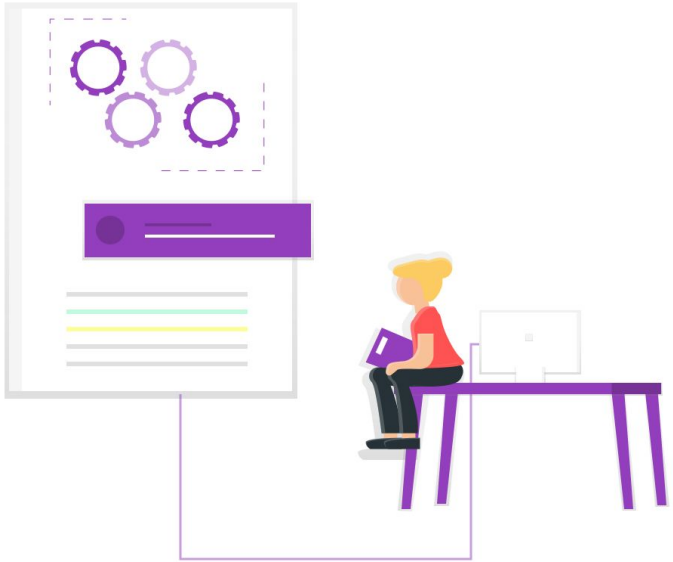
ML Quiz and Algorithms

Session 6-7, 26th Sep 2020

Let's Discuss Quiz

How would you define Machine Learning?

Machine Learning is the science (and art) of programming computers so they can learn from data.



Can you name four types of problems where it shines?

Analyzing images of products on a production line to automatically classify them

Detecting tumors in brain scans

Automatically flagging offensive comments on discussion forums

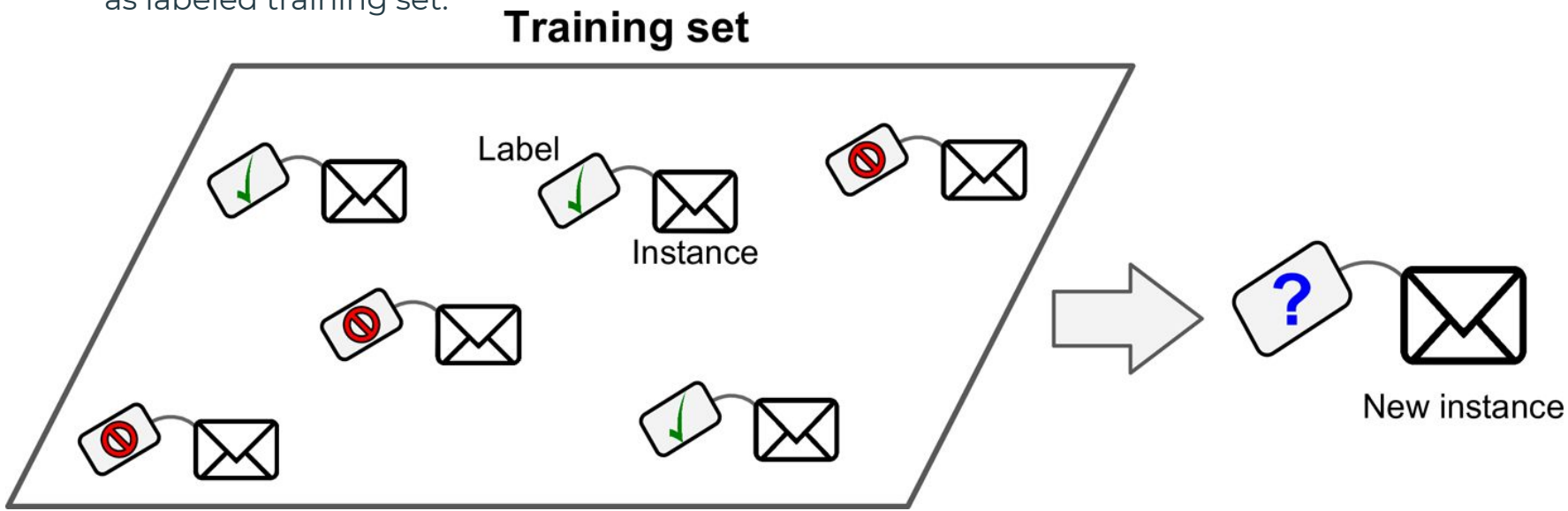
Summarizing long documents automatically

Creating a chatbot or a personal assistant

Detecting credit card fraud

What is a labeled training set?

The training set we feed to the algorithm includes the desired solutions, which is known as labeled training set.



