Predicting Neighborhood Gentrification in New Jersey

by

Justin Sapun

A Thesis submitted to the Faculty

in partial fulfillment of the requirements for the

BACHELOR OF ARTS

Accepted

Amanda Landi, Thesis Advisor

Jack Burkart, Second Reader

John B. Weinstein, Provost and Vice President

Bard College at Simon's Rock
Great Barrington, Massachusetts
Spring 2024

Acknowledgments

I would like to express sincere gratitude to my thesis advisor, Dr. Amanda Landi, for her excellent mentorship, encouragement, and guidance throughout my research and project development. Her expertise and insights have been pivotal to my success. I am also grateful to another committee member, Dr. Jack Burkart, for his support and invaluable assistance. Special thanks go to my undergraduate advisor, Dr. Michael Bergman, for his guidance, instruction, and inspiration, which sparked my passion for research. Lastly, I am profoundly grateful to my family and friends for their unwavering support despite the physical distances that often separated us.

Table of Contents

List of Figures

List of Tables

Abstract

Investors seek confidence in upscaling properties, while policymakers need information to prevent displacement in gentrifying communities. In an attempt to better help stakeholders, this thesis focuses on a newfound approach to detecting noticeable changes in neighborhoods. Traditionally, census data has been used to detect trends in known classifiers. This thesis incorporates spatial analysis with the United States Census and green space data to reveal changes not evident from field-sourced data alone. Visual data will be sourced from pairs of historical and current images using the Google Static Street View API in New Jersey. I show the effectiveness and accuracy of my novel approach in predicting gentrification by comparing it with current studies in New Jersey. My framework is capable of enhancing future research in the respective field.

1. Introduction

Gentrification has gained considerable attention in the United States (US) in recent years due to its increased prevalence in many neighborhoods across the country, especially in lower income areas. The term "Gentrification" first emerged in London during the early 1960s by German-British sociologist and city developer Ruth Glass. This term extends from its roots in the word gentry, referring to people with a good social position. When Glass coined the term, she referred to a British-specific class, just below the nobility. Glass was the first to model economic transformation once she saw unconventional groups with money take over houses and rehabilitate them in London neighborhoods [1]. Gentrification has grown to encompass broader socio-economic shifts from lower income to upper middle class within neighborhoods worldwide. It remains relevant to urban change, but recently, there has been a discussion of rural gentrification. For instance, the middle class, comfortable with their retirement, may leave the busy urban environment to move to a rural community or an isolated property. This shift alters the social class structure in rural areas, resulting in a housing shortage and the potential to replace residents. This is just one example of rural displacement, as it has been shown that lower property costs and a perceived higher quality of life in a rural setting attract urban residents. A direct result is increased property values, making the rural areas unaffordable, similarly displacing residents, and altering rural demographics and economics [2] [3] [4]. This process parallels urban gentrification and should be considered when identifying New Jersey (NJ) gentrification patterns. This demographic and economic shift can appear simple on the surface, but social processes contain numerous underlying variables that may not seem related. These elements interact in unpredictable ways, making it challenging to interpret specific causes and effects. As such, observing and recognizing the signs of gentrification within a neighborhood has become increasingly popular over the last half-century. Nowadays, gentrification involves a complex dynamic of urban development, economic change, and cultural transformation. Although

gentrification is associated with gradual improvement for neglected regions, it still threatens the residents and cultures. The influx of wealth can severely increase the cost of living, making it unaffordable for existing inhabitants. It is easy to understand why gentrification is essential to model and predict for future generations.

2. Literature Review

As I have previously described, the original idea of gentrification has come to have many new meanings over the years. I will be referring to Jeffery Lin and his definition of the increased investment and influx of residents of higher socioeconomic status into a lower socioeconomic status neighborhood [5]. On the one hand, the influx of investment and displacement of lower socioeconomic residents are seen as drivers for gentrification. Conversely, these factors can positively correlate and become key aspects relating to gentrification [6]. Using this definition, metrics are needed to discover the inflow of new investment and displacement of lower socioeconomic residents to determine a neighborhood's state of gentrification adequately. Scholars have spent decades trying to quantify displacement, unable to come to a clear conclusion from the multitude of attempts [7]. This thesis contributes to the field by taking an alternative approach to predicting gentrification that does not solely rely on feature selection from surveys.

Previous work in identifying gentrification stemmed from public datasets like the US Census and the American Community Survey (ACS), which provide researchers with a national summary of statistics and estimates of neighborhood characteristics. Often, there is null data and gaps between two decennial censuses filled in with approximations from the Census Bureau. These datasets provide meaningful information but can be unreliable due to their infrequent sampling rate. Arguably, difficulties arise in finding changes in a five-year time series for instance [8]. Nevertheless, there have been many research publications consisting of just

that with decent results. For example, a machine learning model on longitudinal census features to predict home prices in American cities can achieve a median error of around 8% using two preceding time periods [9]. This is a unique approach to predicting gentrification partly because of endogenous gentrification and its relationship with housing prices. If a poor neighborhood boarding a wealthier neighborhood begins to gentrify, this process can be called endogenous gentrification. Ultimately, introducing more affluent residents into neighborhoods will increase prices, causing the native residents to migrate out [10]. The challenge when working with the Census and ACS surveys is the mass amount of data returned. Each research publication tries to differentiate itself by changing its feature selection in hopes of achieving better results. For example, one experiment used a principal components analysis (PCA) to identify neighborhoods actively gentrifying [11]. They applied the PCA to four features in three Canadian cities:

1. Mean individual income
2. Proportion of tenants
3. Employment rate
4. Percentage of local artists

Then, a review of quantifying gentrification reveals that the following features may be best in predicting gentrification [7]:

1. Mean Individual Income
2. Housing Tenure Changes
3. Socioeconomic Status
4. Educational Level
5. Percentage of local artists
6. Proximity to Green Spaces

Logically, these features signal changes that attract higher residents, altering the neighborhood's demographics and economics. The rising mean incomes and education levels suggest an influx of affluent and educated classes. Meanwhile,

the housing tenure and presence of artists can explain the wealth cultural influx. Although not discussed, it would be appropriate to emphasize the use of building permits in gentrification studies. As the Census releases data on new building permits, it is a widely normalized feature to include in one's analysis. Lastly, the report also mentions the environmental aspect of location desirability, paying tribute to public green spaces as a quantifiable feature. Green space is considered to be any land with natural vegetation that is open and accessible to the public. Interestingly, a study was done in London using green space and commute time as features, which suggests that survey data might be better than spatial models [12]. This result is due to the complex relationships between social and environmental variables that cannot often be seen with visual data. Another study in the US discusses the critical role of local parks, emphasizing the challenges in characterizing them due to the lack of centralized data [13]. It also highlights how improvements in park facilities paired with increased investment in local recreation can signal and even accelerate changes in the neighborhood's demographic and economic profile. Both studies show that green space plays a vital role in neighborhood development, so I will incorporate that data into my analysis.

The method of analyzing tabular data captured from physical surveys is becoming increasingly less popular thanks to new ideas and methods being introduced rapidly. As the process for collecting census surveys and economic data can be both time and labor intensive, it can be challenging to capture a short time series with the 10-year spacing [14]. Many researchers have been switching to a more readily accessible form of gentrification data, including development permits, current infrastructure, and even social media statements for sentiment analysis. However, it was only in the mid-2010s when publicly available geospatial data was released that researchers could use another dimension for urban research. Image change detection in predicting gentrification involves analyzing visual data to identify changes in urban landscapes over time. With detailed imagery, one could analyze finer components like color, size, and minor renovations to individual properties.

By making these resources publicly available, organizations, like Google and NASA, empower everyone to explore and understand the world around them. Over the years, advancements in image capture have included higher resolution and increased frequency of updates. As it stands, satellite and Street View images date back more than 15 years. Because many businesses, like commercial developers, rely on said imagery, constant updates are giving researchers the ability to incorporate them into analyses. These enhancements can significantly impact results, which is why researchers are opting to include geodata in their urban analysis and decision making.

Remote sensing technology was initially used for environmental and agricultural applications [15] with NASA's launch of the Landsat 1 satellite in 1972. Satellite imagery became publicly available for the first time to aid in research efforts. Due to its course resolution, it was not until 2005 when Google Earth released detailed satellite imagery that researchers were able to to monitor urban change and land use. The emergence of studying gentrification via remote sensing stemmed from the realization that satellite imagery offers a consistent, affordable, and rapid collection of large areas. An aerial perspective can also help identify building densities, green spaces, and infrastructural developments, which all contribute to gentrification. Some recent studies used high resolution satellite imagery to map urban poverty and gentrification with the assistance of Geographic Information Systems (GIS) and machine learning models. This approach has been proven to detect changes in urban areas with greater speed and lower cost than traditional census surveys. Critical indicators of gentrification described in the studies were land use, emergence of new construction, and physical appearance of neighborhoods [16] [17]. Unlike street-level imagery images, an analysis using satellite imagery heavily relies on the spatial resolution of the images. Satellite technology continues to advance, ensuring its role in understanding gentrification is expected to grow.

In the late 2000s, with developments in Global Positioning System (GPS) technology and geocoding, Google first released Google Maps in 2006, followed by

its initial Street View captures in 2007, which created the first virtual map of the world. It was not until the mid-2010s that researchers began leveraging the street-level imagery for gentrification analysis. In most urban areas, Google releases new images every couple of years giving researchers an extensive compilation. To simplify the intensity of the project, a lot of research using Street View uses image pairs to observe differentiation. For instance, see [18] for their analysis of three major cities using image pairs. By referencing two time points (roughly a decade apart), there is an argument to be made that it may fail to accurately depict the process of urban change [7]. In my exploration, I will use image pairs to classify whether there is a visual difference, as it only makes sense to include another image if I consider how rapidly the neighborhood has changed. Quantifying gentrification through an image can be difficult, so it helps to automatically analyze bigger objects in more samples than to focus on detecting small objects in each image pair [7]. Thus providing an overall efficient way of detecting socio-economic change. Big objects in my study would include new construction or major renovations. When using Street View to predict gentrification, there are only so many novel approaches. Most use neural networks to analyze changes in property aesthetics or image differencing techniques to detect changes in neighborhood characteristics over time. One study used a Siamese Network with ResNet Backbone to detect new constructions and renovations in buildings. This method embeds the input image pairs into an Euclidean space, allowing the comparison of images taken at different times to identify significant evidence of gentrification [18]. These methods complement the traditional census approach by providing real-time visual cues of neighborhood change. Like satellite imagery, Street View will further develop into a critical aid in predicting gentrification by offering more profound insights into the complexities of the phenomena.

