Queen's
Master of Management in Artificial Intelligence

MMAI 847

Capstone Project

FINAL REPORT

Aug 30, 2019 11:59 AM

Geoffrey Le Fevre (20140347)

Table of Contents

EXECUTIVE SUMMARY

Every network designer faces 3 key challenges when designing their governance processes which include, how to capture network input optimally, how to measure the performance of their mechanisms, and how to increase voter turnout of the right participants (and limit turnout for bad actors). The framework proposed in this paper focuses on analysing voter types, voter preferences, and strategies in the hope of creating a voting mechanism that is able to effectively capture network input for policy making. Analysing Aragon's voting mechanism with this framework the conclusions were that Aragon should implement flexible timing into its vote to reduce vote 'sniping', as well as working on further incentivising voters as part of the clusters that most exhibit 'proxy' voting behaviours.

The framework explored in this paper for analysing and evaluating network governance processes is just a start. The hope is that it will be expanded, and iterated upon in the future to build out more robust descriptive, predictive, and prescriptive tools for network designers.

As small decentralised economies gradually scale and become more complex, for their long term success it is crucial that proper governance processes are in place to effectively navigate the increasing complexity of decision making.

INTRODUCTION

10,000 ft

Blockchain first splashed into the public consciousness in 2009 when 'Satoshi Nakamoto' released the seminal whitepaper "Bitcoin: A Peer-to-Peer Electronic Cash System", describing trust-less peer-to-peer digital cash that could be sent and received without any intermediaries. [1] Since then, the underlying technology that enabled Bitcoin has continued to evolve, enabling more complex use cases. This evolution towards greater sophistication and complexity in the protocols themselves and the layers of

---

[1] Satoshi Nakamoto, https://bitcoin.org/bitcoin.pdf

interactions that have emerged above them, has meant that the codification of trust promised by blockchain is now left largely incomplete in all but the simplest networks where the consensus mechanism of the underlying chain is able to cover all interactions of the entire end-to-end system.[2]

A useful framework for viewing this emergent property of crypto-networks is through contract theory.[3] Contract theory posits that “all but the simplest contracts are incomplete”.[4] The underlying blockchains that enable these tokenised ecosystems are simple computers that are only able to codify trust for a small subset of the total interactions on a network, thus leaving many of the interactions that happen outside the simple logic of the consensus algorithm largely unaccounted for in the “decision logic” of the ecosystem.[5] For distributed and open source networks this presents serious challenges the must be overcome for them to scale successfully. For effective scaling network designers, they must either restrict the design space to ensure some level of contract completeness, or implement governance processes that allow the network to dynamically navigate the increasing complexity of decision making.

Network designers focus largely on the impact mechanisms have on the behaviours of network participants, but designing all required mechanisms to optimally align all possible incentives at network launch is an impossible task. Instead, best practices in the space follow an iterative agile approach starting with a basic MVP (minimum viable product) that is designed to be iterated throughout the lifecycle of the network as new behaviours and incentives gradually emerge. This approach means that the long term success of these networks requires sound governance frameworks and mechanisms to navigate the ever increasing complexity of decision making. “With good governance you can have any feature you want”

---

[2] Jesse Walden https://a16z.com/2019/07/22/incomplete-contracts/
[3] Jesse Walden https://a16z.com/2019/07/22/incomplete-contracts/
[4] Oliver Hart https://scholar.harvard.edu/files/hart/files/incomplete_contracts_and_control.pdf
[5] Jesse Walden https://a16z.com/2019/07/22/incomplete-contracts/

and network founders are increasingly realising that governance frameworks and mechanisms are critical for the long run success of their network.[6]

Governance

Governance frameworks and mechanisms are the structure and processes that take input from the network and output policies. It is centered around who makes decisions, what the scope of those decisions are, and how those decisions are implemented. Input can be captured very informally, such as through informal conversations and exchanges, or very formally by voting mechanisms. At first these governance mechanisms start off very informally, but over time naturally become more formal as changes to the network accumulate. The reason is because there is a lot of great historical work on voting in political systems and a lot of ongoing research and experimentation around network voting mechanisms - from futarchy (prediction markets), liquid democracy, quadratic voting, conviction voting, and pairwise auction mechanisms. The more complex voting mechanisms have yet to reach production standards, but there are some networks that have already deployed simple voting systems into live networks as a way to capture input for the purpose of setting policy.

The primary challenge network designers look to solve when designing and managing their network's governance framework is how to capture user input and turn it into a tangible outcome, in this case, policy. For the purpose of this paper I will examine three subsets of this high level problem that covers the most critical issues network designers face when looking to implement governance processes:

1) Input Capture: How should input be captured and used? *i.e. What voting mechanism should be used?*
2) Measurement: How does one measure the performance of voting mechanisms and intelligently iterate your governance processes going forward?

[6] Chris Burniske: https://www.businesswire.com/news/home/20181016005342/en/Decred-Launches-Politeia-Self-Governance-System-Release-20-Million

3) Voter Turnout: How does one properly engage network participants and incentivise them to participate in network voting?

Solution

Network designers need a framework for evaluating the performance of their voting mechanisms in order to intelligently iterate their designs. For this, clear voter profiles are required to properly analyse, incentivise, and engage network stakeholders to participate and strengthen voter turn-out. Through the lenses of political science and market design, we will analyse Aragon's voter data across 16 proposals focused on generating a framework to analyse which factors have the greatest influence over policy.[7] To support this, a voter taxonomy will be proposed that highlights voter strategies, preferences, and the overall effectiveness of the voting mechanism in capturing input, as well as a predictive model that takes this voter taxonomy into consideration.

BACKGROUND

Aragon

Aragon is a network that leverages the Ethereum blockchain to create the first online jurisdiction for people looking to create borderless organisations. They offer a suite of customisable tools and apps for organisational management including finance, token issuance, and voting. Aragon currently has live voting mechanisms in production that are being leveraged to decide on network proposals that deal with resource and fund allocations, product roadmaps, and governance structures.

At the date of data collection, Aragon's voting mechanism followed a five phase process. The first four phases walk a user through the process of writing and submitting a proposal to be voted on by the

[7] Martin Gilens and Benjamin I Page, Testing Theories of American Politics: Elites, Interest Groups, and Average Citizens, Princeton University Press 2012

network. In the last phase, submitted proposals are brought to a vote open to any ANT[8] token holders. Aragon follows a '1 token = 1 vote' design where each ANT token represents 1 vote. Voters are able to vote for or against new proposals by staking any amount of ANT against their vote, giving weight to their vote equal to the amount staked. Votes are cast over a fixed time limit, where the time of the end of the vote is known ahead of time. Proposals are accepted if they reach a quorum of at least 51%, meaning at least 51% of all tokens staked must all vote 'for' or 'against' a proposal for a decision to be reached.

Political Lens

In 2002, Martin Gilens a political scientist from Princeton analysed American data from 1779 policy issues to "test predictions about which sets of actors have how much influence over public policy: average citizens; economic elites; and organised interest groups.[9] Data collection for the research was a monumental undertaking, and the results showed that economic elites and organised groups have significant influence on public policy whereby the average citizen have little to no influence on public policy (**See APPENDIX/Supplementary Materials/Political Lens**). The key insight pulled from this report was that different types of voters have varying impacts on the outcome of public policy. This insight was very influential for the proceeding analysis of Aragon's voting system.

Economic/Market Design Lens

In 2002 Alvin Roth, an Economist from Harvard analysed how different auction designs had an impact on bidding strategies and outcomes. Auctions across Ebay and Amazon were analysed. Both were ascending auctions, but differed in timeline. Ebay's time was fixed, where the auction stopped at a pre-set time, and Amazon's was flexible and ended only when no new bids were made for 10 minutes. [10] In auctions there are generally two classes of behaviour; proxy, and snipe bidding. Proxy bidding is a strategy to

[8] ANT is Aragon's native token.
[9] Martin Gilens and Benjamin I Page, Testing Theories of American Politics: Elites, Interest Groups, and Average Citizens, Princeton University Press 2012
[10] Alvin E. Roth & Axel Ockenfels, 2002. "Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet," *American Economic Review,* vol. 92(4), 1093-1103.

incrementally bid up to a bidder's maximum valuation at the beginning of an auction, whereas sniping is a bidding strategy to bid at the last moment. The analysis found that the fixed time line of Ebay's auction led to more 'sniping' bids. **(See APPENDIX/ Supplementary Materials/Market Design Lens)**. As a market designer, sniping signals that a market is not thick enough (not enough bids at any period of time) or safe (strategically not safe for a bidder to reveal their true preferences). As Aragon's voting system operates very similarly to an auction where voters stake a specific amount of ANT based on a voter's own valuation of the proposal, the insights and framework presented were very influential for the proceeding analysis of Aragon's voting system.

Initial Project Direction & Pivot

The initial objective of the project was to use a generative adversarial network (GAN) to generate realistic network data that could be used to inform design decisions for early stage networks that have yet to launch. After closer evaluation, the original project was deemed unfeasible. Generating nuanced, realistic economic data using a GAN was intractable , and the potential insights gained from the data that could be generated could not be used in a production setting. Maintaining the original objective of providing a tool that can inform design decision-making, the objective of the project shifted to analysing network data from pre-existing networks and eventually narrowed to the topic of network governance, a key factor for the long term sustainability of these networks. Using existing data the following models aim to be a starting point for a more robust descriptive, predictive, and prescriptive tool for network designers that are starting to approach and analyse their own network's governance mechanisms.

IMPLEMENTATION

Data Exploration

The data used for this analysis was collected from Alethio, an Ethereum Data analytics Platform.[11] Alethio did a great job of cleaning the data, there were no missing observations and no data cleansing was necessary. The original dataset contained 6 features **(SEE APPENDIX/GITHUB REPOSITORY/DATA/aragon-all-votes.csv for full dataset)**:

- **voting_number**: (int64) Integers between 0-15, marking which of the 16 AGPs proposals analysed the vote took place in.
- **timestamp**: (object) The timestamp of the vote.
- **voter**: (object) The wallet address of the voter. What is interesting to note is that wallet balances and transaction history can be obtained by copying the wallet address into Etherscan's search.[12]
- **vote**: (object) Either a "Yes", or "No", marking the preference of the voter.
- **staked_ANT_amount:** (float64) a non-negative number indicating the ANT balance staked on the vote.
- **tx**: (object) The transaction hash. This hash links to the transaction on the Ethereum blockchain and confirms the provenance and legitimacy of the vote. Similarly this transaction hash can be searched by copying the transaction hash into Etherscan's search.

An additional feature was engineered and added to the data frame that labelled the final result of each vote. This feature was added using a simple excel formula that for each proposal summed, all the staked ANT for "Yes" and "No" votes and determined for which decision quorum was reached (51% majority).

- **Result:** (object) Either "Yes" or "No", indicating whether the AGP proposal passed or failed.

[11] https://aleth.io/
[12] https://etherscan.io/

Correlation

Correlations amongst the features were analysed. (**See APPENDIX/ DATA EXPLORATION/ Correlation)**. Most interestingly timestamp and **voting_number** were highly correlated (0.85) indicating that perhaps certain wallet addresses were highly correlated with similar timestamps. The correlation analysis indicated that all features were not too closely correlated to not be included in any prediction model, except for **timestamp** & **voting_number**.

Unique Features

The shape of the dataset was inspected as well. The dataset included 639 rows and 8 columns. What is interesting to note is that when inspecting unique voter values there were only 113 rows, indicating that there were at least 113 voters that voted in more than one proposal. The network was visualised and the visualisation confirmed this. (**See APPENDIX/DATA EXPLORATION/UNIQUE VOTERS & NETWORK GRAPH**).

Data Preparation

To prepare the dataset for modelling some of the features had to be transformed. First, vote and Result were transformed into either 0 or 1 (int) to match either their previous "No" or "Yes" values, respectively. Second the timestamp (object) was turned into a usable timestamp format for modelling (datetime64). The last step was to normalise the timestamp and **staked_ANT_amount** between 0-1 for each proposal. I.e. the first vote and last vote in a proposal received a 0 and 1, respectively.[13]

[13] Similarly, the lowest stake in each proposal received a 0, and the highest stake received a 1.

MODEL SETUP

Clustering

In order to reconcile the political and market design lenses described above, a modelling decision was made to cluster voters based on voting strategies. Given a voter preference, each voter's strategy was limited to the time of the vote and the amount to stake. Therefore to cluster voters based on their voting strategies, voter timestamp and **staked_ANT_amount** were the two factors considered when clustering voters. These scatterplots were analysed across all votes and each vote individually, using the normalised timestamp and **staked_ANT_amounts**. (**See APPENDIX/ CLUSTERING/ Data Grouped by voting_number**).

