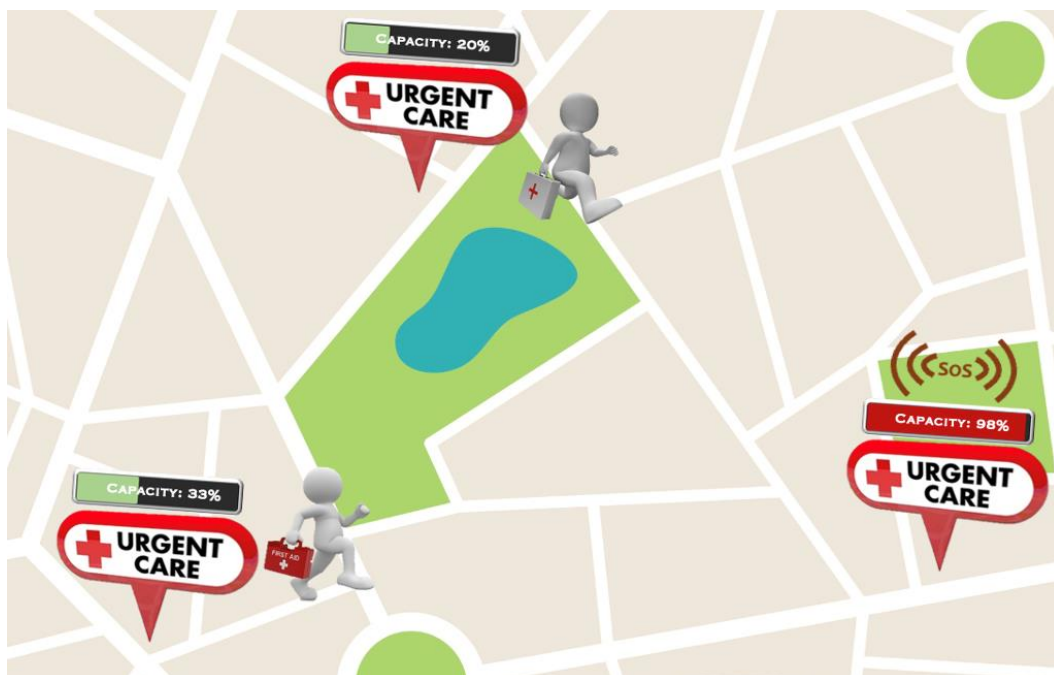# TecNav

Clinic-Monitoring & Technician-Navigation Application

Romith Challa
Tomer Danon
Data Science Capstone
University of Denver

Executive Summary

According to a 2019 benchmark report from the Urgent Care Association, the number of urgent-care clinics increased by 9.6% from the previous year[1]. As the pandemic continues to put a strain on the healthcare system, the appeal for convenient and affordable care-outlets is greater than ever. To meet this growing need for on-demand care, urgent-care clinics have been rapidly expanding the scope of their services to provide patients with a quick and affordable alternative to the emergency room. This proves to be a mutually exclusive endeavor as minimizing wait-times entails maximizing staff availability. Consequently, this leads to higher operational costs that eventually trickle down to patient bills. Due to the inconsistency in patient traffic, it is difficult to schedule staff to be able to handle the peak hours, without inevitably wasting resources during less busy hours. Some chain-operated clinics creatively handle this by transferring technicians between their various locations to account for the constantly evolving needs at each clinic. This requires continuous monitoring of each clinic to manually make navigation decisions based on the fluctuating patient traffic. Instead, a software application that streamlines this process to make real-time, data-driven navigation decisions would serve as an invaluable tool for these clinics. That is the motivation behind *TecNav*—an automated application that employs machine-learning to forecast the dynamic patient flow and navigate technicians without any human supervision. Based on the preliminary results of the prototype, *TecNav* can save clients over $90,000 for each clinic per year. This could potentially finance their mission to expand clinical services, without burdening patients with higher costs. Designed with an emphasis on translatability, *TecNav*'s algorithm is customizable to fit the needs of any potential client.

Project Overview

To better organize a project of this scale, it was broken into two distinct phases to separate the data science portion from the software engineering components. The first phase involved research-based fabrication of the necessary datasets, an exploratory analysis of the generated data, and the construction and evaluation of machine-learning models that would serve as the decision-makers within the application. The objective of the second phase was the design and development of the software algorithm and simulation of a "real-time" stream to evaluate *TecNav* on its ability to make optimal navigation decisions.

Data Fabrication

Due to the unique scope of this project, it was difficult to source a singular, pre-constructed dataset that would be conducive for execution. In fact, stitching together a collection of public datasets would still be insufficient as this project requires specific information that is not widely available. Starting at the broadest level, the first pre-requisite is the fundamentals of a particular chain-operated urgent-care system. This includes the names of their various facilities, the geographical coordinates, and the distances or anticipated travel-time between each pair of clinics. Next, an employee database that contains staff members and their corresponding roles is necessary for any scheduling or navigation tasks. Lastly, a set of patient records consisting of logged check-in and check-out times and reasons for visit is imperative for the modeling stage of the project. Although some ancillary materials are available through APIs and behind paywalls, these datasets are still missing the primary attributes needed to carry out this project. Despite an unsuccessful endeavor of finding a publicly available source that reasonably fits the mold, this project was still worth pursuing as it has immense applicability in today's healthcare landscape. Therefore, the best action plan was to fabricate the necessary datasets for building a prototype application that will be demonstrated to prospective clients. To that extent, this stage was strategically approached to emulate real-world data as closely as possible. The objective was to generate a compilation of translatable datasets that would contain attributes that can be expected of client-provided data. Secondary research materials were referenced, and personal work experience was relied on to gather the patterns and characteristics of urgent-care clinics in order to build datasets that were representative of the real-world. While this spared the need for extensive data cleaning, there were still plenty of opportunities for feature-engineering that demanded meticulous planning. Finding balance between adding randomized variations to represent the imperfections of real data while still maintaining a sensible amount of consistency proved to be a challenging task.