Machine learning has significantly evolved in quantitative research because of large datasets and advanced algorithms with high accuracy. Many years ago, analysts would use descriptive statistics to summarize demographic and economic

data from census reports and other available surveys. By looking at trends over time, they could report on urban development based on median income, home values, and demographic changes. Initially, machine models were trained to predict patterns in Census data. This approach, while slow, returned good results. However, modern approaches have advanced to more dynamic and sophisticated data collection streams. As a result, models are faster and more accurate. For instance, one study used real-time USPS data and Housing Choice Voucher data to provide a more immediate understanding of gentrification impacts [19]. Other studies have used datasets scraped from Zillow, which can provide region-specific insights into housing market dynamics [20]. In turn, more accurate predictions are possible.

In a broader context, gentrification is often seen as the leading cause of displacement and economic polarization. Many studies have determined that managing gentrification has the potential to benefit disadvantaged neighborhoods. This thesis presents a unique machine learning approach to identifying gentrification using Google Street View paired with a traditional survey analysis. Firstly, this study will have developed two models. One uses binary classification to detect changes in a pair of panoramic images to identify gentrification patterns. The results have shown that this model is reasonably accurate, given the low training sample size. The second model uses an unsupervised clustering technique to group features obtained from analyzing Street View and surveying data. Secondly, this study attempts to utilize alternative data sources to identify urban and rural change in New Jersey. Due to time constraints, I was not able to extend the scope of this thesis to gather additional results for multiple counties. As Morris County borders a rural and urban setting, this county would be the best focus of this thesis. What can be described as the typical setting in New Jersey, a semi-rural environment, this thesis will uncover which neighborhoods are at the most risk of gentrifying in Morris County. Findings from this project could serve as a valuable resource to aid and improve existing methods, and future work could expand this analysis to other counties in New Jersey or America by applying similar methods.

3. Data

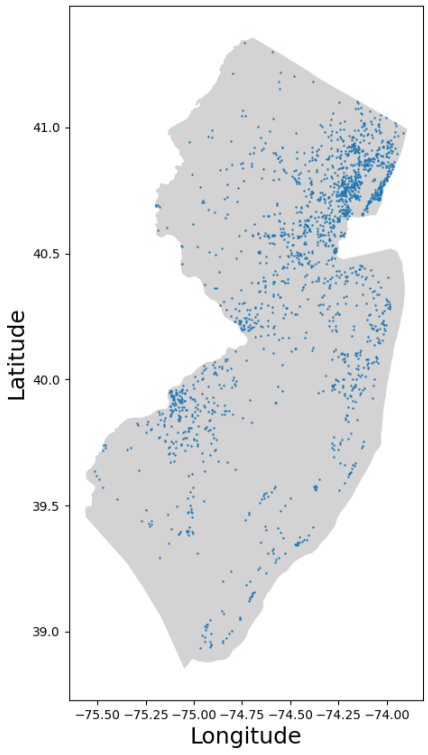
I retrieved data from many sources, including the 2022 Census Estimates [21], 2024 Census Building Permit Records (BPS) [22], Trust for Public Land [23], NJ Geographic Information Network (NJGIN) [24] [25] [26], and the Google Static Street View API [27]. The US Census aims to provide the nation's leading quality data about its people and economy through physical surveying. The last Census survey was completed in 2020, and official estimates for various economic indicators were released in 2022. In addition to demographic and financial data from physical surveying, the Census also releases other datasets like Preliminary Building Permits data compiled from New Residential Construction press releases every month. The BPS provides local statistics extending to the national scale on new privately owned residential construction. The BPS also reports estimates on the construction industry, home ownership rates, and other statistics, but I will not use them to limit survey features. The NJGIN was founded to help share geospatial content in the NJ research community. I will also be using NJGIN to easily access valid New Jersey addresses to help in my analysis of different datasets. The network produces accurate state spatial data, including boundaries, shapefiles, and addresses. I will also be using NJGIN to easily access valid New Jersey addresses to help in my analysis of different datasets. Interestingly, I found a novel green space dataset produced by the Trust for Public Land. It was created by local volunteers nationwide and contains sufficient data from New Jersey to help in my in-depth analysis. Lastly, I will use Google Street View panoramas through the Google Static Street View API. Now, I will describe each dataset in detail and its purpose in my analysis.

3.1. New Jersey Addresses/Street View

I obtained a geodatabase (GDB) file from NJGIN, which can store, query, and manage spatial and non-spatial data. Due to its large size of 2.64 GB, it takes

approximately 6 minutes to open with the necessary columns dealing with location data. Inside the GDB file, I found entries for each address with its associated shapefile, making it very easy to plot. A shapefile is a geospatial vector data format for storing geographic information. Figure 1 is a random sample of addresses in New Jersey. Unfortunately, I had to deal with computational headaches to retrieve a couple thousand valid addresses, an intrinsic issue due to the large amount of data preprocessing needed. However, by following my steps in Section 4.3.1, anyone replicating a similar experiment can simplify the work for themselves. It is important to note that I will be using random Street View image pairs from the years 2009-2012 to 2021-2024.

Figure 1: 1.7k Addresses in New Jersey



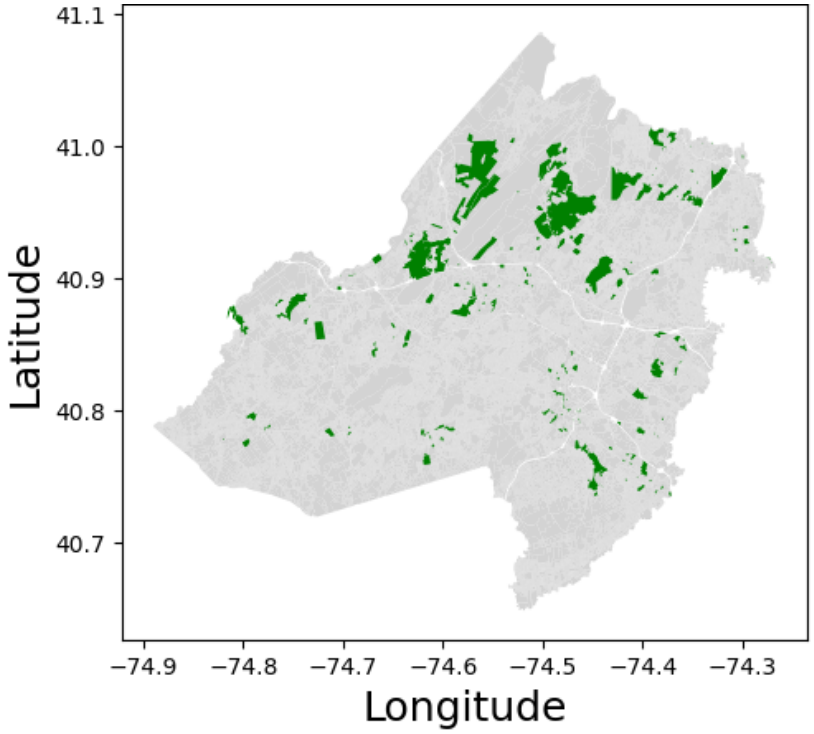
Caption: Created in 03_plot_address_func.py

3.2. Green Space

I have access to a dataset containing detailed information about most parks in the United States in the form of a geodatabase file. This file was made accessible by

the Trust for Public Land organization for research and exploration. The folder is 1.15 GB and contains publicly sourced data on 15,000 cities and towns. This data must be opened with the "FileGDB" driver to read the vector layers of the "ParkServe_Parks_05152023" layer. This layer contains park data with a location identifier along with meaningful estimates. After opening it, I later saved it as a GeoPackage (GPKG) for easier handling in the future. There are over 180 green spaces in Morris County (see Figure 2), so I know the data will be helpful in my analysis.

Figure 2: All Green Spaces in Morris County, NJ



3.3. U.S. Census

The US Census attempts to make accessing data from their website efficient, but it is not always clear. After downloading incorrect data multiple times, I could finally download a dataset displaying meaningful economic indicators by County in New Jersey. Examples include median home value, gross rent, and the year a home was built. You can sort by municipalities in each county as well. The data

was estimated in 2022 based on the 2020 Census and was packaged as a CSV. This data offers key insights into the housing market and economic status of areas, crucial for assessing urban developmental trends.

3.4. Building Permit Survey

The Residential Building Permits Survey released by the US Census contains information beginning from January 2024 and following monthly per new releases. The information is downloadable via CSV and contains numerical representations of Permits by home type and location. This data provides important insights into the pace and nature of urban development. It shows where construction is happening, what type of construction (commercial, residential, etc), and the scale of these developments.

4. Methods

4.1. Census Analysis

With the growth in technology, new data sources are becoming available, like Zillow, mobile phone data, credit card transactions, and even utility usage. Integrating multiple data sources to develop early warning systems for neighborhood change is paramount to achieving optimal results. Raw data and estimates released by the Census Bureau still remain as one of the fundamental resources in predicting gentrification due to its comprehensive time series [28]. It provides essential baseline information on demographics, housing conditions, and socioeconomic aspects. This abundance of data helps researchers identify shifts in neighborhood characteristics and overall development over time. Here, I will explore two datasets released by the Census Bureau that highlight social, economic, and housing development trends.

4.1.1. US Census

I begin by importing houseInfoMorris.csv, an official release by the American Community Survey (ACS) on their 2022 estimates of housing characteristics. This data is evaluated from the decennial Census. The data is categorized by Morris County towns already as seen in 01_dataCleaning.ipynb. The first step of any analysis is to preprocess the dataset, so I start by dropping columns which do not include an estimate in the column name using regular expressions. It is important to understand that the degree of uncertainty for these estimates is represented by a margin of error (calculated from the sampling variability). I calculated the average margin of error percent to be below 5%, showing that they are somewhat negligible compared to the estimate itself. Therefore, this will not be included in the analysis. Then, I carry out more preprocessing techniques like dropping NAs, dropping duplicates, replacing column names with only the town identifiers, and removing bizarre whitespaces. At this stage, I am able to select features from the dataset. After researching, I select the features in Table 1 to include as inputs into the final model.