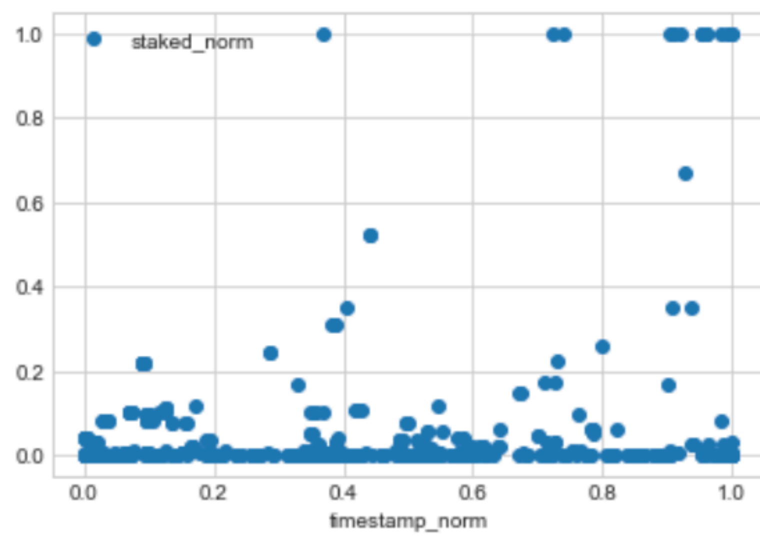


Figure 1: Data used for Clustering[14]

To cluster voters a K-means algorithm was used, with the optimal amount of clusters found using the 'Elbow method' (**See APPENDIX/CLUSTERING/Selecting K**). From this, each vote was assigned to a cluster, and a new cluster feature was added to the dataset, indicating which cluster a vote was assigned to.

[14] X Axis: Normalied timestamp Y Axis: normalised amount staked

The K-means clustering algorithm works by minimising the "cluster variation" which is calculated by summing all the pairwise "squared Euclidean distances" between all observations in a particular cluster.[15] The Euclidian distance refers to the "length of the straight line between two points.[16]

Gini Calculation

To further support the analysis of voter strategies and their impact on public policy a Gini - coefficient was calculated across all votes and for each vote individually (**See APPENDIX/GINI**). To calculate the Gini the normalised time stamp (X-axis) and normalised stake amount were used (Y axis). (**See APPENDIX/GINI**) The purpose of this was to output a number that indicated how evenly votes and stakes were spread across the total time of the vote. Similarly to how Alvin Roth showed the impact of votes in the last minutes of an auction, the rational behind this calculation was to demonstrate and visualise how votes and stakes were distributed across the time of a vote to help identify specific voting behaviours (sniping or proxy voting), and propose a potential measurement to track voting performance over time.

Therefore, the Gini- Coefficient is the number that measures the "degree of inequality in a distribution".[17] The Gini-Coefficient ranges between 0 and 1, with 0 meaning complete equality, and 1 meaning complete inequality.

Predictive Model Setup

A prediction model needed to be developed to analyze the impact of various voting features on the policy outcome of voting. Due to the limited data available and its explainability, a Logistic Regression model

---

[15] 1.4 MMAI 869 Unsupervised ML – Clustering
[16] 1.4 MMAI 869 Unsupervised ML – Clustering
[17] https://en.wikipedia.org/wiki/Gini_coefficient

was chosen, where the **Result** was the target for prediction (y = Result).[18] A logistic regression model predicts the classification of an observation based on its features.

The dataset was biased towards Passing, 378 pass, 133 fails so to avoid overfitting oversampling (SMOTE) was used to add additional “Fail” results to balance the dataset equally between “Pass” and “Fail” before building the Logistic Regression models (**See APPENDIX/GITHUB REPOSITORY/Predictive-Modelling/Voter Regression.ipynb**).

Three separate logistic regression models were instantiated (**See APPENDIX/LOGISTIC REGRESSION**)

- Model 1: The first looked solely at the normalised timestamp (**timestamp_norm**), and amount of ANT staked (**staked_norm**).
- Model 2: The second model added the cluster feature from the clustering analysis (cluster).
- Model 3: The third model added voter preference into the analysis (vote).

All three models were evaluated and compared using the following metrics:

- Confusion matrix[19]: Looks at the number of true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN).
- Accuracy score: The percentage of votes that were classified correctly.[20]
- Precision: How many of the pass predictions were actually true.[21]
- Recall: Of the actual pass predictions, how many were actually a pass.[22]

[18] 1 for pass, 0 for fail
[19] 3.1 MMAI 869 Performance Measures
[20] 3.1 MMAI 869 Performance Measures
[21] 3.1 MMAI 869 Performance Measures
[22] 3.1 MMAI 869 Performance Measures

- F1-score: Combines precision and recall into one number using the harmonic mean between the two.[23]
- ROC Curve and Area Under the Curve (AUC): The ROC curve graphs the True Positive Rate, and the False Positive Rate. From this curve the AUC can be calculated, which is the area unto the ROC curve, where 1 means a perfect prediction, 0.5 means prediction is random, and 0 which means predictions are exactly wrong.[24]
- Cross validations core: Splits the data into K-folds and iterates through each fold using 1 fold for testing and the others for training. At the end of the loop takes returns the average accuracy score across all fold iterations.[25]

RESULTS

Gini

The Gini Coefficient across all network votes was 0.21. This number suggests that stakes and votes were relatively equally distributed across voting times. However closer inspection of the Lorenz curve suggests that towards the end of the votes (~0.9 of the normalised timestamp), less then 60% of all stakes have been cast. This suggests that there is quite a lot of snipping occurring in Aragon's voting periods, and furthermore that Aragon's voting system is not safe for voters to reveal their true preferences early in the voting period, negatively impacting Aragon's ability to efficiently collect network input.

[23] 3.1 MMAI 869 Performance Measures
[24] 3.1 MMAI 869 Performance Measures
[25] 3.1 MMAI 869 Performance Measures

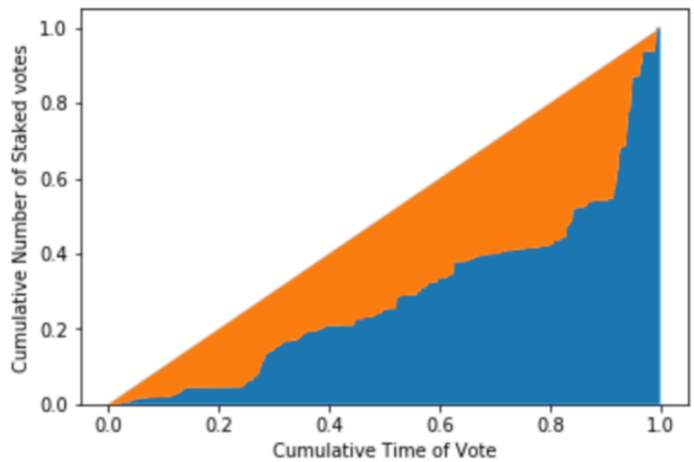


Figure 2: Lorenz Curve Across All Votes

Due to the Simpson's paradox, which discusses how micro trends can disappear when aggregated together, another concern was that trends in the Gini Coefficient across votes was not being captured by this aggregated number. To mitigate this risk the Gini was also calculated and tracked for each proposal.

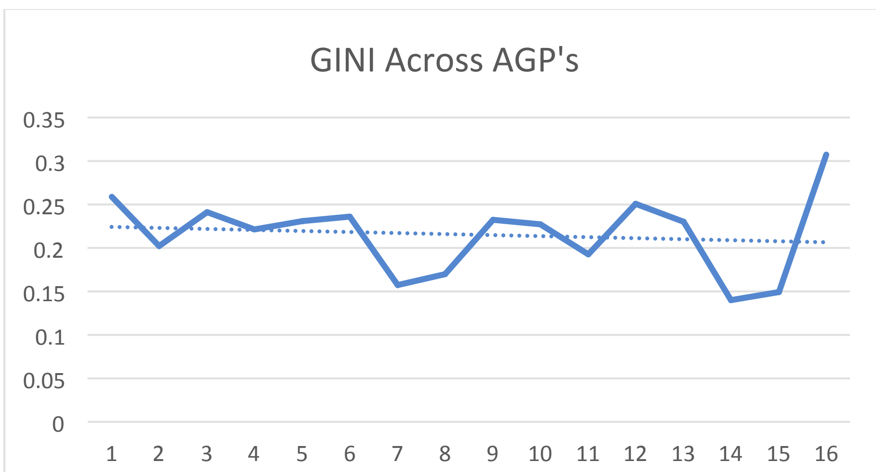


Figure 3: Gini Across All Proposals

CLUSTERING

Using some trial and error and the elbow method, the optimal number of clusters found was to be 4.

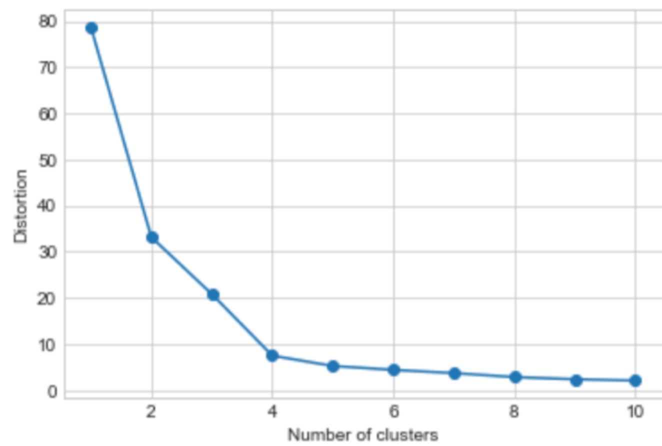


Figure 4: K Cluster Selection

Plotting the four clusters showed that Cluster 1 is strongly associated with sniping behaviour. Cluster 2,3,4 take up the majority of the votes, and due to the tight cluster suggest a lot of proxy bidding behaviour.

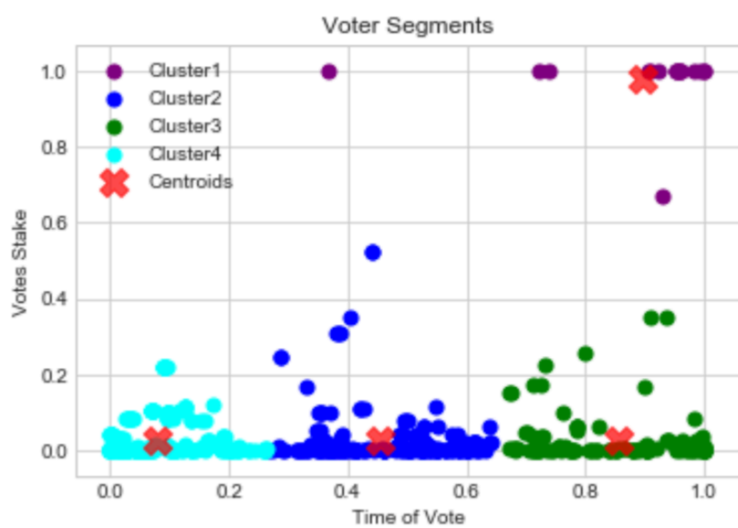


Figure 5: Cluster Visualised

Each cluster was further analysed to see its impact on policy performance. It was found, perhaps unsurprisingly that Cluster 1, despite being the smallest in terms of votes, having the greatest influence on public policy.

	Total Votes	Total Pass	Total Fail	Pass %
Cluster 1	17	16	1	0.94117647
Cluster 2	231	174	57	0.75324675
Cluster 3	110	81	29	0.73636364
Cluster 4	281	204	77	0.72597865

Table 1: Cluster impact on public policy

REGRESSION

(**See APPENDIX/LOGISTIC REGRESSION for full details**)

Model 1

The first logistic regression model just used the normalised timestamp and amount staked to predict whether a proposal was passed or not. This model only obtained an accuracy of 59% with an AUC of 0.57, and cross validation accuracy score of 55%.

Model 2

The second model added each votes cluster label as a new feature, and was able to achieve an accuracy of 63%, and an AUC of 0.57, almost identical to the first model's AUC, and cross validation accuracy score of 55%.