Clinics

For the purpose of this project, five fictitious clinics were chosen in the following municipalities of the Denver Metropolitan area: Denver, Wheat Ridge, Edgewater, River North Arts District (RiNo), and Lakewood. These locations represent a realistic spread of distances from one another, and their clinic capacities were designed to reflect the population of each corresponding area. Initially, the Google Maps API[2] was explored for potentially gathering distances and traffic-based driving times in real-time. However, these services ended up being pay-per-utility which made it unfeasible for building this

prototype project. Therefore, the distances and drive times were manually retrieved instead, with the goal of adding randomized variations to mimic traffic delays during the navigation phase of the project.

Employees

To generate employee records, names were drawn from the *Faker* module and corresponding two-digit IDs were created to serve as unique identifiers. A total of fifty medically trained employees were fabricated for the set of five clinics. Fifteen of these clinicians were distinguished as "providers"—the umbrella term chosen to classify primary caregivers (i.e., physicians and physician-assistants). The remaining thirty-five members were categorized as "technicians" to represent the medical assistants, scribes, and lab technicians. Their duties entail conducting initial examinations, performing lab or blood work, and filling paperwork[3]. As technicians are not bound to any specific patient, these employees would be transferable between the various locations, without interfering with the care-giving process.

Patients

Patient registration data is arguably the most important pre-requisite for the execution of this project. Ideally, these records would contain personal patient information, as well as the details pertaining to their actual visit. To serve as the source for initial explorations and model training, past patient logs were generated to encompass an entire year of past data (*May 2021 - April 2022*). Additionally, another month patient records were also created to be used as the test-set during the evaluation stage (*May 2022*). Each patient was assigned a *Faker*-generated name, along with a unique 7-digit ID. To reflect real-world patient demographics, age breakdowns closely adhered to the distributions available in the Journal of Urgent Care Medicine, which surprisingly showed 21-30 to be the most popular age-group[4]. This intuitively makes sense since younger populations without extensive insurance coverage look for convenient and affordable care options. *Faker* module was utilized to generate a set of date-of-births for each patient to accurately reflect the proportions of these age groups.
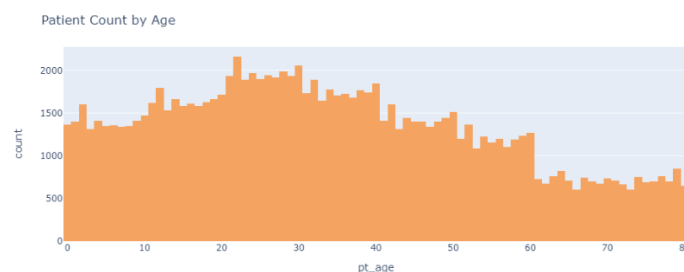

Figure 1 - Age Distribution

Visits

An important component of the patient records is their reason for visit. Unlike the age breakdown, secondary research did not yield a real-world set of proportions to replicate. Instead, several sources were referenced to gather the actual common reasons for visit. Although CDC provided a similar study for ER visits, it would be dangerous to equate ER data to urgent-care experiences as these care-facilities see a completely different patient demographics[5]. On the other hand, relying purely on personal work experience would be biased to that singular experience. Therefore, these were all referenced in tandem to tune the number of patients that came in for each reason as best as possible. Fortunately, this specific component of the fabrication stage is not dependent on replicating the exact proportion values—rather, capturing the real-world variation that would exist between different visit reasons is the true objective. Therefore, the existence of the variation itself is the most important data characteristic for this feature. Furthermore, each visit reason was distinguished by a severity code. This code reflects the level of urgency a provider would place on seeing the patient. The set {3,4,5} was chosen to follow a 3-tiered system based on work experience at urgent-care clinics in St. Louis, MO. For instance, a patient citing chest pain would get priority over a patient coming in for a physical to avert any immediate dangers. In addition to aiding the process of fabricating other features, this attribute will come in handy during the navigation phase.
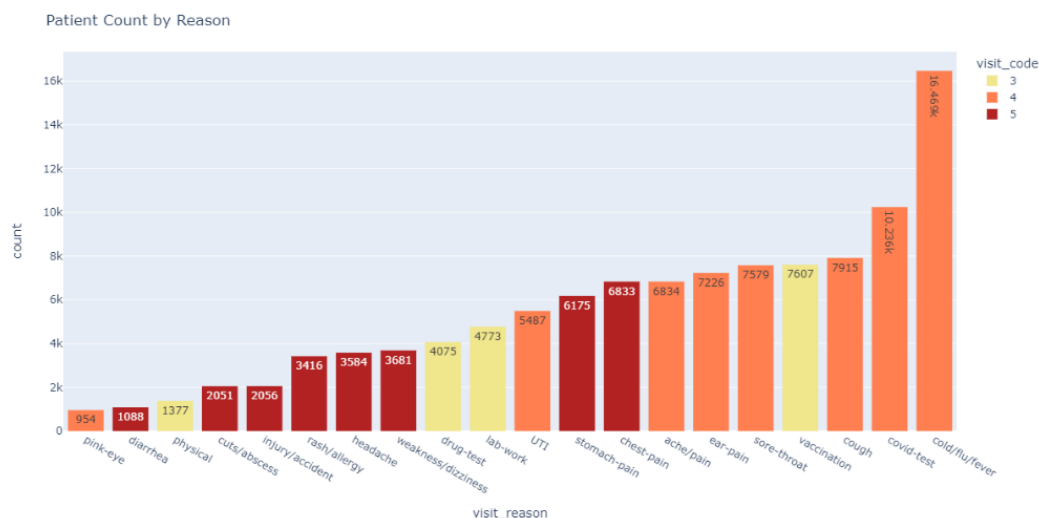

