Forecasting of Staffing Needs in Health Care

Marcelle Chiriboga, Weifeng (Davy) Guo, Patrick Tung, Luo (Iris) Yang

June, 2019

Executive Summary

For most positions in the health care business, any staff absences must always be filled in by another staff and the costs of substituting absences with short notice are usually significantly higher than regular staffing. Hence, preparing for potential shortages by predicting the short-term staffing needs can significantly improve the operational efficiency of health care institutions.

The purpose of the project was to help the People Analytics and Innovation Team from Providence Health Care (PHC) to predict the short-term staff needs in order to prepare for unexpected potential costs and staff shortages. The predictions are made based on the historical records of scheduled exceptions, i.e. staff absences due to unexpected or previously arranged reasons such as sick time, vacation, maternity leave, etc.

Introduction

An increase in patients' waiting time at hospitals or the postponement of important procedures, such as surgeries, are known to be critical, which is why medical institutions try to make sure that their clinical positions have backups whenever possible. On the other hand, more than 70% of the operational costs in health care are tied to staffing and, overstaffing can result in a significant increase in these costs.

PHC is a government agency that operates more than 16 healthcare facilities in British Columbia, with almost 7,000 staff, including 1,000 medical staff. At their scale, under or over staffing can have significant impacts both in terms of cost to the organization and in quality of care provided to patients, and for this reason, accurately forecasting staffing needs can have a very positive impact.

In this project, we partnered with PHC to predict staff needs based in their historical exception records, focusing our predictions on the *operational level*, i.e. short term needs, specifically on a time horizon of less than a month. The goal was to answer the question: *"How should PHC prepare for their weekly staffing needs in order to effectively operate with a full staff?"*, giving them more time to handle the exceptions. More specifically we focused on building models for:

- Forecasting staffing needs on a weekly basis, allowing PHC to estimate how many back up staff are needed per site, subprogram, and job family;
- Forecasting how many exceptions will fall under the urgent groups (i.e. overtime and relief not found), so PHC can be better prepared to find relief for critical exceptions;
- Classifying each exception logged on PHC's internal system in one of three possible categories, allowing PHC to prioritize which exceptions to pay extra attention to in finding relief for.

Data Science Methods

Exception Count Predictions

To begin with the predictions of exception count, because we had to make sure that we did not overfit our model when training, we first split our data into three separate portions: training, validation, and testing. The training dataset consisted of data from 2013 to 2016. 2017 and 2018 were the datasets for validation and testing respectively. As all analysis should start, we performed some exploratory data analysis. First, we plotted out the number of exceptions.

As you can see from the above EDA, we can see that there seems to be some sort of pattern to the number of exceptions throughout the years, so we explore it as a time series data. As you can see, we have decomposed the data into the trend and seasonality portions.

Given the temporal nature of the data, we attempted to model this problem using regression, time series analysis, and even neural networks . We then tried to fit different time series models using techniques and tools such as seasonal decomposition and Facebook's open source tool called Prophet. Ultimately, we chose to move forward with Facebook's Prophet tool. Not only did the model provide the best results when comparing it with our validation set, it was also one of the easier models to implement on a large scale.

Because our goal is to provide PHC with more actionable predictions, providing a forecast of an aggregate exception count for the whole of PHC is not effective. For this reason we decided to group our models into specific combinations of SITE and JOB_FAMILY. However, several of these combinations have very little exceptions for our model to capture any meaning, Therefore, we chose to focus only on the largest six health care facilities, which are: St Paul's Hospital, Mt St Joseph, Holy Family, SVH Langara, Brock Fahrni, and Youville Residence. In regards to JOB_FAMILY, we chose to focus only on nurses, but specifically the top 3 nurses: DC1000, DC2A00, DC2B00.

