Business Case Study

1. Autonomous Shipment roll-out: Autonomous Delivery
2. Value of Customers: What makes a customer valuable?

- **Mohamed Imthiyas Abdul Rasheeth**

Table of Contents

Part 1 - Autonomous Shipment roll-out: Autonomous Delivery

Part 2 – Value of customers: What makes a customer valuable?

Part 1 - Autonomous Shipment roll-out: Autonomous Delivery

Introduction:

Autonomous Shipment, a start-up venture, aims to use robot drones for last-minute deliveries to the customer's doorsteps. Several venture capitalists and the UK Government backed this project. Before the official launch, the company wants to run a trial in Grocery, Clothing and Sports stores to find its efficiency. We have been provided with the data and the goals to achieve for the trial run.

Background to Problem:

The company has developed four robot prototypes: Robot A032 – Archer, Robot B23 – Bowler, Robot CJKL – Corner and Robot DSXX – Deviant. We have a limited budget of **£250,000** for the one-month project. We have two decisions to make. They are,

1. Decide on a robot for the trial run based on the requirements.
2. Allocate the number of robots to maximise delivery and be under budget.

Findings:

This section of the report deals with the approach to our business problem and proposes an optimal solution to each goal. All the R codes and links to the Excel workbook can be found in the appendix.

Goal 1 – Decision on Robot Prototype:

Our first task is to find a robot prototype that can be used for the trial run across all the stores. We have four types of robots to choose from, and the decision should be taken based on the following criteria:

- Carrying Capacity, Battery Size, Average Speed and Reliability should be as high as possible.
- The Cost per unit should be as low as possible.

We are only focused on the hardware part of the robot and not the software to make our decision.

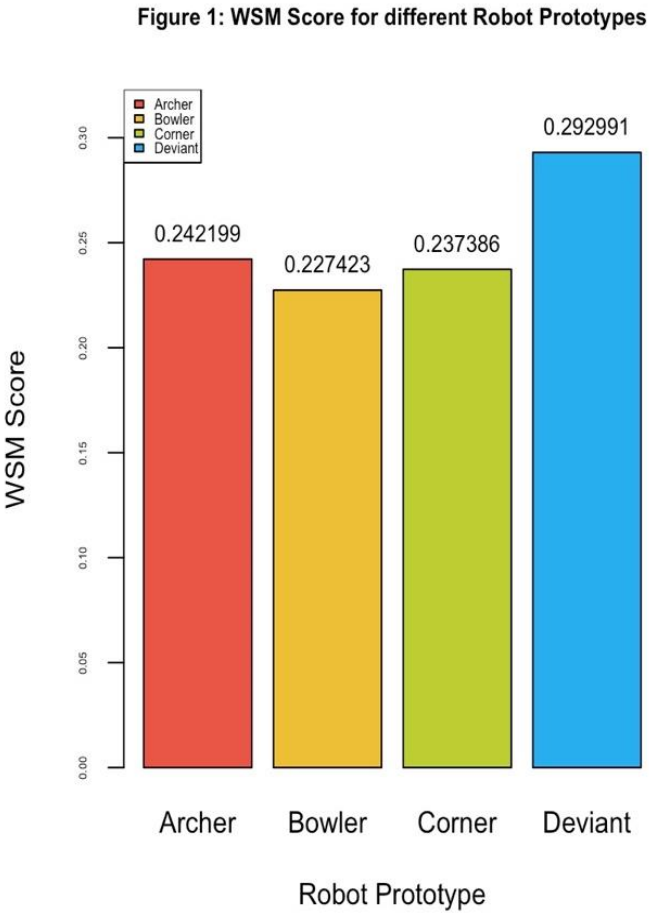
Since we must find the best solution, we can use the WSM (Weighted-Sum Method) or the AHP (Analytic Hierarchy Process) method to reach our goal. We can also use TOPSIS or VIKOR methods, but they are less effective as they deal with near and away from an optimal point. We prefer **WSM** to AHP, as we are supposed to find the **maximum utility** solution, while AHP deals with **Pairwise Comparison**, which we don't seem to have the proper data to work with.

