Substance Use and Abuse

Team 12688

March 2, 2019

1 Executive Summary

Substance use and abuse may have a variety of physiological effects, and in light of the recent rise in e-cigarette usage among teenagers, concerns have been raised as to how to address the proliferation of substance abuse and usage. However, before we begin to develop policies, initiatives, and programs, it is important to understand how and how fast these substances spread, who is at risk, and how they affect entire communities. Together, our solutions to the three given problems address these fundamental questions.

For the first part of the problem, we used an epidemiological model to predict the trend of vape users within the next 10 years in the United States. We predict that vape users, currently estimated at around $3-5\%$ of the population, will quickly reach a peak of almost 12% within a year, then slowly decline to about 6% over the next 10 years. When compared to the past usage of smoking, we predict that vaping will reach its popularity peak much faster but will retain a group of loyal users over a long period of time.

Our second model is a logistic regression model which estimates the risk an individual is at to use a given drug given a variety of characteristics which we detail further later. Our model predicted that out of 300 seniors, 1 student would use illicit opioids, 36 seniors would use nicotine, 132 seniors would use marijuana, and 94 students would use alcohol. Our predictions for opioid, marijuana, and alcohol usage are fairly consistent with national averages. However, our prediction of nicotine usage was not consistent, as national average nicotine use out of a representative sample of 300 students would be over 100 students. This inaccuracy may be due to our data model, which may not be representative.

Our third model is a logistic regression model with constrained coefficients that estimates the impact of a particular drug on an entire community. (describe the purpose of constrained coefficients) In particular, we examined the relationship between drug use and three indexes: a socioeconomic index, a health-care index, and a safety index. Each index encompasses a variety of aspects pertinent to determining the well-being of a community.

2 Problem 1: Darth Vapor

2.1 Problem Interpretation

Vaping is exposing nicotine to a new generation of teenagers. In addition, smokers may be enticed to switch to vaping to cut down on their cigarette consumption. The goal of the problem is to use a model to predict the spread of vaping over the next ten years compare our predictions to the historical rise and fall in popularity of cigarettes. We chose to focus on the growth in the United States.

2.2 Assumptions

Assumption 1: The more addicted persons there are around an individual, the more likely this individual is to become addicted.
Justification: The effects of peer pressure and the environment with respect to addiction density has long been of a great interest from both a sociological and biological perspective.

Multiple studies have found that individuals surrounded by many addicted individuals are more likely to become addicts themselves than individuals who are not.

A 2012 study conducted by Ramirez et al. [1] found that individuals who have four or more drug-addicted friends, were significantly more likely to become drug addicts themselves compared to individuals with (strictly) less than four drug-addicted friends ($p = 0.0002$).

Assumption 2: Individuals once addicted and recuperated are more likely to relapse into addiction than individuals who have never used aforementioned substance.
Justification: According to the National Institute of Drug Abuse [2], even a short-term or temporary addiction significantly alters the brain's dopamine mechanisms. This indicates that previous drug users are more likely to become addicted than non-users.

Assumption 3: The chances that an addicted individual quits and recuperates decreases as time under the influence increases.
Justification: According to the Canadian Cancer Society [3], the longer a smoker is addicted, the harder it will be for the individual to quit. We assume that since smoking and vaping both use nicotine, they share the addictive qualities that will make it harder to an individual to quit.

Assumption 4: Conversely, the risk that a recuperated and non-addicted individual relapses decreases as this aforementioned individual spends more time without the influence.
Justification: As with most of our assumptions, this results has been demonstrated multiple times in the scientific and epidemiological literatures, most notably in (Garcia-Rodriguez et al, 2013). In this work, researchers found that recuperated drug users experienced hyperbolic decreases in relapse risk as time without the influence increased.

We will likewise hold the assumption that this probability follows a hyperbolic function.

Assumption 5: Locally speaking, the probability of two (identical) persons of falling into addiction is the same.
Justification: This is for the ease of computation.

Assumption 6: Children become susceptible to vaping once they are 13 years old.
Justification: We based this assumption off of a study in the Tobacco Control journal [4], which measured nicotine and marijuana usage beginning in the 8th grade, when most children are around age 13.

Assumption 7: Immigrants to the United States have age demographics roughly the same as the United States' age distribution, and immigrants initially are addicted to vaping at the same rate as those in their age group in the United States.
Justification: According to the Pew Research Center [5], the proportion of immigrants in each group were within 5-10% of the proportions we used for the total US population.

Assumption 8: Vaping will face a slowly growing backlash from health organizations and the government in the next ten years.

Justification: While vaping may be seen as "safer" than smoking as it does not contain tobacco or harmful carcinogens, as it becomes more prevalent in society, the media and scientific research may question its benefits and discourage individuals from vaping.

2.3 Variables

Variable	Definition
$S(t)$	The number of susceptible but not addicted individuals in a local community after t days.
$I(t)$	The number of addicted (infected) individuals in a local community after t days.
$f(t)$	The fraction of individuals yet to be addicted after t days: $\frac{S(t)}{S(t)+I(t)}$.
$l(q)$	The length of time, in days, that individual q has been in their current state (susceptible or infected).
$k(t)$	The *anti-vaping* factor, a measure of how much public anti-vaping sentiment (from media, awareness programs) there is after t days.
$p_i(q)$	The *infection probability* function, which determines the probability that non-addicted (susceptible) individual q picks up vaping (becomes infected).
$p_s(q)$	The *susceptibility probability* function, which determines the probability that addicted (infected) individual q stops vaping (becomes susceptible).

2.4 Constants

Constant	Definition
$r = 0.1233$	The *relapse* factor, a measure of how likely a previously infected susceptible user is to becoming infected again. This constant is derived from a logarithmic fit of a graph of the fraction of nicotine relapses over time. (The derivation of this value can be found in Appendix A)
$b = 3$	The number of people born per day in a sample of 100,000 individuals. This constant is interpolated from an estimation of 4,000,000 births annually in the United States. (The actual number ranges from 3,950,000-4,300,000 annual births in the past ten years.)
$i = 1$	The number of people immigrating to the United States per day in a sample of 100,000 individuals. This constant is interpolated from an estimation of 900,000 immigrants annually.
$T = 3650$	The total duration of the model in days. We chose this value since we're trying to model the spread of vaping for the next 10 years.

2.5 Model

We chose to model the spread of vaping in the United States using a stochastic model with a representative sample of 100,000 individuals. Our model is inspired by the SIR (Susceptible, Infected, Recovered) model from epidemiology and views vaping as an infectious disease spread by media and peer influences. In our model, every individual is either in the S (susceptible) or I (infected) state. As opposed to the SIR model, we chose not to include a R (recovered) state since someone who quits an addictive substance is prone to relapse and "rejoin" the S state.

Our initial population of 100,000 had age demographics and a number of initial vapers proportional to that of the United States' (specific values can be found in Appendix B). At every time step (a day), a susceptible individual has a probability of becoming "infected" (addicted to vaping) based on the fraction of individuals who are already infected, and an infected individual has a probability of losing their infection (quitting vaping) which is proportional to the level of anti-vaping sentiment and inversely proportional to the amount of time they've been vaping. Additionally, if an individual has been addicted in the past, this adds an additional probability of becoming infected, which is inversely proportional to the time since they last vaped.

