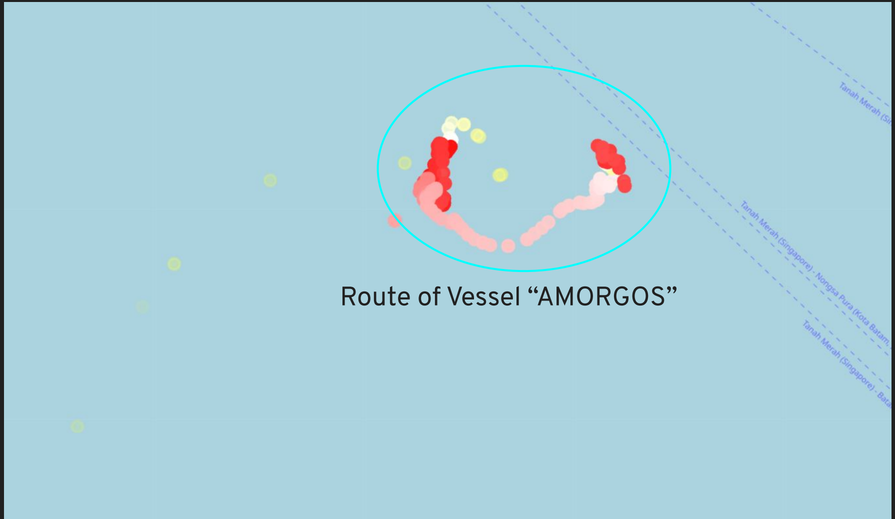
EMISSION REDUCTIONS

Through Just-In-Time (JIT) Arrival Strategies

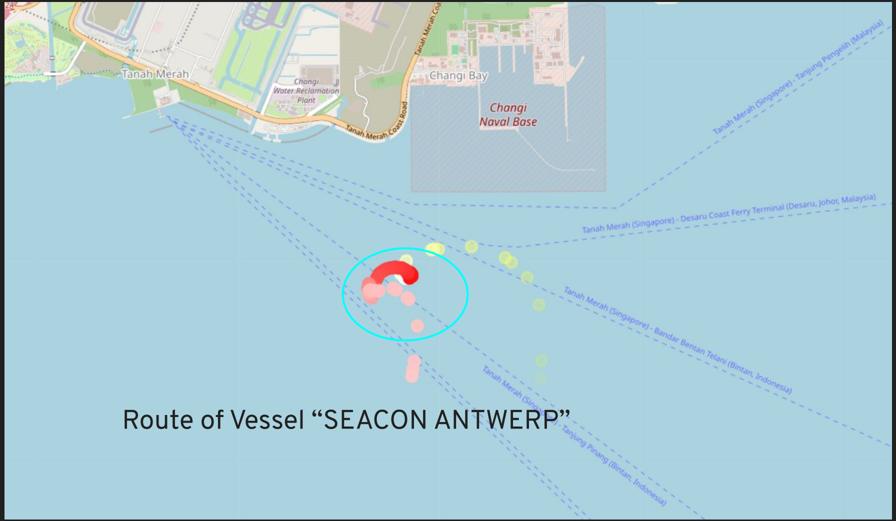
Made by Team participants – Soh Jung Xuan, Loke Sheng Jun, Tan Kai EI

PROBLEM

- Congestion of berths in Singapore causes long delays for ships
- Consumes fuel on idle and in turn increases carbon emissions



Route of Vessel “AMORGOS”