A labeled training set for spam classification

What are the two most common supervised tasks?

Can you name four common unsupervised tasks?

Group Similar Users/Visitors

Visualization algorithms are also good examples of unsupervised learning algorithms: you feed them a lot of complex and unlabeled data, and they output a 2D or 3D representation of your data that can easily be plotted

Anomaly Detection

Association Rule

What type of algorithm would you use to segment your customers into multiple groups?

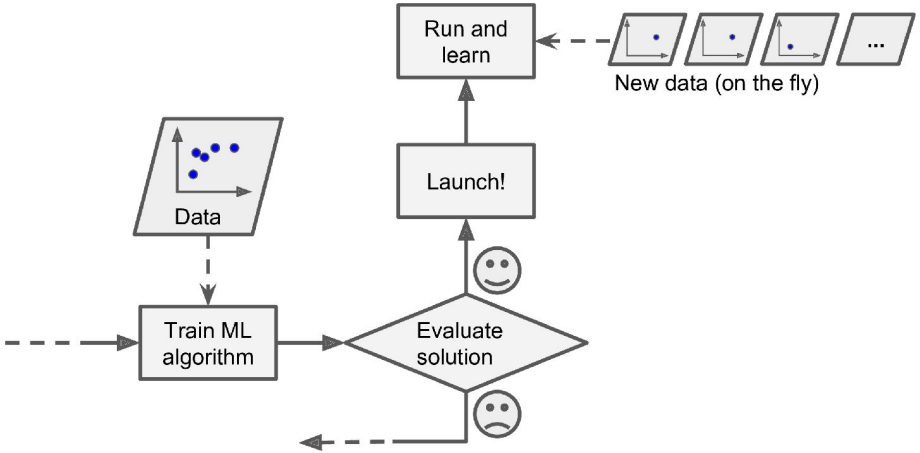
You may want to run a clustering algorithm to try to detect groups of similar visitors. At no point do you tell the algorithm which group a visitor belongs to: it finds those connections without your help. For example, it might notice that 40% of your visitors are males who love comic books and generally read your blog in the evening, while 20% are young sci-fi lovers who visit during the weekends. If you use a hierarchical clustering algorithm, it may also subdivide each group into smaller groups. This may help you target your posts for each group.

What type of Machine Learning algorithm would you use to allow a robot to walk in various unknown terrains?

Many robots implement Reinforcement Learning algorithms to learn how to walk. DeepMind's AlphaGo program is also a good example of Reinforcement Learning: it made the headlines in May 2017 when it beat the world champion Ke Jie at the game of Go. It learned its winning policy by analyzing millions of games, and then playing many games against itself. Note that learning was turned off during the games against the champion; AlphaGo was just applying the policy it had learned.

What is an online learning system?

In online learning, we train the system incrementally by feeding it data instances sequentially, either individually or in small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives



What is out-of-core learning?

The data is so huge that all of it cannot be used at the same time for training the model. Instead, the model is trained in smaller sets of the data. Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory (this is called out-of-core learning). The algorithm loads part of the data, runs a training step on that data, and repeats the process until it has run on all of the data

