# STAFF MANAGEMENT TOOL

Exploring an approach towards optimising number of employees while maximizing customer satisfaction and minimizing cost to organization.


**TEAM THINKQUE**

**ABHYUDAY PATIL | ANSH BHATT**

# PROBLEM STATEMENT

To devise a model which, given the historical volume of Calls and Chats received by the remote IT service desk, predicts the number of employees required to handle the incoming service requests on a given day in the future on an hourly basis.

## WHY A PREDICTIVE MODEL?

### INCOMING SERVICE REQUESTS

The incoming calls or chats are randomly distributed across each hour. Hence, we can not directly use the call hours to determine the optimal number of employees.

### CUSTOMER SATISFACTION

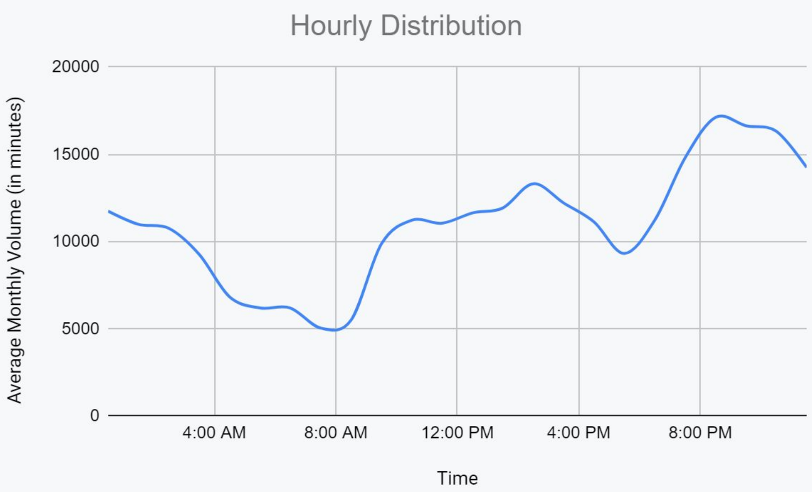
Customer Satisfaction is largely impacted by the wait time in the queue and by the frequency of call drops, which our model aims to minimize.

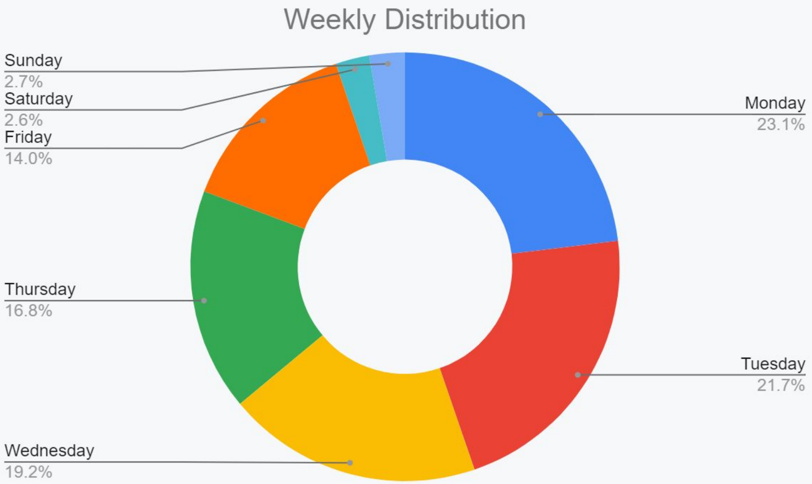
### OPTIMISING RESOURCES

Increasing resources will steadily reduce the waiting time of customers. By fixing a target service level and properly planned shifts, we can optimise the resources required.

# OUR CUSTOMER ANALYTICS

Visualizing the minutes spent on addressing the incoming service requests on an hourly and weekly basis.




With the hourly distribution, we notice that the service requests peak twice during the day. Once close to 3:00pm and the other close to 8:00 pm. This can give us valuable insight regarding shift rotations.

With the Weekly Distribution, we notice that the service requests peak on Monday and almost steadily decrease through the rest of the week, with weekends seeing very little traffic.

