Synthetic Population Generation Using Census Data

by

Abu Darda
23141038
Rahat Khan
20101212
Ashabul Yamin Raad
23141088

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2023

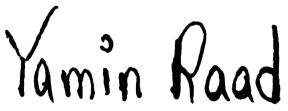
Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Abu Darda
23141038



Ashabul Yamin Raad
23141088

Rahat Khan
20101212

Approval

The thesis/project titled "Synthetic Population Generation Using Census Data" submitted by

1. Abu Darda (23141038)
2. Rahat Khan (20101212)
3. Ashabul Yamin Raad (23141088)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 23, 2015.

Examining Committee:

Supervisor:
(Member)

Dr. Mohammad Nur Yanhaona
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Rafeed Rahman
lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

An open-source population simulator is being developed to address the limitations posed by current closed-source population simulators. The close nature of these existing simulators restricts access to demographic modeling and contact tracing, hindering individuals from predicting crowd and implementing disaster management measures. Moreover, current map services only offer traffic projections without detailed land usage information. In response to these challenges, Our simulator is capable of generating synthetic populations. The simulator aims to provide projections, forecasts, and models of diverse population behaviors under specific stimuli, such as disease outbreaks, natural disasters, and significant congestion. Additionally, we are working with US census data, and the simulator primarily focuses on the geographic area of the US, considering the availability of relevant data in Western countries. By making this simulator openly accessible, it seeks to empower both individuals and states to make informed decisions and effectively plan for various scenarios.

Keywords: Synthetic Population Simulation; US census data; Public use Metadata Sample; National Household Trip Survey; open-source population simulator.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ACS American Community Servey

IPF Iterative Proportional Fitting

IPU Iterative Proportional Updating

$NHTS$ National Household Trip Servey

$PUMS$ Public Use Metadata Sample

RDF Resource Description Framework

Chapter 1

Introduction

1.1 Overview

In an era marked by the unprecedented challenges of the COVID-19 pandemic, the world witnessed a glaring need for a more advanced approach to understanding and managing activity dynamics, particularly in densely populated countries like Bangladesh, India, and Brazil. As nations struggled with the complexities of contact tracing and the instantiation of crucial actions, it became abundantly clear that a novel solution was needed. The effectiveness of vaccination campaigns could have been improved by the necessity of tracking both vaccinated and unvaccinated individuals, amplified by the economic constraints that made prolonged lockdowns unviable. Additionally, natural disasters, such as the devastating floods in Pakistan and Bangladesh and the catastrophic wildfires in Australia, underscored the urgency of comprehending population demographics in times of crisis. Although synthetic populations have been integrated into various contexts, accessing these resources has often proven infeasible, with many sources carrying significant cost metrics. Moreover, the accuracy of population simulations depends heavily on advanced datasets, which are often rare in many countries. These challenges persist in addressing population management concerns during natural disasters and in the heart of densely congested urban areas. It aspires to bridge these critical gaps by introducing an open-source population simulator that empowers governments and the public with a comprehensive understanding of demographic patterns. Grounded in the belief that access to demographic insights is a fundamental right, this initiative aims to equip individuals and authorities with the knowledge needed to make informed decisions about their living, working, and overall quality of life. Furthermore, during crises like pandemics or urban congestion, this open-source tool promises to enhance situational awareness and facilitate effective response strategies.

1.2 Research Motivation

During times of covid 19, countries struggled to trace the contacts. As a result, overpopulated countries like Bangladesh, India, and Brazil needed to learn how to prioritize areas while taking necessary action. Besides, the vaccination process was ineffective during the second and third waves because the governments needed to track the movement of vaccinated and unvaccinated people. Also, developing countries needed more time to afford a very long-term lockdown. The recent floods in Pakistan and Bangladesh and wildfires in Australia urged understanding of the population demographic during the disasters and in the future.

There have been multiple uses of synthetic populations before. For example, there are multiple instances of population synthesis, including Dortmund and Netanya, US. However, none of the sources are free and publicly accessible to the people. Besides, we need advanced datasets unavailable in most countries to make these simulations more accurate. As a result, problems with population management in times of natural disasters and highly congested cities still need to be solved, and these problems continue again and again.

We mentioned earlier that our simulator would be identical to one existing one. Our philosophy is that it is the right of the people and government to understand the demographic as a whole. The reason behind it is it fundamentally allows people and the government to choose locations more wisely regarding living, working, and having a better life. Secondly, this will allow the government and the people to be aware in times of crisis to a greater degree. Thirdly, this open source will encourage other countries to have the requirements(proper datasets and logistic support to the people) to welcome the idea of population synthesis.

1.3 Problem Statement

Population Forecast is essential for disaster management, traffic congestion management, and urban planning. Currently, services like Google Maps provide a prediction, but it does not foretell for a more extended period. Moreover, no open-source simulator forewarns anything other than traffic congestion in a geographic location. For this reason, no population forecaster allows the government or the people of underdeveloped countries to understand the demography in times of crisis, higher congestion in an overpopulated city, hospitals, and amusement parks. Besides, there is only one simulator in the world, which is among the USA and Germany, and they did not make it open source.

Understanding congestion in parks, restaurants, and other places is essential for us daily, too. In the areas of dense population, we often see these places becoming overloaded with people in crucial times. The primary reason is that people need to understand the optimum time required to complete an activity in the park or somewhere else since they do not have any forecast of the congestion of that environment. As a result, despite having capacity, we often see people gathering within a few locations despite having capacity in other places. Especially on weekends, we

often see that parks and cinema halls are overcrowded, and we need to predict which park or cinema hall will take us to a less congested area.

Our research will find answers to these two core questions:

Q1: Can we predict the activity of the population based on the attributes available in the census data?

Q2: What are the attributes that we need to know in order to predict the activity of a person in a given time interval?

While some researchers were working on creating a synthetic population, some applied IPF, and some used Monte Carlo Sampling along with IPF and IPU at first. So here, we choose to work with IPF and IPU and any method to assign individuals in the household depending on the statistical validity of the algorithms. Besides, we will use a distance function for the location assignment to match individuals from home. Though some papers used social networking for contact tracing, we will use different datasets to measure the contact estimation in different places, i.e., schools, hospitals, or other workplaces. We plan to work on more datasets and with fewer features if possible to make our simulator as flexible and accessible as we can. It will allow us to measure more realistic contact estimation rather than holistically.

Our visualization will be as simple as traditional maps available so that an average human can understand it without having any technical knowledge. Furthermore, we want to create an interface that should be easier for the user to access. To exemplify, we plan to create a simulator that will take time intervals as input and project the activity of the population in any part of the United States.

1.4 Research Objective

Prediction, Visualization and User-Friendliness are the core qualities that we want our simulator to have. Additionally, we must ensure the maximum accuracy of projection.

1.4.1 Core Objectives

We aim to create a projection that will be beneficial for the entire population. Here are our research objectives in brief:

1. Determine how much of a person's identity attributes and employment status is dependent on their activity.

2. Develop a simulator that is free for all time.
3. Create a simulator that people can access even without technical knowledge.
4. We want to ensure that no privacy is harmed during the research process and after the simulator is publicly available.
5. We want to design a simulator that doesn't require any no real-time internet access.

1.5 Thesis Structure

Our aim was to develop a simulator capable of projecting population forecasts for a specific region in the US. To achieve this, we needed to create synthetic populations through a series of processes. Initially, the data was collected from six different sources, focusing on a targeted US region. Subsequently, we performed data pre-processing, which involved employing IPF and IPU techniques to refine the data and make it suitable for the subsequent steps. For activity choice, we utilized the NHTS datasets to assign activities to each household population. By using the HERE dataset, we successfully assigned location choices to each individual and traced their contacts.

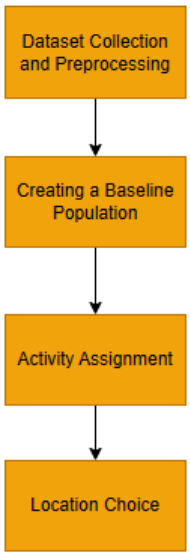


Figure 1.1: Detailed Research Methodology

1.5.1 Dataset Collection

Since we are working with Census Data, we need to collect two sets of data: Public Use Microdata Sample(PUMS) and Census Data. PUMS Data will help us to determine the attributes of 5% of the total population, while the statistics from Census Data will provide us with the marginals from each detail. Secondly, we collect the National Health and Travel Agency's dataset. The NHTS Data consists of trip samples from around 1 million people from different states of the United States. Thirdly, we will collect RDF Datasets from HERE MAP. RDF Datasets contain the longitude and latitude of Schools, Houses, Parks, and so on.

Source	**Dataset(s)**
US Census Bureau	PUMS Data and Census Data
National Health and Travel Survey	Survey of 2021
HERE Maps	RDF Datasets

Table 1.1: Used Datsets

1.5.2 Population Reconstruction

We need to create a dataset identical to the entire population of the United States. So, we must develop different households for them and assign them to a home. Baseline Population is when we simulate all the families and people within the households. We will use Iterative Proportional Fitting and Iterative Proportional Updating to implement this. Our Primary Dataset is the PUMS Dataset. We will take the marginals or targeted value from US Census Data. Census Data allows us to get the totals of the attributes for a state or the United States. Using IPF and IPU will allow us to maintain statistical validity.

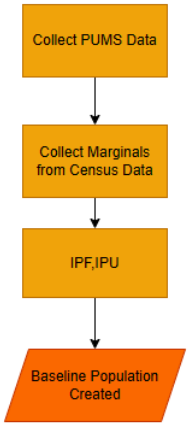


Figure 1.2: Steps of Creating a Baseline Population

1.5.3 Activity Assignment

After creating a baseline population, our challenge is to find a classification model that can predict the trips of NHTS Data at best. Our activity assignment is done as soon as we find a model with good accuracy.

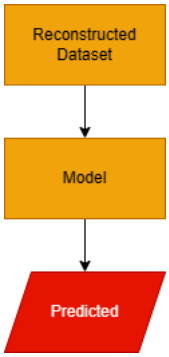


Figure 1.3: Simplified Steps of Activity Assignment

In 1.3, we can see that we will apply classification models on NHTS Data first to compare the accuracies. We will save the best model and then apply it to our baseline population. Our model will provide the trip activities as outputs.

1.5.4 Location Assignment

We will assign the individuals to appropriate locations based on their activity. For this step, we need a map where we can project the population. At first, we assign Households to the population and then according to each person's activity at any given time, we assign the nearest location available for that person for that specific time.