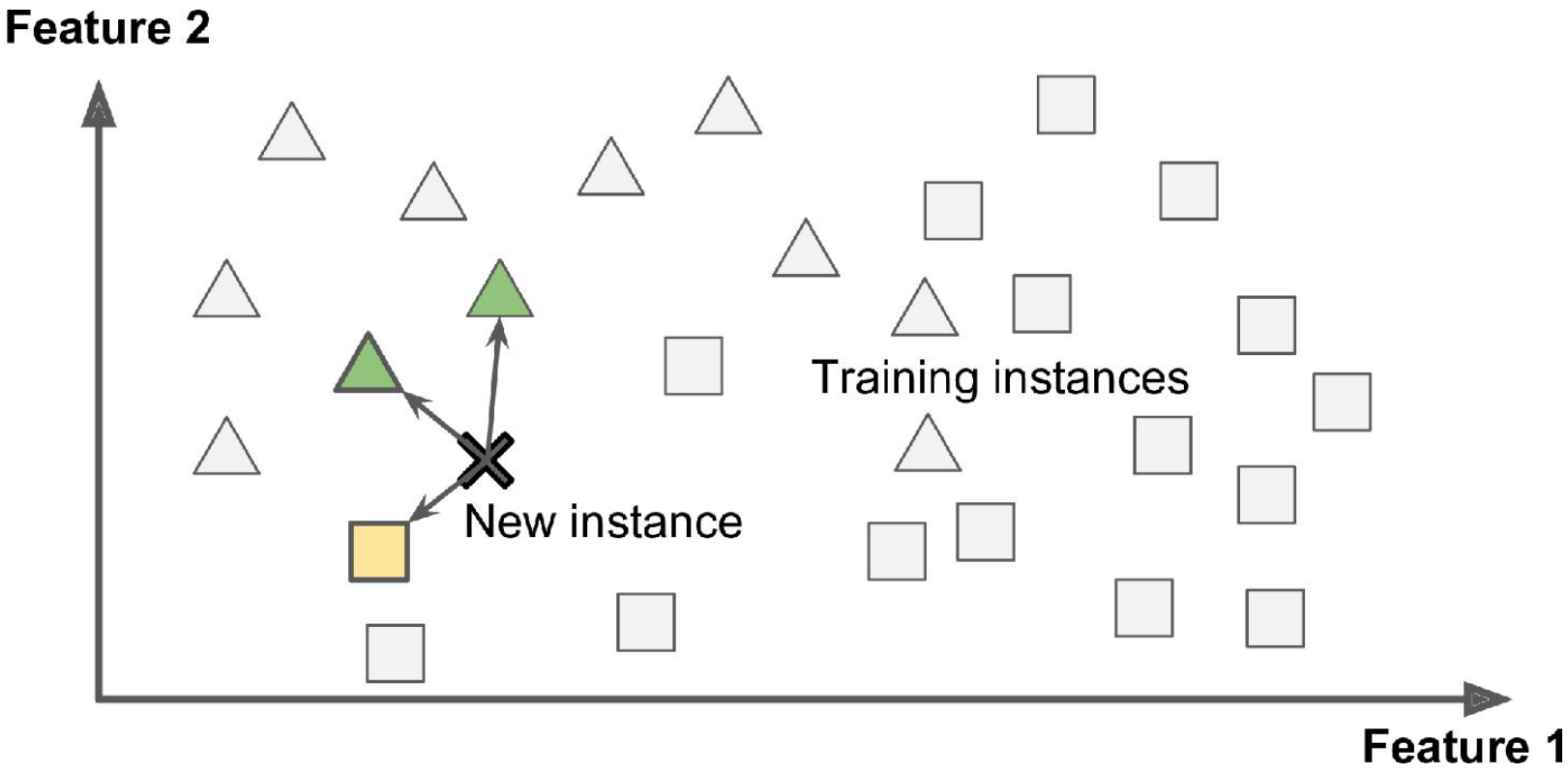
Would you frame the problem of spam detection as a supervised learning problem or an unsupervised learning problem?

The spam filter is a good example of supervised learning: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

What type of learning algorithm relies on a similarity measure to make predictions?

INSTANCE-BASED LEARNING, Instead of just flagging emails that are identical to known spam emails, your spam filter could be programmed to also flag emails that are very similar to known spam emails. This requires a measure of similarity between two emails. A (very basic) similarity measure between two emails could be to count the number of words they have in common. The system would flag an email as spam if it has many words in common with a known spam email.

This is called instance-based learning: the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples (or a subset of them). For example, in the new instance would be classified as a triangle because the majority of the most similar instances belong to that class.


Feature 2
Training instances
New instance
Feature 1

What is the difference between a model parameter and a learning algorithms hyperparameter?

The amount of regularization to apply during learning can be controlled by a hyperparameter. A hyperparameter is a parameter of a learning algorithm (not of the model). As such, it is not affected by the learning algorithm itself; it must be set prior to training and remains constant during training. If you set the regularization hyperparameter to a very large value, you will get an almost flat model (a slope close to zero); the learning algorithm will almost certainly not overfit the training data, but it will be less likely to find a good solution. Tuning hyperparameters is an important part of building a Machine Learning system

What do model-based learning algorithms search for? What is the most common strategy they use to succeed? How do they make predictions?

Model-based algorithms tunes some parameters to fit the model to the training set (i.e., to make good predictions on the training set itself), and then hopefully it will be able to make good predictions on new cases as well. If the algorithm is instance-based, it just learns the examples by heart and generalizes to new instances by using a similarity measure to compare them to the learned instances.

Can you name four of the main challenges in Machine Learning?

Insufficient Quantity of Training Data

Nonrepresentative Training Data

Poor-Quality Data

Irrelevant Features

If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?

Say you are visiting a foreign country and the taxi driver rips you off. You might be tempted to say that all taxi drivers in that country are thieves. Overgeneralizing is something that we humans do all too often, and unfortunately machines can fall into the same trap if we are not careful. In Machine Learning this is called overfitting: it means that the model performs well on the training data, but it does not generalize well.

Solutions

Simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data, or by constraining the model.

Gather more training data.