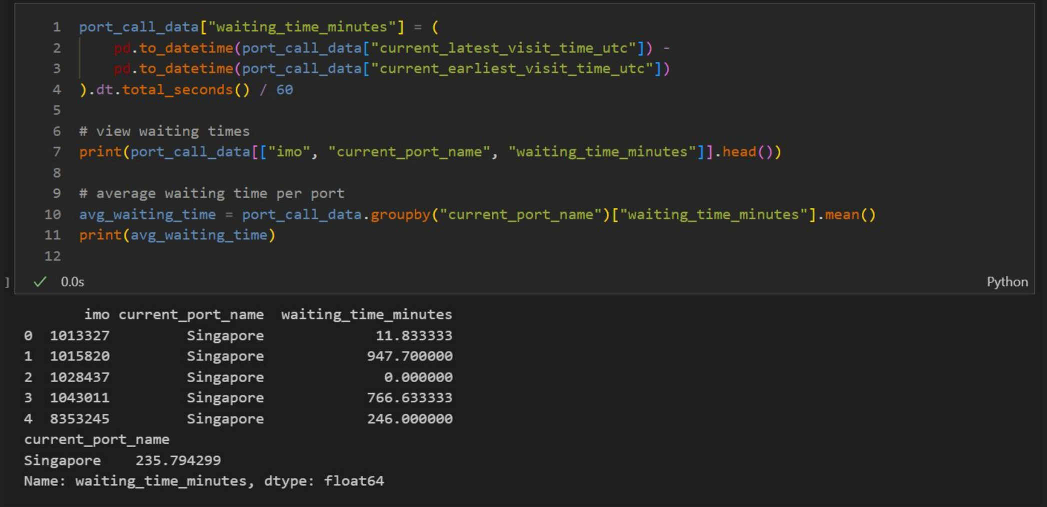
Route of Vessel “SEACON ANTWERP”

Vessels are unable to dock due to congestion and therefore linger at anchorage – near the docks.

ANALYSIS

Waiting Duration

Idle Time = current latest visit time (UTC) - current earliest visit time (UTC)



```python
port_call_data["waiting_time_minutes"] = (
    pd.to_datetime(port_call_data["current_latest_visit_time_utc"]) -
    pd.to_datetime(port_call_data["current_earliest_visit_time_utc"])
).dt.total_seconds() / 60

# view waiting times
print(port_call_data[["imo", "current_port_name", "waiting_time_minutes"]].head())

# average waiting time per port
avg_waiting_time = port_call_data.groupby("current_port_name")["waiting_time_minutes"].mean()
print(avg_waiting_time)
```

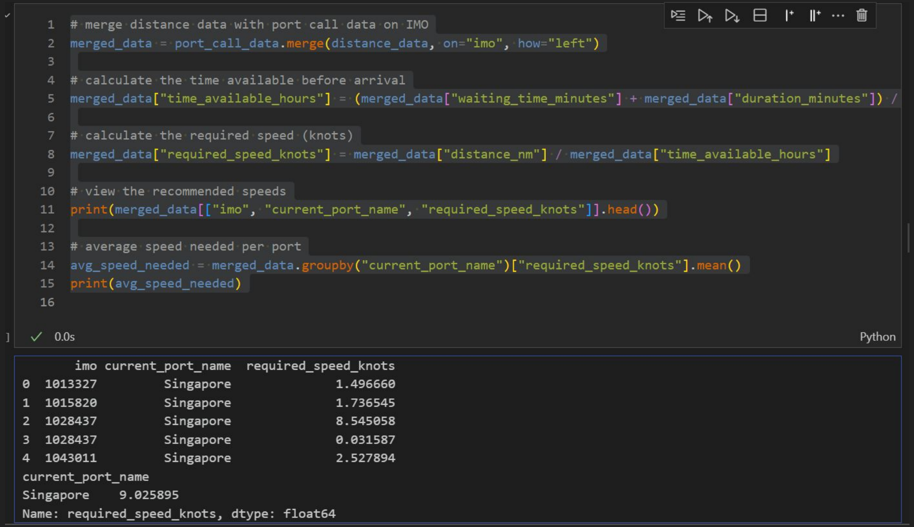
```
       imo current_port_name  waiting_time_minutes
0  1013327         Singapore             11.833333
1  1015820         Singapore            947.700000
2  1028437         Singapore              0.000000
3  1043011         Singapore            766.633333
4  8353245         Singapore            246.000000
current_port_name
Singapore    235.794299
Name: waiting_time_minutes, dtype: float64
```