At every time step, we add new members to the population due to immigration and children reaching their teenage years. As stated in the Constants section, we assume a constant rate of childbirth of 4,000,000 births each year in the United States, which equates to a rate of 3 per day in our population of 100,000. Assuming that infant and child mortality are negligible, we can conclude that the rate of children entering their teenage years is equal to the rate of childbirth. Therefore, we add 3 people to the population every day, whom we assume to all be unexposed to vaping. The United States also receives about 900,000 immigrants into its population every year, which equates to an approximate rate of 1 person per day in our scaled-down population. This person's vaping status is determined randomly depending on age demographics and chance of vaping per age demographic.

2.6 Functions

Before we can define our probability functions p_i and p_s, we must first define a function on which they both depend: the *anti-vaping* factor, or $k(t)$. This function represents how much public anti-vaping sentiment (from media, awareness programs) there is after t days. It is evaluated as the following:

$$k(t) = 0.7 + 0.3 \tanh\left(2f(t) + \frac{t}{T}\right)$$

As can be seen through the use of the hyperbolic tangent, this function is sigmoidal. However, since the domain of t is from 0 to infinity, the function is merely half-sigmoidal with a range of from 0.7 to 1.

This function incorporates f, which is the fraction of the total population which vapes, within the argument of the hyperbolic tangent function, meaning that a higher f would result in a higher level of anti-vaping sentiment. This is because a higher prevalence of vaping would alert the media to inform the population of the adverse addictive effects of vaping.

Additionally, the fraction of time which has passed is implemented within the hyperbolic tangent's argument. This is because we predict that the technology will still receive increasingly negative media coverage regardless of the prevalence of vaping due to its dangerous levels of nicotine.

The probability of an susceptible individual's infection (becoming addicted), or p_i, is defined in the following way:

$$p_i(q) = \left\{ \begin{array}{ll} f(t)(1-k(t)) & \text{if individual } q \text{ hasn't been infected before} \\ 1 - \Big(1 - f(t)(1-k(t)\Big)\Big(1 - \frac{r}{\frac{l(q)}{10}+1}\Big) & \text{if individual } q \text{ has been infected before} \end{array} \right\}$$

Note that there are two cases for this function. The first case is when individual q has never experienced vaping addiction in the past, the the second case is when they have. This way, we appropriately augment the probability of infection using the relapse factor since the probability of relapse is greater than the probability of becoming addicted for the first time (Assumption 2).

Beginning with the first case, we simply have the fraction of the population which is currently addicted to vaping, or f, multiplied by complement of the anti-vaping factor k. This is because a higher prevalence of vaping leads to a higher chance of an individual interacting with those who could spread the trend to them, and the higher the anti-vaping sentiment is, the lower the chance that one decides to adopt the trend.

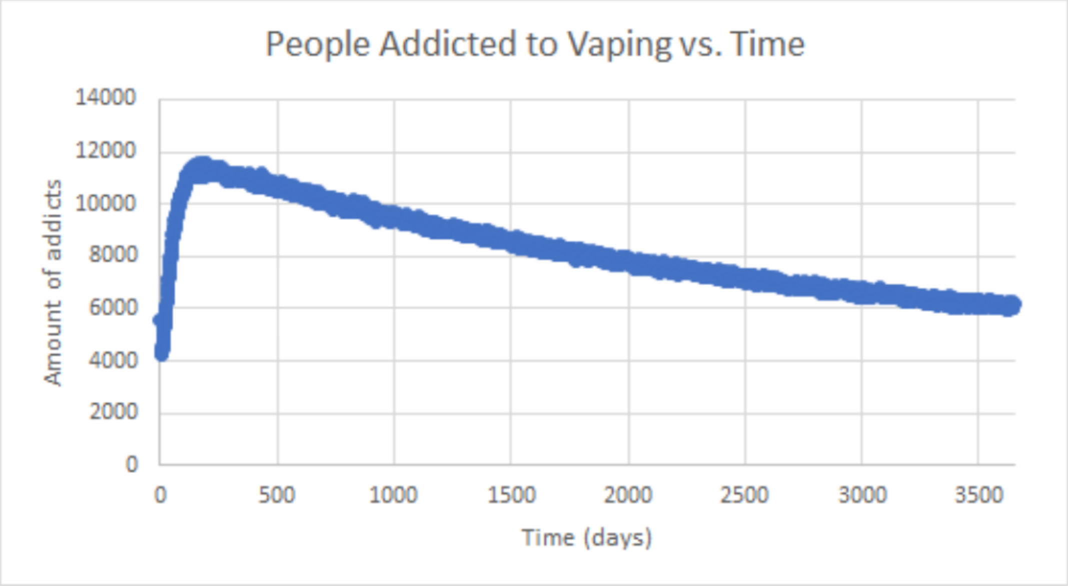
For the second case, we take the complement of the probability of the first case, multiply it with the complement of the relapse probability, and then take the complement of the product. The relapse probability is defined as $\frac{r}{\frac{l(q)}{10}+1}$ since, as described in Assumption 3, the chance of relapsing decreases the more time passed without vaping. By multiplying the complements of the first case's probability and the relapse probability, we find the probability that neither of these events happen. Therefore, taking the complement of that product returns the probability that at least one of these events happen, leading to re-addiction. Thus, the second case returns a augmented version of the first case as a result of the relapse factor.

The probability of an infected individual losing their infection (quitting vaping) and becoming susceptible, or p_s, is defined in the following way:

$$p_s(q) = \frac{k(t)}{\frac{l(q)}{10}+1}$$

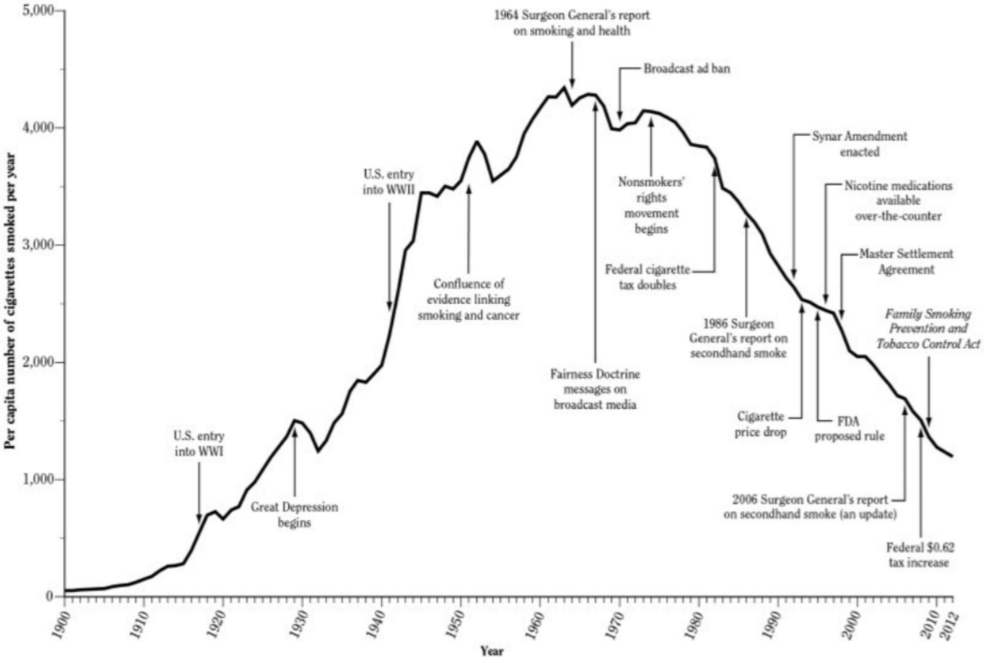
Here, we simply take the anti-vaping factor and divide it by $\frac{l(q)}{10}+1$, so the probability of an individual quitting is directly proportional to the anti-vaping factor and inversely proportional to the amount of time since they began vaping. This is due to the fact that higher anti-vaping sentiment will place more pressure on an individual to quit, while the longer one has experienced addiction, the more difficult it is for them to stop.