Reduce the noise in the training data (e.g., fix data errors and remove outliers).

What is a test set, and why would you want to use it?

We split our data into two sets: the training set and the test set. As these names imply, we train our model using the training set, and we test it using the test set. The error rate on new cases is called the generalization error (or out-of-sample error), and by evaluating our model on the test set, we get an estimate of this error. This value tells us how well our model will perform on instances it has never seen before.

What is the purpose of a validation set?

The problem is that you measured the generalization error multiple times on the test set, and you adapted the model and hyperparameters to produce the best model for that particular set. This means that the model is unlikely to perform as well on new data.

A common solution to this problem is called holdout validation: you simply hold out part of the training set to evaluate several candidate models and select the best one. The new held-out set is called the validation set (or sometimes the development set, or dev set). More specifically, you train multiple models with various hyperparameters on the reduced training set (i.e., the full training set minus the validation set), and you select the model that performs best on the validation set. After this holdout validation process, you train the best model on the full training set (including the validation set), and this gives you the final model. Lastly, you evaluate this final model on the test set to get an estimate of the generalization error.

What is the train-dev set, when do you need it, and how do you use it?

We create yet another set train-dev set. After the model is trained (on the training set, not on the train-dev set), we can evaluate it on the train-dev set. If it performs well, then the model is not overfitting the training set. If it performs poorly on the validation set, the problem must be coming from the data mismatch. You can try to tackle this problem by preprocessing the web images to make them look more like the pictures that will be taken by the mobile app, and then retraining the model. Conversely, if the model performs poorly on the train-dev set, then it must have overfit the training set, so you should try to simplify or regularize the model, get more training data, and clean up the training data.

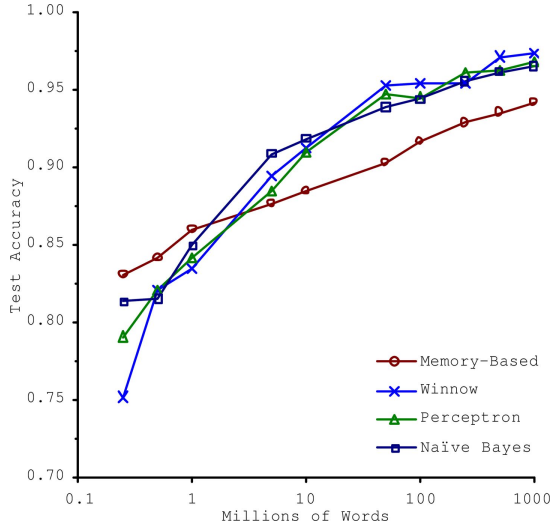
What can go wrong if you tune hyperparameters using the test set?

If we tune hyperparameters using the test set, we risk overfitting the test set, and the generalization error we measure will be optimistic (we may launch a model that performs worse than we expect).

You are given two algorithms. One takes more time than the other but is more accurate. You have absolutely no limitation of labelled data. What algorithm will you go with and why?

We can go with either according to our requirements. However, one must know the effect of data on the accuracy of models.

It has been seen that regardless of what algorithms we choose, they perform comparably good when we have a large amount of data.

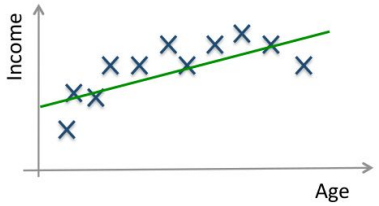


Underfitting and Overfitting

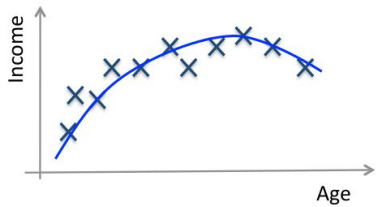
Underfitting and Overfitting

→ We don't want if our model doesn't get generalised. (Undefitting)

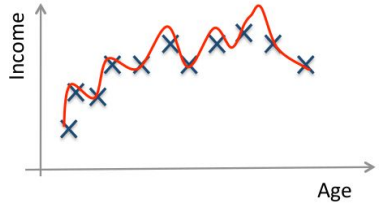
→ Also, we don't want if it gets too generalised only for the available data. (Overfitting)



High bias (underfitting)



Just right!



High variance (overfitting)

How to avoid Overfitting

1. Reduce the number of features manually or do feature selection.
2. Do a model selection.
3. Use regularization (keep the features but reduce their importance by setting small parameter values).
4. Do a cross-validation to estimate the test error.

