INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES

Term Paper

University of Applied Science - Online

Study-branch: Data Science

Rescue Drivers Deployment

Koen Bothmer

Student Number: 92014567

Ankersweg 39

realized in Holtum (NL)

Advisor: Markus Pak

Code and environment available at https://github.com/KoenBothmer/rescue_drivers_prediction

Delivery date: 03-06-2021

Contents

I List of Figures

1 Introduction

This paper describes and justifies the results and methodology used to deliver the project "Rescue Drivers Deployment" as performed in a case study about Berlin's Red Cross organization [Kreuz, 2019], referred to as the client for the remainder of this paper. This work is part of an examination exercise for the course "Case Study: Model Engineering" as part of my Data Science Master's program at IU International University.

As demanded by the client, this project was performed after the crisp-dm methodology [Wirth and Hipp, 2000] and the paper is structured after it's steps:

- Business Understanding: General description of the client and the case study's requirements
- Data Understanding: Data quality assessment and exploratory data analysis
- Data Preparation: Preparing data for forecasting and creation of a simple prerequisite linear regression model
- Modelling: Creation and analysis of the forecasting models used in the final product
- Evaluation: Evaluating the results of the modelling approach
- Deployment: Reasoning behind the app provided, conclusion and discussion.

Accompanying this paper, there is a Github repository [Bothmer, 2021] which provides the source code for all of the steps requiring code. This repository serves two purposes: To support and justify the methodology as described in this paper but also as a deliverable as it was demanded by the client. The work was documented with care so that another data scientist or engineer would be able to expand on this work.

All coding was done in Python making extensive use of the packages Pandas [pandas development team, 2020], Numpy [Harris et al., 2020], Plotly [Inc., 2015], Pyramid Arima [Smith et al., 17], dtale [Schonfeld, 2019] and Statsmodels [Seabold and Perktold, 2010]

More care than the client has demanded has been spend on deployment as it is an often forgotten and underrated part of the data science process [Davenport and Malone, 2021]. Under the hood of a modelling case study like this lies a much harder problem: In order for all of our modelling efforts to be a success the employees at the clients human resource planning department need to adapt to a data driven way of working which, in effect, requires them to change. Because leading organizational change is already a large challenge in itself [Kotter, 2012], it is my opinion that it is vital to provide end users with a workflow that is as effortless as possible. This is why an application that runs on a Docker container [Merkel, 2014] was provided which can be incorporated seamlessly in any common office software workflow.

2 Business Understanding

The German Red Cross is part of the International Red Cross and Red Crescent Movement, known for providing comprehensive aid for more than 150 years. They help people in emergency situations. One of their main areas of work is rescue services and first aid [Kreuz, 2019]. The goal of this case study is to improve the planning of the rescue drivers.

In the current approach, a steady number of drivers are on duty and a steady number of drivers is kept stand-by. The client claims this approach is ineffective as there are days in which none of the stand-by drivers are needed, there are also days in which there are more drivers needed than those kept stand-by. In these cases drivers are dafted; drivers who are not on duty are called and deployed to the task anyway.

The aim of the project is to improve the efficiency. The number of drivers that are stand-by should be activated more often but also, having to daft drivers should be prevented. The prevention of dafting drivers is most important to the client.

In order to solve this problem, the number of expected drivers should be known ahead of time, more specifically: On the 15th day of the month the planning department needs all input for the planning of the upcoming month. To meet this criterion a forecasting system is to be developed based on the provided dataset.

It must be noted that forecasting in these COVID-19 dominated times might be challenging. Human behavior has changed a lot because of all COVID-19 related regulations, this more than likely has a large effect on the deployment of Red Cross drivers. It is important to discuss this issue in an early stage of the project.

3 Data Understanding

3.1 Overall data quality assesment

Overall, the data seems of high quality. Using D-tale [Schonfeld, 2019] the process of exploratory data analysis is dramatically simplified. there are no missing values and extreme outliers were not observed. Figure 3.1 shows a histogram of each variable to get an overall idea of what the data looks like. A short description of the data quality of each column is given below:

- Unnamed contains a row number for each of the 1151 records in the dataset. As is to be expected, each value is unique, there are no missing values.
- As date contains only unique values and the date difference between start date, April 1 2016, and end date, May 27 2019, is exactly 1151 days, it can be concluded that there dates are 1151 consecutive days.
- n_sick contains the number of drivers on duty that are called in sick. It is slightly right skewed causing the mean of 68.8 to be higher than the mode of 57. The largest outlier of 119 sick drivers does not seem infeasible in a population of 1800 employees. Most of the outliers were reported in a consecutive range (19-28 of september 2019), which is to be expected. These outliers should be discussed with the client, for the sake of the case study, the outliers are assumed to be true and will not be removed.
- The 'calls' column shows the number of calls on a given day. It is fairly symmetrically distributed and does not contain extreme outliers.
- n_duty shows the amount of rescue car drivers on duty and contains three levels. This was further researched in paragraph 3.2. Overall the column contains three unique values and has no missing data.
- n_sby contains the number of standby drivers which is always 90.
- sby_need contains the stanby drivers that were needed during each day. The feature is 0 in 73.7% of all cases. This makes it a highly skewed distribution. Is is also remarkable that the number of stand-by drivers needed can exceed the number of stand-by drivers, this will have to be handled carefully as the mistake of counting dafted drivers twice is easily made.
- dafted is 0 in 85.2% of all days. It is a column similar to sby_needed and thus also heavily right skewed.