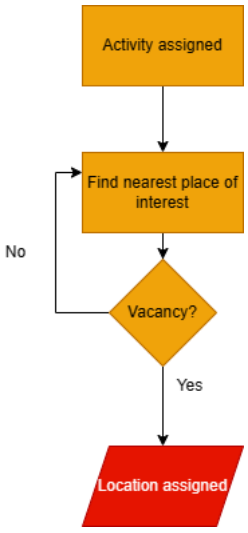


Figure 1.4: Steps of Location Choice

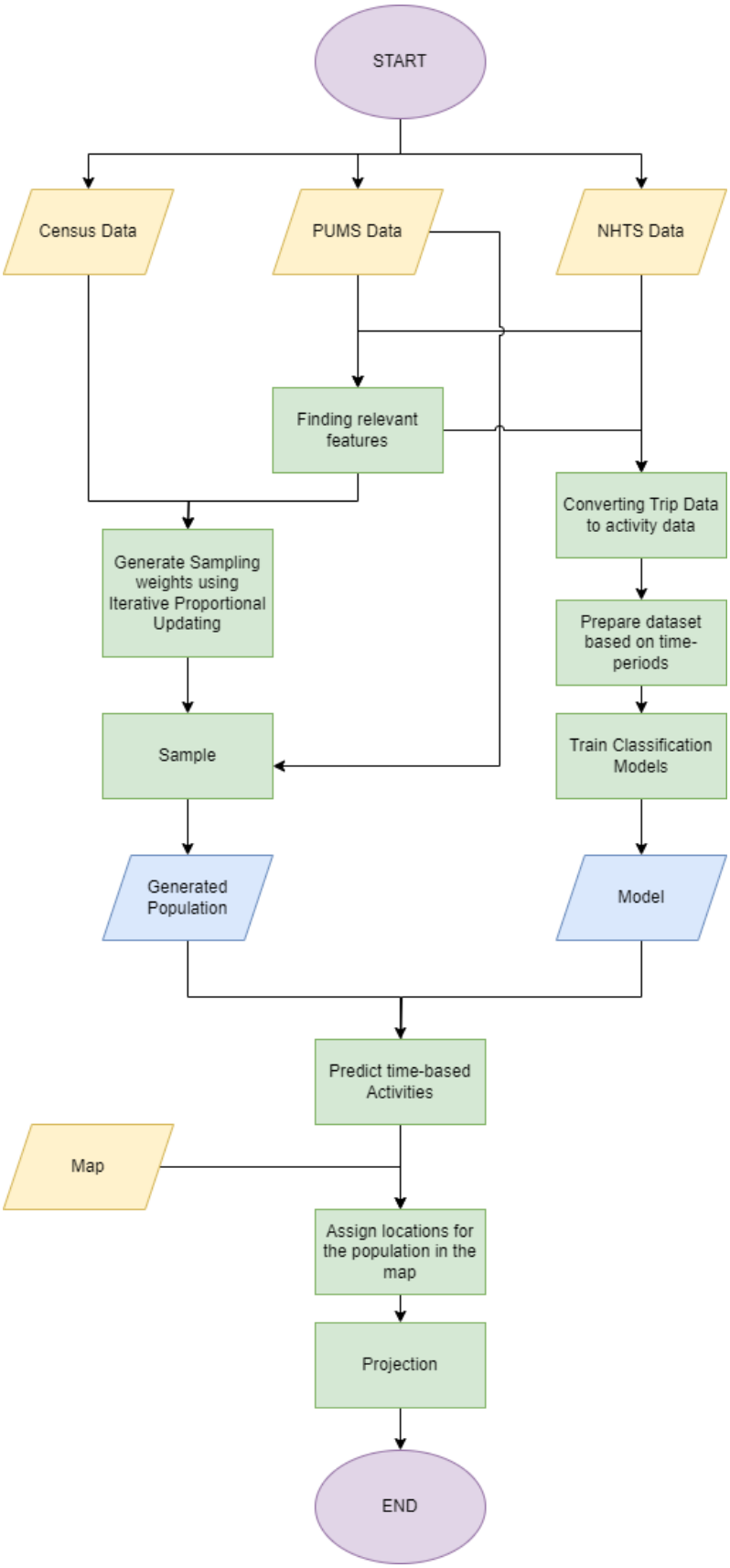


Figure 1.5: Detailed Workmap of the Research

1.6 Contributions

The simulation involves multiple stages. Initially, there were no dynamic Python modules to implement any of the algorithms. Therefore, the additional contributions of this research are:

1. We created functions for Python that can run Iterative Proportional Fitting and Iterative Proportional Updating directly from PUMS Data and Census Data Marginals. In other words, the baseline population can be directly implemented through our code.
2. We extracted relevant census data features that can work as the core for predictions.
3. we predicted heuristics for assigning household and activity locations for the population.
4. Our simulator is open source. All the codes are available in Github.
5. We provided a brief case study of a small sample of the United States population.

Chapter 2

Literature Review

Our research consists of three different stages. To begin with, we reconstruct a population from PUMS Data. Next, we assign activity. Lastly, we assign them to the appropriate location. We came across numerous academic papers that covered multiple algorithms for population reconstruction, acitivity assignment and location assignment.

2.1 Background Study

In each phase of our research, we encountered multiple studies and algorithms and came across numerous academic papers that covered various subjects, including Iterative Proportional Fitting and Iteractive Proportional Updating for population reconstruction, Activity assignment using social networks, and activity-based simulation models. Additionally, three prominent papers on closed-source simulators explore creating a synthetic population.

2.1.1 Population Reconstruction

We studied Iterative Proportional Fitting and Iterative Proportional Updating to create a module that helps us to generate a baseline population. Some papers used Monte Carlo Sampling for assigning the individuals into the households. In 1996, R. Beckman et al. introduced Iterative Proportional Fitting to generate a population from a small portion of its sample and the marginal values [4]. Later, Xin Yee et al. introduced Iterative Proportional Updating [10]. The drawback of IPF is that it only deals with the population distribution and ignores the relationship between the population and its distribution within the households.

Iterative Proportional Fitting

Deming and Stephan (1940) first used IPF in demography in 1940 [1]. It is a process for changing a data table's data cells to add up to chosen totals for the table's columns and rows. Researchers recognize it as the basic procedure for IPF. We need to work for more than two or three dimensions. The reason for the circumstance is that the dimension for our datasets is much higher than just two or three.

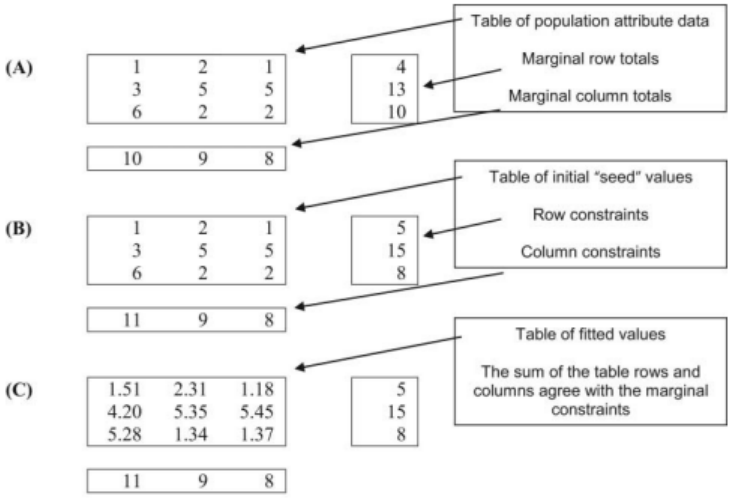


Figure 2.1: IPF Algorithm (Taken from [28])

The first step involves proportionally adjusting each row of seed cells to make them equal to marginal row totals. This process divides each cell by the row's actual sum before being multiplied by the marginal row total. In the next step, they adjust each column of previously row-adjusted cells to match the marginal column totals. These procedures are continued until we obtain the specified rate of convergence [9]. The most commonly used iteration method is adjusting the cell with respect to the ratio of row constraints and column constraints [17].In this case, each step is divided into two parts: adding the ratio of row and row constraints, and then adding the ratio with column and column constraints.IPF is a mathematically proven and widely used procedure [11].

Iterative Proportional Updating

Iterative Proportional Updating is an algorithm researchers use to match personal data with household data by step-by-step iterations simultaneously. So, after we update the individual and household data with the constraints from census data, matching the number of persons with the number of households is essential because assigning individuals to the home is more accurate [3], but IPF fails to address that. In this case, we have an initial dataset with some counts of households based on their type and individual counts. Then, it starts iterating based on the given constraints for household and individual data and a scalar to calculate convergence in each step. In each phase, the weights are adjusted according to the constraints. Also, it alters

the weighted sum in each iteration. Finally, the iteration breaks when it achieves a specific convergence value and stores the final result. These weights determine the total number of samples that we need to take from each type of household and person combination [10].

IPU was first used by Xin Yee et al. [10] in 2009 for creating a synthetic population[4]. Next, Guo and Bhay(2014) used it for simulations at the block group level of the Maricopa County region of Arizona[14].

2.2 Activity Assignment

Our activities are divided in multiple classes 4.1. Therefore, we applied classification models for the prediction. In US, activity based models are still in its former phase [35].

2.2.1 Classification Models

Our targeted feature was divided into multiple categories. Therefore, we used numerous classification models. Classification models are a type of model that predicts the variety of the targeted segment. It is a supervised learning approach where the targeted features are usually divided into labels. Our targeted datasets are mainly divided into multiple categories. For example, when we work with Trip Purpose Summary('WHYTRP1S'), we get 10 categories as our purpose of the Trip. Classification models are good in terms of predicting the activities. Within the datasets of the National Household Travel Survey (NHTS), the recorded activities are intricately divided into various distinct classes. This division arises from the need to capture the nuanced behaviors and choices of individuals when it comes to travel and daily activities. As a result, in our research, we embark on a comprehensive exploration of multiple classification models. The primary objective behind this approach is to meticulously assess and evaluate these models, aiming to identify the most robust and accurate predictor among them. By employing this rigorous methodology, we endeavor to uncover the model that not only excels in its predictive capabilities but also sheds light on the intricate patterns and dependencies inherent in the NHTS data.

Logistic Regression

Logistic regression is a widely used statistical method for binary classification tasks. Despite its name, it is a classification algorithm, not a regression algorithm.

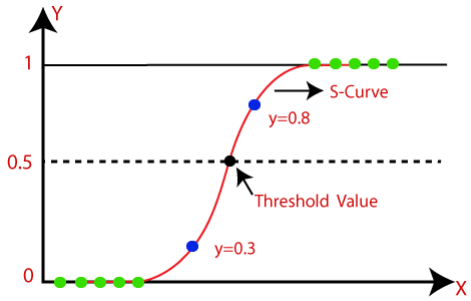


Figure 2.2: Classification Technique of Logistic Regression(Taken from [32])

It models the probability of a binary outcome (0 or 1) based on input features. The algorithm uses a logistic or sigmoid function to transform a linear combination of features into a probability score between 0 and 1[7]. If the probability is above a certain threshold (usually 0.5), the prediction is assigned to class 1; otherwise, it belongs to class 0. Logistic regression is interpretable, computationally efficient, and works well with linearly separable data. It finds applications in various fields, including medical diagnosis, spam detection, and credit risk assessment[6].

Logistic regression is a fundamental algorithm in machine learning and statistics due to its simplicity and interpretability. It is beneficial when dealing with binary classification problems, where the goal is to predict one of two possible classes. The model is based on the logistic or sigmoid function. It smoothly maps the linear combination of input features to a probability value, generating a confident prediction between 0 and 1.

Despite its simplicity, logistic regression can be surprisingly robust when the data is linearly separable. It is easy to implement, computationally efficient, and requires relatively little data preprocessing. Additionally, it can handle both numerical and categorical features with appropriate encoding.

However, it's important to note that logistic regression assumes a linear relationship between the features and the log odds of the target class. This means it may need to perform better on datasets with complex nonlinear relationships, as it cannot capture intricate interactions between features[31].

To extend logistic regression to handle multi-class classification, one can use techniques like one-vs-rest or multinomial logistic regression.

While logistic regression has its strengths, it's essential to evaluate its performance and consider using more complex models like decision trees, random forests, or deep learning approaches when dealing with more intricate datasets. Logistic regression remains a valuable tool in the data scientist's toolkit, especially when interpretability and simplicity are crucial for understanding the relationship between input features and binary outcomes.

Random Forest

Random Forest is an ensemble learning algorithm that has gained significant popularity in machine learning and data science. It is an extension of decision trees and operates by constructing multiple trees on randomly sampled subsets of the training data and features. The final prediction in Random Forest is determined by aggregating the individual forecasts from these trees, leading to improved accuracy and robustness. This algorithm's strength lies in handling complex, high-dimensional datasets and addressing issues like overfitting and variance inherent in individual decision trees. Moreover, Random Forest can perform well on classification and regression tasks, making it a versatile and widely used tool for various real-world applications. The algorithm's interpretability, feature importance assessment, and scalability further enhance its appeal, making it an essential component of the modern machine-learning toolkit.

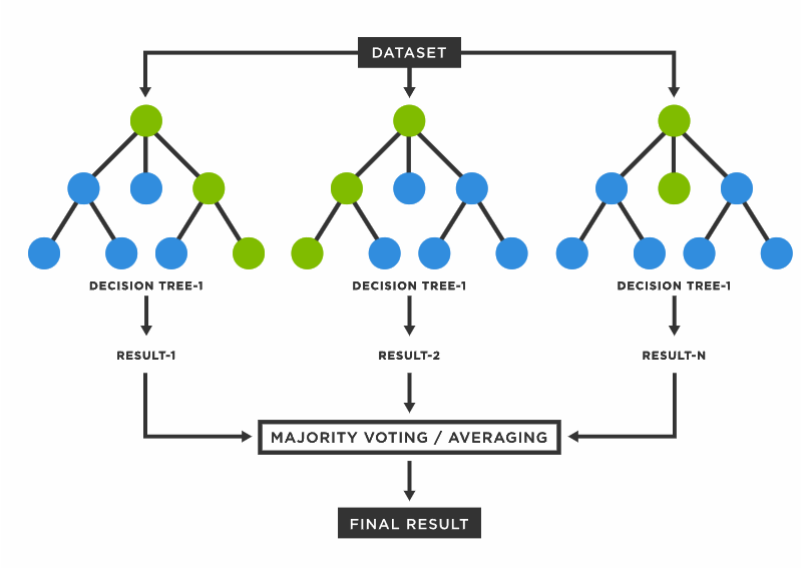


Figure 2.3: Random Forest Steps(Taken from [36])

The algorithm operates through bootstrapping, where random subsets of the training data are drawn with replacements for each tree's construction. Additionally, at each split in the tree, only a randomly selected subset of features is considered, introducing variability in the learning process. Through this mechanism, Random Forest harnesses the principle of "wisdom of the crowd," as the collective decisions of diverse trees result in a more robust and reliable model. The final prediction is obtained by averaging the outputs in regression tasks or employing majority voting in classification tasks. This approach enhances the algorithm's ability to handle high-dimensional data, handle nonlinear relationships, and provide insights into feature importance. Despite its strengths, Random Forest may suffer from higher computational costs due to its ensemble nature. Nevertheless, its widespread adoption across various domains underscores its efficacy and appeal in contemporary machine-learning applications.

