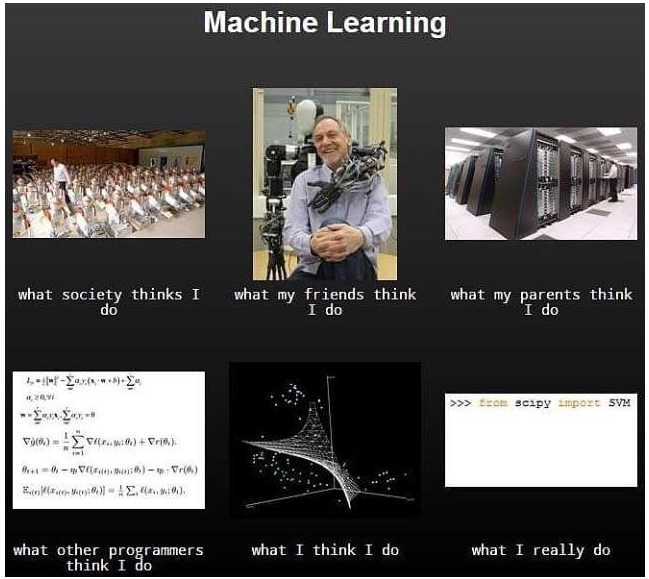
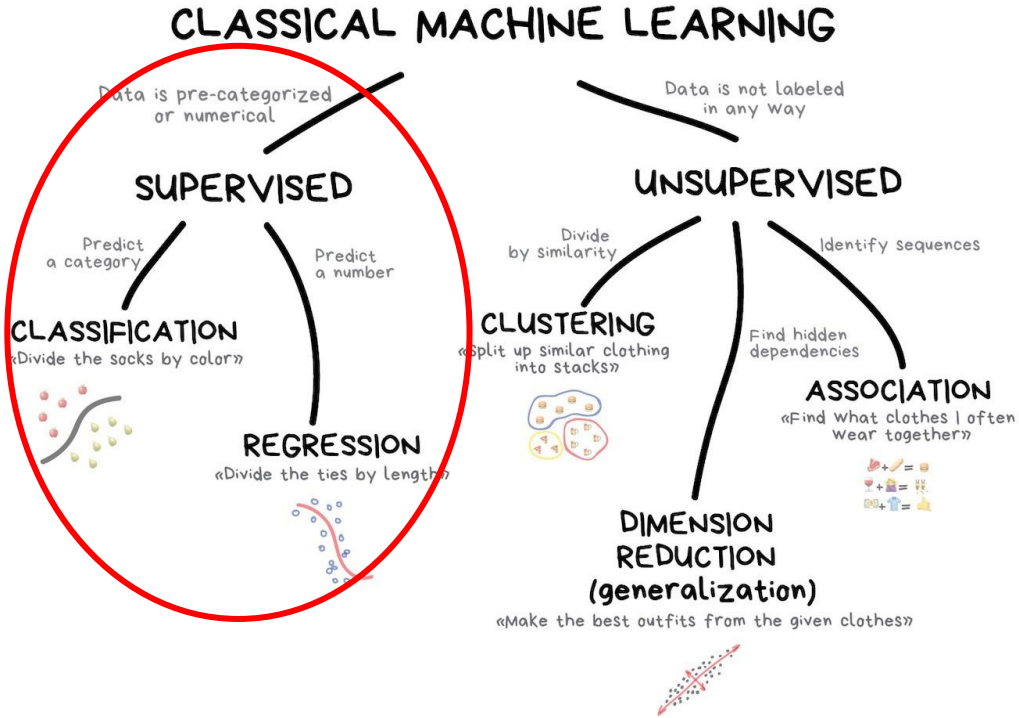
Evolutionary Algorithms

