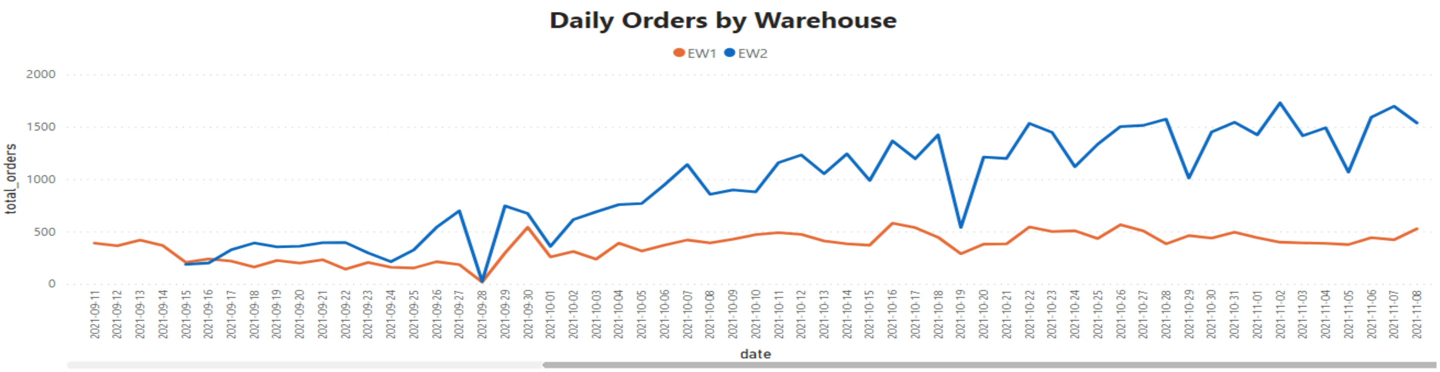
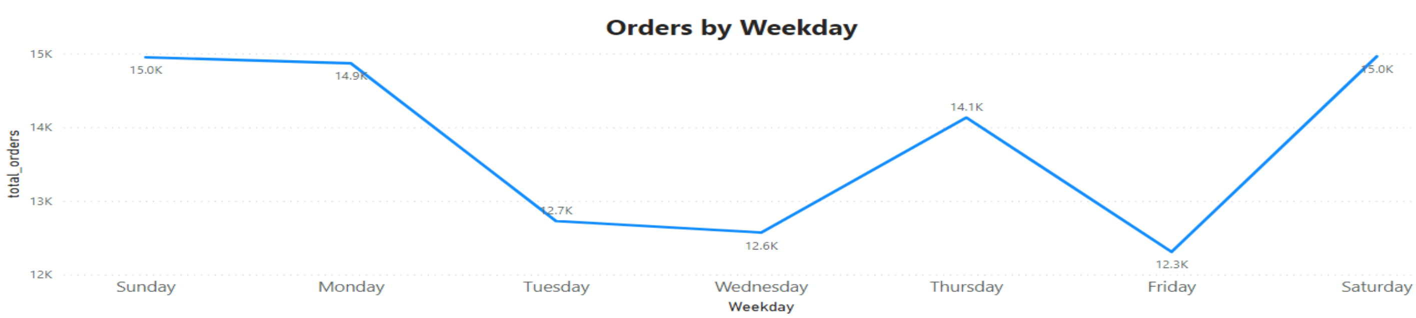
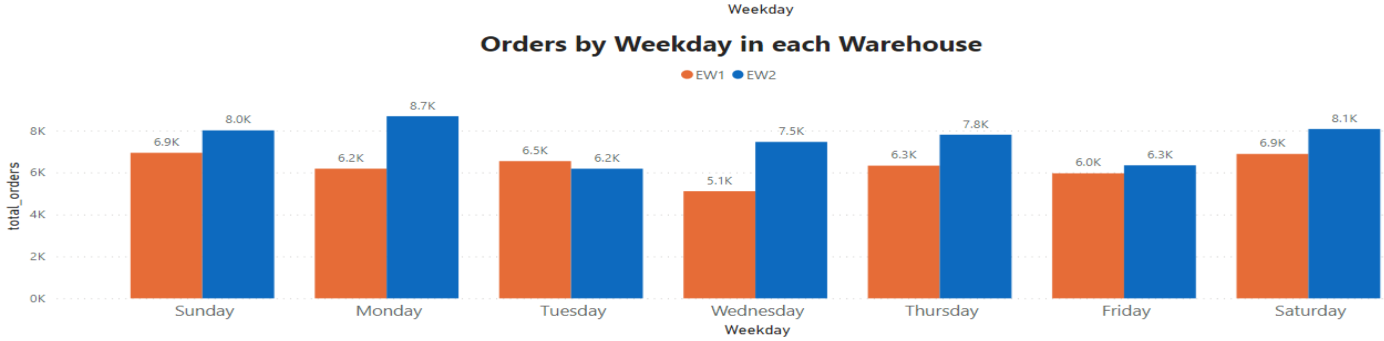
Question 1 - Mncedisi Mncwabe

Tools used: *Python and Microsoft Power BI*

Problem 1







- Warehouse EW2 didn't have orders until the 15th of September 2021. Of which this warehouse then consistently received more daily orders than warehouse EW1. Warehouse EW2 also had more monthly orders than warehouse EW1 except in September.