Decision Tree

The decision tree is a non-parametric and interpretable machine learning algorithm widely used for classification and regression tasks. The algorithm constructs a tree-like model in a hierarchical structure, where each internal node represents a decision based on a specific feature, and each leaf node corresponds to a class label or a predicted value. The critical objective of decision tree learning is to partition the feature space into regions that are as homogeneous as possible regarding the target variable. The splits are determined based on measures such as Gini impurity or information gain for classification and mean squared error reduction for regression. Decision trees are advantageous due to their simplicity, ease of interpretation, and ability to handle numerical and categorical data. However, they may suffer from overfitting, especially when they grow deep, necessitating pruning techniques or ensemble methods like Random Forest to improve generalization performance. Despite this limitation, decision trees remain valuable in machine learning, particularly in scenarios where interpretability and insight into the decision-making process are crucial.

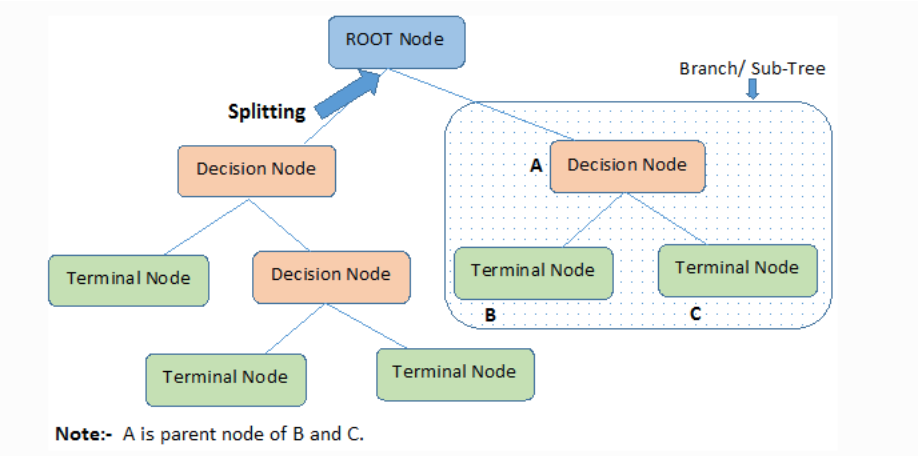


Figure 2.4: Decision Tree Steps(Taken from [33])

The decision tree algorithm consists of the following steps. Firstly, the algorithm starts with the entire dataset as the tree's root node. It then evaluates different features and their thresholds to find the best split that maximizes the information gain or reduces impurity. The dataset is divided into subsets based on this split, and the process is recursively repeated for each subset until a stopping criterion is met, such as reaching a maximum tree depth or having a minimum number of samples in a node. During this process, the algorithm assigns a class label or a predicted value to the leaf nodes. To prevent overfitting, pruning techniques can be applied to simplify the tree by removing branches that do not significantly contribute to the model's performance. The resulting decision tree provides a hierarchical set of rules that can be used for classification or regression tasks and offers interpretability and insights into the decision-making process.

Gradient Boosting

Gradient Boosting is an ensemble learning method in machine learning[18]. It involves iteratively refining weak base models, typically decision trees, to create a robust predictive model. The algorithm accentuates the importance of data points with previous prediction errors at each step, enabling subsequent models to prioritize these instances. This iterative approach efficiently amalgamates multiple weak models into a potent composite model. Gradient Boosting exhibits superior predictive accuracy by minimizing residual errors and adeptly captures intricate data patterns, rendering it a valuable tool for regression and classification challenges.

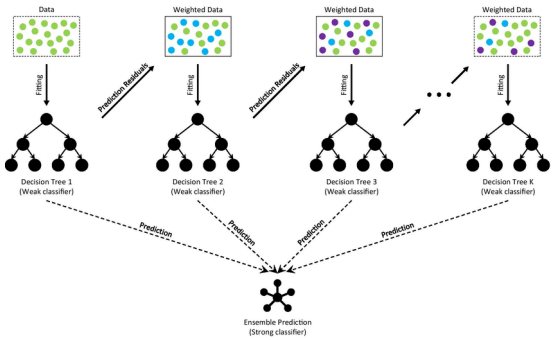


Figure 2.5: Gradient Boosting Architecture(Taken from [19])

Gradient Boosting operates through a sequential and iterative optimization process to construct an enhanced predictive model. A weak learner, often a decision tree with limited depth, is initially fitted to the data[13]. Subsequently, the algorithm identifies the discrepancies between the weak learner's predictions and the actual outcomes, focusing on instances where errors occur. The subsequent step involves fitting a new weak learner to the residual errors, emphasizing the misclassified data points. This learner attempts to rectify the previous model's deficiencies by capturing the nuances associated with these challenging instances. This process is repeated iteratively, each time generating a new model that aims to correct the errors of its predecessors. The final predictive model emerges from the cumulative contributions of these individual learners. Gradient Boosting optimally combines these weak models by systematically addressing prediction errors, yielding a potent ensemble model with a heightened capacity to capture intricate data relationships and produce accurate predictions.

Naive Bayes

Naive Bayes is a probabilistic classification method rooted in Bayes' theorem. It assumes that features are conditionally independent, which is often unrealistic but simplifying. It calculates the posterior probability of a class-given input feature by multiplying the conditional probabilities of each feature given the class. It results in a predictive model that estimates the most probable class label for a given set of features [20]. Despite its "naive" assumption, Naive Bayes excels in tasks like text categorization and sentiment analysis due to its efficiency and ability to handle

high-dimensional data. It is valuable for quick classification tasks but can suffer when strong feature dependencies exist [15].

First, it requires labeled data for training. Data preprocessing follows, involving cleaning and converting features into numerical format. Next, it calculates the class priors by determining the prior probabilities of each class. Then, it calculates the likelihoods, representing the probability of observing each feature given a class. Bayes' theorem determines posterior probabilities d for each class based on the prior probabilities and feature likelihoods. These posterior probabilities are compared, and the class with the highest probability is assigned to the data point. Finally, the model's performance is evaluated using accuracy, precision, and recall metrics, and we can fine-tune it for better results [5].

2.3 Location Assignment

There are multiple works on models that work on activity based predictions. The researches include predicting next locations using social media checkin, R. Beckman et al. [12] worked on location choice using census data and many more.

A recent work from Luo et al. [24] for location assignment using social media check-ins. This research consists of four components:

- **History Encoder:** This component encodes the user's historical check-in sequences to extract mobility patterns.
- **Query Generator:** This component generates a query embedding that represents the user's current state and preferences.
- **Contrastive Learning:** This component enhances the query embedding by leveraging contrastive learning, a technique that learns to distinguish between positive and negative pairs of examples.
- **Preference Decoder:** This component decodes the enhanced query embedding to generate a personalized next-location prediction.

The POI Former model, as described in the paper, is trained end-to-end using a contrastive learning objective. The model is given a set of positive and negative examples during training. The positive pairs are pairs of locations that the user has visited in the past, while the opposing pairs are pairs of places that the user has not seen in the past. The model distinguishes between positive and negative teams by minimizing a loss function.

Once the model is trained, it can be used to generate personalized next-location predictions for users. The model is given the user's current location and historical check-in sequences to do this. The model then encodes the historical check-in sequences to extract mobility patterns and generates a query embedding. The query embedding is then enhanced using contrastive learning and decoded to develop a personalized next-location prediction.

POIFormer has been shown to outperform state-of-the-art next-location prediction models on a variety of public datasets. It is a powerful tool that can be used to develop personalized recommendation systems for various applications, such as location-based services, travel planning, and social media.

2.4 Related Works

In 1996, Beckman made the first and most prominent approach to creating a synthetic population [4]. It had two parts: adjusting a multi-way demographic table using IPF and creating a synthetic population. It does not deal with activity assignments. First, a proportional multi-way demographic table is estimated. It is calculated via proportional fitting iteratively. Then, A synthetic population of households is derived from the PUMS to match the proportions in the estimated table. This paper introduced a lot of essential ideas to population simulation. First, this paper uses Iterative Proportional Fitting for the PUMS data for the first time so that we can create the dataset for household details and individual details for working in a geographic location [12]. To illustrate, we cannot work with PUMS datasets since it is a sample of randomly chosen people and does not contain data for every person in the geographic location [25]. We use marginals to create a dataset that finally represents the prediction of the whole population. Secondly, the dataset we get after running Iterative Proportional Fitting will likely be sparse. For For this reason, this paper shows how we can solve this problem by adding a minimal number before running Iterative Proportional Fitting.

This paper used PUMS data to simulate household details by providing only a proportion of persons and vehicles. In this case, they use a proportion of household details and vehicle status and fit it with IPF. Later, it compares the proportions' ratios and Mean Absolute Deviation using IPF and PUMS. Also, this paper allows the independence of any statistical method that suits. However, as we can see, this paper does not deal with any activity assignment and location choices. For this reason, unlike the other two articles, it fails to provide a sophisticated synthetic population because it only focuses on household data and simulation. In May 2003, the first of the synthetic population models was introduced at the 8th International Conference on Computers in Urban Planning and Urban Management (CUPUM) [8].The the simulation demonstrated household population, using IPF and Monte-Carlo Sampling for data preprocessing [4].In the next layer, they examine the household workers and assign a zone for each operational activity. Then, they created two synthetic models- Netanya, Israel, and Dortmund, Germany. Firstly, IPF adjusts the probability of household size according to the household's head's age.[5] Then, they use Monte Carlo Sampling to get household details by adding corresponding features. After the iterations, details of the household, i.e., household head, non-adult members, number of cars, incoming persons. Let us take the generated simulation of Netanya, for example. It produced a population of One hundred fifty-nine thousand people live in around 50,000 homes for Netanya. The head of the household was chosen to create a home since many household elements depend on that family member's features. The head of the home was then selected based on age, gender, religion, and level of education. Next, the data was used to estimate

the size of the household. There was no need to add any extra people if a one-person family was chosen. If it had chosen a multi-person home, each additional member was chosen until the entire household had been formed. The zone's residential site was then decided. A micro-location, including the row and column coordinates of a raster cell, made up the housing location. The number of earners was chosen next, followed by the household income. All of this information was used to estimate the number of cars. Finally, a workspace was allocated to each employee. The simulator kept creating new ones until all of the zone's households were established.

Later, the simulation of Dortmund was projected. The main drawback of Netanya's synthetic population is that it is only limited to household simulation since the data are less featured, and the location datasets of Dortmund were more liberal than Netanya's [16]. Because it allowed them to project all types of land usage, i.e., business, residential, and non-residential zones. For this reason, the simulation was more accurate and useful.

Compared to Netanya's household, the synthetic population of Dortmund reflects individual actors in the form of households and household members. In addition to the number of vehicles, it distinguishes the eleven categories of vehicles, ownership of monthly season tickets produces car-sharing memberships, and generates a monthly mobility budget. Compared to Netanya portrays each individual in this city by employment status and driver's license possession [16] . While the revenue is shown on a household level in Netanya, each individual's income is created independently in the Dortmund area. For this reason, this simulation provides a better demonstration [4]. As we have mentioned earlier, there is a closed-source we are working on regarding population synthesis titled 'Generating a synthetic population of the United States.' This paper focused on four aspects: Baseline Population Synthesis, Activity Assessment, Location Choice, and Contact Estimation. The four layers are the essential roadmap of our thesis. First, baseline Population Synthesis focuses on creating a primary dataset to an observable dataset using IPF and IPU algorithms. Then, IPF scales the PUMA dataset into the PUMS dataset to become usable for the whole region. Finally, IPF and IPU are mathematically proven and used to scale individual and household datasets. In the next layer, they assign activities to the individuals from those households. They collect the data from the National Household Travel Survey(NHTS), Dun Bradstreet (DB), and HERE (formerly NAVTEQ)[4]. The paper denotes that Hausdorff Distance is used to calculate person-person distance, preferring it over Euclidean and Mahalanobis distance. Finally, the researchers identify the worst person-person distance as household distance. In short, selecting a survey household, best-matching individuals in the home, and assigning activities to each individual are the three essential coatings of this process. Kristian Lum et al. [23] worked on another optimized activity assignment method. It consists of three simple steps:

1. Find the closest household to the synthetic household from the survey.
2. Find the most immediate individual to synthetic individuals in that household.
3. Find the most immediate individual to synthetic individuals in that household.