Table 1: Selected Features in Census Analysis

Census Feature	Description
Built 2020 or later, Built 2010 to 2019, etc.	Estimated year the homes was built
Occupied housing units	Estimated number of occupied housing units
Median home value, median gross rent	Estimated home values and rents
Owner-occupied, Renter-occupied	Estimated number of owner-occupied and renter-ocupied housing units

I chose these features because they encompass both economic and social dimensions that can provide a comprehensive view of neighborhood dynamics. Economically, "Median home value" and "median gross rent" reflect affordability and housing market dynamics in each neighborhood. Socially, housing units built recently indicate modernization and potential cultural shifts. Then, the "Owner-occupied" and "Renter-occupied" help identify changes in home ownership trends. As discussed in the introduction, the cause of gentrification is intertwined with every aspect of

a neighborhood and its infrastructure. The features I selected provide a layered understanding of how the Morris County neighborhoods have evolved in the past five years. Lastly, I needed to convert the town names to the NJGIN standard for Morris County municipalities. I was able to easily map the changes with a dictionary. I saved this processed dataset to censusMorrisCleaned.csv for the final predicting model.

4.1.2. Building Permit Survey

The new privately owned residential construction permit data is fetched from the Census Bureau, which releases new data monthly. Each row of permit data consists of the permit issue date, category, location, number of units, and unit values. I started by opening BPS_NJ.csv and subsetting it to Morris County in 01_dataCleaning.ipynb, yielding approximately 35,861 entries. This dataset contains reports and estimates with imputation. The estimates include reported data for respondents and imputed data for nonrespondents. Given that my prediction is on the neighborhood scale in one county, I need as much data filled in as possible, so I will use the estimates in this analysis. As the estimates are made from previous records over the years, they are not subject to sampling errors. Of course, there are nonsampling errors, but that comes with every survey, including the census, hence a limitation of my data sources. I continued on to select influential features that would signal a neighborhood is gentrifying given its compilation. I choose the features in Table 2:

Table 2: Selected Features in BPS Analysis

BPS Feature	Description
TOTAL_UNITS	Estimated total number of units
TOTAL_VALUE	Estimated total value of units
UNITS_1_UNIT,UNITS_2_UNITS,...	Estimated number of varying unit types
VALUE_1_UNIT,VALUE_2_UNITS,...	Estimated value of varying unit types

These features are critical in predicting gentrification as they offer a detailed outlook of new construction and its economic impacts. The total number of units

and their value provide insights into the scale of development in each region, signaling new investments. The breakdown count and value of each unit can show trends in single-family homes versus multi-family homes, indicating a change in housing demand and demographics. In order to get a housing development perspective for each neighborhood, I decided to group by each neighborhood and sum the values for each building permit. The resulting DataFrame would include the number of residential units in development and their value. I saved the result to morrisCountyBPS.csv for the final model.

4.2. Green Space Analysis

Access to green space is highly desirable and often associated with green gentrification, the process of increased local appeal due to environment greening resulting in higher property values. Green spaces, like parks, community gardens, water surroundings, and trails can provide significant mental and physical health benefits [29]. Correlated to social and economic growth in partly rural areas, these green spaces are one of the most sought-after spaces in a neighborhood. Needless to say, the relative quality and proximity of green spaces is dependent on many factors in a neighborhood. It's understood that underinvested communities may contain less parks, or in general, lower-quality green spaces. Measuring the quality of a green space can be challenging without a visual analysis. In this section, I will walk through how I analyzed the ParkServe dataset to extract useful information on green space data for each neighborhood in Morris County.

To follow along, navigate to the *GreenSpace* directory. As the dataset downloaded from ParkServe is rather large, to reduce the time to open the large file over and over, I decided to start by saving a subset of the greenSpace.gdb GeoDatabase file as a GeoPackage. In 01_*countySubset.py*, I used GeoPandas and Fiona to read and write the multi-layered GIS formats. The greenSpace.gdb file can be opened using the "FileGDB" driver and "*ParkServe_Parks*_05152023" layer. The dataset contains three separate DataFrames, I will be using the DataFrame

comprised of park attributes explained in *Schema.pdf*. I then proceed to only save the Morris County entries to *Morris_subset.gpkg*. From here, I began exploratory data analysis (EDA) in 02_*morrisAnalysis.ipynb* to inspect the green space data. By loading the GeoPackage and a Morris County shapefile, which both contain spatial data, I was able to convert them to the same coordinate system and plot their boundaries (see Figure 2).

I was surprised to see the amount of green space data captured in Morris County considering that this data was all collected by volunteers. Upon further inspection, I found that the dataset only contained 24 towns, signifying that some towns did not have any green space recorded. It would be wrong to assume that an entire township has no green space, so I will not heavily rely on this data in my final gentrification prediction. Following, I need to convert the township names to NJGIN standards to make my life easier when merging all my processed datasets. I noticed that 21 green spaces do not have a neighborhood identifier, and considering that this dataset is so small (less than 200 entries for Morris County), I should try my best to find its town. First, I map all the towns in the green space data to the NJGIN official township name. Next, I used the spatial geometry from the official NJGIN municipalities shapefile to identify the town location for each of the 21 green spaces. Once that conversion was completed, I was able to explore the data more. View 02_*morrisAnalysis.ipynb* to see how different park attributes change with respect to each township.

The final course of action is to select features and create features for the final model. The DataFrame contains many attributes for each park, but most do not have enough non NA-values to include in my analysis. Other attributes are not sensibly correlated with gentrification and hence are excluded. This dataset also contains surveyed data from residents within 10 minutes of a green space, which I could use to dictate cultural and economic factors in relation to green spaces. I selected the attributes as listed in Table 3:

I grouped the data by township to see how each attribute would compare. To

Table 3: Selected Features in Green Space Analysis

Park Attribute	Description
SUM_BLACK_SVC	Black Non-Hispanic population within a 10-minute walk
SUM_WHITE_SVC	White Non-Hispanic population within a 10-minute walk
SUM_HISP_SVC	Hispanic population within a 10-minute walk
SUM_HHILOWSVCA	Number of low income households (<75% urban area median income) within a 10-minute walk
SUM_HHIMEDSVCA	Number of middle income households (75% - 125% urban area median income) within a 10-minute walk
SUM_HHIHIGHSVCA	Number of high income households (>125% urban area median income) within a 10-minute walk
Park_Size_Acres	Park size in acres
SUM_TOTPOPSVCA	Total population within a 10-minute walk

standardize each attribute so it is not dependent on the magnitude of data, such as county size and population, I choose to make proportions of each surveyed attribute and the total population within a 10-minute walk. I also summed the parks in each township. I then saved the resulting DataFrame to *green_space_estimates.csv* to finish the basic analysis. This CSV contains the estimates that have been processed for the final prediction.

4.3. Street View Analysis

I define a street view image pair $t_i = (t_{i,1}, t_{i,2})$ to be the pair of images capturing the same street-level perspective of the same property where $t_{1,2}$ is obtained later than $t_{1,1}$. The average date of $t_{i,1}$ is $2009 - 03 - 14$, and the average date of $t_{i,2}$ is $2019 - 03 - 21$. The Street View time series is similar to the Census sampling rate. Next, I define a neighborhood N_j to be a set containing street view image pairs: $N_j = \{t_1, t_2, t_3, t_j\}$. In this part of the analysis, I aim to identify changes in each image pair t_i with an accurate and understandable model. To accomplish this, I introduce this approach:

1. Obtain random images in Morris County
2. Apply manual labeling techniques to each image pair
3. Use computer vision analysis for model input data

4. Create the model
5. Run the model for N_j sets in Morris County and aggregate to the neighborhood level to extend my analysis later to predict gentrification

4.3.1. Obtaining Random Images

The first step can be followed in the manualTrainingSetSelection subdirectory, and like the rest of the folders, the files are numbered to help understand the workflow. I begin in 01_getAddressListAllNJ.py by opening the *Addr_NG*911 geodatabase file with only specific columns of interest to minimize runtime (6 minutes). Remember, this file contains all valid New Jersey addresses. In this section, I will be using the word geometry; it can be described as the polygon boundaries for each address, allowing for easy plotting. For now, I do not need the geometry of each address. Next, I do some basic data cleaning, like dropping NAs and converting data to int64. Finally, I build a new column that contains the entire address string from the different components and saves to nj_address_list.csv. Unfortunately, I cannot downsize my dataset just yet as I have to verify there is a good Street View panorama first. This process is handled in 02_cleanMetaData.py with my first exposure to the Street View API. At this point, I needed to create a Google Cloud account to create my personal API key. API keys are necessary authentication codes for the host to track requests and API usage. In addition, I used the requests module to send HTTP 1.1 requests and unpack retrieved data with JSON easily. Setting up the HTTP request was as simple as obtaining the base URL from documentation, an address string from the address CSV, and my API key. I can unpack the returned data with JSON, which gives us a dictionary with various elements. I obtain the status, pano_id, latitude, and longitude. Next, I drop entries with a bad status and duplicates to reduce errors in finding good addresses. After some trial and error, I sampled 2000 addresses and saved 1741 cleaned addresses to a new CSV cleaned_nj_meta_df.csv (see Figure 1).

After I create a cleaned metadata CSV containing valid Street View locations,

I can use their coordinates to retrieve the oldest and newest snapshot t_i of each location. Reference 04_randomImagePairs.py. Unfortunately, the Street View API does not allow you to access old photos, so I decided to use an open source module in Python called streetview [30]. This module uses unique access points on the Street View interface to directly download each panorama available for public viewing. Without going into too much detail, I created two functions: *save_image_pair(lat, lon, dir, sample)* and *get_sample_size(df, N, dir)*. The first function searches for a panorama given a coordinate pair, removes panoramas without a date label, and then subsets to get the oldest and current Street View images. Each image in the pair now has its own respective pano_id. From there, this function uses the pano_id to actually get each Street View panorama using the Street View API and my API key. This function then saves the two images t_i with an easily understandable name, the current sample, and a date identifier ("old", "new"). Note this function also saves each pair t_i in streetviewImg/ as a PNG and JPEG now so I do not have to convert them in a later section. The second function iterates through the cleaned metadata DataFrame to obtain Street View image pairs of a particular sample size. As the Street View API may fail or the addresses turn out to be invalid, I have to make sure. This function also saves a CSV that contains information like dates and coordinates about each image pair t_i which can be found in randomImagePairInfo.csv. When I ran the file, I chose to save 400 Street View image pairs, which took approximately 3 hours.