Although weights are not provided, we will **estimate** them based on the comparisons. The comparison follows **Carrying Capacity < Average Speed < Battery Size < Cost per unit < Reliability**. We can see that it is compared using a **scale of 5** and that the Cost per unit is favoured by at least 25% of the total criteria. We estimated the weights by dividing 100 into 15 (=5+4+3+2+1) parts, which gives us 6.666% for each part. With this calculation, we assigned the weights as follows:

Robot_Prototype	Weights	Archer	Bowler	Corner	Deviant
Carrying Capacity - Max	6.67%	45	50	60	40
Battery Size - Max	20%	18	18	12	24
Average Speed - Max	13.33%	6	4	4	10
Cost Per Unit - Min	26.67%	5210	6250	4500	7100
Reliability - Max	33.33%	22	24	24	32

Table 1: Weights and Data used in the WSM Method to find the best robot for trial run.

We can confirm this is the correct estimate of the weights by adding all the percentages, which sums to 100, and the Cost per Unit is over 25% in Table 1.



The WSM method is used to find an optimal solution when we have the weightage of the criteria. We first maximise all the criteria so that we have **consistency** throughout. The higher the value, the better the score. We maximise by taking the reciprocal of that value.

Since each criterion is on a different scale, we will then **normalise** the criteria so that the results will not be biased towards criteria with larger values. The scale after normalisation will be between 0 and 1. To normalise, we divide each value by the sum of all the values for a criterion.

Finally, we calculate the Weighted Sum for each robot to get the best robot. The **higher** the WSM score, the **better** solution for our task.

WSM Score = Sum (Criteria * Weight)

After calculation, we get a bar plot in Figure 1, which shows that the **Deviant** has the highest WSM score, followed by Archer, Corner and Bowler.

Although the WSM method gives us the best solution based on the importance of the criteria, the limitation of this method is that the decision changes when we add or remove criteria during analysis or when we change their weights. The **Deviant** robot is the best prototype for our trial run if we only use these criteria without changing their weights.

Goal 2 – Allocation of Robots:

Our second goal is to come up with the number of robots to be allocated to the stores where each store has its requirements and constraints. The participating stores for the trial run are **Grocery**, **Clothing** and **Sport Equipment**. We have the following details below,

- The budget is **£250,000** for one month.
- The total labour hours are **250** hours a week.
- Grocery Store: We need at least 5 robots, 10 labour hours, 9 orders per robot per day and £8700 (£7100 Robot + £1600 Operation Cost) per robot per month.
- Clothing Store: We need at least 5 robots, 7 labour hours, 6 orders per robot per day and £8100 (£7100 Robot + £1000 Operation Cost) per robot per month.
- Sports Store: We need at least 5 robots, 5 labour hours, 4 orders per robot per day and £7700 (£7100 Robot + £600 Operation Cost) per robot per month.
- The number of deliveries should be as **maximum** as possible.

Since it is an optimisation problem, we can use either a linear or goal programming method. We will not use Goal Programming since we don't know the exact maximum number of deliveries to

be made each day. Thus, we use **Linear Programming** to maximise the number of deliveries while strictly not exceeding budget and labour hours.

With the given objective and constraints, we can formulate mathematical expressions as follows,
Input Variable:

X1 = Grocery Store
X2 = Clothing Store
X3 = Sports Equipment Store

Objective:

Maximise (Number of Delivery) $Z = 9X1 + 6X2 + 4X3$

Problem Constraints:

Cost -> $8700X1 + 8100X2 + 7700X2 \leq$ £250,000 (Monthly)
Labour -> $10X1 + 7X2 + 5X3 \leq 250$ (Weekly)

where $X1, X2, X3 \geq 5$.

We can use Solver in Excel to find the solution. The Decision Variable row gives us our optimal allocated number of robots.

Goals	X1 (Grocery)	X2 (Clothing)	X3 (Sports)	Actual	Target
Cost (Monthly)	8700	8100	7700	244300	250000
Labour (Weekly)	10	7	5	250	250
Order (Daily)	9	6	4	221	
Decision Variable	19	5	5		

Table 2: Linear Programming table in Excel worksheet

Table 2 shows that we need **19 Deviant** robots for the **grocery** store, **5 Deviant** robots for the **clothing store,** and **5 Deviant** robots for **sports** stores. If we allocated these numbers of robots, we would utilise our **250** labour hours a week, with a total cost of **£244300** and can also deliver **221** orders a day by three stores combined.