Model 3

The third model further incorporates each voter's preference as a predictive variable in the model. This model received significant jump in performance, achieving an accuracy score of 81%, and an AUC of 0.78, and cross validation accuracy score of 79%.

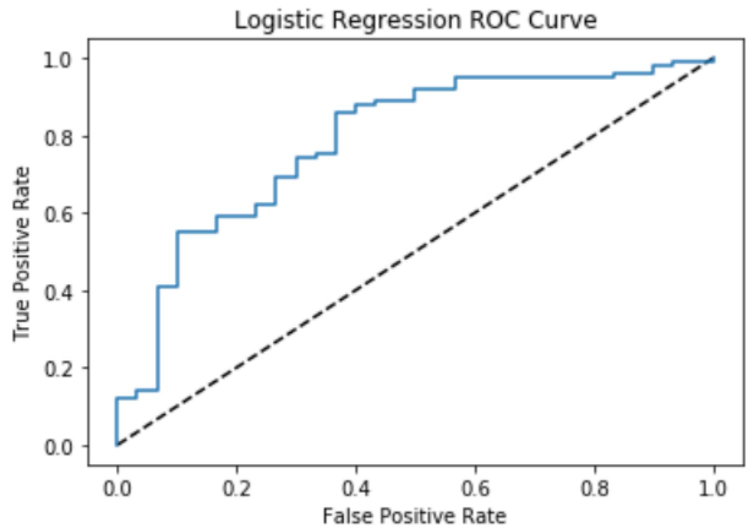


Figure 6: Model 3 AUC

CONCLUSION

Gini Coefficient

The Gini for each proposal suggested that stakes were relatively evenly distributed across the time of the vote. Furthermore when the Gini was calculated for each vote separately and plotted (**See APPENDIX GINI**) it showed the Gini was exhibiting a decreasing trend over time which indicates that network votes and stakes generally become more evenly distributed across votes. This suggests the sniping was becoming less prevalent. However upon closer inspection of the Lorenz Curve for each specific vote, and the aggregate Lorenz Curve, it appeared the sniping was still occurring, and in fact occurred in each vote. This means that Aragon is not efficiently collecting input as it is not safe for the majority of voters to reveal their true preferences and valuations early in the vote. A very generalised observation is that in many of the votes roughly 50% of all staked votes had occurred before the last moments, but that in the very last moments of a vote almost always sniping votes were submitted. However due to the volume of proxy bidding behaviour found throughout the majority of the vote (Cluster 2,3,& 4), this had very little effect on the measure of equality used. This measure can be improved by incorporating the number of total votes over time as opposed to just the amount staked.

Voter Clustering

Voter clustering is an extremely valuable exercise and should be conducted for all networks that have a voting mechanism. Knowing who the different types of voters are can help design better mechanisms that optimise how these networks are able to capture input and engage the right participants and support greater voter turn-out. These clusters presented in this paper were a preliminary exploration but should be expanded. Identifying Cluster 1 was the most important cluster in terms of valuable insights for network designers. The presence of vote sniping is not an optimal behaviour and needs to be minimised. All wallet addresses associated with sniping behaviour have been grouped for future tracking (**See APPENDIX/ GITHUB REPOSITORY/SNIPERS**). There was not much valuable distinction between Cluster 2,3, and 4. These clusters could be grouped into a single cluster as all the notable behaviour in these clusters seem to suggest proxy voting. Proxy bidding is in fact a desirable quality so efforts should be made to further incentivise this behaviour. Voter clustering was a valuable exercise and in this case highlighted the different strategies that voters follow in Aragon's voting system, and the impact that various clusters had on the impact of public policy. This work should be expanded to include data on the wallet balance and transaction history of each voting wallet address, and other clustering algorithms should be explored such as an SVM or bi-clique approximation.

Policy Predictions

The fact that the accuracy improved as more features were added to the logistic regression should not be a surprise and in and of itself does not prove which feature are the most important predictors of public policy. What is an important takeaway from this experiment is that mapping out the types of voters, their preferences and strategies are all critical factors that determine the final outcome of public policy, a cross validation score of 79% is still significant. Voter types (cluster membership), preferences (vote of Yes, or NO) and voting strategies (measured by the timestamp and amount staked) are all critical areas that need to be analysed when determining the effectiveness of a networks governance framework in turning network input into public policy. These three pillars should be the focus of any analysis of a network's voting mechanisms going forward.

Aragon's Voting System

The most important takeaway from this analysis for Aragon's governance designers is that due to the fixed time limit on the vote, vote sniping is a prevalent behaviour, and as discussed should be minimised. The simplest solution to this is perhaps implementing a flexible time limit on votes to make sniping more difficult. Furthermore to improve their ability to capture input, voters part of Cluster 2,3, and 4 should be analysed and increasingly targeted and incentivise to continue their proxy bidding behaviour.

Identifying Cluster 1 Voters is not necessarily helpful for designers due to the ease of setting up a new wallet address and transferring tokens to it. To make this list of Cluster 1 Voters more helpful (**See APPENDIX/ GITHUB REPOSITORY/SNIPERS**) reputation mechanisms should be explored to incentivise voters to set up a single voting address where a voter identity can be verified.

Future Work & Roadmap

The framework for analysing network governance, specifically voting mechanisms in this paper should just be viewed as a starting point. All code will be posted on Github[26] with public access rights in the hope that others interested in this topic will be able to build on it. Future development of this framework should center around four pillars.

1) Data Collection: Aragon's voting system is relatively new so data was limited, but since collecting data there has been more proposals brought to a vote and more still in the pipeline. There is also opportunity to expand this dataset with proposal specific data such as proposal topic,

[26] Github Repository URL: https://github.com/gtklefevre/MMAI-Capstone

public sentiment etc. Natural language processing (NLP) would be a natural fit for this type of data extraction. Another opportunity to expand the dataset would be to do forensic analysis on each voter's wallet address to extract features such as token balances, transaction history, and time the wallet has been active/inactive.

2) Voter Clustering: Further work and study should be done around the optimal way to group voters into segments. This paper proposes using strategic considerations for clustering, but more analysis into wallet address token balance and transaction history should also be included.

3) KPIs for Vote Performance: Further work and study should be done around developer key metrics for evaluating voter performance over time. This paper proposed using a Gini Coefficient calculated from the cumulative amount staked over the time of a vote, however this should be expanded to include the frequency of votes themselves.

4) Public Policy Predictions: This paper suggested three key factors to consider when creating predictive models regarding public policy (voter types, voting strategies, and voter preference), however further features should be explored to improve the effectiveness of this model.

APPENDIX

GITHUB REPOSITORY

https://github.com/gtklefevre/MMAI-Capstone

DATA EXPLORATION

Features

```
df.dtypes
```

```
voting_number          int64
timestamp             object
voter                 object
vote                  object
staked_ANT_amount    float64
tx                    object
Vote Weight            int64
Result                object
dtype: object
```

Data Shape

```
df.shape
```

```
(639, 8)
```

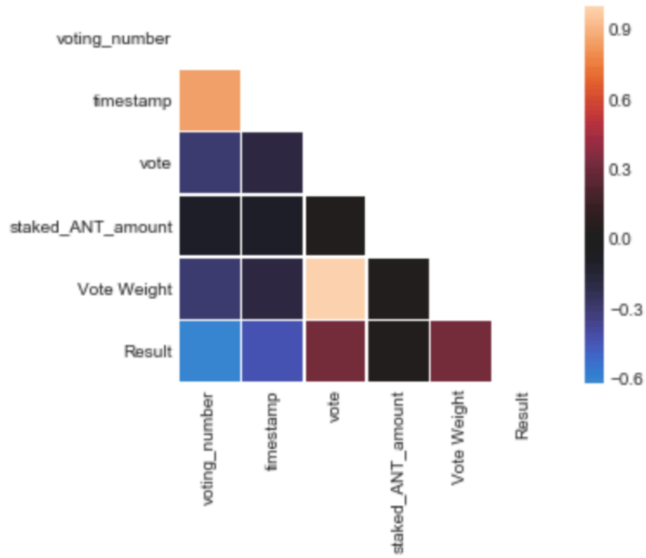
Unique Voters

```
df.voter.unique().shape
```

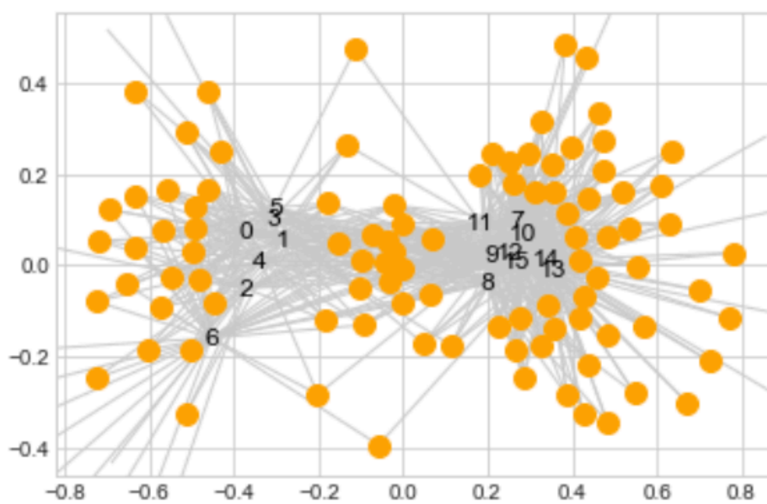
```
(113,)
```

Correlation

	voting_number	timestamp	vote	staked_ANT_amount	Vote Weight	Result
voting_number	1.000000	0.849591	-0.287119	-0.083480	-0.287119	-0.613199
timestamp	0.849591	1.000000	-0.184360	-0.080246	-0.184360	-0.431954
vote	-0.287119	-0.184360	1.000000	0.032972	1.000000	0.338292
staked_ANT_amount	-0.083480	-0.080246	0.032972	1.000000	0.032972	0.041700
Vote Weight	-0.287119	-0.184360	1.000000	0.032972	1.000000	0.338292
Result	-0.613199	-0.431954	0.338292	0.041700	0.338292	1.000000



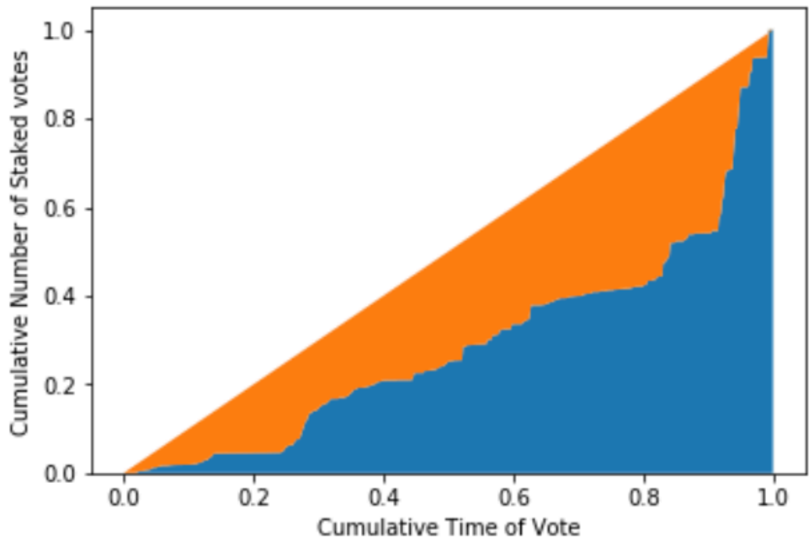
Network Graph



GINI

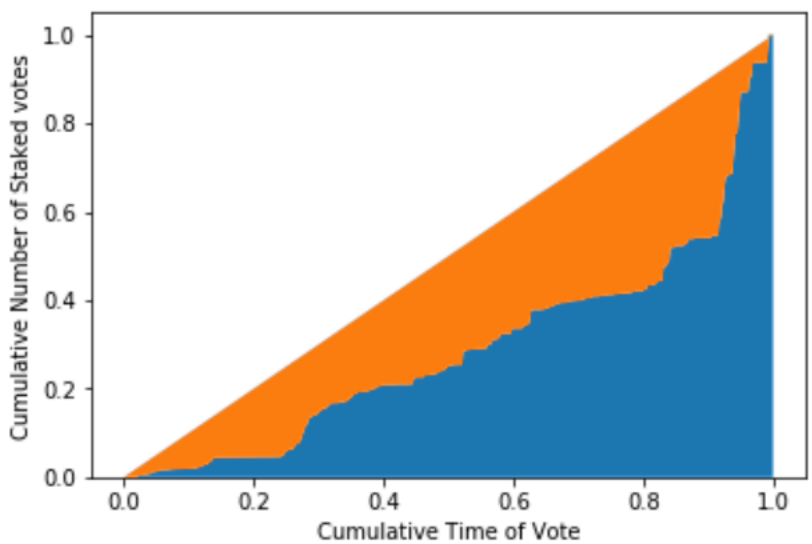
Aggregate

```
Gini: 0.211, Max. Gini: 0.211, Normalized Gini: 1.000
```