4.3.2. Manually Classifying Images Pairs

The next step is to label the change manually for each of the 400 Street View image pairs. I created a Tkinter Graphical User Interface (GUI) for a more accessible manual image analysis. See Figure 3 to follow along with the description below. It gives the option to search through different pairs of images, index images, and save images according to the predefined categories. The file 05_bestImgPairs.py can appear complicated, but it is simply two functions *next_pair(index: int)* and

processInput(string: str), CSV input/output logic, and basic Tkinter framework. Tkinter is a Python module that allows users to create a GUI that runs indefinitely until user termination. This package allowed me to position image pairs t_i side-by-side, along with buttons and text. The buttons are attached to a specific command, which is handled by *processInput()*. In some cases, the button command will require another image pair to appear, so it calls the *next_pair()* function to visualize another pair. This function utilizes the Python Imaging Library to load images based on the specified index concatenated with the image folder path. It is crucial to keep track of the current image index as I load and save images according to their index. I found it best to split images into four categories: change, no change (good), no change (bad), and unhelpful. I define change as any image pair t_i showing signs of clear development or renovation. Some examples include new office buildings, unfinished construction, demolition, home remodeling, new paint, new roads, etc. I define no change (good) as images that showcase no change over time, but the properties appear opulent, and the nearby environment is thriving. I define no change (bad) as images that display no change over time, but the properties and surrounding areas are in poor condition. If the images did not fit these categories, I would exclude them from the training set. In the 400 samples, there were instances of undeveloped properties, trees, and other problems that would cause difficulties in showing evidence of change. As a result, I need to further subset my dataset to only include the best visual evidence of no change and change. The results of the manual labeling are updated after program execution in imgPairsForTraining.csv.

Next, for simplicity, I created a Python script to automatically save the labeled images to a new directory. 06_saveBestImagePairs.py uses the Shutil module which provides high-level file operations for copying and removal. The script begins by loading the labeled images dataset and deleting images from the specified output directory as a safety measure. Then, it creates a string for each image using a predefined image path and image index to help copy the file. As before, I am

Figure 3: Tkinter Window Graphical User Interface

saving images as PNG and JPEG, and keeping the same predefined categorical subdirectories.

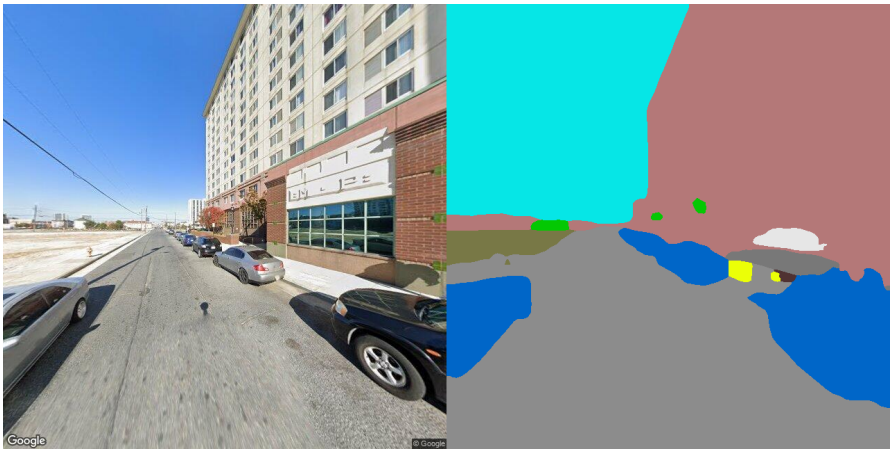
Ultimately, I decided against using three labels in my classification model for the next step as that was not the original goal of my Street View analysis. To stay true to my intentions, I created 07_forTrainingPrep.ipynb to combine no change (good) and no change (bad) into one label, no change. The different images should give good exposure to what the model should identify as no change overall. In this file, I transformed the manually labeled images dataset into a new dataset imgsForTrainingByChange.csv, which only contains the image index and the change label. A 1 is given for a visual change, while a 0 represents no visual change. There are 43 entries, although a bit low, the number should be sufficient.

4.3.3. Computer Vision Numerical Analysis

The next step in my Street View analysis is to develop numerical representations of image changes to feed the classification model. Previous research has shown that semantic segmentation and classification can be a good measure of the quality of urban appearances and their changes using street-level imagery [31]. Semantic segmentation is a computer vision algorithm that classifies each pixel in an image into predefined labels. This process involves analyzing images by inspecting every pixel to classify them as different objects, such as roads, people,

and buildings. Semantic segmentation gained popularity in the field of computer vision around 2010, coinciding with advancements in deep learning and neural networks. Thankfully, its implementation has already been vastly studied and improved, so I decided to use a PyTorch implementation on the ADE20K dataset [32]. Released by the MIT Computer Vision team, ADE20K is considered the largest open source dataset for scene parsing. Due to conflicts with Python version 3.12 and Pytorch, I created a separate virtual environment embedded in the main project root directory. In the SemanticSegmentation subdirectory, you will find the contents for the semantic segmentation model utilizing Python 3.11. The documentation provides instructions on training the model for different architectures as the implementation splits the model into encoders and decoders. For a Residual Network (ResNet), essentially a deep learning model, encoders are usually modified directly from classification networks, while decoders consist of final complexities and interpolation. I choose to use the ResNet50dilated + PPM_deepsup architecture because of the compromise between a reported accuracy of 79.73% and a reported inference speed of 8.3 frames per second (fps). I began by downloading the ADE20K dataset and then ran the 01_train.py script to train my selected encoder and decoder architectures. After training, checkpoints are saved in ckpt by default. Then, I can evaluate the model on traingImgs, a copy of my 400 addresses, using 02_test.py. As this implementation takes images in the JPEG format, I no longer need to do the conversion as I already saved this format. The semantic segmentation model output for the 400 addresses can be found in trainingResults (See Figure 4 for an example output). I chose to run the model on this sample and not the training subset in case I wanted to include more images. Compellingly, this implementation was built to use multiple Graphics Processing Units (GPU) to assist in complex computation. When I first trained the model, it took me three days to complete as I had to use Intel Iris Xe integrated graphics on my laptop. After obtaining a separate Radeon RX 7900 XT graphics card for my PC, I could train and evaluate the model in 3 hours.

Figure 4: Example of Semantic Segmentation Output



Caption: Notice how each object is color encoded. The cars are represented by blue, buildings by the brown, and roads by the grey.

Now that I have run the semantic segmentation model on the manually labeled images, the subsequent course of action involves acquiring numerical representations of the change in image pairs. This process is handled in the CompareVis directory. The first file 01_analysis.py uses CV2 from OpenCV and the priority queue algorithm to compute the analysis. The numerical analysis can be split up into two different ideas:

- Way 1: Get building object percentage of the total image and get difference for each pair
- Way 2: Overlap images, take percentage of total difference between image pairs

In the first script, I created a function *load_images_from_folder(folder)* to load all the images in a folder using the OS module and *CV2.imread()*. From there, I iterate through the image pairs t_i, computing both methods. Note that before computing, the output from the semantic segmentation model produced an image of the original panorama concatenated with the object detection. As a result, I needed to crop every image to keep only the object detection portion. When computing way 1, I created *building_percent_diff(image1, image2)*, which takes a percent difference of building pixels from 2 images. I also created *get_percent_color(image,*

color), which essentially returns the percent of specific colored pixels of an image. It does this by computing the number of unique pixels for each color in the image. Then, this function tries to index the dictionary using the color for buildings in the segmentation model (e.g. [180, 120, 120] in RGB). For way 2, I created *overlap_binary_image(image1, image2)* to overlap two images and return a black and white representation of differing pixels. This process can be done by starting with a blank black image (Numpy array representation), iterating through every pixel in the image pairs, and only updating the blank image with a white pixel if the pixels in the image pair are the same. I also created *overlap_percent_diff(image1, image2)*, which uses the previous two functions to compute the overall percentage of different pixels in the binary image representation. Finally, this script saves the computed numbers to trainingNumericalAnalysis.csv with each image index.

The last thing to do is inner merge the numerical analysis with the manual classification labels. This is completed in 02_makeTrainDataset.ipynb and the resulting DataFrame is saved to trainingSet.csv.

4.3.4. Model Selection – Classification

Model selection is one of the most critical steps in any machine learning development. Clearly, the prediction task at hand requires a classification model as the target variable is discrete. As a Python programmer, the Scikit Learn library is my personal preference for machine learning implementations. As I will be performing *classification*, some options come to mind given the problem (see Table 4).

I started by fitting potential models to the training set CSV to see which produced good accuracy. At this point, there is no cross-validation (CV) being performed, as it is more important to use CV when fine-tuning hyperparameters for specific models. The code and scores can be found in 03_train.py. Evidently, the Kernel Support Vector Machine (SVM) had the highest average accuracy across multiple tests yielding 95%. The SVM showed promise, given that the data input only had two primary features. However, I ultimately decided to go with the K

Table 4: Qualitative and Quantitative Comparison of Classification Models

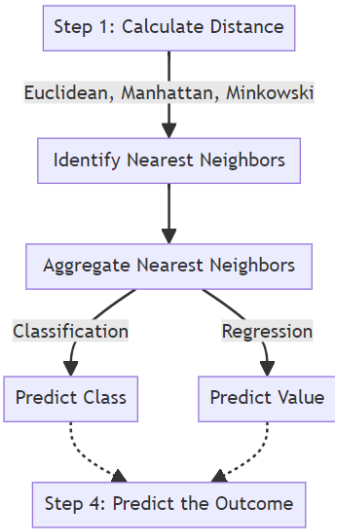
Model	Advantage	Disadvantage	Accuracy
K Neighbors Neighbors Classification (KNN)	Simple to understand, fast and efficient	Computational expense (takes up more memory and data), sensitivity to k parameter	85%
Decision Tree Classification	Interpretability, no need for feature scaling, good for linear or nonlinear data	Poor results on very small datasets, overfitting can easily occur	65%
SGD Classification	Efficiency with large datasets and is well-suited for learning from data streams in real-time	The updates are noisy and have a high variance, less stable optimization	90%
Kernel Support Vector Machine (SVM)	Good performance on nonlinear data, not affected by outliers, not sensitive to overfitting	Not ideal for many features, more complex	95%
Random Forest Classification	Strong and accurate, high performance on a wide range of data inputs	No interpretability, overfitting can easily occur, need to choose the number of trees carefully	90%

Nearest Neighbors (KNN) Classifier for binary classification with an average score of 85%. The model is efficient and straightforward and does not pose any serious disadvantages for the problem scale.