3.2 Exploratory Data Analysis

Figure 3.2 shows a simple scatter chart of day of year plotted against sick employees which shows the seasonality of drivers being sick. as described in chapter 2 this supports the claims by the business of a seasonal illness patern.

Another plot of number of calls against day of year also shows a seasonal pattern as shown in figure 3.3. The column 'day of year' was added to the data set as it is a convenient feature for visualization. However, for modelling, this feature should be used with care as the numerical property of the model might throw off linear

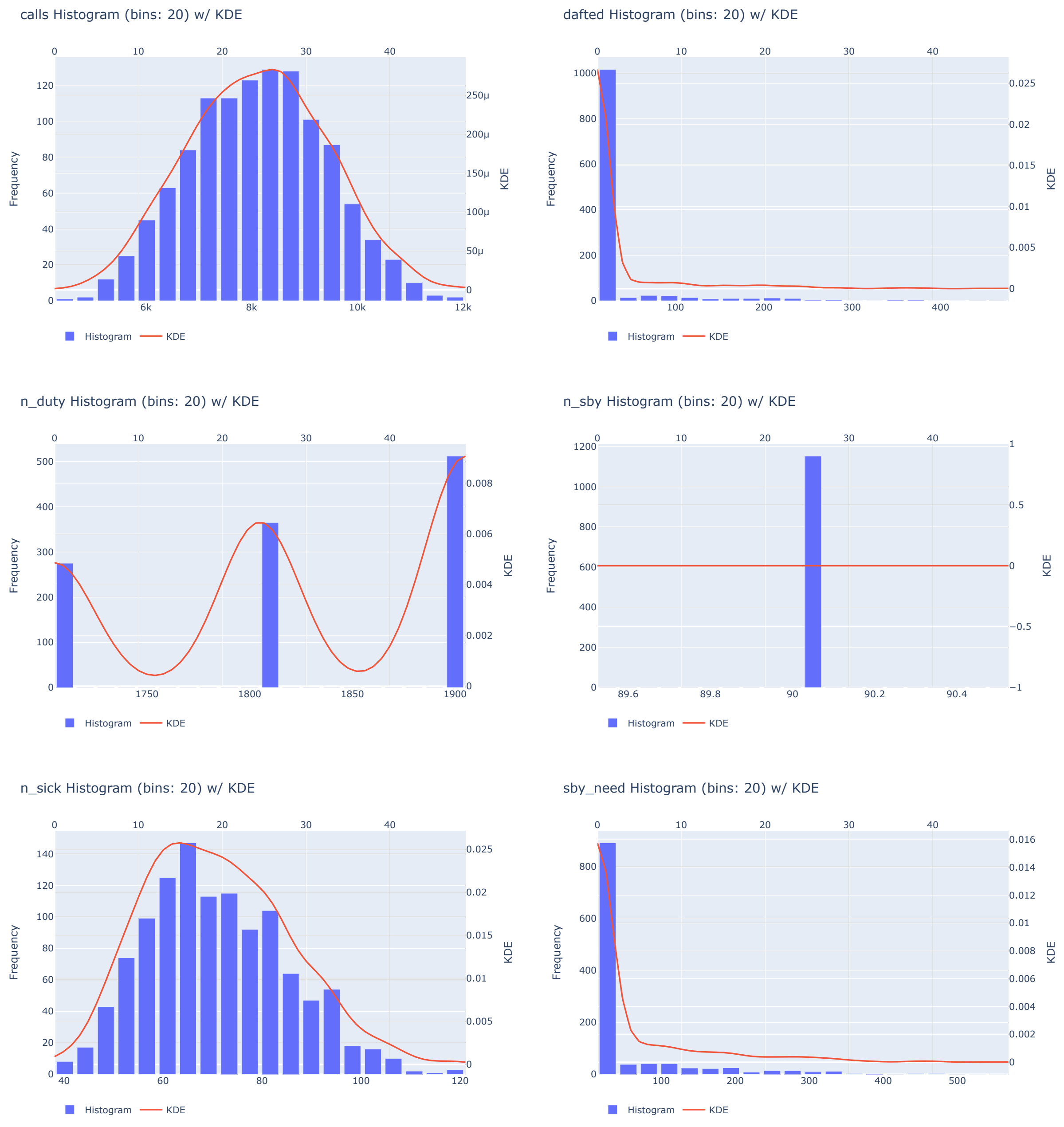


Figure 3.1: A histogram with Kernel Density Estimation of each variable

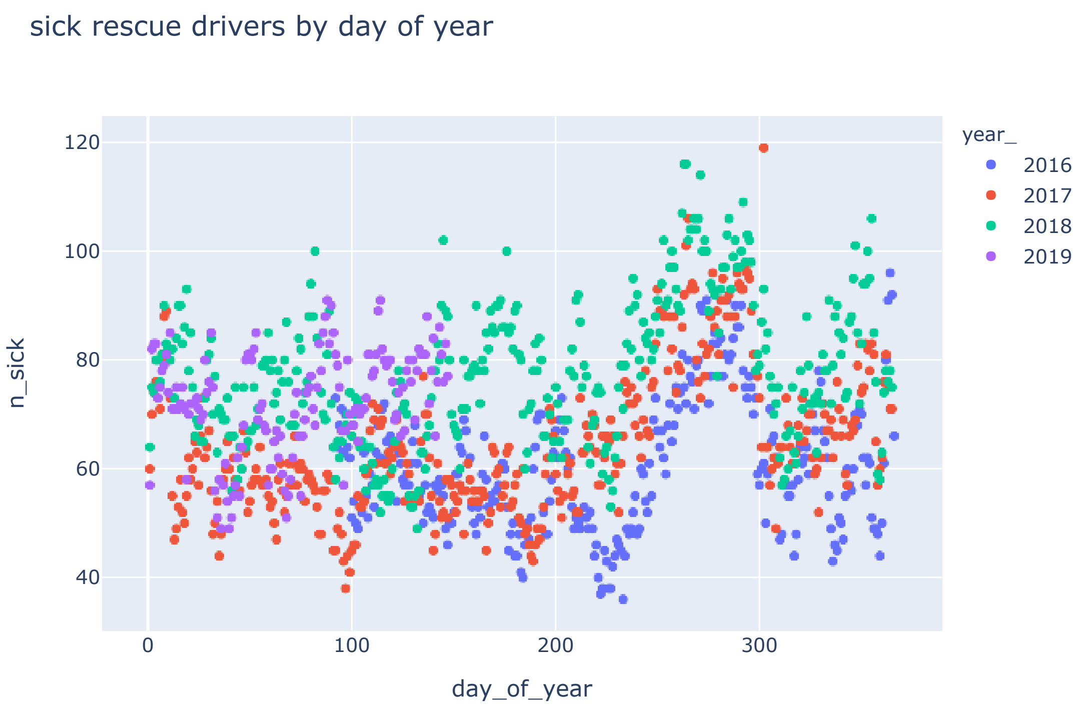


Figure 3.2: A scatter chart of sick employees per day of year

modeling approaches. A better way to include seasonality into the model is most likely needed in the final model.

The number of calls per day does not only seem to show a seasonal trend, there is also an upward trend over the years as shown in figure 3.4.

The number of rescue drivers on duty seems to have increased twice over the recorded period. The amount of drivers on duty could act as a buffer that influences the amount of stand-by drivers needed. Therefore, in a later stage, it is necessary to engineer a feature that takes this into account.

Figure 3.5 shows the number of stand-by units needed and the number of non-stand-by units that were dafted (y) in a scatter plot against the number of calls (x). The pattern that arises seems to be model-able by a rectified linear unit (ReLU) [Schmidt-Hieber et al., 2020]. A ReLU function behaves like a linear function from when a certain threshold has been passed. It is noteworthy that the threshold shows to be highly dependent on the number of drivers on duty. It seems like the total number of drivers needed is heavily correlated with the number of calls received, in a later step in the project a feature should be engineered that contains the total number of drivers deployed on a given day which could serve as the dependent value of a modelling effort.

Figure 3.5 also shows that the number of stand-by drivers that were needed surpasses the 90 drivers that are stand-by. It seems that the drivers that are dafted are also counted as stand-by needed. This should be discussed with the client and investigated further in a later project step.