To tune our models, we took a look at the Mean Absolute Error. The MAE provides a clear image for us to see how many exceptions we have predicted incorrectly averaged on a weekly basis. Overall, our MAE for our validation set and testing set were 118.42 and 131.57 respectively. Given that there are thousands of exceptions occurring each week, this MAE is fairly small and the predictions of the model can facilitate management to make better decisions in regards to staffing. In terms of the errors for each facility, for the year 2018, we had the following MAEs:

SITE	MAEs
St Paul's Hospital	120.48
Mt St Joseph	40.69
Holy Family	17.54
Brock Fahrni	8.98
Youville Residence	6.23
SVH Langara	12.42

The graph below is a visualization of the Actual Exception Count vs Predicted Exception Count as a whole for Providence Health Care:

Urgent Exception Predictions

Besides predicting the total count of exceptions, we also focused on the number of exceptions that are urgent. According to the strategies of backfilling the exceptions in PHC, "Overtime" is the last solution to consider due to the high cost it brings. If no backfill is available, the exception will be marked as "Relief Not Found". These two types of exceptions is considered as "urgent". We want to forecast approximately how many urgent exceptions PHC will have, giving HR an insight so that they can make any arrangements beforehand to minimize costs.

We used linear regression instead of time series for this part due to higher accuracy. We removed data from 2014 from the training set because the pattern is very different compared to the years after. Similarly, we also consider only the top three nurse groups, which are `DC1000`, `DC2A00` and `DC2B00` in `JOB_FAMILY` in the original data. As PHC mentioned that they are working on the system to switch shifts among different sites, we did not split by sites like in the previous portion.

The predictors of the linear regression model are shift dates and productive hours. Shift dates are transformed into one-hot encoding, considering day of week, day of month, week of year and month of year. For the

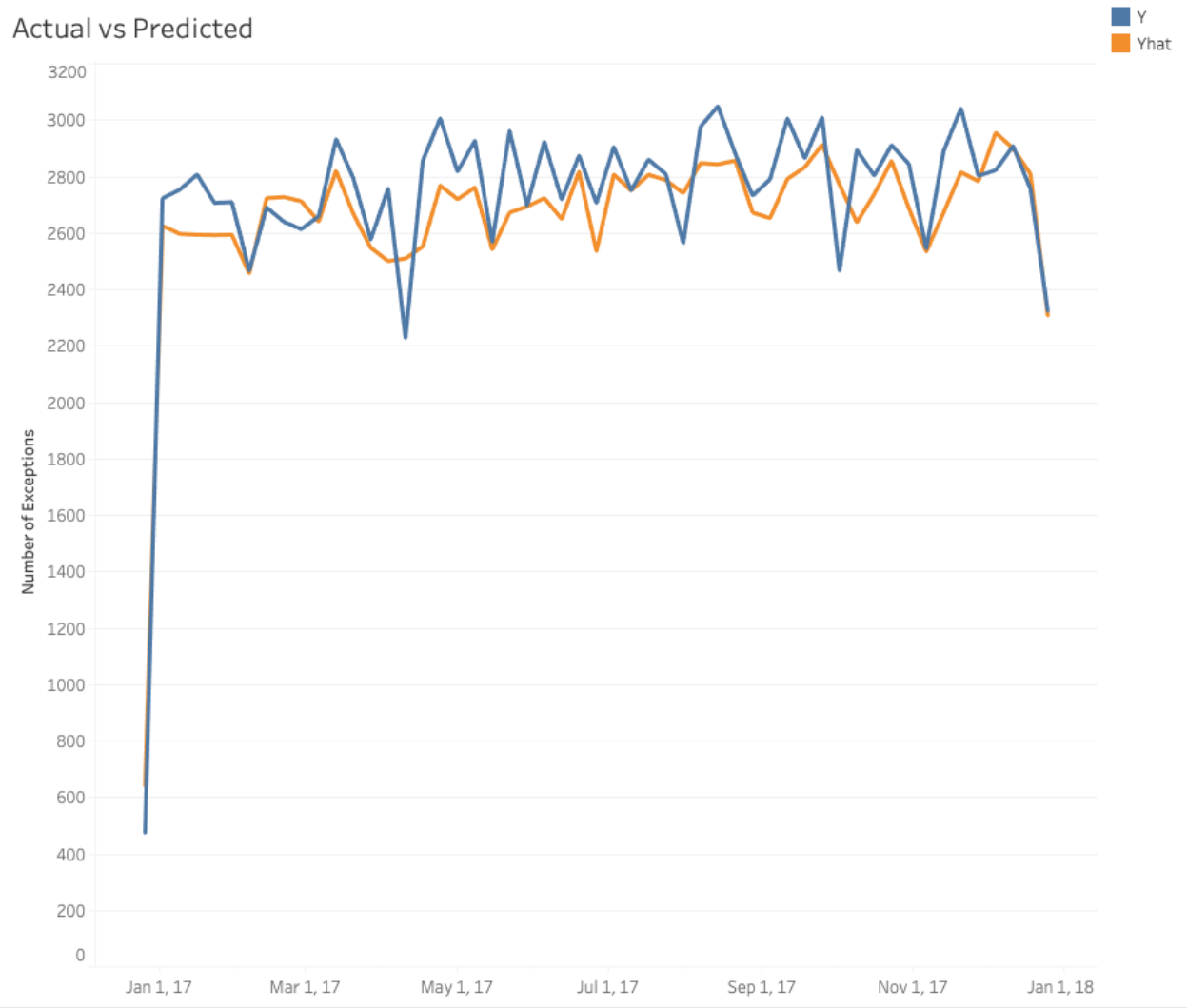
Actual vs Predicted
Y
Yhat
Number of Exceptions
3200
3000
2800
2600
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Jan 1, 17
Mar 1, 17
May 1, 17
Jul 1, 17
Sep 1, 17
Nov 1, 17
Jan 1, 18