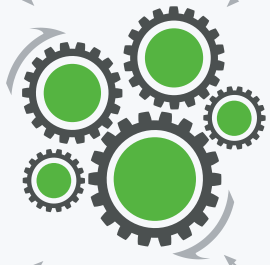
# INPUT PARAMETERS

### 01 HISTORICAL VOLUME

Two months of Historical Service Volume which includes both Calls and Chats.

### 02 AVERAGE HANDLING TIME

The Average Time that an employee must spend on a Service Request, as obtained from the historical data.



### 03 TARGET ANSWER TIME

The Average Time in which the service desk strives to Connect the Customer to one of the Employees.

### 04 WORKING HOURS

The amount of time that an Employee is expected to be available per day.

### 05 MAXIMUM OCCUPANCY

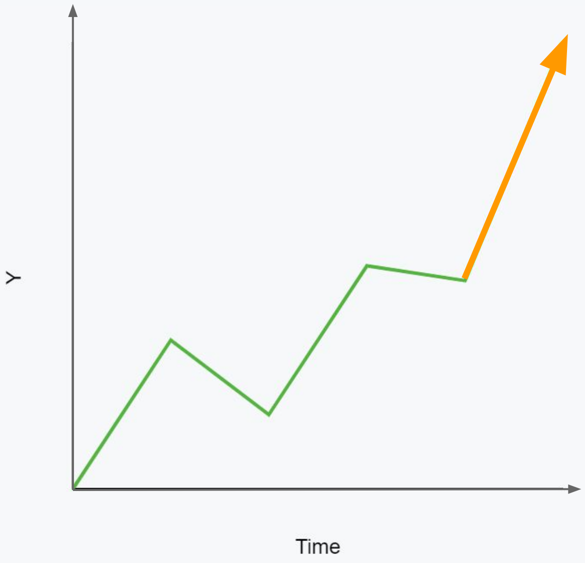
The Maximum Amount of time agents spend on the phone over the course of an hour, expressed as a percentage.

### 06 SHRINKAGE

The Amount of Paid Time that an employee is not available at work, despite being scheduled.

# TWO PHASE PREDICTIVE MODEL

We have devised a two phase model which takes in the previously discussed input parameters and outputs the optimal number of employees for the organization




01

## TIME SERIES FORECAST

A Time Series Forecast Model will be fit on Historical Call and Chat Volumes and will be used to predict the Service Volumes for the Future. This Model was preferred over Point Estimation and Averaging Models since it is able to capture seasonalities. This will be done by Preprocessing the raw data, estimating and eliminating Seasonality by Differencing, using seasonal ARIMA, Regression with ARIMA errors, Facebook's Prophet and other Forecasting models. RMSE, MAE and MAPE will be used to evaluate the Forecast.

## QUEUEING SIMULATION

In this model, we assume that the same set of employees cater to both the incoming calls and the incoming chats. This ensures optimum utilisation of resources and even distribution of workload among all employees.

What separates this from regular queueing models is that there are two different queues. Requests in different queues need to be treated differently because the Average Handling Times for both types of requests are significantly different. To achieve this, our model simulates the queues as well as the request handling times. Thus, we can get a close to real life estimate of the waiting time for the customers.

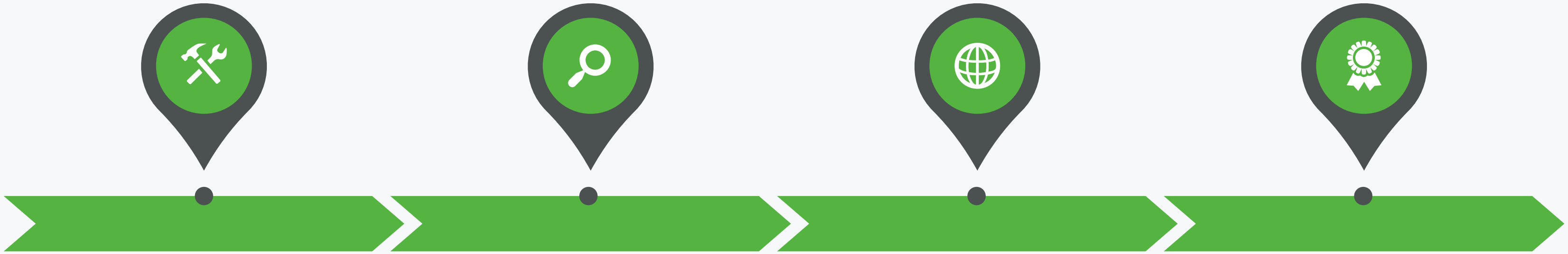
Since we are using a simulation, we can also check for the optimum utilisation of the resources. If a server lies idle during a particular hour, we can remove it.

