NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing, and Cyber Systems

Technology Feasibility

Date: March 22nd, 2024
Team Name: FairyMander
Sponsor: Bridget Bero
Professor: Michael Leverington
Mentor: Vahid Nikoonejad Fard

Team Members:
Izaac Molina (Team Lead)
Dylan Franco
Jeysen Angous
Sophia Ingram
Ceanna Jarrett

Table of Contents

1 Introduction

1.1 Big Problem

Every ten years congressional lines are redrawn to create new voting districts. These districts must comply with state laws and represent the population. People then vote for the member of Congress that they want representing their district. This is an important decision since members of Congress revise and create laws. However, a problem that can arise when redistricting is gerrymandering. Gerrymandering is the act of manipulating district lines in a way that benefits certain political parties. Representatives are often in charge of creating districts, giving them the power to manipulate electoral outcomes. This power poses a threat to our democracy, potentially making voting obsolete. Gerrymandering is done using two popular strategies: packing and cracking. Packing packs like-minded voters into one district, giving them as few districts as possible, whereas cracking splits like-minded voters into different districts, thus diluting their votes. These strategies give a particular political party a district majority, giving them the upper hand. This can also be used to silence minority groups, preventing them from having political representation. Ultimately, gerrymandering takes voting power away from US citizens by predetermining political outcomes.

1.2 Current Solutions

Our sponsor, Dr. Bridget Bero, is a professor at NAU working for the Department of Civil Engineering, Construction Management, and Environmental Engineering. During her 30-year tenure, she has also earned a position as Associate Chair for Civil-Environmental Engineering Programs at NAU. Through her work, she has seen how corrupt the government can be, as politicians often make decisions that go against the will of US citizens. While working in the field, Dr. Bero witnessed the defunding of the Environmental Protection Act, an important act that preserves natural resources and protects human health. Witnessing this injustice directed her attention toward how voting is conducted, leading to the discovery of more government corruption.

There are active solutions in place for gerrymandering, but they are not as widespread as they should be. One solution against gerrymandering is for voting districts to be drawn by the Independent Redistricting Commission (IRC). The IRC is separate from the legislature, meaning that politicians do not have the opportunity to manipulate voting districts. This makes the redistricting process fair and transparent. The IRC must adhere to certain criteria when creating districts, such as equal population, protecting racial minorities, and making districts compact. However, only eight state districts are currently being drawn using this system. This means the other 42 states' voters are being gerrymandered and suppressed. In search of a better, more widespread, solution Dr. Bero reached out to us to create an open-source, non-biased redistricting algorithm.

1.3 Solution Vision

Our proposed solution is to create an unbiased redistricting algorithm that fits Dr. Bero's needs by utilizing census data and geographical information systems. We plan to make our algorithm easily accessible and user-friendly by creating a website. Users will select which state they would like to see "Fairymandered", providing a visual representation of the proposed voting districts. We will prevent packing and cracking by creating districts that are compact, contiguous, and have equal populations. For our algorithm to be fair, different social and economic communities will also be considered. We want to ensure that everyone's votes have an impact, allowing for districts that reflect every community's needs. Our goal is to combat unfairness in voting, ensuring that votes have an impact, and everyone is politically represented.

2 Technological Challenges

To form our envisioned solution, we will need to consider several problems involved in both creating and visualizing a redistricting algorithm. In this section, we describe the challenges we see being most relevant to the development of our solution.

2.1 Drawing State Maps

- Properly drawing an accurate representation of all 50 states to represent our new districts, as well as all cities/counties/geographical divides.
- Populating these representations with any information related to why/what/how the districts are being drawn.

2.2 Making an Interactive Map/UI

- Create an interactive map/UI to work on cross-platform compatibility, mainly focusing on the variety of screen sizes.
- Being able to use coordinates (absolute and relative) to accurately select a specific state.
- Using a proper framework to create an interactive map/UI.

2.3 Handling Geographic and Demographic Data

- Create an unbiased algorithm to make contiguous, compact districts with equal populations.
- We will need our algorithm to account for white space on the graph, which will represent geographical features where no one lives.
- Find a way to account for different demographics when creating districts.