The similarity of a household is calculated by two metrics: probability and

the minimum distance[11]. Hausdorff Distance calculates the distance between two households. Here, the researcher prioritizes Hausdorff Distance over Euclidean Distance, Manhabolis Distance, and Fitted Value Approach because Hausdorff Distance covers most of the covariates. It is an important method because it simplifies activity assignments by allowing calculations of real-world activity factors from a survey. Secondly, the activities are taken from statistics rather than used as a sequence over time.

Next, They assigned locations for the individuals. Location choice means assigning locations probabilistically to the selected individuals. HERE (formerly NAVTEQ) datasets help to count the distance between households and the preferred areas of those individuals [11].

They use Traffic Analysis Data(TAZ) to optimize the calculations to avoid computing population all over the location. In the first place, the location of the TAZ is chosen. Then, they assign the capacity of all the TAZ locations by summing up that particular geographical area. Contact estimation is generated by combining the location choice and activity time. After this process, it creates a table of vertices, denoting the people and edges representing the social network.The method of contact estimation of Delhi and Los Angeles showed a high level of accuracy using this method [2].

Steps	**Paper 1: Creating Synthetic Baseline Populations by Richard J. Beckman et al.**	**Paper 2: Creating a Synthetic Population by Rolf Moeckel et al.**	**Paper 3: Generating a synthetic population of the United States (2015) by Abhijin Atigya et al.**
Algorithms Used for Baseline Population	IPF	IPF, Monte Carlo Sampling	IPF and an algorithm for household assignment
Activity Assignment	This paper does not deal with activity assignments.	Ignores travel behavior and commercial data	Works with employment status, travel behavior, healthcare population, and Land Use Data
Location Choice	This paper does not deal with location choice.	Only assigned to workplaces	It deals with amusement parks, healthcares, educational institutions, and workplaces
Contact Tracing	This paper does not deal with contact tracing.	This paper does not deal with contact tracing.	It shows the individuals along with other individuals they have been contacted with.

Table 2.1: Comparison of Papers

Chapter 3

Population Reconstruction

The initial goal of this research is to build a population that can replicate an actual population of any given area. This population has to have near near-perfect ratio of certain household types and certain individual types in those households. For example, if, in any area, there are H amount of households with features H_1, H_2, and H_3 and on average in those houses, there are I_1 number of persons with p_1, p_2 and p_3 feature and I_2 number of persons with p_4, p_5, and p_6 features, our generated population have to match those numbers. Obviously, perfectly matching those numbers is impossible but we tried to match them as close as possible in our work.

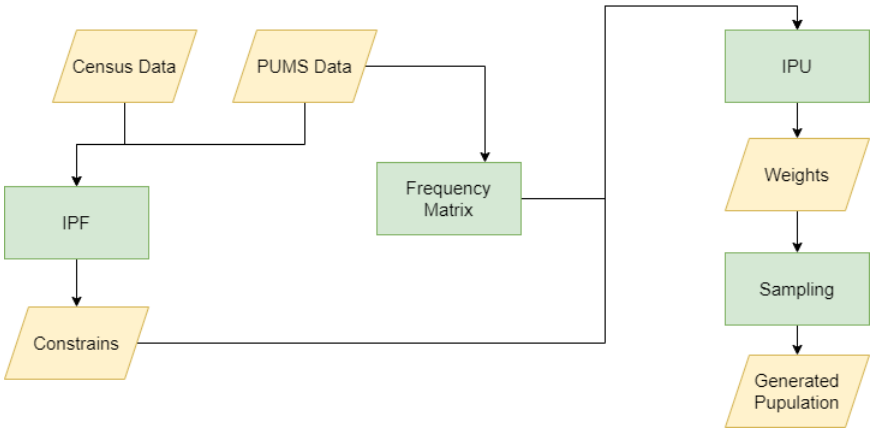


Figure 3.1: Population generation Workflow

3.1 Introduction to Datasets

To create the large instance, we have two datasets initially: Census Data, and Public-Use Microdata Samples(PUMS) dataset.[26]

3.1.1 Census Data

The term "Census dataset" refers to an extensive compilation of data acquired through a national census. The Census Bureau of the United States conducts the survey. [26] The census aims to systematically collect a wide range of demographic, social, economic, and housing information about the population residing within a specific geographic area. This dataset includes crucial aspects like age, gender, race, education, employment, income, and housing.

The US Census Bureau collects two types of datasets: the Decennial Census and the American Community Service. The Decennial Data is the major instance where we have information regarding Gender, Race, Age, Ethnicity, and Housing. However, the ACS is an ongoing survey that comprehensively gathers information on demographics, social factors, economic indicators, and housing statistics for the American populace. In contrast to the decennial census that takes place every ten years, the ACS continually collects data throughout the year. By sampling a smaller portion of the population each month, it accumulates a larger and more comprehensive sample size over the ten years. This approach ensures that the ACS offers a detailed and accurate representation of the population, thereby enhancing the reliability and robustness of the collected data. Covering millions of households across the United States, the ACS addresses a broad spectrum of topics, including age, gender, race, ethnicity, education, employment, income, and housing conditions. The survey provides estimates for various geographic levels, ranging from states and counties to cities and smaller localized units. As a result, it serves as a valuable resource for understanding local communities, conducting research, and gaining insights into the evolving societal and economic dynamics.

In this research, we are using the Census Data of 2021. The Census Data consists of many parts but out interest is in four main parts: Demographic, Economic, Housing, and Social. For sampling, the Census Bureau has a sample of 6.5 million households while only 1 in every 20 households has been sampled. The total number of instances is available to the public.

3.1.2 PUMS Data

In the United States, the Census Bureau conducts a national census, known as the United States Census, which gathers data on the entire population. Additionally, the Census Bureau offers supplementary datasets, one of which is the Public Use Microdata Sample (PUMS) dataset.

The Public Use Microdata Sample (PUMS) dataset is a state or region-based dataset, containing anonymized individual-level records. PUMS data allows researchers and analysts to access detailed information about individuals and households while ensuring data confidentiality. It provides a representative sample of the population, enabling customized analysis and research on various socio-economic characteristics.

There are two types of Public Use Microdata Samples: the 1-year PUMS and

the 5-year PUMS. We are using the 5-year PUMS which was conducted in 2021. Our study utilized the specific Public Use Microdata Sample (PUMS) data for the state of North Dakota. The PUMS dataset, derived from the American Community Survey (ACS), provides anonymized individual-level information that allows for in-depth analysis of various socio-economic factors.

Unlike aggregated statistics, the PUMS data offers a more detailed perspective by providing individual-level records for a representative sample of households within North Dakota. This dataset includes comprehensive information on demographics, education, employment, income, housing, and other relevant variables. By examining the percentage distribution of ages, one can gain a deeper understanding of the population's needs, including those related to education, healthcare, and transportation. For instance, a high percentage of children in the population indicates a demand for additional schools and childcare facilities to cater to their needs. This information enables effective planning and allocation of resources to meet the specific needs of different age groups within the population.

We are working with some limited features of PUMS Data since our travel dataset has only a few in common. PUMS data has many classes within the elements that project the details of a person.

Feature Type	Feature Name	Values	Total Length of Classes
Household	Number of persons in household	0-24	17
Household	Vehicle count	0-12	4
Household	Family Income	0-200,000+	11
Household	Number of workers in family	0-14	4
Person	Age	1-99	5
Person	Sex	male/female	2
Person	Educational Attainment	Graduate, Bachelor, college, High-school, below-high	5
Person	Race	Asian, Black, Native, Pacific-islander, white, other	6

Table 3.1: Household and person features with their size

3.2 Methodoloy

The traditional Iterative Proportional Fitting (IPF) 3.2.3 technique has been used by the majority part of population synthesis algorithms. But, as mentioned by Xin Ye [10] if we apply it to a dataset where there is more than one level meaning a person can belong to two different levels, it creates a problem. Here, with our data, we have both household and person characteristics. So, if we try to estimate the

household and person level joint distribution it will result in two different sets of weights matching household or person distributions. And, if we force person-level distribution weights to match household-level distribution weights, the generated population will not be able to reflect a realistic population.

3.2.1 Combining Person and Household Data

The first step for generating the population is to combine households and their respective Person data. To achieve that goal, we need to build a frequency matrix. The frequency matrix would be a two-dimensional matrix containing information about each person type in each household type. The Public Use Metadata Sample (PUMS) [26] provides separate datasets for each category. However, which person belongs to which household can be easily generated from that person's data. Although there is no unique identifier for each person, each House has a unique identifier and each person has the identifier of which household he is from.

The size of the matrix would be $N \cdot m$, where N is the size of the total households in the sample and m is the summation of household type combinations and person type combinations. By combination we mean, from table 3.1, there are 5, 2, 5, and 6 classes for age, sex, educational attainment, and race respectively, so there would be $5 \cdot 2 \cdot 5 \cdot 6 = 300$ combinations for persons. The procedure is the same for households as well. Although we would have to exclude combinations that have no instances in the initial dataset. Once we have the initial frequency matrix built, we just iteratively update the counts for each person to its corresponding feature combination. A demonstration of the following algorithm is provided in table 3.2.

Algorithm 3.1: Algorithm to combine Household and person-level data

```
for each house do
    create column for the feature combination of
    this house and set value to 1 ;
    all other household columns will have value 0 ;
    for each person in this house do
        if feature combination of this person already exists then
            increase the count;
        else
            create column for the feature combination of
            this person and set count to 1;
```

Household data		
HOUSE_ID	Area	income category
1	urban	1
2	urban	2
3	rural	2

Person data		
HOUSE_ID	race	sex
1	white	male
2	asian	female
2	asian	female
2	asian	male
3	white	male

Combined data						
ID	urban_1	urban_2	rural_2	white_male	asian_male	asian_female
1	1	0	0	1	0	0
2	0	1	0	0	1	2
3	0	0	1	1	0	0

Table 3.2: Demo frequency matrix

3.2.2 Joint Distributions of Household and Person type Constrains

Now that we have the combined Household and Person level data, our next goal is to get the joint distributions of those Household and Person types. To achieve that we would apply the classical IPF method [4] in census data. We will apply the procedure for both person and household columns separately. It will generate the sums of each household and person type combinations. And according to these constraints, we will proceed to generate the weights for each household.

3.2.3 Iterative Proportional Fitting

Iterative Proportional Fitting or raking in survey statistics is a procedure that is widely used in population generation techniques. Here, the goal is to find a matrix M of size $n.m$ that is the closest to another matrix Z such that the row and column marginals of M get as close as possible to two column vectors X of size n and Y of size m

So, for a two-dimensional data,

$$\sum_i M_{ij} = X_j \tag{3.1}$$

$$\sum_j M_{ij} = Y_i \tag{3.2}$$

$$Minimize \frac{\sum (M_{ij} - Z_{ij})}{Z_{ij}} \tag{3.3}$$

Now there are two main methods of achieving this goal as described by W. Edwards Deming and Frederick F. Stephan [1]

- The classical IPF

- factor estimation method

The classical IPF

For ease of demonstration of the algorithm, we will consider a 2D data. The process for multidimensional data is the same. This algorithm iteratively adjusts cell values to satisfy target row marginals X and then column marginals Y until it converges. The steps are as follows:

Algorithm 3.2: Classical IPF

$M_{ij}^0 \leftarrow Z_{ij}$;
for $k = 1$ *to max_iter* **do**
  if *L2-norm distance of* M *and* $Z > \epsilon$ **then**
    break;
  for $j = 1$ *to* M **do**
    for $i = 1$ *to* N **do**
      $M_{ij}^k \leftarrow \frac{M_{ij}^{k-1} \cdot Y_i}{\sum_j M_{ij}^{k-1}}$;
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* M **do**
      $M_{ij}^k \leftarrow \frac{M_{ij}^{k-1} \cdot X_j}{\sum_j M_{ij}^{k-1}}$;

For dimensions more than 2, in each iteration, we do weighted normalization for each dimension according to that dimension's target marginals.

The following table 3.3 shows how the cell values get changed in each iteration until the row and column marginals converge to the target row and column marginals.