Figure 2 -Visit Reason Distribution

With each visit reason, it was important to fabricate a corresponding value of expected visit length that would determine patient check-out times. As with visit proportions, secondary research proved to be moot for this segment. Therefore, personal work experience was relied on to create a base length for each visit, which will then be varied to add randomized fluctuations. As with the previous section, the variation

itself is the primary objective. Overall, level-5 severity code tends to take the longest. The visit length is often correlated to the severity code in the real-world, as higher codes usually indicate the need for extensive diagnostic measures such as imaging or advanced therapeutics such as intravenous fluids.
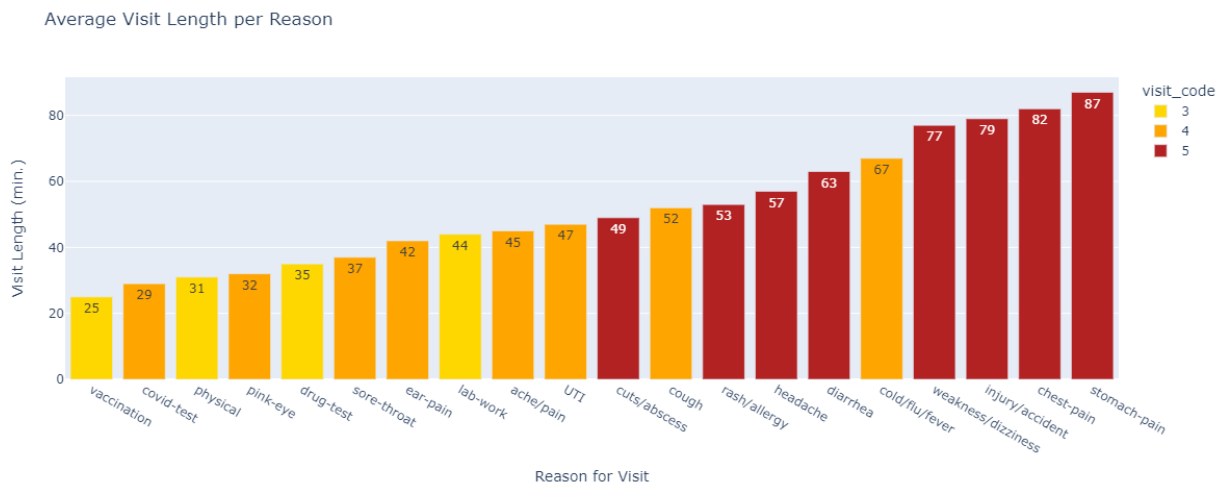

Figure 3 - Average Visit Length

      After the ancillary attributes were fabricated, the next step was to generate the central components of this dataset: the patient check-in and check-out times. Unlike the previous features, capturing patient traffic to best emulate the real-world was crucial for the navigation results to be a realistic reflection of client expectations. Therefore, the "*popular times*" feature in Google Maps was referenced to study the patterns of patient traffic in urgent-care clinics of the Denver area[2]. It became immediately apparent that multi-modal distributions best capture this data. Patient influx seemed to be characterized by periods of high peaks and low valleys that vary based on day and location. Through an exploration of various statistical Python libraries, these times could be mimicked through the combination of several different normal distributions with varying means (μ) and standard deviations (σ). Therefore, this stage was set up by creating a dictionary for each location with a different set of possible μ and σ values for the peak of each day. These values were randomly drawn in the process of generating the check-in times for each patient record. Based on these check-in times and the generated visit lengths, check-out times were computed with added variation to mimic real-world fluctuations.

      With these check-in and check-out times, other important attributes were engineered to be used in stages further down the project pipeline. One of these attributes is the patient rolling count—this signifies the number of patients currently at the clinic. Ultimately, this variation of patient traffic between clinics presents the opportunity for the navigation of technicians throughout the day.
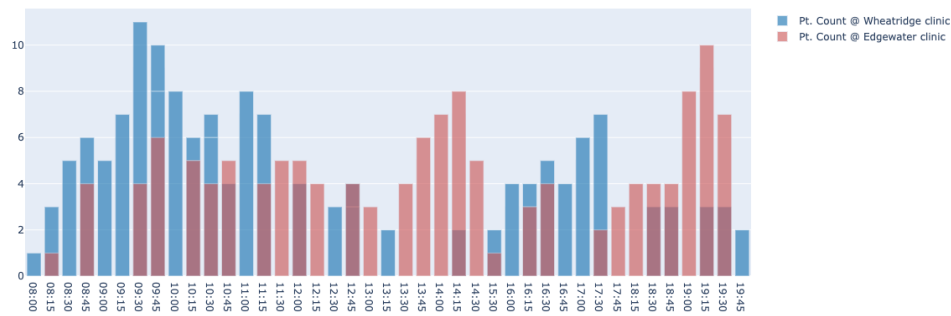
Figure 4 - Rolling Patient Count (Wheat Ridge, Edgewater, 05/19/21)