Conclusions:

We employed WSM as our Multi-Criteria Decision Analysis (MCDA) method for our first task. We chose **Robot DSXX – Deviant** for our trial run since the WSM score is the highest among others. Furthermore, we used Linear Programming as our MCDA method for our second task because we are dealing with the optimisation technique to maximise the number of deliveries to the local customers. Thus, we allocated **19** Deviant robots for the grocery store, **5** Deviant robots for the clothing store, and **5** Deviant robots for the sports equipment store, which would deliver **221** daily orders by all three stores combined.

Part 2 – Value of customers: What makes a customer valuable?

Introduction:

An e-commerce website, drinks@home.uk, operating in Great Britain, deals with Alcoholic and Non-alcoholic beverages from across the world. The company wants to study customer behaviour and how it impacts the Revenue on the website.

Background to Problem:

We are given data on 400 customers from previous orders. We have two objectives to meet using the provided data. They are,

1. Factors that positively and negatively influence customer's spending on the website.
2. Decide on the best method to increase profit on the website from below.
   i) Run an advertisement targeting customers over 45 years old.
   ii) Provide a £20 offer voucher on their next order.
   iii) Advertise with an influencer.

Findings:

This section deals with our data and models it to make a better decision based on the insights. All the R codes and links to the Excel workbook can be found in the appendix.

Data Understanding:

We have data of 400 customers on the website. The data includes the following columns,
Revenue – Revenue generated in GBP from the latest order.
Advertisement Channel – The channel that brought the customer to the website.
(1-Leaflet, 2- Social Media, 3- Search Engine, 4-Influencer)
Estimated Age – The age of the customer according to tracking software.
Estimated Income – Income of the customer according to tracking software.
Time on Site – Time spent on the website per week in seconds.
Seen Voucher- If they saw a voucher pop-up.

Since our data has labels, we will use a **supervised** model. Thus, we use a Linear Regression or a Logistic Regression based on our objectives.

Goal 1 – Factors Affecting Revenue:

Our first task is to find the factors that positively and negatively affect the website's Revenue. Since **Revenue** is our **numerical** dependent variable (Y), we will use linear regression to model our data.

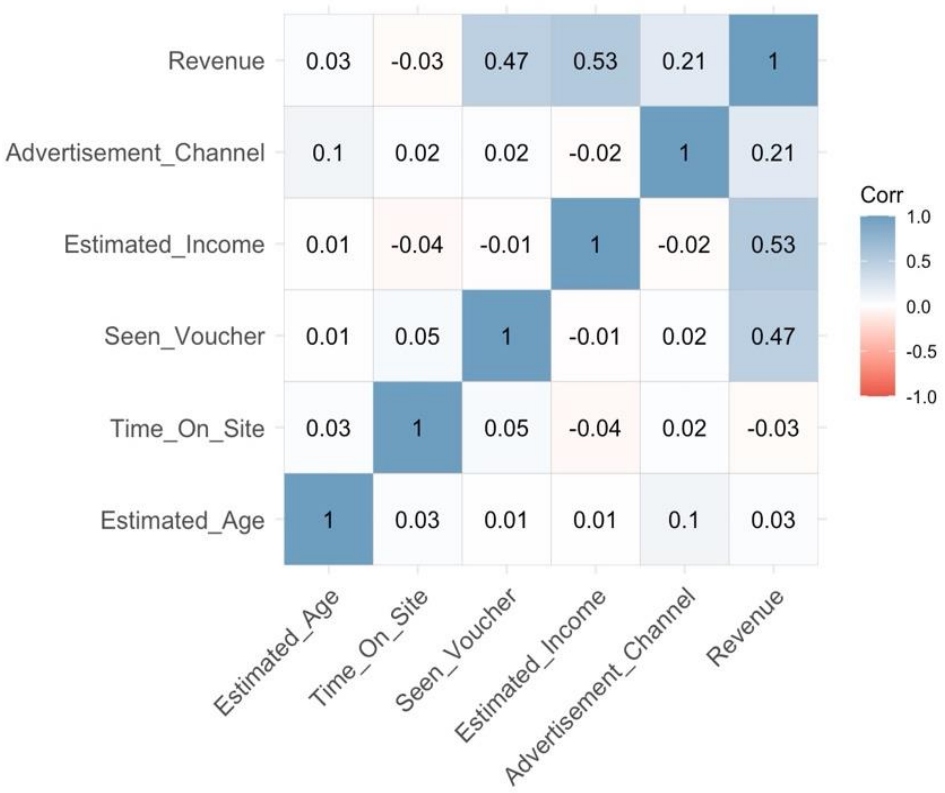
Data Preparation:

We first check for the missing values or NaN values in the data. From the data exploration, we found that we don't have any missing values in the data.