- There's a peak on number of orders received by the store on Saturdays, Sundays and Mondays and both warehouses receive the highest orders during these days compared to the other days of the week.

Problem 2

Forecasted Daily Orders

EW1

Date	Forecasted Orders
11/9/2021	435
11/10/2021	422
11/11/2021	405
11/12/2021	449
11/13/2021	502
11/14/2021	463
11/15/2021	480

EW2

Date	Forecasted Orders
11/9/2021	1801
11/10/2021	1455
11/11/2021	1778
11/12/2021	1329
11/13/2021	1648
11/14/2021	1776
11/15/2021	1822

- The suggested number of store operators for EW1 is 7 store operators a day for EW1 warehouse and 26 store operators a day for EW2 warehouse. This based on the forecasted number of orders between 9-15 Nov 2021 for each warehouse. Since in the exploratory data analysis showed that the two warehouses tend to have a peak on weekends and Mondays, the forecasted order values are also highest on these weekdays (13,14 and 15 November 2021)

Problem 3

- The data is a univariate time series data. The model(s) that is more suitable for this data is a time series model. The data has some seasonality (order peaks on weekends and Mondays), hence the appropriate is a model that can account for seasonality. I selected SARIMA (Seasonal Autoregressive Integrated Moving Average) to forecast the daily orders. This model accounts for seasonal component in the data with decent accuracy.

- Other features that can be used in the model would be promotion features since promotions can increase the number of orders the store receives.

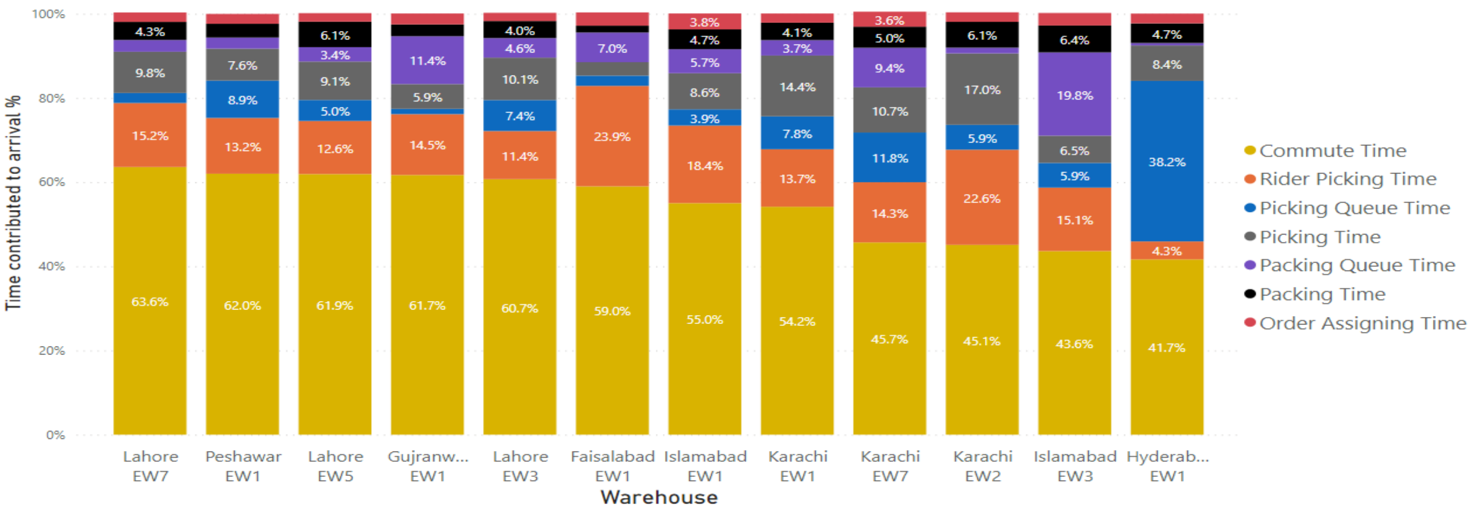
Problem 4

- The forecast accuracy is measured by comparing the forecasted order value and the actual order value. For example, if the forecasted order value is 5 and the actual order value is 10, then the accuracy of the forecasting model is (5/10)*100 = 50%.