As part of the data engineering process, additional features such as rolling severity code were extracted based on the average severity code of patients at the clinic at a given moment. Additionally, the number of needed technicians based on the dynamic rolling count was extracted to study the extent to which resources are wasted when scheduling solely based on peak hours. Shown in the plot below, this highlights the opportunity for technician transfers using a navigation software that accounts for the daily traffic flow.
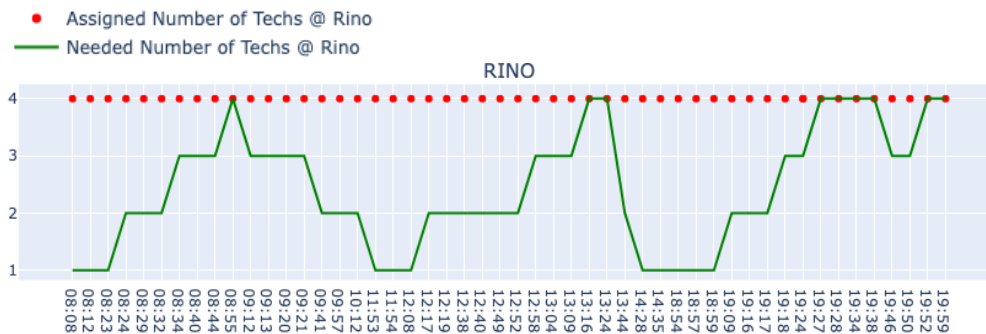

Figure 5 - Assigned Vs. Needed Technician Count (RiNo, 05/01/21)

Modeling

Past work experience has shown that in the real-world, urgent-care chains can benefit from navigating technicians between their clinics. Prior to this, clinics would review past data and schedule technicians based on the anticipated peak hours. In order to avoid excessive wait times and over-worked technicians, it is important to maintain an ideal patient to technician ratio. The technicians would be scheduled for a particular day based on this ratio and the past peak hours. This in turn leads to inefficient use of resources, as the needed technicians for peak hours are not needed all day. Given that the peak

hours are slightly different across the chain of clinics, there is an opportunity to optimize the scheduling and navigation of technicians. This requires a human navigator to monitor the clinics throughout the day and make real-time decisions to initiate the transfers. For a clinic to operate under normal conditions, the ratio of patients to technicians must be less-than-or-equal-to the ideal ratio. If this threshold is exceeded, it signifies that the clinic is getting busy. If another patient checks-in to that clinic, and the ratio is still greater than ideal, the navigator would look to see if there are technicians available for transfer at the other clinics. If the ratio is less than the ideal, it signifies that there are technicians available for transfer. If this is the case, the navigator will initiate a transfer. It can get risky, however, when the available number of technicians at the other clinics is one. It is possible that this technician will be needed in the next hour in the clinic they are currently at. This is something that the navigator would have to decide based on their experience or hunch. Instead, a machine-learning model would provide a more data-driven approach. *TecNav* aims to remove the human supervision from this process and replace them with a machine learning model that would make the decisions based on past data it was trained on.

Regression

In order to substitute human intuition for a data-driven approach, a machine-learning model that is trained on past patient logs has the potential to serve as a decision-maker in the application. While there are numerous ways to approach this setup, the feature-engineered attribute 'needed number of technicians' would serve as an ideal target. In the real-world, it is difficult to plan for an unknown, client-provided feature-set. Thus, a target that is extracted from a basic feature such as the 'patient rolling count' maximizes the chances of model translatability to any client. Since this attribute is a numeric variable, albeit with a small discrete range, this is a task suited for regression.

Several pre-processing steps took place to prepare the data for modeling. The visit day information was converted to Boolean values to designate if it was a weekend. Since the ultimate goal of this model is to predict how many technicians will be needed the next hour, check-in times were rounded down to the nearest hour to standardized times of day. The visit location was encoded to reflect its multi-level categorical nature. Lastly, rolling severity code was scaled to prepare for modeling. This data was partitioned to use the past year (*May 2021 – April 2022*) as the training set, and the new data (*May 2022*) as the test set.

Once the feature set was finalized and the data were partitioned, a number of baseline regressors were constructed and evaluated to model the data before further tuning. Linear-based techniques such

as Lasso, Ridge, ElasticNet, and Linear Regression were employed to study the effects of different penalization weights on the final predictions. Non-linear based techniques were also employed such as the distance-based K-Nearest Neighbors, and the tree-based Decision Tree and Random Forest. These baseline models were evaluated among relevant regression metrics. The highest consideration was given to the root-mean-squared error, as RMSE score is measured in the same units as the target variable. This would enable easier communication with prospective clients. From this, it was evident that the tree-based models outperform all others and could be tuned further to see if the scoring can be improved. Given that the decision tree performs equally well when compared to the random forest, it proves to be an appealing model of choice for further tuning. This is due to its simpler model-complexity, and therefore, it would be easier to explain to stakeholders. However, there are other advantages to selecting random forest for the final implementation. As this project is being conducted on fabricated data with a limited feature-set, using an ensemble tree model might prove to be more robust and translatable when working with client-given datasets. Therefore, the random forest regressor was further tuned and optimized through a cross-validated grid-search. Although this resulted in only negligible improvements, the model generalized well to the test-set. It should be noted that this is to be expected, since both the train and test sources were fabricated. An RMSE score of 0.72 indicates that the model is off by approximately 1 when predicting the number of technicians a clinic needs at a given time. Plotting the learning curve shows there is no immediate danger of model overfitting, as the increased number of train instances leads to a better validation RMSE score. Examining the feature importance showed the hour of the day to be the biggest contributor to the model.

