Measuring Delivery Agent Efficency using DEA

For the MSBA project, we applied the data envelopment analysis(DEA) to evaluate the efficiency of delivery agents working for Zomato. Using performance data from a public Kaggle Dataset, we explore factors such as delivery time, traffic conditions, and agent age and how these affect customer satisfaction. DEA offers a fair and rigorous way to identify high-performing agents and benchmark improvements for others.

Business Introduction:
Zomato is a food delivery platform that connects restaurants, freelance delivery agents, and customers. Their operations rely heavily on delivery speed & service quality. Like its American counterpart, Doordash, Zomato hires thousands of freelance delivery agents to operate under various conditions.

Data & Methodology:
Kaggle Dataset: [Zomato dataset]

The dataset was sourced from Kaggle's Zomato Delivery Operations Analytics dataset. It includes delivery agent demographics, timestamps, location data, traffic conditions, and customer ratings. After processing, the following variables were used:

- Inputs:
  - Avg Time Taken(min)
  - Avg Traffic Score(Converted to a 1-4 scale)
  - Avg Agent Age
- Output:
  - Avg Customer Rating(1-5)

```r
library(tidyverse)
library(readxl)
library(deaR)


zomato_raw <- read.csv("C:\\Users\\jroch\\Downloads\\Zomato Dataset.csv\\Zomato Dataset.csv")


# "Low" = 1, "Medium" = 2, "High" = 3, "Jam" = 4
traffic_map <- c("Low" = 1, "Medium" = 2, "High" = 3, "Jam" = 4)
zomato_raw$Traffic_Score <- traffic_map[zomato_raw$Road_traffic_density]

#Remove NA or problematic rows
zomato_clean <- zomato_raw %>%
  filter(!is.na(Delivery_person_ID),
         !is.na(Time_taken..min.),
         !is.na(Delivery_person_Ratings),
         !is.na(Traffic_Score),
         !is.na(Delivery_person_Age),
         !is.na(multiple_deliveries))

# Rename columns for simplicity
zomato_clean <- zomato_clean %>%
  rename(
    Agent_ID = Delivery_person_ID,
    Time_Taken = Time_taken..min.,
    Agent_Rating = Delivery_person_Ratings,
    Agent_Age = Delivery_person_Age,
    Multiple_Deliveries = multiple_deliveries
  )

#Aggregate per Delivery Agent
dea_df <- zomato_clean %>%
  group_by(Agent_ID) %>%
  summarise(
    Avg_Time_Taken_Min = mean(Time_Taken, na.rm = TRUE),
    Avg_Traffic_Score = mean(Traffic_Score, na.rm = TRUE),
    Avg_Multiple_Deliveries = mean(Multiple_Deliveries, na.rm = TRUE),
    Avg_Agent_Rating = mean(Agent_Rating, na.rm = TRUE),
    Avg_Agent_Age = mean(Agent_Age, na.rm = TRUE)
  ) %>%
  ungroup()

# Save Aggregated DEA Table
write.csv(dea_df, "DEA_Agent_Table.csv", row.names = FALSE)

#View the result
View(dea_df)
```

DEA Model:
I implemented an input-oriented DEA model with constant returns to scale (CRS) using the deaR package in R. This setup minimizes input use (delivery time) while maintaining output levels (customer satisfaction).

```
#run dea
dea_result <- model_basic(dea_data,
                          orientation = "io",
                          rts = "crs",
                          dmu_eval = 1:nrow(df),
                          dmu_ref = 1:nrow(df))
```

Results & Analysis:
DEA classified only 7 out of 1320 agents as fully efficient(efficiency score =1), indicating that DEA is a strict evaluator. However, the majority of agents scored between 0.85 and 0.8,5, meaning that most agents are not far behind this efficiency metric and need minor adjustments.

```
# Inputs: Time, Traffic, Age (cols 2,3,6), Output: Rating (col 5)
dea_data <- read_data(df,
                          ni = 3,
                          no = 1,
                          dmus = 1,
                          inputs = c(2, 3, 6),
                          outputs = 5)



#run dea
dea_result <- model_basic(dea_data,
                          orientation = "io",
                          rts = "crs",
                          dmu_eval = 1:nrow(df),
                          dmu_ref = 1:nrow(df))

#effeciency scores
eff <- efficiencies(dea_result)
print(eff)
write.csv(eff, "agent_efficiencies2.csv")

# targets for improvement

targets_df <- targets(dea_result)
write.csv(targets_df, "dea_targets.csv")

#plot and summary reprot

plot(dea_result)
report <- summary(dea_result)
write.csv(report, "dea_summary_report2.csv")
```

Age and Efficiency Correlation:
A Pearson correlation analysis found a strong negative correlation between agent age and DEA efficiency ($r = -0.74$, $p<0.001$). This may reflect a slower pace, reduced adaptability, or higher input strain. This doesn't mean older agents are worse; it highlights that additional training may be needed as agents progress in age.

```r
# Agent age vs correlation
cor.test(df$Avg_Agent_Age, eff, method = "pearson")

eff <- efficiencies(dea_result)
df$Efficiency <- eff

library(ggplot2)

ggplot(df, aes(x = Avg_Agent_Age, y = Efficiency)) +
  geom_point(alpha = 0.6, color = "steelblue") +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "Agent Age vs DEA Efficiency",
       x = "Average Agent Age",
       y = "Efficiency Score") +
  theme_minimal()
```

Recommendations:
Based on DEA results, Zomato can take the following steps:

- Assign high-traffic agents to less congested routes
- Offer time-efficiency training for slower but well-rated agents
- Build agent performance dashboards based on DEA insights
- Train others using reference agents as models
- Monitor new hires against benchmark profiles

Conclusion:
This project demonstrates how DEA can bring clarity and fairness to performance evaluation in a noisy, real-world environment. By adjusting for traffic, age, and time conditions, DEA allows Zomato to identify top-performing delivery agents and give everyone else a roadmap to improve. These insights can be applied across logistics, delivery, and service platforms looking to boost efficiency without oversimplifying performance metrics.