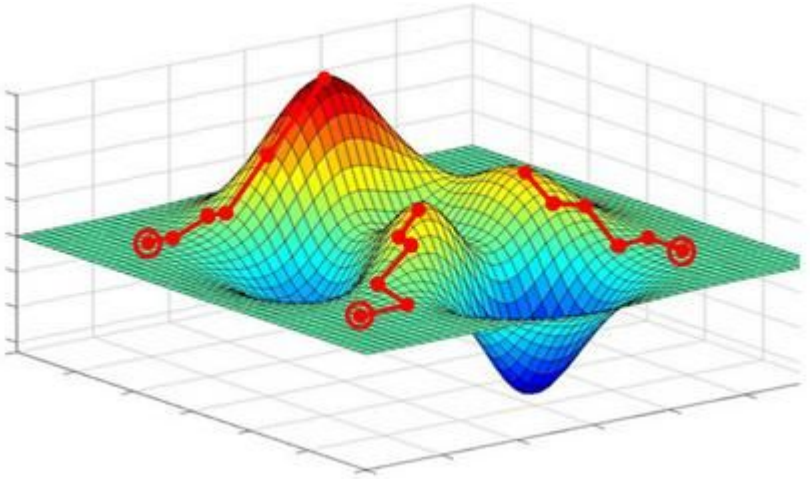
CS @ ILLINOIS Sail
April 2019

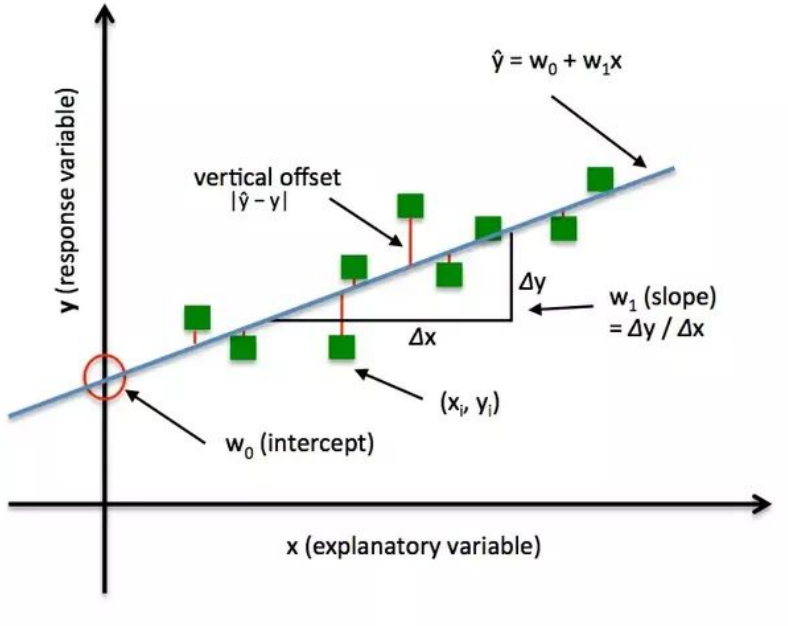
Machine Learning



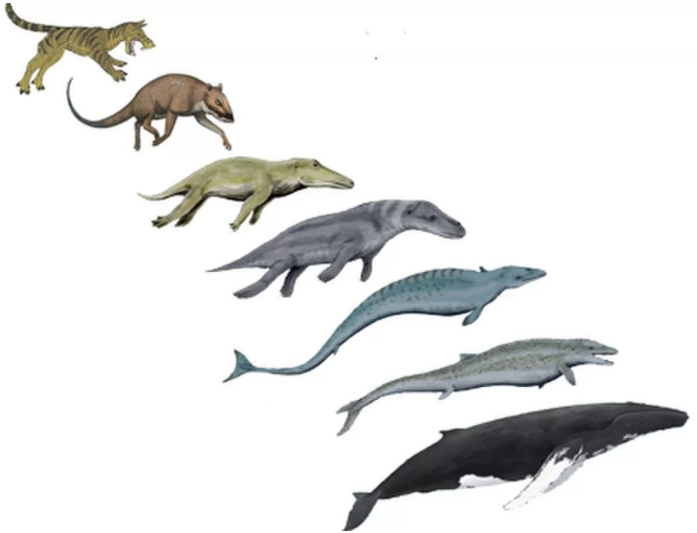


