Agent-Based Modelling

# Assets in ABM Lead to Emergent Specialization

**Group 16**

Nitai Nijholt Calvin Law
Dan Dong Jasper Timmer Zijian Zhang

# Motivation

- **Preventing Income Inequality:** Can wealth inequality be prevented in the presence of initial wealth heterogeneity?
- **Effects of Asset Introduction:** What is the effect of introducing assets on the emergent behavior and income of rational agents?
- **Reproducing Emergent Behavior:** Zheng et al.'s (2022) paper uses an unrealistic asset model and directly programs in "emergent" specialization behavior. Can we reproduce this behavior without direct programming it into the model?

# Motivation

## What we add compared to Zhang et. al 2022

- Remove skills heterogeneity, introduce only initial wealth heterogeneity
- Introduce assets (houses, which give agents income over time)
- Introduce cities (houses close together, produce more income)
- Static vs. dynamic tax policy
- 2 agent types: expectation maximizing and neurevolution

# Problem Definition


RQ1: How does introducing an asset, influence the emergent behavior classes?

RQ2: What is the effect of static, vs. dynamic tax policy on wealth distribution?

# Why ABM?

Heterogeneity

Agent States
(wealth, inventory)

Asset Income
(increases if other assets (houses) are nearby)

Spatial/Local Interactions
(Gathering, Building, Cities)

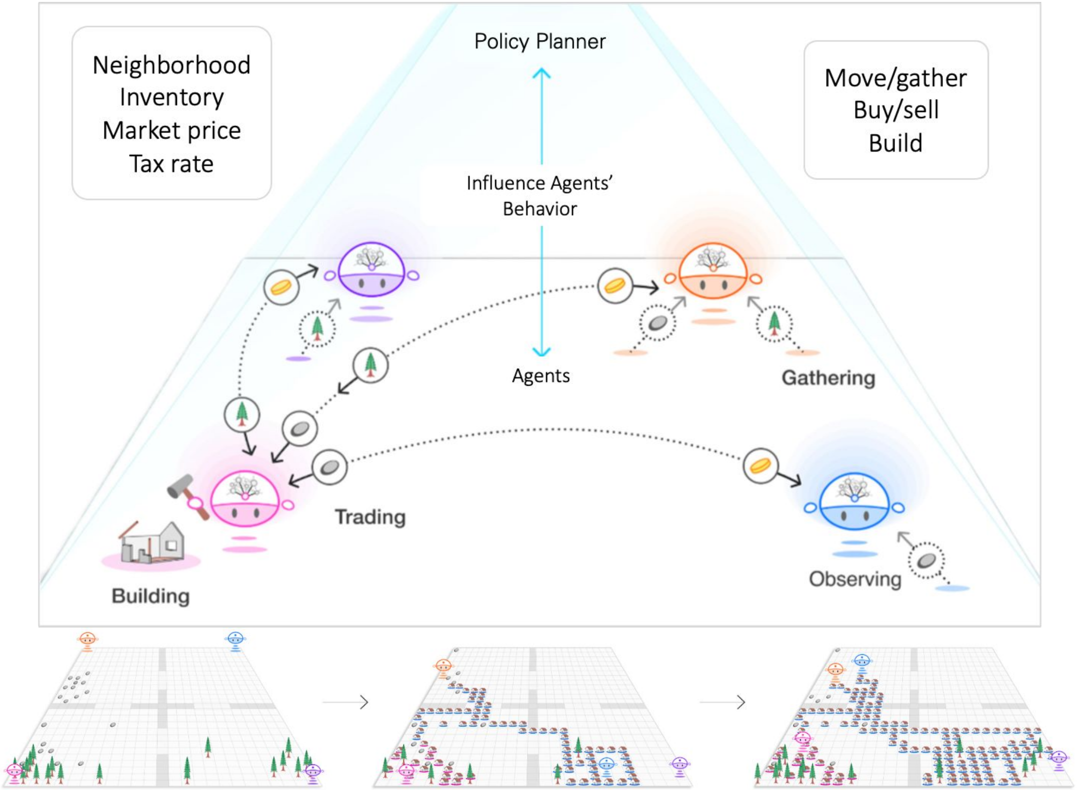
Emergence
(Classification of Agent Behavior)

# Model Description

- Overview
- Tax Policy
- Market Order Book
- Decision Making EV
- Decision Making Neuroevolution

# Model Overview

- **Objects:** Agent, Tax policy Planner, Market, Assets (houses)
- **Environment:** 2d grid with stochastic spawning resources
- **Agent Actions:** move, gather, build houses, buy or sell resources
- **Agent Decision Making:** choose an action based on the logit model
- **Market:** dynamic, agents place bid or ask in order book pricing resources at expected value
- **Assets:** Houses grant the agent +1 income per timestep, or more if adjacent to other houses
- **Tax policy:** Collect and redistribute income tax every timestep

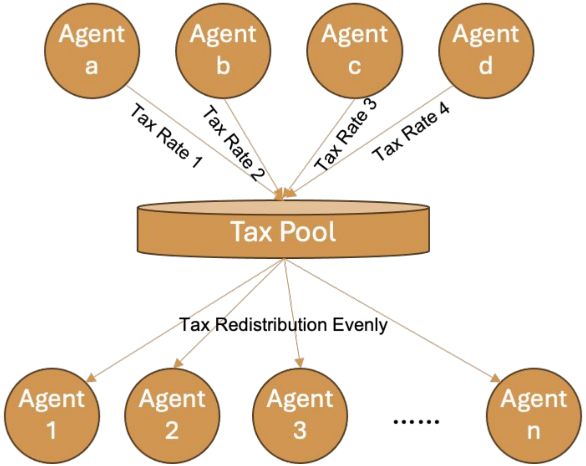



Source: Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. Science advances, 8(18), eabk2607.