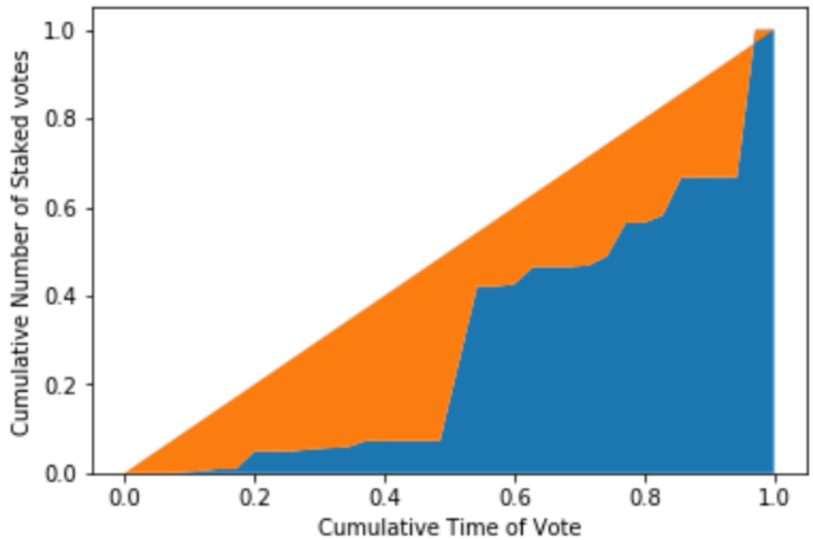
AGP 1

```
Gini: 0.259, Max. Gini: 0.259, Normalized Gini: 1.000
```



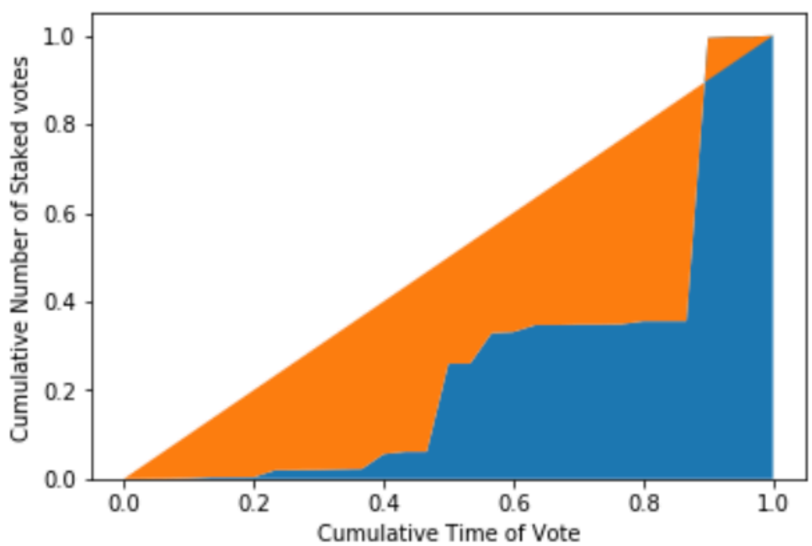
AGP 2

```
Gini: 0.202, Max. Gini: 0.202, Normalized Gini: 1.000
```