2.4 Obtaining Data for Algorithm Computations

- Constructing the redistricting algorithm(s) requires quality data that accurately reflects the US population and demographics.
- This data will need to be up to date and reflect the data that current redistricting committees use to make decisions.
- We need to define how we will access this data.

2.5 Define what it means to be fair

- We need to define what it means to be fair when creating the algorithm.
- We need to decide if we want to define fairness through democratic fairness or authoritarian fairness.
- We need to define how we will evaluate the equality and fairness of our algorithm.

3 Technological Analysis

Next, we'll conduct a thorough evaluation of the existing strategies designed to tackle the challenges outlined previously. By establishing a comprehensive set of criteria for assessing each potential solution, we aim to determine the most suitable option for our needs. The table shown in each "Chosen Approach" section summarizes this analysis.

3.1 Drawing Lines on State Map

In order to visually display the data we generate from our algorithm, we need to be able to accurately represent maps of every district in the US. Doing this by hand in the time provided is out of the question, so we need to find a library that can produce the assets we require.

3.1.1 Desired Characteristics

Ideal characteristics for map drawing and analysis include:

- **User-friendly (5 points):** The library chosen should be user-friendly not only on the user's side but also for us. This would help minimize room for any calculational errors when putting our algorithm on a map.
- **Graphical output (5 points):** An existing graphical component would be ideal to help better represent the area being discussed.
- **GIS support (5 points):** Finally, the library preferably has some form of Geographic Information System (GIS) compatibility to allow for the integration of geological data down the line.

3.1.2 Alternatives

Shapely

Shapely is a Python library for manipulating and analyzing geometric objects.

Geopandas
Geopandas is an addition to the already popular Panda's library that adds the ability to integrate with other existing spatial analysis libraries.

Folium
A Python interface for creating interactive web-based geospatial maps.

3.1.3 Analysis

Shapely
While Shapely seems to be really good at precisely computing geometric information, it does not have a built-in visual component and relies on Pandas for graphics. It also does not seem great for larger datasets, which is primarily what we'll be working with.

Geopandas
Was incredibly easy to work with. We were able to integrate data from Shapely and not only continue to manipulate it but also output it to a visual map. Even better than that, Geopandas supports shapefiles, which are already full of GIS data and other information we can use to properly represent entire areas. The worst part about it would have to be that it's a bit slow.

Folium
It is similar to Geopandas, also very easy to use and able to integrate shapefile data. On top of this, it's also able to create interactive maps that can be embedded into a web page. Unlike Geopandas, though, the customization options for these maps are very limited.

3.1.4 Chosen Approach

Each of these options is exceptional in its own way, but only two of them truly have what we need to progress.

Figure 3.1.4 - Analysis of Map Visualization Modules

Characteristic Score	Shapely	Geopandas	Folium
User-friendly	5/5	5/5	5/5
Graphical output	0/5	5/5	4/5
GIS support	0/5	4/5	5/5
Total	5	14	14

3.1.5 Proving Feasibility

Because Geopandas and Folium are compatible with each other, and both allow for the use of shapefiles, we feel the best approach would be to use *both* options and leverage each of their strengths: Geopandas' exceptional map customization and Folium's web-based interactivity.

3.2 Making an Interactive Map/UI

Many factors go into creating an ideal interactive map/UI within a web application that will be visually appealing and user-friendly.

3.2.1 Desired Characteristics

Some characteristics or features that would be ideal and would stand out for an interactive map can include features such as:

- **Cross-platform compatibility (5 points):** The website needs to be responsive to all types of devices, including web browsers with the main focus being screen sizes.
- **User-friendliness (5 points):** We must ensure that the interactive map/UI will be enhanced for maximum user-friendliness. To do this, the website needs to have simple navigation, clear instructions on how everything works, and provide easy handling features for the user, such as proper clickable areas and even clear guidance.
- **Responsiveness (5 points):** We want to include proper hover responsiveness as it would stand out and make the website look much more professional. For example, when someone hovers over a state they can see the state name and can click on that certain state.

3.2.2 Alternatives

Use of SVG

Scalable Vector Graphics (SVG) is a web-friendly vector file format that displays two-dimensional graphics and charts on a website.

Dash Plotly

Dash is an open-source Python framework mainly used for building data visualization interfaces on web applications. Plotly, on the other hand, is a library used to create plots and charts when combined, you have the capability of creating a powerful interactive map by combining both capabilities.