2.7 Model Results



As shown by the graph, the number of addicts starts at 4,000 and initially spikes at around 12,000 out of the population of 100,000. Then the number of addicts slowly decreases and begins to level off at around 6,000 after 9-10 years. We may interpret the first spike as a large amount of initial interest in vaping. This in turn causes the anti-vaping factors to rise, decreasing the number of addicts. The factors of relapse and the anti-vaping factors seem to roughly cancel each other out, eventually resulting in the number of addicts reaching an equilibrium by the end of the 10 years.

2.8 Validation



When comparing our model to the historical spread of cigarettes in the United States, they both share an initial peak followed by a decline. However our predicted model reaches its peak very quickly, after less than a year, and declines more slowly. We speculate that in a more modern, interconnected world our anti-vaping factor would have a much more immediate impact on vaping than smoking taxes and health reports had on cigarettes. Additionally, since vaping is seen as a healthier alternative to smoking, it may not have as sharp a decline as smoking had and may level off in popularity eventually.

When testing our model, we found that two main factors, the anti-vaping factor and the relapse factor, had a large effect on the number of vape users. We initially used a more simplistic model which held the anti-vaping factor constant, and we found that setting a value too low would cause the number of vape users to dominate the population. Additionally, lowering the relapse factor would cause previous users to be too unlikely to relapse, lowering the number of vape users to an unrealistically low level. This suggests that our model is highly dependent the accuracy of these two parameters.

2.9 Strengths & Weaknesses

Strengths:

1. Extensiveness
   (a) In our model, we accounted for a variety of factors when calculating the probability that a susceptible person becomes infected, and vice versa. For the former scenario, we included factors such as past nicotine use history, fraction of infected people, and the anti-vaping factor. For the latter scenario, we included factors such as the anti-vaping factor, and the amount of time they've been infected for.
2. Informed by current statistics
   (a) Our model's initial parameters are based of the United States' current population and fraction of people who are vapers.
   (b) Additionally, our model takes into account the large influx of teenagers who become susceptible to vaping each year, as well as immigrants to the United States who may be already addicted to vaping.

Weaknesses:

1. Uniformity
   (a) We did not account for factors mentioned in problem 2, such as personality traits, which would determine the probability of a susceptible person to be infected. With more time, we could integrate our model for problem 2 into problem 1 to better represent the probability of each individual being infected.
   (b) We assumed each person interacts with the same number of infected people. In reality, the rate at which an individual interacts with other infected people would vary. If we integrated our model for problem 2 into our model for this problem, the interaction rate per individual could be dependent on their personal traits.
2. Accounting For Anti-Vaping Factor
   (a) Our formula for the anti-vaping factor rests on an assumption that the media will portray vaping more negatively as a larger fraction of the population becomes addicted. However, if there exists a more quantifiable method of calculating the anti-vaping factor, which is contingent on social media campaigns, rehabilitation & awareness programs, among other things, our model may readily utilize it.

3 Problem 2: Above or Under the Influence?

3.1 Problem Interpretation

We are asked to develop a model which estimates the probability that an individual with particular characteristics will use a particular substance. In addition, we are asked to predict how many students out of 300 high school seniors with varying characteristics will use nicotine, marijuana, alcohol, and un-prescribed opioids. We chose to streamline this process by building a model that focused on high school students, but had the ability to be generalized to any individual.

3.2 Assumptions

Assumption 1: The population we sourced from the dataset on which we ran logistic regression was representative of the national student population.

Justification: While many records were either incomplete or corresponded to adults, the sheer size of the dataset and scope of the project (which considered a variety of different states) allows us to reasonably assume that this was representative.

Assumption 2: Other drug users that were not included in our population can be reasonably approximated by slightly changing data values for drug users included in our population.

Justification: In order to avoid a skewed dataset, we ran a method known as SMOTE (explained below) which essentially does this in order to generate more data on drug users. However, it can logically be assumed that other drug users would be fairly similar to those in our dataset and therefore can be ”generated” in this way.

Assumption 3: Survey responses from students will be reasonably close to the actual truth.

Justification: Clearly, many students may choose to lie while taking the survey and misreport their own personal information or past experience with drugs. However, as there is no real way to account for dishonesty in survey responses, we chose to simply rely on survey data for our model.

3.3 Model

We opted to treat this task as four separate classification problems, each of which involving predicting whether or not students (defined as people younger than 18 in the dataset used) would use nicotine, marijuana, alcohol, or un-prescribed opioids. In order to solve each of these four problems, we utilized a logistic regression model that sought to use regression to find the variable t and map each individual to some point on the sigmoid curve $y = \frac{1}{1+e^{-t}}$. This specific model was selected due to its utility in binary classification problems in supervised machine learning.

Our dataset was taken from the 2016 National Survey on Drug Use and Health [6], a study conducted by the Substance Abuse and Mental Health Services Administration, a subdepartment of the Federal Government’s Department of Health and Human Services. While this dataset included over 57,000 data points and over 2,000 dimensions, we opted to simplify this, using only individual samples under the age of 18 and hand-selecting roughly 40 dimensions that we wanted to consider in our model. In the end, this reduced our dataset to just 14,273 samples, many of which were incomplete.

Two of the dependent binary variables used in our classification problems were directly taken from the dataset; MJEVER represented whether or not survey respondents had previously used marijuana, while ALCEVER represented whether or not survey respondents had previously used alcohol. The other two dependent binary variables used were composite variables that took multiple variables in our dataset into account; OPIEVER considered both heroin usage and illicit painkiller usage (usage of either represented OPIEVER = 1, while usage of neither represented OPIEVER = 0). Similarly, NICEVER considered cigarette, smokeless tobacco, cigar, and pipe usage.

We also regularized our data in order to ensure that none of the factors would have an overly large effect on the algorithm; this was done by replacing each term x_{ij} with $\frac{x_{ij}-\mu_j}{\max_j - \min_j}$, where $\mu_j, \max_j, \min_j$ represent the mean, maximum, and minimum, respectively, of column j.