# Tax Policy

Tax Collection and Redistribution : Dynamic and Static Strategy

- Define Income Brackets and Tax Rates: Based on the Dutch Income Data, Over-progressive Tax System
- Set Standard Deviation of Income (Only for Dynamic Policy)
- Set Adjustment Factor (Only for Dynamic Policy)
- Redistribute Collected Tax Evenly
- Update Wealth




- Progressive Tax System

Tax — Income — Number of Brackets — Marginal Tax Rates — Income limits

$$T(z) = \sum_{j=1}^{B} \tau_j \cdot [(b_{j+1} - b_j) \cdot \mathbf{1}\{z > b_{j+1}\} + (z - b_j) \cdot \mathbf{1}\{b_j < z \leq b_{j+1}\}]$$

- Tax Redistribution

$$\Delta C_i = -T(z_i) + \frac{1}{N} \sum_{j=1}^{N} T(z_j)$$

Wealth Change

# Tax Policy

- Social Equality

$$\mathrm{eq}(\boldsymbol{C}_t) = 1 - \frac{N}{N-1}\mathrm{gini}(\boldsymbol{C}_t),\ 0 \le \mathrm{eq}(\boldsymbol{C}_t) \le 1$$

$$\mathrm{gini}(\boldsymbol{C}_t) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|C_{i,t}-C_{j,t}|}{2N\sum_{i=1}^{N}C_{i,t}},\ 0 \le \mathrm{gini}(\boldsymbol{C}_t) \le \frac{N-1}{N}$$

- Social Productivity

$$\mathrm{prod}(\boldsymbol{C}_t) = \sum_i C_{i,t}$$

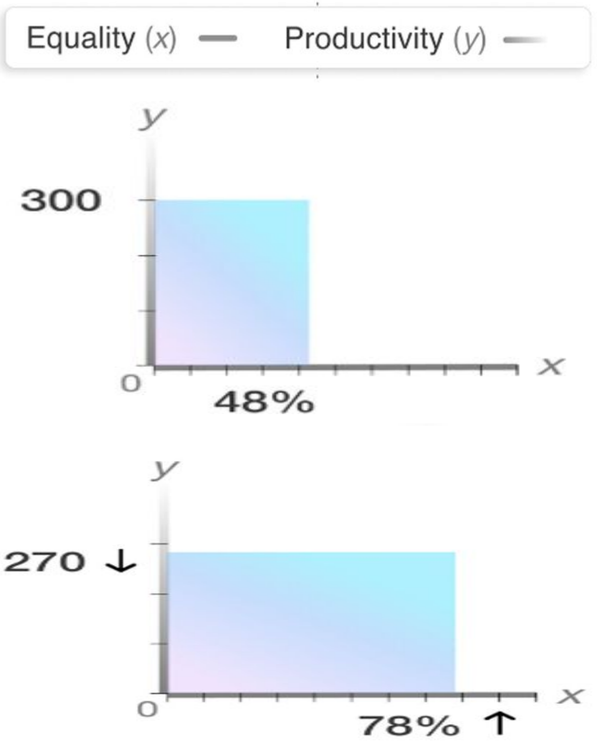
Total Wealth at Time t　　Wealth of Agent i in Time t

- Social Welfare

$$\mathrm{swf}_t = \mathrm{eq}(\boldsymbol{C}_t)\cdot\mathrm{prod}(\boldsymbol{C}_t)$$

$$\max_{\pi_p}\mathbb{E}_{\tau\sim\pi_p,a\sim\pi}\left[\sum_{t=1}^{H}\gamma^t r_{p,t} + \mathrm{swf}_0\right],\quad r_{p,t} = \mathrm{swf}_t - \mathrm{swf}_{t-1}$$

Tax Policy　Action Policy　Discounted Factor　Initial Welfare　Social Benefits or Costs　Social Welfare at Time t

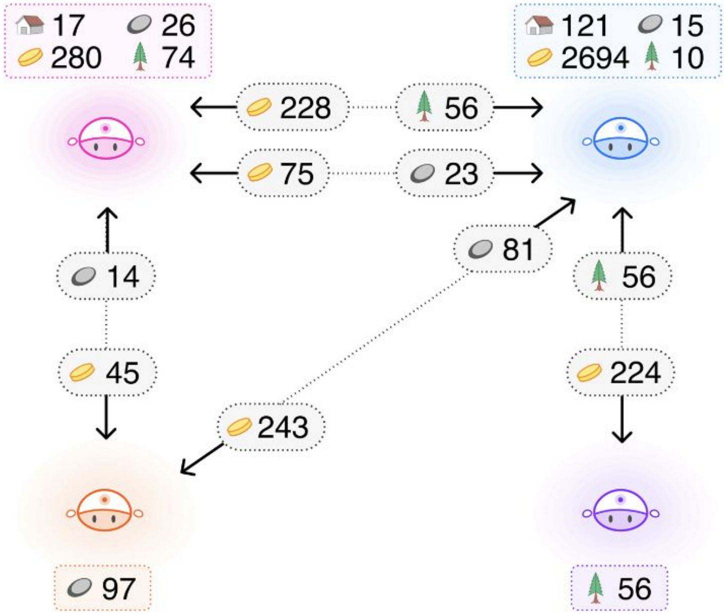



Source: Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. Science advances, 8(18), eabk2607.

# EV Based Decision Making


Decide Next Action

Expected
Value




$$\text{Net extra income} = [z_0 + \mathrm{EV} - T(z_0 + \mathrm{EV})] - [z_0 - T(z_0)]$$

For impossible actions: $\mathrm{EV} = -\infty$

Source: Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. Science advances, 8(18), eabk2607.

# EV Based Decision Making

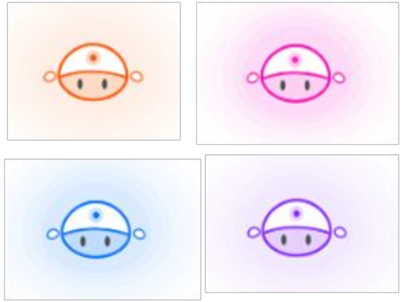
## **EV** of Building:

- Earning Rate

$$r(i,j) = n_v + 1$$

Earning Rate of a House

Number of Houses in its Von Neumann Neighbors




Γ Distribution

- Expected Value of Building

Building Position

Earning Rate

Life Expectancy

Agent Age

Time of Building a House

$$\mathrm{EV}_{\text{building}}(i,j) = \frac{r_{building}(i,j) \cdot (\tilde{t}_{\text{life}} - a - t_{\text{building}})}{t_{\text{building}}}$$

# EV Based Decision Making

## **EV** of Buying:

$$\mathrm{EV}_{\text{buying}}(i,j) = \frac{r_{building}(i,j)\cdot(\tilde{t}_{\text{life}} - a - (t_{\text{building}}+1)) - \text{cost}}{t_{\text{building}}+1}$$