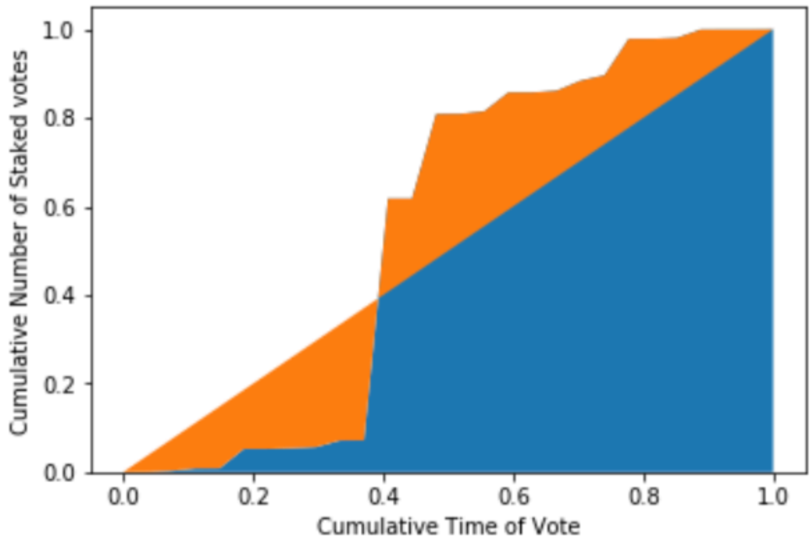
AGP 3

`Gini: 0.241, Max. Gini: 0.241, Normalized Gini: 1.000`



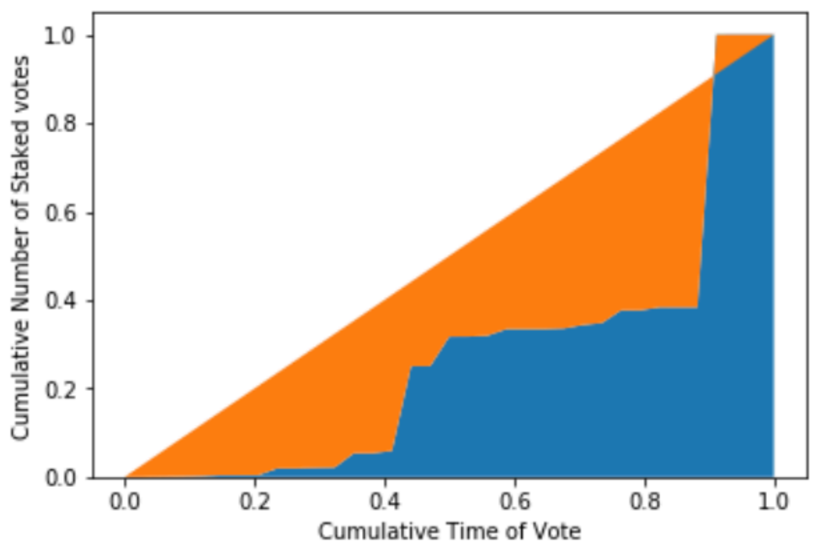
AGP 4

`Gini: -0.050, Max. Gini: -0.050, Normalized Gini: 1.000`



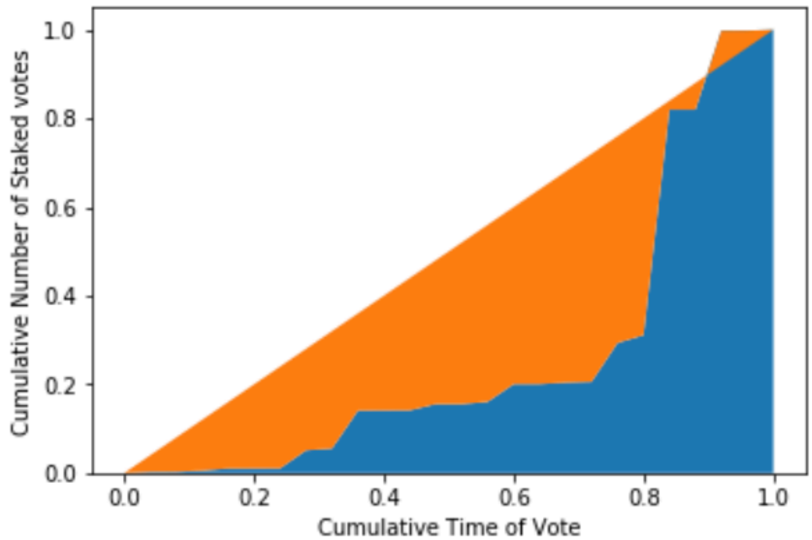
AGP 5

Gini: 0.231, Max. Gini: 0.231, Normalized Gini: 1.000



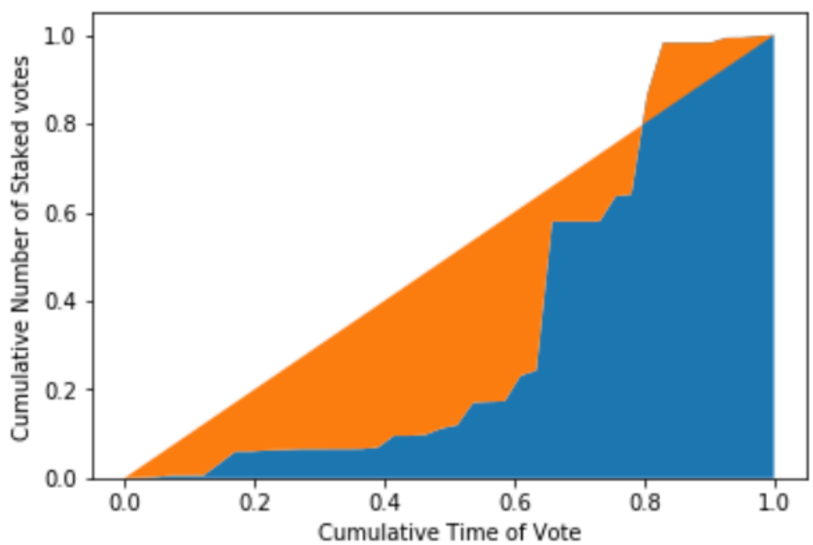
AGP 6

Gini: 0.236, Max. Gini: 0.236, Normalized Gini: 1.000



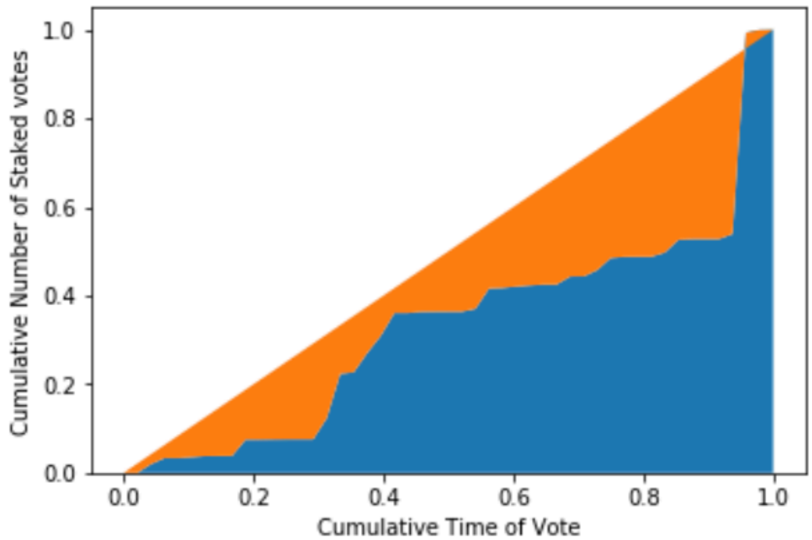
AGP 7

Gini: 0.157, Max. Gini: 0.157, Normalized Gini: 1.000



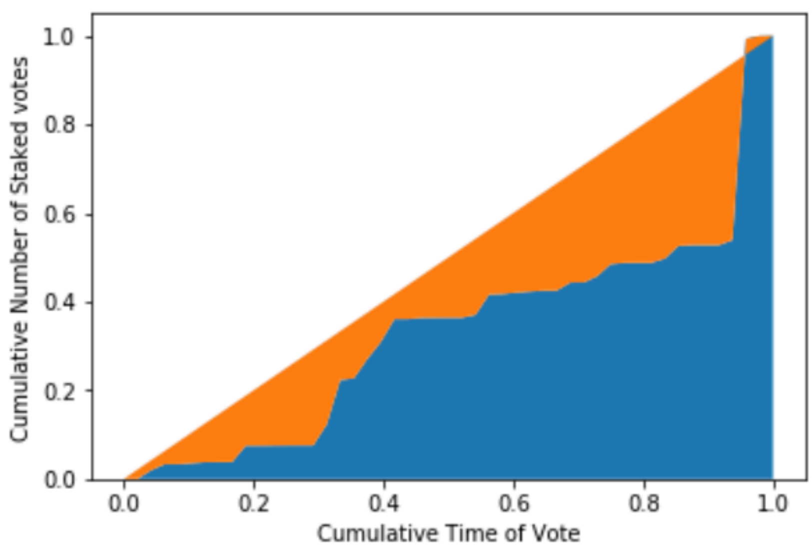
AGP 8

Gini: 0.170, Max. Gini: 0.170, Normalized Gini: 1.000



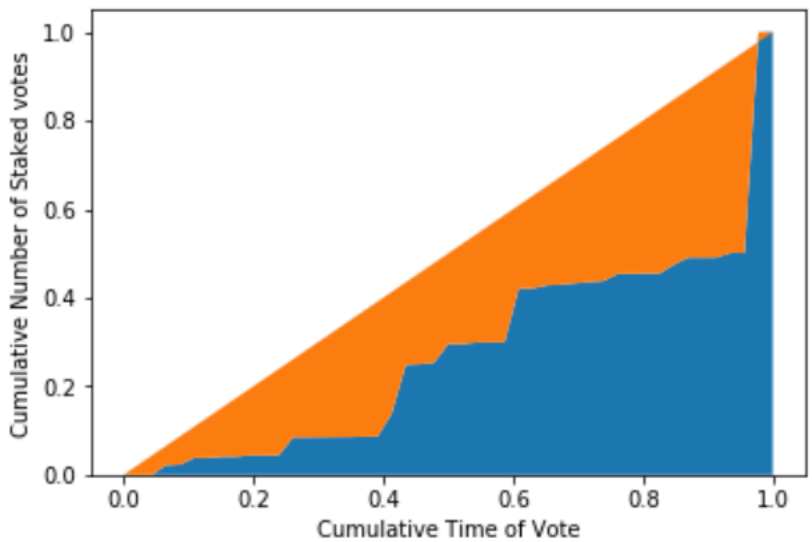
AGP 9

Gini: 0.232, Max. Gini: 0.232, Normalized Gini: 1.000



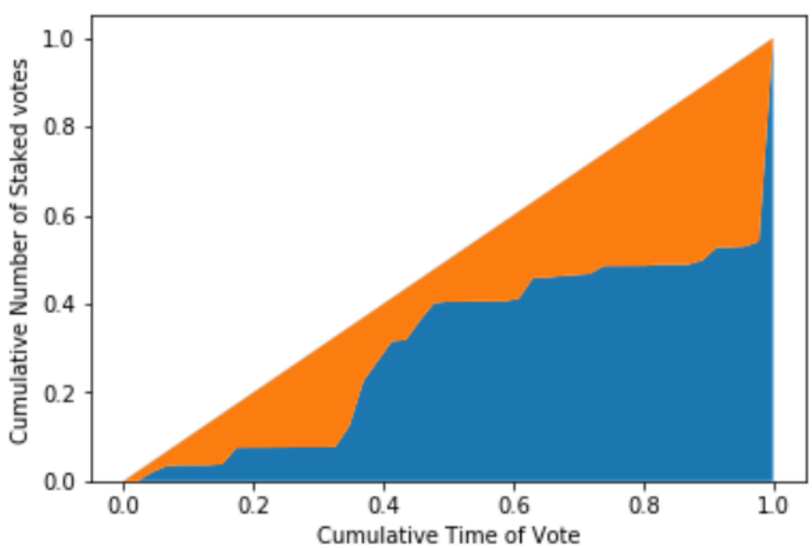
AGP 10

Gini: 0.227, Max. Gini: 0.227, Normalized Gini: 1.000



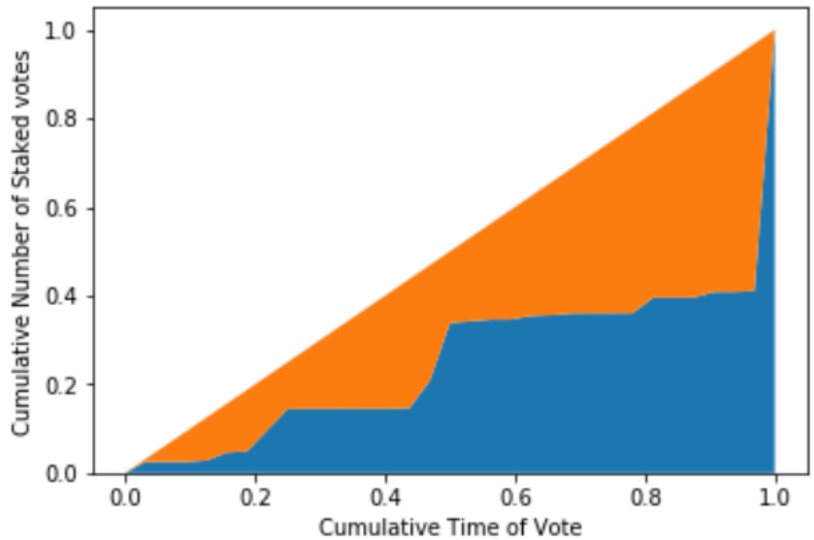
AGP 11

`Gini: 0.193, Max. Gini: 0.193, Normalized Gini: 1.000`



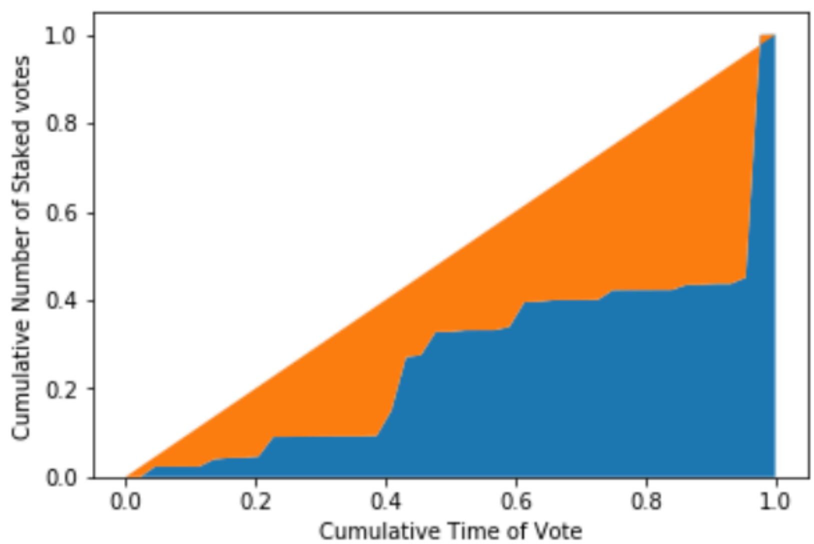
AGP 12

`Gini: 0.251, Max. Gini: 0.251, Normalized Gini: 1.000`



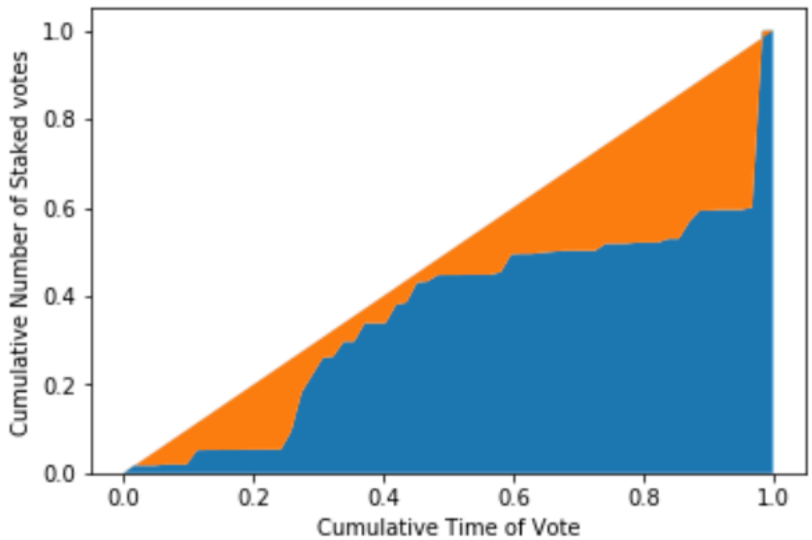
AGP 13

Gini: 0.230, Max. Gini: 0.230, Normalized Gini: 1.000



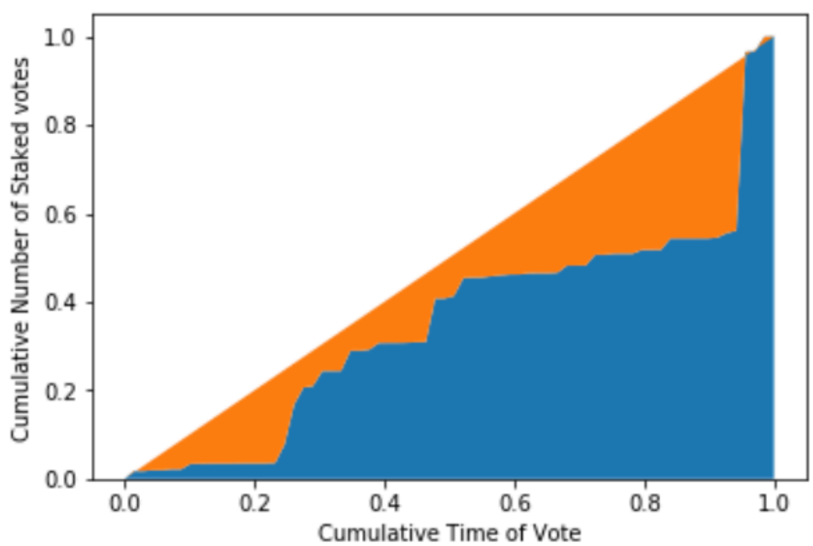
AGP 14

Gini: 0.140, Max. Gini: 0.140, Normalized Gini: 1.000



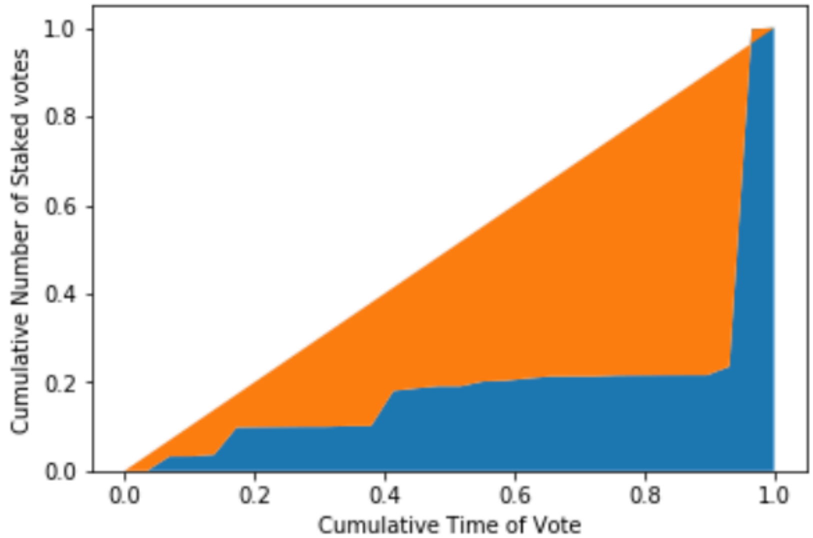
AGP 15