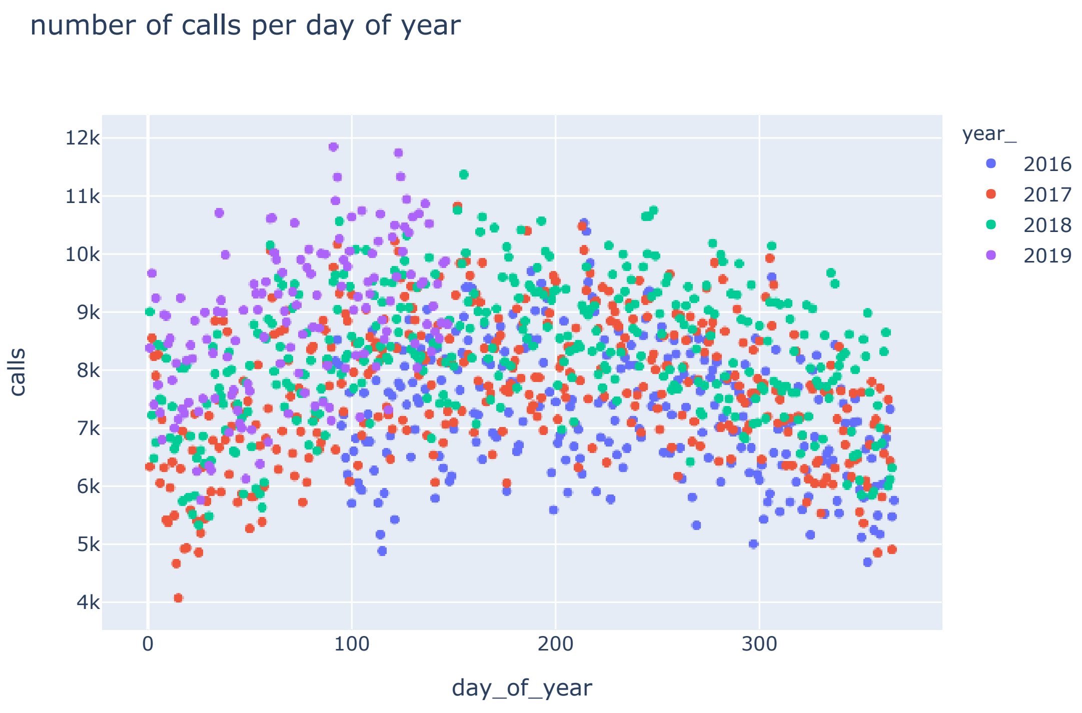


Figure 3.3: A scatter chart of number of calls per day of year

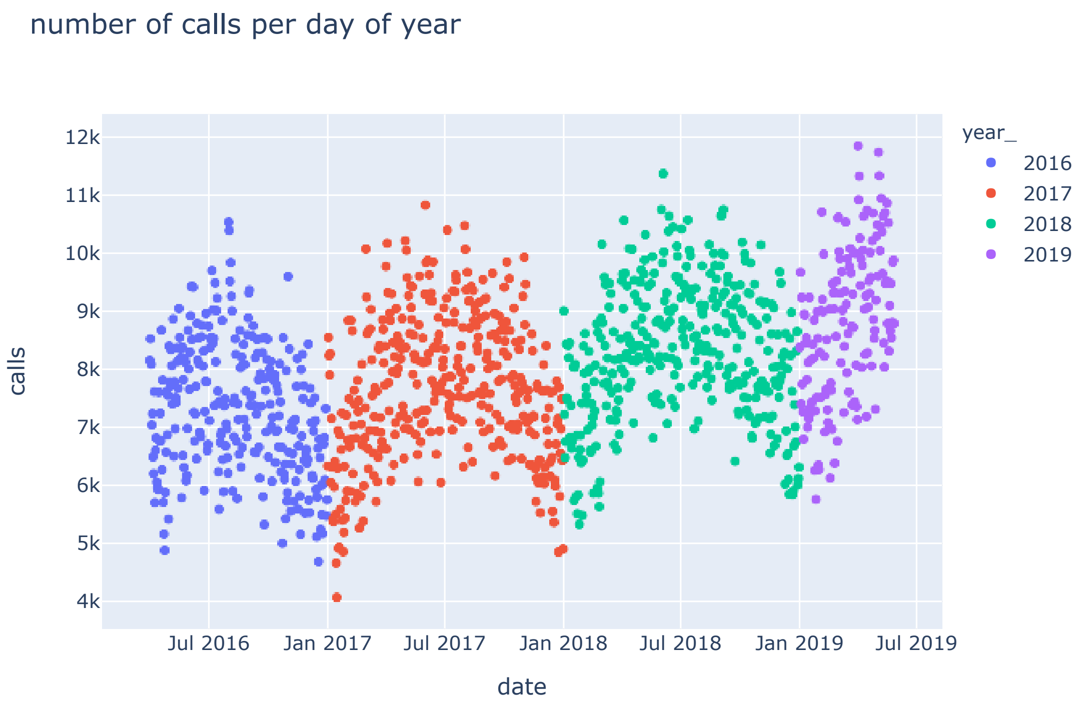
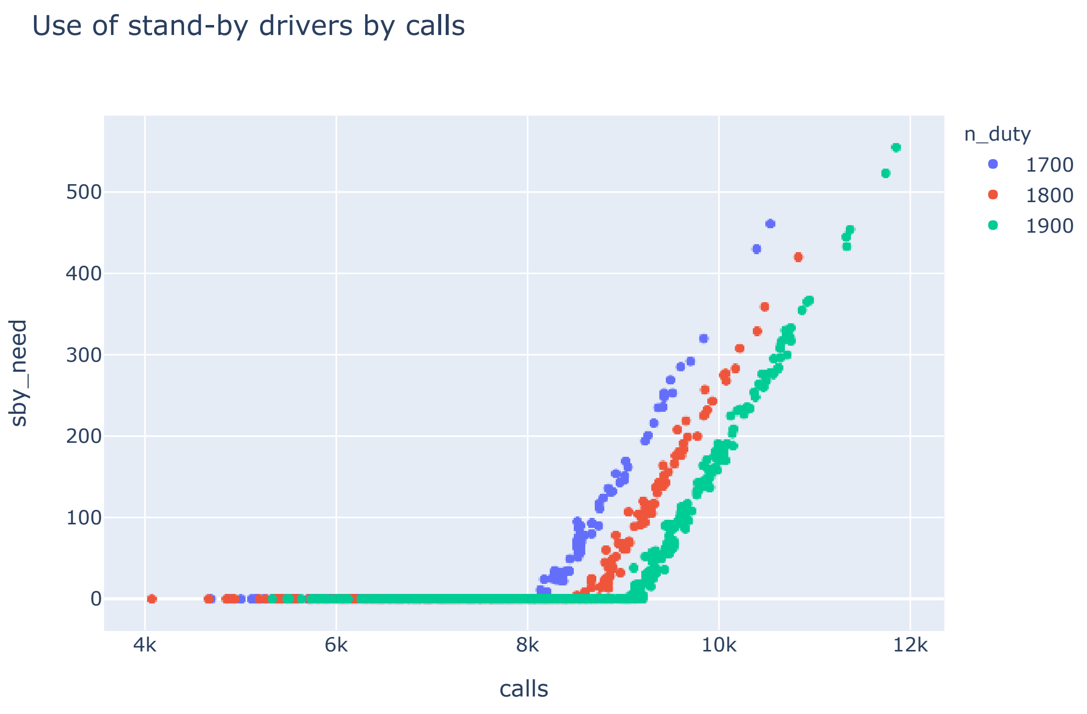