Cost of Buying Resources

## **EV** of Selling:

Wood Price Amount of Wood Owned Stone Price Amount of Stone Owned

$$\mathrm{EV}_{\text{selling}}(i,j) = r_{\text{wood}}\cdot n_W + r_{\text{stone}}\cdot n_S$$

## **EV** of Gathering:

Amount of Wood at Position (i, j) Amount of Stone at Position (i, j)

$$\mathrm{EV}_{\text{gathering}}(i,j) = r_{\text{wood}}\cdot\min(1, n_{W_{ij}}) + r_{\text{stone}}\cdot\min(1, n_{S_{ij}})$$

## **EV** of Moving:

$$\mathrm{EV}_{\text{moving, gather}}(i,j) = \frac{r_{\text{wood}}\cdot n_{W_{ij}} + r_{\text{stone}}\cdot n_{S_{ij}}}{\min(n_{W_{ij}}, n_{S_{ij}})+1}$$

$$\mathrm{EV}_{\text{moving, build}}(i,j) = \frac{r(i,j)\cdot(\tilde{t}_{\text{life}} - a - t_{\text{building}} - 1)}{t_{\text{building}}+1}$$

# Decision Making Neuroevolution

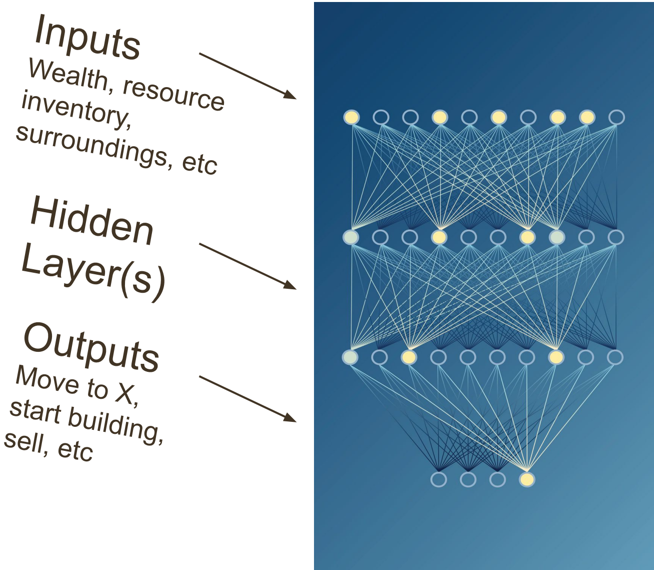
Expected value calculations replaced by a neural network

Same input values as EV calculation

Choose action with highest output value

Networks are not trained, but evolved

| Parameters of Neural Net | Value |
| --- | --- |
| Input Size | 34 |
| Output Size | 9 |
| Hidden Size | 40 |
| Number of Hidden Layers | 2 |
| Total Parameters | 3409 |



# Decision Making Neuroevolution

1

Initial agents receive neural networks with random weights and biases

2

Birth and death is synchronized

3

Calculate fitness of agent as

$$F = W_t - W_0$$

With $W_t$ as wealth of agent at time $t$

## After each generation:

1. **Parent Selection:** 'Tournament selection': Fittest agents reproduce

2. **Recombination:** 'Line recombination': merging the parents' networks

3. **Mutation:** Pick new values for small portion of network parameters

4. **Survivor Selection:** Replace all agents with new generation

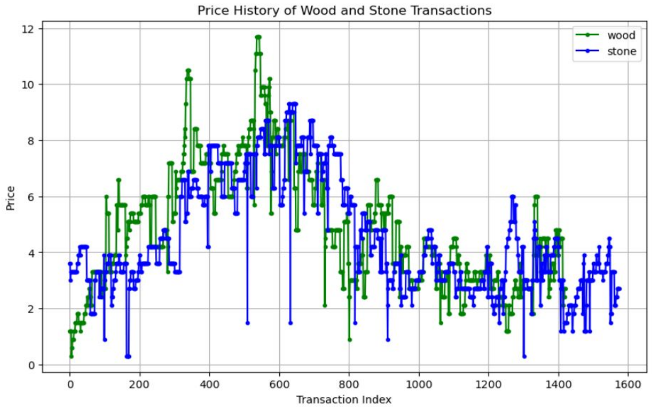
Over time, agents will evolve strategies!

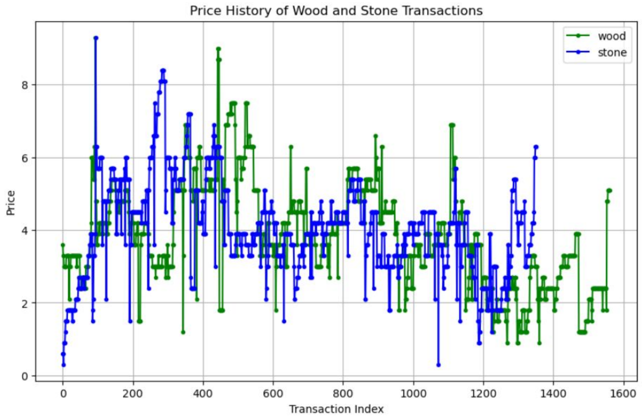
# Market Order Book

- Buy Orders ( Bids )
- Sell Orders ( Asks )

| | | |
|---|---|---|
| 2.6670 | 14.98 | 32,204.37 |
| 2.6660 | 12.68 | 32,189.39 |
| 2.6650 | 83.65 | 32,176.71 |
| 2.6640 | 0.72 | 32,093.06 |
| 2.6630 | 0.57 | 32,092.34 |
| 2.6620 | 4.13 | 32,091.77 |
| 2.6610 | 4,645.18 | 32,087.64 |
| 2.6600 | 2.28 | 27,442.46 |
| 2.6590 | 4,503.87 | 27,440.18 |
| 2.6580 | 22,936.31 | 22,936.31 |
| ↑ 2.6571 | ≈2.65 USD | |
| 2.6560 | 637.12 | 637.12 |
| 2.6550 | 3,373.68 | 4,010.80 |
| 2.6540 | 2.28 | 4,013.08 |
| 2.6530 | 2,829.52 | 6,842.60 |
| 2.6520 | 241.80 | 7,084.40 |
| 2.6510 | 266.88 | 7,351.28 |
| 2.6500 | 244.22 | 7,595.50 |
| 2.6490 | 401.55 | 7,997.05 |
| 2.6480 | 2,426.74 | 10,423.79 |
| 2.6470 | 721.88 | 11,145.67 |
| B 24% | | 76% S |