Constructing a model with a limited feature-set makes it easily translatable to client-provided data. Despite this advantage, the predictive power of new datasets is a challenge that cannot be planned for. To hedge against these risks, it would be prudent to explore modeling techniques that do not rely on specific attributes other than the target variable itself. To that extent, an alternative forecasting approach was investigated - Time-series.

Time-series - ARIMA

Autoregressive Integrated Moving Average (ARIMA) is a popular time-series technique that has immense applicability in the context of rolling target values[6]. The "autoregressive" (AR) component refers to the use of past target values at predetermined time intervals to predict a future point. On the other side, the "moving average" (MA) part incorporates the error terms at each past interval into the regression equation. A pre-requisite for executing ARIMA is conducting an Augmented Dickey-Fuller test (ADF) to

determine if the data are stationary. In other words, this test ensures the mean stays constant throughout the time-series[7]. If this test fails, some differencing steps must be implemented to transform the non-stationary data. The extent to which this is done is noted by the "integrated" (I) portion of the ARIMA acronym.

Since the time-series implementation only considers a rolling target (i.e., the needed number of technicians), there is no direct way to incorporate other features such as the specific clinic branch. Unlike traditional machine-learning models, ARIMA requires a separate execution for each clinic to account for the varied location-based patient-traffic. After exploring potential series lengths, one-week proved to be the most tenable. Examining the target on a more micro, day-to-day level would often be characterized by non-stationary values. This would lead to additional differencing steps for each day that unnecessarily increase model-complexity at every stage of the execution. On the other hand, examining the target on a more macro, month-to-month basis would lead to different issues. Since the patient traffic was fabricated to reflect the real-world variation that exists based on the day, each week of data essentially represents a "season" for time-series analyses. This would require incorporating an additional parameter that leads to a specialized ARIMA model known as SARIMAX[8]. This in turn, also scales the model-complexity, which must be weighed against the additional benefit of using a longer series. Since the objective of the modeling stage was to construct a forecaster that can be implemented for repeated use within a software application, great consideration was given to avert any model-complexity issues. Thus, one-week series is the ideal choice. Implementation of ADF tests consistently showed weekly data to be stationary.
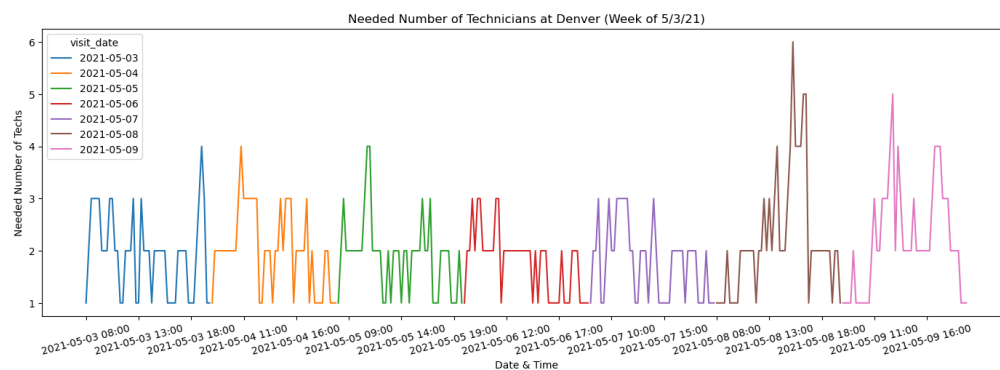
Figure 6 - Needed Number of Technicians (05/03/21 - 05/09/21, Denver)

Once the series length was decided, the next step was to engineer the weekly patient records based on a strategic set of time intervals, known as lags. First, the check-in time of each patient record was rounded down to the nearest 15-minute mark, and the mean target value was aggregated based on each grouped 15-minute interval. Forward-fill methodology was followed to account for missing interval

marks to sustain the needed number of technicians from the previous interval. To help decide which of these pre-processed intervals should be used for ARIMA forecasting, autocorrelation (ACF) and partial autocorrelation (PACF) of the intervals were plotted. The objective of these plots is to single out a particular number of lags that are outside of the shaded area in order to determine the interval parameter for the MA and AR components, respectively[9]. However, literature suggests that if the series is stationary and is autocorrelated, the first-order model could suffice[10].
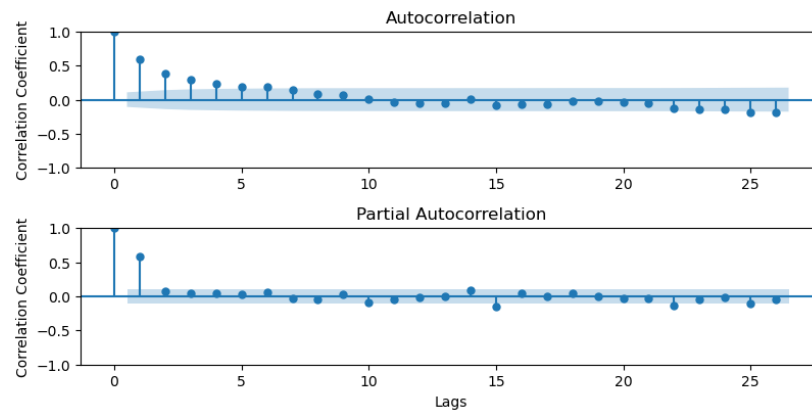