02

# PHASE I - TIME SERIES FORECAST

# MACHINE LEARNING PIPELINE

We look to develop a manual ML Pipeline since our goal is to produce a model to solve our staffing problem



PREPROCESSING

MODEL SELECTION AND TRAINING

MODEL EVALUATION AND VALIDATION

DEPLOYMENT AND MONITORING

# PRE-PROCESSING DATA

Step 1 of the Pipeline

- The Preprocessing will include incorporating both the date and the time duration of the service time using the datetime library of Python, concatenating the Call and Chat Request data in a single dataframe and indexing it with the datetime feature. The final dataframe can be visualised as follows:

| Date-Time | Call Requests | Chat Requests |
|---|---|---|
| 2/1/2021 6:30:00 | 3 | 12 |
| 2/1/2021 7:30:00 | 5 | 4 |
| 2/1/2021 8:30:00 | 9 | 15 |

- We observe that the data contains double seasonality with an intraday pattern for hourly calls within a day and an inter-day pattern within a week. Therefore, the data has two seasonal cycles: a daily cycle of length 24 and a weekly cycle of length 168 (24 hours x 7 days).
- We define Calendar-related dummy variables for time of day (24 hours) and day of week (7 days) to use for the regression part of the models. This is similar to the One-Hot Encoding method which is widely used while working with pure-regression based models.
- We also consider the natural-logged series of the Call and Chat Requests, which is necessary to stabilize variance, satisfy model assumptions and produce better forecasts.