Figure 1:

productive hours, even though we do not have the exact data for the future periods, we can provide estimations according to shift arrangements. A graph of testing result for this model is shown below.

For some peaks in the daily basis, the linear model is not able to predict them accurately due to random events. The problem becomes more obvious in the groups with smaller counts like `DC2B00`. We expect that the model can be improved by providing more features, perhaps even new variables related to the operation for each hospital. But in general, the model has captured the general pattern sufficiently and made good predictions for the majority of the days. The Mean Absolute Error for each group of 2018 is listed below.

JOB_FAMILY	MAEs
DC1000	18.35
DC2A00	3.79
DC2B00	2.19

Exception Classification

The classification model uses random forest classifiers to predict the possible outcome for an exception. We are aiming to generate insights for exceptions which have already been created but yet to be fulfilled, so the HR may change their priority to handle some exceptions and to avoid unnecessary cost. After applying logistic regression, random forests, and gradient boosting, we discovered that random forests had the best performance overall. Not only did it provide the best performance, it also fairs well against the other two methods in terms of interpretability.

The label for our classification model is `EARNING_CATEGORY`, but it has 12 values and is too detailed for our prediction, hence affecting the accuracy of the model. As per our partner's advice, as long as the relief type (e.g. straight time) is the same, we can treat them as the same. Therefore, we grouped the 12 labels into the following 3 labels:

- Straight Time: which contains all sorts of straight time reliefs, the pay rate is the same as the normal rate which is positive
- Overtime and Beyond: which contains `Relief Not Found` and all kinds of relief which needs to be paid more than normal rate, which is negative to the company.
- Relief Not Needed, which is neutral to the company.

For feature selection, we applied the forward selection method. We used `EXCEPTION_HOURS`, `EXCEPTION_CREATION_TO_SHIFTSTART_MINUTES`, `NOTCIE`(i.e. staff response time) to setup accuracy baseline, then we added other features to see if it could increase the model accuracy. After several rounds of selection, we ended up adding the following features to our model: `SITE`, `PROGRAM`, `SUB_PROGRAM`, `EXCEPTION_GROUP`, `MONTH`, `DEPARTMENT`, `SHIFT`.

The following table shows our validation accuracies.

	Accuracy
Validation	0.841
Straight Time	0.936
Overtime and Beyond	0.638
Relief Not Needed	0.308

As you can see, the overall accuracy is pretty accurate. However, if we break it down to every category, the difference is obvious. As we have mentioned Overtime costs more than Straight time, so it is more harmful for PHC to have Overtimes compared to Straight times. Therefore, we need to improve the accuracy of Overtime. The reason for the large discrepancy between the categories is imbalanced data. We found that

the number of Straight time entries is much higher than the other two. Which makes sense that the model is more likely to predict an exception as straight time instead of the other two. Because we see this problem, we updated the model to train based on a balanced set of data.

	Amount
Straight Time	262,608
Overtime and Beyond	76,863
Relief Not Needed	11,806

Below is the comparison of our model accuracies. We can see that the accuracy of Overtime and Relief Not Needed has increased, but the accuracy of Straight time has decreased. However, as we have mentioned, because Overtime is more critical to PHC, the sacrifice of Straight time's accuracy is acceptable, and our final test accuracy is listed in the right column.