Example of an orderbook
Source: ByBit

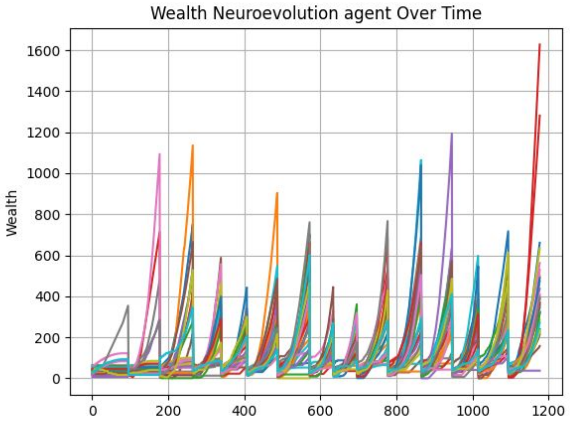


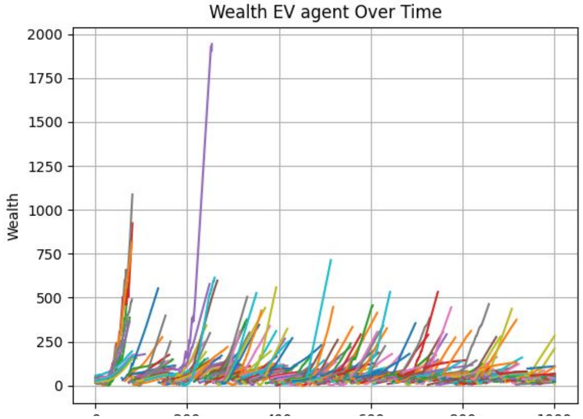



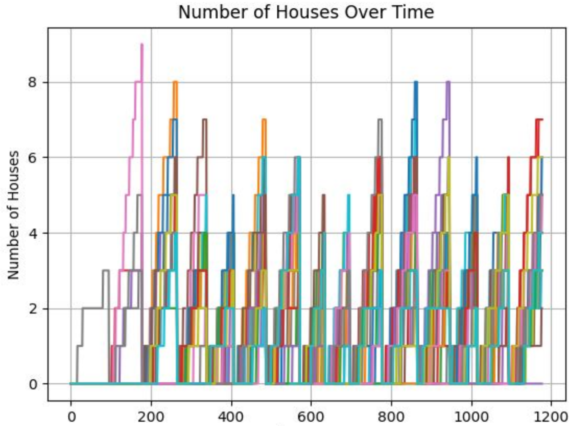
# Experiments and Results

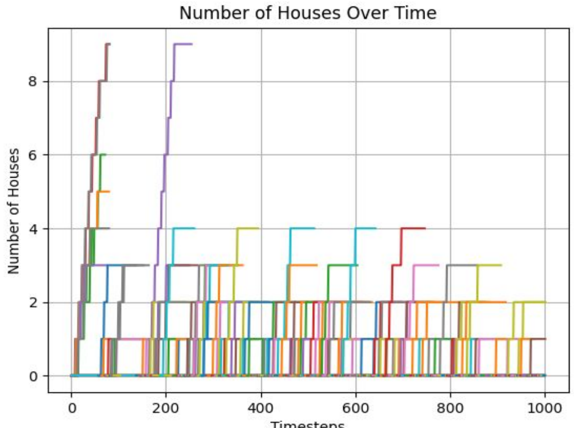
# Experiments: EDA

| | Parameter | Value |
|---|---|---|
| 1 | Grid Size | 40 x 40 |
| 2 | Number of Agents | 30 |
| 3 | Number of Timesteps | 1000 |
| 4 | Number of Resources | 500 |
| 5 | Resource Respawn Rate | 0.5 |
| 6 | Lifetime Mean | 80 |
| 7 | Lifetime Std Dev | 10 |
| 8 | Order Expiry Time | 5 |
| 9 | Tax Period | 1 |
| 10 | House Income per Timestep | 1 |
| 11 | House cost | (2,2) |
| 12 | lamba | 1 |



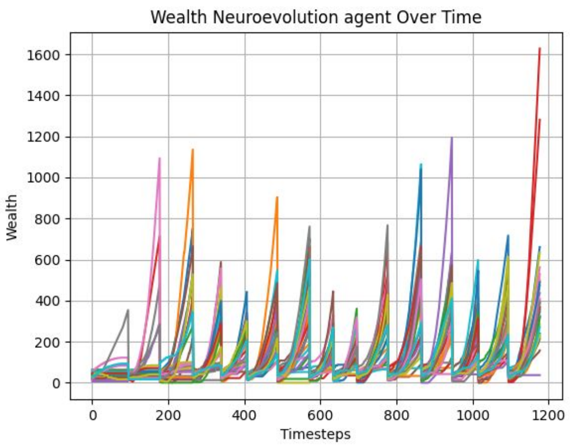



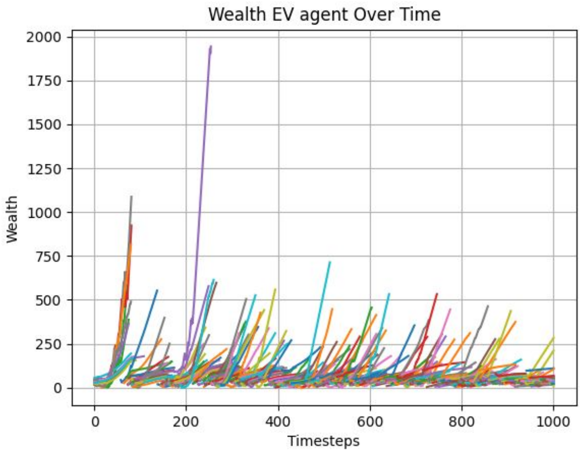



# Experiments: EDA

| | Parameter | Value |
|---|---|---|
| 1 | Grid Size | 40 x 40 |
| 2 | Number of Agents | 30 |
| 3 | Number of Timesteps | 1000 |
| 4 | Number of Resources | 500 |
| 5 | Resource Respawn Rate | 0.5 |
| 6 | Lifetime Mean | 80 |
| 7 | Lifetime Std Dev | 10 |
| 8 | Order Expiry Time | 5 |
| 9 | Tax Period | 1 |
| 10 | House Income per Timestep | 1 |
| 11 | House cost | (2,2) |
| 12 | lamba | 1 |