Confusion Matrix

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions that are correct
Precision	TP / (TP + FP)	The percentage of positive predictions that are correct
Sensitivity (Recall)	TP / (TP + FN)	The percentage of positive cases that were predicted as positive
Specificity	TN / (TN + FP)	The percentage of negative cases that were predicted as negative

Cost/Loss Functions

Cost/Loss Function

1. A function whose value has to be minimized or maximized depending on the context
2. Difference between estimated and true values for an instance of dat

- Let y be the actual expected output
- Let $\hat{y}$ be the output predicted by our model

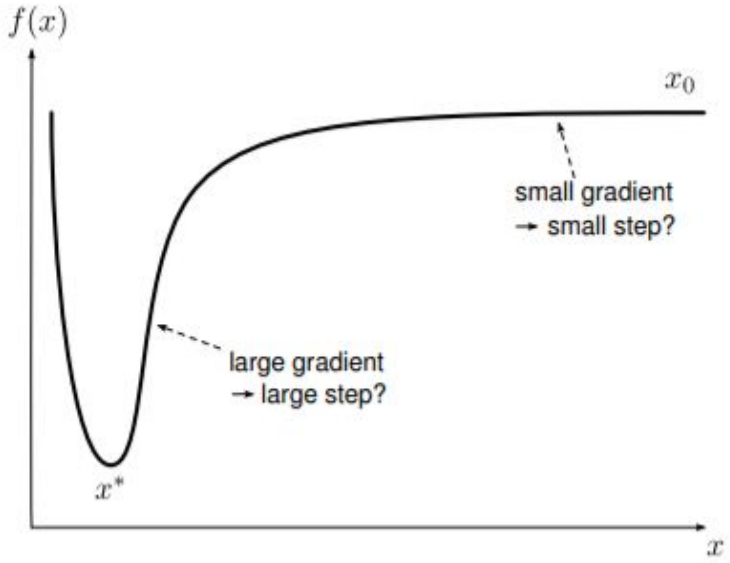
Then our loss function L can be defined as

$$L = (y - \hat{y})^2$$

Gradient Descent

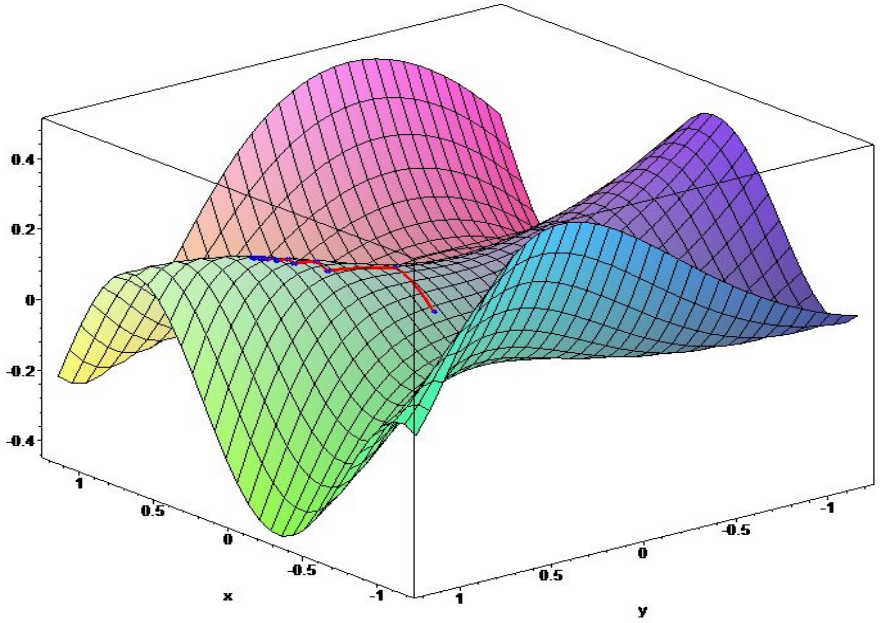
Gradient Descent

- Algorithm for minimizing a cost function
- Gradient simply means “inclined plane”
- Given a function **f(x)** we need to find **min f(x)**