Folium

Folium is a powerful Python library that creates several different types of maps. It is built on leaflet.js, an open-source Javascript library that can be used to create interactive maps along with additional features. Since folium happens to be interactive, the sharing capabilities between Python and HTML become uncomplicated.

3.2.3 Analysis

Use of SVG

We can use SVG to create the interactive map/UI by using SVG elements such as paths, markers, styles, data name, and data-id in which each state name with an acronym will be defined. SVG can also use an attribute called path data element which will be used to accurately select the relative and absolute coordinates of each state.

Dash Plotly

Since Dash Plotly is a Python framework for building interactive web applications, we can use the Plotly library to create lines, mainly district lines, using x and y coordinates as needed.

Folium

We can use Folium to draw district lines on the map that use both longitude and latitude coordinates. Furthermore, designing interactive maps with features like popups, heatmaps, markers, and much more is made simple by its ease of use.

3.2.4 Chosen Approach

Through conducting our research, we found solid alternatives that would work best in making an interactive map. Furthermore, while doing the necessary research, we found all the pros and cons of each alternative to decide if it was the right choice to use.

Figure 3.2.4 - Analysis of Web Frameworks for a Map UI

Characteristic Score	No Framework	Dash Plotly	Folium
Cross-platform compatibility	2/5	4/5	5/5
User-friendliness	2/5	4/5	5/5
Responsiveness	2/5	4/5	4/5
Total	6	13	14

Moreover, following the research conducted, we determined that Folium would be the best Python library to choose from as it received a score of fourteen based on our analysis. The score was given based on capabilities such as cross-platform compatibility, user-friendliness, and responsiveness that each framework/library could perform mainly focusing on the ease of use.

3.2.5 Proving Feasibility

To validate Folium as a visualization framework, we need to properly implement the library within Python and ensure that the implementation is done correctly. We will ensure that it can create responsive map interfaces that we can adapt to our other chosen methods.

3.3 Handling Geographic and Demographic Data

Our algorithm will work by representing people as coordinate points on a map, we will then use those coordinates to create compact and equally populated districts. We will start by finding the densest points on a graph, which will represent population centers. These centers, and their surrounding points, will make up a district. This strategy could cause our algorithm to run very slow since state populations are in the millions. We encounter another issue if we do not consider demographics (such as political party, race, etc) and geography. This could cause us to unknowingly crack or pack minorities and create unfair districts.

3.3.1 Desired Characteristics

There are a few methods that could be used to achieve our goal, but we have to find the best fit for our algorithm in particular. Here are the criteria our technology must meet to fit our algorithm's needs best:

- **Speed (5 points):** Since we will be dealing with populations that reach 38 million, we will need to use the fastest method for our algorithm.
- **Accuracy (5 points):** The algorithm must accurately find areas of high populations, create equally populated districts, and create the correct amount of districts. Geographical areas such as mountains and rivers are not densely populated, our algorithm must consider this when creating districts.
- **Fair (5 points):** Our algorithm needs to be fair by giving underrepresented groups a voice in our government without packing or cracking them. To do this, we need to incorporate demographic data in order to accurately identify populations with similar demographics.
- **Accessibility (5 points):** The software we choose to use for our algorithm must be easily accessible.

3.3.2 Alternatives

Density Peaks Clustering

Density peaks clustering, or DPC, is an algorithm to create clusters and identify cluster centers based on density and distance. DPC is commonly used for data mining in fields such as astronomy and bioinformatics. This algorithm was developed by Alex Rodriguez and Alessandro Laio in 2014. The advantages of DPC is that it is able to detect arbitrarily shaped clusters, it is easy to use and adjust, and it works well for outlier detection.

ArcGIS API for Python

ArcGIS API is a Python library that allows for the use of ArcGIS tools. ArcGIS is a geographical mapping and analysis tool, released in 1999 by Esri. It is used by industries such as law enforcement, government agencies, development companies, transportation, etc. With uses such as disease tracking, crime reports, development planning, and traffic reports. ArcGIS is a popular tool due to its ability to map states and cluster populations, allowing for visualization and customization.