Data Modelling:

We check for the correlation among the variables using R. From Figure 2, we say there is a positive correlation for Revenue among **Seen Voucher**, **Estimated Income** and **Advertisement Channel**. But correlation does not mean causation. We further probe into checking for the individual Revenue across each positively correlated value.

Figure 2: Correlation Matrix



We will use a Boxplot with the Advertisement channel for the Seen Voucher and Revenue.

Figure 3: Boxplot - Seen Voucher and Revenue with Advertisement Channel

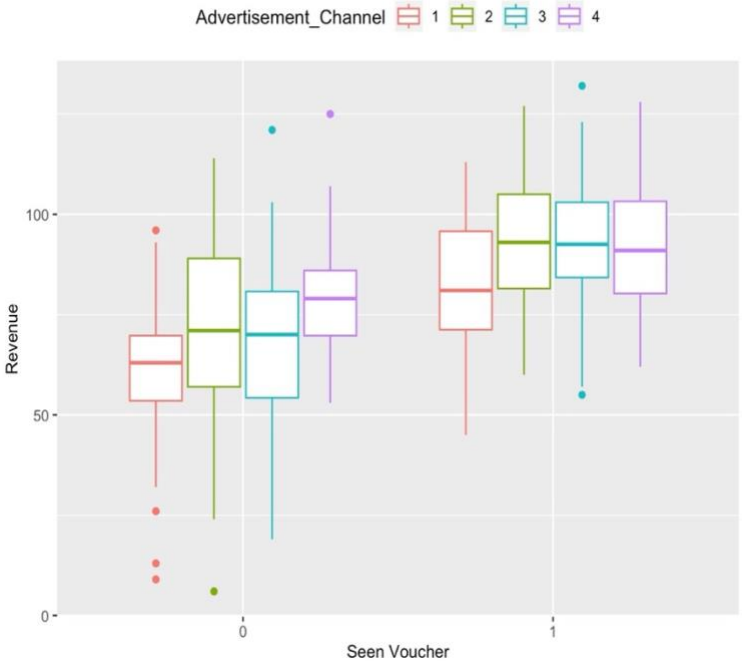


Figure 3 shows the boxplot of Seen Voucher to Revenue, where 0 denotes the customer did not see a discount voucher, and 1 denotes the customer saw a discount voucher. The colours represent the advertisement channel that brought the customers to the website.

Here, we can see that the **Revenue generated** is significantly **higher** among those who saw a voucher than those who did not.

We then take a scatter plot for the Estimated Income and Revenue with the Advertisement Channel to check for the spread of the graph.