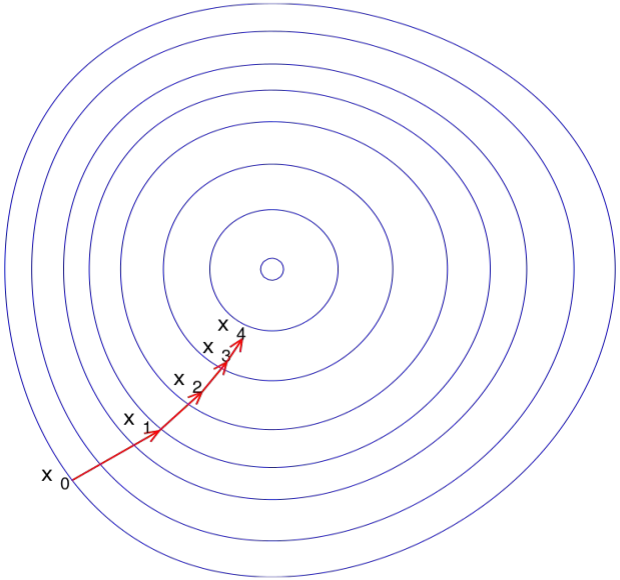


Gradient Descent

A graph of cost function with respect to our learning parameters

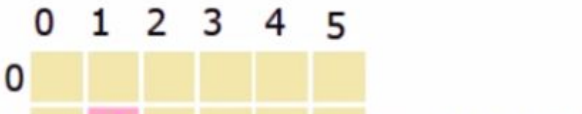


Convergence using Gradient Descent



Calculating Distance and Similarity

Types of distance between two points



Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

Cosine Similarity

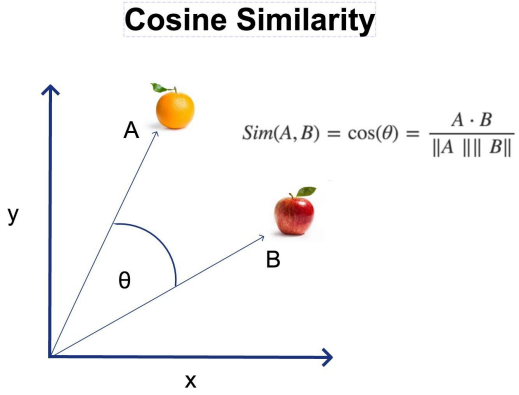
Let A and B be two vectors.
Their dot product is given by,

$A \cdot B = |A| \, |B| \cos\theta$

Therefore,
$\cos\theta = (A \cdot B) / (|A| \, |B|)$

Now we know that,
for θ = 0, cosθ value is 1 (maximum)
for θ = 90, cosθ value is 0 (minimum)

Hence, we can conclude that, when the vectors are similar, the angle between them is near 0 and cosine similarity is near 1.



Supervised Learning Algorithms

Supervised Learning Algorithms

1. Task of learning a function that maps an input to an output based on example input-output pairs

2. Labelled training data

3. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances

4. A wide range of supervised learning algorithms are available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problem