The actual model implementation can be seen in 04_knnModel.ipynb. Initially, during parameter tuning and performing cross validation techniques, I realized the model was severely overfitting to the no change class. This problem was a direct result of having an unbalanced dataset. To solve this issue, I undersampled the no change class and sampled 25 entries. Using cross validation techniques, I found the optimal model to consist of a test split of 20% and a k-neighbors value of 3. I then set up a pipeline that processed sklearn's standard scalar and sklearn's KNN: the standard scalar scales the data so the mean is 0 and the standard deviation is 1, and the KNN classifier calculates a distance; in my model, I use the default Euclidean distance. Next, the model tries to identify the nearest neighbors, which is essentially just an optimization problem of finding one point in a set closest to another given point. Then, the model aggregates the nearest neighbor. In other words, the model determines the predominant class label among the neighbors. Then, the last step is to predict the outcome, which for classification is the class

label. Reference Figure 5 below for an illustration of KNN basics. After model implementation on the training set, I had a test set prediction accuracy of 71%, which is fair. Finally, I saved the model to a pickle file with the Joblib module, allowing me to save objects through serialization.

Figure 5: KNN Model Workflow



Caption: Courtesy of Christain Leo.

4.3.5. Morris County Classification

The last step in my Street View analysis is to run the final model on N_j sets in Morris County to obtain a developmental perspective for each neighborhood. In Step 1, retrieving addresses was simple as they were randomly sampled. Here, it is going to be much more complicated as I need to obtain samples i for t_i for each neighborhood N_j. Follow along in 01_getMorrisAddress.ipynb. Once again, I needed to open Addr_NG911.gdb (valid NJ addresses) and subset to Morris County as the initial retrieval in Step 1 (Obtaining Random Images) did not save the municipality. From there, I had to open Municipalities.shp to read the official municipalities in Morris County. I created a simple town list and saved it to morris_towns.csv for future reference. The idea was to use the town list to get a sample i from the valid address DataFrame for each of the 39 neighborhoods N. With two different datasets, it is possible that the data may be entered entirely

differently. Upon inspection, I found the NJ address list to have almost a quarter of township names different from the official municipalities list. There are two ways to handle this situation:

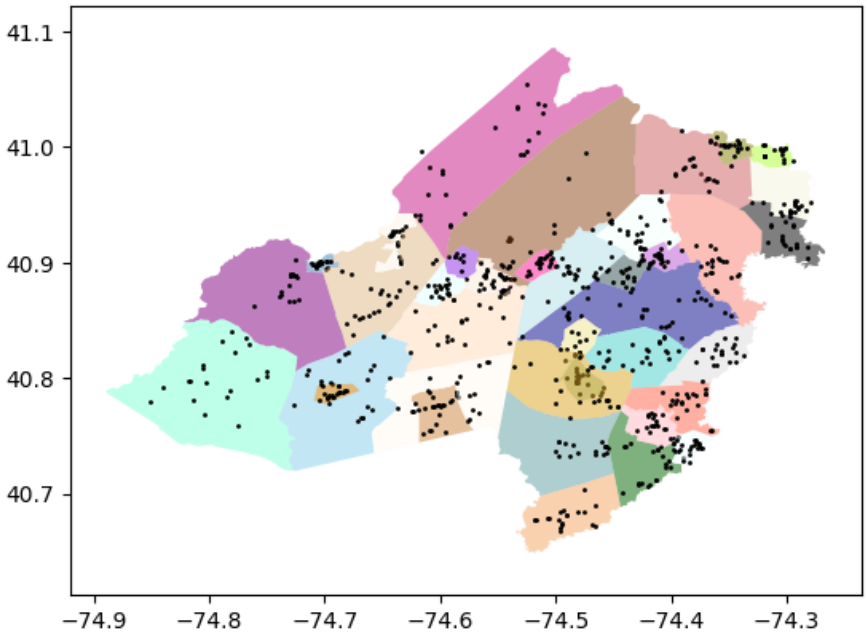
1. Use coordinate data to group each New Jersey address into a town enclosure.
2. Map every different township name to the correct township name.

Using coordinate data for each address was too computationally expensive to execute for a large dataset. While the second option was not possible because Morris included towns like 'Chatham Borough' and 'Chatham Township' when the NJ address list only referred to its location as 'Chatham'. Unfortunately for me, neither of them was a good solution. So, I decided to combine them. I started by mapping the towns without conflicting problems, as described in one DataFrame. Interestingly, the NJ address list referenced Census-Designated Places (CDP), essentially a population concentration defined for statistical purposes. These places made the mapping slightly difficult as I needed to complete some research on neighborhoods and CDPs in Morris County. One example is Long Valley, a CDP in Washington Township. Now, to handle the towns with similar names, I had no choice but to use coordinate data. By using official geographical boundaries for each township, I was able to group each address into a township in New Jersey, taking approximately 12 minutes for 55k entries. Now, I was able to merge two DataFrames to obtain one combined address list with corrected municipal addresses. I saved this information to morris_address.csv containing 171k addresses.

This step in the process is very similar to step 1 (Obtaining Random Images) when retrieving addresses in NJ, so I will not go into detail, but I will note that the implementation is slightly different. Once I have the addresses and their town identifier, I can begin verifying metadata to prepare for obtaining Street View imagery. The code can be seen in 02_cleanMorrisMeta.py, where I use the Street View API to validate each address. I sample 300 addresses per town during this validation step. I output the cleaned addresses to cleaned_morris_address.csv with 8k entries. Next, I download i images for each neighborhood N_j. This

procedure is relatively straightforward as it is identical to the first time I used the streetview module with the Street View API and saved the images to JPEG and PNG format. It is also at this point that I save the image information to randomImagePairInfo.csv so I could later map each image to its location. At this point, I have 780 images to input into the Semantic Segmentation model. See Figure 6 for the image pair distribution among the towns.

Figure 6: Morris County Sample Distribution



Caption: Created using 04_figureOutput.ipynb.

Now that I have preprocessed the data, I begin feature engineering like step 3 (Computer Vision Numerical Analysis). I copy the Streeview JPEG files into the Semantic segmentation folder and run 02_test.py using the same ResNet50dilated + PPM_deepsup architecture I previously trained. The output is located here. Now that I have each Morris sample image categorized into objects, I have to obtain a numerical representation of the change in image pairs. This computation is handled in the CompareVis directory, where I copy the segmentation output to this folder. I computed the numerical analysis in 01_analysis.py on the Morris input by finding the difference in building object percentage for each pair and finding the difference in percentage of overall pixel difference between image pairs.

I saved the engineered features to morrisNumericalAnalysis.csv.

Finally, I am ready to predict change for each neighborhood in Morris County. I have now defined $j = 39$ for N_j neighborhoods and $i = 20$ for t image pairs. Reference the prediction in 05_predictChange.ipynb. I start by loading the KNN model, the Morris numerical analysis, and the Morris image pair information. I could easily predict using the model on the numerical analysis and added the predicted series to the info DataFrame. Now, I can see which locations had the most visual change predicted by summing up the prediction column. This data was saved to morris_pred_grouped.csv for later use. In Table 5, you can see the top 5 neighborhoods with the most predicted change, an example of one Street View pair in Figure 7, and a plot of the overall predictions in Figure 8.

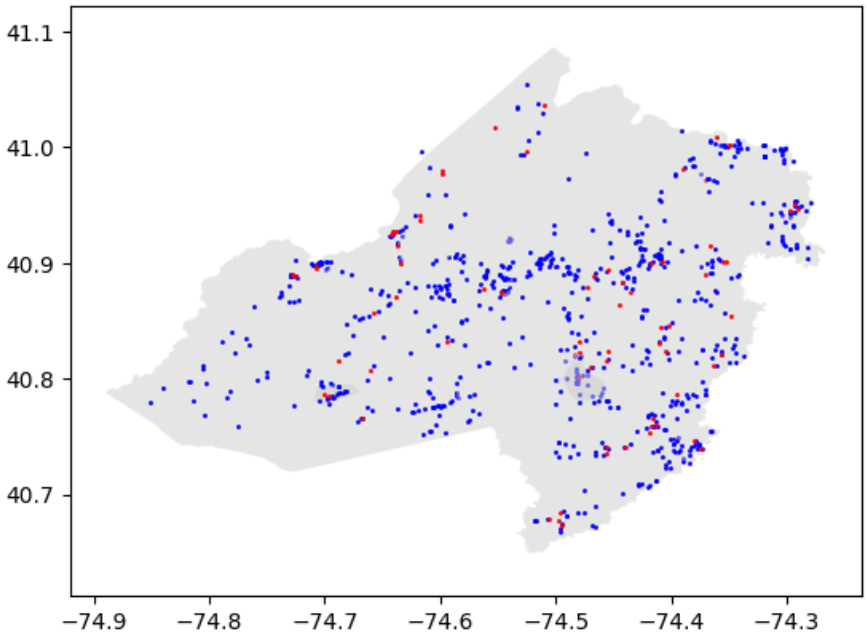
Table 5: Top neighborhoods in Morris County with the most visual changes

Town	Visual Changes Detected
Jefferson Township	7
Mount Arlington Borough	6
Parsippany-Troy Hills Township	5
Long Hill Township	4
Madison Borough	4

Figure 7: Example Image Pair 143

Caption: This image pair is located in Mount Arlington. The old image is on the left, and the new image is on the right. The KNN model correctly classified this image pair as change detected as you can tell by the new driveway and construction.

Figure 8: Morris County Visual Change Prediction



Caption: Above is a figure created to showcase the visual change predictions in Morris County, where red is change detected and blue is change not identified.

5. Results

5.1. Predicting Gentrification

Everything in the analysis leading up to this point has been in preparation for creating a second model. Essentially, the previous methods I explained in detail were for feature engineering inputs to the second model. Now, I can further elaborate on how I plan to predict gentrification in Morris County. As the goal of this thesis is to predict neighborhood gentrification in one county, Morris County, I am very limited in how to proceed given the scale of the data. The current feature space is 39 N x 35 N attributes. Since my feature space is limited to only 39 neighborhoods, most machine learning techniques are automatically excluded. For example, I cannot use a classification model. It does not matter if I have hundreds of attributes for each neighborhood because there are still only 39 entries, it is nonsensical to train a model. If I were able to include multiple counties in this thesis, I would consider a second binary classification model that would classify

neighborhood gentrification. Seeing as this is not the reality, I resorted to using an unsupervised clustering algorithm in PredictingGentrification/. Clustering in machine learning involves automatically discovering natural groupings in data. Contrary to supervised algorithms, clustering algorithms only need input data to interpret and find clusters in the feature space. The clustering algorithm is ideal for predicting my final goal because there is an abnormal feature space that other machine learning algorithms could not handle well.