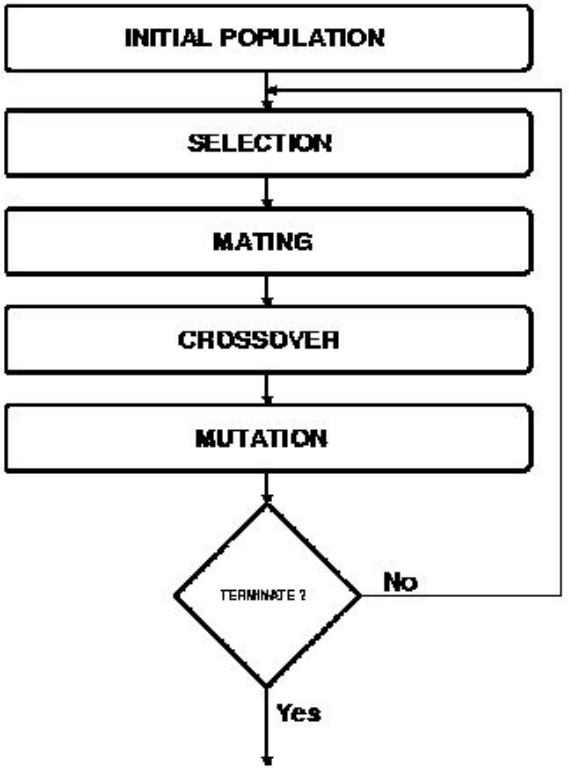
Optimization



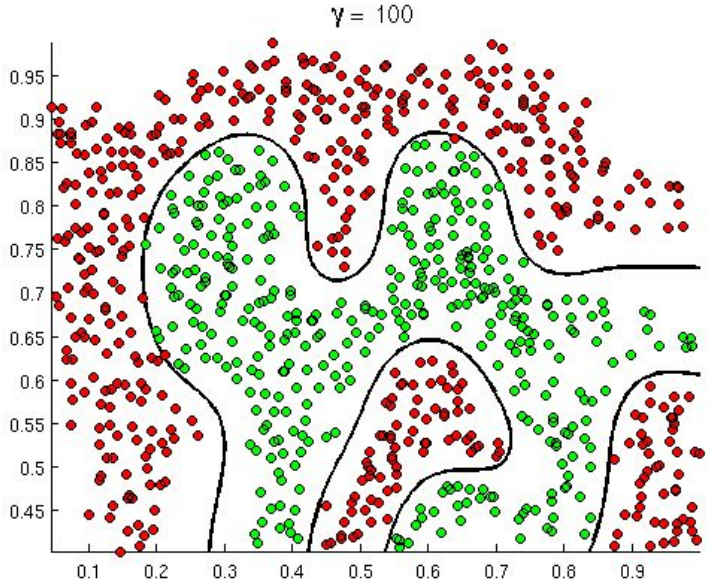


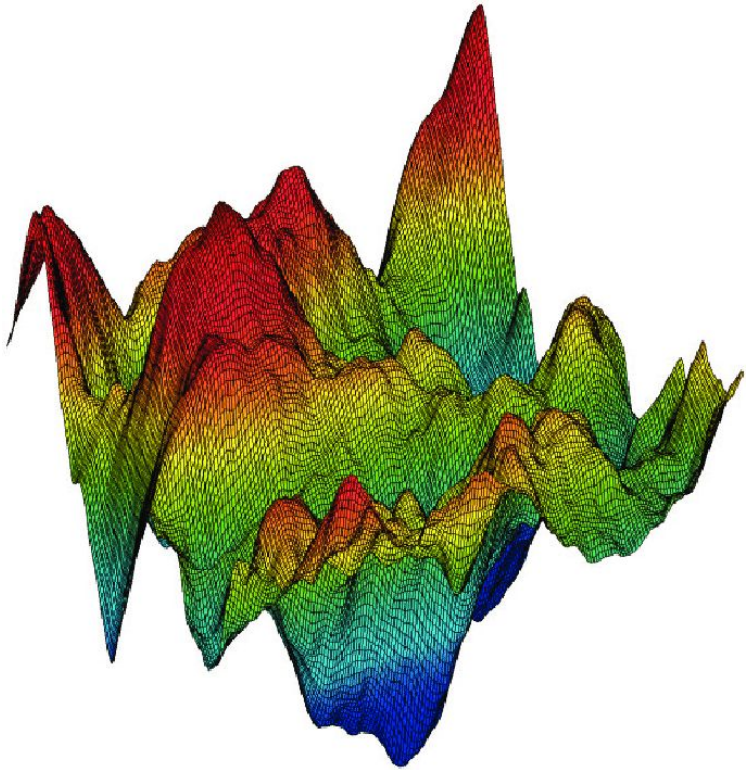
Genetic Algorithms?