# Experiments: RQ1

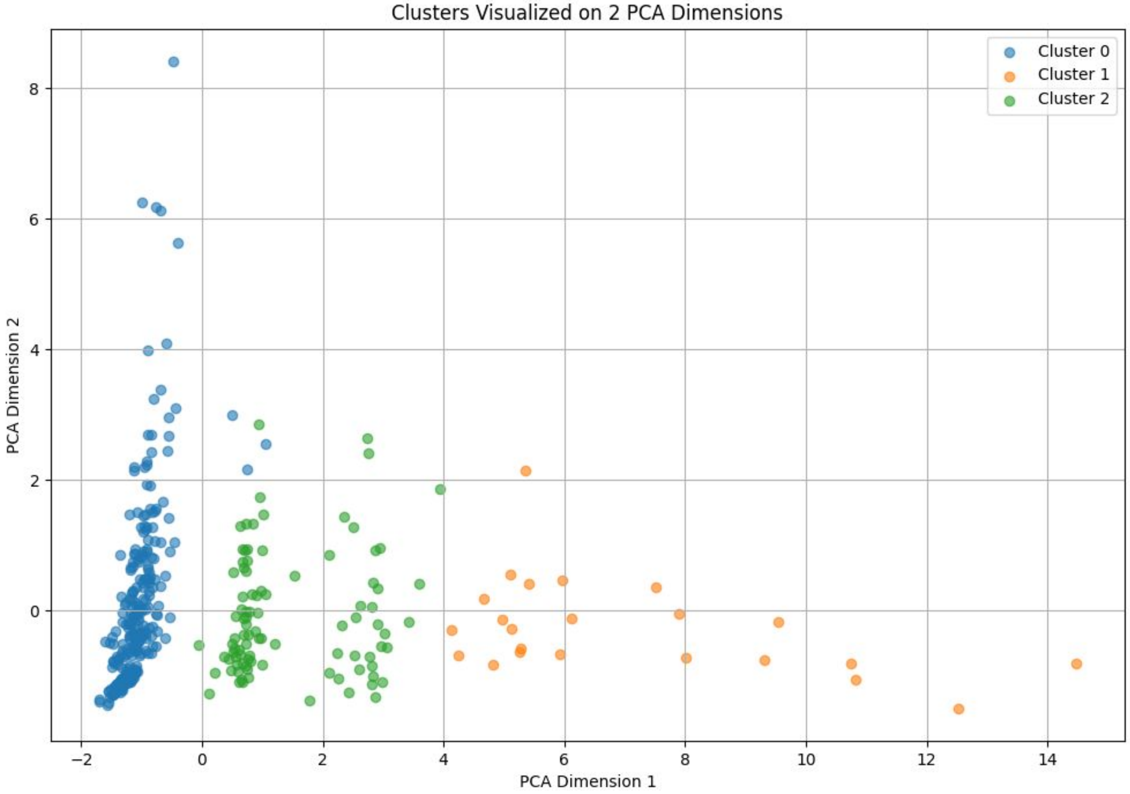
RQ1: How does introducing an asset, influence the emergent behavior classes?

H0a: We do get clusters with significantly different characteristics in buying and selling, gathering and selling and buying and building behaviors.

Use ANOVA on cluster feature importances.

# Experiments: RQ2

Agglomerative clustering on action space with n_clusters = 3




**Agglomerative Clustering**

1. Initialize Clusters:

Start with each data point as its own cluster.

2. Find and merge the closest pair of clusters.

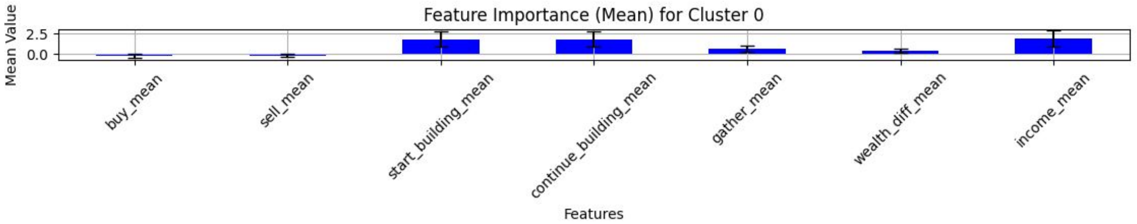
Update distances based on WARD linkage criterion (minimum variance within clusters)

3. Repeat Until Stopping Criterion:

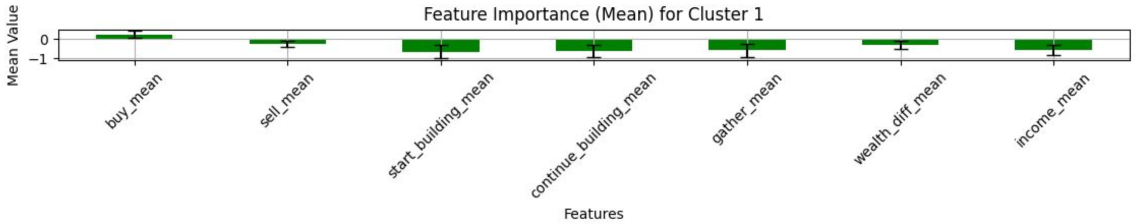
Repeat until n_clusters = 3

# Experiments: RQ2

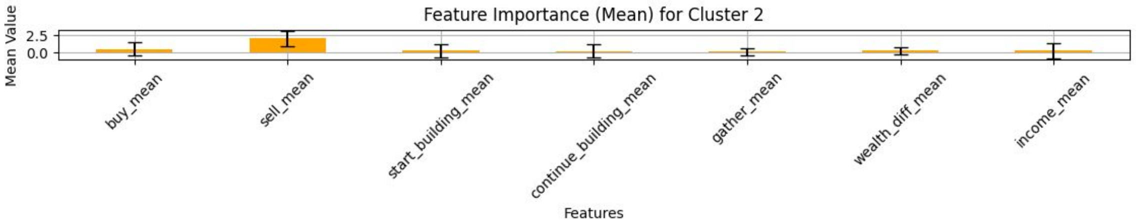
## Neuroevolution agents: Agglomerative clustering feature importances & ANOVA Results (10 runs)