Geospatial Clustering

Geospatial clustering is an algorithm that identifies groups of similar demographics and clusters them. It takes geographical features (rivers, mountains, etc.) into account. This clustering method ensures that clusters are spatial and are made up of objects with high degrees of similarity. Spatial clustering algorithms have been around since the 90s and are often used to identify patterns in urban planning, transportation, and more. Python offers a variety of libraries to implement geospatial clustering; offering customizable options for the programmer.

3.3.3 Analysis

Density Peaks Clustering

Our analysis on density peaks clustering was conducted through testing. It was evaluated for the speed criteria by being tested with 5.3 million data points. It can handle this large scale of coordinates through the use of an R-Tree, but it could potentially slow down with 38 million points. The algorithm is very accurate, finding all the highest density points and marking them as a center. Points within a center's user-specified radius are added to a cluster. DPC does not pass the fairness test since it does not take demographics into account. Modifications would need to be made for it to meet our algorithm's requirements.

ArcGIS API for Python

Through our research, we found that ArcGIS accepts a shape file, which we will be getting from the census website, and creates clusters based on user-specified constraints (demographics, number of clusters, etc). ArcGIS features different clustering algorithms, the one that meets our requirements is spatial k means clustering. This clustering algorithm creates a user-specified number of contiguous clusters. Once clustering has been completed the data will be translated into a shapefile. We can then enter the shapefile into GeoPandas and fine-tune our provided clusters from there. ArcGIS meets all of our algorithm's requirements: it is fast, accurate, and fair. The ArcGIS software is easily accessible through Python API; however, it requires a paid subscription. This forces the sponsor, school, or us to continue buying the subscription for our algorithm to work.

Geospatial Clustering

Our analysis on geospatial clustering took place through research and testing. This method of clustering meets all of our algorithm's requirements. Through our analysis, we found that mountains and rivers do not affect the creation of clusters. Although the population in these areas are small, they are easily identified using geospatial clustering. We also found that geospatial clustering allows us to specify cluster requirements, such as demographics. Through the use of specifications, we will be able to ensure that our algorithm is fair. Implementing geospatial clustering will be quick and easy due to the wide availability of Python libraries.

3.3.4 Chosen Approach

Figure 3.3.4 - Analysis of Methods for Handling Geographic and Demographic Data

Characteristic Score	Density Peaks Clustering	ArcGIS	Geospatial Clustering
Speed	4/5	5/5	5/5
Accuracy	5/5	5/5	5/5
Fairness	3/5	5/5	5/5
Accessibility	5/5	4/5	5/5
Total	17	19	20

Density peaks clustering accurately creates clusters based on the highest density points. However, demographics are not taken into account, and clustering is slow when large amounts of coordinates are used. ArcGIS meets all of our requirements, except accessibility. Many of the tools offered by ArcGIS can only be accessed through a paid subscription, forcing us to look at other alternatives. Geospatial clustering has proved to be our best alternative by meeting all of our algorithm's requirements.

3.3.5 Proving Feasibility

The use of geospatial clustering for our project needs to be validated and tested further. In order to do this, we will import and test available geospatial clustering libraries in Python. Each of the library's clustering functions will be tested on the same population and demographic data of a chosen state. All libraries will be put through a requirements test with the chosen library being the fastest, most accurate, fairest, and easiest one to use.

3.4 Obtaining Data for Algorithm Computations

To create an effective voting district algorithm, it is important to consider where we will acquire our data. Population and geographic data will need to be obtained to draw district maps, and various data fields, such as demographic and electoral data, will assist in evaluating our algorithm's effectiveness. We want to ensure that this data is not only able to be collected but is also collected from reputable and reliable sources. Additionally, this data must be easy to access and work with so that it may be seamlessly integrated into our application.

3.4.1 Desired Characteristics

Numerous data sources exist that could potentially serve our needs. To pick a source best suited for our redistricting algorithm, the following criteria describe aspects of the source that make it best suited for use in FairyMander.