Figure 4: Scatter Plot - Estimated Income and Revenue with Advertisement Channel

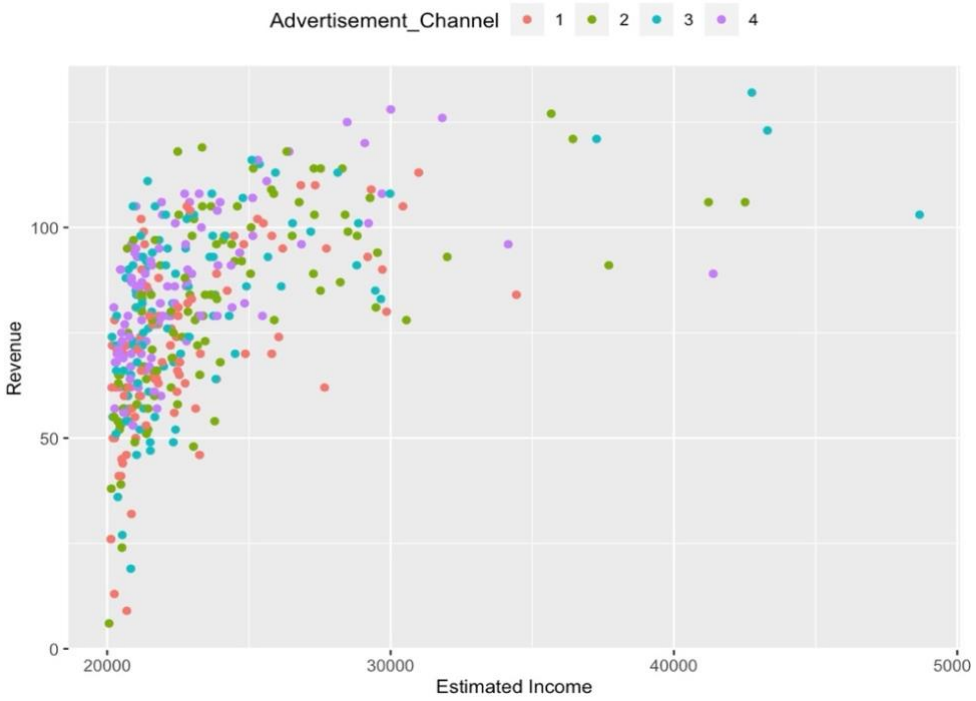
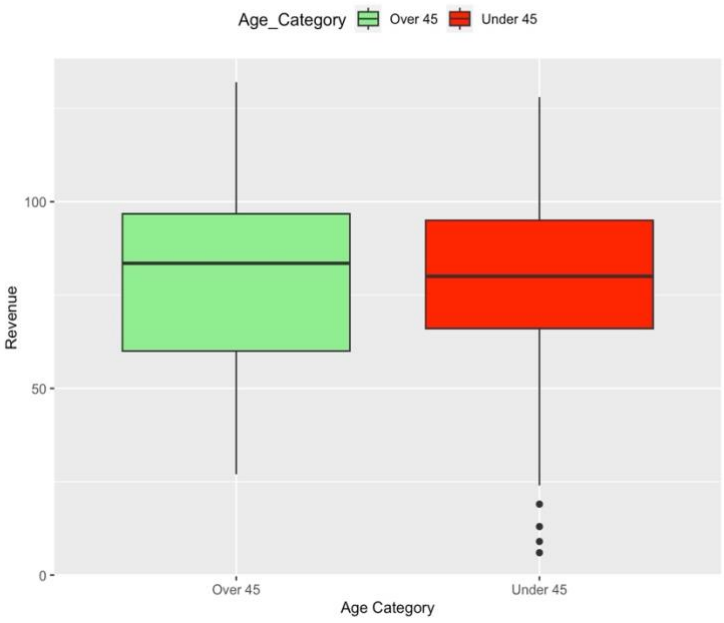


Figure 4 gives us the scatter plot for the Estimated Income and Revenue with the Advertisement Channel. Here, as the Income increases, the Revenue **slightly increases**. There is no apparent difference among the spread of various Advertisement Channels, but we can see many points from 2 – **Social Media** and 3 – **Search Engine** are on the higher Income side.

Since targeting customers over 45 years old is one of the decisions we should make, we will also see how it impacts the Revenue by splitting the data into two categories: over 45 years old and under 45 years old.

Figure 5: Boxplot of Revenue by Age Category



We plotted a bar plot in Figure 5, which shows the spread of the data (by age) to the Revenue generated on the website.

From Figure 5, the spread is almost similar for the customers over 45 and under 45 years old.
Thus, the customer's age does **not influence** the Revenue accrued on the website.

Thus, with the correlation and the plots, we infer that **Seen Voucher**, **Estimated Income** and **Advertisement Channel** have **positively** impacted the Revenue. In contrast, the time spent has minimal impact on the Revenue.

Goal 2 – Recommendation to Increase Profits:

Our second task is to recommend a method to increase profits. We will model a linear regression to find the best solution.

Data Preparation:

We use the *as.factor()* function in R or create a dummy variable in Excel for **Seen Voucher** and **Advertisement Channel** variables, as these are **categorical** data. We do this step to ensure that our model knows they are categorical variables and further improve the model.

Data Modelling:

We will use R's *lm()* function to model a linear regression. Revenue will be our dependent variable (Y) against the rest of the variables as our independent variables. We can then take the summary to see how well Revenue has a linear relationship among other variables.