Figure 3.4: A scatter chart number of calls by date



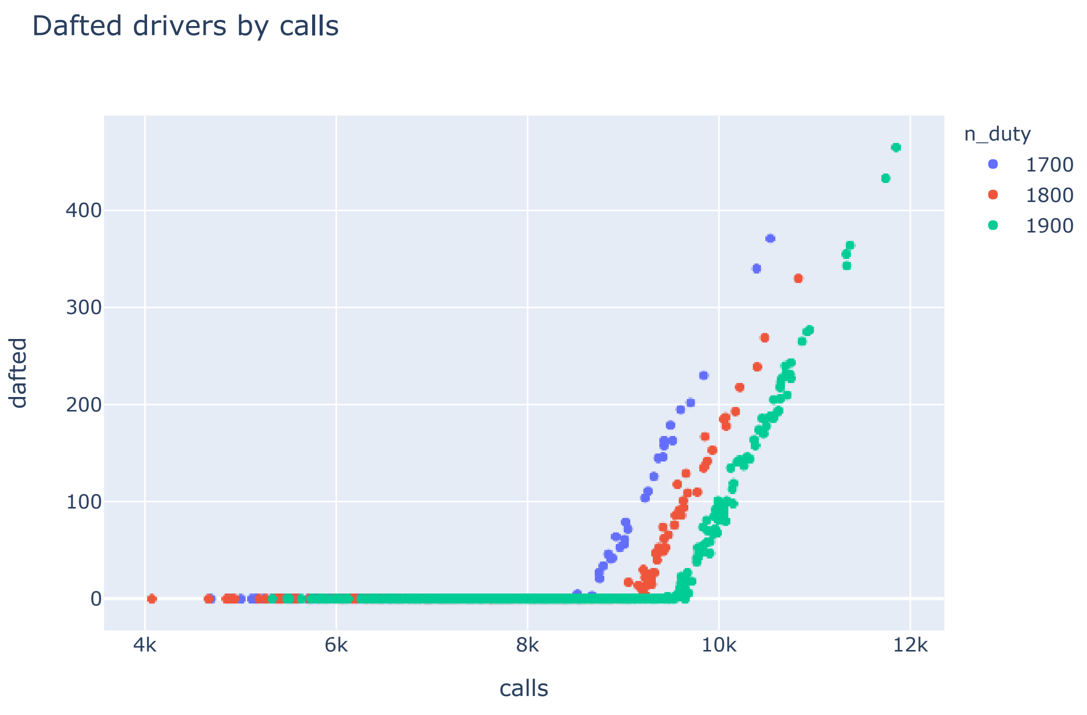


Figure 3.5: Use of stand-by drivers and dafted drivers by number of calls

4 Data Preparation

As described in chapter 3 plotting the use of stand-by drivers against number of calls shows a strong linear dependency after the threshold of the number of drivers on duty has been passed. To expand on this a feature representing the drivers needed was created. As we can only determine how many drivers would have been needed on days that stand-by drivers or dafted drivers were needed, the data was filtered to consider only these rows in the exploration of this feature. The number of drivers were corrected for the number of sick drivers in order to represent the number of drivers actually deployed on a given day. Plotting this against the number of calls shows what visually seems to be a perfect linear relationship as shown in figure 4.1. This strongly suggests the data has been manipulated or that it is synthetic. This will have to be discussed with the client. As the intercept and coefficient can be easily retrieved exactly, fitting a model is somewhat overkill. For the sake of the exercise and also because the real-world data after deployment can be expected to behave more stochastically, a linear model to predict drivers needed was created anyway in the subsequent step of the project.

The model was not extensively evaluated as, rounding aside, it predicts without error on both train and test set. As the relationship is perfectly linear, the model could have been trained on 2 samples.

To extrapolate this relationship to days the number of drivers on duty were sufficient (days without stand-by drivers used), a simple linear regression model was trained. It can be discussed that this step belongs in chapter 5, however, as this step is relatively simple and prepares the dataset for more sophisticated methods the extrapolation of the drivers needed was included in this chapter.

As figure 4.2 shows, the column 'drivers needed not sick' is distorted by the number of drivers on duty. By using a linear regressor, the column 'drivers predicted' was created which extrapolates the relation.

The subsequent modeling step, described in chapter 5, now has a clear purpose: creating a forecasting model for the number of calls and the number of sick drivers. Given the strong relationship between calls and drivers needed, a good estimate of these values will enable the optimization of the planning problem.