`Gini: 0.149, Max. Gini: 0.149, Normalized Gini: 1.000`

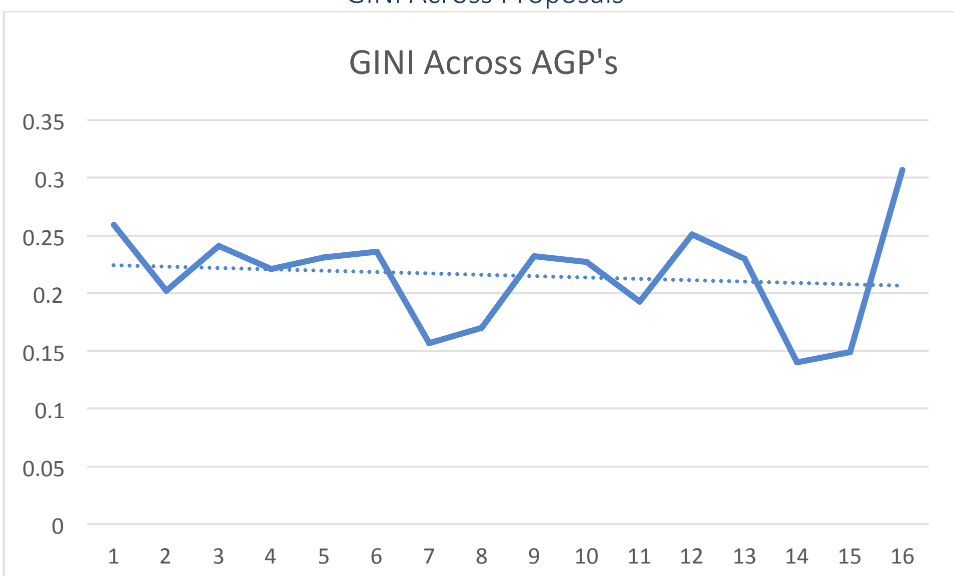


AGP 16

`Gini: 0.307, Max. Gini: 0.307, Normalized Gini: 1.000`



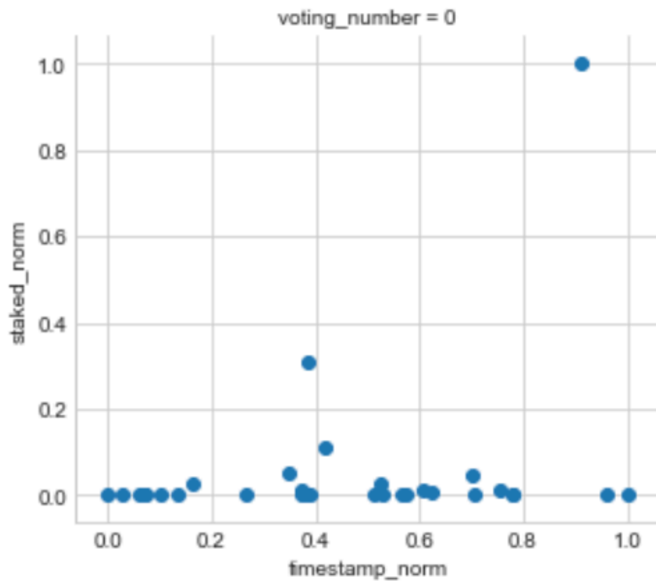
GINI Across Proposals



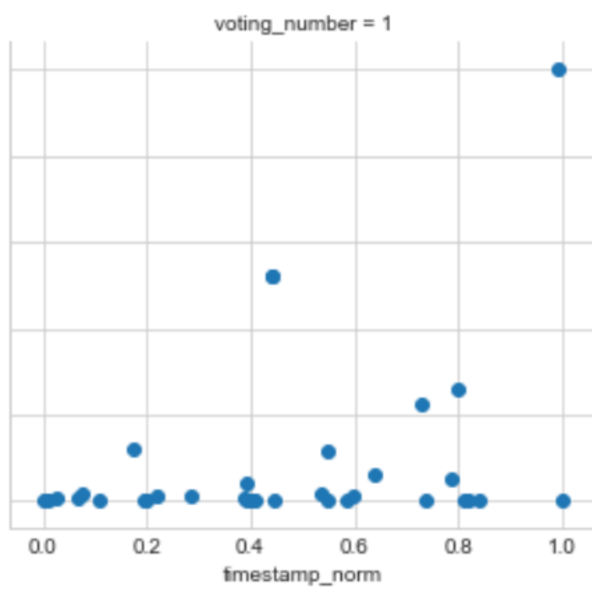
CLUSTERING

Data Grouped by voting_number

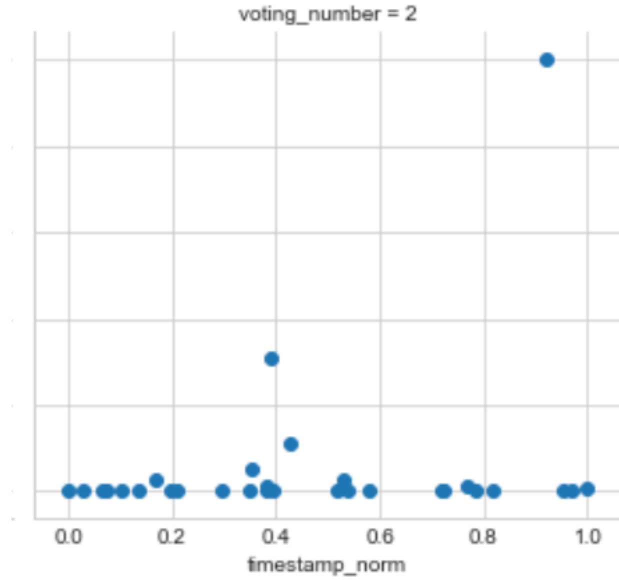
AGP 1



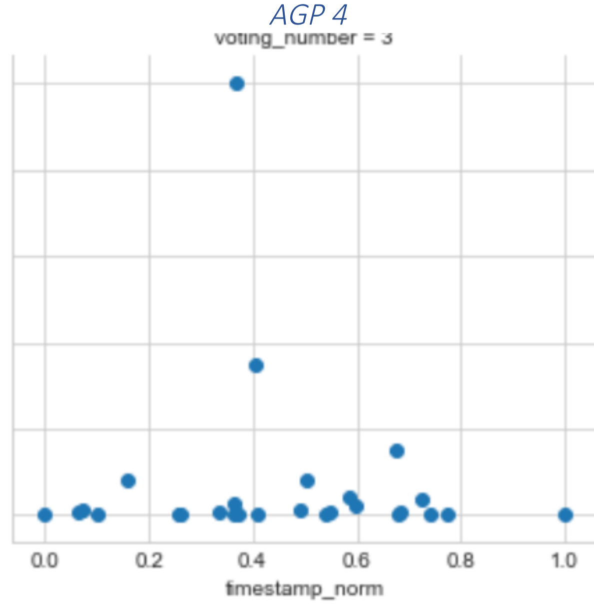
AGP 2



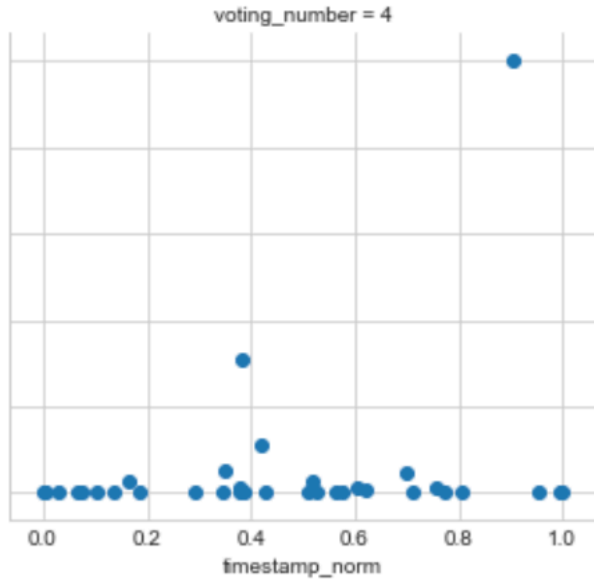
AGP 3


voting_number = 2
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

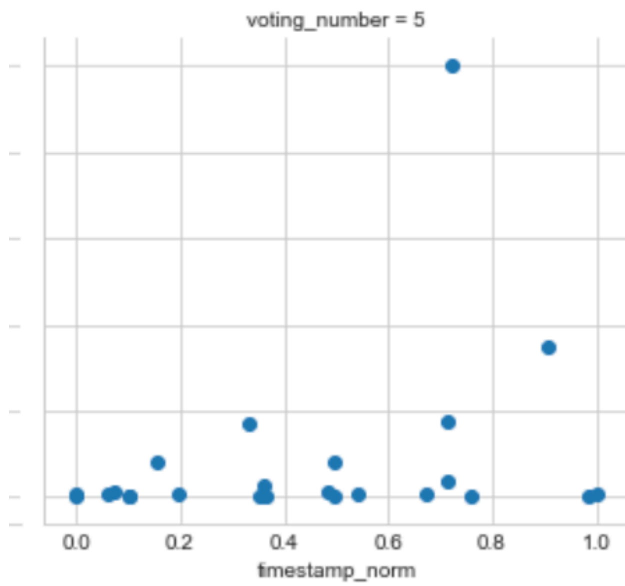
AGP 4


voting_number = 3
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

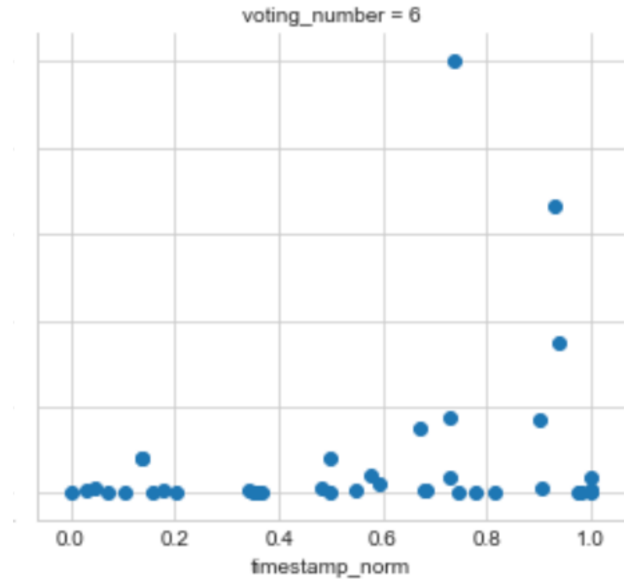
AGP 5


voting_number = 4
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

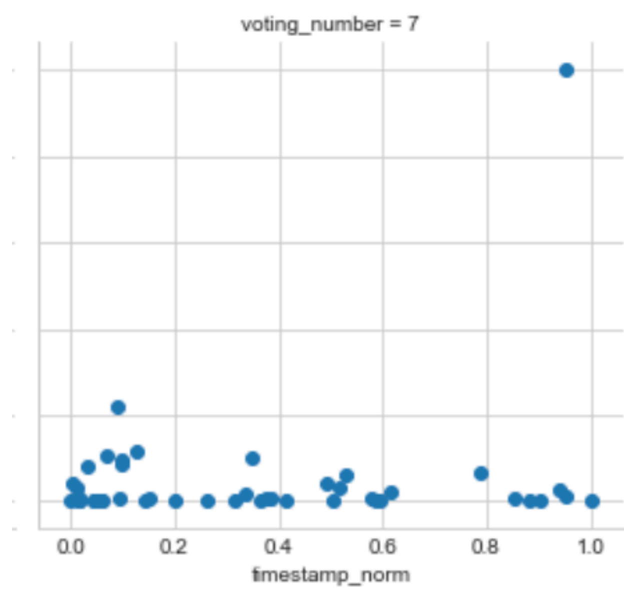
AGP 6


voting_number = 5
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

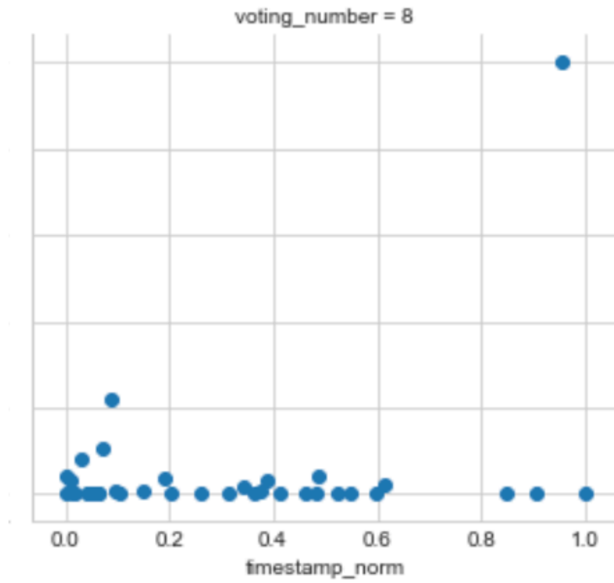
AGP 7


voting_number = 6
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

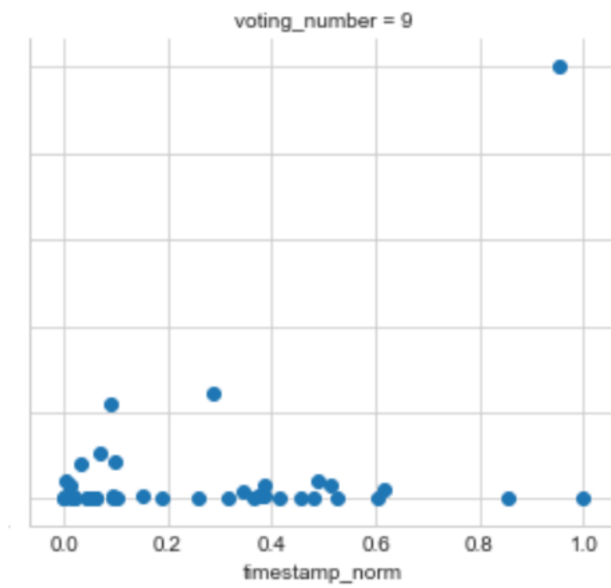
AGP 8


voting_number = 7
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

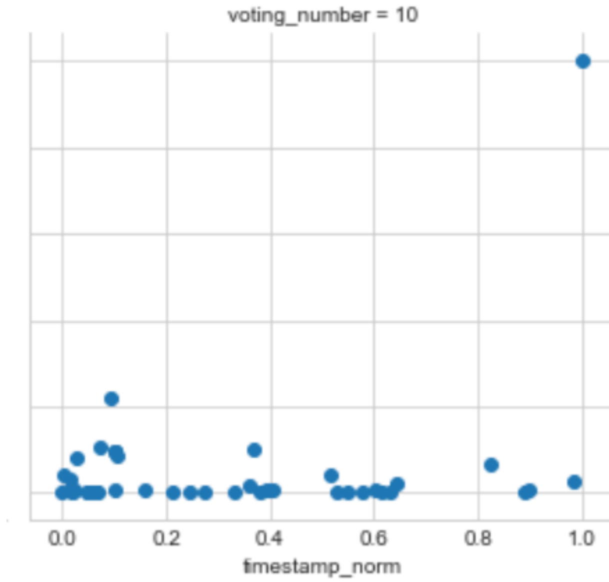
AGP 9


voting_number = 8
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

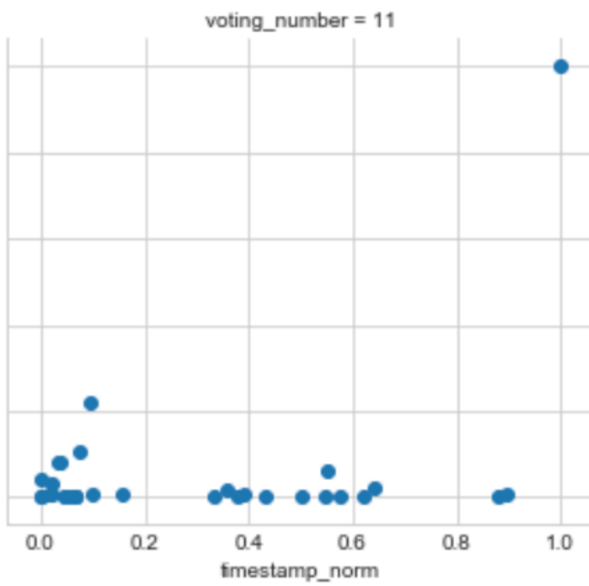
AGP 10