- **Relevancy (5 points):** The method must adequately obtain data relevant to the redistricting algorithm. This primarily includes data on raw population values, as well as demographic information for interest groups, political affiliations, and racial distributions. Obtaining such data is necessary for ensuring the redistricting algorithm remains fair and accounts for specific voting populations.
- **Ease of Use (5 points):** Due to the sheer volume of data we will need to use, the data collected must be as easy to work with as possible. If similar data can be collected from different methods, but one method offers a simpler interface or less data "rummaging" to find what is needed, then that method is more favorable.
- **Up to Date (5 points):** To ensure the redistricting lines reflect the current political conditions, the obtained data needs to be as up-to-date as possible. Political interest groups can change fairly quickly over the course of just a few years, so our algorithm must use the most current information available.
- **Reputability (5 points):** We want to ensure our data is being obtained from a reputable source. Our goal is to create an algorithm that could be used in a real redistricting practice, so it is absolutely essential that our data reflects real-world conditions as accurately as possible. A data source that has a long history of use in other applications or is otherwise well respected as a credible source is more favorable for this reason.

3.4.2 Alternatives

The US Census

The US Census is a longstanding system that collects demographic information with the goal of providing a comprehensive data source for a multitude of purposes. The census website also provides numerous SHAPE files, which can be used to analyze US geographic data such as state, county, and census block borders. The census also has an Advanced Programming Interface (API) that can be used to query data from it within a program.

VEST

The Voting and Election Science Team (VEST), operating out of the University of Florida, has assembled a dataset of election data based on voting precincts. They provide data for both the general and midterm elections, providing the number of votes each candidate in the presidential, congressional, and state elections received. This data is provided in a series of downloadable files from the VEST website.

Redistricting Data Hub
As part of the Fair Representation in Redistricting Initiative, a nonpartisan fund with the goal of ending gerrymandering, the Redistricting Data Hub (RDH) provides numerous datasets that are designed to aid in the redistricting process. This includes census and electoral data, as well as reports the RDH has created related to current redistricting plans. The RDH also has an API for acquiring data through it, which is primarily configured to work in Python.

3.4.3 Analysis

The US Census
The census contains comprehensive data related to population and demographics, which is critical information for both drawing and evaluating district maps. It also includes geographic data through SHAPE files, which is exceedingly relevant in visually representing drawn districts, as well as determining adjacent geographic units. Beyond this, the US census is the most up-to-date data source available for demographic information. No other data sources are as robust and up-to-date, as the census has a well-established collection procedure enabled by the US government. However, the census does not provide electoral data which is a significant hole as such data will likely be necessary to ensure the nonpartisanship of our algorithm. Additionally, while the census contains data appropriate for redistricting, it also contains fairly irrelevant information such as housing unit data, marital status, and fertility data. The census API can also be difficult to work with as it requires long queries with many different options that can convolute the data collection process.

VEST
VEST provides electoral data, which is of particular interest as this data is not available from the census. However, it does not provide population and demographic data. The current VEST datasets are all updated appropriately, as its most recent data sets are from the 2022 general election. They have maintained this since 2000, so it is likely that the team will continue to update the dataset accordingly. Additionally, the head faculty behind VEST, Dr. Michael P. McDonald, is a well-established expert in political science, contributing to the source's credibility. While the data from VEST is very relevant to our algorithm, it suffers from a lack of comprehensiveness in relation to population and geographic data. VEST also offers no API to work with its data files, meaning we will have to develop our own interface for retrieving the data. This is a significant detriment to VEST's ease of use, as it requires the development of a more complex system to obtain desired data.

Redistricting Data Hub

The RDH contains a comprehensive dataset for redistricting purposes, including population, geographic, and electoral data. This data is pulled from various sources such as the census, VEST, and various state government departments up to 2022, with 2023 data currently in progress. It also contains accessory information that could prove to be useful for research purposes, such as reports related to the fairness of current districts, as well as data on public testimonies related to gerrymandering. Since the RDH is specifically targeted at providing redistricting data, it excludes fields unlikely to be relevant, such as those described in the section on the census' ease of use. This makes the data much easier to work with, as data queried with the API requires far less parsing in comparison to the pure census data. The RDH website also has numerous helpful video tutorials for using its tools. Aside from this, The RDH also pulls much of its data from reputable sources, as mentioned previously, making it a quality source in its own right. It has been used for news coverage on the redistricting process, assisting organizations dedicated to stopping gerrymandering, and official submissions to redistricting bodies in various states. Overall, the RDH is highly credible in that it is used in multiple redistricting scenarios and pulls from well-established data sources. However, it is worth mentioning that due to the myriad of sources the RDH pulls from, it is harder to evaluate its overall reputability, making it certainly less reputable than a well-established institution such as the census.