Iteration 0						
	1	2	3	4	TOTAL	TARGET
1	40	30	20	10	100	150
2	35	50	100	75	260	300
3	30	80	70	120	300	400
4	20	30	40	50	140	150
TOTAL	125	190	230	255	800	
TARGET	200	300	400	100		100

Iteration 1						
	1	2	3	4	TOTAL	TARGET
1	60.00	45.00	30.00	15.00	150.00	150
2	40.38	57.69	115.38	86.54	300.00	300
3	40.00	106.67	93.33	160.00	400.00	400
4	21.43	32.14	42.86	53.57	150.00	150
TOTAL	161.81	241.50	281.58	315.11	1000.00	
TARGET	200	300	400	100		100

Iteration 2						
	1	2	3	4	TOTAL	TARGET
1	74.16	55.90	42.62	4.76	177.44	150
2	49.92	71.67	163.91	27.46	312.96	300
3	49.44	132.50	132.59	50.78	365.31	400
4	26.49	39.93	60.88	17.00	144.30	150
TOTAL	200.00	300.00	400.00	100.00	1000.00	
TARGET	200	300	400	100		100

Iteration 3						
	1	2	3	4	TOTAL	TARGET
1	64.61	46.28	35.42	3.83	150.13	150
2	49.95	68.15	156.49	25.37	299.96	300
3	56.70	144.40	145.06	53.76	399.92	400
4	28.74	41.18	63.03	17.03	149.99	150
TOTAL	200.00	300.00	400.00	100.00	1000.00	
TARGET	200	300	400	100		100

Table 3.3: Demo IPF (Taken from [27])

Factor estimation method

The idea for factor estimation is to find factor vectors A and B such that $M_{ij} = A_i \cdot B_j \cdot Z_{ij}$. Initially, all values in A and B are 1. and in each iteration, we update the values. The steps are as follows:

Algorithm 3.3: IPF using factor estimation

initialize A and B with size N and M with values 1 in them.;
for $k = 1$ *to max_iter* **do**
  if *L2-norm distance of* M *and* $Z > \epsilon$ **then**
    break;
  for $j = 1$ *to* M **do**
    for $i = 1$ *to* N **do**
      $B_j^k \leftarrow \frac{Y_j}{\sum_i Z_{ij} \cdot A_i^{k-1}}$;
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* M **do**
      $A_i^k \leftarrow \frac{X_i}{\sum_j Z_{ij} \cdot B_j^{k-1}}$;

Applying the IPF technique to census data, we get the marginal values alongside the frequency matrix of Household and Person data. Table 3.4 shows how our demo combined data may look like after getting the IPF constraints.

ID	urban_1	urban_2	rural_2	white_male	asian_male	asian_female
1	1	0	0	1	0	0
2	0	1	0	0	1	2
3	0	0	1	1	0	0
Constrains (IPF)	4	5	3	7	5	9

Table 3.4: Combined Data with IPF constraints

3.2.4 Generating weights for each household

From the previously generated combined dataset and joint distribution constraints, our next goal is to generate weights for each household that would be used for sampling. The weights have to be calculated in such a manner that the sum of the product of weights and household and person type values for each household is close to the constraints.

$$Constaints_i \approx \sum_{j}^{j=N} frequency_{ij} \cdot weights_j \quad \text{N=total houses} \tag{3.4}$$

The procedure begins by setting weights for each household to 1. Then in each iteration, for each household or person type, we adjust the weights for each of the households one by one. Here we need the weighted sums of each feature as well. As initially all the weights are set to one, the weighted sums would be just the sums of counts of each type. Then, in one iteration, first, for each feature or type, we

calculate a variable *mulpilier* 3.5.

$$multiplier = \frac{constraint}{weighted\_sums} \tag{3.5}$$

Then, we update all the weights by multiplying them with the multiplier. And, calculate the new updated weighted sums. We do this for each feature or type in every iteration until convergence. The convergence condition here is if, for any two consecutive iterations, the change of δ is too low. 3.6

$$\delta \leftarrow \sum_{j}^{m} \left[\left| \frac{(\sum_{i}^{N} frquency\_matrix_{ij} \cdot weight_i - constraints_j)}{contraints_j} \right| \right] \tag{3.6}$$

Some exceptions

As described by Xin Ye, there can be two major problems while running this algorithm.[10] First, if there is any zero value in the frequency matrix. If we run the algorithm as it is while calculating weights, it would make the weight zero, and for all the subsequent iterations, the value for this weight will remain zero as well. Having zeroes in the frequency matrix is inevitable as all the houses and individuals will not certainly be of the same type. To tackle this problem, Beckman proposed to replace the zeroes with a small value. [4] But doing so in our approach may introduce a bias. [14] So instead, whenever we get a zero in the frequency matrix, we will just carry the previous weight instead of updating it.[10]

Another problem stated in the paper is the zero marginal problem which is, having zero values as the constraints that we got by implementing the classical IPF. Although this phenomenon is not as common as the previous one, it can happen in small geographic areas. In our case of implementing this procedure to the population of North Dakota, we faced this issue. Some of the households that contribute to a zero marginal, may need to take non-zero weight to satisfy other constraints. Also, it would lead to a zero division error. So, to solve this issue the zeroes had to be replaced with some small number.

Algorithm 3.4: Generating weights

input : frequency_matrix, constraints
output: Weights for each Household
$weighted\_sums \leftarrow$ summation of each column
$weights, weights_{prev} \leftarrow$ array of size N with all values as 1
$N \leftarrow total\_households$
$m \leftarrow total\_columns$
while *equation 3.4 is not fulfilled* **do**
  calculate δ_{init} using equation 3.6
  for $j \leftarrow 0$ *to* m **do**
    $multiplier \leftarrow \frac{constraints[j]}{weighted\_sums[j]}$
    for $i \leftarrow 0$ *to* N **do**
      if $frequency\_matrix[i][j] == 0$ **then**
        $weights[i] \leftarrow weights_{prev}[i]$
      else
        $weights[i] \leftarrow weights[i] \cdot multiplier$
    for $k \leftarrow 0$ *to* m **do**
      $weighted\_sums[k] \leftarrow \sum_{i}^{i=N} frequency\_matrix[i][k] \cdot weights[i]$
  calculate δ using equation 3.6
  if $|\delta - \delta_{init}| \leq \epsilon$ **then**
    break

Table 3.5 illustrates one iteration of the following algorithm in our previous demo data.

ID	urban 1	urban 2	rural 2	white male	asian male	asian female	w1	w2	w3	w4	w5	w6
1	1	0	0	1	0	0	4	4	4	4	4	4
2	0	1	0	0	1	2	1	5	5	5	4	4.5
3	0	0	1	1	0	0	1	1	3	3	3	3
Constraints (IPF)	4	5	3	7	5	9						
ws1	4	1	1	5	1	2						
ws2	4	5	1	5	5	10						
ws3	4	5	3	7	5	10						
ws4	4	5	3	7	5	10						
ws5	4	4	3	7	4	8						
ws6	4	4.5	3	7	4.5	9						

Table 3.5: Generating weights for Households

3.2.5 Sampling

Now that we have weights for each of the Household types our next and final goal for reconstructing the population is to sample from the PUMS data. We do this in a probabilistic manner. Here, There will be multiple occurrences of the same household type in the joint distribution as the person types can be different in each of them. The probability of a house being chosen is the weight of that house divided by the summation of the weights of all houses belonging to the same type. [10] We rounded the weights to the nearest integer as almost all of the weights are decimal numbers. However, as a result of this rounding, some of the house types may get excluded from the generated population as some weights can be near zero. But the number of this kind of house is negligible. This approach for choosing households does not have the problem of being biased towards household-level or person-level constraints as the weights are adjusted by not only the household constraints but also the person-level constraints,

3.3 Limitations

While working with population reconstruction, we ensured that the constraints of census data were maintained through the algorithms. But, we still encountered multiple limitations of our population simulator.

3.3.1 Limited Type of Housing Units

We generated the whole population based on the dataset of North Dakota. Since our algorithm generates the weights of household units, North Dakota is a small area compared to the entire United States.

1. We encountered small finite household types in North Dakota, while in others, the Housing Units can be different based on PUMS's randomized algorithm[21]. Therefore, though we are maintaining the constraints of an entire population and its attributes, we are working with a limited type of housing unit.

2. In PUMS data, only a very small portion (less than 5%) sample of total housing units is collected.[21] For this reason, the reconstructed population is entirely based on our household units, which may exclude many persons with different attribute values who are not present in the household units.

3.3.2 Household-Based Population Reconstruction

Our household assignment is simultaneous with population reconstruction. It can create two problems:

1. If an additional layer of assigning them to households (Monte Carlo Sampling, Kristan Lum et al.'s method[23] could provide a more accurate reconstruction or not is not tested.

2. The areas of all the household is not generated.

3.3.3 Rounding off weights

During the procedure of the algorithm, we rounded off the weights before sampling. This rounding off is done by taking the nearest integer as weights. By doing so, some houses that have weights less than 0.5 will be excluded from the population. So, in the population, in certain cases, there can be a small amount of household types that will not be included to satisfy other household types constraints that contribute more to the population.

Chapter 4

Activity Assignment

Activity assignment constructs a model of the daily activities of the individuals of a simulated population. It must be done before location assignments as an individual's location at a particular time is related to the activity he/she performs at that time. Here, we try to predict a person's activity within a specified time. We used the NHTS Dataset and classification models for our activity forecast

For the ease of our experiment, we tried to predict the activity of the previously generated population for each hour of a normal weekday. In other words, we are looking for the relationship between someone's activity with their personal information. For this reason, we used multiple Machine Learning algorithms and compared the accuracies to find the best model for the prediction. Our activity assignment involves challenges like dataset preparation, preprocessing, feature extraction, and engineering before applying regression and classification models including Neural Networks.

First, we performed our Data Pre-Processing. Our primary goal is to create a dataset from the NHTS Data which can be compared to Census Data directly in terms of the features. Therefore, we extracted similar features and mapped them accordingly. Next, we performed some feature engineering since many samples in the NHTS Dataset do not contain appropriate information since the person declined to provide them. Finally, We take our previously generated population and use our classification model to predict the activity at any given time.

To predict the activity of a person at any specific time of the day we divide the prediction into two parts. To begin with, we take trip data for each person from the NHTS data [34].Then, we try to predict in starting of each time interval if the person is in a certain place or on his way from one place to another. In activity assignment, we have a total of 10 classes. To exemplify, when someone is starting a trip from home and going to work, his origin and destination number is 1 and 2.

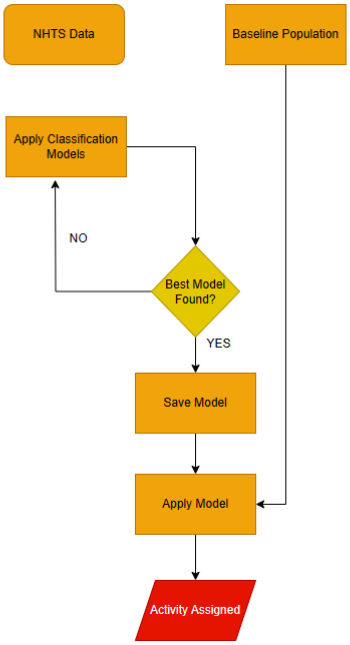


Figure 4.1: Detailed Steps of Activity Assignment

4.1 Introduction to NHTS Data

The National Health and Travel Statistics (NHTS) is comprised of four datasets: Household Data, Personal Data, Trip Data, and Vehicle Data. These datasets all pertain to the same individuals, with identification based on Household Identifier and Person Identifier.

The Household dataset contains information on around 130,000 households, comprising 115 features. It includes details such as Household Identifier, Travel Day (day of the week), Primary Sampling Stratum Assignment, Home Ownership, Count of household members, Count of household vehicles, Household income, and Frequency of Desktop or Laptop Computer Use to Access the Internet, among others. The Household Identifier is used to identify each household, while the Count of household members represents the number of people within a household. Travel-related information is captured through features like Travel Day (day of the week) and Number of Workers in the household. The dataset also provides insights into the medium of transportation used by households, which is indicated by features such as 'Frequency of Walking for Travel', 'Frequency of Bicycle Use for Travel', 'Frequency of Personal Vehicle Use for Travel', 'Frequency of Taxi Service or Rideshare Use for Travel', 'Frequency of Bus Use for Travel', 'Frequency of Train Use for Travel', and 'Frequency of Paratransit Use for Travel'. These factors include features like 'Price of Gasoline Affects Travel,' 'Travel is a Financial Burden,' 'Walk to Reduce Financial Burden of Travel,' 'Bicycle to Reduce Financial Burden of Travel,' 'Public Transportation to Reduce Financial Burden of Travel,' and 'At least two household persons are related,' 'Number of drivers in the household.' These features are used to track elements that might impact their traveling decisions.