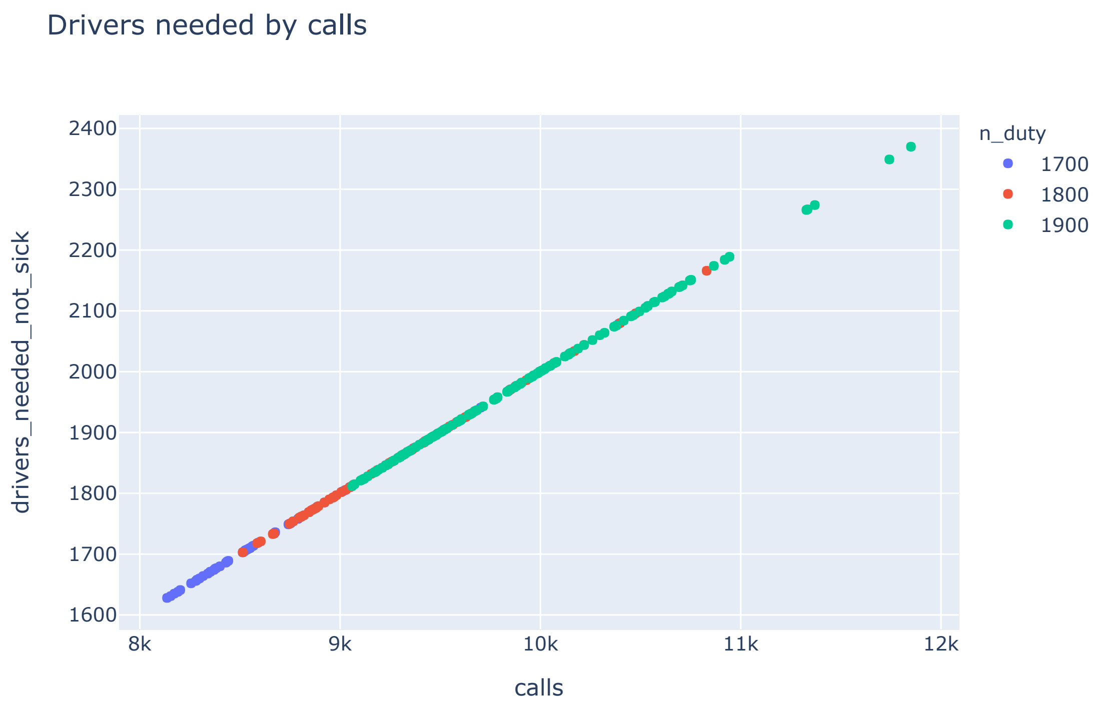


Figure 4.1: A scatter chart of number of drivers needed per number of calls

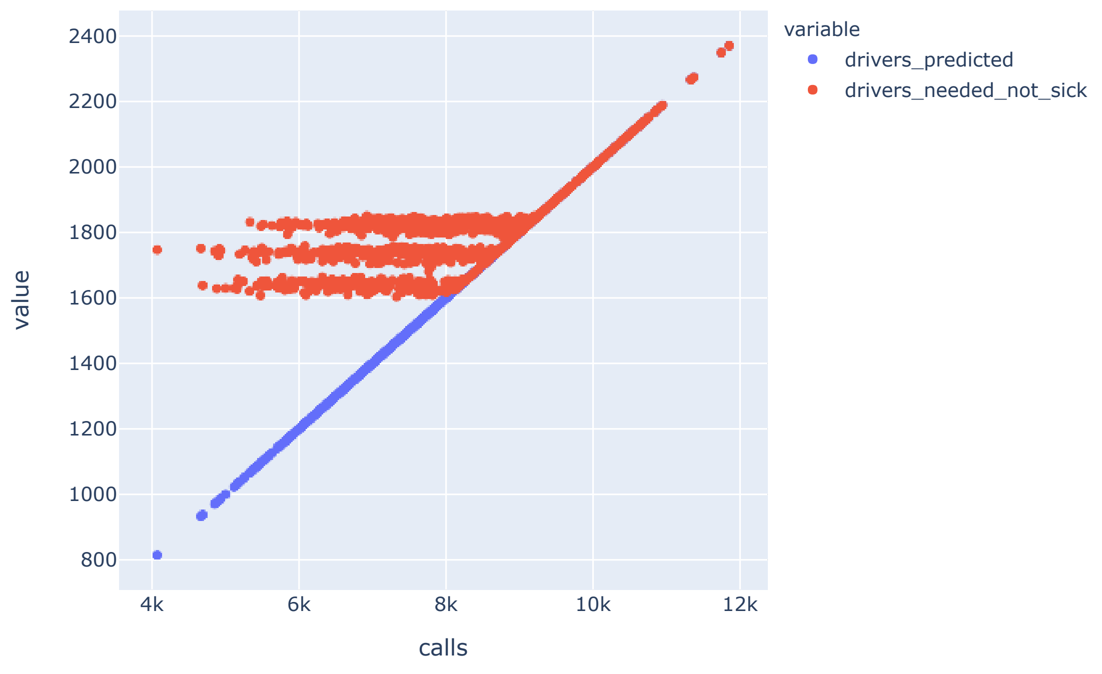


Figure 4.2: A scatter chart of predicted and needed drivers by calls

5 Modelling

Because the client needs to create the planning for each month on the 15th of the prior month, the problem cannot be solved by the classical machine learning approach where the model learns a function of dependent variables to estimate an independent variable. Instead, a forecasting technique is required. Because of the seasonal and linear trends observed in chapter 3 a seasonal ARIMA (SARIMA) is a natural candidate to model this behavior [Valipour, 2015].

As was described in chapter 4, a prediction of the number of calls and the number of sick drivers is needed to provide a solution for the clients planning problem. Both of these features show a high daily variance (figure 3.3, figure 3.2), therefore monthly averages were aggregated that are more feasible to forecast.

It must be noted that the time span of the dataset is limited, there is about 3 years of available data. As the seasonal patterns repeat yearly there are only three cycles to work with. Especially in model selection this causes some difficulties. This is why it was decided to use the full dataset in model selection. This potentially causes some bias, it is recommended to validate the deployed model as described in chapter 6 on the first few months of new data.

The last 6 months of the data were used as a test set, all previous data was used to train the selected models for number of calls and number of sick drivers. As shown in figure 5.1, the model seems to be able to reasonably estimate the data based on the historic trend. However, this approach does not resemble how this model will be used in practice. The business will not want to forecast 6 months ahead in time. To get a better approximation of how the model will perform in practice, the model should be evaluated using using an approach in which the model is retrained monthly, respecting the clients deadline of the 15th day of the previous month. A more thorough explanation of this principle is given in chapter 6.