Figure 7 - ACF (top), PACF (bottom) - 05/03/21-05/09/21, Denver

Despite these efforts, the initial ARIMA forecasts were discouraging. Further research into these methods led to a common methodology in time-series techniques known as walk-forward validation. This method incorporates the new target value at each interval into the training for the prediction at the next interval, yielding improved forecasting results[11].
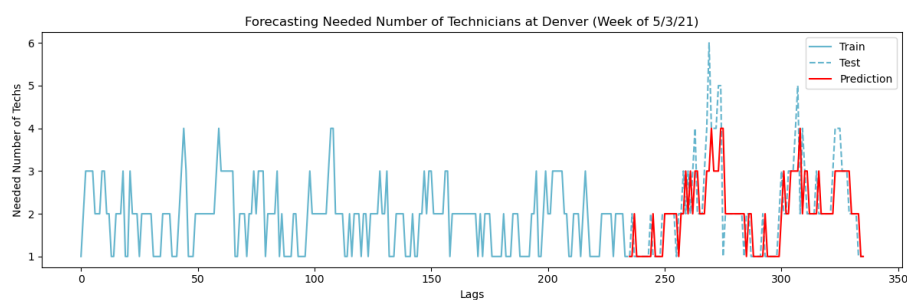

Figure 8 - ARIMA results - 05/03/21-05/09/21, Denver

Although this strategy has immense potential to improve the forecasts, re-training at each stage implies higher time-complexity. Therefore, to determine if the results are worth the expense, several ARIMA models were trained and evaluated for each week of May 2021 for each location. The resulting RMSE scores only showed marginal improvements from the original random forest regressor, which did

not require any re-training. Therefore, incorporating the random forest model into the software application proved to be the smarter choice at this stage. However, this exploration of time-series techniques gives immense flexibility when working with a potential client that may not have sufficient or predictive data to provide. This also opens the door for further investigation into improving time-series techniques without burdening clients with higher computational demand to run *TecNav*.

TecNav Algorithm

The *TecNav* algorithm is the core of the application. It monitors the status at the clinics throughout the day and uses a machine-learning model in specific situations to make navigation decisions. In order to monitor the activity at the clinics throughout the day, a dynamic dictionary is created that will log the changes in number patients and technicians after each check-in and transfer. In addition, it has a key named 'flag' to signify that a clinic is getting busy. A clinic getting busy means that the number of technicians needed, based on the ideal ratio, is greater than the number of technicians currently at the clinic. Anytime a patient checks into one of the clinics, the algorithm begins the work with an update to the dynamic dictionary. Next, it looks for the flag to see if the clinic that the patient checked into was getting busy. If it was, then it is important to find out if the clinic is *still* busy, or if the flag was updated due to a short rush and help is no longer needed. If the clinic was no longer busy, no action would be taken. Otherwise, it means the clinic could use an extra technician. At this point, the algorithm refers to the dynamic dictionary to check which of the nearby clinics have availability and sorts them in a list by the driving time. If they have availability, it means that the number of technicians at the clinic is greater than the number of technicians needed there. If this number is greater than 1, then the transfer gets initiated, the dynamic dictionary gets updated, and the algorithm will restart when the next patient checks into one of the clinics.

Prior to the implementation of *TecNav*, the human navigator would have to decide when to initiate a transfer. If the available number of technicians is 1, *TecNav* would use the machine-learning model to predict how many technicians are needed in the next hour and assess whether or not this technician could be transferred. The human navigator would have to base this decision on their hunch or potentially biased experience. The model makes decisions based on past data it was trained on. Frequent re-training of the model could help discover health trends and improve prediction accuracy. If the model predicts that the technician will be needed at the clinic they are at, the algorithm will move to the next clinic in the availability list. If none of the clinics are able to transfer, the algorithm will note this, and no

action would be taken. If the model assess that the technician can be transferred, it will initiate the transfer and update the dynamic dictionary. Finally, the algorithm will restart when the next patient walks into one of the clinics.

Application Development

The *TecNav* algorithm was translated into an application through object-oriented programming that allows for the integration of several customizable parameters. This application is set up to log navigation activity and print out clinical status at each step throughout the day. In order to demonstrate the prototype algorithm in action, an interactive dashboard was set up through *Streamlit*. This enables a prospective client to choose any past day to simulate incoming patient records, track what navigation movements would have occurred, and observe the optimization of staff allocation throughout that day. This dashboard consists of an animated map of rolling patient count to track the evolving needs at each clinic. It also provides a visual that highlights the dynamic technician count in response to the navigations. If a clinic already has a manual navigation system in place, this prototype application will enable them to compare and contrast a past day of activities. On the other hand, if a clinic chain is adopting a navigation system for the first time, this demo will detail the process for a deeper understanding.

Recommendations

TecNav was designed with an emphasis on translatability and thus customizable to fit the needs of any potential client. The needs of each client vary depending on their clinic locations, and their business model. *TecNav* was evaluated on a series of parameters, such that numerous scenarios can be presented to clients. This was done through a manual grid search of three parameters: ideal patient-to-staff ratio, scheduled number of technicians for the day, and rolling code. The grid-search uses the cartesian product of the given parameters and represents every possible scenario.

Parameters

Each clinic would like to maintain an ideal ratio of patients-to-technicians. This ratio is chosen based on past-patient peak-traffic and aims to avoid overworking the staff. Based on past work experience, a 3:1 ratio was the default value used in development. For the grid search, the ratios of 2:1 and 4:1 were also examined.