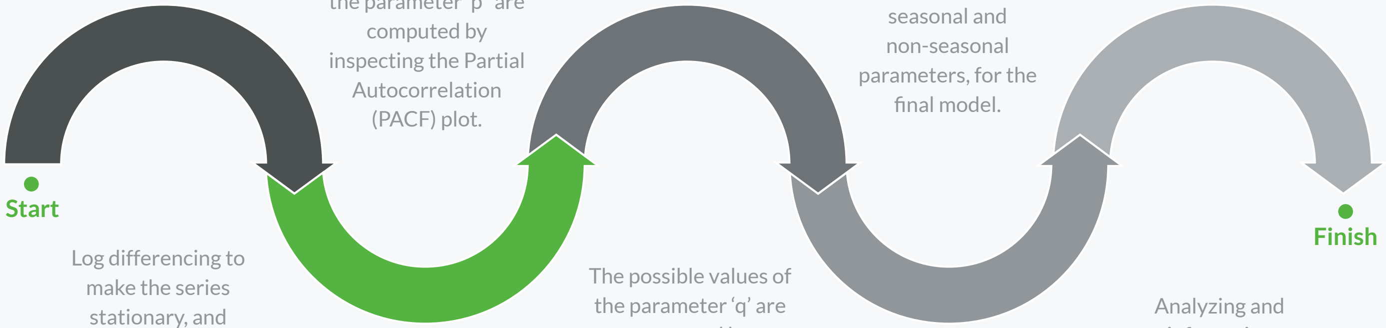
# MODEL SELECTION

| DATA | METHOD | FORECASTING MODEL |
| --- | --- | --- |
| ORIGINAL DATA | SEASONAL ARIMA | $\left(1-\varnothing_1 B-\varnothing_2 B^2-\varnothing_3 B^3\right)\left(1-\varnothing_{24} B^{24}\right)\begin{pmatrix}1-\varnothing_{168} B^{168}-\varnothing_{336} B^{336}\\ -\varnothing_{504} B^{504}-\varnothing_{672} B^{672}\end{pmatrix}\left(1-B^{24}\right) y_t=e_t$ |
| | REGRESSION WITH SEASONAL ARIMA ERRORS | $y_t=c+\sum_{i=1}^{23} \beta_i H_i+\sum_{j=1}^{6} \beta_j D_j+n_t$ <br> $\left(1-\varnothing_1 B-\varnothing_2 B^2-\varnothing_3 B^3\right)\left(1-\varnothing_{168} B^{168}\right) n_t=e_t$ |
| LOGGED DATA | SEASONAL ARIMA | $\left(1-\varnothing_1 B-\varnothing_2 B^2-\varnothing_3 B^3\right)\left(1-\varnothing_{24} B^{24}\right)\begin{pmatrix}1-\varnothing_{168} B^{168}\\ -\varnothing_{336} B^{336}\end{pmatrix}\left(1-\varnothing_{672} B^{672}\right)\left(\log\left(y_t\right)-c\right)=e_t$ |
| | REGRESSION WITH SEASONAL ARIMA ERRORS | $\log\left(y_t\right)=c+\sum_{i=1}^{23} \beta_i H_i+\sum_{j=1}^{6} \beta_j D_j+n_t$ <br> $\left(1-\varnothing_1 B-\varnothing_2 B^2-\varnothing_3 B^3\right)\left(1-\varnothing_{168} B^{168}\right) n_t=e_t$ |

- A seasonal ARIMA model includes Autoregressive and Moving Average terms at lag s, where s is the number of seasons, by differencing to remove additive seasonal effects
- Regression with ARIMA errors combines two powerful statistical models namely, Linear Regression, and Seasonal ARIMA , into a single super-powerful regression model for forecasting time series data.
- Regression with ARIMA errors method can incorporate the effects of special days by adding dummy variables to the regression terms of the model. For the seasonal ARIMA method, the solution is to smooth out the special days using an adjustment method or filtering method prior to model fitting.

# MODEL **TRAINING**

The Training Process of Seasonal ARIMA and SARIMAX Models




## 01 SEASONAL DIFFERENCING

Log differencing to make the series stationary, and cross-verifying it by using the ADF Statistic.

## 02 AUTO-REGRESSIVE MODEL

The possible values of the parameter ‘p’ are computed by inspecting the Partial Autocorrelation (PACF) plot.

## 03 MOVING-AVERAGE MODEL

The possible values of the parameter ‘q’ are computed by inspecting the Autocorrelation (ACF) plot.

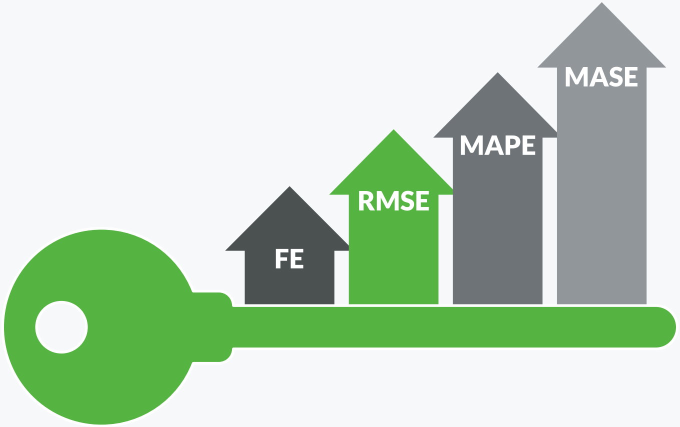
## 04 HYPERPARAMETER TUNING

Using GridSearchCV to choose the best seasonal and non-seasonal parameters, for the final model.

## 05 RESIDUAL ANALYSIS

Analyzing and inferencing auto-correlations from the residuals (the training set error).

# MODEL EVALUATION




## KEY METRICS

01

### FORECAST ERRORS

These are measured as the difference between the observed value and it's forecast. It denotes the unpredictable part of an observation.

02

### SCALE-DEPENDENT ABSOLUTE ERRORS

These include the most commonly used measures namely Mean Absolute Error(MAE) and Root Mean Squared Error(RMSE) . A forecast method that minimises the MAE will lead to forecasts of the median, while minimising the RMSE will lead to forecasts of the mean.

03

### PERCENTAGE ERRORS

Percentage errors have the advantage of being unit-free and hence widely used to compare forecasts. The most commonly used measure is Mean Absolute Percentage Error (MAPE).