voting_number = 9
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

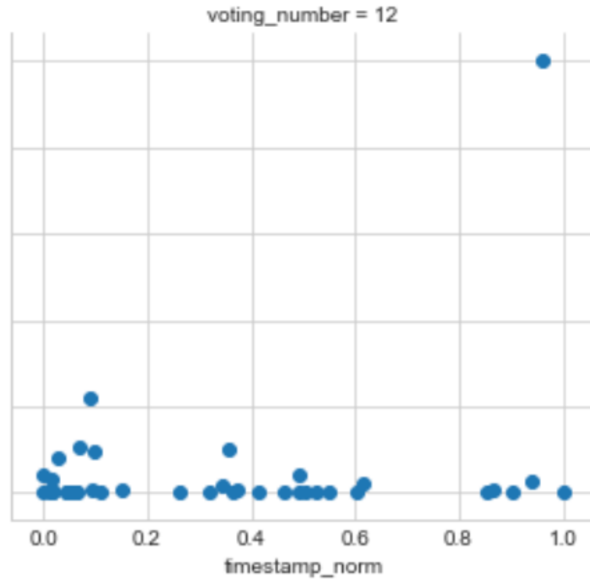
AGP 11


voting_number = 10
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

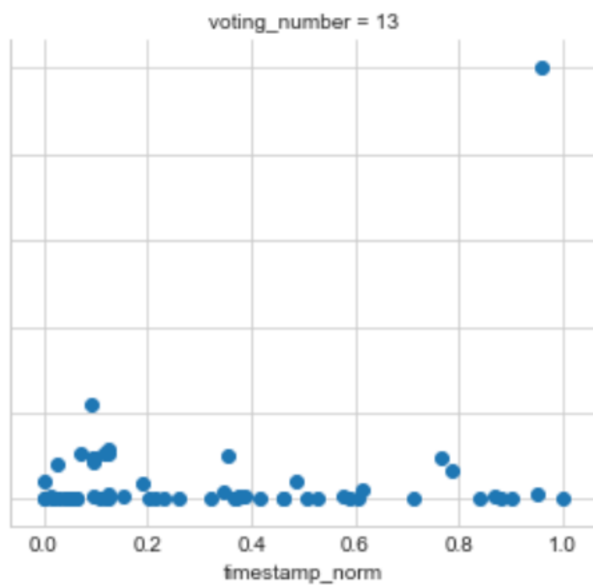
AGP 12


voting_number = 11
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

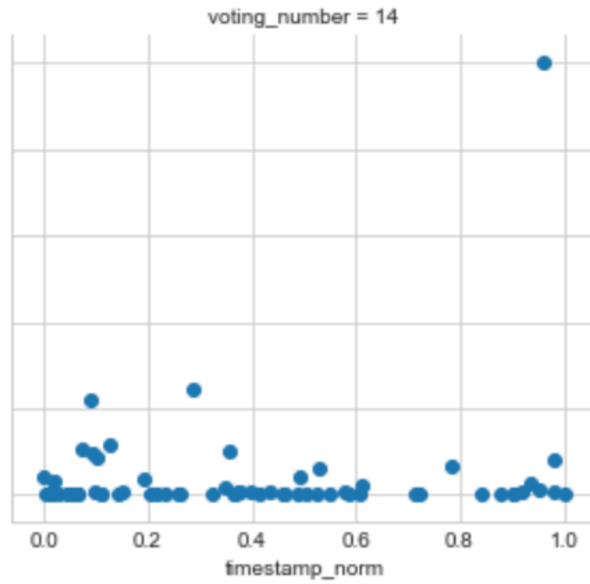
AGP 13


voting_number = 12
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

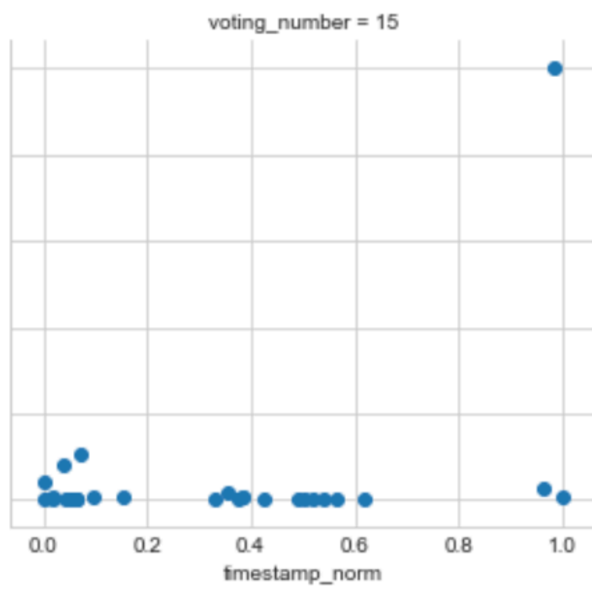
AGP 14


voting_number = 13
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

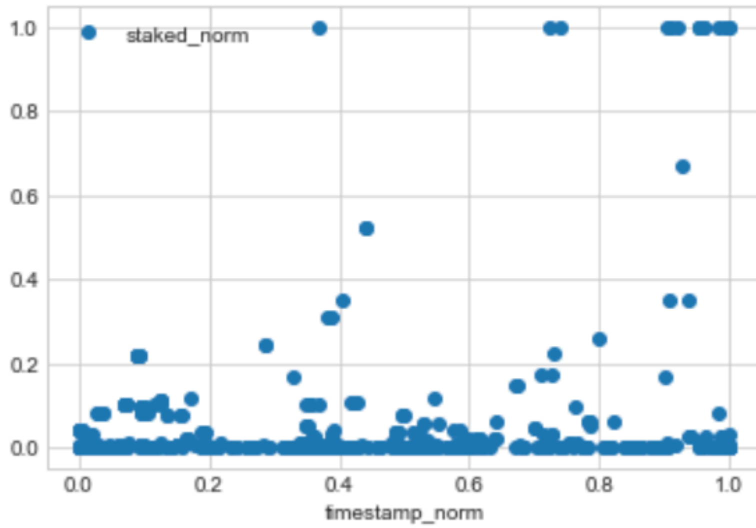
AGP 15


voting_number = 14
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

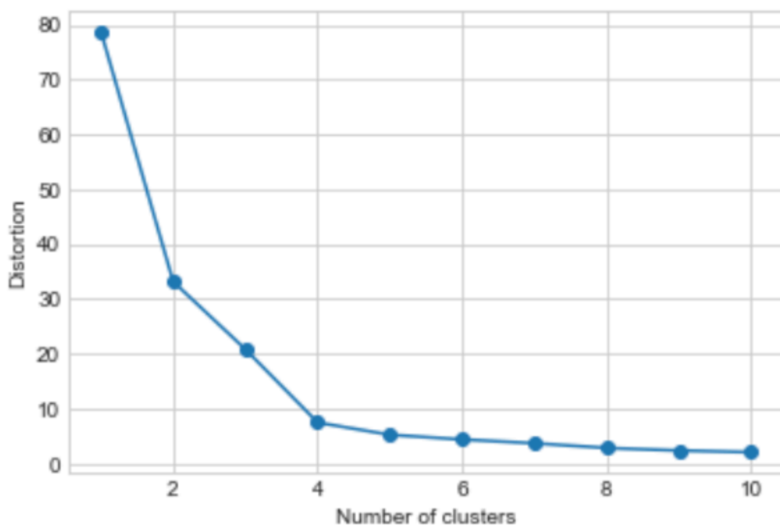
AGP 16


voting_number = 15
0.0
0.2
0.4
0.6
0.8
1.0
timestamp_norm

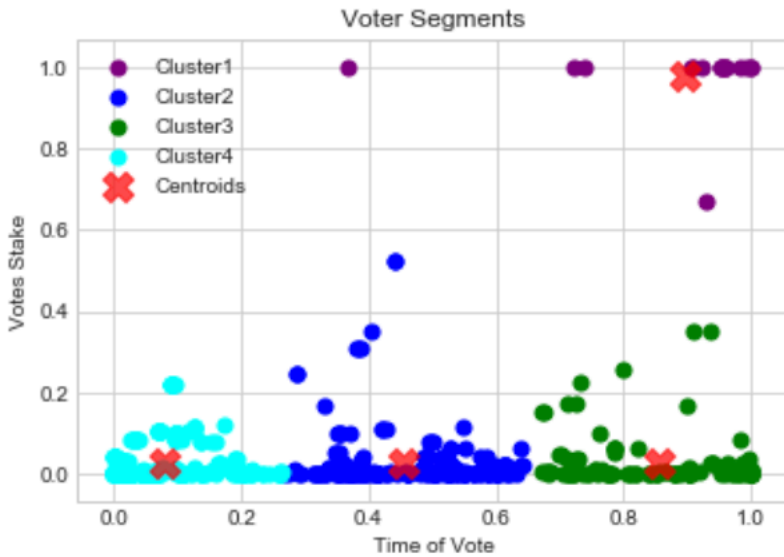
Across ALL AGPs



Selecting K



Clusters



LOGISTIC REGRESSION

MODEL 1

DATA

```
df1.head()
```

	Result	timestamp_norm	staked_norm
0	1	0.000000	0.000131
1	1	0.028919	0.000790
2	1	0.059519	0.001561
3	1	0.069232	0.002929
4	1	0.069483	0.002929

```
#printing confusion matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[19 11]
 [41 57]]
```

```
#accuracy
reg_all.score(X_test,y_test)
```

```
0.59375
```

```
#printing confusion matrix
print(confusion_matrix(y_test,y_pred))
```