Popular Supervised Learning Algorithms

K-nearest neighbour algorithm

Linear Regression

Logistic Regression

Support Vector Machines

Neural networks

Decision Trees

Linear Regression

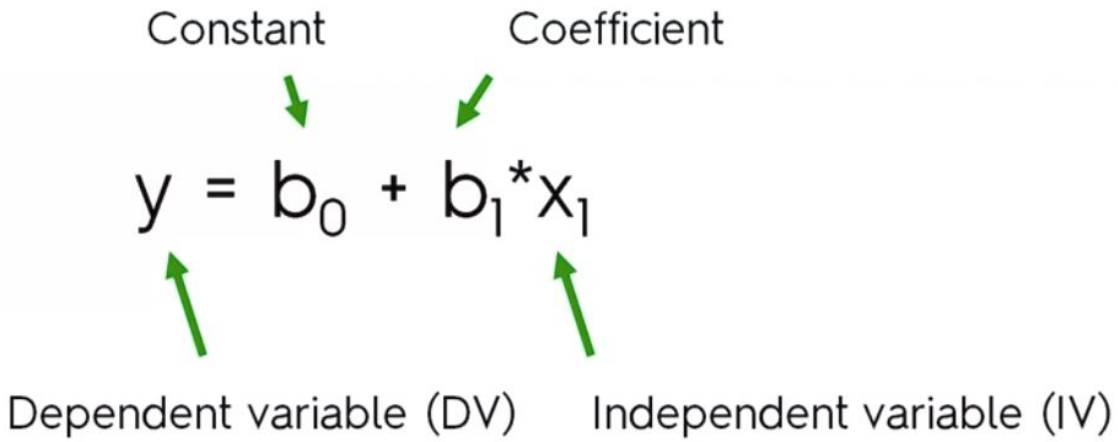
Simple Linear Regression

1. Technique used for the modeling and analysis of numerical data

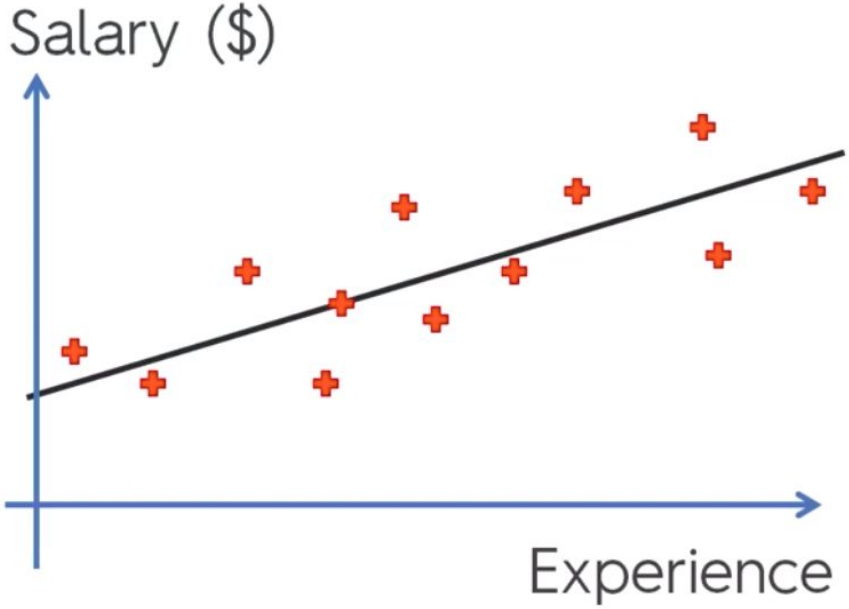
2. Exploits the relationship between two or more variables so that we can gain information about one of them through knowing values of the other

3. Regression can be used for prediction, estimation, hypothesis testing, and modeling causal relationships

Linear Regression Equation



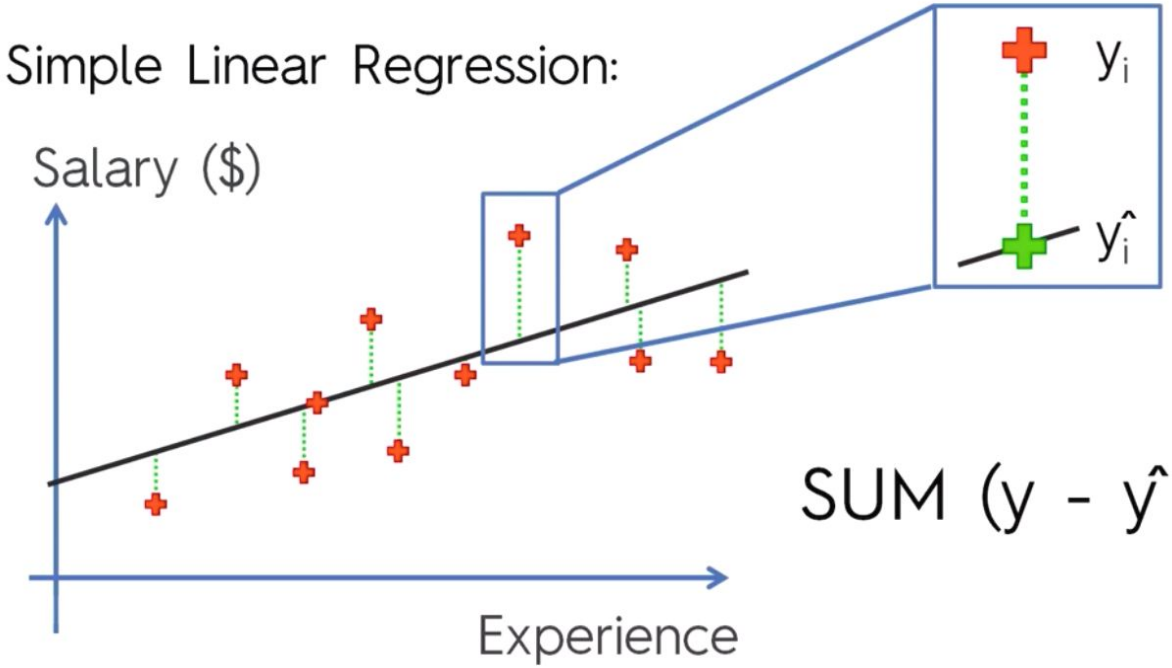
Linear Regression Example



$$y = b_0 + b_1 * x$$

$$\text{Salary} = b_0 + b_1 * \text{Experience}$$

Fitting the model (Least Mean Square)



$$SUM\ (y - \hat{y})^2 \rightarrow min$$

Logistic Regression

Logistic Regression

1. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

Week Work

Approach

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.