04

### SCALED ERRORS

These errors are obtained by scaling the errors based on the training MAE from a simple forecast method. A scaled error is less than one if it arises from a better forecast than the average naïve forecast computed on the training data. The most commonly used measure is Mean Absolute Scaled Error (MASE).

# PHASE I COMPLETE

The model trained as above shall provide us with the Call and Chat volumes for each hour of each day of the future month(s), which will be utilized in Phase II of our Predictive Model.

The model is re-trainable on new datasets which can be created by appending the new data to the old dataset and by monitoring the results,  the hyperparameters of the model can be tuned to accommodate new discernable patterns, if any.

# PHASE II - QUEUEING SIMULATION

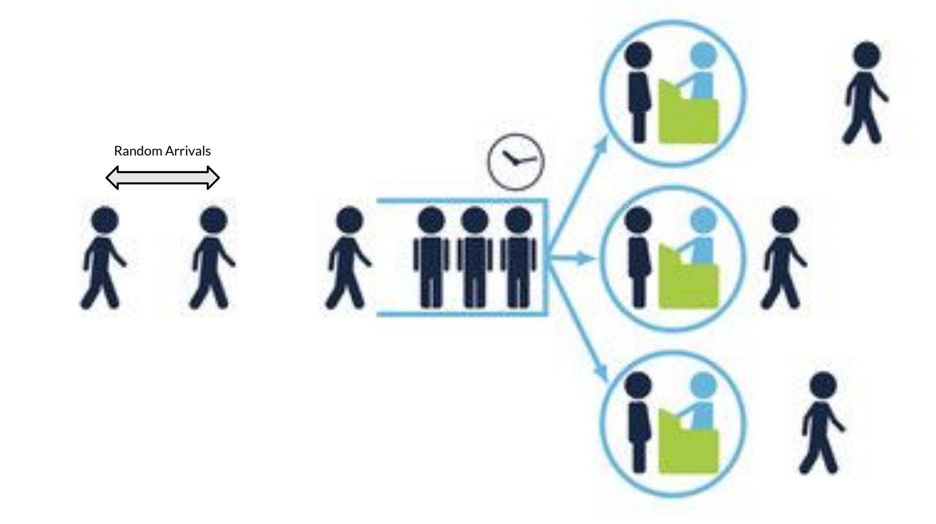
# INDEPENDENT ARRIVALS

The receipt of each service request is independent of all the previous requests

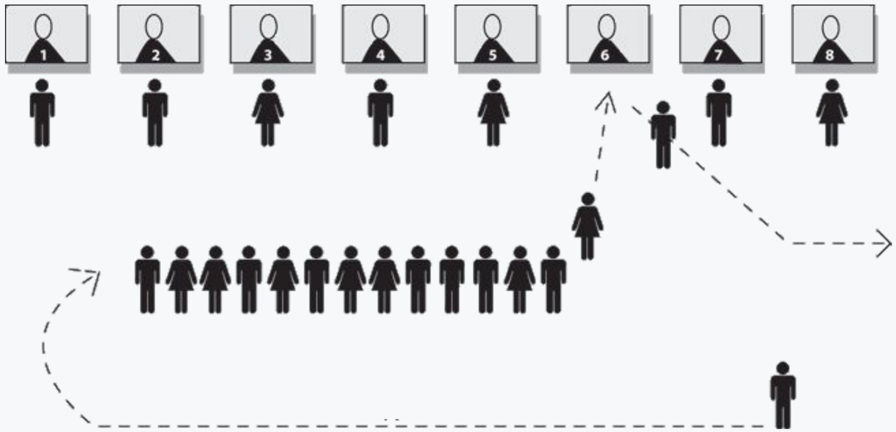
Since all the service requests can, in general, be assumed to be independent of each other, we expect that the inter-arrival times between subsequent requests are also all independent. Thus we can model the arrivals as a Counting Process.

**The POISSON DISTRIBUTION is the easiest representation of a Counting Process. While there can be other representations, most queueing models utilize the PD for convenience.**

Given a Poisson Distribution, the inter-arrival times are Exponentially Distributed. Thus, we can simulate the time points for all incoming requests using the Exponential Distribution. What remains, is to simulate service by the employees and check for proper workload distribution. No employee should be over-burdened.