To execute a clustering algorithm, I must first prepare the data. The data is obtained from the conclusions of each method's analysis. I start in 01_clusteringModel.ipynb by loading the data from the Street View, Census, BPS, and green space analysis. I also import the Morris County shapefile to obtain each town's size in acres. I immediately start by merging DataFrames. Fortunately, this process is not intricate as I already standardized the neighborhood names for each dataset according to the NJGIN standard. The only noticeable complication was converting empty data to zeros and performing a left merge with the green space dataset due to it not containing data on 11 towns. In the Census and BPS methods, I never explicitly mentioned why I did not standardize values irrespective of town size as I did in the green space methods. This was because each dataset did not have enough data to create new features. Having arrived at this stage with combined housing characteristics from the Census and BPS dataset, I can perform some more feature engineering to remove town size as a consideration in the model. I created the following new features in Table 6 and dropped the unhelpful ones.

Table 6: New Features created from the Census and BPS processed datasets

Feature	Description
OccupiedHousingUnits	Ratio of occupied housing units to total units
OwnerOccupied	Ratio of owner-occupied housing units to total units
RenterOccupied	Ratio of renter-occupied housing units to total units
BuiltRecently	Ratio of homes built recently to total units
AvgUnitValue	Ratio of total unit value to total units
ParkSpace	Ratio of total green space to county size

The resulting feature space is $39N \times 16$ attributes, further validating my choice in

a clustering algorithm. Because the feature space is small, I attempt to improve the performance of the clustering model by assigning appropriate weights to different data features. Giving feature weights can reduce the noise and effect of less relevant features while intensifying important features simultaneously. The breakdown for each feature is in Table 7.

Table 7: Features and Their Assigned Weights

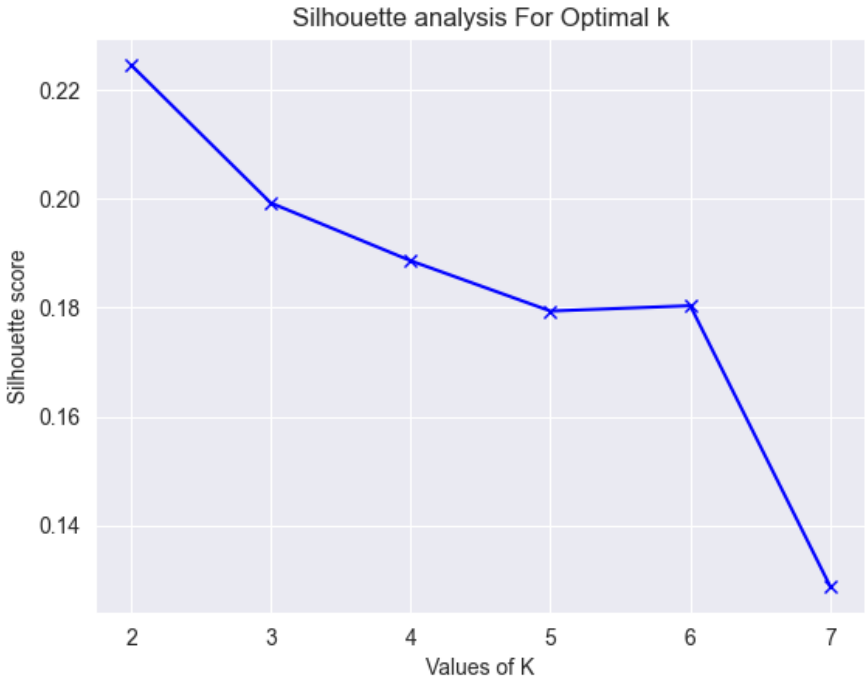
Feature	Weight
Median Home Value	0.8
Median Gross Rent	0.2
pred	1.0
SUM_BLACK_SVC	0.4
SUM_WHITE_SVC	0.4
SUM_HISP_SVC	0.6
SUM_HHILOWSVCA	0.3
SUM_HHIMEDSVCA	0.6
SUM_HHIHIGHSVCA	0.2
OccupiedHousingUnits	0.6
OwnerOccupied	0.3
RenterOccupied	0.3
BuiltRecently	0.7
AvgUnitValue	0.1
ParkSpace	0.5

Caption: As this thesis set out to prove Google Street View is capable of identifying Gentrification, I wanted my results from the first machine learning model to be the most influential. I also put emphasise on Median home value, middle class populations, how recently the homes were built, and the occupied housing units.

With preprocessing and feature engineering concluded, I can now construct a clustering model. I choose to use the K-Means algorithm because it is straightforward to implement and understand. The algorithm partitions a dataset in K distinct, non-overlapping clusters by minimizing the variance with each cluster. This algorithm is also advantageous compared to others when you know how many clusters exist in the feature space. To determine the most optimal number of clusters for the data, I performed a Silhouette analysis. The silhouette method is a great way to evaluate the quality of a cluster as it measures how each data point fits within its cluster and how far it is relative to other clusters [33]. From

the sklearn metrics package, the silhouette score ranges from -1 to 1, where one dictates the data point is centered in its cluster but far from other clusters. By taking an average of all the silhouette scores, I can understand how well the data is clustered. Intuitively, to find the optimal number of clusters, find where the average silhouette score is highest. I found the best number of clusters to be $K = 2$ (see Figure 9). So, I initialized the K-means algorithm and fitted it to my scaled features. K-means is super fast, especially since I have a small feature space. After joining the labels to the DataFrame, I can see from the clustering that there are twenty towns in Group 0 and nineteen in Group 1.

Figure 9: Average Silhouette K-Means Cross Validation



Caption: Silhouette coefficient decreases with higher k values.

In supervised machine learning, clustering generally groups similar data points based on their features. A label for a cluster is arbitrary for the algorithm, as its only goal is to group objects and not name each group. This differs from a supervised approach, which compares each cluster to known sets of clusters to get the closest match and, in turn, produce a label. Now that I have run unsupervised clustering on the data, I am left to interpret the results myself. I will be using

domain knowledge and external information to interpret said clusters and assign meaningful labels. I start by grouping by the result labels and taking the median, average, and sum for each respective feature. For instance, I take the median of each neighborhood for the median gross rent for each result label. Reference Table 8 below.

Table 8: First Five Features Grouped by Clustering Result

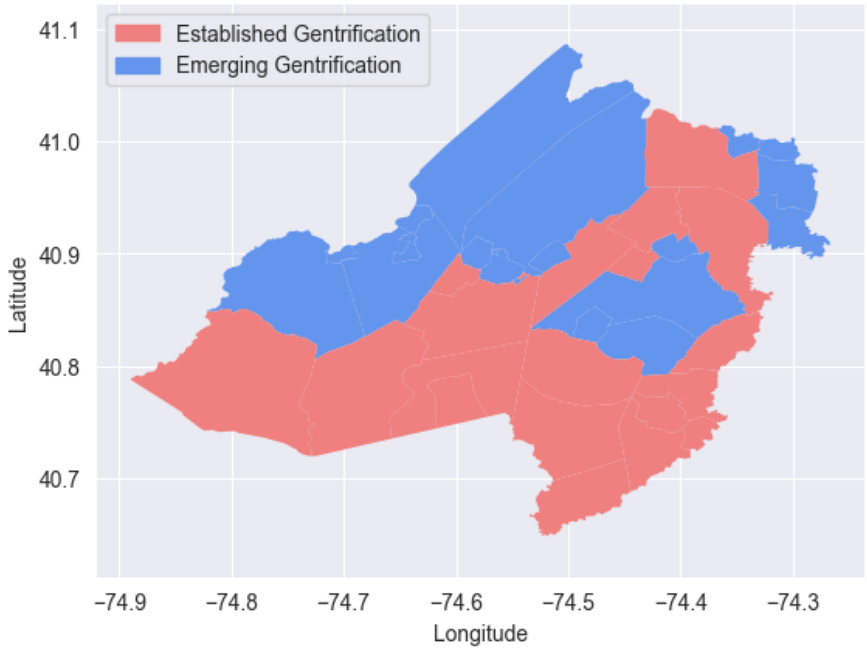
Result	Median Home Value	Median Gross Rent	pred	SUM_BLACK_SVC	SUM_WHITE_SVC
result0	688200	2161	32	0.005655	0.391592
result1	407700	1698	45	0.028078	0.625125

Caption: Reference 01_clusteringModel to see more features from the clustering result.

There are a couple of data points that caught my attention, making it clear that these neighborhoods cannot be split into gentrifying and non-gentrifying. Result 0 seems to characterize neighborhoods with established gentrification, evident from their high median home value of $688,200 and median gross rent of $2,161 as compared to the 2022 New Jersey medians of $454,000 and $1,577 respectively. The demographic profile also shows a higher proportion of services predominantly utilized by the white demographic (39%) with lesser representation from Hispanic and black communities. The dominance of owner-occupied housing and moderate park space confirms that these neighborhoods are stable and mature, typical of areas that endured extensive gentrification. Result 1 seems to characterize neighborhoods with emerging gentrification, apparent from their lower median home value and median gross rent. Almost identical to New Jersey's reported values for 2022. These neighborhoods also exhibit a significant amount of recent housing development, where 5% of current units were built in the past ten years. This, paired with the larger park spaces, suggests ongoing investments appealing to new residents. This result also contains a higher 27% renter-occupant ratio, nearing the owner-occupant ratio of 40%. It is common in gentrifying neighborhoods to have more renters than homeowners, so I can conclude that these dynamics reflect neighborhoods in the early stages of gentrification. Finally, after interpreting the

clustering results, I proceeded to visualize these results to see the geographical distribution of each cluster (as shown in Figure 10).

Figure 10: Predicted Gentrification in Morris County



Caption: The two clusters are visualized on Morris County with their derived labels.