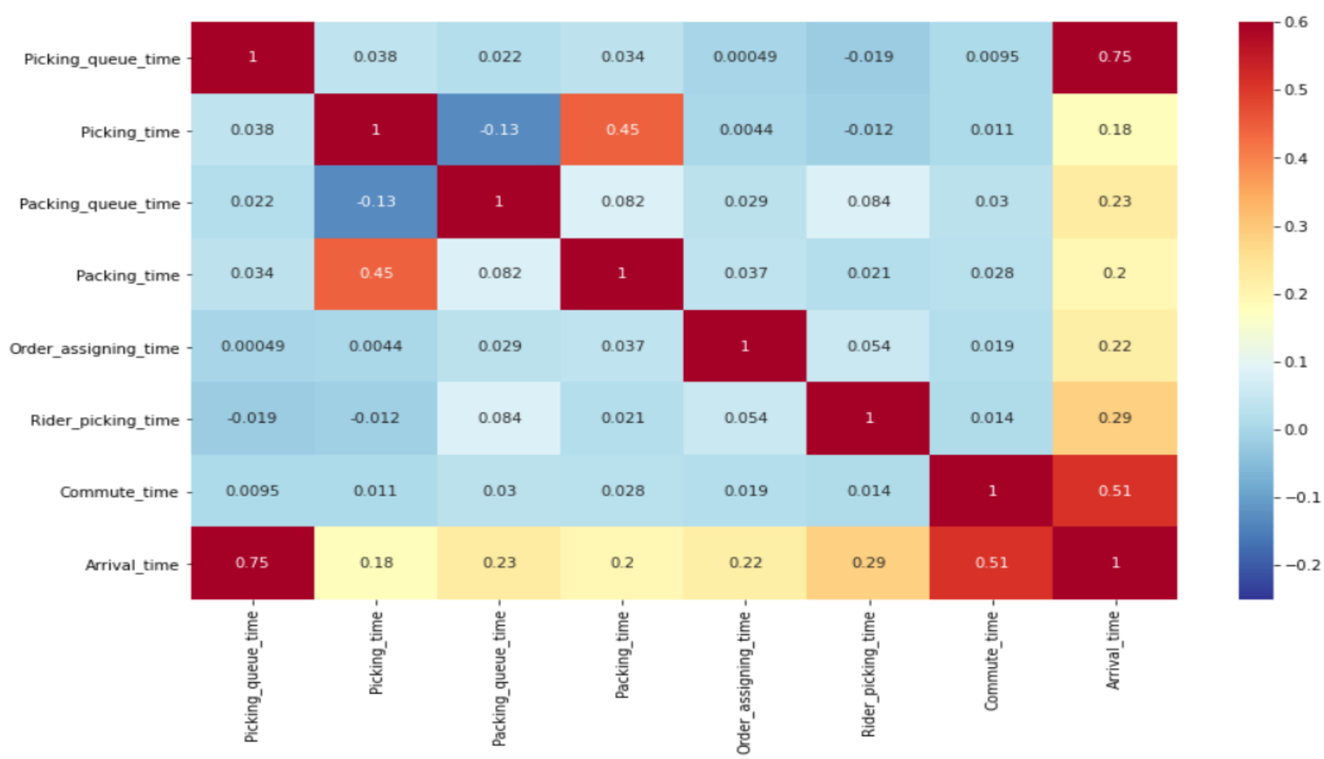
NB: Question 2 (next page)

Question 2

Which part of the delivery chain contributes the most percentage time taken to total arrival time?



Correlation between total arrival time and other delivery chain times.



Methodology: By looking at the data it's clear that the total arrival time of an order is how much time was taken by each part of the delivery chain. For example, if the total arrival time for order 123 was 40 minutes, then the 40 minutes is made up of all the times taken by each delivery chain parts. So, by computing time percentages contributed by each delivery chain part to total arrival time of an order I identified which part of the delivery chain contributes the most time to total arrival time.

- The first plot (stacked bar chart) displays the percentage time taken by each part of the delivery chain for each warehouse. Commute time contributes the most time that an order takes to arrive to the customer for all warehouses and should be a focus point to find a solution to minimize the time taken in this part of the delivery chain. The second part that should be focused on is Rider Picking time for all warehouses except for Hyderabad EW1 warehouse which has Picking Queue time as the second part of the delivery chain which should be focused on.

- The second plot (heatmap correlation) confirms what can be seen from the stacked chart. This plot shows the correlation between arrival time and other delivery chain parts. Picking Queue time, Rider Picking Time and Picking Queue time all have the highest positive correlation with the arrival time, which indicates a strong relationship between arrival time and these delivery chain parts.