Prior to the implementation of *TecNav*, the scheduled number of technicians for the day was based on the peak hours of past-patient data. Given *TecNav*'s goal is to decrease the number of technicians scheduled per day, the grid-search was setup to evaluate the past data starting with 1 and 2 less technicians per day than prior to implementation.

There are often situations where the current number of patients at the clinic exceeds the ideal ratio, but given the reason of the visit, this does not always require the initiation of a transfer. For example, if a family of four checks in to get tested for COVID-19, the average rolling code does not increase significantly. This type of visit is routine and has a consistent visit-length regardless of wait-times. Therefore, it doesn't overwhelm the clinic to the point help is needed. The grid search scenarios included rolling code thresholds of 0, 4.2, 4.4, and 4.6. The development stages of *TecNav* used the default value of 0, meaning the rolling code was not considered when the algorithm assess whether to transfer. The other values were chosen to reflect times in the day where the majority of patients at clinic have high severity codes.

Evaluation

The evaluation metrics of the grid-search were chosen meticulously. Their purpose is to help a client decide which parameters are best for their business model and clinical needs. The following metrics are important evaluators of the *TecNav* application: number of transfers, percentage of transfers that were reversed within an hour, percentage of instances that adhere to the ideal ratio, percentage of transfers that could not be fulfilled, and instances where the availability at another clinic is of only one technician. These parameters can be evaluated on any time period from one day to a number of years.

As the number of transfers increase, so do the gas reimbursements to the transferred technicians. Therefore, some clients may desire to minimize this parameter. A greater number of transfers could also signify inefficient scheduling – if a clinic transfers a technician and shortly-thereafter requests a transfer back, then there is probably a different set of parameters that will better suit this situation.

The total number of instances where a technician was transferred and needed back within the hour is reflected in the parameter, percentage of moves of that were reversed within the hour. Closely related to this metric, is the percentage of transfers that could not be fulfilled. While transferring a technician and needing one back within the hour can be an inconvenience, not fulfilling a request for help could be more hurtful to the business and the flow of the day. The importance of each of these is determined by the client and their needs.

TecNav was designed to deploy a machine-learning model when the availability at a clinic is of only one technician. There is a probability that this technician would be needed at the clinic they are currently at, as this clinic could experience an increase in traffic over the next hour. Therefore, this scenario is important to keep track of. Ideally, there is only one technician available, and they will not be needed in the next hour. Keeping track of this parameter gives insight into how well the parameter choices work with the business flow of the client.

Results

If a client's primary motive is to minimize expenditure and maximize savings, the grid-search results recommend increasing the maintained ratio of patients-to-technician, while scheduling two less technicians to begin each day than prior to implementation. However, this extreme measure would result in overworked technicians and yields a lower percentage of threshold adherence. On the opposite side of the spectrum, if the client prefers to implement a navigation system that strictly adheres to a pre-determined ratio, TecNav recommends maintaining a ratio of 2:1 or 3:1 and scheduling a higher number of technicians to begin the day.

On a more balanced scale, the grid-search results show that a 3:1 ratio, scheduling one less technician to start the day at each location, and not implementing rolling code is the ideal set of parameters. In this scenario, it leads to 241 total transfers for the month that were triggered based on 685 total potential transfers. While 64.8% of potential transfer scenarios were unconverted to actual transfers, this conservative approach yielded 0 re-transfers. This means there were no scenarios where a technician was borrowed from a clinic and needed back within the hour. This is a strong indicator of the robustness of the machine-learning model that is indirectly being evaluated on a more practical metric (number of re-transfers). Essentially, the model is doing a thorough job of preventing the need for re-transfers by accurately anticipating clinical needs before transferring a technician. Lastly, a 91.7% adherence to the ideal ratio indicates strong performance of TecNav with the recommended parameters. As a result, it is feasible to schedule a total of five less technicians per day, and still maintain an appropriate ratio without overwhelming staff. Accounting for gas reimbursements for the transferred technicians, TecNav could yield an estimated savings of approximately $89,000 for each clinic, per year.

Conclusion

It is important to contextualize the preliminary results and recommendations with the caveat that this project was only possible through the fabrication of datasets. Despite conducting extensive secondary

research to emulate real-world data as much as possible, it is difficult to fully capture the varying intricacies and imperfections of actual clinical data. Therefore, this project was designed to deliver a prototype application that can simulate a navigation system in action. Based on client-provided data, this base product can then be customized to fit their specific needs. In order to maximize the project translatability, each phase was strategically designed to utilize components that can be anticipated from client data. However, opting for generalizability did not mean sacrificing utility. In fact, *TecNav* incorporates several layers of sophistication when making navigation decisions such as scheduling preferences, monitoring clinical caseloads for severity level, and considering drive-times. This resulted in a data-driven software application that is customizable in all facets—from the model deployment, all the way down to the actual navigation parameters. In the end, this led to the development of a software application that can monitor the constantly evolving needs at each clinic and transfer technicians in real-time, without any human supervision. Implementing a navigation system such as this can optimize clinical workflow and generate financial savings that could help urgent-care clinics retain their reputation as "low-cost, ER-alternatives". *TecNav* gives a glimpse of the immense potential of leveraging clinical data to deliver a product that can transform the healthcare industry.