**Wealthy Builder Class (Cluster 0)**
Shows significantly higher mean build and income rates compared to cluster 1, but not to 2. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc



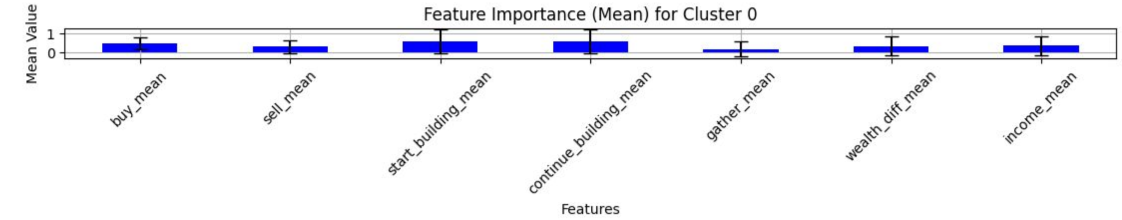

**Low Activity 'Poor' Class (Cluster 1)**
Significantly lower mean buy count, sell count, build time, wealth differential, and income compared to Clusters 0 and 2. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc



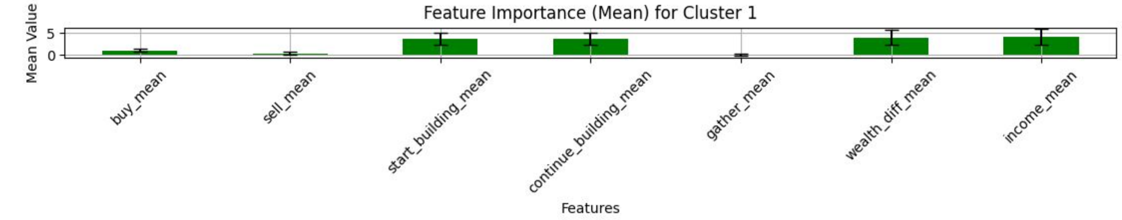

**Trader Class (Cluster 2)**
Significantly higher mean sell rate compared to Clusters 0 and 1 and higher buying, but not significant. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc

# Experiments: RQ2

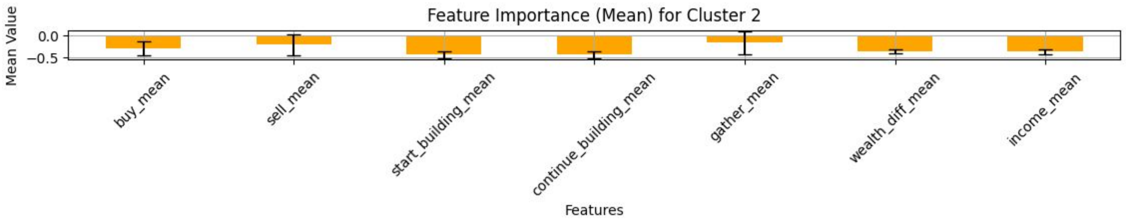
EV agents: Agglomerative clustering feature importances & ANOVA Results (10 runs)




**Average Builder Class (Cluster 0)**
Significantly higher mean buy count, build time, wealth differential, and income compared to Cluster 2. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc




**Wealthy Builder Class (Cluster 1)**
Significantly higher mean build count, build time, wealth differential, and income compared to other groups. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc




**Low Activity 'Poor' Class (Cluster 2)**
Significantly lower mean buy count, sell count, build time, wealth differential, and income compared to Clusters 0 and 1. Significant at alpha = 0.05 level using ANOVA with Tukey Post-hoc

# Experiments: RQ2

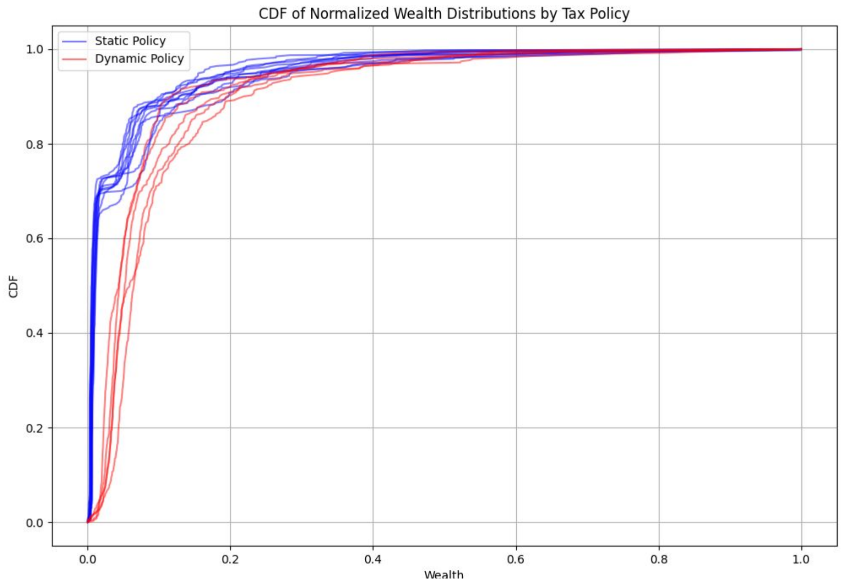
RQ2: What is the effect of static, vs. dynamic tax policy on wealth distribution?

H0b: There is no significant difference in total welfare between simulation runs with dynamic tax policy and those with static tax policy.

H0c: There is no significant difference in the Gini coefficient of wealth distribution between simulation runs with dynamic tax policy and those with static tax policy.