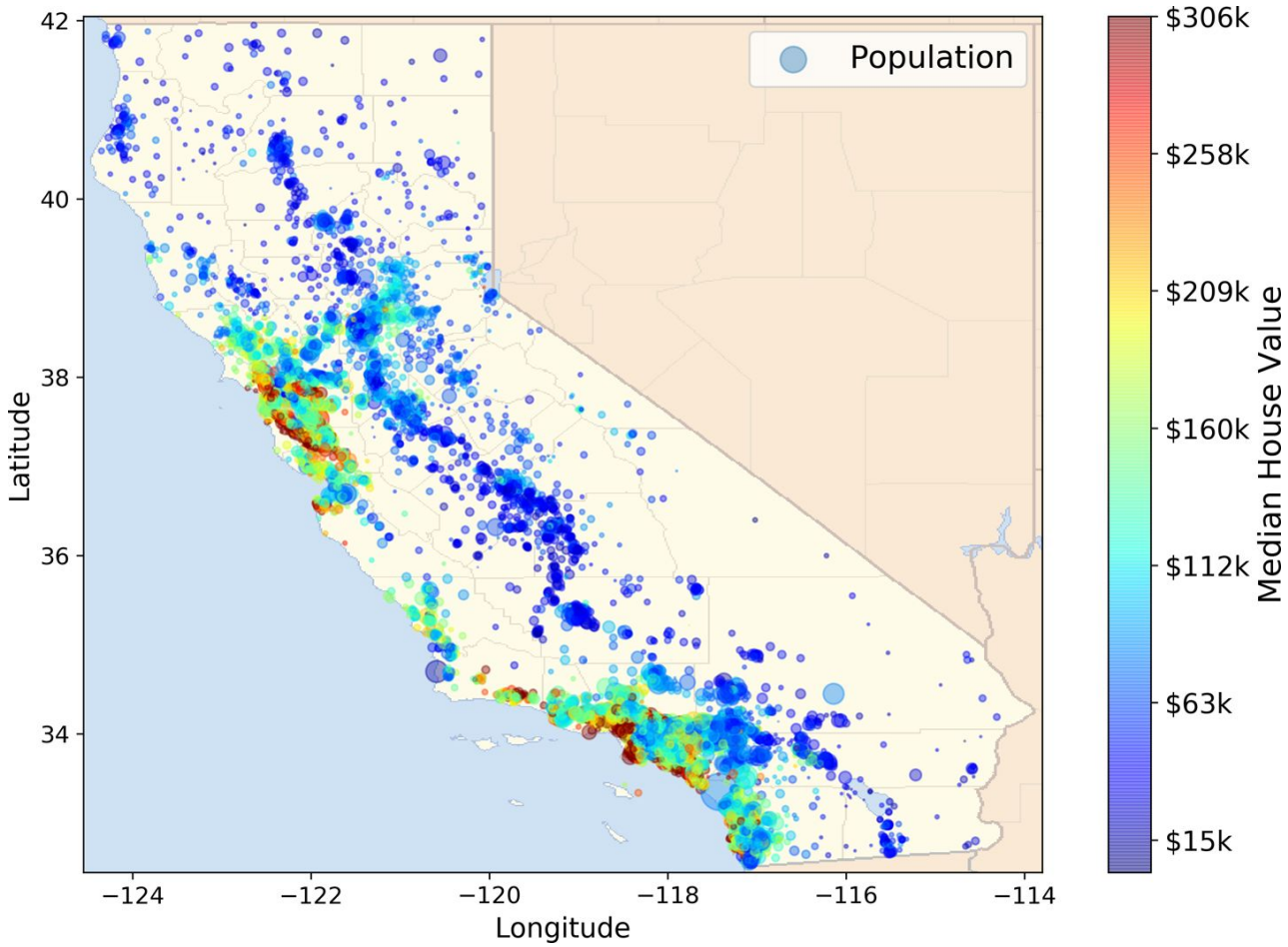
Launch, monitor, and maintain your system.

Task: Beginner

Welcome to the Machine Learning Housing Corporation! Your first task is to use California census data to build a model of housing prices in the state. This data includes metrics such as the population, median income, and median housing price for each block group in California. Block groups are the smallest geographical unit for which the US Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people). We will call them “districts” for short.

Your model should learn from this data and be able to predict the median housing price in any district, given all the other metrics.

Dataset: https://github.com/ageron/handson-ml2/tree/master/datasets/housing


Population
Latitude
Longitude
42
40
38
36
34
−124
−122
−120
−118
−116
−114
Median House Value
$306k
$258k
$209k
$160k
$112k
$63k
$15k

CV AI CHAMP

Task: Intermediate/Advanced

https://www.kaggle.com/hojjatk/mnist-dataset

CV AI CHAMP