= ***236 minutes*** *(3s.f.)*

ANALYSIS

Optimal Speed

Distance / Time, where Time = Idle Time + Remaining Voyage Duration



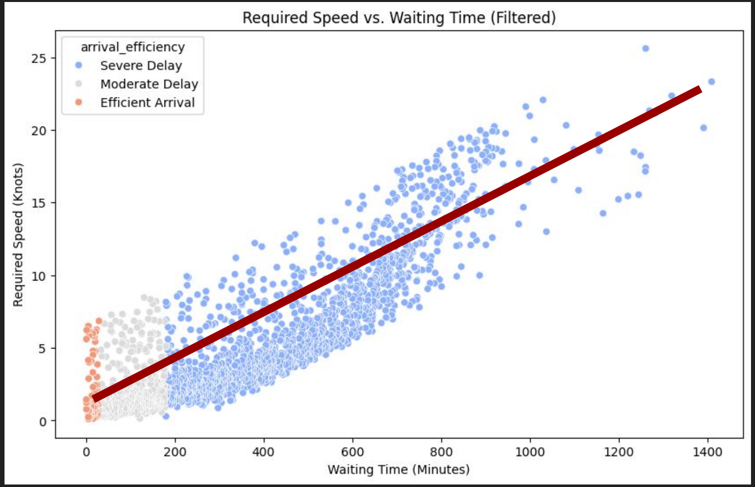
```python
# merge distance data with port call data on IMO
merged_data = port_call_data.merge(distance_data, on="imo", how="left")

# calculate the time available before arrival
merged_data["time_available_hours"] = (merged_data["waiting_time_minutes"] + merged_data["duration_minutes"]) /

# calculate the required speed (knots)
merged_data["required_speed_knots"] = merged_data["distance_nm"] / merged_data["time_available_hours"]

# view the recommended speeds
print(merged_data[["imo", "current_port_name", "required_speed_knots"]].head())

# average speed needed per port
avg_speed_needed = merged_data.groupby("current_port_name")["required_speed_knots"].mean()
print(avg_speed_needed)

```

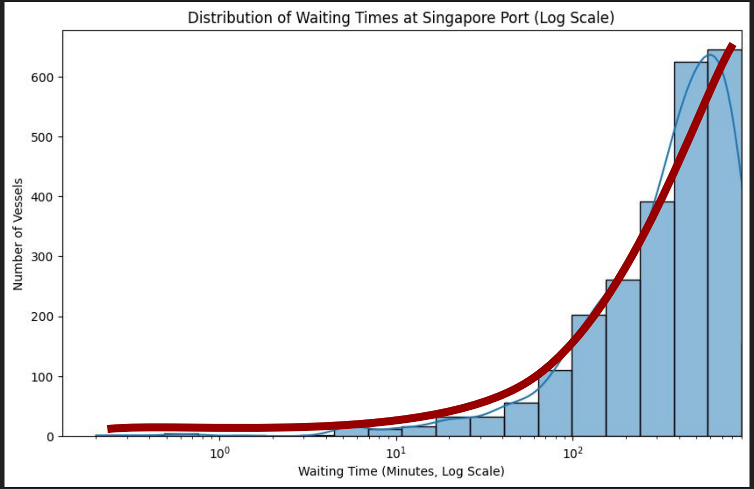
```
       imo current_port_name  required_speed_knots
0  1013327         Singapore              1.496660
1  1015820         Singapore              1.736545
2  1028437         Singapore              8.545058
3  1028437         Singapore              0.031587
4  1043011         Singapore              2.527894
current_port_name
Singapore    9.025895
Name: required_speed_knots, dtype: float64
```