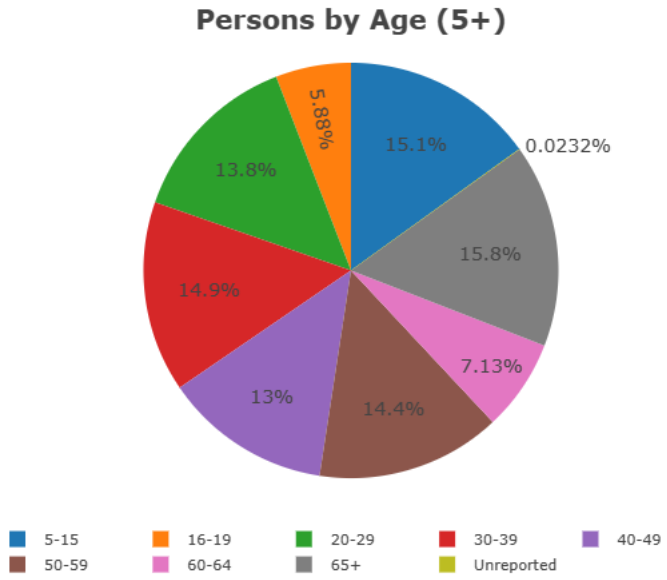


Figure 4.2: Samples from different age groups

Personal Dataset has 121 personal features. The primary focus of Personal Data is on the individual information within a household. Each person within the household is identified by numbers 1, 2, and 3, and their information, such as age, sex, race, and educational attainment, is stored. Additionally, the dataset contains travel-related information for each individual. This Personal Data is essential as it allows the identification of individuals within households, enabling the correlation of their travel data with the trip dataset. In total, the dataset consists of 121 features and 2,64,234 samples, providing information about 2,64,234 individuals from 1,29,696 households.

The Trip Dataset contains comprehensive information about two or three trips taken by individuals from the Personal Dataset. This dataset stores essential details such as trip purpose, origin, destination, travel distance, trip start time, and trip end time. The Trip Information plays a crucial role in providing an overview of primary trips made by multiple individuals on various days. With a total of 923,572 samples, this dataset includes information about 264,234 individuals and their respective 923,572 trips. It comprises 115 features, encompassing attributes like 'Trip Start Time', 'Trip End Time', 'Trip Origin Purpose', 'Trip Destination Purpose', 'Generalized Purpose of Trip', 'Primary Activity in the Previous Week', and other relevant information. Our information on Trip is solely based on the trips we get from the Trip Dataset.

Trip Dataset contains all the trips of an individual in 24 hours. Other than Trip Start Time and Trip End Time, we also get Trip Distance in Miles, and the Dwell time of that trip. Initially, we tried to calculate the total time spent at home, work, and other activities. First of all, we generated the average dwell time for the age groups and visualized the time period.

Code	Activity Description
01	Home
10	Work
20	School/Daycare/Religious activity
30	Medical/Dental services
40	Shopping/Errands
50	Social/Recreational
70	Transport someone
80	Meals

Table 4.1: Activity Description in NHTS Data

There is another dataset which is named Vehicle Dataset. The dataset contains vehicle-related information for households, such as details about the driver, vehicle age, number of vehicles, primary vehicle, and months of vehicle ownership. Analyzing this dataset will enable us to apply heuristics in understanding the location choices made by individuals.With a total of 60 features and 256,115 samples, the dataset provides information about 256,115 vehicles from 129,696 households.

Therefore, NHTS Data provides us the details of trips with multiple features related to it. Additionally, the dataset contains a lot of similar features to PUMS Data. For this reason, this dataset is our primary basis of forecasting.

4.2 Methodology

Our activity assignment involves a two-step process. Initially, we engaged in pre-processing our NHTS data to align it with our specific needs, as we didn't require all the available features. This entailed performing feature mapping to harmonize the dataset with census data, ensuring compatibility and uniformity.

Subsequently, we implemented classification models to analyze our selected features, namely 'From' and 'To,' with the aim of predicting the purpose of trip origin and destination. By employing these models, we sought to gain insights into the underlying patterns and factors influencing travel decisions, thereby enhancing our understanding of transportation dynamics and contributing to more effective planning and decision-making in this domain.

4.2.1 Dataset Preparation and Pre-processing

Our NHTS Data contains 200+ features in total. However, we could only work with a limited amount of features since most of the features are not available in our generated population, or are irrelevant for making predictions. Basically, it contains a lot of features that are trip-specific such as, what model of car was used in the trip or even how many wheels the car has. Besides, most of the features are encoded differently, therefore we ensured a feature mapping by creating functions for all

the features. Additionally, in a lot of instances, the interviewee declined to answer some important features and values like Household Income, Educational Attainment, and Gender. For this reason, we ensured some feature scaling. Lastly, there are some features that denote the purpose of the trip such as From('WHYFROM'), To('WHYTO'), Trip Purpose Summary('WHYTRP1S'), and Generalized Purpose of the Trip('TRIPPURP').

Mapping NHTS Data to the generated population

The features that we are working with, are labeled in different classification labels in the datasets. For example, when we are working with race, we get the races Asian, American Indian or Alaska Native, Native Hawaiian, or other Pacific in Census Data. But in NHTS Data, not everyone asserted their race. Also, the other races are labeled as 'others' while Census data has more race details. Therefore, we used others for all the answers in the NHTS Data before training.

For the working members in the household, Census Data has a limitation of 0-3. But, NHTS Data has a wider range, which is 0-7. To solve this issue, we used the maximum value of 3, as per the Census Data. Household income is distinguished into 9 separate groups and it is represented with an integer from 1 to 9.

Census Data	**NHTS Data**	**Meaning**
NP	HHSIZE	Household Members
VEH	HHVEHCNT	Household Vehicle Count
FINCP	HHFAMINC	Household Income
WIF	WRKCOUNT	Working members in the household
AGEP	R_AGE	Age
EDUC	SCHL	Education
SEX	R_SEX	Gender
RACE1P	R_AGE	Race

Table 4.2: Census Data vs. NHTS Data

Creating a Dataset with time-based Trip Details of Individuals

In NHTS Data, we only get details of a trip when a person changes his destination. For example, if person A moves from Home to their workplace, or returns home from their workplace, we get a sample of their trip. Therefore, the trip samples are only trip-based. Now, to determine a person's activity at a certain time, we need the location of that person in that specific time period. For our experiment, we fixed the time period to be 1 hour. So, now the new challenge is to calculate each person's location on an hourly basis. We had a total of around 1 million trip data for around 250 thousand people. So, making a dataset for each hour for each person, the size becomes about 24 million.

Now, we determined the person's location to be L_1, if, at the beginning of the time period, he was in location L_1. However, the location can be the place where he

spent the majority of the time period as well. But that would make the calculation much more costly. And, If we make the time interval shorter, which would make the simulator almost live, the highest time spent location and the location at the beginning of the period will not be different for maximum cases.

Now, the problem is at the beginning of the time period, or in our case the beginning of the hour, the person can be on his way from L_1 to L_2. To solve this issue we defined two locations for each period of time for each person. We will label them as $FROM$ and TO. If someone is on his way from L_1 to L_2 his $FROM$ will be L_1 and TO will be L_2. In other cases, both $FROM$ and TO will be the same.

The algorithm to assign $FROM$ and TO for each person is pretty straightforward. We assume each person is at his home at the beginning of the day and iteratively for each trip he/she makes we update his location.

Algorithm 4.1: Activity Assignment for each person for each time interval using trip data

```
Set all locations to be Home;
for each person do
    trip ← 0;
    L1 ← trip[starting_location];
    L2 ← trip[starting_location];
    for each interval do
        if trip[starting_time] and trip[ending_time] is between this interval
         then
            FROM[interval + 1[starting_time]] ← L2;
            TO[interval + 1[starting_time]] ← L2;
        else
            FROM[interval + 1[starting_time]] ← L1;
            TO[interval + 1[starting_time]] ← L2;
```

Initial Data					
PERSONID	HOUR	...	FROM	TO	
1	5	...	Home	Home	
1	6	...	Home	Home	
1	7	...	Home	Home	
1	8	...	Home	Home	
1	9	...	Home	Home	
1	10	...	Home	Home	
Trip Data					
PERSONID	TRIPNo.	...	STARTTIME	ENDTIME	DESTINATION
1	1	...	5.30	5.45	Work
1	2	...	7.45	8.15	Meal
1	3	...	9.15	9.30	Home
Combined Data					
PERSONID	HOUR	...	FROM	TO	
1	5	...	Home	Home	
1	6	...	Work	Work	
1	7	...	Work	Work	
1	8	...	Work	Meal	
1	9	...	Meal	Meal	
1	10	...	Home	Home	

Table 4.3: Example of time-specific dataset from trip data

Here, 4.3 the person was initially at home. At 5.30 he went to work and reached there at 5.45. So at the beginning of the 6^{th} hour, he was at work. Then, at 7.45 he started his trip to have a meal and reached there at 8.15. So, At the beginning of 8^{th} hour, he was on his way from work to meal.

4.2.2 Prediction

After performing data pre-processing, the features of our final dataset are appropriately mapped and engineered. Firstly, We used all the classification models to compare the accuracies. Based on the presumption that this is a classification problem, we used all classification models including Gradient Boosting, Random Forest, Support Vector Machine, Logistic Regression, Decision Tree, and Naive Bayes. We also performed a Neural Network with 3 dense layers but classification models provided us better performances.

Model	Accuracy (%)
Gradient Boosting	89.47
Random Forest	88.75
Support Vector Machine	88.12
Logistic Regression	87.88
Decision Tree	86.95
Naive Bayes	86.21

Table 4.4: Comparison of Classification Models

Gradient Boosting provided us the best accuracy 4.4 which is slightly better than Random Forest. So, We applied gradient boosting as it provided the highest accuracy of 89.47%. The accuracies are for predicting the $FROM$ feature. But for the TO feature, the accuracies are more or less the same. As in most cases, both feature have same values.

Algorithm 4.2: Predicting activity/location for each person

H **input :** model, population
for *each person* **do**
└ Create rows for each time interval
predict $FROM$ for the population with time;
predict TO for the population with time;

Person	HOUR	R_SEX	R_AGE	HHVEHCNT	HHSIZE	EDUC	FROM
1	0	1	4	3.0	3.0	2	1
1	1	2	4	3.0	7.0	1	1
1	2	1	4	2.0	5.0	2	1
1	...	...	...	...	...	...	...
2	0	2	4	0.0	4.0	3	1
2	1	2	5	2.0	2.0	5	2
2	2	1	2	3.0	3.0	2	3
2	...	...	...	...	...	...	...
3	0	2	3	1.0	1.0	2	1
3	1	2	4	2.0	5.0	1	1
3	2	1	5	3.0	3.0	4	1
...	...	...	...	...	...	...	...

Table 4.5: Predicting Origin

Person	HOUR	R_SEX	R_AGE	HHVEHCNT	HHSIZE	EDUC	TO
1	0	1	4	3.0	3.0	2	1
1	1	2	4	3.0	7.0	1	1
1	2	1	4	2.0	5.0	2	10
1	...	...	...	...	...	...	...
2	0	2	4	0.0	4.0	3	1
2	1	2	5	2.0	2.0	5	2
2	2	1	2	3.0	3.0	2	3
2	...	...	...	...	...	...	...
3	0	2	3	1.0	1.0	2	1
3	1	2	4	2.0	5.0	1	1
3	2	1	5	3.0	3.0	4	30
...	...	...	...	...	...	...	...

Table 4.6: Predicting Destination

Here, the Trip Origin and Trip Destination determine the activities of the people. Here, we have two steps: Predicting the trip origin and predicting the trip destination. After using our Gradient Boosting model, we are successfully able to predict and assign the activities. It is the basic forecast of people's tasks, which allows us to proceed to assign them appropriate locations.

4.3 Limitations

The main limitation for predicting the activity of the generated population is the lack of trip data. The NHTS dataset only provides about Nine million trips over a very specific and short time period. It is really a small and unbalanced dataset compared to the large population of the United States. If a more elaborate and detailed dataset was available, the activity prediction would have been more accurate. Moreover, the activities of a person on normal weekdays and weekends are much different. If there were separated datasets of weekdays and weekends, separate activities in weekends could have been predicted.

Chapter 5

Location Assignment

Location assignment is the next step after assigning activities. This part aims to assign a real-world geographic location to our synthetic population. As the population already has their activities in any specific time period of a day we need to input a real-world map to have their locations in it. HERE provides a detailed enough data for this procedure.