# SIMULATING SERVICE




## 01 ASSUMING DISTRIBUTION

While we have the AHT for both calls and chats, the actual time taken over handling a request may vary. We have assumed the Service Times to be Exponentially Distributed.

## 02 IDENTIFYING SERVICE

Since we are dealing with both calls and chats simultaneously, before simulating the service time, we must identify whether the request is a call or a chat.

## 03 INCREMENTING AGENTS

We begin each simulation with a single agent attending all requests. We increase the agents till we finally achieve our desired Target Service Level.

## 04 ENSURING SIMULTANEITY

While one agent serves one request, another agent should be able to serve another request. We ensure this by keeping a track of each employee's status (i.e. 'BUSY' or 'FREE')

# JUDGING SIMULATION

To accept or reject whether a particular number of Employees would be enough

## TARGET ANSWER TIME

To always answer all calls or chats under a fixed pre-decided time might require a large number of employees under certain scenarios. Thus, we define the Target Answer Time with respect to a predefined Service Level. The Industry Standard for these is taken to be 20 seconds and 80% respectively. Put together, this means that we strive to answer 80% of all requests within 20 seconds of waiting time.

We can define a different Target for both types of requests. To check whether a certain number of employees meets the set target, we store the waiting times for each call and chat in two separate arrays. From there we can easily check if SL% of the waiting times are less than the TAT.

## ADDRESSING RANDOMNESS

Since a simulation is bound to be random, we run each simulation 100 times and take the average of each output.

## MAXIMUM OCCUPANCY

We might come across scenarios where despite the model meeting the Target Answer Time and the Service Level, the employees may be expected to work over 70 minutes every hour!

This above scenario should obviously not arise. And we must reject that particular simulation if it does. Therefore, we must define what is the maximum amount of time an employee may be expected to work per hour. We can in principle take this to be any amount less than 60 minutes, however, the employees must get rest between calls to perform well on the future Service Requests. As an Industry Standard we can take the Maximum Occupancy to be 80% - 90%.

# UNDERSTANDING SHRINKAGE

**20% - 35%**

of the employees at any point of time are assumed to be not available at work despite being paid to work

We must take into account, that despite employees being paid to work, employees could be absent due to a variety of reasons. Employees could be on paid or sick leaves. They could even be on a vacation.

Shrinkage also takes into account the employees who might have to leave their desks for bio-breaks or for lunch. During this time, they wouldn't be available to attend a Service Request. All of this absenteeism amounts to roughly 20% - 35% of the total work force. It is very important to take this into account. If we do not, then the rest of the employees may be over burdened and the service will deteriorate.

Thus, if our model gives us that the number of employees required to achieve the Service Level is, say, x, we must remember that x should constitute just about 70% of the workforce. The total workforce should be calculated accordingly.

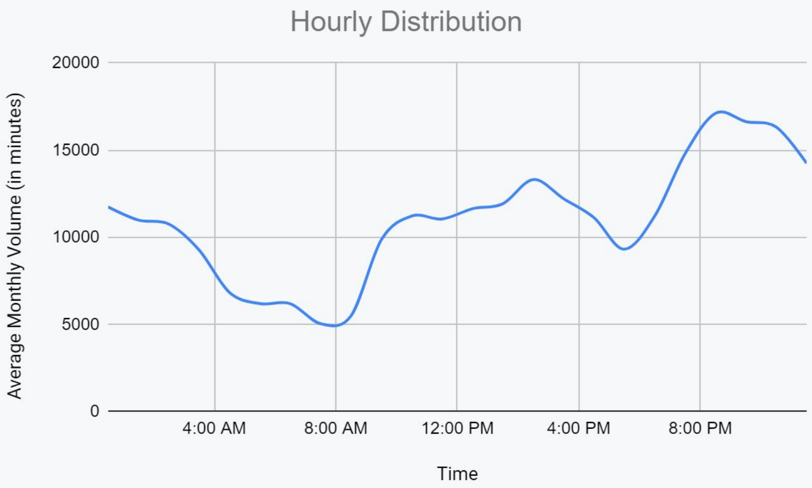
# PHASE II COMPLETE

The two-phase model, as described in the previous slides, will provide us with the number of employees required to attend to all the incoming service requests, while maintaining the desired service level. This shall also take into account factors like Shrinkage and Maximum Occupancy.