Our dataset was initially very skewed, as a majority of students had never used these illicit drugs. This is represented in the table below, which shows what proportion of students had used marijuana, alcohol, opioids, or nicotine in our original dataset.

Drug	Users	Non-Users	Usage %
Opioids	276	13996	1.9%
Nicotine	2290	11982	16.0%
Marijuana	2173	12089	15.2%
Alcohol	3866	10396	27.1%

To rectify this skewed dataset, which could have adverse effects on our model, we utilized a method known as SMOTE (Synthetic Minority Over-Sampling Technique) [7], which created synthetic copies of samples from the minor class (drug users) by slightly altering existing samples. We implemented this through usage of the Python package imblearn.over-sampling. This had the effect of allowing us to train our logistic regression algorithm on a more balanced dataset.

As we had an abundance of features relative to the amount of data we had and sought to avoid a model with high variance, we opted to use Recursive Feature Elimination (RFE) in order to select only the most relevant individual characteristics to predict usage of each of our four illicit drugs. Recursive Feature Elimination repeatedly constructs models and chooses the best or worst performing feature before repeating the process with the remaining features. This process is repeated until all features in the dataset are exhausted, leaving the user with a smaller subset of features that performs well. We opted to have RFE select 12 features for each logistic regression problem; this figure was arrived at by informal testing of the effects of feature number on model accuracy.

3.3.1 Features Used

Note that a full list of what all of the factors represent can be found at [6]

Features Selected for Consideration by RFE:

Health/Disability Factors: HEALTH, DIFFHEAR, DIFFSEE, DIFFTHINK, DIFFWALK, DIFFDRESS, DIFFERAND

Socioeconomic Factors: GOVTPROG, INCOME, IRINSUR4, IRWRKSTAT18

Demographic Factors: IRSEX, NEWRACE2, MILTFAMLY

Household Factors: IRHHSIZ2, NRCH17-2, IMOTHER, IFATHER, IRHH65-2, YEPCHKHW, YEPHLPHW, YEPCHORE, YEPLMTTV, YEPLMTSN, YEYARGUP

Social Factors: YESTSCIG, YESTSMJ, YESTSALC, YESTSDNK
School Factors: YESCHFLT, YESCHWRK, YELSTGRD
Employment Factors: IRWRKSTAT
Mental Health Factors: YMDELT, YSDSOVRL

Features Used to Predict MJEVER: YESCHWRK, YESTSCIG, YESTSDNK, YEPCHKHW, YEPCHORE, YEPLMTSN, YMDELT, YSDSOVRL, MILTFAMLY, IMOTHER, NRCH17-2, IRHHSIZ2

Features Used to Predict ALCEVER: YESTSCIG, YESTSMJ, YESTSALC, YEPLMTTV, YEPLMTSN, YMDELT, YSDSOVRL, DIFFSEE, MILTFAMLY, IMOTHER, IRINSUR4, GOVTPROG

Features Used to Predict OPIEVER: YESCHWRK, YEPLMTTV, YEYHGUN, YMDELT, YSDSOVRL, IRSEX, MILTFAMLY, IMOTHER, NRCH17-2, IRHHSIZ2, IRHH65-2, 'IRINSUR4

Features Used to Predict NICEVER: YESCHWRK, YESTSDNK, YEPCHORE, YEPLMTSN, YEYHGUN, YMDELT, YSDSOVRL, MILTFAMLY, IRWRKSTAT, IMOTHER, NRCH17-2, IRHH65-2

After randomly splitting our data into training and test sets (using the standard 70-30 ratio of training to test data) and supplementing our training data with our SMOTE-generated artificial data. We were then able to run separate logistic regressions with the above features in order to predict each of our four binary variables. This was done using the Python package SKLearn's built-in logistic regression algorithm. Finally, we used the results of our algorithms' performance on the test set and scaled it down to 300 students in order to predict the number of students who would use each individual drug. We assumed our data set to be representative of 300 high school seniors, since our model found age to not significantly affect whether or not individuals used a substance.

3.4 Results

Note that in the below results (representing the performance of our logistic regression model on the test dataset), 0 represents a student who would not use the drug and 1 represents a student who would use the drug.

Opioids	Predicted	Precision	Recall	Fl-score	Support
0	332	0.95	1.00	0.97	314
1	1	0	1	0.05	19

Our model predicted that approximately $\boxed{1}$ of the 300 students would use illicit opioids.

Nicotine	Predicted	Precision	Recall	Fl-score	Support
0	273	0.68	0.93	0.79	213
1	40	0.65	0.22	0.33	120

Our model predicted that approximately $\boxed{36}$ of the 300 students would use nicotine.

Marijuana	Predicted	Precision	Recall	Fl-score	Support
0	186	0.58	0.88	0.70	188
1	147	0.53	0.17	0.26	145

Our model predicted that approximately $\boxed{132}$ of the 300 students would use marijuana.

Alcohol	Predicted	Precision	Recall	Fl-score	Support
0	229	0.73	0.81	0.77	216
1	104	0.56	0.45	0.50	117

Our model predicted that approximately $\boxed{94}$ of the 300 students would use alcohol.

3.5 Validation

It is reported that in 2018 40% of 12th graders have used marijuana [11], in 2018 30% of 12th graders drink alcohol [9], in 2017 3.4% of 12 graders have used opioids [12], and in 2018 37% of 12th graders have vaped [13].

Based on the above data, and if our sample was representative, we could expect around 120 students to have used marijuana, 90 students to have consumed alcohol, 10 students to have used illicit opioids, and 111 students to have vaped. Our model seems to be relatively consistent with marijuana, opioids, and alcohol. However, our nicotine prediction is too small, especially considering that we can expect 111 students to vape, which is only a subset of nicotine products.

3.6 Strengths & Weaknesses

Strengths:

1. Comprehensiveness of Features Considered

   (a) We considered a variety of features from each student's life. The breadth of the dataset used allowed us to take into account social circles and a variety of characteristic traits. At different times, we considered factors as disparate as household size, usage of drugs in social circles, performance and interest in school, and individual disabilities, as well as many others.

   (b) While we eventually significantly narrowed the number of factors used, we still considered a wide range of features while searching for the most effective factors with RFE. The wide net we cast allowed us to pursue, in theory, what should have been the cream of the crop from the thousands of features in the dataset.

2. Algorithm Strength

   (a) Logistic Regression was specially chosen due to its ability to make binary predictions in such problems. While our model could have been substantially improved upon, we believe this was a result of the implementation of the algorithm rather than the algorithm itself. Logistic Regression is notably better-performing than other algorithms that deal with classification problems with a low signal to noise problem (or a problem with a high level of uncertainty), which makes it suited to analyze a complex dataset such as the one we ultimately used.