Figure 6: Summary of the Linear Regression of Revenue as Dependent Variable

```
Call:
lm(formula = Revenue ~ ., data = dataplot)

Residuals:
    Min      1Q  Median      3Q     Max
-53.677  -7.657   1.429   8.967  40.283

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            0.3877205  6.3658721   0.061 0.951465
Estimated_Age         -0.0152422  0.0894058  -0.170 0.864718
Time_On_Site          -0.0221743  0.0219252  -1.011 0.312467
Seen_Voucher1         19.6954714  1.4145999  13.923  < 2e-16 ***
Estimated_Income       0.0028609  0.0001838  15.567  < 2e-16 ***
Advertisement_Channel2  6.8284251  2.0170930   3.385 0.000783 ***
Advertisement_Channel3  8.0909325  1.9997523   4.046 6.28e-05 ***
Advertisement_Channel4 12.9736091  2.0003277   6.486 2.66e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.09 on 392 degrees of freedom
Multiple R-squared:  0.5547,    Adjusted R-squared:  0.5467
F-statistic: 69.74 on 7 and 392 DF,  p-value: < 2.2e-16
```

Figure 6 shows us the linear regression coefficients against Revenue. With Multiple R-squared and Adjusted R-squared scores, the model fits **55%** of our data. This score is quite more than the average, and it is better score because it does not underfit or overfit our data.

Seen_Voucher_1 implies that given all the other variables are the same if a customer has **seen a voucher**, the Revenue accrued on the website is **£19.69** more than the customer that did not see a voucher.

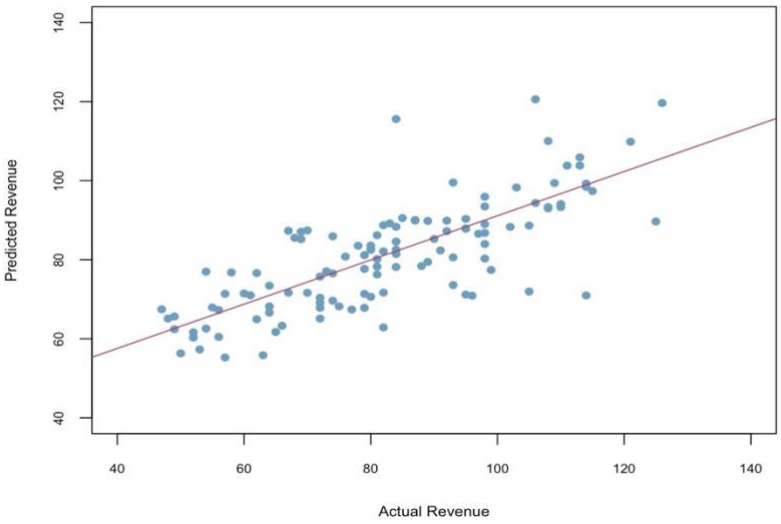
Also, Advertisement Channel 2 (Social Media) generates £6.82, Advertisement Channel 3 (Search Engine) generates £8.09, and Advertisement Channel 4 (Influencer) generates **£12.97** more Revenue than the Advertisement Channel 1 (Leaflet) if all the other variables are identical. This comparison shows that the **Influencer** gains more Revenue than others.

Thus, the Pr(>|t|) values and the significant codes also show that **Seen Voucher**, **Advertisement Channel** and **Estimated income** have significantly **higher importance** than the rest of the variables. At the same time, **Estimated Age** and **Time on Site** have **no significance** to the Revenue generated on the website.

We can also predict the Revenue by training and testing the dataset to evaluate the model's ability to predict any specific value in future projects correctly. We split the dataset into a 70:30 ratio, where 70% will be used to train the model and the rest 30% will be used to test the efficiency of the prediction. We use the prediction to find out how well our **linear regression** model **fits** the data so that we can make a **confident** decision to increase profits.

Figure 7 shows the graph of predicted values using linear regression and actual values from the dataset.

Figure 7: Actual vs Predicted Revenue



We can see that the points are closer to the diagonal line. The closer the points to the line, the better model it is. To put in exact numerical terms of its efficiency, we have **RMSE = 13.242**.