	Original Validation	Adjusted Validation	Adjusted Test
Overall	0.841	0.794	0.800
Straight Time	0.936	0.823	0.830
Overtime and Beyond	0.638	0.735	0.756
Relief Not Needed	0.308	0.625	0.633

Similar to our other two products, this model's output file is also a `.csv` file. However, we are adding two columns to the input data, one is the shift of exceptions, per partner's request. The other is our prediction results for the `EARNING_CATEGORY`.

Difficulties, Limitations, and Potential Improvements

Throughout the whole project, there were 2 main difficulties.

The first one is that we had missing data. Due to technical reasons, some of our data (e.g. `MIN_TO_MAX_MINUTES`) was missing. We had to remove some records in order to maintain the quality of our training data, which affected the model accuracy.

The second difficulty is feature selection. Though the current features performs quite well, if time allows, we will still want to improve and bring it to the next level. We will focus on discovering new features to improve the accuracy. Perhaps we could even create our own features for better performance.

After having heard the presentation from our program, we learned that using a LightBGM model might perform better than random forests, which is impressive. However, due to the time and resource limitations, we were not able to attempt this model.

Data Product and Results

Exception Count Predictions

The Exception Count Prediction Tool is delivered through a script that generates a user interface that is easy to use. The interface can be run on both Windows computers and Mac computers, and it uses the same code to run. The interface has two parts, the top asks for the user to input the training data, which is the raw `exception_hours.csv` file that PHC has provided us. After including the correct type of data, the user will need to input a prediction timeframe. After clicking "Submit", our time series models will run and generate a `.csv` file that would provide all the relevant predictions regarding the number of exceptions Providence Health Care would have within the prediction timeframe.

Urgent Exception Predictions

The Urgent Exception Prediction Tool is also delivered through a script that generates a user interface similar with the Exception Count Predictions. It requires a file of exception hours and a file of productive hours for past years as training data, as well as a file of productive hours for the period to predict, which can be an estimation. The output is a `.csv` file with prediction results in the given time period.

Dashboard

We implemented the dashboard using Tableau, where we consolidated the three models. It has three different tabs:

- Predictions

This tab displays two charts stacked vertically.

The top chart shows how many exceptions are being predicted in a weekly basis by different sites, job families, and sub-program. The orange series corresponds to the predicted numbers of exceptions, while the grey ones represent the 95% confidence interval.

The bottom chart displays how many urgent exceptions, which includes overtime and relief not found, are being predicted. The different bar colors correspond to different job families.

- Exceptions Classification

This tab displays a summary table, where the user can easily see how many exceptions of each label is already on PHC's system. The user can filter by date and site.

- Productive vs. Exception Hours

This tab displays a comparison between the productive and exception hours based on historical data. The user can filter by date, site, and job family.

Conclusions and Recommendations

Using the three data products we have developed, we are able to provide Providence Health Care with the insights they need to be able to answer the question that was posed at the start of the project: *"How should PHC prepare for their weekly staffing needs in order to effectively operate with a full staff?"*. Not only the data products provide accurate classifications and forecasts of PHC's exceptions, they can also help plan for the short-term and long-term staffing needs. However, it should be noted that we recommend using each of the tools separately.

In regards to the Exception Count and the Urgent Exception Predictions, because our forecasts will stay relevant for quite a while, it is not necessary for PHC to re-train the models every day or even every week. In fact, we recommend that PHC re-train the models once every month with updated data, which will definitely increase the efficiency for PHC.

As to the Exception Classification, the model was trained on all exceptions logged in PHC's internal systems. The model does not require periodic re-training to stay accurate, but it's important to run the classifier for newly logged exceptions to obtain their respective classifications. The frequency should be determined by PHC based on their needs, but it'll likely need to be more than once a month.

Finally, for the prediction models, we focused only on the top 3 nurse job families (i.e. DC1000, DC2A00, and DC2B00). In order to also be able to make predictions for other groups, it's important to continue to gather data so the models can be expanded by training with these new observations. With the amount of data currently available, including the other groups – as well as taking into account other variables, such as department – has a negative impact in the models overall accuracies. This is the most significant improvement opportunity.