Weaknesses:

1. Dataset Sparseness
   (a) While our dataset contained a lot of individual data entries, most of them were incomplete or did not satisfy our condition of only using student data. As a result, we were only able to use 1,090 complete records for each of our four logistic regression problems, which resulted in serious underfitting and a high-variance model.
   (b) Many of the questions in the survey that the dataset we used was created from were only asked to specific demographics (e.g. 18+ y/o students or 12-17 y/o students), which greatly constrained our ability to create a meaningful model that included all age groups.
2. Inaccurate Feature Selection
   (a) We hand-picked certain features out of the large variety of features provided by the dataset. While this was much quicker than running RFE on over 2,000 distinct features, it meant that we were likely suboptimally selecting features.
   (b) However, as mentioned above, running RFE on so many different features would have been infeasible due to the relatively small amount of complete data we ended up working with. Any attempt to do so would have. In the end, we would have needed orders of magnitudes more of complete data in order to effectively run an RFE on so many features. However, it is undeniable that with better feature selection and more data, we would have had a more effective model.

4 Problem 3: Ripples

4.1 Problem Interpretation

We are asked to determine the impact of methamphetamines, marijuana, heroin, cocaine, and opioids on a community. We will consider the socioeconomic, health, and safety aspects of each community.

4.2 Assumptions

The assumptions follow the ones justified in the previous problems.

4.3 Data

For this problem, we consulted the University of Maryland's National Drug Early Warning System. Throughout the past five years, NDEWS has collected large datasets on substance abuse throughout the continental United States. We considered the following cities:

Chicago, Detroit, Denver, Atlanta, Miami, Seattle, San Francisco, Los Angeles, New York City, and Philadelphia.

For each city, the dataset provided various information on each drug: the number of crimes committed while under the influence of this substance, the number of rehab admissions due to this substance, the number of overdose deaths, high school usage, etc.

However, we found that only consistently usable data was in the following two categories: rehab admissions and crimes with substance found in the evidence. Using this data, we divided by the population of these regions and multiplied by 1000 (instances per 1000 capita).

Finally, we attempted to quantify the quality of life in a city. For this, we consulted three indices: the New American Economy and the numbeo index. The NAE provided a socioeconomic index for each of the ten cities from 0 to 5. The Numbeo indices provided a safety index and a healthcare index for each of the ten cities from 0 to 100. We scaled all of these numbers to a range of 0 to 1. Clearly, higher is better. For clarity, we will refer to these as "ranking indices."

4.4 Variables

Variable	Definition
$F_{admit}(X,Y)$	The number of individuals admitted into rehab for drug X in the year 2015 of city Y divided by the population and multiplied by 1000.
$F_{crime}(X,Y)$	The number of crime scenes were traces of drug X were found in the year 2015 of city Y divided by the population and multiplied by 1000.
$F(X,Y)$	$\frac{F_{crime}+F_{admit}}{2}$
$\mathbf{G}(X)$	The vector (*F(X, marijuana), F(X, cocaine), F(X, heroin), F(X, opioids), F(X, meth)*)
$a_{1,x}, a_{2,x}, a_{3,x}, a_{4,x}, a_{5,x}$	Five coefficients all initialized to 0.2. Their sum is always 1, and they are the coefficients in the logistic function for each of the drugs. These are the coefficients in the logistic function that predicts the ranking index x
$\mathbf{a}$	The vector $(a_1, a_2, a_3, a_4, a_5)$.
$logits(\mathbf{a}, x, X)$	$$\frac{1}{1+e^{-\mathbf{a}\cdot\mathbf{G(X)}}}$$ for the ranking index x
$y(x, X)$	The ranking index value for the city X for the ranking index x.
$L(\mathbf{a}, y)$	The loss value for a set of coefficients. We used the sum of squares, or $\|\|logits(\mathbf{a}, x, X) - y\|\|^2$

4.5 Model

Using this data, our model is a logistic regression model that correlates drug use with the three indexes of 10 different cities. Given a certain index x and a city X and the set of coefficients **a**, we take the dot product of the drug prevalence factor, $G(X)$ and we plug this into a logistic function. We defined the loss as the squared difference between the predicted value and the actual index value.

The coefficients were constrained for a reason: after fitting our model for all cities, we could simply add up the coefficients for each drug. Then, the lowest coefficients would be the most benign drugs, as they contribute the least to directly lowering the index score and the highest would be the most harmful to the community, as they would contribute the most to directly decreasing the index score.

4.5.1 Model Optimization

Due to the fact that this is a complex constrained optimization problem, there are few methods to minimize the loss. Instead, we opted to use a genetic algorithm. Each generation, every set of coefficients would randomly select a tuple of two coefficients. The first coefficient would be subtracted by a random positive number less than 0.05. And we added this number to the second coefficient. This ensured that the sum always remained one. Then, the loss function was evaluated, and the 50 % of coefficients that resulted in the highest losses were pruned.

5 Results

However, due to a lack of time, we could not finish the model, due to an excess of programming bugs.

5.1 Strengths & Weaknesses

Strengths:

1. Constrained Parameters
   (a) By using a genetic algorithm, we were able to rank the drugs very easily.
2. Logistic Regression
   (a) As with our solutions to the previous problems, we again used the logistic regression. It has many notable properties: boundedness and monotonically decreasing (or increasing).
   (b) It can often predict sociological phenomena and has worked effectively in this problem.
3. Little Assumptions
4. this model is almost entirely data-based, and there are very few assumptions made.

Weaknesses:

1. Data Set
   (a) Our data set only contained the three indices of 10 cities in 2015. (citations) Given more time and computational resources, our data set could include more cities and could include data from various years in order to strengthen the predictive power of our model.
   (b) In addition, our data set is limited to information on substances through users being admitted to rehabilitation programs, and drug-related crime incident reports. This is particularly limiting as it is clearly representative of all substance users.

(c) Crime data for alcohol and nicotine is unavailable, so we were unable to account that into our model.

2. Correlation Vs. Causation

(a) A model simply finds the relationship between the prevalence of particular substances and the well-being of a community. Hence, especially with a limited data set, it is very hard to identify in a quantifiable context whether a drug causes certain changes to a community.

6 Bibliography

[1]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3287367/
[2]https://www.drugabuse.gov/publications/drugs-brains-behavior-science-addiction/drugs-brain
[3] https://download.lww.com/wolterskluwer_vitalstream_com/PermaLink/AA/A/AA_2013_05_23_LEE_203797_SDC1.pdf
[4]https://tobaccocontrol.bmj.com/content/26/4/386
[5] http://www.pewhispanic.org/2018/09/14/facts-on-u-s-immigrants-current-data/
[6] https://samhda.s3-us-gov-west-1.amazonaws.com/s3fs-public/field-uploads-protected/studies/NSDUH-2016/NSDUH-2016-datasets/NSDUH-2016-DS0001/NSDUH-2016-DS0001-info/NSDUH-2016-DS0001-info-codebook.pdf
[7]https://arxiv.org/pdf/1106.1813.pdf
[8]https://www.cdc.gov/tobacco/data_statistics/fact_sheets/youth_data/tobacco_use/index.htm
[9]https://www.cdc.gov/alcohol/fact-sheets/underage-drinking.htm
[10]https://www.childtrends.org/indicators/marijuana-use
[11]https://www.drugabuse.gov/publications/drugfacts/monitoring-future-survey-high-school