3.4.4 Chosen Approach

Figure 3.4.4 - Analysis of Redistricting Data Sources

Characteristic Score	US Census	VEST	Redistricting Data Hub
Relevancy	4/5	2/5	5/5
Ease of Use	4/5	3/5	5/5
Up to Date	5/5	5/5	4/5
Reputability	5/5	4/5	4/5
Total	18	14	18

Ultimately, the US Census and Redistricting Data Hub tied in our analysis. The census is very up-to-date and is a highly reputable source, but the RDH provides a dataset better fitted to redistricting needs and is easier to work with. This is a case where trade-offs will be made regardless of data choice, but ultimately, the comprehensiveness and usability of the RDH make it the most appealing option. The census will still be considered as a data source if the RDH does not include all of the census data we find appropriate to include. If this scenario does occur, we will combine the RDH and the census for our data needs.

3.4.5 Proving Feasibility

To prove that the RDH will be sufficient in obtaining the data needed, we will learn how to use the RDH to obtain the geographic, demographic, and electoral data needed for our application. We will then ensure that said data is compatible and translatable into a visual representation, as discussed in section 3.1.

3.5 Define what it means to be fair

When creating an algorithm that is unbiased, we must look at exactly what it means to be unbiased. We must consider how the data will be used in determining fair redistricting criteria. This algorithm must uphold what it means to be fair, but in order to do so, we must define what it means to be fair.

3.5.1 Desired Characteristics

When one is determining what is fair, they must first define fairness itself. Fairness must be broken down into requirements, which will lead to an unbiased way to look at redistricting. These requirements have been explicitly stated by our sponsor, and are described below:

- **Equal Population (5 points):** Each district must have an equal population that is representative of the people living in that district.
- **No Discrimination (5 points):** There must be adequate representation for minority populations within a given state map.
- **Unbiased (5 points):** The algorithm should remain impartial by considering multiple sides of an issue, ensuring that nonpartisanship is achieved.

3.5.2 Alternatives

Process of Fairness

The process of fairness is broken down into three principles, engagement, explanation, and clarification. Engagement means to make sure that all sides are spoken for in the agreement. When it comes to this project, each side will be represented by ensuring no party is given an unfair advantage. Every voter has the right to be brought into the process, providing meaningful engagement with our democracy. The explanation part of the process is the reasoning for the decision to be made. This decision will be made through requirements that each state must meet in the algorithm. The last principle of the process is clarification, where we will express that our reasoning is fair. The map of the district lines that we create will be the clarification.

Equity vs. Equality

Fairness is an abstract idea; it has different meanings depending on the situation. Some definitions of fairness have an emphasis on equality. However, adapting this view for our algorithm would force us to take power from groups. Forcing people to sacrifice their power and resources to give to other groups would be unfair. This caused us to come to the conclusion that fairness does not always equal equality. It would be unfair for us to impose our biases on which demographics need to sacrifice power in order to provide it to another set of demographics. In order to be fair, we will not be able to fully enforce equality in our algorithm.

3.5.3 Analysis

Process of Fairness

For the process of fairness, each person will be accounted for in every district, giving them fair representation in the electoral process. Additionally, minority groups must be accurately represented to ensure that these populations are given a proper voice. For the process of fairness, we will make sure that we do not stray from the requirements of what it means to be fair, ensuring no bias in the algorithm.

Equity vs. Equality

Equality means that resources are completely equal. However, suppose two people are buying a cake that costs ten dollars, and one person pays six dollars. The fair method would be that the person who paid six gets sixty percent of the cake, and the other person gets forty. It is fair to split the cake that way because that is how they paid for it. This example shows how being equal and being fair are two different things. Basing our algorithm on equality would produce gerrymandered districts. In order to adhere to equality, we would need to take and redistribute power between groups, therefore implementing our own biases.

3.5.4 Chosen Approach

Figure 3.4.4 - Analysis of Fairness Metrics

Characteristic Score	Process of Fairness	Equity vs. Equality
Equal Population	5/5	3/5
No Discrimination	5/5	3/5
Unbiased	5/5	3/5
Total	15	9

At the end of the analysis, the Process of Fairness came out on top. The process of fairness hits each requirement for our evaluation of fairness. It ensures adherence to all the requirements and ensures that each person understands why the districts are drawn this way. The equity vs equality fell short of what we need for the algorithm. We can not have an algorithm that is based upon wealth. An algorithm based on wealth is a form of discrimination that would ultimately lead to bias.