References

1. Finnegan, J. (2020, February 26). *Now more than 9,000 urgent-care centers in the U.S., Industry Report says*. Fierce Healthcare. Retrieved May 2022, from https://www.fiercehealthcare.com/practices/now-more-than-9-000-urgent-care-centers-u-s-industry-report-says

2. Google. (n.d.). Google maps. Retrieved May 2022, from https://www.google.com/maps

3. *Medical technician job duties*. All Allied Health Schools. (2022, March 9). Retrieved May 2022, from https://www.allalliedhealthschools.com/medical-technician/medical-technician-careers/

4. JUCM Chart Audit Research. (2019, June 25). *Urgent-care is an appropriate setting for any age-but what ages are showing up the most?* Journal of Urgent-care Medicine. Retrieved May 2022, from https://www.jucm.com/urgent-care-is-an-appropriate-setting-for-any-age-but-what-ages-are-showing-up-the-most/

5. Scripps (2020, April 9). *Should you go to the emergency room or Urgent Care*. Scripps Health. Retrieved May 2022, from https://www.scripps.org/news_items/4231-should-you-go-to-the-emergency-room-or-urgent-care

6. Lusardi, A. (2021, February 8). *Machine learning can now predict how many customers to expect in your restaurant tonight.* LinkedIn. Retrieved May 2022, from https://www.linkedin.com/pulse/machine-learning-can-now-predict-how-many-customers-expect-lusardi

7. Prabhakaran, S. (2022, April 4). *Augmented dickey-fuller (ADF) test - must read guide - ml+*. Machine Learning Plus. Retrieved May 2022, from https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

8. *Sarimax and Arima: Frequently asked questions (FAQ)¶*. statsmodels. (n.d.). Retrieved May 2022, from https://www.statsmodels.org/devel/examples/notebooks/generated/statespace_sarimax_faq.html

9. Iamleonie. (2022, March 15). *Time-series: Interpreting ACF and PACF*. Kaggle. Retrieved May 2022, from https://www.kaggle.com/code/iamleonie/time-series-interpreting-acf-and-pacf/notebook

10. Fuqua School of Business. (n.d.). *Introduction to ARIMA: nonseasonal models*. Introduction to arima models. Retrieved May 2022, from https://people.duke.edu/~rnau/411arim.htm

11. *Time-series - Walk Forward Validation*. Tutorials Point. (n.d.). Retrieved May 2022, from https://www.tutorialspoint.com/time_series/time_series_walk_forward_validation.htm

12. Findlay, S. (2018, May 4). *When you should go to an urgent-care or walk-in Health Clinic*. Consumer Reports. Retrieved May 2022, from https://www.consumerreports.org/health-clinics/urgent-care-or-walk-in-health-clinic/

13. Life Point Medical, LLC . (n.d.). *Top 10 reasons to seek an urgent-care visit*. Top 10 Reasons to Seek an Urgent-care Visit: Timothy Scott Beck, MD: Primary Care Physician. Retrieved May 2022, from https://www.drtsbeck.com/blog/top-10-reasons-to-seek-an-urgent-care-visit

14. Admin, I. and O. (2022, April 6). *Top 9 reasons to go to urgent-care - common urgent-care complaints*. In and Out Express Care. Retrieved May 2022, from https://inandoutexpresscare.com/top-9-reasons-for-an-urgent-care-visit/

15. CDC/NCHS, National Hospital Ambulatory Medical Care Survey. (2012). *Data table for Figure 23. Patient's primary reason for emergency department visit, by age and reason: United States, average*

annual, 2009–2010. CDC. Retrieved May 2022, from
https://www.cdc.gov/nchs/data/hus/2012/fig23.pdf

16. Weiss, A. J., & Jiang, J. H. (2021, December). *Most Frequent Reasons for Emergency Department Visits, 2018*. HCUP - Healthcare Cost And Utilization Project. Retrieved May 20, 2022, from https://www.hcup-us.ahrq.gov/reports/statbriefs/sb286-ED-Frequent-Conditions-2018.jsp

17. *Top 20 most common ER diagnoses*. Definitive Healthcare. (n.d.). Retrieved May 2022, from https://www.definitivehc.com/blog/top-20-most-common-er-diagnoses

18. ASPE. (2021, March 2). *Trends in the utilization of emergency department services, 2009-2018*. ASPE - Office of Health Policy. Retrieved May 2022, from https://aspe.hhs.gov/sites/default/files/private/pdf/265086/ED-report-to-Congress.pdf

19. Ayers, A. A. (2020, June 29). *What are the reporting obligations of Urgent-care Centers for covid-19 patients?* JUCM - Journal of Urgent-care Medicine. Retrieved May 2022, from https://www.jucm.com/what-are-the-reporting-obligations-of-urgent-care-centers-for-covid-19-patients/

20. Higuera, V. (2022, March 28). *6 things to know about urgent-care before you go*. Healthline. Retrieved May 2022, from https://www.healthline.com/health/right-care-right-time/know-before-you-go

21. The American Academy of Urgent-care Medicine. (n.d.). *Frequently Asked Question*. AAUCM. Retrieved May 20, 2022, from https://aaucm.org/faq/

22. 2U. (n.d.). *What is Arima Modeling?* Master's in Data Science. Retrieved May 2022, from https://www.mastersindatascience.org/learning/what-is-arima-modeling/

23. University of Illinois Chicago. (n.d.). *Chapter 23: Using ARIMA for Time-series Analysis*. A Language, not a Letter: Learning Statistics in R. Retrieved May 2022, from https://ademos.people.uic.edu/Chapter23.html

24. *Statsmodels.tsa.arima.model.ARIMA*. statsmodels. (n.d.). Retrieved May 2022, from https://www.statsmodels.org/devel/generated/statsmodels.tsa.arima.model.ARIMA.html