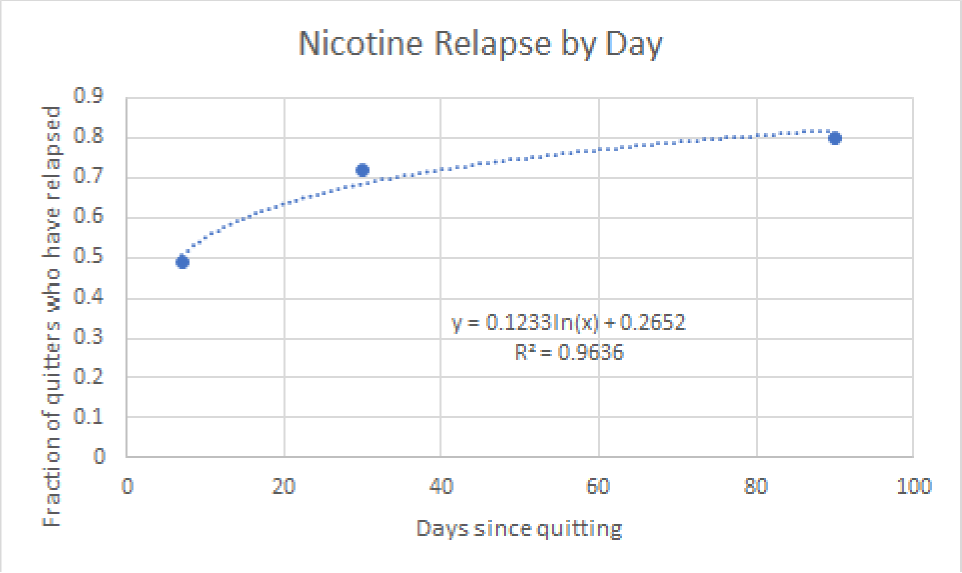
[12]https://www.affirmhealth.com/blog/teens-and-opioids
[13]https://www.drugabuse.gov/news-events/news-releases/2018/12/teens-using-vaping-device

7 Appendices

7.1 Appendix A

Graph of the fraction of nicotine quitters who have relapsed at certain days after quitting. The graph was fitted with a logarithmic trendline. This is so the derivative of the trendline would be the percentage of nicotine quitters who relapse per day which is inversely proportional to days since quitting (a.k.a. hyperbolic, as stated in `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3723776/?fbclid=IwAR0IBLKVfS7D_V7L5mbdAkF1EaQmqZlk-J9ufmut3TveYoK_5jn0UGeoLJ0`, Results section).

Data for the graph provided by `https://www.publichealthlawcenter.org/sites/default/files/Nicotine%20Dependence,%20Relapse%20and%20Quitting%20Smoking.pdf`



The derivative was found to be .1233/x, where x is the days since quitting. This is where we got the relapse factor of .1233 from.

7.2 Appendix B

Age distributions of the United States provided by:
`https://en.wikipedia.org/wiki/Demography_of_the_United_States`
Age distribution:
0–14 years: 18.62% (male 31,255,995/female 29,919,938)
15–24 years: 13.12% (male 22,213,952/female 21,137,826)
25–54 years: 39.29% (male 64,528,673/female 64,334,499)
55–64 years: 12.94% (male 20,357,880/female 21,821,976)
65 years and over: 16.03% (male 22,678,235/female 28,376,817)
Normalized to exclude 0-14 yr olds: (81.38% 15 and up)
15-24 y/o: 16.12%
25-54 y/o: 48.28%
55+ y/o: 35.60%

7.3 Appendix C

The rest of this document is all of the code (written in Jupyter iPython notebooks) we used during the formation of this model.

In [1]:

```
# Import Libraries
import math

from random import random

import csv

import matplotlib
import matplotlib.pyplot as plt
```

In [2]:

```
# Constants
relapse_rate = .1233

population = 100000
days = 3650

teen_population = .1612
adult_population = .4828
old_population = .3560

teen_vape = .054
adult_vape = .074
old_vape = .031

teen_vaped = .166
adult_vaped = .167
old_vaped = .055

teen_rate = 3
immigration_rate = 1
```

In [3]:

```
# Define the person class
class Person(object):
    time_in_state = 0

    def __init__(self, infected, been_infected):
        self.infected = infected
        self.been_infected = infected or been_infected
```

In [4]:

```
susceptible = []
infected = []
```

In [5]:

```
# Population fraction function
def fraction_infected():
    susceptible_count = len(susceptible)
    infected_count = len(infected)
    return infected_count / (susceptible_count + infected_count)
```

In [6]:

```
# Media influence at a certain day
def media_influence(day):
    return .7 + .3 * math.tanh(2*fraction_infected() + day/days)
```

In [7]:

```
# Probability function for a susceptible person to become infected
def probability_infected(person, day):
    original_prob = fraction_infected() * (1 - media_influence(day))
    if person.been_infected:
        return 1 - (1 - original_prob) * (1 - relapse_rate / (person.time_in_s
tate / 10 + 1))
    else:
        return original_prob
```

In [8]:

```
# Probability function for an infected person to become susceptible
def probability_susceptible(person, day):
    return media_influence(day) / (person.time_in_state / 10 + 1)
```

In [9]:

```
# Adds new people to the population following both age distribution and vape
# user distribution depending on age
def new_people(count):
    for i in range(count):
        teen_chance = teen_population
        adult_chance = teen_chance + adult_population

        age_prob = random()
        vape_prob = 0
        vaped_prob = 0

        if age_prob <= teen_chance:
            vape_prob = teen_vape
            vaped_prob = teen_vaped
        elif age_prob <= adult_chance:
            vape_prob = adult_vape
            vaped_prob = adult_vaped
        else:
            vape_prob = old_vape
            vaped_prob = old_vaped

        if random() <= vape_prob:
            infected.append(Person(True, True))
        else:
            if random() <= vaped_prob:
                susceptible.append(Person(False, True))
            else:
                susceptible.append(Person(False, False))
```

In [10]:

```
# Determines who among the susceptible become infected and vice versa.
# Also adds new people due to children becoming teenagers and immigration
def recalculate_population(day):
    global susceptible, infected

    new_susceptible = []
    new_infected = []

    for person in susceptible:
        person.time_in_state += 1
        if random() <= probability_infected(person, day):
            new_infected.append(person)
            person.infected = True
            person.time_in_state = 0
            person.been_infected = True
        else:
            new_susceptible.append(person)

    for person in infected:
        person.time_in_state += 1
        if random() <= probability_susceptible(person, day):
            new_susceptible.append(person)
            person.infected = False
            person.time_in_state = 0
        else:
            new_infected.append(person)

    susceptible = new_susceptible
    infected = new_infected

    for i in range(teen_rate):
        susceptible.append(Person(False, False))

    new_people(immigration_rate)
```

In [11]:

```
%matplotlib notebook
```

In [12]:

```
def run():
    new_people(population)

    with open("vape_data.csv", "w", newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(["Day", "Media Influence", "Susceptible", "Infected",
"Total"])

        for day in range(days):
            if (day % 10 == 0):
                print("Time step:", day)
                print("Media influence:", media_influence(day))
                print("Susceptible:", len(susceptible))
                print("Infected:", len(infected))
                print("")

            writer.writerow([
                day,
                media_influence(day),
                len(susceptible),
                len(infected),
                len(susceptible) + len(infected)
            ])

            recalculate_population(day)
```