Why Use Them?


γ = 100
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9



Implementation

Step One: Generate the initial population of individuals randomly

Step Two: Evaluate the fitness of each individual in that population

Step Three: Repeat the following regenerational steps until termination:

1. Select the best-fit individuals for reproduction
2. Breed new individuals through crossover and mutation operations to give birth to offspring.
3. Evaluate the individual fitness of new individuals.
4. Replace least-fit population with new individuals.

Setup

```
model = {feature1, feature2, feature3, ..., featureN}

individual_DNA = {A, B, C, ..., N} // we call A, B, C, ... points
```

Generate Initial Population

```
for (int i = 0; i < INITIAL_POPULATION_SIZE; i++)
    Create_Individual(random_DNA) // add to population
```

Evaluate Fitness

```
predicted_out = A * feature1 + B * feature2 + ... + N * featureN

fitness = LossFunction(predicted_output, actual_output)
```

Select Best-Fit Individuals

```
for (individual: population)
    if (individual_fitness < MIN_FITNESS)
        remove individual from population
```

Crossover

```
for (point: DNA)
    if (random > CROSSOVER_RATE)
        Swap(this_feature, random_individual_feature)
```

Mutation

```
for (point: DNA)
    if (random > MUTATION_RATE)
        point = random; // different random number
```

Web API's

A server-side web API is a programmatic interface consisting of one or more publicly exposed endpoints to a defined request–response message system, typically expressed in JSON or XML, which is exposed via the web—most commonly by means of an HTTP-based web server.

Get Real-Time Apple Stock Quote

https://cloud.iexapis.com/beta/stock/market/batch?token=pk_18e2ba4b80cb471aa96ba58fadec6b05&symbols=aapl&types=quote

Requests Library

```
API = "https://cloud.iexapis.com/beta/stock/market/batch"
TOKEN = "pk_18e2ba4b80cb471aa96ba58fadec6b05"
SYMBOLS = “aapl”
TYPES = “quote”


query = {"token": TOKEN, "symbols": SYMBOLS, "types": TYPES}
r = requests.get("https://cloud.iexapis.com/beta/stock/market/batch", params=query)
```

Stock Information

- open
- high
- low
- close
- volume

Historical Stock Data is Scarce...