# Experiments: RQ3 Tax effect on Wealth

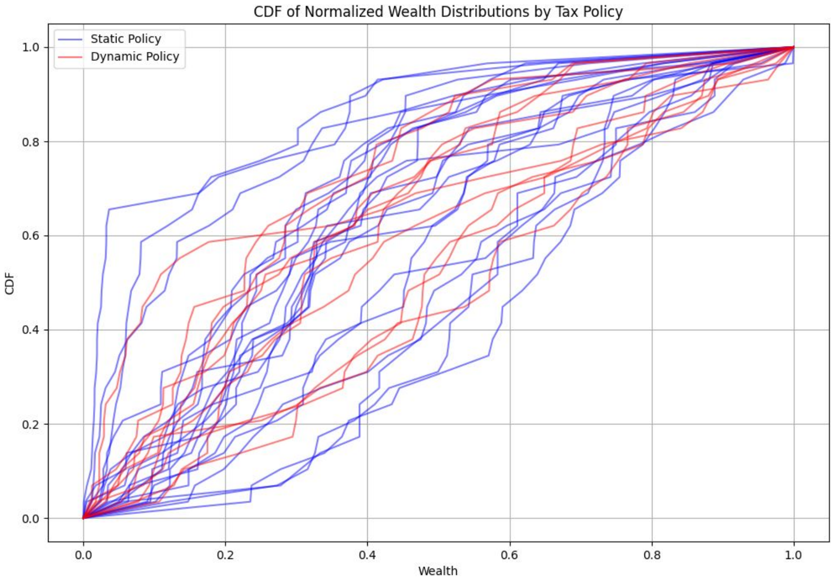
EV agents: Wealth distribution across 5 runs




| Metric | Static Policy Mean ± 1SD | Dynamic Policy Mean ± 1SD | KS Statistic | P-Value |
|---|---|---|---|---|
| Welfare | 22.43 ± 4.00 | 36.87 ± 5.86 | 0.8000 | 0.0193 |
| Gini | 0.7389 ± 0.0203 | 0.4858 ± 0.0284 | 1.0000 | 0.0007 |
| Productivity | 22.43 ± 4.00 | 36.87 ± 5.86 | 0.8000 | 0.0193 |
| Equality | 0.2592 ± 0.0203 | 0.5130 ± 0.0285 | 1.0000 | 0.0007 |
| Social Welfare | 5.83 ± 1.23 | 18.99 ± 3.52 | 1.0000 | 0.0007 |

# Experiments: RQ3 Tax effect on Wealth

Neuroevolution agents Wealth distribution across 5 runs




| Metric | Static Policy Mean ± 1SD | Dynamic Policy Mean ± 1SD | KS Statistic | P-Value |
|---|---|---|---|---|
| Welfare | 10.59 ± 3.43 | 11.14 ± 2.62 | 0.2500 | 0.7838 |
| Gini | 0.3794 ± 0.1175 | 0.3938 ± 0.0798 | 0.2500 | 0.7838 |
| Productivity | 10.59 ± 3.43 | 11.14 ± 2.62 | 0.2500 | 0.7838 |
| Equality | 0.6075 ± 0.1216 | 0.5926 ± 0.0825 | 0.2500 | 0.7838 |
| Social Welfare | 6.77 ± 3.11 | 6.76 ± 2.39 | 0.2000 | 0.9491 |

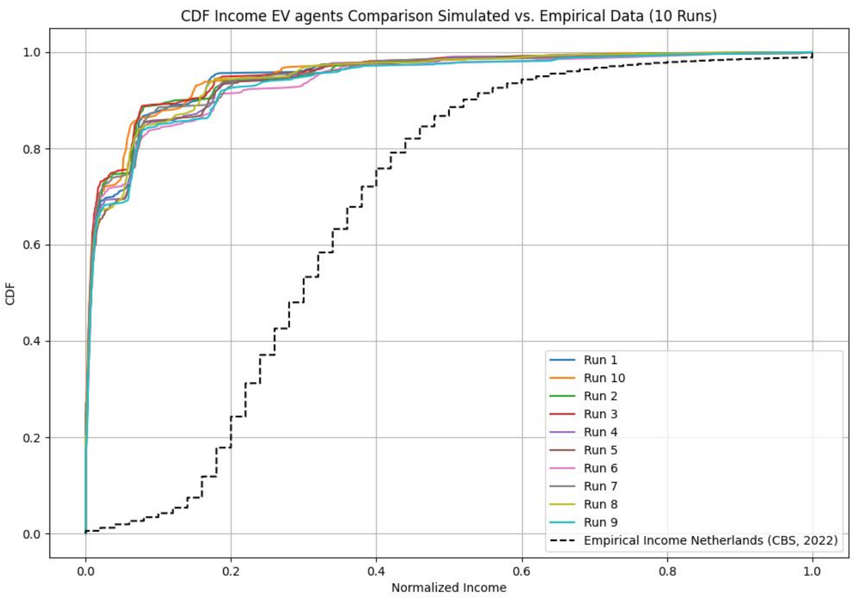
# Experiments: Validation

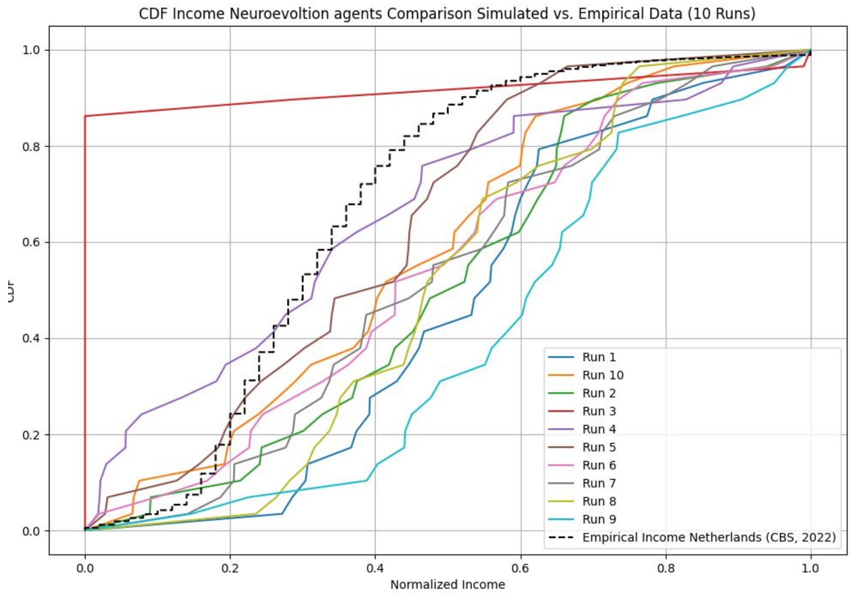
Validation: is simulated income distribution similar to Netherlands?