RMSE (Root Mean Squared Error) estimates how well the model can predict the target value. The **lower** the RMSE, the **better** the model is in accuracy. Thus, the RMSE value of 13.24 shows that **Linear Regression** is an **excellent** model for our task.

From our analyses, it is **recommended** that the company provides a voucher of **£20 off** their next order because this significantly increases the chances of a customer spending on the website. The second-best option is to advertise using an **Influencer**. We can see that age does not play a role in generating Revenue on the website; therefore, targeting customers **older than 45** should be **avoided**.

Conclusion:

Since our data is labelled, we used linear regression to know how each variable impacts the Revenue on the website. We see that the customer's income, the advertisement channel that brought him to the website and the possibility of seeing a voucher make the customer spend on the website. On the other hand, the customer's age has nothing to do with his spending habits. Finally, it is recommended that a voucher for **£20 off** their next order is provided to **existing** customers to make the website more profitable. It is also advisable to advertise using an **Influencer** to get **new** customers to the website. We do not have to target customers over 45 years as there is no difference in Revenue generated based on age.

Appendix:

The Excel workbook can be accessed here.

Part – 1

```
#Part 1
#Task 1 - Weighted Sum Method
#Install Packages
install.packages("./MCDA_0.1.0.tar.gz")
install.packages(c('Rglpk','triangle','plyr','ggplot2','glpkAPI','combinat'))
install.packages('MCDA')
library('MCDA')

#Data Preparation
dataWSM <- read.csv('/Users/Imthiyas/Library/CloudStorage/OneDrive-UniversityofLeeds/MSc Data Science & Analytics/Semester 1/Business Analytics & Decision Science/Courseworks/Robot_Info.csv')
# Insert new column by estimating Weights
dataWSM$Weight <- c(0.0667, 0.2, 0.1333, 0.2667, 0.3333)
performanceTable <- dataWSM
performanceTable

#Weighted Sum
performanceTable <- data.frame(t(performanceTable))
colnames(performanceTable) <- performanceTable[1,]
performanceTable <- performanceTable[2:5,]
performanceTable <- sapply(performanceTable,as.numeric)
performanceTable

#Criteria
weights <- dataWSM[,c(1,6)] # Retain only column that is related to alternative characteristic
weights <- data.frame(t(weights))
weights <- as.numeric(weights[2,])
weights

#Inversion - Min to Max
performanceTable[,4] <- performanceTable[,4]^-1
performanceTable

#Normalization
performanceTable <- performanceTable/colSums(performanceTable)[col(performanceTable)]
performanceTable

#Final Calculation
overall1 <- weightedSum(performanceTable, weights)
names(overall1) <- colnames(dataWSM[,c(2:5)])
overall1

#Visualization
#Figure 1
bp <- barplot(overall1, main="Figure 1: WSM Score for different Robot Prototypes", xlab="Robot Prototype", ylab ="WSM Score" , ylim = c(0, max(overall1) + 0.03), col=c("#ea5545","#edbf33","#bdcf32", "#27aeef"), cex.main=0.8, cex.lab=1, cex.axis=0.4, cex.names=1)
legend("topleft", legend=names(overall1), fill=c("#ea5545","#edbf33","#bdcf32", "#27aeef"), cex=0.5)
text(x = bp, y = overall1, label = round(overall1, 6), pos = 3, cex = 0.8, col = "black")
```