In [13]:

```
run()
```

In [ ]:

```
#import - pandas, numpy, sklearn (including LogisticRegression, RFE, Metrics),
SMOTE
import pandas as pd
import numpy as np
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE
```

In [ ]:

```
#loads dataset with specific columns we considered
df = pd.read_csv('/Users/andyliu/Downloads/NSDUH_2016_Tab.csv', usecols=['CIGEVER',
'SMKLSSEVR',
'CIGAREVR',
'PIPEVER',
'ALCEVER',
'MJEVER',
'HEREVER',
'PNRNMLIF',
'HEALTH',
'DIFFHEAR',
'DIFFSEE',
'DIFFTHINK',
'DIFFWALK',
'DIFFDRESS',
'DIFFERAND',
'GOVTPROG',
'INCOME',
'IRINSUR4',
'IRWRKSTAT18',
'SEXIDENT',
'IRSEX',
'NEWRACE2',
'MILTFAMLY',
'IRHHSIZ2',
'NRCH17_2',
'IMOTHER',
'IFATHER',
'IRHH65_2',
'YEPCHKHW',
'YEPHLPHW',
'YEPCHORE',
'YEPLMTTV',
'YEPLMTSN',
'YEYARGUP',
'YEYFGTSW',
'YEYFGTGP',
'YEYHGUN',
'SNYATTAK',
'SNYSTOLE',
'YESTSCIG',
'YESTSMJ',
'YESTSALC',
'YESTSDNK',
'YESCHFLT',
'YESCHWRK',
'YELSTGRD',
'IRWRKSTAT',
'K6SCYR',
'SPDYR',
'YMDELT',
'YSDSOVRL',
'AGE2',])
```

In [ ]:

```
#Some preprocessing; defines the compound variables NICEVR and OPEVER, restricts space to students, and replaces invalid survey responses.
df['NICEVR'] = ((df['SMKLSSEVR'] == 1) | (df['CIGAREVR'] == 1) | (df['PIPEVER'] == 1) | (df['CIGEVER'] == 1)).astype(int)
df['OPEVER'] = ((df['HEREVER'] == 1) | (df['PNRNMLIF'] == 1)).astype(int)
df = df.query('AGE2<7')
df = df.replace(to_replace=['85','94','97','98','99'], value='')
```

In [ ]:

```
df.to_csv('drugdata_students_final.csv')
```

In [ ]:

```
#normalizes data; some manual processing (moving columns around, etc) was done in Excel here
df = pd.read_csv('/Users/andyliu/Documents/drugdata_students_final.csv')
df_norm = (df - df.mean()) / (df.max() - df.min())
df_norm.to_csv('drug_data_normalized.csv')
```

In [ ]:

```
dic = {'alcohol':'ALCEVER', 'marijuana':'MJEVER', 'opioid':'OPEVER', 'nicotine':'NICEVR'}
```

In [ ]:

```
for i in ['alcohol', 'marijuana', 'opioid', 'nicotine']:
    filename = '/Users/andyliu/Documents/LR_' + i + '.csv'

    df = pd.read_csv(filename, header=0)
    df = df.dropna()

    X = df.loc[:, df.columns != dic[i]]
    y = df.loc[:, df.columns == dic[i]].astype(np.int16)

    #runs SMOTE
    os = SMOTE(random_state=0)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, r
andom_state=0)
    columns = X_train.columns
    os_data_X,os_data_y=os.fit_sample(X_train, y_train.values.ravel())
    os_data_X = pd.DataFrame(data=os_data_X,columns=columns )
    os_data_y= pd.DataFrame(data=os_data_y,columns=[dic[i]])

    #runs RFE
    logreg = LogisticRegression(solver='lbfgs')
    rfe = RFE(logreg, 12)
    rfe = rfe.fit(os_data_X,os_data_y.values.ravel())
    print(i + " results:")
    rfe_approved = (rfe.support_).tolist()
    rfe_indices = []
    for j in range(0, len(rfe_approved)):
        if rfe_approved[j]:
            rfe_indices.append(j)

    print(df.iloc[:, rfe_indices].columns)

    #logistic regression
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, r
andom_state=0) #this line breaks
    logreg = LogisticRegression(solver='lbfgs')
    logreg.fit(X_train, y_train)

    #statistics (confusion matrix, accuracies)
    y_pred = logreg.predict(X_test)
    print(y_pred)
    cm = confusion_matrix(y_test, y_pred)
    print(cm)

    print('Accuracy of logistic regression classifier on test set: {:.2f}'.for
mat(logreg.score(X_test, y_test)))
    print(classification_report(y_test, y_pred))
```

In [1]:

```
# importing necessary packages
import pandas as pd
import numpy as np
```

In [2]:

```
# reading csv
df = pd.read_csv('drugcitydata.csv', row)
```

In [3]:

```
# what the data looks like
df
```

Out[3]:

	admit_mj	report_mj	admit_coc	report_coc	admit_meth	report_meth	admit_her	rep
0	1342	380	845	3359	736	5328	631	111
1	4613	30090	2059	9957	103	620	6335	116
2	2594	2771	886	1499	2290	2771	2313	142
3	1042	3699	1556	1381	12	29	5207	998
4	4835	7490	1332	3919	7626	10610	9392	201
5	4779	3991	2088	7763	166	399	2366	165
6	15347	12333	8596	13989	471	532	26217	668
7	1086	5880	676	6001	2	108	1206	494
8	1180	102	432	214	911	418	3016	431
9	584	42	922	18	1465	32	4125	33

In [21]:

```
# list of columns to be modified
modifiers = ['admit_mj', 'report_mj', 'admit_coc', 'report_coc', 'admit_meth',
'report_meth', 'admit_her', 'report_her',
            'admit_ops', 'report_ops']
```

In [22]:

```
# cleaning data
data = []
for index, row in df.iterrows():
    pop = row['pop']
    new = []
    for mod in modifiers:
        x = 1000 * row[mod] / pop
        new.append(x)
    new.append(row['nae'] / 5)
    new.append(row['safe'] / 100)
    new.append(row['health'] / 100)
    data.append(new)
```

```
In [23]: # list of new columns
         cols = ['admit_mj', 'report_mj', 'admit_coc', 'report_coc', 'admit_meth', 'rep
         ort_meth', 'admit_her', 'report_her',
                     'admit_ops', 'report_ops', 'nae', 'safe', 'health']
```

```
In [24]: len(data[0])

Out[24]: 13
```

```
In [25]: df2 = pd.DataFrame(data=data, columns=cols)
```

```
In [26]: df2

Out[26]:
```

	admit_mj	report_mj	admit_coc	report_coc	admit_meth	report_meth	admit_her	rep
0	0.231791	0.065634	0.145949	0.580168	0.127122	0.920255	0.108987	0.1
1	0.485579	3.167368	0.216737	1.048105	0.010842	0.065263	0.666842	1.2
2	0.894483	0.955517	0.305517	0.516897	0.789655	0.955517	0.797586	0.4
3	0.242537	0.860983	0.362176	0.321443	0.002793	0.006750	1.211987	0.2
4	0.477767	0.740119	0.131621	0.387253	0.753557	1.048419	0.928063	0.1
5	0.775960	0.648013	0.339026	1.260468	0.026953	0.064785	0.384164	0.2
6	1.788278	1.437078	1.001631	1.630040	0.054882	0.061990	3.054882	0.7
7	0.178942	0.968858	0.111386	0.988796	0.000330	0.017795	0.198715	0.8
8	0.557129	0.048159	0.203966	0.101039	0.430123	0.197356	1.423985	0.2
9	0.076042	0.005469	0.120052	0.002344	0.190755	0.004167	0.537109	0.0

```
In [28]: # normalize between 0 and 1

         normalized_df=(df2-df2.min())/(df2.max()-df2.min())
```