The model provides us with this answer for each hour of every day of the upcoming month(s). So, what remains is to understand this output and to estimate the employees required for the month.

# SHIFTS

# DAILY EMPLOYEE REQUIREMENT




Each day is divided into three shifts. The shift timings are given to be:

- 06:00 AM - 03:00 PM
- 03:00 PM - 11:00 PM
- 11:00 PM - 06:00 AM

Since an employee can work only one shift in a day, the daily employee requirement is equal to the sum of the employee requirement in all the three shifts.

The same set of employees work the whole shift. So the employee requirement of the shift can be inferred from the busiest hour in the shift. For this graph, we can see that for the first shift the busiest hour is the final hour. So, the number of employees required from 02:00 PM to 03:00 PM, as obtained by the model, should be taken as the employee requirement of the first shift. The same can be done for each shift.

If we have the employee requirement for each shift, the employee requirement for that day can be obtained by adding the three up.

It should be noted, that this graph is based on historical data. The future data might differ and the analysis for that should be done using the same.

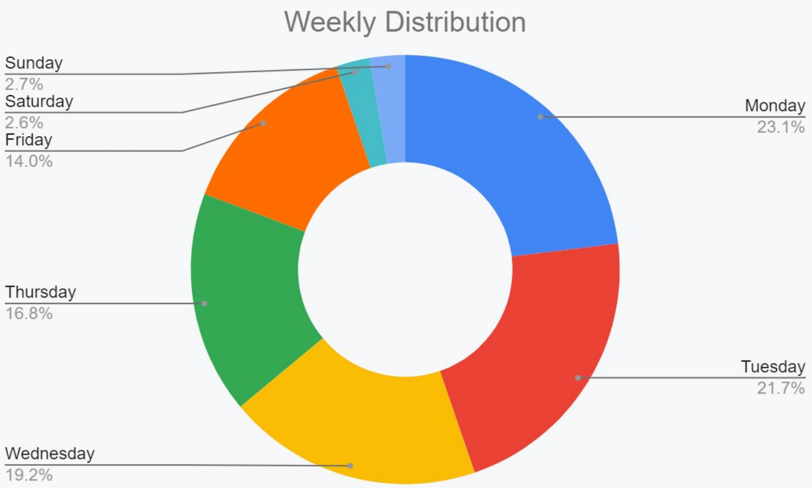
# WEEKLY EMPLOYEE REQUIREMENT

While calculating the Weekly Requirement, there are a few things which need to be taken care of. If we are forecasting the Employee Requirement of the month of February, we will have four data points corresponding to the Employee Requirement of each day of the week. While these data points are not expected to vary greatly, for each day of the week the Employee Requirement can be taken to be the Median of these data points. This shall eliminate large deviations from the trend, if any.

Once we have determined the Employee Requirement for each day of the week, the Weekly Employee Requirement can be determined as below.

Since each employee works five days a week, if we take the sum of the Employee Requirements of each day of the week, we should end up having counted each employee five times. Therefore, dividing this sum by five should give us the Weekly Employee Requirement.

$$WR = \frac{\sum_{i=1}^{7} DR_i}{5}$$



# CONCLUDING NOTE

We have discussed a predictive model which incorporates Time Series Forecasting and Simulation Techniques to predict the optimum number of Employees required at a remote IT Service Desk to achieve the desired service level.

The model attempts to determine a close to real life estimate of the **WAITING TIME** as well as the **SERVER OCCUPANCY** both of which are very important factors for predicting the number of employees.

- With new data available every passing month, the model can be retrained, and the hyperparameters can be recalibrated for more accurate results.
- This shall ensure that the organisation is able to maximise customer satisfaction while minimising cost to the organisation. Thus, it is a win-win situation for both parties.

## “IN GOD WE TRUST, ALL OTHERS MUST BRING DATA.”

## THANK YOU!