5.1 Introduction to HERE Maps

We used HERE Maps as our visualizer[29]. Because the HERE map allows us to get a detailed overview of the map, unlike Google Maps. For example, in HERE Maps we can even access the households. Additionally, in the HERE Map Studio, we can visualize our data on the map so easily. We need to create a dataset that contains the longitude and latitude of the people to visualize them. Therefore, our initial challenge is to assign a location and calculate the longitude and latitude of the place. We can visualize our data in HERE Maps through the Resource Description Framework(RDF) Dataset.

5.1.1 RDF Datasets

Resource Description Framework of HERE Maps contain the longitudes and latitudes of points of interests [30].

Facility Type	Count
Shopping	318
School	663
Bank	580
Amusement Park	10
Place of Worship	497
Restaurant	411
Government Office	359
Speciality Store	282
Grocery Store	323
.....	.....

Table 5.1: An example dataset of Abuja, Nigeria

To simplify, we provide an example dataset of Abuja, Nigeria. In Table 5.1, we can see that the dataset contains 318 places for shopping, 497 schools, 411 restaurants, and many more places. It is the primary representation of the total number of areas of interest in Abuja. We can put pointers in these places by providing RDF data as input. To conclude, the RDF data of a city helps us understand a town's areas of interest.

5.2 Methodology

Our location assignment is divided into two parts. First, we assign them into the households. Next, we assign them in the preferred locations after activity assignment.

5.2.1 Assigning households

The first step to assigning location is to assign households to our generated population as the population was generated mainly based on household and person-level constraints were matched later. Also, except under extremely unrealistic conditions, the number of households in the generated population and the map where we are projecting the population will never match. The reason for these unbalanced numbers is we had to round off the weights while picking houses. So, if we have a map of the same area where the generated population is based on, the numbers will not be the same. Also, as we did not have a map of our test dataset, North Dakota, we had to work with a map of Abuja, Nigeria, so we had to cut off some houses from the generated population to actually match the map.

To assign households, we prioritized houses that have a higher number of people in them. So, the house that has the highest number of people in it will get a house first. The procedure is to iteratively assign households from the map to the population.

Algorithm 5.1: Assigning households to the population

Sort Households according to the number of people in them
$House\_iterator \leftarrow 0$
for *each house in the map* **do**
    lat and lon of all person in $House\_iterator^{th}$ house $\leftarrow$ lat and lon of this house
    increase $House\_iterator$

5.2.2 Assigning locations

Now that all the people have their houses, they need to go outside and do their work. We have already calculated their activity in each period of time of a day in the previous chapter and we have all the locations of doing these activities in the provided HERE map. We just need to plug in these values. To do so, the heuristic that we used is if a person does Z activity for a period of time he will go to the place Z that is nearest to his/her house. We used the Manhattan distance to find the nearest distance.

$$Location = \arg\min_{l_i \in L} \sum |h - l_i| \tag{5.1}$$

- $Location$ = location of a person
- h = location of the household of that person
- L = Set of all possible locations of his activity

But this approach has a major problem that is, in every area, there are densely populated zones and zones that are not much populated. So assigning the nearest location of activities will result in some locations having too many people at a time and some having too less. To resolve this a threshold to the maximum capacity of any location had to be set. We assumed all the locations have the same capacity so the maximum limit of any location will simply be,

$$maximum\_capacity = \frac{\text{total number of that activity in the map}}{\text{total population}} \tag{5.2}$$

Algorithm 5.2: Assigning Locations

```
Make sets of each activity on the map;
counts ← frequency array of persons in each location;
for each person do
    for each time period do
        locations ← subset of locations of activity in this specific time
          period;
        min_distance = ∞;
        maximum_limit ← subset_size / population_size;
        for each location in locations do
            if |house[lat] − location[lat]| + |house[lon] − location[lon]| ≤
              min_distance then
                if count[location] ≥ maximum_limit then
                    continue;
                person[time_period][lat, lon] ← location[lat, lon];
                increase count[location];
```

5.2.3 Assigning Locations to people who are on the way from some place to another

In the previous chapter, we have seen that at the beginning of the time period, a person can be on his/her way from one place to another and to solve this issue, we predicted two different locations for that person. That is, there are two locations for each person, l_1 and l_2(FROM and TO), and, if, for any person, l_1 and l_2 are the same then he/she was at that place at that time. But if l_1 and l_2 are not the same then he/she was on his way from l_1 to l_2. To assign locations for these people we have to implement another heuristic, that is. how far he/she is from l_1 and l_2. We can get this value from his/her previous and next locations. if X = the number of intervals l_1 and l_2 is different then we divide the distance from l_1 to l_2 into X pieces and assign locations.

Algorithm 5.3: Update of algorithm 5.2 where any person is on his way from one place to another

```
...
for each time period do
    if l_1 and l_2 are not the same then
        from ← location for l_1 using previous algorithm
        to ← location for l_2 using previous algorithm
        count ← Count of subsequent intervals where l_1 and l_2 are not the
          same
        d ← distance between l_1 and l_2
        for i ← 0 to count do
            person[time_period + i][lat, lon] ← from[lat, lon] + i * d[lat, lon]
    else
        follow previous algorithm
...
```

5.3 Limitations

The main limitation of this part of the research is the unavailability of HERE map data of our desired location. As described in the Population Reconstruction chapter. Our preliminary data was based on US census data and Sampling was from North Dakota. However, we could not project the population in a real-world map because RDF files for that region are not available. Not only that, RDF files for any US or nearby region are unavailable for public usage.

Chapter 6

Technical Architecture

Our targeted output is the location of the individuals. Our input datasets are PUMS Dataset, NHTS Dataset, Census Dataset as we have mentioned earlier.

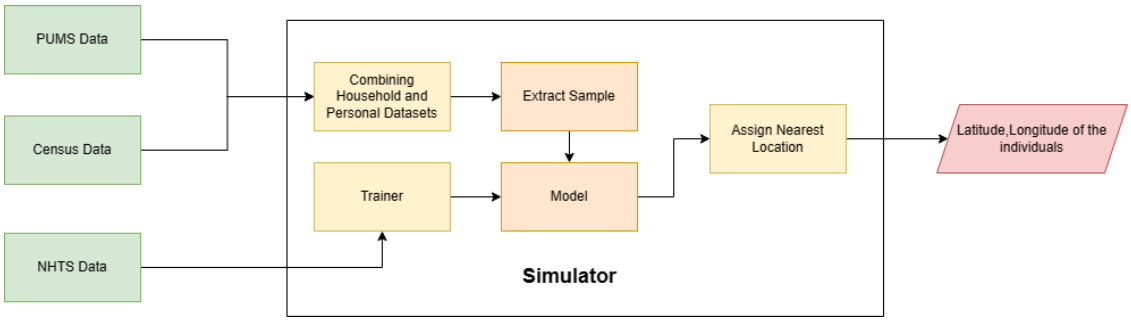


Figure 6.1: Simulator Architecture

Our work revolves around the PUMS Dataset, which comprises two distinct segments: Household Data and Personal Data. To begin, we merge these two datasets, creating a unified dataset. Subsequently, we derive a comprehensive distribution of features by drawing insights from Census data through the use of Iterative Proportional Fitting (IPF), as detailed in Beckman et al.'s work [4].

In the next step, our simulator assigns weights to each household utilizing the Iterative Proportional Updating (IPU) technique, as introduced by Xin et al. [10]. These weights play a crucial role in our analysis.

Finally, we employ this enriched dataset to perform sampling from the PUMS dataset, tailoring the number of samples to the total count of households in the region. This methodology forms the foundation of our thesis, enabling us to draw meaningful insights and conclusions from the data.

At the same time, our trainer uses NHTS Dataset to train the best classification model. Once the model is ready, it takes the extracted number of samples as an input and provides the activities of the individuals as output.

Once the activity is predicted, we use our algorithm 5.2 for the location assignment. The final dataset contains the details of the person, 'From' and 'To' activity

in the hour interval, and the latitude and longitude of the person. Then we use a visualizer [29] to project our results.

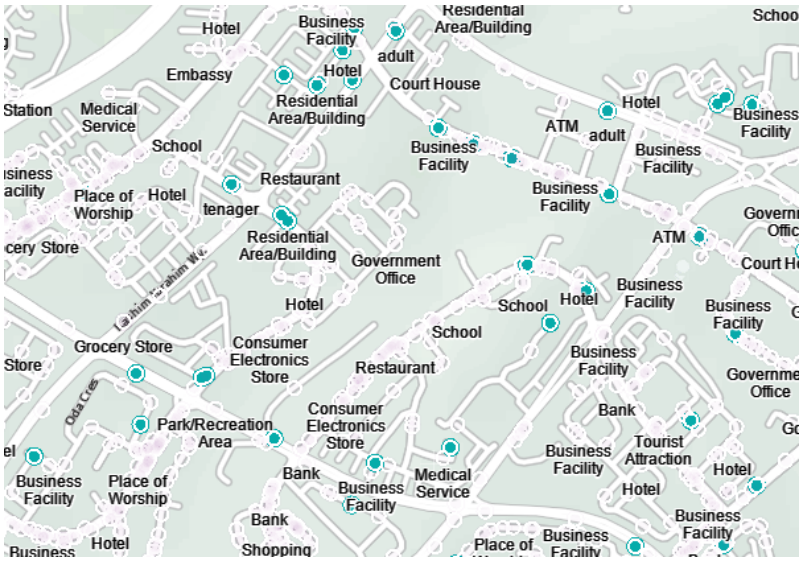
Chapter 7

Case Study

We used the population of the United States and the map of Abuja, Nigeria to visualize our data. Our simulator is able to provide us with the behavior of the population over time in 24 hours. Here, we are providing a brief case study that we get from our model.

7.1 Assignning each person to Households

First, we assign everyone to the available houses of the city. Since we are using the map of Abuja, we can't visualize individuals in their appropriate addresses. So, we assigned 10,000 individuals using our Algorithm 5.1 .



(a) Everyone Assigned to Home

(b) Everyone Assigned to Home(Closer View)

Figure 7.1: Household Assignment

As we can see in 7.1, everyone is assigned to their home, and the rest of the POIs are empty.

7.2 Activity Assignment

After assigning activities in time intervals, we see the change in behaviors. For example, when we are taking the time interval of 9 a.m. to 10 a.m. on an average working day, we see that people are mostly busy in all the activity classes. People were already at their workplaces, schools, restaurants and shops.

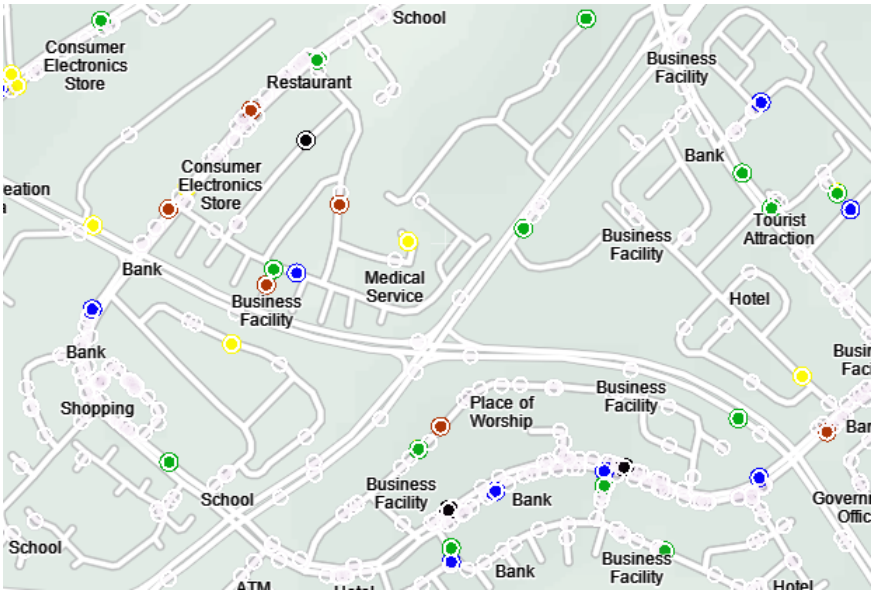


Figure 7.2: Visualization after Activity Assignment at 9 a.m.

Activity Types	Number of People
Home(Transparent White)	1166
Work(Green)	4972
School/Daycare/Religious Activities(Red)	2202
Medical/Dental Service(Black)	122
Shopping/Errands(Blue)	375
Social/Recreational Activity (Yellow)	1163
Total	**10000**

Table 7.1: Activities between 9 to 10 a.m.

In Table 7.1, we can see that out of our 10000 visualized individuals, 4972 went to their workplaces. Second most common activity type is School/Daycare activity and the most uncommon activity is taking medical services at that time.