= ***9.02 knots*** *(3s.f.)*

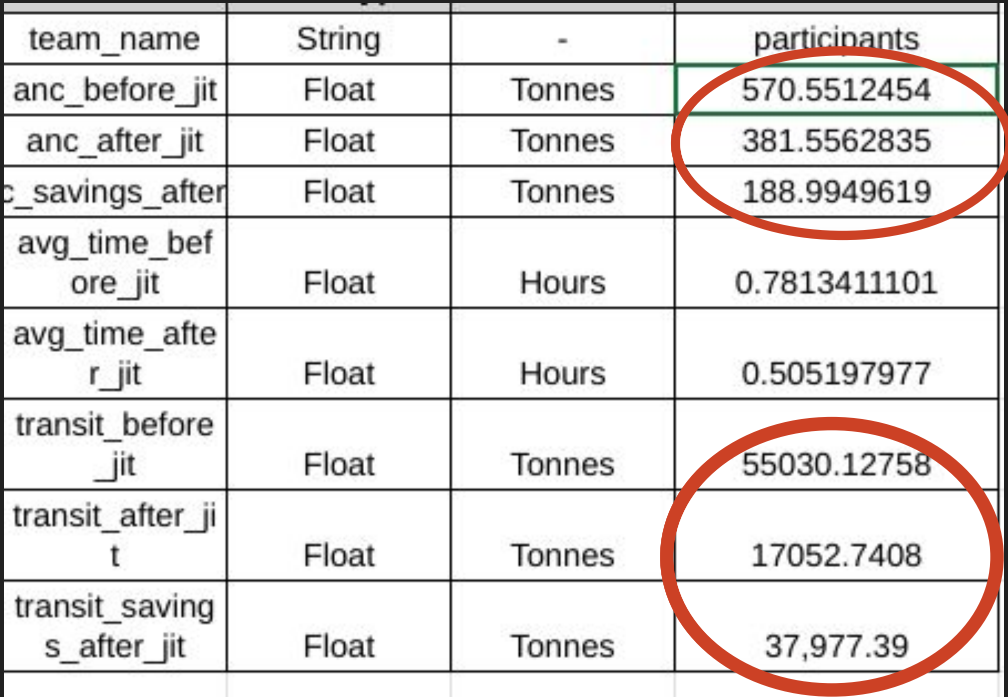
OUR SOLUTION

A machine learning model that predicts optimal **vessel speed** and optimal **number of vessels** allowed to dock at a certain point in time with respect to **idle time**.





RESULTS



team_name	String	-	participants
anc_before_jit	Float	Tonnes	570.5512454
anc_after_jit	Float	Tonnes	381.5562835
c_savings_after	Float	Tonnes	188.9949619
avg_time_before_jit	Float	Hours	0.7813411101
avg_time_after_jit	Float	Hours	0.505197977
transit_before_jit	Float	Tonnes	55030.12758
transit_after_jit	Float	Tonnes	17052.7408
transit_savings_after_jit	Float	Tonnes	37,977.39

32%

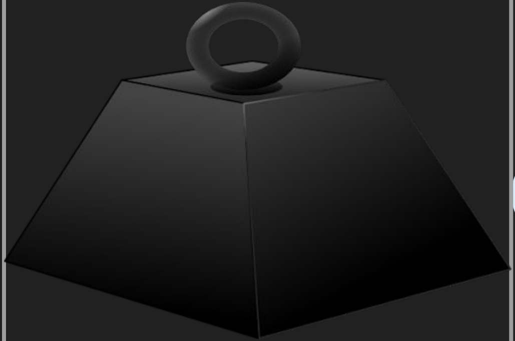
decrease in anchorage emissions

69%

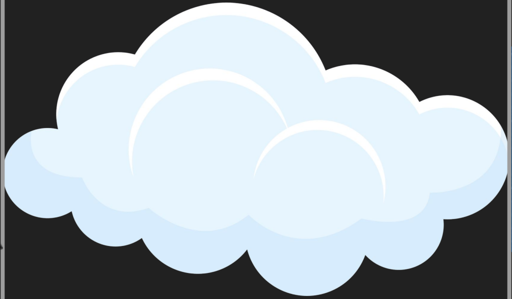
decrease in transit emissions

FUTURE IMPROVEMENTS

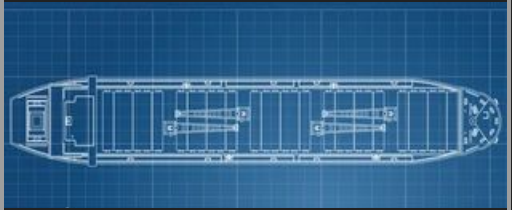
Allow model's predictions to account for the load of a ship



Reschedule based on weather conditions



Adjustments based on dimensions and features of ships



THANK YOU!