H0d: There is no significant difference on income distribution in the model and in the Netherlands.

- Use Kolmogrov Smirnov test on distribution.

# Experiments: Validation




- H0d: The 2 samples come from the same distribution, tested using KS test obtained p-value: 0.0 (10/10 times)-> H0a Rejected

# Conclusion

# Conclusion

RQ1: How does introducing an asset, influence the emergent behavior classes?

Buyer and Builder Class & low activity poor class emerges using EV and Neuroagents, Traders emerge in some simulations.

RQ2: What is the effect of static, vs. dynamic tax policy on wealth distribution?

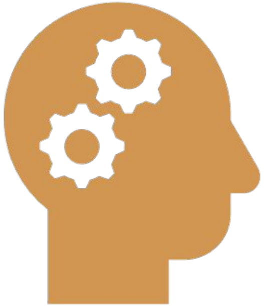
Significantly higher Gini coefficient and total welfare for EV agents, not for Neuro agents

Validation: Income distribution found significantly different from Netherlands for both EV and Neuro evolution agents.

Wealth distribution GINI coeff is similar (0.73 simulated vs. 0.71 empirical) for EV agents using static tax policy

# Limitations

# Limitations & Future work



**Limitations**

- Income from labour and from wealth is taxed (not the case in Netherland tax code)
- Burn in time Neuroevolution agents
- Only explored 1 set of parameters
- Limited amount of runs

**Contributions & Future work**

- Include assets and wealth heterogeneity instead of hard coding skill difference to produce emergent behavior
- Use dynamic tax policy to lower inequality in model
- Relevant for policy makers (as wealth is easier to measure than skill) and researchers (Including assets leads to emergent classes)

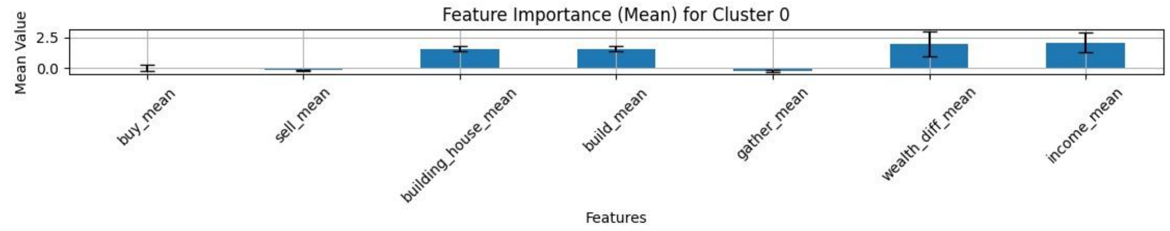
# References


- Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances, 8*(18), eabk2607. https://doi.org/10.1126/sciadv.abk2607Epstein, J. M. (2006). *Generative social science: Studies in agent-based computational modeling*. Princeton University Press.
- Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8
- Statistics Netherlands (CBS). (2022). Income distribution (standardised income). Retrieved June 7, 2024, from https://www.cbs.nl/en-gb/visualisations/income-distribution
- Ministerie van Financiën. (2023). *Vermogensverdeling in beeld*. Retrieved from https://www.rijksoverheid.nl/documenten/begrotingen/2023/04/28/voorjaarsnota-2023

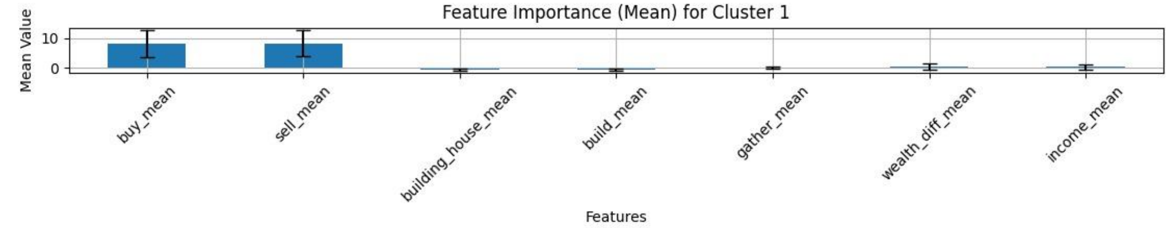

Thank You
Any Questions?

# Experiments: RQ2

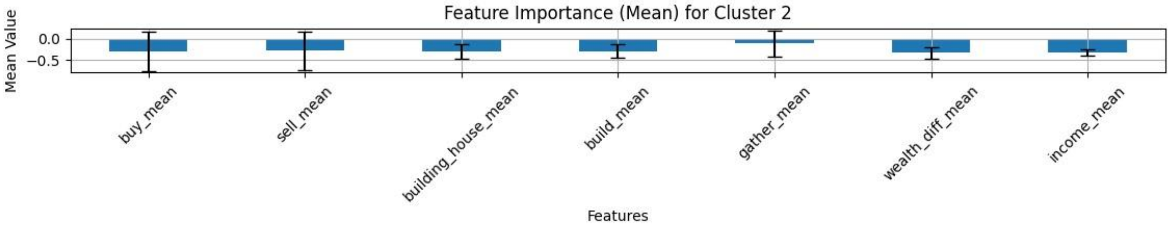
## Agglomerative clustering feature importances & ANOVA Results




**Wealthy builder Class**
Significantly higher mean build count, build time, wealth differential, and income compared to other groups.p-adj values (vs. cluster 1, 2)
Buy: 0.001, 0.9831
Build: 0.000, 0.000
Wealth differential: 0.0389, 0.0028
Income p-adj values: 0.0016, 0.0002




**Trader Class**
Significantly higher buy and sell rates.
Buy p-adj values (vs. cluster 1, 2): 0.001, 0.0007
Sell p-adj values (vs. cluster 1, 2): 0.0006, 0.0006




**'Poor' looking Class**
Lower wealth and higher gathering variability, although not significant
Gather p-adj values (vs. cluster 1, 2): 0.161, 0.7237
Wealth differential p-adj values (vs. cluster 1, 2): 0.4768, 0.3311
Income p-adj values (vs. cluster 1, 2): 0.3864, 0.5197