Part – 2

```
#Part 2
#Install Packages
install.packages("corrgram")
library(dplyr)
library(ggplot2)
library(corrgram)
library(readr)
install.packages("tidyverse")
library(tidyverse)

#Data Preparation
transactions_customer <- read_csv("/Users/Imthiyas/Library/CloudStorage/OneDrive-
UniversityofLeeds/MSc Data Science & Analytics/Semester 1/Business Analytics & Decision
Science/Courseworks/Transactions_Customer.csv")
options(width = 700) #Adding more columns to the output
head(transactions_customer) #No need of dummification as all the variables are numeric

#Data Exploration - Looking for Missing Values
# Check if there are any null values in the dataframe
any_null_values <- any(is.na(transactions_customer))
# Print the result
print(any_null_values)

#Task 1 - Exploring the data for positive and negative impacts on Revenue (Target Variable)
cor(transactions_customer)
corrgram(transactions_customer)

#Making a better Correlation Matrix Plot
# Install and load the ggcorrplot package
install.packages("ggcorrplot")
library(ggcorrplot)
# Calculate correlation matrix
correlation_matrix <- cor(transactions_customer)
# Create a correlation plot with correlation coefficients
#Figure 2
ggcorrplot(correlation_matrix, lab = TRUE, colors = c("#ea5545", "white", "#6D9EC1")) +
ggtitle("Figure 2: Correlation Matrix") +
theme(plot.title = element_text(hjust = 0.5, size = 15))

#Seen Voucher, Estimated Income, Advertisement Channel
#Assumption 1: Linear relationship between the variables
dataplot <- transactions_customer
dataplot$Seen_Voucher <- as.factor(dataplot$Seen_Voucher)
dataplot$Advertisement_Channel <- as.factor(dataplot$Advertisement_Channel)

#Figure 3 for Boxplot for Seen Vouchers & Advertisement Channel have Variance
ggplot(data=dataplot) + geom_boxplot((aes(Seen_Voucher,Revenue,color=Advertisement_Channel)))+
labs(title = "Figure 3: Boxplot - Seen Voucher and Revenue with Advertisement Channel",,x = "Seen
Voucher",y = "Revenue") +
theme(plot.title = element_text(hjust = 0.5, size = 12), axis.title.x = element_text(size = 10), axis.title.y =
element_text(size = 10), legend.position = "top")
#Seen Voucher gets more revenue across.
ggplot(data=dataplot) + geom_boxplot((aes(Seen_Voucher,Revenue)))+
labs(title = "Seen Voucher and Revenue with Advertisement Channel", x = "Seen Voucher",y = "Revenue")


#Figure 4 for Scatterplot for Estimated Income have Variance
ggplot(data=dataplot) + geom_point(aes(Estimated_Income,Revenue,color=Advertisement_Channel))+
```

```
labs(title = "Figure 4: Scatter Plot - Estimated Income and Revenue with Advertisement Channel",x =
"Estimated Income",y = "Revenue") +
theme(plot.title = element_text(hjust = 0.5, size = 12), axis.title.x = element_text(size = 10), axis.title.y =
element_text(size = 10), legend.position = "top")

# Create a new variable for age category
dataplot$Age_Category <- ifelse(dataplot$Estimated_Age >= 45, "Over 45", "Under 45")
#Figure 5 for the boxplot
ggplot(dataplot, aes(x = Age_Category, y = Revenue, fill = Age_Category)) +
  geom_boxplot() +
  labs(title = "Figure 5: Boxplot of Revenue by Age Category",
       x = "Age Category",
       y = "Revenue") +
  scale_fill_manual(values = c("Over 45" = "lightgreen", "Under 45" = "red")) +
  theme(plot.title = element_text(hjust = 0.5, size = 12), axis.title.x = element_text(size = 10), axis.title.y =
element_text(size = 10), legend.position = "top")

#Task - 2: Fitting a linear regression model to predict Revenue
#Fitting Model: This is for better way of advertising
dataplot$Seen_Voucher <- as.factor(dataplot$Seen_Voucher)
dataplot$Advertisement_Channel <- as.factor(dataplot$Advertisement_Channel)
model <- lm(Revenue ~ ., data=dataplot)
summary(model)

#Predicted vs Actual Revenue
set.seed(123) #Using the same seed to get the same random numbers
datasplit <- transactions_customer
datasplit$id <- 1:nrow(datasplit)
dim(datasplit)
trainingdata <- datasplit %>% sample_n(40*7)
testdata <- anti_join(datasplit, trainingdata, by = 'id')
print(dim(trainingdata))
dim(testdata)
modeltrainsplit <- lm(Revenue ~ ., data=trainingdata)
prediction <- predict(modeltrainsplit, newdata = testdata)
sqrt(mean((testdata$Revenue - prediction)^2)) #RMSE = 13.2412

# Create a scatterplot
plot(testdata$Revenue, prediction, xlim = c(40,140) ,ylim = c(40,140),xlab = "Actual Revenue", ylab =
"Predicted Revenue", main = "Figure 7: Actual vs Predicted Revenue", col ="#6D9EC1", pch=16,
cex.main=1, cex.lab=0.8, cex.axis=0.7)
# Add a line to the plot
abline(lm(prediction ~ testdata$Revenue), col = "maroon")
```