In [29]:

```
normalized_df
```

Out[29]:

	admit_mj	report_mj	admit_coc	report_coc	admit_meth	report_meth	admit_her	rep
0	0.090962	0.019028	0.038824	0.354995	0.160634	0.877267	0.000000	0.1
1	0.239183	1.000000	0.118339	0.642480	0.013318	0.058507	0.189367	1.0
2	0.477995	0.300468	0.218065	0.316123	1.000000	0.911035	0.233749	0.3
3	0.097239	0.270570	0.281709	0.196044	0.003121	0.002474	0.374419	0.1
4	0.234620	0.232344	0.022729	0.236475	0.954267	1.000000	0.278040	0.1
5	0.408774	0.203215	0.255705	0.772948	0.033730	0.058050	0.093411	0.2
6	1.000000	0.452769	1.000000	1.000000	0.069113	0.055373	1.000000	0.6
7	0.060097	0.304687	0.000000	0.606042	0.000000	0.013051	0.030459	0.6
8	0.280970	0.013501	0.103994	0.060635	0.544507	0.185003	0.446383	0.1
9	0.000000	0.000000	0.009735	0.000000	0.241251	0.000000	0.145329	0.0

In [31]:

```
data_final = []
for index, row in normalized_df.iterrows():
    new = []
    for i in range(5):
        val = (row[modifiers[2*i]] + row[modifiers[2*i+1]]) / 2
        new.append(val)
    new.append(row['nae'])
    new.append(row['safe'])
    new.append(row['health'])
    data_final.append(new)
```

In [32]:

```
col_finals = ('mj', 'coc', 'meth', 'her', 'ops', 'nae', 'safe', 'health')
```

In [34]:

```
df_final = pd.DataFrame(data=data_final, columns=col_finals)
```

In [35]:
```
df_final
```

Out[35]:

	mj	coc	meth	her	ops	nae	safe	health
0	0.054995	0.196910	0.518951	0.076715	0.493843	0.726562	0.292839	0.079558
1	0.619591	0.380410	0.035913	0.594684	0.216571	0.898437	0.272040	0.539227
2	0.389232	0.267094	0.955518	0.316159	0.519342	0.000000	1.000000	0.825414
3	0.183904	0.238877	0.002798	0.280361	0.375002	0.882812	0.000000	0.028729
4	0.233482	0.129602	0.977134	0.218775	0.201956	0.687500	0.373671	0.262983
5	0.305995	0.514326	0.045890	0.154871	0.798831	0.507812	0.373671	0.060773
6	0.726384	1.000000	0.062243	0.816257	0.825627	1.000000	0.615221	0.000000
7	0.182392	0.303021	0.006526	0.346031	0.402313	0.937500	0.216734	1.000000
8	0.147236	0.082314	0.364755	0.304576	0.251110	0.804687	0.808792	0.996685
9	0.000000	0.004867	0.120626	0.072664	0.076008	0.976562	0.780194	0.850829

In [36]:
```
df_final.to_pickle('cleaned.pkl')
```

In [ ]:

```
import pandas as pd
import numpy as np
import random
import math
from copy import copy, deepcopy
```

In [ ]:

```
df = pd.read_pickle('cleaned.pkl')
```

In [ ]:

```
df
```

In [ ]:

```
COUNT = 10
RANGE = 0.05
```

In [ ]:

```
cities = []
for i in range(10):
    species = []
    for i in range(COUNT):
        initial = [0.2, 0.2, 0.2, 0.2, 0.2, 0]
        species.append(initial)
    cities.append(species)
```

In [ ]:

```
def evaluate(specie, city_data, y):
    specie = specie[:-1]
    x = np.dot(specie, city_data[:-3])
    value = 1 / (1 + math.exp(-x))
    loss = (value - y) ** 2
    return loss
```

In [ ]:

```
def mutate_spec(spec, city_data, y):
    copy = spec
    tup = random.sample(range(0, 5), 2)
    diff = RANGE * random.random()
    copy[tup[0]] += diff
    copy[tup[1]] -= diff
    x = evaluate(spec, city_data, y)
    copy[5] = x
    return copy
```

In [ ]:

```
testrow = df.iloc[0].tolist()
```

In [ ]:

```
testrow
```

In [ ]:

```
species1 = []
for i in range(COUNT):
    initial = [0.2, 0.2, 0.2, 0.2, 0.2, 0]
    species1.append(initial)

species2 = []
for i in range(COUNT):
    initial = [0.2, 0.2, 0.2, 0.2, 0.2, 0]
    species2.append(initial)

species3 = []
for i in range(COUNT):
    initial = [0.2, 0.2, 0.2, 0.2, 0.2, 0]
    species3.append(initial)
```

In [ ]:

```
species1
```

In [ ]:

```
def mutate_specs(specs, city_data, y):
    copy = specs
    news = []
    for spec in specs:
        new = mutate_spec(spec, city_data, y)
        news.append(new)
    return news
```

In [ ]:

```
def mutate_city(spec1, spec2, spec3, city_data):
    storage1 = deepcopy(spec1)
    storage2 = deepcopy(spec2)
    storage3 = deepcopy(spec3)
    l1 = storage1+mutate_specs(spec1, city_data, city_data[5])
    l2 = storage2+mutate_specs(spec2, city_data, city_data[6])
    l3 = storage3+mutate_specs(spec3, city_data, city_data[7])
    l1 = sorted(l1, key = lambda x: int(x[-1]))
    l2 = sorted(l2, key = lambda x: int(x[-1]))
    l3 = sorted(l3, key = lambda x: int(x[-1]))
    return [l1[:10], l2[:10], l3[:10]]
```

In [ ]:

```
def avgloss(species):
    total = 0
    for spec in species:
        total += spec[-1]
    return total / len(species)
```

In [ ]:

```
row = df.iloc[0].tolist()
i=0
for j in range(10000):
    new = mutate_city(species1, species2, species3, row)
    species1 = new[0]
    species2 = new[1]
    species3 = new[2]
    if i%100 == 0:
        print(i)
        print(avgloss(species1))
    i+=1
```

In [ ]:

```
for i in range(10):
    row = df[i]
```

In [ ]:

```
print(species1)
```