Our time interval for next simulation is 4 to 5 p.m. . In this case, people are mostly at home and some portion of them are still at work. We dont see any portion of the sample is at School/Daycare/Religious Activities at that time.

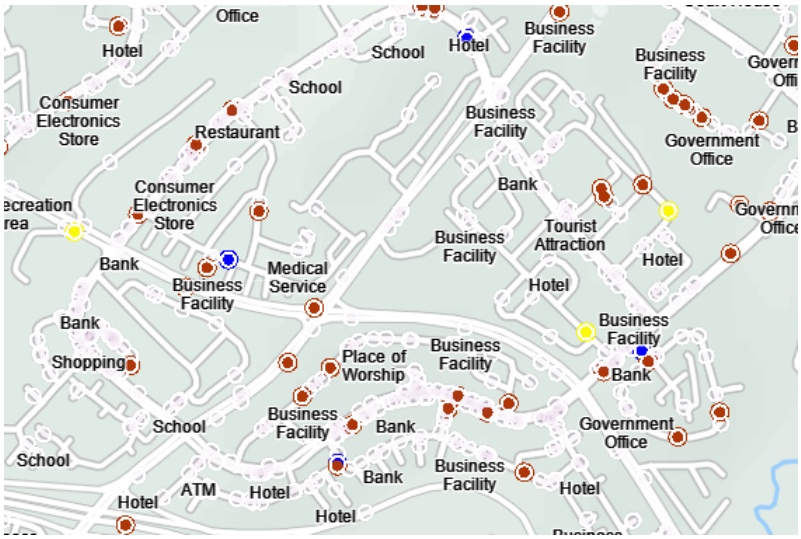


Figure 7.3: Visualization after Activity Assignment at 4 p.m.

Activity Types	**Number of People**
Home(Transparent White)	7565
Work(Green)	2263
Shopping/Errands(Blue)	100
Social/Recreational Activity (Yellow)	62
Total	**10000**

Table 7.2: Activities between 4 to 5 p.m.

So, by the end of the day, our simulator simulates that people are mostly spending time at their home.

Chapter 8

Conclusion

Synthesizing a representative population through population generation synthesis using census data allows for a comprehensive grasp of a region's demographics and socio-economic characteristics. This process provides valuable insights into the population's composition, distribution, and dynamics, enabling informed decision-making and effective policy formulation.

Moreover, population synthesis enables the projection of future population trends, the anticipation of demographic changes, and the evaluation of the impact of different scenarios and interventions. This information is crucial for urban planning, resource allocation, infrastructure development, and service provision, facilitating the fulfillment of evolving community needs.

Additionally, population synthesis facilitates the analysis of social disparities, identification of vulnerable populations, and evaluation of resource distribution equity. It uncovers patterns, correlations, and relationships among various demographic variables, supporting evidence-based decision-making and targeted interventions. In conclusion, population generation synthesis using census data empowers an understanding of population complexities, the anticipation of future trends, and informed decision-making. It significantly contributes to policy shaping, resource allocation improvement, and the promotion of equitable development. Harnessing the potential of population synthesis enables the creation of sustainable, inclusive, and resilient communities, ultimately benefiting society as a whole.

8.1 Future Works

We generated a population based on census data and travel information. However, we expect some characteristics in future models so that the simulation is more dynamic.

8.1.1 Contact Tracing Using Social Network

In this section, researchers use three types of the social network: education, work, and household. Firstly, the household data are taken from the census data. Also, they take living space and work location from road use data. In the next step, they place households, workplaces, and educational institutions and assign daytime locations for each individual. Lastly, the output shows the social network that has been created.

This method helps understand location choice and contact tracing. However, it does not deal with detailed contact tracing, i.e., consumer behavior for shops, travel behaviors, and use of healthcare by the people. For this reason, this method is very good at giving insights regarding contact tracing, but it is not the best method to use because the location assignment is less accurate.

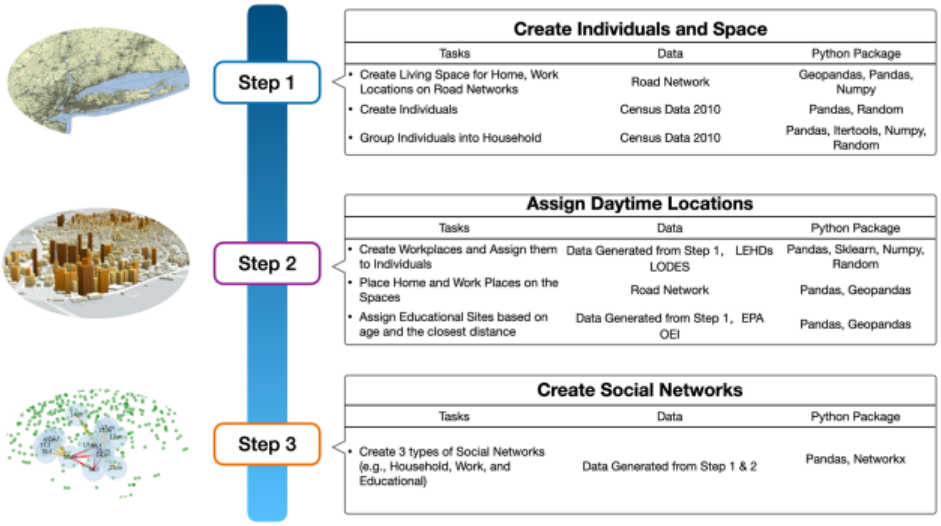


Figure 8.1: Contact Tracing Using Social Networks (Taken from [22])

It generated a synthetic population of 23,004,272 people and 8,457,710 households. However, due to the inconsistency of 116 census tracts, the number of households was slightly higher and caused 0.36% less accuracy.[22]

A future open-source simulator should be able to trace the contacts. Contact Tracing can become an important addition to this simulation because it can work as a primary basis to track upcoming outbreaks of contagious diseases. Besides, it'll help to study social relationships in the future.

Bibliography

[1] W. E. Deming and F. F. Stephan, "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known," *The Annals of Mathematical Statistics*, vol. 11, no. 4, pp. 427–444, 1940, ISSN: 00034851. [Online]. Available: http://www.jstor.org/stable/2235722 (visited on 08/26/2023).

[2] R. Brown, "Comparing census data in 90 countries," *The American Statistician*, vol. 25, no. 1, p. 32, 1971.

[3] W. Recker, "The household activity pattern problem: General formulation and solution," *Transportation Research Part B: Methodological*, vol. 29, no. 1, pp. 61–77, 1995, ISSN: 0191-2615. DOI: https://doi.org/10.1016/0191-2615(94)00023-S. [Online]. Available: https://www.sciencedirect.com/science/article/pii/019126159400023S.

[4] R. J. Beckman, K. A. Baggerly, and M. D. McKay, "Creating synthetic baseline populations," *Transportation Research A - Policy and Practice*, vol. 30, pp. 415–429, 1996.

[5] I. Rish, "An empirical study of the naïve bayes classifier," *IJCAI 2001 Work Empir Methods Artif Intell*, vol. 3, Jan. 2001.

[6] J. Peng, K. Lee, and G. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research - J EDUC RES*, vol. 96, pp. 3–14, Sep. 2002. DOI: 10.1080/00220670209598786.

[7] J. Cramer, "The origins and development of the logit model," Aug. 2003. DOI: 10.1017/CBO9780511615412.010.

[8] R. Moeckel, K. Spiekermann, and M. Wegener, "Creating a synthetic population," in *8th International Conference on Computers in Urban Planning and Urban Management (CUPUM)*, Sendai, Japan, 2007.

[9] J. Auld, A. Mohammadian, and K. Wies, "Population synthesis with subregion-level control variable aggregation," *Journal of Transportation Engineering*, vol. 135, pp. 632–639, 2009.

[10] X. Ye, K. Konduri, R. Pendyala, B. Sana, and P. Waddell, "Methodology to match distributions of both household and person attributes in generation of synthetic populations," Jan. 2009.

[11] J. Rich and I. Mulalic, "Generating synthetic baseline populations from register data," *Transportation Research Part A: Policy and Practice*, vol. 46, no. 3, pp. 467–479, 2012. DOI: 10.1016/j.tra.2011.11.002.

[12] R. Beckman, K. Channakeshava, F. Huang, *et al.*, "Integrated multi-network modeling environment for spectrum management," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1158–1168, Jun. 2013.

[13] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, Dec. 2013. DOI: 10.3389/fnbot.2013.00021.

[14] J. Y. Guo and C. Bhat, "Population synthesis for microsimulating travel behavior," *Transportation Research Record: Journal of the Transportation Research Board*, 2014.

[15] Vikramkumar, V. B, and Trilochan, "Bayes and naive bayes classifier," *CoRR*, vol. abs/1404.0933, 2014. arXiv: 1404.0933. [Online]. Available: http://arxiv.org/abs/1404.0933.

[16] H. Xia, J. Chen, M. V. Marathe, and S. Swarup, "Comparison and validation of synthetic social contact networks for epidemic modeling (extended abstract)," in *Proceedings of The Thirteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Paris, France, May 2014.

[17] N. Watthanasutthi and V. Muangsin, "Generating synthetic population at individual and household levels with aggregate data," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2016, pp. 1–6. DOI: 10.1109/JCSSE.2016.7748838.

[18] Z. He, D. Lin, T. Lau, and M. Wu, "Gradient boosting machine: A survey," 2019. arXiv: 1908.06951 [stat.ML].

[19] H. Deng, Y. Zhou, L. Wang, and C. Zhang, "Ensemble learning for the early prediction of neonatal jaundice with genetic features," *BMC Medical Informatics and Decision Making*, vol. 21, Dec. 2021. DOI: 10.1186/s12911-021-01701-9.

[20] S. Sumahasan, U. Kumar, A. Kintali, and S. Bontu, "Content-based sms spam messages classification using natural language processing and machine learning," Jul. 2021.

[21] U.S. Census Bureau. "2021 accuracy of the public use microdata sample (pums)." (2021), [Online]. Available: https://www2.census.gov/programs-surveys/acs/tech_docs/pums/accuracy/2021AccuracyPUMS.pdf.

[22] N. Jiang, A. Crooks, H. Kavak, A. Burger, and W. G. Kennedy, "A method to create a synthetic population with social networks for geographically-explicit agent-based models," *Computational Urban Science*, vol. 2, no. 1, Feb. 2022. DOI: 10.1007/s43762-022-00034-1.

[23] P. Ye, B. Tian, Y. Lv, Q. Li, and F.-Y. Wang, "On iterative proportional updating: Limitations and improvements for general population synthesis," *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1726–1735, Mar. 2022. DOI: 10.1109/TCYB.2020.2991427.

[24] Y. Luo, Y. Liu, F.-l. Chung, Y. Liu, and C. W. Chen, "End-to-end personalized next location recommendation via contrastive user preference modeling," 2023. arXiv: 2303.12507 [cs.IR].

[25] US Census Bureau. "Census.gov." (May 2023), [Online]. Available: https://www.census.gov/.

[26] *Census bureau - access to acs microdata*, https://www.census.gov/programs-surveys/acs/microdata/access.html, Accessed on September 12, 2023.

[27] "Iterative proportional fitting." Accessed: [10 may, 2023]. (), [Online]. Available: https://en.wikipedia.org/wiki/Iterative_proportional_fitting.

[28] N. Lomax and P. Norman, "Estimating population attribute values in a table: 'get me started in' iterative proportional fitting,"

[29] HERE Developer. "HERE Studio - HERE Developer." (Year (or date of access)), [Online]. Available: %7Bhttps://developer.here.com/products/platform/studio%7D.

[30] HERE Technologies. "Here platform: Map data." (Year of Access), [Online]. Available: https://www.here.com/platform/map-data.

[31] IBM. "Logistic Regression." (Year (or date of access)), [Online]. Available: %7Bhttps://www.ibm.com/topics/logistic-regression%7D.

[32] JavaTpoint. "Logistic Regression in Machine Learning." (Year (or date of access)), [Online]. Available: %7Bhttps://www.javatpoint.com/logistic-regression-in-machine-learning%7D.

[33] KDnuggets. "A gentle introduction to decision tree algorithm." Accessed on Date. (Year), [Online]. Available: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html.

[34] Oak Ridge National Laboratory. "National household travel survey (nhts)." Accessed on Date. (YYYY), [Online]. Available: https://nhts.ornl.gov/.

[35] TF Resource. "Activity-based models." (Year of the Web Page), [Online]. Available: https://tfresource.org/topics/Activity_based_models.html.

[36] TIBCO. "What is a random forest?" Accessed on Date. (Year), [Online]. Available: https://www.tibco.com/reference-center/what-is-a-random-forest.