5.2. Discussion

5.2.1. Model Performance

In order to have confidence in my results, I need to evaluate the performance of my machine learning models. The binary classification model developed in this study demonstrates a promising approach to detecting visual changes using Google Street View images. Despite the challenges posed by the small training sample size, given the time constraints, the model achieved reasonable accuracy (see Figure 11). From the training and validation set, the model mainly showcases 80% average precision and recall. There is one exception when predicting the positive change class. The recall is 56%, indicating the model would rather label an image pair as no change over change. I tried to adjust for this by undersampling, but there is still an imbalance in the training set. Considering the general model performance

and barely having any errors in classifying positive change, I believe this model is adequate.

Figure 11: KNN Classification Model Test/Training Set Scores

	precision	recall	f1-score	support
0	0.82	0.95	0.88	19
1	0.83	0.56	0.67	9
accuracy			0.82	28
macro avg	0.83	0.75	0.77	28
weighted avg	0.82	0.82	0.81	28

	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.00	0.00	0.00	1
accuracy			0.71	7
macro avg	0.42	0.42	0.42	7
weighted avg	0.71	0.71	0.71	7

Caption: To learn more about sklearn's classification report and the evaluation metrics, visit the official sklearn documention.

As for the unsupervised clustering model, the silhouette score for the clustering result is 0.18, which according to Kaufmann and Rousseuw [33], means the data is unstructured. However, I accept this conclusion because my features describe many socio-economic aspects, and there is no numerical cookie-cutter definition of various gentrification stages. In addition, I computed the highest average silhouette score earlier to be around 0.22, meaning that my model is close to its peak potential. To further inspire confidence, I used a Principal Component Analysis (PCA) to reduce the feature space to 2 columns and visualized the clustering as well. Once a feature space extends three dimensions, a dimension reduction algorithm like PCA or T-distributed Stochastic Neighbor Embedding (TSNE) is needed to visualize high-dimensional data. Refer to sources [34] and [35] to learn more about these methods. As shown in Figure 12 below, there are visible clusters with minor

overlap.

Figure 12: Lower Dimension Visualization of Clustering Results

Caption: This plot uses PCA to show similarities between the two reported groups. The labels 0 & 1 are ambiguous from the clustering result. The x and y axis don't mean anything physically, just a combination of principal components.

5.2.2. Interpreting the Result

Now that I have established that the models are able to correctly identify visual changes and separate data clusters, I must assess the predictive capability of this entire analysis to see if there is any truth to my predictions. By comparing my results against documented instances of gentrification, I should be able to observe nuanced alignments with the known gentrification patterns in various neighborhoods.

A study released in January of 2024 by New Jersey Future set out to compare Morris and Monmouth counties on their effectiveness in implementing affordable housing policies. The report particularly analyzed their policies on compliance, impact on segregation, and community diversification. Furthermore, the increase in affordable housing was shown to be linked to a decrease in both racial and economic segregation [36]. In a general sense, this is a perfect example of how

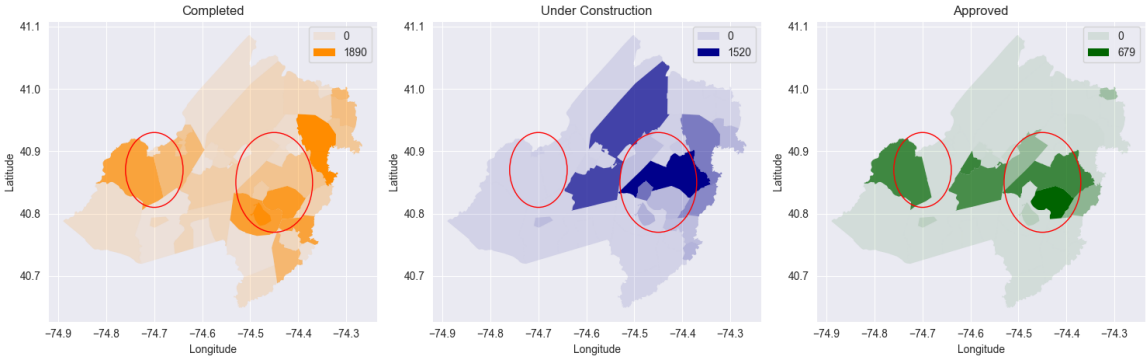
predicting gentrification can help influence neighborhood planning and policies.

Affordable housing is often introduced into a region to help diversification without negatively impacting economic stability. In Morris County (and other counties in New Jersey), I see the introduction of affordable housing as one of the state's major contributions to counteract the effects of gentrification. I downloaded a dataset of all affordable housing units from the County website to see how my results compare to their development [37]. It is a bit of a novel idea, but by visualizing the distribution of affordable housing across Morris County, I can see what neighborhoods the state, researchers, and investors believe to be at risk of gentrifying. This understanding allows me to forgo a long time-series analysis and compare my results from Figure 10 quickly.

I imported the affordable housing dataset and the Morris County Municipalities Shapefile in 02_morrisAffordableHousing.ipynb. Once again I renamed the town names to match the NJGIN standard. As I wanted to create three separate plots for units split by completion status, I found it easier to streamline the plotting process by creating the function *plot_func(subset, ax, df, num)*. This function subsets the big dataset, preprocesses it, groups the data to get the sum per town, normalizes the values from 0.1 to 1, and manually plots a heatmap. I had trouble plotting the county boundaries with varying colors based on the housing unit proportions, so the function became messy when I tried to use the transparency parameter of one color to convey the proportions. Overall, I am happy with the results of this comparison as I can confirm that my results in Figure 10 align with state investments. In Figure 13, you will find three plots of affordable housing already completed, under construction, and approved. Notice that most units are distributed in the two red enclosed regions. Although my results diverge outwards a little past those red areas, this almost parallels the state gentrification predictions. Notably, I previously explained my reasoning for interpreting one group as established gentrification, which is also slightly verified by these plots. As you can see, there are pronounced neighborhoods where my model predicted

"established" (red) in the completed plot. This signifies that units were being built in these regions decades ago, and since then, these areas are more likely to be at a late stage of gentrification.

Figure 13: Affordable Housing Units in Morris County



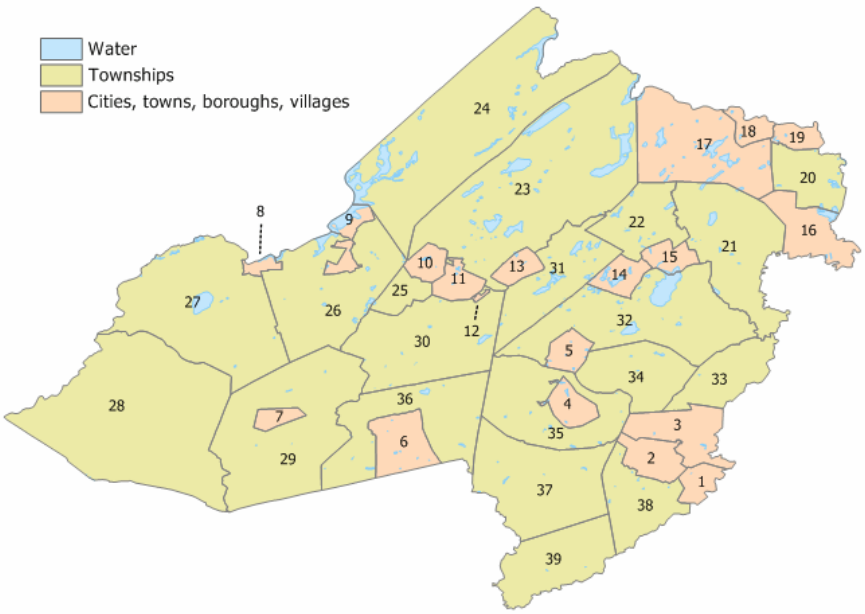
Caption: These plots showcase the concentration of affordable housing in Morris County. Each plot is distinguished by current affordable housing completion status. The red circles in each plot are there for reference to highlight the major developments. Helpful to compare against my results in Figure 10.

6. Conclusion

In this thesis, I introduced an innovative approach to predicting neighborhood gentrification by utilizing computer vision algorithms on street-level images, statistical analysis on survey data, and advanced machine learning techniques. With temporal gaps in traditional survey census data and advancements in technology, using other sources has become increasingly more popular in gentrification studies. The primary objective of this research was to assess the stages of gentrification by using Census estimates, green space data, and Google Street View imagery for the first time. With existing gentrification research relying on large data tables, this thesis relied on a visual analysis to make up for using a small portion of available survey data. Nevertheless, the results prove that using unorthodox data sources can be used to predict gentrification accurately. Two machine learning models were developed to handle the data for Morris County in New Jersey. This research was carried out in Morris County because of its diverse mix of urban and suburban

landscapes, and its history with economic growth and developmental pressures. The validations for each model reveal the feasibility of identifying and predicting neighborhood gentrification from a combined analysis in a semi-urban environment. In the one county I investigated, I found the neighborhoods classified with early signs of gentrification to be in the NorthWest half of the region, encompassing many of the larger population centers (see figure 14). Interestingly, Morris County borders rural counties on its NorthWest boundary. The remaining neighborhoods were deemed at a much (relative) later gentrification stage and were distributed on the SouthEast border, adjacent to Essex and Union counties, both major urban regions. A prime example of endogenous gentrification. By verifying my results with state investments into affordable housing, my approach has showcased its potential to be a valid aid for future research on the topic.

Figure 14: Population Centers in Morris County



Caption: Courtesy of Jim Irwin.

Although my work establishes a pioneering approach to data integration by merging image-based analysis with traditional survey techniques, several limitations

impact the overall ability of my methods to detect urban and rural change. One limitation is the poor random samples of Street View images, see Figure 15, which provide no insight into neighborhood characteristics (e.g. slight perspective changes, trees, differing camera quality, etc). More data to train the model would help reduce the noise. A second limitation (which was mentioned previously as a result of studying one county) is the severe reliance on user input in manually classifying image pairs and interpreting clustering results. A unique solution could be to create an interactive website where volunteers could help classify image changes to create a large dataset. For example, CMU Create Lab created an online classifier interface to have online users train artificial intelligence to Recognize Industrial Smoke Emissions (RISE). While my classification model performed okay, a third limitation could arise from the low attributes computed for each image pair. As technology and algorithms advance, spatial data will be a practical data source for many fields. If I were to lead an attempt to predict gentrification solely using street-level imagery, I would seriously consider these limitations, how to amass data, and what criterion to use when classifying change before training the model.

Figure 15: Example Image Pair 33

Caption: This image pair located in Hillsborough is one of the bad samples that could be retrieved from the Google Street View API. With such a large obstruction to the house in question, the perspective is squandered.