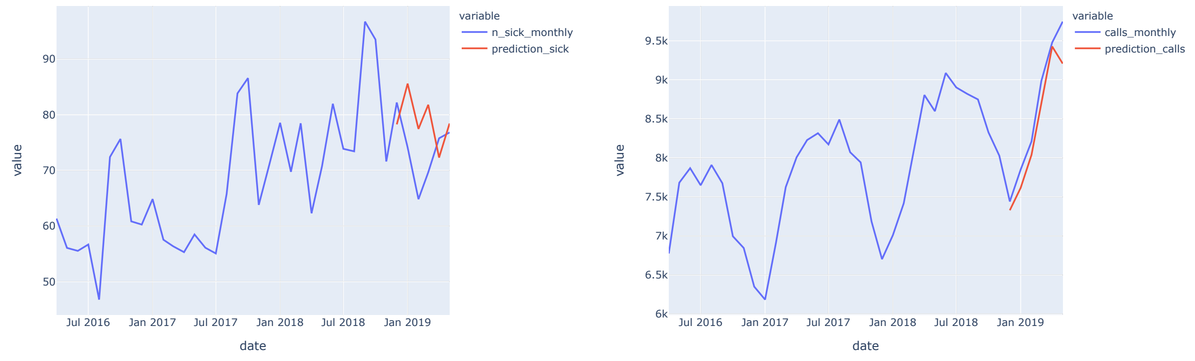


Figure 5.1: Forecasts on the test set

6 Evaluation

6.1 Iterative Evaluation

The model was evaluated iteratively in order to best resemble the way it will be deployed in practice. The last year of the dataset was split up into monthly subsets spanning from the 15th day of the previous month to the 15th day of the current month. In each iteration the subset is merged with all the previous data and used to forecast sick drivers and number of calls. The root mean squared error of the forecast vs the actual value was used as evaluation metric.

In order to understand whether the model adds value versus a naive approach, two dummy 'models' where created predicting the average of the previous month and the average value of previous year. The root mean squared error of these metrics is a benchmark to compare the forecast to.

RMSE for predicted calls is 391.9, RMSE for the previous month dummy model is 418.7, RMSE for the previous year dummy model is 765.4 RMSE for predicted sick is 12.0, RMSE for the previous month dummy model is 10.3, RMSE for the previous year dummy model is 14.6

As it shows, the model-based approach of predicting the number of calls is improved by some margin. The previous month dummy of the sick prediction is actually a bit more accurate then the predicted amount of sick drivers. Because less then 3 years of training data were used, the ARIMA model can be expected to improve over time as more data becomes available. Therefore the ARIMA approach was still used in the final solution.

The drivers needed where predicted using the model that inferences the needed drivers from the number of calls as described in chapter 4. The number of needed drivers was corrected for the expected number of sick drivers. These steps result in a monthly prediction of the amount of drivers that are needed. These were joined to the original daily data so that the final predictions can be evaluated. This process was visually summarized in Figure 6.1.

6.2 Error Analysis

Figure 6.2 shows a histogram of the prediction error and a percentile chart of the error distribution. The percentile table can be used to plan an appropriate amount of drivers. As the client does not want to make use of drivers

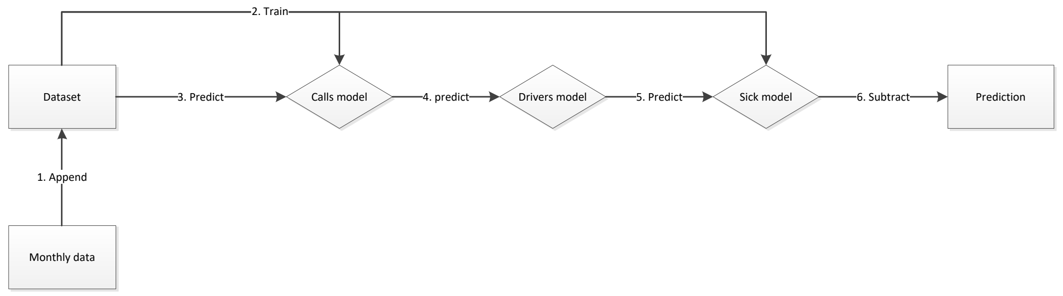


Figure 6.1: A diagram explaining the modelling steps of the deployed product

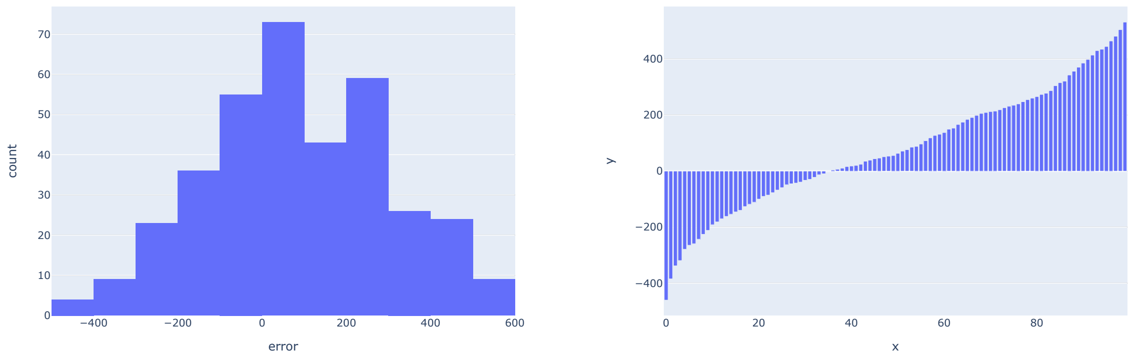


Figure 6.2: The distribution of the prediction error

that are not stand-by, the number of stand-by drivers should be chosen at a high confidence level. The exact numbers will have to be discussed with the client but a first suggestion would be to have enough drivers stand-by in 95% of all cases. This would mean to have the predicted amount of drivers and the error on the 5th percentile (264 drivers, the visual is reactive in the notebook on Github [Bothmer, 2021]) on duty so:

$$Standby = Predicted + 264 - n_{duty}$$

For example, on a month of high expected driver demand, like march 2019 the number of suggested stand-by drivers as predicted on 15-02-2019:

$$Standby = 1957 + 264 - 1900 = 321$$

A second example from a low expected driver month, For example, november 2018, as predicted on 15-10-2018:

$$Standby = 1567 + 264 - 1900 = -69$$

As it shows, there are months in which a very high number of stand-by drivers is suggested, in other months, the model even suggests a negative amount of needed drivers. It is strongly recommended to also include the number of drivers on duty in the dynamic planning of the drivers. For example, have a monthly number of drivers on duty which will suffice in 85% (15th percentile) of all days, which would mean:

$$n_{duty} = Predicted + 145$$

$$Standby = Predicted + (264 - 145)$$

This method would, in the same examples respectively, have 2102 drivers on duty with 145 stand-by on the high month march 2019 and 1712 drivers on duty with 145 drivers stand-by on the low month november 2018.

Off course, the described confidence levels are merely suggestions, the domain expertise of the client regarding contracts and acceptability of dafting risk will have to be consulted to decide on the most balanced set of confidence levels.

7 Deployment

7.1 Application

As the end-users of the suggested methodology are not expected to be of high technical skill it is vital to the success of this undertaking to supply a simple interface to make use of the models. In research, methodology created in Python Jupyter notebooks is notoriously difficult to share [LeVeque, 2013]. These challenges also apply to the deployment of data science projects. The complex environment of open-source packages with all their dependencies and versions is difficult to manage. One way to provide a solution to this problem is containerization using Docker [Merkel, 2014] [Boettiger, 2015] which provides a cross-platform solution with all dependencies included.

For the deployment of this project a simple application that can be run from a Docker container was produced. In the Github repository supporting this paper [Bothmer, 2021] this application is available. Further details can be found in the readme file but in essence, after setting up a Docker image from a Docker file, the application takes the data as provided by the business and outputs the predictions generated by a Python script. The batch file that starts the script assumes the user to be on a Windows computer but can be easily adopted to another environment as the container that it starts will always run a Linux kernel so it is platform independent.

The application is created to be used monthly, the forecasting models included will retrain at each iteration, because of this the accuracy is expected to improve over time.

If the input data can be provided in an automated way the client is strongly encouraged to do so as the batch file can then be scheduled on a task scheduler so that the output will automatically be available to the end users.

7.2 Conclusion and Discussion

The project as described in this paper resulted in an application which can enhance the planning routine of the rescue car drivers for Berlin's Red Cross. The provided solution includes a two step model in which two self learning forecasting algorithms provide input to a linear regression model which is used to predict the number of drivers for the upcoming month. Using the predictions in combination with the provided error percentile chart, the client can determine how much risk to take of having to daft drivers.

The number of calls per day has proven to be the strongest predictor for the number of drivers needed. The provided forecasting model to estimate this metric outperforms all attempts at a naive 'dummy' model by a large margin. The model is further enhanced by also predicting the number of sick drivers. This model does not perform as well as estimating the previous month number of sick drivers. It was still adopted in the final solution because the model is expected to improve over time.

The assessment of the model's performance should also be included in the monthly planning routine. A dashboard visualizing the prediction error of the previous month is suggested to be adopted. The creation of this dashboard is beyond the scope of this paper but is not too complex to create with a small adaption of the Python script in the final application.

Bibliography

[Boettiger, 2015] Boettiger, C. (2015). An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79.

[Bothmer, 2021] Bothmer, K. (2021). Rescue drivers prediction. `https://github.com/KoenBothmer/rescue_drivers_prediction`.

[Davenport and Malone, 2021] Davenport, T. and Malone, K. (2021). Deployment as a critical business data science discipline. *Harvard Data Science Review*.

[Harris et al., 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

[Inc., 2015] Inc., P. T. (2015). Collaborative data science.

[Kotter, 2012] Kotter, J. P. (2012). *Leading change*. Harvard business press.

[Kreuz, 2019] Kreuz, D. R. (2019). A self-portrayal of the red cross.

[LeVeque, 2013] LeVeque, R. J. (2013). Top ten reasons to not share your code (and why you should anyway). *Siam News*, 46(3).

[Merkel, 2014] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2.

[pandas development team, 2020] pandas development team, T. (2020). pandas-dev/pandas: Pandas.

[Schmidt-Hieber et al., 2020] Schmidt-Hieber, J. et al. (2020). Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875–1897.

[Schonfeld, 2019] Schonfeld, A. (2019). D-tale is the combination of a flask back-end and a react front-end to bring you an easy way to view analyze pandas data structures.

[Seabold and Perktold, 2010] Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

[Smith et al., 17] Smith, T. G. et al. (2017–). pmdarima: Arima estimators for Python. [Online; accessed 3 July 2021].

[Valipour, 2015] Valipour, M. (2015). Long-term runoff study using sarima and arima models in the united states. *Meteorological Applications*, 22(3):592–598.

[Wirth and Hipp, 2000] Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1. Springer-Verlag London, UK.