2.3$$

goal: minimize J(w)

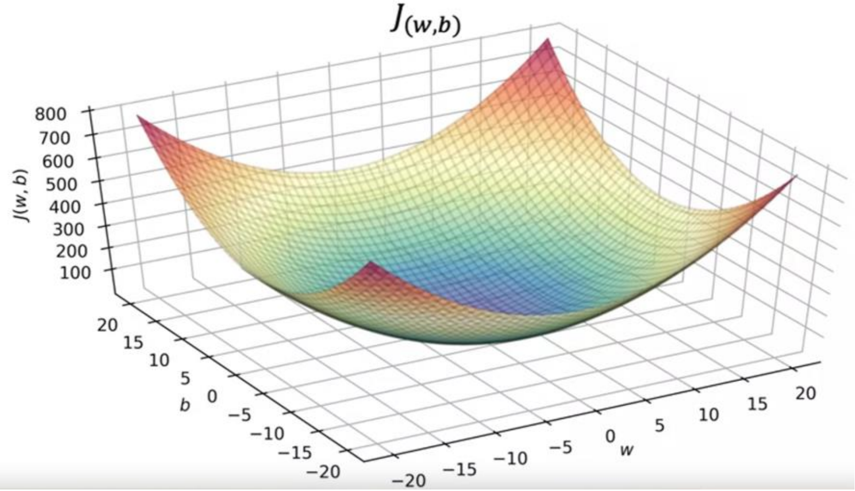
Modified from Andrew Ng, “machine learning specialization” (Coursera)

How does a regression algorithm learn?

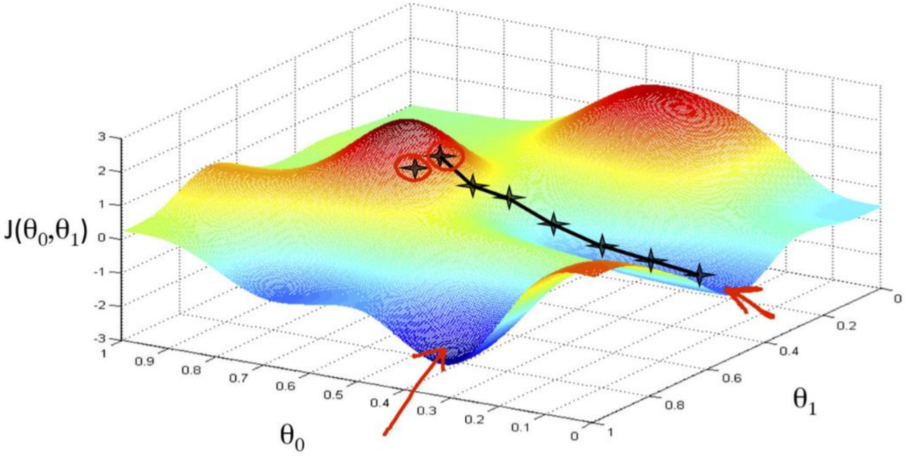
Any point represents a particular choice of w and b. The high in that point is the value of J(w,b).

(GPT-3 parameters)

linear case (one local minimum)



non-linear case (multiple local mínima)



Andrew Ng, “machine learning specialization” (Coursera)

Q
&
A