In spite of these limitations, the research, methods, and results described in this thesis provide evidence that gentrification is not only tied to demographic and economic data for instance. The dynamics can be partially explained with a visual analysis, but further exploration into only using spatial data will determine if this is possible. Regardless, identifying signs of gentrification at any stage can benefit local governments, residents, and outside investors. Early detection allows stakeholders to proactively manage the changes brought by gentrification to better address the needs of residents and newcomers. Leading to a more balanced development without displacement and significant cultural shift.

A. GitHub Repository

Figure A.1: GitHub QR Code

Caption: This QR Code contains a link to
https://github.com/JSapun/PredictingGentrificationNJ
which stores my source code for this project.

References

[1] J. Brown-Saracino, *The Gentrification Debates*. Routledge, Sep. 2013, ISBN: 9781134725649. DOI: 10.4324/9781315881096. [Online]. Available: http://dx.doi.org/10.4324/9781315881096.

[2] N. Smith, "New globalism, new urbanism: Gentrification as global urban strategy," *Antipode*, vol. 34, no. 3, pp. 427–450, Jul. 2002, ISSN: 1467-8330. DOI: 10.1111/1467-8330.00249. [Online]. Available: http://dx.doi.org/10.1111/1467-8330.00249.

[3] G. Gotti, *A review on rural gentrification—what happens when urban development migrates to the countryside?* Rural Radicals, Dec. 2023. [Online]. Available: https://www.linkedin.com/pulse/review-rural-gentrificationwhat-happens-when-urban-development-5r3xe/.

[4] M. Phillips, D. Smith, H. Brooking, and M. Duer, "Re-placing displacement in gentrification studies: Temporality and multi-dimensionality in rural gentrification displacement," *Geoforum*, vol. 118, pp. 66–82, Jan. 2021, ISSN: 0016-7185. DOI: 10.1016/j.geoforum.2020.12.003. [Online]. Available: http://dx.doi.org/10.1016/j.geoforum.2020.12.003.

[5] J. Lin, "Understanding gentrification's causes," *Economic Insights*, vol. 2, no. 3, pp. 9–17, Jul. 2017. [Online]. Available: https://ideas.repec.org/a/fip/fedpei/00020.html.

[6] L. Ding and J. Hwang, "The consequences of gentrification: A focus on residents' financial health in philadelphia," *Cityscape*, vol. 18, no. 3, pp. 27–56, 2016, ISSN: 1936007X. [Online]. Available: http://www.jstor.org/stable/26328272 (visited on 05/06/2024).

[7] S. Easton, L. Lees, P. Hubbard, and N. Tate, "Measuring and mapping displacement: The problem of quantification in the battle against gentrification," *Urban Studies*, vol. 57, no. 2, pp. 286–306, Jul. 2019, ISSN: 1360-063X. DOI: 10.1177/0042098019851953. [Online]. Available: http://dx.doi.org/10.1177/0042098019851953.

[8] M. Cohen and K. L. S. Pettit, "Guide to measuring neighborhood change to understand and prevent displacement," *National Neighborhood Indicators Partnership*, Apr. 2019. [Online]. Available: https://www.urban.org/sites/default/files/publication/100135/guide_to_measuring_neighborhood_change_to_understand_and_prevent_displacement.pdf.

[9] K. Steif, A. Mallach, M. Fichman, and S. Kassel, *Predicting gentrification using longitudinal census data*, Urban Spatial, 2016. [Online]. Available: https://urbanspatialanalysis.com/portfolio/predicting-gentrification-using-longitudinal-census-data/.

[10] V. Guerrieri, D. Hartley, and E. Hurst, "Endogenous gentrification and housing price dynamics," *Journal of Public Economics*, vol. 100, pp. 45–60, Apr. 2013, ISSN: 0047-2727. DOI: 10.1016/j.jpubeco.2013.02.001. [Online]. Available: http://dx.doi.org/10.1016/j.jpubeco.2013.02.001.

[11] R. A. Walks and R. Maaranen, "Gentrification, social mix, and social polarization: Testing the linkages in large canadian cities," *Urban Geography*, vol. 29, no. 4, pp. 293–326, May 2008, ISSN: 1938-2847. DOI: 10.2747/0272-3638.29.4.293. [Online]. Available: http://dx.doi.org/10.2747/0272-3638.29.4.293.

[12] C. Hamnett and J. Reades, "Mind the gap: Implications of overseas investment for regional house price divergence in britain," *Housing Studies*, vol. 34, no. 3, pp. 388–406, Mar. 2018, ISSN: 1466-1810. DOI: 10.1080/02673037.2018.1444151. [Online]. Available: http://dx.doi.org/10.1080/02673037.2018.1444151.

[13] M. Walls, *Parks and recreation in the united states: Local park systems*, Resources for the Future, Jun. 2009. [Online]. Available: https://media.rff.org/documents/RFF-BCK-ORRG_Local20Parks.pdf.

[14] C. R. Nwanna, "Gentrification in lagos state: Challenges and prospects," British Journal Publishing, Inc., 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:6783985.

[15] U. G. Survey, *Landsat 1*. [Online]. Available: https://www.usgs.gov/landsat-missions/landsat-1#multimedia.

[16] L. Lin, "Monitoring and modeling gentrification using remote sensing and geospatial technologies," AAI29161537, Ph.D. dissertation, USA, 2022, ISBN: 9798841754503. [Online]. Available: https://www.proquest.com/docview/2716091195?pq-origsite=gscholar&fromopenview=true&sourcetype=Dissertations%20&%20Theses.

[17] L. Lin, L. Di, C. Zhang, L. Guo, and Y. Di, "Remote sensing of urban poverty and gentrification," *Remote Sensing*, vol. 13, no. 20, p. 4022, Oct. 2021, ISSN: 2072-4292. DOI: 10.3390/rs13204022. [Online]. Available: http://dx.doi.org/10.3390/rs13204022.

[18] T. Huang, T. Dai, Z. Wang, *et al.*, *Detecting neighborhood gentrification at scale via street-level visual data*, arXiv, Jan. 2023. DOI: 10.48550/ARXIV.2301.01842. [Online]. Available: https://arxiv.org/abs/2301.01842.

[19] A. Stern and G. Gilling, *How we used machine learning to predict neighborhood change*, Medium, Oct. 2021. [Online]. Available: https://urban-institute.medium.com/how-we-used-machine-learning-to-predict-neighborhood-change-bf52c9f32fda.

[20] J. Yoo, *Identifying gentrification using machine learning*, U.S. Census Bureau, Apr. 2023. [Online]. Available: https://www.census.gov/content/dam/Census/library/working-papers/2023/demo/sehsd-wp2023-15.pdf.

[21] U. C. Bureau, *New jersey 2022 housing characteristics estimates*. [Online]. Available: https://data.census.gov/table?t=Financial%20Characteristics:Housing%20Value%20and%20Purchase%20Price:Insurance,%20Utilities,%20and%20Other%20Fees:Renter%20Costs&g=050XX00US34027$0600000.

[22] U. C. Bureau, *Residential building permits survey dataset*. [Online]. Available: https://www2.census.gov/econ/bps/Master%20Data%20Set/.

[23] T. for Public Land, *U.s. green space parkserve dataset*. [Online]. Available: https://www.tpl.org/park-data-downloads.

[24] N. J. O. of Geographic Information Network, *New jersey addresses*. [Online]. Available: https://njgin.nj.gov/njgin/edata/addresses/#!/.

[25] N. J. O. of Geographic Information Network, *New jersey boundaries*. [Online]. Available: https://njgin.nj.gov/njgin/edata/boundaries/index.html.

[26] N. J. O. of Geographic Information Network, *Parcels and mod-iv of morris county*. [Online]. Available: https://www.arcgis.com/home/item.html?id=c8cb1abeedc54b1c89e251268de80222.

[27] Google, *Google street view static api*. [Online]. Available: https://developers.google.com/maps/documentation/streetview/overview.

[28] S. Greene and K. L. S. Pettit, "What if cities used data to drive inclusive neighborhood change?" *Urban Institute*, Jun. 2016. [Online]. Available: https://www.urban.org/sites/default/files/2022-08/2000807-what-if-cities-used-data-to-drive-inclusive-neighborhood-change.pdf.

[29] P.-Y. Nguyen, T. Astell-Burt, H. Rahimi-Ardabili, and X. Feng, "Green space quality and health: A systematic review," *International Journal of Environmental Research and Public Health*, vol. 18, no. 21, p. 11 028, Oct. 2021, ISSN: 1660-4601. DOI: 10.3390/ijerph182111028. [Online]. Available: http://dx.doi.org/10.3390/ijerph182111028.

[30] A. Letchford, *Streetview python package*. [Online]. Available: https://github.com/robolyst/streetview.

[31] N. Naik, S. D. Kominers, R. Raskar, E. L. Glaeser, and C. A. Hidalgo, "Computer vision uncovers predictors of physical urban change," *Proceedings of the National Academy of Sciences*, vol. 114, no. 29, pp. 7571–7576, Jul. 2017, ISSN: 1091-6490. DOI: 10.1073/pnas.1619003114. [Online]. Available: http://dx.doi.org/10.1073/pnas.1619003114.

[32] M. C. C. Vision, *Semantic segmentation on mit ade20k dataset in pytorch*. [Online]. Available: https://github.com/CSAILVision/semantic-segmentation-pytorch?tab=readme-ov-file.

[33] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, Mar. 1990, ISBN: 9780470316801. DOI: 10.1002/9780470316801. [Online]. Available: http://dx.doi.org/10.1002/9780470316801.

[34] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15, MIT Press, 2002. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf.

[35] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, Nov. 1901, ISSN: 1941-5990. DOI: 10.1080/14786440109462720. [Online]. Available: http://dx.doi.org/10.1080/14786440109462720.

[36] T. Evans, *Breaking barriers: A comparative analysis of affordable housing compliance and diversity in morris and monmouth counties, new jersey*, New Jersey Future, Jan. 2024. [Online]. Available: https://www.njfuture.org/wp-content/uploads/2024/01/Morris-vs.-Monmouth-Segregation-Report-1.29.24-FINAL.pdf.

[37] M. C. O. of Planning and Preservation, *Low moderate income and inclusionary affordable housing developments*. [Online]. Available: https://www.morriscountynj.gov/files/sharedassets/public/v/1/departments/human-services/2023-morris-county-affordable-housing-map-and-resources.pdf.