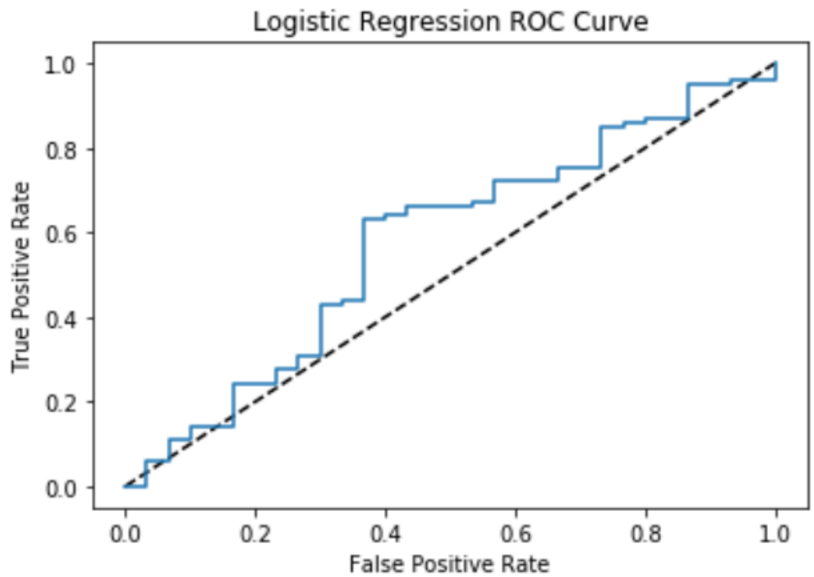
```
[[19 11]
 [41 57]]
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.32      0.63      0.42        30
           1       0.84      0.58      0.69        98

   micro avg       0.59      0.59      0.59       128
   macro avg       0.58      0.61      0.55       128
weighted avg       0.72      0.59      0.62       128
```




```
roc_auc_score(y_test, y_pred_prob)
```

```
0.5758503401360544
```

```
Accuracy of Model with Cross Validation is: 55.456140350877206
```

MODEL 2

Data

```
df2.head()
```

	Result	cluster	timestamp_norm	staked_norm
0	1	3	0.000000	0.000131
1	1	3	0.028919	0.000790
2	1	3	0.059519	0.001561
3	1	3	0.069232	0.002929
4	1	3	0.069483	0.002929

Measuring Performance

```
#accuracy
reg2_all.score(X2_test,y2_test)
```

```
0.6328125
```

```
#printing confusion matrix
print(confusion_matrix(y2_test,y2_pred))
```

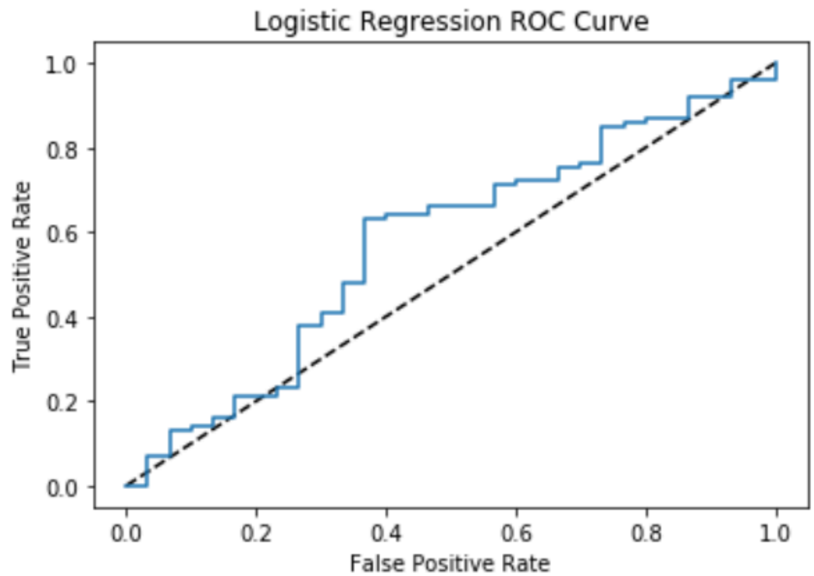
```
[[18 12]
 [35 63]]
```

```
print(classification_report(y2_test,y2_pred))
```

```
              precision    recall  f1-score   support

           0       0.34      0.60      0.43        30
           1       0.84      0.64      0.73        98

   micro avg       0.63      0.63      0.63       128
   macro avg       0.59      0.62      0.58       128
weighted avg       0.72      0.63      0.66       128
```




```
roc_auc_score(y2_test, y2_pred_prob)
```

```
0.5741496598639456
```

```
Accuracy of Model with Cross Validation is: 55.666666666666664
```

MODEL 3

Data

```
df3.head()
```

	vote	Result	cluster	timestamp_norm	staked_norm
0	1	1	3	0.000000	0.000131
1	1	1	3	0.028919	0.000790
2	1	1	3	0.059519	0.001561
3	1	1	3	0.069232	0.002929
4	1	1	3	0.069483	0.002929

Measuring Performance

```
#accuracy
reg3_all.score(X3_test,y3_test)
```

```
0.8046875
```

```
#printing confusion matrix
print(confusion_matrix(y3_test,y3_pred))
```

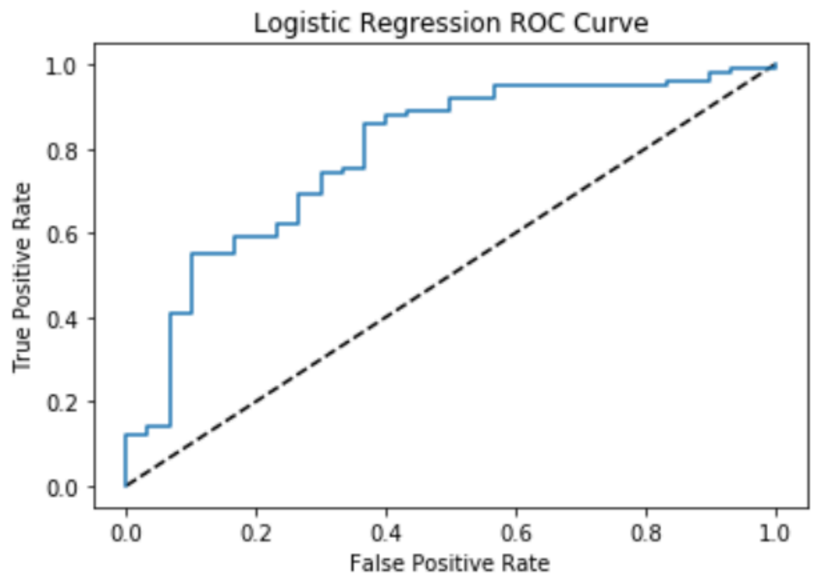
```
[[19 11]
 [14 84]]
```

```
print(classification_report(y3_test,y3_pred))
```

```
              precision    recall  f1-score   support

           0       0.58      0.63      0.60        30
           1       0.88      0.86      0.87        98

   micro avg       0.80      0.80      0.80       128
   macro avg       0.73      0.75      0.74       128
weighted avg       0.81      0.80      0.81       128
```




```
roc_auc_score(y3_test, y3_pred_prob)
```

0.7863945578231292

Accuracy of Model with Cross Validation is: 79.26315789473685

Supplementary Materials

Market Design Lens

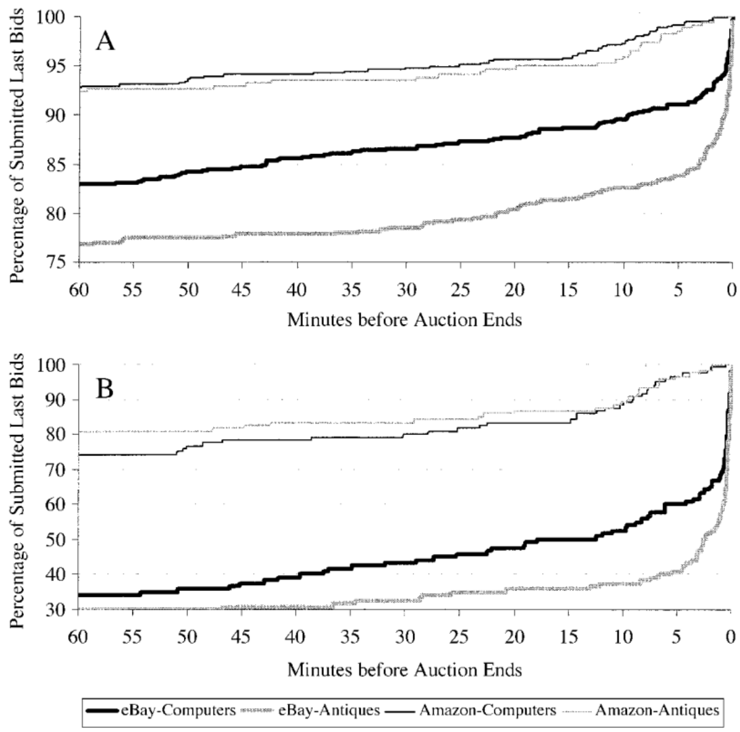
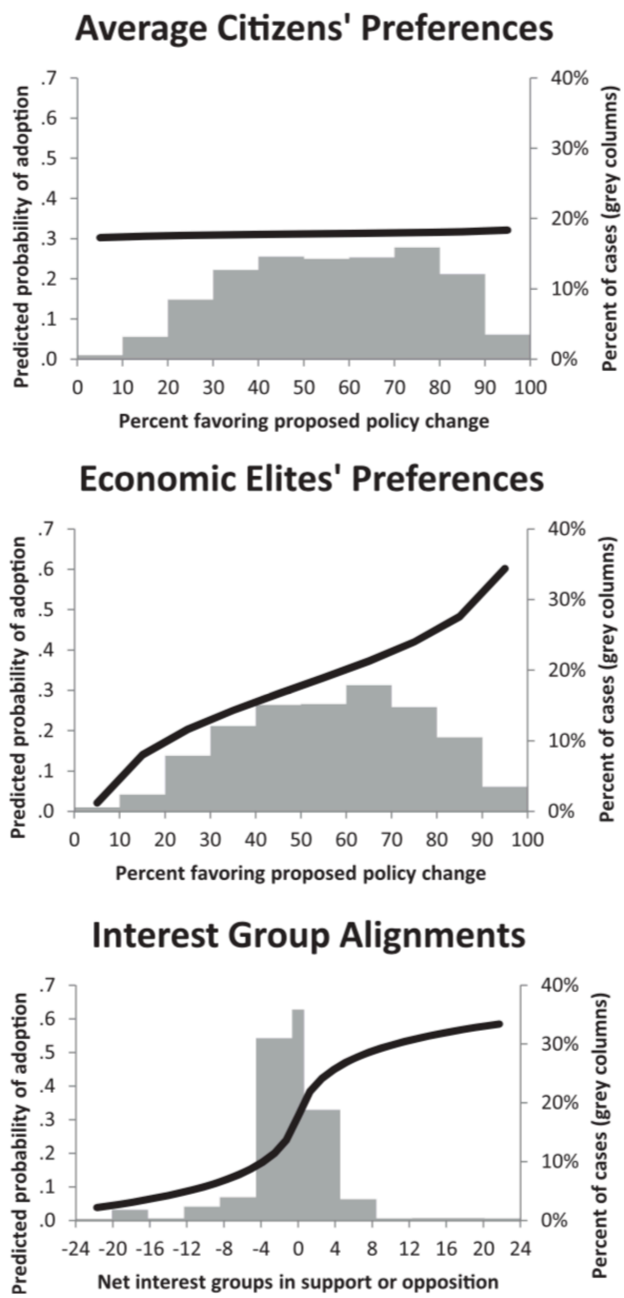


FIGURE 1. CUMULATIVE DISTRIBUTIONS OVER TIME OF (A) BIDDERS' LAST BIDS AND (B) AUCTIONS' LAST BIDS [27]

[27] Alvin E. Roth & Axel Ockenfels, 2002. "Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet," *American Economic Review,* vol. 92(4), 1093-1103.

Political Lens

Figure 1
Predicted probability of policy adoption (dark lines, left axes) by policy disposition; the distribution of preferences (gray columns, right axes)



[28]

[28] Martin Gilens and Benjamin I Page, Testing Theories of American Politics: Elites, Interest Groups, and Average Citizens, Princeton University Press 2012