3.5.5 Proving Feasibility

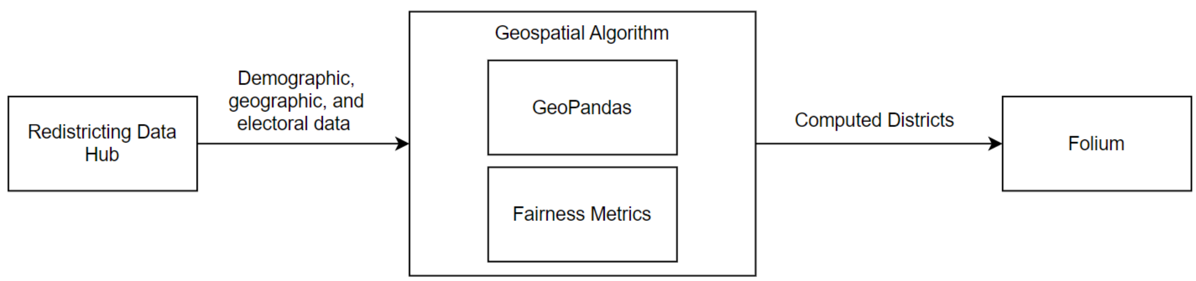
With this, we have defined a set of standards that we will use to determine fairness. The process of fairness is established so that we can communicate these standards to the public. When creating this algorithm, we have to remember that requirements might change for each state. Furthermore, we need to figure out if there are any more requirements of fairness.

4 Technological Integration

Our ultimate goal is to assemble a redistricting algorithm that accurately reflects US demographics, impartially forms district maps, and produces meaningful output that can be easily visualized. To do this, we will need to design a system that integrates all of our chosen methodologies described in Section 3.

A summary of this integration is described in Figure 4.1, seen below.

Figure 4.1 - Outline of System Design



We will first acquire the necessary data by making API calls to the Redistricting Data Hub, which will provide us with key data such as population, racial demographics, voting outcomes, and geographic layouts. This will then be processed using a Geospatial algorithm that utilizes the acquired data to draw districts according to the criteria that we will set. Geopandas will be used to record and input data related to the algorithm, as it provides an easy-to-use framework for orienting data in a geographic space. Once the maps are generated, they must also be evaluated using the metrics for fairness discussed in 3.5 to ensure that the algorithm is producing maps that are most aligned with our goal of producing impartial redistricting plans. Finally, to have our results available in an accessible UI, we will use Folium to output the maps generated from the algorithm, allowing us to embed our maps into a web-based visualizer.

This design enables us to keep a simple architecture so that we may focus the majority of our efforts on developing the algorithm itself. The most critical portion of this project is to ensure we can produce quality district maps that enable fair representation. By providing an easy-to-use foundation for our system, we can spend more time developing, evaluating, and calibrating our algorithm without getting caught up in complex implementation details. Above all, we are confident that the provided design will enable us to succeed in creating and visualizing our redistricting algorithm.

5 Conclusion

In addressing the issue of gerrymandering, our FairyMander team is on a mission to democratize the redistricting process through the power of data science. By leveraging census data and GIS, we have outlined our strategy to generate unbiased, equitable voting districts. This project not only targets the corrupt manipulation inherent in current redistricting practices but also proposes a tangible solution that champions fairness and transparency. Though there may be technological and ethical challenges, we feel that we have effectively evaluated our options through a lens of feasibility, effectiveness, and our core values of fairness and accessibility.

Our analysis will lead with a carefully considered approach. Applying a blend of Geopandas and Folium for map visualization, alongside geospatial clustering for data analysis, ensures our algorithm meets and exceeds the requirements for creating electoral districts that are both just and representative. Utilizing the data we can gather from the Redistricting Data Hub also helps to bolster our confidence in the reliability and relevance of our sources. As we move forward, we are confident that our efforts will contribute significantly to making voting fair again, emphasizing the importance of equal representation and the idea that each vote should count as much as the next.