Quick discussion on the open source movement

Welcome back

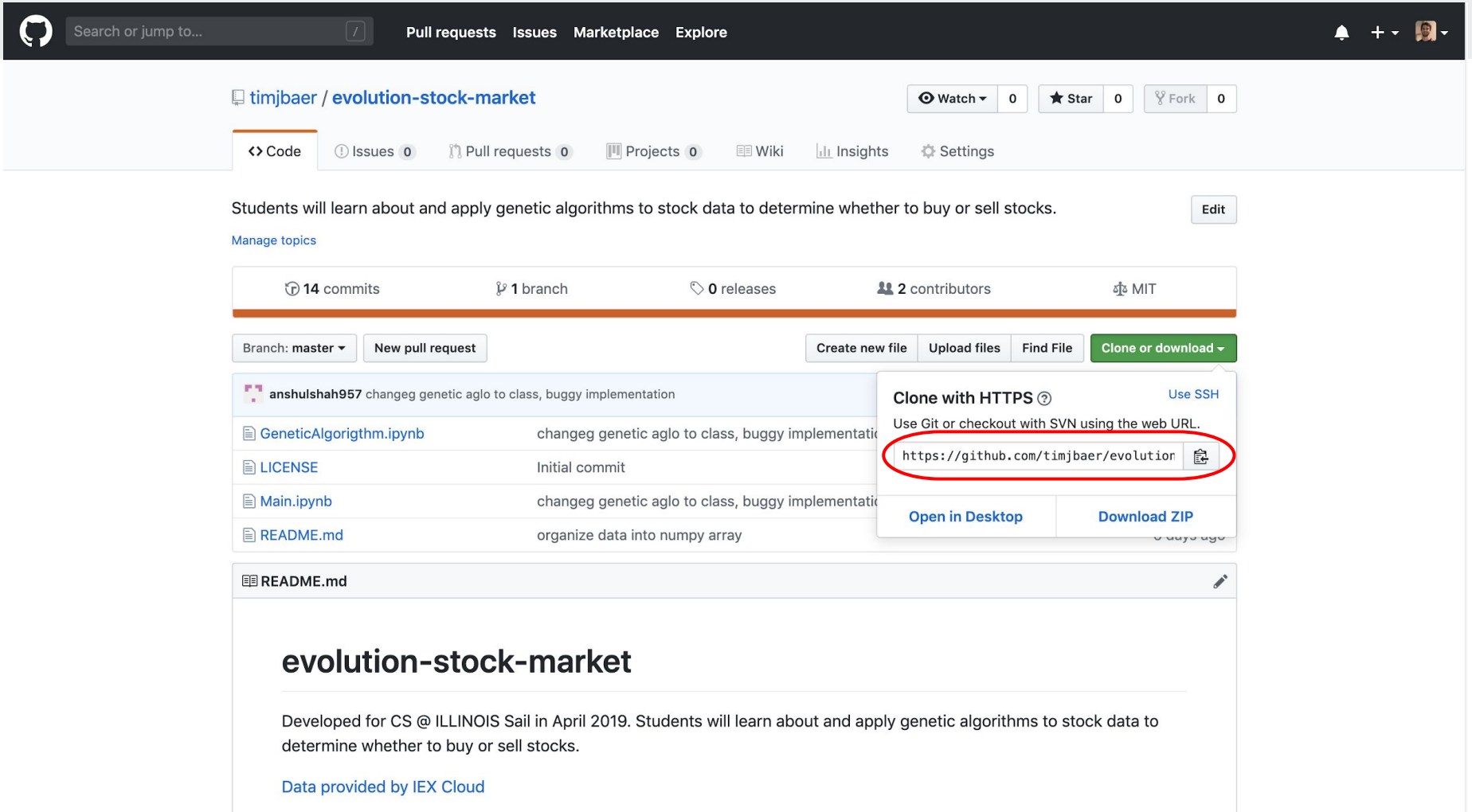
Email

Password

Log in

Forgot your password?

Don't have an account? Create an account →


Search or jump to...
Pull requests
Issues
Marketplace
Explore
timjbaer / evolution-stock-market
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Students will learn about and apply genetic algorithms to stock data to determine whether to buy or sell stocks.
Edit
Manage topics
14 commits
1 branch
0 releases
2 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
anshulshah957 changeg genetic aglo to class, buggy implementation
GeneticAlgorigthm.ipynb
changeg genetic aglo to class, buggy implementati
LICENSE
Initial commit
Main.ipynb
changeg genetic aglo to class, buggy implementati
README.md
organize data into numpy array
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/timjbaer/evolution
Open in Desktop
Download ZIP
README.md
evolution-stock-market
Developed for CS @ ILLINOIS Sail in April 2019. Students will learn about and apply genetic algorithms to stock data to determine whether to buy or sell stocks.
Data provided by IEX Cloud

Getting Started

Click here to see the steps in a notebook format

Open terminal and type:

```
git clone https://github.com/timjbaer/evolution-stock-market.git

cd evolution-stock-market

python Run.py ← Run our code!

vim Run.py ← Edit to make it run better!
```

Questions?

- Ask about genetic algorithms, programming, our experiences at UIUC, the Cubs, etc.